Predicting Customer’s Chrun

First Author1, Second Author2, Third Author3 & Fourth Author4

1first author student number and e-mail contact

2second author student number and e-mail contact

3third author student number and e-mail contact

4fourth author student number and e-mail contact

Abstract**:** maximum of 250 words, font Times New Roman, size 10, line spacing 1.0

Keywords: maximum three keywords separated by semicolon

Statement of Contribution: clearly state the contributions of each group member to the project

*Table of Contents*

[Abstract I](#_Toc531539791)

[Keywords I](#_Toc531539792)

[Statement of Contribution I](#_Toc531539793)

[List of Figures III](#_Toc531539794)

[List of Tables III](#_Toc531539795)

[I. Introduction 1](#_Toc531539796)

[II. Description 1](#_Toc531539797)

[a. Data 1](#_Toc531539798)

[b. Problem 3](#_Toc531539799)

[III. Preprocessing 4](#_Toc531539800)

[IV. Modelling 8](#_Toc531539801)

[a. Decision Tree Classifier 8](#_Toc531539802)

[b. Random Forest Classifier 8](#_Toc531539803)

[c. Support Vector Machine (SVM) 9](#_Toc531539804)

[d. Naive Bayes Classifiers 9](#_Toc531539805)

[e. Logistic Regression (PCA/LDA) 9](#_Toc531539806)

[V. Results and Discussion 10](#_Toc531539807)

[VI. Model Tuning 10](#_Toc531539808)

[VII. Conclusions 11](#_Toc531539809)

[VIII. Acknowledgements IV](#_Toc531539810)

[IX. References V](#_Toc531539811)

### List of Figures

[Figure 1 7](file:////Users/fruechtnicht/Library/Containers/com.apple.mail/Data/Library/Mail%20Downloads/FDE18E52-969D-410B-BCDA-1E5402850CDA/Template.docx#_Toc531534914)

[Figure 8: Decision Tree 12](#_Toc531534915)

[Figure 9: Feature importance 13](#_Toc531534916)

### List of Tables

[Table 1: Dataset 3](#_Toc531539661)

# Introduction

This report is part of the syllabus the course Introduction to Programming in the Master of Data Science and Advanced Analytics at Nova IMS and has the purpose to demonstrate programming skills in Python in a Data Science related manner.

The data set used for our project, further described later (cf. a) contains sample data of bank customers of the year Y0. One of the variables is the binary variable “exited” indicating whether a customer is or no longer is a customer after time T1.

Our goal is to explore some machine learning techniques in order to predict if a customer is going to exit or not that is predicting the binary variable.

The remainder of this report is structured as following: first, the data set and problem is described (cf. II. Data and Problem description), then the data is preprocessed (cf. III. Preprocessing) and different prediction models are introduced (cf. IV Modelling). Afterwards the results of the models are compared (cf. V. Results and Discussion). Then the two bests are being optimised (cf. VI. Model Tuning). In the end a conclusion is drawn (cf. VII. Conculsion).

# Description

In this section the dataset is described in terms of each variable. Also, the problem is described.

## Data

The main and only dataset used for this project contains detailed information of a random sample of customers from a European bank. The data in question was taken from the bank’s database on the 31st of December 2017. Meaning the data is static and of very high quality as it will be seen on the following sections of the report.

Six months after the initial data collection, it was recorded if the customers continued to be in business with the bank. This additional data will be used as the independent variable.

The data previously described can be found on the file *Churn\_Modelling.csv*. The columns contained are the following:

