About Dataset
Predict Term Deposit

Introduction

Bank has multiple banking products that it sells to customer such as saving account, credit cards, investments etc. It wants to which customer will purchase its credit cards. For the same it has various kind of information regarding the demographic details of the customer, their banking behavior etc. Once it can predict the chances that customer will purchase a product, it wants to use the same to make pre-payment to the authors.

In this part I will demonstrate how to build a model, to predict which clients will subscribing to a term deposit, with inception of machine learning. In the first part we will deal with the description and visualization of the analysed data, and in the second we will go to data classification models.

Strategy

-Desire target
-Data Understanding
-Preprocessing Data
-Machine learning Model
-Prediction
-Comparing Results

Desire Target

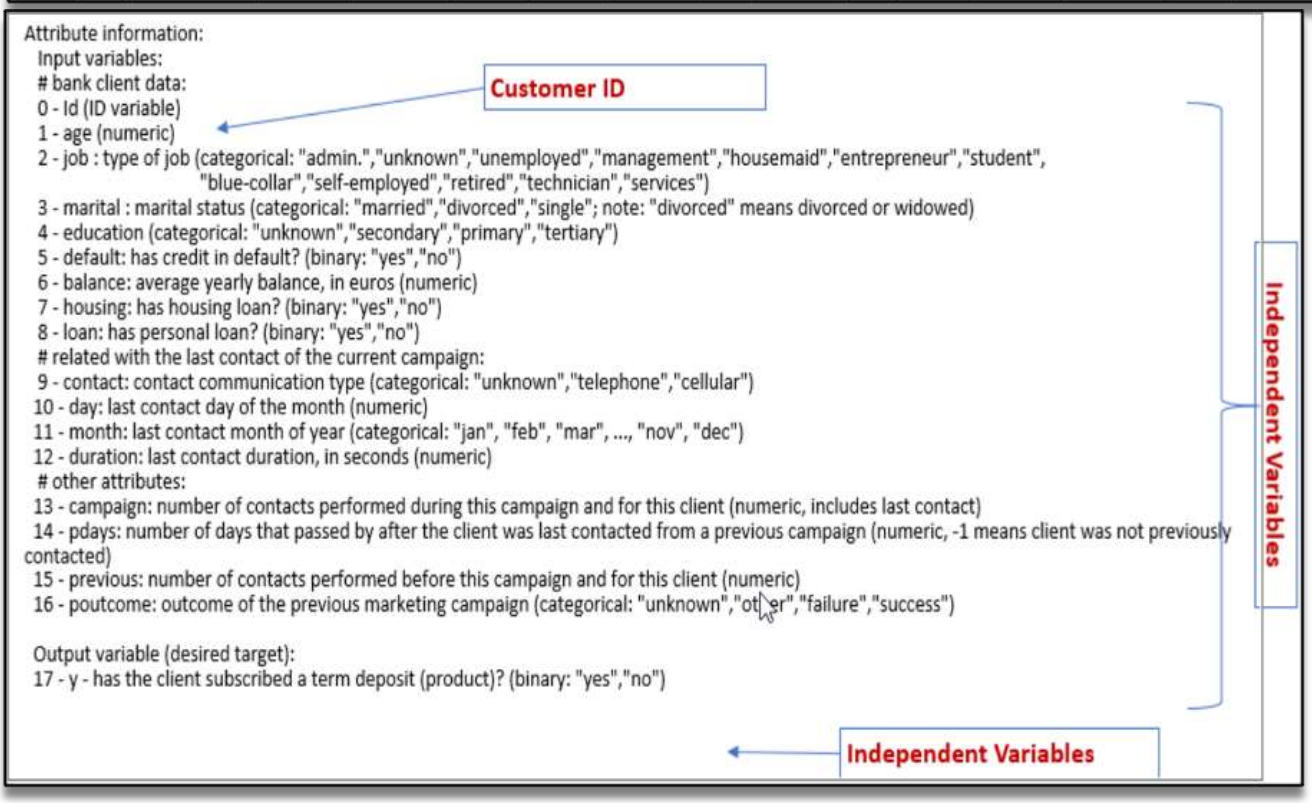
Predict if a client will subscribe (yes/no) to a term deposit — this is defined as a classification problem.

