Report I

Anna Szymanek (230042), Patryk Wielopolski (234891)

Introduction

In this report we will be analysing the data [1] related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

This specific problem is extremly important for call centers to manage their limited human resoruces. In order to make such a campaign succesful we have to point out people who are most likely to subscribe bank term deposit.

Our goal of this project is to conduct data analysis to better understand available data for modeling and then create classifier which will be correctly assigning classes.

Structer of this document is as follows. In Section 2 we will describe methods used to perform whole modeling process. In the next section we will analyse our results and sum up whole project.

Methods

This section is divided to 3 parts - data description, data analysis, classification.

In first subsection we will describe dataset from official documentation which will give us general overview about dataset and information about variables meaning and types. Also we might there spot some interesting facts which could be interesting for us from modeling perspective, i.e. variable coding or they might give us real-world application context.

Then in the next subsection we will conduct statistical analysis of given dataset. After this part we would like to know basic properties of variables / features (range, properties, distribution), find all missing values and outliers, be familiar with correlations between features in the dataset and give initial assessment of discriminative ability of consecutive features (i.e. ability to separate objects from different classes).

In the last subsection we will describe our approach to the modeling task. We will define our objective, describe when, where, and how was the study done, what materials were used and who was included in the study groups. Also we will describe methods and algorithms used in the project.

Data description

Telemarketing is one of the forms used to encourage clients to buy new bank's product. If we imagine real-world scenario it may be very hard to decide to which customer we should call in order to achive our goal (in this case bank term deposit subscription) because it's not possible to call them all as we have limited human resources of telemarketers. Such in this case we could use historical data about calls done in previous marketing campaigns to formulate conclusions about what type of clients are our target group and what part of the day / week / year is a good time for such a projects. Moreover after that we can create classification models which will learn to give us good recomendations which clients we should call in the first order.

Variable name	Type	Description
Bank client data		
Age	Numeric	Age of client
Job	Categorical	Type of job
Marital	Categorical	Marital status of client
Education	Categorical	Education status of client
Default	Categorical	Has credit in default?
Housing	Categorical	Has housing loan?
Loan	Categorical	Has personal loan?
Variables related with the last contact of the current campaign		
Contact	Categorical	Contact communication type
Month	Categorical	Last contact month of year
Day of week	Categorical	Last contact day of the week
Duration	Numeric	Last contact duration, in seconds
Other attributes		
Campaign	Numeric	Number of contacts performed during this campaign
Pdays	Numeric	Number of days that passed by after the client was last contacted from a previous campaign
Previous	Numeric	Number of contacts performed before this campaign
Poutcome	Categorical	Outcome of the previous marketing campaign
Social and economic context attributes		
Emp.var.rate	Numeric	Employment variation rate - quarterly indicator
Cons.price.idx	Numeric	Consumer price index - monthly indicator
Cons.conf.idx	Numeric	Consumer confidence index - monthly indicator
Euribor3m	Numeric	Euribor 3 month rate - daily indicator
Nr. employed	Numeric	Number of employees - quarterly indicator
Outcome variable		
y	Categorical	Has the client subscribed a term deposit?

Table 1: Input variables.

In general our dataset can be splitted in five categories: bank client data, last contact of the current campaign, other attributes, social and economic context, outcome as we can observer in Table 1. First category describes general information about client - age, job, etc. Second category describes how the last contact with client was performed. There is also important note that Duration attribute highly affects the output target (e.g., if duration=0 then y="no"). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and we will discard it because our intention is to have a realistic predictive model. Third category in general tell us about previous campaigns and previous contacts with given person. There is also a note from dataset authors that Pdays equal 999 means client was not previously contacted. We will also note that in our data. Fourth category is about social and economic context attributes such as employment rate. This might give us information how was economy in this time and might be driving factor for some people. The last category is our outcome variable, i.e. flag if the client subsribed a term deposit.

Data analysis

We will begin our data analysis with some important numbers about our dataset. As we can observe in Table 2 we have 41188 rows and 21 with 11 discrete columns and 10 continuous columns which agrees with data description. Luckily, we don't have all missing columns.

Now, let's take a closer look into missing values. It's very important because most of the classification algorithms doesn't support missing values and in case of such variables we should somehow deal with them. Based on our results from Table 3 we can observe that we have only 2 columns which has any missing value -

	Info
rows	41188.00
columns	21.00
discrete_columns	11.00
continuous_columns	10.00
all_missing_columns	0.00
total_missing_values	73098.00
complete_rows	0.00
total_observations	864948.00
memory_usage	6763504.00

Table 2: Basic summary about dataset.

	feature	num_missing	pct_missing
1	age	0	0.00
2	job	0	0.00
3	marital	0	0.00
4	education	0	0.00
5	default	0	0.00
6	housing	0	0.00
7	loan	0	0.00
8	contact	0	0.00
9	month	0	0.00
10	day_of_week	0	0.00
11	duration	0	0.00
12	campaign	0	0.00
13	pdays	39673	0.96
14	previous	0	0.00
15	poutcome	0	0.00
16	emp.var.rate	0	0.00
17	cons.price.idx	0	0.00
18	cons.conf.idx	0	0.00
19	euribor3m	0	0.00
20	nr.employed	33425	0.81
21	y	0	0.00

Table 3: Basic summary about missing values in dataset.

pdays, nr.employed.

Let's begin with the second one - *nr.employed*. We can see that almost 81% of values are missing. In Table 4 we can observe that we have only missing value and value 5191 and there is no special difference between frequencies in class of y variable. Probably that may be an error during data completion. Because of that observations we want to exclude that variable from further analysis.

Let's go back to the first variable - *pdays*. It's more interesting because that was us who made that number of missing values, in particular 96% of all records. As we remember missing values was assigned to clients who has never been called by telemarketers. If we look into Table 5 it turns out that more than half of the customers bought the product. So in our context it would be more convinient to analyse client who was never reached by telemarketers in the previous campaigns. In order to that we will select only that subset of clients and drop that column.

	nr.employed	y	n
1	0.00	no	29025
2	0.00	yes	4400
3	5191.00	no	7523
4	5191.00	yes	240

Table 4: Analysis of missing values of nr.employed

	pdays	y	n
1	FALSE	no	548
2	FALSE	yes	967
3	TRUE	no	36000
4	TRUE	yes	3673

Table 5: Analysis of missing values of pdays

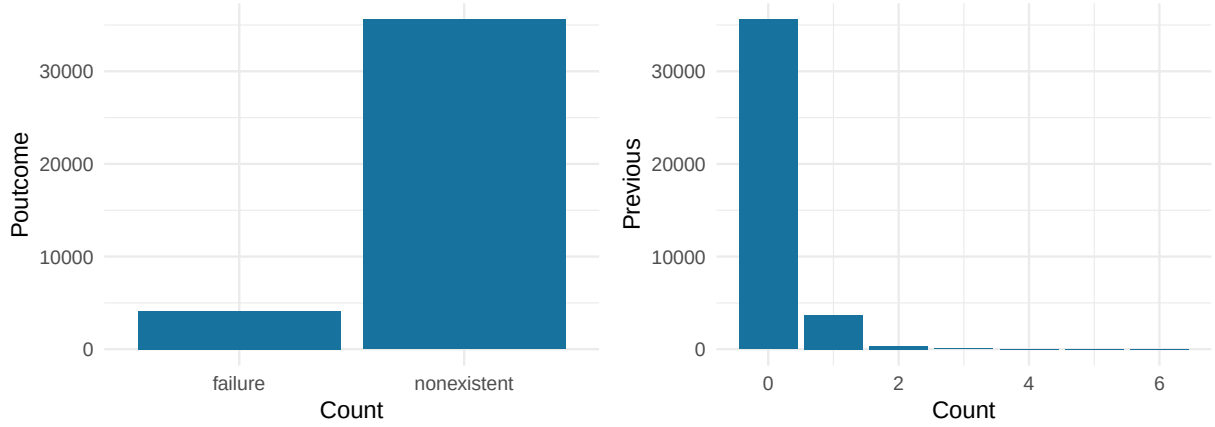

Figure 1: Analysis of linked variables to *pdays* - *poutcome*, *previous*.

But the column *pdays* is strictly connected to other two columns: *previous* and *poutcome*. They should probably also be droped because if there was no contact before there should be values indicating no contact. As we could see on Figure 1 that's not the case in our situation. We have some group of people who were contacted before, even more than one time, but it was a failure. We want to follow the set goal, i.e. only analyse first-contacted in this campaign people, so we will filter out these observations and drop these two columns.

