From Feature Selection to Data Augmentation: the ADA Algorithm

Eduard Cruset Pla
Supervisors: Rihan Hai and Andra Ionescu
EEMCS, Delft University of Technology, The Netherlands
22-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

From Feature Selection to Data Augmentation: the ADA Algorithm

Eduard Cruset Pla
Supervisors: Rihan Hai , Andra Ionescu
EEMCS, Technische Universiteit Delft, The Netherlands

Abstract

The democratization of data science, and in particular of the machine learning pipeline, has focused on the automation of model selection, feature processing, and hyperparameter tuning. Nevertheless, the need for high-quality data for increased performance has sparked interest in the inclusion of data augmentation in these automatic machine learning techniques. This research approaches this topic by examining different feature selection techniques that will ultimately allow devising what makes a feature desirable. We introduce an automatic data augmentation process, tailored for support vector machines, that employs sample joins. This approach is evaluated through different setups, datasets, and other machine learning models: CART, random forests, and XGBoost. The results are mixed: the algorithm identifies the features containing the signal, resulting in accuracy scores close to the models trained with all the data. However, the computational time is higher. A theoretical analysis suggest that the methodology might be helpful in particular cases where data is structured in specific ways.

1 Introduction

Advances in machine learning (ML) have launched a burgeoning literature on the democratization of data science [17][26] in pursuit of the automation of the ML pipeline: from model selection, and hyperparameter tuning, to feature processing. Particular focus has been placed on the latter [8][19], which aims at the minimisation of the dependency on domain expertise by exploiting statistical knowledge [17]. Indeed, more advanced techniques of automatic ML have demonstrated to outperform experts [31] and arguably suggest a rapprochement to the upper bound set by the signal contained in a certain dataset.

Evidently, model performance relies highly on the quality of the data, thus focusing on a single table might fail to exploit the power of all available data [9]. For instance, a taxi service company might not collect enough data to accurately predict the time a taxi ride will take; and consequently would benefit from publicly available datasets such as the current and forecast weather, or that of construction work taking place on the specified route. Nevertheless, the existence of thousands of repositories with millions of datasets makes finding pertinent data a tedious burden [3]; even after correct identification of the tables, determining what features will increase model performance can be a remarkably computationally expensive task. Extensive work has aimed at discovering the characteristics a table should possess to determine *ex-ante* the benefits of performing a join with the base table [5][9][18]. In their current form, these results prompt the question: can data augmentation techniques also be automated and included in the ML pipelines in an efficient manner?

This research attempts to reverse-engineer the process of feature selection to first analyse what characteristics are desirable for a model. Such scrutiny will allow the formulation of an approach for the automation of feature discovery that will be thoroughly evaluated to derive insights into its performance, robustness, and efficiency.

This approach is thus divided into three subcategories: **(i)** What characteristics make a feature appealing? By reverse engineering the results of different feature selection techniques we will study what makes a variable desirable. Moreover, this paper investigates whether a multivariate analysis is required to define the appeal of a variable. **(ii)** Can these findings be used for an efficient and automatic feature discovery process? Transposing the previous results into the creation of an algorithm that automatically and efficiently performs feature discovery. **(iii)** Does this hold for other models? By evaluating this approach under different setups to study the robustness and extension of the results.

The use of support vector machines (SVMs) is justified under the context of this research, where the objective is to analyse whether tree-based classification models would require a different process of automatic data augmentation compared to a linear model, such as SMVs. Moreover, this model is chosen for its rich literature and popularity [2][22][29]. The research is narrowed down to binary classification problems and will not discuss regression techniques.

The main contributions of this paper can be summarized as follows: **(i)** The use of sample joins helps identify the desirability of variables, albeit at a large computational expense. **(ii)** The automatic data augmentation (ADA) algorithm presented outperforms the use of the base table. However, joining all tables and performing feature selection techniques *ex-post*, yields the best results. Nevertheless, this could arguably be due the format and dimensions of the data. **(iii)** Finally, the results suggest the presence of overfitting when using the ADA algorithm. This indicated that a more stringent desirability computation could be beneficial [1].

These findings add to the existing literature on automatic feature discovery presented in section 2. Following, sec-

[1] The algorithm, its evaluation, and the data can be accessed in the following repository: github.com/mrcruset22/bachelorthesis.

tion 3 introduces the framework under which the feature study is performed: model, feature selection techniques, data, and results obtained. Consequently, an approach to automatic feature discovery is defined in 4. This, in turn, is scrutinized in an evaluation outlined in section 5. Finally, sections 6 and 7 discuss the results of the research and state some concluding remarks.

2 Related Work

Automatic data augmentation techniques constitute a *still* unripe literature when compared to other stages of the automatic machine learning pipeline [5]. Consequentially, no *golden rules* have been yet devised, and the exploration of many different methodologies defines best the literature's current state. Nevertheless, two main approaches can be identified: automatic data augmentation through synthetic oversampling from existing data, and dataset augmentation through feature discovery in other repositories and tables.

2.1 Synthetic Oversampling

Deep learning is particularly vulnerable to insufficient amounts of data on which to train a model [13]. Most notably, in the image domain, great focus has been given to automatic data augmentation; techniques range from the introduction of new features through processes such as applying flips or translations to an image [11], the inclusion of filters and kernels to the model's architecture [14], and more complex approaches such as the renowned *AutoAugment* [7] or DADA [20].

These approaches service specific problems in the subfield of media analysis. Their transposition into other domains, such as biology, has been recently explored [33][27], and their generalisation to support any reinforcement learning has similarly been investigated [24]. These approaches, however, constitute a parallel branch of research to the one this paper will follow, which rather than focusing on data augmentation in its strictest definition, will delve into the practices of dataset augmentation through joins.

2.2 Automatic Dataset Augmentation

Feature discovery through join-based data augmentation has often relied on domain expertise. In order to avoid or reduce human-in-the-loop interaction and its disadvantages, the inclusion of feature discovery into the automatic machine learning pipeline has been successively proposed. Different approaches have been presented that can be grouped into two categories.

First, a negative approach to the issue by analysing the tables that are safely disregardable. Kumar's notable *"to join or not to join"* [18] develops on this idea of avoiding joining tables that do not yield meaningful increases in prediction accuracy. It is based on the Vapnik-Chervonenkis (VC) dimension to avoid including redundant information. However, this approach focuses on whether a table or feature should be used for augmentation, rather than discovering these in the first place.

Second, finding the most suitable tables within the data set through an external score system. ARDA [5] joins the base table with those that obtain the highest scores, i.e., are selected in the top-k. Aurum [10], a join discovery system, attaches a score to each of the tables of the database that can be joined to the base table, on which ARDA is based. This work is complementary to the one in this paper as it relies on an external scoring metric that determines the relevance of tables and features and does not measure *ex-ante* the impact on accuracy they will entail, indeed it proposes a reverse-engineered-based external scoring system to elicit the appeal of performing a particular join. Similarly, COCOA [9] is a data augmentation technique that utilizes correlation to extract insight into the desirability of a table or feature. However, this technique uses virtual join for a more efficient approach to the feature discovery process. Alternatively, this paper presents a solution based on sample joins to disregard unwanted tables, but it performs full-fledged joins in the primary key-foreign key (PK-FK) relational path traversal.

3 Background

This section introduces the models and techniques that will be used and examined for the automatic dataset augmentation approach. First, an overview of SVMs in section 3.1, followed by the introduction of different feature selection techniques used in section 3.2. Finally, the data and metrics are presented in sections 3.3 and 3.4.

