UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

SEMINAR

Targeted marketing using machine learning methods

Duje Medak

Zagreb, April 2020.

CONTENTS

1. Introduction

1.1. Targeted marketing

In order to increase the profit and expand business, companies typically invest part of their capital into marketing strategies. Traditional way of marketing consists of mass campaigns, without focus on some specific clientele. According to [15] the effectiveness of this kind of marketing is less than 1% . Targeted marketing is a new business model of interactive one-to-one communication between marketer and customer [13]. Focusing on a specific subset of clients makes the marketing more cost effective, especially because the subset of clients is determined by matching the needs and desires of customers with the offered product or service. Knowing the desires and wishes of clients is not always easy and companies usually need to implicitly determine those from the available clients data and previous knowledge. Using those information companies try to develop a decision support system (DSS) in order to help the managers with decision making.

1.2. Decision support systems

Decision support system (DSS) are comprehensive computer systems and related tools that assist managers in making decisions and solving problems [8]. Decision support systems can be completely computerized, human-powered or a combination of both. In this work, human expertise was used to select some important attributes but no further input is needed from human in order for this system to work after the models were developed. The traditional approach to data analysis for decision support has been to couple domain expertise with statistical modeling techniques to develop handcrafted solutions for specific problems [2]. One of the DSS related concepts is Business Intelligence (BI). Business intelligence comprises the strategies and technologies used by enterprises for the data analysis of business information [6]. One of those technologies often used in the process of building a DSS is Data Mining.

1.3. Data mining

Data mining is the process of sorting, organizing or grouping large amount of data in order to extract useful information. What is a useful information depends on the area where it is being used. In this work a useful information would be whether some client would be interested in a specific new product or service or not. This information, however, is not trivial to obtain. Data mining should reveal relationship, and regularity of the data that is usually to complex to simply be noticed from data visualization. Data mining is often used to extract usable information from the data usually by hand picking, transforming and filtering the data with the help of an expert. In the end the important information can be visualized in a clean, readable way, helping the domain expert in reaching a conclusion and discovering new information. The other use case of Data mining is predicting the outcome of some new event or instance data using the knowledge of previously seen data. This is often done by using the methods from the area of machine learning, statistics, evolutionary algorithms or database systems.

1.4. Predicting the marketing outcome using machine learning methods

The goal of this work is to predict whether some client will buy specific banking product upon offer. This way, the efficiency of marketing campaign could be increased. Many information about the clients are at disposal in order to make that decision. If this prediction is possible the bank can reach out only to specific subset of customers and this has two advantages:

- The customers that are likely not interested will not be bothered by the banking marketing campaigns.
- The bank will save the money on the marketing expenses.

Banks often have large amount of data about their customers and some of those information can be used for described targeted marketing. However, manually inspecting the data is usually impossible. Manual analysis of the data also relies too much on one's abilities and knowledge and it is not objective. That is why in the recent years a new data-driven approach to data analysis is getting more popular. Various machine learning algorithms learn from the data itself and the final product is a mathematical model that is able to generalize on new examples. In this work the following models models will be tested: Decision Tree, Random Forest and Gradient Boosting Trees. Detailed description of aforementioned methods is given in the chapter 3.

2. Dataset

2.1. Dataset description

There are 2 version of this dataset both available at [21]. The older version of the dataset was described in [18]. In this work the extended (newer) version of the dataset is used. Comprehensive description of used dataset was given in [19]. The attributes from the database can be seen in Table 2.1. The data was preprocessed in a number of ways before making it publicly available. The authors picked 20 features from the original 150 features. This was done in a semi-automated way by first asking the banking expert to manually choose the features that are considered important for this task. After that the features were further filtered by using a forward selection method [9]. Ten of these features are categorical type and ten of them are numerical type. The rows with missing data were removed and the final data consist out of 41188 rows where each rows represents one contact of some client. Only 12.7 % (4640 cases) of contacts resulted in the selling the bank long-term deposit. The goal of the work is the create a model that can pick a subset of clients such that this percentage is increased. This has to be done with a minimum impact on the total number of positive outcomes. To test the models' performance a 10 % of the dataset was left aside and was not in any way used in the training.

2.2. Dataset exploration

Before starting with the model development it is always a good practice to explore the dataset. It is crucial to identify problematic and useful attributes and values. Some of the dataset related issues which can cause further problems in development are:

- missing values
- non-existent values
- duplicate values

Table 2.1: Features (attributes) of the bank marketing dataset

Attribute name	**Description**
age	age of the client - numeric
job	type of job - categorical: admin, student,...
marital	marital status - categorical: divorced, married,...
education	education level - categorical: basic4y, high.school,...
default	whether the client has credit in default - categorical: yes, no, unknown
housing	whether the client has housing loan - categorical: yes, no, unknown
loan	whether the client has personal loan - categorical: yes, no, unknown
contact	offer communication type - categorical: cellular, telephone
month	month of the last contact - categorical: jan, feb,...
day_of_week	day of the last_contact - categorical: mon, tue,...
duration	last contact duration in seconds - numeric
campaign	number of contacts performed during this campaign and for this specific client - numeric
pdays	number of days that passed after the client was last contacted from a previous campaign - numeric
previous	number of contacts performed before this campaign and for this client - numeric
poutcome	outcome of the previous marketing campaign - categorical: failure, nonexistent, success
emp.var.rate	employment variation rate - quarterly indicator - numeric
cons.price.idx	consumer price index - monthly indicator - numeric
cons.conf.idx	consumer confidence index - monthly indicator - numeric
euribor3m	euribor 3 month rate - daily indicator - numeric
nr.employd	number of employees - quarterly indicator - numeric
y	**did the client subscribe for the long term deposit - categorical: yes, no**

- outliers

It this dataset the same client can be found in multiple rows only if multiple calls were made to this client. All the others duplicate rows were removed. There are several missing values in some categorical attributes, all coded with the "unknown" label. This category was not handled any different than other categories from the same row. One of the simple methods for getting the feel of the dataset and determining possible problems is to plot the data distribution and correlation. Calculating the correlation for the numerical data is quite straightforward and one of the common ways to calculate correlation in that case is by using a Pearson correlation coefficient:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.1}$$