Also before we will go the visualization part we want to mention column *Duration* which was stated that it's not known before contact so we also want to exclude it from further analysis.

After this transformations our dataset has 35563 rows and 16 columns. Moreover our modeling goal is more specified and now we want to predict which new clients who we never called before in the previous campaigns we want to target.

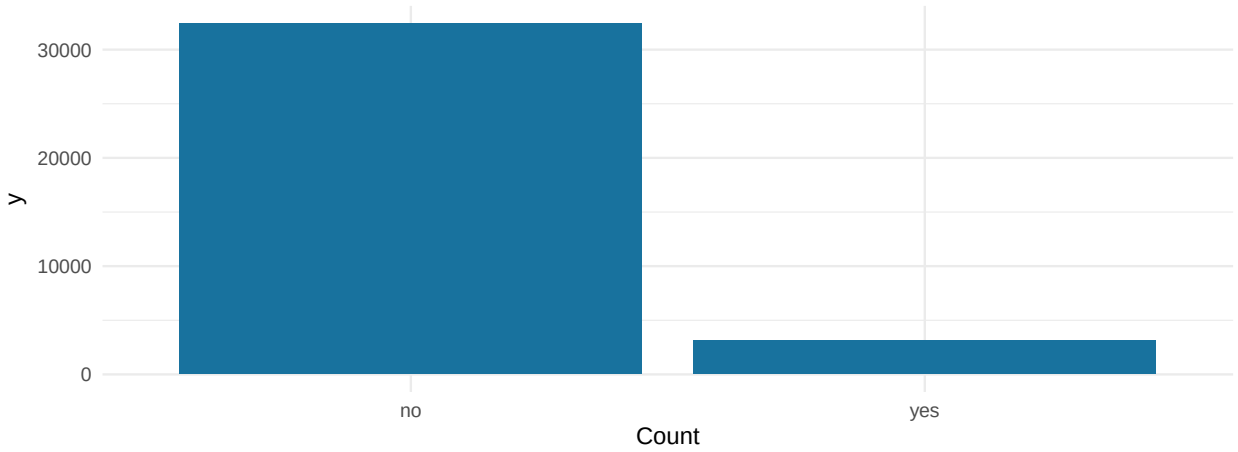

Figure 2: Analysis of target variable.

Now, we will conduct univariate analysis of the dataset. Let's begin with our target value y. As we can observe on Figure 2 we have higly imbalanced data which may cause problems during modeling part but we will take care of that in next section of the project.

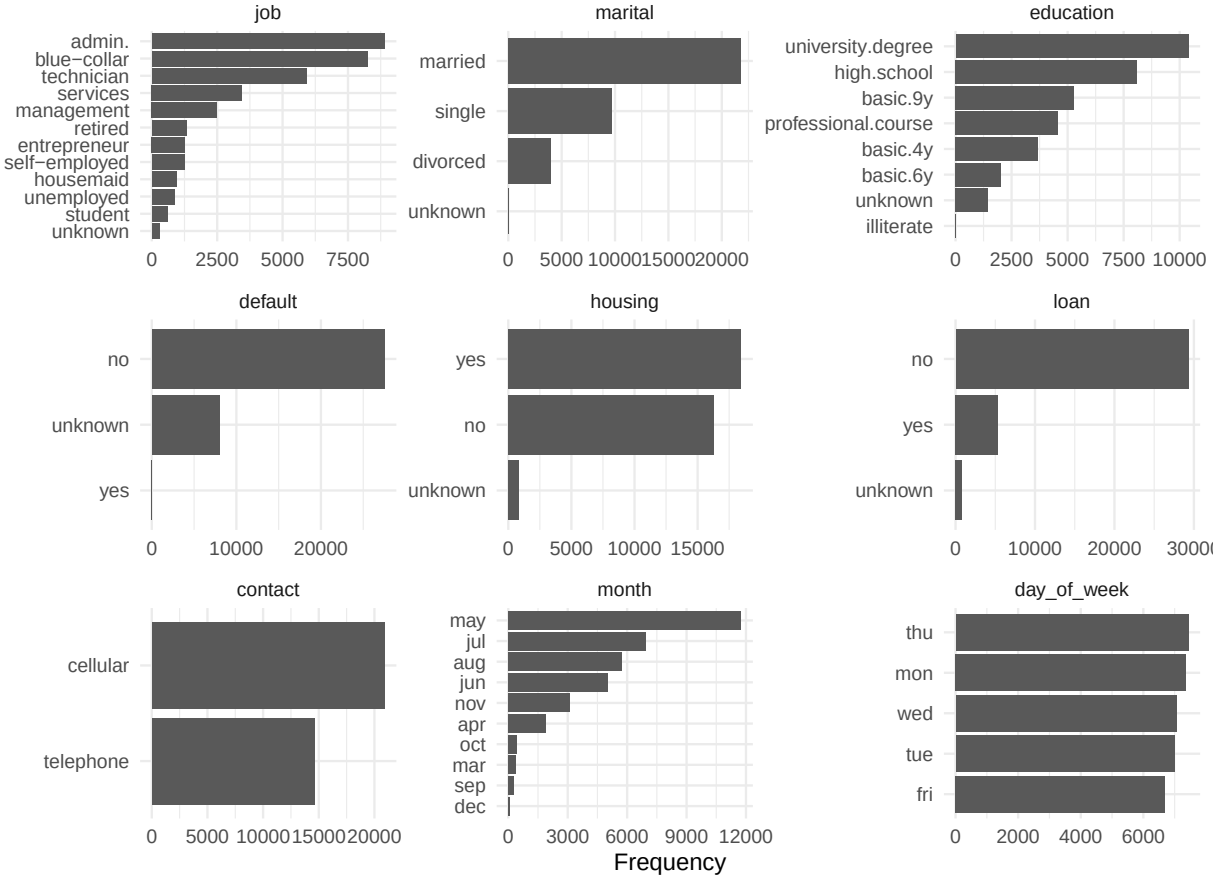

Figure 3: Analysis of categorical variables.

	loan	default	housing	n
1	unknown	no	unknown	637
2	unknown	unknown	unknown	214

Table 6: Analysis of unknown values in load, default, housing.

At the Figure 3 we've plotted all categorical variables in our dataset. At the first subplot we have *job* variable. Most of the jobs are connected to administration - *admin* and *blue-collar*, which covers most of the all jobs. The least frequent category is *unknown*. On the next plot we can see that most of the clients are *married* and only about a few of them we don't know what is client marital status. Moreover majority of the clients have accomplished *high school* or *university degree*. What's interesting we have a some observations with illiteracy. Most of the people have never *default* or we don't know about that. Only 2 examples had ever *default* and it would be probably good to exclude that variable because it cannot properly say anything about client - because answer is *no* or *we don't know*. Approximately half of the people have own house and second half does not. About a small portion of clients we don't know. Moreover most of the clients doesn't have a *loan*. What's interesting and we can observe in Table 6 that it's a case that we don't known more than one information about client's *loan*, *default* and *housing*. It may be caused by data collection process. *Contact* was mostly performed by *cellular*. By looking at the *month* of the call we can guess that most of the data was collected between *April* and *August* especially the most intensive month was *May*. *Day of the week* was uniformly distributed and calls was only performed during work days. By intuition month and day of the week will probably won't have any particular predictive power because it's random situation when we will call particular client. However it may be a case when we take into account socio-economic variables. Similar argumentation may be applied to the *contact* variable.