3.1 Support Vector Machines

Support Vector Machines (SVMs) is a computer algorithm that learns by example to assign labels to objects [2]. Their success has been in many fields from biology to economics and finance [22]. There are four key aspects to the understanding of an SVM: (i) the separating hyperplane, which entails geometrically dividing the two classes with a line in two dimensions or a hyperplane in higher dimensions. (ii) The maximum-margin hyperplane; this technique, based on information theory [29], chooses the hyperplane that maximizes the distance to the nearest data point. (iii) The soft margin, which is used when data is not perfectly linearly separable. It allows for certain data points to fall under the wrong side of the hyperplane, by introducing a penalty in the loss function. Finally, (iv) the kernel trick; which allows for an increase in the dimensionality of the feature space to tackle non-linear problems [1]. Given the context of this research and the interest in deriving differences between linear and non-linear models, this trick will be not used unless stated otherwise. Equation (1) presents the base loss function for a soft margin SVM [6]:

$$C||\mathbf{w}||^2 + \frac{1}{n}\sum_{i=1}^{n} max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i - b) \qquad (1)$$

Where $\mathbf{w}$ is the feature vector, y_i is the class label (either 1 or -1), $\mathbf{w}^T\mathbf{x}_i - b$ represents the separating hyperplane, and C establishes the rigidness for the hinge loss [25].

3.2 Feature Selection Techniques

1. *Variance threshold* is a baseline filter feature selection technique [4]. It selects the independent variables that have a larger variance than a pre-set threshold. Alternatively, it also offers a relative implementation by which the user can select the top percentage of features with the largest variance. Although this methodology can be very useful when in need of a simple technique that is not computationally expensive, it will not be much used in this analysis as it

already relies on the heuristic that variables with high variance are appealing.

2. *Correlation with class*. How the variable is correlated with the target variable is closely examined [12].

3. *Linear discriminant analysis (LDA)* is not *per se* a feature selection technique but is commonly use as a tool for dimensionality reduction [28]. Alternatively, it offers a vector in the feature space which best separates both classes. There exists a well-known implementation for multi-class tasks. In this analysis is used to derive insight into the correlation between each independent variable and the LDA-created feature.

4. *Select K best*. This methodology consists in ranking all variables using a score function. Three will be used:

(a) *ANOVA* This technique uses the ANOVA test; i.e., it analyses the difference among the class-conditional means. The resulting score is the p-value derived from performing an F-test [27].

(b) *Mutual Information*. Similar to correlation, it analyses the amount of entropy shared by two variables and thus, the information that one variable contains about a second variable [30].

(c) *Chi2* It uses the value resulting from the chi-squared test [21]. Therefore, it requires non-negative features. Since it mostly relies on frequencies it will not be used in many dataset instances containing purely numerical variables.

5. *Recursive feature elimination* is as straightforward as the name indicates. This wrapper method relies on a model that attaches a score or coefficient to each feature and recursively deletes that with the lowest value [4].

6. *Regularization: L1 and L2*. Regularization is an embedded method that is not native to SVMs; nevertheless, extensive literature exists on its analysis and benefits [16][23][32]. Given the problem at stake, the regularisers for sparsity L1 and L2 are preferred. These tend to help highlight the relative importance of each feature. Equations (2) and (3) are added to the loss function (1) of the SVM, where the hyperparameter λ determines the rigidness of the regularisation.

$$\lambda \sum_{1}^{n} \mathbf{w}_i \qquad (2) \qquad\qquad \lambda \sum_{1}^{n} \mathbf{w}_i^2 \qquad (3)$$

7. *Sequential feature selection: backward and forward*. This exhaustive wrapper method is usually the most computationally expensive technique. The backward (forward) format requires the target model to be trained by excluding (including) each feature at a time and only exclude (include) that that decreases (increases) the selected score metric the least (most). Due its complexity is often disregarded; however, it usually yields the best results [15].

3.3 Data

In order to extract heuristics that are both comprehensive and robust, the datasets need possess analogous qualities. Therefore, in order to maximize the information to extract, the data used was carefully chosen to be as representative as possible, rich —i.e., different number of numerical and categorical features, ratio of rows and columns, proportions of NaNs per variable, etc.— and evidently, belong to different fields —namely economics, biology, medicine, etc. Table 1 includes their names, together with their dimensions, specifying the number of categorical and numerical features. All datasets used are publicly available.

	#var. (num., cat.)	#rows
Banknotes	4 (4, 0)	1.372
Heart disease	13 (13, 0)	297
Card fraud detection	29 (29, 0)	284.807
Income estimation	14 (6, 8)	48.842
Mushroom poisonousness	22 (0, 22)	8.124
NBA contracts	20 (19, 1)	1.340
Stroke prediction	11 (6, 5)	5.110
Water potability	9 (9, 0)	3.276
Cancer detection	30 (30, 0)	569

Table 1: Data sets for variable examination

Categorical data and NaNs

SVMs are models that require numerical features that do not contain any missing values. Unfortunately, this prerequisite is largely unsatisfied in most datasets; thus, needing certain pre-processing of the data before this can be used for training. Regarding categorical data, whenever a SVM is used, the creation of dummy variables will (unless the categorical variable consists of a hierarchic variable in whole numbers; namely, the satisfaction of a client on a scale from 0 to 10). On the other hand, NaNs will be handled through simple univariate imputation techniques: using the mean of the known values for numerical data, and using the most frequent value for categorical. Further research could be carried out on the effects of performing more complex imputation techniques on missing values. Nevertheless, such examination falls beyond the scope of this paper.

3.4 Metrics

In order to assess the performance of each feature selected by the different feature selection techniques, different metrics are used: ***(i)*** *Accuracy* computed as the percentage of correctly classified predictions on unseen data. It is also compared to the accuracy of the training set to study the possibility of overfitting. ***(iii)*** *Receiver operating characteristic (ROC) curve* which allows the study of the sensitivity vs. specificity trade-off in binary classification. ***(iv)*** *F1-score* metric combines true positives, false positives, and false negatives which allows for model performance comparison. Both the precision and recall, used to compute the F1-score are also individually assessed. ***(v)*** *Coefficients* of the features and hyperplane. After controlling for the variance of the feature, a small coefficient is interpreted as having small importance in the classification task.

4 Automatic Data Augmentation (ADA) Algorithm Design

Sections 3.1 to 3.4 presented the techniques, data, and metrics to analyse feature desirability for linear SVMs. These

will be used for the design of the Automatic Data Augmentation (ADA) algorithm. The development of this approach has been the result of an iterative study of how the feature selection techniques select certain variables over others, findings that will be transformed into heuristics and will be ultimately transposed into the automation of feature discovery.

First, the data format the algorithm expects is discussed in section 4.1. Second, the process of designing and tuning the desirability of variables and tables is presented in section 4.2. Last, the ADA algorithm is detailed in section 4.3.

4.1 Data Format

This research assumes the input data to have the following format: a base table b_0 containing m features $f_{0j} \in b_0$ where the first subscript refers to the table it belongs to, and the second the feature it represents. The data set contains $b_i \in B$ where n represents the depth of B. Graph G represents the PK-FK relationships —i.e., relationship expressed as $b_a - b_b$ allows you to join tables b_a and b_b. Moreover, the graph is restricted to be a tree; thus, containing a root b_0 and no cycles. This restriction allows for the path to join a particular table to be unique. Therefore, P represents the set of paths p_i. P has the same depth as B and each path p_i expresses the path from b_0 to b_i in the format $b_0 - b_j - ... - b_i$. Figure 1 constitutes an example of this format.

