Shapley Values

A Comparison of Definitions and Approximation Methods

J. T. de Jong

Shapley Values

A Comparison of Definitions and Approximation Methods

by

J. T. de Jong

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday August 13, 2021 at 11:00 AM.

Student number:	4604202	
Project duration:	December 1, 2020 – September 1, 2021	
Thesis committee:	dr. N. Parolya,	TU Delft, supervisor
	dr. R. J. Fokkink,	TU Delft
	prof. dr. M. K. Francke,	Ortec Finance, University of Amsterdam

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

The Shapley value method is an explanatory method that describes the feature attribution of Machine Learning models. There are three different definitions of the Shapley values, namely Conditional Expectation Shapley, Marginal Expectation Shapley and Baseline Shapley. A comparison is made between the three definitions and they are applied to one statistical and two Machine Learning models that predict house transaction prices. Most existing methods to approximate Shapley values assume independence, which is in practice often violated. An existing copula-based method that tries to take into account the dependency is extended to apply to problems with continuous and discrete features. This copula-based method approximates the Shapley values more accurately than other methods. The Conditional Expectation Shapley values give unnatural explanations, therefore other definitions of the Shapley values are more suitable. The Baseline Shapley values seem to be the most promising since there is an accurate and fast approximation method and the B Shapley values are the easiest to interpret.

Keywords— Shapley values, Machine Learning, explanatory method

Preface

This Master Thesis is performed at Ortec Finance as part of the Applied Mathematics program with specialisation Financial Engineering at the TU Delft. The topic of this Master Thesis is Shapley Values, how they are defined and how they can be approximated. This project is part of a larger plan to apply Machine Learning in real estate valuation, especially for taxation purposes. For that, it is necessary that decent explanations can be extracted from the Machine Learning model. I hope that this Master Thesis can contribute to making that happen.

At first, I would like to thank Ortec Finance for offering me the chance to write this thesis with them. In particular, I would like to thank Marc Francke and David Kroon for their time, effort, feedback, critical questions and suggestions during our weekly meetings and outside those meetings. Furthermore, I would like to thank Nestor Parolya for his supervision from the TU Delft and for providing valuable feedback on my project. Moreover, I would like to thank all my colleagues from Ortec Finance that provided me with the right tools, equipment and data with special mentions to Alex de Geus, Raymond Havekes, Frank Verbeek and Robert Melman. Lastly, I would like to thank my family and friends for their support outside office hours.

J. T. de Jong
Delft, August 2021

Contents

Acronyms

ALE Accumulated Local Effects. 10

CFI Counterfactual Instances. vii, 12, 13

FI Feature Interaction. 10

HTM Hierarchical Trend Model. vii, 25, 26, 30, 32–37, 39, 41–43, 45, 46, 53

ICE Individual Conditional Expectation. 10

LGBM Light Gradient Boosting Machine. 26, 32–35, 37, 39

LIME Local Interpretable Model-agnostic Explanations. vii, 10, 13

ML Machine Learning. iii, v, 5, 7–13, 19, 26, 32, 33, 37, 39, 43

MLP Multi-layer Perceptron. 26, 32–34, 36, 37, 39

PDP Partial Dependence Plot. 10

PFI Permutation Feature Importance. 10

SHAP Shapley. vii, 11, 13, 15, 16

SR Scoped Rules. vii, 11, 13

XAI Explainable Artificial Intelligence. 5

1 Introduction

Machine Learning methods are getting increasingly popular. Nowadays, there are many advanced Machine Learning (ML) models and more are invented every day. Combining Machine Learning models in ensemble models give endless possibilities of combinations. The main purpose is to get models that are more accurate in making predictions. However, the increased accuracy is accompanied by increased complexity very often. The Machine Learning models can make very accurate predictions but what is happening inside to come to that prediction is difficult, sometimes impossible, to interpret for humans. For that reason, Explainable Artificial Intelligence (XAI) is an upcoming subject in the Machine Learning discipline. Many applications of Machine Learning models demand an explanation next to a prediction. That opens the way for explanatory methods, algorithms that extract an explanation from a Machine Learning model by looking at the data and evaluating the model.

One specific application that will be highlighted in this research is the real estate valuation problem. The is the problem of predicting house prices, which is a regression problem with tabular data. But this problem has certain properties that apply in other situations. For instance, house prediction is based on features that are discrete, continuous and categorical and there is often and strong correlation between the features.

Due to the high demand for explanation techniques, many explanatory methods have been invented. Some methods are model specific, but there are also methods that can be applied to any ML model, the so-called agnostic explanatory methods. The advantage of agnostic methods is that you can apply any ML that fits the problem well.

There are many different methods, but the most promising methods will be explained in Chapter 2. For that, first, an explainability framework is sketched. From those methods, the most suitable method is chosen, which turns out to be the Shapley values method. However, there are still some practical problems with the method. First, there are multiple definitions of the Shapley values, so one needs to find out which definition suits the problem the best. That brings us to the main question of this research.

> *Which of the three Shapley definitions is most suitable for the real estate valuation problem?*

It turns out that the existing approximation methods do not determine all three Shapley values accurately. These methods often assume independence between features, while this is not the case in practice. That leads to a sub-question that will be investigated.

> *For each of the three Shapley values definitions, which approximation method determines the Shapley values most accurately?*

To answer the main question, there are three definitions of Shapley values given in Chapter 3. Also, the theoretically properties and interpretation of each definition will be explained. To test how the different definitions behave in practice, the Shapley values are applied to the real estate valuation problem. This is the problem of predicting house prices using a model. House transaction data of the Netherlands from 2008 to 2016 is used to train a statistical and two ML models to which the Shapley values method is applied.

For an answer to the sub-question, a Shapley values approximation method of Aas et. al. [1] and a new approximation method are introduced in Chapter 4. Also, the copula method of Aas et. al. [1] is extended to discrete features and is tested in a simulated environment. Discrete features occur in many problems, for instance, the real estate valuation problem, but in many more.

Chapter 5 describes the experiments that will be performed to find solutions for the two problems stated above. The results of these experiments are presented in Chapter 6. Finally, a conclusion of these results will be drawn in Chapter 7.

2

Model Explainability

The definition of an explanation is not straightforward. Many researchers have attempted to define what an explanation is. One that is used often is:

> To explain an event is to provide some information about its causal history. - Lewis [2]

Miller put this definition in a more ML framework:

> The degree to which an observer can understand the cause of a decision. - Miller [3]

Both refer to the word cause or causal for the definition of explanation. There are often many causes possible to explain the same phenomena. That makes that explanations of the same event can differ much because a different explanation can refer to a different cause of the event. Some explanations are more useful than others. To understand when an explanation is useful, it first needs to be clear what the reason for an explanation is. The red line through this chapter will be the real estate valuation framework.

2.1. Reasons for an Explanation

In general, there are multiple reasons why people demand not only a prediction but also an explanation [4, 5]. In most cases where machine learning models are applied, the prediction alone does not tell the whole story.

If there is an explanation why the ML predicts what it predicts, there is more **trust** in the ML model. Imagine that you and your neighbour both have your house for sale. You let an ML model predict a reasonable asking price for both of the houses. The ML model tells you that your neighbour's house is worth €20.000 more than your house. You are completely thunderstruck since you have approximately the same house at approximately the same location. You demand an explanation from the ML model why it predicts that your house is less worth than your neighbour's. A clarifying explanation that your neighbour has a garage will give you more trust that the model is reasonable.

A good explanation could give the user of an ML model an **informative** explanation. For instance, which features are the most important, on what domain the model works appropriately, how the model could be improved and what the weak spots of the model are. For every prediction that an ML model makes, there could be a lot happening inside the model, but the model only returns a single value. Getting more insight into the processes inside the machine can be more useful than only looking at the prediction.

ML could be used to find indications of possible **causal** relationships. If the explanation of an ML model hints at a relationship, then a researcher can investigate further whether this is relation is causal. Nevertheless, let it be clear that ML models are not based on causal relationships. The objective of training an ML model is to maximize the accuracy of predictions. That means that an ML method can base its predictions on non-causal grounds if this increases the predictive power. So one should always handle the results of an ML model with care.

Another actual problem of ML being a black box is that it could be **unfair** or **unethical**. ML have a risk of implicit bias which could have an undesired effect when an ML model makes a prediction based on associations, such as sex, age or race. This might also become a problem in real estate valuation when more features are added, think of personal data of the inhabitant or the neighbours. An explanatory method might reveal implicit bias in ML methods, such that the creator or regulator can come into action.

2.2. Properties of an Explanation

An explanation has multiple properties that say something about the quality of the explanation. Molnar made an extensive list of properties [6]. Here the most important properties are summarized.

Comprehensibility

An explanation should not be too simplified and also not too complex. This is of course dependent on the person that gets the explanation, or the explainee as Miller called it [3]. So before a comprehensible explanation can be made, you need to determine the knowledge of the explainee. Then you can build an explanation on top of that knowledge in understandable language.

Comprehensibility is difficult to get a grip on. It is hard to measure since it depends on the level of understanding of the explainee. For some ML models, the level of comprehensibility can be approximated by model internals, such as the depth of a decision tree or the number of non-zero weights in a linear regression model. For other ML methods is that more difficult. The definition of Ribeiro can be helpful to get a measure of comprehensibility [7]

> whether a human understands a model enough to make accurate predictions about its behaviour on unseen instances.

This can be tested in an experiment where a model and a human both predict unseen instances, which is what Ribeiro did in fact [7].

Goal-Oriented

The goal of the explanation should be kept in mind when an explanation is constructed. If someone asks for an explanation, then he will only be satisfied with an explanation that answers his questions. So after you have determined what who your explainee is, you should find out what he wants to know and why. The four reasons of explanation described in Paragraph 2.1 can all have a different approach to explaining.

Stability

Also a desirable property of an explanation is that it is stable. If two instances have very similar features and the prediction of a fixed model is also similar, then both explanations should also be comparable. If the explanations differ much then the explainee would not have much trust in the explanatory method.

For instance, you and your neighbour have (almost) the same house and you value both houses with a model. You both get the same house value, however, the explanation tells you that the price of your house is due to the neighbourhood you are living in while your neighbours' house price is dominated by the living space. This is an undesirable situation since you don't know what the real most important feature is for both houses. Therefore high stability is a good property of an explanation.

Certainty

Statistical methods do not only produce predictions, but also confidence intervals. These intervals are very useful since they tell you something about how certain the model is about its prediction. If you get a prediction with wide prediction intervals, then at least you know that your model was struggling with the input. So maybe for that specific input instance, the model has a weak spot.

However, most ML methods do not produce confidence intervals. So from the single prediction, you don't get the information if the input instance is in a well-trained region of the model, or that it might be the first time that the model sees an instance of this kind. That is especially an issue if you have an instance that is from a region far from the training data of the model. Most ML models do not generalize so well for instances far from their training data, but they won't tell you that, so they produce a prediction. To which extent the explanation reflects the certainty of the model is also a property.

Representativeness

The representativeness of an explanation is the range of instances the explanation covers. An explanation can cover the whole model, for instance, that the amount of living space is a dominant factor for the house price. On the other hand, an explanation could also only be accurate in a local region. For example, in New York, a high-level apartment means a nice city view which increases the price. However in general a high-level apartment is often a small flatlet which reduces the price. A good explanation is not necessarily local or global. Both can be desired, depending on the explainee and the goal.

2.3. Properties of Explanatory Methods

If the intrinsic explainability of a model is not enough, then one can make use of a post hoc explanatory method. The main idea of these methods is that you already fitted a model on your data. Then you use the explanatory method to extract more information on how the model comes to a prediction.

There are different dimensions in given an explanation for an ML prediction. There are differences in how the explanation is derived, the range to which the explanation applies and the form of the explanation. Here follows a short description of these three aspects, inspired by Molnar [6].

Applicability

The applicability does not tells much about the explanation itself, but more about how the explanation is derived. The method for deriving the explanation might apply to every type of model. The main requirement is a function with input and output variables, and the method comes up with an explanation. This means that it is not only applicable to ML models but that it could also be applied to statistical models. These methods are called model-agnostic methods. The advantage is that they are flexible and that the results of different models can easily be compared.

On the contrary, some methods only work for a specific model. Sometimes the method uses certain model internals or the explanation is derived intrinsically. These are called model-specific explanatory methods.

Scope

An ML model could be explained on a global scale. Then you can say something about the features and interactions that influence the prediction in general. The scope of the explanatory translates directly to the representativeness of the explanation.

On the contrary, an explanation could describe how the prediction of a specific instance is made. However, this does not necessarily mean that this explanation also generalizes well to the whole range of the model. For example, the value of an apartment in a big city could very much be based on its livings space, because of the limited available space. On the other hand, this would probably be less important for a house in a rural area, since there is space enough.

Furthermore, there is everything between the extremes of the explanation of the whole model and the explanation of one single instance. One could think of an explanation for a certain subgroup or cluster.

Result

There are many ways of giving a good explanation and there is no one best way. It always depends on the situation and the explainee. Here follows a list of possible outcomes of an explanatory method.

The result could be a **feature summary statistic**. This is a list or table with a value for each feature that says something about the importance of the feature. One could do this for every feature, but this might be inconvenient in problems with dozens of features, so one could also choose to do this only for a subset or for the most important features. It could also have a more complex interpretation, for instance, the effect of the interaction of features on the prediction.

Another result that is close to the feature summary statistic is the **feature summary visualization**. Instead of a list or table, this is a figure. That could, for instance, be a 2 or 3-dimensional plot, but also a pie chart is possible. Humans have problems with imagining more than 3 dimensions. That makes that these methods are often constrained to visualizing only 2 features.

The **model internals** are the outcome of parameters after the model is trained. That could be the coefficients of a linear regression model or the weights of a neural network. But that could also be the structure of a decision tree. For some models, this gives a human interpretable explanation such as the coefficients of a linear regression model. But for example, the weights of a deep neural network give no interpretable explanation of how the model predicts. Besides, very deep decision trees can be hard to interpret.

The result of an explanatory method could be a **data point**, another instance of the problem. That could be an instance that is representable for the input instance or that is contrastive to the input instance. If you want to know why your house has a certain value according to the model, then a useful explanation could be that your neighbour sold their house for the same price. Another useful explanation could be that your other neighbours sold their house for €20.000 more because they have a garage.

Another outcome could be an **intrinsically interpretable model**. If the model itself is not interpretable, then one could train an interpretable model that mimics the behaviour of the machine learning model. Linear regression with little fea-

tures or shallow decision trees are still interpretable by human, especially for machine learning specialists. Also, a simple set of rules could be seen as an interpretable model, which looks like a decision tree but with a planar structure.

Finally, the outcome of an explanatory method could also be a **conditional statement**. The conditional statement contains some conditions about feature values that need to be satisfied for the outcome to be of a certain kind. An example of such a conditional statement would be: if your house was 20 m^2 bigger and it had a garage, then the value would be €20.000 more.

2.4. Existing Explanatory Methods

There are many different explanatory methods with different properties as described above. Some explanatory methods are more sophisticated than others. Some easy to apply methods are Partial Dependence Plot (PDP), Individual Conditional Expectation (ICE), Accumulated Local Effects (ALE), Feature Interaction (FI) and Permutation Feature Importance (PFI). Molnar gives a good explanation of all these individual explanatory methods in his book [6]. All these methods give a feature statistic or feature visualization where you need certain knowledge to interpret the data. A more detailed review of the newest and most promising methods will follow below.

2.4.1. Local Interpretable Model-agnostic Explanations (LIME)

The LIME method is invented by Ribeiro et al. [8]. The idea is that you train a local surrogate model that tries to approximate the predictions of the underlying ML model.

Suppose you have an ML model and you can do predictions with it but you do not know anything about the model. If you want to understand why the model makes a prediction y on a specific instance x, you can use the LIME method. First, you generate many samples x^i in the neighbourhood of x and let the model make predictions y^i. Then you choose an interpretable surrogate model, for instance, a linear regression model. You train this model on the samples (x^i, y^i). If you have trained your model, then the β coefficients of your linear regression model, tell you something about how the ML attributes the features around your instance x.

The difficulty is how to sample in the neighbourhood of x. A logical choice is to perturb x in each direction with a normal variable with mean zero. But then one has to choose the size of the variance. If the variance is too small, then the neighbourhood around x is too small and the explanation has a very local scale. If the variance is too big, then the neighbourhood is too big and the surrogate model might not pick up the local relations. Then the explanation is not telling the whole story. What the right size of the neighbourhood is, is a difficult task and needs a lot of fine-tuning. In certain cases, the explanation changes completely if the neighbourhood size is modified.

Advantages
The choice of the surrogate model can make the explanation human friendly. For instance, the lasso model can create a short explanation in problems with many features, where only the most explaining features appear in the explanation.

Another advantage is that the features of the ML model are not necessarily the same as the features where the surrogate model is trained on. One could choose to aggregate some features in one feature to create a more interpretable feature. This is very useful where ML is applied to image classification. Then the input feature of the ML model are colour values of a pixel, but this can be aggregated to picture segments as input for the surrogate model.

The accuracy of the fitted surrogate model tells something about the certainty of the explanation. For instance, linear regression with very small confidence bounds is likely to give an accurate explanation.

The choice of the surrogate model does not depend on the ML model that is used. Suppose you choose a certain surrogate model because the explainees prefer that model, for instance, linear regression. Then the ML model can be exchanged by another model that has higher accuracy. The surrogate model can stay the same, it only has to be trained on the new ML model.

