Predicting Customer’s Churn

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](#_Toc531554527)

[Keywords I](#_Toc531554528)

[Statement of Contribution I](#_Toc531554529)

[List of Figures III](#_Toc531554530)

[List of Tables III](#_Toc531554531)

[I. Introduction 1](#_Toc531554532)

[II. Description 1](#_Toc531554533)

[a. Data 1](#_Toc531554534)

[b. Problem 3](#_Toc531554535)

[III. Pre-processing 4](#_Toc531554536)

[IV. Modelling 11](#_Toc531554537)

[a. Multilinear Regression 11](#_Toc531554538)

[b. Logistic Regression (PCA/LDA) 11](#_Toc531554539)

[c. Decision Tree Classifier 12](#_Toc531554540)

[d. Random Forest Classifier 12](#_Toc531554541)

[e. Support Vector Machine (SVM) 13](#_Toc531554542)

[f. Naive Bayes Classifiers 14](#_Toc531554543)

[g. Perceptron or Single-Layer Neural Network 14](#_Toc531554544)

[V. Results 14](#_Toc531554545)

[VI. Model Tuning 16](#_Toc531554546)

[VII. Conclusions 17](#_Toc531554547)

[VIII. Acknowledgements IV](#_Toc531554548)

[IX. References V](#_Toc531554549)

# 

### List of Figures

[Figure 1: Problem peculiarities 3](file:////Users/fruechtnicht/Library/Mobile%20Documents/com~apple~CloudDocs/IP%20Report.docx#_Toc531554550)

[Figure 2: Null value check 4](file:////Users/fruechtnicht/Library/Mobile%20Documents/com~apple~CloudDocs/IP%20Report.docx#_Toc531554551)

[Figure 3: Missing value check 4](file:////Users/fruechtnicht/Library/Mobile%20Documents/com~apple~CloudDocs/IP%20Report.docx#_Toc531554552)

[Figure 4: Column Histograms 5](#_Toc531554553)

[Figure 5: Column Boxplots 6](#_Toc531554554)

[Figure 6: Exited vs Geography and Age 7](#_Toc531554555)

[Figure 7: Exited vs Country 7](#_Toc531554556)

[Figure 8: Number of Exits among Active and Inactive Members 8](#_Toc531554557)

[Figure 9: Pair plot on selected variables 9](#_Toc531554558)

[Figure 10: Heatmap 10](#_Toc531554559)

[Figure 11: Example Decision Tree 12](#_Toc531554560)

[Figure 12: Feature importance 13](#_Toc531554561)

[Figure 13: Monetary score of each model 15](#_Toc531554562)

[Figure 14: Accuracy of each model 15](#_Toc531554563)

[Figure 15: Monetary score of each model after Cross-Validation 16](#_Toc531554564)

[Figure 16: Final Monetary value of each model 16](#_Toc531554565)

### List of Tables

[Table 1: Dataset 2](#_Toc531554566)

**The graphs/visualizations presented on the report were produced by the members of the group.**

# Introduction

This report is part of the syllabus of Introduction to Programming in the Master of Data Science and Advanced Analytics at Nova Information Management School, IMS, and has the purpose of demonstrating Python programming skills in a Data Science environment.

The project was initiated by finding a suitable dataset (cf. a. Data) where we could practice data processing and analytics techniques.

The dataset used contains detailed information of a random sample of customers from a European bank collected on the 31st of December 2017. This seemed like a very good source of high quality data as it will be seen on the pre-processing section of the report (cf. III Pre-processing).

The second step was to create a hypothetical case study that would justify exploring said dataset (cf. b. Problem).

Six months after the initial data collection, it was recorded if any of the sampled customers had left the bank. Given this additional data we were able to build a case around the bank’s data. The customers’ information was to be used as predictive factors, independent variables, to measure how these affect the customers’ decision to leave the business, dependent variable.

The end goal of our project was to run the best independent variables through different machine learning techniques (cf. IV Modelling) to test and compare which model is the most accurate in predicting if a customer leaves the business or not (cf. V. Results).

The two bests models are then optimised for better performance (cf. VI. Model Tuning) and we finally reach a conclusion (cf. VII. Conclusions).

# Description

This section of the report will describe the dataset used and introduce the case study. It is divided into two subparts where the first one provides a brief description of the dataset’s variables and the second part provides solutions to 4 possible scenarios related to the research question.

In this section at first the dataset is described in terms of each variable. then, the problem is described.

## Data

The dataset used for the project contains static data of a sample of bank customers during the year Y0. The dataset is artificial and not confidential. One of the variables is the binary variable “exited” that indicates whether a customer is or is not a customer after time T1.

The data previously described can be found on the file *Churn\_Modelling.csv*. The columns contained on the csv file 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 is also used 6 months later. and ranges 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. |
| **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 bank’s marketing department as developed a marketing campaign that aims to prevent customers from leaving the business. This campaign is to be introduced in year Y1.

Our goal as aspiring Data Scientists, is to maximize the return on investment of the marketing campaign. This would be done by predicting which customers are going to leave and which are not based on the data from Y0.

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: 260€
* Average customer value till T1: 300€,
* Average customer value from T1 till T2: 850€
* 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 300€**

(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 590€**

(no loss of 300€ by T1, a gain of 850€ because the customer is going to stay till T2, a loss for applying the campaign of 260€)

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

**loss of 250€**

(no loss of 300€ by T1, a loss for applying the campaign of 260€, 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*

# Pre-processing

In the pre-processing part we perform 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 3 and Figure 2). We are then able to skip the null handling part focus on other aspects of data pre-processing.

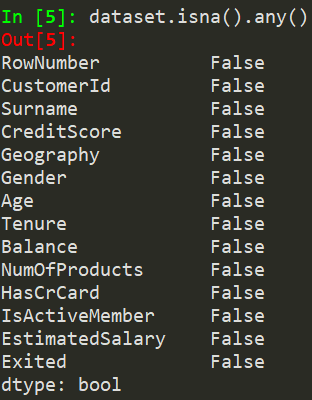
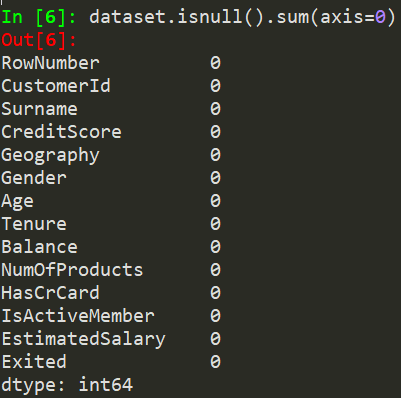


