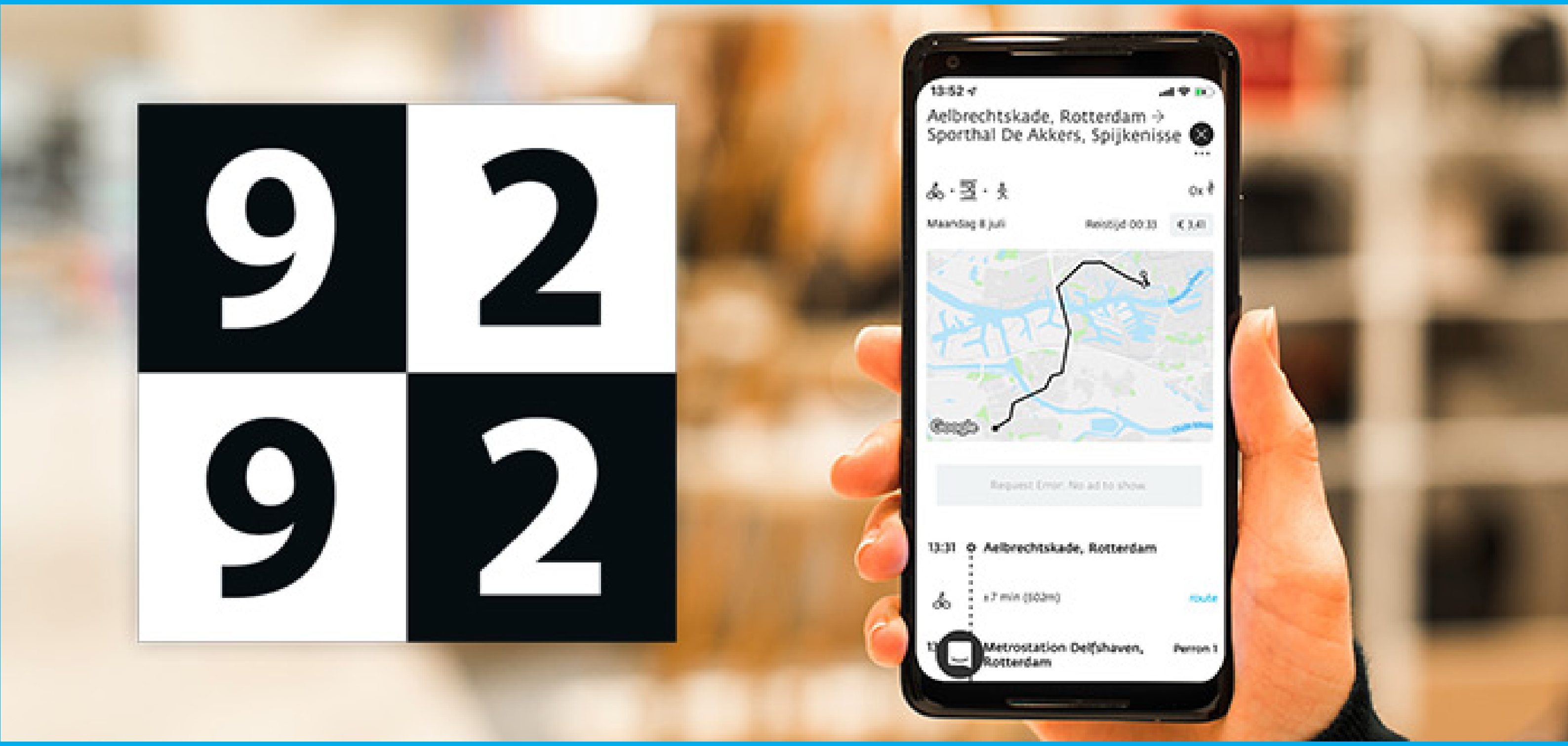
Predicting Short-Term Bus Ridership with Trip Planner Data

A Machine Learning Approach

Ziyulong Wang


9 2
9 2
13:52
Aelbrechtskade, Rotterdam → Sporthal De Akkers, Spijkenisse
Maandag 8 juli
Reistijd 00:33
€ 3,41
Request Error: No ad to show
13:31 Aelbrechtskade, Rotterdam
±7 min (602m)
route
Metrostation Delfshaven, Rotterdam
Perron 1

TUDelft

Predicting Short-Term Bus Ridership with Trip Planner Data

A Machine Learning Approach

by

Ziyulong Wang

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4890973	
Date:	August 18, 2020	
Thesis committee:	Dr. ir. N. Van Oort,	TU Delft, chair
	Dr. ir. A. J. Pel,	TU Delft, supervisor
	Dr. ir. T. Verma,	TU Delft, supervisor
	Dr. ir. P. Krishnakumari,	TU Delft, supervisor
	P. Van Brakel,	9292 | REISinformatiegroep BV

Preface

This thesis is written in a very difficult and uncertain time during the infamous and notorious COVID-19 pandemic when our daily lives have changed dramatically. Millions of lives with their families have been affected and no one knows whether the current measurement will become a new normal…

Thankfully, I never fell along and abandoned during this extremely hard time and I would like to take this opportunity to express my gratitude to all those who have accompanied this journey with me. I would not make this research without the vital roles played by many people who encouraged, inspired and supported me.

What wisdom can you find that is greater than kindness? I would like to thank my Daily Supervisor Dr.ir. A.J. (Adam) Pel, for your amazing knowledge, for your creative mind, and for your valuable guidance and enthusiasm through all the stages of my study. I still remember a joke you made in the lecture of Advanced Transport Modelling when you said "Speaking of our assignment, some have already finished while some have just started" by almost the end of the course. You always provided me critical suggestions for improvements and answered all questions that I can not figure out by myself, even would like to help to connect me to researchers and experts from academia.

Your network is your net worth. Great thanks to Dr. Ir. N. (Niels) van Oort who provided me this excellent opportunity for shaping me to think as a researcher and offered me significantly contributing advice to this study. Your invaluable feedback and networking always made my research more practically applicable and challenged me to rethink and to be critical. Public Transport is your life-long pursing filed and you affect me to be involved as well.

Fairness means everyone gets what they need and to be the one they like to be. This is a saying I came across that is similar to your motto, Dr.ir. T. (Trivik) Verma. I would like to thank you for providing priceless suggestions and feedback from an urban scientist and policy support perspective. I really appreciate that as an external supervisor (twice) for your involvement and welcome me to your lab CUSP. I will always think twice about my audience and the aim when I make visualizations and all things I may encounter further.

Practice makes perfect. Indeed, I would like to thank Dr.ir. P. (Panchamy) Krishnakumari for your insightful feedback and suggestions and your technical help along with my study, for your invaluable advice concerning all my arrangements and your practical experience. I would like to thank my Company Supervisor Peter van Brakel for your active involvement in this research, and for helping me bridge all the colleagues that could answer my questions, and for all the efforts to help me blend in the big family Reisinformatiegroep B.V.(9292). Great gratitude to this marvelous company and all the staff and also data scientists from Lynxx. Moreover, big thanks to OV-bureau Groningen Drenthe and Translink for providing the precious smart card data to me.

Time spent with family and friends is worth every second. Special thanks to my family, my mother Guang Long, my father Lei Wang, my aunt Shaofang Wang and Lingyun Long for your countless and constant love and support without whom I could not be able to make this journey. I would like to thank all my friends (George Weijs, Darshik Parejiya, Chen-Yen Chou, Qingyuan Zhu, etc.) and football teammates who have made this journey more meaningful and enjoyable. Lastly, I would like to thank my girlfriend Xinyan Zhao for your sweet love and warm accompany.

Ziyulong Wang
Delft, August 2020

Contents

List of Figures

List of Tables

List of Acronyms and Glossaries

AFC Automatic Fare Collection System

ANN Artificial Neural Network

APC Automatic Passenger Counting System

APPs Applications

AR Autoregressive

ARIMA Autoregressive Integrated Moving Average

AVL Automatic Vehicle Location System

BI Business Intelligence

CA Cluster Analysis

CDR Call Detail Record

CML Composite Marginal Likelihood

EA Empirical Analysis

EM Elasticity Model

GBDT Gradient Boosting Decision Tree

GBM Gradient Boosting Machine

GBRT Gradient Boosting Regression Tree

GPS Global Positioning System

GSM Global System for Mobile Communications

IMM Interactive Multiple Model

LR Linear Regression

MA Moving Average

MAE Mean Absolute Error

MDI Mean Decrease in Impurity

ML Machine Learning

MPD Mobile Phone Data

MPE Mean Percentage Error

NLM Nested Logit Model

OD Origin-Destination

ORP Ordered Response Probit

OV Public Transport in Dutch

PCA Principal Component Analysis

PDP Partial Dependence Plot

PT Public Transport

RBF Radial Basis Functions

RF Random Forest

RFR Random Forest Regression

RMSE Rooted Mean Square Error

RTI Real-time Transit Information

SLEF Sliding Window Ensemble Framework

SVM Support Vector Machine

SVMOM Support Vector Machine Online Model

SVR Support Vector Regression

VRU Voice Response Unit

WTVP Weighted Time Varying Poisson Model

Desired time of a travel advice The time when a passenger desires to depart of a trip

Requested time of a travel advice The time when a passenger sends a travel request to trip planner

Answer from Trip Planner The recording(s) of a trip advice for passengers, multiple legs if applicable

Question from Trip Planner The recording of a trip request from passengers

Vehicle Start Time from Trip Planner The arrival time of a bus trip at the origin from a trip planner request

Timing Advance The time difference between the vehicle start time and the request travel time from the trip planner, if not specified

Executive Summary

To address the increasing passenger demand in the coming years and make public transport less crowded and delayed, insights into predicted passenger flow are needed. A wide range of studies has used and validated that smart card data can be one of the sound bases for predicting short-term passenger demand (Ding et al., 2016, Van Oort et al., 2015a, Zhou et al., 2016). However, it also has several disadvantages, such as the relatively long collection time, the insufficiency to reflect the relationship between passenger behavior and ridership. Trip planner data, which emerged as a type of real-time transit information, could reduce the perceived waiting time of passengers and increase the transit ridership due to the improved satisfaction. Combining these two types of data could potentially cater to the interest of operators in matching the vehicle supply and passenger flow demand at an operational level.

Background and Research Objectives

The proliferation of one kind of trip planner application has significantly enhanced the mobility options for the public. It is essentially a multi-modal trip advice app that offers different choices for the details from origin to destination. It can improve user travel experience through static (timetable, fare, map, etc.) and real-time (delay, re-route, crowdedness, etc.) information. Correspondingly, the aggregated information also help public transport operators to understand their service performance more and lead to their goals.

Nevertheless, hitherto very limited researches have been carried out to combine smart card data and trip planner data for predicting short-term bus ridership to the best of our knowledge. Moreover, it is still unclear to what extend the trip planner data would help in such a prediction and what is the effective method to perform. Hence, the initiative of this study is built upon the ever-increasing provision of real-time travel information, namely the trip planner data. The reasons are bifold: first, it is common that people schedule their trips before they realize so that the dynamic future passenger flow can be revealed by using this kind of information; second, to combine smart card data and trip planner data is promising as they share the same level of spatial and temporal information, i.e. they can provide accurate information on the origin and destination of a trip at a specific time point.

Another drive of this study is to offer public transport operators a suitable method of short-term ridership estimation. Currently, they schedule most buses a week in advance, and therefore it possibly neglects the changes and results in some negative impacts on the passengers. Not only a reliable short-term prediction model can help to efficiently schedule the fleet with adequate ability to avoid crowdedness, delay, or bunching, but also it can cope with demand fluctuations during abruptions, weather changes, events, etc. Additionally, it also benefits public transport operators to select the appropriate bus size to cater to the demand as they usually have more than one type of bus to serve different scales of operation. Correspondingly, the quality of service would increase and thus a higher passenger satisfaction.

Notwithstanding, the abundance of available data challenges the traditional data mining methods. But the emergence of machine learning with its efficiency, automation, and effectiveness facilitates the fulfillment of this research gap. Thus, we conclude our objective is to derive the relationship between trip planner data and the ridership of bus trips by applying machine learning algorithms to see if we can conduct the short-term prediction with an acceptable precision at stop-level. We present the formulated research questions below, beginning with the main question:

To what extent can trip planner data contribute to short-term bus ridership prediction and what are the important influencing factors in trip planner data and other data in such a prediction model?

This main research question is subdivided into five sub-questions in this study. We first explore the existing literature to find out what are the state-of-the-art short-term ridership prediction models and influencing factors that internally from trip planner data and externally from other data affect the short-term ridership prediction, with respect to spatial, temporal, and other characteristics. Then, we analyze the trip planner data to unveil the dimensions that have been stored and the short-term prediction models to unravel what parameters and variables stand. Next, we establish the connection between those dimensions and unfold the correlation between observed ridership from smart card data and trip planner data. Following, we leverage this correlation for predicting the short-term bus ridership. Finally, we examine and determine the performance and benefits of this model.

Research Methodology

In general, this study leverages the regression problem to predict the short-term bus ridership and we select supervised learning to accomplish this task. The regression task approximates a mapping function from input variables to the continuous output variable(s). We iteratively explore the raw data in order to create a well-knitted set of variables (features).

To begin with, we first describe, clean, and integrate three datasets that we have used in this study, including trip planner data, smart card data, and automatic vehicle location data. We investigate the temporal and spatial dimensions and discover the correlation matrices and variance-covariance matrices to find the contributing variables.

Then, we choose the machine learning methods with the emphasis on interpretability in both configuration and results. By results, we mean that we can derive the feature importance. The methods include *Linear Regression*, *K-Nearest Neighbour Regression*, *Gradient Boosting Decision Tree Regression*, and *Random Forest Regression*. For several machine learning models among them, we need to tune a few parameters to optimize their performances. Thus, we divide the whole dataset into a training set and a test set. Following, we apply nested cross-validation in which we fine-tune the parameters in the inner-loop by k-fold cross-validation. And we compute the robustness and accuracy of the models via outer-loop by random permutation cross-validation. Moreover, we create a comparison between machine learning models and the baseline models (public transport current model and public transport current with seasonal trend). The outperformed model is used to examine the feature importance.

After data analysis, we find out that both smart card data and trip planner data are imbalanced. The imbalance is not inherently a problem. However, we have a non-uniform preference across the ridership domain, and the most important ranges are poorly represented, which is the crowded cases. Therefore, we employ the sampling design to tackle this imbalanced data. We first take the natural logarithm of the ridership to smooth the highly right-skewed distribution, and it is in line with the regression models that work better with more symmetrical, bell-shaped distributions. Next, this transformation facilitates the binning of continuous data into discrete one as there is less data on the right tail of the distribution. The binning strategy with Doane's formula helps us to resample the data by SMOTE (synthetic minority oversampling technique) that enhances the influence and representations of the minority but interested classes. We also practice a sensitivity analysis to determine the optimal sample designs.

Finally, we set up a set of performance metrics that consist of MAE, $RMSE$, and R^2 to compare the multiple models. They measure the absolute error, the spread of the error, and goodness-of-fit, respectively. Additionally, the main objective of the study is to discover the usefulness and utility of trip planner data. Only get to know the performance of the model is not enough. Thus, we introduce feature importance to the study, and we select the best model out of the omnibus to explore the feature importance. For different models, the way of computing feature importance is dissimilar. The permutation feature importance is comparably convincing. For the tree-based models, we also add the mean decrease in impurity to measure the feature importance if they outperform. Lastly, we implement the partial dependency plot to assess the influence of a single feature on the average prediction of the model.

Data

We utilize the data provided jointly by 9292 and OV-bureau Groningen and Drenthe in October 2019 to validate our methodology. The case study area is suitable for researching with the feature of bus-oriented, youth-prone, and density-divers. The bus line types are various, including inter-city, inner-city, and rural. It could potentially offer insights into the comparison for the prediction of different line types.

Results

First of all, *Random Forest Regression* outperforms than the other models with a landslide in the majority of the cases (3 cases out of 4). Only one line with a different sample design that gives us a different result, originated from its spares distribution at the less representative domain. In this particular case, *K-Nearest Neighbour Regression* is the best model based on the criteria of MAE, $RMSE$, and R^2, however, we have a better R^2 from the cross-validation by applying *Random Forest Regression*. Due to the sample design varies in this specific case, we consider it is much more robust to adopt *Random Forest Regression* as it is less sensitive to training data.

Second, we find out that the current public transport model performs worse due to the missing recordings of bus trips. If operators can come up with a better way to deal with this problem (other than filling 0 or drop the recordings in this study), it could potentially elevate the performance of this model. It is indeed efficient and effective when the ridership is low. The simple model with the weekly trend fails in this study as it exaggerates the ridership when the same trip that was too crowded last week on several specific sections. A future step could be adding smoother, such as the weekly trend coefficient of the last trip or the last two trips.

Then, we explore the results from different perspectives. The actual vs. prediction plots tell us that machine learning models generally can not only better capture the quiet trips but also outperform than the baseline models when the trips become busy, regardless of the line. However, all models have larger average error and bias when the actual value becomes larger, which implies the existence of heteroscedasticity. When the actual value is low, all models can function well while *Gradient Boosting Decision Tree Regression* tends to overestimate, *K-Nearest Neighbour Regression* tends to underestimate while *Random Forest Regression* is relatively neural.

Next, we analyze the residual plots. Residuals are differences between the predictions from the model and the measured outputs from the validations dataset. From the residual plots, we once again testify the existence of heteroscedastic. It is not inherently a problem but implies that the model can be improved. As we have already transformed the target, it indicates the possible inclusion of other significant contributing factors could solve the issue. Except for the case that has a different sample design, all other cases show a symmetrical and balanced distribution of residuals around zero, which means a balance in the estimation. And this specific case has the problem of the y-axis unbalanced slightly due to the larger oversampling design.

Besides, these prediction residuals are not entirely similar distributed across the cases, in terms of periods. In most cases, it is the evening peak that has the highest variance because it is the second-highest commuting time. People have a non-uniform off-duty time so that the ridership is easy to fluctuate, and therefore a higher variance of prediction error. In contrast, one line out of four has a higher variance of residuals during the morning peak because it has a comparatively high commuting passenger flow during the morning peak due to its line characteristics. It results in higher variance and predictive difficulty. Concerning day type, the variance of the weekday is higher than the weekends in every case.

Particularly, *Random Forest Regression* models roughly reach a balanced estimate. However, it is another time the case that has a large oversampling strategy (by enlarging the minorities with 250%) that has the most difference in biased prediction. The margin between overestimation and underestimation is as high as 30%. Even though we did not undersample, but only oversample the minority classes by 5%. The model still tends to underestimate the actual values, and this tendency is much more profound when we see the average of the top 5 percentile residuals.

With the outperformed model Random Forest Regression, we uncover the importance of variables (features). Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction. The most important feature is always the average number of ridership, despite the methods in this study. Regarding trip planner data related variables, the sum of the number of trip planner requests, the average of requests, and the variance of requests can take approximately 20% of the importance in the ridership prediction on average. When the number of requests is low, there is a strong positive correlation between ridership and request, between ridership and average request, and between ridership and variance of the requests as revealed by the partial dependency plot. However, when it is the high-value domain, the effect is marginal because of the fewer recordings so that the model can not learn a meaningful prediction.

Furthermore, we investigate the performance of the *Random Forest Regression* model with trip planner data that is further ahead in time. We examine the scenario with all requests, requests that are

10, 15, 30 minutes ahead with the same configuration of the model, and the same sampling design. This timing advance is measured by the requested travel time and the vehicle start time, namely the margin between the time forward a travel request and the time that passengers potentially realize the trip. The performance does not deteriorate to a great extent. Two of the four cases show us that model with requests that are 10-minute ahead outperforms, indicating that the information stores nearest are more valuable.

Lastly, we desire to unveil how the trip planner information could function during different times, and during what period, the trip planner data is of the most usefulness. The reason is that people behave differently during different times by using such a trip planner from the data analysis. We find out that three cases out of four have the best performance with requests from 10:00 to 16:00, compared to all requests included. Users send very few requests at night (from 22:00 to 4:00), which is around 5% of all requests. But one of the cases gives the best performance by utilizing requests from the night. In notable contrast, the other one gives very unsatisfied results. Therefore, we conclude that the case in which we have a different sample design can learn a meaningful result when the sample is large enough. In other cases, it is during the evening (from 16:00 to 22:00) when *Random Forest Regression* model captures less.

Conclusions and Recommendations

This research provides scientific and practical contributions respecting ridership prediction with trip planner data. It fills the research gap where hitherto there is still no study using such kind of data in short-term ridership prediction with machine learning methods. Furthermore, we construct more interpretable models so that we can analyze the feature importance with the emphasis on the trip planner data. We also utilize the data with a certain time ahead to validate the usefulness of trip planner data in predicting the short-term bus ridership. The developed method shows that *Random Forest Regression* is approximately two times better than the public transport current model with a MAE of around 3 at a stop-level on average. Moreover, it also presents that *Random Forest Regression* has a goodness-of-fit (R^2) from repeated random 5-fold cross-validation at almost 0.8 on average. By stop-level, we mean at a specific stop during a trip. Lastly, the model does not deteriorate to a large extent when it only works requests with a certain lead time.

We list the main contributions of this research as following:

- It is novel to incorporate trip planner data in short-term ridership prediction, however, solely based on this kind of data would be inaccurate. It is useful to combine the trip planner data and the historical ridership, derived from the smart card, to realize the ridership prediction. In this way, we can avoid the long collection time of smart card data and able to capture the temporal and spatial influence such that we can fulfill the ridership prediction at an operational level.
- By sampling design, the less represented data domain can be pronounced but with a necessary compromise of the overall performance of the model.
- Regardless of the line, machine learning models generally capture not only the quite trips but also the busy trips.
- Random Forest Regression among all six models that we have selected in this study outperforms than others. It has the highest goodness-of-fit from repeated random 5-fold cross-validation of every case study line and the best performance of absolute error, the spread of error in three of the four.
- The Random Forest Regression model roughly reaches a balanced estimation, i.e. no tendency towards over- or underestimation.
- The Random Forest Regression model does not deteriorate to a large extent when we only utilize trip planner requests with a short time ahead. For the majority of the cases, it also does not degrade with the requests sent from a specific period.
- The request-related features (variables) can take up 20% of the importance in short-term ridership prediction in this study, including the number of requests, the average number of requests, and the variance. This means that it is useful and rational to take trip planner information in ridership prediction in the future works.

We recommend a high-level collaboration among 9292 (trip planner company), public transport operators, and the authorities. So that using smart card data as a sound basis and adding trip planner data to predict the ridership can avoid the long collection time of the smart card data and also capture the passenger behavior. This collaboration also benefits the cleaning and merging of datasets, including stop names, route names, etc. Besides, it is beneficial that 9292 could store all provided choices such that we can dive into the analysis of passenger behavior and reduce the overestimation. The logged alternatives and the actual choices are similar to a huge questionnaire. Benefitting from it, we can know how people trade-off between the fastest route or the cheapest route, how people select between the routes that traversed specific regions or districts, how people choose the exact modality, and do they have a preference. Those questions can be interesting, and also the results would help to personalize the service and maximize the profit. Finally, if user type and IP tracing are applicable, we can reduce the overestimation and distinguish the travel pattern in a user-wise.

1 Introduction

This chapter begins with the background information of the context and the relevant research at current. This chapter also provides the background and research motivation, including the reveal of the research gap and the contribution of the application. Thereafter, we present the research objective with the scope to fill in the revealed research gap. Finally, we propose the research questions.

1.1. Background and Research Motivation

To pursue livability and mobility, one of the public transport (PT) goals of the government of the Netherlands is traveling in comfort by PT without hassles. However, with the anticipated 30% to 40% growth of passenger demand in the coming twenty years, the PT network is approaching overload, bringing the challenge of matching the transport supply and increasing travel demand (Ministry of Infrastructure and Water Management, 2019). The underlying mismatch could not only lead to extended travel time and decreased comfort due to crowdedness and delay in the short-term, but also result in a modal shift in the long-term (Pel et al., 2014, Van Oort et al., 2012). This will run in the opposite direction of the goal to make PT a viable and comfortable alternative to driving.

Fortunately, in an era with massive data resources and computing tools, solving the problem of matching the imbalance is not entirely impossible. Researchers like Van Oort et al. (2015b) expounded that in order to design more optimal PT networks, timetables and develop more reasonable operation strategies, insights into predicted passenger flow is needed. Many other similar studies have reasoned that reliable and effective ridership prediction is beneficial for both passengers and transit operators (Ding et al., 2016, Ohler. et al., 2017, Van Oort et al., 2015). With the predicted passenger demand information, public transit operators could allocate sufficient but not superfluous rolling stocks and inform passengers in advance or optimize their timetable tactically to avoid bus delay or bunching, which results in more positive travel conditions for passengers. From the perspective of passengers, people would like to be informed in advance about expected arrival and accurate traffic information to help them with adjusting their mode or departure time choice to avoid the crowdedness. By predicting the future passenger flow, it is able to reach a reliable balance between passenger travel demand and PT supply. Accordingly, this will improve the PT service and potentially attract more ridership. Hence, many researchers have studied the topic of demand prediction with different types of methodologies.

Conventionally, a simple 4-step model will be deployed to predict long-term transit ridership, including trip generation, trip distribution, modal split, and trip assignment (Horowitz, 1984). The typical prediction method in trip generation is to construct a linear or non-linear model between passenger demand and contributing factors such as demographics, socio-economic factors, transit attributes, geoinformation, etc (Chan and Miranda-Moreno, 2013, Idris et al., 2015, Taylor et al., 2009). The advantages of these types of models are that it is easy to be fulfilled by spreadsheets with simple rules and it only focuses on long-term ridership prediction served for strategic and tactical level through analyzing the pattern of travel demand. Yet, it is not able to represent multi-modality and its level of detail does not always match the level of operation in the PT company. Most of the PT operators still do not have a transport model that could provide valuable insights into the operational level (Van Oort et al., 2015a). Thus, researchers began to unveil the possibilities from new massive data sources such as smart card data, Global Positioning System (GPS) and Mobile Phone Data (MPD) to forecast the passenger demand.

In the field of PT, the above-mentioned new data can be retrieved from Automatic Fare Collection System (AFC), Automatic Vehicle Location System (AVL), Automatic Passenger Counting System (APC) and all mobile related devices and applications(Van Oort et al., 2015). Van Oort et al. (2015) show that combining the data from APC and AVL can facilitate the study into bottleneck identification, service reliability investigation, which can be used to measure the performance of the public transit system and accordingly elevate it. In another study, smart card data from AFC and an elasticity model are combined to develop a short-term prediction model (Van Oort et al., 2015a). In this way, we can observe the current demand by smart card data as origin-destination (OD) matrix and it is a sound basis for the short-term demand prediction, which leads to a substantial improvement on the PT level of service. However, the disadvantage of using smart card data is that it is hard to execute on an hourly basis or shorter because the collection of smart card data needs time up to days. It needs to be transmitted from vehicles to transit operators and then to the transportation authorities and therefore it is hard to predict the ridership real-timely or in a shorter term. Additionally, only leveraging smart card data to predict ridership is insufficient to reflect the relationship between passenger behavior and ridership.

MPD, more than just containing the position of users but also their dynamic mobility information, is another popular discussed data source and is able to help operators gain an insight into the planning stage (Elias et al., 2016). Generate the data out of the cellular network and convert it possible to represent the transport network that can be beneficiary. MPD generally can provide various types of data, such as Call Detail Record (CDR), Global System for Mobile Communications (GSM), Voice Response Unit (VRU), etc. De Regt et al. (2017) fused GSM data with smartcard data to reveal the spatial and temporal pattern and it is able to offer insightful mobility patterns from strategical and tactical level but since it is not real-time data, limited value is provided at an operational level. Whereas the social media data or data following from smartphone applications (apps) often offer those real-time data and those apps have emerged as tools for gathering and aggregating mobility information. Shaheen et al. (2017) reasoned that understanding the role of apps in the field of mobility is important for planning and development. For instance, social media apps data can grant knowledge on the usage of public transport (Bregman, 2012). Sharing mobility apps, such as car-sharing (Uber, Lyft, etc.) and bike-sharing (Mobike, OV fiets, etc.) can be used to estimate and predict the ridership in order to reveal the demand pattern and to match the demand and supply better (Li et al., 2015c, Vogel et al., 2011, Xu et al., 2018). These apps may vary one from another but all with the basic provision of the mobility information about travels to its users. However, limited studies have been conducted in the field of PT, which indicates a lack of knowledge on incorporating real-time apps data to understand the demand pattern and its correlation with ridership at an operational level.

Trip planner apps, as one of the important PT mobility apps, is basically a multi-modal trip advice application, providing the information from origin to destination with various options. These apps aggregate knowledge from PT, walking, cycling, and other modes. Furthermore, these apps improve the mobility of users by static (timetable, fare, map and so forth) and real-time (delay, re-route, crowdedness and so on) information. From the user side, it is convenient to use these trip advice to make travel decisions like the least travel time, the least travel cost and the like. From the operator side, the aggregated trips can help them understand the performance of their service from diverse angles and can lead to their goals. As indicated by the previous studies, real-time transit information (RTI) apps could reduce the waiting time of passengers, the overall travel time due to route choice and increase the ridership of transit as increased satisfaction as a result of overall transit service and perceived personal security (Brakewood and Watkins, 2019). Van Roosmalen (2019) attempted to predict the short-term bus ridership by the usage of this type of data and several machine learning (ML) algorithms, but the results can still be improved considerably and more selected features and methods can be discussed.

Given these points, the PT industry shows substantial interests in matching the vehicle supply and passenger flow demand via short-term ridership prediction at an operational level. However, current information on the interaction between PT usage and the future demand is provided by solely smart card data and other demographic information. It can capture the future flow to help at strategic and tactical levels but it is hard to predict ridership in short-term or real-time and it is neglecting the passenger behaviors as above-mentioned, indicating the necessary incorporation of the information of real-time dynamic demand. Furthermore, very limited studies have been conducted in the integration of smart

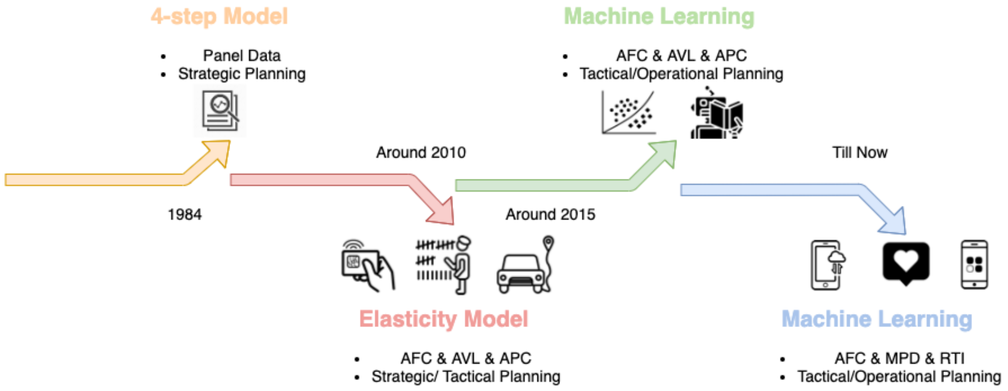


Figure 1.1: A brief overview of the timeline in ridership prediction methods and data

card data and trip planner data to the best of our knowledge and we do not know how the trip planner data could advantageously help to forecast ridership in short term and what is the suitable method to be performed yet. With the increasing provision of real-time information to passengers by industry practice, the dynamic future passenger flow can also be revealed by trip planner apps as it is common that people plan their trips before acting. Those two types of data share the same level of spatial and temporal information, which means they are able to provide detailed information on the origin and destination of a trip at an exact timestamp. This shows the possibilities of the fusion will lead to information on the interaction of PT supply and overall demand, including temporal and spatial dynamics. The dynamics will enable PT operators to allocate enough vehicles for specific routes during desired periods.

In a nutshell, if it is able to predict the ridership based on the requests with certain accuracy within an acceptable organizing time, this will benefit both PT operators and passengers. Plus, PT operators presently schedule most buses a week in advance (Van Roosmalen, 2019). This will result in some negative impact on the passengers. For instance, “Students at the expense of the holiday timetable, Kas (14) from Zelhem comes to school late every day” because the PT operator made the timetable based on the schedule of school last year (Van Sloten and Van Rooijen, 2019). Another example, “Busy buses to Deltion Zwolle: Jos (18) has to wait for seven full buses” because of overwhelming travels during peak hours without sufficient vehicles (Frasa, 2018). This indicates the deficit of efficiency and dynamics. An appropriate short-term ridership prediction method would help to plan the fleet dynamically with sufficient capacity to avoid crowdedness and the bus delay or bunching that it may incur. Moreover, it is able to cope with the fluctuations of demand during sudden change of weather, events, etc. Furthermore, PT operators normally have more than one type of bus to server different operation scale and some rural dedicated lines to help residents access to major cities. When the predicted demand is fewer, smaller sizes of buses with lower emission could be dispatched to save costs from multi-perspectives. In turn, it will improve the level of service and bring more convenience and comfort to passengers. Eventually, it would boost the public reputation of PT operators, attracting more passengers to realize the PT goal as a viable alternative of driving.

1.2. Research Objective

Following from the research gap and the possible contributions, the objective of this study is formulated as follows:

To derive the relationship between trip planner data and the ridership of bus trips by applying machine learning algorithms to see if the short-term prediction can be conducted with an acceptable precision at stop-level.

We will try to uncover the correlation between the trip planner and ridership and determine whether this type of data can be used to predict the ridership of bus trips. We assume there should be a positive correlation between these two types of data, namely if there are more trip requests consulted for a particular period, it is likely to have more travelers expected to use PT during the desired travel period.

However, the accuracy of this type should be acceptable from the perspective of PT operators in order to be valuable. More specifically, a PT operator should be able to predict the ridership for a certain bus trip by incorporating this type of data and know what is the importance of this type of data. Otherwise, if there is no correlation or the correlation is not significant, it means that trip planner data is not an effective new information source for predicting short-term bus ridership.

1.3. Research Questions

Based on the research gap, contributions of application and the research objective, the formulated research questions are presented below, beginning with the main question:

To what extent can trip planner data contribute to short-term bus ridership prediction and what are the important influencing factors in trip planner data and other data in such a prediction model?

The sub-questions are formulated as followed:

1. What are the existing short-term ridership prediction models and influencing factors that internally from trip planner data and externally from other data affect the short-term ridership prediction according to literature?
2. What are the dimensions of analysis in the trip planner data (what data have been collected and stored) and in short-term ridership prediction models (what parameters and variables are there)?
3. Along the above-mentioned dimensions, how does trip planner data correlate with observed ridership from AFC data in the short-term?
4. How can such correlations be leveraged for short-term bus ridership prediction?
5. What is the performance and benefit of using such a prediction model?

In this sense, the aim of the study is to explore the possibilities and identify the importance of trip planner data in the prediction of bus ridership at a stop-level. By ridership at stop-level, it means the occupation rate on board per stop per bus trip. In order to build up a short-term prediction model at stop-level eventually, the travel planner data of 9292 will be used. 9292 is a daily source of PT travel information for passengers in the Netherlands, integrating all information from all transport companies together. This trip planner data is considered as the internal environment to predict the ridership and other datasets used are considered as the external, in terms of spatial, temporal and other characteristics. Moreover, smart card data as the ground truth will be used to correlate and validate the ridership prediction. If the correlation can be constructed, influencing factors can then be selected to investigate based on the assessment and availability of data internally from trip planner dataset and externally from other datasets. For instance, time of the day, the day of the week and the segment of the line, etc. Specifically, the case study will be the bus network in Groningen and Drenthe.

The remainder of this study is organized as follows: the second chapter displays the literature review that uncovers the state-of-art methods and influencing factors that are followed by the description of the case study in the third chapter. The fourth chapter presents the data preliminary and features exploration, including data description, cleaning, merging, and analysis. The fifth chapter presents the methodology that we formulate. Then, the sixth chapter reports the application and development of the method, and the seventh chapter compares the results and analyzes the importance of features. The study wraps up with the eighth section where the conclusions are drawn with the discussion of the future research direction.

2 Literature Review

This chapter describes what are the state-of-the-art methods and contributing factors in short-term ridership prediction. To build a prediction model, the extensive treatment of relevant theories and findings in the scientific literature is needed as knowing relatively more important factors and outperformed models that testified by previous studies can facilitate the establishment and the better performance of the prediction model. The sub research question has been answered in this chapter is:

What are the existing short-term ridership prediction models and influencing factors that internally from trip planner data and externally from other data affect the short-term ridership prediction according to literature?

In order to answer the question, this chapter consists of three parts, which are the exploration of contributing factors and ML algorithms (in a relationship revelation or classification) and conclusion. First, we identify the various contributing factor from multi perspectives. Second, we conclude the current methods of ridership prediction models with an emphasis on ML algorithms. Finally, we draw the conclusions of this chapter.

The search engines are Scopus and Google Scholar and the search terms are constructed using related or broader terms as follows:

*prediction** OR *forecast**
AND
"public transport" OR *"public transit"*
AND
ridership OR *"number of passengers"*

With the aim of investigating the influencing factors for the bus ridership, the first searching term is optional and therefore the literature with a * means that the target outcome is not predicted ridership in the following summary tables. After the search, other than the type of relationship between ridership and contributing features, the existing studied prediction model for transit ridership can also be classified based on the length of the prediction horizon. The prediction horizon is hard to be defined since there is not a universal or uniform rule for it. Researchers choose the time horizon to focus based on their research scope and motivation. Hence, in this study, the long-term models are deemed as models with a prediction horizon of a year. Those models are mainly used to help decide on capital-intensive transit-oriented investments and to investigate the impact of major changes in service and environment on a strategic and tactical level. On the other hand, the short-term model is normally with a prediction horizon of days or hours. These models can be used by public transport operators to increase/decrease supply (dynamic traffic management) and to timely notify travelers on possible crowding (Pereira et al., 2015).

2.1. Influencing Factors of Bus Ridership

To know the influential factors in bus ridership is vital. We carry out a thorough investigation into the data dimensions internally and externally in order to construct the feature set of ML later on. Current literature studies the influential factors as follows:

A consensus of the influence of time and date has been reached. Temporal factors can be diverse, including the time of day, day, week, and month (Ding et al., 2016). The embodiment of the temporal factors starts from a long time ago when Stopher (1992) calibrated a separate model for peak (combination on AM and PM peak), day and night for weekdays. Ding et al. (2016) found that the time of day is associated with the periodic feature of subway ridership that subway ridership is usually high during peak hours and maintains at a moderate level during non-peak hours. Even the feature, peak-hour, could be different as the AM peak tends to be sharper, since work and education journeys coincide at that time, whereas the PM peak is flatly spread due to education trips being earlier (Xue et al., 2015). When it comes to long-term ridership prediction, seasonality can be investigated due to continuing observations. Chiang et al. (2011) included August and October as binary predictors for explaining seasonality. Those two months are significant in the study area and hence cause an effect on the ridership compared to other months.

Among all temporal attributes, events and holidays can be special and influential. Karnberger and Antoniou (2020) discovered that adding event information can improve prediction quality for a few links significantly while others are no use. Since all events would normally have a certain location and starting time associated with them, some people leave early to escape the rush or have to work the next day. Ohler. et al. (2017) investigated four different types of holidays: public holidays, school holidays and semester breaks of the local university and cultural events. To inspect those factors in a more elaborate way, not only the day has been modeled as a binary dummy variable but also one day before and the day starting, the day after and the day ending because the increased demand can be expected at both the start and the end of these periods.

Aside from temporal attributes, spatial attributes are also vital. If the aim of the bus prediction is just one route, the spatial feature is always neglected. While if the scope is at regional or stop-level, this variable is analyzed as the building environment for the attractiveness of passengers. Different modes of stops or stations could have different impacts on ridership. Ding et al. (2016) concluded that the bus transfer activities around the subway station have the most potentially significant effects on the subway ridership, and the bus transfer activities around the subway station have little effects on the subway ridership. Chakour and Eluru (2016) have conducted a study that exhausted the infrastructure and built environment influence on bus ridership at stop-level, which indicated that highway affects bus ridership negatively, while the presence of public transportation around the stop has a positive and significant effect, for instance, metro and train stations and bicycle paths. In addition, residents in urban and rural areas could have a significant difference in mode choice and mobility preference originated from the low density and dispersed locations of origins and destinations of rural areas (Pucher and Renne, 2005). Moreover, parks, commercial enterprises, and residential areas, amongst others, have various effects across the day on boardings and alightings at bus stops.

Weather can result in an impact on ridership. In the article written by Li et al. (2015b), humidity, wind speed, rainfall, and temperature were found to be negatively correlated with bus ridership, although the magnitude of the impact varies depending on route clusters and seasons. In another paper composed by Tao et al. (2016), three scopes of different sizes have been studied, including system, sub-system, and route level. However, there was essentially no significant association between changes in weather and system-wide bus ridership but heavier rain appeared to markedly prompt bus ridership along the busway. Furthermore, Tao et al. (2016) indicated that higher humidity depresses the bus ridership while wind induces the increase in bus ridership in less dense areas.

The characteristics of transit and demand influence the ridership. Van Oort et al. (2015b) utilized the data from the smartcard as the historic demand to forecast future demand by using the elasticity model. Variables, for instance, in-vehicle time, waiting time, number of transfers are also Incorporated as transit features. The same concept but different method can be also seen in the work done by Gummadi and Edara (2019), where historical demand, reflected by tickets, has been used to predict the transit ridership by the artificial neural network. Zhou et al. (2016) steered their efforts on the inclusion of GPS location of passengers and the bus parking positions. The connection between bus and metro, indicated by the transfers is regarded as another transit feeder service feature (Ding et al., 2016).

Other factors, for example, socioeconomic attributes influence the occupation of vehicles as well(Li et al., 2015a). Unemployed people tend to have less need to travel to work and a declining economy alike(Chiang et al., 2011). The presence of alternative modes, for instance, the preference of using rail will hinder the ridership of the bus, given the same situation (Scherer and Dziekan, 2012). And personal travel behavior or mode choice as a consequence of household mode choice chain could also result in a change in the ridership (Dissanayake and Morikawa, 2010).

Table 2.1: Summary of influencing factors of ridership

Article	Temporal[1]	Holiday[2]	Spatial[3]	Weather[4]	Characteristics of transit[5]	Characteristics of demand[6]	Socioeconomic status[7]	Others
Stopher, 1992	✔		✔		✔		✔	
Pucher and Rene, 2005*			✔				✔	
Dissanayake and Morikawa, 2010*					✔	✔	✔	Mode chain
Chiang et al., 2011	✔						✔	Gas price
Scherer and Dziekan, 2012 *			✔					Rail bonus
Li et al., 2015a			✔				✔	
Xue et al., 2015	✔							
Van Oort et al., 2015a	✔				✔			
Li et al., 2015b*	✔	✔		✔	✔			
Zhou et al., 2016	✔				✔			
Ding et al., 2016	✔		✔		✔	✔		
Tao et al., 2016*			✔	✔				
Chakour and Eluru, 2016*	✔		✔		✔			
Ohler et al., 2017	✔	✔		✔	✔			
Gummadi and Edara, 2019	✔				✔			
Karnberger and Antoniou, 2019	✔	✔	✔		✔			

[1] Temporal means the factor relates to time. For example, time of the day or day of the week.
[2] Holiday means the factor relates to holiday specifically. For example public holidays or school holidays.
[3] Spatial means the factor relates to space, infrastructure and built environment. For example, the location of origin or destination is suited in a residential area or business area.
[4] Weather means the factor relates to the weather condition. For example, precipitation or wind velocity.
[5] Characteristics of transit means the factor relates to the service of PT. For example, the PT routes or stops.
[6] Characteristics of demand means the factor relates to the behavior of passengers. For example, the trip distance or travel time.
[7] Socioeconomic status means the factor relates to the economic and sociological combined total measure of a passenger.
* The literature with this symbol means that the topic does not directly predict the ridership.

2.2. Machine Learning Algorithms for Bus Ridership Prediction

Predictive modeling is the problem of developing a model using historical data to make predictions on new data where we are supposed to not have the answer (Geisser, 1993). In general, the predictive modeling can be described as the mathematical problem in which we approximate a function $f(x)$ from input variables (x) to output variables (y). Typically, we can leverage regression and classification to a ridership prediction problem. Both regression and classification are approximating a mapping function $f(x)$ from input variables (x), but the fundamental difference lies in the type of the output variables (y) (Loh, 2011). In a classification problem, the output variables (y) are discrete class labels, e.g. a range of ridership or whether the ridership is beyond capacity or not. In contrast, the output variables are continuous quantities in a regression problem, for instance, the exact number of ridership on board.

The development of manifold payment systems upon various aspects of the public transport system enriches the abundance of available data but meanwhile challenges the traditional data mining methods, such as regression, classification, clustering, etc. ML, as a data mining method, is shown to have the capability to handle high-dimensional, high-volume and multivariate data in a complex and dynamic system, and is able to identify the patterns in the data and the relevant influential factors (Tang et al., 2020).

Recently, there has been a considerable increase in the number of studies in the analysis of public transport data using ML. For instance, Yamaguchi et al. (2019) utilized probe data of buses to predict the bus delay with various methods, including linear regression (LR), artificial neural network (ANN), support vector regression (SVR), random forest (RF), and gradient boosting decision tree (GBDT). In addition, Tang et al. (2020) also applied the GBDT algorithm to estimate the alighting stop for general bus trips in an open AFC system which only has the boarding stop information. At the same time, how to exploit various data sources to predict ridership is also an interesting topic that has been studied.

This study takes the study from Van Roosmalen (2019) as a benchmark, focusing on presenting a clearer research methodology and elevating the results. Van Roosmalen (2019) compared five different ML models, including LR, DT, RF, NN and SVR with three-month AFC data and trip planner data. By validating the result on trips of one route during the morning peak, Van Roosmalen (2019) drew the conclusion that RF predicted the number of people boarding most accurate. However, the proposed methods on first predicting the number of people boarding and alighting were unsuccessful and the selection of features could be given more thoughts. Moreover, some features could be added such as holidays, crowdedness levels on board, etc. The comparison of scenarios could be more in-depth, such as peak and off-peak hours, lines with many and few people, city line and rural line, etc.

The ML algorithms in this study are to discover the relationship between ridership and other contributing factors. To this end, regression and classification should be the main functionality of the method.

To predict the ridership, it can be modeled by using a time series analysis, which only considers the historical pattern but not other external factors. The simplest relationship is a linear one where it includes a desired traffic parameter and a set of variables. The most common model is the ARMA model, consisting of the autoregressive (AR) and the moving average (MA) model. This type of model applies MA to eliminate the past error and uses AR to perform a regression over the previous observations. However, given the ridership of PT is not stationary, it is more likely to build an ARIMA (autoregressive integrated moving average) model to consider the trends in the data. The typical ARIMA should have three components, containing an autoregressive lag, a moving average lag and the difference in the order(Chiang et al., 2011, Gummadi and Edara, 2019, Ohler. et al., 2017, Zhou et al., 2016).

A decision tree is a highly flexible ML algorithm, which can be used. This kind of tree-like method segments the data into layers or regions by a certain rule, learned by training data. The prediction is given by calculating, for instance, the mean of training data in the regions that it belongs to. Those regions are named leaves of the tree. One of the ways to enhance the algorithm is by boosting (Freund et al., 1999). Boosting refers to a general approach that yields accurate prediction rule by incorporating rules of thumb. The combined methodology is called gradient boosting decision tree (GBDT) which can be seen in several papers (Ding et al., 2016, Karnberger and Antoniou, 2020). An example of the decision tree is shown in Fig. 2.1.

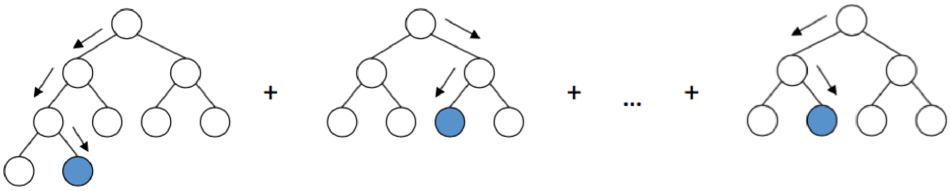


Figure 2.1: Example of decision tree (Shin, 2015)

A reduction in variation should lead to better predictability. Karnberger and Antoniou (2020) first presented a preliminary on the spatial-temporal influential factors to get knowledge on the importance by simulating the decision tree with the aim to reduce the variance. They also mention that the inference is important as it is concerned with understanding the relationship between variables. For example, *How much does Y change when X changes?* GBDT package is able to retrieve the importance of influential factors and then they forward them to the visualization on the network to know the inference. The same methodology can be identified in Ding et al. (2016).

There are methods that can map the non-linear input space into a linear one and can be regarded as dimension reduction methods. For example, SVM is used for classification as well as for regression. For classification, the machines determine a hyperplane based on historical data and try to separate two classes as well as possible (Ohler. et al., 2017). For regression, it is done by finding an area that is as small as possible while holds all historical data. Kernel is the key to the application of SVM. It is capable of mapping non-linear data into higher dimensional spaces that we can find a hyper-layer to make the samples linear. The kernel functions are used to describe the kernels that can reduce the complexity of finding the mapping function. Hence, kernel functions are seen as the inner product of the transformed space. Some of the most commonly used kernel functions are Linear, Polynomial, and Radial Basis Functions (RBF includes Gaussian). In the field of the metro, Wang et al. (2018) applied a combined online SVM model (overall online model and partial online model) to forecast the 5-min ridership of the metro. An example of SVM is shown in Fig. 2.2.

Other than SVM, Principal Component Analysis (PCA) is another popularly used method for dimension reduction. Rather than creating a hyper-layer like SVM, PCA maps the input data onto a new set of axes, which is a coordinate transformation. The new set of axes is called the principal axes or

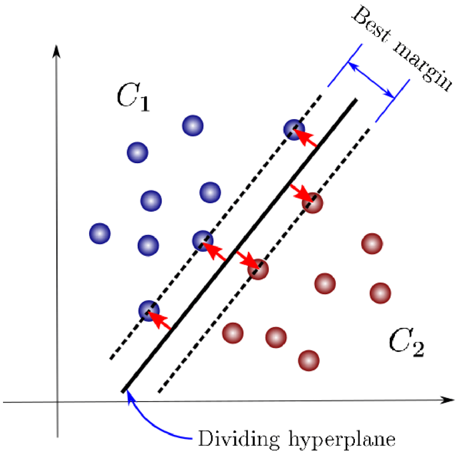


Figure 2.2: Example of support vector machine (Cortes and Vapnik, 1995)

components. Those components combine input variables in a specific way, and by such, it drops the least important variables while still retaining the most valuable parts of all of the variables and those created integrated variables are independent of each other. But till this point, bus ridership prediction with PCA has not been read by the author, although it has been discussed in the field of road traffic, for example by using it to identify loop detector fault detection (Jin et al., 2008). An example of PCA is shown in Fig. 2.3.

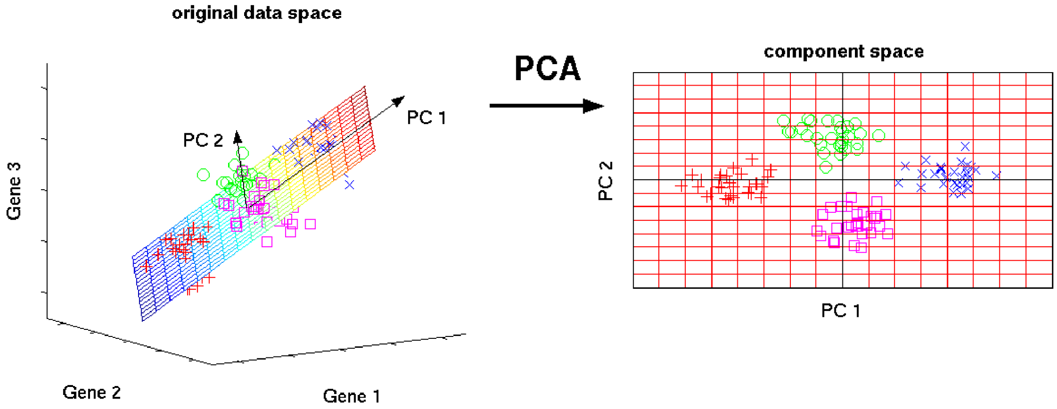


Figure 2.3: Example of principal component analysis (Wold et al., 1987)

Regardless of the different architectures and operations of neural networks, they all share some common features. A neural network is composed of several nodes, which are the processing elements or called neurons to represent a human brain like system (Chiang et al., 2011). These nodes take data as sources and then compute the output dependently via some ways on the values of the inputs, applying an internal transfer function. The nodes are connected with weighted links and typically the relationship between output and its corresponding inputs is non-linear. The weights are adjusted to minimize the deviation of the output through training historical data while it is a black box procedure and therefore it is not able to be reconstructed and deduced how the network comes to its conclusions (Ohler. et al., 2017). Naturally, a structure of the neural network is as Fig. 2.4 shows, including three types of layers, namely the input layer, hidden layer, and output layer.

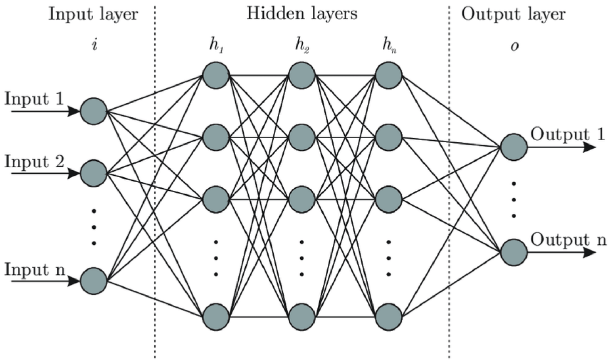


Figure 2.4: Example of neural network (Bre et al., 2018)

Table 2.2: Summary of machine learning methods for ridership prediction

Article	ANN[1]	ARIMA	DT	GBDT	LR	SVR	RF	Others
Jin et al., 2008*								Principal Component Analysis
Chiang et al., 2011	✔	✔			✔			
Xue et al., 2015		✔						Interactive Multiple Model
Ding et al., 2016				✔				
Zhou et al., 2016		✔						
Ohler et al., 2017		✔			✔	✔		
Wang et al., 2018						✔		Suport Vector Machine Combined Online Model
Gummadi and Edara, 2019	✔	✔						
Karnberger and Antoniou, 2019				✔				
Van Roosmalen, 2019	✔		✔		✔	✔	✔	
Yamaguchi et al., 2019*	✔			✔	✔	✔	✔	
Tang et al., 2020*				✔				

[1] Abbreviation of methods has been adopted, details refer to the list of acronyms and glossaries.
* The literature with this symbol means that the topic does not directly predict the PT ridership.