Disadvantages
Finding the right size of the neighbourhood is one of the main problems when using the LIME method. One needs to do a lot of fine-tuning to the neighbourhood size. Modifying the neighbourhood size can change the explanation, which makes that the explanatory method is inconsistent.

Another problem is that the method is not stable. Two instances that have similar features and a similar prediction can have a very different explanation [9].

2.4.2. Scoped Rules (SR)

The Scoped Rules or Anchors method explains in the form of local sufficient conditions for certain predictions. The method constructs rules that are called anchors. An anchor states something like 'if feature 1 falls in the interval $[a, b]$'. If all anchors hold, then you can give an estimate of the prediction that holds with high probability. The anchors method takes an input instance and returns which features should be kept (approximately) the same as the input instance, to keep the prediction also the same as the input instance.

A concrete example of an anchors explanation in the real estate valuation framework would look like this:

> If the property has a living space between $100m^2$ and $120m^2$,
>
> and if the property is between 5 and 8 years old,
>
> and if the property is located in the city of Amsterdam,
>
> then the value is between €300.000 and €320.000 with a probability of 0.95.

The `alibi` package in `Python` contains an implementation of the anchors method that is based on the method of Ribiero [7]. This implementation can handle numerical and categorical input features. However, the prediction should be categorical. Therefore the predictions have to be put in bins before the method can be applied.

Advantages

The output of the method is a text that is understandable for people, even for laymen. One can limit the number of anchor rules such that the explanation is selective and easier to understand.

This method overcomes the shortcoming of some other methods that try to locally linearize the model. If the model is locally highly non-linear, a linear model is too confident about its result and should not be trusted. The anchors method handles this issue by returning a coverage value, which indicates how much of the input data is covered by the explanation.

Disadvantages

If the feature space is large (in the order of tens or hundreds) then the anchors method needs a lot of computation time. The implementation of Alibi makes bins for every continuous feature and samples new instances from these bins. If there are many features and the bin size is chosen small, then there are a lot of permutations on the input instance possible. Therefore this implementation can have a very high computation time.

Furthermore, the result of the method can be unsatisfying for the explainee. For instance, if one wants a high probability bound for the prediction, then the bin size of the input feature might be so small, that the coverage becomes almost 0 and the anchors only apply for that specific instance. Furthermore, it might need a lot of anchors to get a high probability. If 30 anchor rules need to apply to an instance to make a probable prediction, then the interpretation of these rules becomes a bit cumbersome. One can force the method to limit the number of rules, but then the precision of your prediction could reduce significantly.

The method assumes that the input features are independent since it samples new instances. This is often not the case and it could be possible that if two features are highly correlated, that the method samples instances that are impossible in real life.

The method has a lot of hyperparameters that need to be tuned. Therefore it is not so easy, and might even be impossible to get a satisfying explanation from this method.

2.4.3. Shapley (SHAP)

The Shapley method is a feature attribution method, with its origins in Game Theory. The Shapley values were first used to assign the contribution of a single player to the payoff of a coalition. This is translated to a prediction game, where the features are the players and the prediction is the payoff. Then the Shapley method is a feature attribution method that describes how much each feature attributes to the predicted outcome. The attribution of a feature is called the Shapley value.

Here follows an example of the Shapley method to illustrate how it works. Suppose there is an ML model that predicts house prices. As input, it takes the surface area, the house type and the city of the house. Now, you give the ML an apartment with a surface area of $100m^2$ in Amsterdam. The ML predicts that the house is worth €300.000, but how much is each feature contributing to this price? For that, the Shapley values explain the difference between the average house price, which is determined at €250.000 for example, and the predicted house price, which was €300.000. Then a result of the Shapley method could be that the surface area of $100m^2$ contributes €30.000, the house type being apartment contributes -€20.000 and the fact that the house is in Amsterdam contributes €40.000. The sum of the Shapley values of

all features equals the difference between the prediction and the average,

$$€300.000 - €250.000 = €30.000 - €20.000 + €40.000.$$

To determine the Shapley values, one uses a method from Game Theory. The features are gathered in coalitions. Then the attribution of a feature is the prediction if the feature is in the coalition minus the prediction if the feature is excluded from the coalition. The Shapley value is the average over all possible coalitions. The prediction of a coalition is not straightforward, since the ML model is trained on all features, so how can it make a prediction when only a subset of features are known. For that, there are different solutions, as will be described in Chapter 3.

Advantages
The Shapley values have a few desirable properties, such as *efficiency*, *symmetry* and *dummy*. The efficiency property means that the Shapley values describe the full difference between the prediction and the average. The symmetry property tells that if two features contribute equally, then the Shapley values are equal. The dummy property makes that the feature that never changes the prediction, has a Shapley value of 0. The above properties are in line with human intuition. Also, these properties are a solid foundation when an explanatory method needs to hold in court.

The scope of the Shapley values can be modified by changing the background data set. The Shapley can be compared to a big data set or to a single instance. That makes that Shapley values have the possibility to give contrastive explanations by choosing a certain background data set. For instance, one can explain where the difference between two predictions comes from by making the background data set only a single point. In Chapter 3 these Shapley values will be defined as B Shapley values.

Disadvantages
The computation time for the Shapley values is exponential in the number of features. The Shapley values are namely the sum over all possible coalitions, so for N features this is a sum with 2^N terms. For models with many features, the Shapley values become very expensive to calculate. Luckily, there is an approximation method named the kernel Shap method. This method is implemented in `Python` in the `shap` package. This method makes it possible to calculate the Shapley values for models with more features, however, the approximation of the Shapley values to the exact Shapley values becomes worse for higher numbers of features.

Another disadvantage is that a background data set is necessary to calculate the Shapley value. A prediction function of the ML model is not enough, because of two reasons. First, the Shapley values explain the difference between the prediction and the average of the background data set. Second, the calculation method of the Shapley value makes a new random instance by drawing some features from the background data set. Without a background data set, the random samples can have feature values that are not possible in the real world.

The last and maybe the biggest issue is that the calculation of Shapley values has problems when there are correlated features. Current implementations assume independence between features, for instance the kernel Shap implementation [10]. Random sampling goes wrong when there are correlated features because each feature is sampled from the marginal distribution. If the features are dependent, then the random samples can have very unlikely features values. For example, if the features surface area and volume are drawn from their marginal distribution, then a house can occur with a surface area of $100m^2$ and a volume of $100m^3$, but this is a very unlikely instance.

2.4.4. Counterfactual Instances (CFI)

A type of explanation that is easily understandable for people without any knowledge of models is a counterfactual explanation. Counterfactual explanations can be defined as:

> "How the world had to be different for the outcome to be different" [11].

An example of a counterfactual explanation for the commercial real estate valuation problem would be:

> Now my house has 50 m^2 living space and is worth €100.000,-. If my house had 20 m^2 extra living space, then my house would be worth €30.000,- more.

The advantages of counterfactual explanations are that this way of explaining is comprehensible for people without any knowledge of the model. Furthermore, the explanation is selective and someone can get to know which feature (if possible) he needs to change to increase the value of his property by a certain amount.

Multiple algorithms in literature generate counterfactual examples[11, 12]. The most useful for this problem seems to be the method of Dandl et al.[12], since their method is superior in performance and can handle numerical and categorical features. The method is based on simultaneously optimizing four different objective functions.

The four objective functions make that the counterfactual example is understandable and sensible to humans. The objectives are

1. The prediction of the counterfactual example should be close to the desired prediction.
2. The counterfactual example should be close to the original instance.
3. The counterfactual example may differ from the original instance only on a few features.
4. The counterfactual example should be realistic. So it should be close to another instance in the input data.

The four objective functions are optimized simultaneously with an optimisation method called Nondominated Sorting Genetic Algorithm II.

Advantages
Counterfactual explanations are easy to interpret. The resulting outcome of the explanatory method is close to how humans explain to each other. That makes the method understandable for people with all kinds of backgrounds.

The method only needs the models' prediction function. A background data set is not necessary to create counterfactual explanations.

Counterfactual explanations can also be created for non ML models. That makes them widely applicable and easy to compare between different (non) ML models.

Disadvantages
In most cases there will be many counterfactual explanations. There is no best explanation since every explanation is true, but some of the explanations won't be very useful. It is hard for people to handle an overload of 20 or more explanations. It becomes an even bigger problem when some explanations are contradicting.

It could also be the case that the counterfactual explanations are not useful for the explainee. For instance, if the explainee wants a counterfactual explanation of why his mortgage request is rejected by the bank. If the result is a counterfactual explanation that tells that the mortgage would not be rejected if he was a female, then the explanation is not very useful for the explainee. It is not a feature that he can change, such that his mortgage request is accepted.

2.4.5. Summary
A summary of the explanatory methods of Section 2.4 is given in Table 2.1. Remember that the framework is the real estate valuation problem. The explainees of our explanation are house appraisers. These have high knowledge of houses and which features drive the price. However, they are probably not familiar with mathematics. So difficult to interpret attributions or plots should be avoided. The goal is to assist the appraisers with appraising real estate by using an ML model and accordingly an explanation. The main application is real estate valuation for taxation purposes. Therefore, an explanation with a solid theoretic foundation is preferred. In that light, the Shapley values seem to be the most promising explanation method, since this method satisfies certain properties as symmetry, dummy, efficiency and linearity. Therefore the Shapley value method is the topic of further investigation. Also, the disadvantage of needing a background data set is not a problem, since there is a big data set available of over 1 million house transactions. The disadvantage of assuming independence in the approximation method will be part of the research.

method	**scope**	**applicability**	**result**	**advantages**	**disadvantages**
LIME	local	model-agnostic	interpretable model	- easy interpretable - feature engineering - accuracy of explanation	- a lot of hyperparameter tuning - unstable
SR	local	model-agnostic	conditional statement	- interpretable for laymen - coverage value	- computationally heavy - possible useless explanation - assumes independence - a lot of hyperparameter tuning
SHAP	local	model-agnostic	feature summary statistic	- solid theory - select background data	- computationally heavy - needs background data set - implementation assumes independence
CFI	local	model-agnostic	conditional statement	- easy interpretable	- many (contradicting) explanations - possible useless explanation

Table 2.1: A summary of the four explanatory methods described in Section 2.4

3

Shapley Value Definition

In Section 2.4 already a brief introduction of the Shapley method is given. Since it is the subject of the rest of the research, it is necessary to dive deeper into the method. This chapter will give the exact definition (in fact there are three) and some properties of the Shapley values.

The Shapley method is a way to attribute the value of a single player i to the value of the coalition S, where S is a subset of all players N. The value or value function of a coalition S is defined as $v(S)$. Then the Shapley value of player i compared to the whole group of players N is defined as

$$\varphi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left(v(S \cup \{i\}) - v(S)\right) \tag{3.1}$$

The Shapley value can be interpreted as the contribution of player i to the value averaged of all possible coalitions S. The game-theoretic Shapley values can be translated to feature attribution of a model cleverly. Observe a model prediction as a game in which the players are features and the outcome is the model prediction. For instance, an instance $x = (x_1, ..., x_{|N|})^T$, model f and prediction $f(x) = y$. Of course, the coalition of all features have as value the prediction of the model, so

$$v(N) = f(x) = y$$

Also, when none of the features is in the coalition, the empty coalition, the value should be 0,

$$v(\emptyset) = 0$$

But for all other possible coalitions it is not clear what the value function should be. There are different options for how to define the value function. A different value function gives different Shapley values.
In Section 3.1 the most important properties of the Shapley values will be enumerated. These properties hold for all value functions. In Sections 3.2, 3.3 and 3.4 there will follow three different definitions of the value function, which give different Shapley values. The question is which of these definitions is most suitable for the real estate valuation problem.

3.1. Properties

Regardless of the definition of the value function, the Shapley values have certain desirable properties. Proofs of these properties are straightforward and can be found in many different papers and books [13, 14]. The most important properties are

1. **Symmetry** If f is symmetric in i and j and if the input data has $x_i = x_j$, then $\varphi_i = \varphi_j$

2. **Dummy** If i is a dummy variable, then the Shapley value is zero. A variable is a dummy variable if for any x_i, x_i' and for all $x_{N\setminus\{i\}}$ it holds that $f(x_i; x_{N\setminus\{i\}}) = f(x_i'; x_{N\setminus\{i\}})$. In other words, a dummy variable is a variable that does not add to the prediction, no matter what the values of the other features are.

3. **Efficiency** The sum of the Shapley values over all features is equal to the difference between the prediction and the global average, $\sum_{i=1}^{|N|} \varphi_i(x) = f(x) - \mathbb{E}\left[f(X)\right]$

4. **Additivity** If f and g are both prediction functions, then the Shapley values of the sum of the predictions is equal to the sum of the Shapley values of both prediction functions.

3.2. Conditional Expectation Shapley (C)

First, one could choose the conditional expectation as the value function. Then one defines the explicand x and the distribution of the input data $\mathscr{D}$. Also define the restriction of x to a set of features S as $x_S = \{x_i : i \in S\}$. Then the conditional expectation value function is defined as

$$v_x^C(S) = \mathbb{E}_{X' \sim \mathscr{D}}\left[f(X') | x'_S = x_S\right] - \mathbb{E}_{X' \sim \mathscr{D}}\left[f(X')\right].$$

This is the conditional expectation of the prediction function conditioned on the feature values of x that are in S. This is how Lundberg et. al. [10] defined the value function when they invented the SHAP method. But because of their approximation method, they essentially calculated the M Shapley values, which are defined in a coming paragraph. If the prediction function is a linear regression model, then the value function of the C Shapley values get the explicit form

$$\begin{aligned} v_x(S) &= \mathbb{E}_{X' \sim \mathscr{D}}\left[f(X') | X'_S = x_S\right] \\ &= \sum_{i \in S} \beta_i x_i + \sum_{i \in N \setminus S} \beta_i \mathbb{E}_{X' \sim \mathscr{D}}\left[X'_i | X'_S = x_S\right]. \end{aligned} \tag{3.2}$$

Combining the value function of the Shapley values with the definition of the Shapley values, defines the C Shapley values, namely

$$\varphi_i^C(x) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left(v_x^C(S \cup \{i\}) - v_x^C(S)\right).$$

It is difficult to interpret the C Shapley values directly. Note that one needs to know (or approximate) the underlying distribution $\mathscr{D}$ to calculate the C Shapley values. Two solutions for that are given in Sections 4.2 and 4.3.

3.3. Baseline Shapley (B)

On the other hand, one could choose the baseline value function. For that, a baseline instance r (reference instance) needs to be chosen. Then the value function becomes

$$v_{x,r}^B(S) = f\left(x_S; r_{N \setminus S}\right) - f(r).$$

The composite instance $\left(x_S; r_{N \setminus S}\right)$ has the feature values of x for features in set S and has feature values of r otherwise. This can be interpreted as the explicand x against the baseline r. Note that it is not necessary to know the underlying distribution $\mathscr{D}$ to calculate the B Shapley value, because there is no expectation in the value function. The definition of the B Shapley value becomes

$$\varphi_i^B(x, r) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left(v_{x,r}^B(S \cup \{i\}) - v_{x,r}^B(S)\right).$$

The B Shapley values do not give an overall attribution of the features to the prediction. Instead, the B Shapley values explain how much a feature contributes to the difference between the prediction of x and the prediction of b. This can be useful if you want to compare the prediction of specific cases. For instance, if two adjacent houses have a significantly different predicted house price, then the B Shapley values can give an indication of where the difference comes from. To get an overall attribution from the B Shapley values, one can average the baseline value function over different baselines, which give the M Shapley values.

3.4. Marginal Expectation Shapley (M)

The B Shapley values can be combined to an overall Shapley value, the M Shapley values. The value function is the expectation of the baseline value function over every possible baseline,

$$\begin{aligned} v_x^M(S) &= \mathbb{E}_{R \sim \mathscr{D}}\left[v_{x,R}^B(S)\right] \\ &= \mathbb{E}_{R \sim \mathscr{D}}\left[f(x_S; R_{N \setminus S} - f(R)\right] \\ &= \mathbb{E}_{R \sim \mathscr{D}}\left[f(x_S; R_{N \setminus S})\right] - \mathbb{E}_{R \sim \mathscr{D}}\left[f(R)\right]. \end{aligned}$$

It is called M Shapley values because it essentially calculates the marginal expectation. Note that is different from the conditional expectation value function in the sense that there is not conditioning on the features in S, it is just the expectation over the underlying distribution $\mathscr{D}$. The definition of the M Shapley values is

$$\varphi_i^M(x) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left(v_x^M(S \cup \{i\}) - v_x^M(S)\right).$$

This can be interpreted as explicand x against the average instance of $\mathscr{D}$, where the average instance is the instance with every feature value the average of that feature. Note that one needs to know (or approximate) the underlying distribution $\mathscr{D}$ to calculate the M Shapley values. Two solutions for that are given in Sections 4.2 and 4.3.

In some literature, the M Shapley value is referred to as the random baseline Shapley value [14]. A short proof shows that this is equal to the expected B Shapley value [15]

$$\begin{aligned}
\mathbb{E}_{R\sim\mathscr{D}}\left[\varphi_i^B(x,R)\right] &= \mathbb{E}_{R\sim\mathscr{D}}\left[\sum_{S\subseteq N\setminus\{i\}} \frac{|S|!\,(|N|-|S|-1)!}{|N|!}\left(v^B_{x,R}(S\cup\{i\}) - v^B_{x,R}(S)\right)\right] \\
&= \sum_{S\subseteq N\setminus\{i\}} \frac{|S|!\,(|N|-|S|-1)!}{|N|!}\left(\mathbb{E}_{R\sim\mathscr{D}}\left[v^B_{x,R}(S\cup\{i\})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[v^B_{x,R}(S)\right]\right) \\
&= \sum_{S\subseteq N\setminus\{i\}} \frac{|S|!\,(|N|-|S|-1)!}{|N|!}\left(v^M_x(S\cup\{i\}) - v^M_x(S\cup\{i\}\right) \\
&= \varphi_i^M(x).
\end{aligned}$$

A summary of the possible Shapley values is in Table 3.1.