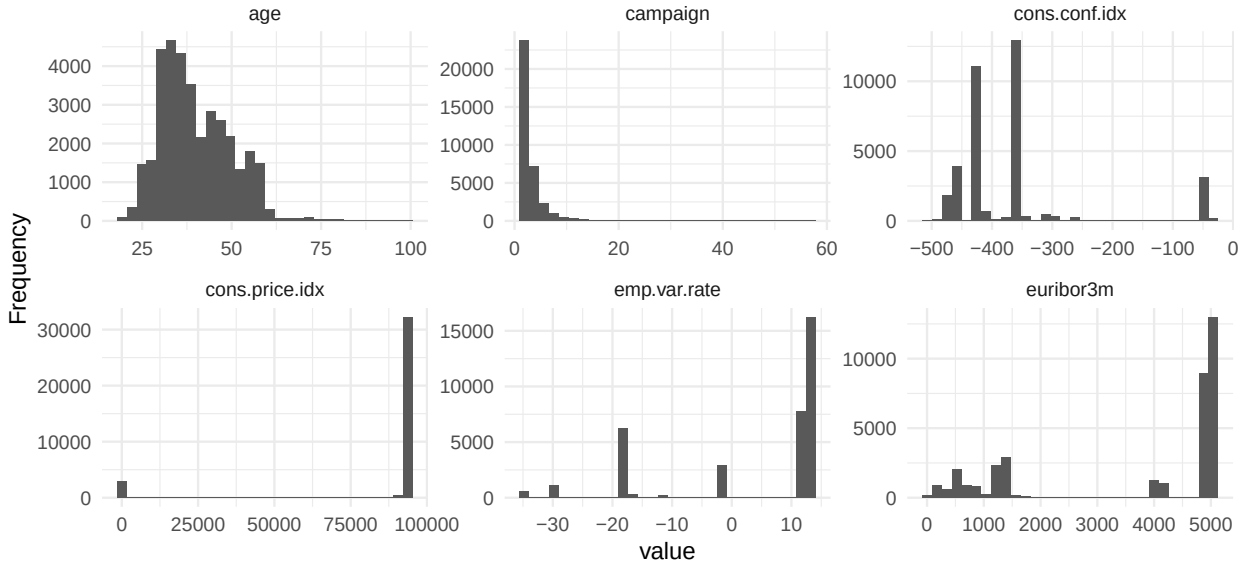

Figure 4: Analysis of continuous variables.

Now, let's take a closer look at continuous variables. We've plotted histograms on Figure 4. We can observe that most of the clients are young adults which is understandable because they may have in this age some saving which may be worth to invest. *Campaign* variable looks very similar to the exponential distribution which is quite natural for a such variable. *Consumer confidence index* provides an indication of future developments of households' consumption and saving, based upon answers regarding their expected financial situation, their sentiment about the general economic situation, unemployment and capability of savings. Negative values for this variable are not a good indicator for our goal to encourage people to save money

in bank term deposits. *Consumer price index* is a measure that examines the weighted average of prices of a basket of consumer goods and services, such as transportation, food, and medical care compared to the base year. High positive values are a bad indicator for us because people probably will have less money to invest and could be less willing to subscribe a bank term deposit. *Employment variation rate* is essentially the variation of how many people are being hired or fired due to the shifts in the conditions of the economy. When the economy is in a recession or depression, people should be more conservative with their money and how the spend it because their financial future is less clear due to cyclical unemployment. When the economy is at its peak, individuals can be more open to risky investments because their employment options are greater. Both positive and negative values may potentialy mean good predictive power of this variable. The last variable - *3 month Euribor interest rate* is the interest rate at which a selection of European banks lend one another funds denominated in euros whereby the loans have a maturity of 3 months. We may suspect that very high values such as 5000 may indicate wrong scale of the variable which should be around 1-5% in this case. High values of Euribor may be also a good indicator for subscribing a term deposit.

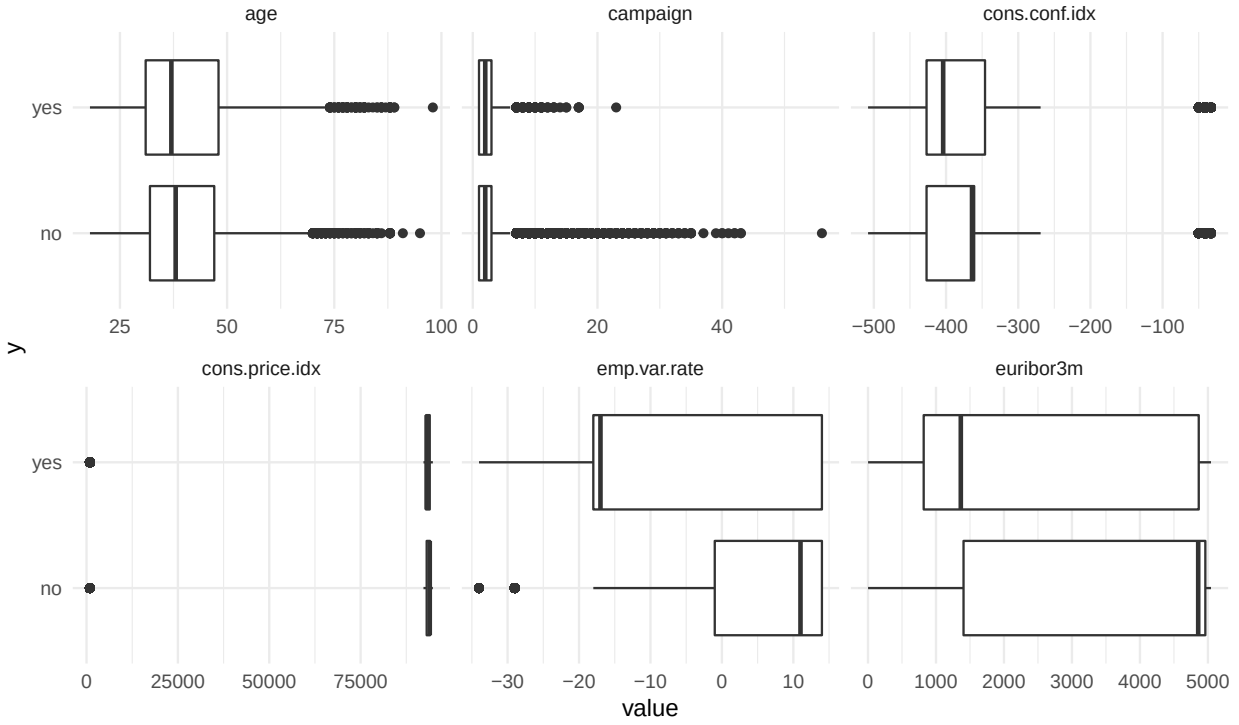

Figure 5: Analysis of continuous variables in context of target variable.

After univariate analysis there is a time for multivariate analysis. We will analyse continuous variables in context of target variable. The results can be found on Figure 5. In the *age* variable it's hard to distinguish between target category what values of age can determine subscription. More interesting is next chart with *campaign* variable where we can observe that calling someone more than 20 times won't end up with success. The next variables connected to socio-economic factors have very different distributions except *Consumer price index* and may be a very good features for a predictive model.

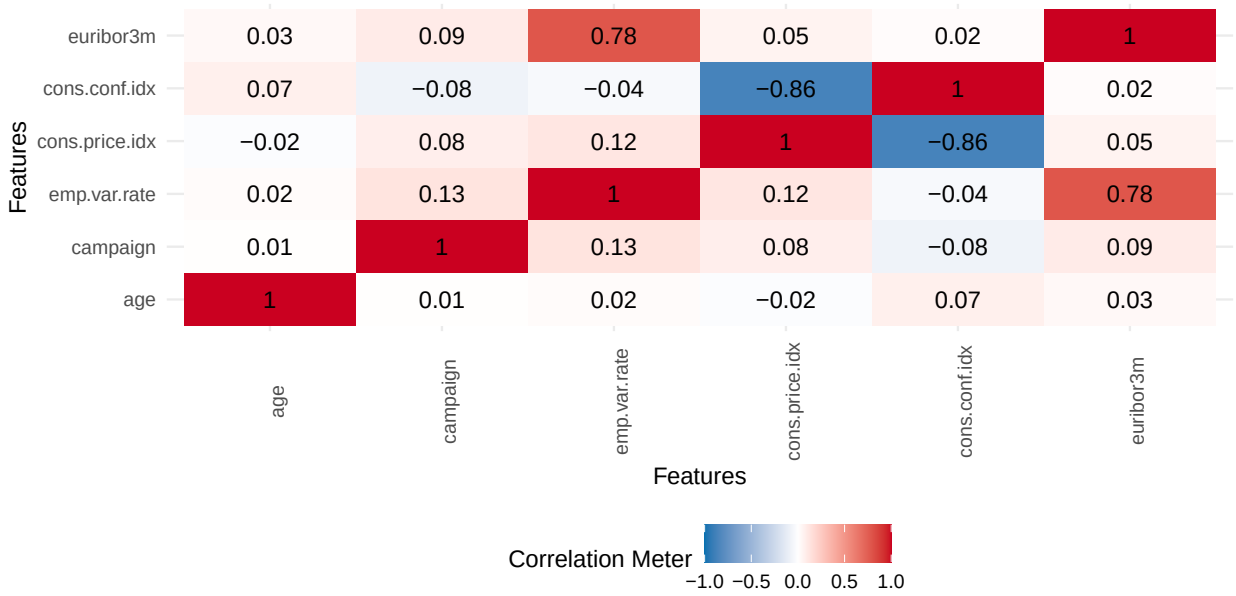

Figure 6: Analysis of correlations.