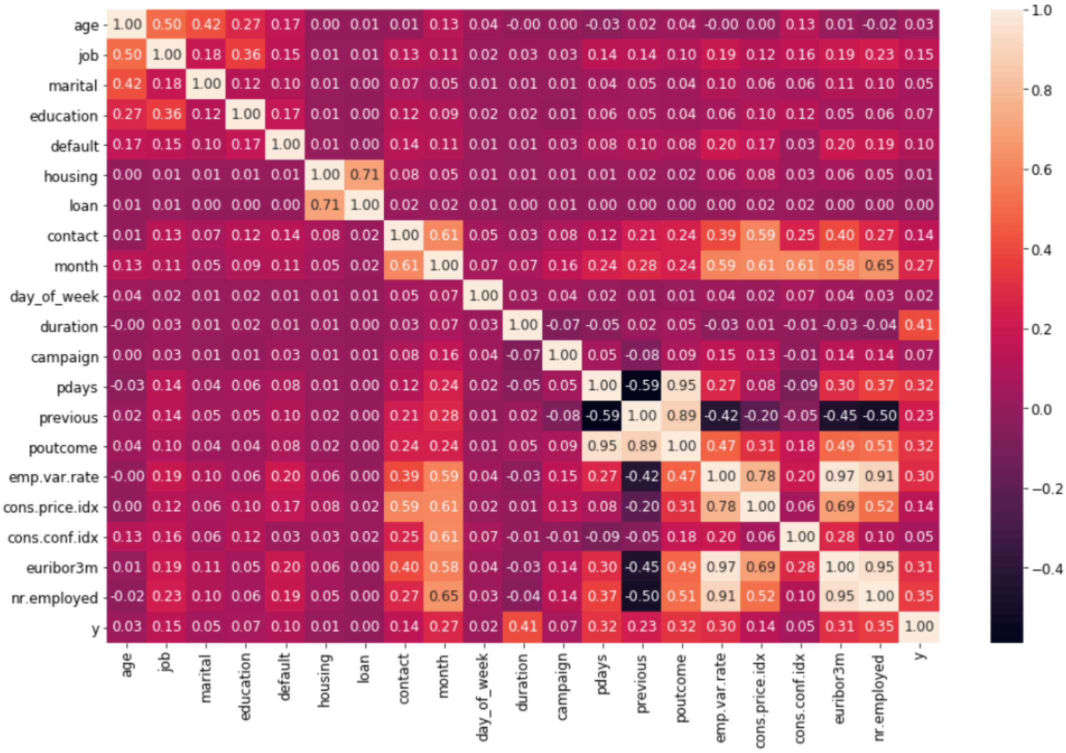
This coefficient results in a value between [-1,1]. In the case of categorical values it is not possible to use this coefficient so different approach is needed. Some popular methods for plotting the categorical variable features are:

- One-hot encoding of categorical values
- Cramér's V measure
- Theil's U (Uncertainty coefficient)

For dataset containing both numerical and categorical features Correlation ratio [22] measure can be used. The output of this measure is in the interval [0,1].

Correlation Ratio answers the following question: Given a continuous number, how well can you know to which category it belongs to? [20]

Correlation ratio for the bank marketing dataset used in this work can be seen in the figure 2.1. It can be seen that the duration variable is highly correlated with the target value (y). By plotting the probability density function of the attribute duration with respect to the target value (figure 2.2), it is easily seen that the distribution of the values belonging to the positive class differs from the distribution of the values belonging to the negative class. This also means that duration attribute will be very useful and indicative in the target value prediction. This was also noticed by the authors of the dataset but with a different method. They noticed this only after doing a sensitivity analysis of the trained neural network. It is logical that the duration of the made call impacts the success of the sales. After all, all the calls with duration value 0 will definitely have a negative outcome, and the calls with longer duration probably mean that the client was more interested. However this value can not be know before making a call and thus it can not be used for selecting the subset of clients which will be