Shapley value	**value function**	$v(S)$
B Shapley	function evaluation	$f(x_S; r_{N\setminus S}) - f(r)$
M Shapley	marginal expectation	$\mathbb{E}_{R\sim\mathscr{D}}\left[f(x_S; R_{N\setminus S})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[f(R)\right]$
C Shapley	conditional expectation	$\mathbb{E}_{X'\sim\mathscr{D}}\left[f(X')\vert x'_S = x_S\right] - \mathbb{E}_{X'\sim\mathscr{D}}\left[f(X')\right]$

Table 3.1: A summary of the three different Shapley value definitions.

4

Shapley Value Approximation

The Shapley values are elegant in theory, but in practice, there are two main problems to overcome. First, the definition of the Shapley values in Equation 3.1 is exponential in the number of features $d = |N|$. For each Shapley value, one needs to calculate a sum with 2^d terms, which is computationally heavy for high dimensions. In that case, an approximation method is necessary. Luckily, the kernel approximation method is invented by Lundberg et al. [10] which will be explained in paragraph 4.1.

Furthermore, the C and M Shapley values both take the expectation of the prediction function f over the data distribution $\mathscr{D}$. There are two problems with calculating this expression. First, the value function f is often not analytically known, for instance when a complex ML model is used. Nevertheless, there is the possibility to gain knowledge of f by evaluating it for different data points. Second, the data distribution $\mathscr{D}$ is unknown. A solution is to approximate $\mathscr{D}$. Two ways to approximate $\mathscr{D}$ are described in respectively paragraphs 4.2 and 4.3. When approximating the C Shapley values and the features are dependent, then it is important to take this into account. Chapter 5 describe the experiments that try to find out what is the best method to take the dependence into account.

4.1. Kernel Approximation

The definition of the Shapley values implies

$$f(x) = \phi_0 + \sum_{j=1}^{d} \phi_j.$$

where $\phi_0 = \mathbb{E}_{X\sim\mathscr{D}}\big(f(X)\big)$ is the average prediction of the model over the data set. Then the Shapley values can also be stated as a weighted least squares problem [10] where you want to minimize

$$\sum_{S\subseteq N} \left(v(S) - \left(\phi_0 + \sum_{j=1}^{d} \phi_j \right) \right)^2 k(d,S) \tag{4.1}$$

and

$$k(d,S) = \frac{d-1}{\binom{d}{|S|}|S|(d-|S|)}.$$

Equation 4.1 can be rewritten in matrix form [1]. Let ϕ be the vector of Shapley values. Let Z be the $2^d \times d+1$ coalition matrix, where every row is a different coalition and the cells with value 1 are members of the coalition and cells with value 0 are not. Note that the first column contains only value 1 since ϕ_0 is always included. Also, let v be the value function vector where every row of Z has value function $v(S)$. Finally, W is the $2^d \times 2^d$ diagonal matrix with values $k(d,S)$. Then equation 4.1 becomes

$$\sum_{S\subseteq N} \left(v(S) - \left(\phi_0 + \sum_{j=1}^{d} \phi_j \right) \right)^2 k(d,S) = (v - Z\phi)^T W (v - Z\phi).$$

The solution for ϕ of this problem is

$$\phi = (Z^T W Z)^{-1} Z^T W v.$$

For a problem with many features d, this becomes a huge matrix product. Therefore, Lundberg et al. [10] came with the idea to sample coalitions S from a probability distribution proportional to the kernel weights $k(d,S)$. The subset $\mathcal{M}$ of M is the sampled subset of coalitions S and $|\mathcal{M}| \ll |M|$. Then the solution of the Shapley values becomes

$$\phi = (Z_{\mathcal{M}}^T W_{\mathcal{M}} Z_{\mathcal{M}})^{-1} Z_{\mathcal{M}}^T W_{\mathcal{M}} v_{\mathcal{M}} = R_{\mathcal{M}} v_{\mathcal{M}}.$$

The advantage of this kernel method is that the matrix product is smaller than the original matrix product because only a subset of all possible coalitions is included. Another advantage is that $R_{\mathcal{M}}$ only has to be determined once.

4.2. Monte Carlo Approximation

The other problem is that the expectation cannot be determined since the data generating distribution $\mathscr{D}$ is unknown. The solution is to use a Monte Carlo integral to approximate the expectation, namely for M Shapley values [1]

$$\mathbb{E}_{R\sim\mathscr{D}}\left[f(x_S; R_{N\setminus S})\right] \approx \sum_{k=1}^{K} f\left(x_S; R_{N\setminus S}^k\right) \quad \text{with } R^k \sim \mathscr{D}'_X,$$

and for C Shapley values

$$\mathbb{E}_{X'\sim\mathscr{D}}\left[f(X')|x'_S = x_S\right] \approx \sum_{k=1}^{K} f\left(x_S; R_{N\setminus S}^k\right) \quad \text{with } R_{N\setminus S}^k \sim \mathscr{D}'_{X|x_S}.$$

where $\mathscr{D}'_X$ is an approximation of the distribution $\mathscr{D}_X$ and $\mathscr{D}'_{X|x_S}$ is an approximation of the conditional distribution $\mathscr{D}_{X|x_s}$. Now the next problem arrives, namely how to approximate the distributions to sample from. It turns out that there are several possibilities.

The **Input distribution** $\mathscr{D}^{Inp}$, where you use a background data set to sample from. Sampling from $\mathscr{D}^{Inp}$ is drawing samples independently from the background data set. A drawback of this method is that a big background data set is needed if one is using many features. In theory, it should also be possible to sample conditional from $\mathscr{D}^{Inp}$. However, in practice, this is often very hard, especially when continuous features are included. For instance, when conditioned on a continuous feature, there is (almost no) sample in the background data set with the specific continuous feature value.

The **Joint Marginal distribution** $\mathscr{D}^{JM}$ is the composition of univariate marginal distributions. Sampling for $\mathscr{D}^{JM}$ is sampling every feature independently from their marginal distribution. The correlation between features is completely ignored. It is possible to sample from the conditional distribution $\mathscr{D}^{JM}_{X|x_S}$ but is the same as sampling from the unconditioned $\mathscr{D}^{JM}$. Therefore, the approximated C and M Shapley values are exactly the same for $\mathscr{D}^{JM}$.

The **Multivariate Gaussian distribution** $\mathscr{D}^{MG}$ is a multivariate Gaussian distribution that is fitted on $\mathscr{D}$. This method is invented by Aas et al. [1]. The mean and variance of the fit will be respectively the sample mean and the sample covariance matrix of the background data set. It is easy to sample from a Gaussian distribution. Besides, it is possible to sample from the conditional multivariate Gaussian distribution, see for instance [1]. The conditional distribution $\mathscr{D}^{MG}_{X|x_S}$ is different from the unconditional distribution $\mathscr{D}^{MG}_X$, therefore the approximated C and M Shapley values will be different. Namely, this distribution also takes into account the correlation between features. A drawback of this distribution is that it does not handle discrete and categorical features well. Discrete and categorical features do not fit in a Gaussian distribution.

The **Gaussian copula distribution** $\mathscr{D}^{GC}$ is a fitted Gaussian copula with fitted marginals on the background data set. The mean and covariance of the fit will be respectively the zero vector and the sample spearman correlation matrix of the background data set. The marginals of the copula will be the empirical density function of the background data set. One can sample from the unconditional distribution and the conditional distribution $\mathscr{D}^{GC}_{X|x_S}$ [16]. This distribution takes into account the dependence between the marginals. Therefore, the approximated C and M Shapley values will be different for the Gaussian copula distribution. Aas et al. [1] invented this method for continuous variables. It turns out that it can be extended to discrete variables by fitting certain marginals. This is an advantage of $\mathscr{D}^{GC}$ compared to $\mathscr{D}^{MG}$. Therefore this distribution seems to be the most promising for the real estate valuation problem.

There are many more distributions possible. Any distribution that fits the data well, could be a good candidate. For instance, it is possible to fit another copula than the Gaussian copula. However, for this research, these are the distributions that are chosen to investigate. One reason is that for the Multivariate Gaussian and the Gaussian Copula it is easy to determine the conditional expectation. Then experiments can be done where the samples are drawn from these distributions and the analytic Shapley values can be determined, such that the approximated and the analytical Shapley values can be compared.

4.3. Likelihood Weighted Approximation

As described above, the Monte Carlo integration method depends on approximating $\mathscr{D}$ and sampling from that distribution. To determine the C Shapley values even the conditional distribution needs to be determined. Determining the conditional distribution is even more difficult than determining the unconditional distribution. The likelihood weighted method overcomes this problem by not depending on sampling from a conditional distribution but weighting the Monte Carlo sum to its likelihood. That looks as follows

$$\mathbb{E}_{X' \sim \mathscr{D}}\left[f(X') | x'_S = x_S\right] \approx \sum_{k=1}^{K} w\left(x_S; R_{N\setminus S}\right) f\left(x_S; R^k_{N\setminus S}\right) \quad \text{with } R^k \sim \mathscr{D}^{Inp},$$

where R^k are just sampled from the background set. The weights are defined as follows,

$$w\left(x_S; R^k_{N\setminus S}\right) = \frac{\mathbb{P}_{\mathscr{D}'}\left(x_S; R^k_{N\setminus S}\right)}{\sum_{i=1}^{K} \mathbb{P}_{\mathscr{D}'}\left(x_S; R^i_{N\setminus S}\right)}.$$

In this method, it is less important from which distribution you sample R^k, since the samples are weighted to their likelihood with respect to $\mathscr{D}'$. For example, it could be that $(x_S; R^k_{N\setminus S})$ is a very unlikely instance, then the weight $w(x_S; R^k_{N\setminus K})$ corrects for that and gives the instance low weight. Therefore it is less important from which distribution the samples are drawn, and it could just be $\mathscr{D}^{Inp}$. The method is inspired by Aas et al. [1], where they use a certain multivariate empirical distribution for $\mathscr{D}'$. It was not possible to use the empirical distribution in the Monte Carlo method, however, it could be used in the likelihood weighted method. The likelihood weighted method opens the door for many more distributions than the Monte Carlo method since you do not need to sample from that distribution. Any distribution that fits the data well could be used, even non-parametric distribution, as long as one can determine the likelihood for an instance.

5

Experiments

There are two questions that need to be answered by performing experiments. Remember that the main problem was which of the three Shapley definitions is most suitable for the real estate valuation problem? This lead to a sub-question which approximation method determines the C Shapley values most accurately? First, the experiments for the sub-question will be performed and after the experiments for the main question follows.

The experiments for the sub-question will be explained in Section 5.1. Then the step is made towards the real estate valuation problem, namely which method approximates the C Shapley value the most accurate for house transaction data, which will be explained in Section 5.2. Finally, the comparison between the three Shapley values can be made as described in Section 5.3.

5.1. Simulated Environment

To test which method approximates the C Shapley values most accurately, experiments are performed. The first experiments are done with simulated data, where the simulated data is from a predefined data generating distribution $\mathscr{D}$. Furthermore, the value function f is also predefined and known. In this synthetic environment, the conditional expectation can be calculated, hence the theoretic C Shapley values are known. Therefore the approximated C Shapley values can be compared to the theoretic C Shapley values and the best approximation method can be determined.

Simulated data

The data can be simulated from any distribution, however, there needs to be an analytical expression for the conditional expectation such that the theoretic Shapley values can be determined. Besides, it is useful to use a distribution with discrete and continuous variables combined, because the house transaction data contains mixed features too. One distribution that satisfies both conditions is a d-dimensional Gaussian copula with continuous and discrete marginals. The Gaussian copulas is defined as

$$C_{\Sigma}^{Gauss}(\mathbf{u}) = \Phi_{\Sigma}\left(F_1^{-1}(u_1), ..., F_d^{-1}(u_d)\right),$$

where $\Phi_{\Sigma}(\mathbf{x}) = N_d\left(\mathbf{0}, \Sigma_d(\rho)\right)$. The inverse cumulative distribution of the marginals are F_i^{-1}. The marginals are predetermined and such that the variables are continuous and discrete. The variance matrix $\Sigma_d(\rho)$ will be chosen as follows

$$\Sigma_d(\rho) = \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & & \rho \\ \vdots & & \ddots & \vdots \\ \rho & \rho & \cdots & 1 \end{bmatrix},$$

such that ρ is a metric for the correlation between the variables. By increasing ρ, the correlation between the features will increase. During the experiments, ρ will be varied between 0 and 0.95. For every experiment, the Shapley values of 200 samples are calculated. A background data set of 1,000 is used. Each setup will run 10 simulations and the results will be the average over the 10 runs.

Copulas are not unique when the marginals are discrete. Also, if F is a discrete CDF and u is a uniform random variable, then $F(F^{-1}(u)) \neq u$. Therefore, the discrete random variables X_j are transformed to continuous variable via the transformation [17]

$$Y_j = X_j + V_j - 1$$

where V_j are independent uniform (0,1) random variables. Y_j are now continuous variables and

$$\{Y_j \leq x_j\} = \{X_j \leq x_j\}.$$

A two-dimensional copula of this transformation is called a checkerboard copula.

Prediction Function

The prediction function is also predetermined. In this experiment, the prediction function is a simple linear model, namely

$$f(X) = x_1 + ... + x_d + \varepsilon,$$

where $\varepsilon \sim N(0, 0.01)$ is the noise term. A linear model will be fit on the background data, which looks like

$$\hat{f}(X) = \beta_1 x_1 + ... + \beta_d x_d + \hat{\varepsilon}.$$

This setup is comparable to the setup of Aas et. al. [1], such that the results can be compared.

Theoretic C Shapley value

Since the data generating distribution and the prediction function are known, the theoretic C Shapley values can be determined. For that, one needs to calculate the value function, so the conditional expectation. The conditional distribution of a Gaussian copula is known. For that, the steps of Kaarik et al. [16] can be used. Let $X_1, ..., X_d$ be the (discrete or continuous) random variable with cumulative density functions $F_1, ..., F_d$. Suppose without loss of generality that X_{k+1} is the unobserved variable and that $X_i's$ with $i \in \{0, ..., k\}$ are the observed variables.

1. Use the normalizing transformation $Z_j = \Phi^{-1}\left(F_j\left(X_j\right)\right)$ for all $j = 1, ..., k+1$, such that $Z_1, ..., Z_{k+1} \sim (\mathbf{0}, \Sigma_{k+1})$.

2. Define

$$\Sigma_{k+1} = \begin{bmatrix} \Sigma_k & \sigma_k \\ \sigma_k^T & 1 \end{bmatrix}$$

   and let $\sigma_k = 1 - \sigma_k^T \Sigma_k^{-1} \sigma_k$ The conditional probability density function of Z_{k+1} given $Z_1, ...Z_k$ is

$$f_{Z_{k+1}|Z_1,...Z_k}\left(z_{k+1}|z_1, ..., z_k; \Sigma_d\right) = \frac{1}{\sigma_{k+1}} \varphi\left(\frac{z_k - \sigma_k^T R_k^{-1} z_k}{\sigma_{k+1}}\right)$$

   where φ is the pdf of a standard normal variable. This means that $Z_{k+1}|Z_1, ...Z_k \sim N\left(\sigma_k^T R_k^{-1} z_k, 1 - \sigma_k^T R_k^{-1} \sigma_k\right)$, such that

$$\mathbb{E}\left[Z_{k+1}|Z_1, ..., Z_k\right] = \sigma_k^T R_k^{-1} z_k.$$

3. Transform back the conditional expectation of Z_{k+1} via the inverse transformation $\mathbb{E}\left[X_{k+1}|X_1, ...X_k\right] = F_{k+1}^{-1}\left(\Phi(\mathbb{E}\left[Z_{k+1}|Z_1, ..., Z_k\right]\right.$

The theoretic value function for instance x and arbitrary coalition S is stated in Equation 3.2, where the conditional expectation can be calculated via the above steps.

Approximation method

Six different approximation methods have been chosen to be used for the experiments, which are

1. **mc copula exact** is a Monte Carlo approximation method where the samples are from a conditional copula. The marginals are determined by the empirical cumulative density function of the background data. The copula is determined by setting $\mu = \mathbf{0}$ and approximate Σ with the spearman correlation matrix. The exact method is used to calculate the Shapley values sum, not the kernel approximation method. This is possible for small dimensions since it takes 2^d calculations.
2. **mc copula kernel** is the same as mc copula exact except that Shapley values sum is approximated with the kernel method.
3. **mc gaussian** is a Monte Carlo method where the samples are drawn from a multivariate Gaussian. The MG distribution is fitted on the training data by approximating μ by the sample mean and Σ by the sample covariance. Furthermore, the sum is approximated with the kernel method. This is the same method as proposed by Aas et. al. [1].