Figure 2: Null value check

Figure 3: Missing value check

Feature selection

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

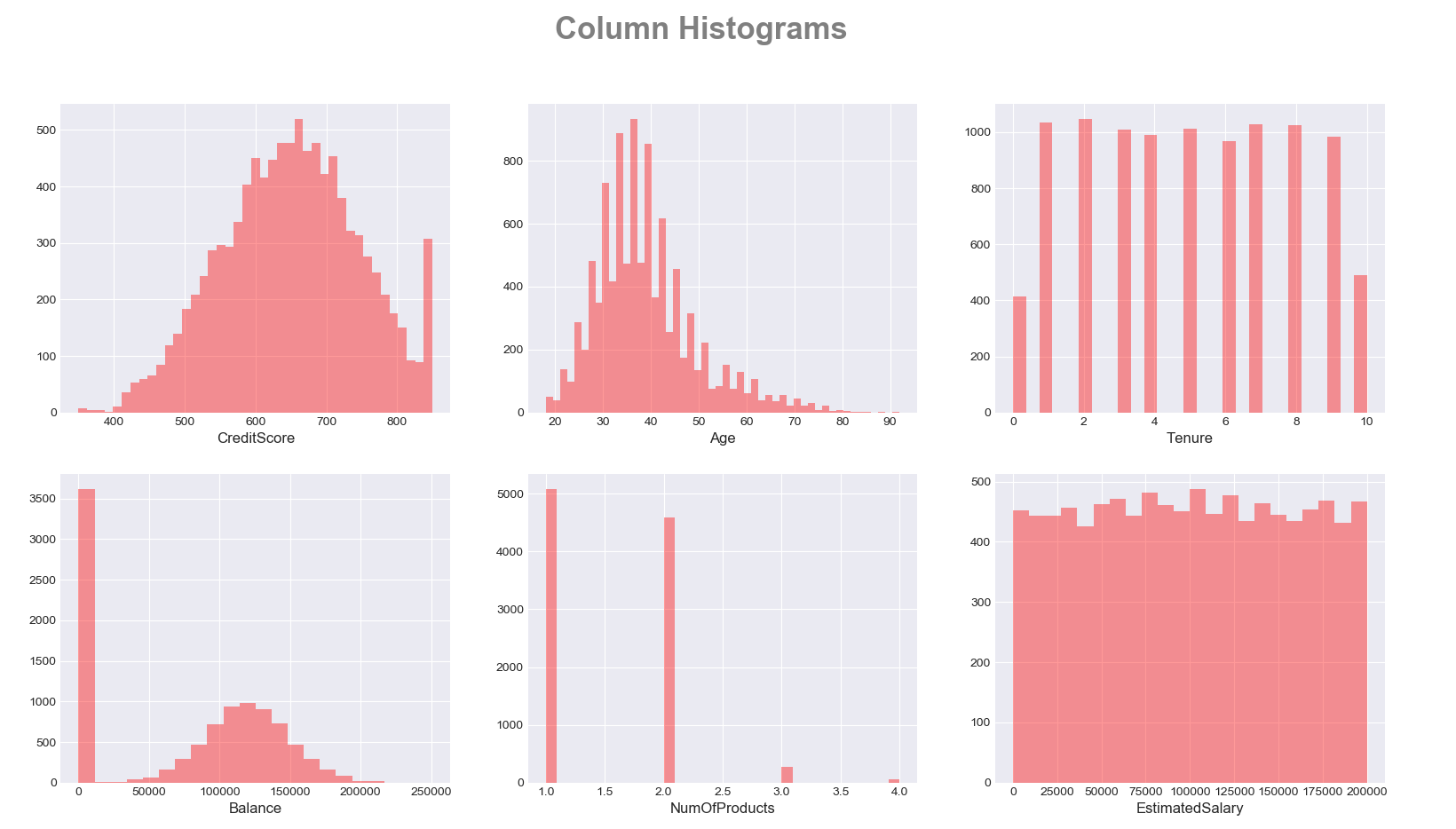


Figure 4: Column Histograms

Handling skewness

The third step of data pre-processing includes logarithmic transformation of columns with skewed distribution (cf. [‘Age’] in Figure 4). As a result, the distribution of transformed column is more similar to normal as can be seen in Figure 5: Column Boxplots.

Outliers detection

The forth step of data pre-processing includes outlier identification. For that purpose, boxplots of non-binary columns were plotted (cf. 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.

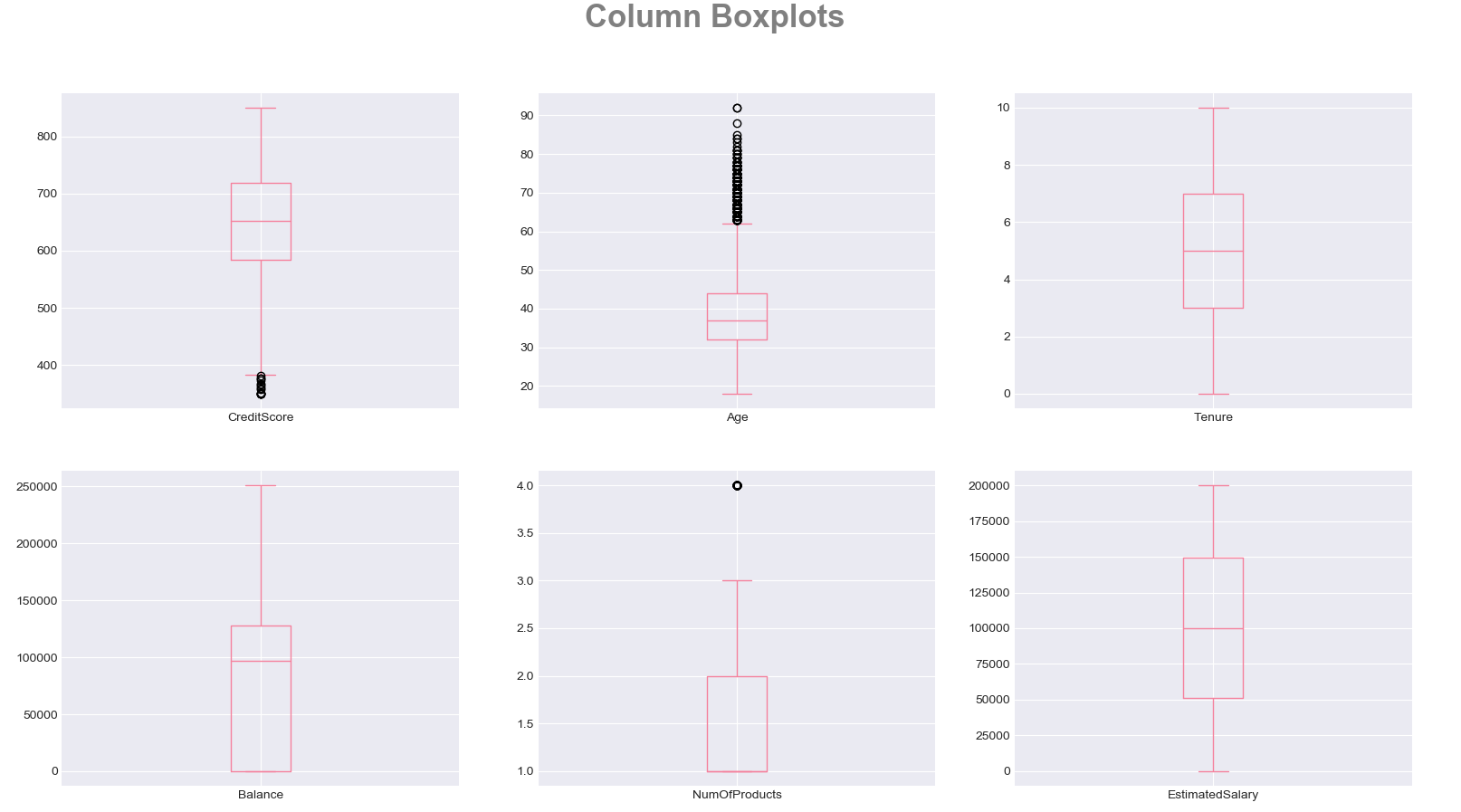


Figure 5: Column Boxplots

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 [6].

In the following, we will be looking at the categorical variables and at how they 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 (cf. Figure 6). 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.