	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
age	1.00	0.50	0.42	0.27	0.17	0.00	0.01	0.01	0.13	0.04	-0.00	0.00	-0.03	0.02	0.04	-0.00	0.00	0.13	0.01	-0.02	0.03
job	0.50	1.00	0.18	0.36	0.15	0.01	0.01	0.13	0.11	0.02	0.03	0.03	0.14	0.14	0.10	0.19	0.12	0.16	0.19	0.23	0.15
marital	0.42	0.18	1.00	0.12	0.10	0.01	0.00	0.07	0.05	0.01	0.01	0.01	0.04	0.05	0.04	0.10	0.06	0.06	0.11	0.10	0.05
education	0.27	0.36	0.12	1.00	0.17	0.01	0.00	0.12	0.09	0.02	0.02	0.01	0.06	0.05	0.04	0.06	0.10	0.12	0.05	0.06	0.07
default	0.17	0.15	0.10	0.17	1.00	0.01	0.00	0.14	0.11	0.01	0.01	0.03	0.08	0.10	0.08	0.20	0.17	0.03	0.20	0.19	0.10
housing	0.00	0.01	0.01	0.01	0.01	1.00	0.71	0.08	0.05	0.01	0.01	0.01	0.01	0.02	0.02	0.06	0.08	0.03	0.06	0.05	0.01
loan	0.01	0.01	0.00	0.00	0.00	0.71	1.00	0.02	0.02	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.00	0.00
contact	0.01	0.13	0.07	0.12	0.14	0.08	0.02	1.00	0.61	0.05	0.03	0.08	0.12	0.21	0.24	0.39	0.59	0.25	0.40	0.27	0.14
month	0.13	0.11	0.05	0.09	0.11	0.05	0.02	0.61	1.00	0.07	0.07	0.16	0.24	0.28	0.24	0.59	0.61	0.61	0.58	0.65	0.27
day_of_week	0.04	0.02	0.01	0.02	0.01	0.01	0.01	0.05	0.07	1.00	0.03	0.04	0.02	0.01	0.01	0.04	0.02	0.07	0.04	0.03	0.02
duration	-0.00	0.03	0.01	0.02	0.01	0.01	0.00	0.03	0.07	0.03	1.00	-0.07	-0.05	0.02	0.05	-0.03	0.01	-0.01	-0.03	-0.04	0.41
campaign	0.00	0.03	0.01	0.01	0.03	0.01	0.01	0.08	0.16	0.04	-0.07	1.00	0.05	-0.08	0.09	0.15	0.13	-0.01	0.14	0.14	0.07
pdays	-0.03	0.14	0.04	0.06	0.08	0.01	0.00	0.12	0.24	0.02	-0.05	0.05	1.00	-0.59	0.95	0.27	0.08	-0.09	0.30	0.37	0.32
previous	0.02	0.14	0.05	0.05	0.10	0.02	0.00	0.21	0.28	0.01	0.02	-0.08	-0.59	1.00	0.89	-0.42	-0.20	-0.05	-0.45	-0.50	0.23
poutcome	0.04	0.10	0.04	0.04	0.08	0.02	0.00	0.24	0.24	0.01	0.05	0.09	0.95	0.89	1.00	0.47	0.31	0.18	0.49	0.51	0.32
emp.var.rate	-0.00	0.19	0.10	0.06	0.20	0.06	0.00	0.39	0.59	0.04	-0.03	0.15	0.27	-0.42	0.47	1.00	0.78	0.20	0.97	0.91	0.30
cons.price.idx	0.00	0.12	0.06	0.10	0.17	0.08	0.02	0.59	0.61	0.02	0.01	0.13	0.08	-0.20	0.31	0.78	1.00	0.06	0.69	0.52	0.14
cons.conf.idx	0.13	0.16	0.06	0.12	0.03	0.03	0.02	0.25	0.61	0.07	-0.01	-0.01	-0.09	-0.05	0.18	0.20	0.06	1.00	0.28	0.10	0.05
euribor3m	0.01	0.19	0.11	0.05	0.20	0.06	0.00	0.40	0.58	0.04	-0.03	0.14	0.30	-0.45	0.49	0.97	0.69	0.28	1.00	0.95	0.31
nr.employed	-0.02	0.23	0.10	0.06	0.19	0.05	0.00	0.27	0.65	0.03	-0.04	0.14	0.37	-0.50	0.51	0.91	0.52	0.10	0.95	1.00	0.35
y	0.03	0.15	0.05	0.07	0.10	0.01	0.00	0.14	0.27	0.02	0.41	0.07	0.32	0.23	0.32	0.30	0.14	0.05	0.31	0.35	1.00

Figure 2.1: Correlation of the attributes

targeted. This means that the duration attribute should be removed from the dataset in order to get realistic results.

2.3. Feature selection

One easy way to determine the importance of some feature is to train a classifier and then check how the final score will be change if some feature is removed. If some feature is important removing it from the input data will significantly decrease the performance of the model (whichever metric we choose to evaluate the performance of the model). We can scale all the features coefficients so that they add up to one and then visualize those coefficient. In figure 2.3a coefficients of all the features are shown and in figure 2.3b only the first half is visualized. It can be seen that model trained only on the first half of the features is still able to perform 90% as good as the model that is trained on all of the features. This can be used to select the most informative features and reduce the time of training and inference which can be useful when lot of the data is available and large models are used. From the feature importance plots it is also possible to confirm that classifiers rely a lot on duration feature since its importance coefficient is 0.10 (the second largest from all of the features). As already

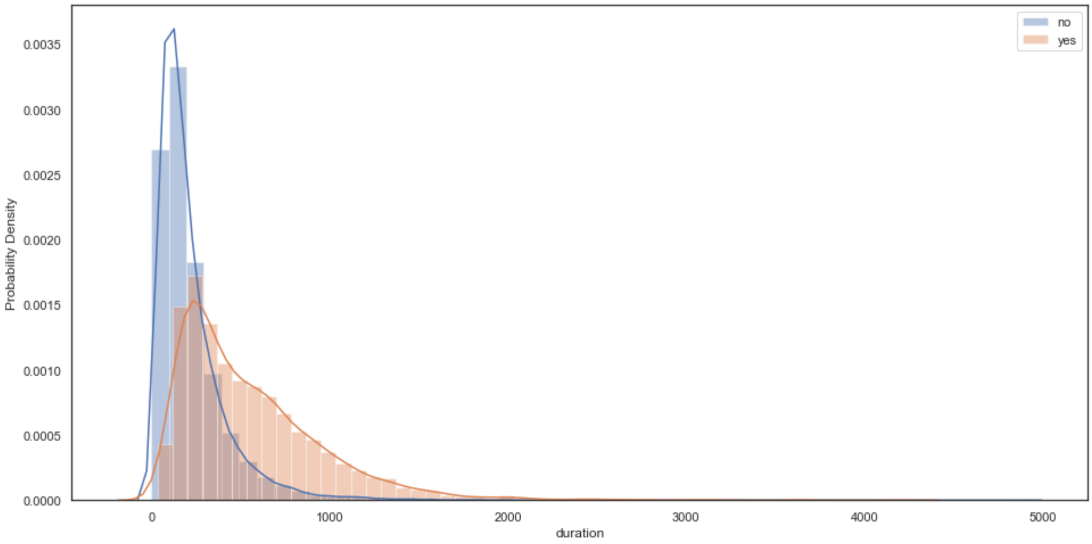
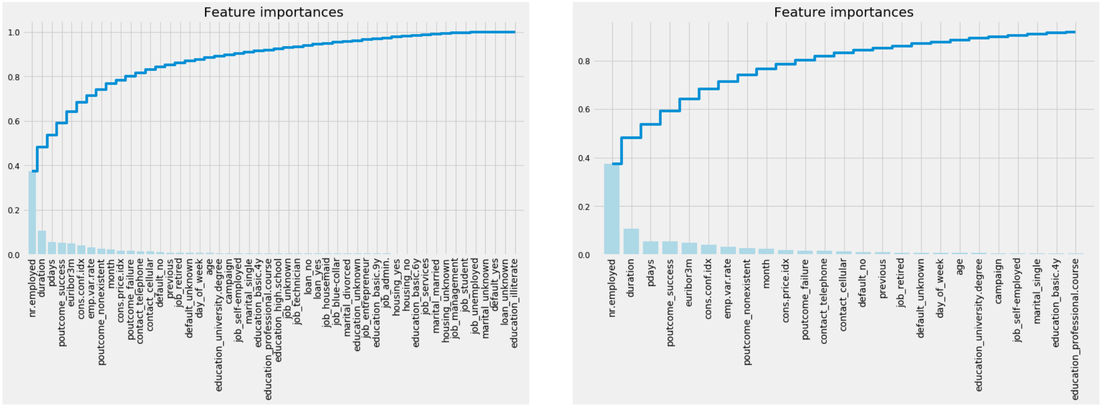


