Project

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 a part of the syllabus of Data Mining in the Master of Data Science and Advanced Analytics at Nova Information Management School, IMS, and has

Data is regarding a fictional insurance company in Portugal

In the ABT (Analytic Based Table) we have data regarding 10.290 Customers

The project was initiated by

The dataset used (what the data contains)

The problem was given, briefly describe it

As a Data Mining/Analytic Consultant, you are asked develop a Customer Segmentation in such a way that it will be possible for the Marketing Department to better understand all the different Customers’ Profiles. You are expected to define, describe and explain the clusters you chose. Invest time in reasoning how you want to do your clustering, possible approaches, and advantages or disadvantages of different decisions. Simultaneous, you should express the marketing approach you recommend for each cluster.

The end goal of our project was to…

Explain more or less how the report is structured

# Description

This section of the report will briefly describe the dataset and its variables. The used dataset contains data regarding 10.290 customers of a fictional insurance company in Portugal.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Format** | **Description** |
| Customer Identity | int64 | ID |
| First Policy’s Year | float64 | First year a customer enters the business. Also refered to as Customer’s First Year Policy. |
| Birthday Year | float64 | Customer’s year of birth.  Note: The current year of the database is 2016. |
| Educational Degree | object | Educational level of a customer. |
| Gross Month Salary | float64 | Gross monthly salary (€). |
| Geographic Living Area | float64 | Living area. |
| Has Children (Y=1) | float64 | Binary variable representing if a customer has children. 0 for “does not have Children” and 1 for “has Children”. |
| Customer Monetary Value | float64 | Customer’s monetary value. Calculated the following way:  Lifetime value = (annual profit from the customer) X (number of years that they are a customer) - (acquisition cost) |
| Claims Rate | float64 | Claims rate. Amount paid by the insurance company (€)/ Premiums (€)  Note: in the last 2 years |
| Premiums in LOB: Motor | float64 | Annual Premium (€) in LOB: Motor |
| Premiums in LOB: Household | float64 | Annual Premium (€) in LOB: Household |
| Premiums in LOB: Health | float64 | Annual Premium (€) in LOB: Health |
| Premiums in LOB: Life | float64 | Annual Premium (€) in LOB: Life |
| Premiums in LOB: Work Compensations | float64 | Annual Premium (€) in LOB: Work Compensations  Note: Negative Premiums may manifest reversals occurred in the current year, paid in previous one(s) |

Table 1: Dataset

# Pre-processing

This section of the report will cover the process of analysing the initial dataset and preparing it for further observations. It will also cover the reason and process behind data transformation. This will be done by going through every column making sure our investigation is as thorough as possible.

The first step taken was to rename the columns of the dataset into shorter yet still meaningful variable names as the original ones were very long. The new variable names will be shown in between brackets as we examine each of them in the following pages.

**Dealing With Outliers**

Customer Identity (CustID)

As it was of no value to the current project this variable was dropped.

First Policy’s Year (1stPolYear)

While observing the list of values included in ‘1stPolYear’ it came to attention that there were values higher than 2016 and given the fact that the dataset was of static 2016 data these were considered error data and removed (values removed > 2016).

Brithday Year (BirthYear)

On the assumption that there are no individuals born in the 19th century values under the year 1900 were removed as they were considered to be error data.

Gross Monthly Salary (GrossMthSalary)

The description of this variable showed a mean of approximately 2.507 and a maximum value of 55.215. The last value seemed odd so a boxplot of the entire column was plotted for a more visual observation of the list. It was concluded that there were 2 extreme values and these were removed making the data more naturally distributed (values removed > 30.000).

Customer Monetary Value (CustMonetVal)

The description of ‘CustMonetVal’ showed a minimum value that was extremely low compared to others. It seemed very unlikely to be correc. As there were other values very far from the mean it was decided to drop all values under -2000 smoothing for extreme values and possible wrong data, leading to the loss of 14 data pionts.