4. **mc independent** is a Monte Carlo method where the variables in S and $N \setminus S$ are drawn independently. The sum is approximated with the kernel method. This is the original kernel method from Lundberg et. al. [10] and essentially calculated the M Shapley values.
5. **lw gaussian** is a likelihood weighted method where the likelihood is a fitted multivariate Gaussian distribution with the sample mean and sample covariance. The sum is approximated with the kernel method.
6. **lw gaussian mixture** is also a likelihood weighted method but instead, the fitted distribution is a Gaussian Mixture with 2 components. The sum is approximated with the kernel method.

All approximation methods are implemented in `Python`. The implementation makes use of the `shap` package. The package is extended if necessary for the approximation method.

Comparison metric

To compare the approximated and theoretic Shapley values, the mean absolute percentage error is used

$$\textbf{MAE} = \frac{1}{md} \sum_{i=1}^{m} \sum_{j=1}^{d} \left| \phi_j(i)^{theoretic} - \phi_j(i)^{approximate} \right|$$

where d is the number of features and m is the number of samples used for the experiment. The errors for the largest Shapley values are the most interesting since the largest Shapley values will be the most explaining factors. Therefore this metric is chosen instead of, for instance, the mean absolute percentage error, which puts relatively more emphasis on the small Shapley values.

5.2. Approximated Environment

If the experiments in the simulated environment are good enough, the step to are more realistic environment can be made. Suppose that a fit on the real-world data is perfect, so the real data generating distribution is known. Then one can check how the methods perform. For that, a copula is fit on real-world house transaction data. More information on the house transaction data set can be found in Appendix C. Then it is assumed that this fitted is the true copula and the same experiment as above can be done.

The choice of the copula is the Gaussian copula, which is fitted on a part of the house transaction data. The choice of the Gaussian copula is because it is the most standard copula and more importantly it is easy to determine the conditional expectation for a Gaussian copula. The included features are in table 5.1. The choice is such that any type of variables is included, but the number of variables is low enough to keep the computation time reasonable.

feature	**type**
construction year	discrete
transaction date	discrete
storage dummy (y/n)	discrete
garage dummy (y/n)	discrete
surface area (log)	continuous
house type	categorical
province	categorical

Table 5.1: Features of house transaction data that are included in the Gaussian copula fit.

There are 1,332,458 house transactions in the data set. The continuous surface area marginal is just fitted as the empirical density function, which is reasonable for this many samples. Next, the storage and garage dummy marginals are fitted as a Bernoulli variable with p equal to the sample mean. For simplicity, the construction year and transaction date marginals are fitted as discrete random variables within the observed range and with probabilities corresponding to the sample frequencies. Note that the transaction date is on a monthly level. It is not possible to fit a density function on the categorical house type variable, therefore the variable is ordinally encoded such that a discrete density can be fitted. This is not the most elegant way, but if the samples look plausible, it might not be too bad. The categorical feature could also be one-hot encoded, but that gives troubles with sampling. Namely, samples occur with 1's for different house types, while a house can only be of one type. Also, during the calculation of the Shapley values, permutations are made on an instance. If the house type is one-hot encoded, then the instance can be permuted such that is has a 0 for all house types, which is a troublesome situation, especially when working with the HTM model. The mean of the Gaussian copula is set as the zero vector and the correlation matrix is the approximated spearman correlation matrix.

The C Shapley values will be approximated for 200 samples from this fitted copula. A background data set of 1,000 samples from the copula will be used for the approximation method. This simulation will be repeated 10 times and the results will be the average over all 10 runs.

5.3. Real-World Environment

The above two experiments try to show that the C Shapley values can be calculated appropriately. Then a comparison can be made between the three different Shapley values. The B and M Shapley values have been implemented already in the `shap` package in python. The implementation is checked by comparing the theoretic Shapley values of the HTM model (see Appendix B) and the result of the implementation. If the theoretic Shapley values and the approximated Shapley values are close enough, the conclusion that the implementation is good can be made. Then an approximation method for the B, C and M Shapley values are working and can be applied to real-world problems.

The different Shapley Values are calculated for different models. An HTM model is fitted with a local linear trend and the time-invariant features as a linear regression model. Furthermore, two different ML models will be used. A Multi-layer Perceptron model, neural network method, and a Light Gradient Boosting Machine model, a tree-based method, will be used, such that the two most important types of ML models are included. The MLP model will have two hidden layers with 20 neurons. The LGBM model will have 1,000 trees. Both ML models will be used twice, once fitted on the real transaction prices and once on samples from the HTM model. The details of the sample generation can be found in Appendix A. The reason for this is that if the ML model is fitted on samples from HTM, then the underlying model is known. So one can check whether the ML picks up the real structure of the data, or that it finds different relations that are good for prediction but are not close to the underlying model. The Shapley values are a tool to check this because the Shapley values indicate how important the feature values are to come to the prediction. If a certain instance gets the same prediction from two different models, but the Shapley values of both models are very different, then the models are using different relations to come to a prediction.

model	**fitted on**	**abbreviation**
Hierarchical Trend Model	real transaction prices	HTM
Light Gradient Boosting Machine	samples from HTM	$\text{LGBM}_{\text{sample}}$
Light Gradient Boosting Machine	real transaction prices	LGBM_{y}
Multi-layer Perceptron	samples from HTM	$\text{MLP}_{\text{sample}}$
Multi-layer Perceptron	real transaction prices	MLP_{y}

Table 5.2: The models that are fitted and for which the different Shapley values are approximated.

The different Shapley values can be calculated for the five models, which will be done for 100 sampled transactions. For the M Shapley values a background data set of 1,000 samples from the original data set will be used. For the C Shapley values, the kernel copula method will be used where there are 1,000 samples drawn from the fitted Gaussian Copula to compute the C Shapley values. The B Shapley values will be calculated to a baseline instance. The baseline instance is chosen to be an average house transaction, such that it is not an outlier. The baseline instance that is chosen is in Table 5.3. The baseline house transaction is a house in Alkmaar, a midsize city in the Netherlands. The house lies in the province Noord-Holland, one of the provinces with the most transactions. All the other feature values of the house are also close to the sample mean.

feature	**value**
province	Noord-Holland
municipality	Alkmaar
house type	attached
transaction date	2018-02-01
construction year	1973
surface area	115 m^2
transaction price	€250.000

Table 5.3: The feature values of the baseline house transaction used to calculate the B Shapley values.

6

Results

This chapter will present the results of the experiments that are explained in Chapter 5. First, the experiments in the simulated environment will be handled in Section 6.1. Then the results of the fitted copula on the house transaction data will be treated in Section 6.2. Finally, the comparison between the B, C and M Shapley values will be made for the five different models in Section 6.3

6.1. Simulated Environment

The first experiment is with two standard normal marginals and one Bernoulli marginal with $p = 0.5$. Remember that the Shapley values are calculated for 200 samples with 1,000 background samples. The correlation ρ is varied between 0 and 0.95 with steps of size 0.1 and the last step of size 0.05. For every correlation, the experiment is repeated 10 times and the results are averaged. These results are in Figure 6.1. On the x-axis stands ρ. Remember that ρ is the value of the off-diagonal elements of the Gaussian Copula where the samples are drawn from. This is a measure of the correlation between the features. On the y-axis stand the mean absolute error of the approximated C Shapley values compared to the true C Shapley values. The different approximation methods have different colours. From the 10 runs also a 95%-confidence interval of the MAE can be determined, which is included in the plot.

The second experiment is with two standard normal marginals, one Bernoulli marginals with $p = 0.01$, one Bernoulli with marginal $p = 0.5$, one random discrete marginal $\{-1,0,1,2\}$ with probabilities (0.25; 0.25; 0.25; 0.25) and one random discrete marginal $\{-1,0,1,2\}$ with probabilities (0.49; 0.01; 0.01; 0.49). The choice is such that all different types of marginals are included and some extreme situations are included.

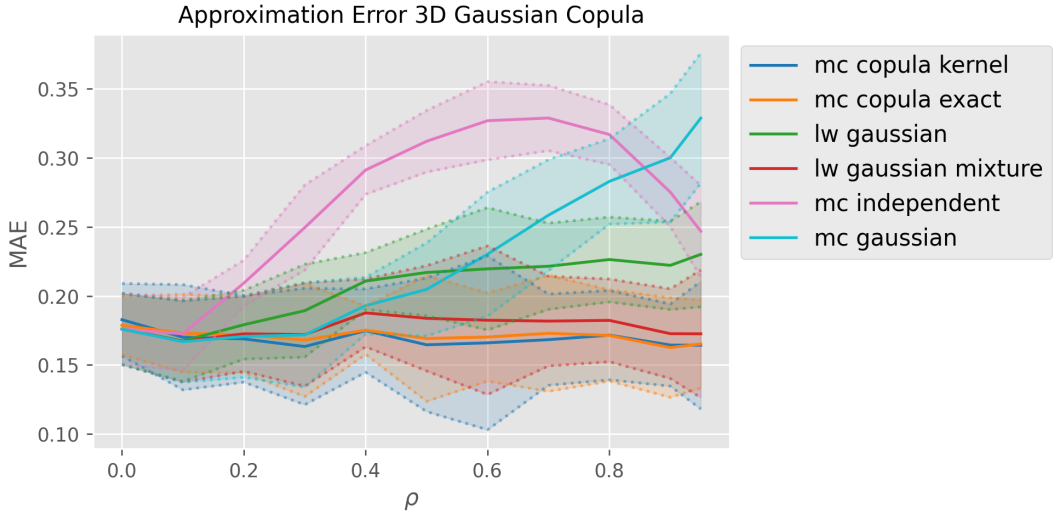


Figure 6.1: The results of different C Shapley value approximation methods. The marginals of this experiment are two standard normal marginals and one Bernoulli marginal with $p = 0.5$.

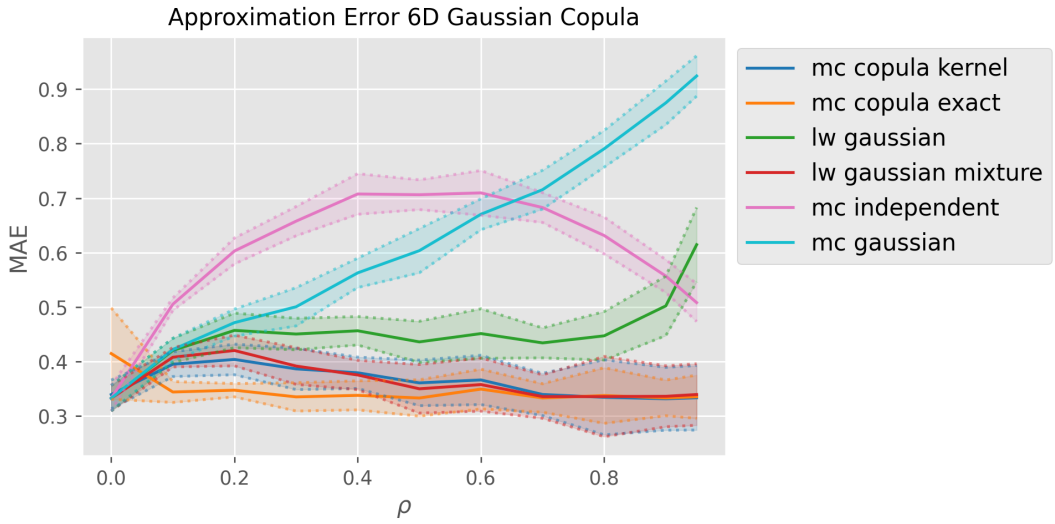


Figure 6.2: The results of different C Shapley value approximation methods. The marginals of this experiment are two standard normal marginals, one Bernoulli marginal with $p = 0.5$, one Bernoulli marginal with $p = 0.99$, one random discrete marginal $\{-1,0,1,2\}$ with probabilities (0.25; 0.25; 0.25; 0.25) and one random discrete marginal $\{-1,0,1,2\}$ with probabilities (0.49; 0.01; 0.01; 0.49).

Both experiments show that for higher correlation the copula methods are better than the other methods for the two data generating Gaussian copulas. The difference between the methods is bigger for the second experiment. Maybe this is due to the higher number of variables that is used. The second experiment shows that the mc independent method already performs significantly worse than the other methods from a correlation of 0.1. The copula methods are significantly better than the other methods from a correlation of 0.3, where the exact copula method is better than the kernel copula method as expected. Also, the likelihood weighted method with a Gaussian Mixture distribution performs almost as good as the copula methods. The results of these experiments suggest that the copula methods can be used for estimating the C Shapley values for the house transaction data. But before that, a copula is fit on (a part of) the house transaction data, which is the subject of Section 6.2.

6.2. Approximated Environment

A Gaussian copula is fitted on a part of the house transaction data. The data consists of 1,332,458 house transactions in the Netherlands, more details about the data set are in Appendix C. The features that are used for the fit are the surface area (logarithmic), the construction year, the transaction month, the house type, the province and whether the house contains a storage and/or garage. Note that the province and house type are ordinally encoded. That means that a certain ordering is applied to the categories. That is an undesired effect because there is not an ordering in house type or province. The downside is that the correlation does not represent the real correlation. But if the samples from the fitted copulas resemble the real examples, then this effect might not be too dominant.

Figure 6.3 shows the Spearman correlation matrix between the features. The diagonal is contains only ones. Further, there is a negative correlation between having a storage and having a garage. One could expect that since a storage and a garage are interchangeable for many people. Besides, having a garage is highly correlated with the surface area, which is expected since often only big houses have a garage. Furthermore, the house type has two significant negative correlations, namely with the garage and with the surface area. It depends on the ordering of the house type, but it is understandable that certain house types have more often a garage and have often a bigger surface area.

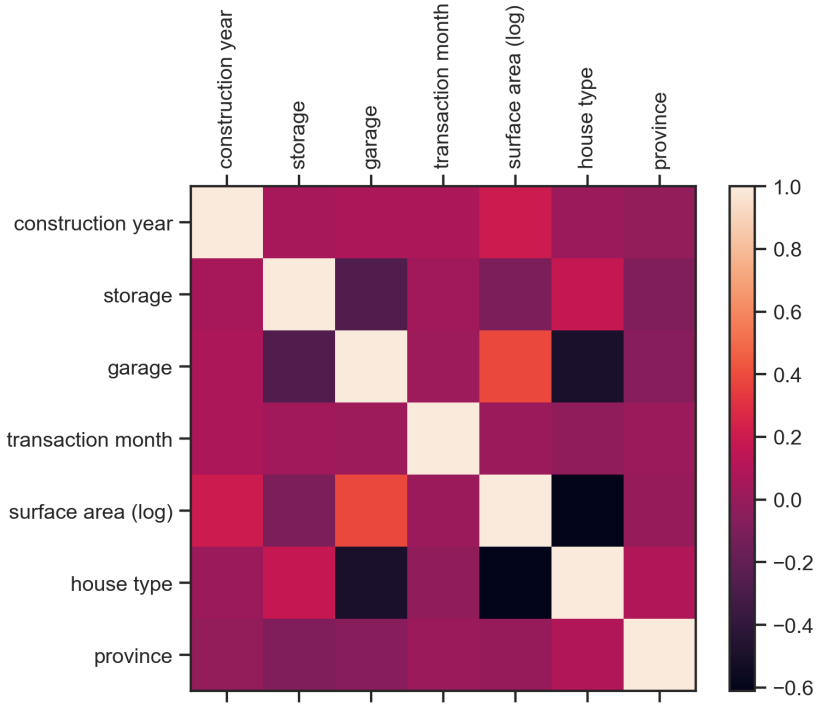


Figure 6.3: The spearman correlation matrix of the house transaction data used for fitting a Gaussian copula.

It is possible to draw samples from a Gaussian Copula. Samples drawn from the fitted copula are compared to the real data instance to check whether the fit of the copula is good. The histogram density plots of the real data and the samples are in Figure 6.4. The true data points are the blue bars. The orange bars are samples from the fitted Gaussian copula. The density histograms of the feature of the samples are very close to the density histograms of the real data points. The surface area, house type, storage and garage seems to fit almost perfectly. The construction year, transaction month and province show some differences, but this is probably due to limited sample size in combination with many different category types.

The marginal densities of the real data and sampled data can be alike, but that is not enough. There might be samples with an unlikely combination of feature values. To check that these kinds of problems are not occurring, a pair plot of densities is created. That is a grid where every combination of features is represented. This plot is shown in Figure 6.5. The lines represent the density of points, the higher the density of lines the higher the density of points. The choice of this kind of plot is made because a scatter plot would have a lot of overlapping points, such that it is not clear what is underneath certain points. The blue lines represent the true data points and the yellow lines the sampled data points. For every combination of features, the blue and yellow lines make up almost the same domain. The density of blue and yellow lines is almost everywhere comparable. That means that the true data points and the sampled data points have almost

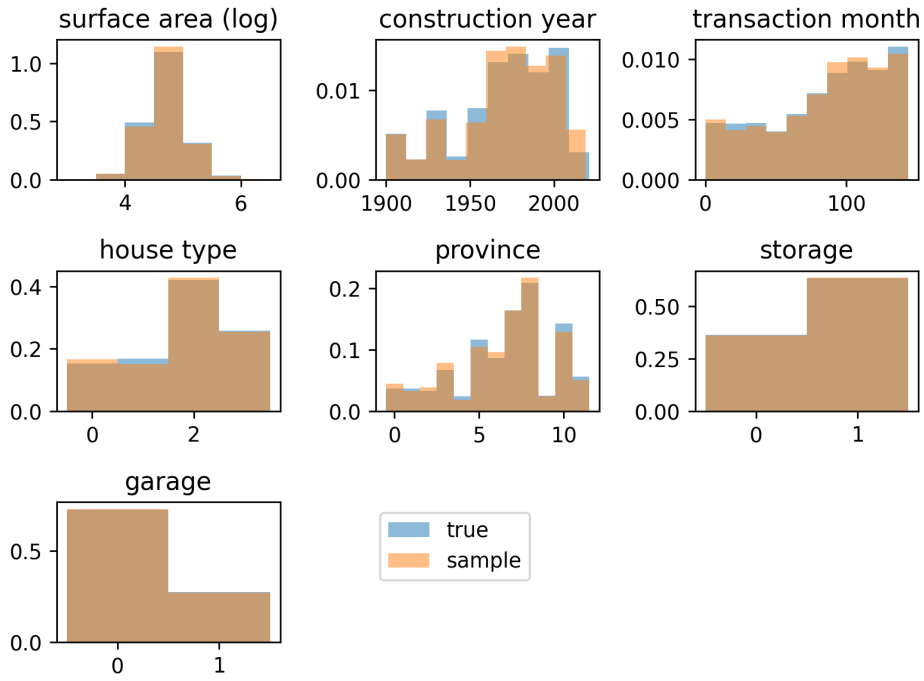


Figure 6.4: The density histograms of the house transaction features. The blue bars are the true data points and the orange bars are the data points sampled from the fitted Gaussian copula.

the same densities for each combination of features. Therefore, the fitted Gaussian copula is a good approximation of the house transaction data.