Data

The dataset (Assignment-2*data.csv) used in this assignment contains bank customers' data. File name: Assignment-2*Data
File format: . csv
Numbers of Row: 45212
Numbers of Attributes: 17 non- empty conditional attributes attributes and one decision attribute.

Assignment_2_Data

Id	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
1001	999	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1002	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
1003	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
1004	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
1005	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
1006	35	management	married	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	unknown	no



Exploratory Data Analysis (EDA)

Data pre-processing is a main step in Machine Learning as the useful information which can be derived it from data set directly affects the model quality so it is extremely important to do at least necessary preprocess for our data before feeding it into our model.

In this assignment, we are going to utilize python to develop a predictive machine learning model. First, we will import some important and necessary libraries.

Below we are can see that there are various numerical and categorical columns. The most important column here is y, which is the output variable (desired target): this will tell us if the client subscribed to a term deposit(binary: ‘yes’,’no’).

	Id	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	1001	999.0	management	married	tertiary	no	2143.0	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	1002	44.0	technician	single	secondary	no	29.0	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	1003	33.0	entrepreneur	married	secondary	no	2.0	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	1004	47.0	blue-collar	married	unknown	no	1506.0	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	1005	33.0	unknown	single	unknown	no	1.0	no	no	unknown	5	may	198	1	-1	0	unknown	no

We must to check missing values in our dataset if we do have any and do, we have any duplicated values or not.

```
Id           0
age          9
job          0
marital      0
education    0
default      0
balance      3
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

We can see that in 'age' 9 missing values and 'balance' as well 3 values missed. In this case based that our dataset it has around 45k row I will remove them from dataset. on Pic 1 and 2 you will see before and after.

```
Id             0
age            0
job            0
marital        0
education      0
default        0
balance        0
housing        0
loan           0
contact        0
day            0
month          0
duration       0
campaign       0
pdays          0
previous       0
poutcome       0
y              0
dtype: int64
```

From the above analysis we can see that only 5289 people out of 45200 have subscribed which is roughly 12%. We can see that our dataset highly unbalanced. we need to take it as a note.