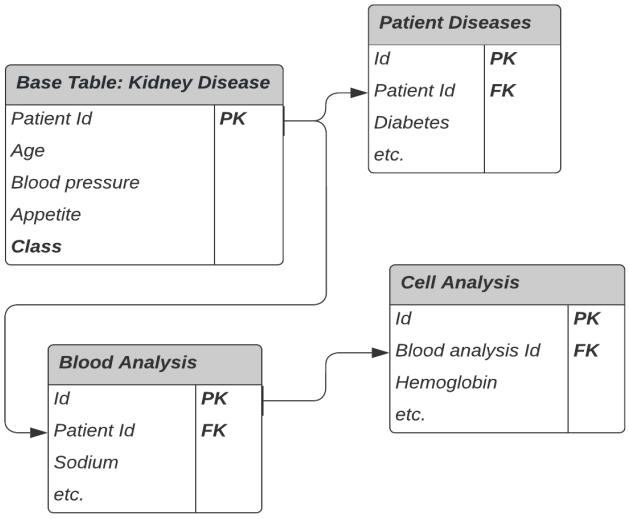

Figure 1: Data set example

4.2 Computing desirability

When deciding the suitability and consequent inclusion of a variable, not only are the traits of such variable important, but also the context: features already included, current model performance, residual variance, etc.

Therefore, the results are subdivided into two categories: univariate, and multivariate. The former will examine what intrinsic characteristics the variables have. The latter will be subdivided into bivariate and multivariate. A bivariate examination will help understand what relation a desirable variable has with the target variable, namely the class. Finally, the multivariate analysis will also look into the relationships with the other variables already included in the model, either present in the original base table, or already included through the same process. These design choices are a product of the feature selection analysis, whose summary can be found in appendix A.

Univariate

There only exist so many characteristics that define a single feature. For this study, the focus has been given to data type (continuous, discreet, or categorical), the mean and variance, and the distribution that the variable approximately follows. These are the components that will determine the desirability. Moreover, simple statistics of the class variable will also be used, since these can be computed before joining a table.

• *Data type*. Numerical variables, both continuous and discreet, are recurrently chosen over their categorical counterparts. This is particularly true when the categorical variable contains a large number of unique values that, after performing one-hot encoding, increase drastically the dimensionality of the feature space, which leads to overfitting. Therefore, the design of the univariate desirability algorithm will attach a higher score to numerical variables.

• *Mean and variance*. Mean does not seem to be a determinant factor in the suitability of a variable. Regarding the variable's variance, the results seem to reassert the well-known heuristic of higher variance is preferred. However, after a closer look, at binary classification this threshold is relatively low: variables with variance at least as high as that of the target feature can be good candidates. For all this, the algorithm will use the target's class variance as a threshold and will not incorporate the mean of the variable when computing its desirability.

• *Distribution*. The specific distribution that a variable approximately follows yields inconclusive results on its impact on desirability. Nevertheless, two different traits can help shed some light:

1. *Frequencies*. In categorical variables, when most data points contain the same category, in a larger percentage than the most common class, the variable is never selected. Therefore, the algorithm will compute the percentage of the most repeated category for categorical variables and compare it to the percentage of the most frequent class. It will only consider the feature as long as it has a smaller percentage.
2. *Outliers* can sometimes be very helpful, particularly when the target feature contains a very unbalanced class. Nevertheless, this will not be incorporated into the algorithm as it can be a very computationally expensive task.

Multivariate

Features might seem highly desirable after a univariate analysis, and yet contain no information or signal on the classification task at hand. Therefore, we analyse the feature from both bivariate and multivariate perspectives, which will ultimately help better determine its desirability.

• *Bivariate*. There are three suitable approaches that can help determine the desirability of a feature based on bivariate analysis with the target feature:

Algorithm 1 Univariate desirability

Input: Table t. Proportion p of target feature.
$desirability \leftarrow \alpha \cdot length(t)$
for $f_i \in t$ **do**
  if $dtype(f_i)$ **is** numerical **then**
    $desirability \leftarrow desirability + \beta \cdot variance(f_i)$
  else
    if $p \geq FrequencyMax(f_i)$ **then**
      $desirability \leftarrow desirability + \gamma$
return $desirability$

1. Correlation with the class. A higher correlation with the target feature consistently implies good results in the classification task.
2. The F-statistic results of an ANOVA f-test. The F-statistic illustrates the probability that the class-conditional distributions come from the same distribution. A smaller statistic suggests they do not and usually implies a higher prediction power.
3. Correlation with the one-dimensional projection of the linear discriminant analysis (LDA). This technique computes a one-dimensional line —for binary classification— that best separates both classes and projects each datapoint in it. Variables that have a high correlation with it imply that they can partially separate both classes and be desirable.

Although all three metrics were initially considered for the computation of desirability, it became apparent that adding all three significantly slows down the desirability computation and they add little extra information to each other. Using the results of appendix A: **1.** correlation between class-correlation and LDA-correlation: 0.9776; **2.** correlation between class-correlation and ANOVA's F-statistic: 0.8927; **3.** correlation between LDA-correlation and ANOVA's F-statistic: 0.9039.

This led to the choice of only including the correlation with the one-dimensional projection of the LDA as a decision to include numerical variables. First, it will decrease the time of computation. Second, the data suggest it is the most correlated with the other two. Third, it also accounts for explainability within the base table.

• *Multivariate*. Multivariate analysis can become much more complex, as well as cumbersome, compared to the two above. However, a variable that satisfies all heuristics presented thus far might still not be actually desirable. This is because most information that the feature contains is already present in the base table. This is particularly dangerous for features that are linear combinations of variables already in use since it usually implies the modeling of the noise in the data. A technique considered to take this phenomenon into account is the computation of the residual variance. Unfortunately, this approach contains two drawbacks. First, it implies that the model is trained every time there is a new variable incorporated into the base table, significantly slowing down the computation of the algorithm. Second, it risks overfitting by trying to capture the noise in the data. Therefore, the multivariate analysis will be limited to the bivariate evaluation described above.

Algorithm 2 Multivariate desirability

Input: Table b after sample join, set F of features to analyse.
Initialize array A
for $f_i \in F$ **do**
  $d \leftarrow d + \gamma \cdot LDA(f_i, class)$
  if $d \geq threshold_0$ **then**
    Add f_i in A.
return A

4.3 The ADA Algorithm

Having presented the methodology to determine the desirability of a table and variable, this section will outline the format of the table traversal and the exact functioning of the ADA Algorithm.

The first step is to iterate through all of the tables that can be joined using a breadth-first search algorithm. This allows for a quicker and easier implementation since most foreign keys refer to the same private key on the base table. Then, for each table, the univariate desirability will be computed. Given that this analysis is better at identifying those variables that should not be joined, rather than those that should, only a sample join and subsequent multivariate analysis will be performed as long as it yields a high enough score. Moreover, this also ensures a speed-up in the process. Finally, when a sample join is performed, the multivariate analysis will determine the variables in the table that are desirable and join these. These can range from none to all. Finally, after the complete traversal, the augmented base table is returned, on which the model can be trained.

Algorithm 3 ADA

Input: Base table t_0, other tables and path pair (t_i, p_i) in T.
$\rho \leftarrow proportion(target\_feature)$
for $(t_i, p_i) \in T$ using BSF **do**
  $d \leftarrow univariateDesirability(t_i, \rho)$
  if $d_i \geq threshold_0$ **then**
    $s_i \leftarrow sampleJoin(t_i)$
    $A \leftarrow multivariateDesirability(s_i)$
    $Join(A, t_0)$
return t_0

5 Evaluation

An essential step for the assessment of the ADA algorithm is to evaluate it using unseen databases, formatted in the required way, and compare it with other approaches to data augmentation. This section incorporates all aspects of such evaluation: the data used, the metrics, and the different setups that have been carried out. Finally, the results are discussed.

5.1 Data

In order to evaluate the ADA algorithm, four different data sets have been selected. They all follow the format specified in section 4.1 and are collected in table 2. Moreover, although it is not a strict requirement for the application of

the algorithm, all tables that can be joined in the data sets used contain the same number of rows as the base table.

	#rows	#tables	#features per table
Kidney disease	400	4	**4**, 11, 9, 7
Titanic	891	4	**3**, 2, 6, 5
Steel plate fault	1941	8	**9**, 7, 6, 5, 6, 5, 5, 8
Football	1182	9	**5**, 13, 5, 5, 6, 6, 6, 6, 6

Table 2: Data sets for evaluation. The first number in *features per table* corresponds to the base table.