Now the assumption is that the fitted copula is the true copula. Then the same experiment can be done as before, namely, calculate the theoretic Shapley values and compare the approximation methods. To achieve that, also the prediction function is necessary, so a linear regression model is fitted on the logarithmic house prices. The coefficients of the linear regression are in Table 6.1a. The logarithmic has the biggest coefficient, which is expected because this is also the case with a fitted HTM model.
The results of the experiment are in Table 6.1b. The exact copula method is the most accurate. Then the kernel copula

feature	β**-coefficients**
construction year	$-3.73 \cdot 10^{-4}$
transaction date	$2.66 \cdot 10^{-3}$
storage dummy (y/n)	$2.67 \cdot 10^{-2}$
garage dummy (y/n)	$6.43 \cdot 10^{-2}$
logarithmic surface area	$8.55 \cdot 10^{-1}$
house type	$-6.69 \cdot 10^{-3}$
province	$1.94 \cdot 10^{-2}$
intercept	$8.73 \cdot 10^{-0}$

(a) The coefficient of the linear regression model fitted on the house transaction data.

method	**MAE**
mc copula exact	0.426 ± 0.050
mc copula kernel	1.658 ± 0.195
lw gaussian	2.243 ± 0.319
lw gaussian mixture	1.992 ± 0.292
mc independent	4.352 ± 0.327
mc gaussian	2.224 ± 0.229

(b) The results of the experiment where the copula is the fitted copula on the transaction data.

method is the most accurate, but the gap between the exact and kernel approximation is significant. The kernel method seems to be an accurate approximation method, which will turn out in Table 6.3. However, when using the copula method there is a significant difference between the exact Shapley values and the kernel approximated Shapley values.

The exact copula method is the most accurate but is often not possible because the computation time of the exact method scales exponentially with the number of features. Then one could use the copula kernel method because this method is the best of all other methods. However, it is noticeable that the copula kernel method is not much better than the lw gaussian mixture or mc gaussian method.

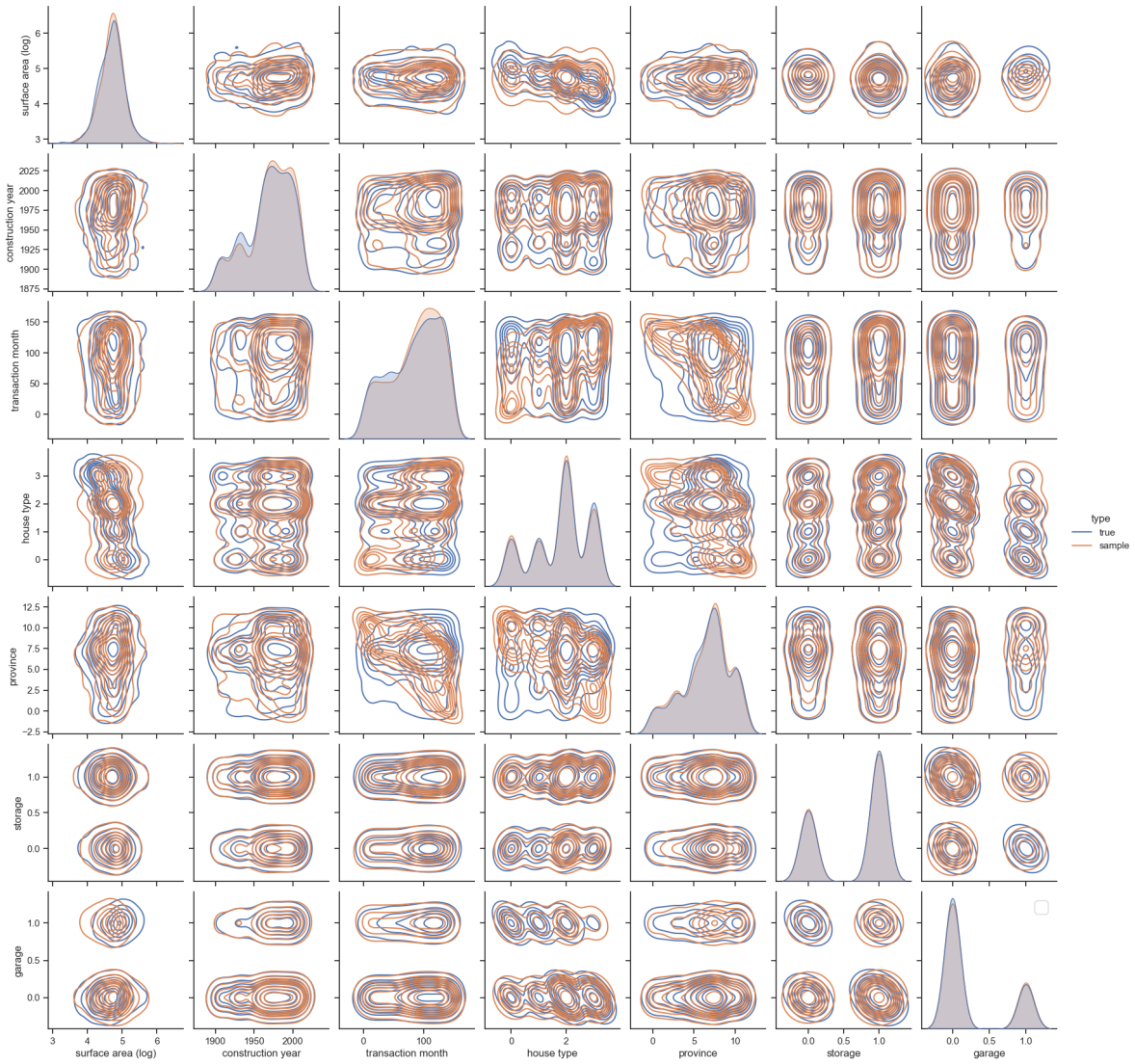


Figure 6.5: Density plots of real data points and sampled data points drawn from the fitted Gaussian copula.

6.3. Real-World Environment

The results from Section 6.2 show that the most practical and accurate method is the kernel copula method to approximated the C Shapley values. The same house transaction data set is used as in Section 6.2. 10 per cent of the data set is split off to create a test data set, so 90 per cent of the data is used to fit the models. There are five different models fitted, an HTM model, an LGBM model on samples, an LGBM model on real data, an MLP model on samples and an MLP on real data. The samples are created from the fitted HTM model, more details about this are in Appendix A The performance of the models is presented in Table 6.2.

For all models, the in-sample and out-sample performance is almost equal, which is an indication that none of the models is significantly over-fitted. Moreover, it is expected that the ML models fitted on the sample from the HTM model do not perform better than the HTM model itself, which is indeed the case. From the three models fitted on the real transaction price, the LGBM_y performs the best. The MLP_y performs worse than the HTM model. However, not much time has been spent on tuning the hyperparameters during this research, since it is not the goal to create the best possible ML model. Nevertheless, the performance of all models is reasonable and good enough to use for comparing the different Shapley values.

model	**MAPE in sample**	**MAPE out sample**
HTM_y	0.1624	0.1624
$\text{LGBM}_\text{sample}$	0.1821	0.1823
LGBM_y	0.1505	0.1512
MLP_sample	0.2067	0.2072
MLP_y	0.2141	0.2180

Table 6.2: The in and out sample performance of the fitted models. The $\text{LGBM}_\text{sample}$ and MLP_sample are fitted on samples from the fitted HTM model. All other models are fitted on the real transaction prices.

The B and M Shapley values can be theoretically determined for the HTM model (see Appendix B) but can also be approximated with the kernel method and sampling from $\mathscr{D}^{Inp}$. The approximated Shapley values can be compared to the theoretical Shapley values following from the HTM model. This comparison is made for all models and presented in Table 6.3.

The approximation of the Shapley values for the HTM model is good since this is the performance of the kernel approximation method versus the exact method. So the kernel approximation method is a good method to determine the B and M Shapley values.

The results for the $\text{LGBM}_\text{sample}$ and MLP_sample are also interesting because they are fitted on the samples from the HTM model. This means that the underlying model is known. So one expects that if the ML model performs well on the samples, then it has picked up the structure of the HTM model. Then the Shapley values of the ML model should be close to the Shapley values of the HTM model because the Shapley values give an indication of how important a feature is for the prediction. If the Shapley values differ between models then the prediction is coming from different key features. The results in Table 6.3 show that the B and M Shapley values of the $\text{LGBM}_\text{sample}$ and MLP_sample differ much from the theoretic Shapley values of HTM. So $\text{LGBM}_\text{sample}$ and MLP_sample are picking up other patterns than the HTM model, which is the true underlying model. That means that these models are good at making predictions, but not based on the true underlying structure.

The Shapley values from LGBM_y and MLP_y differ even more from HTM. But this can be explained since the underlying model is the real-world data. So these models might pick up different relations than the HTM model. From the results in Table 6.3, it is clear that it is indeed the case that the ML models pick up different patterns than the HTM model. That means that for the ML models different features are important to come to the prediction.

It takes a reasonable time to compute the Shapley values. The computation times for 100 Shapley values are presented in Table 6.4. The experiments are performed on an Intel i7-5600U 2.60GHz core processor. Another depending factor is which Shapley value is approximated. The B Shapley values take by far the least computation time since a specific instance is only compared to the baseline. The M Shapley values take significantly more computation time because there one instance is compared to 1,000 background instances. Finally, the C Shapley values take the most time, due to that here one instance is compared to 1,000 samples that need to be sampled from a Gaussian copula.

Moreover, the computation time depends on how fast the implementation of the prediction function is of a specific model. For instance, the prediction function of the HTM model is not fast, since there is not much time spent to optimize the code. The implementation uses the `shap` package in `Python`. Furthermore, the speed of the predicting with an

model	MAE B Shapley	MAE M Shapley	MAPE B Shapley	MAPE M Shapley
HTM_y	$7.4 \cdot 10^{-3}$	$1.2 \cdot 10^{-2}$	$4.9 \cdot 10^{-2}$	$3.0 \cdot 10^{-1}$
$LGBM_{sample}$	$1.4 \cdot 10^{-1}$	$1.2 \cdot 10^{-1}$	$4.6 \cdot 10^{-1}$	$9.4 \cdot 10^{-1}$
$LGBM_y$	$1.5 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$	$6.3 \cdot 10^{-1}$	$1.2 \cdot 10^{0}$
MLP_{sample}	$2.6 \cdot 10^{-1}$	$2.0 \cdot 10^{-1}$	$1.1 \cdot 10^{0}$	$3.2 \cdot 10^{0}$
MLP_y	$3.1 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$	$2.8 \cdot 10^{0}$	$7.8 \cdot 10^{0}$

Table 6.3: The error of the approximated B and M Shapley values compared to the theoretic B and M Shapley values of the HTM model as described in Appendix B.

ML model depends on which model you are using and the hyperparameters of the model.

model	Shapley type	computation time (s)
HTM_y	Baseline	19
	Marginal	6756
	Conditional	21595
$LGBM_{sample}$	Baseline	5
	Marginal	1028
	Conditional	4711
$LGBM_y$	Baseline	6
	Marginal	980
	Conditional	4864
MLP_{sample}	Baseline	5
	Marginal	374
	Conditional	1973
MLP_y	Baseline	6
	Marginal	409
	Conditional	2298

Table 6.4: The computation time to calculate 100 Shapley values with 1,000 background samples.

The M, C and B Shapley values for the 100 house transactions for the five different models are summarized in Figures 6.6 till 6.10. Note that all models are trained on the logarithmic transaction price, but for determining these Shapley values the exponent of the prediction is taken such that the normal transaction prices are returned. That results in more natural Shapley values that are represented in euros. The y-axis is labelled according to the features of the house. The x-axis represents the value of the Shapley values. The colour of the dots indicates the feature value according to the Shapley value. Pink means a high feature value and blue is a low feature value. Remember that the categorical features are ordinally encoded, so the colour has no meaning except that a different colour means a different category. Note that the multiplicative factor of the x-axis could differ per Shapley value. The vertical displacement of the dots within one feature is to illustrate that there are multiple instances with that specific Shapley value.

First, observe the results for the HTM model in Figure 6.6. All Shapley values seem to have the most variation in the surface area and surface extras features. Remarkable is that the M and B Shapley values of the surface area monotonic increase, where a small surface area has low Shapley values and a big surface area high Shapley values. That is a natural explanation since it is commonly known that houses with a bigger surface area are more expensive. This is in contrast to the C Shapley values, where the Shapley values have mixed surface area values.

Further, the C Shapley values of the surface area are all negative. Again this is likely caused by a correlation with other features, such as the surface extras features. A negative effect of the surface area is an unnatural explanation since the surface area has the highest coefficients in the HTM model. So an explanation that tells that the surface area lowers the predicted house price is unlikely to be accepted by the explainee. Only for this reason, the C Shapley values seem to be unsuitable in this case to apply to the HTM model.

Also, an interesting feature to note is the house type EGWO. This feature has small M and B Shapley values, but significant C Shapley values. The coefficients of the HTM model have low values for the house type EGWO feature. That means that the high C Shapley values are likely due to correlated other features, for instance, surface area. The same holds probably for the garage feature, which has a small coefficient and therefore small M and B Shapley values, nevertheless it has sometimes high C Shapley values.

Furthermore, if the M and B Shapley values of the surface extras features are observed, then something interesting can be noticed. The M Shapley values of the surface extras features are never 0, while the B Shapley values of the surface extras features are very often 0. The reason is probably that the surface extras features for most instances have value 0, so comparing two instances with both feature value 0 will give a B Shapley value of 0. On the other hand, the M Shapley values refer to the whole group, where on average the surface extras feature values are not 0, therefore a difference with the surface extras value of the specific instance occur and the M Shapley values are not 0.

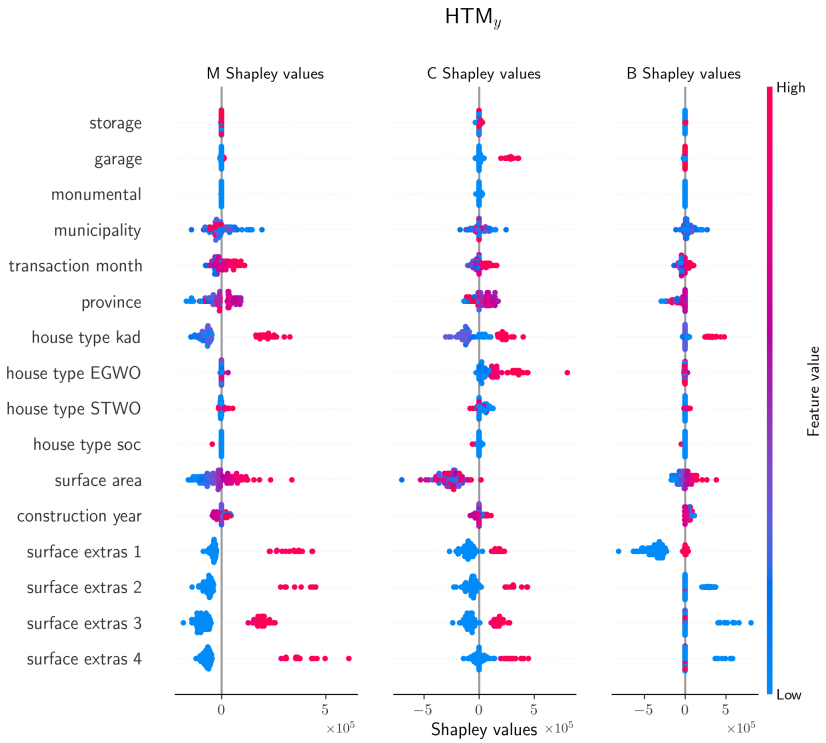


Figure 6.6: Summary plots of the Shapley values for the HTM model.

Now observe the summarized Shapley values for both LGBM models in Figures 6.7 and 6.8. Interesting is that the Shapley values of both LGBM models are similar. That probably means that both models are internally very similar, even though they are trained on different data.

Again the C Shapley values of the surface area are all negative, but the surface extras C Shapley values are all positive. Probably is the effect of the surface distributed over those features, because the surface area and surface extra feature are highly correlated. But the fact that this is happening, makes the C Shapley values not very useful in practice.