Figure 2.2: Distribution of duration attribute value



(a) all features coefficients

(b) first half of the coefficients

Figure 2.3: Feature importance

stated this feature will not be used during the model training to provide significant, objective results.

3. Machine Learning methods

3.1. Decision Tree

Decision Tree is one of the common data mining methods used for determining the outcome based on an input feature set. More specifically it is one of the supervised machine learning techniques. It is called a Decision Tree because it can be displayed as a tree structure, typically drawn upside down, in which the observations about an item are represented in the branches and the outcomes are represented in the leaves of the tree. Since the Decision Tree representation is a good analogy with an actual decision making process it is easy to understand and interpret obtained results. This is one of the main advantages of using a Decision Tree over other classification methods. Decision Trees can be applied to both regression and classification problems. In this work a Decision Tree is used for binary classification, but in a general case, Decision Tree can be applied for classification of any number of categories. In the branches of the tree it is possible to see the features used to make a decision and they are usually displayed in a way that the most important features are found at the top of the tree. It is very easy to understand why some decision was made just by looking at the graphic representation of the tree. After the learning process we get a tree-like structure with root at the top. Each node represents a subset of the training dataset starting from the root node which represents entire population (all the instances from the dataset). In each node a decision is made depending on some attribute value and the current population (subset of the dataset) is divided into new subsets. The attributes and exact values which are used for splitting are found during the building process which is described in the next section.

3.1.1. Building a Decision Tree

The first node of the tree is called the root. Final nodes of the tree (nodes in which the splitting stops) are called leaf nodes. From the root the splitting is done depending on the values of some attributes. In order to select a good attribute, an impurity metric which determines how good some attribute is needs to be used. There are many impurity metrics that can be used. For example:

- Gini index (used in CART)
- Entropy
- Information gain (used in ID3)
- Gain Ratio (used in C4.5)
- Reduction in Variance
- Chi-square

Some of these metrics are more suitable for categorical values and some for numerical. The most popular choices for this metric are Gini index (for numerical data) and information gain (for categorical or mixed data). Regardless of the used metric, the goal is always the same: picking an attribute from features such that the nodes obtained after splitting are as pure as possible. That means that the distribution of classes in some node should be as homogeneous as possible. Figure 3.1 shows and example of Decision Tree of level 1 built by using an Gini index. The attribute used for splitting the root node was "duration". That node was chosen because it had the best Gini index compared to other attributes. Since the Gini index metric measures the total impurity the best attribute is the one that has the lowest Gini index. Gini indices of the nodes in the Fig 3.1 were calculated in the following way:

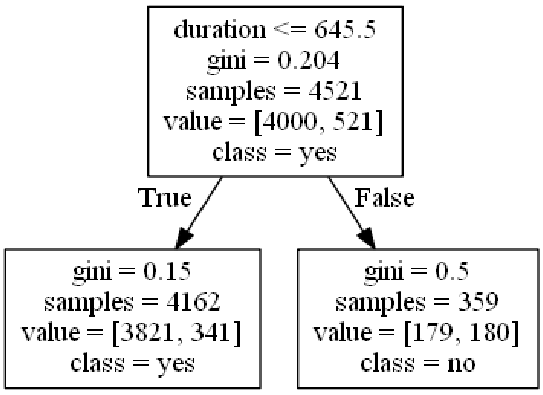


Figure 3.1: Example of simple Decision Tree

$$G_i = 1 - sum(p)^2$$

Gini index of the left node is :

$$G_i = 1 - (\frac{3821}{4162})^2 - (\frac{341}{4162})^2 = 0.1504$$

Gini index of the right node is

$$G_i = 1 - (\frac{179}{359})^2 - (\frac{180}{359})^2 = 0.499$$

The total Gini index for the attribute duration is calculated as a weighted sum of Gini indices of nodes. The weight are determined from the sizes of the subsets:

$$G_i : \frac{4162}{4521} * 0.15 + \frac{359}{4521} * 0.5 = 0.178$$

Gini index before making a split was 0.204 (as seen from the Figure) so making this split improves the impurity metric of the leaf nodes so it makes sense and it will be executed. The process of node splitting will be continued as long as we can improve the leaf node impurity or until some stopping criterion is met. If a built Decision Tree is too complicated it is more likely have a poor generalization ability. There are many know methods to handle this disadvantage (overfitting) and they can be divided into two categories:

- Prevent the algorithm from building a complex tree by introducing some early stopping mechanism (number of leafs, tree depth, etc.)
- Build a complex tree but then reduce its complexity. (Tree pruning)

By using these methods one can make sure the tree doesn't have a large bias or large variance.