Claims Rate (ClaimsRate)

The maximum value of ‘ClaimsRate’ appeared to be 4.33 but even though it was a high value it did not seem out of the ordinary. It was only after seeing the boxplot that the maximum value was considered an extreme value. It was a single value away from the rest of a naturally distributed list so it was dropped.

Premiums in LOB: Motor (PremLOBMotor)

Following the same strategy of looking at a visual representation of the data, extreme values were once again spotted. This time there were 6 items away from the bulk of the data. It was decided to drop all 6 of them (values removed > 2.000).

Premiums in LOB: Health (PremLOBHealth)

A boxplot of ‘PremLOBHealth’ showed 2 values excessively far from the rest of the data. They were considered too extreme and removed from the set (values removed > 5.000).

BOXPLOTS here

**Handling Skewed Data**

Until here the only action taken towards the data was the removal error data / extreme values. From this point on skewed values were encountered and further actions were required to normalise it.

Explain 3 sigma rule strategy

#rows: 9862 --> meaning 434 rows dropped (4.2%)

#3.5 sigma: 9927 rows --> 369 rows dropped (3.6%)

#4 sigma:10022 rows --> 274 rows dropped (2.6%)

The next 3 paragraphs contain the same process. Consider changing overall format.

Premiums in LOB: Household (PremLOBHousehold)

After visualizing the data it was detected that the data was skewed and contained possible outliers. The first strategy was applying a logarithmic transformation. After this, the data appeared to be closer to a normal distributkion and it was now clear that extreme values were still to be dealt with. The 3 sigma rule was the second strategy applied to solve this issue.

Premiums in LOB: Life (PremLOBLife)

The same was observed in this column so the same strategies were applied. First a log transformation to fix skewed data following the 3 sigma rule to deal with extreme values.

Premiums in LOB: Work Compensations PremLOBWorkCompensation)

The same was observed in this column so the same strategies were applied. First a log transformation to fix skewed data following the 3-sigma rule to deal with extreme values.

**Handling missing values**

The process conducted so far as looked at fixing extreme values / error data and skewed data but it did not evaluate whether or not there are any missing values in the dataset. Table X was created with the use of a function and it displays what columns contain missing values (blank spaces are missing values for us too, therefore we repaced them beforehand) and how many are missing per column.

INSERT TABLE HERE

From the table we can see that there are 2 premiums containing missing values: ‘PremLOBHealth’ and ‘PremLOBMotor’. To fix the missing values in these two columns the null values were replaced for zeros. This was done because one would assume that null values mean a customer did or does not pay for this specific premium.

The column ‘GeoLivArea’ also had one missing value but after exploring the row from which they belonged it was concluded that that specific row were of poor quality as other data was missing too. Therefore, we decided to drop the whole data point.

Mention that 20% of the data contains 1stPolYear < BirthYear – where to put this

**Applying Machine Learning Algorithms to Replace Missing Values**

The remaining columns containing missing values were the following: ‘GrossMthSalary’, ‘HasChild’, ‘EduDegree’ and ‘1stYearPol’. As the data on these four columns is essential for the continuation of the project and it cannot be simply dropped or replaced by zeros like previously seen a new solution had to be found. This solution is the usage of machine learning models to predict possible values for the missing data set, replacing like this null values of the remaining columns.

Three different models made this process possible: Logistic Regression, K-Nearest Neighbour and Decision Tree.

Do we include an explanation of each model?

Do we mention train and test set?

GrossMothSal – Regression

HasChild – KNN

EduDegree – Trees

1stYearPol – Trees

**Exploratory Data Analysis (heatmap)**

INSERT HEATMAP HERE

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.

**Feature Engineering and Selection**

The variable ‘EduDegree’ is a categorical variable containing 4 possible values representing different stages or levels of education. It is anticipated that this column will provide some insights in the next section of the report. A quick fix is to turn its 4 possible values into 2 new categories: ‘Lower Education’ and ‘Higher Education’.