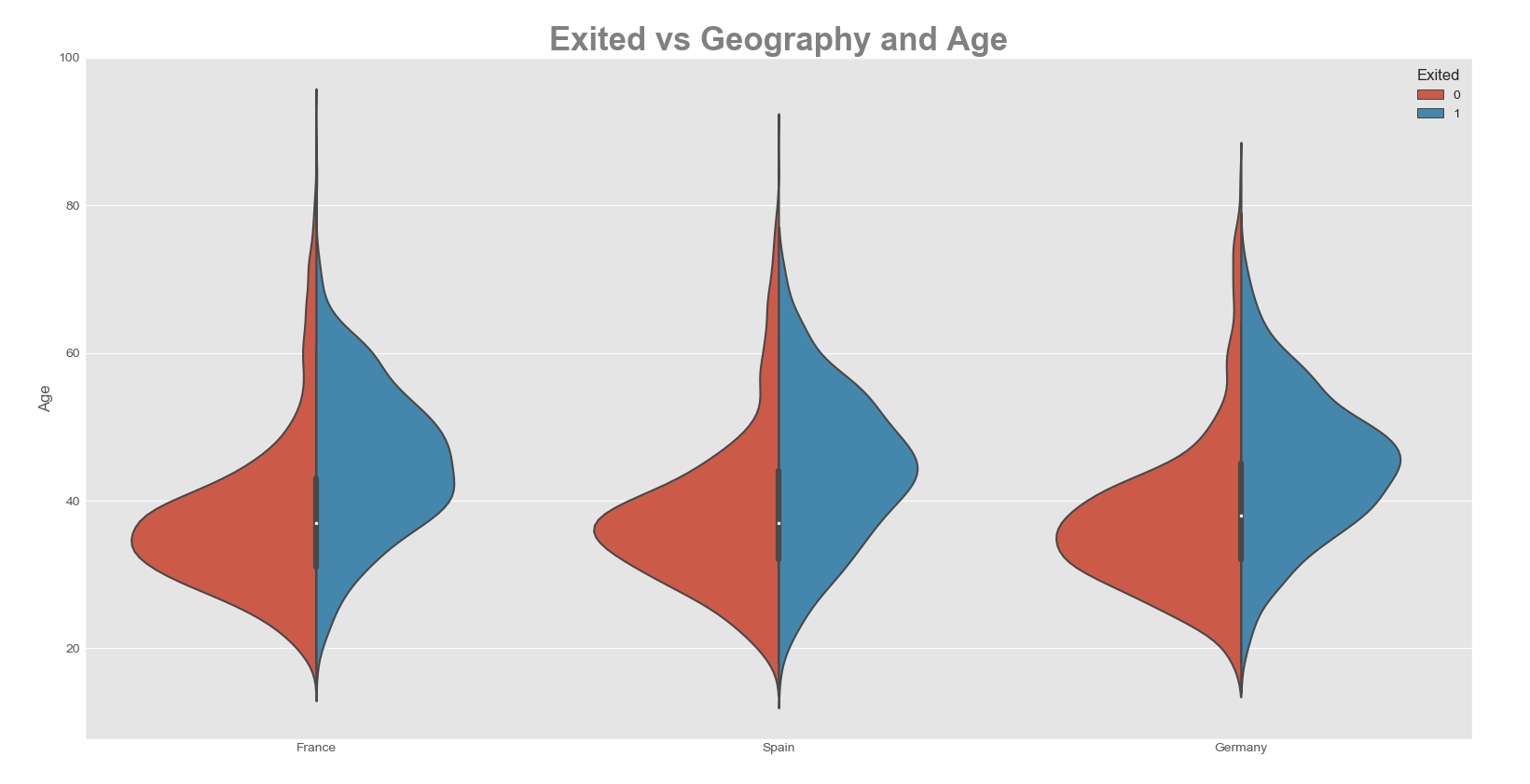


Figure 6: Exited vs Geography and Age

Exited vs Country

Figure 7 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.

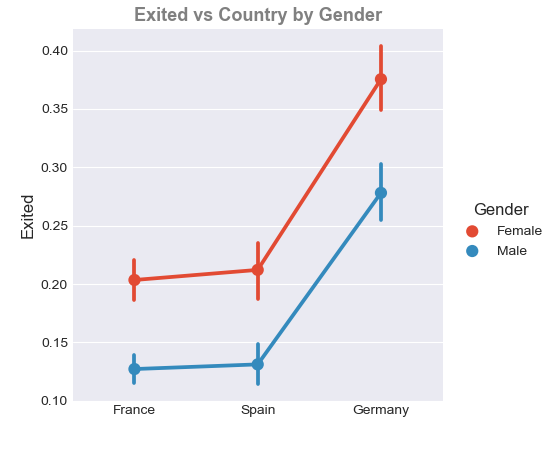


Figure 7: Exited vs Country

Exited vs Members Activity

The Figure 8 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.

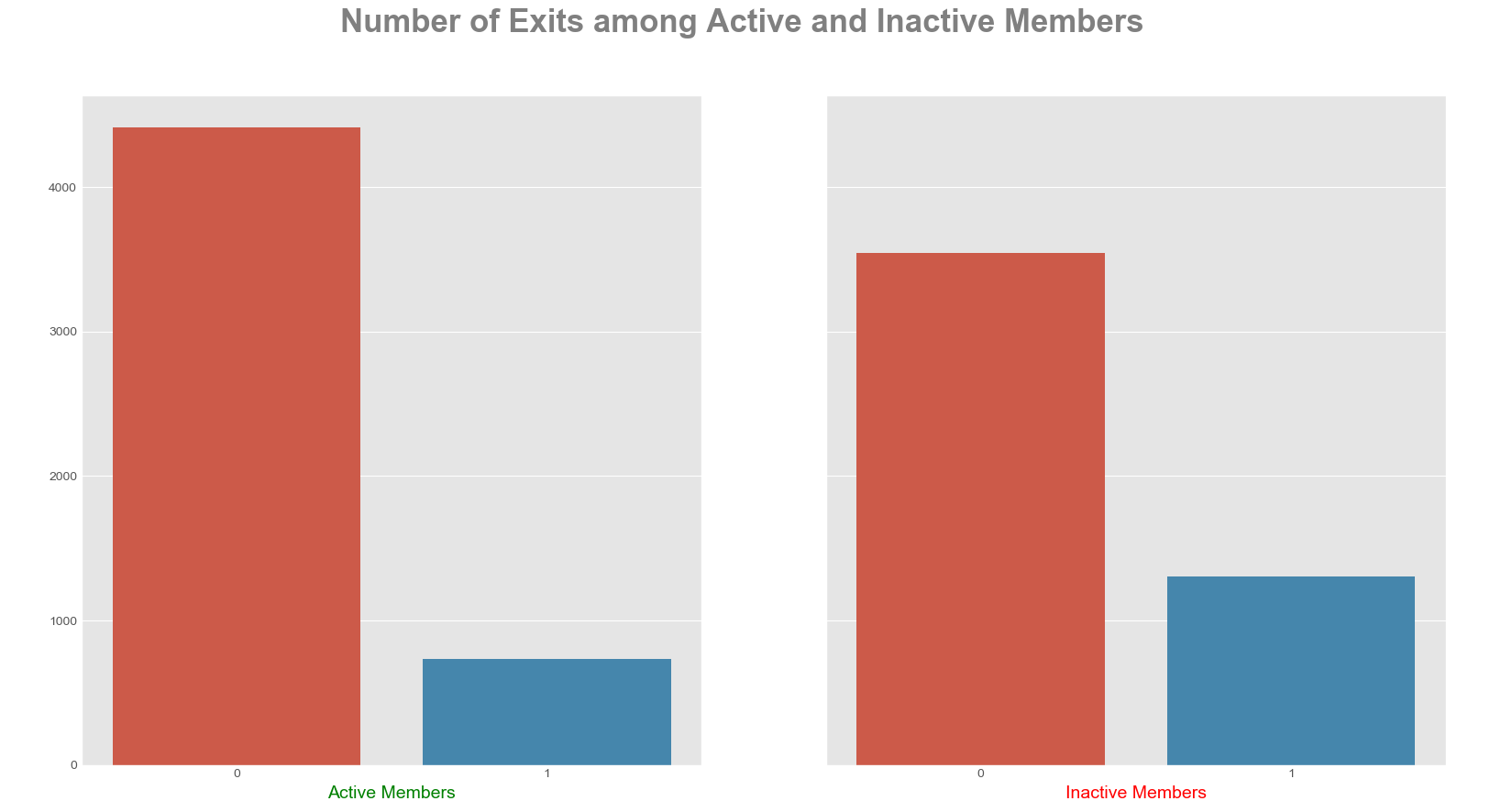
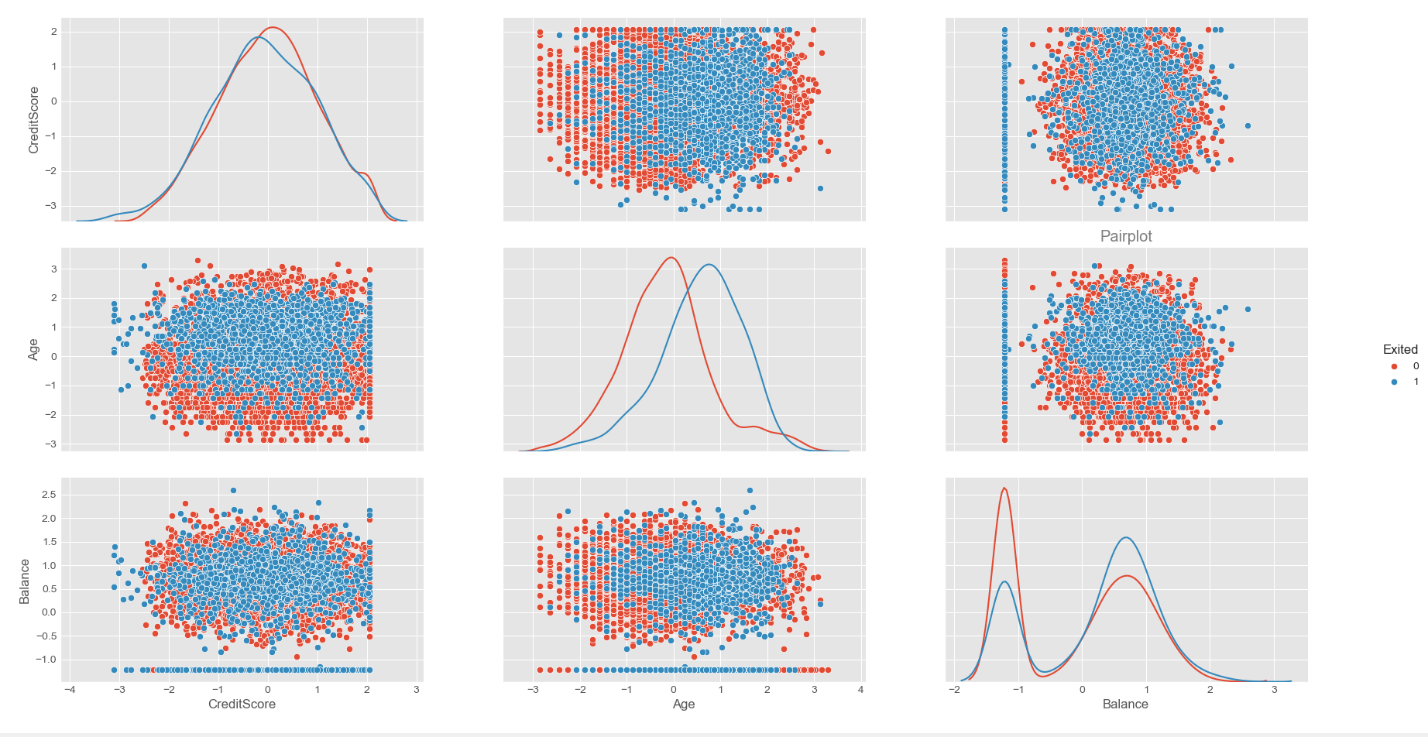