3.2. Ensemble methods

Ensemble models are models that are obtained by combining multiple base models into a new better model. This base models are usually referred to as weak learners and their ensemble a strong learner. Even if none of the weak learners are able to achieve good performance, their combination might be able to do so. Diversification among the weak learners helps the ensemble model to reach better performance than any of its components could achieve. This means that the ensemble model's performance will be better if the base model fail in predicting different instances of some dataset. This way there will always be at least one weak learner that is able to correctly predict the outcome of some input that some other weak learner was not able to predict correctly. In the following sections ensemble methods based on Decision Tree weak learners will be consider. Decision Trees have been around for a long time and also known to suffer from bias and variance. Simple trees have large bias while the complex Decision Trees have large variance[1]. Ensemble method try to deal with these disadvantages and there are two commonly used techniques of creating an ensemble from multiple homogeneous weak learners:

- Bagging (**B**ootstrap **agg**regat**ing**)
- Boosting

These techniques are discussed in the following sections.

3.2.1. Bagging

Bootstrap aggregation or Bagging is an approach used to deal with the high variance of a basic Decision Tree model. This technique is used to improve the stability and accuracy of the base model and helps with the reduction of the base model overfitting. In order to get a final decision, prediction from multiple weak learners are used. Used weak learners are obtained by training on different subsets that are sub-sampled from the full original training dataset. Training samples that are picked from the original dataset are replaced, thus the name: bootstrap aggregation (Bootstrap is a statistical technique used to generate samples of some size from an initial dataset by randomly drawing observations with replacement [11]). This method for sampling is used in practice because we want to train many independent models but the number of observations in the dataset is limited and not big enough to divide into subsets that could be used to successfully train independent models.

Random Forest is one of the bagging technique that uses Decision Tree model as a base weak learner. This ensemble method is based on bagging procedure but it has one extra step in the process of selecting the subsets from which the base learners will be trained. Instead of just taking random instances (rows of the data), random attributes (columns of the datasets) are also selected. For regression task the output is a weighted average of the base learners' outputs:

$$output(SL) = \frac{1}{N} sum_{n=1}^{N} output(WL)$$

For classification task the majority voting is used:

$$output(SL) = argmax[card(l|ouput(WL) = k]$$

This approach may not give satisfying results when used for regression task because the averaging process could diminish the success of the good base model. However, in the classification task it usually leads to an improved performance compared to the basic models. Since the subsets of the dataset and feature can be selected beforehand, all the base models can be trained in parallel and they don't depend on each other performances. This characteristic is the one that differs the most bagging methods from boosting methods.

3.2.2. Boosting

Boosting technique is an approach used to deal with the high bias of a basic Decision Tree model. However boosting can also lead to a variance decrease in some cases. Unlike bagging, base models are learned sequentially with early learners fitting simple models to the data and then analyzing data for errors. Consecutive trees are fitted at every step, and in each step the goal is to reduce the error from the prior tree [1].

Gradient Boosting is an extension over boosting method. It can be used to improve basic regression or classification models. Typically a Decision Tree model is used as a weak classifier. In order to use the gradients for optimizing the ensemble, a loss function must be specified. Any differentiable function can be used for this purpose but it should of course objectively capture how good some models is. This way optimizing this function (usually minimizing) would lead to better model performance. The idea of gradient boosting was first proposed in the work of Leo Breiman [3]. This idea continued to develop and it was the topic of many articles including [16] [17] which introduced the idea of gradient descent algorithms. This type of algorithms iteratively pick a cost functions (often called loss function or objective function) and then calculate the gradients by differentiating the wanted output with the one gotten by the model. In this work an improved version of gradient boosting algorithm was used. It is called **Extreme gradient boosting (XGBoost)** and this algorithm can often be seen among winning methods in various classification and regression task competitions (like Kaggle competitions). This algorithm was proposed in 2016 in [5]. Extreme gradient boosting (XGBoost) improves basic gradient boosting approach through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting. Figure 3.2 visually shows all the enhancement that were proposed in the XGboost article. Since Extreme gradient boosting method is very popular it is not surprising there are many popular python libraries implementing this method. Some of them are:

- XGBoost [25]
- LightGBM [14] (based on [12])
- H2O [10]
- Catboost [4] (based on [7])

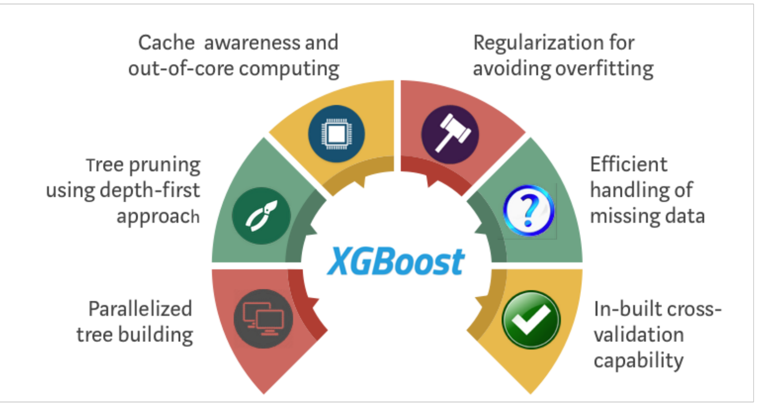

Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient handling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

Figure 3.2: XGBoost introduced improvements
source: [23]

In this work, a LightGBM implementation was used.

4. Results

4.1. Evaluation metrics

There are many different metrics that can be used for evaluating the classification model. Commonly used ones are: accuracy, precision, recall, F1-score, ROC and AUC but there are also many other that can be used when these metrics aren't appropriate. Picking the correct metric is crucial in order to get an objective measure of model performance. For example, choosing accuracy to evaluate models from this work would be a mistake because the dataset is not balanced. Even the classifier that always predicts 'no' as an output would have accuracy over 85% since more than 85% of data samples have negative outcome. One of the commonly used metrics in target marketing is cumulative gain. This metric can be used to compare the performances of binary classifiers. It is very useful because it shows the amount of population that needs to be selected in order to get some response. Figure 4.1 shows an example of cumulative gain curve that was obtained with the Decision Tree classifier.

