Data Glacier

Your Deep Learning Partner

Slides for technical users

Bank Marketing (Campaign) - Group Project

Group Name - Bloodhounds,
Batch code - LISUM09,
Specialization: Data science.
Group member details:

- Name - Margarita Prokhorovich,
- email - marusya15071240@gmail.com,
- Country – Thailand,
- Submission date – 30 July, 2022

Data understanding

Characteristics

- Data Set Characteristics: Multivariate
- Attribute Characteristics: Real
- Associated Tasks: Classification
- Area: Business
- Date Donated: 2012-02-14

Source

- [Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

Description

- The data is related with 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[4].

Goal

- The binary classification goal is to predict if the client will subscribe a bank term deposit (variable y).

Data set used

- bank-additional-full.csv with all examples, ordered by date (from May 2008 to November 2010).

- This data set is an updated bank-full.csv data set. The data is enriched by the addition of five new social and economic features/attributes (national wide indicators from a ~10M population country), published by the Banco de Portugal.

- It was found that the addition of the five new social and economic attributes (made available here) lead to substantial improvement in the prediction of a success, even when the duration of the call is not included.

4. Bank Marketing Data Set. UCI Machine learning repository, URL

Type of data

Categorical features

bank client data:
- job : type of job
- marital : marital status
- education
- default: has credit in default?
- housing: has housing loan?
- loan: has personal loan?

related with the last contact of the current campaign:
- contact: contact communication type
- month: last contact month of year
- day_of_week: last contact day of the week

other attributes
poutcome: outcome of the previous marketing campaign

Output variable - y - has the client subscribed a term deposit? (binary)

Numeric features

bank client data:
- age

related with the last contact of the current campaign:
- duration: last contact duration, in seconds

other attributes
- campaign: number of contacts performed during this campaign and for this client
- pdays: number of days that passed by after the client was last contacted from a previous campaign
- previous: number of contacts performed before this campaign and for this client
- poutcome: outcome of the previous marketing campaign

social and economic context attributes
- emp.var.rate: employment variation rate - quarterly indicator
- cons.price.idx: consumer price index - monthly indicator
- cons.conf.idx: consumer confidence index - monthly indicator
- euribor3m: euribor 3 month rate - daily indicator
- nr.employed: number of employees - quarterly indicator[5].

5. Bank Marketing Data Set. UCI Machine learning repository, URL

Data cleansing and transformation

These data transformation steps were presented in a previous week report. For these issues in the data I use only one technique because

- drop duplicates is the most convenient way to handle duplicates;
- I don't see any options to work with numeric pdays feature and decide to move to categorical one.

```
n_duplicates = df.duplicated().sum()
print(f"Number of duplicates - {n_duplicates}.")
```

✓ 0.1s

```
Number of duplicates - 12.
```

```
df = df.drop_duplicates()
df.shape
```

✓ 0.1s

```
(41176, 21)
```

Duplicate values

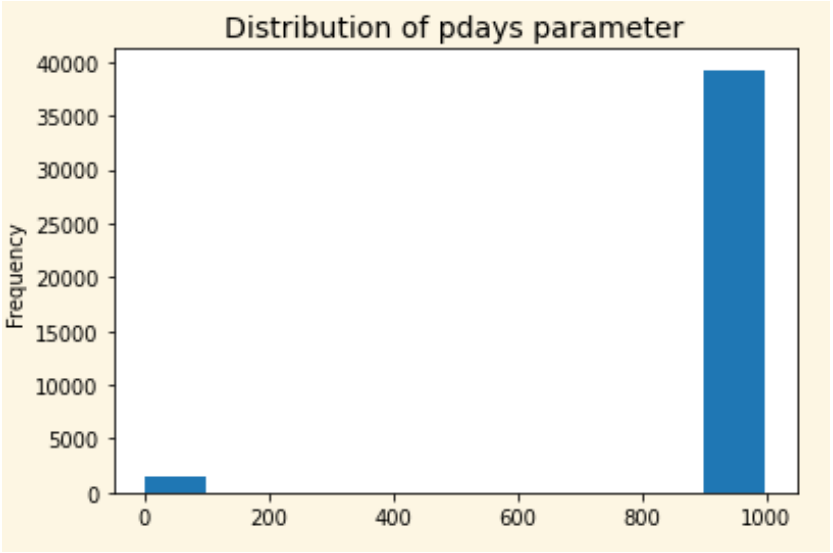
- Since we have duplicates in our data set, we need to delete them.
- We have 12 duplicates and remove them using drop_duplicates() method. This method deletes complete duplicates from the data set.

Pdays parameter values

Pdays parameter has a lot of 999 values. 999 means client was not previously contacted. Since the variable is numeric, it can affect the interpretation of the model. Replacing 999s with 0 is also not effective since interpretation can be wrong - 0 days passed by after the client was last contacted from a previous campaign. Therefore it was decided to move from numeric variable to a binary one. It's also reasonable because except for 999s, another values lie in not large range.

```
df['pdays_categ'] = [0 if pday == 999 else 1 for pday in df.pdays]
df = df.drop(['pdays'], axis = 1)
```

✓ 0.1s



Data cleansing and transformation

‘Unknown’ values

Options for handling

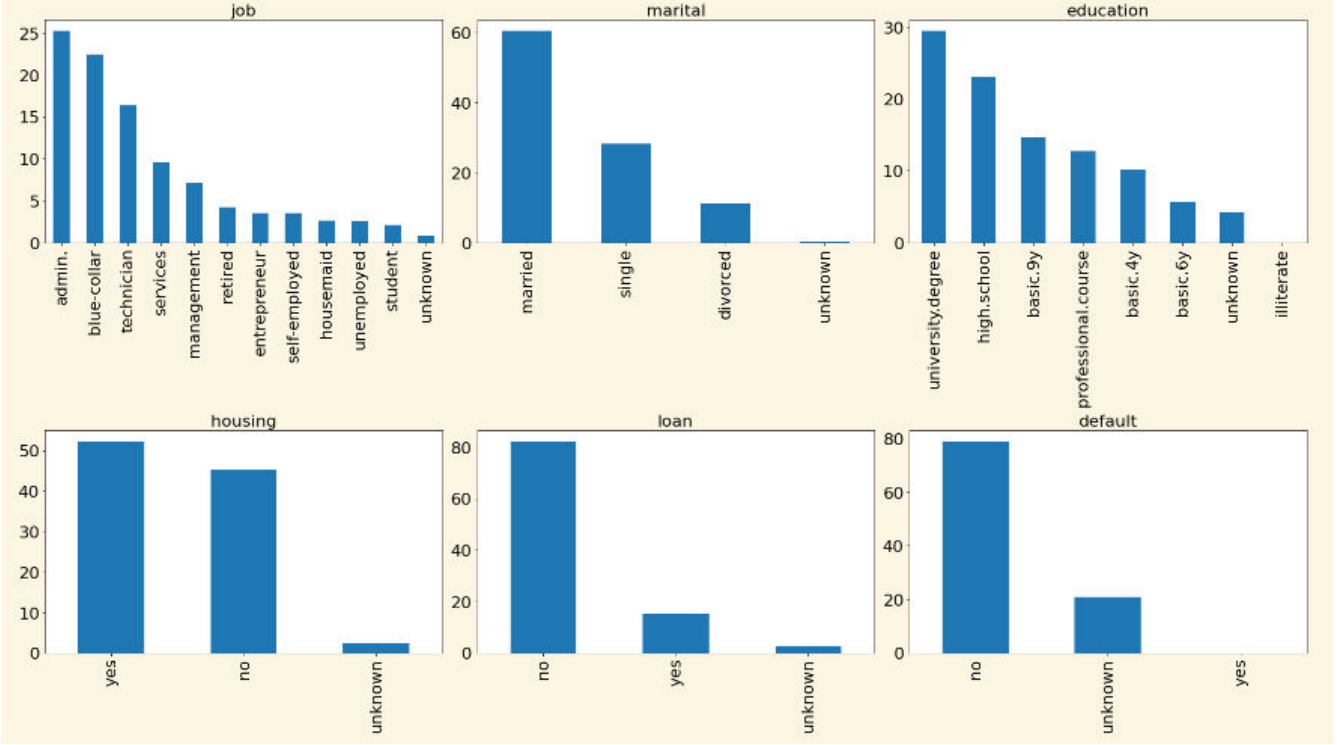
- Data set has no null values but some categorical features have values marked ‘unknown’. Following options can be considered:
- Delete all the rows with ‘unknown’ values or delete them only in certain columns
- Populate ‘unknown’ values with a major category.

The first approach – fill unknown values with values of a major class

The suggested approach is to populate the variables with the most common category. The exception is the default variable, since the proportion of unknowns is quite large, with a high probability we will consider ‘unknown’ as a separate category.

The plots on the right show distribution of categorical features before the processing.

```
Number of "unknown" occurrences:
- feature -  job , number -  330 , percentage - 0.8014 %
- feature -  marital , number -  80 , percentage - 0.1943 %
- feature -  education , number -  1730 , percentage - 4.2015 %
- feature -  default , number -  8596 , percentage - 20.8762 %
- feature -  housing , number -  990 , percentage - 2.4043 %
- feature -  loan , number -  990 , percentage - 2.4043 %
```