5.2 Metrics

In order to evaluate the performance of the different models, we will rely on classification accuracy in both the training and testing data. This will also allow shedding light on whether the model is overfitting the data, a major issue in data augmentation. Moreover, in order to capture the differences in computational power required, the time elapsed for the completion of each technique will also be computed and discussed.

5.3 Setups

1. *Baseline.* This evaluation will be used as the baseline for result comparison. The model will only be trained with the base table and will not use any feature selection or dimensionality reduction techniques. However, as it is standard, L1 regularization will be added with the default parameter.

2. *Join-All.* On the other side of the spectrum is the inclusion of all data available. This setup will perform a join for all PK-FK relationships that exist in the dataset. This setup is further subdivided into two:

(a) *Naive.* This approach will train the model with the features from all tables and will not perform feature selection or dimensionality reduction. The time metric will include both, the joining process and the training of the model.

(b) *post Feature Selection.* Conversely to the naive approach, this setup will perform feature selection *ex-post*, i.e., once all features have been added to the base table. Similarly, all three steps will be included in the time measurement: joining, feature selection, and training of the model. After the consideration of the techniques used in section 3.2, the ANOVA F-statistic was chosen for its fast and robust results.

3. *ADA Algorithm.* This setup will evaluate the algorithm introduced in section 4. The time measurement will include both the running of the algorithm and the training of the model with the yielded variables.

4. *Extensions and Robustness.* This final evaluation consists of the reproduction of all four setups described above for different models. In order to capture the robustness of the ADA algorithm, the same exercise will be performed with three tree-based models: CART, Random Forest, and XGBoost, as well as an extension of SVMs utilizing the non-linear kernel trick. Note that the building of the algorithm is tailored to linear SVMs; this final exercise aims at answering whether the approach is, however, model agnostic.

5.4 Results

Baseline

The baseline setup is the simplest of all four that are analysed. Therefore, it is expected that the prediction power of the data is the smallest in comparison and that it requires the least amount of computational time.

The results presented in table 3 seem to reflect this idea. Indeed, the time required to compute these results took the shortest out of all approaches. In fact, most processes required less than one second of computational time.

Regarding the accuracy, the results indicate a suboptimal outcome that becomes more apparent when compared to the other setups. All in all, the base tables with which these computations were performed only contain approximately 15% of the available data, greatly explaining these worse results.

Naive Join-All

This setup combines all the tables that are available, thus using 100% of the data. Table 4 shows the results for this setup.

A first glance at the table reveals an important increase in accuracy of the model —particularly for the kidney and football databases. However, a more thorough examination of it indicates the presence of overfitting in a majority of cases, where the accuracy of the training set is significantly higher than that of the testing set. This increase in overfitting compared to the baseline approach is explained by the increase in data available: by including all available variables the dimensionality of the feature space is increased and so does the flexibility of the model. This implies that the different models are capturing and modeling inherent noise within the data that does not help predict the class of unseen data points.

Regarding the computational time, the increase is noticeable but perhaps much lower than anticipated. The explanation is twofold: first, this is a direct consequence of the stringent format of the data. For instance, the tables are of the same size, and therefore the joining process is accelerated. Second, the databases that have been used for evaluation contain a very small number of rows. Given that the complexity of joining two tables is $\mathcal{O}(n \cdot log(n) + m \cdot log(m))$ —uniquely depending on the length of such tables— the computational time derived from the joining process is concealed by the training process.

Feature Selection Join-All

This setup builds on the previous one. After combining all tables, we perform a simple feature selection technique to reduce the number of features used for training. The feature selection technique is the filter method of the ANOVA F-statistic.

A first look at table 5 shows the significantly lower computational times than for the naive join-all approach. This results is surprising until the times are discerned into subcategories. For instance, for the kidney database, the joining process of all tables only takes 89 milliseconds, the use of the feature selection technique takes 189 milliseconds. However, the training of the SVM with all features takes 26,8 seconds, and the training with the subset of features chosen takes 2,9 seconds. This distinction makes it clearer on how to interpret the results of the table. Regarding accuracy, there seems to be an overall slight improvement in the

	Linear SVM			Non-linear SVM			CART			Random Forest			XGBoost		
	Train	Test	Time	Train	Test	Time	Train	Test	Time	Train	Test	Time	Train	Test	Time
Kidney Disease	0,729	0,675	33ms	0,711	0,625	31ms	0,896	0,717	33ms	0,896	0,725	207ms	0,871	0,767	185ms
Titanic	1	0,586	287ms	1	0,586	579ms	1	0,586	536ms	0,998	0,586	1,12s	0,629	0,586	803ms
Steel Plate Fault	0,712	0,705	229ms	0,772	0,763	265ms	1	0,765	32ms	1	0,799	645ms	0,856	0,798	405ms
Football	0,682	0,549	126ms	0,813	0,538	204ms	0,891	0,53	28,9ms	0,891	0,544	396ms	0,783	0,555	452ms

Table 3: Baseline evaluation results.

	Linear SVM			Non-linear SVM			CART			Random Forest			XGBoost		
	Train	Test	Time	Train	Test	Time	Train	Test	Time	Train	Test	Time	Train	Test	Time
Kidney Disease	0,979	0,983	17,2s	0,621	0,633	112ms	1	0,975	91ms	1	1	298ms	1	0,992	195ms
Titanic	1	0,784	537ms	0,643	0,461	332ms	1	0,755	404ms	1	0,676	450ms	0,996	0,735	503ms
Steel Plate Fault	1	1	320ms	1	0,997	387ms	1	1	396ms	1	0,997	1,26s	1	1	1,34s
Football	1	1	14,4s	0,616	0,623	1,22s	1	0,995	17,9ms	1	0,953	646ms	1	0,995	1,64s

Table 4: Naive join-all results.

results. In most cases, the accuracy for the test set is either improved or kept constant. Moreover, in some cases, the gap between the accuracy for the training set and testing set is decreased, implying a diminished tendency to overfit. Finally, this also indicates that the signal contained in the data is present in a subset of variables that are selected in this approach. Therefore, some features do not add any prediction power to the exercise.

ADA Algorithm

This section aims to evaluate the algorithm presented in section 4. The results are shown in table 6.

The most noticeable result of all is the substantial increase in computational time: orders of magnitude compared to the baseline approach and a threefold increase in average with the other two. This is largely due to the number of operations that the algorithm computes.

However, it is important to notice that this increases in. The complexity of joining all the data is $\mathcal{O}(n \cdot log(n) + m \cdot log(m))$ where n and m are the tables to be joined. The ADA algorithm has a complexity of $\mathcal{O}(nmt + t^3)$ where n is the length of the table, m is the number of features and t is $min(n, m)$. Therefore, in particular cases where the data is structured in many small tables that only contain a handful of features, the ADA algorithm could result advantageous.

On the other hand, in the datasets used, all tables contained a significant amount of appealing variables. This hampered the univariate analysis and barely disregarded the tables before computing multivariate desirability. In cases were desirable data is contained in a small subset of table, the algorithm would not need to perform sample joins and LDA as often, greatly reducing the computational time in relation to joining all tables. Nevertheless, further evaluation should be carried to confirm all these theoretical propositions.

On the brighter side, the ADA algorithm provides good results in terms of accuracy, greatly over-performing the results of the baseline approach. The small decrease in accuracy in comparison with the third approach indicates that the algorithm correctly identifies the features where the signal is contained, albeit at a much larger expense. Moreover, the results are very similar for all the models —particularly the tree-based models. This implies that the ADA algorithm is somewhat model-agnostic and can be of use in different situations.