Figure 8: Number of Exits among Active and Inactive Members

Pair Plot

We are using a pair plot to see relationships between two variables [14]. Figure 9 is a pair plot on the three variables CreditScore, Age and Balance with the hue of Exited. People who exit are on average older ((2)(2)). Moreover, Age also has an influence on the distribution of customers Balance ((3)(2)) and likewise on the CreditScore ((1)(2)). Other variable distributions (other than Age) do not allow much interpretation (therefore this selection).



**Pair plot**

Hidden

Figure 9: Pair plot on selected variables

Data Correlation

Before modelling it is important to get insights on the correlation between variables. Figure 10 illustrates it graphically and with percentage notation using the Pearson correlation in a heat map (for simplicity the binary variables are included in the visualization, noting that geography has influence on savings [16]). The correlation of each continues variable is less than 6% which we consider small enough for the assumption of uncorrelated data.

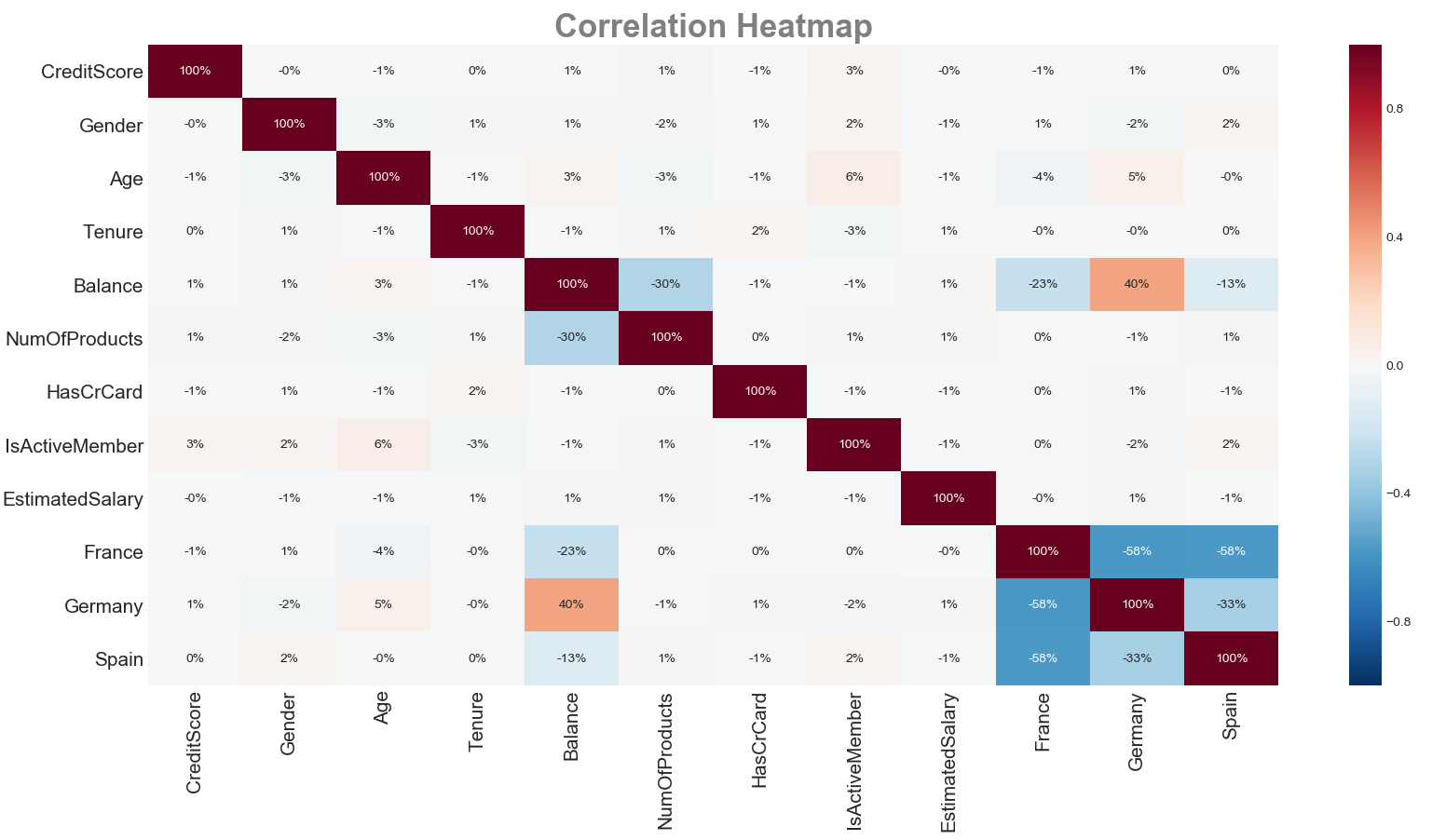


Figure 10: Heatmap

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 [3].

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 neighbours, 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 [22]. 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 designed to make binary classification you can predict continuous value with it and afterwards use some threshold to classify those values into two categories. There are two popular approaches in doing this. One is in using a 0.5 threshold, if predicted value is above that level, you predict 1 and when exactly that or less it you predict 0. This method seems to be logical from mathematic point of view, but the predicted values don’t necessary lay in a range <0;1> that is why a new approach has been developed, to calculate threshold by taking the 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. We use the Linear regression only to get a deeper understanding of our data and will not show it in model comparisons.

## Logistic Regression (PCA/LDA)

Logistic Regression is a classification algorithm that tries to predict the probability of a binary categorical variable. It is said to have a linear boundary between two results [13]. This is perfect to test with our data as the dependent variable is if a customer exited the business or not. A disadvantage is that the model assumes variables are independent, which is no problem for us [15]. For a Dataset with many variables the course of dimensionality can affect the model, this is why we also used the Principal Component Analysis an unsupervised method to reduce dimensionality and correlation of variables [2]. In the future we could also use the Linear Discriminant Analysis, which needs class labels in order to reduce dimensionality by separating classes along their linear discriminants [5].