Data cleansing and transformation

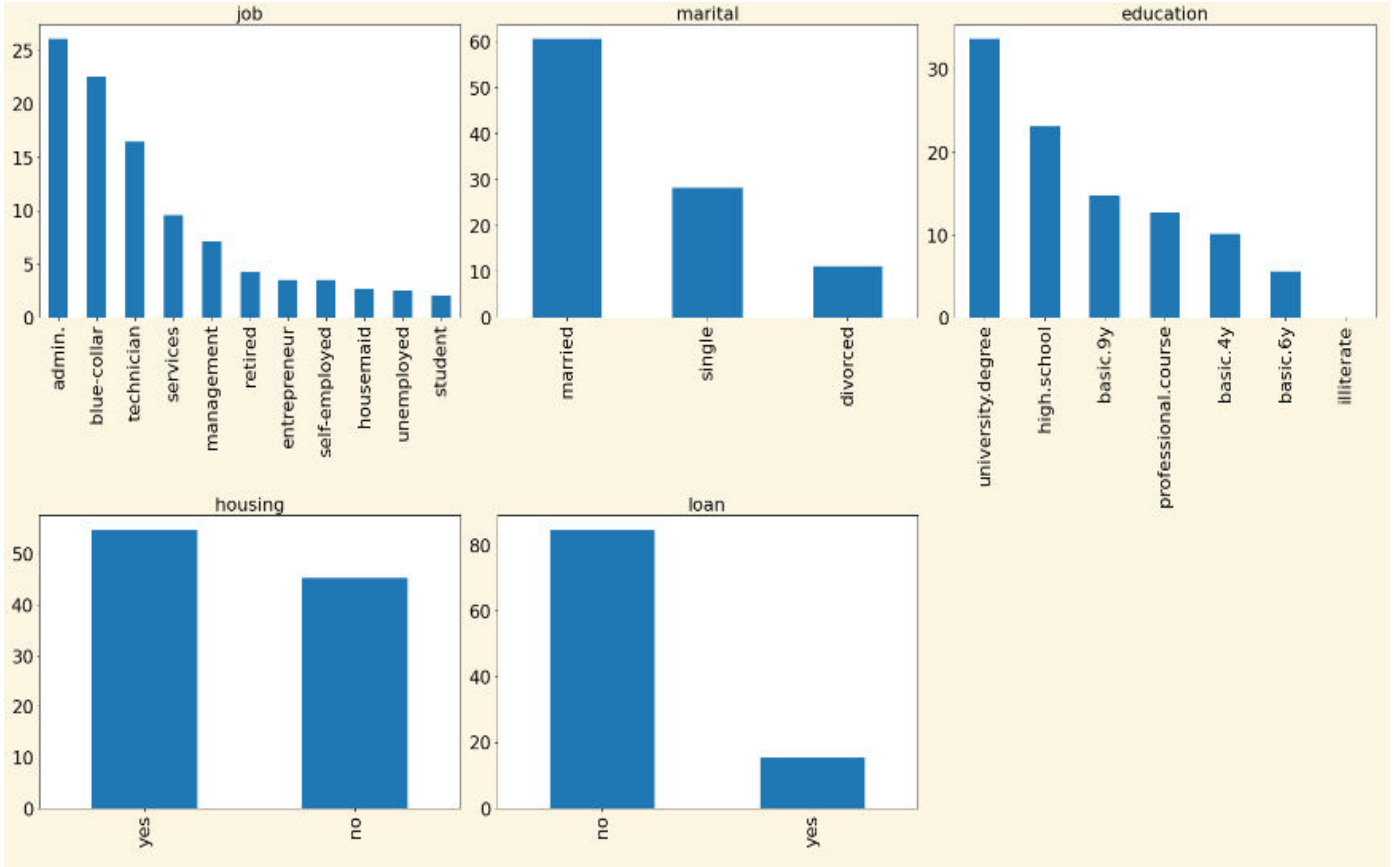
‘Unknown’ values

The first approach – fill unknown values with values of a major class

```
df_option1 = df.copy()
categorical = ['job', 'marital', 'education', 'housing', 'loan']

for i in categorical:
  df_option1[i] = df_option1[i].replace(to_replace = 'unknown', value = df_option1[i].mode().iloc[0]
```

We could see on the previous chart that in all the features values, except default, unknown values take quite a small percentage and adding it to the most frequent category won't affect imbalance between different categories badly. Since default feature has too many unknown values and this data is quite sensitive, it could be better to keep 'unknown' as a separate category.

After filling unknown variables with the most frequent category values we can see that change in values distribution isn't dramatic. However, this approach makes not very gentle assumption that all unknown values belong to a 'mode' category. It especially affects binary features with high imbalance, such as housing and loan.



‘Unknown’ values

The second approach – building logistic regression to predict unknown values in each category

We could solve a classification problem for each column with unknown values and build a model to predict such values. Since we have six features containing unknown values, it's needed to develop six different models.

We will use all the rest variables in prediction cause if we use some individual set of features (which affects each output feature the most), we can lose a part of information, we need to use the same set of features for prediction to save the whole picture. We'll treat ‘unknown’ values in other categorical input features as a separate class.

First, we need to encode out input and output categorical features. I'll use get_dummies method and Label Encoder for this purpose respectivelly.

Further steps are the following:

- Divide the data into two parts. One part will have the present values of the column including the original output column, the other part will have the rows with the missing values.
- Divide the 1st part (present values) into cross-validation set for model selection.
- Train the models and test their metrics against the cross-validated data.
- Finally, with the model, predict the unknown values[4].

```
job : 12
marital : 4
education : 8
default : 3
housing : 3
loan : 3
contact : 2
month : 10
day_of_week : 5
poutcome : 3
y : 2
```

Number of classes in each category (including ‘unknown’)

4. CHIRAG GOYAL. How to Handle Missing Values of Categorical Variables? Analytics Vidhya, URL

‘Unknown’ values

The second approach – building logistic regression to predict unknown values in each category

We can see that models' accuracy isn't high. It could be associated with big number of classes in some features, low relationship between input and output variables in general. Maybe the models could perform better with limited set of input features but, since all features are placed within the current data set, we find it more appropriate to include the same set of variables for all models. The only one model with high accuracy is a model for default feature. Since this feature values distribution is highly imbalanced, even adding class weight parameter to the model didn't solve the problem. So, this approach isn't acceptable for this feature and we again treat 'unknown' as a separate class here.

```
Model accuracy for job is 40.31 %.
Model accuracy for marital is 51.31 %.
Model accuracy for education is 49.61 %.
Model accuracy for housing is 54.47 %.
Model accuracy for loan is 52.97 %.
Model accuracy for default is 99.87 %.
```

	job	marital	education	housing	loan	default
0	housemaid	married	basic.4y	no	no	no
1	services	married	high.school	no	no	no
2	services	married	high.school	yes	no	no
3	admin.	married	basic.6y	no	no	no
4	services	married	high.school	no	yes	no

```
Number of "unknown" occurrences:
- feature -  default , number -  8596 , percentage - 20.8762 %
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week
14817	29	admin.	single	university.degree	no	yes	no	cellular	jul	wed
11819	40	blue-collar	married	basic.9y	unknown	no	no	telephone	jun	fri
6497	27	admin.	single	university.degree	no	yes	no	telephone	may	wed

By and large, after filling the missing values ratio between classes within each feature is kept. Probably, in comparison with the first method this one fills missing values in a smoother way.

Data cleansing and transformation

‘Unknown’ values

The third approach – delete all the rows with unknown variables

Although it's not the best choice, we can perform removal of all the rows with unknown variables.

```
df_del_ukn = df.copy()
for i in df_del_ukn.columns:
    if 'unknown' in set(df_del_ukn[i]):
        df_del_ukn = df_del_ukn[df_del_ukn[i] != 'unknown']
print(f'Dataframe has {df_del_ukn.shape[0]} examples after the removal.')
```

✓ 0.3s

```
Dataframe has 30478 examples after the removal.
```

```
Model accuracy for job is 47.43 %.
Model accuracy for marital is 62.34 %.
Model accuracy for education is 47.81 %.
Model accuracy for housing is 51.32 %.
Model accuracy for loan is 84.38 %.
Model accuracy for default is 99.99 %.
```

	job	marital	education	housing	loan	default
0	housemaid	married	basic.4y	no	no	no
1	services	married	high.school	no	no	no
2	services	married	high.school	yes	no	no
3	admin.	married	basic.6y	no	no	no
4	services	married	high.school	no	yes	no

The fourth approach – use unsupervised learning (KNN)

In this approach, we use unsupervised machine learning, concretely K-Nearest-Neighbors algorithm. The idea is that we use a feature with unknown values as a target variable, other features as input variables and try to find the category in which each unknown value falls. We will use 10 nearest neighbors. The algorithm of preprocessing the data is similar to one we used in logistic regression, we just change the classifier.

We can see that the models perform even better than logistic regression models. However, for default feature we are not going to use this method again and keep 'unknown' class.

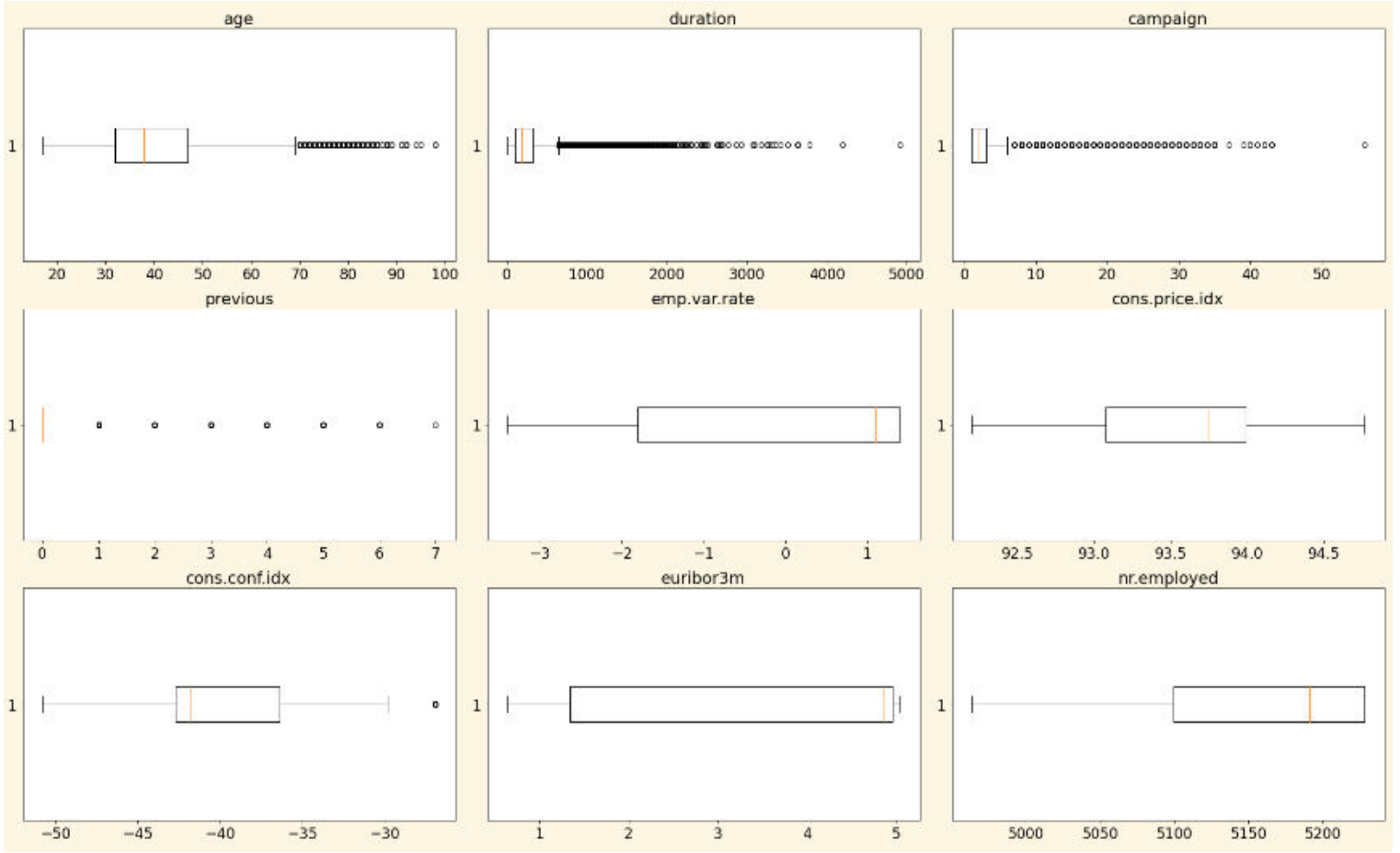
Data cleansing and transformation

Outliers

Since we detected outliers in some variables, we can either keep or remove them.

- Usually outliers cannot be removed without analyzing.
- We could keep the outliers in age variable cause removing the oldest clients will affect the customer base understanding.
- We could keep outliers in duration, campaign and previous cause they are just technical parameters and these outliers aren't related to specific customer groups.
- Also we could keep an outlier in consumer confidence index also for a reason that removing a customer with a higher confidence index will not display diversity of the customers.

Outliers graphs



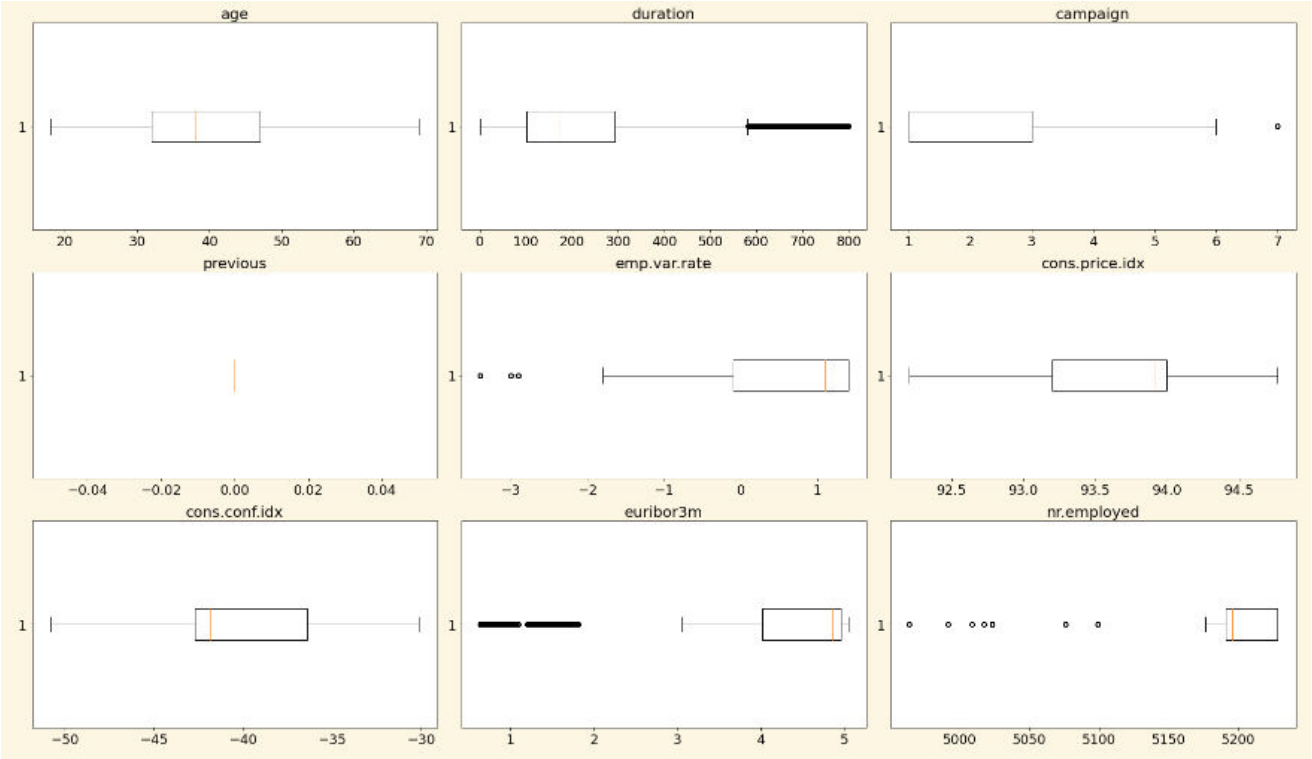
However, there are several outlier removal approaches worth considering.

Data cleansing and transformation

Outliers

The first approach – use boxplot data
First option for detecting outliers is to visualize them. For this purpose we can use boxplot (i.e. whisker plot). We need to find indexes of rows with outliers (according to the graphs) and remove them. We look at the graphs and set limits for each column. After that we sequentially filter dataframe columns.