Hidden

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 learn 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, it is possible to 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, we predict 1 and when it equals that threshold level or less then it is predicted as 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 this data is skewed. Additionally, by applying backward elimination strategy, we can get more insights about the 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.

## 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]].

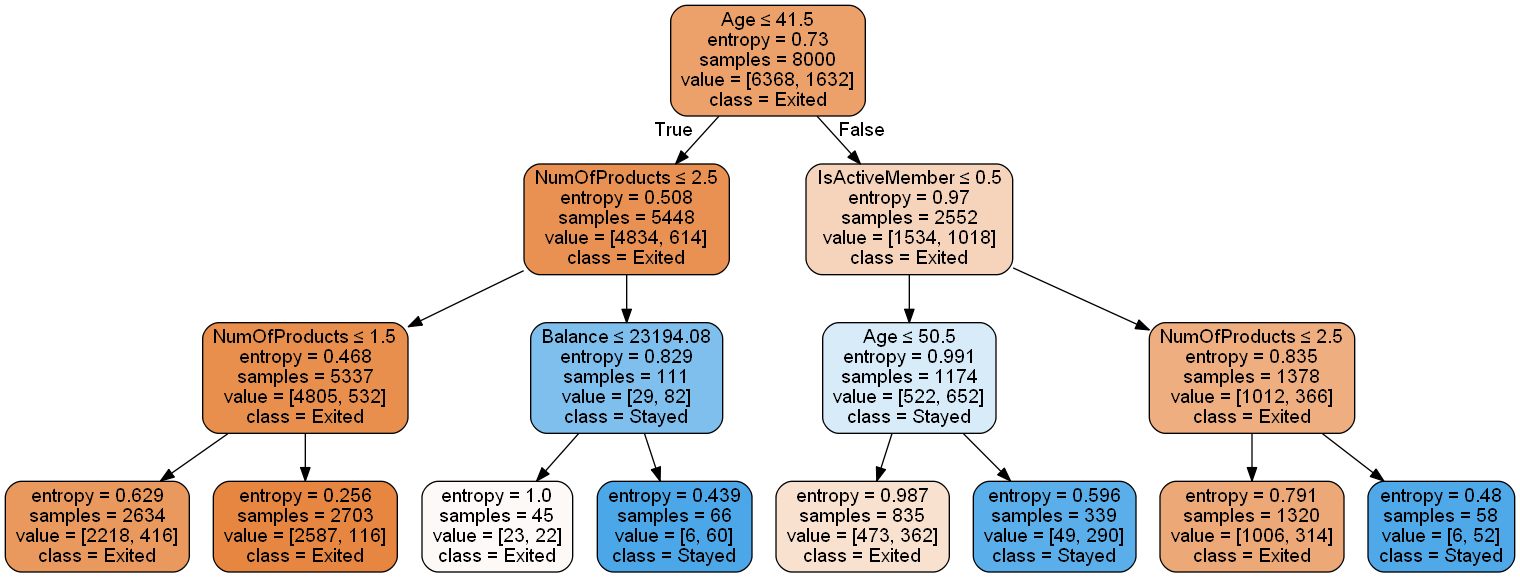


Figure 11: Example Decision Tree

## 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].

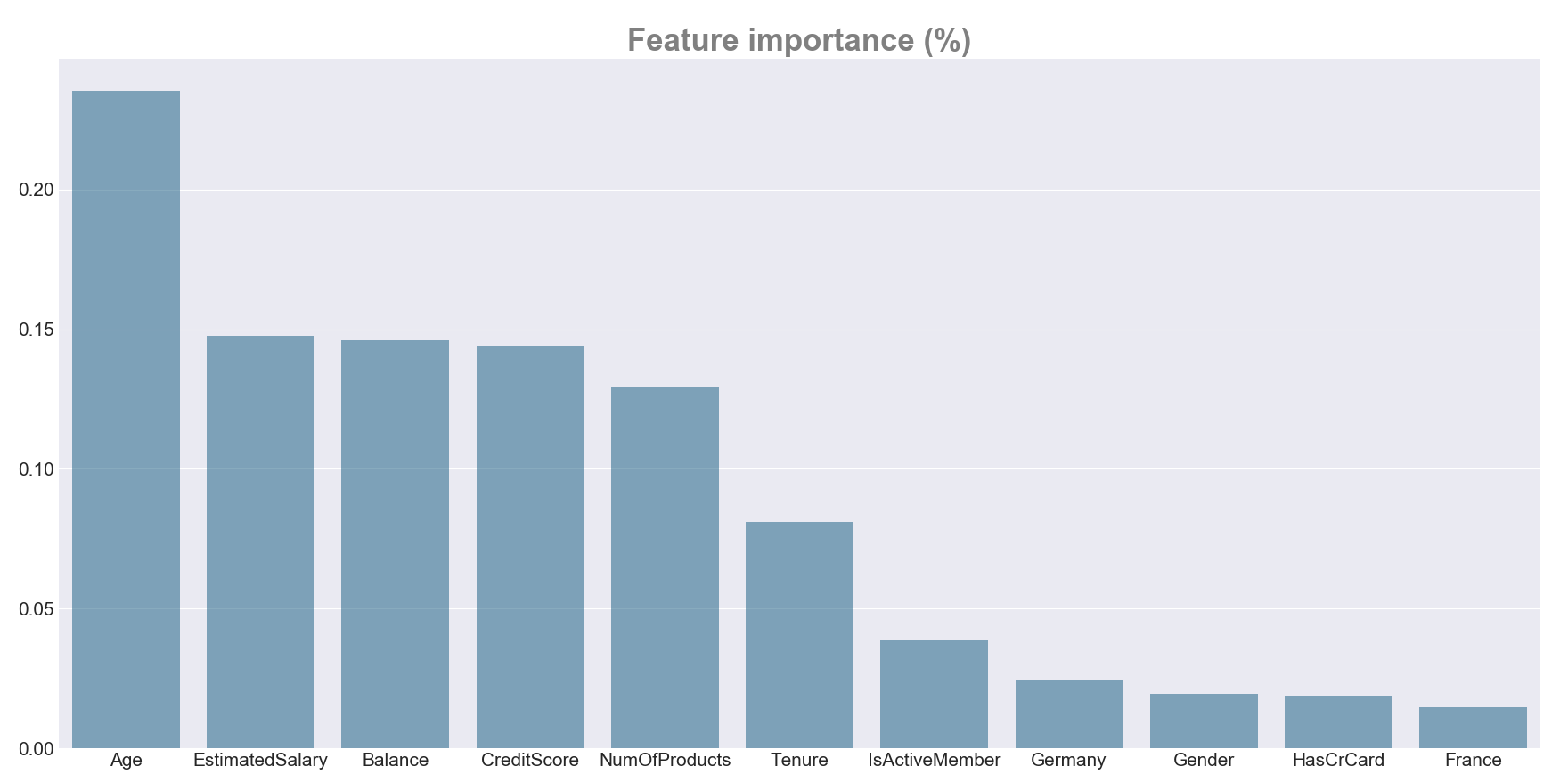


Figure 12: Feature importance

# Results

After 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).