## Decision Tree Classifier

The decision tree classifier is a supervised learning method, which can be applied in both regression and classification problems [12]. Given our binary classification problem described [cf. I Introduction] 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 [12]. An additional benefit is that decision trees are able to perform multi class classification problems [17], 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 [[11], [20]] [cf. a. Data], which makes their implementation easier. Moreover, they can also handle incomplete data [[11], [18]].

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 11) [[11], [19]].

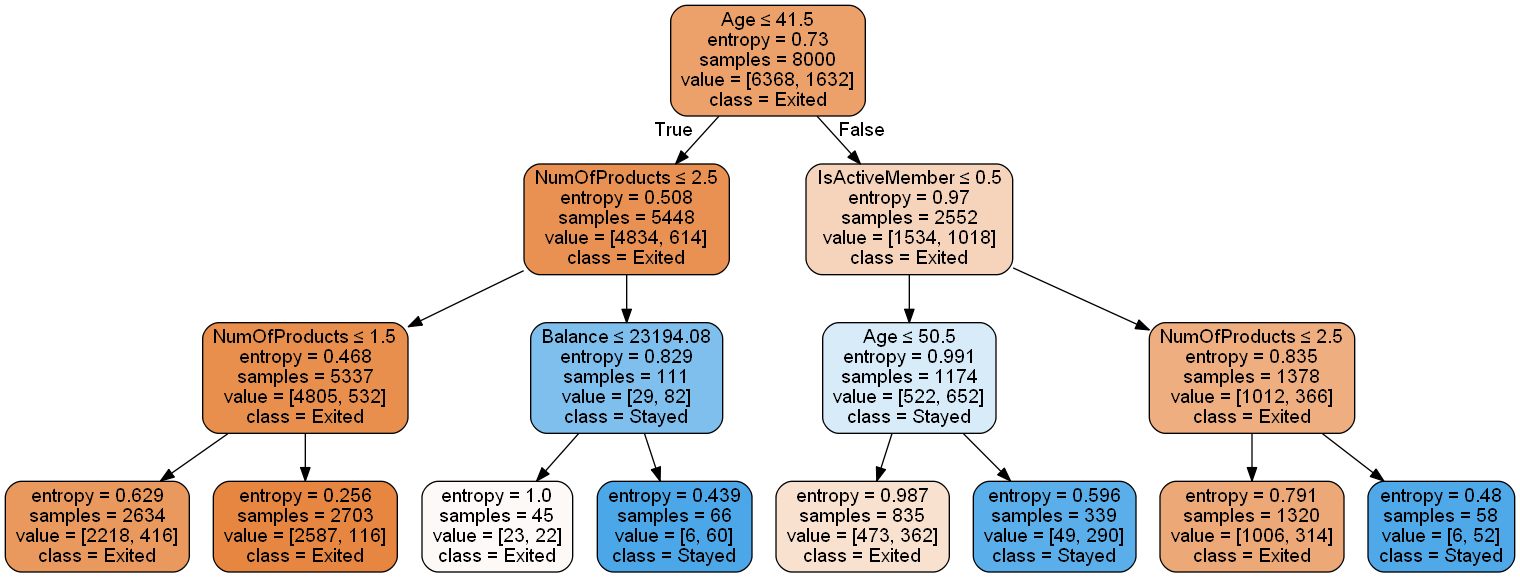


Figure 11: Example Decision Tree

## 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 [[1], [11]]. 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 [[11], [12]] 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 [[7], [12]]. However, the relative feature importance can be derived (cf. Figure 12) [11]. 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 [[12], [17]]. 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 [[11], [17]]. A more accurate prediction requires more trees, which results in a slower model, following the no free lunch theorem [25].

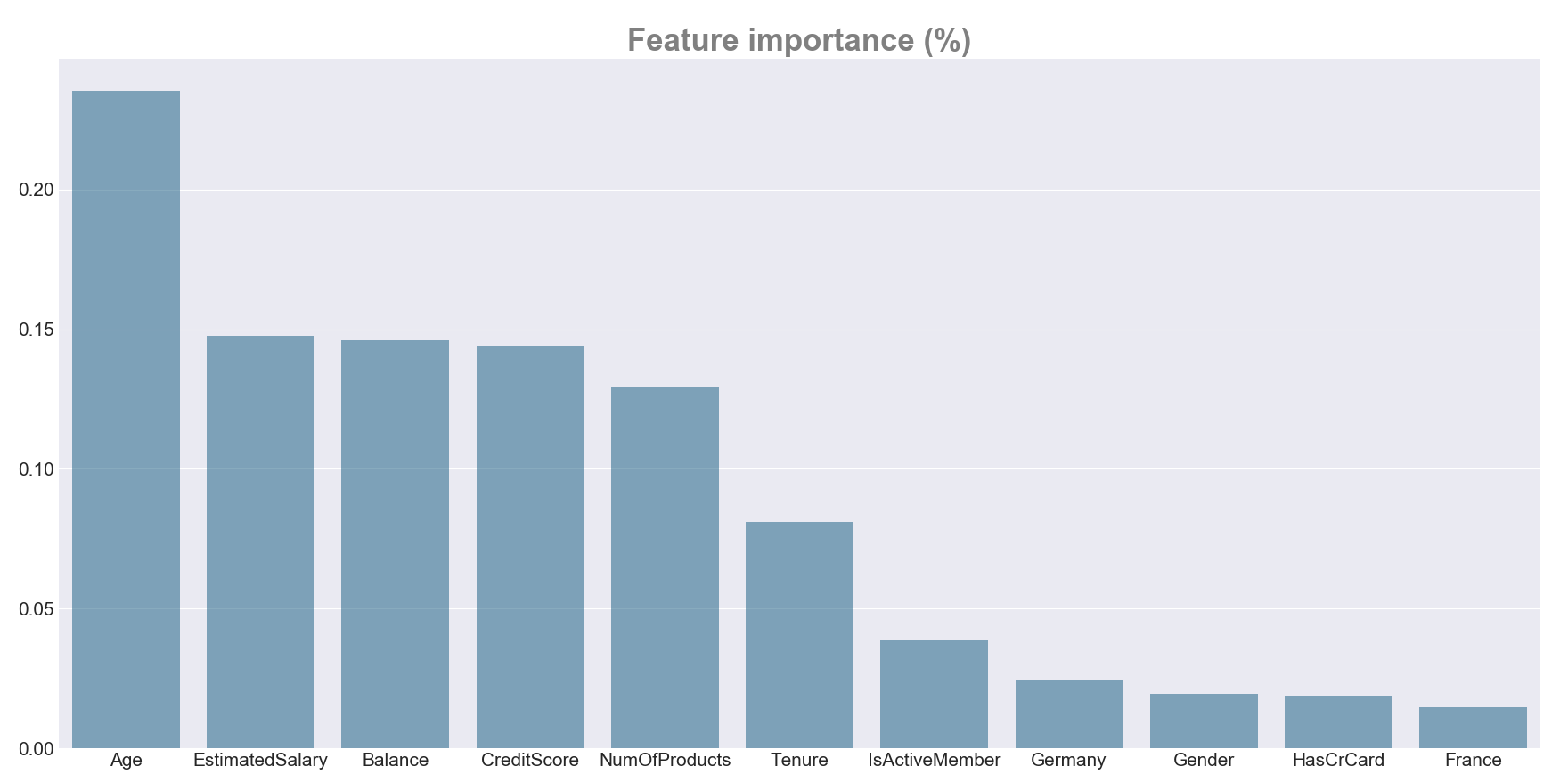


Figure 12: Feature importance

## 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 [23]. 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 can take a long time, but this is a suitable algorithm for the volume of data that we are working with in this case [12].

## 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 [[24], [26]]. Shortly explained, Bayes’ theorem uses prior knowledge of an event to more accurately predict the outcome of said event [8].

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 [9].

To sum up, Naive Bayes is a very fast module due to being very simple and easy to implement [4] . 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

The Perceptron or Single-Layer Neural Network is a machine learning technique belonging to the supervised learning 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 [10].