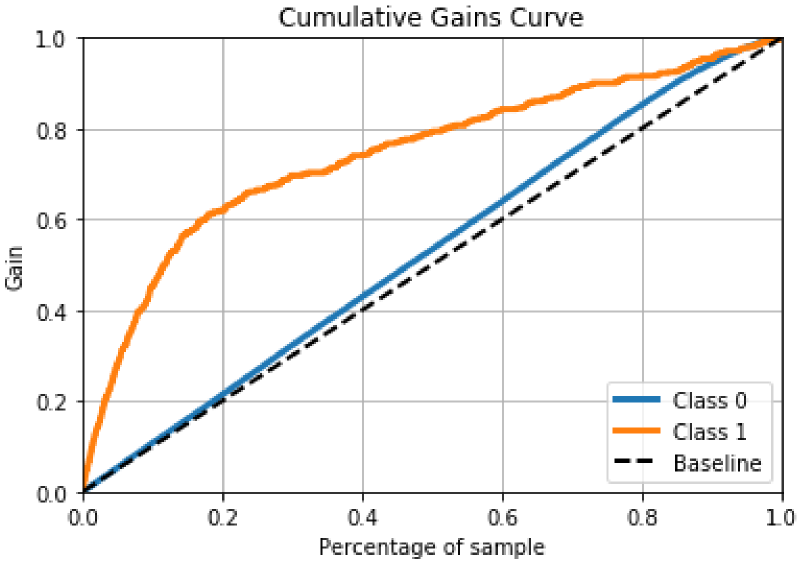


Figure 4.1: Example of cumulative gain curve

Cumulative gain measure can be calculated from true positive rate (Eq. 4.1) and support (Eq. 4.2) measures:

$$tpr = \frac{TP}{TP + FN} \quad (4.1) \qquad sup = \frac{TP + FP}{N} \quad (4.2)$$

To plot the cumulative gain curve the prediction of the model are first sorted by their probability score. This means that for a good classifier most of the positive prediction will be in the beginning of the list. The cumulative gain curve is obtained by taking examples from this list and in each step (in which we consider one more new example) support and tpr measures are calculated. As the number of examples taken into consideration is increasing the tpr will also be increasing. Support measure is plotted in the x-axis of the chart and tpr measure is plotted in the y-axis.

Another way to visualize the performance of the model is to use the lift charts. Lift chart shows a different perspective on the same result. In marketing terminology, the increase in response rate, is known as a lift factor yielded by the learning tool (model). This factor is often combined with the cost of marketing and that combination is used to determine the payoff implied by a particular lift factor [24]. The cost of marketing is not know for this particular work so lift chart shows equivalent information as cumulative gain curve. In real life scenario lift chart with incorporated cost of the marketing can be a very useful tool for a manager to determine the exact amount of people that will be targeted in some marketing campaign.

4.2. Results

As already discussed in the previous section, the best way to evaluate the performance of some classifier for the task of targeted marketing is by looking at the cumulative gain curve. A chart showing cumulative gain curves of all the models used in this work is shown in the figure 4.2. It can be seen that XGBoost method is superior than other two methods, especially if some particular interval is taken into consideration (for example between 0.4 and 0.6). The lift chart for these three models is shown in 4.3. It shows that targeting small subsets of clients with the help of methods proposed in this work improves the efficiency of marketing by several times. Even though it is possible to get a feeling of which model performs the best by looking at presented charts, objective way to determine this would be to measure the areas underneath cumulative gain