The surface extras B and M Shapley values have lower values than in the HTM model. On the other side, the surface area B and M Shapley values seem to be bigger than in the HTM model. So the LGBM models are putting more emphasis on the surface area than the surface extras features when making a prediction.
Finally, observe the results for the MLP models in Figures 6.9 and 6.10. Here the C Shapley values for the surface area and surface extras features are again respectively all negative and all positive. Also notice the multiplication factor of the C Shapley values, which makes that the C Shapley values are 100 to 1,000 times bigger than the M and B Shapley values, such that the C Shapley value of one feature could be bigger than the predicted house price. For those reasons, the C Shapley values are not useful as an explanation.

Another noticeable thing is that the B and M Shapley values seem to be very small for most instances with only a few outliers. It is at least clear that the MLP models are predicting completely different from the LGBM models. If the Shapley values are compared between both MLP models, then they are significantly different, for instance for the surface area or the surface extras 3 features.

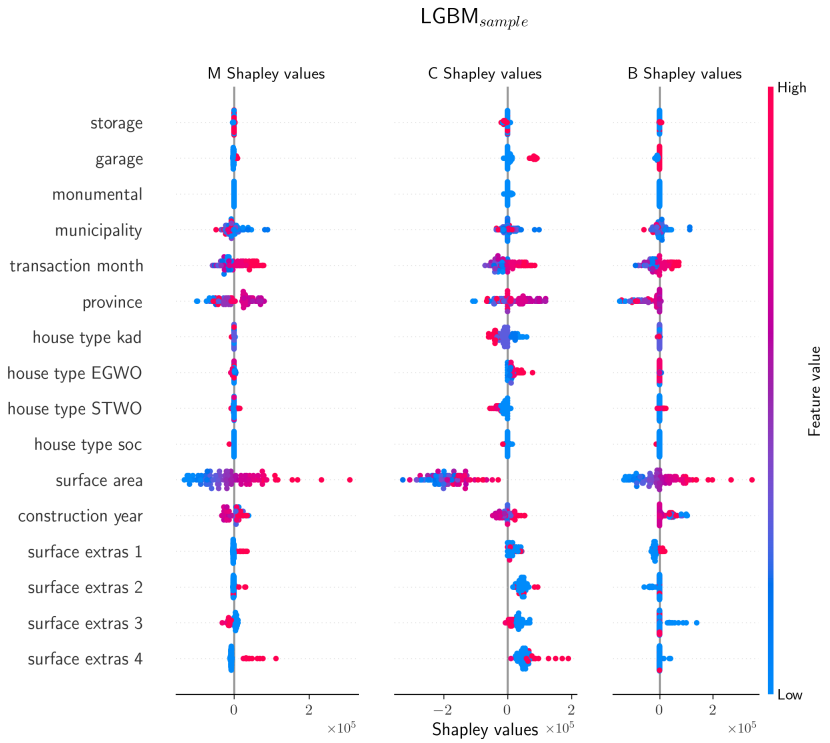


Figure 6.7: Summary plots of the Shapley values for the LGBM model fitted on samples of the HTM model.

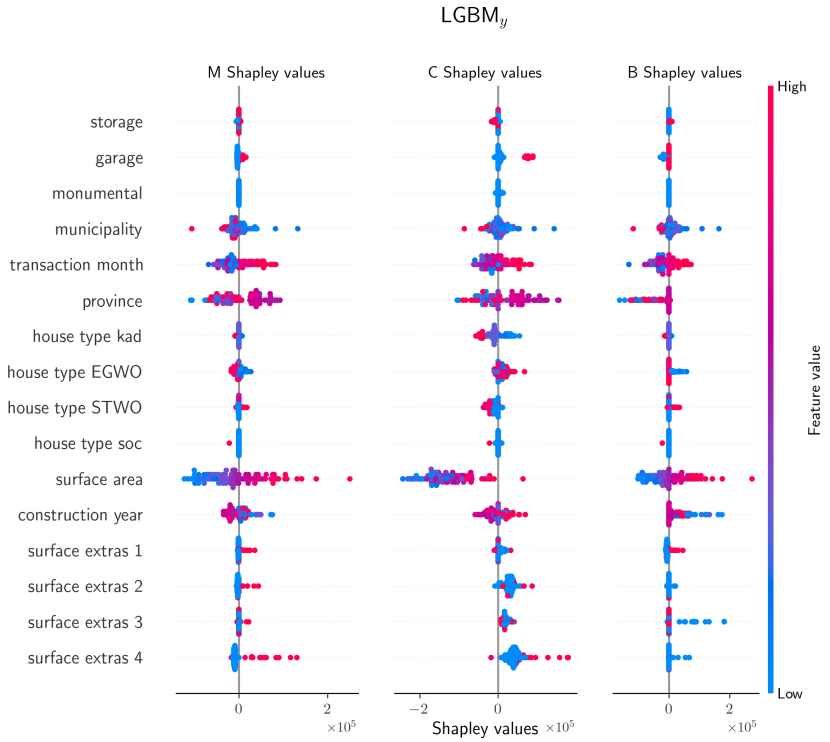


Figure 6.8: Summary plots of the Shapley values for the LGBM model fitted on the real transaction prices.

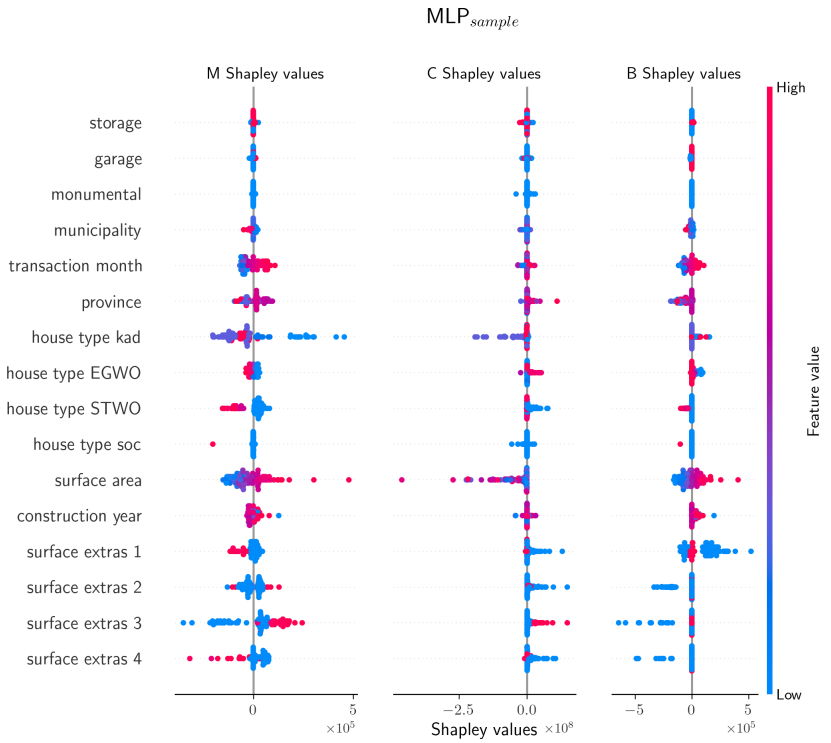


Figure 6.9: Summary plots of the Shapley values for the MLP model fitted on samples of the HTM model.

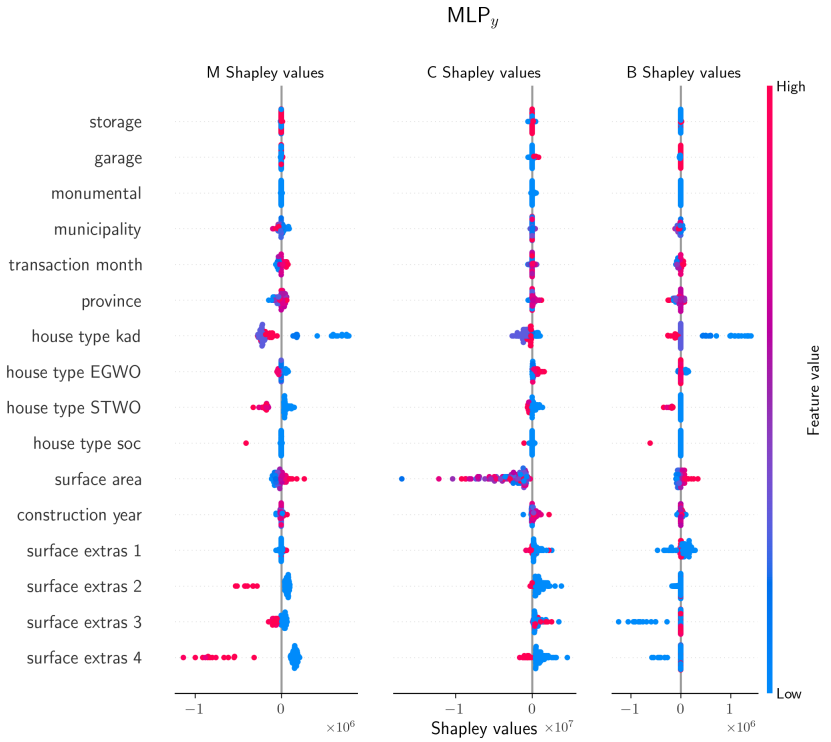


Figure 6.10: Summary plots of the Shapley values for the MLP model fitted on the real transaction prices.

7

Conclusion

Many real-world problems ask for a prediction together with an explanation. The real estate valuation problem is an example of that. When ML models are applied, an explanation can be extracted via multiple explanatory methods. The Shapley values method seems to be the most suitable method because this method has a strong theoretic base with desirable properties. However, there are still some problems with the Shapley values to overcome.

There are different definitions of the value function, which result in a different Shapley value. The most used value functions are the function evaluation (B Shapley values), the marginal expectation (M Shapley values) and the conditional expectation (C Shapley values). The difference between the M and C Shapley values is that for the C Shapley values the correlation between features is taken into account. Methods that assume independence to calculate the C Shapley values are basically calculating the M Shapley values. When the approximated C Shapley values of these methods are compared to the theoretical C Shapley values, then these methods make a significant error.

A solution is to assume that the data is not independent but of a certain shape, then fit a distribution of that shape on the data and (conditionally) sample from the fitted distribution. Previous research has shown these methods approximate the C Shapley values more accurately than the methods that assume independence. This research extended these results to distributions with continuous and discrete variables by making use of copulas, especially the Gaussian copula. The result is that fitting a copula and conditionally sample from this copula gives a better approximation of the true C Shapley values than methods that assume another distribution shape or methods that assume independence.

The M, C and B Shapley values are determined for a Hierarchical Trend Model model, a Light Gradient Boosting Machine and a Multi-layer Perceptron. From the comparison could be concluded that the C Shapley values are not useful as explanations in this case, because they return unnatural explanations.

Both M and B Shapley values had reasonable outcomes for the HTM and LGBM models. It seems that the M and B Shapley method can be applied to those models. The MLP model should be handled with more care, but it holds in general that these kinds of ML models are more difficult to build.

The M Shapley values describe the contribution of the feature value to the sample average. This could be sometimes difficult to interpret, especially for laymen. The B Shapley values describe how much the features contribute to the difference between two predictions. This is easier to interpret because it is a comparison between two instances. Moreover, the B Shapley values can be approximated faster and more accurately. So in practice, the B Shapley values seem to be the most suitable to extract an explanation from an ML model used for real estate valuation.

8

Discussion

This research is limited to the real estate valuation problem. A recommendation for further research is to apply the different definitions of Shapley values to other problems. For instance, apply the Shapley values to problems that are not similar to the real estate valuation problem, for instance, image recognition. It is interesting to check if the C Shapley values in other problems also give unnatural explanations.

Another limitation of this research is the choice of models. Here one statistical model (HTM) and two Machine Learning models (LGBM and MLP) are used. There are many more statistical and Machine Learning models. It would be interesting to apply the different Shapley values to those different models. That could give more insight into the Shapley values but might also give interesting results on how these models behave.

Furthermore, the C Shapley approximation methods can be investigated further. This research was limited to a specific choice of methods and fitted distributions. It might be that fitting other distributions gives different results. One could think of fitting non-parametric distributions or other copulas than the Gaussian one.

Another suggestion for a research direction is to try to define a Shapley value that is in between the B and M Shapley values. The B Shapley values compare one instance to another and the M Shapley values compare one instance to the whole group. It should be possible to define a Shapley value that lies in between, so it compares one instance to a subset or cluster. That could be for instance a set of comparable instances or instances from the same neighbourhood or cluster. There is a lot of freedom in the choice of the background data set when defining the Shapley values. Tuning that background data set might give interesting results.

A

Hierarchical Trend Model (HTM)

An econometrical model that is often used in practice for real estate valuation is the Hierarchical Trend Model (HTM) designed by Francke et al.[18–20]. This is also the model used by Ortec Finance for the real estate valuation used for taxation purposes. It is a state-space model designed to predict the value Y of a house based on its features X.

The regression of the model is done with data of transactions of houses done in the past. The logarithm of the transaction price is put into a state vector y_t. The length of y_t is n_t which depends on time due to the fact that the number of transactions is not the same in every time period. The corresponding house characteristics are in the X_t matrix of size $n_t \times m$.

In a qualitative manner, the HTM model can be summarized as

> The `log selling price` of house i in province j of type k in neighbourhood l at time t =
> the `general trend level` at time t +
> the `province` j `trend level` at time t +
> the `house type` k `trend level` at time t +
> the `neighbourhood` l `effect` +
> the effect of the `individual characteristics of house` i +
> an error term.

A quantitative description of the individual effects follows in the next sections.

Time-independent variables

Many characteristics of a house are time-independent, or at least for the time span of about a decade, for instance, the house size, lot size, year of construction, whether it has a garage or the distance to the nearest grocery store. These variables come together in a function $f(X,\beta)$ and can have many forms. The easiest choice is to define it as a linear model, e.g. $f(X,\beta) = X^T\beta$, where X are the time-independent variables. The choice of $f(X)$ is such that the computations for the HTM model become easy. In practice, $f(X)$ is a more sophisticated function where some assumptions can be put in the model. For instance, the assumption that the surface area of a house is decreasingly contributing to the house price, so the first square metres are weighing heavier than the latter square metres. An example is the function of Francke et. al. [20]. The linear regression model $f(X) = X^T\beta$ tries to replicate this effect by adding some extra time-independent features to $f(X)$. Some extra surface features are added as spline transformation of the original surface feature.

Time-dependent variables

The model distinguishes different types of trends that are time-variant. First, there is the general trend μ_t level that influences all houses prices. The general trend is specified as local linear trend model,

$$\begin{aligned}
\mu_{t+1} &= \mu_t + \kappa_t + \eta_t, \\
\kappa_{t+1} &= \kappa_t + \zeta_t, \\
\eta_t &\sim N(0, \sigma_\eta^2), \\
\zeta_t &\sim N(0, \sigma_\zeta^2).
\end{aligned}$$

Next, there are different cluster trends that only affect the prices in the specific cluster. The different clusters are province trends θ_t and house type trends λ_t. The length of θ_t is the number of distinct provinces n_d and for λ_t it is the number of

different house types n_h. Both cluster trends are modeled as a random walk,

$$\begin{aligned}\theta_{t+1} &= \theta_t + \omega_t,\\ \lambda_{t+1} &= \lambda_t + \varsigma_t,\\ \omega_t &\sim N(0, \sigma_\omega^2 I_{n_d}),\\ \varsigma_t &\sim N(0, \sigma_\varsigma^2 I_{n_h}).\end{aligned}$$

Furthermore, the model takes into consideration a neighbourhood effect ϕ. A neighbourhood is like a province but on an even smaller scale (there are multiple neighbourhoods in one province). The length of ϕ is the number of distinct neighbourhoods n_n. The neighbourhood effect is just a random variable drawn from a normal distribution, $\phi \sim N(0, \sigma_\phi^2 I_{n_n})$. Also, there is a noise term $\varepsilon_t \sim N(0, \sigma_\varepsilon^2 I_{n_t})$.

The time-invariant characteristics of the house are captured in the $f(X_t, \beta)$ term. To put it in a clear equation, there are selection matrices D defined to select the appropriate cluster. This means that the entries in the D matrices are either 0 or 1. Also, the D matrices change over time because we have different observations every time period. The size of the D matrices is the number of observations y_t times the number of different clusters, for instance, the number of provinces n_d.