Table 2.3: Summary of literature review

Article	Target	Type [1]	Data Source	Spatial Horizon	Temporal Horizon	Feature[2]
Stopher, 1992	Ridership	EM	Service data	Route	Month	Temporal
Chiang et al., 2011	Ridership	LR+ANN+ARIMA	Several	Region	Month	Temporal socio-economic
Li et al., 2015a	Ridership	4-step model	Panel data	Route	Year	Socio-economic
Xue et al., 2015	Ridership	IMM	AFC	Route	Weekly +Daily +15 minutes	Temporal
Van Oort et al., 2015a	Ridership	EM	AFC	Stop	Hour	Demand transit
Zhou et al., 2016	Ridership	WTVP+ ARIMA+ SLEF	AVL+AFC	Stop	Minute	Demand transit
Ding et al., 2016	Ridership	GBDT	AFC	Stop	15 minutes	Temporal spatial
Ohler et al., 2017	Capacity	LR+SVR+ARIMA	APC	Stop	Trip	Temporal spatial weather
Wang et al., 2018	Ridership	SVMOM	AFC	Stop	5 minutes	Temporal demand
Gummadi and Edara, 2019	Ridership	ANN+ARIMA	Tickets	Route	Day	Demand
Van Roosmalen, 2019	Ridership	LR+ DT+ SVM+ RF+ NN	Trip Planner+ AFC	Stop	Trip	Temporal spatial demand transit
Karnberger and Antoniou, 2019	Ridership	GBDT	APC	Link	30 minutes	Temporal spatial
Pucher and Rene, 2005*	N\A	EA	Panel data	Region	N\A	Spatial
Jin et al., 2008*	Traffic flow	PCA	Loop detector	Detector	5 minutes	N\A
Dissanayake and Morikawa, 2010*	N\A	NLM	Panel data	Region	N\A	Household
Scherer and Dziekan, 2012*	N\A	EA	Panel data	Region	N\A	Rail bonus
Li et al., 2015b*	N\A	CA+LR	AFC	Route	N\A	Weather
Chakour and Eluru, 2016*	N\A	CML+ORP	AVL+AFC	Stop	N\A	Spatial
Tao et al., 2016*	N\A	EA	AFC	System level + sub-system level + route level	N\A	Weather
Yamaguchi et al., 2019*	Bus delay	LR+ ANN+ SVM+ RF+ GBDT	Probe	Region	5 minutes	Bus delay
Tang et al., 2020*	Alighting stop	GBDT	AFC	Route	N\A	Temporal demand transit weather

[1] The column of type represents the method that the article(s) has (have) used. The abbreviation of methods can be found in the list of abbreviations.
[2] The feature has the same meaning as shown in the annotation in Table. 2.1.
* The literature with this symbol means that the topic does not directly predict the ridership.

2.3. Conclusion

We present a summary of the literature review in Table. 2.3. From the literature review, it can be seen that the variables used in the different studies differ a lot. Depending on the time, location and level of temporal and spatial aggregation, the impact of variables differs. The used spatial and temporal level is generally chosen so that the input variables fluctuate. The long-term, intermediate-term and short-term are also defined based on the horizon that researchers want to focus on. Long-term demand forecasting models use variables that only change slowly over time. Intermediate-term demand forecasting models use variables that change per month or season. Short-term demand forecasting models use variables that change with the time unit used in the forecasting method, such as lagged demand, the occurrences of event and weather.

The different variables can be categorized in the following groups: temporal, demand characteristics, weather, event, holidays, transit characteristics, other mode characteristics, spatial/built environment, socio-economic and socio-psychological. The first six of these groups can be useful to predict short term demand. Variables form the last four groups vary mostly only in the long-term. Depending on the location, time and aggregation level, different variables are used. Even when the influencing factors are known it matters how they are used as input in the model. For instance, it is possible to use relative values, moving averages or it could be useful to divide the variable in multiple dummy variables. Moreover, Table 2.1 also reports that researchers begin to use revealed preference more frequently, namely smart card data, AVL data and AFC data. This is due to collecting stated preference data can be time and money consuming. Normally, the stated preference is revealed by panel data or survey and it takes months or years to build up a complete and detailed dataset. In contrast, the revealed preference data can be retrieved much faster even could be real-time. With the advantageous feature of short-term and real-time, it can provide more insights into the tactical and operational levels of transit management.

Besides, regarding the ridership prediction with ML, there are a lot of different models that have been applied but there is no such a model that works best in every scenario as we can see from Table. 2.2 (Raschka, 2015). Depending on the motivation of the researcher, some studies dive into the more accurate prediction results while others try to explore the importance of features. Remarkably, almost no article hitherto has investigated the relationship between trip planner data and ridership, except for Van Roosmalen (2019). Even Van Roosmalen (2019) has not presented promising results with the methodology he formulated. Moreover, it focuses too many variables and too broad scope in the first place which shows a lack of well-knitted data analysis before performing the ML models. Finally, Van Roosmalen (2019) predicts the boarding and alighting passengers separately which would possibly introduce errors twice.

Table 2.4: Summary of interpretable machine learning model (Molnar, 2019)

Algorithm	Linear[1]	Monotone[2]	Interaction[3]	Task
Linear Regression	Yes	Yes	No	Regression
Logistic Regression	No	Yes	No	Classification
Decision Trees	No	Some	Yes	Classification Regression
RuleFit	Yes	No	Yes	Classification Regression
Naive Bayes	No	Yes	No	Classification
K-Nearest Neighbors	No	No	No	Classification Regression

[1] Linear means the relationship between features and targeted variable is linear or not.
[2] Monotone represents the relationship goes in the same direction or not.
[3] Interaction describes the connections among features to predict the target outcome and this can improve predictive performance.

A table of interpretable ML models is shown in Table. 2.4 (Molnar, 2019). In this table, linear means the relationship between features and the targeted variable is linear and monotone means the relationship goes in the same direction (e.g. an increase in the feature always leads to an increase of the targeted variable). Moreover, interaction stands for the connections among features to predict the target outcome and this can improve predictive performance. However, too complex interactions would lead to less interpretability. In most of the cases, the decision trees are monotone, however, some decision trees are not monotone. Monotonic classification problems refer to classification problems that are required to be monotonic with respect to the attribute values. Decision trees should be ideally monotonic, regardless of whether or not the training set is monotonic. Ben-David (1995) explains that information-theoretic top-down induction decision tree algorithms that use entropy as the criterion for attribute selection may produce non-monotonic decision trees. Since then, we have witnessed the increasing amount of literature that enhance the monotonicity of decision tree, for example applying an induction approach to generate monotonic decision trees from sets of examples which may not be monotonic or consistent (Lee et al., 2003) or to adjust the probability estimated in the leaf nodes in case of a monotonicity violation (Van De Kamp et al., 2009).

This study intents to find out to what extend that trip planner data can facilitate short-term ridership prediction and we will apply the ML algorithms to bridge this research gap, in which the interpretation of features is vital and necessary. Thus, this study mainly refers to the ML methods that have more abilities in feature interpretation as suggested in Table 2.4. In this way, the property of “black box” is avoided and we can get to know the importance of features, especially trip planner data. Moreover, we will also construct a baseline model to establish the comparison, which is the model that PT operators are currently using. It estimates the ridership of this week based on the same trip of last week.

3 Case Study

Following the literature review, this chapter describes the case study in the following way. First, we introduce the case study area of Groningen and Drenthe, especially the bus network and types of bus service. This provides background knowledge, in terms of spatial and geographical. Then, we present the general introduction for the trip planner that is selected for researching. Finally, we illustrate the time scope of this study and the case study lines.

3.1. Background and Bus Network of Groningen and Drenthe

To understand the development of the PT service network and make the study comparable to other studies with different spatial scope, an introduction of the overall demand and supply characteristics is needed.

Groningen and Drenthe, two provinces adjacent to each other, are suited in the northeast of the Netherlands. Groningen is the seventh-largest province of the Netherlands with a population of 583,990 and a total area of 2,960 km^2 and Drenthe is the ninth largest with 492,167 residents and 2,680 km^2 large as 2019 Central Bureau for Statistics (CBS) reported[1]. Groningen consists of 23 municipalities where the city of Groningen as the largest city covers around 40 % of the province population. Drenthe has 12 municipalities and its large cities - Emmen, Assen and Hoogeveen- contain more than 40 % of the province population as well. This indicates that those two provinces have a relatively denser population and their residents favor living in the urban cities more, which in turn shows the demand for PT. Figure 3.1 shows the number of residents per municipality based on the same data from CBS. It can be seen from the figure that densely populated cities are limited and clustered in certain corridors that spread from Groningen city, mainly towards south or east, namely Assen-Hoogeveen, Emmen and Midden-Groningen-Oldambt.

Surprisingly, these dense areas have no trams or light rails that can provide a substantial supply for the travel demand due to historical issues. Bus is the only mode connecting cities, towns or villages for these two provinces and it is vital to the daily life of residents, living in those two provinces. By providing a more reliable bus service, the overall living condition can be correspondingly elevated. The complete bus network of Groningen and Drenthe is shown in Fig A.1 (Appendix).

The service provided by Qbuzz (the only bus operator in region Groningen and Drenthe) and OV (PT) bureau Groningen Drenthe can be broadly defined into four kinds, including intercity bus, city bus, neighborhood bus and others. The inter-city bus is operated on a regional level and is divided into two types, namely Q-link and Qliner as shown in Fig 3.2 (although the year of the map in Fig 3.2 is 2020 and the case study time is 2019, there is almost no change on this inter-city bus map). Q-link on the left side of Fig 3.2 consists of 8 lines with one line - line 12 - from Emmen to Emmen South that is not shown on the figure while Qliner has 6 lines is presented on the right side of Fig 3.2. Except for line 15 of Q-link that connects Groningen central station and Zernike Campus (a major working area) and line 12 above-mentioned, all other lines are linking large residential and working areas in provinces Groningen and Drenthe directly. These lines serve as the backbone of the commuters, students and are essential for the daily life of their passengers. Hence, these Q-link lines have a relatively high speed with fewer stops

[1] Bevolkingsontwikkeling; regio per maand, 2019.

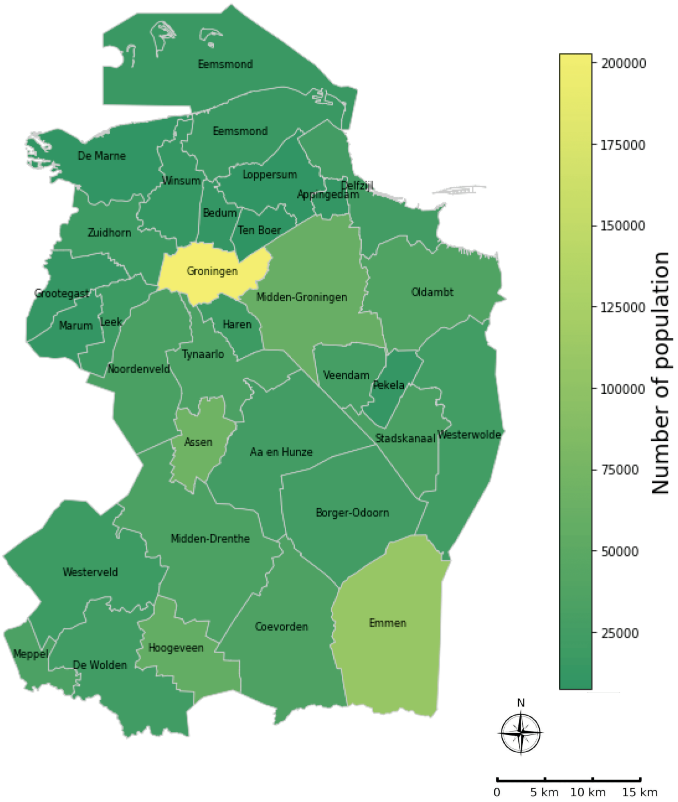


Figure 3.1: Number of residents per municipality in Groningen and Drenthe

along the line with a frequency of 10 minutes and even less during peak hours. Moreover, these lines would use dedicated lines (e.g. emergency lines) to gain priorities if traffic congestion happens. Qliner is another comfortable, fast and direct line, sharing a lot of commons with Q-link, but with longer routes. Normally, they would traverse between big cities and towns in provinces Groningen and Drenthe at a decent speed. From the Q-link and Qliner networks, it can be seen that Groningen as the biggest city can be regarded as a hub, connecting other big cities and its neighboring towns.

Most buses in Groningen and Drenthe are city buses and this is the second type of bus operated in that region. Those buses not only run in cities such as Groningen, Assen, Emmen, Hoogeveen, Meppel, and Veendam but also run in villages or towns. These city bus lines have to stop many times along the route to cover the train stations, shopping centers, hospitals, and living and working areas. Generally, most of the cities have a frequency of twice an hour and some busier routes could have more frequencies. In contrast, if it is operated in villages or towns, the frequency could be decreased to once or twice per hour during weekdays and much less during weekends.

The third type of bus is buurtbus in Dutch, which means the neighborhood bus in English. Those buses are organized locally and driven by volunteers. Other types of buses, including night bus, hub taxi, and bell bus are operated under specific conditions that are left out of scope.

3.2. Background of Trip Planner 9292

The 9292 journey planner is an interactive trip planner, established in 1992, the Netherlands. However, it is not the only trip planner that a traveler can use in the country due to the market competition with NS (the biggest train service operator), Google Maps, ANWB and, etc. These kinds of trip planners have less difference in functions and interfaces but have one common goal that is to provide integrated travel information to its users. Traveler can choose use or do not use such an application before traveling and can also choose any of the trip planners when they plan the trips according to their habits and preferences. The penetration rate of trip planner and the market competition of 9292 are regarded as

[2] https://www.qbuzz.nl/gd/reis-plannen/soortenbussen

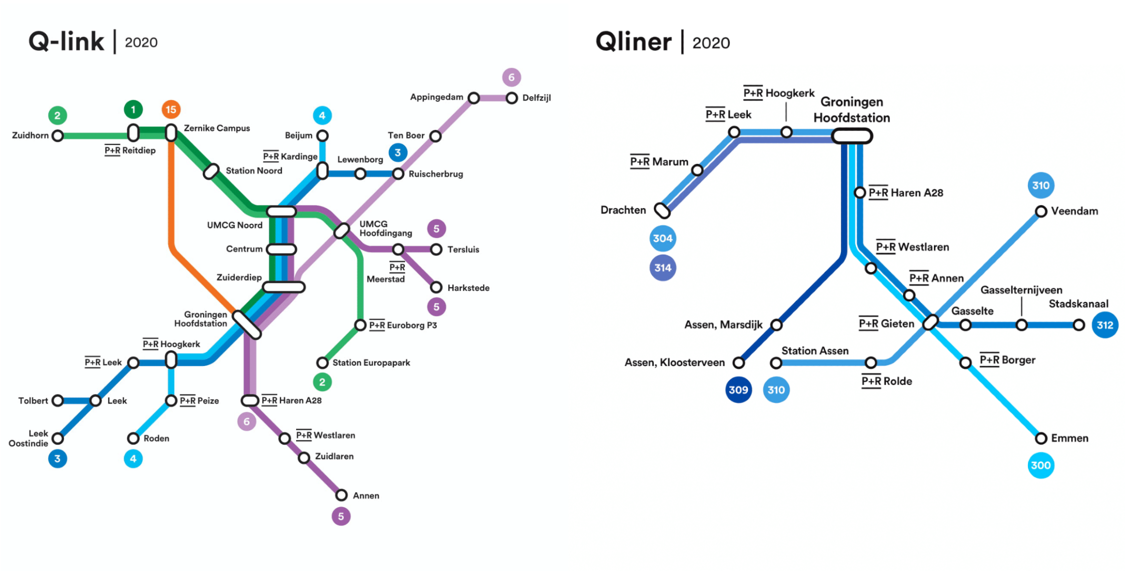


Figure 3.2: Q-link and Qliner network[2]

the limitations of this study. Because we only analyze the trip planner data from 9292, although it is representative, it is still partial. This could underestimate the importance and contribution of trip planner data in ridership prediction.

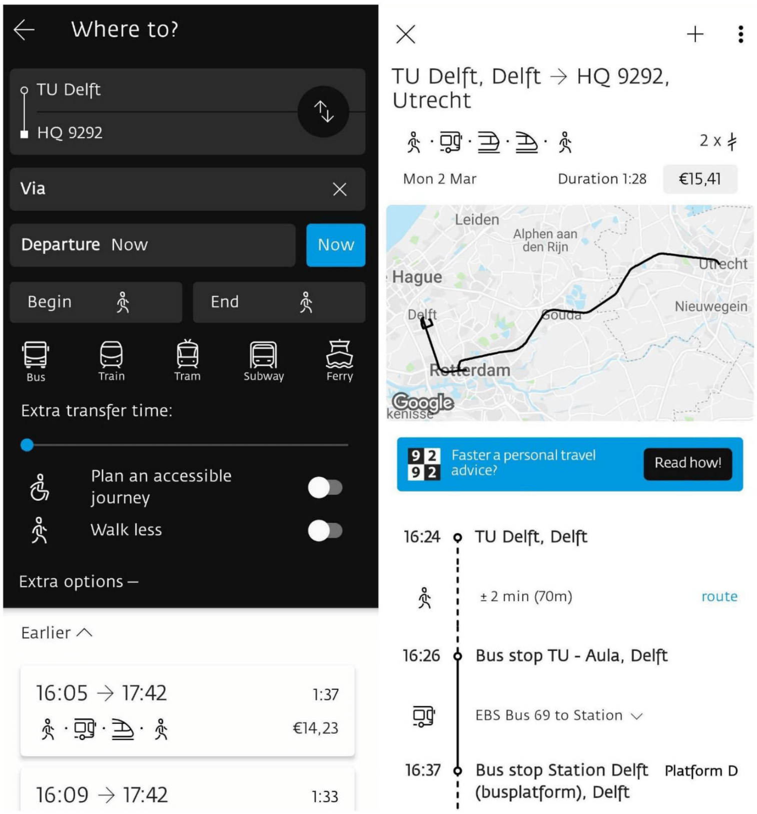
However, 9292 is notably the biggest one and with the largest market share of approximately 46 % [3]. Every day, it has 600,000 active devices with 4 to 5 requests per device on average, resulting in around 3 million requests per day [4]. It is more local in the Netherlands with the PT information of all modes and more detailed in bus, metro, tram, light rails, which matches the interest of this study. A traveler can access such a trip planner by mobile phone and tablet with this app installed or a web browser. An interface of the 9292 mobile app is shown in the left part of Fig. 3.3 where the trip planner requires its users to select origin, destination and intermediate stop based on his/her willingness. Moreover, a user could choose the desired departure time or arrival time in a one-minute basis plus the access and egress mode, namely walking or cycling. In addition, 9292 allows its users to choose additional options, for example, desiring traveling with a specific mode desired or exclude one explicitly and choose to have more buffer transfer time, less walking, or travel with certain disabilities during the journey.

With the filled-in information, the planner searches in the database for the transport supply and then provides the suitable and possibly multi-modal trip alternatives as shown in the lower left part of Fig. 3.3. The alternatives contain the information of departure time, arrival time, trip duration, fare and involved mode(s) based on user preference. The first alternative is always with the least travel time and has been recorded into the database. The recorded data, regardless of the departure or arrival time chosen, will always be saved as the departure time, i.e. the arrival time will be converted as departure time. Moreover, if a user selects one of the alternatives and dives into details, 9292 will show the map with the route marked as shown in the right part of Fig. 3.3. Suppose a specific section of the route has been selected, the planner will narrow down the scope to that section. Furthermore, it also brings the delay, disruptions or the cancellation information of a trip, which means real-time travel information.

Real-time transit information provided by trip planner facilitate the travel of passengers and the understanding of travel patterns of operators, however, the need of travel information and the importance of that information differs at a different stage of a trip and differs along with the purpose of using such a trip planner and the characteristics of users. For instance, frequent travelers - 5+ trips per week - use trip planner 2.5 times greater than that of infrequent travelers - <1 trip per week (Yeboah et al., 2019). Mulley et al. (2017) concluded that age has a strong impact on the usage of the trip planner, for example, people older than 50 years old would prefer printed timetables. From this regard, the

[3] The market share comes from independent research by 9292, which the company regularly has carried out by Newcom Research: https://www.newcom.nl/

[4] The statistics of the active device is another study carried out by 9292. 9292 uses Flurry for the app, measuring the number of unique devices per day on which the app is used at least once: https://www.flurry.com/



TU Delft
HQ 9292
Departure Now
Now
Begin
End
Extra transfer time:
Plan an accessible journey
Walk less
Extra options –
16:05 → 17:42 1:37
€14,23
16:09 → 17:42 1:33

TU Delft, Delft → HQ 9292, Utrecht
2 x
Mon 2 Mar Duration 1:28 €15,41
Faster a personal travel advice?
16:24 TU Delft, Delft
± 2 min (70m)
16:26 Bus stop TU - Aula, Delft
EBS Bus 69 to Station
16:37 Bus stop Station Delft (busplatform), Delft Platform D

Figure 3.3: Example of 9292 trip planner interface

age limitation is introduced where not every traveler is using the trip planner and this limitation could potentially hamper the importance of trip planner data. Figure 3.4 shows that the percentage of people above 45 years old in the provinces Groningen and Drenthe. From Fig. 3.4, we can see that half of the population of those two provinces in general are below 45 years old and it is even younger in big cities like Groningen. Besides, bus users are much more inclined to be mobile app users, compared to train users as the timetable is not constantly adjusted (Mulley et al., 2017). This matches the interest of this study and potentially provide insight into the prediction of bus ridership with trip planner data.

There are several other limitations in the study, such as, users can plan their journey into parts instead of using the “via” option and due to the fact that we can’t trace the IP address, it is hard to know if it is a single journey with multiple legs (trips) or different journeys, which would bring in “more” trip planner data than it should be and thus incur the overestimation of the importance of trip planner data. Besides, due to the same privacy issue, we don’t know if it is a bunch of people traveling or just a single person. If it is just a single person, the influence is understandable while if it is a bunch of people, referring to the same advice given by the trip planner, an underestimation of the importance of this type of data would be introduced.

As above-mentioned, the trip purpose will also introduce noise in the data. More often, travelers request travel advice in advance in order to facilitate their travel, however, there could be other causes. The motivation of a traveler for using the trip planner can be roughly summarized as:

1. To find the travel advice with the least travel time directly between origin and destination as provided by 9292.

2. To find the travel advice with the least generalized cost directly or indirectly between origin and destination.

3. To check the travel decision that made, namely trace back to the travel plan that chosen, including transfer time, platform, line or location, etc.

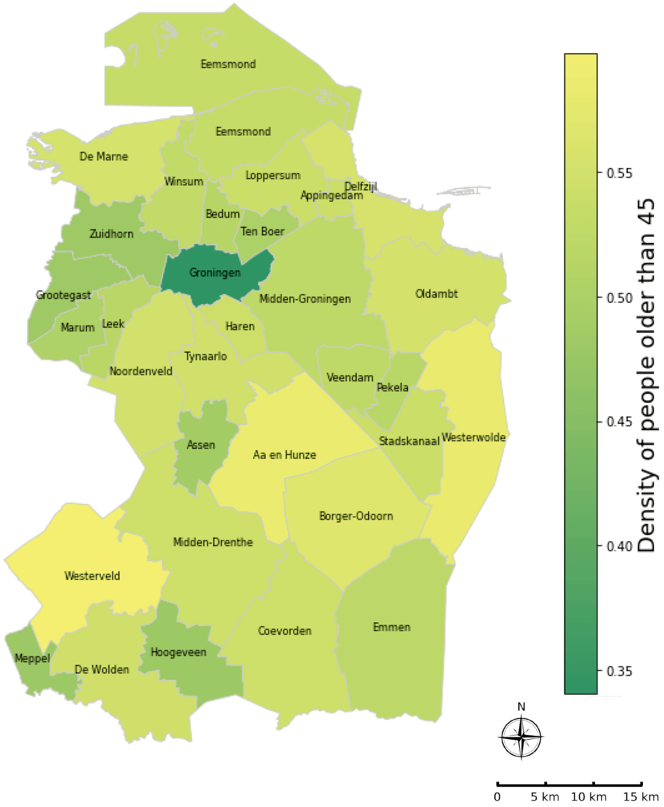


Figure 3.4: Density of residents above 45 years old per municipality in Groningen and Drenthe

4. To check the real-time transit information, containing delays, disruptions or other unexpected situations spontaneously or force majeure.

5. To investigate the timetable of PT. For example, get the knowledge of evenly distributed patterns or hourly patterns.

6. To find an alternative for other modes, including private vehicles.

7. To know the disruptions in advance in order to change or even cancel their travel plan.

8. To illustrate or to look up historic journeys for declaration purposes.

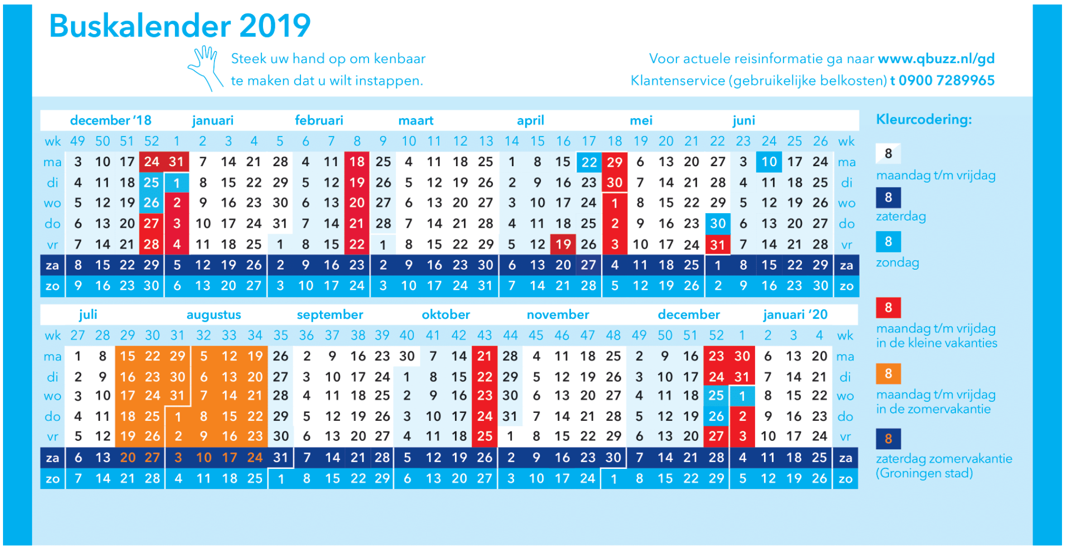
9. To request a new piece of trip advice with a later departure (arrival) time due to unexpected situations.

Hence, some trips may not be realized due to the trip purpose. Depending on the trip purpose, the number of requests, the selected alternative and the intended departure (arrival) time for a single journey varies. The main objective of this study is to predict the ridership on board in advance and thus only pre-travel requests are interested. Nevertheless, it is almost impossible to differentiate different purposes for requests and only the first trip advice is saved in the dataset. Due to this design, noises in data are inevitable.

3.3. Time Scope Selection

The study utilizes the data from the smart card and trip planner in October (2019), namely the 1st to the 31st of October. First, this is the most recent data available when the study started. Second, on 15th December Qbuzz changed their timetable to the new one. Accordingly, it takes time for residents in Groningen and Drenthe to adapt to the new timetable and thus the questions for travel advice will surge, resulting in bias for the prediction. Lastly, the payment has been changed as well with the introduction of E-ticketing and cancellation of cash payment.

The bus calendar of Qbuzz in Groningen and Drenthe of 2019 is shown below in Fig. 3.5. During this October, there is an official holiday for schools and therefore influences the travel of students, teachers and other school-related jobs from 19th to 27th in this region. Hence, we will investigate this influence in the later chapter. Other than this week, all weekdays in October 2019 are normal days without obvious and significant disruptions.



Buskalender 2019

Steek uw hand op om kenbaar
te maken dat u wilt instappen.

Voor actuele reisinformatie ga naar **www.qbuzz.nl/gd**
Klantenservice (gebruikelijke belkosten) **t 0900 7289965**

	december '18					januari				februari				maart				april					mei				juni				
wk	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	wk
ma	3	10	17	24	31	7	14	21	28	4	11	18	25	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	ma
di	4	11	18	25	1	8	15	22	29	5	12	19	26	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	di
wo	5	12	19	26	2	9	16	23	30	6	13	20	27	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	wo
do	6	13	20	27	3	10	17	24	31	7	14	21	28	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	do
vr	7	14	21	28	4	11	18	25	1	8	15	22	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	vr
za	8	15	22	29	5	12	19	26	2	9	16	23	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	za
zo	9	16	23	30	6	13	20	27	3	10	17	24	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	zo

	juli				augustus					september				oktober				november					december				januari '20				
wk	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	wk
ma	1	8	15	22	29	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	ma
di	2	9	16	23	30	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	di
wo	3	10	17	24	31	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	wo
do	4	11	18	25	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	do
vr	5	12	19	26	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	vr
za	6	13	20	27	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	za
zo	7	14	21	28	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	zo

Kleurcodering:

8	maandag t/m vrijdag
8	zaterdag
8	zondag
8	maandag t/m vrijdag in de kleine vakanties
8	maandag t/m vrijdag in de zomervakantie
8	zaterdag zomervakantie (Groningen stad)

Figure 3.5: Bus calendar of Qbuzz in 2019[5]

From another point of view, we look into the number of smart card transactions per month to explore the seasonal passenger flow. The line chart of the number of smart card transactions from June to November is shown in Fig. 3.6 below. During summer (from June to August), the number of transactions is below the average with little fluctuations due to holidays and summer break. The semester usually begins in September and therefore the number surges and becomes steady from September to November. Thus, October as in between September (the start of the term) and November can be a decent month to be selected.

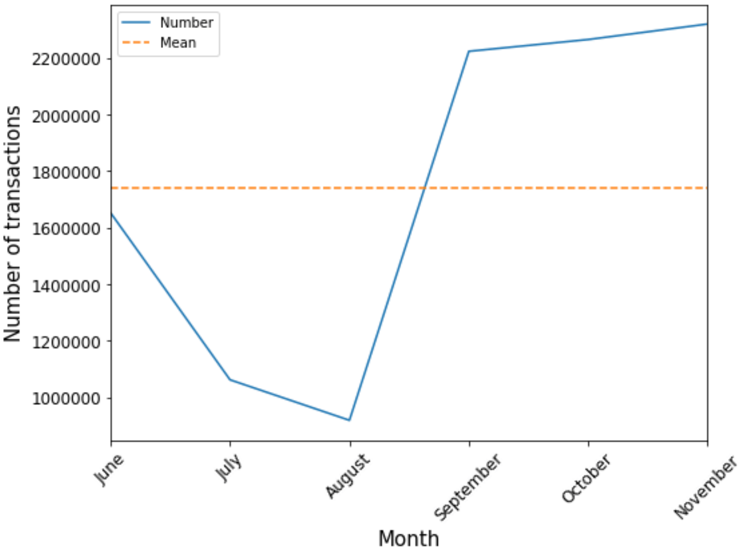


Figure 3.6: Number of smart card transactions per month from June 2019 to November 2019

[5] https://www.qbuzz.nl/gd/reis-plannen/busboekje-gd

3.4. Scope of Lines

This study will utilize Qliner 300 (inter-city, fast service), Q-link 1 (within a major city, traversing a dozen of important locations), Line 50 Groningen-Assen (city-city bus line) and Line 35 Groningen-Oldehove (city-village bus line) to validate the methodology and perform ridership prediction, as shown in Fig. 3.7. The objective of creating such a dataset is to include different types of lines and thus represent different characteristics of ridership (demand) and line (supply). Moreover, these three types of buses have more traffic than others, according to the number of requests per mode shown in Fig. 3.8.

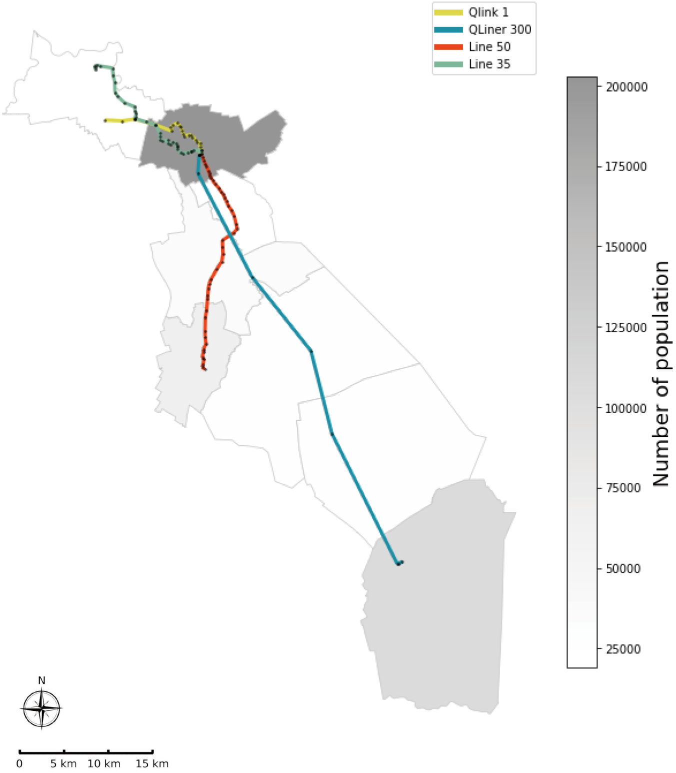


Figure 3.7: Map of case study lines

The general information of every case study line is shown in Table. 3.1 (For the complete information of every case study line, see Table A.1 (Appendix).). This table takes the outbound line as an example because the difference between the outbound line and the inbound line is minor, in the sense of organization, fleet, etc. Moreover, the lines recorded in the table are the options with the highest frequency, which will be the prediction line in the further chapter since the highest frequency option is normally 20 times higher than other options.

Qliner 300 is an intercity line that travels between two major cities -Groningen and Emmen- in provinces Groningen and Drenthe with only 8 stops and an average stop distance of 7335.6 meters. The average stop distance is approximately 20 times larger than that of other lines and the seating capacity is relatively large as the trip length is long as high as around 107 minutes. It is comfortable and safe to provide seats for long-distance traveling passengers. During weekdays, from the first trip to the end of the evening peak, Q-liner 300 operated in every 12 minutes and with 13 vehicles actively serving. Moreover, during off-peak and weekend daytime, it operators in every 30 minutes. Even amid the night, there is a headway of 60 minutes for providing essential inter-city passenger transport. Hence, this line provides a fast inter-city service for the residents with sparse stops and a large distance between stops.

Q-link 1 is traveling within Groningen city to cover several vital locations, including campus, working area, two railway stations (north station and central station), and a medical center. Correspondingly, the average stop distance is low. This design also determines that the ridership pattern of Q-link 1 should be different from Qliner 300 in which we can expect a comparatively flat one from Q-link 1 while a bi-modal pattern at starting and ending stop is assumed to have from Qliner 300. It serves regularly with a headway of 30 minutes, except for on Saturdays that it has two ways of operation. As it traverses in the city center, the duration of weekdays is longer and hence a lower speed, compared to the weekend. The fleet of buses is larger during the weekday to provide sufficient supply for passenger transport demand.

Bus line 50 drives between Groningen city and the third-largest city Assen with dense stops, compared to Qliner service. It is worthwhile noticing that there are three parallel lines, providing the same inter-city service between Groningen and Assen. Bus line 50, among them, is with less frequency but more stops. No matter the type of vehicles, line 50 has a large total capacity with a more standing place for passengers. During weekday daytime and weekend off-peak, it has more frequency and lower speed, especially morning peak hours for catering to the passenger demand. In contrast, during weekday night, Saturday morning and evening, and the whole Sunday, it has less frequent service and capacity. This design is in line with the conventional understating of passenger demand patterns to avoid supply surplus.

Bus line 35 runs between the city Groningen and its adjacent small town Oldehove. It has the same bus type and capacity as that of line 50. It also has a dense stop setting with a short distance between stops to provide the necessary accessibility for citizens. The strategy of bus line 35 operation is similar to line 50, namely more frequency during weekday peak hours and weekend daytime while less frequency during other periods. Overall, line 35 ends earlier than other lines, particularly on Saturday. Furthermore, there is no service on line 35 on Sundays. Thus, we can infer that the connection between Groningen and its surrounding towns or villages is limited and less resilient than other ones.

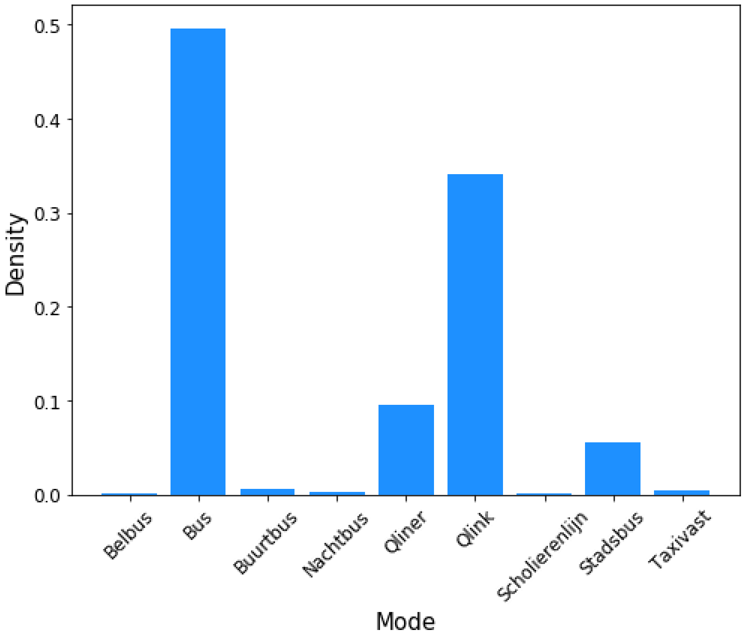


Figure 3.8: Density of trip planner requests per mode in October 2019

3.5. Conclusion

In this chapter, we introduce the context and the background of the study, including area and time scope. We will further utilize the data provided jointly by 9292 and OV-bureau Groningen and Drenthe in October to validate our methodology. The case study area is suitable for researching with the feature of bus-oriented, youth-prone, and density-divers. The bus line types are various, including inter-city, inner-city, and rural. This case study could potentially offer insights into the comparison for the prediction of different line types. The trip planner - 9292 - allows us to study with the largest trip advice database and hence is a relatively strong representation for trip planning in the Netherlands. Next, the study chooses October as the time scope of this study with the possible least noise. It will facilitate investigation and analysis through the same bus trip with different temporal settings. Finally, this chapter illustrates the scope of lines in order to build up a complete set of case studies, namely to include all types of PT services in the case study area. In this way, we will perform the prediction on different services with significant dissimilar characteristics for validating the methodology.

Table 3.1: Main information on case study lines*[6]

Line (outbound)	**Qliner 300**	**Q-link 1**	**Line 50**	**Line 35**
Number of Stops	8	19	43	38
Average Stop Distance (m)	7335.6	401.3	664	592.2
Capacity (seats+standing)	73+55/69+42/81+0	43+82	34+87/47+118/40+50	34+87/47+118/40+50
Weekday Headway (min)	12/30/60	30	16/30/60	30/60/33/60
Saturday Headway (min)	30/60	43/29	59/30/60	60
Sunday Headway (min)	30	60/30	59	No operation

* The symbol / means the line has different types of operation. For detailed information, see Table A.1 (Appendix).

[6] https://www.qbuzz.nl/gd/reis-plannen/busboekje-gd

4 Data

Following the case study set-up in the previous chapter, this chapter analyzes the data that we use in this study. We present the workflow of this chapter below in Fig. 4.1 in which we show which analysis we conduct and the reason why we need to do it.

We begin by presenting the context of three datasets that we use in this study, including trip planner data, smart card data, and AVL data. We utilize the trip planner data to explore the usefulness and contribution to predict the short-term bus ridership. Smart card data is not only the prediction target but also will be reasonably split into a validation set as the ground truth of this study. Also, we derive several variables from the smart card dataset as independent variables, such as the monthly average and the ridership last week of the predicted trip this week, etc. We refer to AVL data to map the trip planner data and smart card data onto the desired bus trip. Then, we report the data cleaning process of the three datasets in which we detect, remove, and correct corrupt or inaccurate records from the raw data and build the link among the three datasets. Next, since these three data come from different sources, we integrate these datasets into one single, unified, and clean data frame to facilitate the analysis following.

The second section of this chapter explores the data from two aspects, separately. In the beginning, we unravel the passenger behavior of using such a trip planner app by investigating the timing advance, compared to the desired travel time and vehicle start time. This step enlightens the user preference and the prediction time horizon. Moreover, we derive the distribution of the trip planner data for discovering the interrelation among the requested, the desired, and vehicle start time of travel advice over an average day to deepen the understanding of this user preference. Then, we uncover the influence of temporal variables on the number of trip planner requests over an average day, for example, the influence of holiday or the weekends. Next, we take smart card data into account to study the realized requests over an average day. It helps us roughly understand the relationship between trip planner and smart card data and the variance of such a realization on an average day. Furthermore, we dive into the smart card data by first constructing the distribution of smart card data over an average day to testify the influence of temporal variables on ridership. Finally, we develop the ridership profile of representatively average trips to unfold the impact of spatial variables on the ridership.

The last section jointly analyzes the trip planner data and smart card data. This joint analysis starts from two temporal scopes, day-level and stop-level, to establish a comparison. We first examine the joint distribution of thees two data to reveal the relationship between them and its strength. Intuitively, people would realize their trips at a day-level. However, when narrowing down to a stop-level, people may change their minds to a different route or another travel time. Plus, there are no recordings for the selected trip advice in the trip planner data. Then, we only focus on the stop-level as this is the level of interest. Second, we construct the covariance matrix to measure how much two variables change in tandem. Lastly, we introduce the correlation matrix to emphasize the normalized relationship, which is not affected by scale. When interpreting the relationship, we give priority to the correlation related to ridership to determine the highly correlated variables that we can put into the model.

The sub research question has been answered in this chapter is:

What are the dimensions of analysis in trip planner data (what data have been collected and stored) *and in short-term ridership prediction models (what parameters and variables are there)?*

Part of the sub research question has been answered in this chapter is:

Along the above-mentioned dimensions, how does trip planner data correlate with observed ridership from AFC data in the short-term?

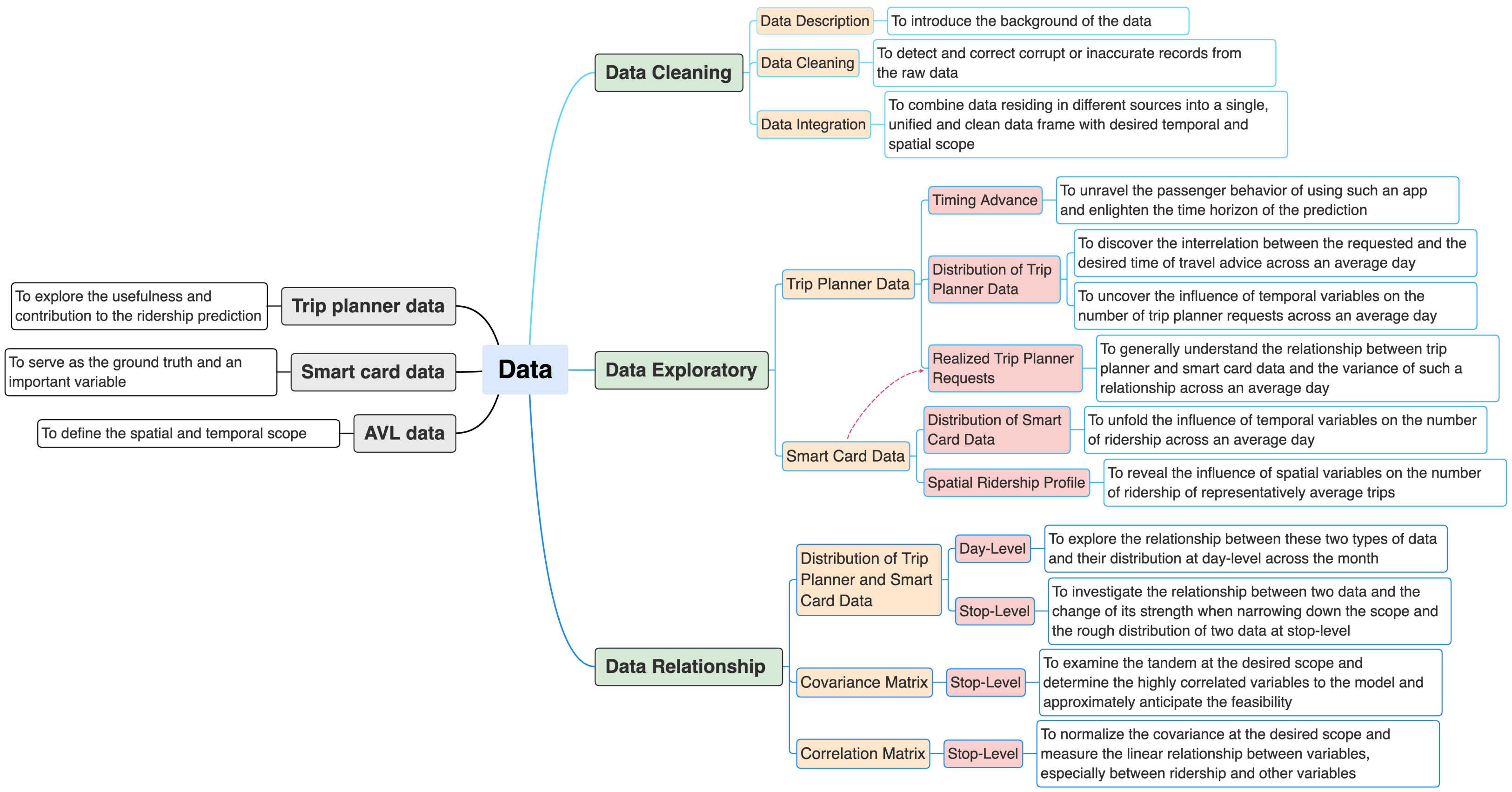

Data
Trip planner data
To explore the usefulness and contribution to the ridership prediction
Smart card data
To serve as the ground truth and an important variable
AVL data
To define the spatial and temporal scope
Data Cleaning
Data Description
To introduce the background of the data
Data Cleaning
To detect and correct corrupt or inaccurate records from the raw data
Data Integration
To combine data residing in different sources into a single, unified and clean data frame with desired temporal and spatial scope
Data Exploratory
Trip Planner Data
Timing Advance
To unravel the passenger behavior of using such an app and enlighten the time horizon of the prediction
Distribution of Trip Planner Data
To discover the interrelation between the requested and the desired time of travel advice across an average day
To uncover the influence of temporal variables on the number of trip planner requests across an average day
Realized Trip Planner Requests
To generally understand the relationship between trip planner and smart card data and the variance of such a relationship across an average day
Smart Card Data
Distribution of Smart Card Data
To unfold the influence of temporal variables on the number of ridership across an average day
Spatial Ridership Profile
To reveal the influence of spatial variables on the number of ridership of representatively average trips
Data Relationship
Distribution of Trip Planner and Smart Card Data
Day-Level
To explore the relationship between these two types of data and their distribution at day-level across the month
Stop-Level
To investigate the relationship between two data and the change of its strength when narrowing down the scope and the rough distribution of two data at stop-level
Covariance Matrix
Stop-Level
To examine the tandem at the desired scope and determine the highly correlated variables to the model and approximately anticipate the feasibility
Correlation Matrix
Stop-Level
To normalize the covariance at the desired scope and measure the linear relationship between variables, especially between ridership and other variables

Figure 4.1: Overview of the workflow and objective of data analytic in the data chapter

4.1. Data Description, Cleaning and Integration

This section provides an understanding of the context of the data that we use in this study, including trip planner data, smart card data, and AVL data. We first present the background and the structure of the data, i.e. data description. Then, we report the cleaning process of the data to detect, remove, and correct the data corruption and inaccurate recordings. Next, we integrate the dataset separately and internally if needed as the same type of data could be stored individually based on their utility. Finally, we merge all three datasets into one single, unified, and clean data frame to facilitate the analysis later and in a desired spatial-temporal dimension.

4.1.1. Data Description

This subsection is unfolded with three parts based on the type of data, containing trip planner data, smart card data, and AVL data. We describe the data collection process and the database structure of each dataset.

Trip Planner Data

We present the complete data structure of the 9292 trip planner in Fig. 4.2. 9292 owns five types of data and integrates the five datasets into the trip planner system to provide trip advice, including timetable data, real-time data, geographic data, messages disturbances, and fare. Timetable data represents the timetable of all PT modes in the Netherlands and it is provided by every PT operator in advance. Real-time data contains the live information of vehicles, in terms of temporal and spatial. Geographic data has the location information of stops and lines, generating the street map and transport network. Messages disturbances have certain overlaps with real-time data in the way of disruptions. However, it grants the knowledge with more time in advance and the format of plain text. The last dataset that the trip planner posses is fare data, which can help to estimate the tick price.

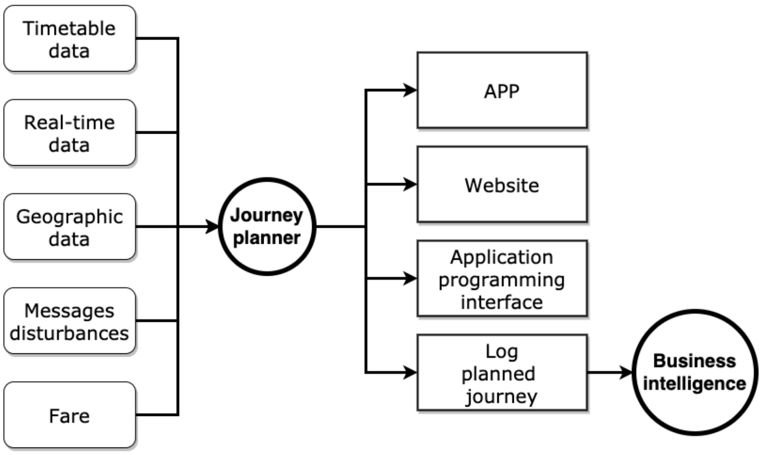


Figure 4.2: Complete database structure of 9292

With the integration of the data, the journey planner [1] could provide the trip advice to its users via different platforms and interfaces. It can be an application installed on Android or IOS or the website of 9292 and the PT operators. Every time 9292 provides a piece of trip advice, it would be logged into the database, maintained and organized by the business intelligence (BI) team for the commercial and analytical usage.

This study utilizes the BI data from 9292 as the trip planner data over the whole of October, which we show in Fig. 4.3. The database contains four parts, namely stops, modality, answer, and question. The primary keys - links - among sub-datasets are marked with color. The link between answer and question is established via question ID, which is a unique ID logged in the dataset. In the question dimension, users could type in the exact address as origin or destination. However, it is not possible to know the user type nor the exact location that passenger inquiries due to privacy issues. It is worthwhile noticing that the way that 9292 names stop is different from that of the OV bureau, i.e. smartcard dataset.

[1] The term of journey planner and trip planner in this study is interchangeable.

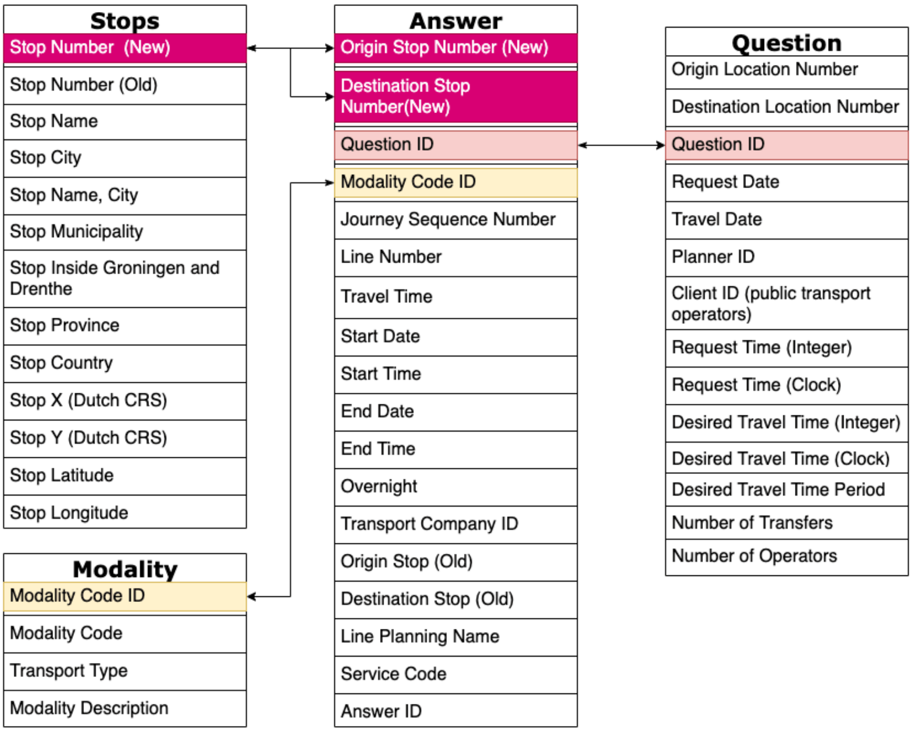


Figure 4.3: Database structure of 9292 business intelligence data

Smart Card Data

In 2012, the public transport sector adopted the Dutch smart card system (OV-chipkaart) as the primary fare system in the Netherlands. Ever since only 5 % of the journeys are made with a paper ticket, and 2 % of the fares are dodged (Van Roosmalen, 2019). It means that smart card data is an accurate recording and the objective reflection of passenger trips. However, the availability of this data is limited and there is a certain storage time due to the privacy issue. Moreover, it is impossible to differentiate the user type or subscription type as traveler IDs are hashed.

In this study, we regard the data from the smart card as the realized demand. It will not only be trained in the prediction model as a variable when applicable and will also be used to validate the result separately and independently. This means several independent variables can be derived from this dataset to investigate the importance, such as the monthly average ridership and the ridership last week on a certain trip.

OV bureau Groningen-Drenthe provides the smart card data of Qbuzz to this study, authorized by Translink (responsible company for processing the transaction data). Normally, the collection cycle of this data is large. It would be first transmitted from the physical smart card to terminals (devices on vehicles or gates in stations) and then forwarded to local storage systems at PT company (e.g. Qbuzz). The local storage systems would gather the data and send it to the central storage system of the PT company. The procedure ends up with keeping the data in the central database of the national data collecting agency, which is Translink in this study. A data structure of smart card data provided is shown in Fig. 4.4. The smart card data is already split into trips, namely a tap-in and tap-out of a single leg of a journey.

Note that the smart card data we used in this study does not contain the information on the type of the smart card. Thus we can not distinguish students, subscription holders, and other travelers. It can be a follow-up study that we will mention in the last chapter. Moreover, smart card data only has the route ID but not the vehicle ID. It means that we have to map the ridership onto the vehicles but not directly linked.

AVL Data

9292 provides the AVL data, which is regarded as the PT supply of this study. It helps to map the

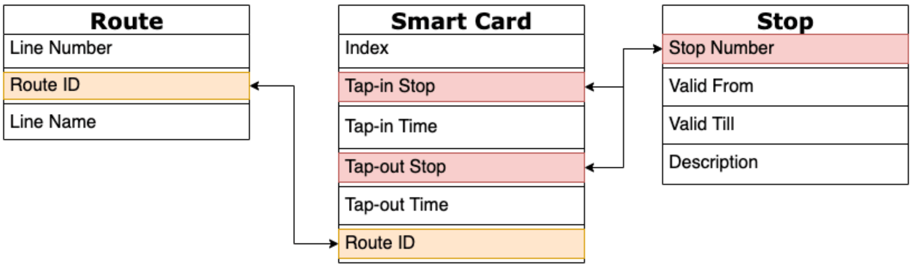


Figure 4.4: Database structure of smart card data

ridership and the travel requests onto the right vehicle at a desired spatial-temporal scope. The dataset contains detailed information about all trips of Qbuzz buses at a vehicle-stop level.

The data structure of AVL is shown in Fig. 4.5. The AVL dataset has the following information: data owner company, line planning number, journey number, operating day, sequence number, type of mode, arrival time, departure time, user stop type, user stop code, and the Dutch coordinate of the stop. The stop code in this data does not directly match the trip planner data in the previous section.