# Results

As a result of data cleaning, data exploratory analysis, and 6 machine learning algorithms implementation with default settings, we were able to predict the customer decision about exiting the store with 70-83% of accuracy (cf. Figure 13).

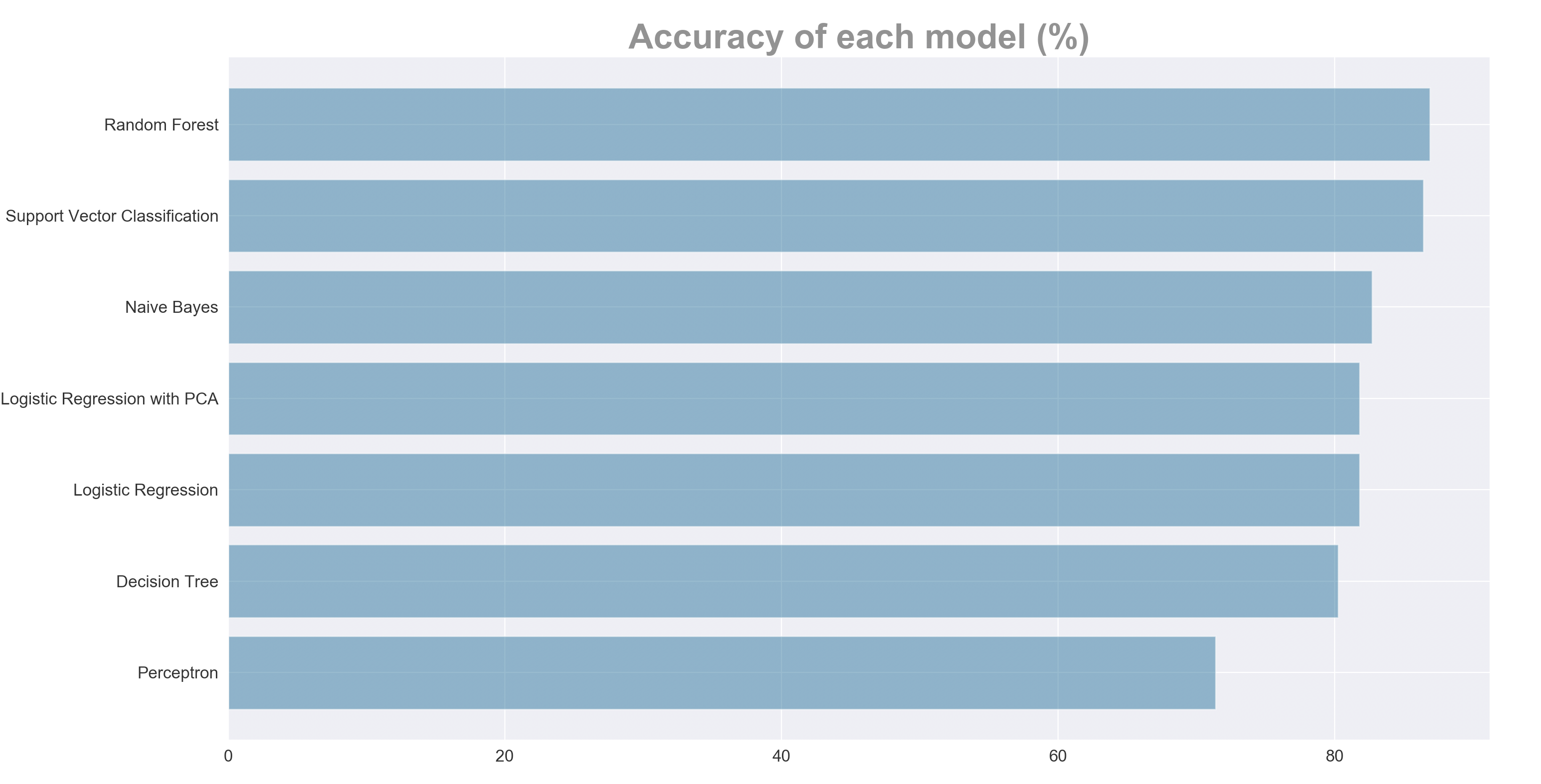


Figure 13: Accuracy of each model

But since output is not equally distributed in the data set and we have a specific problem at hand, it is not advisable to focus merely on the statistical accuracy (that is one of the reasons why we introduced a new monetary formula to measure how precise our models are in real business terms). Although differences in accuracy were almost indistinguishable, a significant difference in model performance have been identified in the Monetary score of each model. This is because the correct prediction of customers who are going to stay is much less valuable to the company than the correct predictions of customers who are going to leave. Additionally, a wrong prediction of customers who are going to leave costs the company three times as much as the wrong prediction of customers who are going to stay.

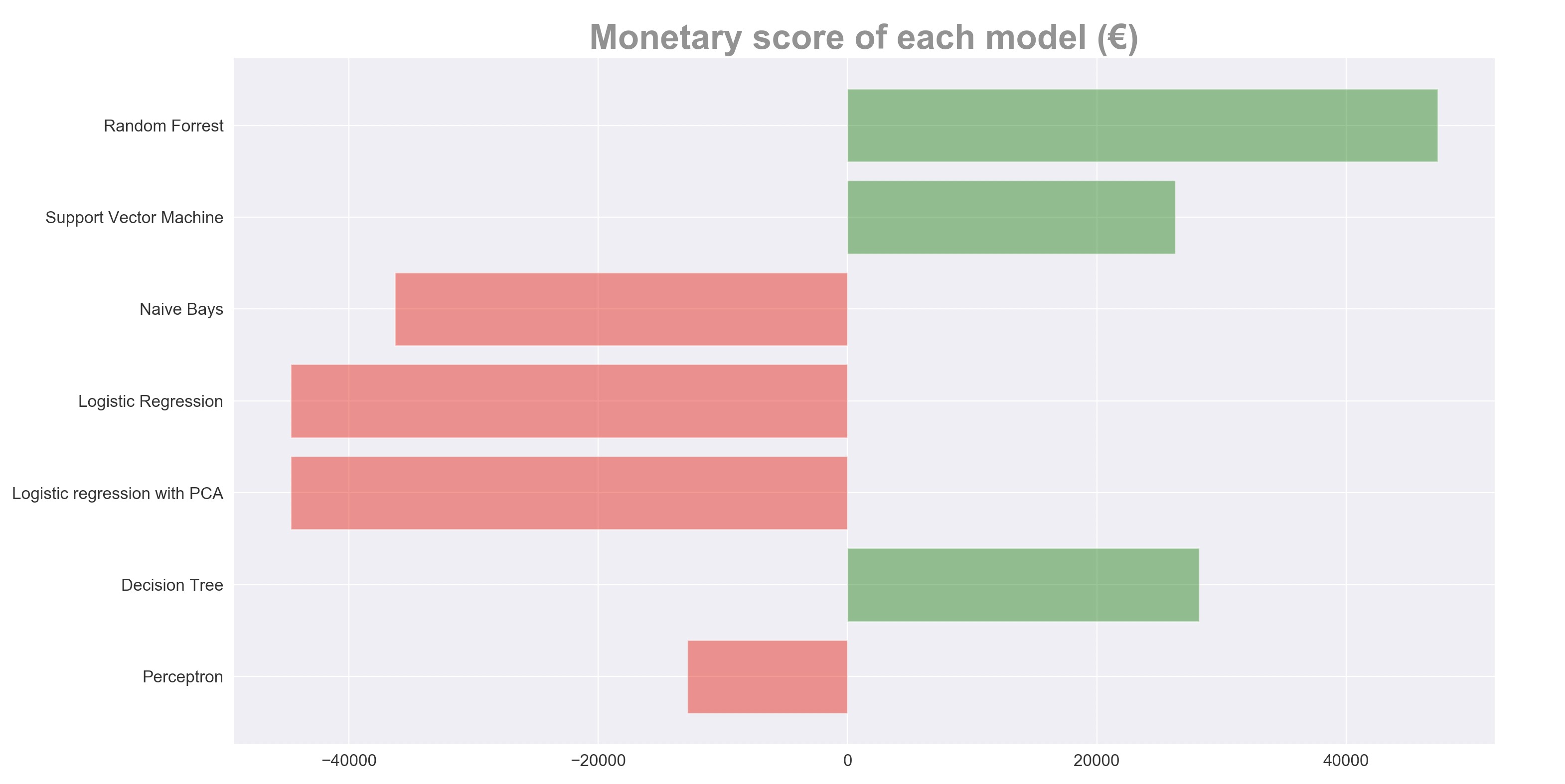


Figure 14: Monetary score of each model

Before we decide how to make our model better, let's determine which of our models are not worth keeping for further analysis. Since three of the models (Linear Regression and Perceptron) are significantly worse than the others, we decided to exclude them from further analysis. Before deciding on which models are the most promising, a cross validation has been performed in order to exclude the possibility of making wrong conclusions caused by randomness which could be a result of a one-time model testing on a single test data set [21]. In order to make results comparable we extrapolated the cross-validation results to have the equal size of the test set.