|  |  |  |
| --- | --- | --- |
| **Field** | **Format** | **Description** |
| **RowNumber** | int64 | The row number (index from database table). Numeric value in the range of <1;1000> |
| **CustomerId** | int64 | The unique Customer ID ranging between 15565701 and 15815690. This is the most granular attribute in the dataset. |
| **Surname** | object | Customer’s surname. |
| **CreditScore** | int64 | Credit Score of a customer ranging between 300 and 850. A higher implies a higher ‘credit worthiness’. |
| **Geography** | object | Branch location that a customer belongs to. There are three possible discrete values: ‘France’, ‘Germany’ or ‘Spain’. |
| **Gender** | object | Gender of the customer. There are two possible Boolean values: ‘Male’ or ‘Female’}. |
| **Age** | int64 | Customer’s age on the day the data was collected (31/12/17). This age will also be applicable 6 months later. The ages range between 18 and 92. |
| **Tenure** | int64 | Number of years a customer has been with the business. The years range between 0 and 10. |
| **Balance** | float64 | Average amount of money in the bank account on December 2017. The Average was calculated by summing account balance for each day in the month and divided by number of days in that specific month. The Sum of the average amount can be significantly different from sum of the actual account balance of customers but for static data using only one-day info can be very misleading. |
| **NumOfProducts** | int64 | The number of products a customer has with the bank. The number ranges between 1 and 4. |
| **HasCrCard** | int64 | Boolean value representing if a customer has a credit card or not. 1 if the customer has and 0 if the customer doesn’t. The dataset contains 7055 customers who possess a credit card (value 1) and 2945 who don’t (value 0). |
| **IsActiveMember** | int64 | Boolean value representing if a customer is active or not. 1 if the customer is and 0 if the customer isn’t. |
| **EstimatedSalary** | float64 | Estimated annual salary of the customer. The salaries in USD range between 11.58 and 199992.48. |
| **Exited** | int64 | Boolean value representing if a customer left the bank after 6 months or not. 1 if yes and 0 if no. |

Table 1: Dataset

## Problem

For the purpose of this report the following fictional problem is introduced:

The marketing department of a bank has developed a campaign which prevents customers from leaving the company. This campaign is to be executed a data set from the current year Y1.

The goal for us, as aspiring Data Scientists, is to maximize the return on investment of the marketing campaign, that is predicting which customer is going to leave and which is not.

Later, we are going to use the campaign information to evaluate the performance of different prediction models.

The marketing department specified the following parameters of the campaign:

* Cost of applying the campaign per customer: 150€
* Average customer value till T1: 450€,
* Average customer value from T1 till T2: 650€
* Moreover, applying the campaign to a customer which had not been leaving still has some positive effect on the customer’s likelihood to leave the bank in the future. This effect is quantified as: 10€
* Assumption at time T0: no costumer leaves till T1 (for simplicity)
* Customers who were prevented from leaving will stay till T2

Four cases can be derived from the above for the training data set:

Figure 1: Problem peculiarities

If a customer leaves till T1 and no campaign is applied:

**loss of 450€**

(money expected to be gained by T1)

If a customer does not leave and no campaign is applied:

**no loss or gain**

(everything happened as excepted till T1

If a customer does not leave because the campaign is applied:

**gain of 500€**

(no loss of 450€ by T1, a gain of 650€ because the customer is going to stay till T2, a loss for applying the campaign of 150€)

If a customer does not leave and was not going to leave but the campaign is applied:

**loss of 140€**

(no loss of 450€ by T1, a loss for applying the campaign of 150€, a gain for the positive effect of the campaign of 10€)

Customer does not leave

Customer leaves

Customer does not leave

Customer leaves

*Model prediction (applying the campaign or not)*

*Reality*

# Preprocessing

In the preprocessing part we perform an exploratory data analysis in order to gain more insights on the dataset and justify data transformation performed on the dataset, preparing it for the modelling part.

Handling null values

As previously mentioned the analysed dataset presents high quality which can be proved with no missing values (Figure 2 and 3). We are then able to skip the null handling part focus on other aspects of data preprocessing.

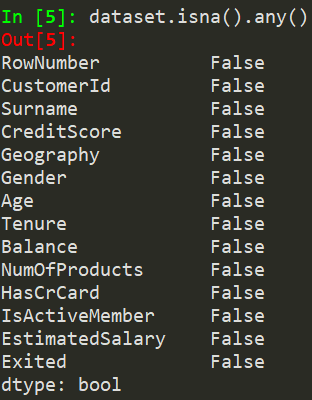


Figure 2: Missing value check

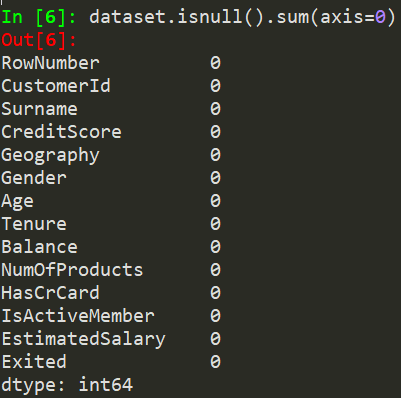


Figure 3: Null value check

Feature selection

As a second step of data preprocessing (after identifying null values), non-useful for the problem columns have been dropped ([Surname], [CustomerId], [RowNumber]). Afterwards, histograms of non-binary columns have been created for insights about data distribution. (Figure 4).