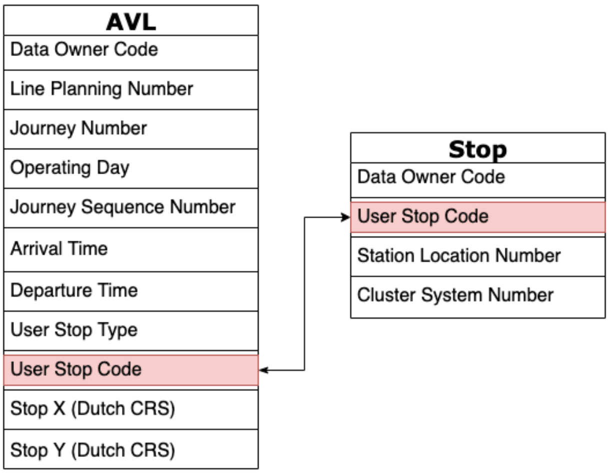


Figure 4.5: Database structure of AVL data

4.1.2. Data Cleaning

The subsection is unfolded with three parts based on the type of data, including trip planner data, smart card data, and AVL data. We present the data cleaning process and the corruptions and inaccurate recordings of each dataset. The technical implementation of the data cleaning process for each dataset is lengthy and therefore we offer visual schematic overviews of the data cleaning and linking process in which we show what steps we execute and how much data is filtered.

Trip Planner Data

Figure 4.6 demonstrates the cleaning process of the trip planner dataset. There are 7,841,229 answers and 5,908,679 questions. Directly merging the answer and question datasets cause crashes of the computer due to a shortage of memory, so developing strategies to deal with the problem is needed: First, we drop the irrelevant columns, namely irrelevant information. Second, we remove the duplicate data stored out for unknown reasons. In this way, the number of answers decreases to 6,048,246, and the number of questions drops to 4,533,563. Then, we can merge the question and answer dataset based on the unique question ID. Next, we merge the question-answer dataset with the modality dataset based on the modality code ID in order to differentiate the bus type.

Moreover, we merge the question-answer-mobility dataset with the stop dataset. We refer to the stop numbers from the answer instead of from questions because users could type in Zip-code or exact location as origin and destination. 9292 stores this type of information as a hashed number to protect

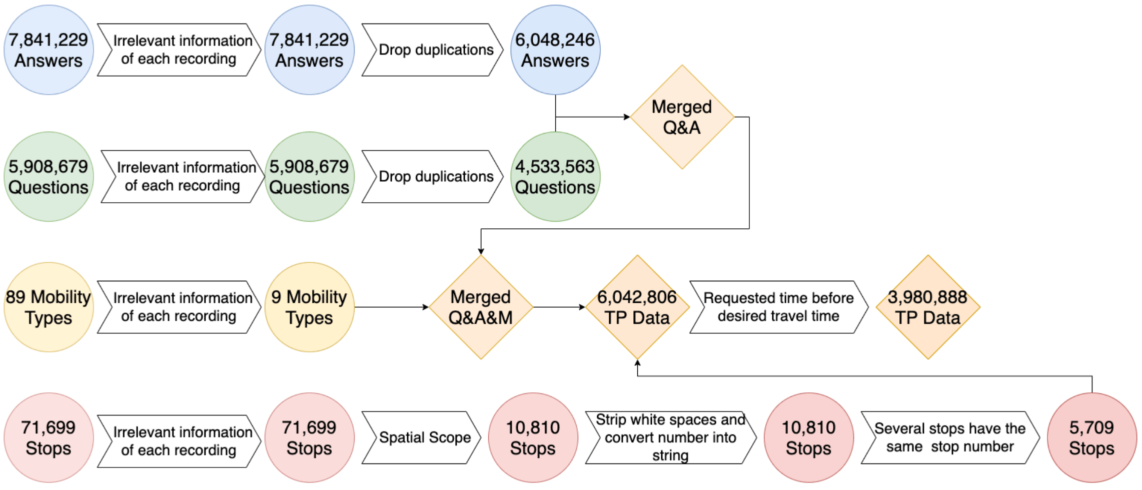


Figure 4.6: Cleaning process of trip planner data

information security and therefore is not traceable and correspondingly inaccurate. However, several stop names could have the same stop number. For instance, there could be a couple of stops with the same stop name around the railway station with different platforms or around an intersection. In this study, the focus is on the occupation rate on board, and hence those locations are regarded and clustered into one unique stop number as the first appearance in alphabetical order. Eventually, 5,709 stops are in the dataset.

The final step of cleaning the trip planner is to select the trip advice that is asked before the trip could be realized, which means that the requested date and time should be before the travel date and time. The reason is bifold: first, the topic is to predict the ridership with trip planner data. If the request is sent after the trip, it becomes meaningless. Second, it helps to filter out the trip purpose of tracing back the journey that has been realized, which means to help filter out the same IP sending the same requests, although not all of them. Finally, there are 3,980,888 rows of data.

Smart Card Data

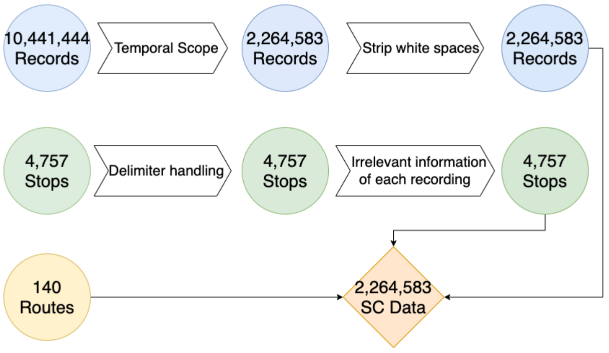


Figure 4.7: Cleaning process of smart card data

Figure 4.7 presents the cleaning process for the smart card dataset. We first filter out the data that are out of the scope. Then, we drop the irrelevant information and reformat the data to facilitate the following study. Altogether, the cleaned smart card dataset has 2,274,195 rows of data.

AVL Data

We describe the data cleaning workflow for AVL data in Fig. 4.8. In the AVL data, each trip of the bus from the same line of the same day is marked with a unique journey number, regardless of the same rolling stock. This journey number is refreshed every day.

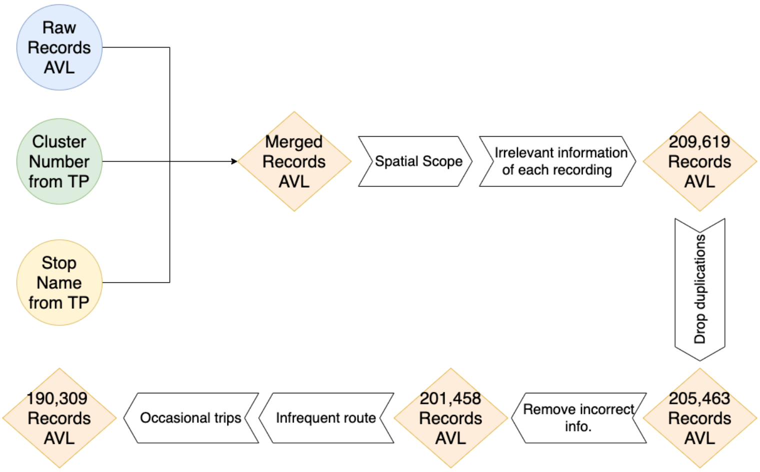


Figure 4.8: Cleaning process of AVL data

We initially filter out the vehicle trajectories that we need. Following, we unify the stop names to help the integration later based on the trip planner data. Then, we leave out the special trips of each route and the comparatively infrequent route options. Normally, the highest frequency option is 20 times higher than other options.

After the cleaning, there are 3992 Qliner 300 trips (bi-directional), 2300 Q-link 1 trips, 927 bus line 50 outbound trips, 923 bus line 50 inbound trips, 912 bus line 35 trips (bi-directional). In addition, we mark the direction of buses with the outbound as “0” while inbound as “1”.

4.1.3. Data Integration

After we describe the structure of databases and execute the cleaning process for each dataset separately, we move to build the link between them. Moreover, we also map the expected ridership from the trip planner data and the realized ridership from the smart card onto the vehicles in this subsection. Note that, as above-mentioned in the smart card data structure, we do not have the vehicle number that is associated with the smart card and thus we have to link the ridership and the vehicles. Figure 4.9 reports the general integration process for the three datasets that we utilize in this study.

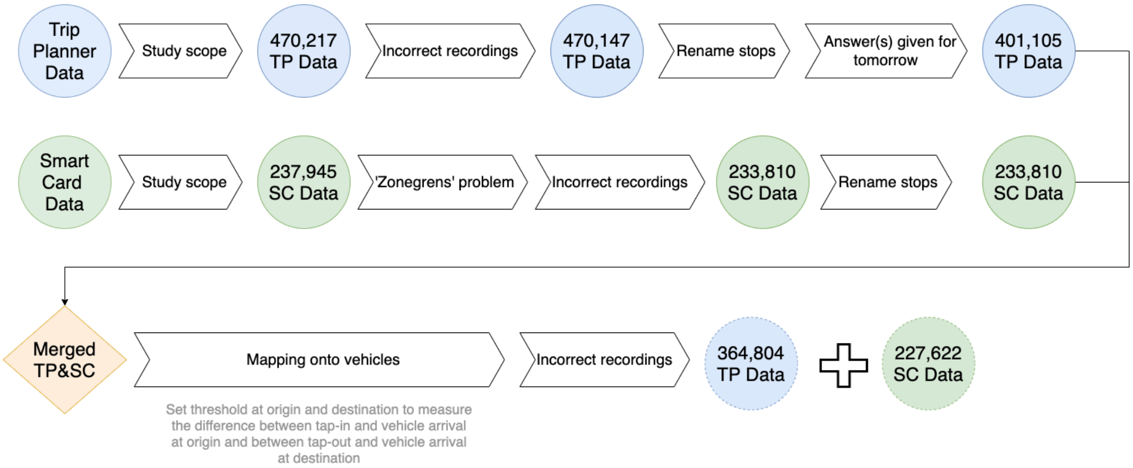


Figure 4.9: Integration process of trip planner, smart card and AVL data

OV-bureau and 9292 have different ways to number the stops, which passes some challenges in this study and implies further collaboration. OV-bureau keeps a record per platform whereas 9292 aggregates these platforms to clusters. Therefore, we establish the connection via stop names. We refer to the stop names from the interactive line map of OV-bureau as the benchmark to establish the parallel comparison.

Trip Planner Data and Smart Card Data

For integrating trip planner data and smart card data, we mainly deal with inaccurate recordings, corruptions, and uninterested shorter routes (infrequent options). During this process, a lot of data are left out due to the issue “Zonegrens+number”[2]. This issue is originated from the financial settlement of certain transactions and can not be traced to a specific existing stop. We recommend further treatment from the OV-bureau side to avoid unnecessary data loss.

After integration, there are 164,379 trip planner data and 78,797 smart card data of Qliner 300, 157,484 trip planner data and 81,527 smart card data of Q-link 1, 117,525 trip planner data and 57,724 smart card data of line 50, and 43,059 trip planner data and 15,777 smart card data of line 35.

Mapping Request and Ridership onto Vehicles

Then, we map requests and ridership onto vehicles. We adopt the strategy of setting a threshold at both ends as the gate machine could be located at the station but not on the vehicles. This threshold is determined based on the headway of that specific line, the frequent the smaller.

Eventually, there are 3% of smart card data left out without a matching vehicle. Moreover, we lose around 20% of trip planner data by applying the same method. It shows a big contrast, compared to the smart card. However, 15% of data are dropped when we set the criteria that we should remove the advice given for tomorrow. Hence, we consider this amount of data loss is acceptable.

4.1.4. Brief Summary

To summarize all the assumptions that we make during the data cleaning and integration, we provide the list of assumptions below:

1. We refer to the interactive map from OV-bureau as the baseline to convert all the stop names in this study into a uniform one. And these stop names are the links between trip planner data, smart card data, and AVL data.

2. Since we are only interested in the trip requests that were asked before the desired travel time, we filter out the requests that have requested time before the desired travel time.

3. Several stops have the same stop number, and we only save the first one appeared in the dataset in alphabetical order.

4. When a bus line has multiple options, we only keep the frequent route into account. Q-link 1 is an exception that we cut the feeder route, converting the longer route to the shorter one as they have almost the same frequency. In this way, we get the best use of our data.

5. We filter out the trip advice given for tomorrow due to the impossible transport supply at night or during weekends. This threshold is set as 5 hours.

6. We filter out the smart card transactions with “Zonegrens” issues as we can not infer the right boarding/alighting stop from the data.

7. When we map the trip planner data and smart card data onto vehicles, we set a threshold at both ends to measure the time difference between vehicle arrival and tap-in time at origin and between vehicle arrival and tap-out time at destination. We set the threshold as 10 minutes for Qliner 300 and 15 minutes for all other case study lines.

4.2. Data Exploration

This section introduces the dimensions that are contained in the trip planner data and smart card data for facilitating the selection of variables for ML. It is unfolded into two parts and begins with the exploration of trip planner data, following the investigation of smart card data.

[2] Dutch word “Zonegrens” means financial zone boundary.

4.2.1. How Informative Is Trip Planner Data?

The analysis of trip planner data begins with the study into how informative this kind of data can be. Intuitively, people plan their trip before they realize the trip. However, some questions arouse our interests. How much time do people plan before they realize the trip? What is the distribution of trip requests per period, per type of the day? How many requests have been realized (namely, the relative value between trip planner requests and smart card transactions)?

Timing Advance for Asking A Piece of Trip Advice

We begin with uncovering the knowledge of the timing advance. We calculate this advance in two ways, namely by using the requested time compared to the desired travel time and the vehicle start time. We focus on the latter one as the vehicle start time is when the passenger will possibly realize the trip while comparing the desired travel time can unveil the user preference in such a trip planner app.

It is worthwhile mentioning again that IP tracing is impossible. Therefore, there are possibilities that a single person could contribute to many requests, which would introduce overestimation to the results. Moreover, we are only interested in the trip requests that are not yet realized, which means people requested travel advice before their desired travel time and the vehicle start time from the trip planner question.

To better present the visualization and unravel the timing advance, we group the time difference in the following order: “[0,10] minutes earlier”, “(10,30] minutes earlier”, “(30,60] minutes earlier”, “(60,120] minutes earlier”, “(120,240] minutes earlier”, “(240,480] minutes earlier”, “more than eight hours but still within a day” and how many day(s) earlier. The top 10 timing advance of whole lines is shown below in Fig. 4.10. On the left is the time difference between requested time and desired time while the difference between requested time and vehicle start time is on the right.

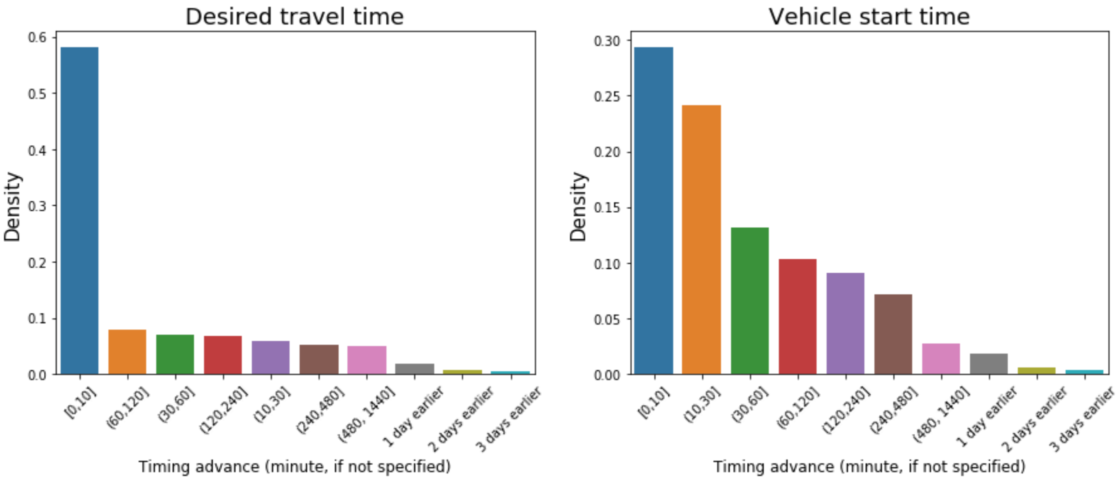


Figure 4.10: Timing advance of trip requests

From the figure, we can see that the range of real-time to 10 minutes for requesting trip advice before realizing the trip is dominant in the timing advance, regardless of whether we can compare the requested travel time with desired travel time or vehicle start time. If we compare with the desired travel time, this margin is dramatic. It means that people generally prefer using a trip planner application like 9292 in time instead of preparation. If we look at the timing advance by comparing vehicle start time, it shows a descending trend along with the increase of the timing advance. Moreover, the gap between the range of real-time to 10 minutes and the range of 10 to 30 minutes is less profound. Still, it is partially due to the increase of the aggregation level. The figure only shows the largest 10 timing advance groups whereas the time advance can be as long as 61 days ahead.

To illustrate the frequency of appearances, we construct the cumulative curve to investigate how often is the timing advance of request per 10 minutes as shown in Fig. 4.11. The cumulative curve truncates at 2 days advance as the curve becomes flat. The dotted line represents the time of one day, equally 1440 minutes.

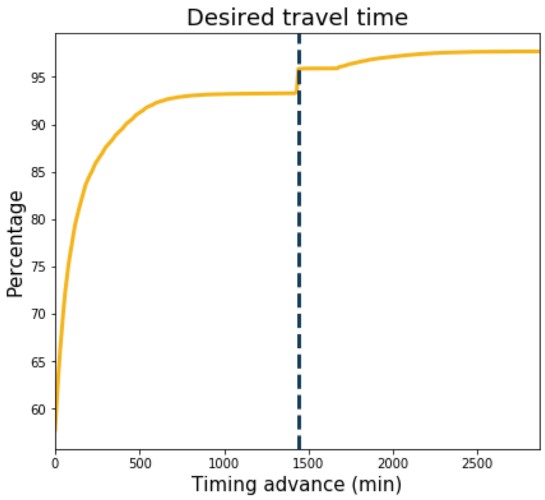


Figure 4.11: Cumulative curve of the timing advance per 10 minutes

From the figure, we can see that most of the trip advice has been given within a day and can be as high as 93.44%, compared to the desired travel time. Besides, if we calculate the timing advance by vehicle start time, 95.91% of the trip advice is asked within a day in advance. It indicates people tend to use such a trip planner within a day in advance.

Then, we adopt the same concept to the case study lines in order to see whether passengers behave differently, according to the feature of the line. The resulted bar charts are shown in Fig. 4.12. In this figure, we use abbreviations for the ticks of the x-axis where only the number of timing advances within a day has been presented in minutes.

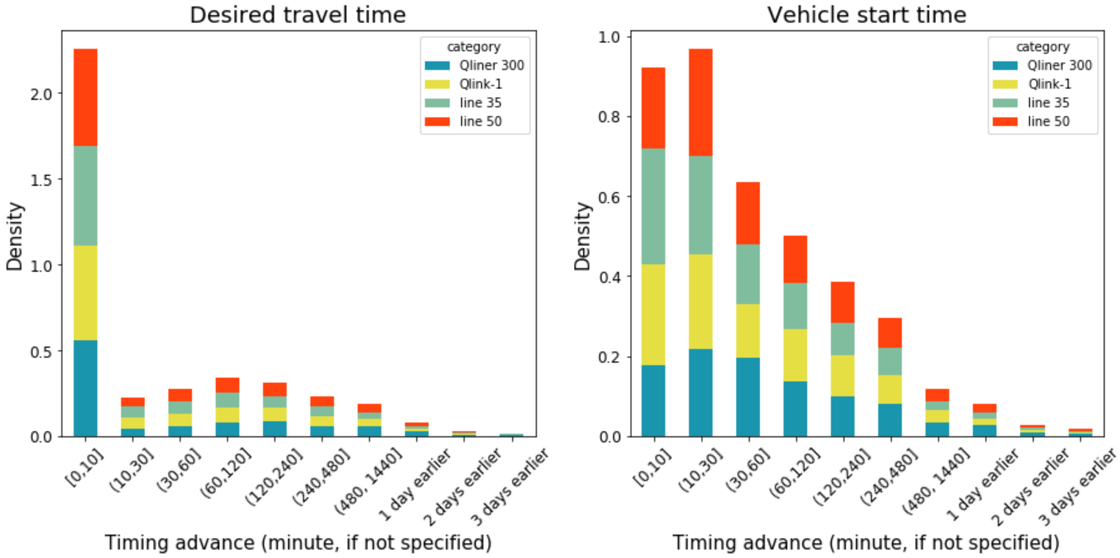


Figure 4.12: Timing advance per case study line

In general, we see a descending trend of the requests with the increase of the prediction lead time in every case study line. If we compare the requested time with the desired travel time, the dominant range is still from real-time to 10-minute advance before traveling in every case study line. However, if we compare to vehicle start time, people would normally prefer asking the travel advice for inter-city lines 10 to 30 minutes before. Following, it is the real-time to 10 minutes. The gap between real-time to 10 minutes and 10 to 30 minutes is subliminal. But, if we are interested in further ahead of time, it becomes larger. It shows that if we only utilize the data further ahead of time, we can potentially encounter a shortage of data.

It is intuitive that during different periods, people behave differently for using such a trip planner. We differentiate the periods of a day and cluster it into four groups with the same time horizon of 6 hours, namely from 4:00 to 10:00 (including morning peak), 10:00 to 16:00 (containing off-peak hours), 16:00 to 22:00 (taking evening peak into account), and 22:00 to 4:00 (considering the before sleep plans).

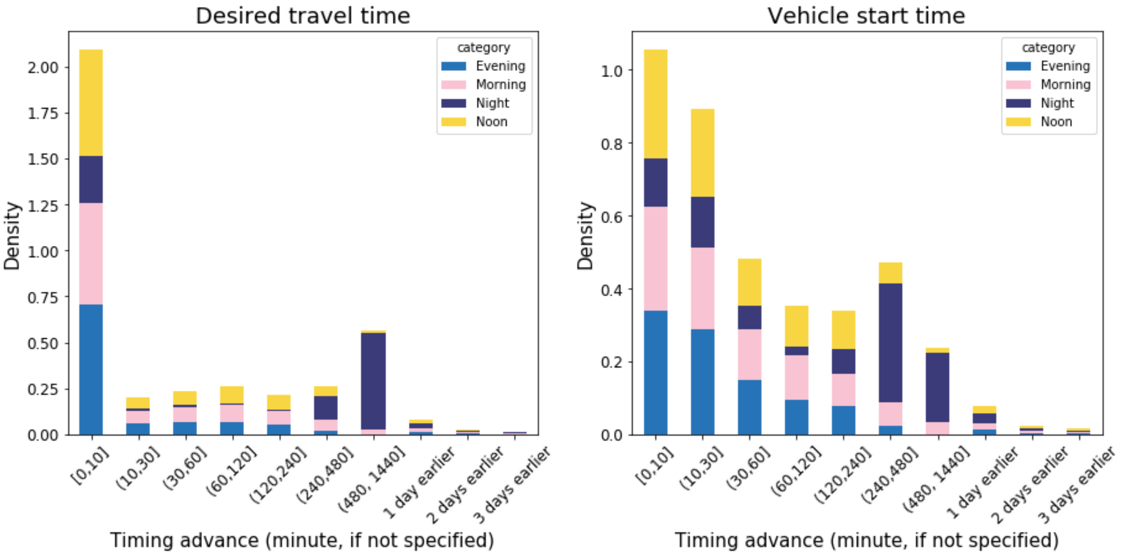


Figure 4.13: Comparison of timing advance per time period

Figure 4.13 shows the number of requests per period. It presents that people behave differently in the night, compared to the other three time periods. Compared to the desired travel time, the most preferable timing advance at night is from 8 hours to a day, which refers to the trip tomorrow. Then, the range from 0 to 10 minutes follows it. Compared to vehicle start time, people prefer preparing from 4 hours to 8 hours in advance at night while the range of 8 hours to a day follows it. Intriguingly, the trip purpose of the daytime, from 10:00 to 16:00, is supposed to be different from peak hours while it does not show a different pattern. This reflects that the aim of using such a trip planner varies over the whole day. The difference between desired travel time and vehicle start time lies in the availability of trips.

Therefore, we conclude that people prefer checking travel information in 0 to 10 minutes of short-time range during the daytime. In contrast, people would like to plan their trip at least 8 hours before night. Although if we can predict the ridership further ahead of time can grant more flexibilities to transit operators, the data analysis, unfortunately, reveals a real-time oriented usage, which implies a shortage of data. However, if we compare to the vehicle start time, there is still ample (roughly half) data that we can utilize if we are interested 30 minutes in advance. Therefore, we will include variables in the following chapter, such as all trip planner requests, 10-minute ahead, 15-minute ahead, and 30-minute ahead to investigate how the model would develop in the later chapters.

Distribution of Trip Planner Requests on An Average Day

Next, we derive the distribution of time requests over an average day and for different types of the average day. Since 0 to 10 minutes advance are favored by travelers, we set the time aggregation of 10 minutes. Figure 4.14 shows the distribution of time requests of all lines per 10-minute time aggregation.

From the figure, we know that the trend of requested travel is flatter than that of the desired travel and the vehicle start time. We notice there are spikes in the distribution of desired travel and the vehicle start time that are always on a rounded-up minute (namely, 10, 20, 30, etc.). The reasons are trifold. First, passengers often prefer a rounded up number as their departure time. Second, some clients (PT operators) of 9292 provide departure time options with a quinary system (whenever meets 5 round-up). Third, there is a timetable of the bus operation, and the departure time is usually quinary. Generally, the distribution is a bimodal distribution with one peak during the morning peak and the other one

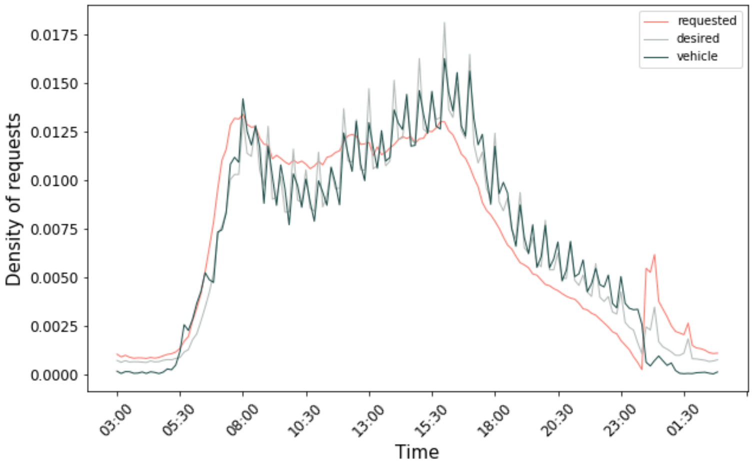


Figure 4.14: Distribution of requests per 10 minutes on an average day

in the evening. Besides, we can see that both the requested and the desired travel time during the evening peak is comparatively high due to the non-uniform off-duty time. Lastly, there are still quite a few requests at mid-night, compared to the early morning. Notably, there is a spike around 00:00.

Table 4.1: Percentage of timing advance (compared to desired travel time) from 00:00 to 00:09 of all lines

Timing Advance Group (minute, if not specified)	(480,1440]	(240,480]	1 day earlier	2 days earlier	3 days earlier
Percentage	74.389%	15.845%	5.468%	1.357%	0.688%

Generally, people plan their trip before they go to sleep, which would result in the spike. However, bedtime spreads out among people. Hence, we take the request sent during the time aggregation from 00:00 to 00:09 to investigate the reason. Table 4.1 reports the percentage of timing advance from 00:00 to 00:09, compared to the desired travel time in order to unveil the preference. The majority of the people who use the trip planner during midnight (from 00:00 to 00:09) would plan their trip for the next day, namely 8 hours to 24 hours ahead. Following the group plan between 6 to 8 hours ahead, still can be regarded as planning the trip for tomorrow. Thus, we conclude that the spike during the night is due to the planning for tomorrow at a rather dense and interesting mid-night time.

Then, we derive the distribution of requests per 10-minute on an average day from our case study lines, shown in Fig. 4.15. The oscillation of requests based on vehicle start time is because the operation of the bus has a timetable, and we aggregate the requests per 10 minutes. The mismatch of the time could lead to this oscillation. In terms of the requested time, it is obvious to see that Qliner 300 and Q-link 1 have a relatively flat trend during the daytime, compared to bus lines 50 and line 35 whose fluctuations are much more visible. It is probably due to the frequency of Qliner 300 and Q-link 1 is higher than other bus lines. Plus, people would like to check transit information in time. The bimodal distribution is much more apparent in normal bus lines with two clear peaks, one in the morning and the other one in the evening. The two peaks are significantly evident in bus line 35 that connects the city to a town. Throughout all case study lines, it seems that the morning peak is in line with the conventional understanding that is from 07:00 to 8:30 while the evening peak is left-shifted.

Furthermore, we explore whether different types of the day would impose an impact on the number of requests. We distinguish the average number of requests on an averagely regular weekday, an average day of the weekend, and an average day of the autumn holiday to see such an impact. At this step, we only consider the timestamp from the desired travel time because this is when passengers want to travel. Correspondingly, it reflects the behavior. Moreover, only case study lines are interested at this stage, shown in Fig. 4.16.

Figure 4.16 presents us with the influence of the day type. In every sub-graph, the orange curve represents the number of requests on an averagely regular working day, the green line indicates the requests sent on an average autumn holiday, and the blue line means the number on an average day during the weekend, respectively. In general, we see a shrinking pattern of the number on holidays,

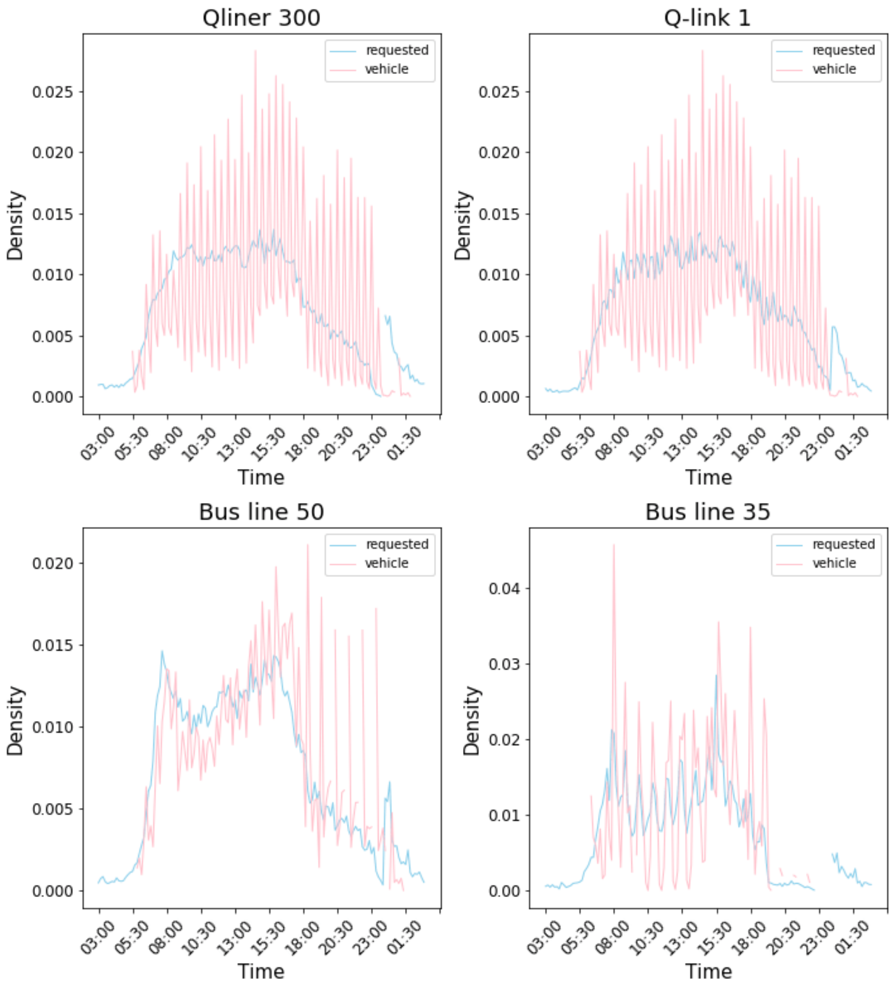


Figure 4.15: Distribution of requests per 10 minutes per line

compared to a regular working day. It is because students, teachers, and all other education-related jobs have almost no demand for commuting. In contrast, the pattern during the weekend shows a dramatic difference. Not only the number of requests drop notably, but also the peak hours are different. Every line shows a bimodal distribution of requests on a weekday, and uniform distribution over the daytime on the weekend, except for Q-link 1. On the contrary, Q-link 1 has an increasing trend of requests on each type of day with the difference that the peak reaches late on the weekend. This is due to the characteristics of the line as it is traversing within a major city and offers accessibility for several vital locations. For an average weekday, requests for asking a home-bound off-duty trip are dominant while a home-bound entertainment-ended trip is possibly popular during the weekend. To conclude, the type of day can influence evidently on the number of requests, and both weekends and holidays impose a negative effect on the number.

Relative Value between Trip Planner and Smart Card Data

The last point in this subsection is to deal with the relative value between the number of trip planner data and smart card data, namely the realized trip. We desire to generally understand the relationship between trip planner and smart card data and the variance of such a relationship over an average day. The relative value is calculated as Eq. 4.1 shows:

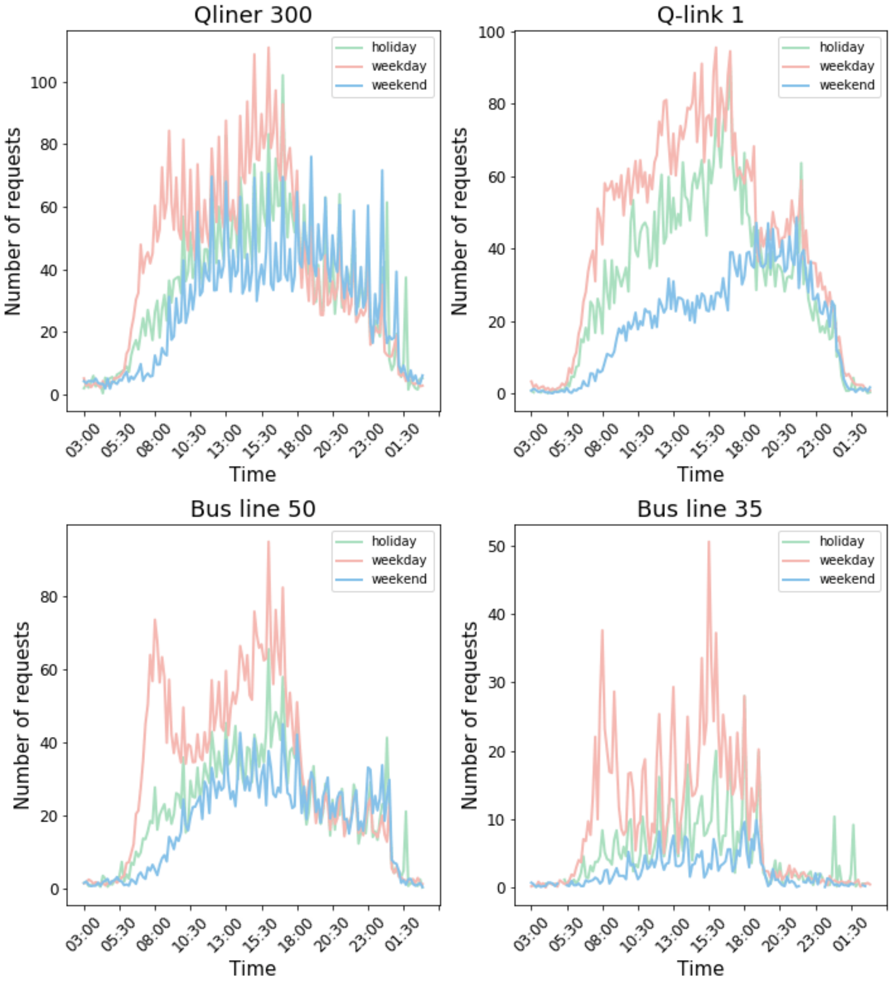


Figure 4.16: Distribution of requests per 10 minutes per day type per line

$$\delta = \frac{n_{trip\,planner,t}}{n_{smart\,card,t}} \tag{4.1}$$

where, $n_{trip\,planner,t}$ represents the number of trip planner requests during a certain span while $n_{smart\,card,t}$ is the number of smart card data during that span, namely ridership. The t at this subsection is set as 10 minutes.

Since the objective is to investigate the realization of the trips, we take the multiple legs of a trip into account when it is possible as the transactions of smart card data are split into legs, although this could bring in overestimation. We keep the temporal aggregation of 10 minutes as this is the preferred timing advance by travelers. Again, we use the desired travel time from trip planner requests in this subsection since it is when the passengers want to travel.

Figure 4.17 shows the relative value distribution of all lines between trip planner requests and smart card transactions over the whole of October. The left part of Fig. 4.17 shows the entire day while the right truncates the time before 06:00 and after 23:30, focusing on the daytime when values are at a similar level. The left part of Fig. 4.17 shows that the recordings of the trip planner are unrealistic in the early morning, which can be as high as more than 800 times smart card transactions because there is almost no bus service during that period. During the daytime, the trips are realized more than in the evening, and it becomes more and more unrealized when it is approaching night. One of the reasons for this exaggeration in the evening could be the requests for tomorrow or future trip(s). Another reason

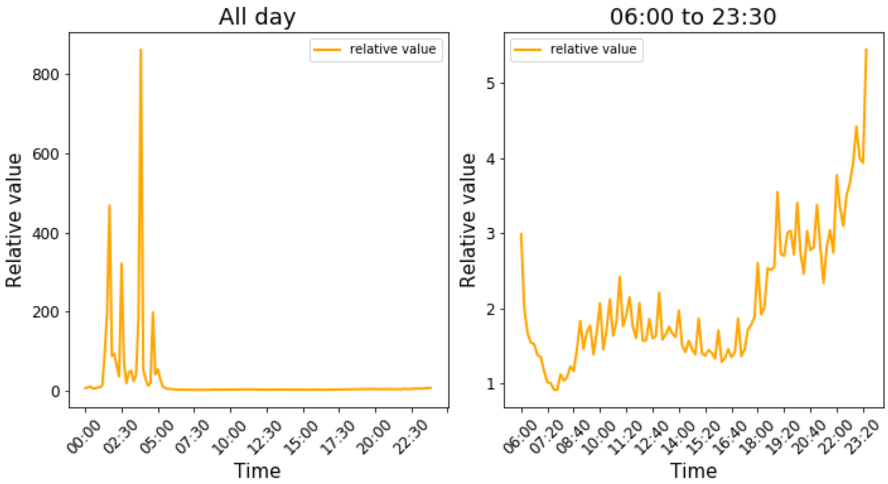


Figure 4.17: Relative value per 10 minutes on an average day

could be the headway of services becomes longer during the night, and consequently, it is harder to have a supply for the desired travel time. If we laterally compare the number of requests and the relative value, we can draw the conclusion that when the number of requests is large, the relative value tends to be small as more travels have been realized.

Following, we consider the relative value of case study lines in Fig. 4.18 in which we set the boundary from 06:00 to 19:30. From the figure, we notice that there are eye-catching blanks in every graph. It is possibly due to the non-operational hours. In general, the trend of Qliner 300 and Q-link 1 match that of all lines because the number of recordings is large. The bimodal distribution is more evident in the truncated curve of Qliner 300 and Q-link 1 due to their high frequency, but not in regular bus lines. In contrast, the fluctuations of bus line 50 (up to 250 times) and 35 (up to 50 times) are higher during the daytime. It shows that when the headway of a line is short, the realization of trips is correspondingly high. Additionally, the span of magnification is lower of Q-link 1 compared to other lines because it has constant passenger traffic as it traverses within the city.

Thus, the variance of case study lines is generally high during the night and sometimes for a specific line during the daytime. It could incur troubles when we build the model to capture the relationship between smart card data (ridership) and trip planner data (travel advice) because the model essentially is to learn a rule between them. Suppose the fluctuation is huge, it will be hard to capture the rule. However, for the majority time over the whole of the case study temporal span, it is comparably regular. Therefore, we can expect the model performs better when the realization of trips is high.

4.2.2. Smart Card Data

In this sub-section, we analyze the smart card dataset with the aim of finding out what are the dimensions in this type of data. We still keep a 10-minute aggregation level in order to make sure that it remains consistent with trip planner data. First, we elaborate on the dataset with all lines included and then we narrow down our scope to our case study lines. Second, we illustrate the ridership spatially by revealing the ridership at the stop level and measure the ridership with the crowdedness level.

Temporal Influence on Ridership

We aggregate the smart card transactions per 10-minute and present the distribution per temporal group of an average day in Fig. 4.19. In this figure, we exhaust two different cases, including the comparison among an averagely regular working day, an average day during the weekend, and an average day of the autumn holiday (left part) and the comparison among average days of the week (right side).

First, we investigate the influence of the day type. Note that we exclude the working days during the autumn holiday and average the transactions by day. However, we still include the weekend during

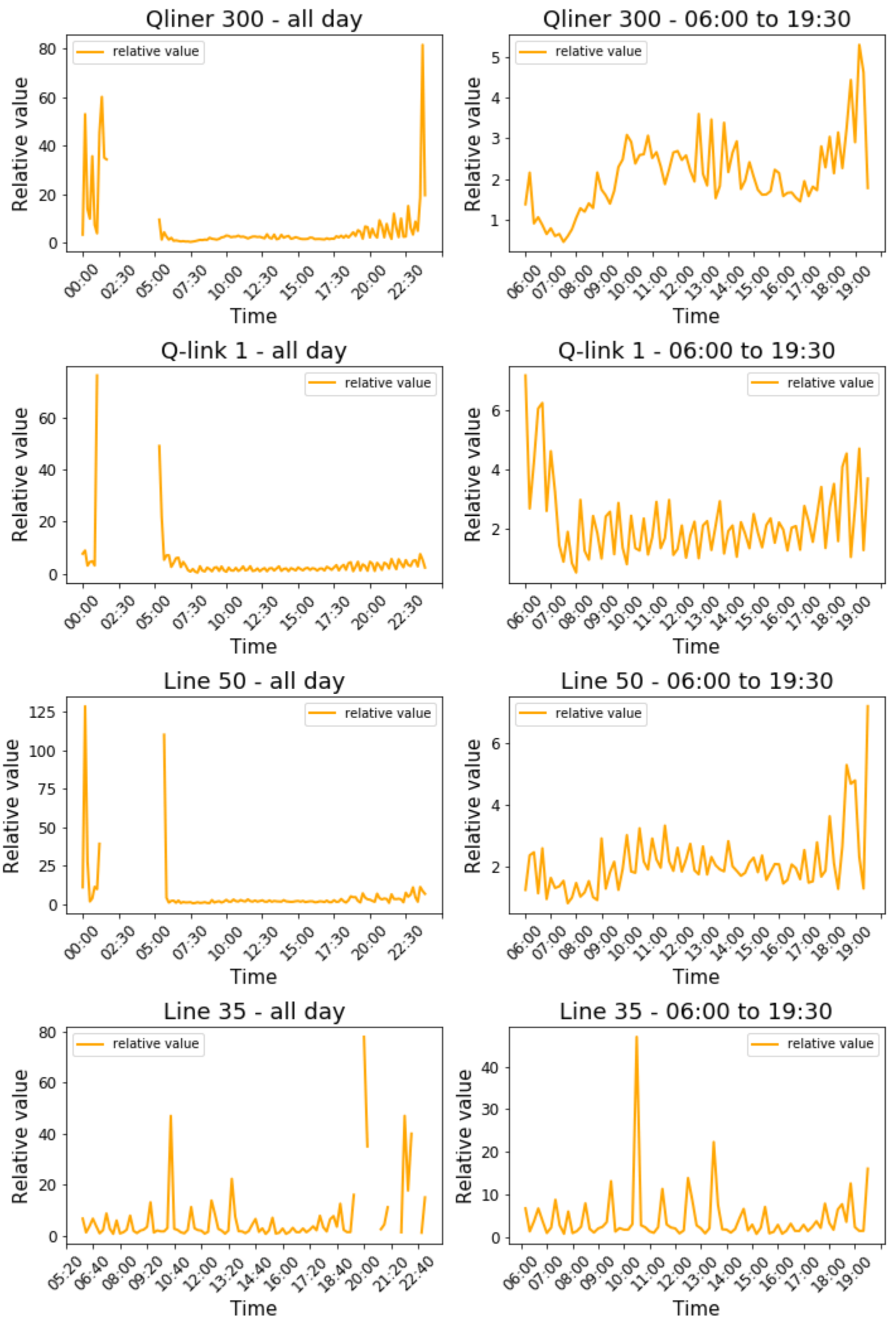


Figure 4.18: Relative value per 10 minutes per line

that holiday as the impact would be negligible. From the overall distribution shown in the left part of Fig. 4.19 with blue color, we can see a bimodal distribution with two peaks. It is different from the distribution of travel requests that shows a flat curve between morning and evening peak. The margin of requests between peak and off-peak is not as significant as that of the smart card. We then take the autumn holiday into account that is unique during this period when students and teachers have a week off. As mentioned in the literature review, the influence of the holiday can be substantial, which is testified by the left part of Fig. 4.19. During the morning peak, we see almost a five-time drop on the transactions,

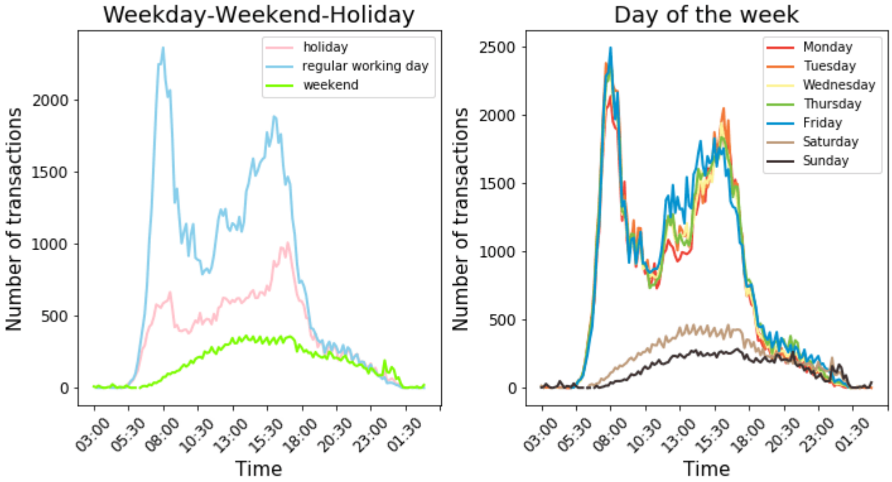


Figure 4.19: Distribution of smart card transactions per group on an average day

and it is nearly one-third of an averagely regular day during the evening peak. As for off-peak daytime hours, the decrease is only two times. Next, we uncover the impact of ridership by day of the week as commuters almost do not make trips during the weekend, and hence the travel purpose and pattern should be contrasting with the weekdays. Weekday and weekend show a completely different pattern where the weekend only has one peak in the afternoon, and the total travel demand is lower. Lastly, we are interested in that if every day of the week has a particular influence on the validate ridership. Again, we exclude the working day during holidays that could introduce errors. As shown in the right part of Figure 4.19, the difference over the day of the week is not apparent. But the attributes of weekday or weekend are obvious.

Second, we assess the impact of the temporal factors on the case study lines, shown in Fig. 4.20. In general, we see a similar pattern of ridership over the day as the overall pattern. Distributions of an average holiday and an average weekday have the same trend as that of Fig. 4.19. However, distributions of an average weekend and an average weekday present a different pattern plus a lower number of transactions. The ridership during an average weekend always starts from the middle of the day. Also, we see every graph has many spikes with the only difference of the span. It is due to the headway, and for Qliner 300 and Q-link 1, the headway is short, and therefore the graph tends to be densely multi-peak while the spikes are much looser of other lines as the headway is longer.

Additionally, for Qliner 300 and bus line 50, those two inter-city lines have an identical morning peak starting at around 7:00 and an evening peak at approximately 15:30. The number of transactions during an average holiday is almost 5-time fewer than an averagely regular day. Interestingly, the margin between an average holiday and an average working day is less profound of Q-link 1 with a near 2-time decrease of the transactions. The reason could be the line connects the residential area with working places, and it is still busy as only students and education-related jobs are having that holiday. As for bus line 35, it does not operate on Sundays, and thus the ridership on an average day of the weekend only represents the average value of Saturdays. From the chart, we can see that the difference between an average weekday and an average off-duty day is the most significant of line 35 since it connects a town with Groningen, offering more functionalities in residence.

To conclude, we define the morning peak from 7:00 to 8:30, and the evening peak from 15:30 to 17:00 on a weekday. Moreover, the study takes the tap-in time to measure if a passenger boards during peak hours or not. Regarding how to define a bus trip is during peak hours or not, we take the departure time of a bus at the stop to match the boundary of peak-off-peak.

Spatial Influence on Ridership

Finally, we unveil the spatial dimensions that exist in the realized ridership per stop per trip during different periods of case study lines, which are ridership profile shown in Fig. 4.21, Fig. 4.22, Fig. 4.23,

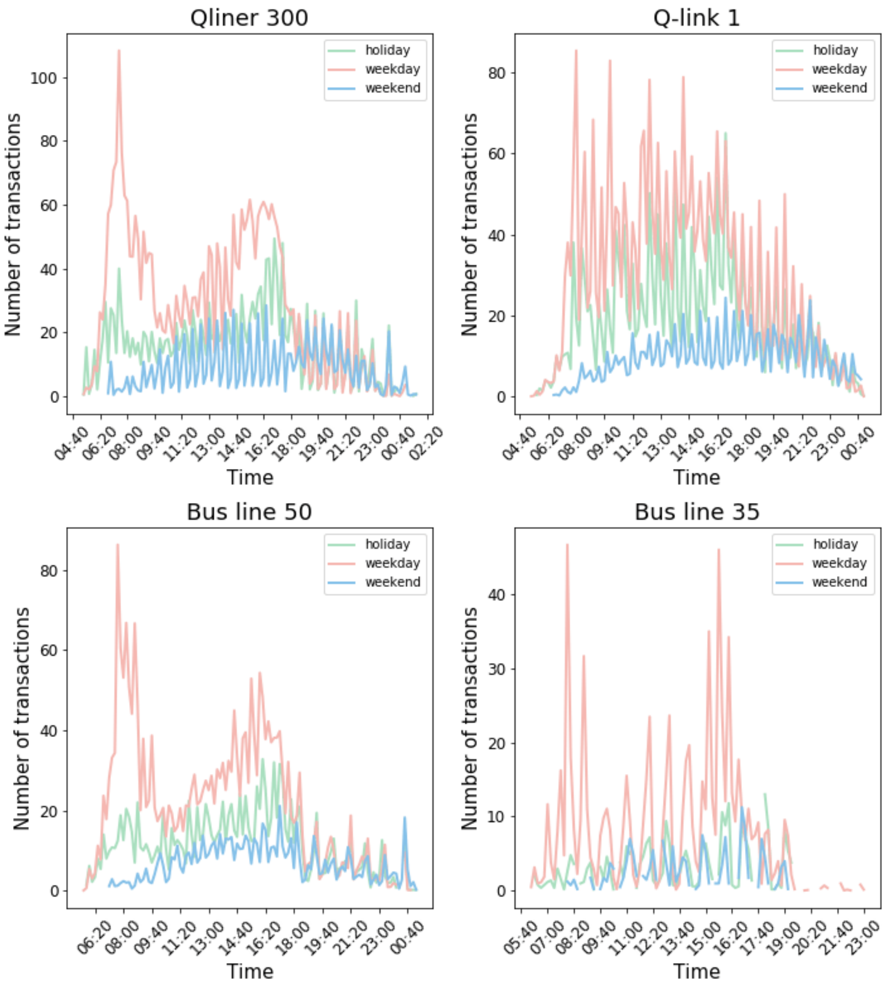


Figure 4.20: Distribution of smart card transactions per day type per line

and Fig 4.24, respectively. In each graph, we look closely into the different periods, including weekday peak hours, holidays, and other off-peak hours. To calculate the average number of ridership on trips, we count the number of trips during each period and use it as the denominator for the ridership during the corresponding period. Moreover, we also take the busiest three trips into account and add them to the graph to show the severity of crowdedness in some extreme circumstances. The grey dot line in each graph is the seating capacity of the most common vehicle in the fleet of that line, according to Table. 3.1. Besides, we present the statistics on the number of trips per line during each period in Table. 4.2. Moreover, we distinguish the outbound trip and the inbound trip and present it on the left part and the right side, respectively.

Table 4.2: Frequency of case study lines per period

Time Period	QLiner 300	Q-link 1	Line 50	Line 35
Autumn Holiday (Outbound)	310	190	155	70
Autumn Holiday (Inbound)	310	190	155	70
Weekday Peak Hours (Outbound)	324	108	126	108
Weekday Peak Hours (Inbound)	396	108	144	90
Off-Peak Hours (Outbound)	1362	852	646	278
Off-Peak Hours (Inbound)	1290	852	624	296
Total	3992	2300	1850	912

In general, the outbound and inbound ridership per stop of each line show an asymmetric pattern, which means that the busy common corridor is the same, no matter the direction of the line. Those common corridors are with a railway station, a P+R stop, a working area, or a city center.

It is interesting to see that when we average the ridership by frequency, the passengers on board are always lower than the seating capacity, even during peak hours. However, if we focus on the busiest trips, the ridership at the busy corridor is more than the seating capacity, which indicates an unsatisfied level of service. For Q-link 1 and line 50, the crowded trips can load as many as approximately two times the seating capacity, showing a closed-packed state and implying safety issues. The direction of outbound is always from Groningen city towards elsewhere. It shows that people flow out of Groningen with a decreasing trend whereas, for inbound trips, it is the opposite.

The ridership profile of Q-liner 300 and line 50 (two inter-city lines) is similar to stairs as it either starts from Groningen railway station (outbound) or ends at it (inbound). And this is where people board and alight the most. When it comes to Q-link 1 or line 35, as they connect a couple of vital points in Groningen city after they depart the railway station or when they enter the city, the pattern shows a unimodal distribution where the peak depends on the appearance of Groningen city center. It means a high attraction of Groningen city.