The next step in our data understanding process will be correlation analysis. As we can observe on Figure 6 there are two set of variables which are highly correlated - *emp.var.rate* with *euribor3m* and *cons.conf.idx* with *cons.price.idx*, which is quite understandable because these are socio-economic indices which often are based on similar variables. We will have that in mind and if necessary exclude one of the variables from pairs.

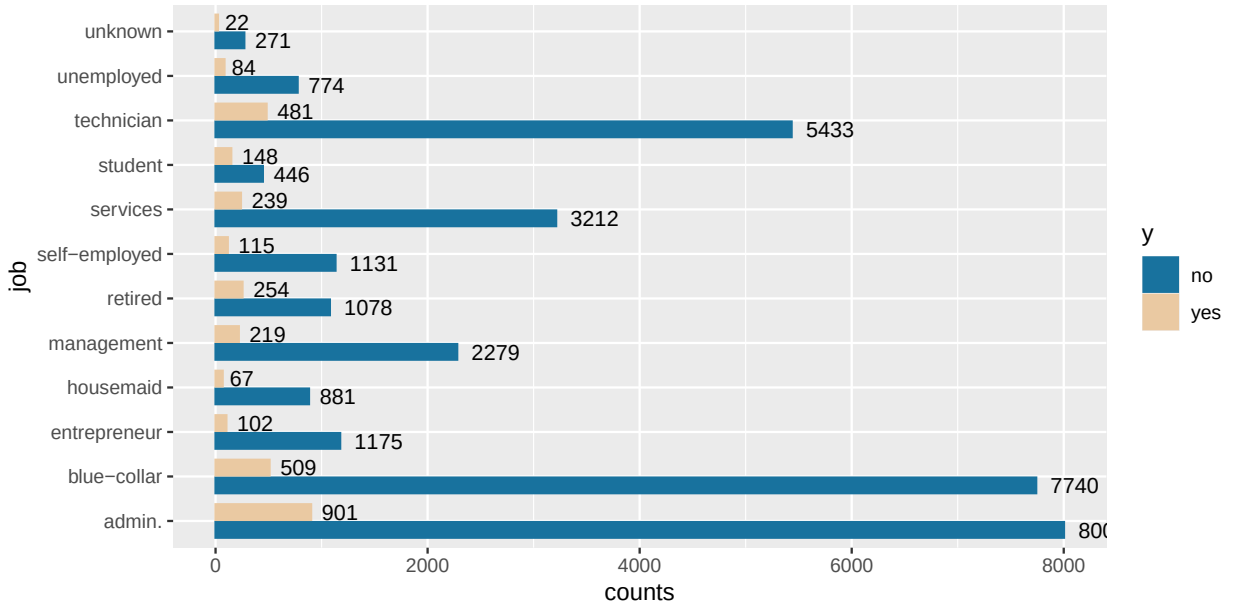

Figure 7: Analysis of *job* variable in the context of target variable.

We also conducted multiavariate analysis for categorical variables. Let's begin with *job* variable on the Figure 7. We can see that proportions between groups are different, especially interesting are *students* which have quite big percent of subscribed bank term deposits. It could be very useful variable for our model.

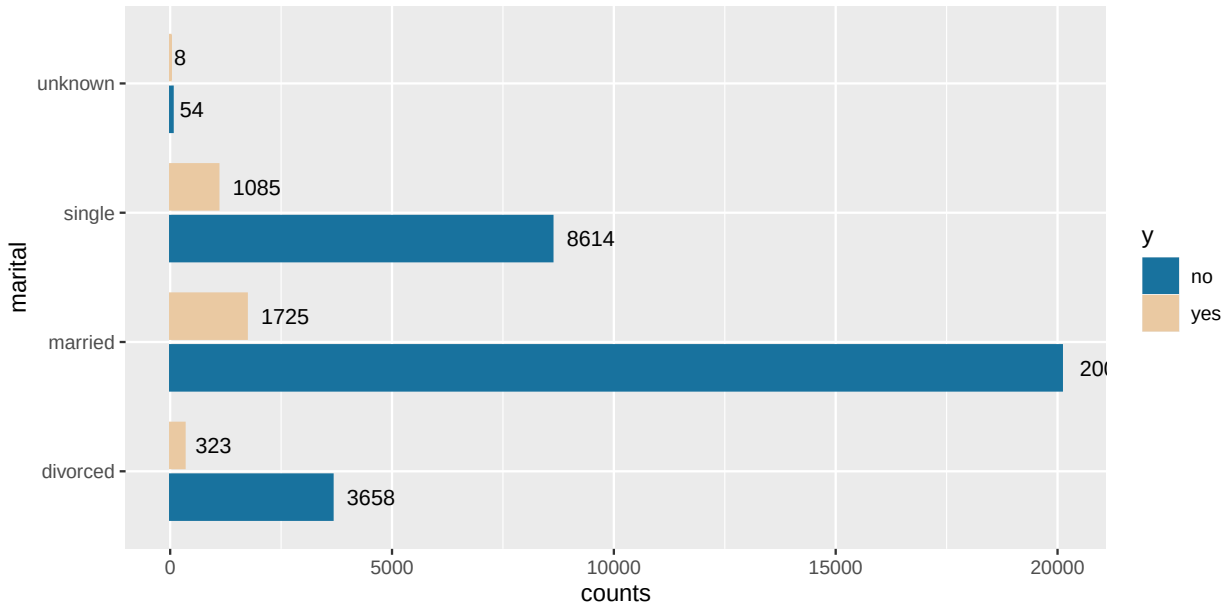

Figure 8: Analysis of *marital* variable in the context of target variable.

Marital status has 4 different categories as we previously see in the analysis. As we can see on Figure 8 they also have similar proportions between classes but we can see that *unknown* category has low number of observations which could be potentialy improved during data collection process. We would like to remove these observations that have unknown marital status.

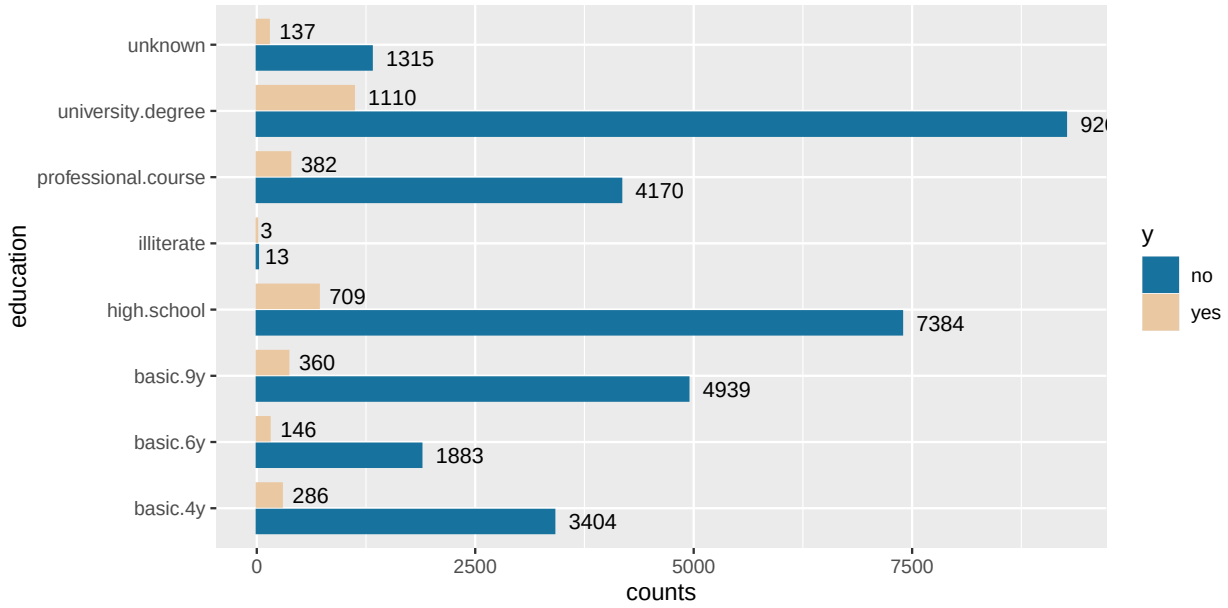

Figure 9: Analysis of *education* variable in the context of target variable.