```
Need to remove 469 outliers from age feature.
Need to remove 1767 outliers from duration feature.
Need to remove 1777 outliers from campaign feature.
Need to remove 5625 outliers from previous feature.
Need to remove 0 outliers from emp.var.rate feature.
Need to remove 0 outliers from cons.price.idx feature.
Need to remove 714 outliers from cons.conf.idx feature.
Need to remove 0 outliers from euribor3m feature.
Need to remove 0 outliers from nr.employed feature.
Total number of duplicates to remove - 9440.
Number of rows after outliers removal - 31748.
Number of rows after outliers removal - 31737.
```



We can see that even after removal boxplots show that new arrays also have values considered as outliers. So, it's quite hard to determine, when we should stop removing outliers. If we keep removing outliers, we can lose some essential part of information.

Data cleansing and transformation

Outliers

The second approach – use quantiles

This method uses IQR (Inter Quartile Range) to find and remove outliers. The idea is to calculate upper and lower bounds and remove the values beyond them.

Although the method is considered very reliable, we can see that if we apply outliers removing to each numeric feature, we lost essential amount of data. Even if we use 2 IQR instead of 1.5 IQR, number of remaining examples almost doesn't change.

Function defined for using quantile approach

```
def remove_outliers(column):
    global df_with_outlier
    #define 1st and 3d quantile - 25% and 75%
    Q1 = np.percentile(df_with_outlier[column], 25,
                       interpolation = 'midpoint')

    Q3 = np.percentile(df_with_outlier[column], 75,
                       interpolation = 'midpoint')
    #define interquantile distance
    IQR = Q3 - Q1
    #calculate lower and upper bounds
    upper = Q3+1.5*IQR
    lower = Q3-1.5*IQR
    #filter the dataframe using the calculated bounds
    df_with_outlier = df_with_outlier[df_with_outlier[column].between(lower, upper)]

#apply the function to each numeric column
for i in columns:   #columns = ['age', 'duration', 'campaign', 'previous', 'emp.var.rate', 'c
  remove_outliers(i)

print(f'Number of rows after removing outliers using quantiles - {df_with_outlier.shape[0]}')
```

```
Initial number of rows - 41188
Number of rows after removing outliers using quantiles - 21384
```

Data cleansing and transformation

Outliers

The third approach – use z-score

Z- Score is also called a standard score. This value/score helps to understand that how far is the data point from the mean. And after setting up a threshold value one can utilize z score values of data points to define the outliers.

- Zscore = (data_point -mean) / std. deviation
- To define an outlier threshold value is chosen which is generally 3.0. As 99.7% of the data points lie between +/- 3 standard deviation (using Gaussian Distribution approach).
- We can see that compared to previous methods, z-score removes quite a small part of data[5].

```
from scipy import stats
import numpy as np
df_z = df.copy()
columns = ['age', 'duration', 'campaign', 'previous', 'emp.var.rate', 'con

list_of_indexes = []
for i in columns:
    z = np.abs(stats.zscore(df_z[i]))
    to_del = (np.where(z > 3))
    print(f'Need to remove {len(to_del[0])} outliers from {i} feature.')
    list_of_indexes.extend(to_del[0].tolist())
print(f'Number of examples to be deleted - {len(set(list_of_indexes))}.')
delete = set(list_of_indexes)
df_z.drop(delete, axis=0, inplace=True)
print(f'Number of examples after outliers removal - {df_z.shape}')
```

