# 1 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 hour project, further described later [(cf. XXX) ]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.

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

The marketing department of the 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, this information is used to evaluate the performance of different prediction models.

[link to below]

[The remainder of this report is structured as following: first, the data set is described (cf. XXX), and different prediction models are introduced (cf. XXX). Afterwards the result of the models are compared (cf. XXX) of which the *two* best are further optimized and the best model is chosen (cf. XXX). At last a conclusion is drawn (cf. XXX).]

# 2 Data and problem description

## 2.1 Data description

## I think it can be good to add either small hist plot for most of the varibales or add description it looks (ex. Add info that data are close to normal distribution or skewed

Dataset contains random sample of customers from bank operating in Europe. Data comes from banks database and because of that the quality of data is very high. Each row represent one client and all information about him stored in a bank. Data set contains only static data and represent state on 31/12/2017. Half year after data were collected bank measured whether client exited from bank or stayed. Dataset contain all information bank had and no information about importance of each information, that was left to be fully determined by THIS\_PROJECT.

Variables contained in the dataset:

**Input Data:**

* **RowNumber** – The row number (index from database table). Numeric value in the range of <1;1000>
* **CustomerId** – The unique Customer ID, which is the most granular attribute in the dataset. Numeric value in the range of <15565701; 15815690 >
* **Surname** – The surname of a customer.
* **CreditScore** – The Credit Score of a customer. Numeric value in the range of <350;850>. Mean value of Credit Score equals 650.53, while median is 652.0
* The Credit Score ranges between 300 and 850 and higher value implies a higher “creditworthiness”.
* **Geography** – The country location of the bank branch. It contains 3 discrete values: France, Germany and Spain.
* **Gender** – Gender of the customer. Boolean values consisting of {‘Female’, ‘Male’}. The dataset contains 5457 males and 4543 females.
* **Age** – Current age of the customer (applicable to the current year of 2018). Numeric value in the range of <18;92>. Mean value of Age equals almost 39, while median is 37.
* **Tenure** – The number of years the customer has been the banks customer. The numeric value in the range of <0;10>. Median value equals 5.
* **Balance** - Average amount of money in the bank account in 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** – The number of products the customer acquired in previous 6 months. Numeric value in the range of <1;4>. Median value equals 1.
* **HasCrCard** – The Boolean value, which equals 1 if Customer has a Credit Card and 0 otherwise. The dataset contains 7055 customers who possess a credit card (value 1) and 2945 who don’t (value 0).
* **IsActiveMember** - The Boolean value, which equals 1 if Customer is considered an active member and 0 otherwise. The dataset contains 5151 of customers who are considered active members (value 1) and 4849 who are considered inactive (value 0). Customer who made any transaction within previous 30 days is considered an active member.
* **EstimatedSalary** – Estimated annual salary of the customer. Numeric value expressed in USD in the range of <11.58; 199992.48>. The average value of estimated salary equals 100090.24, while the median is 100193.92.

**Output Data:**

* **[Exited** - The Boolean value, which equals 1 if within previous 6 months the customer exited the bank and 0 otherwise. The dataset contains 7963 customers who did not exit (value 0), and 2037 of exit records (value 1) in P6M.]

## 2.2 Problem description

[intro ]

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: [750€?]
* 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:

1. If a customer leaves till T1 and no campaign is applied: loss of 450€ (money expected to be gained by T1)
2. If a customer does not leave and no campaign is applied: no loss or gain (everything happened as excepted till T1)
3. If a customer does not leave because the campaign is applied: gain of 200€ (no loss of 450€ by T1, a gain of 450€ because the customer is going to stay till T2, a loss for applying the campaign of 150€)
4. 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€)

[**picture** ]

# 3 Data preprocessing

# 4 Modeling

## 4.1 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 [UDEMY][DM].

## 4.2 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 [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)].

* (<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

## 4.3 Support vector machines (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. 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. For better results, hyperparameters running was performed by means of GridSearch on 2 of algorithm’s parameters.

[NEEDS REFERENCES]

# 5 Modeling selection and tuning

# 6 Conclusion

# Appendix

# 1 XXXX

# 2 Bibliography