The next plot is about *education* variable on Figure 9. There is a lot of categories which will need to be encoded and we see that there is not a lot of differences between particular groups. We would like to try group *basic* category into one and *illiterate* with *unknown* as they are semantically similar.

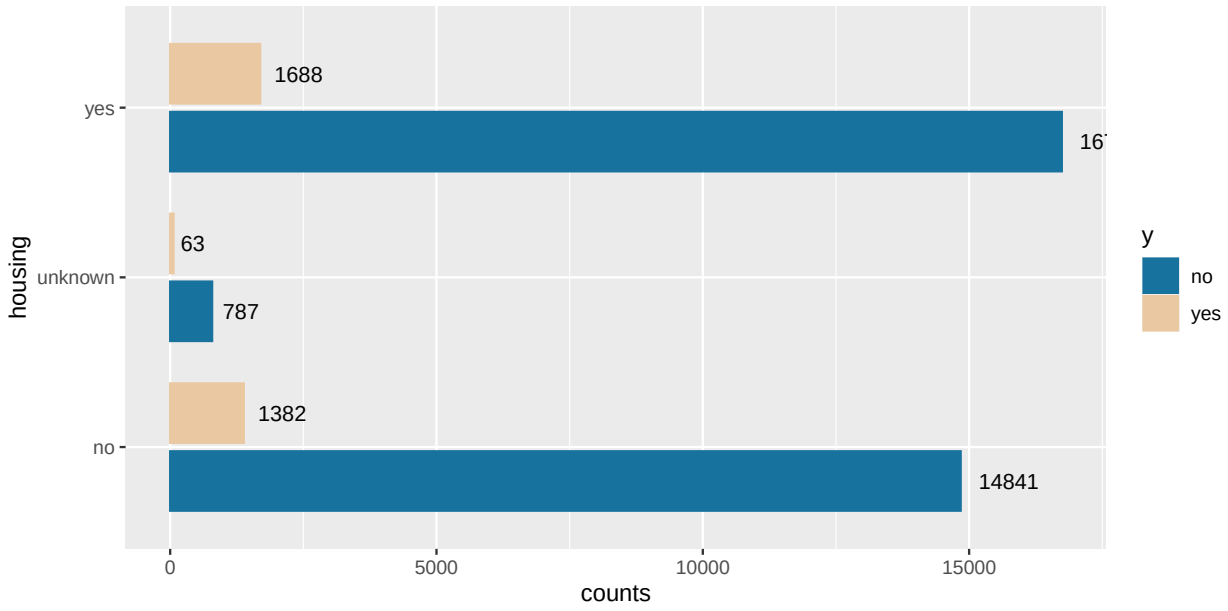

Figure 10: Analysis of *housing* variable in the context of target variable.

Housing is also variable which has a status *unknown* and only 2 other categories. The results of grouping by target category can be found on Figure 10. Similarily to marital, this status could be improved during data collection. Except that we can observe that housing have similar proportions. Moreover after reduction of unknown category we could encode housing as a binary variable.

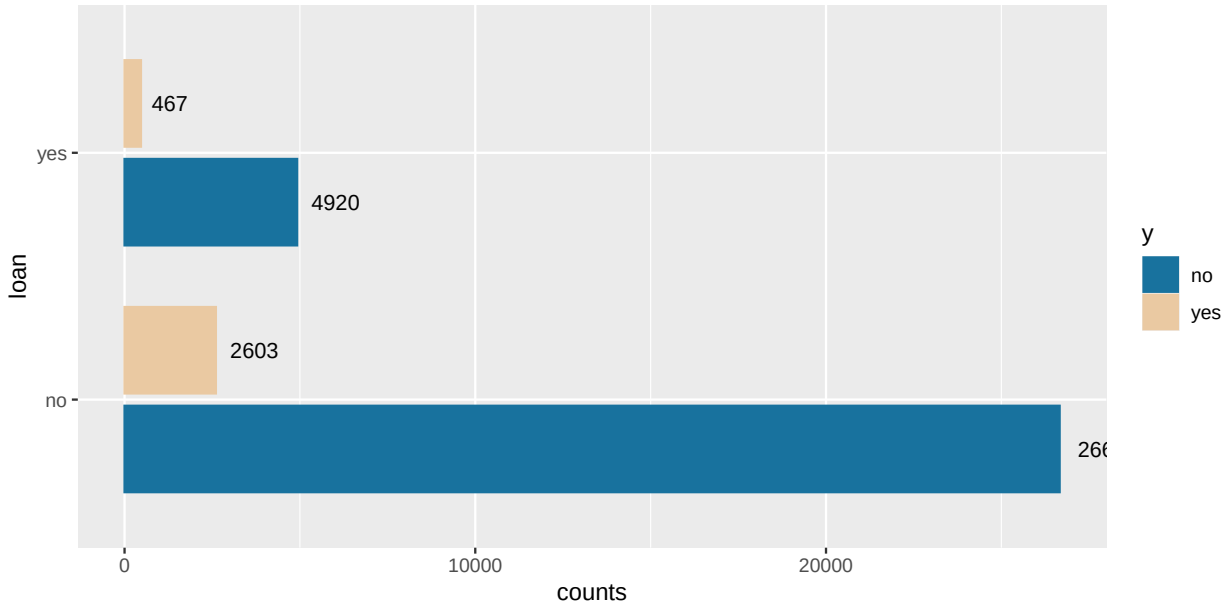

Figure 11: Analysis of *loan* variable in the context of target variable.

On the next chart on Figure 11 we can observe *loan* variable. After reduction of previous variables we can have only two states: *yes* or *no* so we can encode that variable as '0/1' flag. It's hard to say at this moment if this variable will have large impact on model but we can left it for now.

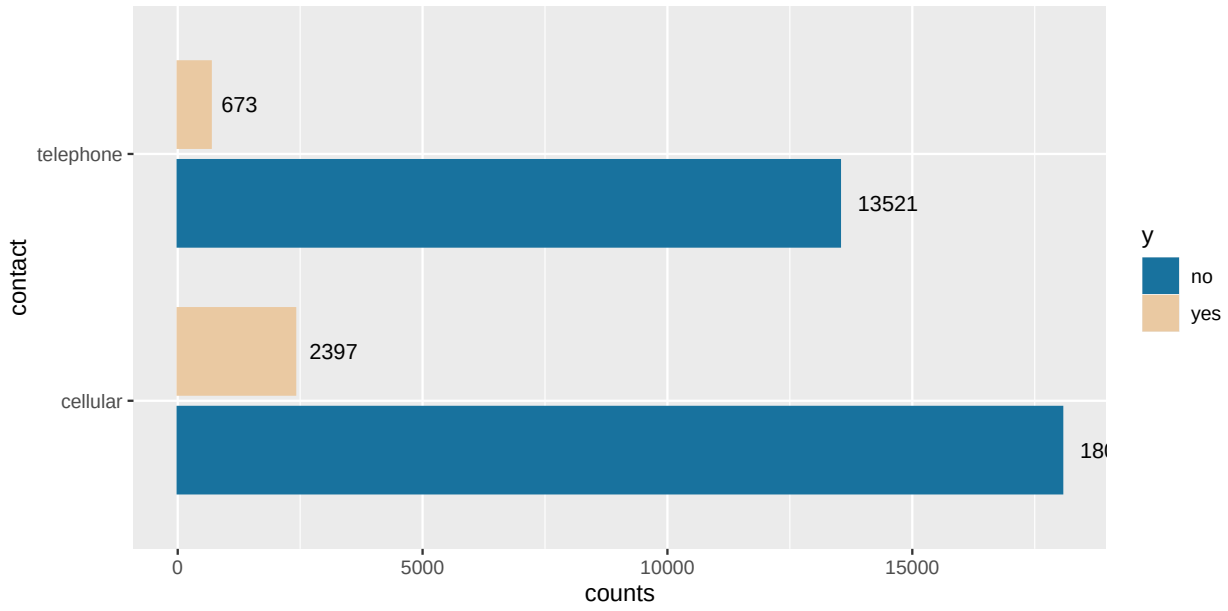

Figure 12: Analysis of *contact* variable in the context of target variable.

Interesting plot is on Figure 12 which describes *contact* variable. We can observe that contact made by *cellular* is more likely to be successful. Because we have only two categories we can encode that variable as '0/1' flag.