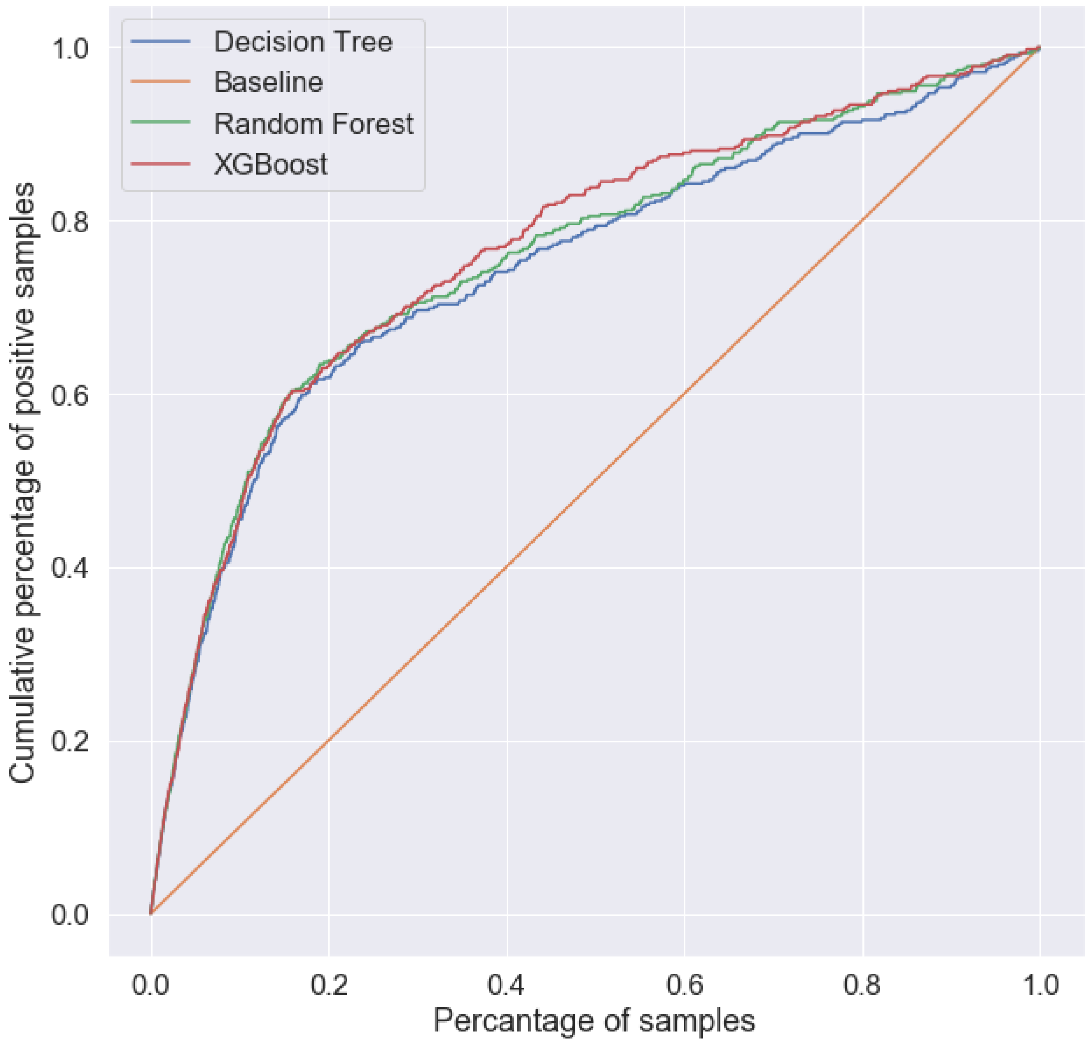


Figure 4.2: Cumulative gain curve for all the classifiers

curves. This metric is shown in the table 4.1.

XGBoost method has the largest area under the surface but it doesn't mean it will outperform other models for every size of picked clients. To demonstrate this, table 4.2 is presented. This table shows how many clients need to be picked from the whole population in order to achieve wanted response. The first threshold of 11% was chosen because it is the percentage of positive outcomes calculated from the test subset. It can be seen that the usage of machine learning models can lead to a drastic decrease in the number of reached clients without affecting the number of positive campaign outcomes. The best performing classifier for each row was highlighted with a bold text font. Even though XGBoost was determined to be the best overall model from the table 4.1, this table shows that XGBoost isn't always the best choice. There are some intervals for which the Random Forest classifier performs better. There is also

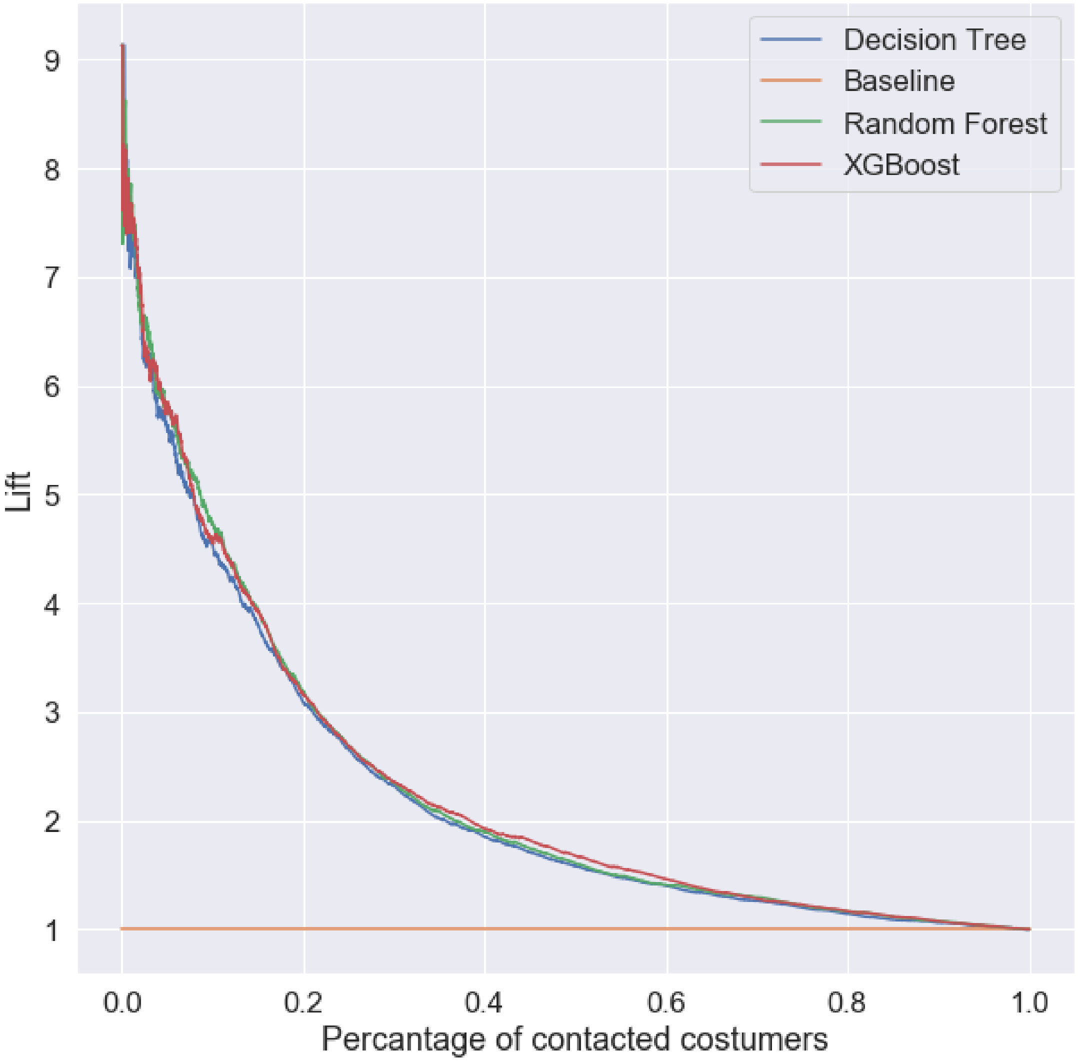


Figure 4.3: Lift chart for all the classifiers

a possibility that there is some other method (like SVM, Neural network etc.) that would perform better than presented models. However, models that were discussed in this work were picked because they can easily be interpreted. This characteristic can often mean more than a slight increase in performance that some (probably more complex) classifier could achieve.