```
y
no      39911
yes      5289
dtype: int64
```

Our list of categorical variables.

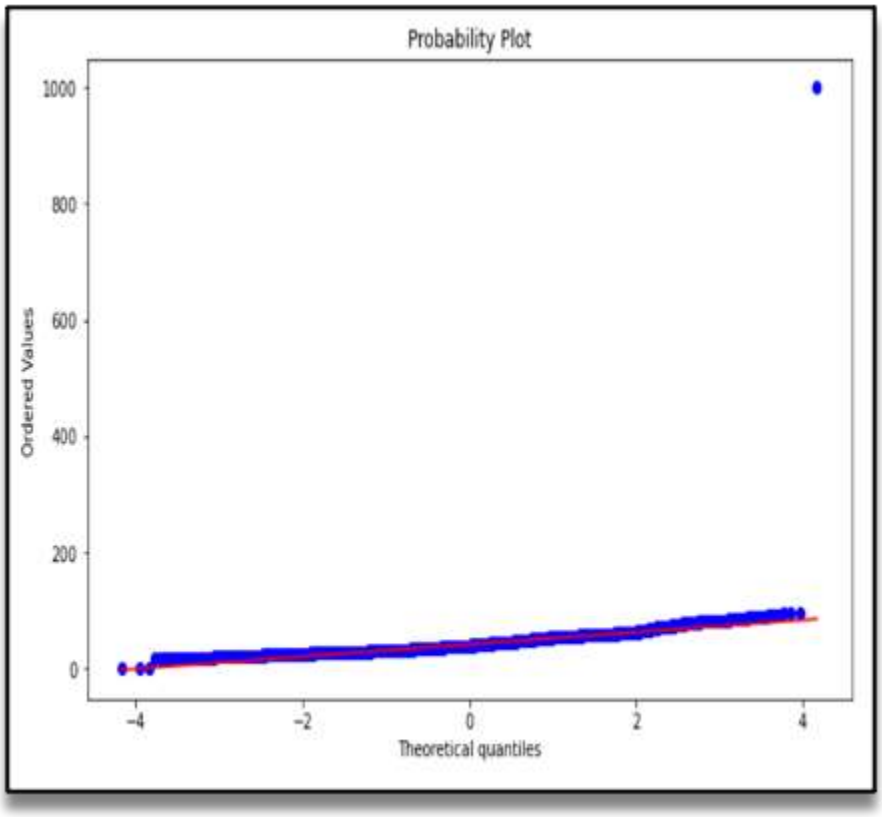
	Id	age	balance	day	duration	campaign	pdays	previous
0	1001	999.0	2143.0	5	261	1	-1	0
1	1002	44.0	29.0	5	151	1	-1	0
2	1003	33.0	2.0	5	76	1	-1	0
3	1004	47.0	1506.0	5	92	1	-1	0
4	1005	33.0	1.0	5	198	1	-1	0

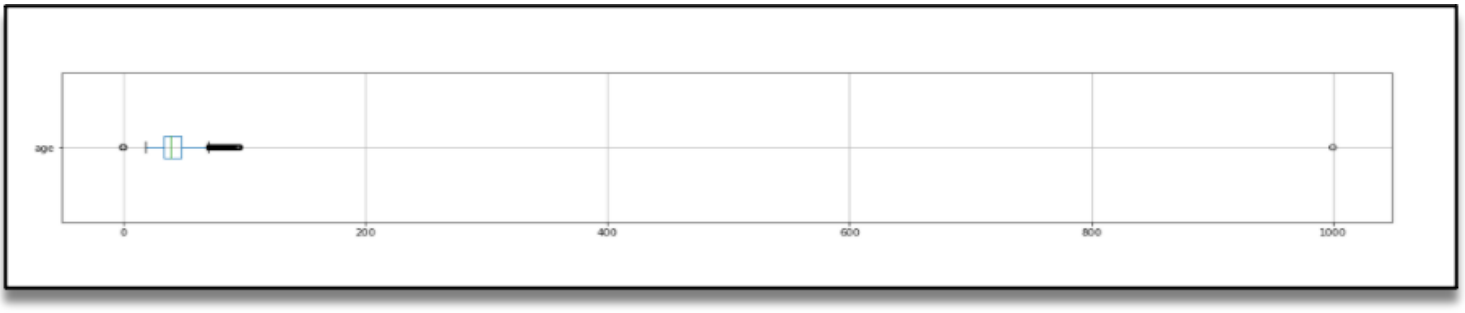
Our list of numerical variables.