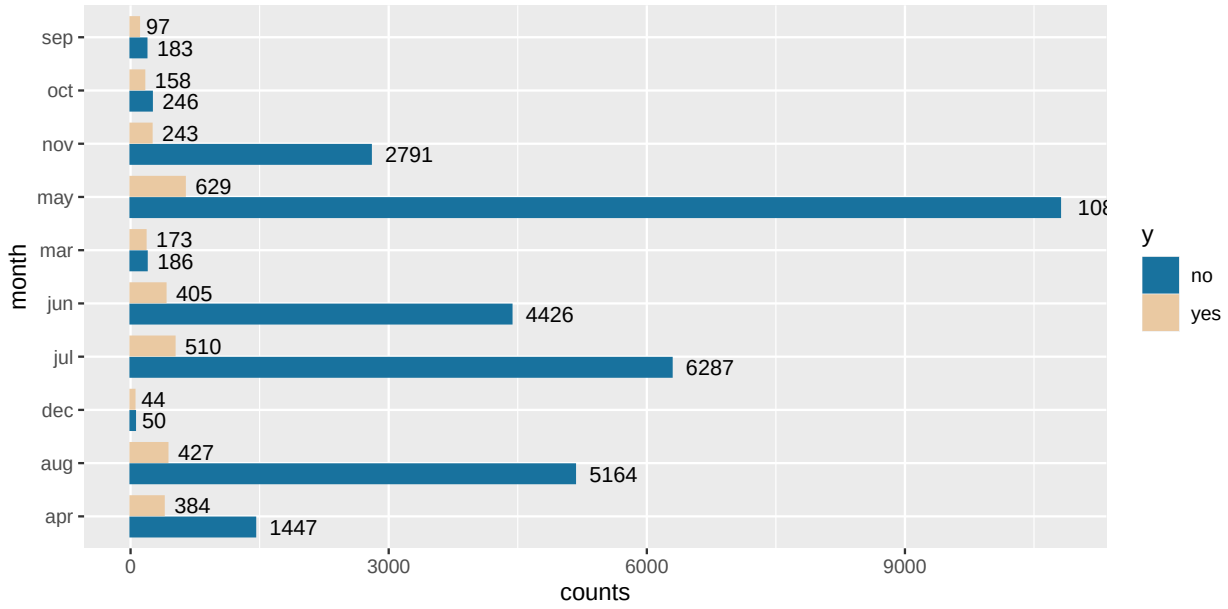

Figure 13: Analysis of *month* variable in the context of target variable.

Very interesting variable is *month*. On Figure 13 we can observe the results. It was very likely to make customer to get deposit on months: *March*, *September*, *October*, *December* comparing to other months. It's worth to think a while about that feature from this disproportions come from. It may be a case that in May, June, July, August we have the largest number of calls and bank only wanted to collect as much data as

possible and for example in following months there was used a predictive model which targeted clients in very accurate way. There is also other possibility. The last examples was targeted after a few other calls in this campaign and there was only negotiations between telemarketer and client which in very large number of cases ended up with success. However this variable shouldn't be used to predictive model because we would probably want to use it during whole year and we've got samples from whole period of that time.

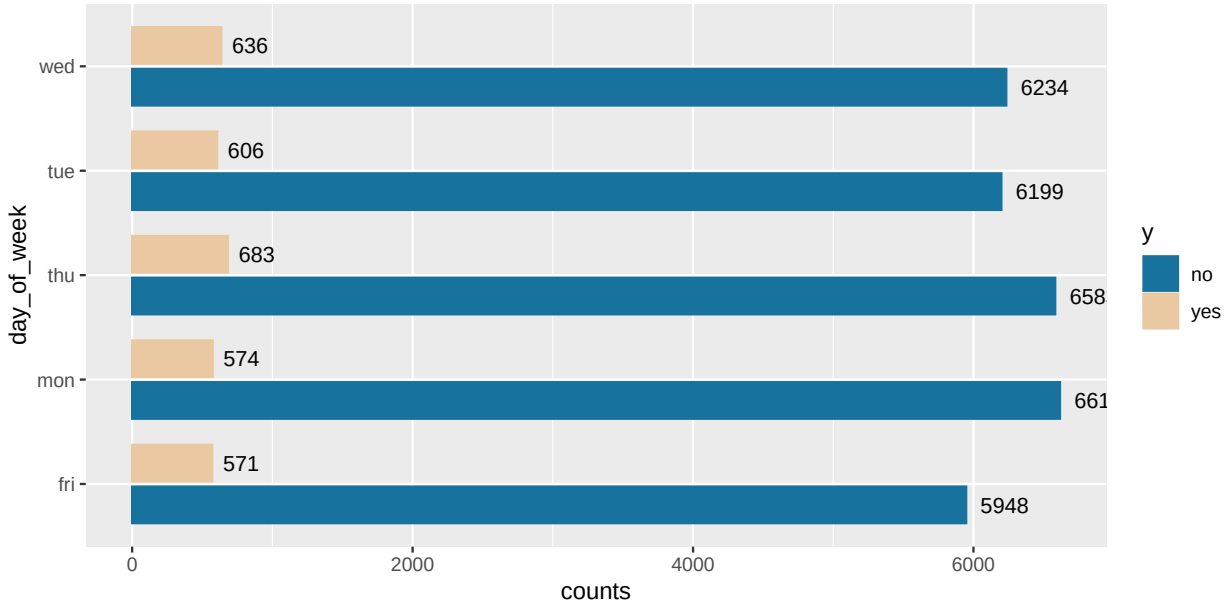

Figure 14: Analysis of *month* variable in the context of target variable.

Similarily to the month we will analyse *day of the week*. Coresponding plot can be found on Figure 14. As we previously expected there is no differences between particular days and we can skip this variable in modeling part.

After all this analysis our resulting dataset have 34651 rows of observations and 13 columns: age, job, marital, education, housing, loan, campaign, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, y, contact_cellular. Some of the variables will require additional encoding but it will be our task in the next subsection connected to modeling part.

Classification

In this part we will focus on classification task. Our goal is to predict for new clients in current telemarketer campaign who will subscribe bank term deposit. As we earlier spotted there exists class imbalance and it's equal to 0.0972103. Due to this class disproportion our main classification metric will be area under precision recall curve (AUCPR). The reason is that in general to measure classification power we use AUC as it does not depend on classification threshold however it's known that it doesn't do well on imbalanced dataset and then it's good to use AUCPR. The short intuition behind that is that it combines precision and recall which both depends on minority class *1* contrary to AUC which depends in their definition also on *0* class which is majority class and overstates the results. Obviously we will also report other metrics such as accuracy, precision, recall and AUC. As a resampling method we will use 5-fold stratified cross-validation. We use stratification because we want to have real proportions of class in each fold and we use cross-validation to have better estimation of real error and our dataset is too small to use only holdout set. In our project we will test algorithms such as kNN, LDA, QDA, Logistic Regression, Decision Tree, Random Forest. We will also use different subset of variables depending of algorithm and check performance.