```
Need to remove 369 outliers from age feature.
Need to remove 861 outliers from duration feature.
Need to remove 869 outliers from campaign feature.
Need to remove 1064 outliers from previous feature.
Need to remove 0 outliers from emp.var.rate feature.
Need to remove 0 outliers from cons.price.idx feature.
Need to remove 0 outliers from cons.conf.idx feature.
Need to remove 0 outliers from euribor3m feature.
Need to remove 0 outliers from nr.employed feature.
Number of examples to be deleted - 3065.
Number of examples after outliers removal - (38123, 21)
```

5. rajeshsharma7. Detect and Remove the Outliers using Python. Geeks for Geeks, URL

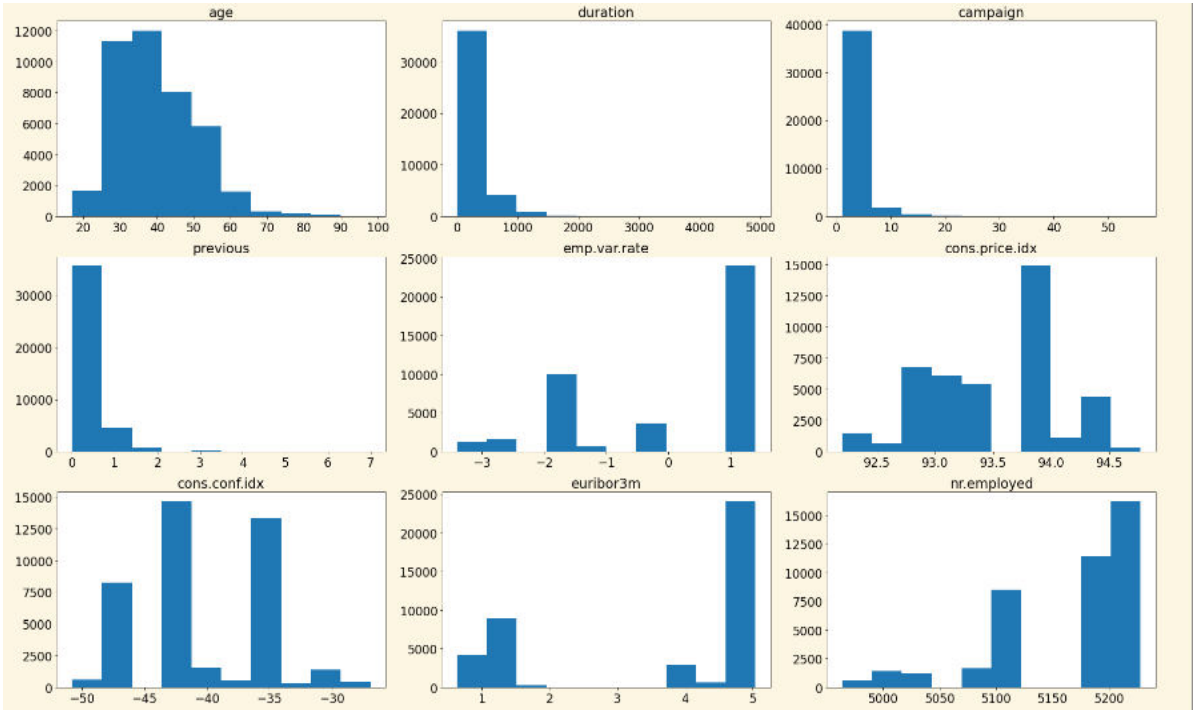
Data cleansing and transformation

Skewed distribution

As for features distributions, each distribution is quite far from a bell shape the normal distribution has. We could transform some of them to normal by applying log function. However, it becomes harder to interpret the results, so we decided to keep the variables distributions in initial condition. Anyway, we can try and see how transformed distributions can look. We can apply this method only to features, where all the values are positive. Also, there is a limitation for features with zero values cause logarithm from zero is not defined.

- After applying this approach we cannot see that forms of distributions have become closer to a bell shape in most cases. Therefore, we'd rather not use this approach and keep the distributions in an initial condition.

Variables distributions

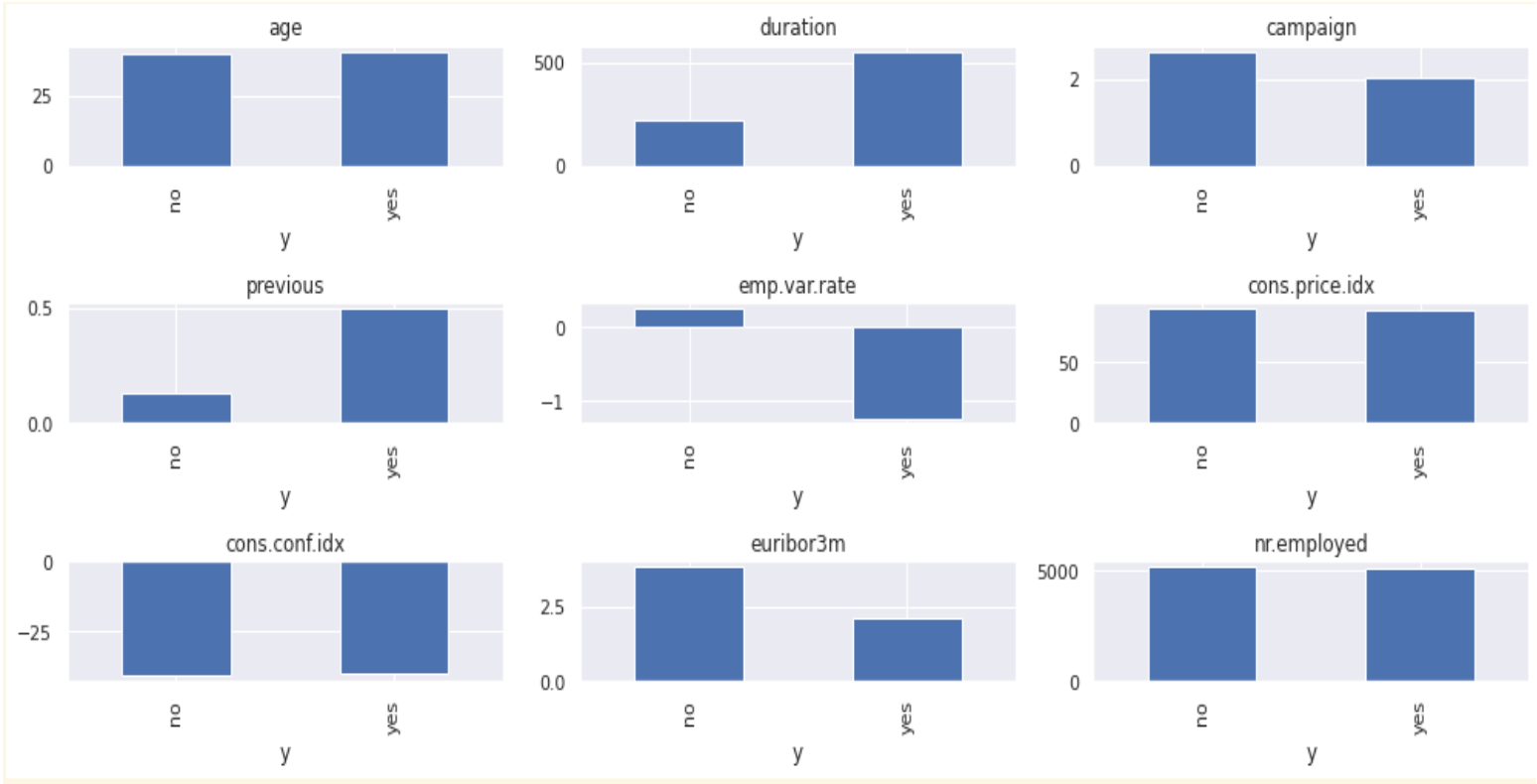



```
columns = ['age', 'duration', 'campaign', 'previous', 'cons.price.idx',

df_skewed = df.copy()
for i in columns:
    df_skewed[i] = [np.log(j) if j != 0 else 0 for j in df_skewed[i]]