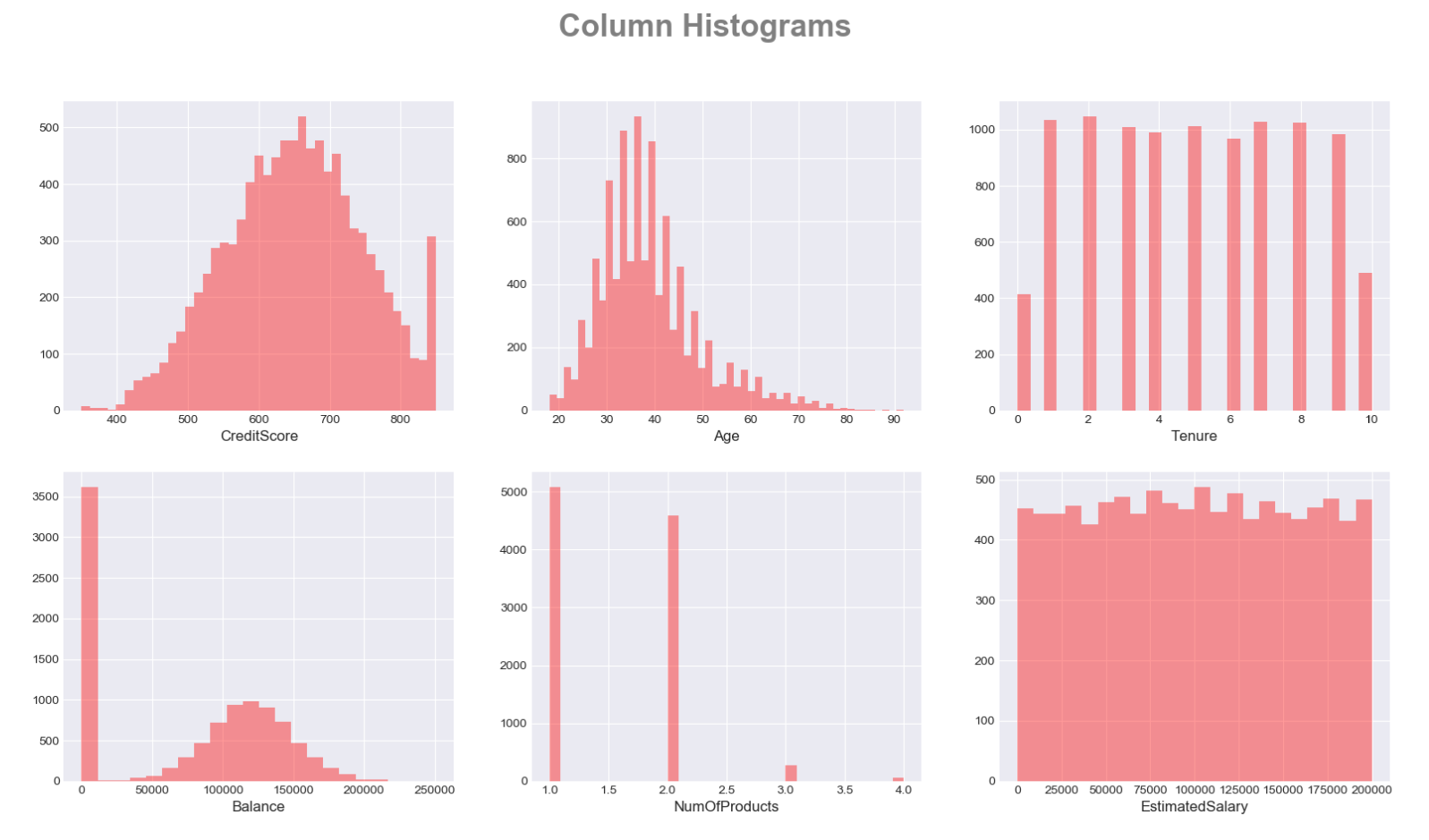


Figure 4: Column Histograms

Handling skewness

The third step of data preprocessing includes logarithmic transformation of columns with skewed distribution (i.e. [‘Age’]). As a result, the distribution of transformed column is more similar to normal, which is proved at the boxplot below.