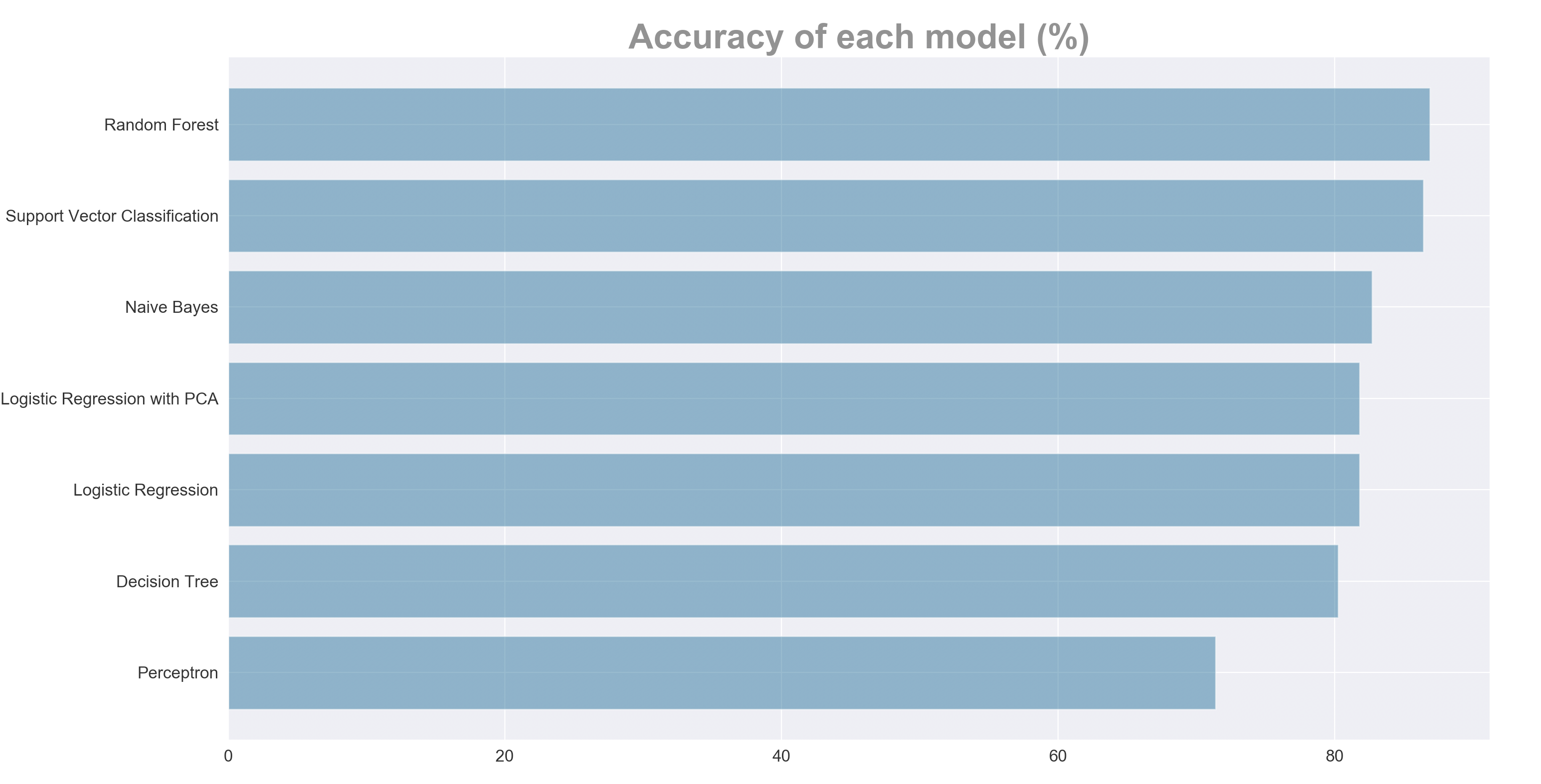


Figure 13: Accuracy of each model

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 for 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.

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]. The perceptron model is providing a significantly worse accuracy than the other models and does not perform well on the monetary score either. This and the long training times are the reason why we excluded it from cross-validation. In order to make results comparable we extrapolated the cross-validation results to have the equal size of the test set.

After cross validating the model performance (cf. Figure 15) it has been concluded 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 cross validates the results to find the best settings for the two models. As a result, the Monetary Score 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.