	job	marital	education	default	housing	loan	contact	month	poutcome	y
0	management	married	tertiary	no	yes	no	unknown	may	unknown	no
1	technician	single	secondary	no	yes	no	unknown	may	unknown	no
2	entrepreneur	married	secondary	no	yes	yes	unknown	may	unknown	no
3	blue-collar	married	unknown	no	yes	no	unknown	may	unknown	no
4	unknown	single	unknown	no	no	no	unknown	may	unknown	no

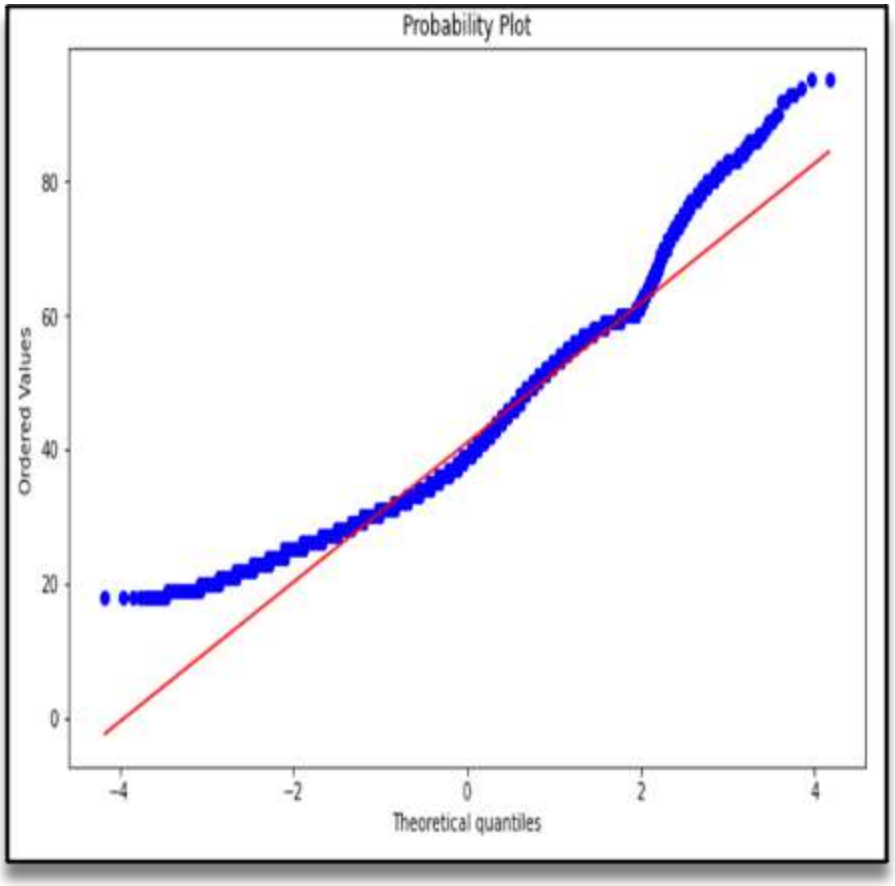
"Age" Q-Q Plots and Box Plot.

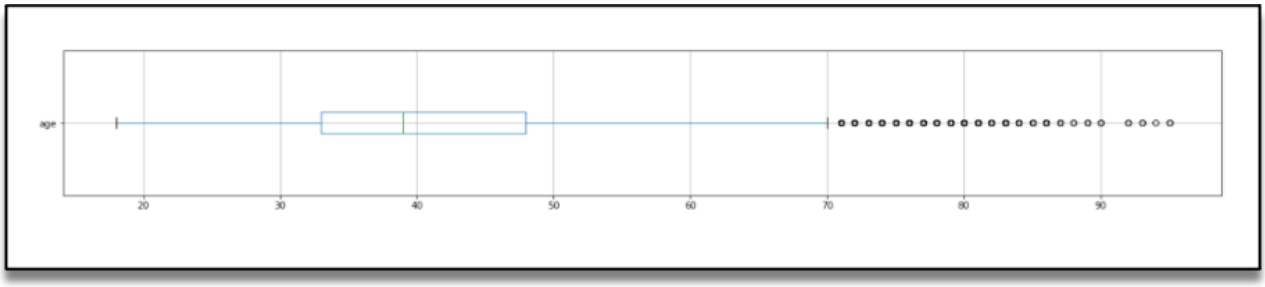
In above boxplot we can see that some point in very young age and as well impossible age. So,



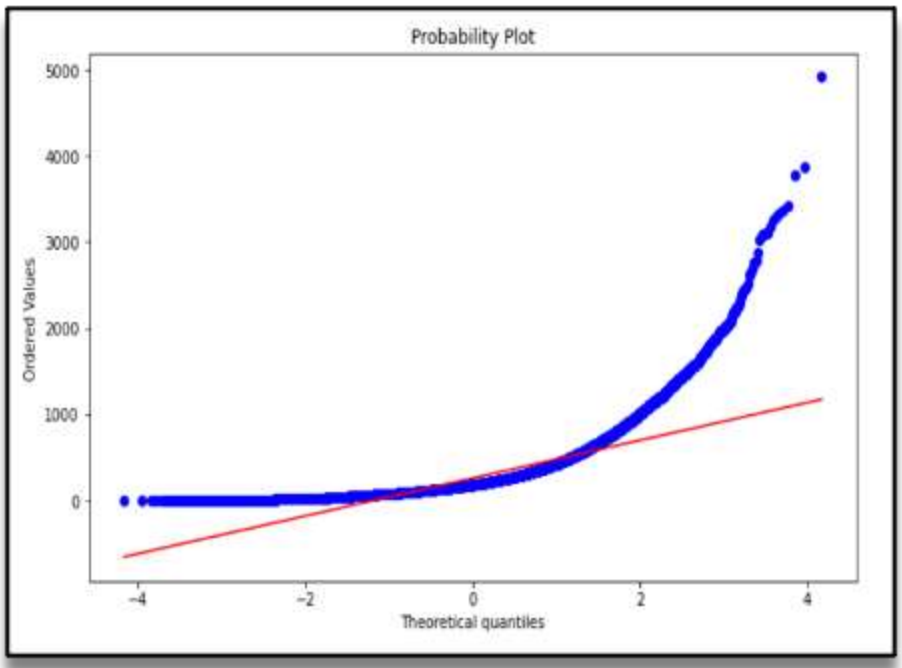


Now, we don’t have issues on this feature so we can use it



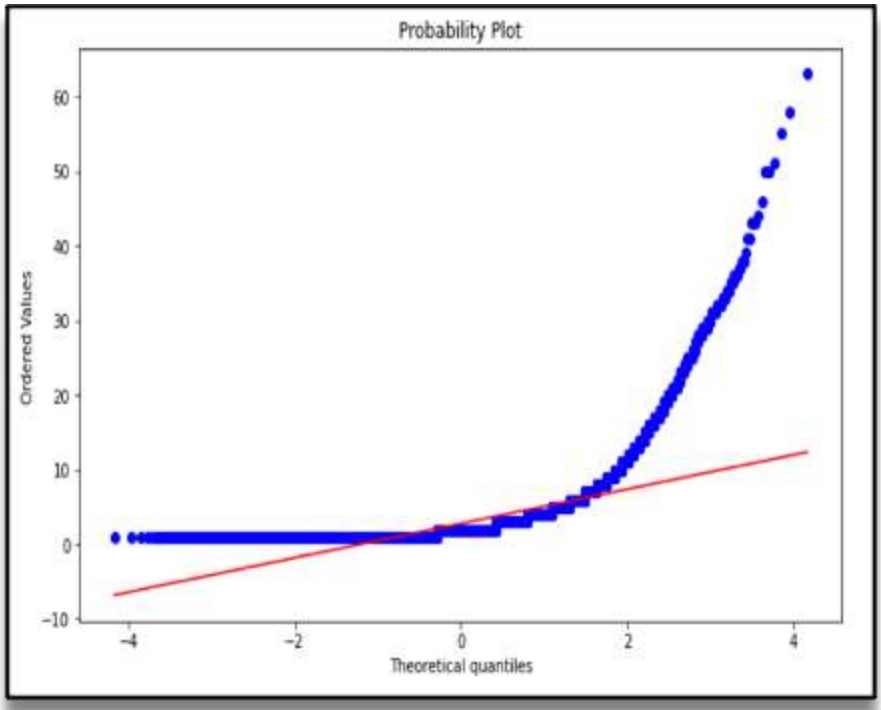


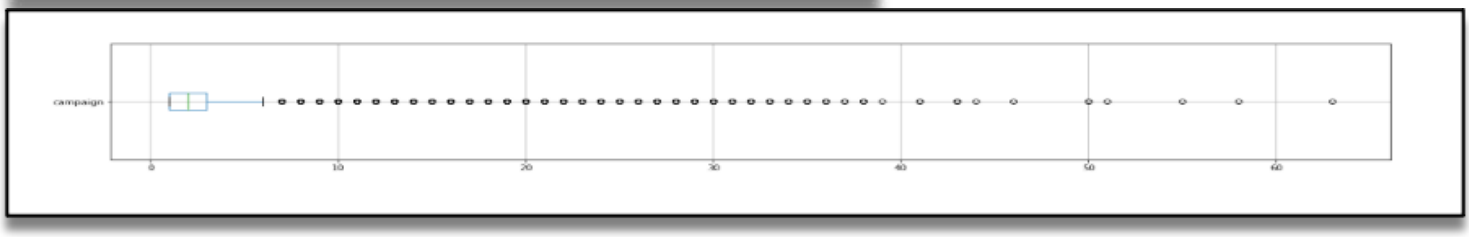