Outliers detection

The forth step of data preprocessing includes outliers identification. For that purpose, boxplots of non-binary columns were plotted (Figure 5). Points above whiskers are considered as extreme values but not outliers, as they present credible values. Data distribution is normal for all variables with the exception of [NumofProducts] and [Balance], where it is appears (former) and left (latter) skewed.

Categorical variables encoding

Categorical variables, such as [Gender] and [Geography] have been one-hot encoded so we are able to include them in the model. To avoid the dummy variable trap we reduce the encoded dummy columns by on only keeping [“Germany”, “Spain”] for Geography [ http://www.algosome.com/articles/dummy-variable-trap-regression.html].

In the following section we will be looking at the categorical variables and at how these correlate with some of the numerical variables analysed previously. This analysis will be conducted with the use of visual representations.

Exited vs Geography and Age

The following graph shows how the age and location of the branch are correlated to the independent variable (figure 10). It can be observed that customers left the business at a similar rate in all three locations. Spain lost the least number of customers and Germany the highest. The bulk of the population that remained a customer of the bank is between the range of 25 and 45 years of age in all branches. It would be expected that the older a customer is the more loyal he would be to a business, but this graph proves the contrary. Ultimately, the graph shows the peak of customers leaving is at around 45 years old in all three countries.

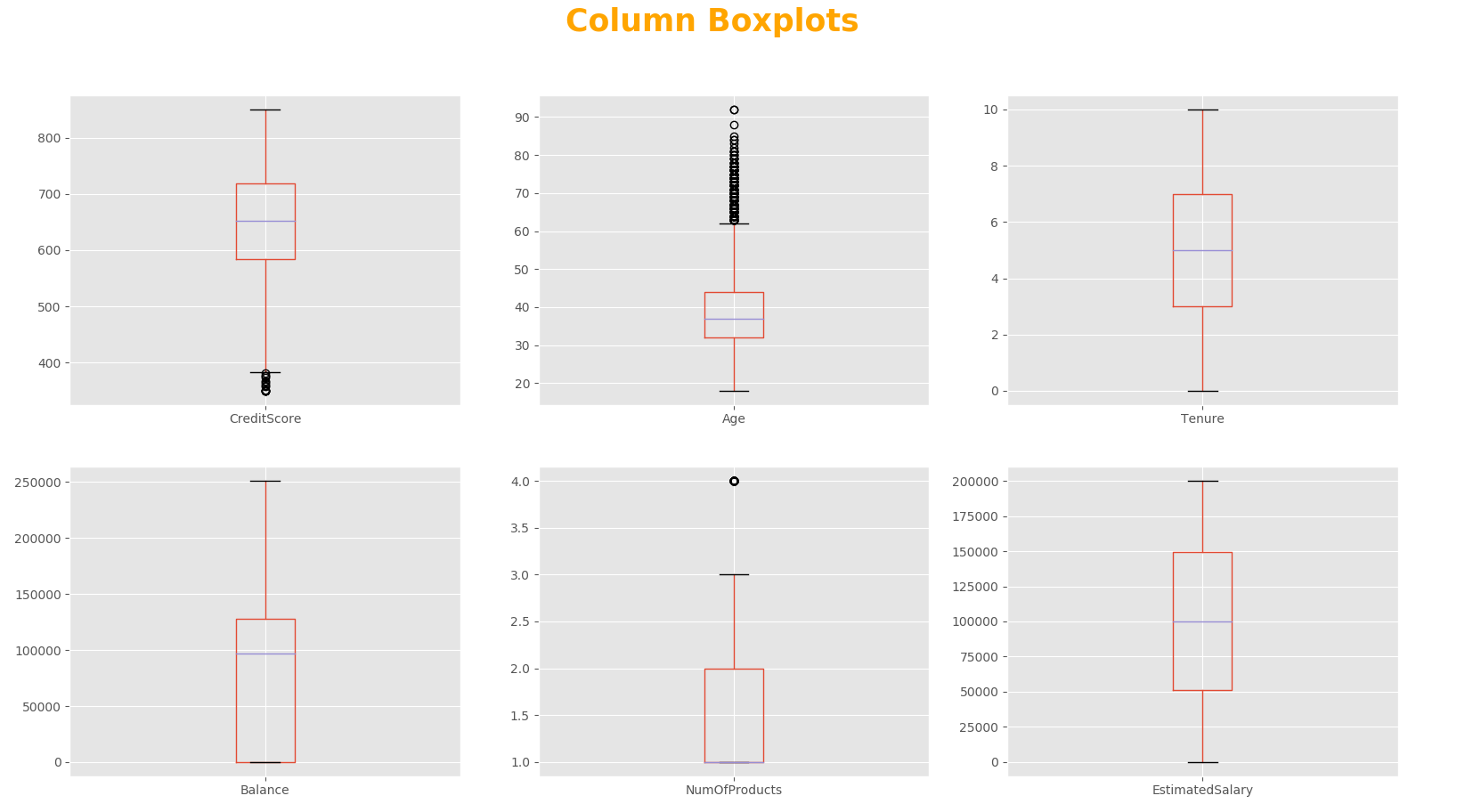


Figure 5: Distribution of values

FIGURE

Exited vs Country

Figure 11 solidifies the point that Germany is the country were customers tend to leave the business the most. As it can be seen France is the country with the lowest number of customers leaving but it does not differ much from Spain.

FIGURE

Exited vs Members Activity

The histogram (Figure 12), shows if both active and inactive customers left the bank or not. It would be to expect that inactive customers have a higher tendency to leave. The histogram shows that the prediction is correct but the difference is minimal.

FIGURE

Pair Plot

We are using a pair plot to see relationships between two variables [https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166]. Figure 13 is a pair plot on the tree variables CreditScore, Age and Balance with the hue of Exited. People who exit are on average older (cf. plot [2][2]). Moreover, Age also has an influence on the distribution of customers Balance (cf. plot [3][2]) and likewise on the CreditScore (cf. plot [1][2]). Other variable distributions (other than Age) do not allow much interpretation (therefore this selection).

FIGURE

Data Correlation

Before modelling it is important to get insights on the correlation between variables. Figure 14 illustrates it graphically and with percentage notation using the Pearson correlation in a heat map. The Balance is influenced by the customers region, a phenomenon which is explained by finance theory [fiancé theory]. Having introduced Dummy variables for location a correlation is normal. The correlation of NumOfProducts and Balance is higher than any other (-30% vs less than +/-5%) however we decided to include in for further modelling as it is not extremely high.