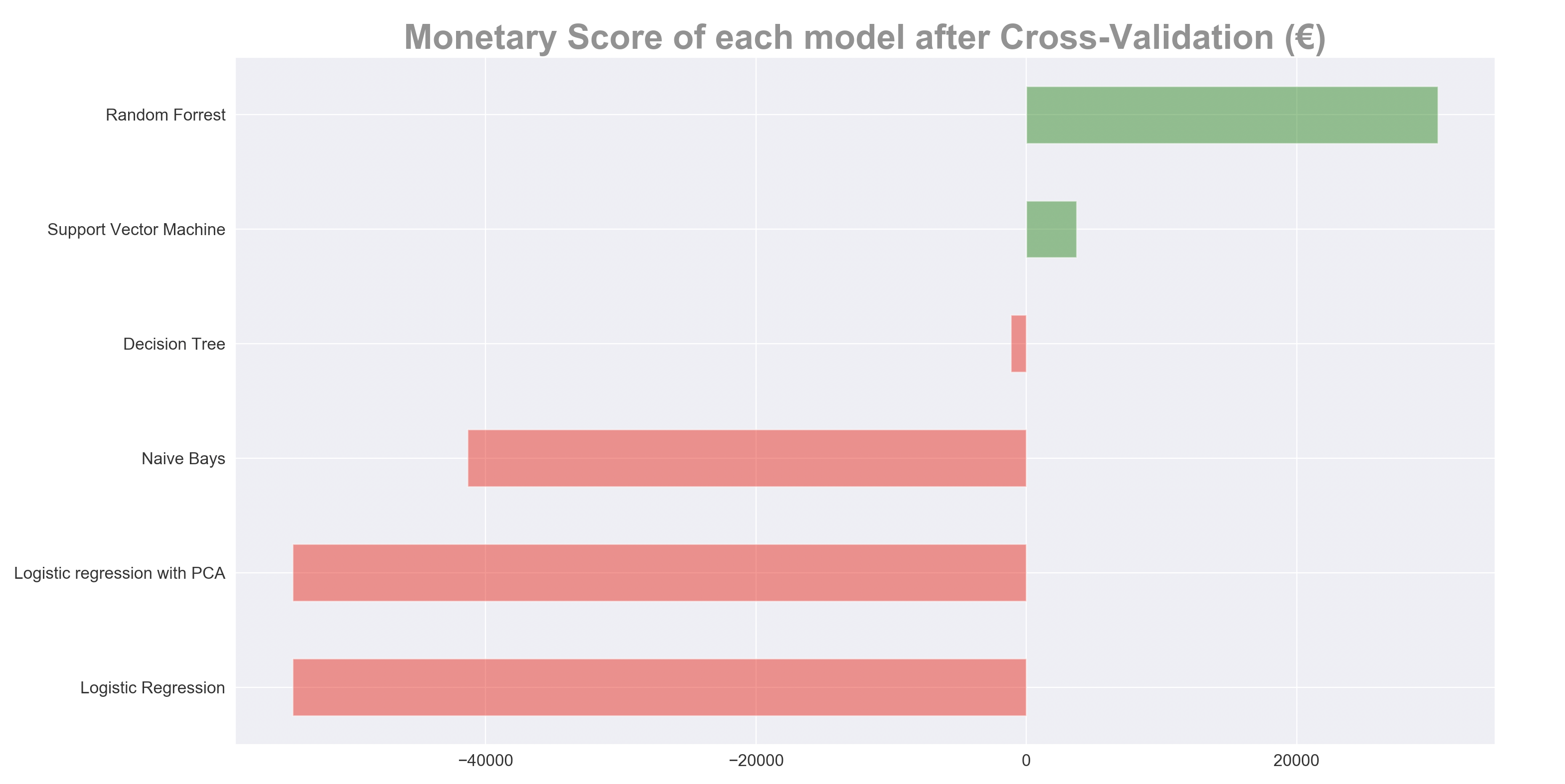


Figure 15: Monetary score of each model after Cross-Validation

After cross validating the model performance (cf. Figure 15) we are able to conclude that two models perform significantly better than any other model. It turned out that in this specific problem instance Random Forest and Support Vector Classification are able to provide the most valuable classification of customers.

# Model Tuning

In order to tune the models for even better performance, we used different parameters in a grid search, which also cross validates these results to find the best settings for the two models. As a result, the monetary is increased from 30500€ to 48260€ for Random Forest and from 3707€ to 38660€ for Support Vector Classifier. Grid search delivered better results for Random Forest (cf. Figure 16), which indicates that good parameter settings turned out to be more relevant for that model.