All together this gives the following set of equations,

$$\begin{aligned}y_t &= \mathbb{1}_{n_t}\mu_t + D_{\theta,t}\theta_t + D_{\lambda,t}\lambda_t + D_{\phi,t}\phi + f(X_t,\beta) + \varepsilon_t,\\ \varepsilon_t &\sim N(0, \sigma_\varepsilon^2 I_{n_t}),\\ \mu_{t+1} &= \mu_t + \kappa_t + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2),\\ \kappa_{t+1} &= \kappa_t + \zeta_t, \quad \zeta_t \sim N(0, \sigma_\zeta^2),\\ \theta_{t+1} &= \theta_t + \omega_t, \quad \omega_t \sim N(0, \sigma_\omega^2 I_{n_d}),\\ \lambda_{t+1} &= \lambda_t + \varsigma_t, \quad \varsigma_t \sim N(0, \sigma_\varsigma^2 I_{n_h}),\\ \phi &\sim N(0, \sigma_\phi^2 I_{n_n}).\end{aligned} \tag{A.1}$$

Estimate HTM

To solve system A.1, it is convenient to put it in a state-space format.

$$\begin{aligned}y_t &= Z_t\alpha_t + f(X_t,\beta) + \varepsilon_t\\ \alpha_{t+1} &= T_t\alpha_t + \xi_t\end{aligned} \tag{A.2}$$

where we defined a few new variables

$$\begin{aligned}Z_t &= \begin{bmatrix}\mathbb{1}_{n_t} & 0 & D_{\theta,t} & D_{\lambda,t} & D_{\phi,t}\end{bmatrix},\\ \alpha_t^T &= \begin{bmatrix}\mu_t^T & \kappa_t^T & \theta_t^T & \lambda_t^T & \phi^T\end{bmatrix},\\ T_t &= \begin{bmatrix}1 & 1 & 0 & \cdots & 0\\ 0 & 1 & 0 & & 0\\ 0 & 0 & 1 & & 0\\ \vdots & & & \ddots & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}.\end{aligned}$$

Also we defined a random variable which is just the combination of single random variables,

$$\begin{aligned}\xi_t &\sim N(0, \Sigma),\\ \mathrm{Diag}(\Sigma) &= \begin{bmatrix}\sigma_\eta^2 & \sigma_\zeta^2 & \underbrace{\sigma_\omega^2 \dots \sigma_\omega^2}_{\times n_d} & \underbrace{\sigma_\varsigma^2 \dots \sigma_\varsigma^2}_{\times n_h} & \underbrace{0 \dots 0}_{\times n_n}\end{bmatrix},\end{aligned}$$

where the off-diagonal entries of Σ are 0. Solving equation A.2 is done via the following steps designed by Francke et al. [18]

1. Calculate the means per cluster $\bar{y}_1, ..., \bar{y}_T$ and the deviation from the means $\bar{y}_1, ..., \bar{y}_T$. So the length of $\bar{y}_t$ is the number of clusters and the length of $\bar{y}_t$ is the number of observations of that time.
2. Solve $f(X_t, \hat{\beta}) = \hat{y}_t$. If f is a linear regression model, then this could be done with Ordinary Least Squares.
3. Use $\hat{\beta}$ as priors to solve equation A.2 with a Kalman filter on the means $\bar{y}_t$.

Properties HTM

The prediction interval for a house i, σ_i, can be calculated via the formula

$$\sigma_i^2 = \left[Z_i X_i\right] \cdot V \cdot \left[Z_i X_i\right]^T + \sigma_\varepsilon^2,$$

where Z_i is the vector with the trend values, X_i is the vector with the time-invariant data, V is the matrix of covariances between all trends and coefficients and σ_ε^2 is the variance of the noise term.

The interpretation of the general trend level mu_t is very natural, it describes the average price development on the whole market. The cluster trend levels θ_t and λ_t describe the development of the house prices in a specific cluster. But the interpretation of the β coefficients can be more difficult and depends on the definition of $f(X, \beta)$. If f is a linear regression model, as will be in this report, then the interpretation is clear. If a time-invariant variable x_j, for instance, surface area, increases by 1%, then the house price will increase with a fraction β_j. If f is not a linear regression model, but of another type then the interpretation can be less obvious.

Sampling from HTM

For the experiments it was necessary to draw samples from the HTM model to train the ML models. The samples are drawn in the following way. The input data X is kept the same as the real-world data. The samples only apply to the transaction prices. In the samples, the real transaction prices y will be replaced by predictions from the HTM model $\hat{y}$. The samples are created as follows

$$\hat{y} = e^{f(X)+\varepsilon}$$

where X is the input data, f prediction function of the HTM model that predicts the logarithmic transaction prices and $\varepsilon \sim N(0, \sigma_\varepsilon)$ is a normal random variable with mean of 0 and variance equal to the noise term of the HTM model.

B

Analytic Shapley Value HTM

The Hierarchical Trend Model (HTM) model can be rewritten in the following way

$$f(x) = \mu(x_t) + \theta(x_\theta, x_t) + \lambda(x_\lambda, x_t) + \phi(x_\phi) + x_0\beta + \varepsilon$$

where x is a specific house transaction with the features x_t (time), x_θ (province), x_λ (house type), x_ϕ (neighbourhood) and x_0 (time-invariant features). Also, the functions of the global trend $\mu(x_t)$, the province trend $\theta(x_\lambda, x_t)$, the neighbourhood trend $\phi(x_\phi)$. The other features are modelled in a Linear Regression model with coefficients β.

If S is an arbitrary coalition without feature i. Then the B Shapley value of feature i depends on the difference in value function with and without feature i.

Analytic B Shapley Value

The exact B Shapley values can be determined for all features. For the time variable i_t, the value function becomes

$$\begin{aligned}
v^B_{x,r}(S\cup\{i_t\}) - v^B_{x,r}(S) =& f\left(x_{S\cup\{i_t\}}; r_{N\setminus(S\cup\{i_t\})}\right) - f\left(x_S; r_{N\setminus S}\right) \\
=& f\left(x_S; r_{N\setminus(S\cup\{i_t\})}; x_{i_t}\right) - f\left(x_S; r_{N\setminus(S\cup\{i_t\})}; r_{i_t}\right) \\
=& \mu(x_{i_t}) - \mu(r_{i_t}) + \begin{cases} \theta(x_{i_\theta}, x_{i_t}) - \theta(x_{i_\theta}, r_{i_t}) & \text{if } i_\theta \in S, \\ \theta(r_{i_\theta}, x_{i_t}) - \theta(r_{i_\theta}, r_{i_t}) & \text{if } i_\theta \notin S. \end{cases} + \begin{cases} \lambda(x_{i_\lambda}, x_{i_t}) - \lambda(x_{i_\lambda}, r_{i_t}) & \text{if } i_\lambda \in S, \\ \lambda(r_{i_\lambda}, x_{i_t}) - \lambda(r_{i_\lambda}, r_{i_t}) & \text{if } i_\lambda \notin S. \end{cases}
\end{aligned}$$

Note that summed over all possible coalitions S, then half of the times $i_\theta \in S$ and half of the times $i_\theta \notin S$ and the same holds for i_λ. This results in the B Shapley value of the time variable i_t,

$$\varphi^B_{i_t}(x,r) = \mu(x_{i_t}) - \mu(r_{i_t}) + \frac{1}{2}\left[\theta(x_{i_\theta}, x_{i_t}) - \theta(x_{i_\theta}, r_{i_t})\right] + \frac{1}{2}\left[\theta(r_{i_\theta}, x_{i_t}) - \theta(r_{i_\theta}, r_{i_t})\right] + \frac{1}{2}\left[\lambda(x_{i_\lambda}, x_{i_t}) - \lambda(x_{i_\lambda}, r_{i_t})\right] + \frac{1}{2}\left[\lambda(r_{i_\lambda}, x_{i_t}) - \lambda(r_{i_\lambda}, r_{i_t})\right].$$

For the province variable i_θ the value function becomes,

$$\begin{aligned}
v^B_{x,r}(S\cup\{i_\theta\}) - v^B_{x,r}(S) &= f\left(x_{S\cup\{i_\theta\}}; r_{N\setminus(S\cup\{i_\theta\})}\right) - f\left(x_S; r_{N\setminus S}\right) \\
&= f\left(x_S; r_{N\setminus(S\cup\{i_\theta\})}; x_{i_\theta}\right) - f\left(x_S; r_{N\setminus(S\cup\{i_\theta\})}; r_{i_\theta}\right) \\
&= \begin{cases} \theta(x_{i_\theta}, x_{i_t}) - \theta(r_{i_\theta}, x_{i_t}) & \text{if } i_t \in S, \\ \theta(x_{i_\theta}, r_{i_t}) - \theta(r_{i_\theta}, r_{i_t}) & \text{if } i_t \notin S. \end{cases}
\end{aligned}$$

This results in the B Shapley value of the province feature i_θ,

$$\varphi^B_{i_\theta}(x,r) = \frac{1}{2}\left[\theta(x_{i_\theta}, x_{i_t}) - \theta(r_{i_\theta}, x_{i_t})\right] + \frac{1}{2}\left[\theta(x_{i_\theta}, r_{i_t}) - \theta(r_{i_\theta}, r_{i_t})\right].$$

The same calculation leads to the B Shapley value of the house type feature i_λ,

$$\varphi^B_{i_\lambda}(x,r) = \frac{1}{2}\left[\lambda(x_{i_\lambda}, x_{i_t}) - \lambda(r_{i_\lambda}, x_{i_t})\right] + \frac{1}{2}\left[\lambda(x_{i_\lambda}, r_{i_t}) - \lambda(r_{i_\lambda}, r_{i_t})\right].$$

The value function of the B Shapley values of the neighbourhood variable i_ϕ,

$$\begin{aligned} v^B_{x,r}(S\cup\{i_\phi\}) - v^B_{x,r}(S) &= f\left(x_{S\cup\{i_\phi\}}; r_{N\setminus(S\cup\{i_\phi\})}\right) - f\left(x_S; r_{N\setminus S}\right) \\ &= f\left(x_S; r_{N\setminus(S\cup\{i_\phi\})}; x_{i_\phi}\right) - f\left(x_S; r_{N\setminus(S\cup\{i_\phi\})}; r_{i_\phi}\right) \\ &= \phi(x_{i_\phi}) - \phi(r_{i_\phi}). \end{aligned}$$

This translates directly to the B Shapley value of the neighbourhood variable,

$$\varphi^B_{i_\phi}(x,r) = \phi(x_{i_\phi}) - \phi(r_{i_\phi}).$$

Finally, the value function for any time-invariant variable i_0 becomes,

$$\begin{aligned} v^B_{x,r}(S\cup\{i_0\}) - v^B_{x,r}(S) &= f\left(x_{S\cup\{i_0\}}; r_{N\setminus(S\cup\{i_0\})}\right) - f\left(x_S; r_{N\setminus S}\right) \\ &= f\left(x_S; r_{N\setminus(S\cup\{i_0\})}; x_{i_0}\right) - f\left(x_S; r_{N\setminus(S\cup\{i_0\})}; r_{i_0}\right) \\ &= \beta_j\left(x_{i_0} - r_{i_0}\right). \end{aligned}$$

That gives the B Shapley value of any time-invariant feature,

$$\varphi^B_{i_0}(x,r) = \beta_{i_0}\left(x_{i_0} - r_{i_0}\right).$$

Analytic M Shapley Value

The same can be done for the M Shapley values. Suppose there is a background data set $\mathcal{N}$ of size $|\mathcal{N}| = N$ used to approximate the expectation. For the time variable i_t the value function becomes,

$$\begin{aligned} v^M_x(S\cup\{i_t\}) - v^M_x(S) =& \mathbb{E}_{R\sim\mathscr{D}}\left[v^B_{x,R}(S\cup\{i_\lambda\})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[v^B_{x,R}(S)\right] \\ =& \mathbb{E}_{R\sim\mathscr{D}}\left[v^B_{x,R}(S\cup\{i_\lambda\}) - v^B_{x,R}(S)\right] \\ =& \mathbb{E}_{R\sim\mathscr{D}}\left[\mu(x_{i_t}) - \mu(R_{i_t}) + \begin{cases}\theta(x_{i_\theta}, x_{i_t}) - \theta(x_{i_\theta}, R_{i_t}) & \text{if } i_\theta \in S,\\ \theta(R_{i_\theta}, x_{i_t}) - \theta(R_{i_\theta}, R_{i_t}) & \text{if } i_\theta \notin S.\end{cases} + \begin{cases}\lambda(x_{i_\lambda}, x_{i_t}) - \lambda(x_{i_\lambda}, R_{i_t}) & \text{if } i_\lambda \in S,\\ \lambda(R_{i_\lambda}, x_{i_t}) - \lambda(R_{i_\lambda}, R_{i_t}) & \text{if } i_\lambda \notin S.\end{cases}\right] \\ =& \mu(x_{i_t}) - \mathbb{E}_{R\sim\mathscr{D}}\left[\mu(R_{i_t})\right] + \begin{cases}\theta(x_{i_\theta}, x_{i_t}) - \mathbb{E}_{R\sim\mathscr{D}}\left[\theta(x_{i_\theta}, R_{i_t})\right] & \text{if } i_\theta \in S,\\ \mathbb{E}_{R\sim\mathscr{D}}\left[\theta(R_{i_\theta}, x_{i_t})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[\theta(R_{i_\theta}, R_{i_t})\right] & \text{if } i_\theta \notin S.\end{cases} \\ &+ \begin{cases}\lambda(x_{i_\lambda}, x_{i_t}) - \mathbb{E}_{R\sim\mathscr{D}}\left[\lambda(x_{i_\lambda}, R_{i_t})\right] & \text{if } i_\lambda \in S,\\ \mathbb{E}_{R\sim\mathscr{D}}\left[\lambda(R_{i_\lambda}, x_{i_t})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[\lambda(R_{i_\lambda}, R_{i_t})\right] & \text{if } i_\lambda \notin S.\end{cases} \end{aligned}$$

Then the theoretic M Shapley value of the time variable can be approximated as,

$$\begin{aligned} \varphi^M_{i_t}(x) =& \mu(x_{i_t}) - \frac{1}{N}\sum_{r\in\mathcal{N}}\mu(r_{i_t}) + \frac{1}{2}\theta(x_{i_\theta}, x_{i_t}) + \frac{1}{2}\lambda(x_{i_\lambda}, x_{i_t}) \\ &- \frac{1}{2N}\sum_{r\in\mathcal{N}}\left[\theta(x_{i_\theta}, r_{i_t}) - \theta(r_{i_\theta}, x_{i_t}) - \theta(r_{i_\theta}, r_{i_t}) - \lambda(x_{i_\lambda}, r_{i_t}) + \lambda(r_{i_\lambda}, x_{i_t}) - \lambda(r_{i_\lambda}, r_{i_t})\right]. \end{aligned}$$

The value function of the province variable i_θ becomes,

$$\begin{aligned} v^M_x(S\cup\{i_\theta\}) - v^M_x(S) &= \mathbb{E}_{R\sim\mathscr{D}}\left[v^B_{x,R}(S\cup\{i_\theta\})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[v^B_{x,R}(S)\right] \\ &= \mathbb{E}_{R\sim\mathscr{D}}\left[v^B_{x,R}(S\cup\{i_\theta\}) - v^B_{x,R}(S)\right] \\ &= \mathbb{E}_{R\sim\mathscr{D}}\left[\begin{cases}\theta(x_{i_\theta}, x_{i_t}) - \theta(R_{i_\theta}, x_{i_t}) & \text{if } i_t \in S,\\ \theta(x_{i_\theta}, R_{i_t}) - \theta(R_{i_\theta}, R_{i_t}) & \text{if } i_t \notin S.\end{cases}\right] \\ &= \begin{cases}\theta(x_{i_\theta}, x_{i_t}) - \mathbb{E}_{R\sim\mathscr{D}}\left[\theta(R_{i_\theta}, x_{i_t})\right] & \text{if } i_t \in S,\\ \mathbb{E}_{R\sim\mathscr{D}}\left[\theta(x_{i_\theta}, R_{i_t})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[\theta(R_{i_\theta}, R_{i_t})\right] & \text{if } i_t \notin S.\end{cases} \end{aligned}$$

Then the M Shapley value of the province variable becomes,

$$\varphi^M_{i_\theta}(x) \approx \frac{1}{2}\theta(x_{i_\theta}, x_{i_t}) - \frac{1}{2N}\sum_{r\in\mathcal{N}}\theta(r_{i_\theta}, x_{i_t}) + \frac{1}{2N}\sum_{r\in\mathcal{N}}\theta(x_{i_\theta}, r_{i_t}) - \frac{1}{2N}\sum_{r\in\mathcal{N}}\theta(r_{i_\theta}, r_{i_t}).$$

Via the same calculation the M Shapley value of the house type variable i_λ becomes,

$$\varphi^M_{i_\lambda}(x) \approx \frac{1}{2}\lambda(x_{i_\lambda}, x_{i_t}) - \frac{1}{2N}\sum_{r\in\mathcal{N}}\lambda(r_{i_\lambda}, x_{i_t}) + \frac{1}{2N}\sum_{r\in\mathcal{N}}\lambda(x_{i_\lambda}, r_{i_t}) - \frac{1}{2N}\sum_{r\in\mathcal{N}}\lambda(r_{i_\lambda}, r_{i_t}).$$

The value function of the neighbourhood variable i_ϕ is,

$$\begin{aligned}
v_x^M(S\cup\{i_\phi\}) - v_x^M(S) &= \mathbb{E}_{R\sim\mathscr{D}}\left[v_{x,R}^B(S\cup\{i_\phi\})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[v_{x,R}^B(S)\right] \\
&= \mathbb{E}_{R\sim\mathscr{D}}\left[v_{x,R}^B(S\cup\{i_\phi\}) - v_{x,R}^B(S)\right] \\
&= \mathbb{E}_{R\sim\mathscr{D}}\left[\phi(x_{i_\phi}) - \phi(R_{i_\phi})\right] \\
&= \phi(x_{i_\varphi}) - \mathbb{E}_{R\sim\mathscr{D}}\left[\phi(R_{i_\phi})\right].
\end{aligned}$$

Then the M Shapley value of the neighbourhood variable is,

$$\varphi_{i_\phi}^M(x) \approx \phi(x_{i_\phi}) - \frac{1}{N}\sum_{r\in\mathscr{N}}\phi(r_{i_\phi}).$$

The value function of any time-invariant variable i_0 becomes,

$$\begin{aligned}
v_x^M(S\cup\{i_0\}) - v_x^M(S) &= \mathbb{E}_{R\sim\mathscr{D}}\left[v_{x,R}^B(S\cup\{i_0\})\right] - \mathbb{E}_{R\sim\mathscr{D}}\left[v_{x,R}^B(S)\right] \\
&= \mathbb{E}_{R\sim\mathscr{D}}\left[v_{x,R}^B(S\cup\{i_0\}) - v_{x,R}^B(S)\right] \\
&= \mathbb{E}_{R\sim\mathscr{D}}\left[\beta\left(x_{i_0} - R_{i_0}\right)\right] \\
&= \beta_{i_0}\left(x_{i_0} - \mathbb{E}_{R\sim\mathscr{D}}\left[R_{i_0}\right]\right).
\end{aligned}$$

Then the M Shapley values for any time-invariant variable i_0 can be approximated

$$\varphi_{i_0}^M(x) \approx \beta_{i_0}\left(x_{i_0} - \frac{1}{N}\sum_{r\in\mathscr{N}} r_{i_0}\right).$$

C

House Transaction Data

The data that is used is house transaction data of houses in the Netherlands that are sold between January 2009 and January 2021. The features that are included in the data set are listed in Table C.1. The histograms of the data are in Figures C.1 and C.2a. Note that the municipality histogram is somewhat different from the others. A histogram of the house transaction prices is in Figure C.2b. A summary of the numerical features in the data set is in Table C.2. The top three and bottom three of the number of transactions in a municipality are in Table C.3.