6 Limitations & Future Work

There exist a few limitations worth exploring of the approach presented thus far. These are:

1. *Multivariate analysis and residual variance*. One of the most important limitations of this approach is the lack of a procedure to avoid incorporating variables that contain information already present in the base table. As discussed in section 4.2, one option would be the training of the model at each step and computing the sample join with the incorrectly classified datapoints —which in binary classification correspond to the residual variance. However, this approach would imply problems of overfitting, and alternative methods should be considered to incorporate this.

2. *Sample joins can drastically slow down the algorithm in certain cases.* The inclusion of sample joins when analysing the desirability of a variable comes with a major caveat: the relationship between the tables. The format of the data chosen satisfies many requirements that are not present in the majority of practical problems. Therefore, the inclusion of sample joins in the algorithm could hamper its speed in cases where the tables to be joined contains much more data, or more relevant yet, when the joining cannot be performed cleanly and alternative techniques, such as fuzzy joins, need to be performed.

3. *Stringent data format*. Related to the previous point, the stringiness of the data format supposes a limitation on the application of the algorithm. First, it assumes that all the data is known, accessible and the paths among tables is known in advance. Ultimately, this requirements require certain pre-processing of the data, which defeats the point of automated dataset augmentation. Further research could be focused on the relaxation of such requirements.

4. *Categorical variables encoding*. Another caveat that comes with the results of this research is the specific encoding technique that has been used. One-hot-encoding, although ubiquitous in the literature, implies problems of overfitting, especially in certain models as SVMs. This resulted in the little desirability of categorical data that might not correspond to the reality. First, other models, such as the tree-based models used in the evaluation, might benefit more from the inclusion of these variables. Second, other encoding could be used in order to maximize their usefulness to SVMs.

	Linear SVM			Non-linear SVM			CART			Random Forest			XGBoost		
	Train	Test	Time	Train	Test	Time	Train	Test	Time	Train	Test	Time	Train	Test	Time
Kidney Disease	0,993	0,992	3,1s	0,864	0,842	129ms	1	1	97ms	1	1	304ms	1	0,992	175ms
Titanic	0,83	0,755	303ms	0,634	0,529	432ms	0,953	0,824	460ms	0,928	0,775	693ms	0,868	0,784	518ms
Steel Plate Fault	1	1	904ms	1	0,998	292ms	1	1	402ms	1	0,999	1,11s	1	1	1,01s
Football	1	1	2,8s	0,997	0,993	425ms	1	0,997	22,9ms	1	0,997	402ms	1	1	511ms

Table 5: Feature selection on join-all results.

	Linear SVM			Non-linear SVM			CART			Random Forest			XGBoost		
	Train	Test	Time	Train	Test	Time	Train	Test	Time	Train	Test	Time	Train	Test	Time
Kidney Disease	0,925	0,908	41,6s	0,718	0,658	254ms	0,957	0,883	398ms	0,957	0,908	606ms	0,932	0,908	500ms
Titanic	0,999	0,731	1,52s	0,889	0,702	2,65s	1	0,713	213ms	1	0,722	1,96s	0,815	0,705	293ms
Steel Plate Fault	0,975	0,975	755ms	0,99	0,982	1,05s	1	0,986	587ms	1	0,987	1,37s	0,995	0,985	2,12ms
Football	1	1	8,67s	0,95	0,921	3,11s	1	1	2,82s	1	0,977	1,46s	1	1	1,19s

Table 6: ADA algorithm results.

5. *Further robustness checks*. Particularly regarding the data sets used, further diversity in the type of data would result in more thorough robustness checks. Although many characteristics could be added to the data, one is of particular relevance: increasing the number of datapoints. The usefulness of performing sample joins might not be clear after the evaluation with four datasets containing a small number of rows. Further inspection could be done, in which an increased number of rows is present in the data. This is critical because it would alter the results of the algorithm, but analogously, would also greatly alter the time measurements of the *join-all* approach.

7 Conclusions

The automation of data augmentation processes is still an unripe section of the automatic machine learning literature. There exist different approaches to the augmentation of data. Synthetic oversampling is a set of techniques, mainly used in the media analysis field, that creates new data through transformations of existing data. However, this research investigates the automatic dataset augmentation techniques. Whereas a large part of the literature has focused on the univariate examination of features [18][5][9], this paper delves into the automation of data augmentation through sample joining and multivariate analysis.

In order to design an automated process, the thorough analysis of different feature selection techniques allowed devising some heuristics and insights into what makes a variable desirable. The main ones are: First, the variance of numerical variables and the frequency of their categorical counterparts, seem the best proxies to feature desirability under univariate analysis. Moreover, the correlation with the class, the correlation with the one-dimensional projection of the LDA, and the ANOVA's F-statistic are helpful in bivariate analysis. Finally, the inclusion of multivariate analysis also seems a very promising approach. It would allow for the disregarding of variables that contain signal that is already present in the base table. This latter finding, however, was not incorporated into the algorithm.

The ADA algorithm presented in section 4 was designed with these insights in mind. First, it uses univariate analysis to determine whether to perform a sample join using the datatype, correlation, and frequency of the variable. Then, if the desirability surpasses the threshold, a bivariate analysis is performed to determine what variables are going to be joined. The latter is based on the correlation with the one-dimensional projection of the LDA.

This process is relatively complex and results in large computational times. Indeed, the algorithm takes longer than joining all the tables. However, the metrics of accuracy are promising. They show only slight decreases, which implies that the algorithm is able to detect what variables contain the signal. All in all, after a more theoretical scrutiny, the ADA algorithm can be helpful in very particular situations: when the number of features per table is small, and the number of tables to be joined is large. This decreases the complexity of the multivariate desirability computation of the ADA algorithm and increases the expense of performing full-fledged joins. Similarly, in cases where most data is not beneficial for the classification task and the tables can be disregarded with the univariate analysis.

Furthermore, the ADA algorithm presents some signs of model-agnosticism. Although the best results are obtained when the linear SVM is employed, the three tree-based classifiers —CART, random forests, and XGBoost— also produce similar results. Applying a non-linear kernel to the SVM, however, does decrease the performance.

Finally, some limitations and propositions of future work include: (i) The relaxation of the stringent format required for the data. (ii) The inclusion of multivariate analysis. (iii) The evaluation of the approach under different data that analysis whether the algorithm can indeed be helpful in the right circumstances.

8 Responsible Research

This research employs thirteen different datasets. All these data are publicly available and can be found in the repository of the paper.

There are two types of data. That used for the feature selection technique analysis and that used for the evaluation of the algorithm presented. Regarding the nine datasets used for the study of feature selection, the choice of these was carried out beforehand, taking different characteristics as the main deciders: first, the data needed to be publicly available as well as commonly used in literature and other machine learning exercises. Second, the data had to have a specific format: binary classification task, a relatively small amount of rows, and a varying number of features. Finally, it was in the interest of the research to make the

data as comprehensive as possible by including different domains, such as biology, finance, and chemistry. Regarding the choice of the four datasets used for the evaluation of the algorithm, the decision was taken by the supervisor in order to ensure a fair and unbiased examination of the approach.

Furthermore, in order to minimize the unconscious biases in the examination of the results of the feature selection techniques, all of the variables were anonymized. This ensured that the different statistics of each feature and the results of the techniques determined the desirability of the feature, beyond its name or its logical relation with the classification exercise.

Concerning the design and evaluation of the algorithm presented in this research, a strict scientific procedure has been utterly followed. The design choices are justified by the different results incorporated in appendix A. Most importantly, the pseudo-code and the actual code of the algorithm are made publicly available for the scrutiny of any interested parties. The results of the evaluation have been carefully produced, and all the code and data that has been used for it can be found in the repository of this paper. This, however, comes with a caveat: whereas the accuracy results can be completely reproduced using the code available, the metric of *time to compute* cannot be fully reproduced, as this highly depends on the machine in which the software is run. In order to derive the fairest results possible, all of the results were computed under the same conditions and machine.