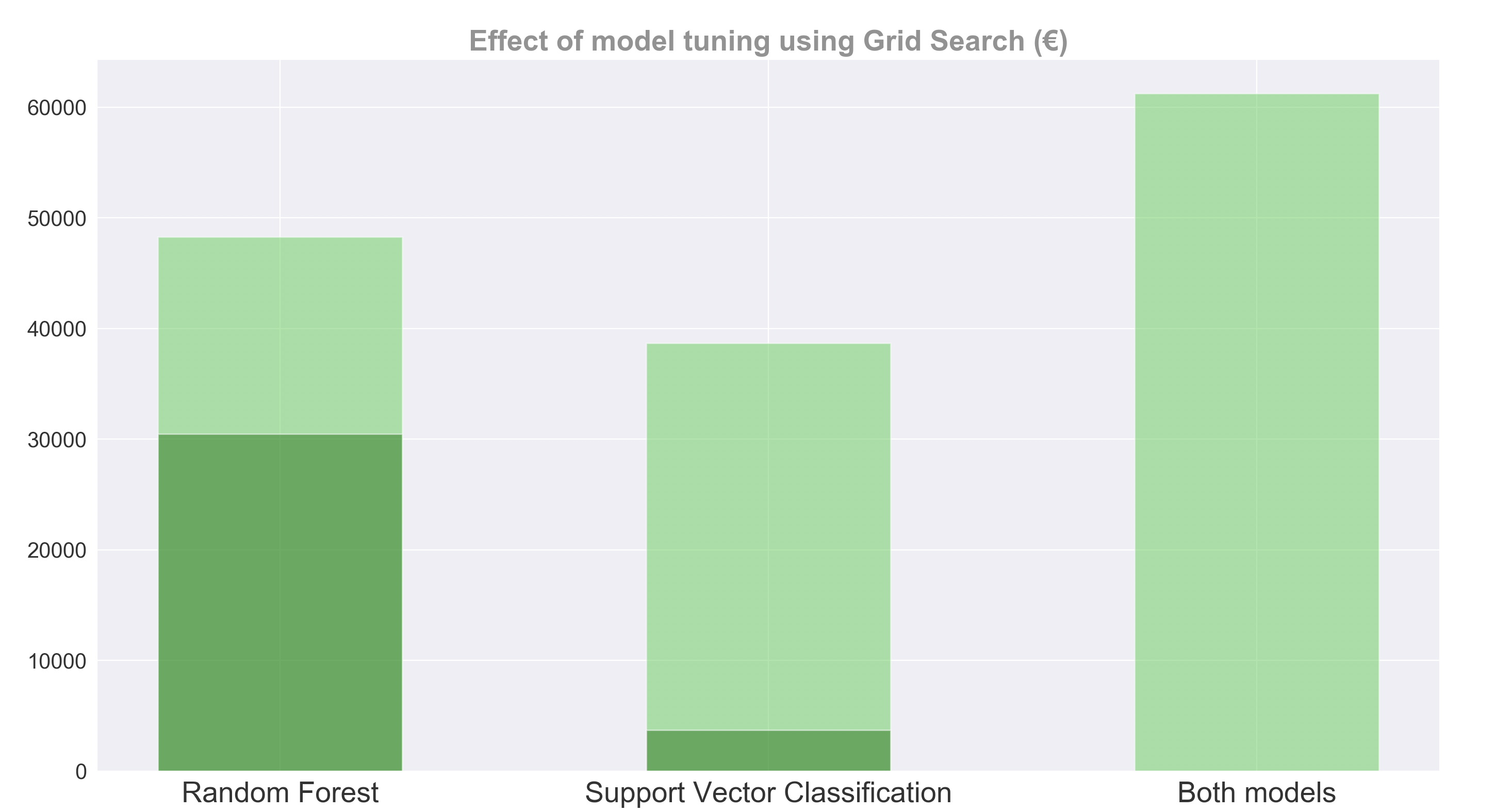


Figure 16: Effect of model tuning using Grid Search

Due to the nature of the problem it is very crucial for us to not have false predictions on customers who do not leave. This is why we first apply the random forest model to perform a first prediction and afterwards perform a second prediction with the support vector classifier on the customers the random forest predicted not leaving. This results in a achieved Return on the Marketing campaign of 61250€ on the test set representing an average of 30.76€ per customer.

# Conclusions

Starting out with a clean dataset to with the goal of maximizing a marketing campaign we found it is still crucial to prepare the data for modelling. Moreover, it is important to define the problem and find appropriate measures for testing different prediction models as standard statistical measures fail to grasp problem specifies. In our project we defined a return on marketing measure to test on each model and identified that it is important to not have false negative predictions. After testing different models, a common base of comparison has to be established as randomness play a decisive role in most of the models which might bias their outcome. We used cross-validation with our self-defined measure of marketing return to compare the models. Random Forest proved to be the best model for our problem case. Moreover, it is extremely important to mention that there was no linear relationship between the accuracy score of the models and their monetary score, solidifying the need for problem specific measures. Only two of our models are able to predict the customers churn in a way that applying the marketing campaign actually generates a profit with our test data. Improving the models by testing out different parameters even bettered their monetary score performance, resulting in a score of 48260€ for the random forest classifier and a score of 38660€ for the support vector machine. Given our insights on the problem we first used the random forest classifier to predict the customers churn and then performed a second prediction on the customers which were predicted not leaving in the first case to avoid false negative results. With this double testing approach, we were able to gain an estimated profit of 61250€ for the marketing campaign. In conclusion, were able to turn theoretical data science knowledge into real profit for the company.

# Acknowledgements

The project was conducted using the following python packages:

* Fabian Pedregosa, et al. (2011). ***Scikit-learn: Machine Learning in Python****, Journal of Machine Learning Research*, **12**, 2825-2830
* John D. Hunter*.****Matplotlib: A 2D Graphics Environment****, Computing in Science & Engineering*, **9**, 90-95 (2007), [DOI:10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
* Seabold, Skipper, and Josef Perktold. (2010). [*Statsmodels: Econometric and statistical modeling with python.*](http://conference.scipy.org/proceedings/scipy2010/pdfs/seabold.pdf) Proceedings of the 9th Python in Science Conference.
* Travis E, Oliphant. (2006). *A guide to NumPy*, USA: Trelgol Publishing
* Waskom, M., Botvinnik, et al. (2018). *seaborn: v0.5.0 (November 2014)*. [online] Zenodo. Available at: https://zenodo.org/record/12710#.XAkqdWj7TIU
* Wes McKinney. (2010). ***Data Structures for Statistical Computing in Python,*** *Proceedings of the 9th Python in Science Conference*, 51-56

The following application was also used:

* Ellson, J. (2018). [online] Graphviz. Available at: https://www.graphviz.org

# References

1. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), pp.5-32.
2. Brems, M. (2017). *A One-Stop Shop for Principal Component Analysis*. [online] Towards Data Science. Available at: https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c
3. Bronshtein, A. (2017). *Train/Test Split and Cross Validation in Python*. [online] Towards Data Science. Available at: https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6
4. Brownlee, J. (2014). *Naive Bayes Classifier From Scratch in Python*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/naive-bayes-classifier-scratch-python
5. Brownlee, J. (2016). *Linear Discriminant Analysis for Machine Learning*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning
6. Cope, G. (2018). *Dummy Variable Trap in Regression Models*. [online] Algosome. Available at: http://www.algosome.com/articles/dummy-variable-trap-regression.html
7. Donges, N. (2018). *The Random Forest Algorithm*. [online] Towards Data Science. Available at: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd
8. Ellinor, A. (2018). *Bayes' Theorem and Conditional Probability*. [online] Brilliant.org. Available at: https://brilliant.org/wiki/bayes-theorem
9. Gandhi, R. (2018). *Naive Bayes Classifier*. [online] Towards Data Science. Available at: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
10. Garbade, M. (2018). *How to Create a Simple Neural Network in Python*. [online] Kdnuggets. Available at: https://www.kdnuggets.com/2018/10/simple-neural-network-python.html?fbclid=IwAR2y-is8FrDb-Ma8f-tSA8eX-hme8-so4IOI1VOHOEn8Gq7Kdwq6y6XWqoo
11. Han, J., Kamber, M. and Pei, J. (2012). *Data mining Concepts and Techniques*. 3rd ed. Waltham: Morgan Kaufmann.
12. James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). *An introduction to statistical learning*. New York: Springer.
13. Joglekar, S. (2015). *Logistic Regression (for dummies)*. [online] Sachin Joglekar's blog. Available at: https://codesachin.wordpress.com/2015/08/16/logistic-regression-for-dummies
14. Koehrsen, W. (2018). *Visualizing Data with Pairs Plots in Python*. [online] Towards Data Science. Available at: https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166
15. Li, S. (2017). *Building A Logistic Regression in Python, Step by Step*. [online] Towards Data Science. Available at: https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
16. Misztal, Piotr (2011) : The relationship between savings and economic growth in countries with different level of economic development, e-Finanse: Financial Internet Quarterly, ISSN 1734-039X, University of Information Technology and Management, Rzeszów, Vol. 7, Iss. 2, pp. 17-29
17. Navlani, A. (2018). *Random Forests Classifiers in Python*. [online] DataCamp. Available at: https://www.datacamp.com/community/tutorials/random-forests-classifier-python#advantages
18. Pedregosa et al., (2011).,[*Scikit-learn: Machine Learning in Python*](http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html), JMLR 12, pp. 2825-2830.
19. Portilla, J. (2016). *Python for Data Science and Machine Learning Bootcamp*. [online] Udemy. Available at: https://www.udemy.com/python-for-data-science-and-machine-learning-bootcamp/learn/v4
20. Scikit-learn. (2018). *1.10. Decision Trees*. [online] Available at: https://scikit-learn.org/stable/modules/tree.html
21. Scikit-learn. (2018). *3.1. Cross-validation: evaluating estimator performance*. [online] Available at: https://scikit-learn.org/stable/modules/cross\_validation.html?fbclid=IwAR3eWbJB8f94qsy\_APgdn4QtL0LKm1PUPl6ZjI1Y6s32oFxTdRrdEVtm3-4
22. Scikit-learn. (2018). *Importance of Feature Scaling*. [online] Available at: https://scikit-learn.org/stable/auto\_examples/preprocessing/plot\_scaling\_importance.html
23. Tandel, A. (2017). *Support Vector Machines — A Brief Overview*. [online] Towards Data Science. Available at: https://towardsdatascience.com/support-vector-machines-a-brief-overview-37e018ae310f
24. Waldron, M. (2015). *Naive Bayes for Dummies; A Simple Explanation - AYLIEN*. [online] AYLIEN. Available at: http://blog.aylien.com/naive-bayes-for-dummies-a-simple-explanation
25. Wolpert, D. and Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), pp.67-82.
26. ZHANG, H. (2005). EXPLORING CONDITIONS FOR THE OPTIMALITY OF NAIVE BAYES. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(02), pp.183-198.

This page is left out intentionally