df_skewed.head()
```

EDA – explore relations between target and numeric features

Let's move to the output variable and start to explore it's relation to the other variables in the data set.
We plot a ratio between positive and negative answers (subscribed a term deposit or not) and see that our data is imbalanced We need to take this fact into account when building the models.

Now we are going to explore relationships between our output variable and numeric input variables. First, we could visually look if means for y = no and y = yes are different for each numeric feature.



y	age	duration	campaign	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
no	39.910994	220.868079	2.633385	0.132414	0.248885	93.603798	-40.593232	3.811482	5176.165690
yes	40.912266	553.256090	2.051951	0.492779	-1.233089	93.354577	-39.791119	2.123362	5095.120069

EDA – explore relations between target and numeric features

We can see that several numeric features have a visualy significant difference in means. For example, duration of a call for negative answers is much less, customers who answered 'no', were contacted more often (campaign). Vise versa, previously they were contacted less often than customers, who answered 'yes' (previous). Negative answers subgroup has a higher employment variance rate but lower short-term lending rates (eurlibor 3 months). We don't see any significant difference for the rest of the variables.

Another way to find out how the input features are related to the target variable - use statistical tests. For this purpose we will use 2 samples Student t-test. For each numeric feature we will compare samples, where y is equal to 'yes' and 'no' respectively When conducting the t-test, we assume that samples have equal variances and that data is normally distributed (for big samples it's not a strict assumption). As a rule of thumb, we can assume the populations have equal variances if the ratio of the larger sample variance to the smaller sample variance is less than 4:1. Hypothesis:

- H0 - positive and negative y samples means are equal
- H1 - positive and negative y samples means are not equal

Feature	Ratio between variances	P-value (t-test)
age	1.95	7.00324e-10
duration	3.75	0
campaign	0.34	2.04343e-41
previous	4.42	1.68378e-161
emp.var.rate	1.2	0
cons.price.idx	1.46	1.62223e-169
cons.conf.idx	1.95	9.13218e-29
euribor3m	1.13	0
nr.employed	1.84	0

EDA – explore relations between target and numeric features

Obtained results show that almost all features have equal variance except for previous feature. For this feature we pass an argument that variances are not equal. As for check for normality of distributions, all p-values are less than 0.05. That means we need to reject null hypothesis - features are normally distributed. However, since we have a large data set, violation of this assumption is not critical.

Finally, for each numeric features Student t-test shows significant difference for two samples means (subscribed/didn't subscribe a term deposit). Each p-value is less than 0.05 - accepted significance level. We can conclude that there is relationship between the target feature values and the input features values.

Additionally, we decided to conduct ANOVA (analysis of variance) test. Typically, a one-way ANOVA is used when you have three or more categorical, independent groups, but it can be used for just two groups.

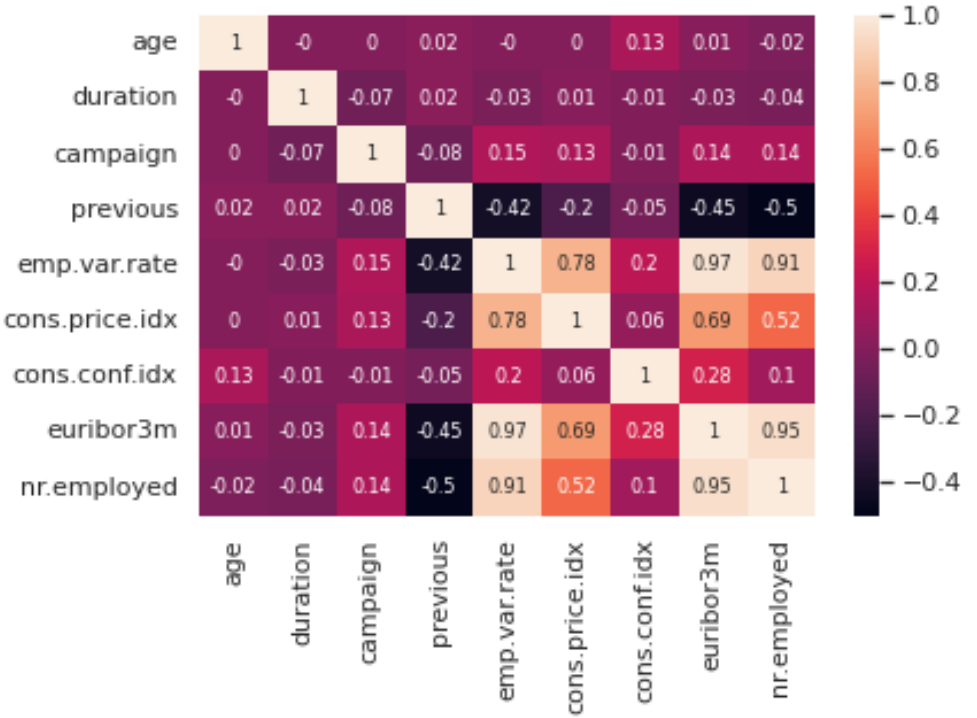
ANOVA results are consistent with t-test results.

Feature	P-value (ANOVA)
age	7.0032438456849 08e-10
duration	0
campaign	2.0434309097339834e-41
previous	0
emp.var.rate	0
cons.price.idx	1.6222328681832695e-169
cons.conf.idx	9.1321757745501 33e-29
euribor3m	0
nr.employed	0

EDA – explore relations between numeric features

Next step is to look at correlations between numeric features themselves.

We can see that several features related to social and economic context are highly correlated. This fact can cause a problem of multicollinearity in the models. We also can check the features for multicollinearity by calculating VIFs (variance inflation factors).
VIFs do not have any upper limit. The lower the value the better. VIFs between 1 and 5 suggest that the correlation is not severe enough to warrant corrective measures.



	Variables	VIF
0	const	528303.388424
1	age	1.018790
2	duration	1.008052
3	campaign	1.038421
4	previous	1.349387
5	emp.var.rate	33.063173
6	cons.price.idx	6.314483
7	cons.conf.idx	2.617565
8	euribor3m	64.331181
9	nr.employed	31.636555

Indeed, several features have extremely high VIF. We should remove some of them to avoid the multicollinearity. Only if we would plan to build a neural network, we can keep highly correlated features. Let's look how the picture can change if we remove 2 variables with the highest VIFs.

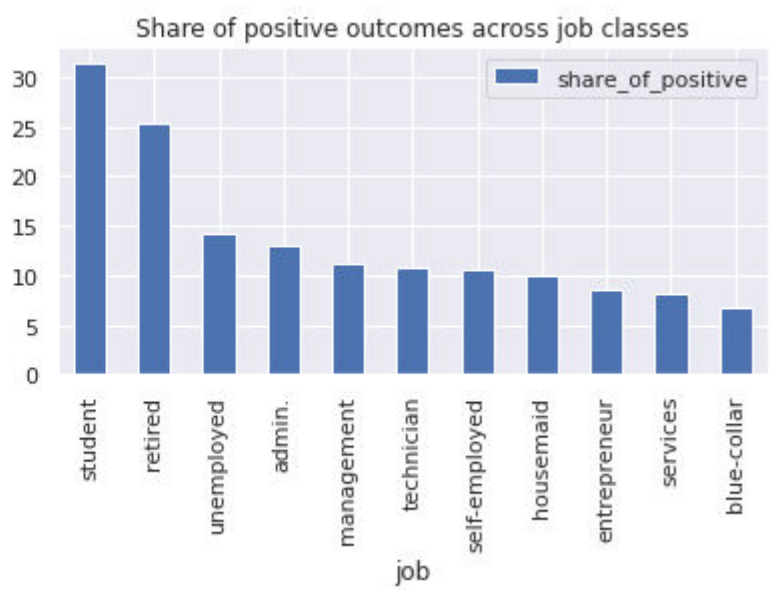
	Variables	VIF
0	const	27962.675266
1	age	1.018470
2	duration	1.007907
3	campaign	1.031206
4	previous	1.344990
5	cons.price.idx	1.390942
6	cons.conf.idx	1.029060
7	nr.employed	1.795510

Now we can see that there's no VIFs greater than 5, so, the multicollinearity problem could be eliminated.

EDA – explore relations between target and categorical features

Job feature

Let's move to analysis of relation between the target variable and input categorical variables. We can plot number of positive and negative answers for each class in categorical features. Also we will plot a percentage of positive answers and provide tabular data.

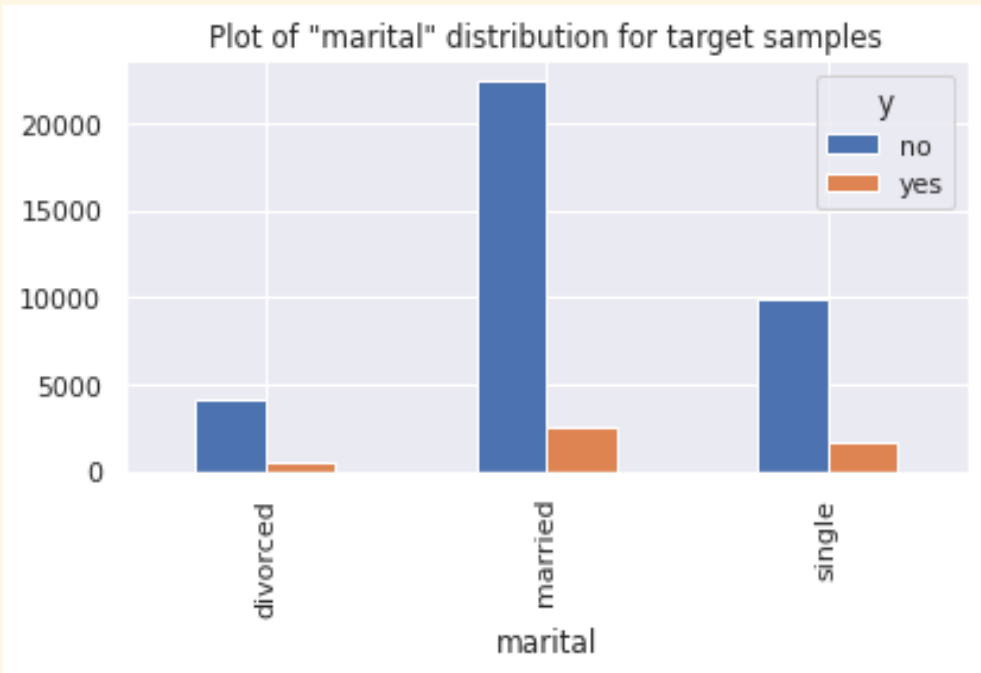


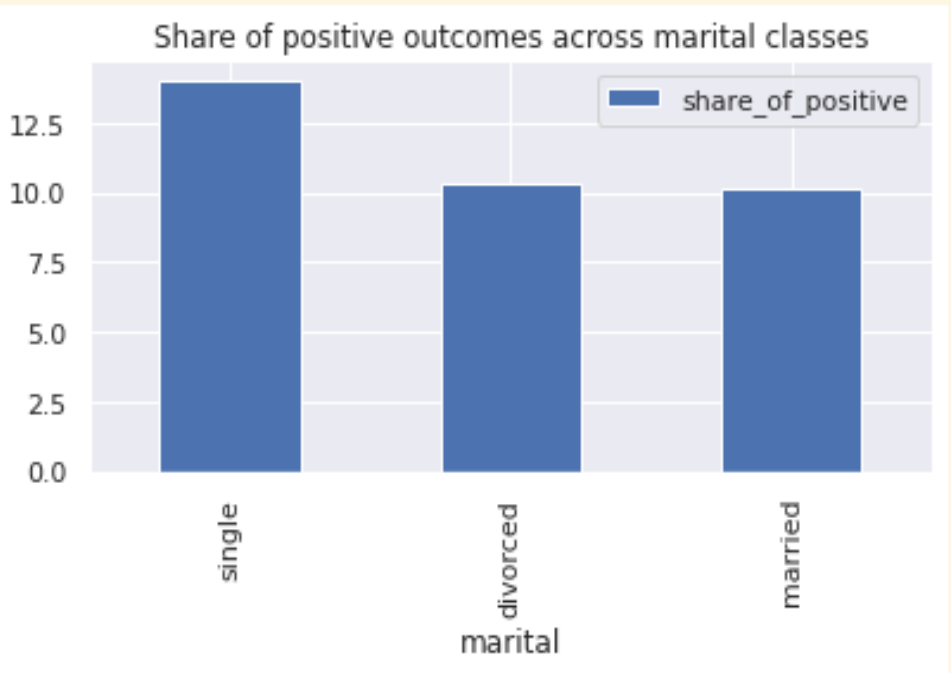
job	share_of_positive
student	31.4286
retired	25.3619
unemployed	14.2012
admin.	13.0385
management	11.2666
technician	10.8300
self-employed	10.4856
housemaid	9.8973
entrepreneur	8.5165
services	8.1366
blue-collar	6.8375

As we can see, ratio between positive and negative answers varies for different job classes. It's quite expectable that for each class share of negative answers is higher. We can see that job feature has no low , top three classes that have the biggest share of y = 'yes' - student, retired, unemployed. Since we can see relation between the target variable and type of job, so, it's needed to keep this feature for a further analysis.