Furthermore, from an ethical standpoint, this paper does not present any controversial result and/or approach. Although Machine Learning is well-known to have created dissension, this paper does not aim to perform any predictions tasks *per se*; conversely, these are only used for the evaluation of the approach proposed, which can be utilized by any dataset under the format introduced in section 4.1.

Finally, the results and approach presented in this paper cannot be essentially used in a malevolent way. The democratization of the machine learning pipeline pursuits the use of these powerful techniques in a ubiquitous manner. Whereas this implies that such techniques, including the one here presented, can be more easily employed by malicious individuals, the intent of this paper, as well as that of the literature, is diametrical. Indeed, the author condemns any mischievous use of this research.

References

[1] Mark A Aizerman. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[2] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[3] Dan Brickley, Matthew Burgess, and Natasha Noy. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference*, WWW '19, page 1365–1375, New York, NY, USA, 2019. Association for Computing Machinery.

[4] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[5] Nadiia Chepurko, Ryan Marcus, Emanuel Zgraggen, Raul Castro Fernandez, Tim Kraska, and David Karger. ARDA: Automatic relational data augmentation for machine learning. March 2020.

[6] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

[8] Asir Antony Danasingh, Suganya Balamurugan, and JEBAMALAR LEAVLINE EPIPHANY. Literature review on feature selection methods for high-dimensional data. *International Journal of Computer Applications*, 136, 02 2016.

[9] Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. Cocoa: Correlation coefficient-aware data augmentation. In *EDBT*, pages 331–336, 2021.

[10] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1001–1012. IEEE, 2018.

[11] Kunihiko Fukushima, Sei Miyake, and Takayuki Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (5):826–834, 1983.

[12] Mark A Hall. Correlation-based feature selection of discrete and numeric class machine learning. 2000.

[13] Xing Hao, Guigang Zhang, and Shang Ma. Deep learning. *International Journal of Semantic Computing*, 10(03):417–439, 2016.

[14] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pages 2146–2153. IEEE, 2009.

[15] Alan Jović, Karla Brkić, and Nikola Bogunović. A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1200–1205. Ieee, 2015.

[16] Yoshiaki Koshiba and Shigeo Abe. Comparison of l1 and l2 support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 2054–2059 vol.3, 2003.

[17] Tim Kraska. Northstar: An interactive data science system. *Proc. VLDB Endow.*, 11(12):2150–2164, aug 2018.

[18] Arun Kumar, Jeffrey Naughton, Jignesh M Patel, and Xiaojin Zhu. To join or not to join? thinking twice about joins before feature selection. In *Proceedings of the 2016 International Conference on Management of Data*.

[19] Vipin Kumar. Feature selection: A literature review. *The Smart Computing Review*, 4, 06 2014.

[20] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. Dada: differentiable automatic data augmentation. *arXiv preprint arXiv:2003.03780*, 2020.

[21] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391. IEEE, 1995.

[22] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[23] Massimiliano Pontil and Alessandro Verri. Properties of support vector machines. *Neural Computation*, 10(4):955–974, 1998.

[24] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[25] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? *Neural computation*, 16(5):1063–1076, 2004.

[26] Zeyuan Shang, Emanuel Zgraggen, Benedetto Buratti, Ferdinand Kossmann, Philipp Eichmann, Yeounoh Chung, Carsten Binnig, Eli Upfal, and Tim Kraska. Democratizing data science through interactive curation of ml pipelines, 2019.

[27] Mansour Sheikhan, Mahdi Bejani, and Davood Gharavian. Modular neural-svm scheme for speech emotion recognition using anova feature selection method. *Neural Computing and Applications*, 23(1):215–227, 2013.

[28] Fengxi Song, Dayong Mei, and Hongfeng Li. Feature selection based on linear discriminant analysis. In *2010 international conference on intelligent system design and engineering application*, volume 1, pages 746–749. IEEE, 2010.

[29] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

[30] Jorge R Vergara and Pablo A Estévez. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1):175–186, 2014.

[31] Xiaowen Chu Xin He, Kaiyong Zhao. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

[32] Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *Journal of machine learning research*, 10(7), 2009.

[33] Zhichang Zhang, Dan Liu, Minyu Zhang, and Xiaohui Qin. Combining data augmentation and domain information with tener model for clinical event detection. *BMC medical informatics and decision making*, 21(9):1–12, 2021.

A Feature Selection Analysis Results

This appendix presents a summary of the results related to the preliminary study for the design of the ADA algorithm. Further analysis was performed, but it was decided to be summed up enough to support the results and design of section 4. For each database analysed, the following is considered:

- The balance of the the target variable, i.e., the percentage of the most frequent class.
- The data type of the variable.
- The variance for numerical variables and frequency of the most common category for categorical variables.
- The mean for numerical variables and the frequency of the least common category for the categorical variables
- The range for the numerical variables and the number of categories for the categorical variables.
- The correlation with the class. Note that this value is presented in absolute value.
- The correlation with the one-dimensional projection of the linear discriminant analysis (LDA). Note that this value is presented in absolute value.
- The score (F-statistic) of an ANOVA test about the class-conditional distributions.
- The score of mutual information.
- The ranking in the Recursive Feature Elimination process.
- The ranking of coefficients of each feature (after normalizing all the data, only for this purpose) after stringent L1 regularisation is introduced.

NB: The actual name of the variables has been dropped deliberately to make the analysis process as unbiased as possible, as well as to ease up the analysis process.

A.1 NBA

- *Balance of the class*: 0.6201

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	integer	303,9441	60,4141	71	0,396833	0,39269	144,7859	0,089153	10	11
v2	float	69,0223	17,6246	37,8	0,317805	0,309631	82,5131	0,113027	11	9
v3	float	18,9882	6,8014	27,5	0,315981	,0.306255	85,7079	0,084976	8	1
v4	float	2,8344	2,6291	9,9	0,317594	0,309093	88,0165	0,078671	9	5
v5	float	12,9132	5,8852	19	0,29266	0,284653	73,5343	0,072585	13	18
v6	float	37,6711	44,1694	49,9	0,227134	0,22307	38,2283	0,02607	12	12
v7	float	0,1472	0,2476	2,3	0,036619	0,38126	0,6689	0,015273	1	19
v8	float	1,1275	0,7791	6,5	0,01811	0,017739	0,0676	0,018494	3	7
v9	float	256,7339	19,3081	100	0,003411	0,008501	0,015	0	20	8
v10	float	0,9747	1,2976	7,7	0,296841	0,282483	74,1989	0,087181	18	17
v11	float	1,7503	1,8219	10,2	0,296089	0,280139	75,8123	0,052375	14	16
v12	float	111,9042	70,3002	100	0,106706	0,113391	5,7974	0,023031	15	10
v13	float	0,6039	1,0094	5,3	0,293307	0,283155	84,4512	0,05913	7	20
v14	float	1,8496	2,0257	9,4	0,284677	0,268262	66,5071	0,05388	17	3
v15	float	4,2344	3,0344	13,6	0,299406	0,285177	79,9703	0,05913	4	2
v16	float	2,1643	1,5505	10,6	0,175353	0,174	21,5774	0,017112	6	13
v17	float	0,1679	0,6185	2,5	0,229811	0,229219	45,616	0,025031	16	6
v18	float	0,1841	0,3685	3,9	0,210114	0,203248	34,0064	0,050345	5	15
v19	float	0,5221	1,1935	4,3	0,272348	0,260543	61,9316	0,064692	2	4
v20	string	0,0067	0,0007	1294	0,034918	0,039523	1,5616	0,022431	19	14
Time	0s	16ms	39ms	18ms	26ms	646ms	2,32s	9,26s	39min	930ms

Table 7: Summary of results for the *NBA* database.