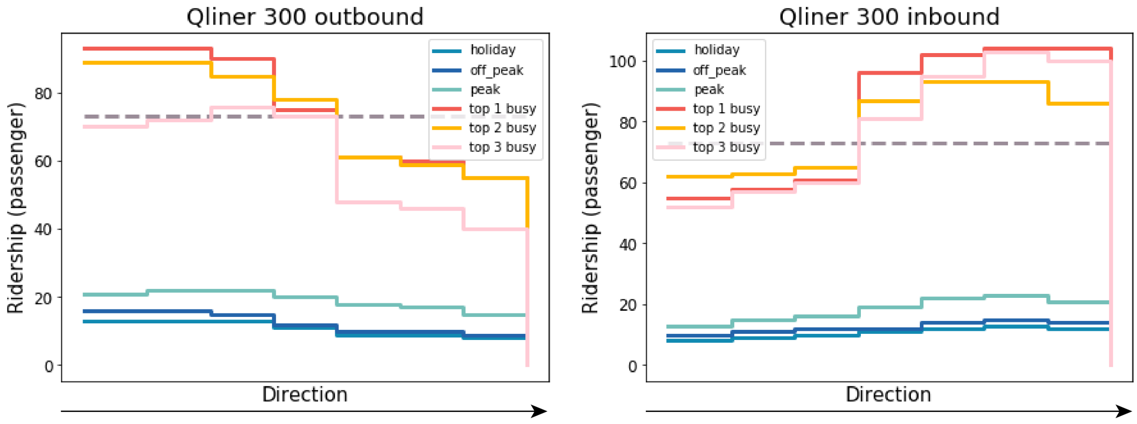


Figure 4.21: Ridership profile of Qliner 300

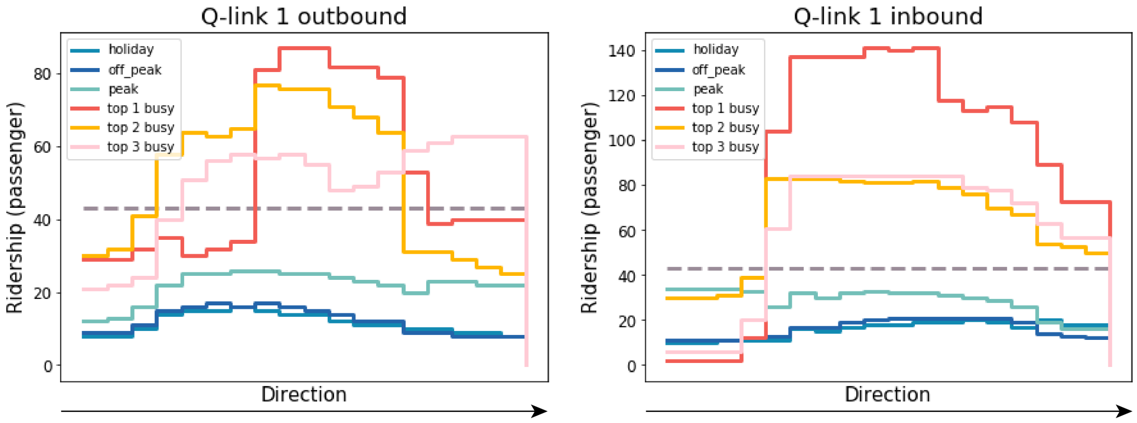


Figure 4.22: Ridership profile of Q-link 1

In short, the average ridership of trips on each case study line is manageable. But, it is the busy trips during peak hours that lower the level of service. As expected, the spatial characteristic is considerable. Around high-volume locations, there are more passengers, and Groningen, as a major city in the case study region, attracts a sizable number of commuting passengers.

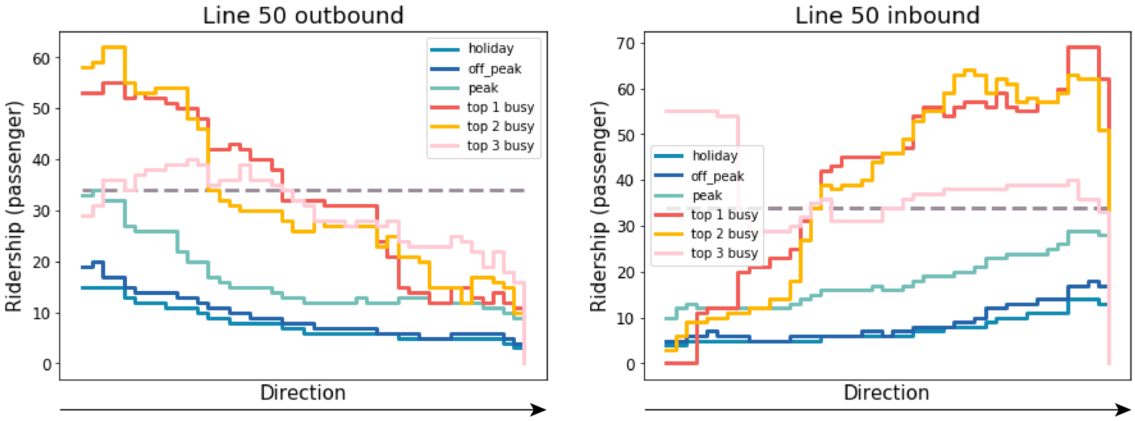


Figure 4.23: Ridership profile of bus line 50

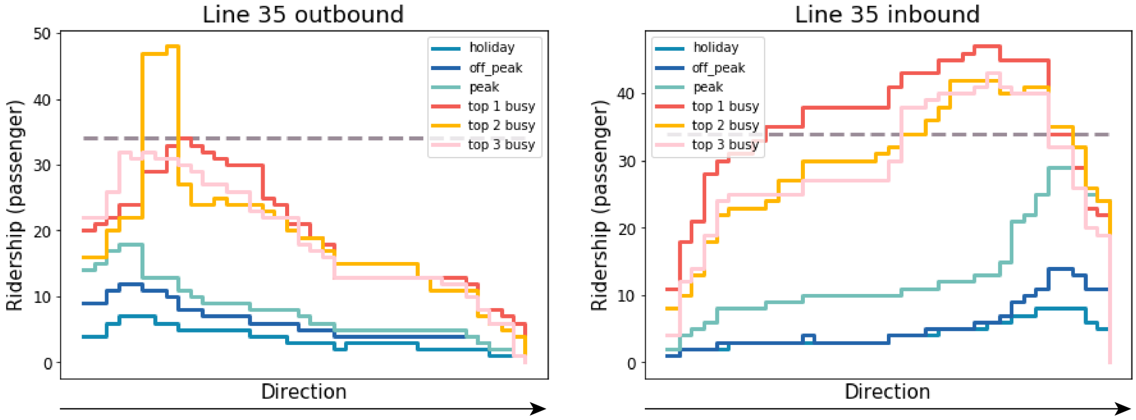


Figure 4.24: Ridership profile of bus line 35

4.3. The Relationship between Trip Planner Requests and Smart Card Transactions

In this section, we explore the trip planner and smart card data collectively to unveil the relationship between two types of data. The first subsection reveals the joint distribution of two kinds of data in different spatial-temporal scope intending to have a quick scan of the correlation between them. Then, the second section derives the correlation between them with all other variables enclosed. In this way, we can select the strongly correlated variables with ridership to construct the machine learning models.

4.3.1. The Joint Distribution of Trip Planner Data and Smart Card Transactions

To unravel the relationship between trip planner data and smart card transactions, we intend to explore the data in two temporal dimensions, including day-level and trip-level. The reasons are bifold: first, we only have the first answer from the trip planner data, and therefore it is unlikely for a person to alter his/her trip decision at a day-level, but it is easier to change at a trip-level. It means that the passenger could turn to the second or other suggestion from the trip planner and thus result in bias. Second, if we can have a general understanding of the type of relationship, for example, linear or non-linear, it can facilitate bridging between them.

Joint Distribution at Day-Level

To begin with, we present the distribution of transactions and requests at the day-level shown below in Fig. 4.25. From the figure, we can see that the two types of distribution have roughly the same trend at the day-level, which means that when the requests of a line are high, the realized trips are correspondingly high. Note that there are no buses of line 35 on Sundays, and thus there are no recordings. Moreover, we see a seasonal trend from the distributions where the weekdays have a high volume while the weekends are lower. There is a decrease in number not only in passengers but also requests from 21th to 25th when the autumn holiday takes place, which again testifies the influence of holidays. Interestingly, we also see a spike of all lines every Friday, which shows high demand for traveling on that day.

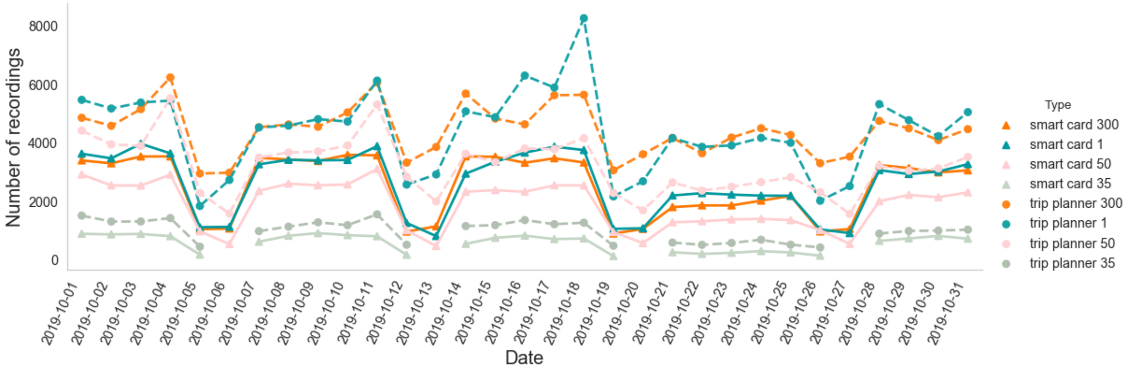


Figure 4.25: Distribution of transactions and requests (desired travel time) at day-level

On 18th October, there is a significant outlier of Q-link 1 requests. We try to figure out the reason by investigating the distribution of requests on that specific day. It turns out that the number of requests during the afternoon (from 12:00 to 18:00) is much higher compared to other normal Fridays. Besides, there are no associative disruptions that happened in this region from the archive. Thus, we conclude this is the influence of pre-holiday emotion and intent to include this outlier inside the dataset to test how well the model can be to deal with such an outlier practically.

Joint Distribution at Stop-Level

Then, we narrow down our scope into stop-level that we have mapped both transaction data and request data onto a specific vehicle trip. We directly explore the ridership of a section, instead of investigating the ridership by inferring boarding and alighting passengers. Note that we do not aggregate any spatial-temporal scope, and each dot represents the ridership of a section of a trip.

We derive a scatter plot at this level shown below in Fig. 4.26, where each dot represents a recording of trip planner requests as y-axis value and smart card transaction as x-axis value. The dashed line assumes a linear relationship between requests and transactions. If the cloud of dots is closer to the line, it indicates a linear relationship between them. However, we can see that although both data go in the same direction, they are scattering along the line, and there are several extreme values far from the line. In general, there are more dots above the line, which means that the requests are larger than the ridership, i.e. the realized trips. When the dots are approaching the high-value range, the margin becomes larger. Therefore, we conclude that a linear relationship between them is hard to find.

Moreover, we can get a hint that both data are right-skewed, which means that we have an imbalanced data. It could pass challenges when we want to capture the rarest and relevant cases equally as the majority. This kind of issue is wildly prevalent in many domains, framed within predictive tasks, and we will provide solutions in the later chapter.

To summarize, the relationship becomes weaker if we dive into the detailed dimension. Additionally, we discover no linear relationship between those two data at stop-level, which means that a linear model is hard to be successful.

4.3.2. Pre-processing for Machine Learning: from Covariance Matrix to Ridership Prediction

In the first place, we introduce some basic knowledge of variance and covariance, which is in the text box below.

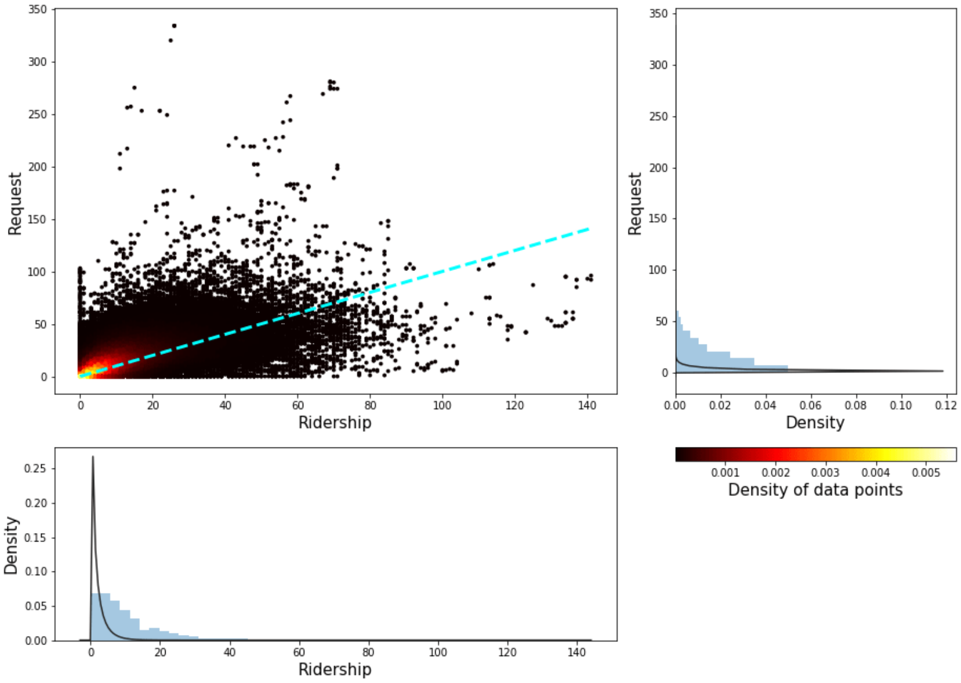


Figure 4.26: Distribution of transactions and requests (desired travel time) at stop-level

The variance of a variable describes how much the values are spread and deviated from its mean.

The covariance is a measure that jointly considers two variables at a time and tells the amount of dependency between two variables. A positive covariance value means that the two variables increase and decrease at the same time, namely when the value of the first variable is high, the value of the second variable will be correspondingly large. While a negative covariance value represents the opposite that the change of two variables values goes the opposite direction, namely when the value of the first variable is high, the value of the second variable will be small. The covariance value depends on the scale of the variable, so it is hard to analyze it. It is possible to use the correlation coefficient that is easier to interpret. The correlation coefficient is normalized covariance.

A covariance matrix is a matrix that summarizes the variances and covariances of a set of vectors where the diagonal corresponds to the variance of each vector.

We list all variables considered in the variance-covariance matrix as following in Table. 4.3. The information in Table. 4.3 includes the name of the variable, the explanation of the variable, the temporal dimension focused, the category that the data is classified, and the unit or coding we have applied. The inclusion of the variable section not only helps us to track the ridership and fulfills the mapping at the stop-level but also tries to reveal the spatial importance that could influence the ridership. However, for four case study lines, there are 204 sections coded as dummy variables[3], and thus it is hard to visualize in the following graphs. Finally, as above-mentioned, the day of the week is meaningful when we see it as a weekday or weekend but less insightful when we dive into each day of the week.

We present the variance-covariance matrix in Fig. 4.27. The top left corner represents the variance of ridership, which is the target we want to predict. This variance is as high as 125.42, which means that there is a large span in the value of ridership. It correspondingly shows that the prediction is valuable.

Since we are predicting the ridership, the variables related to ridership are of most interest, and hence the first row of the heatmap is the most important. These variables associated with the target

[3] One-hot encoding: when it is true, then it is coded as 1 and coded as 0 when it is the other way around.

Table 4.3: Information of variables considered in the variance-covariance matrix

Variable	Explanation	Dimension	Category	Unit/Coding
Ridership	The passengers on board	Day & Stop	Numerical	Person
Ridership_mean	The historical average of ridership	Day & Stop	Numerical	Person
Holiday	The autumn holiday	Day & Stop	Categorical	One-Hot Encoding
Request^1	The trip requests	Day & Stop	Numerical	Record
Request_mean	The historical average of trip requests	Day & Stop	Numerical	Record
Request_var^1	The variance of trip requests, compared to average	Day & Stop	Numerical	Record
Day_of_week	Weekday or Weekend	Day & Stop	Categorical	One-hot encoding
Line	The case study line	Day & Stop	Categorical	One-hot encoding
Section	The section that a vehicle traverses during a trip	Stop	Categorical	One-hot encoding
Direction	The direction of the trip	Stop	Categorical	One-hot encoding
Period	Peak or off-peak hour	Stop	Categorical	One-hot encoding
Ridership_last_week	The passengers on board of the same trip last week	Stop	Numerical	Person
Request_10^1	The trip requests that sent 10 minuets ahead	Stop	Numerical	Record
Request_15^1	The trip requests that sent 15 minuets ahead	Stop	Numerical	Record
Request_30^1	The trip requests that sent 30 minuets ahead	Stop	Numerical	Record
Request_var_10^1	The variance of trip requests that sent 10 minuets ahead	Stop	Numerical	Record
Request_var_15^1	The variance of trip requests that sent 15 minuets ahead	Stop	Numerical	Record
Request_var_30^1	The variance of trip requests that sent 30 minuets ahead	Stop	Numerical	Record

^{1}: We calculate the timing advance by vehicle start time. Variables with this symbol will be put into models in pairs. For instance, request and request_var would be a pair to see how the model performs when we have all the trip planner available, while other variables with timing advance would test how the model performs with fewer data and when the prediction is made further ahead in time.

are also called predictor variables. A high variation in them leads to greater precision of the parameters of a regression model. For example, we are most interested in the relationship between ridership and request, and the variation of requests is as high as 240.12. Thus, the parameter of requests in the regression model will have a lower variation.

The covariance between ridership and other variables matches the previous analysis. The negative covariances are shown in holidays, lines 35 and 50 (two comparably quite lines), weekend, and off-peak hours. Surprisingly, we also see a negative covariance in the variance of requests with at least 30 minutes ahead. This is probably due to the change of usage behavior if we are interested further ahead in time. However, all those negative values are relatively low.

The largest three positive covariances are seen in the request, request with at least 10 minutes ahead, and ridership last week. It confirms our assumption that the relationship between trip planner requests and ridership is strongly positive correlated, which implies the potentials to predict the ridership with trip planner data. Moreover, there is a good sign that all covariances of trip planner variables with prediction lead time are strongly positive. It means that when we only utilize trip planner data with timing advance, the travel purpose and behavior do not change dramatically.

From the variance-covariance, we know that the most influencing factors are ridership-related and requested-related, such as the historical ridership and the number of requests. The strong relationship facilitates the study following. However, we also notice that both line characteristics and temporal-

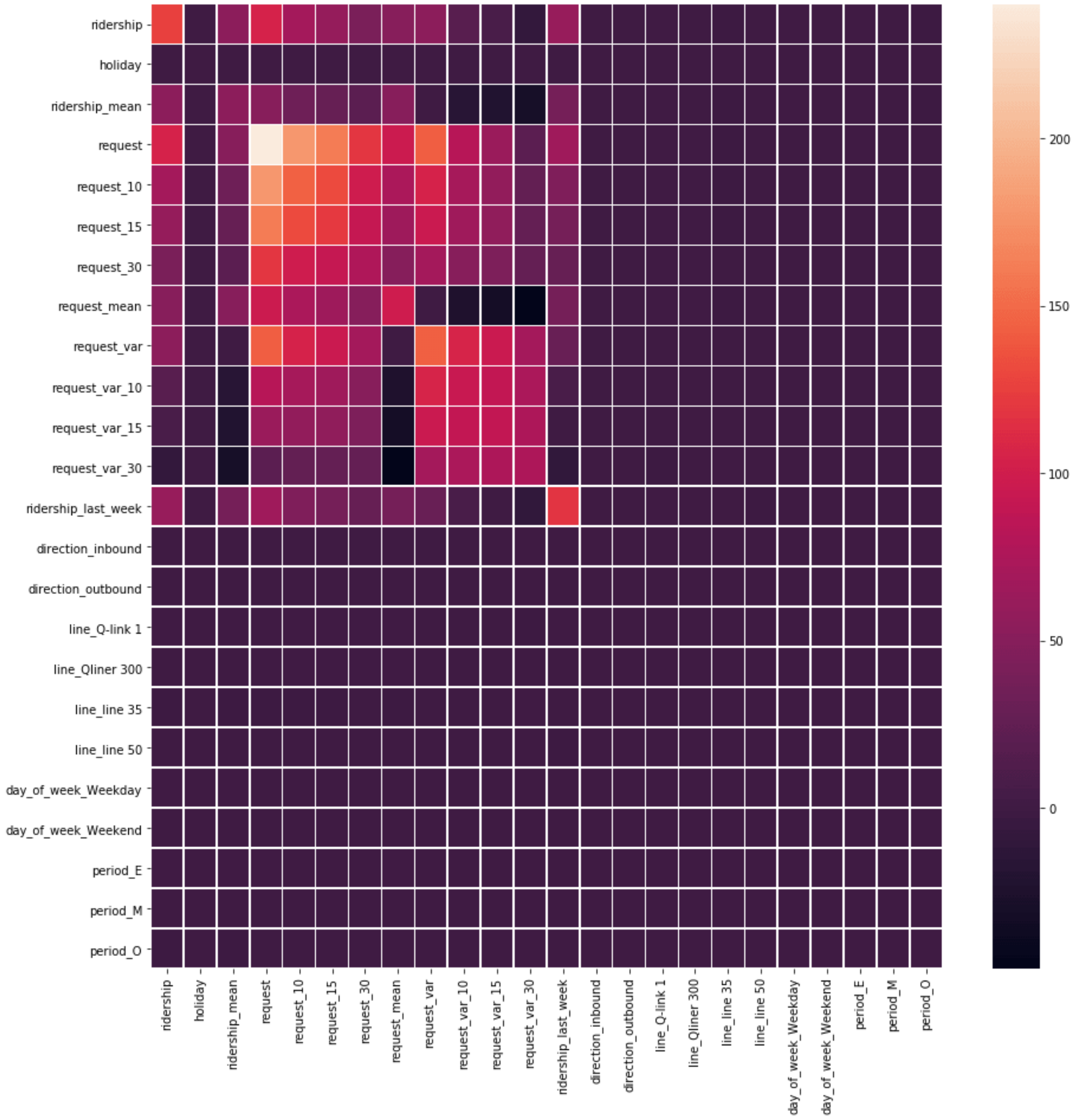


Figure 4.27: Variance-covariance matrix at stop-level

related variables have unexpectedly small covariances. It implies that the influence is marginal.

4.3.3. Pre-processing for Machine Learning: from Correlation Matrix to Ridership Prediction

Since the variables included in the covariance matrix are of different scales, a normalized form of covariance will help us to examine the relationship better. We show the correlation matrix at the stop-level in Fig. 4.28. In the visualization, we use blue color to represent a positive correlation while use red to indicate the opposite. The deeper the color, the stronger the relationship between two variables. In each cell, the size of the cube also shows the magnitude of the correlation. Although both depths of color and size of cubes refer to the strength of the correlation, it is much more apparent and good-looking to visualize in this way, compared to a heatmap without cubes. The important information this matrix conveys mainly in the first row where we show the correlation between ridership and other variables.

The absolute value of the variance of a variable is highest per row and therefore result in an all-one diagonal correlation matrix. It shows that each variable always perfectly correlates with itself. The

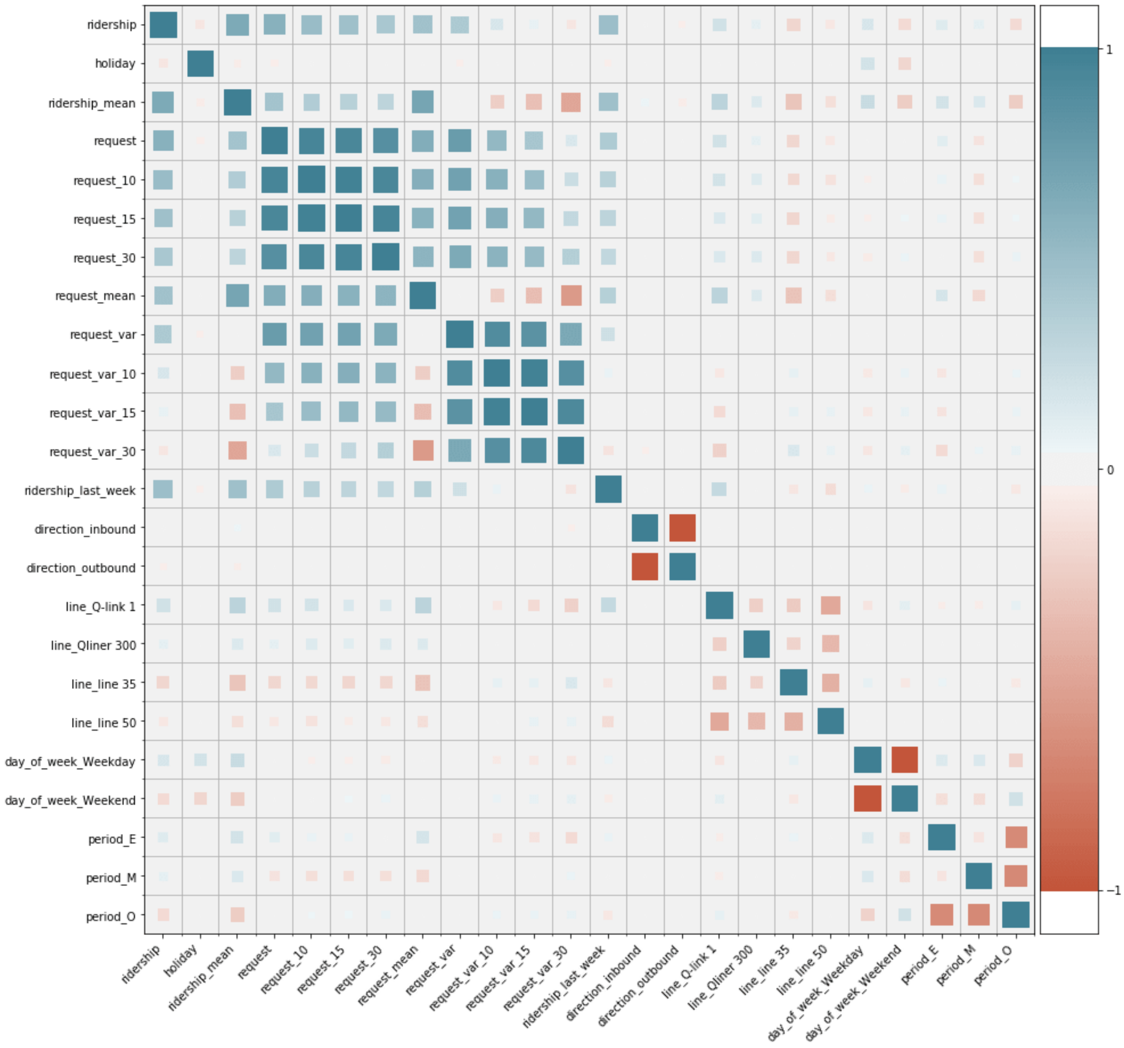


Figure 4.28: Correlation matrix at stop-level

matrix is symmetrical, with the same correlation is shown above the main diagonal being a mirror image of those below the main diagonal. Several -1 values also make sense as they are the opposite, such as the direction, the day of the week. Note that there are no zero values in the matrix. However, the cubes are not displayed due to their small values.

First of all, the request remains one of the most influencing factors with a correlation of 0.60. Second, we see a continuous decrease in the correlation when we consider the trip planner request further ahead in time. Third, all temporal variables are following the conventional understanding that holidays, off-peak hours, and weekends would have a negative effect while others have the counter-effect. Fourth, the historical average of the ridership is of importance when we want to perform ridership prediction. Furthermore, the direction of a line has almost no covariance with the ridership.

Lastly, we also consider how this correlation would develop during different periods. We present the detailed covariance matrix visualizations in *Appendix B*. We can only observe the influence of weekdays and weekends from off-peak periods because the peak hours are only on weekdays. Figures. B.1, B.2, and B.3 show the correlation matrices during the off-peak, morning peak, and evening peak, respectively. The correlation changes slightly when we focus on a specific period. The majority of the positive and negative effects between ridership and other variables remain the same, and all the conclusions we draw above are still valid for each period.

4.4. Conclusion

In this chapter, we described, cleaned, integrated, and analyzed the datasets. We cleaned three separate datasets, including trip planner data, smart card data, and AVL data. Then, we established links between different datasets. Afterward, we mapped the trip requests and the smart card transactions onto the vehicle trips, which we derive from the AVL data. In this step, we set a threshold for measuring the time difference between tap-in time and vehicle origin arrival time and the difference between tap-out time and vehicle destination arrival time. This threshold is based on the headway of the line. Then, we select the least time difference as the best matching alternative. It is worthwhile mentioning that we directly infer the ridership of a section, which is the passengers on-board between two consecutive stops, instead of inferring boarding and alighting separately because we think the later way will introduce error twice. Some data have been discarded due to the match is related to space and time. We lost approximately 20% of trip planner data and around 3% of smart card data.

After merging, we investigated the dataset to reveal the temporal and spatial dimensions that are existing in the dataset. We began with the requests where we found out that people prefer using such an application for real-time most of the time. During the night (from 22:00 to 4:00), it is in contrast that people check the trips for tomorrow with at least four to six hours in advance. All case study lines tend to have a similar distribution of requests per 10-minute aggregation, indicating a peak off-peak pattern. The influences of temporal variables are significant, including the day of the week and holidays. The realized trips vary over the day. During the daytime, it tends to be more realized while it is the opposite during the night. Then, we explore the distribution of transaction data where the temporal influence is again significant. We also discover the impact of spatial variables from the smart card where we learn that the level of service is sufficient on average while it is not satisfied with busy trips. The city Groningen attracts quite many passengers, and the crowded sections are mainly in the city center of Groningen, the railway stations, or the P+R stops in the case study region, which shows the importance of incorporating the spatial variables in the model.

Furthermore, we unveil the relationship between trip planner data and smart card data by developing the joint distribution of them at the day-level and the stop-level and variance-covariance with all variables considered at the stop-level. The variance of ridership is large, which implies the value of our study. We see that there is a considerable positive correlation between requests and transactions, which means that requests can be leverage to ridership prediction. Moreover, we understand that this relationship weakens when we dive into more details, namely from day-level to stop-level. The correlation between trip planner data and smart card data is around 0.63 at stop-level. If we carry out the prediction further ahead in time, the correlation drops from all requests included to requests with timing advance. All correlations of other variables are following the data preliminary above, including the temporal attributes, the characteristics of case study lines, and period. Besides, we know that the historical average of a trip plays an important role when we want to establish a relationship. Lastly, the covariance between ridership and all other variables remains the same during different periods.

The data analysis passed some challenges to this study. There are also some limitations of this study, revealed from this step. First, it would be easier to conduct such research if the OV bureau has the same naming system with 9292 and also other PT-related companies. Second, it would be much more meaningful if we can track the user IP and all the alternatives of a piece of trip advice. Third, 9292 is the largest trip planner company in the Netherlands. It is indeed representative. But there are also other competitive companies, which means that only proportional trip planner data have correlated with the smart card data. Fourth, there is no identical bus number provided, neither on the smart card nor the trip planner and hence the mapping would introduce errors and bias into the study. Lastly, only one-month data explored in this study, which could cause a lack of seasonal trend consideration.

5 Method

In this chapter, we highlight the theoretical framework of this study, including the data used, the model selected and deployed, and the evaluation metrics adopted. The general scheme of the method can be described by Fig. 5.1 below:

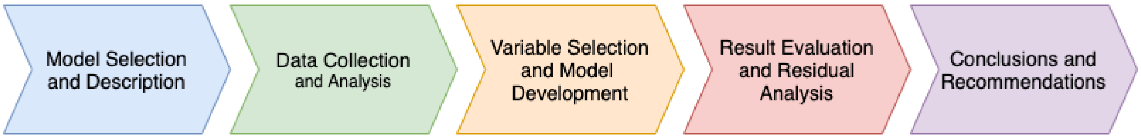

Model Selection and Description
Data Collection and Analysis
Variable Selection and Model Development
Result Evaluation and Residual Analysis
Conclusions and Recommendations

Figure 5.1: A general overview of the method framework

Since ML has been successfully applied to a wide variety of fields, definitions for a prosperous pipeline are born at the right moment but differs from each other based on the objective, such as regression, classification, clustering. However, for a typical ML problem, the components can be summarized as following (Boutaba et al., 2018):

1. Understanding the problem and application domain, and setting the goal.
2. Determining the learning paradigm and set up a strong relationship between data, problem, and the learning paradigm.
3. Collecting data, possibly without bias for building an effective ML model.
4. Processing data to clean the noisy and incomplete data and extracting features[1] that act as discriminators for learning and inference.
5. Choosing the ML algorithm(s) to fulfill the objective.
6. Establishing the ground truth pertains to giving a formal description to the classes of interest.
7. Gauging the performance of the ML model that describes, predicts, or evaluates the outcomes.

Chapter 1 has already executed the first step with the ridership prediction problem defined and the goal introduced. Chapter 2 has examined state-of-the-art literature and the current learning paradigm for predicting transit ridership. Chapter 4 has explored the data and the dimensions existing in the data. But in this chapter, we explicitly state the type of learning and its task in section 5.1. Moreover, the other sections of this chapter provide a theoretical background and preparation for step 3-7. Section 5.2 introduces the data for step 3. We explain step 4 in Section 5.3, focusing on how to clean the data and the process for analyzing the dimensions in the dataset. Section 5.4 presents the selection of models, and section 5.5 illustrates how to tune the model. The chapter wraps up with the introduction of the performance metrics and summary.

[1] The term feature in this study is the combination of variables. For instance, peak hour or not is a variable, and day of the week is a variable while passenger commute during peak hours on weekdays is a feature.

5.1. Learning Paradigm

There are four learning paradigms in ML, namely *supervised*, *unsupervised*, *semi-supervised* and *reinforcement learning*. The choice of paradigm influences the following steps and should be consistent with the objective of the research. Generally, the dataset used to build the model is often regarded as the training data, and labels are associated with data to determine whether the data is informative with meaningful tags. The outcome of an ML model is usually perceived as the identification of membership to a class of interest (Boutaba et al., 2018). Supervised learning applies labeled training datasets to build the model and maps an input to an output based on example input-output pairs (Russell and Norvig, 2009). This learning method is adopted to acquire knowledge of patterns or behaviors that exists in the training datasets. Typically, this approach is used to solve *classification* and *regression* problems that pertain to predicting discrete or continuous desired output values, respectively. For other paradigms, readers interested in them are referred to the work by Dey (2016).

As above-mentioned in the literature review, we can leverage regression and classification to a ridership prediction problem, which is the objective of this study. Both regression and classification are approximating a mapping function $f(x)$ from input variables (x), but the fundamental difference lies in the type of the output variables (y) (Loh, 2011). In a classification problem, the output variables (y) are discrete class labels, e.g. a range of ridership or whether the ridership is beyond capacity or not. In contrast, the output variables are continuous quantities in a regression problem, for instance, the exact number of ridership on board. In this study, we aim to forecast the number of passengers on board. It is a continuous quantity and can be deemed as a regression problem.

Alongside the type of learning, there are two ways of learning methodology: *generative* and *discriminative* (Ng and Jordan, 2002). The underlying difference lies in the Bayes' theorem as shown in Eq. 5.1. Suppose we have two events A and B, the conditional probability is defined as:

$$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{P(B)} \tag{5.1}$$

which can be also denoted as:

$$posterior = \frac{likelihood \times prior}{evidence} \tag{5.2}$$

The joint probability of event A and B is:

$$P(A \cap B) = P(B \mid A) \times P(A) \tag{5.3}$$

The generative model learns the joint probability distribution $P(A \cap B)$. It predicts the conditional probability with the help of the Bayes Theorem. On the other hand, a discriminative model learns the conditional probability distribution $P(A \mid B)$. In other words, a discriminative model learns the boundary between classes. A generative model models the distribution of individual classes. Both of these models were generally used in supervised learning problems, for example, the regression problem. The aim of our study is only to regress. A discriminative model is less expensive than a generative one. Following the generative approach to model input distribution can result in requiring too much training to model complexities in distribution, which is unimportant. Furthermore, a discriminative model relies more on the data itself and make fewer assumptions, compared to the interest in the data distribution and data generating process that generative models offer.

5.2. Data Collection

Commonly, there are three main data types used as input in machine learning: *categorical*, *ordinal* and *numeric*. A categorical data (also called nominal data) is one that has two or more categories, but there is no intrinsic ordering concerning the categories. For example, the time of the day can be defined as a categorical variable having three categories (the morning peak, the evening peak, and the off-peak), and there is no intrinsic ordering to the categories. A purely categorical variable is one that simply indicates where the variable belongs but no clear ordering. If the variable has a clear ordering, then the variable is regarded as an ordinal variable (Rokach and Maimon, 2008).

An ordinal variable is much similar to a categorical variable with the difference of a clear ordering of the variables. For instance, the crowdedness level of a vehicle with three categories (crowded, acceptable, empty) can be an ordinal variable. In addition to being able to classify the crowdedness onboard into three classes, we can order the categories as crowded, acceptable, and empty to represent the order from highest to lowest, vice versa. However, the size of the difference between categories is inconsistent because the spacing between categories is vague. If these categories were equally spaced, then the variable would be numerical.

A numerical variable is similar to an ordinal variable, except that the intervals between the values of the numerical variable are equally spaced. For example, the number of trip planner requests is measured in the number of logs and suppose we have 1, 2 and 3 logs for three different days. 3 logs are larger than 2 logs and are larger than 1 log and the size of these intervals is the same. It is important that our study can handle all three data types due to their possible presence or we can transform them to facilitate the set-up of the ML models.

The aforementioned data types can be structured as *vectors*, and vectors constitute the most basic entity that machine learning could encounter. A vector is a set of features with varying unites and scales (Smola and Vishwanathan, 2008). For instance, to predict the ridership with trip planner data, the data vector could consist of the number of trip planner request, the time of the day and the day of the week, etc. All the data used in our study will be transformed into a compatible form with the selected ML models as vectors. These vectors are associated with the desired individual bus trip at a specific time, retrieved from the smart card data and other related datasets. A set of characteristics captured by a vector will be referred to as a feature set and individual characteristics (i.e. vector elements) as features. The target variable, ridership, is then referred to as a label.

In general and from a practical perspective, it is possible that the more data available, the better. However, an overwhelmingly large dataset used for training the model could be less meaningful as the usefulness of the dataset may not linearly relate to the accuracy of the model and could lead to excessively long training time, eventually resulting in an almost impossible realization. Next, when applying machine learning methods, it is important to avoid overfitting. Thus, splitting the dataset into smaller instances is necessary for validating and testing the performance of the model.

In principle, one way is to split the dataset into three parts, i.e. training, validation, and testing set. We use the first two sets for training the model and tuning the configuration for final usage. The validation set is used for assessing the performance of the model. In other words, the training dataset is the sample of data used to fit the model, and the validation dataset is used to evaluate a given model so that it helps to update higher level hyperparameters. The testing set is for providing an unbiased evaluation of a final model fit on the training dataset.

The other way is to only split the data into two instances, containing only training and testing set. In this context, cross-validation is applicable to fine-tune the model. We introduce the concept of cross-validation in Section 5.5. However, there is no uniform ratio but only rule-of-thumb to determine how we would split the datasets. There are two competing concerns: with less training data, the estimation of parameters would have larger variance. With less testing data, the performance of the model would have greater variance. Thus, a concern with dividing data such that neither variance is too high is needed. Technically, the splitting ratio varies with the context, and it is often a good start point with the 80/20 rule, which is also known as the Pareto Principle.

Finally, we must derive the desired variables from the collected data. All variables need to be prepared before developing the model. These variables are created by drawing predefined sets of variables from the raw and original data. Thus, data cleaning, reorganizing, and reformatting are vital as raw data would cover a wide range of variables and with considerable noises. Some variables are directly adapted from the data without major processing or hassles while some are designed to enhance a selected factor by applying feature engineering techniques described in the next section.

5.3. Feature Engineering

From the data collection, the data we received often is not with the wanted information and can be distributed across multiple data sources. Thus, a preparation to manually organize the data for machine learning algorithms or models is required. A procedure will make sure that the data as input would be in a single table with training examples in the rows and the explanatory variables in the columns. This data representation for machine learning is called *feature matrix* and *feature engineering* is the process to extract features from raw data by using domain knowledge with the help of data mining techniques. Theses extracted features are for improving the performance of machine learning algorithms or models. From some points of view, the process of feature engineering can be regarded as applied machine learning per se (Zheng and Casari, 2018).

A feature is associated with one or more independent variables and determined by which analysis or prediction is to be done. Any attribute or variable can be a feature, as long as it is useful to the model. The purpose of constructing a feature would be much more understandable in the context of a problem. It is important to predictive models, and it will influence the results. The objective of feature engineering can be trifold. First, improving the prediction performance. Second, providing faster and more efficient predictors. Third, providing a better understanding of the underlying process that generated the data (Guyon and Elisseeff, 2003).

The relevance class of a feature can be broadly defined as strongly relevant (i.e., the feature has information that doesn't exist in any other feature), relevant, weakly relevant (some information that other features include), or irrelevant. Even if some features are irrelevant, having too many is better than missing those that are important as some models can neglect the irrelevant features but missing important features will lead to the potentially poor performance of the predictor.

Essentially, the iterative process of feature engineering is as follows: first, brainstorm features by carefully looking into the problem and referring to literature; second, devise features with respect to the problem; third, select features by using different feature importance scorings and feature selection methods to prepare one or more points of view; lastly, evaluate the model accuracy to see if there are unseen data yet captured. An overview of the feature engineering process is shown below in Fig. 5.2.

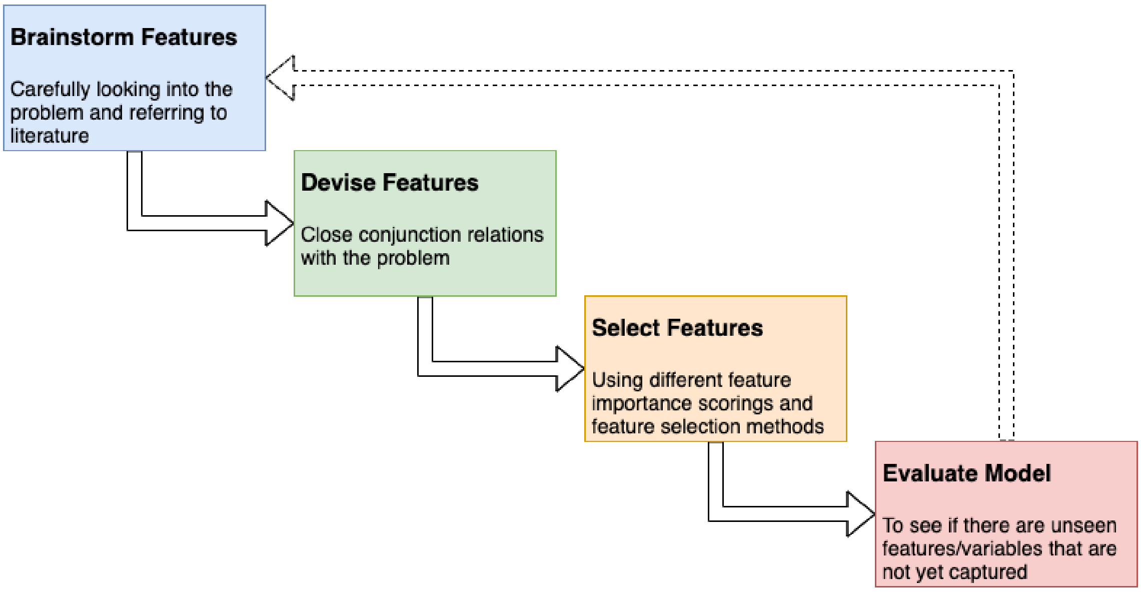

Brainstorm Features
Carefully looking into the problem and referring to literature
Devise Features
Close conjunction relations with the problem
Select Features
Using different feature importance scorings and feature selection methods
Evaluate Model
To see if there are unseen features/variables that are not yet captured

Figure 5.2: A general overview of the feature engineering

In our study, many articles have shown the usefulness of several contributing factors while some variables are remained to testify, for example, trip planner request related data. For the testified factors, we will verify the practicality as it may vary case by case. While for those emerging data sources, we will utilize a correlation study to examine the interrelationship. Eventually, the high correlated variables would be incorporated as features or composed into a feature based on their characteristic(s).

5.4. Model Selection

According to the literature review, there are a lot of models that can be chosen and there is no single model that can be the best for every scenario (Raschka, 2015). Hence, it is common and wise to pick several models for establishing a comparison. Every model has its pros and cons. Some are more interpretable while others are able to reach high accuracy of prediction. In this study, the aim is to unravel to what extent can the trip planner data contribute to the short-term bus ridership prediction and what are the important influencing factors in trip planner data and other data in such a prediction model. Thus, we will turn to more interpretable models suggested by Table. 2.4. We finally decide to include the following models: baseline, a simple model with weekly trend considered, linear regression model, k-nearest regression model, random forest, and gradient boosting regression.

5.4.1. Baseline

The baseline model is the model that PT operators are using currently. In this model, the ridership of a section of a trip this week is estimated by the ridership of last week, which is shown in Eq. 5.4 below:

$$y_{i,\, this\, week} = y_{i,\, last\, week} \tag{5.4}$$

where, y represents the ridership of a section of a trip and i means a particular section of a specific bus trip. In this study, the trip of the bus is identical throughout a single day while it will refresh the following day.

This model has its obvious advantage, which is easy to implement. While it has many disadvantages: first, it fails to capture the ridership in a shorter time, and it does not have detailed insight into the ridership. Moreover, it fails to capture the change of external factors that influence the ridership, which would result in an unreliable result. And when the journey number is missing or wrongly recorded, it could lead to an absent prediction or a huge error.

5.4.2. Simple Model with Weekly Trend

The simple model that we apply in this study is used as another baseline model. This model estimates the ridership of a section of a trip this week by the ridership of last week multiply by a weekly coefficient. This coefficient is decided by the ridership of yesterday divided by the ridership of yesterday last week.

$$y_{i,\, this\, week} = y_{i,\, last\, week} * \frac{y_{i,\, yesterday}}{y_{i,\, yesterday\, last\, week}} \tag{5.5}$$

Many of the pros and cons of the baseline model also apply to the simple model. On the good side, the simple model could capture the weekly trend and could incorporate the effect of the holiday into account.

5.4.3. Linear Regression

Linear regression is one of the most used models. The model establishes a linear relationship between the target (dependent) variable y and the prediction (independent) x. Hence, if there is a linear relation between target and prediction, a satisfying result can be expected. Suppose multiple independent variables are chosen, then the model is called multiple linear regression. The linear regression model is shown in Eq. 5.6.

$$y_{i,\, t} = a_1 x_1 + a_2 x_2 + \dots + a_n x_n + b \tag{5.6}$$

where, t symbolizes a certain day.

By fitting data into this model, we are able to derive the intercept and the slope of each independent variable included. We apply ordinary least squares linear regression with the independent variable(s). We start from the highest correlated independent variable and gradually incorporate more by the correlation hierarchy of variables, one at a time to six maximum. The linear regression aims to find the coefficients that are able to minimize the residual sum of squares between the observed targets in the dataset and the targets predicted by the linear model.

5.4.4. K-nearest Regression

The k-nearest neighbor method can be used for classification and also regression and uses the nearest neighbors of a data point for prediction. For regression, it takes the average of the outcome of the neighbors. The tricky parts are finding the right k and deciding how to measure the distance between instances, which ultimately defines the neighborhood. The right k will be validated by the grid search cross-validation, which will be introduced in the following section.

The k-nearest neighbor model differs from the other interpretable models because it is an instance-based learning algorithm. It is called instance-based because it constructs hypotheses directly from the training instances themselves (Russell and Norvig, 2009). It means that the hypothesis complexity can grow with the data, and the worst case can be a list of n training items that result in a single new instance of $O(n)$. The advantage of applying such instance-based learning is that it can adapt its model to previously unseen data. Moreover, when we make a prediction, we can retrieve the k neighbors that were used for the prediction. Since the case study instance does not consist of hundreds or thousands of features, we can interpret a single instance with manageable features.

5.4.5. Gradient Boosting Regression

As recommended by quite a number of literature with remarkable results, we desire to incorporate gradient boosting in our study to predict the ridership at a stop level. Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

In this study, instead of applying decision trees, we decide to use gradient boosting decision trees and random forest (later in the next sub-section will be introduced) to practice tree models. These kinds of techniques are ensembles of tree models. A decision tree works as follows: it applies a step-wise method that data would be split into increasingly smaller branches. The output value of branches is set to the mean of the output of the samples of the branch. The decision rule adopted is based on one of the explanatory variables and it is based on the most outperformed scenario according to the performance metric. Thus, the method is greedy with the aim of optimizing the current step instead of benefiting the future tree.

Gradient boosting regression is also a greedy algorithm and builds one tree at a time. However, each new tree included is to help to correct errors made by previously trained tree. Gradient boosting regression tree (GBRT) regressors are additive models whose prediction y_i for a given input x_i is in the following form:

$$\hat{y}_i = F_M(x_i) = \sum_{m=1}^{M} h_m(x_i) \tag{5.7}$$

where, h_m is the estimator called weak learners in the context of boosting. Gradient tree boosting uses decision tree regressors of fixed size as weak learners. The constant M corresponds to the number of trees.

Similar to other boosting algorithms, the greedy algorithm is formulated as:

$$F_m(x) = F_{m-1}(x) + h_m(x) \tag{5.8}$$

where, the newly added tree h_m is fitted in order to minimize a sum of losses L_m, given the previous ensemble F_{m-1}:

$$h_m = \arg\min_h L_m = \arg\min_h \sum_{i=1}^{n} l(y_i, F_{m-1}(x_i) + h(x_i)) \tag{5.9}$$

where $l(y_i, F_m(x_i))$is defined by the loss function. The initial model is chosen as the constant that minimizes the lose: for a least squares loss, this is the empirical mean of the target values. Using a first-order Taylor approximation, the value of can be approximated as follows:

$$l(y_i, F_{m-1}(x_i) + h_m(x_i)) \approx l(y_i, F_{m-1}(x_i)) + h_m(x_i)[\frac{\partial l(y_i, F(x_i)}{\partial F(x_i)}]_{F=F_{m-1}} \tag{5.10}$$

In this equation, the quantity $\frac{\partial l(y_i,F(x_i)}{\partial F(x_i)}$ is the derivation of the loss with respect to its second parameter, evaluated at $F_{m-1}(x)$. When we fit the data into the model, we compute the prediction with the aim of minimizing h_m that is proportional to the negative gradient which is the derivation. Therefore, at each iteration, the h_m is fitted to predict the negative gradients of the sample. And these gradients are updated during every iteration.

GBRT is better in prediction but prone to overfitting and thus we need cross-validation to detect if such overfitting exists. Furthermore, GBRT is much more expensive in tuning and training.

5.4.6. Random Forest Regression

Random forest regression is another extension of the decision tree. It is also an ensemble of decision trees. Rather than fit one tree at a time, random forest fits multiple trees. Each tree makes a prediction. By taking the average of these predictions, the final prediction of random forest is generated. It is different from a regular decision tree that trees in random forest have been resampled by using bootstrapping and by taking only a set of the variables into account during each split.

Bootstrapping is a resampling method that randomly picks a sample from the dataset with replacement. In the random forest, the square root of the total number of predictors would normally be the number of variables taken into account during each split. This method ensures that the trees are more exclusive from each other and thus is more stable, which eventually results in a better performance than decision trees. However, less interpretability than decision tree and a much more expensive computation cost is inevitable. The number of trees should be sufficiently large to get a good result but should not be too large to prevent large training times. If the number of trees is large enough, a depth of 1 might also be sufficient. Setting the right minimum number of samples in the end nodes ensures a good variance-bias trade-off.

Based on the concept of random forest regression. There are several parameters in such a model that need to be tuned. First, we need to decide the number of trees in the forest. If the number of observations is large, but the number of trees is too small, then some observations will be predicted only once or even not at all. If the number of predictors is large but the number of trees is too small, then some features can (theoretically) be missed in all subspaces used. Second, we need to decide the maximum number of features considered for splitting a node. The number of randomly selected features can influence the generalization error in two ways: selecting many features increases the strength of the individual trees, whereas reducing the number of features leads to a lower correlation among the trees increasing the strength of the forest as a whole. Third, we need to determine the maximum depth in each decision tree. The depth of the tree meaning length of the tree you desire. A larger tree helps you to convey more info, whereas a smaller tree gives less precise information. So the depth should be large enough to split each node to your desired number of observations. Fourth, we should rule the minimum number of data points placed in a node before the node is split. It can vary between at least one sample to all of the samples at each node. When we increase this parameter, each tree in the forest becomes more constrained as it has to consider more samples at each node. Fifth, the minimum number of data points allowed in a leaf node needs to be regulated. This parameter is similar to the minimum sample split. However, this describes the minimum number of samples at the leaves, the base of the tree. Lastly, it is the bootstrap that we mentioned above, which is a method for sampling data points (with or without replacement).

5.5. Model Tuning

To calibrate and update the hyperparameters of models and investigate the robustness of the calibration, we follow the cross-validation procedure. Normally, we apply *k-fold cross-validation* that splits up the dataset into k-partitions - 5 or 10 partitions, which is recommended as a rule of thumb (James et al., 2014). The way that splits the dataset is making k random and different sets of indexes of observations, then interchangeably using them. The percentage of the full dataset that becomes the testing dataset is $\frac{1}{k}$, while the training dataset will be $\frac{k-1}{k}$. For each partition, a model is fitted to the current split of training and testing datasets. An example of 5-fold cross-validation is shown below in Fig. 5.3. The full dataset will interchangeably be split up into a testing and training dataset, which a model will be trained upon.

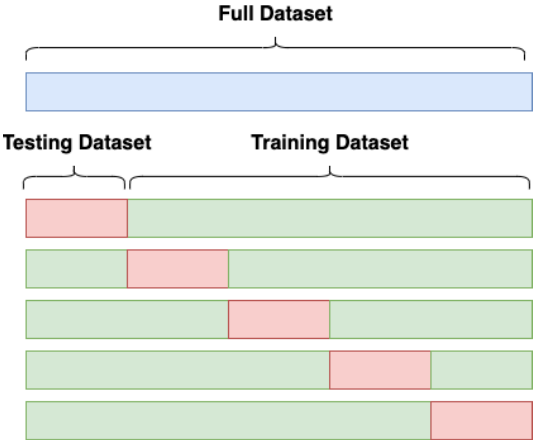


Figure 5.3: The set up of cross validation (Browne, 2000)

When tuning the parameters, we use cross-validation with a search algorithm, where we put a hyperparameter grid - parameters that are selected before training a model. In combination with *Random Search* or *Grid Search*, you then fit a model for each pair of different hyperparameter sets in each cross-validation fold.

However, when we again apply *k-fold cross-validation* to estimate the skill of the model on new data, it would incur information leakage and significant bias (Cawley and Talbot, 2010). Thus, we need another strategy called *nested cross-validation* shown in Fig. 5.4 below. In this *nested cross-validation*, we have two loops. The inner loop is basically normal *k-fold cross-validation* with a search function, e.g. random search or grid search. Though the outer loop only supplies the inner loop with the training dataset, and the test dataset in the outer loop is held back. In this way, the inner loop is for validating the model and update the parameters and the outer loop is for testing the model with an unbiased evaluation of the final model and therefore we can avoid bias.

For the inner loop, we will apply *stratified k-fold cross-validation* in which the partitions are selected so that in each partition the number of elements of the same class is approximately equal. In stratified sampling, suppose the population consists of N elements and this population is divided into H groups, which is called *strata*. Each element of the population can be assigned to one, and only one, stratum. This kind of sampling strategy outperforms than a simple random sampling from three perspectives: first, stratified sampling can provide greater precision than a simple random sample of the same size; second, if we perform the methodology line-wise, the size of the dataset drops, and we can still obtain high accuracy by applying this sampling strategy; third, we ensure that the sample is representative. By applying stratified sampling to train the model is also in line with the interest of PT operators as they have remarkable knowledge about the normal situations of a bus trip, such as quiet region/time, off-peak hours trip, and infrequent line, etc. But they are more interested in the crowded or unusual scenarios which are only represented by limited data and strata sometimes. Thus, they want the model can equally capture or treat every scenario more or less the same and one way to fulfill this objective is by deploying *stratified k-fold cross-validation* to train the model.