FIGURE

Train test split and scaling

Before Modelling we are splitting the data into a training and a test set. The training set contains a known output and the models learns on this data in order to be generalized to other data later on. The test set is used to test our model’s prediction. [https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6].

Further we scale the train data using the Z score normalization because normalized data is an assumption of many machine learning algorithms (such as SVM, K-nearest neighbors, and logistic regression). Standardization involves rescaling the features such that they have the properties of a standard normal distribution with a mean of zero and a standard deviation of one. Afterwards we apply the same scale on our test set[https://scikit-learn.org/stable/auto\_examples/preprocessing/plot\_scaling\_importance.html]. Scaling is done after splitting for test and train because otherwise the training set would include information of the test set.

# Modelling

In this section we introduce the machine learning models we applied to our test data set.

## Multilinear regression

Regression is one of the most basic and popular approaches to prediction, that’s why it is common to perform linear regression as a first model. Although linear regression is not design to make binary classification you can predict continuous value with it and than use some threshold to classify those value into two category. There are two popular approaches to do this. One is in 0.5, if predicted value is above that level, you predict 1 and when exactly that or less you predict 0. This method seems to be logical from mathematic point of view, but predicted values doesn’t necessary lay in a range <0;1> that is why new approach has been developed, to calculate threshold by taking mean of the range of predicted values, or median if distribution of those data are skewed. Additionally by applying backward elimination strategy you can get more inside about importance of each variable and magnitude of the effect.

## Logistic Regression (PCA/LDA)

Logistic Regression is a classification algorithm that tries to predict the probability of a binary categorical variable, two classes. It is said to have a linear boundary between two possible results [**https://codesachin.wordpress.com/2015/08/16/logistic-regression-for-dummies/**]. This is perfect to test with our data as the dependent variable is if a customer exited the business or not. This module is also best used in large data samples. A disadvantage is that the module assumes independent variables have little multicollinearity and as it can be seen in the data extraction section of the report that is not always true [**https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8**]. A solution to this is to transform the data before it goes through the Logistic Regression model. This can be done with Linear Discriminant Analysis, LDA, and Principal Component Analysis, PCA. These are both techniques used for linear transformation.