Classifier	Decision Tree	Random Forest	XGBoost
Area under the cumulative gain curve	74.48%	75.8%	**76.5%**

Table 4.1: Area under the curve

Percentage of positive response (recall)	Decision tree	Random forest	XGBoost
11%	1.58%	**1.51%**	**1.51%**
25%	4.37%	4.22%	**4.18%**
50%	11.68%	**10.83%**	10.88%
75%	41.37%	39.38%	**36.15%**
90%	72.95%	**68.83%**	71.23%

Table 4.2: Percentages of picked clients for particular recall values

5. Conclusion

Many companies perform marketing campaigns in order to attract new clients or to strengthen their position in the market. Sometimes, campaigns are done to offer some new product or service to existing clients. Marketing campaigns always have some costs included so it is crucial to make them as efficient as possible. This can be done by selecting subsets of clients that are likely to be interested in some new offer. However, selecting interested clients is not trivial and usually cannot be done manually. The goal of this work was to research methods that could increase the efficiency of selling long term banking deposit. Data from a Portuguese banking institution, collected from May 2008 until November 2010, was used for this purpose. Three different methods were analyzed, namely: Decision Tree, Random Forest and XGBoost. To evaluate models' performances, different metrics were proposed. XGBoost showed the best overall results, but all of the methods showed a big improvement compared to random picking clients. It was also shown how these metrics could help a manager with marketing optimization and improve the efficiency of long-term deposit sales by several times.

6. Bibliography

[1] Anuja Nagpal. Decision Tree Ensembles- Bagging and Boosting. https://towardsdatascience.com/decision-tree-ensembles-bagging-and-boosting-266a8ba60fd9, 2017. Accessed: 2020-04-05.

[2] Chidanand Apte, Bing Liu, Edwin P. D. Pednault, i Padhraic Smyth. Business applications of data mining. *Communications ACM*, 45(8):49–53, Kolovoz 2002. ISSN 0001-0782. doi: 10.1145/545151.545178. URL https://doi.org/10.1145/545151.545178.

[3] Leo Breiman. Arcing the edge. Technical report, 1997.

[4] catboost developers. CatBoost Documentation. https://catboost.ai/, 2020. Accessed: 2020-04-13.

[5] Tianqi Chen i Carlos Guestrin. Xgboost: A scalable tree boosting system. stranice 785–794, 08 2016. doi: 10.1145/2939672.2939785.

[6] Nedim Dedić i Clare Stanier. Measuring the success of changes to existing business intelligence solutions to improve business intelligence reporting. U A Min Tjoa, Li Da Xu, Maria Raffai, i Niina Maarit Novak, urednici, *Research and Practical Issues of Enterprise Information Systems*, stranice 225–236, Cham, 2016. Springer International Publishing. ISBN 978-3-319-49944-4.

[7] Anna Veronika Dorogush, Vasily Ershov, i Andrey Gulin. Catboost: gradient boosting with categorical features support, 2018.

[8] Margaret Dunham i Dr.Sridhar Seshadri. *Data Mining- Introductory and Advanced Topics*. 01 2006. ISBN 9788177587852.

[9] Isabelle Guyon i André Elisseeff. An introduction of variable and feature selection. *Journal of Machine Learning Research Special Issue on Variable and Feature Selection*, 3:1157 – 1182, 01 2003. doi: 10.1162/153244303322753616.

[10] h2o developers. H2O XGBoost Documentation. `http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/xgboost.html`, 2020. Accessed: 2020-04-13.

[11] Joseph Rocca. Ensemble methods: bagging, boosting and stacking. `https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205`, 2019. Accessed: 2020-04-05.

[12] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, i Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. U I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, i R. Garnett, urednici, *Advances in Neural Information Processing Systems 30*, stranice 3146–3154. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf`.

[13] Nissan Levin i Jacob Zahavi. Data mining for target marketing. U *Data Mining and Knowledge Discovery Handbook*, 2005.

[14] LightGBM developers. LightGBM Documentation. `https://lightgbm.readthedocs.io/en/latest/`, 2020. Accessed: 2020-04-13.

[15] Charles X. Ling i Chenghui Li. Data mining for direct marketing: Problems and solutions. U *KDD*, 1998.

[16] Llew Mason, Jonathan Baxter, Peter Bartlett, i Marcus Frean. Boosting algorithms as gradient descent. U *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, stranica 512–518, Cambridge, MA, USA, 1999. MIT Press.

[17] Llew Mason, Jonathan Baxter, Peter Bartlett, i Marcus Frean. Boosting algorithms as gradient descent in function space, 1999.

[18] Sergio Moro, Raul Laureano, i Paulo Cortez. Using data mining for bank direct marketing: An application of the crisp-dm methodology. U *Proceedings of European Simulation and Modelling Conference-ESM'2011*, stranice 117–121. EUROSIS-ETI, 2011.

[19] Sérgio Moro, Paulo Cortez, i Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22 – 31, 2014. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2014.03.001. URL http://www.sciencedirect.com/science/article/pii/S016792361400061X.

[20] Shaked Zychlinski. The Search for Categorical Correlation. https://towardsdatascience.com/the-search-for-categorical-correlation-a1cf7f1888c9, 2018. Accessed: 2020-03-06.

[21] Sérgio Moro and Paulo Cortez and Paulo Rita. Bank Marketing Data Set. http://archive.ics.uci.edu/ml/datasets/Bank+Marketing#, 2014. Accessed: 2020-02-26.

[22] unknown. Correlation ratio. https://en.wikipedia.org/wiki/Correlation_ratio, 2018. Accessed: 2020-02-26.

[23] Vishal Morde. XGBoost Algorithm: Long May She Reign! https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d, 2019. Accessed: 2020-04-13.

[24] Ian H. Witten, Eibe Frank, i Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd izdanju, 2011. ISBN 0123748569.

[25] xgboost developers. XGBoost Documentation. https://xgboost.readthedocs.io/en/latest/, 2020. Accessed: 2020-04-13.