As for the outer loop, we apply *random permutation cross-validation* after the fine-tuned model by the inner loop. This method is also called *shuffle split* and it will randomly sample the entire dataset during each iteration to generate a training set and a test set (Kuhn et al., 2013). The test size and train size parameters control how large the test and training test set should be for each iteration. As it is an iterative process and the method samples from the entire dataset during each iteration, values selected during one iteration could be selected again during another one. If the data (population) is

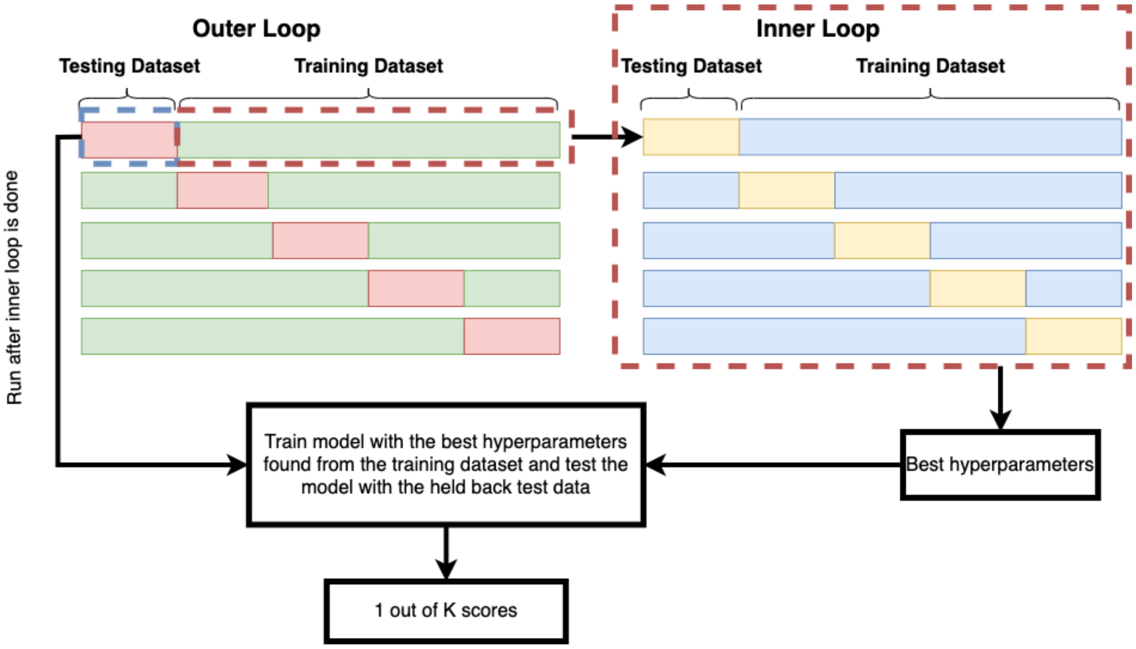

Outer Loop
Testing Dataset
Training Dataset
Inner Loop
Testing Dataset
Training Dataset
Run after inner loop is done
Train model with the best hyperparameters found from the training dataset and test the model with the held back test data
Best hyperparameters
1 out of K scores

Figure 5.4: The set up of nested cross validation (Parvandeh et al., 2020)

imbalanced, then a model trained upon stratified data will compromise its accuracy when it is validated by the random permutation cross-validation as the validation data is not stratified. This study is in favor of shuffle split as it is closer to reality as the appearance of the target is random and is not stratified. Moreover, the configuration of the method can be set as less demanding (for instance, the train/test split is 80/20 and the number of iteration is 5).

5.6. Performance Metrics

A major part of our study is to assess the performance of the prediction models and compare those performances by a set of metrics. The study uses MAE, $RMSE$ and R^2 to determine the goodness-of-fit at a section (stop) level, which is to measure the ridership between two consecutive stops per bus trip.

MAE represents the difference between the original and predicted values extracted by averaged the absolute difference over the data set:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.11}$$

where, $\hat{y}_i$ is the predicted value and y_i is the true value for the i^{th} record. A lower MAE tells that the predictions are close to the true values.

$RMSE$ is equal to the square root of the MSE (mean squared error), which is the most commonly used metric for regression models (James et al., 2014):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.12}$$

A lower $RMSE$ tells that the predictions are close to the true values. Because the prediction error is squared, the $RMSE$ quickly increases when there is a substantial prediction error for some values. The model will be fitted by minimizing the $RMSE$. This study is to minimize the difference between the ridership predicted between stops on a vehicle and the real ridership retrieved from smart card data.

R^2 measures the proportion of variance that is explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.13}$$

where, $\hat{y}_i$ is the predicted value, y_i is the true value and $\overline{y}$ is the mean of the true values. The R^2 normally ranges between 0 and 1. If all the variance can be explained the R^2 will be 1. Since the R^2 is a proportion, it can be used to compare the results with other research because the $RMSE$ is not ideal for such comparisons as it is an absolute measure of the prediction error with units of y. For instance, if the target unit is different in the context of ridership prediction, for one model it is in persons while for another one it is a bunch of people, then the models are not comparable with $RMSE$.

Another performance metric we need to set up is to explore the importance of features. Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction (Kuhn et al., 2013). The scores can be calculated for problems that involve predicting a numerical value, which is the same as in our study, regression problem. If it is for predicting a class label then it is called classification. The reason why we need to understand the feature importance scores are trifold: first, it helps us to understand better of the dataset. The relative importance scores can shed light on the features which are most relevant to the target. This can facilitate the interpretability and could help data gathering and practicing. Second, it provides insights into the model. By inspecting the importance score, we can grant the knowledge of what are the most and least important features when building up such a prediction model. This is invaluable in our study as we want to discover the usefulness of trip planner data. Third, the scores can help us to improve the model by selecting irrelevant features to remove or important features to keep. The pruning process simplifies the model and speeds up the modeling process.

For different models, the feature importance investigation is unfolded differently as it depends on the configuration of the model. In this study, we look at three main types of feature importance in accordance with the prediction models: model coefficients, decision tree and permutation testing.

Linear machine learning algorithms fit the model by the ordinary least squares where the prediction is the weighted sum of the input values. The slope of variables, which is a set of coefficients, can be regarded as a crude type of feature importance score.

For tree-based models, such as GBRT and random forest regression, the relative rank (i.e. depth) of a feature used as a decision node in a tree can be used to assess the relative importance of that feature with respect to the predictability of the target variable (Louppe, 2014). Variables or features used as inputs at the top of the tree(s) contribute to the final prediction decisions of a larger fraction of the input samples. The expected fraction of the samples that the top variables contribute can thus be used as an estimation of the relative importance of the features. For random forest, the prediction of the target variable is concluded by averaging the result from each tree and thus it can reduce the variance of such an estimation. It is known as the *mean decrease in impurity (MDI)* and can be used for feature selection.

Despite the method is easily understandable and computationally light, it has two flaws. First, it is computed based on the training dataset and hence does not necessarily indicate which feature contributes the most to the prediction on the validation dataset. Second, it overestimates the high cardinality features as they tend to have more unique values.

Permutation feature importance is an alternative to the impurity-based feature importance that does not have the above-mentioned flaws. It is a model inspection technique that can be used for any fitted estimator when the data is tabular, such as tree-based models, KNN, etc. The permutation feature importance is defined to be the decrease in a model score when a single feature value is randomly shuffled (Breiman, 2001). It regards a feature that is important when shuffling its values increases the model error as the model relies on the feature for the prediction while when the feature is unimportant, the error remains unchanged because the model ignores the feature for the prediction. Suppose that we have a trained model f, a feature matrix X, a target y and an error measure $L(y,f)$ (which can be R^2 for a regression problem), the permutation feature importance algorithm based on Fisher et al. (2018) proposed is shown as follows: first of all, we estimate the original model error $e_{original}$ by $L(y,f(X))$; secondly, for each feature from $j=1,\ldots,p$ and for each repetition $k=1,\ldots,q$, we randomly shuffle feature j in the data X and generate the feature matrix $X_{k,j}$. This will break the association between feature j and true outcome y. Then we estimate the error $e_{k,j}$ by $L(y,f(X_{k,j}))$ based on the predictions of the permuted data. Next, we calculate the permutation feature importance FI_j by:

$$FI_j = e_{original} - \frac{1}{q}\sum_{k=1}^{q} e_{k,j} \tag{5.14}$$

However, the permutation importance can be computed either on the training set or the test (validation) set. There is no clear answer and no research hitherto addressing the question of training or test set. The underlying difference is whether we want to know how much the model relies on each feature for making predictions (by using training data) or how much the feature contributes to the performance of the model on unseen data (by using test data). In our study, we use the feature importance based on test data. The reasons are if we measure the model error (or say performance) on the same data that our model was trained, the measurement tends to be too optimistic, which in turn indicates that the model works better than the reality. Moreover, the permutation importance is built upon the change of the model error. The unseen test data is more desirable as the change of error based on training data makes us mistakenly believe that features are important. However, in reality, the model can be overfitting, and thus the features are not important at all.

Like any other feature importance technique, permutation importance also has its pros and cons. As it measures the change of the model error (or performance), the interactions among features are taken into account. By permuting a feature, we destroy the original effects with other features. This means we take into account both the main feature effect and the interaction effects on model performance. But this is also a disadvantage, we measure not only the feature importance but the sum of feature importance and the interaction and this could be larger. Another advantage of permutation feature importance is that it is computationally light as it does not require the retraining of the model and the interpretation is easy. However, as aforementioned, whether to apply the technique on training or test set can be unclear, it depends on what the objective is. Moreover, it links to the error of the model which is not inherently harmful. But if we are interested in the robustness of the model, a change of the variance of the performance would be more focused which can not be reflected. Next, the permutation is random. This means the results might vary significantly, but we will repeat the process to minimize the randomness. Lastly, if there are strongly correlated features, the model will still have access to the feature through its correlated feature when one of the features is permuted. This will result in a lower importance value for both features where they might be important. One of the possible ways is to cluster features that are correlated and only keep one feature from each cluster.

One last performance metric to measure the feature importance is the *partial dependence plot (PDP)*. It shows the marginal effect one or two features have on the predicted outcome of a machine learning model (Friedman, 2001). A partial dependence plot can depict whether the relationship between the target variable and a feature is linear, monotonic, or more complex.

The partial dependency function for regression problem is defined as:

$$\hat{f}_{x_S}(x_S) = E_{x_S}\left[\hat{f}(x_S, x_C)\right] = \int \hat{f}(x_S, x_C)d\mathbb{P}(x_C) \tag{5.15}$$

where, x_S is the feature or are the features for which the partial dependence function should be plotted and x_C are the other features used in the machine learning model $\hat{f}$. Usually, there are only one or two features in the set S and the feature(s) in the set S are those for which we want to know the effect on the prediction. The entire feature space x is made up by x_S and x_C. Partial dependence is calculated by marginalizing the machine learning model output over the distribution of the features in set C so that the function shows the relationship between the features in set S that we desire to observe and the predicted outcome. By marginalizing over the other features, we get a function that only depends on features in set S such that interactions with other features are included.

The partial function $\hat{f}_{x_S}$ is estimated by the calculation of the average in the training data, as known as Monte Carlo method:

$$\hat{f}_{x_S}(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)}) \tag{5.16}$$

The partial function shows that for a given value(s) from the feature set S, what is the average marginal effect on the prediction is. In the Eq. 5.16, $x_C^{(i)}$ are the actual feature values from the uninterested feature dataset and n is the number of instances in that dataset. PDP assumes that the features in C are not correlated with the features in the interested feature dataset S. If this assumption is violated, the averages calculated for the PDP will include data points that are very unlikely or even impossible.

The advantages of PDP are obvious: it is quite intuitive that PDP takes only one particular feature value to represent the average prediction if we make all data points to assume that feature value. If the assumption is not violated, then the interpretation of PDP is clear and straightforward as it describes how the feature influences the prediction on average. Moreover, it is easy to implement. Nevertheless, it also has several apparent drawbacks. The realistic maximum number of features in a PDP is two. And PDP assumes that independence of the variables between the interested and the uninterested sets. This leads to a problem when the features are correlated, new data points are generated in the areas of the low actual probability part of the feature distribution. Finally, it misses the heterogeneous effect because it only shows the average marginal effects. Suppose we have a feature, half of the data points are positively associated with the prediction while the other half is negative, PDP will eventually give us a horizontal line as this is the average of the effect. This can be overcome by an individual conditional expectation curve. But we will see whether we will need that or not, based on the outcomes.

5.7. Summary

In this chapter, we provide a theoretical framework of this research to predict the short-term bus ridership with trip planner data by applying ML methods. We illustrated the steps of learning paradigms choice, data collection, feature engineering, model selection, model tuning, and model validation and interpretation via a set of performance metrics.

Our work leverages regression to the short-term bus ridership prediction and supervised learning is chosen to accomplish the task. We will iteratively explore the raw data to construct a well-knitted variable (feature) set. The ML methods that we choose for our study are *LR*, *KNN*, *GBRT* and *RFR* due to a more interpretable configuration and a more interpretable results. Along with baseline models (PT current model and PT current with seasonal trend), we will establish a comparison and find the outperformed one. Several parameters are needed to be tuned to optimize their performance during the training of the model and thus we have to apply nested cross-validation to fine-tune the parameters in the inner-loop by *k-fold cross-validation* and compute the robustness and accuracy of the models via outer-loop by *random permutation cross-validation*.

To compare the multiple models, we set up an overall set of performance metrics that consist of

MAE, $RMSE$, and R^2. Since the main objective of the study is to discover the usefulness and utility of trip planner data, only get to know the performance of the model is not enough. Thus, we introduce feature importance to the study and we will select the best model out of the omnibus to explore the feature importance. For different models, the way of computing feature importance is dissimilar while the permutation feature importance is comparably convincing. But we will take the MDI method for the tree-based model and permutation feature importance both into account if the tree-based model outperforms than the others. Moreover, we will deploy PDP to assess the influence of a single feature on the average prediction of the model.

6

Model Development

After performing preliminary on the data in the fourth chapter, we are able to have a general understanding of the data separately and collectively. In this chapter, we further summarize variables developed in the previous chapter and determine the inputs for the models with the method set up in the previous chapter. Moreover, we develop the strategy to tack the imbalanced data as both PT operators and 9292 want the model to treat every scenario equally. Furthermore, we explain how the models are developed in this study and how we set up the configuration of the model and how the models' hyperparameters are tuned. A flow diagram of the model development is shown in Figure 6.1.

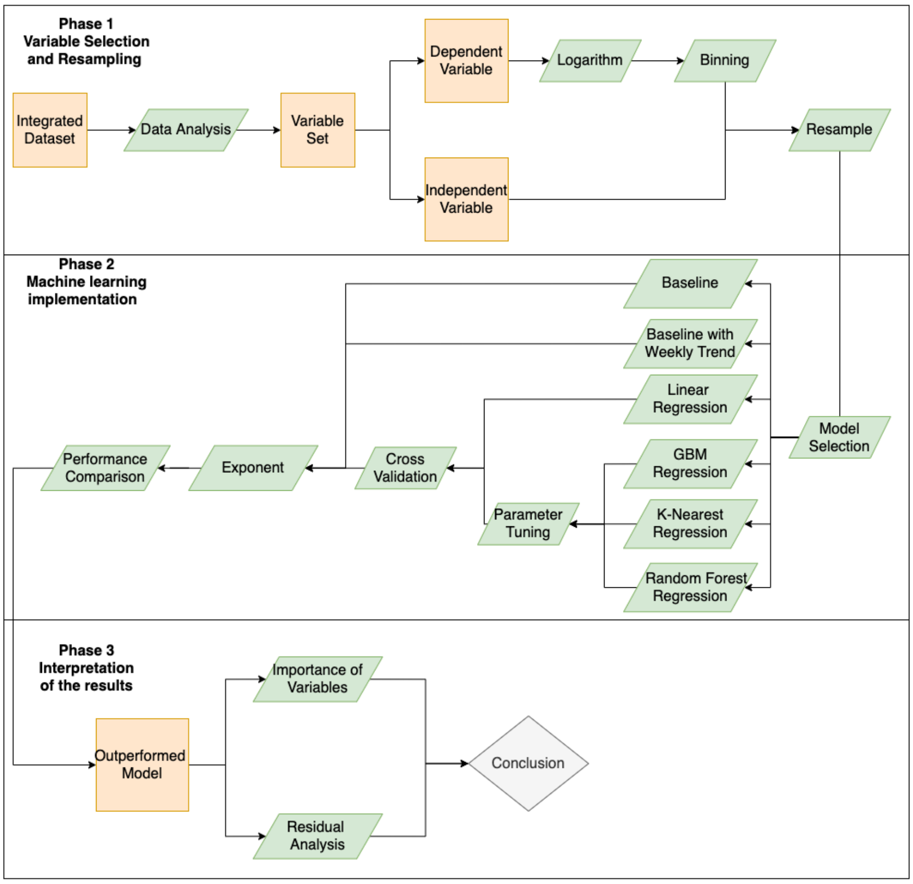


Figure 6.1: Workflow of predicting short-term bus ridership with trip planner data

We divide the whole dataset into two parts, namely the dependent variable set and the independent variable set. The dependent variable is the target that we want to predict. It is the ridership per stop per bus trip. The independent variables make up the training set that we need to explore to establish a relationship with the target. However, as we can see from the previous chapter, the ridership presents a right-skewed distribution, which means that we have more data under the few ridership situations but less in the crowded ones. Thus, we first take the natural logarithm on the ridership to generally smooth the data towards a normal distribution. Then, we apply bootstrap to generate more data under extreme situations. We also test the sensitivity of the bootstrap strategy that we used line-wise.

Next, we input the resampled data into the models that we select in this study, including the baseline model, baseline model with the weekly trend, linear regression, GBM regression, KNN regression, and RF regression. For GBM, KNN, and RF, we need to tune the hyperparameters to reach a better performance while for others this is not necessary. Only the models that have the train/test split procedure need to be cross-validated as this could affect the overall performance of the model. With all the fine-tuned model, we can take the exponent of the ridership that has been taken the natural logarithm before, changing it back to the unit of person and measure the error. The cross-validate in the figure refers to the models that need to be cross-validated, however, this step is the last step we carry out in the implementation. The target variable has been changed back to its measurement scale (take exponent function to person as the unit) before we apply the cross-validation to test the accuracy of the fine-tuned model.

Finally, we determine the best model line-wise and use that model to explore the importance of variables by the feature importance and PDP. We emphasis on the relationship between ridership and trip planner request in this step. Furthermore, we thoroughly analyze the residual plots from different perspectives to assess the problem of our regression models and put forward the future work. We elaborate on this step in the next chapter, including the results of the models.

The other part of research sub-question has been answered in this chapter is:

Along the above-mentioned dimensions, how does trip planner data correlate with observed ridership from AFC data in the short-term?

The research sub-question has been answered in this chapter is:

How can such correlations be leveraged for short-term bus ridership prediction?

6.1. Variable Selection

Variable selection is important before we proceed to apply the machine learning models. If a variable is insignificant, it means that it contributes less to the regressed target. Therefore, we should keep it out of the model. Otherwise, the model would become unnecessarily complex and thus perform poorer than expected.

In this study, we first depend on the literature review as the starting point to explore the multiple datasets. Then, we perform the data preliminary of the fourth chapter to select our variables, especially the last section of variance-covariance/correlation analysis. If we find a strong correlation between two variables, it is likely to have a strong correlation between them. Since we will later train the model per case study line, the variable “line” is left over. Other variables selected are based on Table 4.3 and the conclusion of section 4.3.2 and 4.3.3.

Except for the request related variables that are not examined before, all other variables bring into correspondence with the literature. The consensus of the influence of time and date remains consistent, such as the period of the day and the day type. The influence of special days is considerable, for instance, the autumn holiday. Spatial attributes are also contributing, especially the section around a railway station or a working place.

Before we train the model, it is important to scale the variables. We standardize the train data so that the transformed variables have a mean of 0 and a standard deviation of 1. First, the input dataset has differences in their measurement unit, such as one-hot encoding, person, and record. And the range of those data has large differences between each other. These differences in the ranges of initial variables cause troubles to many machine learning models that are sensitive to the distance, for instance, KNN in our study. Second, we scale the variables so that the prediction has a mean of 0. In our regression problem, this makes the intercept term is interpreted as the expected value of y_i when the predictor values are set to their means (i.e. independent variables). Otherwise, the intercept is interpreted as the expected value of y_i when the predictor values are set to 0, which may lead to an unrealistic or interpretable situation. And the last reason is to simplify calculation and notation.

We choose standardization rather than normalization as it is less sensitive to noise. Moreover, we do not standardize the target variable, which is the ridership. In this way, we can calculate the difference between the predicted value and the actual value in the unit of passengers per stop per trip. It helps us to understand the error (residual) in an understandable manner.

The selection of variables would always be an iterative process to reach a higher performance of the models. Since we choose the model with more abilities in interpretation, we can trace back to the importance of variables and then filter the variables again. We will perform this ad hoc and post hoc analysis for the outperformed model and use that model to illustrate the significance of variables.

6.2. Oversampling and Undersampling

In our data, we have a very imbalanced distribution of the target variable (ridership). We present the distribution of the population per case study line below in Fig. 6.2. In this density plot, we can see that the distribution of every case study line is right-skewed, and the extreme value can be as high as 108. It means that we have more knowledge of the quiet trips but less data on the busy ones, which is in contrast with what 9292 (trip information company) and PT operators want to capture. Thus we have to develop strategies to tack this imbalanced data issue.

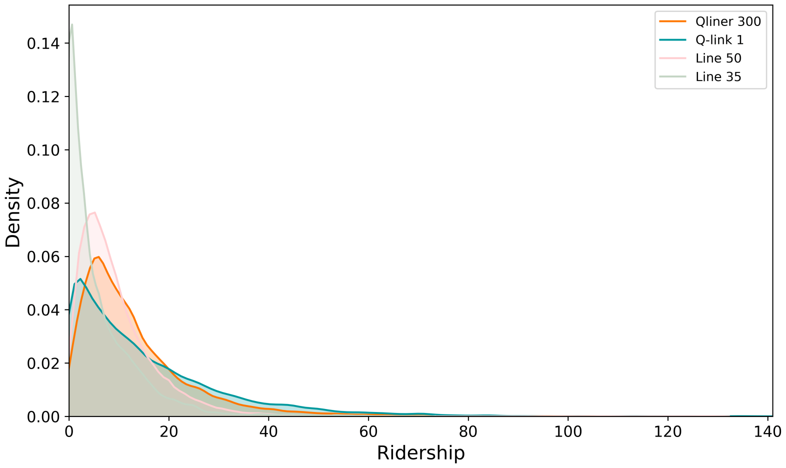


Figure 6.2: The distribution of ridership per line

We regard this problem is important in our study in bifold: first, we have non-uniform preferences across the target variables domain; second, the most interested ranges are poorly represented (Branco et al., 2017). It is in line with part of the objective of this study that we assign more importance to the predictive performance achieved in the poorly represented ranges of the target variable (ridership on busy trips) in comparison with other more frequent ranges (quite trips). However, this conjunction causes a degradation of the performance of the most important and desirable cases. It means that the standard algorithms are biased towards the majority classes (known as “negative”). Therefore, there is a high misleading rate or even be ignored to achieve good performance in the minority class instances (called “positive” classes) (Fernández et al., 2018). The methods that we choose in this study (tree-based RF and GBM, and KNN) will learn the minority class is not as important as the majority class. Hence a balanced method is needed.

The most common method to tackle this imbalanced learning problem is via sampling methods or resampling methods (He and Ma, 2013). The aim is to change the composition of the training dataset. This sampling or resampling method is only performed on the training dataset. It is not carried out on the test dataset as the test set is to evaluate the resulting model but not remove the class imbalance from the model fit and should be both real and representative of the target domain. Assessing a model with a synthesized or transformed dataset would provide a misleading and optimistic estimation of performance.

In general, we have two ways of sampling/resampling strategies, i.e. undersampling or oversampling techniques. Undersampling is intuitive and consists of removing samples from the majority class(es). Oversampling is the reversed procedure of adding samples to the minority class(es). Chawla et al. (2002) proposes a method called Synthetic Minority Oversampling Technique (SMOTE), and it is capable of generating new samples that resemble the original observations but are not duplicates. SMOTE creates new instances of a minority class by using a K-Nearest-Neighbor approach. A random number of original observations are chosen and for each of their K neighbors, a new sample is created as a linear combination of the initial observation and its neighbor. The authors indicate that generally, a combination of SMOTE and undersampling performs the best (Chawla et al., 2002). Therefore, we decide to use the combination of oversampling and undersampling to resample our data (Fernández et al., 2018).The ratio of SMOTE and undersampling is necessary to be testified by a sensitivity analysis.

However, we can not directly practice oversampling. Instead, we have to convert the continuous data into discrete data to ensure that every stratum has enough samples. Moreover, the imbalanced data problem pervades in many domains but mainly studied in the context of classification (He and Ma, 2013). Thus, we apply Doane's formula shown in Eq. 6.1 to decide the number of bins. It is essentially a modification of Sturges' formula, which attempts to improve its performance with non-normal data (Doane, 1976).

$$k = 1 + log_2(n) + log_2(1 + \frac{|g_1|}{\sigma_{g_1}}) \tag{6.1}$$

where g_1 is the estimated 3rd-moment-skewness of the distribution and it is formulated as:

$$\sigma_{g_1} = \sqrt{\frac{6(n-2)}{(n+1)(n+3)}} \tag{6.2}$$

Nonetheless, the conversion still results in a situation where several bins have limited samples. It results in an unsatisfied result from the model after SMOTE if we apply random oversampling to duplicate an instance. Moreover, if we refer to the current distribution of data to conduct the train/test split and train the model, the results from all four case study lines are depreciated and not desirable. Thus, we take the log transformation of the target for all four case study lines.

The transformation of a variable changes the shape of its distribution. In general, regression models work better with more symmetrical, bell-shaped distributions. Logarithmic transformation is a convenient approach to transform a highly skewed variable into a more normalized dataset (Metcalf and Casey, 2016). Tukey (1977) expounded that when the residuals have a "strongly" positively skewed distribution, a log-transformation can reduce this skewness. Besides, the transformation provides an equally spread of the data, which facilitates the performance of the model. However, it also has some downsides, such as it reduces the variability of data, and it is not always functional.

In this study, we took a post hoc analysis and found that log-transformation works well on the dataset and reduces the error, compared to the base scenario. Thus, we first take a natural logarithm function on the target variable and then apply the binning strategy to discrete the continuous data. Note that there are empty trips, and thus we plus one on the target and take the natural logarithm to train the model as shown in Eq. 6.3. Finally, we execute a sensitivity analysis to determine the optimal resampling strategy to emphasize the extreme but less existed instances. We only utilize transformed the target in training the model, and when testing, we take the exponent to convert the unit more comparably and understandably.

$$y_i' = \log_e(y_i + 1) \tag{6.3}$$

The distribution of ridership after the natural logarithm is presented in Fig. 6.3. From the figure, we can see that after taking the natural logarithm, the distribution of the population of every case study line is transforming from a highly right-skewed distribution towards a normal distribution. However, the amount of data that represents the extreme cases are still less profound, which means the resampling strategy to better capture the important domains is necessary.

Table. 6.1 reports the number of bins of each case study line, calculated based on Doane's formula, and the number of bins that we apply SMOTE. We determine the number of bins, the majority, and the

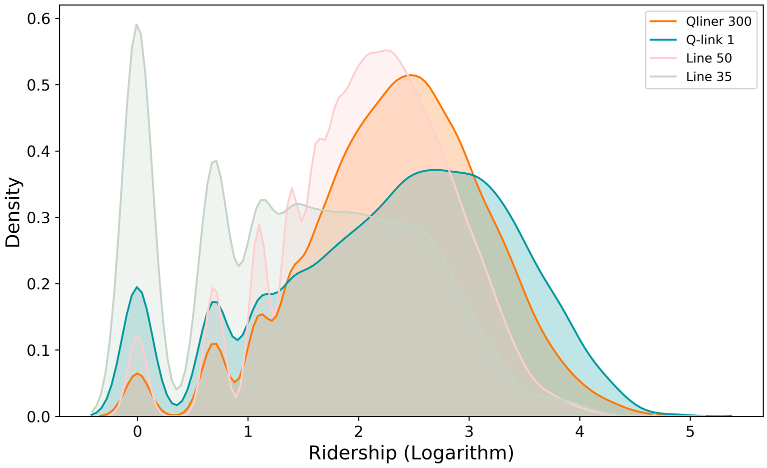


Figure 6.3: The distribution of ridership (log) per line

minority based on the training dataset. The majority should have overwhelmingly large value with more than two times than other bins initially. Except for the bins from the right side of the distribution that we present in Table. 6.1, all other bins are majorities.

Table 6.1: The binning strategy of each case study line

	Number of bins	Oversampling bins
Qliner 300	21	Last 3
Q-link 1	22	Last 4
Line 50	24	Last 7
Line 35	20	Last 4

We develop a pair-wise study with 4 undersampling and 6 oversampling strategies. We test them through 5-fold cross-validation on the dataset of every case study line, for a total of 24 resampling tests. The undersampling strategy consists of reducing the number of samples of the majority class by a percentage of 0%, 25%, 50%, and 75%. The oversampling includes enlarging the minority samples by 50%, 100%, 150%, 200%, 250%, and 300%. Since every case study line shows a different pattern of the distribution, and it is necessary to set up the analysis case-by-case. We use a random forest regressor as the model to perform this sensitivity analysis as it is representative because it suffers from the learning from an extremely imbalanced training dataset (Khoshgoftaar et al., 2007). Moreover, RF is construed to minimize the overall error rate. It tends to focus more on the prediction accuracy of the majority class, which often results in poor accuracy for the minority class. Besides, we test the performance of all models with the original training dataset, and RF primarily is the outperformed model.

We use R^2 to evaluate the resampling strategies of the RF regressor, which is the coefficient of determination. A higher R^2 means that smaller differences between the observed data and the fitted values. It is essentially the percentage of the dependent variable variation that a linear model explains. Figure 6.4 presents the results of oversampling and undersampling of each case study line. The difference among the scores per case study line is tiny, and therefore we use a Min-Max scaler to shrink the range of the result between 0 and 1 to make the difference profound.

From the figure, we can see that all case study lines show a similar pattern, except for line 50. The optimal sampling design is to oversample the minority by 50%, with no undersampling. With the reduction of majorities, the R^2 drops correspondingly, despite the oversampling of minorities. Horizontally, the increase of the oversampling of minorities incurs the decline of the score when undersampling is small. However, when undersampling is large, the role of majority and minority may change, and thus, the score decreases and then increases.

Due to the train/test split is random, we repeat the evaluation of different sampling designs per case study line 10 times and present the other 9 times in the Appendix from Fig. C.1 to Fig. C.9. The

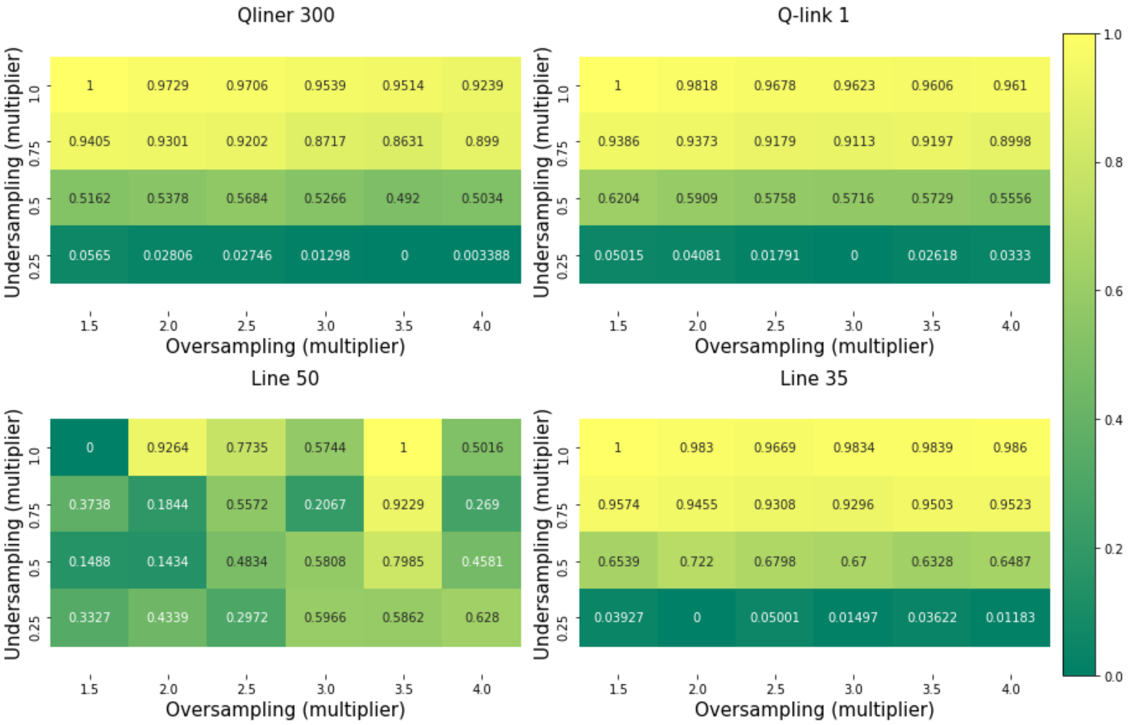


Figure 6.4: Evaluation of different sampling designs per line

results should have some slight variations. But the results from line 50 vary dramatically, and there is no clear rule of the optimal strategy. We consider the reasons are that line 50 has sparse bins when the ridership moves from small to large, especially at the large-value domain. And this is where we oversample and where we are most interested.

Thus, we conclude that we should resample the data without undersampling the majorities but oversampling the minorities by 50% for Qliner 300, Q-link 1, and line 35. While for line 50, we refer to the results from Fig. 6.4 to further our study.

At this step, we also find out two future works. First, the optimal combination of oversampling and undersampling strategy is at the left top corner, except for line 50. It means that the optimal combination can either be further investigated while the results might imply that we do not need to oversample at all. However, this is contradictory to what we want the model to be trained with more poorly represented instances. Some other strategies or methods can be performed here. Another possible work is to treat data like line 50 differently from other lines. The distribution and the binning of line 50 indicate a different entity in comparison with other lines, and therefore coming up with other strategies would lead to a potentially better result.

Aside from sampling and resampling, the imbalanced data problem can be also addressed by a weighted strategy (Chawla et al., 2002). It is also known as assigning different costs to classes during training (cost sensitivity weighting). Since RF tends to be biased towards the majority class, the cost sensitivity weighting places a heavier penalty on misclassifying the minority class. It assigns a weight to each class, with the minority class given larger weight (namely, higher misclassification cost). Chen and Breiman (2004) compared two ways in a classification context, they concluded that there is no clear winner. However, cost sensitivity weighting needs to use the entire training set while the resampling is computationally more efficient with large imbalanced data. Moreover, by assigning weights, it can introduce noises (mislabeled classes) and thus make the model vulnerable to those noises comparatively.

6.3. Model

Except for linear regression, the machine learning models that we selected to use in this study are non-parametric regression algorithm. Parametric tests are built upon assumptions that the distribution of the underlying population from which the sample was taken, e.g. linear regression. In contrast, non-parametric tests are not based on these assumptions, relying on the assumed shape or parameters of the underlying population distribution, such as KNN, GBM, and RF (Hopkins et al., 2018). Thus, the structure of these non-parametric models can fit closely with training data and tend to overfit. On the other hand, we also need to avoid underfitting where the algorithm is not learned enough of the structure of data.

Table 6.2: Optimal hyper-parameters of the non-parametric models

	Qliner 300	Q-link 1	Bus Line 50	Bus Line 35

GBM regression	learning_rate = 0.01
	n_estimators[1] = 13000
	max_depth[2] = 4
	min_samples_split[3] = 2
	min_samples_leaf[4] = 6
	subsample[5] = 1
	max_features[6] = 11
		learning_rate = 0.02
		n_estimators = 25000
		max_depth = 4
		min_samples_split = 2
		min_samples_leaf = 15
		subsample = 1
		max_features = 15
			learning_rate = 0.01
			n_estimators = 100
			max_depth = 2
			min_samples_split = 2
			min_samples_leaf = 1
			subsample = 1
			max_features = 98
				learning_rate = 0.02
				n_estimators = 22000
				max_depth = 4
				min_samples_split = 2
				min_samples_leaf = 20
				subsample = 1
				max_features = 25

K-Nearest regression	n_neighbors[7] = 14	n_neighbors = 10	n_neighbors = 8	n_neighbors = 8

Random Forest regression	bootstrap[8] = False
	n_estimators = 700
	max_depth = 20
	min_samples_split = 2
	min_samples_leaf = 4
	max_features = 10
		bootstrap = False
		n_estimators = 400
		max_depth = 25
		min_samples_split = 2
		min_samples_leaf = 2
		max_features = 20
			bootstrap = False
			n_estimators = 1000
			max_depth = 45
			min_samples_split = 2
			min_samples_leaf = 2
			max_features = 30
				bootstrap = False
				n_estimators = 1300
				max_depth = 20
				min_samples_split = 2
				min_samples_leaf = 2
				max_features = 35

[1]: Number of trees.
[2]: The maximum depth of a tree.
[3]: The minimal number of samples in a node for the node to be split.
[4]: The minimum number of samples in a leaf node.
[5]: The fraction of samples to be used for fitting the individual base learners.
[6]: The number of features randomly chosen as candidates for a split.
[7]: Number of neighbors to use.
[8]: Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.

For tree-based models, a process called regularization can help to use hyper-parameters to control the structure of the decision tree-based models and thereof GBDT and RF (Probst et al., 2019). The regularization hyperparameters of RF include: the number of trees in the forest, the maximum depth of a tree, the minimal number of samples in a node for the node to be split, the minimum number of samples in a leaf node, and the number of features randomly chosen as candidates for a split (details see Section 5.4.6). The hyper-parameters of GBDT involve: the learning rate from each tree to the next, the number of trees in the GBM, the maximum depth of a tree, the minimal number of samples in a node for the node to be split, the minimum number of samples in a leaf node and the number of features randomly chosen as candidates for a split. As that for KNN, it is a relatively simple tool. The only hyper-parameter we need to tune is the nearest K, which is calculated by the Euclidean distance.

We apply GridSearchCV to exhaustive search over specified parameter values for an estimator, which is provided by scikit-learn. The parameters of the estimator used to reach a higher R^2 are optimized by cross-validated grid-search over a pre-defined parameter grid. We again refer to R^2, the coefficient of determination, to determine the best performance of the model on the holdout dataset. Note that this whole process is adopted on the training dataset in the inner loop of the nested cross-validation. The outer loop is kept out to prevent an overestimation of the results and a local optimization.

We present the optimal hyper-parameters below in Table. 6.2 in line-wise. It is noteworthy that due to the random train/test split, it could vary slightly. In terms of line 50, it shows a high variation in the sampling design. Different set-up on the sampling design could result in a differently optimal hyper-parameters. We keep the GBM model as default for line 50 as it has the performance drops sharply after we tune the hyper-parameters. It is because the model learns from the training set and assumes the same distribution exists in the test set. However, this is false as we oversample the minorities by 3.5 times. However, RF regression and KNN do not see such a depreciation. We conclude that it is because only GBM learns the residual to optimize the target and is more sensitive to the noisy data (in our case is the change of the underlying distribution of data).

Different lines have different distributions of the data, and thus the hyper-parameters are not the same from one of another. The searching time of GBDT and RF can be unexpected long while KNN is the shortest due to its simpler configuration and fewer number of hyper-parameters to be found.

6.4. Summary

In this chapter, we select the variables that could strongly influence the ridership, including ridership-related, request-related, spatial-temporal-related variables. We also incorporate variables of request with timing advance. These variables would be put into the outperformed model in pairs to explore the prediction performance with a prediction lead time. We also think it is important to scale the independent variables as different units, different ranges, and distance sensible models are existing in our study.

Moreover, the task of addressing imbalanced data equally passed some challenges. The objective is to learn the different domains better. We apply the sampling/resampling design to tackle the problem over the cost sensitivity weighting because it is comparatively more efficient, effective, and doable. We train the model with the target after taking the natural logarithm because the original distribution of the target is highly right-skewed, and we want to comply with the regression models that work better with more symmetrical, bell-shaped distributions. Furthermore, this transformation facilitates the binning of continuous data into discrete one as there is less data on the right tail of the distribution. The binning strategy with Doane's formula helps us to resample the data by SMOTE that enhances the influence and representations of the minority but interested classes. We also practice a sensitivity analysis to determine the optimal sample designs. For the data distribution similar to line 50, extra attention is needed as it has sparse bins on the right tail, and thus the optimal strategy is random without clear rules. For other lines, the optimal strategies are also suited in the top left corner, which implies that better options can be explored. These can be the future work of this study.

The chapter wraps up with the optimal hyperparameters of the models. For non-parametric models, such as KNN, GBM, and RF, we apply GridSearchCV to exhaustive explore the optimal hyperparameters to prevent overfitting and underfitting, but also to reach a better performance.

7 Research Results

After we tune the hyperparameters of the models and choose the optimal method of sampling, we proceed to analyze the results from four case studies of our models. This chapter first discusses the performance of six models in line-wise based on our performance metrics. Then, the outperformed model is selected to explore the feature importance.

We also explore two ways of constructing the baseline models, containing zeros the missing trips or drops the missing trips. The reason is that the baseline model is built upon the ridership last week. If the trip number does not match, it will lead to a missing recording. Thus, we compare two ways of predicting this ridership and use the comparatively better baseline model to proceed.

Moreover, we investigate the linear regression model by starting with only the highest correlated variable and adding one more strongly correlated variable at a time. The inclusion of a new variable can tell us whether it significantly influences the model.

The assessment of the regression models ends with the construction of residual plots of the outperformed model. Residual plots display the residual values on the y-axis and fitted values, or another variable, on the x-axis. It can help us determine whether the residuals are consistent with random error. When the residuals are around or center on zero, it indicates that the prediction of the model is correct on average rather than over- or underestimation. In a regression problem, we often assume that the distribution of residuals should be normal and the degree of scattering is the same for all fitted values. By checking this plot iteratively, we could potentially improve the model, for instance, taking the natural logarithm and sampling design in the previous chapter.

Then, we discover the feature importance based on the outperformed model. We compare the MDI and permutation feature importance to conclude the usefulness of trip planner data and other influencing variables. Moreover, we apply PDP to assess how the change of trip planner related variables would affect the ridership.

The chapter wraps up with unveiling the usefulness of the trip planner data with certain prediction lead time. It helps us to know if we want to anticipate the ridership with a certain timing advance, what would be the performance and what would be the usefulness and effectiveness of such kind of data.

In the following content, we utilize the results from Qliner 300 for visualization for a clear presentation. However, the discussion refers to the results of all four case study lines. Visualizations of the other three cases see Appendix D.

The sub research question has been answered in this chapter is:

What is the performance and benefit of using such a prediction model?

7.1. Performance of the Models

In this section, we analyze the performance of all six models (baseline model, simple model, linear regression, GBDR, KNN regression, and RF regression) in line-wise. We assess the performance from multiple perspectives. We first examine the performance of models by the performance metric that we set up. The criteria are MAE, $RMSE$, R^2, and the accuracy after repeated random 5-fold cross-validation. To visualize the goodness of fit, we also plot the prediction and actual values, which is regarded as one of the richest information graphs. Then, we explain several interesting points that we find out when we deal with the baseline model and the simple model with weekly trends. Following, we discuss the significance of specific variables in the linear regression model where we include the variable one at a time according to their correlation ranking. Finally, we illustrate the residual plot that indicates the prediction of the model is correct on average or over- and underestimation. The residuals are also evaluated per scenario to see whether it matches our conventional understanding and previous studies.

7.1.1. Performance Metrics

We present the performance of the prediction based on the set of metrics of each case study line in the order of Qliner 300, Q-link 1, line 50, line 35. The set of metrics is from four aspects (details recall Chapter 3): first, MAE is the mean of the absolute errors. The absolute error is the absolute value of the difference between the predicted value and the actual value. Second, $RMSE$ is the standard deviation of the residuals (prediction errors). It is a measure of how spreading out these residuals are or say how concentrated the data is along the line of best fit. Third, R^2 represents the proportion of the variance for a dependent variable, explained by the variables in the regression model. Lastly, we apply a repeated random 5-fold cross-validation to evaluate the skill of the model on new data. The score we present is again R^2 as it is the most important measurement for the goodness-of-fit. The value recorded is the average of each fold, and it helps us to assess whether the model is spot overfitting. However, this cross-validation does not apply to the baseline model and simple model as there is no unseen data.

There are three topics that we will discuss in the following subsections. First, we design two kinds of baseline models due to the missing recordings of the trips of the previous week. One way is to zero the missing value, and the other way is to drop them. We select the outperformed one to apply in our study. Second, the simple model has a weirdly high error and the worst performance, and we will explain the problem later. Third, the detailed inclusion and strategy of implementing linear regression models will be put forward in the following subsection while we only report the best scenario here to establish the comparison.

Table 7.1: Performance of models (Qliner 300)

Qliner 300	**MAE (Person)**	**RMSE (Person)**	**R^2**	**Repeated Random 5-Fold Cross-Validation**
Baseline (Drop Null)	5.761	9.060	0.461	-
Simple Model (incl. weekly trend)	10.721	30.408	-6.179	-
Linear Regression (3 variables)	5.573	9.870	0.276	0.595
GBM Regression	4.664	7.114	0.624	**0.726**
K-Nearest Regression	4.699	7.277	0.607	0.666
Random Forest Regression	**4.123**	**6.357**	**0.700**	**0.726**

From the performance table of Qliner 300, we can see that both RF regression and GBM regression can capture the unseen data the best with a cross-validation score of 0.726. However, RF outperforms than GBM if we compare the other three metrics. Except for linear regression, all machine learning approaches show considerable improvement of the prediction, especially RF regression. The linear regression model only slightly improves the prediction error but functions worse in capturing the variance and has a large discrepancy, compared to the baseline model.

Table 7.2: Performance of models (Qlink-1)

Q-link 1	**MAE (Person)**	**RMSE (Person)**	**R^2**	**Repeated Random 5-Fold Cross-Validation**
Baseline (Drop Null)	7.744	12.588	0.287	-
Simple Model (incl. weekly trend)	9.920	26.071	-2.049	-
Linear Regression (3 variables)	9.095	68.056	-17.879	0.536
GBM Regression	9.462	13.667	0.239	0.796
K-Nearest Regression	5.622	9.235	0.652	0.698
Random Forest Regression	**4.329**	**7.045**	**0.798**	**0.826**

From the performance table of Q-link 1, RF regression beats other models from every criterion. From the aspect of the error and the variance of the errors, it is nearly two times better than the baseline model and has a three-time better goodness-of-fit, compared to the baseline model. The cross-validation score of RF is also the highest among all the models. It is unexpectedly to see that linear regression has a very high RMSE due to some exceptionally wrong prediction and a general overestimation of the target. Moreover, the significant negative value of R^2 indicates that a simple mean could work better than it. Failures of the simple model and linear regression have the same reason as that of Qliner 300, which we will explain in the following subsection. The margin between RF and other ML models is the largest in this Q-link 1 scenario, compared to others.

Table 7.3: Performance of models (line 50)

Line 50	**MAE (Person)**	**RMSE (Person)**	**R^2**	**Repeated Random 5-Fold Cross-Validation**
Baseline (Drop Null)	4.026	5.938	0.494	-
Simple Model (incl. weekly trend)	6.799	12.022	-1.423	-
Linear Regression (3 variables)	4.491	10.959	-0.780	0.515
GBM Regression	11.819	13.693	-1.778	0.657
K-Nearest Regression	**3.248**	**4.864**	**0.639**	0.686
Random Forest Regression	3.578	5.179	0.603	**0.770**

For line 50, GBM performs the worst due to the different distribution of data and a relatively large oversampling strategy. GBM is sensitive to “noisy” data (in our case, it is the underlying difference in the distribution of training set and test set). The reason is that boosting builds each tree on the residuals of the previous trees. Outliers will have much larger residuals than non-outliers, so gradient boosting will focus a disproportionate amount of its attention on those points.

> Bias refers to an algorithm that has limited flexibility to learn the true value while a variance occurs when an algorithm is sensitive to specific sets of training data. A high bias, low-variance model means that it is more consistent but inaccurate on average. In contrast, a high-variance, low-bias model means that it is accurate on average but inconsistent.

Moreover, we have a trade-off between KNN regression and RF regression. For all other cases, the model with the lowest bias tends to have the smallest variance as well but not line 50. The margin between KNN regression and RF regression is small, with respect to MAE, RMSE, and R^2. However, the R^2 score from repeated random 5-fold cross-validation is comparably large. Moreover, line 50 is the only one that has a different sampling design, and the design is relatively arbitrary. Thus, we regard a model that is less sensitive to training data is better. Hence, a higher R^2 from the repeated random 5-fold cross-validation as a repeatable process is more convincing, and therefore RF regression is considered as the best model.

Table 7.4: Performance of models (line 35)

Line 35	**MAE (Person)**	**RMSE (Person)**	**R^2**	**Repeated Random 5-Fold Cross-Validation**
Baseline (Drop Null)	3.423	6.158	0.341	-
Simple Model (incl. weekly trend)	4.225	10.338	-0.858	-
Linear Regression (2 variables)	3.732	12.577	-1.656	0.530
GBM Regression	2.243	3.922	0.742	0.811
K-Nearest Regression	2.458	4.204	0.703	0.747
Random Forest Regression	**1.938**	**3.493**	**0.795**	**0.831**

In terms of line 35, RF regression once again becomes the best model from every measurement. In comparison with the baseline model, RF regression is two times better from each criterion. Different from other cases, GBM regression ranks second and is slightly inferior to RF regression. The best performance of the linear regression model is with only two highly correlated variables, but it is even worse than the baseline model.

7.1.2. Actuality vs. Prediction Plots

To have some knowledge of the density of points, we also apply the same binning strategy with each model to calculate the average error of each bin. The number of bins is the same as Table. 6.2. This subsection has four parts. We first analyze the overall performance of the case study line and then explore the baseline models since there are many trips without a unique trip number. There are two ways to deal with this kind of issue and thus result in some differences. Next, we dive into the linear regression models by including one more variable every time to see the significance of the variable. Finally, we study the reason for the unsatisfied result of the simple models.

Scatter plots of actual vs. predicted values are one of the richest forms of data visualization. Ideally, all the data points should be as close as possible to the regressed diagonal line. For instance, if the actual value is y_i, then the predicted value should be reasonably close to y_i. It depicts that if the model has a high R^2, all the data points should be close to the diagonal line. In contrast, if R^2 is low, it means a weak goodness-of-fit, and thus the points are dispersed (away from the diagonal line).

All models

We present the visualization of actual vs. predicted values of Qliner-300 below in Fig. 7.1 and for other visualizations, see Appendix Fig. D.1, Fig. D.2, and Fig. D.3. There are two graphs in each sub-graph, namely a scatter plot with predicted value on the y-axis and actual value on the x-axis and a line chart of the average prediction error per bin.

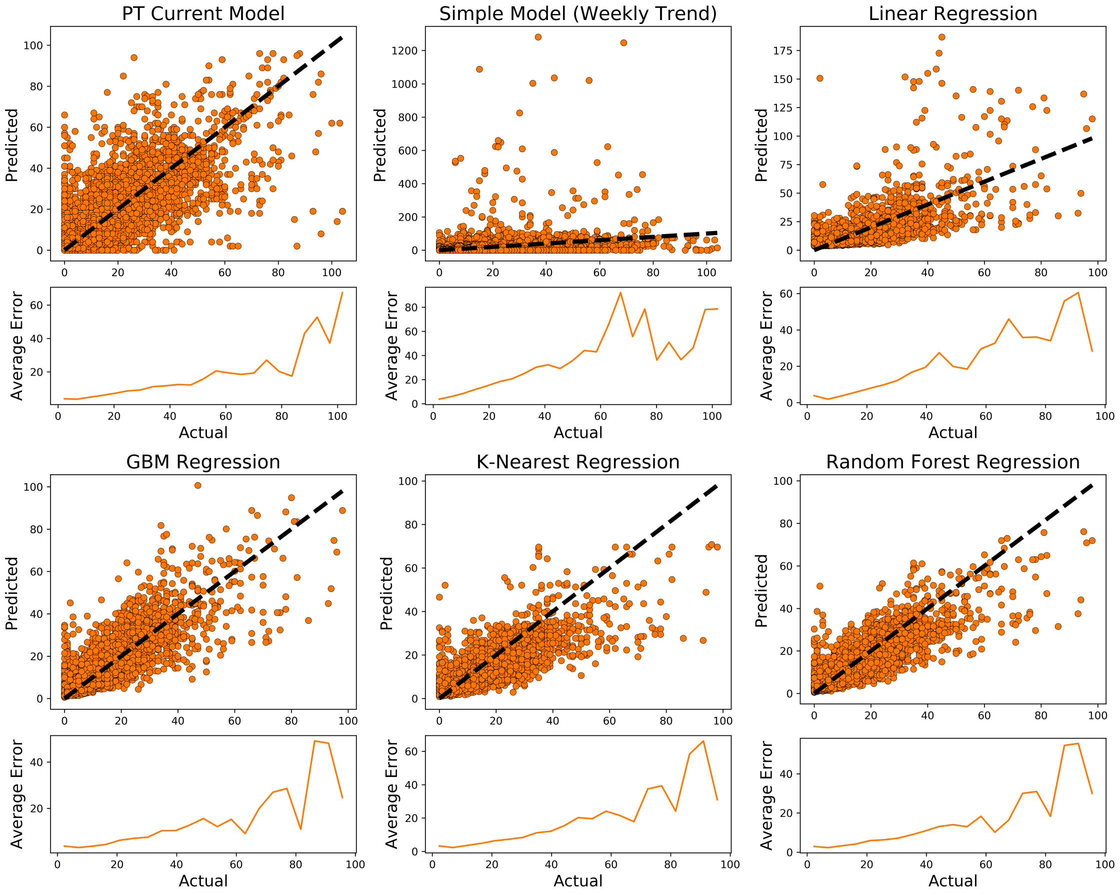


Figure 7.1: Prediction vs. actuality plot of Qliner 300

First off, every model can capture the relationship, and the difference lies to what extend it can function. The distribution of the scatters following what the performance tables present. When the R^2 is high, the points are close to the regressed diagonal line, for instance, random forest regression.

It is obvious to see that GBM regression, KNN regression, and RF regression capture most of the data points well when the actual value is less 70, except for line 50. GBM of line 50 always tends to overestimate the true value. It is not only reflected by the scatter plot but also the average error.

Second, we notice that the residuals are heteroscedastic from the visual presentation. All models have higher average error and bias when the actual value becomes higher. It means that the variance of the error is not constant across various levels of the dependent variable. As a result, it implies that the regression can be understated and potentially improved by adding more statistically significant variables. Although RF regression can capture the prediction the best, it tends to underestimate the result when the actual value is high. Thus, the average error is correspondingly large, implying the existence of heteroscedasticity. But, this ever-increasing average error becomes smaller when it reaches the last bin where fewer data points are suited. When the actual value is low, all models can function well while GBM tends to overestimate, KNN tends to underestimate while RF is relatively neural.

Third, we can assert that regardless of the line, machine learning models generally can not only better capture the quiet trips but also outperform than the baseline models when the trips become busy. However, even the comparatively best model (RF regression) becomes worse when the values become larger.

Fourth, the simple model has some exaggerated values that are far away from the line, which we can also observe in linear regression. The detailed analysis follows in the next subsection.