PCA is used to reduce the dimensions of a dataset, meaning it tries to use fewer components to explain the outcome. It is usually used to make sure there is no correlation between independent, and like this make the logistic regression model more efficient. [https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c] It also does not take into account class membership meaning it is unsupervised.

LDA on the other hand tries to maximize class separation meaning it is supervised. [**https://www.researchgate.net/post/What\_is\_the\_difference\_between\_PCA\_FA\_and\_LDA**]

[**https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning**/]

## Decision Tree Classifier

The decision tree classifier is a supervised learning method, which can be applied in both regression and classification problems [ISLR]. Given our binary classification problem described in the introduction [cf. XX] applying a decision tree model is possible. In addition, decision trees are able to outperform linear (regression) models if the classification boundary is of non-liner type as linear models won’t be able to capture the decision boundary [ISLR]. An additional benefit is that decision trees are able to perform multi class classification problems [SCIKIT], however given our binary classification problem this is of little use. Another aspect making decision tree model very suitable for our problem is, that decision trees are applicable for continuous and categorical data [DM][cf. data description], which makes their implementation easier. Moreover, they can also handle incomplete data [DM][SCIKIT].

Without going into much detail, decision trees are trained through splitting the data into sub categories according to some criteria (Entropy or Gini index etc.). The trained tree then classifies the data based on its relation to the different splits. This allows a high interpretably of the dataset itself and the functioning of the decision tree, especially because a decision tree is easy to visualize (cf. Figure 8: Decision Tree)[UDEMY][DM].

FIGURE

## Random Forest Classifier

Decision-tree models can create over-complex trees that do not generalize the data well. This is called overfitting. This problem can be overcome by aggregating multiple decision trees e.g. in a random forest using ensemble methods [Breiman, L (2001)][ISLR]. Because the random forest method is based on multiple (random generated) trees it inherits many good characteristics as being applicable in both regression and classification problems and being able to deal with categorical and continuous data [DM][ISLR] making it suitable for our project. Using a large number of trees can often result in dramatic improvements in prediction accuracy, compared to single decision trees at the expense of some loss in interpretation. ISLR [towardsdatascience.com] However, the relative feature importance can be derived (cf. Figure 9: Feature importance) [DM]. The random forest decorrelates trees compared to other aggregated tree models [e.g. bagging]; important when dealing with multiple features which may be correlated. which is why Random forests is considered as a highly accurate and robust method [DATACAMP] [ISLR]. However, it is important to mention that a large number of trees can make the algorithm to slow and ineffective for real-time predictions. While random forests are fast to train, they are slow to create predictions once they are trained [DM][DATACAMP]. A more accurate prediction requires more trees, which results in a slower model, following the no free lunch theorem [Wolpert, Macready (1997)].

FIGURE

## Support Vector Machine (SVM)

A special kind of machine learning algorithm that uses the idea of Maximum Margin between the Support Vectors was used as an attempt to outperform previous algorithms for solution to our classification problem. Support Vector Machines is an extremely popular algorithm because of its efficiency and ability to tackle both: classification and regression problems. Additionally, the algorithm can be useful for both Linearly Separable (hard margin) and Non-linearly Separable (soft margin) data thanks to the proper C parameter tuning [www.towardsdatascience.com]. Moreover, the SVM uses the ‘Kernel Trick’ thanks to which it is able to capture complex relationships between data points without having a problem to perform difficult transformations. This algorithm presents some kind of a different approach to our problem, as SVM is ‘rebellious’ itself since unlike most of the common algorithms, it uses extreme cases, close to the hyperplane (boundary) between the classes for its analysis. The downside of SVM is that the training time takes a relatively long time, but this is a suitable algorithm for the volume of data that we are working with in this case. [Introduction to Statistical Learning] Author: Gareth James. For better results, hyperparameters running was performed by means of Grid Search on 2 of algorithm’s parameters.