A.2 Breast Cancer

- *Balance of the class*: 0.6274

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	float	2.069,431583	40,3371	535,3980	0,548236	0,585843	173,78	0,313508	22	23
v2	float	123.843	654,8891	2.357,5000	0,708984	0,787642	381,77	0,343658	29	16
v3	float	324.167	880,5831	4.068,8000	0,733825	0,819032	462,54	0,420501	30	22
v4	float	0,000321	0,0255	0,1331	0,292999	0,288207	33,93	0,050548	18	2
v5	float	0,002789	0,1043	0,3260	0,596534	0,669109	192,88	0,249789	10	3
v6	float	0,024755	0,2543	1,0307	0,590998	0,667218	202,89	0,246165	3	8
v7	float	0,000038	0,0118	0,0528	0,408042	0,430819	67,90	0,158341	26	12
v8	float	0,001506	0,0489	0,2012	0,776614	0,881619	488,05	0,431199	11	20
v9	float	0,004321	0,1146	0,2910	0,793566	0,893823	587,86	0,470558	2	29
v10	float	0,000911	0,0319	0,3960	0,253730	0,244785	51,23	0,167841	25	13
v11	float	0,006355	0,0888	0,4268	0,696360	0,774149	344,84	0,401014	7	19
v12	float	0,043524	0,2722	1,2520	0,659610	0,736408	335,88	0,377748	1	25
v13	float	0,000007	0,0038	0,0289	0,077972	0,047277	2,85	0,028756	23	1
v14	float	0,000050	0,0628	0,0475	0,012838	0,016211	0,15536	0,060283	24	5
v15	float	0,000326	0,0839	0,1525	0,323872	0,354628	49,73	0,083175	17	9
v16	float	4,087896	2,8661	21,2230	0,556141	0,599607	163,21	0,223034	15	27
v17	float	590,440480	91,9690	144,7100	0,742636	0,829211	466,68	0,37646	20	14
v18	float	1.129,130847	107,2612	200,7900	0,782914	0,878088	617,62	0,454524	21	21
v19	float	0,076902	0,4052	2,7615	0,567134	0,611973	176,46	0,223479	14	30
v20	float	12,418920	14,1273	21,1290	0,730029	0,813591	438,58	0,337123	8	15
v21	float	23,360224	16,2692	28,1100	0,776454	0,868490	598,08	0,427867	6	24
v22	float	0,000009	0,0070	0,0294	0,067016	0,065855	5,10	0,045428	27	11
v23	float	0,000198	0,0964	0,1108	0,358560	0,425037	47,29	0,084701	13	7
v24	float	0,000521	0,1324	0,1514	0,421465	0,487610	78,02	0,089119	9	17
v25	float	0,000068	0,0205	0,0711	0,006522	0,005351	0,07752	0,002838	28	4
v26	float	0,000752	0,1812	0,1980	0,330499	0,394430	45,98	0,068949	12	10
v27	float	0,003828	0,2901	0,5073	0,416294	0,498894	88,67	0,117186	4	28
v28	float	0,304316	1,2169	4,5248	0,008303	0,003655	0,052371	0,001579	5	6
v29	float	18,498909	19,2896	29,5700	0,415185	0,471578	77,60	0,100085	19	18
v30	float	37,776483	25,6772	37,5200	0,456903	0,529663	106	0,174918	16	26
Time	0,6274	1,94ms	4ms	1,73ms	3,8ms	19ms	250ms	388ms	1,58s	89ms

Table 8: Summary of results for the *Breast Bancer* database.

A.3 Bank Note

- *Balance of the class*: 0.5554

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	float	8	0,433735	13,8667	0,72	0,77	834	0,348	1	2
v2	float	34,4457	1,922353	26,7247	0,44	0,48	180	0,217	3	1
v3	float	18,5763	1,3976	23,2135	0,16	0,17	12	0,143	2	3
v4	float	4,4143	-1,1917	10,9977	0,023	0,04	1	0	4	4
Time	0s	4,99ms	3,03ms	2ms	94,7ms	23,4ms	400ms	200ms	52,9ms	7,98ms

Table 9: Summary of results for the *Bank Note* database.

A.4 Cleveland: Heart disease

- *Balance of the class*: 0.5387

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	integer	81,8977	54,5421	48	0,23	0,2655	10,59	0	12	13
v2	integer	0,2194	0,6767	1	0,28	0,4745	19,76	0,0436	8	6
v3	integer	0,9309	3,1582	3	0,41	0,4417	25,66	0,1366	3	3
v4	float	315,51	131,693	106	0,15	0,2179	1,51	0,0045	11	9
v5	float	2703,7	247,3501	438	0,08	0,1433	0,09	0,0145	13	10
v6	integer	0,124	0,1448	1	0,003	0,0476	0,15	0	5	7
v7	float	0,9899	0,9966	2	0,17	0,1704	6,74	0	9	12
v8	integer	526,315	149,5993	131	0,42	0,5213	38,3	0,0844	10	5
v9	float	0,22	0,326599	1	0,42	0,4837	38,01	0,0912	2	11
v10	integer	1,3598	1,0555	6,2	0,42	0,5492	34,56	0,088	6	4
v11	integer	0,8816	1,602694	2	0,33	0,2756	16,19	0,0269	7	8
v12	integer	3,7582	0,6767	3	0,46	0,71	63,04	0,2065	1	2
v13	float	0,2493	4,7306	4	0,53	0,7074	65,61	0,1524	4	1
Time	0s	982μms	996μs	2,99ms	219ms	13ms	306ms	344ms	327ms	3,98ms

Table 10: Summary of results for the *Cleveland's Heart Disease* database.

A.5 Mushroom Poisonousness

- *Balance of the class*: 0.5179

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	string	0,450025	0,000492	6	0,182567	0,167103	188,44	0,035425	19	17
v2	string	0,399311	0,000492	4	0,195415	0,198264	193,64	0,019441	20	18
v3	string	0,281142	0,001969	10	0,133683	0,132127	88,95	0,024554	21	13
v4	string	0,584441	0,415559	2	0,50153	0,506122	1.715,38	0,143658	14	2
v5	string	0,434269	0,004431	9	0,785557	0,793071	7.624,06	0,36755	1	1
v6	string	0,974151	0,025849	2	0,1292	0,125576	76,73	0,0171	15	21
v7	string	0,838503	0,161497	2	0,348387	0,342734	673,64	0,073896	9	6
v8	string	0,690793	0,309207	2	0,540024	0,546356	2.044,04	0,168498	4	22
v9	string	0,212703	0,002954	12	0,538808	0,544695	1.995,61	0,186201	12	19
v10	string	0,567208	0,432792	2	0,102019	0,102748	49,77	0	16	14
v11	string	0,464796	0,023634	5	0,302001	0,307199	505,83	0,034815	8	4
v12	string	0,637125	0,002954	4	0,587658	0,592629	2.540,81	0,204639	6	8
v13	string	0,607582	0,034958	4	0,573524	0,573093	2.340,22	0,189801	3	15
v14	string	0,549483	0,000985	9	0,266489	0,27087	386,81	0,044257	17	9
v15	string	0,539636	0,002954	9	0,266489	0,265096	384	0,053011	7	11
v16	string	1	1	1	NaN	NaN	0	0,002533	22	20
v17	string	0,975382	0,000985	4	0,140541	0,138599	91,97	0,011803	18	12
v18	string	0,921713	0,004431	3	0,2046	0,20213	208,20	0,019751	13	16
v19	string	0,488429	0,004431	5	0,540469	0,540262	2.056,02	0,15389	10	3
v20	string	0,293944	0,005908	9	0,490229	0,487734	1.546,09	0,157412	2	7
v21	string	0,497292	0,041851	6	0,443722	0,43943	1.268,18	0,101257	5	10
v22	string	0,387494	0,023634	7	0,323346	0,31769	574	0,054385	11	5
Time	0s	312ms	294ms	152ms	130ms	996ms	373ms	4,72s	8,77s	487ms

Table 11: Summary of results for the *Mushroom Poisonousness* database.