Given these points, RF regression outperforms than other models in most of the cases. However, regarding MAE, $RMSE$ and R^2, KNN performs better than RF of line 50 due to the different sampling design. Still, the RF has a lower variance. Thus, we conclude that RF regression is the outperformed model. It is obvious to see from the actuality vs. prediction plots that ML models usually predict the values better than baseline models, which means the points are close to the precisely fitted line. However, even the best RF regression model has heteroscedasticity. It means that the model performs worse when the actual value is higher and could be potentially improved by adding more statistically significant variables.

Baseline Models

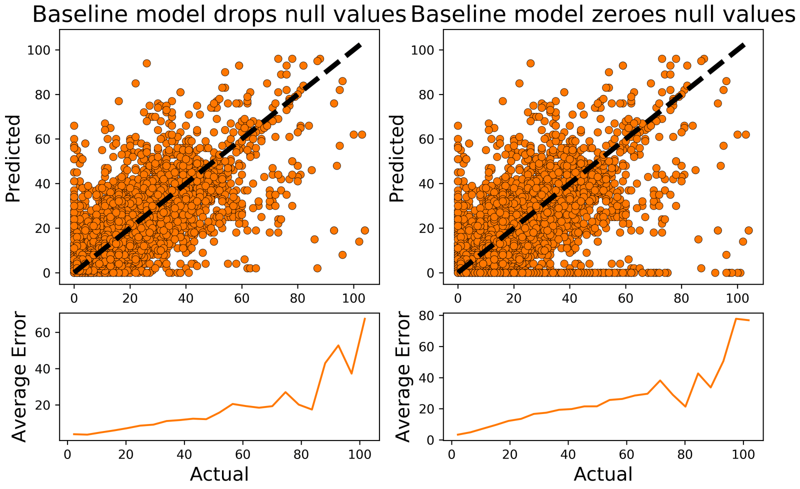


Figure 7.2: Prediction vs. actuality of baseline models (Qliner 300)

Missing recording of the bus trips results in two ways of constructing this baseline model. One way is to drop the lost trips, reflected by a null value, or to zero the null value and therefore make the ridership a 0. By making the missing trips zero ridership, we have a high error and have empty estimation in most parts of the actual value range. Hence, we use the comparatively better model -dropping null values- as the baseline model in our study.

However, by dropping the null values, we do not know what will be the possible number of passengers. From a practical point of view, it is hard to allocate sufficient vehicles to cater to demand. The missing trips could result in an underestimation when the ridership of last week is low or an overestimation when the ridership of last week is high. Moreover, it is hard to assert that the baseline model in our study is biased in which way since we do not know how well the estimation would be. Thus, we recommend that the PT operators can come up with a better way to deal with this problem and potentially elevate the recording system or trip numbering system. This simple model is indeed an efficient and effective one when the ridership is low.

Linear Models

We build the linear regression model with one more variable into account every time to test the significance of such a variable in a linear relationship manner. The order of inclusion is the number of requests, the average number of ridership, the ridership of last week, the average number of requests, the variance of request, and it is a holiday or not.

The first linear regression model only considers that there is a linear relationship between ridership and request. The second one assumes that we can calculate ridership linearly by using the number of requests and the historical ridership. The inclusion of the average number of ridership helps to reduce the error as we observe from the graph with a less average error and data points are more close to the perfectly predicted line. The third graph where the inclusion of the ridership of last week even helps the model become better.

However, none of the models from different lines can capture this relationship, which testifies our analysis above that there is no linear relationship between the two data, and a non-parametric model is needed to find this relationship. Moreover, the average error keeps increasing with the increase of the actual value, which is not like the model with more variables that have fluctuations.

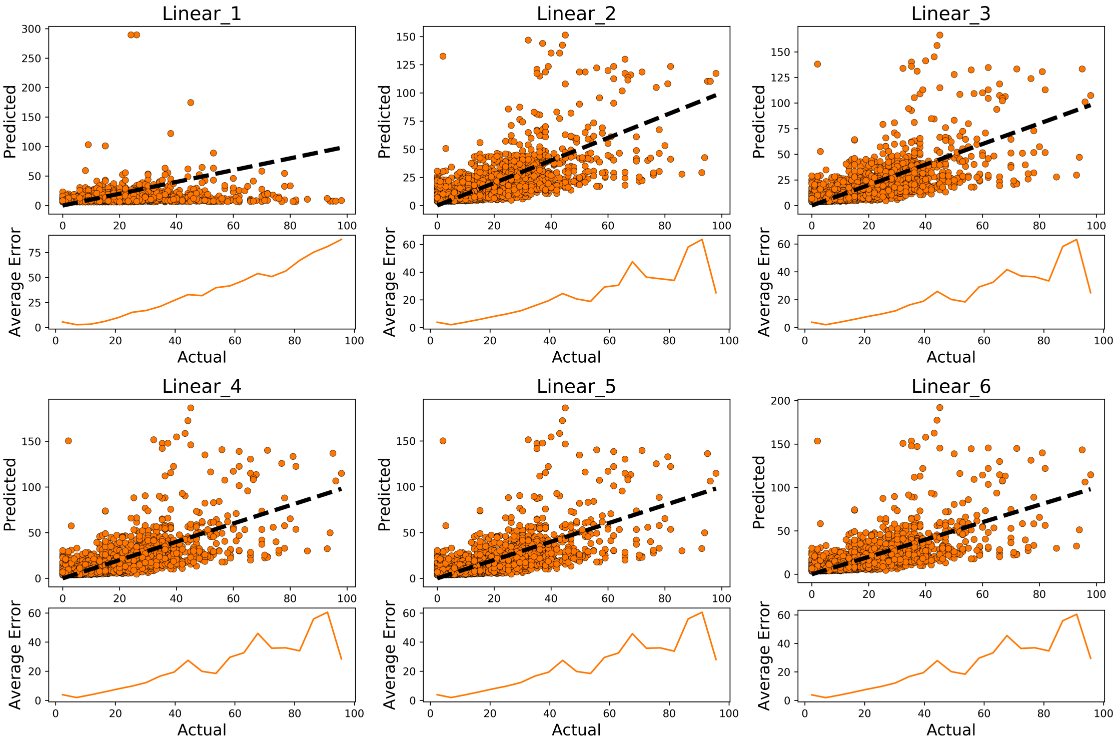


Figure 7.3: Prediction vs. actuality of linear models (Qliner 300)

The linear regression model of Qliner 300 among all case study lines is the best. All other lines have few identifiable outliers in the estimation, which are two exaggerated and thus result in an unsatisfied result. The best regression model of Qliner 300 with three variables is the only one that has a positive R^2. It means that using a simple mean as the estimation will perform better than this linear regression model for all other three case study lines.

Furthermore, the best linear regression model most of the time is with three variables, including the number of requests, the average number of ridership, the ridership of last week while line 35 is an exception with only the first two. When adding the fourth one, the average number of requests, the model performs worse. The best linear model is supposed to have a stable slope and the intercept when adding a new variable such that the coefficient of this new variable is zero. However, in our study, it is the tool that we used results in a higher error if this new variable is not significant and facilitates the prediction. Every time, a new set of slope + intercept will be estimated when a new variable is incorporated. The linear regression package from Scikit-learn can analytically solve the fitting by an ordinary least squares, and there are no iterative algorithms that converge the gradient descent. Also, this indicates that the new included variables that make the model worse are negative confounders that correlate with the variables that we have, for example, the number of requests and the average number of requests.

Simple Models

The simple model with weekly trend performs unexpectedly unsatisfied. We initially assume that by taking the ratio between the ridership of yesterday and yesterday of last week, the model can capture the weekly trend. For instance, if this week is the autumn holiday, by overestimating the demand yesterday, it will lower the estimation today by setting the multiplier to a small value and vice versa. However, from the prediction vs. actual value plots above, we can see the model performs disappointed because there are too many extremely high predictions. Those data points can have a predicted value of more than 1000 persons, which is impossible.

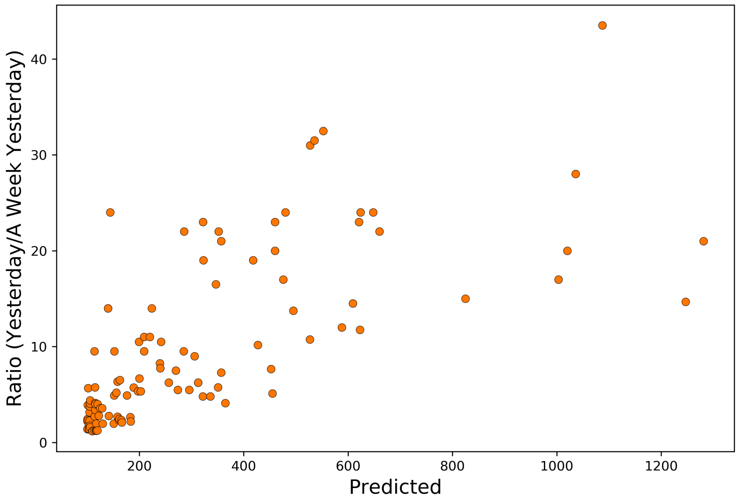


Figure 7.4: Prediction vs. actuality of the simple model (Qliner 300)

Figure 7.4 presents the data points of prediction that are above 100 persons (almost impossible) on the x-axis and the coefficient (ratio) between the ridership of yesterday and the ridership of yesterday a week ago on the y-axis. We can observe that there is a relationship when the prediction is high and too much overstated, the ratio is also too high. Most of the overestimations are related to the same trip that was too crowded last week on several specific sections. Note that the dimension of the ratio is also a section of a trip.

We still consider that this kind of simple model can outperform the baseline model but maintain the feature of efficiency and computationally friendly attributes. The later study can improve the model by adding smoothers, for example, the ratio of the last trip, or the ratio of the last section, or the average ratio of the previous sections.

7.2. Residual Analysis

In this section, we use the outperformed model -RF regression- to further our analysis. Residuals are differences between the predictions from the model and the measured outputs from the validations data set. Thus, residuals represent the portion of the validation data not explained by the model.

A residual plot is a graph that shows the residuals on the vertical axis and the prediction values on the horizontal axis (sometimes, it is one or several independent variables to explore the relationship, linear or non-linear). Each point represents a residual. The distance from the line at 0 is how undesirable the prediction was for that value.

Ideally, the residuals should be symmetrically distributed along the line, and tend to cluster towards the middle of the plot, which is the zero line. The distance between residuals and the line should have a low value, and there is no clear pattern.

We present the residual plot of Qliner 300 below in Fig. 7.5, and the other ones see Appendix D.4.

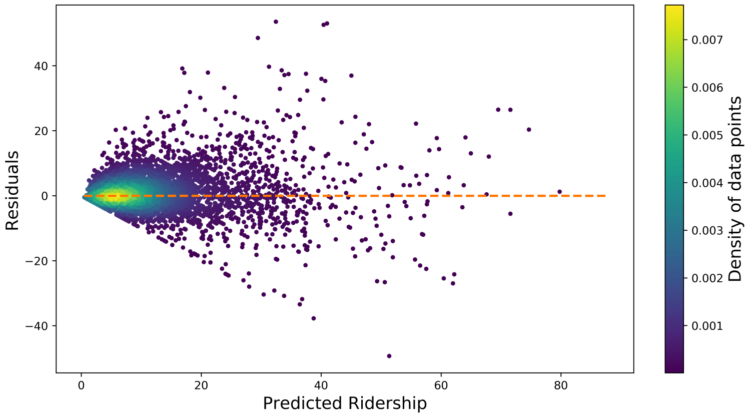


Figure 7.5: Residual plots of Qliner 300 (Random Forest Regression)

In the figure, we use the color bar to represent the density of the data points. Most of the points are densely clustered between 0 and 10 of the predicted ridership and around 0 of the residuals. Only that part is bright color as the density is overwhelmingly large.

However, we do identify a clear pattern of the diagrams, which is the heteroscedasticity. Heteroscedasticity means that the residuals get larger as the prediction moves from small to large (or from large to small, which is not the case here). It does not inherently indicate a problem, but it implies that there are potentials for the improvements. Most often, there are two ways: the first one we have already applied, transforming the variable. Yet, there are various ways of doing so. It could be a different way that gives a better result. The second one is to add more variables so that the model can capture the relationship between dependent and independent variables better.

Moreover, we can spot some outliers in the graphs. In some cases, a model can pivot to try to get closer to the outlying point at the expense of being close to all the others and end up being entirely wrong. However, RF regression is not sensitive to outliers because tree algorithms split the data points based on the same value. So the outlier will not affect that much to the split. Moreover, we deprive the bootstrap in the RF so that it will not create a disproportionate amount due to the large residuals from the outliers. And this is the reason RF performs better than GBM in general, and it testifies that when we oversample line 50 larger than other cases, GBM performs the worst due to more outliers or larger misleading residuals that have been created.

Except for line 50, all other lines show a symmetrical and balanced distribution of residuals along the line. Line 50 has the problem of the y-axis unbalanced slightly. The data distribution of it is the only one significantly different from others, and the sample design is arbitrary. After binning, the distance between bins becomes sparse when it approaches the right side, which is the range of high ridership. We oversample the minorities with 250 % more, which is randomly one of the optimal sample strategies and thus results in too many samples at the high-value domain. As we see from Fig. D.2, the RF regression graph that more data points are above the fitted line and implies that our model overestimates the ridership. Hence, the residual plot shows that between low and medium value ranges, we have more negative residuals, and when it approaches the high value, some positive residuals are easily identifiable. Therefore, ridership distribution that is similar to line 50 needs a different approach to benefit a better performance of the model.

After we have an initial understanding of the residuals, we want to assess whether or not the errors distribute normally. It helps us to understand to what extend the model explains the data and to what extend the heteroscedasticity is. Essentially, it guarantees correct p-values.

We apply the histogram and the Q-Q (quantile-quantile) plot to check whether or not it is reasonable to assume that the random errors inherent in the process have been drawn from a normal distribution. The normality assumption is needed for the error rates we are willing to accept when making decisions about the process. If the random errors are not from a normal distribution, incorrect decisions will be made more or less frequently than the stated confidence levels for our inferences indicate.

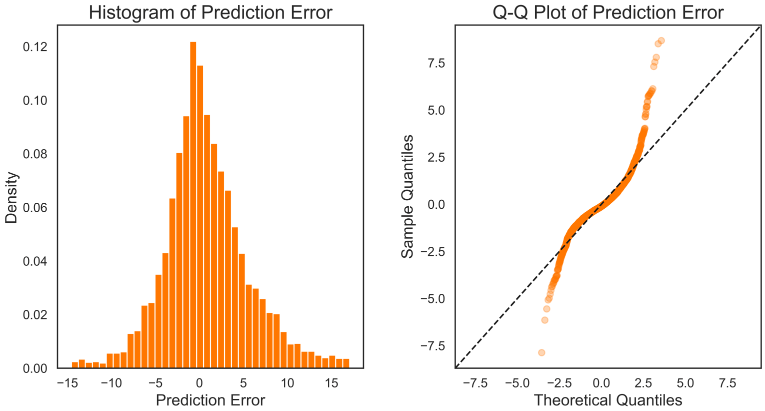


Figure 7.6: Distribution of errors of Qliner 300 (Random Forest Regression)

Figure 7.6 reports the distribution of error in a histogram manner and a Q-Q plot of Qliner 300, respectively (for other lines, see Appendix D.5). Except for line 50, the histogram of the prediction error shows a normally distributed format with the mean around zero and equally spread towards the two ends. For line 50, the distribution is slightly right-skewed (positive skewed), which means that most of the prediction errors are clustered in the negative part of the distribution, representing an overestimation. All histograms of the errors present a bell-shape distribution, regardless of the skewness. However, by solely referring to the histogram, it is hard to discern deviations from normality than with the more specifically-oriented Q-Q plot.

In general, the curvature of Q-Q plots indicates skew distributions. Downward concavity represents negative skewness (long tail to the left). In contrast, upward concavity symbolizes positive skewness. On the other hand, S-shaped Q-Q plots indicate heavy tails, or an excess of extreme values, relative to the normal distribution. And it is intuitive and visible to see if the curve fits the fitted straight line.

The Q-Q plots of our cases show that most of the data points sit on the line. However, it also presents a fat tail and indicates that there is the existence of leptokurtosis. The reason is that too many residuals at the tail that are too far from the predicted line. In other words, there is more data located at the extremes of the distribution and fewer data in the center of the distribution when it approaches the two ends. Combined with the plots and analysis above, this means that when we want to predict a high actual value, a broader fluctuation is expected, which results in a greater potential to return biased (either low or high) values.

We further investigate the residuals by differentiating how many predictions we made are overstated and understated. Overestimation and underestimation cause differences in the PT operations. If the aim is a higher level of service, overestimation is more acceptable than underestimation. The other way around if it is a cost-oriented operation. The analysis of the overestimation and underestimation with residuals is presented in Table. 7.5.

Qliner 300 is the only line that has more underestimation than overestimation. The difference is approximately 7%. Q-link 1 is the most balanced one with a similar percentage of over- and underestimation of the ridership. Line 50 has the most difference in the biased prediction, and the margin

Table 7.5: Residual analysis of overestimation and underestimation

	Percentage		95th Percentile Absolute Error(Person)		Average of top 5 Percentile Absolute Error(Person)	
	Overestimation	Underestimation	Overestimation	Underestimation	Overestimation	Underestimation
Qliner 300	46.77%	53.23%	10.125	14.492	16.486	22.013
Q-link 1	50.51%	49.49%	12.079	18.285	18.353	25.833
Line 50	65.08%	34.92%	11.291	9.839	14.825	15.315
Line 35	57.10%	42.90%	5.733	8.124	9.365	13.504

can be as high as 30% due to its different sample design. We oversample the minority by 250% and thus result in an overestimation. Line 35 has more overstatements, which is around 15% more than understatements.

A percentile says that for that percentage of the time, the data points are below the resulting value. A 95th percentile tells us that 95% of the time data points are below that value, and 5% of the time the points are above that value. We then explore the 95th percentile of the absolute error to know most-of-the-time cases and the top 5th percentile of that to understand what is the average of extreme cases.

Line 35 has the lowest 95th percentile absolute error, no matter overestimation or underestimation. It means that most of the time, the RF regression model for line 35 can anticipate the passengers on board right with the lowest error among all cases. Plus, it has the smallest margin of increase if we compare the 95th percentile with the average of the top 5th percentile.

Line 50 has a higher 95th percentile error of overestimation than underestimation. However, the average top 5th percentile of underestimation error is severer than overestimation. It indicates that the majority of the time, the model tends to have an overestimation, and for extreme cases, it is the opposite. It is in line with the resampling strategy that we applied where we oversample the minority classes at the large-value range.

Even though we did not undersample, but only oversample the minority classes by 5% for Qliner 300 and Q-link 1. They have more tendencies to underestimate the actual values, and this tendency is much more profound when we see the average of the top 5 percentile residuals.

The last step of this section is to investigate the residuals per scenario to see how residuals unfolded in each temporal circumstances. We show the box plot of residuals of Qliner 300 per scenario in Fig. 7.7 and all other cases see Appendix D.6.

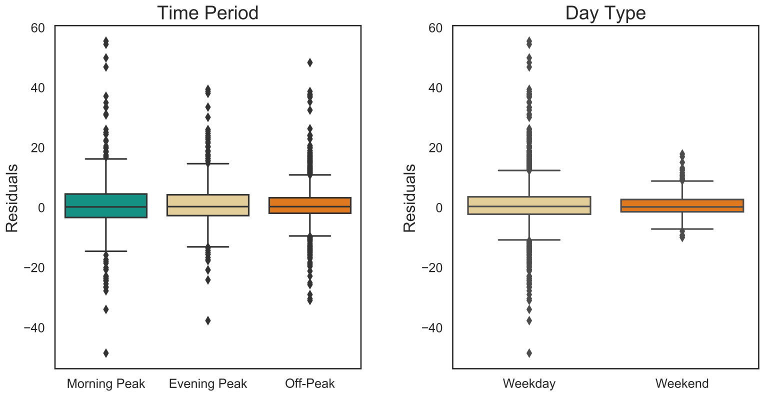


Figure 7.7: Residuals of Qliner 300 per scenario (Random Forest Regression)

Boxplots are a standardized way of displaying the distribution of data based on a five-number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”). By using such plots, we can compare the range and distribution of the residuals from prediction.

In terms of the residuals per period, Qliner 300 is the only one that has a higher variance of residuals during the morning peak, reflected by the box size in comparison with evening peak and off-peak. The reason could be it is an inter-city fast line, and its commuting passenger volume can be very high during the morning peak and thus result in a difficult condition for capturing the prediction variance. While for other lines, it is the evening peak that has the largest variance because it is the second-highest commuting time. However, people have a non-uniform off-duty time, and thus the variance is easy to fluctuate. Off-peak, among all cases, has the smallest variance due to its relatively lower number of passengers. Lastly, the box of line 50 during the evening peak is a bit drifted to negative as the model tends to overestimate the ridership. Plus, there is a higher fluctuation during the evening peak and thus causes trouble for the model. Line 35 performs the best with the smallest box during every period, which means the model can capture the variance better.

Concerning day type, the variance of the weekday is higher than the weekends on every case study line. On weekdays, the outliers are larger and can be as high as 60. It is in line with the common understanding that the passenger flow varies stronger during weekdays as people have more demand for transport while less during weekends. Therefore, the variation during weekdays is higher and is much harder to capture.

7.3. Feature Importance

In this section, we use the outperformed model -RF regression- to explore the feature importance. There are two ways to calculate the feature importance. One is to use MDI, and the other one is to perform permutation on the features. We compare the results from the two methods and conclude the importance.

Since the study has an emphasis on the relationship between ridership and trip planner requests, we further turn to PDP to investigate the development of such a relationship. We focus on the number of requests, the average number of requests, and the variance of the request.

7.3.1. MDI and Permutation Feature Importance

As the outperformed model in our study is RF regression, this kind of tree-based models provides a measure of feature importance via the MDI.

Impurity is quantified by the splitting criterion of the decision trees (Gini, Entropy, or Mean Squared Error). However, this method can give high importance to features that may not be predictive of unseen data when the model is overfitting. Permutation-based feature importance, on the other hand, avoids this issue, since it can be computed on unseen data. This unseen data is referred to as the test dataset.

Another disadvantage of impurity-based feature importance is that it normally favors the high cardinality features (typically numerical features) over low cardinality features such as binary features or categorical variables with a small number of possible categories. However, permutation-based feature importance does not have such a bias.

Each method has its pros and cons. When two features are correlated, and one of the features is permuted, the model will still have access to the feature through its correlated feature. It will result in a lower importance value for both features, where they might be important. This is permutation importance with multicollinear or correlated features.

Therefore, we apply both methods to calculate the feature importance and compare the results from both methods to gain a complete understanding of the importance of features, especially we apply the permutation feature importance on the test set. The results we derive from Qliner 300 is shown above in Fig. 7.8. For other cases, see Appendix D.7. Given the limitation of the page width, we convert the name of the section to a different code the name. The change of section names refers to Appendix D, Table. D.1.

No matter the method is MDI or permutation feature importance, the first five important features are the average number of ridership, the ridership of last week, the number of requests, the average number of requests, and the variance of requests. But the order of the most crucial five variables varies a bit depending on the line. The most important feature is always the average number of ridership, which we derived from the smart card data, except for Q-link 1 MDI feature importance. As many articles expounded, the smart card data is absolutely a sound basis for the ridership prediction. The

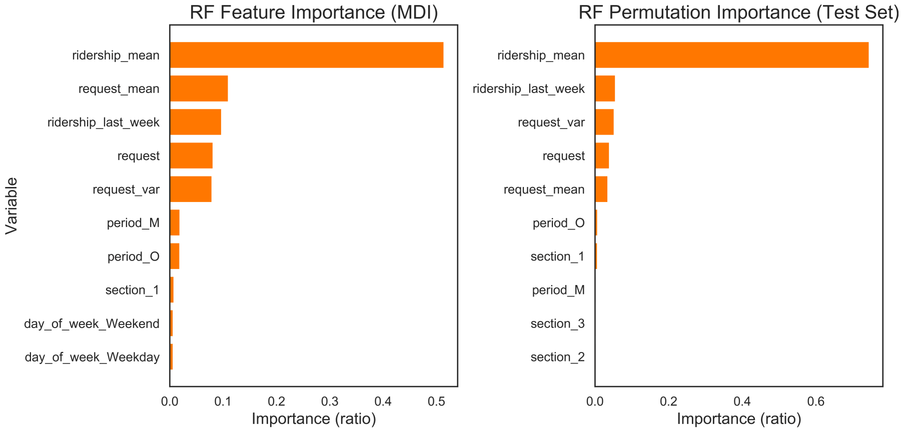


Figure 7.8: Feature importance of Qliner 300 (Random Forest Regression)

number of trip planner requests can also play a vital role in the ridership prediction with the importance of approximately 20%. But this value fluctuates across the lines and depends on the method. Still, it often ranks within the first three substantial features.

In terms of categorical variables, morning peak and off-peak can influence the model but with a minor effect. It holds the same to some specifically spatial sections, such as the section around the working place and the place around the railway station. Unexpectedly, the temporal effect from the day of the week is not significant. The direction of the line, although it does not have a strong correlation with the ridership from the data analysis, it sometimes can be the top 10 influencing factors with a minor effect.

Our feature importance analysis also backs up the argument that the impurity-based feature importance can inflate the importance of numerical features. From the figures, the MDI feature importance of numerical features, starting from the second highest, is always higher than the permutation feature importance. The results support the finding that impurity-based importances are biased towards high cardinality features. Impurity-based importances are computed on training set statistics and therefore do not reflect the ability of feature to be useful to make predictions that generalize to the test set. Although in our study, we do not have a categorical variable that significantly influences the prediction of the target, the permutation importance can be a cheerful alternative when applying RF regression models as it can be performed on the unseen, held out test set. By combining both of the approaches, we can have relatively complete knowledge of what are the important influencing factors in both the training set and test set.

7.3.2. Partial Dependency Plot

A partial dependence (PD) plot depicts the functional relationship between a single variable or a small bunch of input variables and predictions. It shows how the predictions partially depend on the values of the input variables of interest. It is worthwhile mentioning that the unit of partial dependency is not concrete with a specific meaning. It just tells us for the given value(s) of feature(s) what the average marginal effect on the prediction is. Furthermore, PDPs assume that the target features are independent of the complement features, and this assumption is often violated in practice.

We present the PDP of Qliner 300 in Fig. 7.9. For other cases, see Appendix D.8. The objective of this study emphasizes the importance of trip planner data in ridership prediction and how this importance develops with the change of the ridership. Therefore, we analyze how our target, ridership, depends on a single input, including the number of requests, the average number of requests, and the variance of this request. Overall, the PDP of every case study line shows a similar pattern.

The PD between ridership and request develops almost the same across four cases. The distribution of requests is right-skewed, and it is clustered in the low-value range. When the number of requests

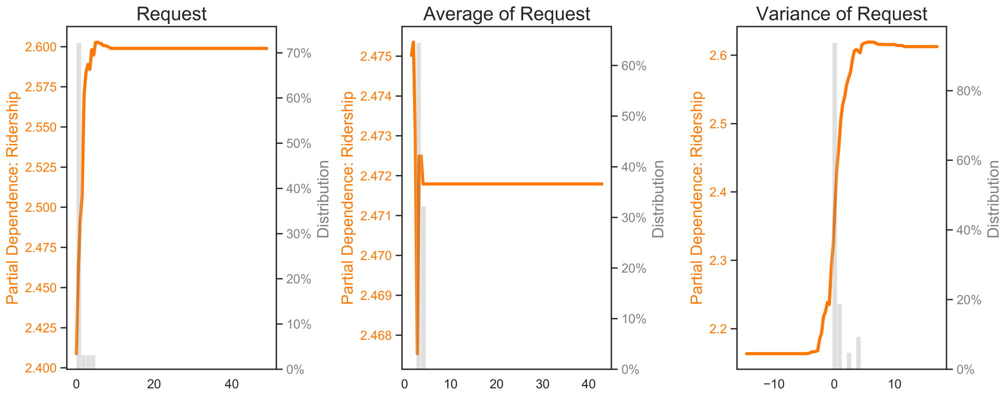


Figure 7.9: PDP of Qliner 300 (Random Forest Regression)

is less than 10 (low-value range), there is a strong positive correlation between ridership and request. When the number of requests is larger than 20, the PD between ridership and request remains stable as there are fewer recordings so that the model could probably not learn a meaningful prediction for this range. Therefore, we can still conclude that ridership shows a positive correlation with requests.

The PD of the average number of requests is stable after around 3 and 4. It is because there is not much training data in all cases. In contrast, it varies in the low-value range, depending on the type of the line. For Qliner 300 and line 50, the potential ridership is increasingly inhibited when the average number of requests is from 0 to 1. It is the same increasing trend of the other two lines. Then, when the average number of requests increases from 1 to 2, the ridership climbs with it. However, the ridership drops with the increasing average request from 2 to 3 and then remains stable.

The last PD we investigate is between ridership and the variance of the request. From the figure, we can see that there is a clear positive correlation between them from small negative values to the minor positive values as both ends remain flat. It means that when the variance increases from negative (less than expected, compared to average) to positive (more than expected), the number of passengers on board would be correspondingly higher.

7.4. Prediction Time Horizon

In this section, we analyze the performance of the RF regression model with the same configuration and the same sampling design with a certain prediction lead time. In other words, how the model performs with trip planner data that is further ahead of time, such as 10 minutes, 15 minutes, and 30 minutes. Note that the timing advance is calculated by vehicle start time from the trip advisor as this is when passengers would realize their trips.

The objective of this step is to investigate how we can well inform PT operators to match the flexibility of them. Although it is common that the more data, the better, trip planner as a trip advisor app, is often used with ahead of time. Thus, what would be the performance of such a model with fewer data and with trip planner data that is further ahead of time arouses the interests.

We explore the model with the optimal hyper-parameters that listed in Table. 6.3 and with the same performance metrics that we study in the first section. Since we also want to know what role the trip planner with timing advance plays and how important it is, we apply MDI feature importance to see the feature importance of the number of requests, the average number of requests, and the variance of requests. Note that we apply MDI because we want to know how it will affect the training of the model. The tables of model performance with trip planner data that is asked earlier than travel time are presented below.

The number of data drops from all-time enclosed to certain lead time. This drop is steady if we are interested in further ahead in time. We see the same trend in correlation. Among all scenarios of Qliner 300, the model outperforms with trip requests with only 10-minute in advance. It is the closest time to the vehicle start time and contains most of the data. In terms of the importance of trip planner data, the number of requests drops. But the average number of requests and the variance of the requests

Table 7.6: Performance of RF regression with timing advance of trip planner requests (Qliner 300)

Qliner 300	Number	Correlation	MAE (Person)	RMSE (Person)	R^2	Request	Request_mean	Request_var
All	124350	0.397	4.131	6.385	0.697	0.090	0.080	0.110
10-minute	101313	0.328	**4.199**	**6.612**	**0.675**	**0.070**	**0.070**	**0.110**
15-minute	91176	0.304	4.269	6.647	0.672	**0.070**	0.060	**0.110**
30-minute	70824	0.274	4.215	6.623	0.674	0.060	0.060	**0.110**

remain almost the same. There is an interesting phenomenon that although 30-minute timing advance is the furthest time ahead, the model performs with lower MAE and RMSE, plus a higher R^2 than 15-minute ahead. It indicates that sometimes further ahead in time could contain more useful information. For Q-liner 300, if we only use trip planner data with a certain time ahead, it will not impose a strong negative effect.

Table 7.7: Performance of RF regression with timing advance of trip planner requests (Q-link 1)

Q-link 1	Number	Correlation	MAE (Person)	RMSE (Person)	R^2	Request	Request_mean	Request_var
All	122913	0.618	4.308	6.992	0.801	0.340	0.090	0.100
10-minute	90408	0.541	4.375	7.156	0.791	**0.300**	**0.070**	**0.100**
15-minute	77920	0.501	**4.317**	**7.099**	**0.795**	0.230	**0.070**	**0.100**
30-minute	56664	0.458	4.387	7.181	0.790	0.200	0.060	**0.100**

Moving to Q-link 1, the number of requests again declines from all-time to further ahead in time, along with the correlation. Unlike Qliner 300, 15-minute ahead of time is the best one out of the three. The second comes with 10 minutes in advance. However, the importance of the number of requests drops sharply. But the average number and the variance of requests stay approximately stable. The model again does not deteriorate significantly if we only utilize requests ahead.

Table 7.8: Performance of RF regression with timing advance of trip planner requests (line 50)

Line 50	Number	Correlation	MAE (Person)	RMSE (Person)	R^2	Request	Request_mean	Request_var
All	91719	0.560	3.609	5.241	0.593	0.210	0.160	0.060
10-minute	72666	0.411	**3.897**	**5.656**	**0.526**	**0.130**	0.100	0.070
15-minute	63489	0.411	4.408	6.337	0.405	**0.130**	**0.110**	0.070
30-minute	44667	0.313	5.853	7.472	0.173	0.090	0.080	**0.090**

Then, it comes to line 50. Line 50 is the only that we apply a different resampling strategy, and it seems that the overall tendency from all-time requests included requests with timing advance is similar to Qliner 300. However, the difference is that the model decays gradually from real-time to further ahead in time from every criterion. Particularly, the scenario of 30-minute ahead has a R^2 of only 0.173. In such a model, the importance of requests and the average number of requests drops almost half. But we see an unexpected increase in the variance of the requests. In short, the model performs increasingly worse when we are interested earlier.

Table 7.9: Performance of RF regression with timing advance of trip planner requests (line 35)

Line 35	Number	Correlation	MAE (Person)	RMSE (Person)	R^2	Request	Request_mean	Request_var
All	25822	0.707	1.839	3.298	0.817	0.310	0.180	0.060
10-minute	18231	0.630	1.910	3.460	0.800	**0.300**	**0.120**	0.060
15-minute	15572	0.576	1.943	3.543	0.789	0.280	0.110	**0.070**
30-minute	10908	0.472	**1.862**	**3.409**	**0.805**	0.210	0.100	**0.070**

The best scenario of line 35 is different from other lines, which happens to be the furthest in time, namely the trip planner requests with at least 30 minutes ahead. It means that the 30-minute scenario holds the most useful information in ridership prediction. The initial model is already the best among the four cases, and all other scenarios executed are just a bit worse. The importance of the number of requests and the average number falls in half. But, there is an increase in the variance once more.

To sum up, by using trip planner data with prediction lead time, the performance of the random forest model does not deteriorate to a great extent, except for line 50. Even though the number of data and the correlation between requests and ridership drops, the model functions essentially the same. It is interesting to see that the importance of the number of requests and the average number of it decreases. But we see an increase in the importance of the variance of the requests. For the majority of the cases, the model does not perform worse with the increase of the timing advance. Two of the four cases show us that model with trip planner requests that are 10-minute ahead outperforms than others, indicating that the information stores nearest are more valuable.

7.5. Prediction with Requests at Different Times

In section 4.2.1, we explore how people behave differently during different times by using such a trip planner. We conclude that people prefer checking travel information in 0 to 10 minutes of a short time range during the daytime. In contrast, people would like to plan their trip at least 8 hours before night. Therefore, we analyze the performance of the RF regression model with the same configuration and the same sampling design during different times in this section.

We desire to unveil how the trip planner information could function during different times, and during what period, the trip planner data is of the most usefulness. So that we could know by utilizing the trip planner during what time of the day would result in a better performance of the ridership prediction.

Like the previous section, We execute the model with the optimal hyper-parameters that listed in Table. 6.3 and with the same performance metrics that we study in the first section. MDI feature importance is again used to check the feature importance of the number of requests, the average number of requests, and the variance of requests. We present the performance of the RF regression model with requests at different times of every case study below from Table 7.10 to 7.13.

Table 7.10: Performance of RF regression with requests at different times (Qliner 300)

Qliner 300	Number	Correlation	MAE (Person)	RMSE (Person)	R^2	Request	Request_mean	Request_var
All	124350	0.397	4.279	6.538	0.682	0.100	0.110	0.090
Morning	29644	0.307	4.546	6.967	0.639	**0.050**	**0.120**	**0.080**
Noon	57783	0.230	**4.401**	**6.774**	**0.659**	**0.050**	**0.120**	0.070
Evening	33792	0.072	4.711	7.337	0.600	0.030	**0.120**	0.080
Night	3131	0.123	4.509	6.907	0.646	0.010	**0.120**	0.090

Table 7.11: Performance of RF regression with requests at different times (Q-link 1)

Q-link 1	Number	Correlation	MAE (Person)	RMSE (Person)	R^2	Request	Request_mean	Request_var
All	122913	0.618	5.098	8.443	0.709	0.360	0.110	0.110
Morning	25423	0.308	5.366	8.803	0.684	0.060	**0.150**	**0.100**
Noon	55479	0.436	**5.191**	**8.499**	**0.706**	**0.130**	**0.150**	0.070
Evening	39173	0.247	7.112	11.527	0.458	0.090	**0.150**	**0.100**
Night	2838	0.145	5.567	9.227	0.653	0.020	**0.150**	**0.100**

Table 7.12: Performance of RF regression with requests at different times (line 50)

Line 50	Number	Correlation	MAE (Person)	RMSE (Person)	R^2	Request	Request_mean	Request_var
All	91719	0.560	4.202	6.096	0.449	0.230	0.070	0.170
Morning	22890	0.376	5.707	8.022	0.046	**0.110**	**0.100**	**0.090**
Noon	42251	0.350	**4.194**	**5.802**	**0.501**	0.100	0.090	**0.090**
Evening	24615	0.096	4.489	6.305	0.411	0.040	**0.100**	**0.090**
Night	1963	0.134	8.126	11.346	-0.907	0.020	0.090	0.060

Table 7.13: Performance of RF regression with requests at different times (line 35)

Line 35	Number	Correlation	MAE (Person)	RMSE (Person)	R^2	Request	Request_mean	Request_var
All	25822	0.707	2.257	3.882	0.747	0.340	0.070	0.200
Morning	7191	0.479	3.293	5.677	0.459	0.130	**0.130**	0.100
Noon	12568	0.489	**2.373**	4.343	0.683	**0.140**	0.110	**0.140**
Evening	5488	0.050	3.601	6.500	0.291	0.050	0.110	0.090
Night	575	0.187	2.451	**4.307**	**0.688**	0.040	0.110	0.080

All four cases receive most of the requests from 10:00 to 16:00 (noon). Thus, the trip planner requests during noon roughly have the strongest correlation with ridership. Three cases out of four have the best performance during that period, except for line 35. Still, all cases show a satisfactory performance during noon, compared to all requests included. All three metrics do not change to a large extend but just slightly. Line 50 is the only case that has a better performance by utilizing requests from noon, although the improvement is tiny. It means that the requests forwarded during noon are the most informative.

Users send very few requests at night (from 22:00 to 4:00), which is around 5% of all requests. However, the majority of the cases do not show a significant deterioration of the performance, except for line 50. Particularly, line 35 even has its best performance of RMSE and R^2 with requests from the night. The notable contrast is with line 50 from which we observe R^2 becomes approximately -0.9. It means the model fails to capture the relationship and yield untruthful results. It is probably due to the sampling design in which we exaggerate the minorities. When the sample size is small, the underlying difference between the training set and test set has been overlearned.

We see a worsening performance of the models during the evening (from 16:00 to 22:00), regardless of the line. Two of the four cases almost see a double MAE. Line 50 is the only one that deals with the problem well in which evening is the second-best scenario. Hence, we consider that when the sample size is large, the RF regression model of line 50 can learn a meaningful result. However, for other cases, it is the evening when the RF model can capture less.

Regarding the feature importance of request-related variables, we see an increase in the average of requests when we utilize partial requests, no matter the scenario. The request importance drops sharply when the number of requests is low. However, we do not observe the same trend in the variance of requests. Those variables are still crucial as they take up around 20% of the importance in total.

7.6. Summary

We execute the models that we selected in the previous chapter with the desired inputs in this chapter. We measure the performance of the models by the metrics that we set up, including MAE, RMSE, R^2, and the R^2 from repeated random 5-fold cross-validation. By applying these metrics, we can know the error, the variance of the error, the extent that our inputs explain the target, and the model performance on reality. For the majority of the cases, Random Forest Regression outperforms than other models with a landslide. It is line 50 that we have a different sampling design that gives us a different result. In such a case, we have KNN regression as the best model on the MAE, RMSE, R^2 but we have a better result from the cross-validation. Still, we consider RF is better as the result from cross-validation shows that it is less sensitive to training data and due to cross-validation is an iterative process, the result is more convincing.

During the analysis, we also explore the actuality vs. prediction plots where we can see that every model can capture the relationship, and the difference lies to what extend it can function. All models have larger average error and bias when the actual value becomes large, imply the existence of heteroscedastic. Moreover, we can assert that regardless of the line, machine learning models generally can not only better capture the quiet trips but also outperform than the baseline models when the trips become busy. When the actual value is low, all models can function well while GBM tends to overestimate, KNN tends to underestimate while RF is relatively neural.

Then, we further investigate the baseline model, the linear model, and the simple model. The baseline model is the one that PT operators currently use, which estimates the ridership of this week based on the same trip last week. However, we find that in our AVL dataset, the missing recordings of the trips are notable. Thus, we recommend that the PT operators can come up with a better way to deal with this problem and potentially elevate the recording system or trip numbering system. For the linear model, we build the linear regression model with one more variable into account every time to test the significance of such a variable in a linear relationship manner. From this ever-increasing inclusion, we conclude that the model with the number of requests, the average number of ridership, the ridership of last week is the best linear regression model. But, the linear relationship can not capture the relationship between target and inputs well. The last point is the simple model with the weekly trend. This model is strongly biased and fails due to most of the overestimations are related to the same trip that was too crowded last week on several specific sections so that the ratio is too high. The possible improvements for this kind of model are to add smoothers, such as the ratio of the last trip or the ratio of the last section or the average ratio of the previous sections so that the model might outperform the baseline model but maintain the feature of efficiency and computationally friendly attribute.

Next, we use the outperformed model -RF regression- to explore the residuals. We identify a clear pattern of heteroscedasticity of all diagrams, and we can spot some outliers in the graphs that in some cases. The possible solution is to include more significant variables or turn to other models to test the performance. Line 50 has the problem of the y-axis unbalanced slightly. Therefore, ridership distribution like line 50 needs a different approach to gain better performance of the model.

All histograms of the errors present a bell-shape distribution, which means a normally distributed format while line 50 shows the pattern with sightly right-skewed, representing the tendency of overestimation. By using Q-Q plots, we discover that the distribution of residuals has a fat tail, namely leptokurtosis. There is more data located at the extremes of the distribution and fewer data in the center of the distribution when it approaches the two ends. It is due to the original shape of data distribution, and also our model can be elevated when dealing with the high-value domain.

Moreover, the models reach a balanced estimation in general. However, line 50 has the most difference in the biased prediction, and the margin can be as high as 30% due to its large oversampling strategy. Most of the cases, the RF regression model for line 35 can anticipate the passengers on board right with the lowest 95th percentile. Even though we did not undersample, but only oversample the minority classes by 5% for Qliner 300and Q-link 1. They have more tendencies to underestimate the actual values, and this tendency is much more profound when we see the average of the top 5 percentile residuals.

These prediction residuals are not entirely similar distributed across the line, in terms of periods. For the majority of the cases, it is the evening peak that has the highest variance because it is the second-highest commuting time. However, people have a non-uniform off-duty time, and therefore the variance is easy to fluctuate. In contrast, Qliner 300 is the one that has a large variance of residuals during the morning peak because the commuting passenger volume can be very high due to its line characteristics and thus result in a large variance and a difficulty in prediction. Concerning day type, the variance of the weekday is higher than the weekends on every case study line.

Furthermore, we apply both MDI and permutation feature importance to calculate the importance of features and compare the results from both methods to gain a complete understanding of the importance of features. Regardless of the methods, the most important feature is always the average number of ridership. The number of trip planner requests can also play a vital role in the ridership prediction with the importance of approximately 20%. Our feature importance analysis also backs up the argument that the impurity-based feature importance can inflate the importance of numerical features.

Following, we put PDP into use the check the functional relationship between request-related variables with ridership. When the number of requests is low, there is a strong positive correlation between ridership and request. However, when it is the high-value domain, the effect is marginal because of the fewer recordings so that the model can not learn a meaningful prediction. It holds the same for the average number of requests in the high-value domain. And the influence goes down and then up when the value is less than 3 or 4. The positive correlation between the variance of the requests and ridership is visible from the small negative value to the minor positive value.

Additionally, we investigate the performance of the RF model with trip planner data that is further ahead in time. We examine the scenario with all requests, requests that are 10, 15, 30 minutes ahead with the same configuration of the model, and the same sampling design. The performance does not deteriorate to a great extent. Even though the number of data and the correlation between requests and ridership drops, the model functions essentially the same. For the majority of the cases, the model does not perform worse with the increase of the timing advance. Two of the four cases show us that model with trip planner requests that are 10-minute ahead outperforms than others, indicating that the information stores nearest are more valuable.

Lastly, we desire to unveil how the trip planner information could function during different times, and during what period, the trip planner data is of the most usefulness. The reason is that people behave differently during different times by using such a trip planner from the data analysis. We find out that three cases out of four have the best performance with requests from 10:00 to 16:00, compared to all requests included. Users send very few requests at night (from 22:00 to 4:00), which is around 5% of all requests. But one of the cases gives the best performance by utilizing requests from the night. In notable contrast, the other one gives very unsatisfied results. Therefore, we conclude that the case in which we have a different sample design can learn a meaningful result when the sample is large enough. In other cases, it is during the evening (from 16:00 to 22:00) when *Random Forest Regression* model captures less.

8 Conclusion, Discussion and Recommendation

The research gaps that we put forward at the very beginning of our study are: 1) incorporating a novel data source in ridership prediction – trip planner data; 2) exploring the dimensions of this data and smart card data (ridership); and 3) applying machine learning methods to predict ridership with trip planner data along the matching dimensions. Thereon, we formulate the research objective to derive the relationship between trip planner data and the ridership of bus trips by applying machine learning algorithms to see if we can conduct the short-term prediction with an acceptable precision at stop-level.

We carry out our research with the scope of two provinces in the Netherlands, Groningen, and Drenthe. The research is also supported and collaborated with 9292 and OV Bureau Groningen and Drenthe, who provided valuable data to the study. We used the data from October to analyze the data, develop the model, and test its performance. The data consists of three parts: trip planner data, smart card data, and AVL data. We examine the inputs by the covariance/correlation, and we explore the dimensions in the data both spatially and temporally. We select the PT current model, simple model with the weekly trend, linear regression, GBM regression, KNN regression, and RF regression to fulfill the prediction task based on a more interpretation aim.

The benchmark of this study refers to Van Roosmalen (2019) who first investigated the usefulness of the trip planner in ridership prediction. Models proposed by Van Roosmalen (2019) anticipates the passenger on board at origin and destination separately and has a relatively higher RMSE, compared to this study. Moreover, the research has a scope of whole lines in the same region but mainly looking into the workday and 8 AM on workdays. However, in our study, we directly predict the passengers on board by sections, namely from stop to stop. It gives a better result in comparison with the extraction method. Furthermore, although we focus on four cases, they tend to be representative, and it gives us a complete temporal dimension to establish the comparison analysis. Following, we also add various analysis, including prediction vs. actuality analysis, residual analysis, feature importance, scenario analysis, and prediction time horizon analysis.

We steer our efforts on how to use visualizations to deliver more information at different data value domain and analyze this difference via various methods. The most innovative analysis of our research is by revealing the utility of trip planner data with further ahead in time. Besides, we offer an extensive analysis of the functionality of the models and compare two measures of the feature importance. Additionally, before diving into the construction of the models, we also present diverse analyses for the two types of data, i.e. smart card data and trip planner data. Plus, we put forward a methodology with visual schematics to convey the workflow in every step.

The structure of this chapter is as follows: we first conclude this study by answering the research quests that we proposed at the very beginning of the research; then, we discuss our research results, the limitations and suggest the future works that this study implies; finally, we list some recommendations for the further research in this field.

8.1. Conclusion

Before answering the main research question, we first answer the research sub-questions that can guide the answer to the main one. We formulate the sub-questions to lead the direction of the study from theory or the state-of-the-art research works, through the exploration of the environment (datasets), to the application of the models and finally navigating to the results and evaluation. The detailed and broader answers immerse in the contexts and summaries of relevant chapters. Here, we present only the essential and critical components of the answers:

1. What are the existing short-term ridership prediction models and influencing factors that internally from trip planner data and externally from other data affect the short-term ridership prediction according to literature?

The variables used in the different studies differ a lot. Depending on the time, location, and level of temporal and spatial aggregation, the impact of variables varies. The used spatial and temporal level is generally chosen so that the input variables fluctuate. The different variables can be categorized in the following groups: temporal, demand characteristics, weather, event, holidays, transit characteristics, other mode characteristics, spatial/built environment, socio-economic, and socio-psychological. The first six of these groups can be useful to predict short-term demand. Variables form the last four groups vary mostly only in the long-term. Depending on the location, time, and aggregation level, different variables are used. The internal factors lie in transit-related characteristics, including line frequency, routes, travel times, fares, and comfort. The other factors can be regarded as external factors.

2. What are the dimensions of analysis in the trip planner data (what data have been collected and stored) and in short-term ridership prediction models (what parameters and variables are there)?

We explore the dimensions of analysis by investigating the merged dataset of trip planner data, smart card data, and AVL data. The objective is to reveal the temporal and spatial dimensions that are existing in the dataset. We found out that people prefer using such an application for real-time most of the time. During the night (from 22:00 to 4:00), it is in contrast that people check the trips for tomorrow with at least four to six hours in advance. All case study lines tend to have a similar distribution of requests per 10-minute aggregation, indicating a peak off-peak pattern. The influences of temporal variables are significant, including the day of the week and holidays. The realized trips vary over the day. During the daytime, it tends to be more realized while it is the opposite during the night. Then, we explore the distribution of transaction data where the temporal influence is again significant. We also discover the impact of spatial variables from the smart card where we learn that the level of service is sufficient on average while it is not satisfied with busy trips. The city Groningen attracts quite many passengers, and the crowded sections are mainly in the city center of Groningen, the railway stations, or the P+R stops in the case study region, which shows the importance of incorporating the spatial variables in the model.

If we take the smart card into account as the ground truth, the realized trips vary over the day. During the daytime, it tends to be more realized while during the night it is the opposite. Then, we explore the distribution of transaction data where the temporal influence is again testified. We also discover the impact of spatial variables from the smart card where we learn that the level of service is sufficient on average while it is not satisfied with busy trips.

3. Along the above-mentioned dimensions, how does the trip planner data correlate with observed ridership from AFC data in the short-term?

The dependent variable (target) in this study is ridership, derived from the smart card. We consider the following dimensions (independent variables) to construct the relationship: the historical average of ridership, ridership of last week, the autumn holiday, the trip planner requests, the historical average of requests, the variance of requests, the day type, the period of the day, the line, the section, and the request-related variables with timing advance.

We unveil the relationship and correlation between trip planner data and smart card data by first developing the joint distribution of them and variance-covariance/correlation with all variables considered at the day-level and the stop-level. We see that there is a considerable positive correlation between requests and transactions, which means that trip planner data can correlate to smart card data. Moreover, we notice that this relationship weakens when we dive into more details, namely from day-level to stop-level. The correlation between trip planner data and smart card data can be around 0.63 at stop-level. It implies that it is comparably easier to leverage trip planner data to predict ridership at the day-level but harder at stop-level.

If we carry out the prediction further ahead in time, the correlation drops continuously from all requests data included to data with timing advance. It indicates that based on the number of requests to predict the short-term ridership becomes more difficult when we are interested in earlier prediction lead time. The historical average of a trip takes an important role when we want to establish a relationship, which means that the collaboration between the OV-bureau and 9292 is highly appreciated to reach a satisfying result. Additionally, the covariance between ridership and all other variables remains the same during different periods.

Except for the request related variables that are not examined before, all other variables bring into correspondence with the literature. The consensus of the influence of time and date remains consistent, such as the period of the day and the day type. The influence of special days is considerable, for instance, the autumn holiday. Spatial attributes are also contributing, especially the section around a railway station or a working place.

4. How can such correlations be leveraged for short-term bus ridership prediction?

We select the variables that could strongly influence the ridership, including ridership-related, request-related, spatial- and temporal-related variables as unveiled by data analysis and correlation examination. We scale the independent variables as different units, different ranges because there are distance sensible models in our study.

Moreover, we apply the log transformation, binning strategy, alone with sampling/resampling design to tackle the data imbalance problem, and to better capture different domains equally. The target, ridership, is trained after taking the natural logarithm because the original distribution of the target is highly right-skewed, and we want to comply with the regression models that work better with more symmetrical, bell-shaped distributions. Furthermore, this transformation facilitates the binning of continuous data into discrete one as there is less data on the right tail of the distribution, and we can only over- and discretely undersample data and then convert it back to continuous. The binning strategy with Doane's formula helps us to resample the data by SMOTE (synthetic minority oversampling technique) that enhances the influence and representations of the minority but interested classes. We also practice a sensitivity analysis to determine the optimal sample designs.

There is one type of data distribution that we need to consider as an exception and treat it differently when resampling, which is the distribution that is similar to line 50. This kind of distribution has sparse bins on the right tail, and thus the optimal strategy is random without clear rules. In contrast, other cases show that no undersampling, but oversample the minorities by 50% is the optimal sampling design.

We leverage the regression problem to the short-term bus ridership prediction with trip planner data, and supervised learning is chosen to accomplish the task. The well-knitted variables as input dataset is trained in models including, *LR*, *KNN*, *GBRT* and *RFR* based on a more interpretable configuration and a more interpretable results. Along with baseline models (PT current model and PT current with seasonal trend), we establish a comparison and find the outperformed one. Besides, we apply nested cross-validation to fine-tune the parameters in the inner-loop by *k-fold cross-validation* and compute the robustness and accuracy of the models via outer-loop by *random permutation cross-validation*.

5. What is the performance and benefit of using such a prediction model?

In order to compare the multiple models, we set up an overall set of performance metrics that consist of MAE, $RMSE$, R^2 and R^2 from *random permutation cross-validation*. Since the main objective of the study is to discover the usefulness and utility of trip planner data, only get to know the performance of the model is not enough. Thus, we introduce feature importance to the study, and we will select the best model out of the omnibus to explore the feature importance.

We execute the models that we selected with the desired inputs. For the majority of the cases, RF Regression outperforms than other models with a landslide. It is line 50 that we have a different sampling design that gives us a different result. In such a case, we have KNN regression as the best model on the MAE, $RMSE$, R^2, but we have a better R^2 from the cross-validation. The result from cross-validation shows that it is less sensitive to training data. Due to cross-validation is an iterative process, it is more convincing. Therefore, we consider RF is the best model.

Every model can capture the relationship, and the difference lies to what extend it can function. All models have higher average error and bias when the actual value becomes large, imply the existence of heteroscedastic. Moreover, we can assert that regardless of the line, machine learning models generally can not only better capture the quiet trips but also outperform than the baseline models when the trips become busy. When the actual value is low, all models can function well while GBM tends to overestimate, KNN tends to underestimate, and RF as the best model is relatively neural.

Next, we analyze the residuals by the outperformed model, RF regression. Other than clear heteroscedasticity, we can also spot some outliers. These imply the potential of improvements, including the inclusion of significant variables, better models. Line 50 has the problem of the y-axis unbalanced slightly, and therefore ridership distribution like line 50 needs a different approach to gain better performance of the model. The distribution of residuals also shows a leptokurtosis. More residuals locate at the extremes of the distribution, and fewer data in the center of the distribution when it approaches the two ends. It is due to the original shape of data distribution, and also our model can be elevated when dealing with the high-value domain.