## Naive Bayes Classifiers

The Naive Bayes model is not a single classifier but a family of classification algorithms based on the statistical theorem of Bayes [**http://blog.aylien.com/naive-bayes-for-dummies-a-simple-explanation**/]. Shortly explained, Bayes’ theorem uses prior knowledge of an event to more accurately predict the outcome of said event **[https://brilliant.org/wiki/bayes-theorem/].**

Every Naive Bayes classifier works from the same principle that every condition of the dependent variable is unique, not taking any correlation between independent variables into account.

Three types of Naive Bayes classifiers were tested: Multinomial, Bernoulli and Gaussian. Multinomial is used for categorical variables. Bernoulli is also categorical focused but it works with binary variables. Gaussian takes numerical continuous variables as predictors [**https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c**].

To sum up, Naive Bayes is a very fast module due to being very simple and easy to implement. Its only disadvantage, as mentioned above, is the fact that it does not take the correlation between independent variables as a predictive factor. This leads to a loss of valuable information.

## Perceptron or Single-Layer Neural Network

Is one of the machine learining techniques which belongs to supervised methods to perform binary classification. A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

# Results and Discussion

# Model Tuning

# Conclusions

# Acknowledgements

Use this section to acknowledge the contribution of a third party for your project, for instance additional mentoring or data acquisition, or if you are using data from your company for instance.

# References

* Aigiomawu, E. (2018). *Introduction to Matplotlib — Data Visualization in Python*. [online] Heartbeat. Available at: https://heartbeat.fritz.ai/introduction-to-matplotlib-data-visualization-in-python-d9143287ae39.
* Brownlee, J. (2014). *Naive Bayes Classifier From Scratch in Python*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/>.
* Anon, (2018). *Dimension Reduction Techniques (PCA vs LDA) in Machine Learning – Part 2*. [online] Available at: http://www.vfirst.com/blog/techfirst/dimension-reduction-techniques-pca-vs-lda-in-machine-learning-part-2/ .
* Han, J., Kamber, M. and Pei, J. (2012). *Data mining Concepts and Techniques*. Waltham: Morgan Kaufmann.
* James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). *An introduction to statistical learning*.
* Koehrsen, W. (2018). *A Complete Machine Learning Walk-Through in Python: Part One*. [online] Towards Data Science. Available at: https://towardsdatascience.com/a-complete-machine-learning-walk-through-in-python-part-one-c62152f39420?fbclid=IwAR01u2cAkJOhJpJM7QFCH821et3MzS9wI00a8W49mmgDNmYoZQDtd8dZpHM.
* Lei, P. and Koehly, L. (2003). Linear Discriminant Analysis Versus Logistic Regression: A Comparison of Classification Errors in the Two-Group Case. *The Journal of Experimental Education*, 72(1), pp.25-49.
* Ramis, J. (2017). *Logistic Regression PCA vs LDA comparison | Kaggle*. [online] Kaggle.com. Available at: https://www.kaggle.com/junad007/logistic-regression-pca-vs-lda-comparison.
* ZHANG, H. (2005). EXPLORING CONDITIONS FOR THE OPTIMALITY OF NAÏVE BAYES. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(02), pp.183-198.
* (<https://www.datacamp.com/community/tutorials/random-forests-classifier-python#advantages>)
* (https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd)
* Wolpert, D.H., Macready, W.G. (1997), "[No Free Lunch Theorems for Optimization](http://ti.arc.nasa.gov/m/profile/dhw/papers/78.pdf)", *IEEE Transactions on Evolutionary Computation* 1, 67.
* UDEMY <https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/learn/v4/> by Jose Portilla
* SCIKIT <https://scikit-learn.org/stable/modules/tree.html>
* ISLR Gareth James • Daniela Witten • Trevor Hastie Robert TibshiraniAn Introduction to Statistical Learning DOI 10.1007/978-1-4614-7138-7
* DM ISBN 978-0-12-381479-1 (our data mining book)
* Random Forest Breiman, L. Machine Learning (2001) 45: 5. <https://doi.org/10.1023/A:1010933404324>
* [https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166]
* <https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html>].
* [Scikit-learn: Machine Learning in Python](http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html), Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
* All Figures are own visualizations using the Seaborn module (10.5281/zenodo.1313201) and the Mathplot lib module (https://doi.org/10.5281/zenodo.1482099)

This page is left out intentionally