k-Nearest Neighbours

We will start our journey with probably one of the simples machine learning algorithm - k-Nearest Neighbours. We analyse two combinations of variables: only numeric variables, all variables with categorical variables one-hot-encoded. Also as the kNN is based on distance we need to scale our socio-economic variables which range is much more different than other variables which are quite close to range (0,1). However we also wanted to do some experiment and check how does scaling affects the results so to sum up we have 4 different datasets which we will be checking. Moreover we will check k = 21.

```
## INFO  [23:59:50.591] Benchmark with 20 resampling iterations
## INFO  [23:59:50.695] Applying learner 'knn 21' on task 'all-variables-scaled' (iter 5/5)
## INFO  [00:00:09.156] Applying learner 'knn 21' on task 'all-variables' (iter 5/5)
## INFO  [00:00:28.047] Applying learner 'knn 21' on task 'all-variables-scaled' (iter 3/5)
## INFO  [00:00:48.256] Applying learner 'knn 21' on task 'only-numeric-scaled' (iter 3/5)
## INFO  [00:00:57.685] Applying learner 'knn 21' on task 'only-numeric-scaled' (iter 2/5)
## INFO  [00:01:07.847] Applying learner 'knn 21' on task 'all-variables-scaled' (iter 4/5)
## INFO  [00:01:28.648] Applying learner 'knn 21' on task 'only-numeric' (iter 1/5)
## INFO  [00:01:38.136] Applying learner 'knn 21' on task 'all-variables-scaled' (iter 2/5)
## INFO  [00:01:58.059] Applying learner 'knn 21' on task 'only-numeric-scaled' (iter 5/5)
## INFO  [00:02:07.611] Applying learner 'knn 21' on task 'only-numeric-scaled' (iter 4/5)
## INFO  [00:02:17.731] Applying learner 'knn 21' on task 'all-variables' (iter 4/5)
## INFO  [00:02:37.202] Applying learner 'knn 21' on task 'all-variables' (iter 3/5)
## INFO  [00:02:57.489] Applying learner 'knn 21' on task 'only-numeric' (iter 5/5)
## INFO  [00:03:06.729] Applying learner 'knn 21' on task 'only-numeric' (iter 3/5)
## INFO  [00:03:14.543] Applying learner 'knn 21' on task 'all-variables' (iter 1/5)
## INFO  [00:03:31.537] Applying learner 'knn 21' on task 'only-numeric-scaled' (iter 1/5)
## INFO  [00:03:41.325] Applying learner 'knn 21' on task 'all-variables-scaled' (iter 1/5)
## INFO  [00:03:59.567] Applying learner 'knn 21' on task 'only-numeric' (iter 4/5)
## INFO  [00:04:08.897] Applying learner 'knn 21' on task 'all-variables' (iter 2/5)
## INFO  [00:04:27.189] Applying learner 'knn 21' on task 'only-numeric' (iter 2/5)
## INFO  [00:04:37.967] Finished benchmark
```

Table 7: Results for LDA algorithm.

	V1	V2	V3	V4
nr	1	2	3	4
resample_result				
task_id	only-numeric	all-variables	only-numeric-scaled	all-variables-scaled
learner_id	knn 21	knn 21	knn 21	knn 21
resampling_id	cv	cv	cv	cv
iters	5	5	5	5
accuracy_train	0.921921449000198	0.920622785961518	0.922145102511612	0.920593925672386
accuracy_test	0.908978142202785	0.908920384669916	0.910392146208046	0.908949232207059
precision_train	0.925265871828541	0.922029405673213	0.925471571886902	0.922095242578969
precision_test	0.918697829346966	0.916069270046811	0.919358821674297	0.915658657201509
recall_train	0.994672423545378	0.997237260967876	0.994680349348207	0.997118523699451
recall_test	0.987524180305804	0.990848944157141	0.988379016462208	0.991418870017403
auc_train	0.927186535443713	0.957546720484326	0.92710307208756	0.957565092437318
auc_test	0.700907442874557	0.68543734108023	0.700427464466651	0.682915864012737
aucpr_train	0.992465783951449	0.995894873821043	0.992435567333326	0.995896543059799
aucpr_test	0.948438823812664	0.947035046678208	0.948185427829552	0.946065165898376

The results can be found in Table **??**. Our first notice about kNN is that it took a long time to train all

clasifiers. It's understandable because it makes comparison all-to-all samples and when we have more than 30 thousands samples it takes a lot of time. Moreover we can observe on AUC and also other metrics that our models are very overfitted, scaling does not affect our results and higher number of neighbours gives a better results but still not ideal.

Linear Discriminant Analysis, Quadratic Discriminant Analysis

The next algorithm which we will use is a LDA and QDA. As it was mentioned during lab classes these algorithms are very stable and often forgives broken assumptions. Similarily to the previous case we will use four options of data: all numerica and all variables with categorical features one-hot encoded with scaled and normal version.

```
## INFO  [00:04:39.114] Benchmark with 20 resampling iterations
## INFO  [00:04:39.120] Applying learner 'lda' on task 'only-numeric-scaled' (iter 5/5)
## INFO  [00:04:39.229] Applying learner 'lda' on task 'all-variables-scaled' (iter 2/5)
## INFO  [00:04:39.582] Applying learner 'lda' on task 'all-variables' (iter 1/5)
## INFO  [00:04:39.813] Applying learner 'lda' on task 'only-numeric' (iter 5/5)
## INFO  [00:04:39.907] Applying learner 'lda' on task 'all-variables' (iter 5/5)
## INFO  [00:04:40.134] Applying learner 'lda' on task 'all-variables-scaled' (iter 4/5)
## INFO  [00:04:40.369] Applying learner 'lda' on task 'only-numeric' (iter 1/5)
## INFO  [00:04:40.461] Applying learner 'lda' on task 'all-variables-scaled' (iter 5/5)
## INFO  [00:04:40.694] Applying learner 'lda' on task 'only-numeric' (iter 2/5)
## INFO  [00:04:40.791] Applying learner 'lda' on task 'all-variables' (iter 3/5)
## INFO  [00:04:41.026] Applying learner 'lda' on task 'all-variables-scaled' (iter 1/5)
## INFO  [00:04:41.266] Applying learner 'lda' on task 'all-variables-scaled' (iter 3/5)
## INFO  [00:04:41.497] Applying learner 'lda' on task 'only-numeric-scaled' (iter 2/5)
## INFO  [00:04:41.594] Applying learner 'lda' on task 'all-variables' (iter 4/5)
## INFO  [00:04:41.949] Applying learner 'lda' on task 'only-numeric' (iter 3/5)
## INFO  [00:04:42.041] Applying learner 'lda' on task 'all-variables' (iter 2/5)
## INFO  [00:04:42.277] Applying learner 'lda' on task 'only-numeric-scaled' (iter 3/5)
## INFO  [00:04:42.373] Applying learner 'lda' on task 'only-numeric' (iter 4/5)
## INFO  [00:04:42.475] Applying learner 'lda' on task 'only-numeric-scaled' (iter 4/5)
## INFO  [00:04:42.569] Applying learner 'lda' on task 'only-numeric-scaled' (iter 1/5)
## INFO  [00:04:42.791] Finished benchmark
```

Table 8: Results for LDA algorithm.

	V1	V2	V3	V4
nr	1	2	3	4
resample_result				
task_id	only-numeric	all-variables	only-numeric-scaled	all-variables-scaled
learner_id	lda	lda	lda	lda
resampling_id	cv	cv	cv	cv
iters	5	5	5	5
accuracy_train	0.90074600821742	0.901128392408108	0.900738800237556	0.901049036178
accuracy_test	0.900724323849414	0.900435744380341	0.900753304631533	0.900839880554207
precision_train	0.92425295949112	0.924687119187312	0.924239456341276	0.924642024279518
precision_test	0.924259701128489	0.924235790653414	0.924233372597335	0.924523019115777
recall_train	0.970646901238261	0.970567745043765	0.970654827354426	0.970528173369879
recall_test	0.970615124923518	0.970298594131079	0.970678631592445	0.970425331767079
auc_train	0.708273237777248	0.719355756951744	0.708267873438201	0.719399044371429
auc_test	0.706984811397241	0.716196715668001	0.707288570850319	0.7162920730406
aucpr_train	0.950697452313912	0.952278570254046	0.950759389730358	0.952322043239382

	V1	V2	V3	V4
aucpr_test	0.95048326533876	0.951538607604478	0.950464622095488	0.95142529191007

Unfortunately for QDA algorithm we have received error about rank deficiency in group *no*. Due to that fact we've only checked LDA algorithm. As we can observe in Table **??** the results are very stable because train and test scores are very similar. Also we can observe that results for Moreover the final result of AUCPR is very high and is approximately equal to 0.95. Such a great result may be more than enough for our problem in real-world scenario.

Logistic Regression

We have also tried simple Logistic Regression without any regularization. The setup will be standard one, i.e. without scaling in two versions - with all numeric variables and with one-hot encoded marital status, education and job.

```
## INFO  [00:04:43.989] Benchmark with 10 resampling iterations
## INFO  [00:04:43.994] Applying learner 'log-reg' on task 'only-numeric' (iter 2/5)
## INFO  [00:04:44.105] Applying learner 'log-reg' on task 'only-numeric' (iter 1/5)
## INFO  [00:04:44.337] Applying learner 'log-reg' on task 'only-numeric' (iter 4/5)
## INFO  [00:04:44.430] Applying learner 'log-reg' on task 'all-variables' (iter 4/5)
## INFO  [00:04:44.699] Applying learner 'log-reg' on task 'only-numeric' (iter 5/5)
## INFO  [00:04:44.799] Applying learner 'log-reg' on task 'only-numeric' (iter 3/5)
## INFO  [00:04:44.897] Applying learner 'log-reg' on task 'all-variables' (iter 5/5)
## INFO  [00:04:45.164] Applying learner 'log-reg' on task 'all-variables' (iter 1/5)
## INFO  [00:04:45.438] Applying learner 'log-reg' on task 'all-variables' (iter 3/5)
## INFO  [00:04:45.710] Applying learner 'log-reg' on task 'all-variables' (iter 2/5)
## INFO  [00:04:45.977] Finished benchmark
```

Table 9: Results for LDA algorithm.