In general, the RF models reach a balanced estimation, namely the amount of over- and underestimation tends to be equal. Qliner 300 is the only line that has more underestimation than overestimation. The difference is approximately 7%. Q-link 1 is the most balanced one with a similar percentage of over- and underestimation of the ridership. Line 50 has the most difference in the biased prediction, and the margin can be as high as 30% due to its different sample design. We oversample the minority by 250% and thus result in an overestimation. Line 35 has more overstatements, which is around 15% more than understatements.

Line 35 has the lowest 95th percentile absolute error, no matter overestimation or underestimation. It means that most of the time, the RF regression model for line 35 can anticipate the passengers on board right with the lowest error among all cases. Plus, it has the smallest margin of increase if we compare the 95th percentile with the average of the top 5th percentile. Line 50 has a higher 95th percentile error of overestimation than underestimation. However, the average top 5th percentile of underestimation error is severer than overestimation. It indicates that the majority of the time, the model tends to have an overestimation, and for extreme cases, it is the opposite. It is in line with the resampling strategy that we applied where we oversample the minority classes at the large-value range. Even though we did not undersample, but only oversample the minority classes by 5% for Qliner 300 and Q-link 1. They have more tendencies to underestimate the actual values, and this tendency is much more profound when we see the average of the top 5 percentile residuals.

Temporal-wise, the evening peak generally is the most difficult period to predict because it is the second-highest commuting time. People have a non-uniform off-duty time, and therefore the variance is easy to fluctuate. While Qliner 300 is the one that has a larger variance of residuals during the morning peak because the commuting passenger volume can be very high due to its line characteristics. Moreover, the variance of residuals is much more profound in the weekdays than the weekend.

Aside from receiving a better performance when using RF regression to predict the ridership, another benefit is that the model functions roughly the same when we incorporate the trip planner data with timing advance. Even though the number of data and the correlation between requests and ridership drops, along with the increase of the timing advance, the model performs almost the same and the feature importance does not change dramatically. Two of the four cases show us that model with trip planner requests that are 30-minute ahead outperforms than others, indicating that the information stores further ahead are more valuable than the nearest. It means it is possible to leverage fewer data to reach a satisfying performance.

To what extent can the trip planner data contribute to the short-term bus ridership prediction and what are the important influencing factors in trip planner data and other data in such a prediction model?

In order to understand to what extent the trip planner data can help in short-term bus ridership prediction. We execute six different types of models, including two baseline models, *LR*, *KNN*, *GBRT* and *RFR*. We measure the performance of the models by the metrics that we set up, including MAE,

$RMSE$, R^2 and R^2 from *random permutation cross-validation*. For the majority of the cases, Random Forest Regression outperforms than other models with a landslide. Especially, the RF regression model can be roughly two-time better than the baseline model.

For Qliner 300, both RF regression and GBM regression can capture the unseen data the best with a cross-validation score of 0.73. However, RF outperforms than GBM if we compare the other three metrics. From the performance of Q-link 1, RF regression beats other models from every criterion in which the R^2 from cross-validation can be as high as approximately 0.83. It is line 50 that we have a different sampling design that gives us a different result. In such a case, we have KNN regression as the best model on the MAE, $RMSE$, R^2 but we have a better R^2 of 0.77 from the cross-validation. Still, we consider RF is better as the result from cross-validation shows that it is less sensitive to training data and due to cross-validation is an iterative process, the result is more convincing. Line 35 is the best our of the four cases with a high R^2 of 0.83 from the cross-validation.

We apply both MDI and permutation feature importance to examine the importance of variables (different types of data) and compare the results from both methods to gain a complete understanding of the importance of features in ridership prediction with trip planner data. Regardless of the methods, the most important feature is always the average number of ridership. The number of trip planner requests can also play a vital role in the ridership prediction with the importance of approximately 20%. This feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction. It does not mean accuracy. The relative importance scores can shed light on the features which are most relevant to the target.

Moreover, our findings of feature importance are in line with the literature but with some differences. In terms of temporal variables, morning peak and off-peak can influence the model but with a minor effect. It holds the same to some specifically spatial sections, such as the section around the working place and the place around the railway station. Unexpectedly, the temporal effect from the day of the week is not significant. It is different from the literature. The direction of the line, although it does not have a strong correlation with the ridership from the data analysis, it sometimes can be the top 10 influencing factors with a minor effect.

Furthermore, since we are interested in the relationship between trip planner data and ridership, we employ PDP to check the functional relationship between request-related variables with predicted ridership. When the number of requests is low, there is a strong positive correlation between ridership and request. However, when it is the high-value domain, the effect is marginal because of the fewer recordings so that the model can not learn a meaningful prediction. It holds the same for the average number of requests in the high-value domain. And the influence goes down and then up when the value is less than 3 or 4. The clear positive correlation between the variance of the requests and ridership is visible from the small negative value to the minor positive value.

Besides, the model performs generally the same when we incorporate the trip planner data with timing advance, namely including requests with further ahead in time, such as 10, 15, and 30 minutes. Even though the number of data and the correlation between the number of requests and ridership drops along with the increase of the timing advance, the model performs almost the same and the feature importance does not change dramatically.

From the data analysis, we explore how people behave differently during different times by using such a trip planner. We conclude that people prefer checking travel information in 0 to 10 minutes of a short time range during the daytime. In contrast, people would like to plan their trip at least 8 hours before night. We find out that three cases out of four have the best performance with requests from 10:00 to 16:00, compared to all requests included. Users send very few requests at night (from 22:00 to 4:00), which is around 5% of all requests. But one of the cases gives the best performance by utilizing requests from the night. In notable contrast, the other one gives very unsatisfied results. Therefore, we conclude that the case in which we have a different sample design can learn a meaningful result when the sample is large enough. In other cases, it is during the evening (from 16:00 to 22:00) when the Random Forest Regression model captures less.

8.2. Discussion

In this section, we discuss the results that could be improved and the implication that the analyses convey.

First of all, noticeable challenges are during data preliminary. It would be easier to conduct a joint study if the OV bureau has the same naming system with 9292 and also other PT-related companies. It includes the uniform naming system, not only stops but also bus trips and routes. For 9292, it would be much more meaningful if we can track the user IP and all the alternatives of a piece of trip advice. Users can plan their journey into parts instead of using the "via" option. We can't trace the IP address, and thus it is hard to know if it is a single journey with multiple legs (trips) or different journeys. It could bring in "more" trip planner data than it should be, and thus incur the overestimation of the importance of trip planner data. Besides, due to the same privacy issue, we don't know if it is a bunch of people traveling or just a single person. If it is just a single person, the influence is understandable. However, if it is a bunch of people, referring to the same advice given by the trip planner, an underestimation of the importance of this type of data would be incurred. 9292 is the biggest trip planner company in the Netherlands, and it is indeed representative. But there are also other competitive companies which means that only proportional trip planner data correlate with the smart card data. Fourth, there is no identical bus number provided, neither on the smart card nor the trip planner and hence the mapping would introduce errors and bias into the study. Additionally, only one-month data are explored in this study, which could cause a lack of seasonal trend consideration.

Second, when dealing with the imbalanced data, we used an integrated method which consists of log transformation, binning, and sampling design. We choose sampling design over the cost sensitivity weighting because it is comparatively more efficient, effective, and doable. The reasons are that RF tends to be biased towards the majority class, and the cost sensitivity weighting places a heavier penalty on misclassifying the minority class. It assigns a weight to each class, with the minority class given higher weight (namely, higher misclassification cost). By assigning such weights, it can introduce noises (mislabeled classes) and thus make the model vulnerable to those noises comparatively. Besides, the machine learning package we used does not support a penalized random forest regression. However, there is no clear winner between the two methods. Line 50 among four cases appears to be a different case and results in an overestimation of the results, and its overall performance is not entirely similar to other cases. Given the limited time and cost, sensitivity weighting needs to use the entire training set to compute and then tune the model and the technical limitations. Thus, further study can use this method to address the imbalanced data issue to treat each data domain equally.

Third, future works can choose better optimal resampling strategies (over- and undersampling). We decide not to undersample the majorities, but oversampling the minorities by 50% for Qliner 300, Q-link 1, and line 35. However, these best scenarios always suit in the top left corner, which implies that better options can be explored, except for line 50 above-mentioned. By potential improvement, we mean that We test them through 5-fold cross-validation on the dataset of every case study line, for a total of 24 resampling tests. There could be more combinations since we set a fixed number of over- and undersampling. The better strategy can be even further "top left" with even less oversampling and may give us a result that we should not resample. It is because we compare the R^2 from the test set. But, this is contradictory to the objective that we want to treat every data domain equally, and we are more interested in the domain that is less representative with fewer data. Further research can pay more attention to decide the strategy, and this will strongly affect the results as we compare the models with and without resampling.

Fourth, several failed models can be substantially enhanced, including the baseline model and the simple model. The baseline model is the one that PT operators currently use, which estimates the ridership of this week based on the same trip last week. However, we find that in our AVL dataset, the missing recordings of the trips are notable, and thus we recommend that the PT operators can come up with a better way to deal with this problem and potentially elevate the recording system or trip numbering system. The simple model with weekly trend considered, this model is strongly biased and fails due to most of the overestimations are related to the same trip that was too crowded last week on several specific sections so that the ratio is too high. The possible improvements for this kind of model are to add smoothers, such as the ratio of the last trip or the ratio of the last section or the average ratio of the previous sections so that the model might outperform the baseline model but maintain the feature of efficiency and computationally friendly attribute.

Fifth, during residual analysis, we find out that the existence of heteroscedasticity. Heteroscedasticity (aka. heteroskedasticity) refers to the circumstance in which the variability of a variable is unequal across the range of values of a second variable that predicts it. In our case, it is that the error becomes higher with the increase of the actuality. Heteroscedasticity is not a big problem, but it implies the model can be advanced. Most often, there are two ways: the first one we have already applied, transforming the variable. Yet, there are various ways of doing so. It could be a different way that gives a better result. The second one is to add more variables so that the model can capture the relationship between dependent and independent variables better. Therefore, future works can blend more significant variables to improve the current model. Also, there are some outliers of residuals. Although tree-based models are less sensitive to those outliers, they still affect the performance to some extent. But keeping the outliers can create a more real circumstance and test how reliable the model is. It depends on the aim of the researcher to handle such an issue.

Sixth, we only test four lines with different characteristics as the case study. Further research can take more cases into account to reveal more valuable information from different line features.

Lastly, from a practical point of view, by using KNN regression to predict ridership is worthy of consideration. KNN regression is less sensitive to resampling strategy, and it reaches a balance between computation time and error minimization. Although RF regression gives the best performance, the tuning process can be unacceptable long, especially for line 50, in which we apply a different sampling design. If we want to obtain a better performance, we can also attempt to apply deep learning where artificial neural networks can take place. But deep learning is often regarded as a “black box” (unsupervised learning) as we can not derive desirable knowledge of feature importance out of it, which is opposite what we want for this study. However, since we have a relatively large dataset, training deep learning can give better results.

8.3. Recommendation

We consider that the future works of this study are ample as it is novel to incorporate trip planner data in ridership prediction. From this study, since the result is satisfactory, it implies this consideration and participation can earn a place in the ridership prediction. Regardless of the case that we selected, the RF regression model outperforms the baseline model (PT current model) by a landslide. We recommend a high-level collaboration among 9292 (trip planner company), public transport operators, and the authorities. So that using smart card data as a sound basis and adding trip planner data to predict the ridership can avoid the long collection time of the smart card data and also capture the passenger behavior. This collaboration also benefits the cleaning and merging of datasets, including stop names, route names, etc. Besides, it is beneficial that 9292 could store all provided choices such that we can dive into the analysis of passenger behavior and reduce the overestimation.

In a ridership prediction task, both the RF regression model and KNN regression model perform better than the baseline model. However, RF outperforms than KNN, in terms of the R^2 from cross-validation. However, from the other three metrics (MAE, $RMSE$, R^2), the difference is less profound for the majority of the cases. Therefore, we recommend 9292, PT operators and authorities can refer those two models to predict the short-term ridership with trip planner data based on our study.

Nonetheless, it is worthwhile mentioning the computation time in an ML study. The calculation is carried out on a laptop equipped with a 2.3 GHz Quad-Core Intel Core i7 and a memory of 16 GB 1600 MHz DDR3.

In general, the computation time of the model is acceptable in this study. However, it is to train an RF model that is very time-intensive. In ML, an algorithm aims to learn some set of rules that can describe the data (a set of inputs and outputs). We intuitively know those rules as a human, but do not precisely know the rules in a way so that we can describe them mathematically with enough concreteness. However, this learning passes challenges of computation complexity and sample complexity. In our study, we have already identified the most contributing variables, which means if the model is put into practice, we can reduce the runtime by removing irrelevant information. Still, we can expect a high running time if we want to utilize the RF model in practice with optimal hyper-parameter. But, our study also explains that once the configuration is determined, the results are robust, no matter utilizing the requests that are sent in advance or during different times. Therefore, it is the training before the usage will be time-consuming. There are two ways to solve the problems: first, we can utilize KNN in practice as it is relatively light, and the only hyper-parameter that we need to tune is the number of neighbors. From our study, it is comparably fast to determine this hyper-parameter. Second, we can carefully select the sample to leverage the minimum number of them required to achieve the goal. It can be also a future work that deals with the trade-off between computation time and results.

It would be significantly beneficial if 9292 could log the other alternatives in the database and also which route passengers choose such that there is more valuable information we can derive. How people trade-off between the fastest or the cheapest routes, how people trade off the routes traversed specific regions or districts, how people choose the exact model and are there a preference, etc. All those behavioral studies can take place since the choice of people is similar to a huge stated preference survey. It can significantly benefit the development of trip planners so that we can design personalized trips. It could also bring benefits to public transit operators as they can build the essential line or introduce the preferable modes.

The distinguish of user type will be advantageous to study the travel preference among different kinds of travelers. The first choice of users varies. For instance, the commuting pattern is different between a regular clerk and a night shift. Another example, a high-income employee would be different from a low-income worker. If we can distinguish the user type, the analysis can be performed in a user type manner such that we have more knowledge different professional has traveled.

If the IP address can be traced, not only we can avoid overestimation issues, but also we can have the preference across the different region/city/district, even community. It can enhance accessibility service and personalized travel. Especially during COVID-19, if there is an infection in a specific community, we can take precautions on the PT vehicles.

There is an issue in the trip planner database that needs to be resolved, which is the advice given for tomorrow. If a traveler wants to travel for the time being, but there is no certain supply, the algorithm now that 9292 would provide the earliest solution for tomorrow. However, it is unlikely people would wait for hours to make the trip, and therefore another method needs to be developed.

If we can manage the availability of GSM (Global System for Mobile Communications) or API (Application Programming Interface), we can fuse the data with trip planner data. Then, we can understand more about the spatial and temporal patterns of public transport usage versus overall travel demand.

A

Appendix A: Case Study

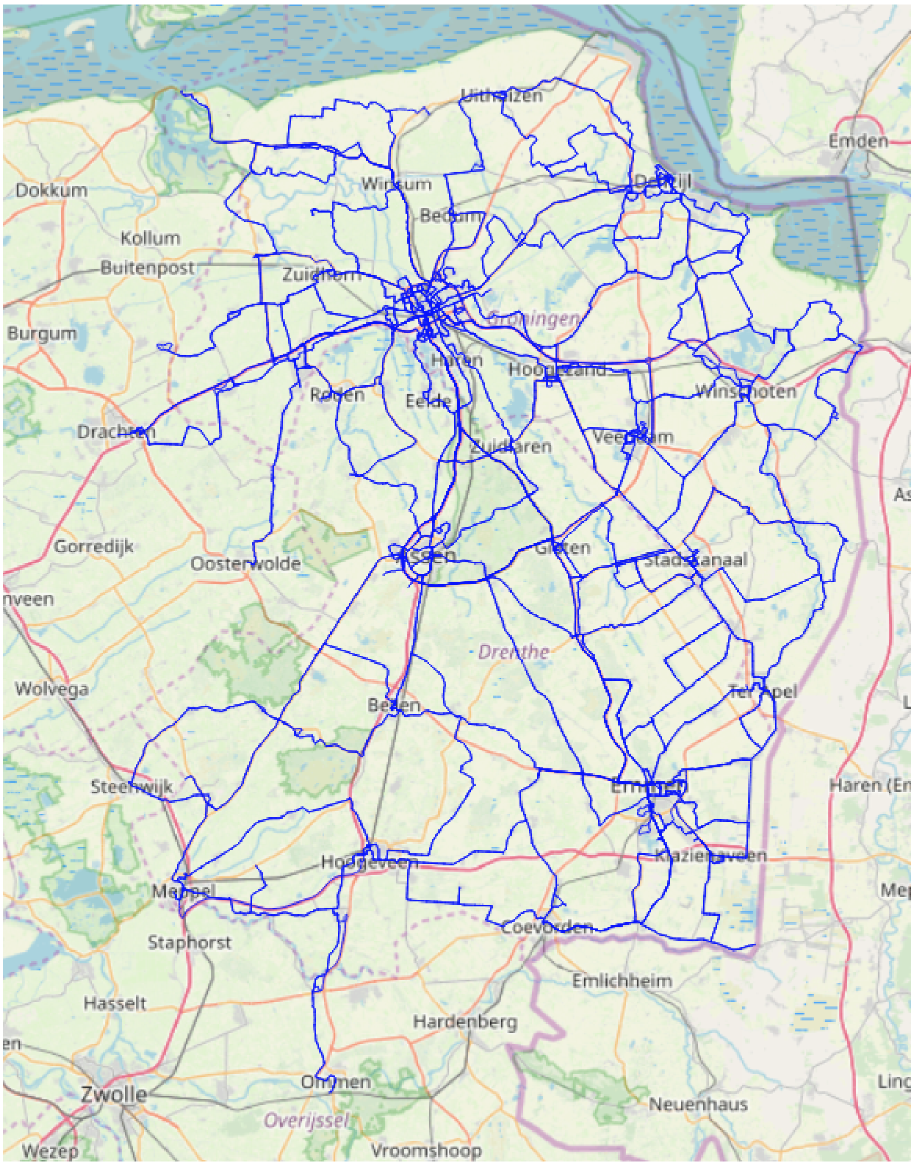


Figure A.1: Qbuzz network in Groningen and Drenthe

Table A.1: Information on case study lines

Line (outbound)	**Qliner 300**			**Q-link 1**		**Line 50**			**Line 35**			
Origin	Station, Groningen			Station, Groningen		Station,Groningen			Station,Groningen			
Destination	Station, Emmen			Station(PerronB), Zuidhorn		Station,Assen			Niehoofsterweg,Oldehove			
Number of Stops	8			19		43			38			
Average Stop Distance (m)	7335.6			401.3		664			592.2			
Bus Type	Setra S419UL	Mercedes Benz Integro L Euro 6	Van Hool TDX27 Astromega	VDL Citea SLFA 181 BRT		Mercedes Benz Citaro LE	Mercedes Benz Citaro C2G Hybrid	Ebusco 2.2	Same as Line 50			
Capacity (seats+standing)	73+55	69+42	81+0	43+82		34+87	47+118	40+50				
Weekday Start Time	5:30 AM	19:00 PM	23:00 PM	5:30 AM		6:00 AM	8:15 AM	18:15 PM	6:00 AM	8:45 AM	14:45 PM	20:45 PM
Weekday End Time	18:45 PM	22:30 PM	1:00 AM	12:15 AM		8:00 AM	17:45 PM	23:15 PM	8:00 AM	13:45 PM	18:45 PM	22:15 PM
Weekday Headway (min)	12	30	60	30		16	30	60	30	60	33	60
Duration (min)	110.6	107	107	51.3		54	53.9	49	96.5	92.9	90.9	70.6
Speed (km/h)	63.7	65.8	65.8	17.9		31.5	31.6	34.7	28	29.1	29.7	38.3
Number of Vehicles	13	4	2	3		7	6	2	4	2	4	2
Saturday Start Time	7:00 AM		23:00 PM	6:30 AM	7:45 AM	7:15 AM	11:15 AM	17:15 PM	7:45 AM			
Saturday Start Time	22:30 PM		1:00 AM	6:30 AM	24:15 AM	10:15 AM	16:45 PM	12:15 AM	17:45 PM			
Saturday Headway (min)	30		60	43	29	59	30	60	60			
Duration (min)	107		107	44.9	45.9	50.7	53	49.5	90			
Speed (km/h)	65.8		65.8	20.4	19.9	33.6	32.1	34.4	30			
Number of Vehicles	4		2	2	2	2	4	3	2			
Sunday Start Time	7:00 AM		23:00 PM	8:15 AM		7:15 AM			No operation			
Sunday Start Time	22:30 PM		1:00 AM	24:15 AM		23:15 PM						
Sunday Headway (min)	30		60	30		59						
Duration (min)	107		107	44.9		49.8						
Speed (km/h)	65.8		65.8	20.4		34.2						
Number of Vehicles	4		2	2		2						

B

Appendix B: Data

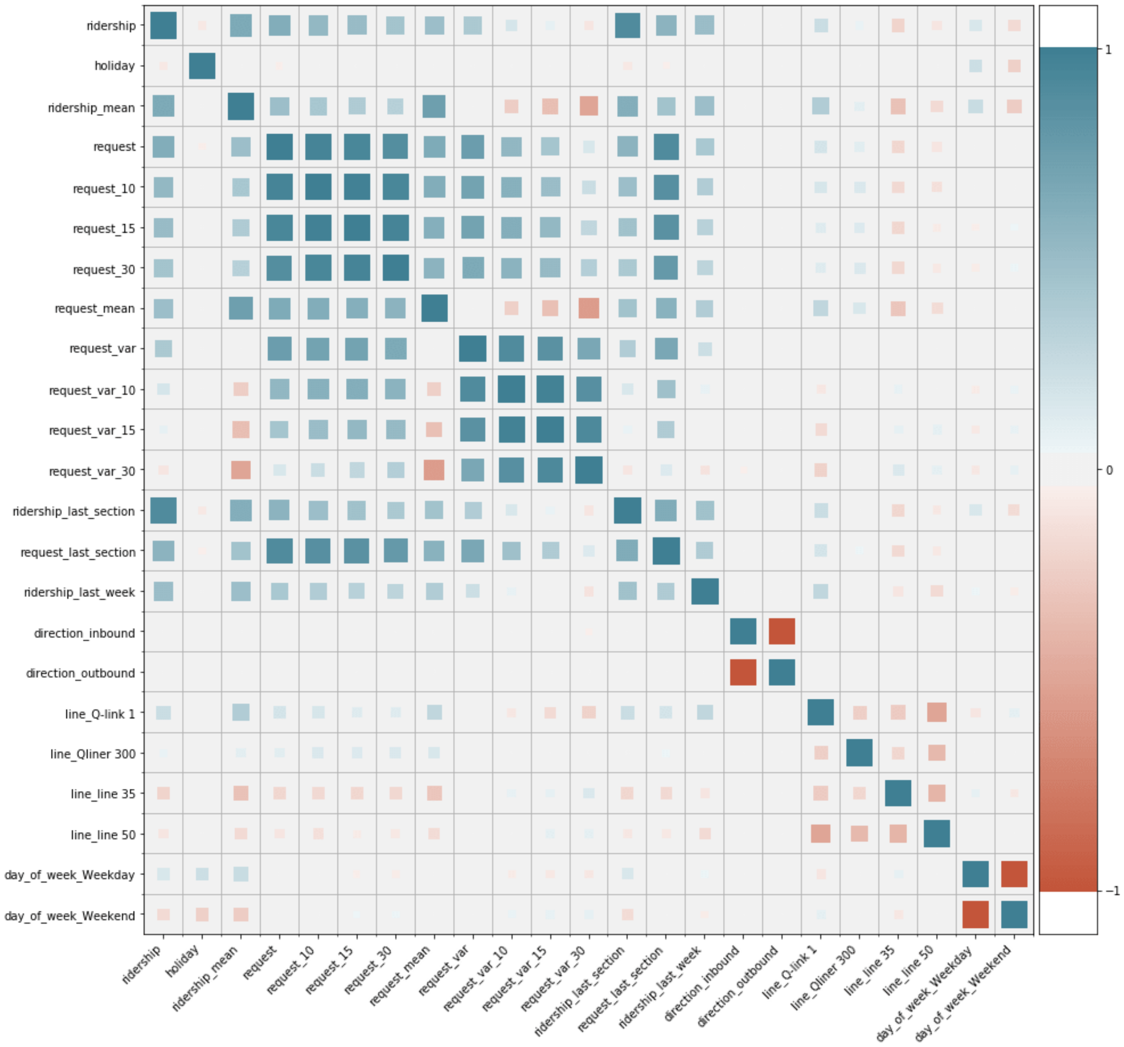


Figure B.1: Correlation matrix at stop-level during off-peak

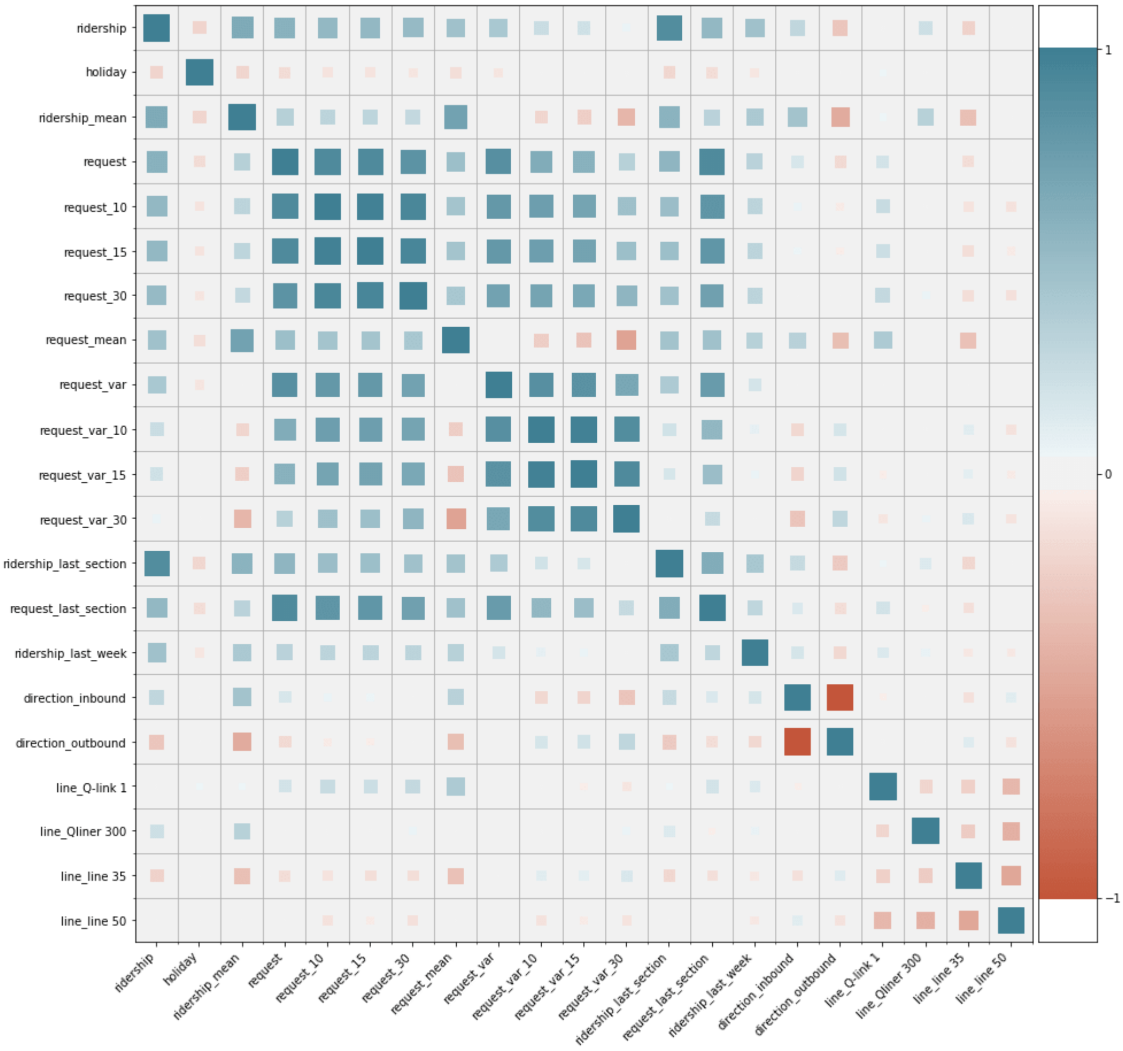


Figure B.2: Correlation matrix at stop-level during morning peak

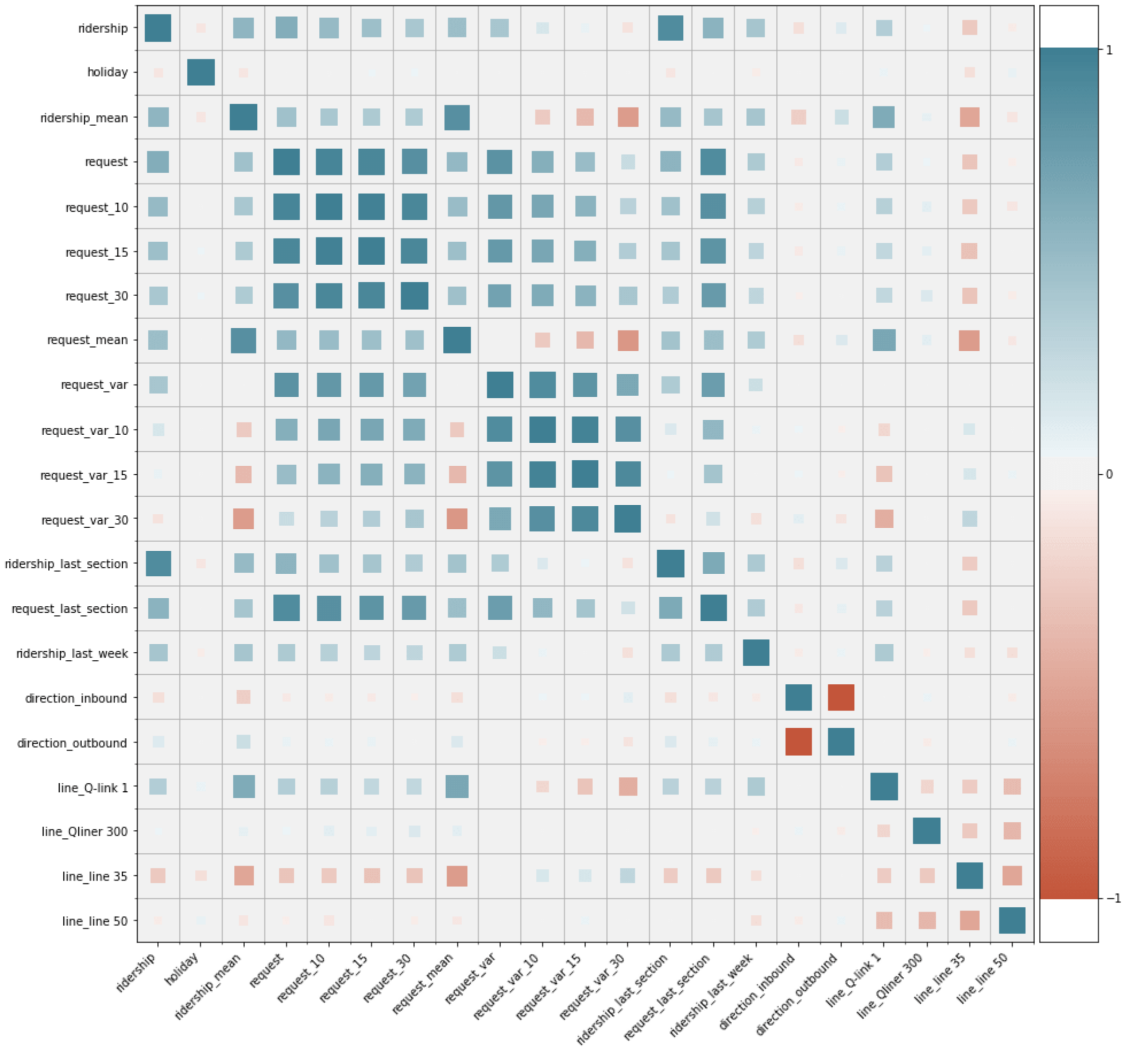


Figure B.3: Correlation matrix at stop-level during evening peak

C

Appendix C: Model Development

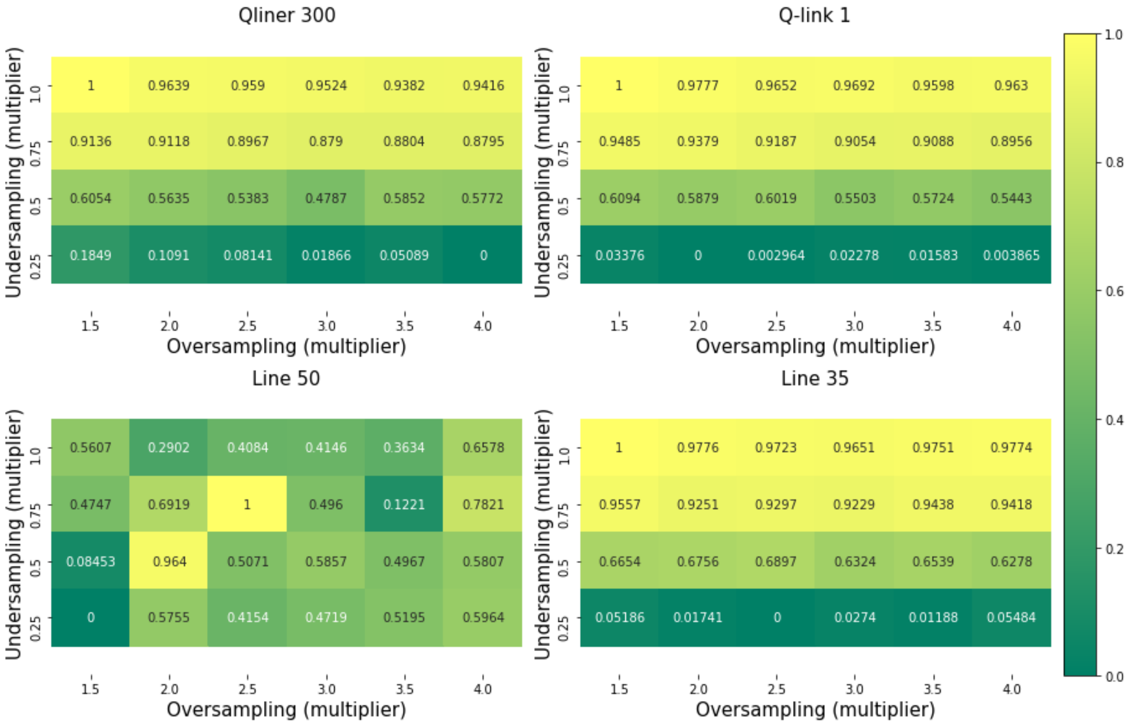


Figure C.1: Evaluation of different sampling designs per case study line 2nd

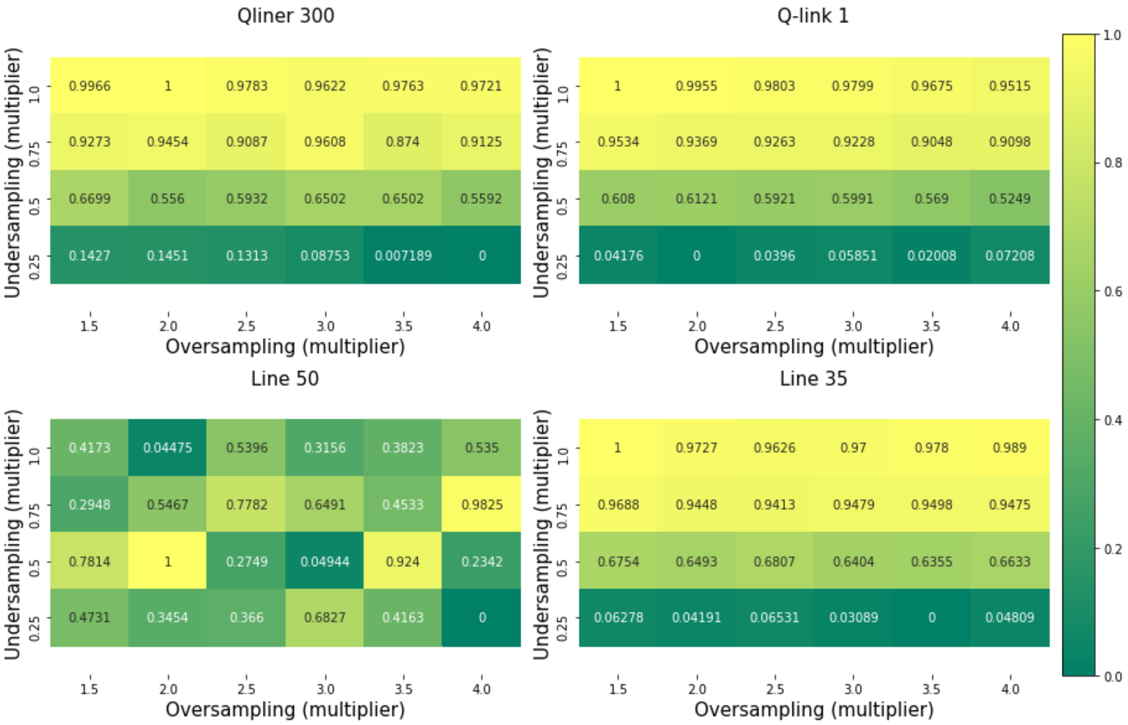


Figure C.2: Evaluation of different sampling designs per case study line 3rd

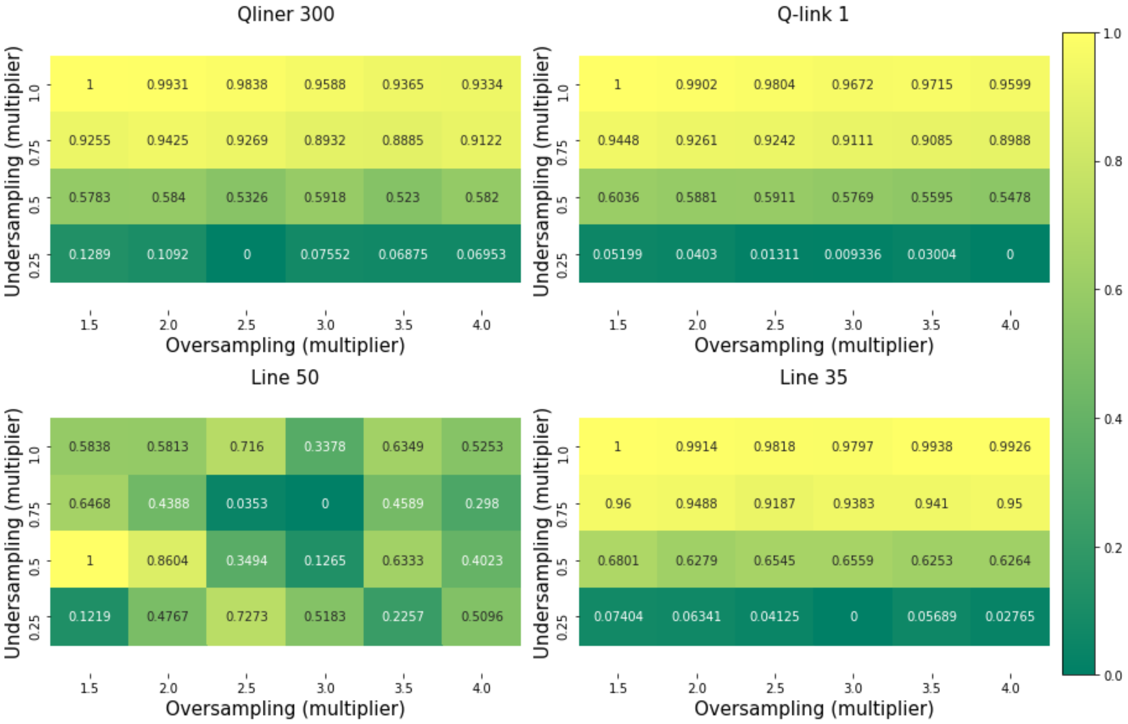
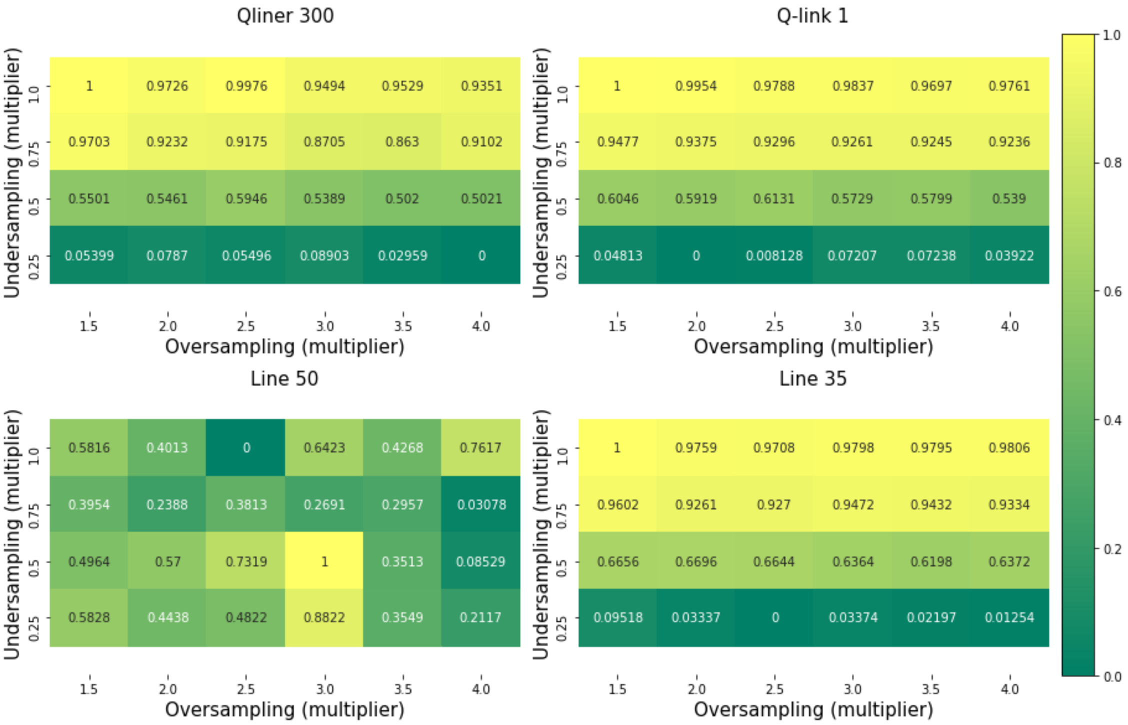


Figure C.3: Evaluation of different sampling designs per case study line 4th



Qliner 300

	1.5	2.0	2.5	3.0	3.5	4.0
1.0	1	0.9726	0.9976	0.9494	0.9529	0.9351
0.75	0.9703	0.9232	0.9175	0.8705	0.863	0.9102
0.5	0.5501	0.5461	0.5946	0.5389	0.502	0.5021
0.25	0.05399	0.0787	0.05496	0.08903	0.02959	0

Q-link 1

	1.5	2.0	2.5	3.0	3.5	4.0
1.0	1	0.9954	0.9788	0.9837	0.9697	0.9761
0.75	0.9477	0.9375	0.9296	0.9261	0.9245	0.9236
0.5	0.6046	0.5919	0.6131	0.5729	0.5799	0.539
0.25	0.04813	0	0.008128	0.07207	0.07238	0.03922

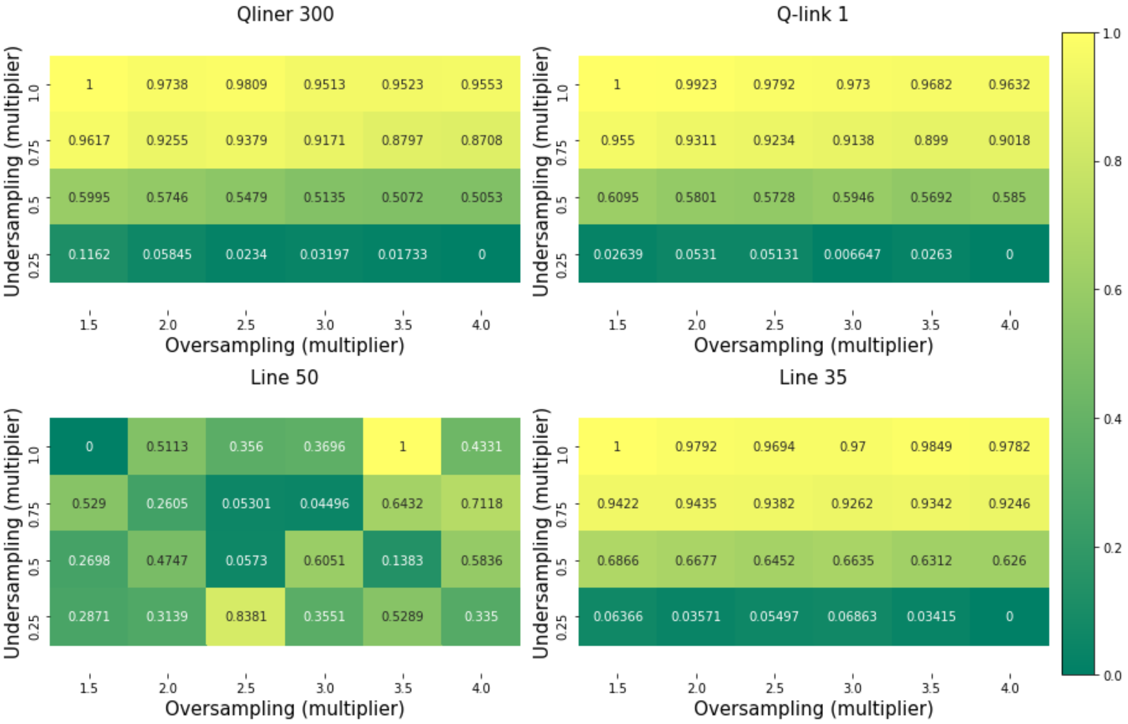
Line 50

	1.5	2.0	2.5	3.0	3.5	4.0
1.0	0.5816	0.4013	0	0.6423	0.4268	0.7617
0.75	0.3954	0.2388	0.3813	0.2691	0.2957	0.03078
0.5	0.4964	0.57	0.7319	1	0.3513	0.08529
0.25	0.5828	0.4438	0.4822	0.8822	0.3549	0.2117

Line 35

	1.5	2.0	2.5	3.0	3.5	4.0
1.0	1	0.9759	0.9708	0.9798	0.9795	0.9806
0.75	0.9602	0.9261	0.927	0.9472	0.9432	0.9334
0.5	0.6656	0.6696	0.6644	0.6364	0.6198	0.6372
0.25	0.09518	0.03337	0	0.03374	0.02197	0.01254

Figure C.4: Evaluation of different sampling designs per case study line 5th



Qliner 300

	1.5	2.0	2.5	3.0	3.5	4.0
1.0	1	0.9738	0.9809	0.9513	0.9523	0.9553
0.75	0.9617	0.9255	0.9379	0.9171	0.8797	0.8708
0.5	0.5995	0.5746	0.5479	0.5135	0.5072	0.5053
0.25	0.1162	0.05845	0.0234	0.03197	0.01733	0

Q-link 1

	1.5	2.0	2.5	3.0	3.5	4.0
1.0	1	0.9923	0.9792	0.973	0.9682	0.9632
0.75	0.955	0.9311	0.9234	0.9138	0.899	0.9018
0.5	0.6095	0.5801	0.5728	0.5946	0.5692	0.585
0.25	0.02639	0.0531	0.05131	0.006647	0.0263	0

Line 50

	1.5	2.0	2.5	3.0	3.5	4.0
1.0	0	0.5113	0.356	0.3696	1	0.4331
0.75	0.529	0.2605	0.05301	0.04496	0.6432	0.7118
0.5	0.2698	0.4747	0.0573	0.6051	0.1383	0.5836
0.25	0.2871	0.3139	0.8381	0.3551	0.5289	0.335

Line 35

	1.5	2.0	2.5	3.0	3.5	4.0
1.0	1	0.9792	0.9694	0.97	0.9849	0.9782
0.75	0.9422	0.9435	0.9382	0.9262	0.9342	0.9246
0.5	0.6866	0.6677	0.6452	0.6635	0.6312	0.626
0.25	0.06366	0.03571	0.05497	0.06863	0.03415	0