EDA – explore relations between target and categorical features

Marital feature



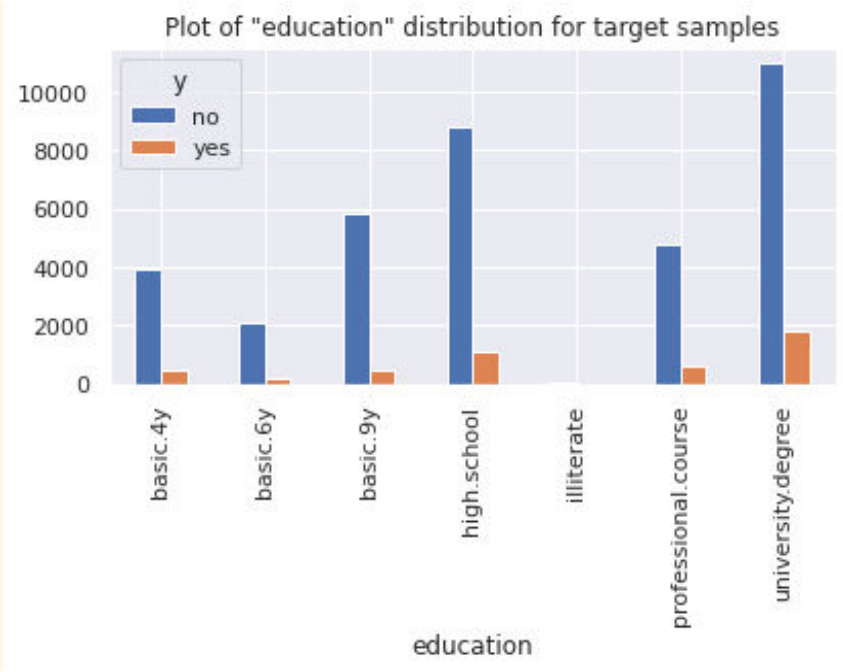


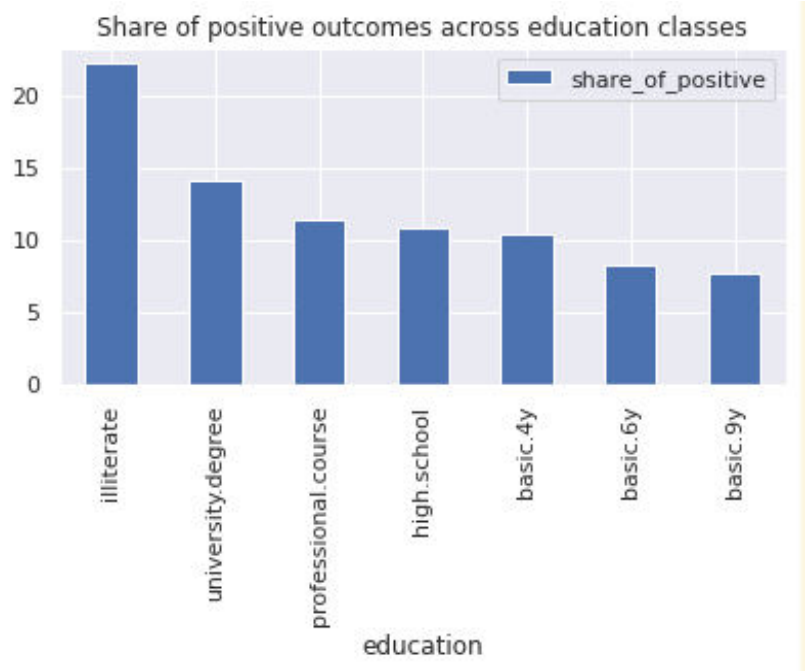
Marital status is also connected with y because single customers are more inclined to give a positive answer and subscribe to the term deposit. However there are much less divorced customers than married ones, shares of positive outcomes is almost equal.

marital	share_of_positive
single	14.0093
divorced	10.3359
married	10.1669

EDA – explore relations between target and categorical features

Education feature



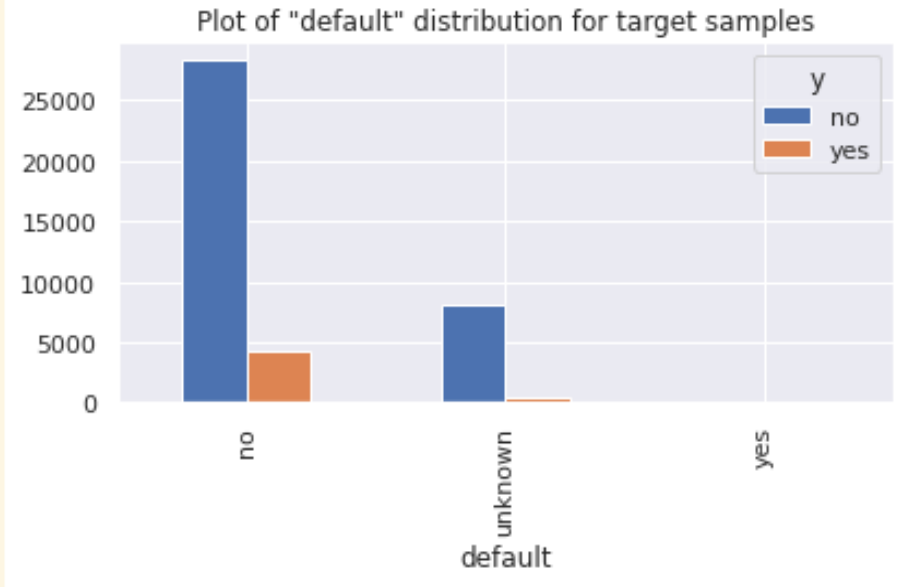


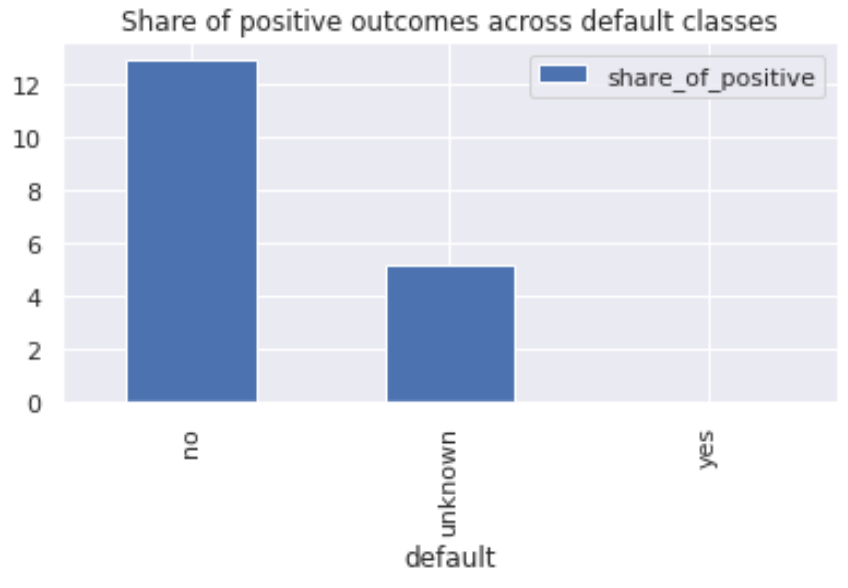
education	share_of_positive
illiterate	22.2222
university.degree	14.1359
professional.course	11.3800
high.school	10.8955
basic.4y	10.4138
basic.6y	8.1861
basic.9y	7.6621

Although a share of illiterate customers is extremely small in the entire data set, this class has the highest share of positive outcomes. It's interesting that there's no obvious relation between illiteracy rate and share of positive outcomes because the second place belongs to customers with university degree, the third one - to customers who completed some professional courses. Anyway, this feature might have an impact on the target feature.

EDA – explore relations between target and categorical features

Default feature



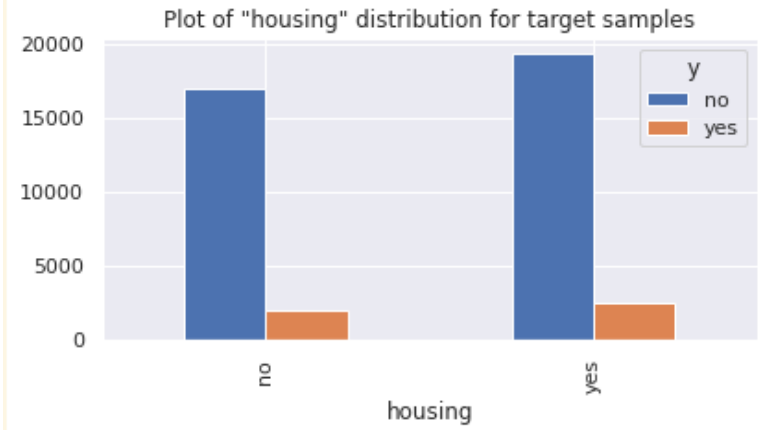


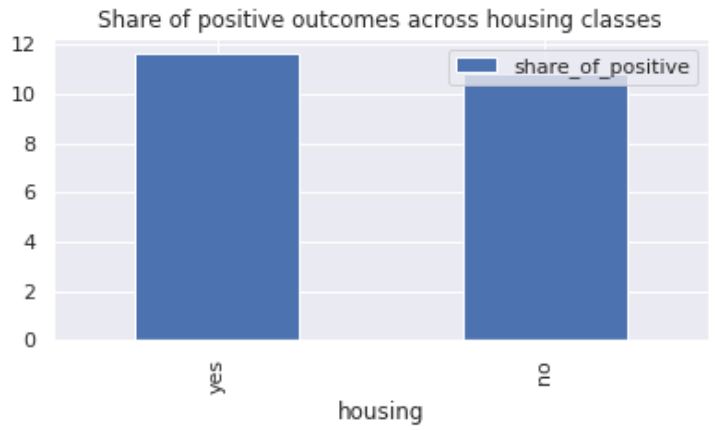
default	share_of_positive
no	12.8803
unknown	5.1536
yes	NaN

As for default feature, share of customers who has credit in default is very low in general as expected. Looking at tabular data we can see that there's no customers with credit in default who subscribed the term deposit. We can consider this feature in model building to find out, how the fact that customer doesn't want to reveal information if he has credit in default or not can affect the target variable. However, in general, this feature seems to be not extremely informative because we have only two classes for further analysis and one of them could potentially belong entirely to the second class ('unknown' to 'no'). In this case we could have zero entropy what isn't good for predicting the target.

EDA – explore relations between target and categorical features

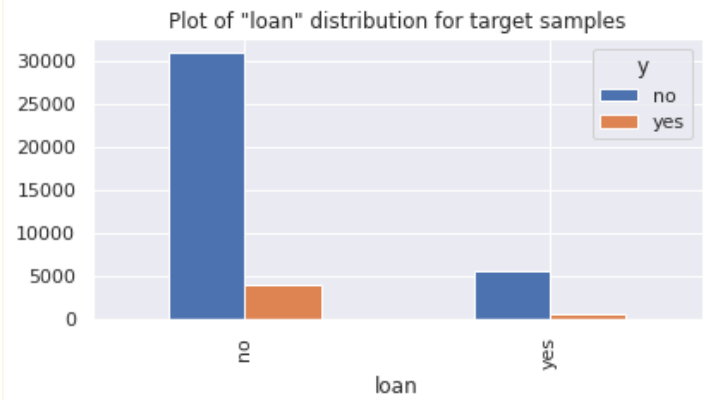
Housing and loan features

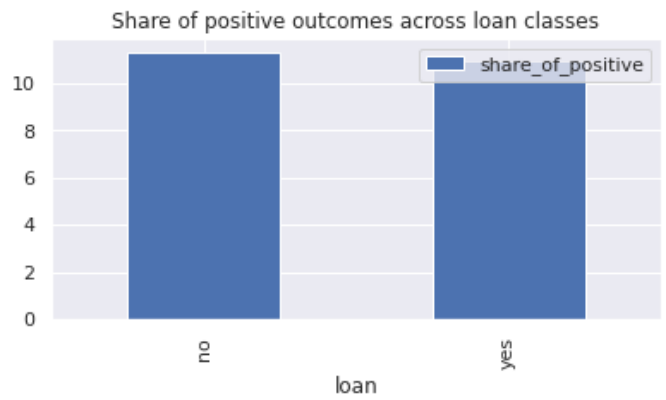




housing	share_of_positive
yes	11.6631
no	10.8110

We can see that there's almost no difference for two y classes, either in number of positive and negative outcomes, or in shares of positive outcomes. So, this feature tends to have low predictive power.



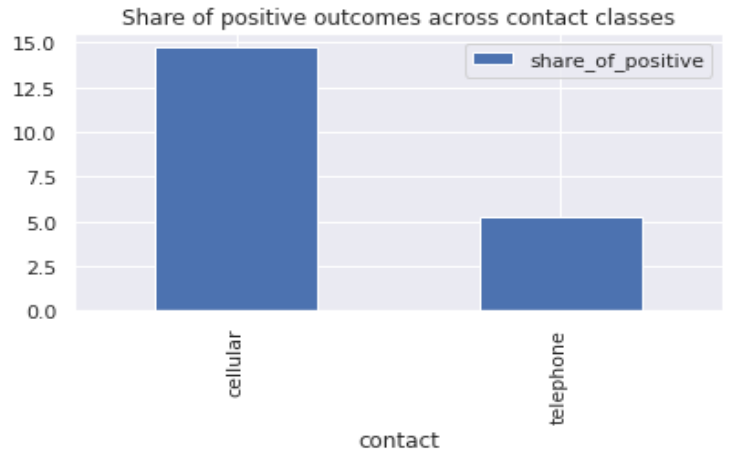


loan	share_of_positive
no	11.3268
yes	10.9280

The situation with loan feature is quite similar to housing feature, except that number of positive outcomes in much lower. However, shares of positive outcomes across two classes are almost equal. So, this feature also seems to be not very informative.

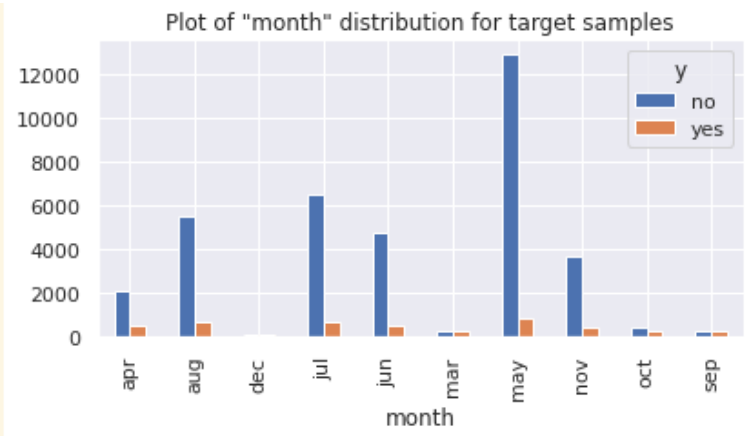
EDA – explore relations between target and categorical features

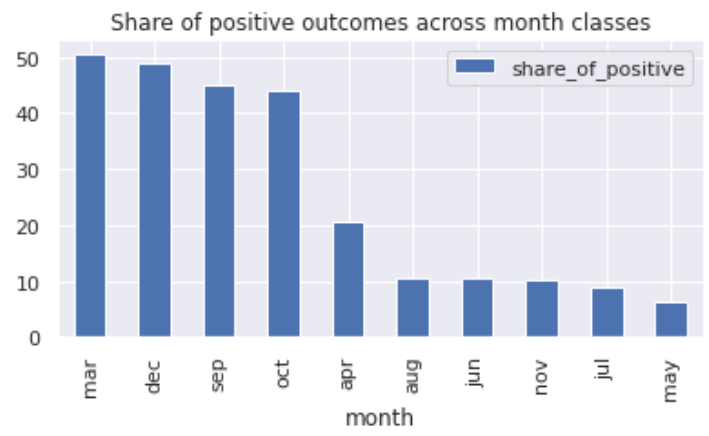
Contact and month features



contact	share_of_positive
cellular	14.7389
telephone	5.2324

As for type of contact, we can see that relationship exists, so we plan to keep this feature for further analysis.



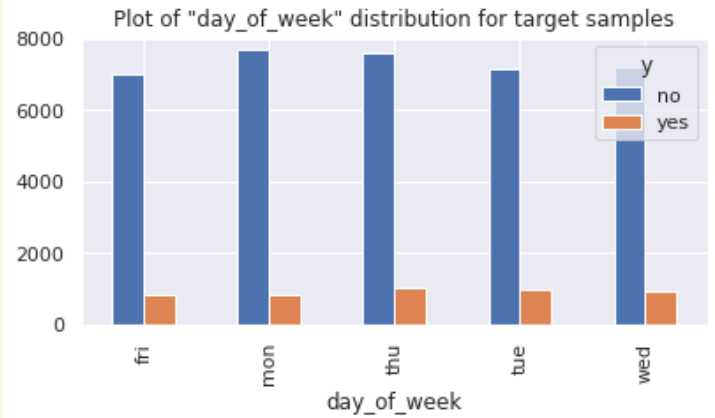


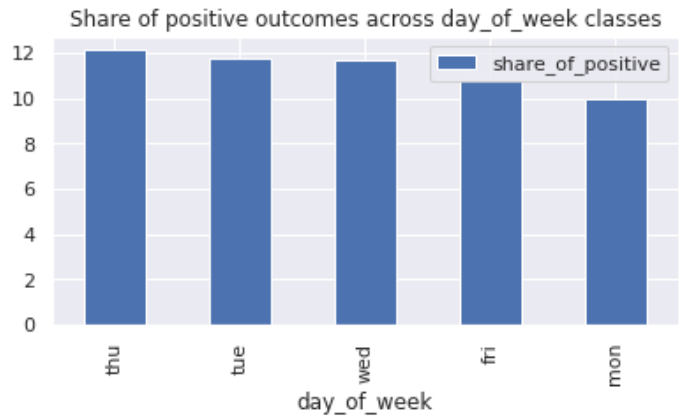
month	share_of_positive
mar	50.5495
dec	48.9011
sep	44.9123
oct	43.9331
apr	20.4865
aug	10.6056
jun	10.5115
nov	10.1463
jul	9.0389
may	6.4357

Last contact month of year can affect the target, as wee can see. However, the relationship isn't obvious - the first three places are taken by months from different year seasons. So, maybe there's no any regularity here.

EDA – explore relations between target and categorical features

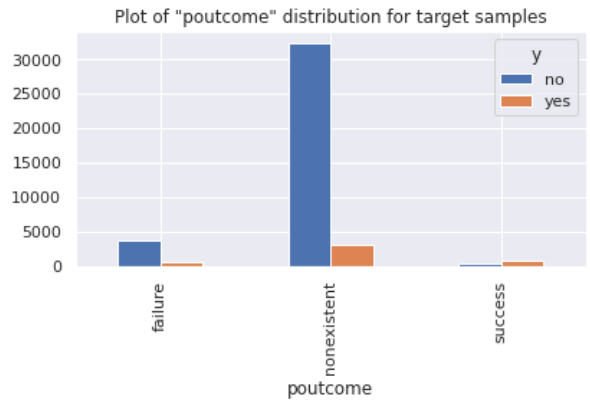
Day of week and poutcome features

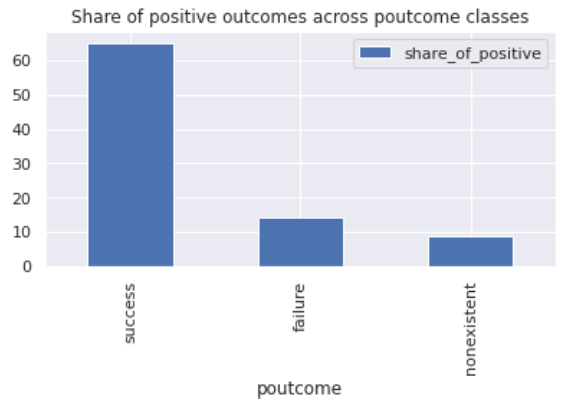




day_of_week	share_of_positive
thu	12.1142
tue	11.7858
wed	11.6671
fri	10.8101
mon	9.9507

As for day of the week, we cannot see any strong relationship. On Mondays and Fridays share of positive outcomes is a little bit less. Maybe due to this slight difference we shouldn't exclude this feature from the further analysis and check it's impact on the target.



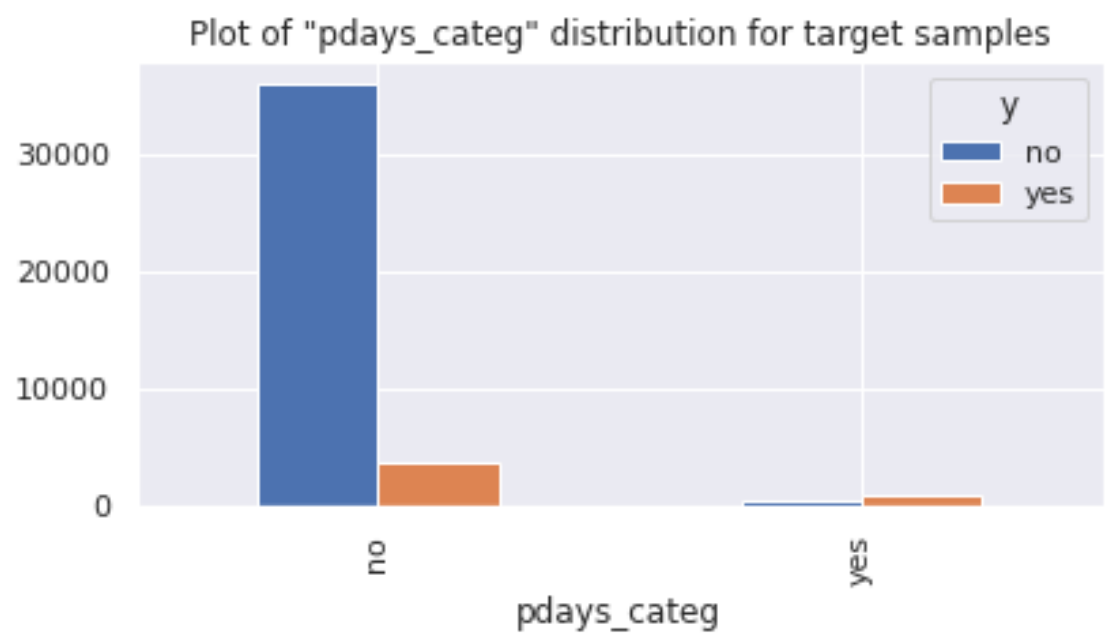


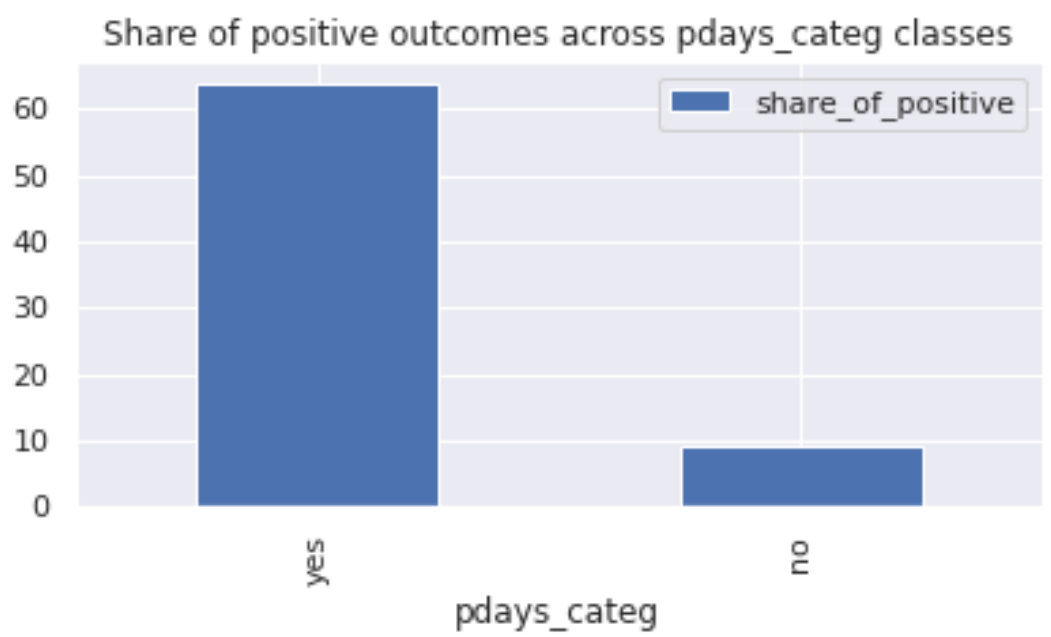
poutcome	share_of_positive
success	65.1129
failure	14.2286
nonexistent	8.8324

Outcome of the previous marketing campaign can influence the outcome of the current marketing campaign. We should keep this feature.

EDA – explore relations between target and categorical features

Pdays_categ feature





Finally, the fact that the client was contacted earlier also affects the target. This created feature is quite close to the previous one in meaning, that's why we should try to build the model with this feature and without it to see its personal impact.

pdays_categ	share_of_positive
yes	63.8284
no	9.2585

EDA – explore relations between target and categorical features

To support our assumptions, we can use a Chi-square test for categorical features.

Hypothesis:

- H0: two categorical variables being compared are independent of each other.
- H1: two categorical variables being compared are dependent of each other.

Also we could use SelectKBest method to select the categorical features which have the highest level of dependence with the target but we'd like to have a look at all the features.

As a result, we can see that all the features except housing, month and loan have p-values less than 0.05. This means that there's dependence between these features and the target. As for three features with p-value more than 0.05, their usefulness for analysis is being questioned. We could compare predictive power of models with and without them.

Chi-square scores and p-values

	ftr	score	pval
0	pdays_categ	4186.8682	0.0000
1	contact	547.7785	0.0000
2	default	321.8911	0.0000
3	poutcome	98.2633	0.0000
4	job	92.9062	0.0000
5	education	86.2995	0.0000
6	marital	27.0094	0.0000
7	day_of_week	10.2336	0.0014
8	housing	3.4657	0.0627
9	month	1.9178	0.1661