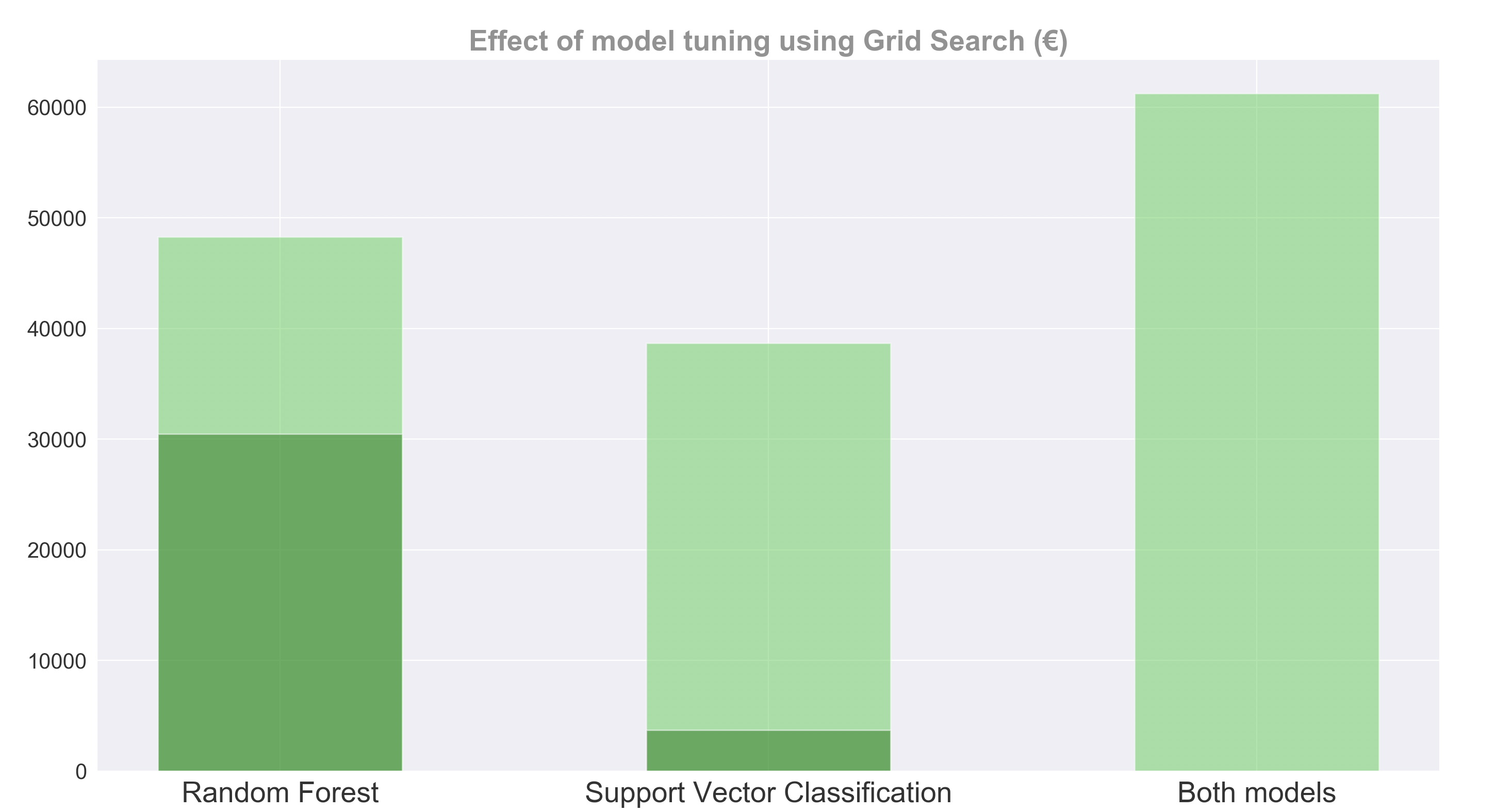


Figure 16: Effect of model tuning using Grid Search

Due to the nature of the problem it is essential 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 total Return on the Marketing campaign of 61250€ on the test set representing an average of 30.76€ per customer.

# Conclusion

Starting out with a clean dataset 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 models quality, since standard statistical measures fail to grasp problem-specific details. In our project we defined a return on marketing measure (Monetary Score) for models evaluation and identified the importance of minimizing the number of 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

# 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