Figure C.5: Evaluation of different sampling designs per case study line 6th

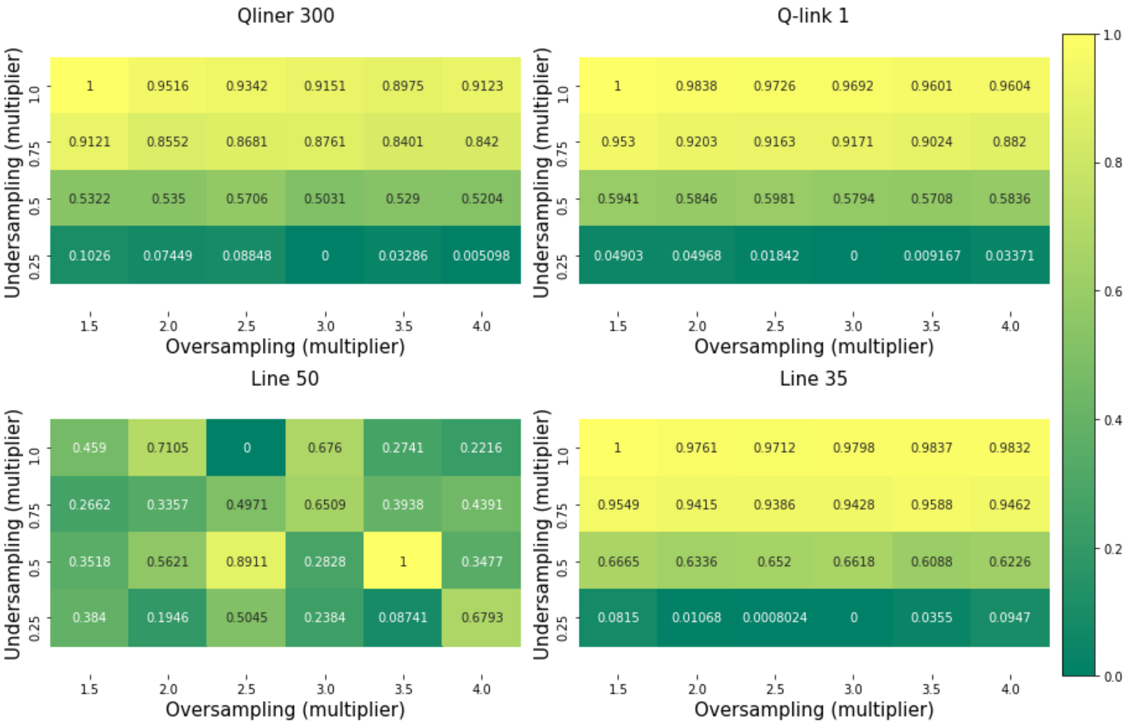


Figure C.6: Evaluation of different sampling designs per case study line 7th

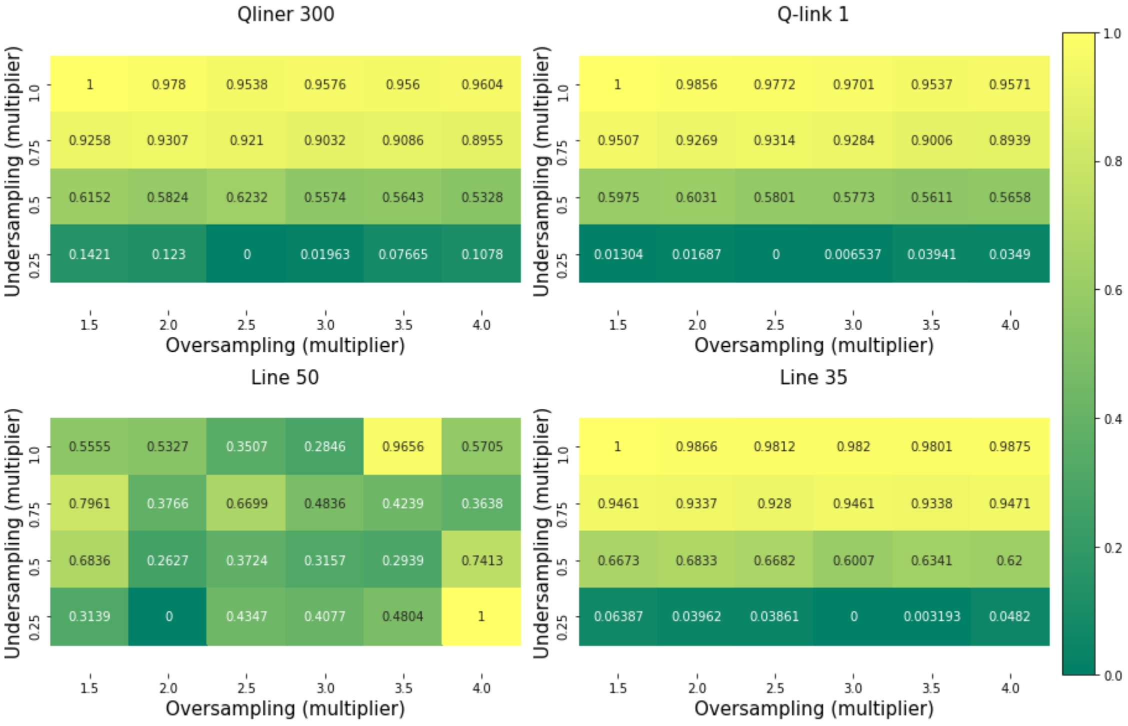


Figure C.7: Evaluation of different sampling designs per case study line 8th

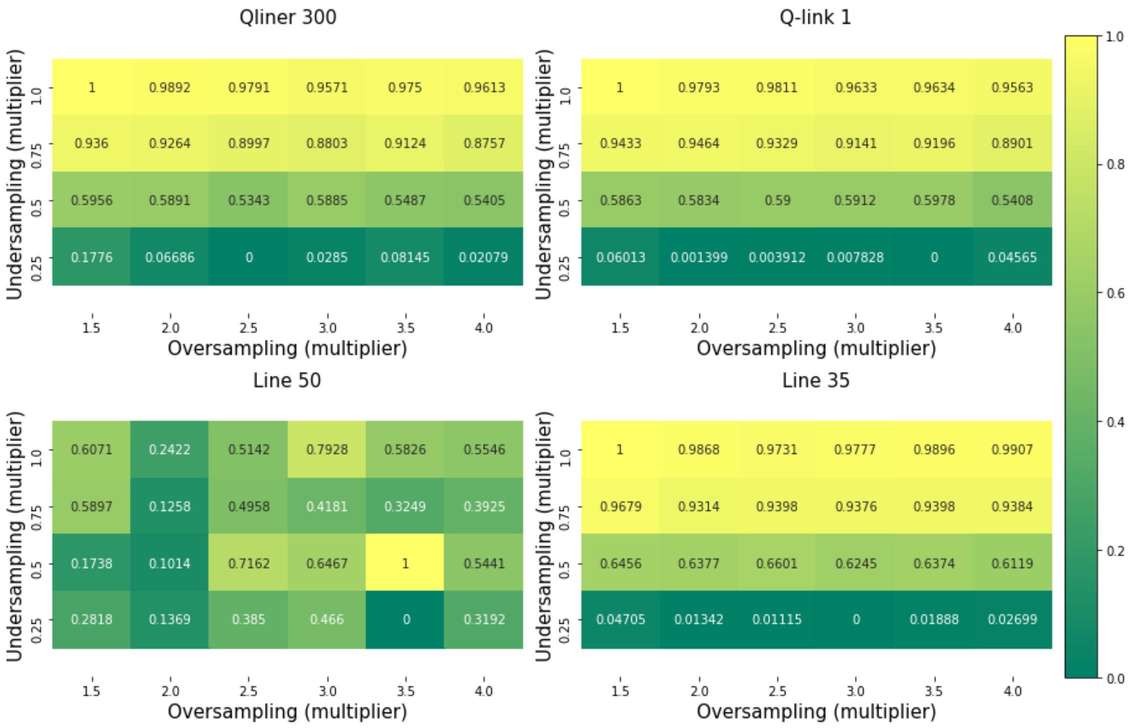


Figure C.8: Evaluation of different sampling designs per case study line 9th

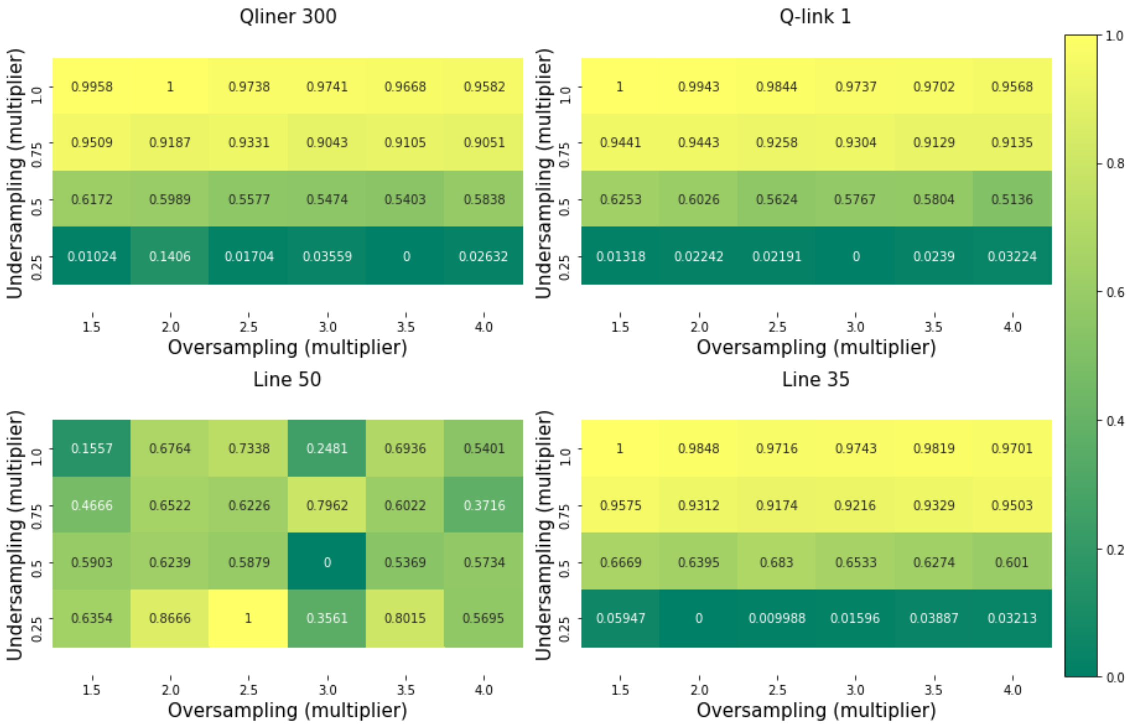


Figure C.9: Evaluation of different sampling designs per case study line 10th

D

Appendix D: Result

D.1. Actuality vs. Prediction Plots

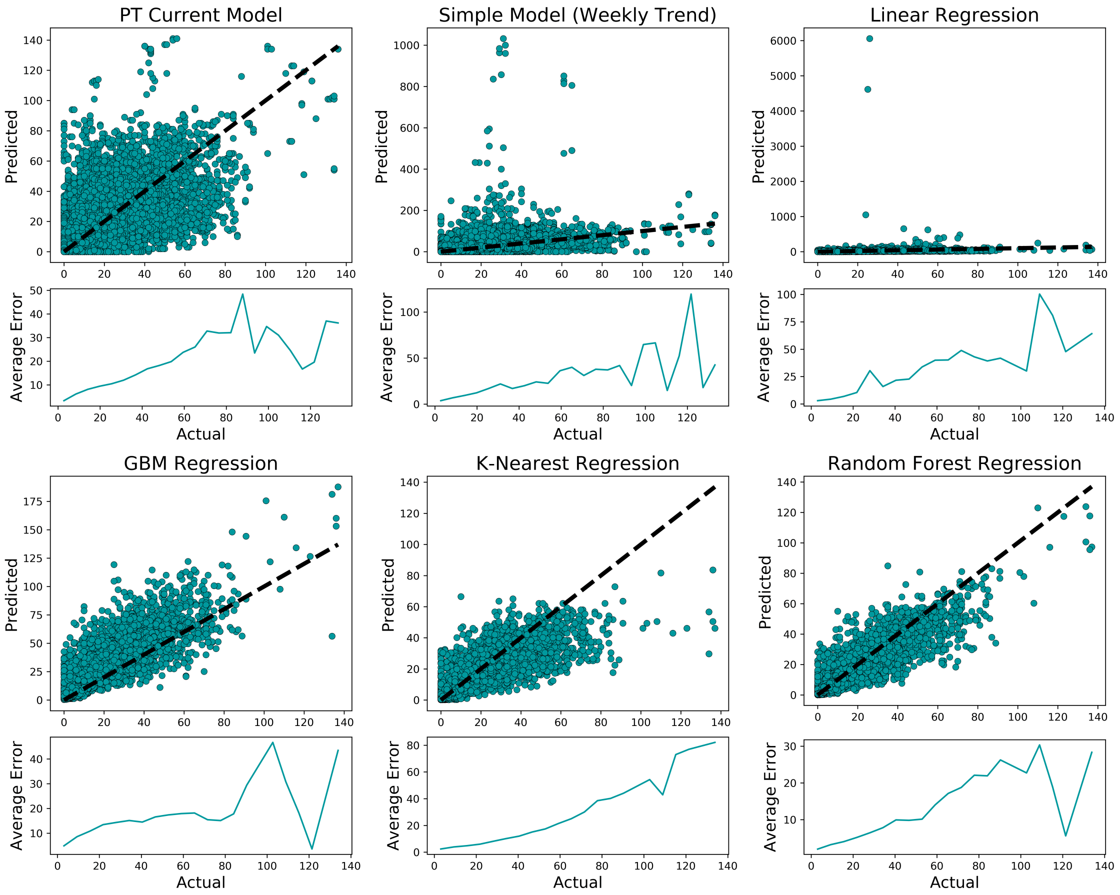


Figure D.1: Prediction vs. actuality plot of Q-link 1

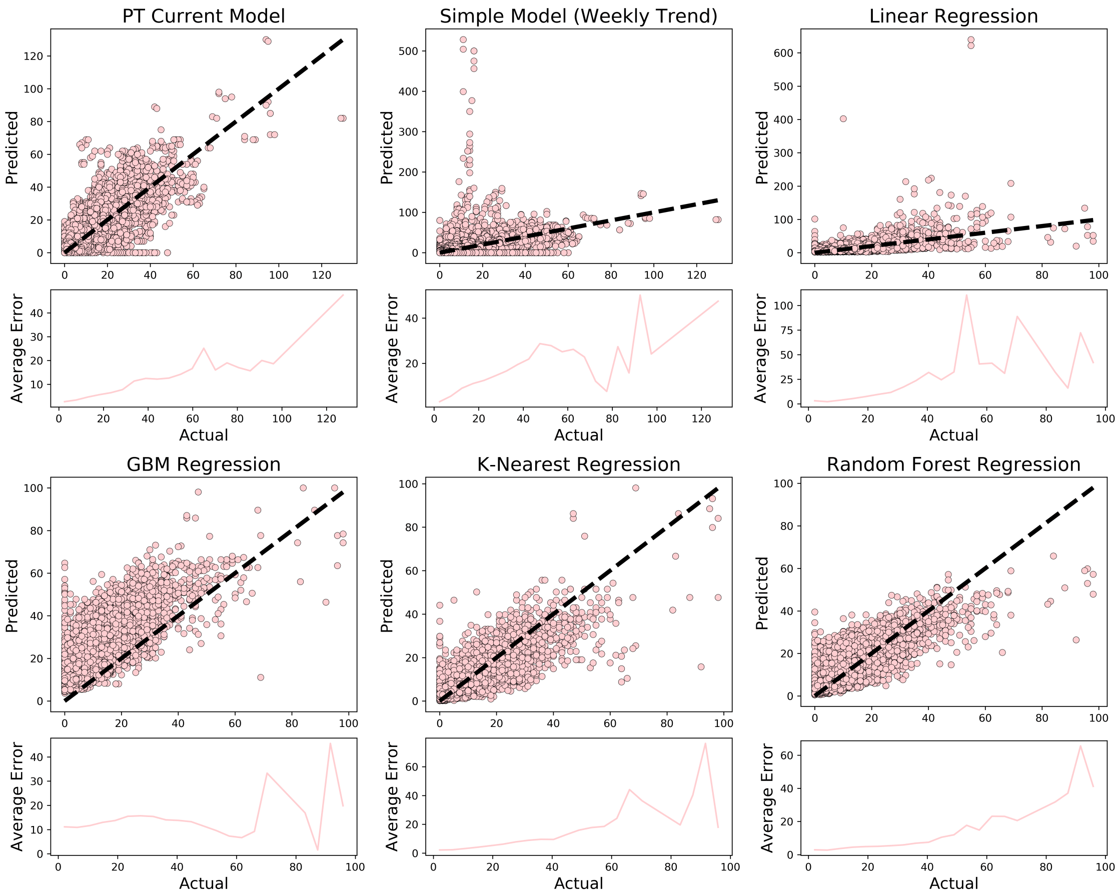


Figure D.2: Prediction vs. actuality plot of line 50

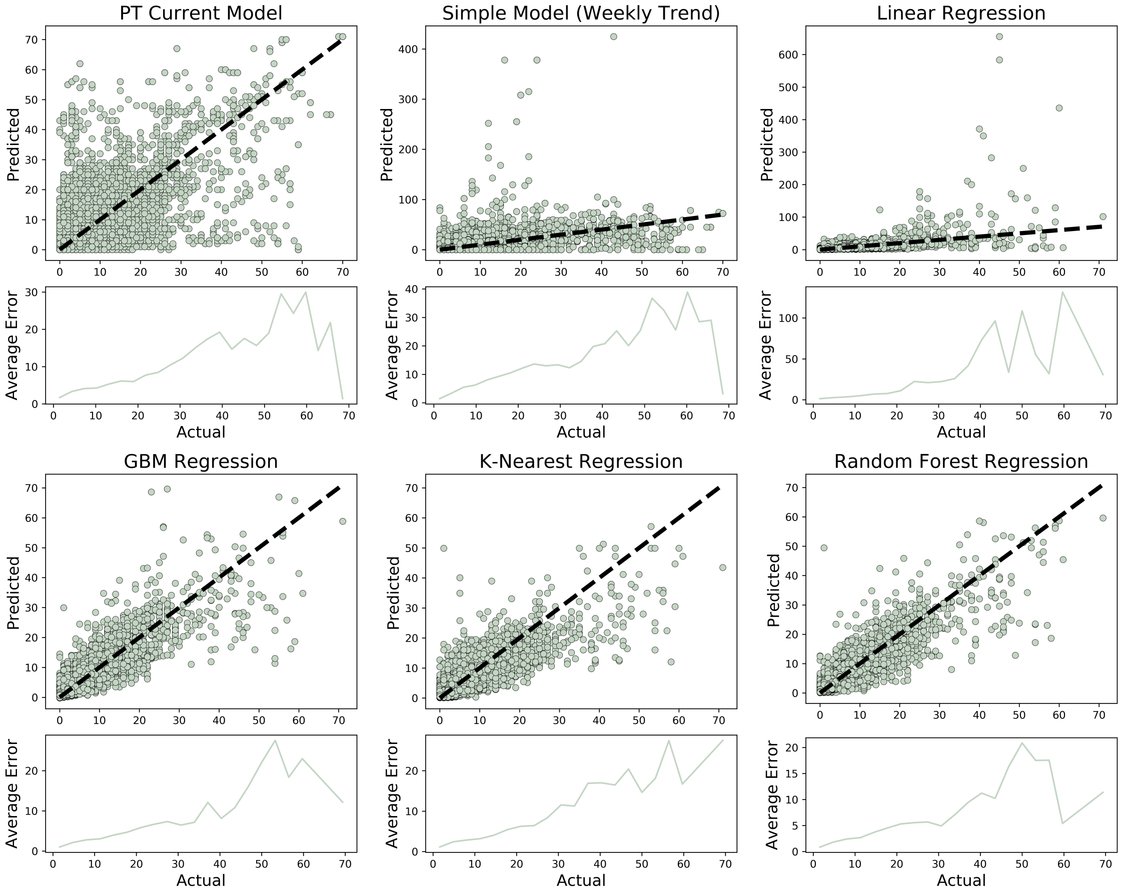


Figure D.3: Prediction vs. actuality plot of line 35

D.2. Actuality vs. Prediction Plots – Baseline Models

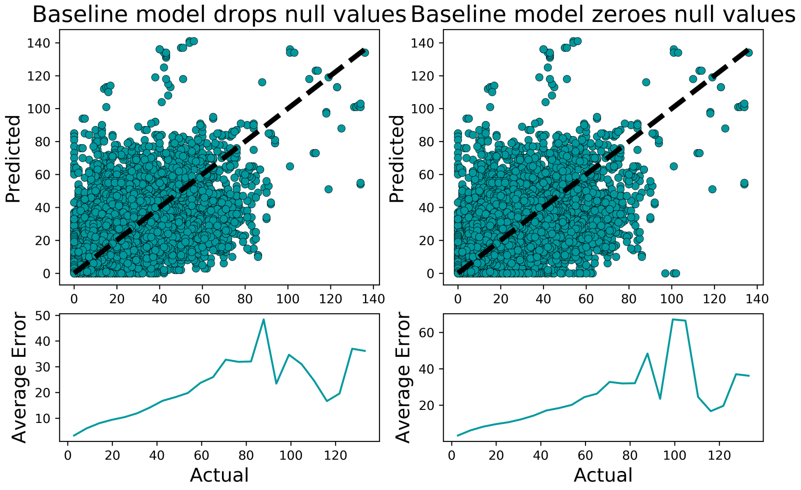


Figure D.4: Prediction vs. actuality of baseline models (Q-link 1)

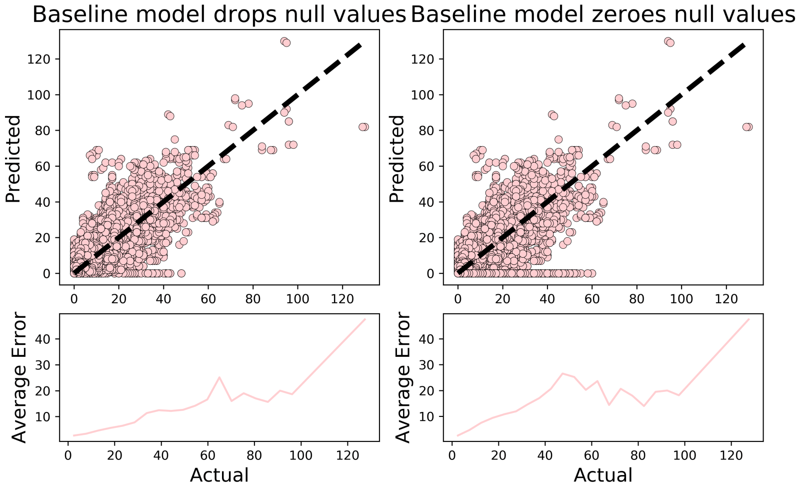


Figure D.5: Prediction vs. actuality of baseline models (line 50)

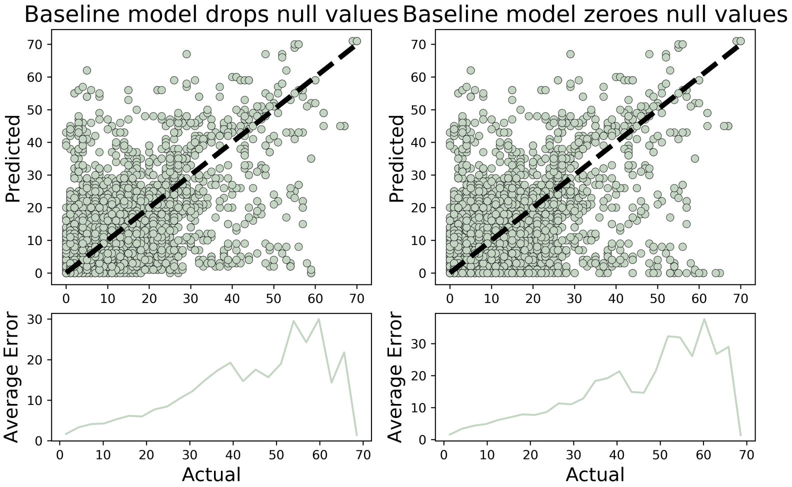


Figure D.6: Prediction vs. actuality of baseline models (line 35)

D.3. Actuality vs. Prediction Plots – Linear Models

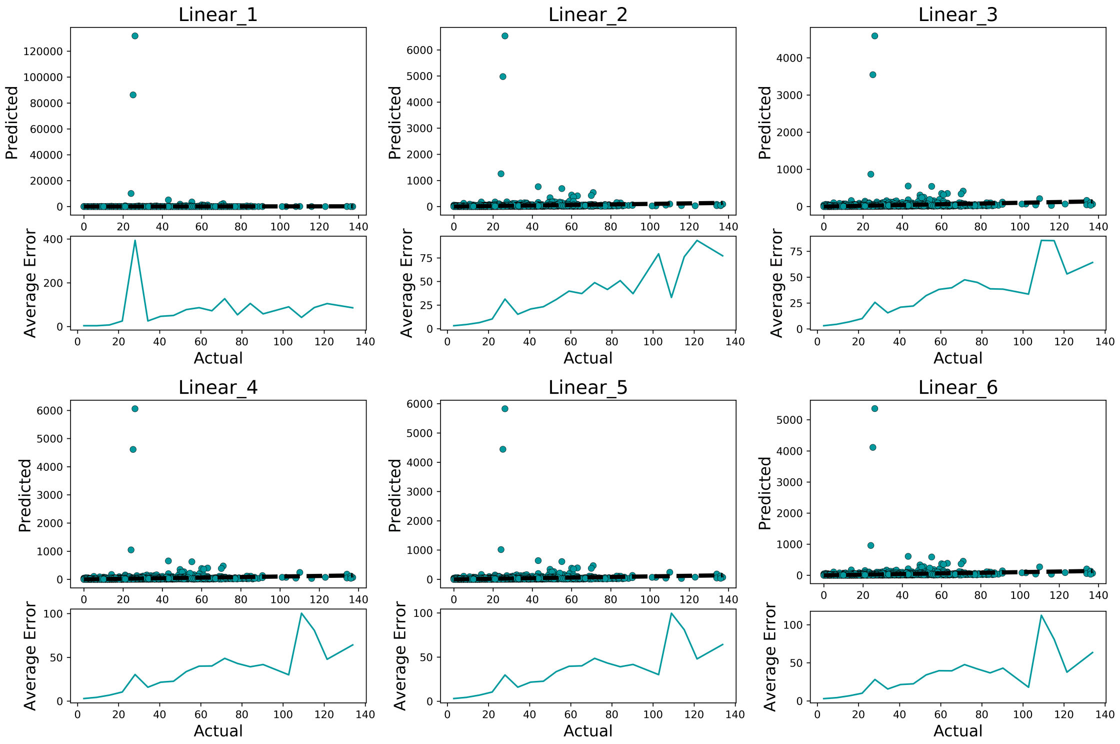


Figure D.7: Prediction vs. actuality of linear models (Q-link 1)

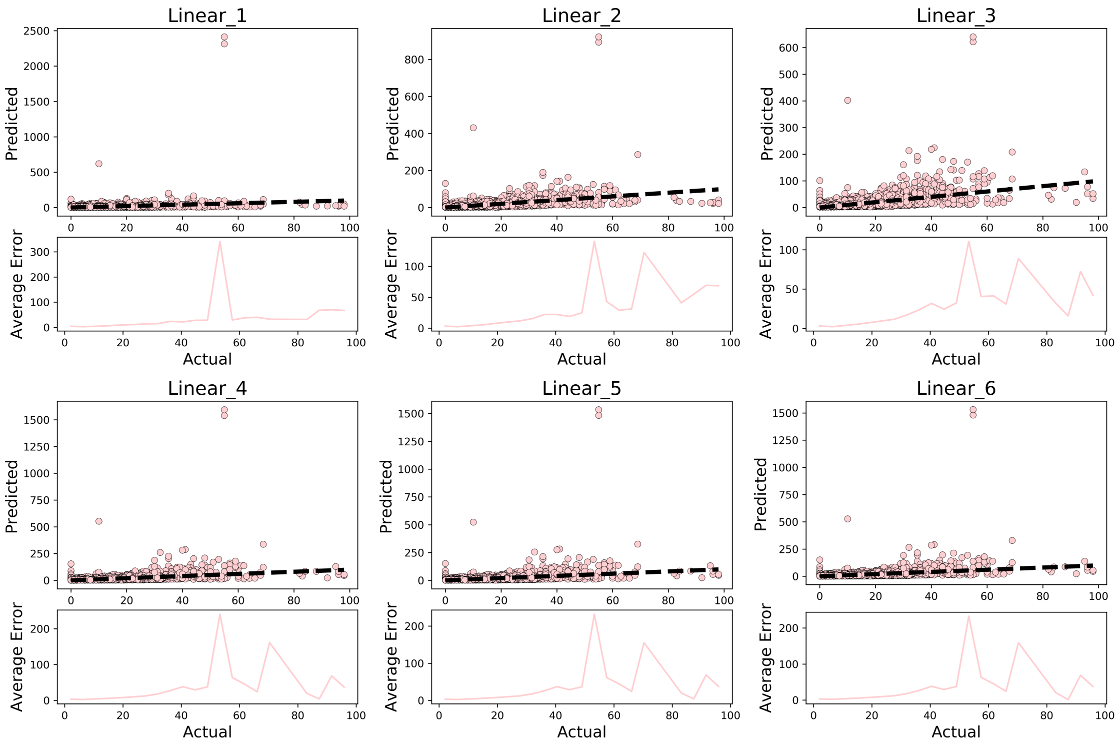


Figure D.8: Prediction vs. actuality of linear models (line 50)

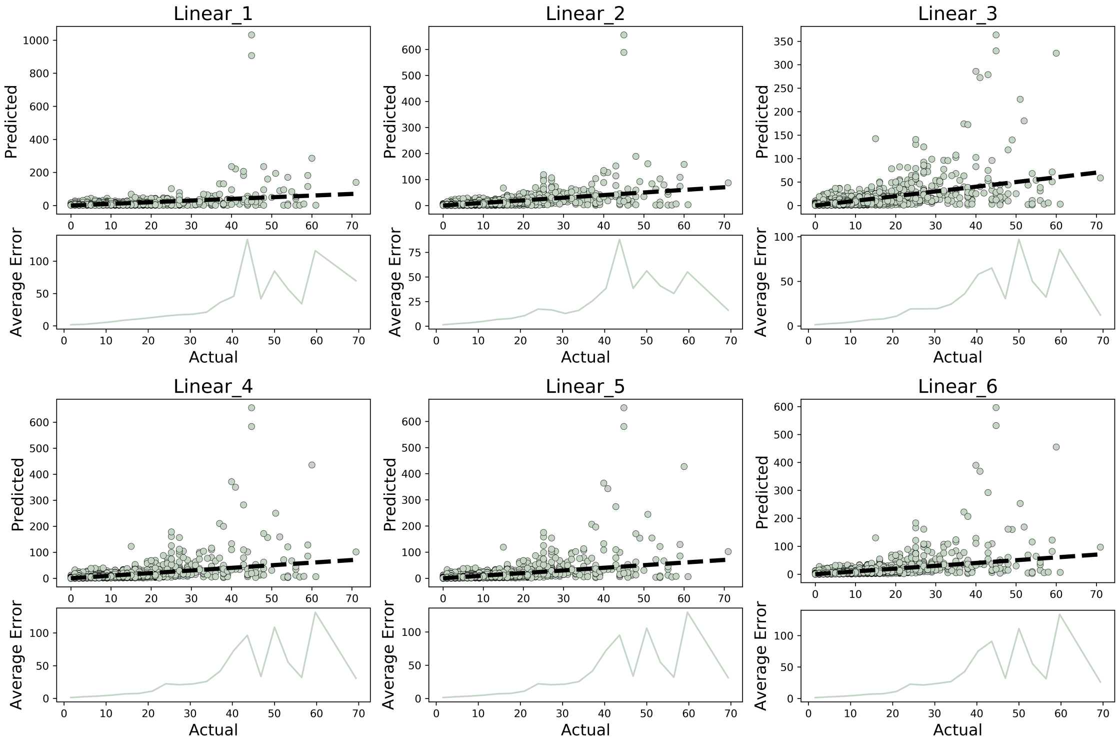


Figure D.9: Prediction vs. actuality of linear models (line 35)

D.4. Residual vs. Prediction Plots

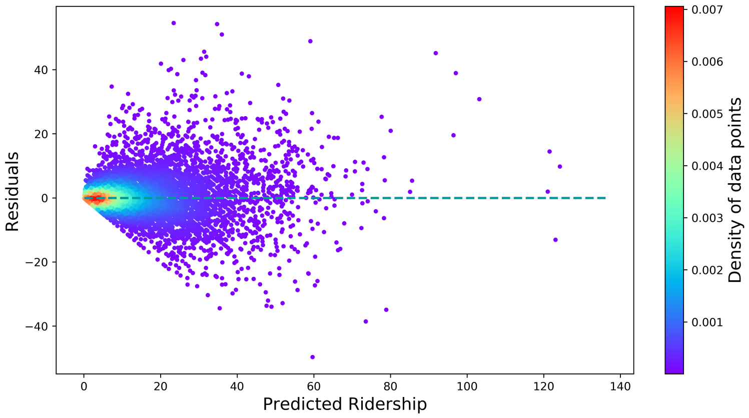


Figure D.10: Residual plots of Q-link 1 (Random Forest Regression)

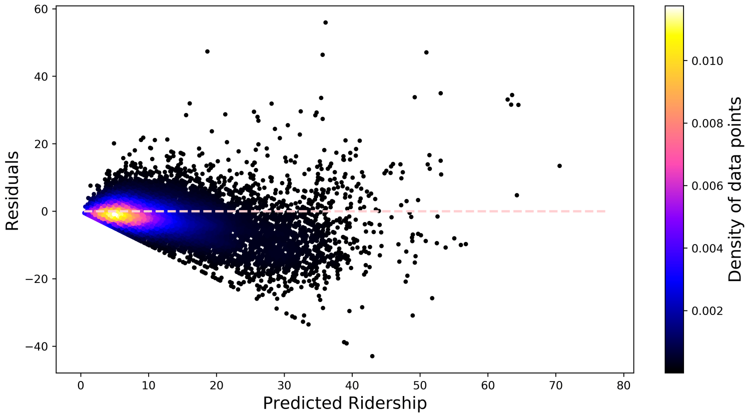


Figure D.11: Residual plots of line 50 (Random Forest Regression)

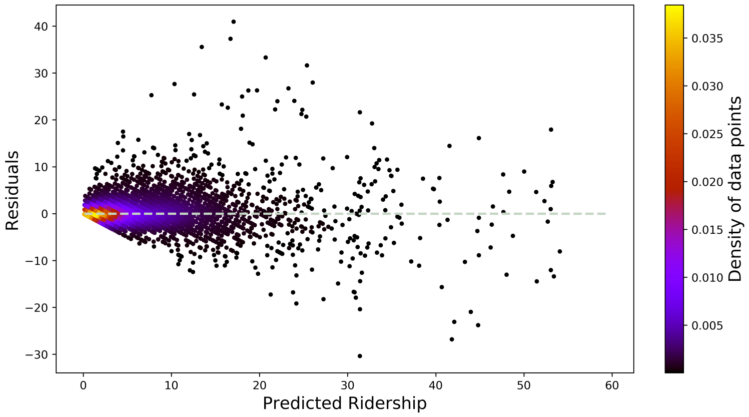


Figure D.12: Residual plots of line 35 (Random Forest Regression)

D.5. Distribution of Errors

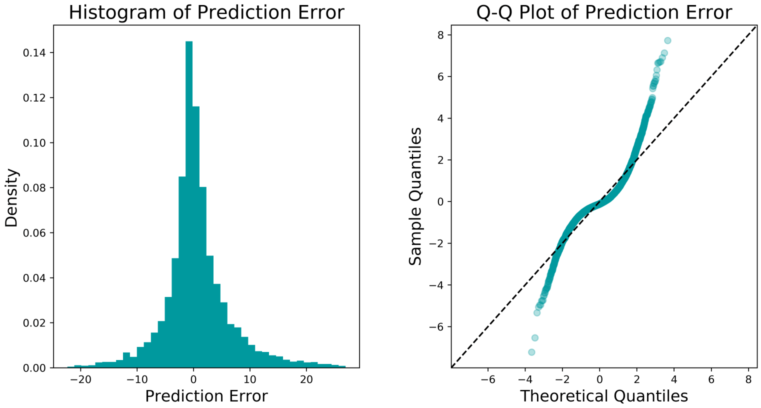


Figure D.13: Distribution of errors of Q-link 1 (Random Forest Regression)

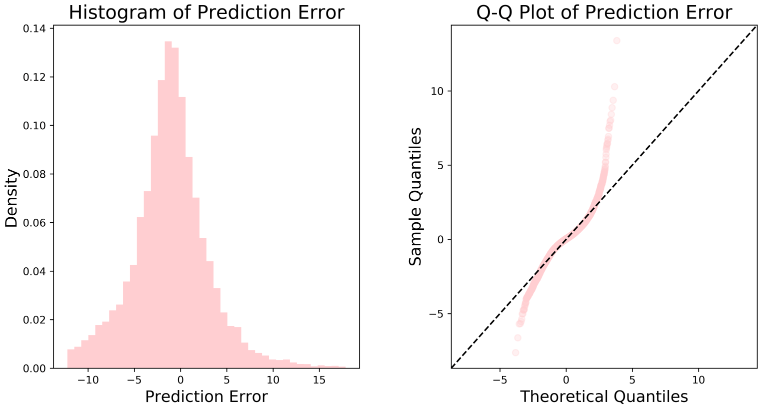


Figure D.14: Distribution of errors of line 50 (Random Forest Regression)

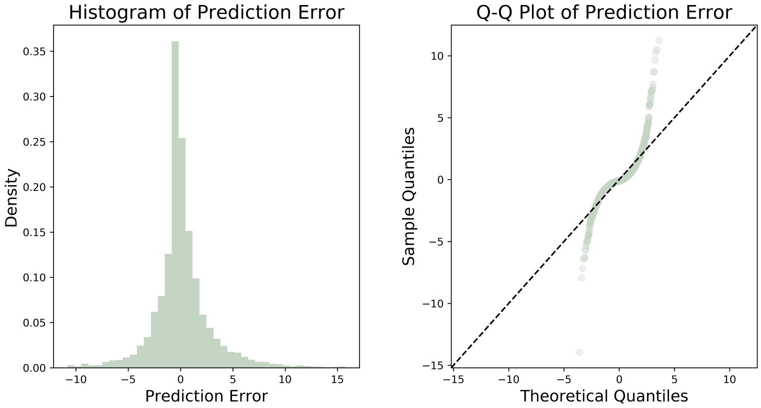


Figure D.15: Distribution of errors of line 35 (Random Forest Regression)

D.6. Residuals per Scenario

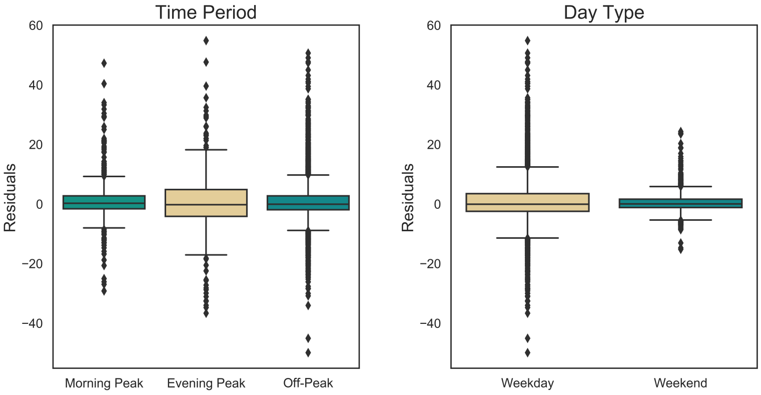


Figure D.16: Residuals of Q-link 1 per scenario (Random Forest Regression)

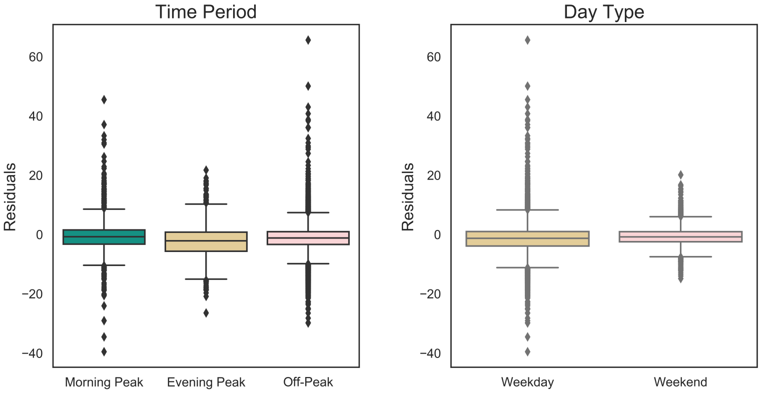


Figure D.17: Residuals of line 50 per scenario (Random Forest Regression)

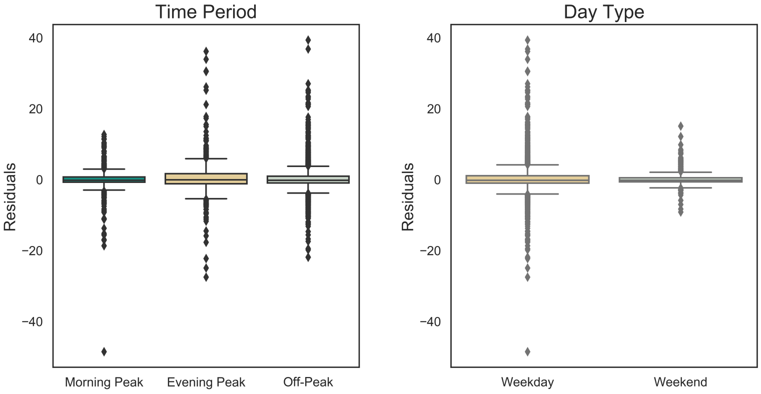


Figure D.18: Residuals of line 35 per scenario (Random Forest Regression)

D.7. Feature Importance

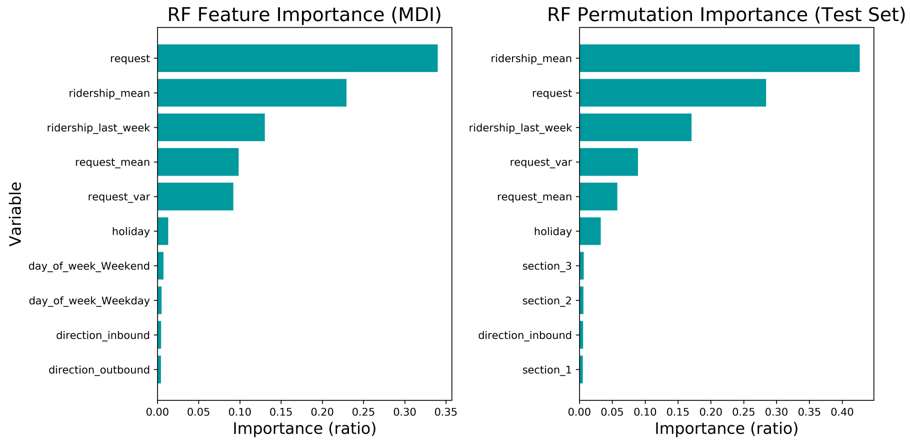


Figure D.19: Feature importance of Q-link 1 (Random Forest Regression)

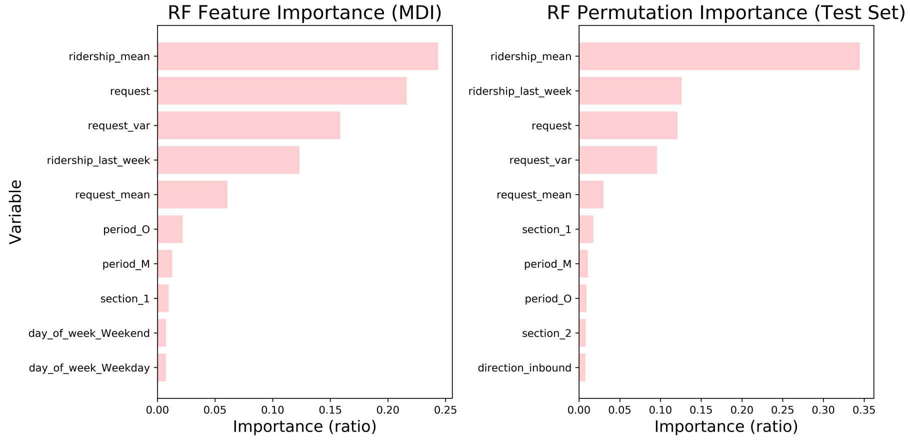


Figure D.20: Feature importance of line 50 (Random Forest Regression)

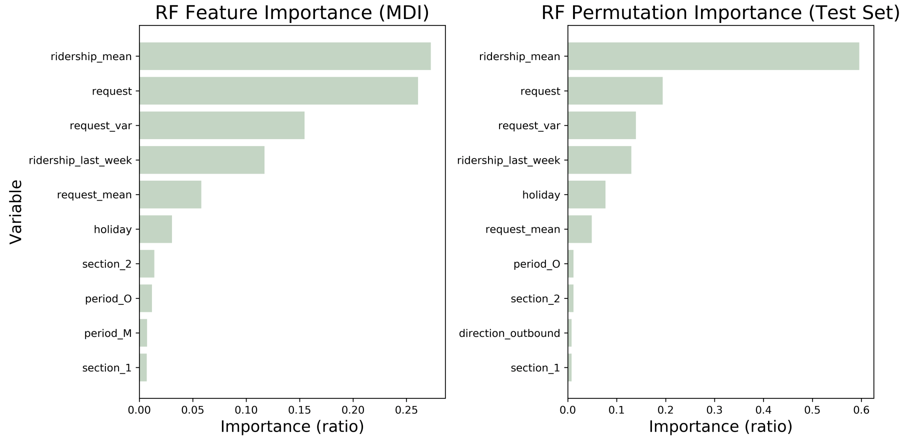


Figure D.21: Feature importance of line 35 (Random Forest Regression)

Table D.1: Parallel table of the section names

	Section_1	**Section_2**	**Section_3**
Qliner 300	Weerdingerstraat,Emmen–Stationuitstaphalte,Emmen	v.K.Verschuurbrug,Groningen–N34,Zuidlaren	Stationuitstaphalte,Emmen–Weerdingerstraat,Emmen
Q-link 1	Nijenborgh,Groningen–Zernikeplein,Groningen	ZernikeNoord(PerronC),Groningen–Hoogeweg,Groningen	Prof.Uilkensweg,Groningen–P+RReitdiep,Groningen
Line 50	Oostersingel,Assen–Station(PerronB),Assen	Rolderstraat,Assen–Oostersingel,Assen	N/A
Line 35	H.Colleniusstraat,Groningen–RembrandtvRijnstr.,Groningen	W.Barentzstraat,Groningen–Westerhaven(PerronB),Groningen	N/A

D.8. PDP

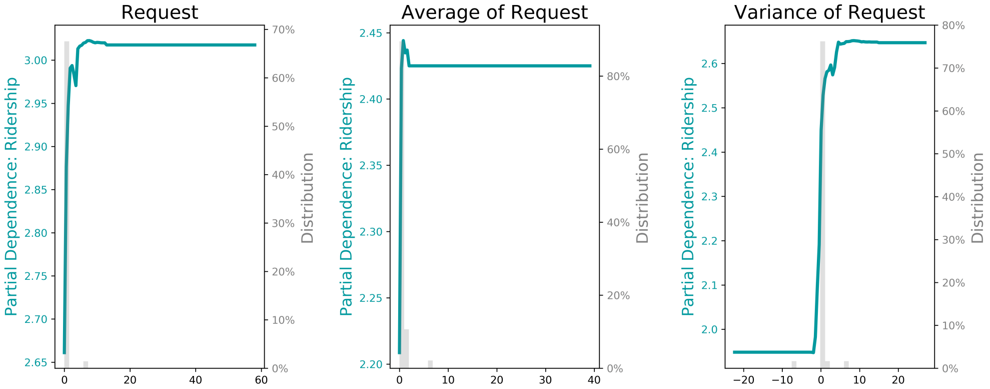


Figure D.22: PDP of Q-link 1 (Random Forest Regression)

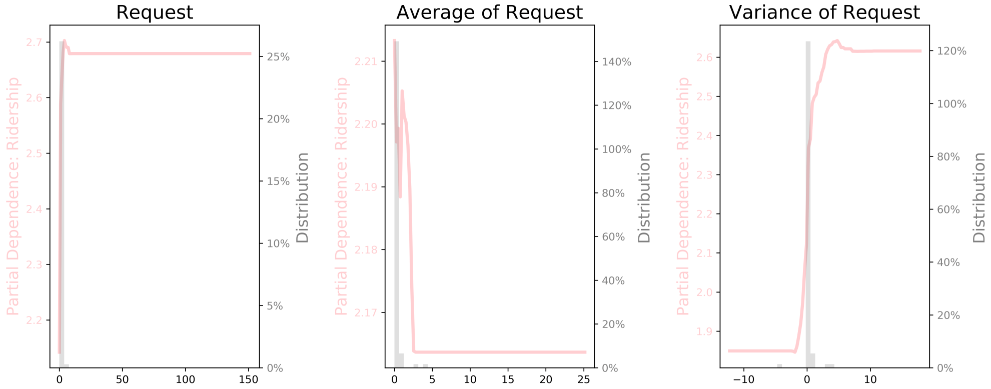


Figure D.23: PDP of line 50 (Random Forest Regression)

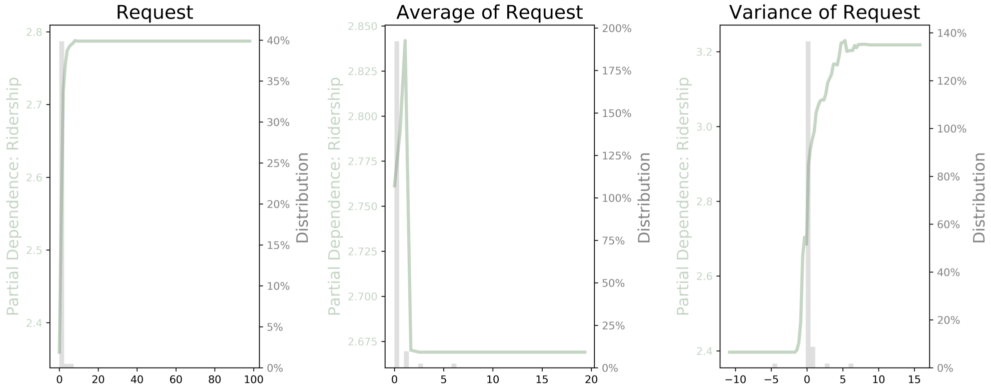


Figure D.24: PDP of line 35 (Random Forest Regression)

Bibliography

Ben-David, A. (1995). Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning*, 19(1):29–43.

Boutaba, R., Salahuddin, M., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., and Caicedo Rendon, O. (2018). A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9.

Brakewood, C. and Watkins, K. (2019). A literature review of the passenger benefits of real-time transit information. *Transport Reviews*, 39(3):327–356.

Branco, P., Torgo, L., and Ribeiro, R. P. (2017). Smogn: a pre-processing approach for imbalanced regression. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 36–50.

Bre, F., Gimenez, J. M., and Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158:1429 – 1441.

Bregman, S. (2012). *Uses of Social Media in Public Transportation*. The National Academies Press, Washington, DC.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Browne, M. W. (2000). Cross-validation methods. *Journal of Mathematical Psychology*, 44(1):108 – 132.

Cawley, G. C. and Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.*, 11:2079–2107.

Chakour, V. and Eluru, N. (2016). Examining the influence of stop level infrastructure and built environment on bus ridership in Montreal. *Journal of Transport Geography*, 51:205–217.

Chan, S. and Miranda-Moreno, L. (2013). A station-level ridership model for the metro network in montreal, quebec. *Canadian Journal of Civil Engineering*, 40(3):254–262.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Chen, C. and Breiman, L. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*.

Chiang, W. C., Russell, R. A., and Urban, T. L. (2011). Forecasting ridership for a metropolitan transit authority. *Transportation Research Part A: Policy and Practice*, 45(7):696–705.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

De Regt, K., Cats, O., Van Oort, N., and Van Lint, H. (2017). Investigating potential transit ridership by fusing smartcard and global system for mobile communications data. *Transportation Research Record*, 2652(1):50–58.

Dey, A. (2016). Machine learning algorithms: a review. *International Journal of Computer Science and Information Technologies*, 7(3):1174–1179.

Ding, C., Wang, D., Ma, X., and Li, H. (2016). Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees. *Sustainability*, 8(11).

Dissanayake, D. and Morikawa, T. (2010). Investigating household vehicle ownership, mode choice and trip sharing decisions using a combined revealed preference/stated preference Nested Logit model: case study in Bangkok Metropolitan Region. *Journal of Transport Geography*, 18(3):402–410.

Doane, D. P. (1976). Aesthetic frequency classifications. *The American Statistician*, 30(4):181–183.

Elias, D., Nadler, F., Stehno, J., Krösche, J., and Lindorfer, M. (2016). Somobil – improving public transport planning through mobile phone data analysis. *Transportation Research Procedia*, 14:4478 – 4485. Transport Research Arena TRA2016.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from imbalanced data sets*. Springer.

Fernández, A., Garcia, S., Herrera, F., and Chawla, N. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, 61:863–905.

Fisher, A., Rudin, C., and Dominici, F. (2018). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously.

Frasa, T. (2018). Overvolle bussen naar deltion zwolle: Jos (18) moet zeven volle bussen lang wachten. *de Stentor*.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Geisser, S. (1993). *Predictive inference*, volume 55. CRC press.

Gummadi, R. and Edara, S. R. (2019). Prediction of passenger flow of transit buses over a period of time using artificial neural network. In Yang, X.-S., Sherratt, S., Dey, N., and Joshi, A., editors, *Third International Congress on Information and Communication Technology*, pages 963–971, Singapore. Springer Singapore.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

He, H. and Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons.

Hopkins, S., Dettori, J. R., and Chapman, J. R. (2018). Parametric and nonparametric tests in spine research: Why do they matter? *Global spine journal*, 8(6):652–654.

Horowitz, A. J. (1984). Simplifications for single-route transit-ridership forecasting models. *Transportation*, 12(3):261–275.

Idris, A. O., Habib, K. M. N., and Shalaby, A. (2015). An investigation on the performances of mode shift models in transit ridership forecasting. *Transportation Research Part A: Policy and Practice*, 78:551 – 565.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.

Jin, X., Zhang, Y., Li, L., and Hu, J. (2008). Robust pca-based abnormal traffic flow pattern isolation and loop detector fault detection. *Tsinghua Science and Technology*, 13(6):829–835.

Karnberger, S. and Antoniou, C. (2020). Network–wide prediction of public transportation ridership using spatio–temporal link–level information. *Journal of Transport Geography*, 82(May 2019):102549.

Khoshgoftaar, T. M., Golawala, M., and Van Hulse, J. (2007). An empirical study of learning from imbalanced data using random forest. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, volume 2, pages 310–317. IEEE.

Kuhn, M., Johnson, K., et al. (2013). *Applied predictive modeling*, volume 26. Springer.

Lee, J. W., Yeung, D. S., and Wang, X. (2003). Monotonic decision tree for ordinal classification. In *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, volume 3, pages 2623–2628. IEEE.

Li, J., Shi, X., Jiang, Z., Li, Y., and Jia, H. (2015a). Transit ridership prediction of Changchun light rail line 3. *Sixth International Conference on Electronics and Information Engineering*, 9794(December 2015):97942R.

Li, L., Wang, J., Song, Z., Dong, Z., and Wu, B. (2015b). Analysing the impact of weather on bus ridership using smart card data. *IET Intelligent Transport Systems*, 9(2):221–229.

Li, Y., Zheng, Y., Zhang, H., and Chen, L. (2015c). Traffic prediction in a bike-sharing system. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '15, New York, NY, USA. Association for Computing Machinery.

Loh, W.-Y. (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, 1(1):14–23.

Louppe, G. (2014). Understanding random forests: From theory to practice.

Metcalf, L. and Casey, W. (2016). Chapter 4 - introduction to data analysis. In Metcalf, L. and Casey, W., editors, *Cybersecurity and Applied Mathematics*, pages 43 – 65. Syngress, Boston.

Ministry of Infrastructure and Water Management (2019). Public transport in 2040: Outlines of a vision for the future. *Government of the Netherlands*.

Molnar, C. (2019). *Interpretable Machine Learning*. Lulu. `https://christophm.github.io/interpretable-ml-book/`.

Mulley, C., Clifton, G. T., Balbontin, C., and Ma, L. (2017). Information for travelling: Awareness and usage of the various sources of information available to public transport users in nsw. *Transportation Research Part A: Policy and Practice*, 101:111–132.

Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848.

Ohler., F., Krempels., K., and Möbus., S. (2017). Forecasting public transportation capacity utilisation considering external factors. In *Proceedings of the 3rd International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS,*, pages 300–311. INSTICC, SciTePress.

Parvandeh, S., Yeh, H.-W., Paulus, M. P., and McKinney, B. A. (2020). Consensus features nested cross-validation. *bioRxiv*.

Pel, A. J., Bel, N. H., and Pieters, M. (2014). Including passengers ' response to crowding in the Dutch national train passenger assignment model. *Transportation Research Part A*, 66:111–126.

Pereira, F. C., Rodrigues, F., and Ben-Akiva, M. (2015). Using data from the web to predict public transport arrivals under special events scenarios. *Journal of Intelligent Transportation Systems*, 19(3):273–288.

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301.

Pucher, J. and Renne, J. L. (2005). Rural mobility and mode choice: Evidence from the 2001 national household travel survey. *Transportation*, 32(2):165–186.

Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd.

Rokach, L. and Maimon, O. Z. (2008). *Data mining with decision trees: theory and applications*, volume 69. World scientific.

Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edition.

Scherer, M. and Dziekan, K. (2012). Bus or rail: An approach to explain the psychological rail factor. *Journal of Public Transportation*, 15 (1):75–93.

Shaheen, S., Cohen, A., and Martin, E. (2017). *Smartphone App Evolution and Early Understanding from a Multimodal App User Survey*, pages 149–164. Springer International Publishing, Cham.

Shin, Y. (2015). Application of boosting regression trees to preliminary cost estimation in building construction projects. *Computational intelligence and neuroscience*, 2015.

Smola, A. and Vishwanathan, S. (2008). Introduction to machine learning. *Cambridge University, UK*, 32(34):2008.

Stopher, P. R. (1992). Development of a route level patronage forecasting method. *Transportation*, 19(3):201–220.

Tang, T., Liu, R., and Choudhury, C. (2020). Incorporating weather conditions and travel history in estimating the alighting bus stops from smart card data. *Sustainable Cities and Society*, 53(July 2019):101927.

Tao, S., Corcoran, J., Hickman, M., and Stimson, R. (2016). The influence of weather on local geographical patterns of bus usage. *Journal of Transport Geography*, 54:66–80.

Taylor, B. D., Miller, D., Iseki, H., and Fink, C. (2009). Nature and/or nurture? analyzing the determinants of transit ridership across us urbanized areas. *Transportation Research Part A: Policy and Practice*, 43(1):60 – 77.

Tukey, J. W. (1977). *Exploratory data analysis*, volume 2. Reading, MA.

Van De Kamp, R., Feelders, A., and Barile, N. (2009). Isotonic classification trees. In *International Symposium on Intelligent Data Analysis*, pages 405–416. Springer.

Van Oort, N., Boterman, J. W., and Van Nes, R. (2012). The impact of scheduling on service reliability: trip-time determination and holding points in long-headway services. *Public Transport*, 4(1):39–56.

Van Oort, N., Brands, T., and de Romph, E. (2015a). Short-term prediction of ridership on public transport with smart card data. *Transportation research record*, 2535:105–111.

Van Oort, N., Drost, M., Brands, T., and Yap, M. (2015b). Data-driven public transport ridership prediction approach including comfort aspects. In *Proceedings of Conference on Advanced Systems in Public Transport, 19-23 July 2015, Rotterdam.* CASPT.

Van Oort, N., Sparing, D., Brands, T., and M.P. Goverde, R. (2015). Data driven improvements in public transport : the Dutch example. *Public Transport*, 7(3):369–389.

Van Roosmalen, J. (2019). Forecasting bus ridership with trip planner usage data : a machine learning application. Master's thesis, Universiteit Twente.

Van Sloten, J. and Van Rooijen, M. (2019). Scholieren de dupe van vakantiedienstregeling, kas (14) uit zelhem komt dagelijks te laat op school. *de Gelderlander*.

Vogel, P., Greiser, T., and Mattfeld, D. C. (2011). Understanding bike-sharing systems using Data Mining: Exploring activity patterns. *Procedia - Social and Behavioral Sciences*, 20:514–523.

Wang, X., Zhang, N., Zhang, Y., and Shi, Z. (2018). Forecasting of short-term metro ridership with support vector machine online model. *Journal of Advanced Transportation*, 2018:3189238.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37 – 52. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

Xu, C., Ji, J., and Liu, P. (2018). The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. *Transportation Research Part C: Emerging Technologies*, 95(July):47–60.

Xue, R., Sun, D. J., and Chen, S. (2015). Short-term bus passenger demand prediction based on time series model and interactive multiple model approach. *Discrete Dynamics in Nature and Society*, 2015(i).

Yamaguchi, T., As, M., and Mine, T. (2019). Prediction of Bus Delay over Intervals on Various Kinds of Routes Using Bus Probe Data. *Proceedings - 5th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, BDCAT 2018*, pages 97–106.

Yeboah, G., Cottrill, C. D., Nelson, J. D., Corsar, D., Markovic, M., and Edwards, P. (2019). Understanding factors influencing public transport passengers' pre-travel information-seeking behaviour. *Public Transport*, 11(1):135–158.

Zheng, A. and Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.".

Zhou, C., Dai, P., Wang, F., and Zhang, Z. (2016). Predicting the passenger demand on bus services for mobile users. *Pervasive and Mobile Computing*, 25(2013):48–66.