Some feature engineering is performed before the data is used in the models. All categorical features are one-hot encoded. That there are columns added for every different category. If an instance is of a specific category, then it has a 1 in the column of that category and a 0 in the columns of all other categories. The transaction date is made discrete by taking the month of the transaction. A lot of feature engineering is performed on the surface area. The surface area is split up into 2 different columns, one column when the house type is an apartment and the other column for all other types. Furthermore, there are 4 more columns that contain information about the surface, such as lot surface area etcetera. These columns are called surface extras 1 till 4. Besides, these surface extras features are such that the first square metres of surface area weigh heavier to the house price than the last square metres. The construction year is split up into splines from 1910 till 2000 in steps of 10 and 2008 where the spline function is $\max(0, \text{construction year} - \text{base year})$. There is also a dummy variable for houses with a construction year before 1900. The storage, garage and monumental features are dummy variables that are 1 if the house has the specification and 0 otherwise.

feature	**type**
transaction date	discrete
province	categorical
municipality	categorical
house type kad	categorical
house type EGWO	categorical
house type STWO	categorical
house type soc	categorical
surface area	continuous
surface extras	continuous
construction year	discrete
storage	dummy
garage	dummy
monumental	dummy

Table C.1: Features in the house transaction data set.

feature	**min**	**0.25-quantile**	**mean**	**median**	**0.75-quantile**	**max**	**standard deviation**
transaction date	2009-01-02	2013-08-23	2016-03-02	2016-10-14	2019-01-09	2021-01-29	
surface area	15	90	117	112	135	1021	43.5
construction year	1900	1955	1970	1974	1994	2021	29.3

Table C.2: Summary of the numerical features in the house transaction data set.

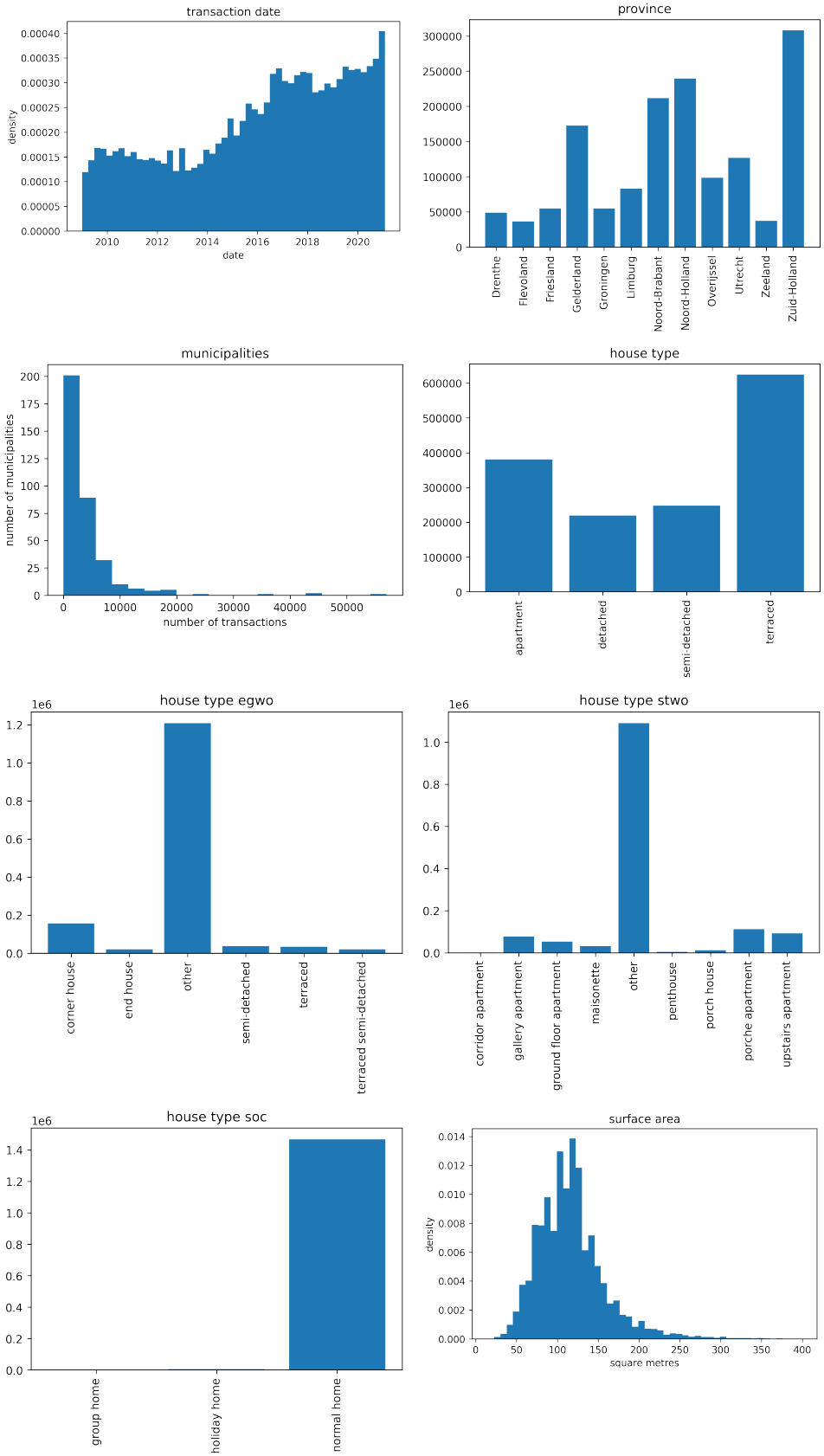
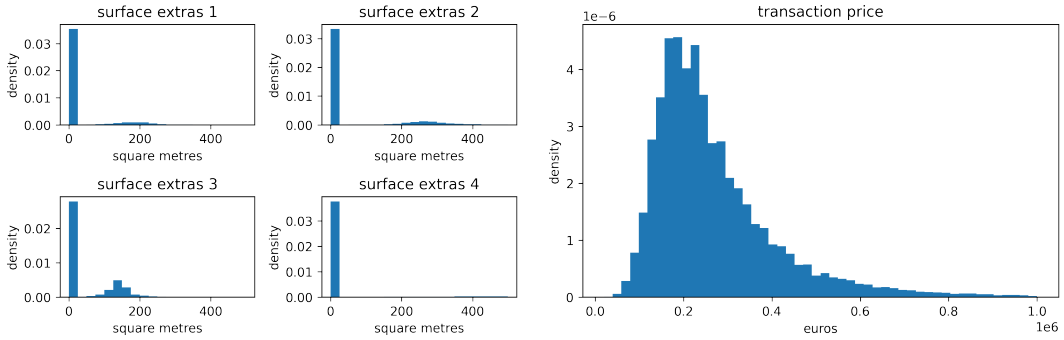


Figure C.1: Histograms of features of the house transaction data set.

top 3	municipality	number of transactions	bottom 3	municipality	number of transactions
1.	Amsterdam	57,023	1.	Vlieland	29
2.	's-Gravenhage	44,691	2.	Schiermonnikoog	110
3.	Rotterdam	42,804	3.	Rozendaal	177

Table C.3: The top three and bottom three of the number of transactions per municipality in the house transaction data set.



(a) Histograms of the surface extras features of the house transaction data set.

(b) Histogram of the house transactions prices.

D

HTM Coefficients

The Hierarchical Trend Model (HTM) model from Appendix A is fitted on the house transaction data of Appendix C. The most important coefficients of the fit will be summarized, to a better understanding of the model. The $\hat{\beta}$-coefficients of the linear regression fit on the time-invariant part are in Table D.1. Only the municipality coefficients are too many to put in a table, so they are shown in a histogram in Figure D.1. The top three and bottom three $\hat{\beta}$-coefficients of the municipalities are in Table D.2. These results are not surprising, since very desired municipalities have high coefficients and less desired municipalities have low coefficients. Remember that the HTM model uses trends of clusters. For this fit, the province and house type are used as clusters. The development of these trends is in Figures D.2 and D.3. The trends are all set to 100 at January 2008. Note that the trends have the expected shape since this is how the housing market developed in the Netherlands during this period.

feature	$\hat{\beta}$	**feature**	$\hat{\beta}$
house type terraced semi-detached (EGWO)	0.003173072	surface area extras 2	0.224999231
house type terraced (EGWO)	0.107246457	surface area extras 3	0.200173825
house type semi-detached (EGWO)	-0.012320248	surface area extras 4	0.224807930
house type corner house (EGWO)	0.019966976	construction year dummy 1900	-0.021647426
house type end house (EGWO)	0.009996303	construction year spline 1910	-0.008036799
house type gallery apartment (STWO)	0.020000000	construction year spline 1920	0.008299518
house type porch apartment (STWO)	0.049779011	construction year spline 1930	-0.003442507
house type corridor apartment (STWO)	0.113889307	construction year spline 1940	0.000940951
house type maisonette (STWO)	-0.043082343	construction year spline 1950	-0.006960054
house type ground floor apartment (STWO)	0.122561914	construction year spline 1960	0.004051666
house type upstairs apartment (STWO)	0.070330908	construction year spline 1970	0.010186315
house type porch house (STWO)	-0.027440450	construction year spline 1980	0.003409437
house type penthouse (STWO)	0.293649257	construction year spline 1990	-0.007069120
house type holiday home (soc)	-0.089998182	construction year spline 2000	0.005342239
house type group home (soc)	-0.022780572	construction year spline 2008	-0.007977338
surface area non apartment log	0.879971803	storage	0.008091202
surface area apartment log	0.832443062	garage	0.031182263
surface area extras 1	0.200000027	monumental	0.135580554

Table D.1: The $\hat{\beta}$-coefficients of the fitted HTM model.

top 3	**municipality**	$\hat{\beta}$	**bottom 3**	**municipality**	$\hat{\beta}$
1.	Terschelling	0.528	1.	Den Helder	−0.464
2.	Groningen	0.483	2.	Hollands Kroon	−0.405
3.	Amsterdam	0.459	3.	Medemblik	−0.367

Table D.2: The top three and bottom three of the $\hat{\beta}$-coefficients of the fitted HTM model.

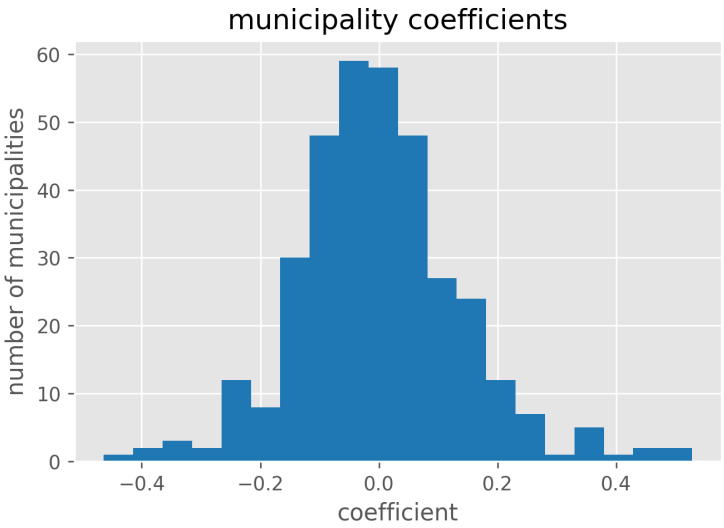


Figure D.1: A histogram of the $\hat{\beta}$-coefficients of all municipalities.

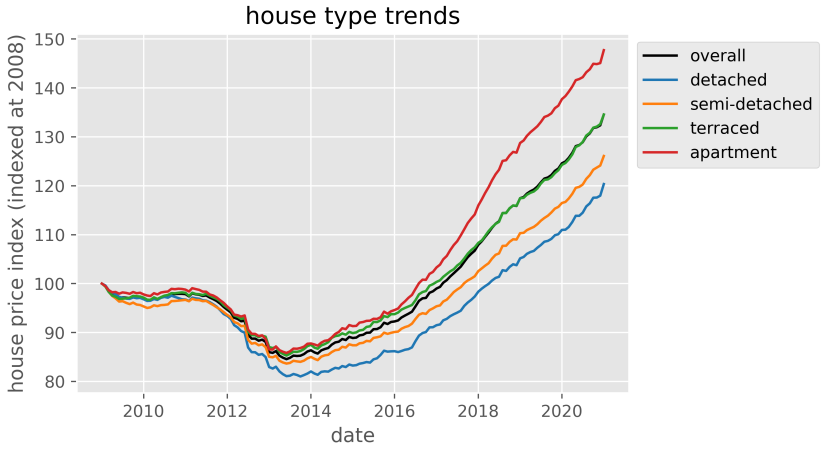


Figure D.2: The house type trends from 2008 to 2021 where 2008 is the base year.

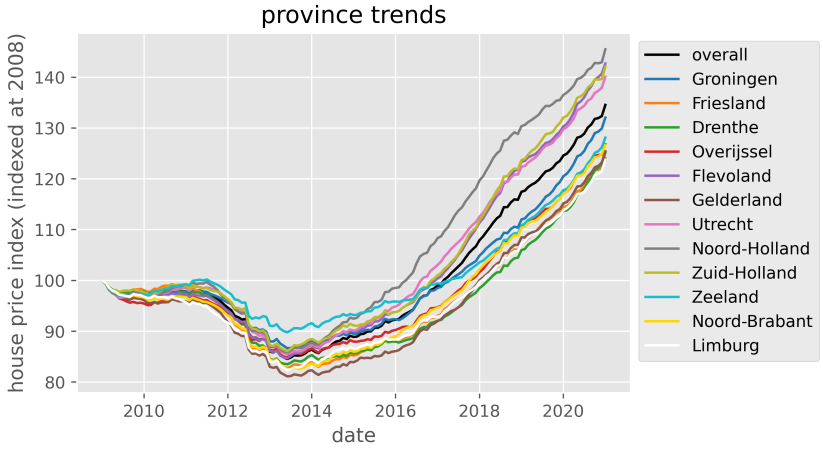


Figure D.3: The province trends from 2008 to 2021 where 2008 is the base year.

Bibliography

[1] K. Aas, M. Jullum, and A. Løland, "Explaining individual predictions when features are dependent: More accurate approximations to shapley values," *Artificial Intelligence*, vol. 298, p. 103502, 2021.

[2] D. Lewis, "Causal explanation," 1986.

[3] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," 6 2017.

[4] A. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[5] Z. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[6] C. Molnar, *Interpretable Machine Learning*. 2019. `https://christophm.github.io/interpretable-ml-book/`.

[7] M. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations.," vol. 18, pp. 1527–1535, 2018.

[8] M. Ribeiro, S. Singh, and C. Guestrin, """ why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[9] D. Alvarez-Melis and T. Jaakkola, "On the robustness of interpretability methods," *arXiv preprint arXiv:1806.08049*, 2018.

[10] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[11] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[12] S. Dandl, C. Molnar, M. Binder, and B. Bischl, "Multi-objective counterfactual explanations," *arXiv preprint arXiv:2004.11165*, 2020.

[13] T. Ferguson, *PART IV Games in Coalition Form*.

[14] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," in *International Conference on Machine Learning*, pp. 9269–9278, PMLR, 2020.

[15] L. Merrick and A. Taly, "The explanation game: Explaining machine learning models using shapley values," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 17–38, Springer, 2020.

[16] M. Käärik, A. Selart, E. Käärik, and J. Liivi, "The use of copulas to model conditional expectation for multivariate data," in *ISI Proc. 58th World Statistical Congress*, pp. 5533–5538, 2011.

[17] J. Segers, "Copulas: An introduction i - fundamentals," 2013.

[18] M. Francke and A. De Vos, "Efficient computation of hierarchical trends," *Journal of Business and Economic Statistics*, vol. 18, pp. 51–57, 2000.

[19] G. Francke, M.K. & Vos, "The hierarchical trend model for property valuation and local price indices," 2004.

[20] M. Francke, *The Hierarchical Trend Model*. Wiley-Blackwell, 2 2008.