10	loan	0.7154	0.3977

EDA – explore relations between categorical features

Additionally to the main analysis of relationships between the features and the target we also can explore relationships between the categorical features using label encoding and visualization. We will explore just several features not to be repetitive.

- more unemployed tend to be singles
- entrepreneurs, management and self-employed have a higher education level
- housemaids are more prone to have a credit in default
- entrepreneurs and unemployed have a little higher probability to have a loan
- unemployed and retired were contacted before more often

- married in average have a little bit higher education level
- married are less prone to have a credit in default
- married are contacted by cellular more often
- married were previously contacted more often

- illiterate, basic 6y and university degree are more prone to have a credit in default
- customers with university degree are contacted by cellular more often
- customers with university degree and professional course were contacted before more often

- unemployed have a higher probability to have a credit in default
- people with a credit in default are less prone to have a housing, they don't have loans and weren't contacted neither by cellular, nor by telephone
- people with a credit in default had a successful outcome in a previous marketing campaign less often
- people with a credit in default weren't contacted before

- people with a housing loan were contacted by cellular more often
- people with a housing loan have a personal loan with a higher probability
- people with a housing loan were contacted before more often

EDA – explore relations between categorical features

Additionally to the main analysis of relationships between the features and the target we also can explore relationships between the categorical features using label encoding and visualization. We will explore just several features not to be repetitive.

- people with a higher education level are contacted by telephone more often
- people who has a credit in default are contacted by cellular more often
- people who was contacted by cellular in average were contacted before less often

- singles had a successful outcome of the previous campaign more often
- people with a higher education level had a successful outcome of the previous campaign more often
- people with a credit in default had an unsuccessful outcome of the previous campaign more often
- people contacted by telephone had an unsuccessful outcome of the previous campaign more often
- people with a successful campaign outcome were contacted more often

Final recommendations

Technical part

Data cleansing and transformation:

- Drop duplicates.
- Transform pdays feature to categorical.
- Fill unknown values using KNN approach.
- Keep outliers (but also consider a z-score approach as a benchmark).
- Keep skewed distribution (but also consider a binning option as a benchmark).

Technical part

Features selection:

- Remove numerical features – euribor3m and emp.var.rate to avoid a multicollinearity problem. In case of building a neural network we can try to keep them.
- Consider models with and without duration feature, as stated in he task.
- Categorical features that might be excluded from the analysis – housing, month, loan, day of week, default. Need to explore their impact on the target by building models with and without these features.

Business part

At the first glance we can conclude, that customers tend to subscribe to a term deposit if:

- They are students, retired or unemployment
- They are single
- They either illiterate or have university degree/professional courses
- They have no a credit in default
- They rather have a housing loan
- They rather don’t have a personal loan
- They were contacted by a cellular
- They rater were contacted in March, December, September or October
- They participated in a previous marketing campaign and subscribed

Recommended models

- Type of problem – classification.
- Type of classification – binary classification.
- Type of learning – supervised learning.

Logistic regression

Estimates the probability of an event occurring. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

Decision Tree

Creates a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Random Forest

Operates by constructing a multitude of decision trees at training time. For classification tasks, the output is the class selected by most trees.

Neural Network

K Nearest Neighbors

Attempts to determine what group a data point is in by looking at the data points around it.

Support Vector Machine

Finds a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

Naive Bayes

Applies Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

Models performance

- We consider a set of models with different hyperparameters. Besides the mentioned models, we also decided to consider a boosting model (Ada Boost classifier).
- We measured the models by accuracy metric. Accuracy is the measurement used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the training data.
- Later we decided to estimate models using f1 score metric, since this metric is more appropriate foe imbalanced datasets.
- Finally, we have chosen Logistic regression with class weights for interpretation and estimating features importance, since it can be conducted easier compared to the other models.

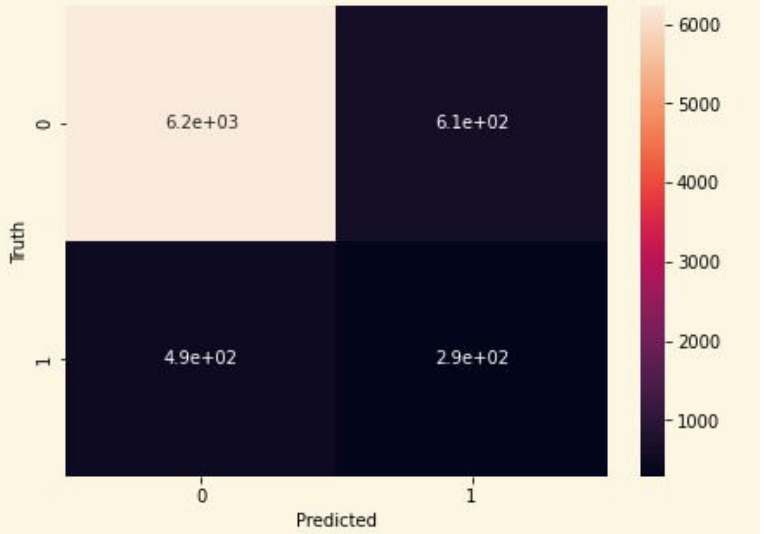
Model performance comparison

Model	Accuracy	f1-score
Random Forest	0.887	0.349
KNN	0.885	0.369
Ada Boosting	0.881	0.357
SVM	0.873	0.362
Decision Tree	0.869	0.283
Logistic Regression	0.864	0.321
Naive Bayes	0.803	0.377
Neural Network	0.897	-
Logistic Regression with class weights	0.820	0.430
Random Forest with binned and encoded numeric features	0.868	0.504
Random Forest with duration	0.871	0.498

Results comparison

Random forest

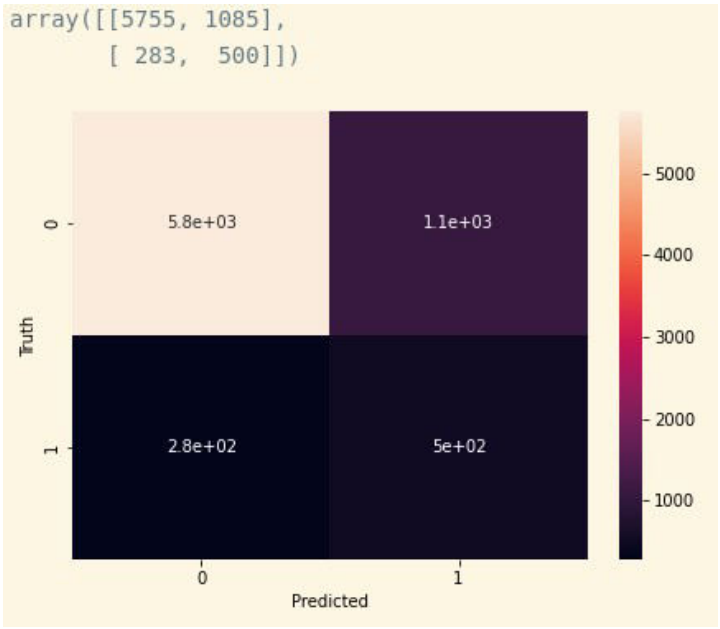
```
array([[6231,  609],
       [ 493,  290]])
```



	precision	recall	f1-score	support
0	0.93	0.91	0.92	6840
1	0.32	0.37	0.34	783
accuracy			0.86	7623
macro avg	0.62	0.64	0.63	7623
weighted avg	0.86	0.86	0.86	7623

Logistic regression with weights