A.6 Stroke

- *Balance of the class*: 0.9513

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	integer	4,478E+8	36617,83	72873	0,006388	0,038614	0,5702	0	11	6
v2	integer	511,33	43,2266	82	0,245257	0,843694	183,5844	0,041657	4	11
v3	integer	0,0879	0,0974	1	0,127904	0,420226	74,5096	0,005363	5	10
v4	integer	0,0511	0,054	1	0,134914	0,467673	72,2818	0,002013	6	9
v5	float	2050,6	106,1476	216,62	0,131945	0,43715	53,9436	0,00663	10	7
v6	float	61,6863	28,8932	87,3	0,043374	0,150982	6,2147	0,008303	9	8
v7	string	0,5859	0,0002	3	0,009117	0,023946	0,9638	0,005922	8	2
v8	string	0,6561	0,3439	2	0,10834	0,346437	28,1711	0,004897	2	3
v9	string	0,5724	0,0043	5	0,083869	0,203523	19,4319	0,006628	3	1
v10	string	0,508	0,4919	2	0,015458	0,05291	0,0991	0	7	5
v11	string	0,3702	0,1544	4	0,064556	0,191425	12,3503	0,0013	1	4
Time	0s	5ms	34ms	6ms	33ms	61ms	367ms	1,06s	357s	63s

Table 12: Summary of results for the *Stroke* database.

A.7 US census: Income Estimation

- *Balance of the class*: 0.7607

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	integer	187,9780	38,6436	73	0,2303	0,3827	1604,7181	0,0693	9	10
v2	integer	1,115E+10	189664,1	1478115	0,0063	0,0077	1,9009	0,0228	10	9
v3	integer	6,6099	10,0781	15	0,3326	0,5539	3683,8590	0,0622	14	13
v4	integer	5,553E+7	1079,0676	99999	0,2231	0,3651	1616,6350	0,0810	13	14
v5	integer	162412,7	87,5023	4356	0,1475	0,2372	592,4670	0,0333	11	11
v6	integer	153,54	40,4224	98	0,2276	0,3811	1618,4490	0,0411	12	12
v7	string	0,7515	0,0002	8	0,1396	0,2375	621,9762	0,0096	4	4
v8	string	0,3231	0,0016	16	0,1803	0,3062	1041,2106	0,0172	1	1
v9	string	0,4582	0,0007	7	0,4459	0,7363	7203,6518	0,1019	2	2
v10	string	0,1839	0,0003	14	0,2109	0,3397	1440,5742	0,0210	6	3
v11	string	0,4037	0,0308	6	0,4037	0,6650	5655,6206	0,0800	3	5
v12	string	0,8550	0,0096	5	0,0904	0,1300	225,0978	0,0064	5	6
v13	string	0,6685	0,3315	2	0,2146	0,3599	1384,4710	0,0303	8	8
v14	string	0,9186	0,0000	41	0,0627	0,1062	116,1263	0,0074	7	7
Time	0s	23ms	3,4s	20ms	106ms	1,78s	433ms	21.7s	111min	73s

Table 13: Summary of results for the *US Census: Income Estimation* database.

A.8 Credit Card Fraud Detection

- *Balance of the class*: 0.9982

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	float	3,836	3,92E-09	58,8624	0,1	0,130599	1459,438	0,001867	30	30
v2	float	2,726	5,68E-10	94,7735	0,091	0,118838	1264,338	0,003112	13	5
v3	float	2,299	-8,76E-09	57,7081	0,19	0,262844	5808,167	0,004641	20	26
v4	float	2,004	2,81E-09	22,5585	0,13	0,18818	2842,29	0,004623	26	29
v5	float	1,905	-1,55E-09	148,5450	0,095	0,120565	1346,811	0,002223	18	23
v6	float	1,775	2,04E-09	99,4621	0,044	0,064608	349,7418	0,002095	17	19
v7	float	1,53	-1,70E-09	164,1467	0,19	0,243015	5494,75	0,00385	16	7
v8	float	1,426	-1,89E-10	93,2239	0,02	0,043969	151,4849	0,001735	9	12
v9	float	1,207	-3,15E-09	29,0291	0,098	0,133149	1535,638	0,004078	12	2
v10	float	1,185	1,77E-09	48,3334	0,22	0,291787	8219,931	0,007295	2	1
v11	float	1,042	9,29E-10	16,8164	0,15	0,214261	4028,27	0,006186	25	24
v12	float	0,998	-1,80E-09	26,5321	0,26	0,361916	12044,93	0,007157	3	6
v13	float	0,991	1,67E-09	12,9188	0,0046	0,002311	1,1408	0,000084	19	17
v14	float	0,919	1,48E-09	29,7411	0,3	0,421284	16481,97	0,007683	6	9
v15	float	0,838	3,50E-09	13,3767	0,0042	0,006586	6,4808	0,000105	11	15
v16	float	0,768	1,39E-09	31,4450	0,2	0,262656	6923,064	0,005851	8	10
v17	float	0,721	-7,47E-10	34,4163	0,33	0,439002	20152,34	0,007937	1	3
v18	float	0,703	4,26E-10	14,5398	0,11	0,145153	2123,348	0,003759	5	11
v19	float	0,663	9,02E-10	12,8055	0,035	0,046895	222,0804	0,001073	27	27
v20	float	0,594	5,13E-10	93,9186	0,02	0,025674	48,2164	0,001061	4	4
v21	float	5,395	1,47E-10	62,0332	0,04	0,04365	393,957	0,002403	21	25
v22	float	0,526	8,04E-10	21,4362	0,0008	0,004081	0,6255	0,000082	24	22
v23	float	0,389	5,28E-10	67,3361	0,0027	0,011617	14,7942	0,00052	22	20
v24	float	0,366	4,46E-09	7,4212	0,007	0,008965	7,8896	0,000441	23	21
v25	float	0,271	1,43E-09	17,8150	0,0033	0,001778	2,3675	0,000274	10	13
v26	float	0,232	1,70E-09	6,1219	0,0045	0,008524	7,1662	0,000361	15	16
v27	float	0,162	-3,66E-10	54,1779	0,018	0,02187	92,7381	0,002096	7	8
v28	float	0,108	-1,22E-10	49,2779	0,01	0,01269	20,7864	0,001442	14	14
v29	float	62.560	8,83E+07	25.691,16	0,006	0,007552	5,0882	0,001085	29	28
Time	0s	30,9ms	19,9ms	32,9ms	1,29s	2,3s	513ms	62s	143min	116s

Table 14: Summary of results for the *Credit Card Fraud Detection* database.

A.9 Water Potability

- *Balance of the class*: 0.514

	dtype	Variance / Max. f.	Mean / Min. f.	Range / # cat.	Corr. class	Corr. LDA	Score ANOVA	Score Mutual Info.	Ranking RFE	Ranking Regularization
v1	float	2,5418	7,0808	14,00	0,003556	0,173632	0,7824	0,03	2	5
v2	float	1081,079	196,3695	275,69	0,013837	0,287703	0,1014	0,0101	7	3
v3	float	76887830	22.014,0925	60.906,25	0,033743	0,652519	3,5531	0,0024	9	2
v4	float	2,5061	7,1223	12,78	0,023779	0,25941	0,7498	0	1	4
v5	float	1715,35	333,7758	352,03	0,023577	0,397129	4,452	0,0628	6	7
v6	float	6532,52	426,2051	571,86	0,008128	0,043403	0,8327	0,0003	8	6
v7	float	10,94	14,2850	26,10	0,030001	0,4611	3,2008	0	3	8
v8	float	261,63	66,3963	123,26	0,00713	0,282378	1,3685	0,0109	5	1
v9	float	0,6089	3,9668	5,29	0,001581	0,071483	0,9332	0,002	4	9
Time	0,514	2,9ms	1ms	1ms	45micros	29ms	214ms	344ms	2,3s	622ms

Table 15: Summary of results for the *Water Potability* database.