	V1	V2
nr	1	2
resample_result		
task_id	only-numeric	all-variables
learner_id	log-reg	log-reg
resampling_id	cv	cv
iters	5	5
accuracy_train	0.911654785498442	0.911431127302128
accuracy_test	0.91148885228816	0.91102710015421
precision_train	0.913399799629389	0.913691144032681
precision_test	0.913391790604159	0.913421980392135
recall_train	0.997656808068796	0.996999780791
recall_test	0.997466881440082	0.996865189713454
auc_train	0.723696978202154	0.729463305186583
auc_test	0.721972821676311	0.72470790344155
aucpr_train	0.954272828086809	0.95492994779743
aucpr_test	0.953782266180975	0.953405222444155

The results of our Logistic Regression can be found in the Table **??**. As we can observe on all metrics this algorithm is very well fitted and has high score for all metrics. What's interesting adding categorical variables does not really improved results of the model.

	sort.m.coefficients.
(Intercept)	-2.79
education_basic	-0.14
marital_divorced	-0.14
marital_married	-0.14
education_professional.course	-0.08
education_high.school	-0.07
campaign	-0.05
housing	-0.04
emp.var.rate	-0.03
loan	-0.03
age	-0.00
euribor3m	-0.00
cons.price.idx	0.00
cons.conf.idx	0.00
job_blue.collar	0.02
education_university.degree	0.03
job_services	0.08
job_technician	0.12
job_housemaid	0.15
job_management	0.15
job_self.employed	0.24
job_entrepreneur	0.24
job_admin.	0.25
job_unemployed	0.26
contact_cellular	0.54
job_student	0.57
job_retired	0.74

Table 10: Impurity feature importance for decision tree on all data.

We wanted to also check coefficients of particular variables so we've trained one more time Logistic Regression on whole dataset with all variables and presented results in table 10. The most driving factor for subscribing a loan is a student status or being a retired and contacting via cellular phone. The opposite ones was Euribor, basic education and marital status (divorced / maried). Also we can observe that the intercept is very low. That means if we want to find people who will subscribe we need to look for a combination of a few very influential variables such as for example mentioned job status (student / retired).

Tree models

Our next models will be decision tree and its variation - random forest. We will use package *ranger* which deals with categorical variables so we will only test one dataset which consists of all variables. Let's check how them perform.

```
## INFO  [00:04:47.145] Benchmark with 15 resampling iterations
## INFO  [00:04:47.150] Applying learner 'random-forest-100' on task 'all' (iter 3/5)
## INFO  [00:04:47.793] Applying learner 'random-forest-100' on task 'all' (iter 2/5)
## INFO  [00:04:48.547] Applying learner 'decision-tree' on task 'all' (iter 4/5)
## INFO  [00:04:48.654] Applying learner 'random-forest-100' on task 'all' (iter 5/5)
## INFO  [00:04:49.262] Applying learner 'random-forest-100' on task 'all' (iter 1/5)
## INFO  [00:04:49.884] Applying learner 'random-forest-50' on task 'all' (iter 2/5)
## INFO  [00:04:50.253] Applying learner 'decision-tree' on task 'all' (iter 2/5)
## INFO  [00:04:50.365] Applying learner 'random-forest-50' on task 'all' (iter 3/5)
```

```
## INFO  [00:04:50.719] Applying learner 'random-forest-50' on task 'all' (iter 5/5)
## INFO  [00:04:51.083] Applying learner 'decision-tree' on task 'all' (iter 3/5)
## INFO  [00:04:51.188] Applying learner 'random-forest-50' on task 'all' (iter 1/5)
## INFO  [00:04:51.567] Applying learner 'random-forest-50' on task 'all' (iter 4/5)
## INFO  [00:04:51.932] Applying learner 'decision-tree' on task 'all' (iter 5/5)
## INFO  [00:04:52.040] Applying learner 'decision-tree' on task 'all' (iter 1/5)
## INFO  [00:04:52.162] Applying learner 'random-forest-100' on task 'all' (iter 4/5)
## INFO  [00:04:52.837] Finished benchmark
```

Table 11: Results for tree algorithms.

	V1	V2	V3
nr	1	2	3
resample_result			
task_id	all	all	all
learner_id	decision-tree	random-forest-50	random-forest-100
resampling_id	cv	cv	cv
iters	5	5	5
accuracy_train	0.917404990085449	0.940658278768556	0.940954088598654
accuracy_test	0.889209484627173	0.910449849610144	0.910623051422359
precision_train	0.939579984270286	0.940428252548666	0.940492684373804
precision_test	0.925284712897267	0.921199449050848	0.921288100344858
recall_train	0.971873926123434	0.998115955357066	0.99838510584511
recall_test	0.955606141554555	0.986099187702133	0.986194219624899
auc_train	0.823630365535946	0.971301600743046	0.97213759639821
auc_test	0.624580561212692	0.738638852682975	0.740261864979842
aucpr_train	0.970481983244569	0.997186479684089	0.997279313318062
aucpr_test	0.931937733427997	0.955708633419468	0.956168741208429

The results of the tested trees can be found in Table **??**. We can observe that difference between training and test metrics are quite high for all models which means that models have overfitted. It can be especially easily seen of AUC metric. It would be probably a good idea to prune these trees and check how this would affect results.

	sort.m.variable.importance..decreasing...TRUE.
age	662.55
euribor3m	533.46
job	345.04
campaign	338.86
cons.conf.idx	325.90
cons.price.idx	305.52
education	243.56
marital	153.88
emp.var.rate	149.85
housing	108.22
contact_cellular	97.64
loan	86.50

Table 12: Impurity feature importance for decision tree on all data.

Also as an experiment we wanted to check what variables was important for a model. In Table 12 we've presented results of impurity feature importance of single decision tree. We can observe that the most importance factor was *age*, *euribor3m* and *job* and the least important was *loan*, *contact*, *cons.price.idx*.

Results and discussion

After the whole analysis let's sum up what we've found out about our telemarketing dataset. First of all we've made extensive data analysis during which we precisely defined our modeling goal - predict whether client which have never been targeted in previous marketing campaigns will subscribe bank term deposit. Moreover we've stated one more important constraint - we wanted our model to be as much real-world applicapled as possible. This condition made us to exclude some variables from our modeling process. After that analysis we've performed time and resources consuming process of model training. We've tested several models from which except predictions we've also extracted some knowledge. By analysis of logistic regression and decision tree we can conclude that *age*, *euribor*, *job*, *cons.conf.idx*, *cons.price.idx* are the most important factors for modeling.

	learner_id	task_id	aucpr_train	aucpr_test
1	random-forest-100	all	1.00	0.96
2	random-forest-50	all	1.00	0.96
3	log-reg	only-numeric	0.95	0.95
4	log-reg	all-variables	0.95	0.95
5	lda	all-variables	0.95	0.95
6	lda	all-variables-scaled	0.95	0.95
7	lda	only-numeric	0.95	0.95
8	lda	only-numeric-scaled	0.95	0.95

Table 13: The best 8 models (ordered by AUCPR) from modeling part.

For our production model we would probably use Logistic Regression or LDA even though they are not the best AUCPR models how it's stated in summary Table 13. It's because Random Forest was overfitted and similarily the 7th and 8th kNN. However results of 0.95 for AUCPR is extremly high and for our mind it would satisfy real-world scenarios. But it should be noted that it's only a beginning of our modeling. There is a lot of work which could be done to deliver better results. The first steps we could take may be better hyperparameter tunning especially for random forest models. What's more we could make a stacked ensemble of the best models. Moreover we could try to investigate models better by techinques from eXplainable Artifical Intelligence which could possibly guide us in better model understanding. We could also probably work more with data - find new variables about customer which could be available in banking data system. The last thing which could be also should done before production deployment is a threshold optimization to decide how much clients we want to address and with what precision.

References

[1] P. Cortez S. Moro and P. Rita. A data-driven approach to predict the success of bank telemarketin. *Decision Support Systems*, 2014.