```
array([[5755, 1085],
       [ 283,  500]])
```



	precision	recall	f1-score	support
0	0.95	0.84	0.89	6840
1	0.32	0.64	0.42	783
accuracy			0.82	7623
macro avg	0.63	0.74	0.66	7623
weighted avg	0.89	0.82	0.85	7623

Categorical features

We can interpret the coefficients of each feature, mentioning “In average, other things being equal”. Features, whose coefficients are not statistically significant, according to the logit model, highlighted in gray.

Poutcome:

- Failure - failure in previous outcome is associated with a 3.34 times increase in the probability that customer will subscribe the term deposit.
- Success - successful previous outcome is associated with a 3.96 times increase in the probability that customer will subscribe the term deposit.

Months:

- Nov – call in November is associated with a 2.81 times increase in the probability that customer will subscribe the term deposit.
- Apr - call in April is associated with a 1.93 times increase in the probability that customer will subscribe the term deposit.
- Oct - call in October is associated with a 1.45 times increase in the probability that ...
- Aug - call in August is associated with a 1.21 times increase in the probability that ...
- Jul - call in July is associated with a 1.08 times increase in the probability that ...
- Jun - call in June is associated with a 24.42% reduction in the probability that ...
- Sep - call in September is associated with a 30.48% reduction in the probability that ...
- Mar - call in March is associated with a 31.06% reduction in the probability that ...
- Dec - call in December is associated with a 45.35% reduction in the probability that ...

Pdays_categ:

- Yes_pdays - calling a customer before is associated with a 1.7 times increase in the probability that ...

- **Education** – having education 1 stage higher is associated with a 1.67 times increase in the probability that ...

Categorical features

We can interpret the coefficients of each feature, mentioning “In average, other things being equal”. Features, whose coefficients are not statistically significant, according to the logit model, highlighted in gray.

Job:

- Self-employed - contacting a self-employed customer is associated with a 1.48 times increase in the probability that customer will subscribe the term deposit.
- Retired - contacting a retired customer is associated with a 1.41 times increase in the probability that customer will subscribe the term deposit.
- Admin - contacting an admin. customer is associated with a 1.11 times increase in the probability that ...
- Services - contacting a services customer is associated with a 1.05 times increase in the probability that ...
- Entrepreneur - contacting an entrepreneur customer is associated with a 1.02 times increase in the probability that ...
- Student - contacting a student customer is associated with a 1.02 times increase in the probability that ...
- Technician - contacting a technician customer is associated with a 1.01 times increase in the probability that ...
- Unemployed - contacting an unemployed customer is associated with a 2.6% reduction in the probability that ...
- Management - contacting a management customer is associated with a 5.51% reduction in the probability that ...
- Blue-collar - contacting a blue-collar customer is associated with a 8.86% times reduction in the probability that ...

Marital status:

- Divorced - divorced customer is associated with a 1.28 times increase in the probability that ...
- Single **-** single customer is associated with a 1.07 times increase in the probability that ...

Contact type:

- Cellular - contacting a customer by cellular is associated with a 40.98% reduction in the probability that customer will subscribe the term deposit.

Results interpretation

Numerical features

We can interpret the coefficients of each feature, mentioning “In average, other things being equal”. Features, whose coefficients are not statistically significant, according to the logit model, highlighted in gray.

Previous :
An increase by 1 call addressed to a customer before is associated with a 41.58 times increase in the probability that customer will subscribe the term deposit.

Cons.conf.idx:
An increase of 1 point in consumer confidence index is associated with a 2.13 times increase in the probability that customer will subscribe the term deposit.

Age:
An increase in customer age by 1 year is associated with a 1.57 times increase in the probability that ...

Cons.price.idx :
An increase of 1 point in consumer confidence index is associated with a 21.33% reduction in the probability that ...

Campaign:
An increase by 1 call addressed to a customer during this campaign is associated with a 85.94% reduction in the probability that ...

Nr.employed:
An increase of 1 person employed is associated with a 91.85% reduction in the probability that ...

Bank Marketing (Campaign) - Group Project

Data Glacier

Your Deep Learning Partner

Thank You