"Duration" Q-Q Plots and Box Plot



This 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 should be discarded if the intention is to have a realistic predictive model.In this case I will not remove it we have very low 0. However, for realistic model we will need to place it to our depended and independent variables.

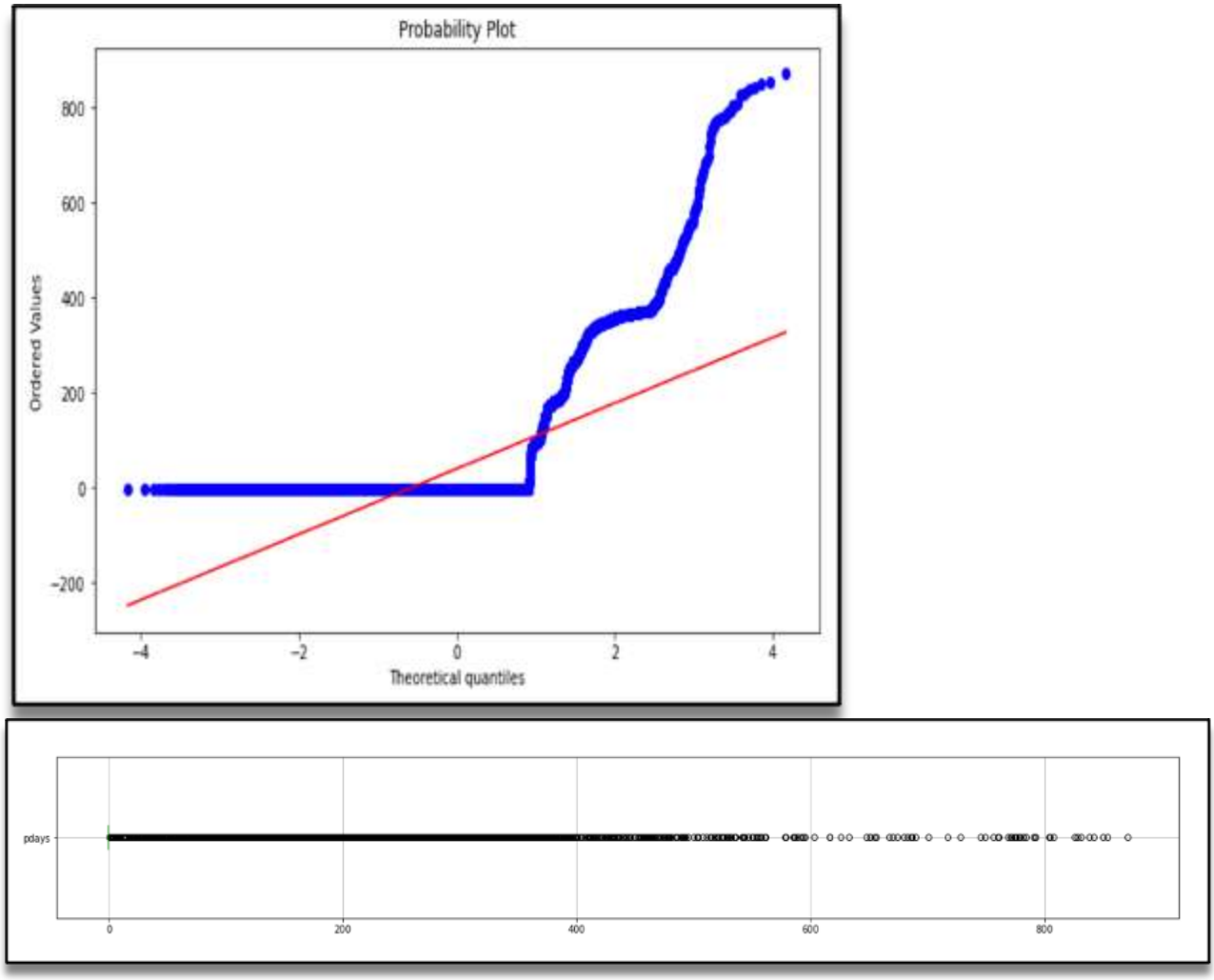
"Campaign" Q-Q Plots and Box Plot.





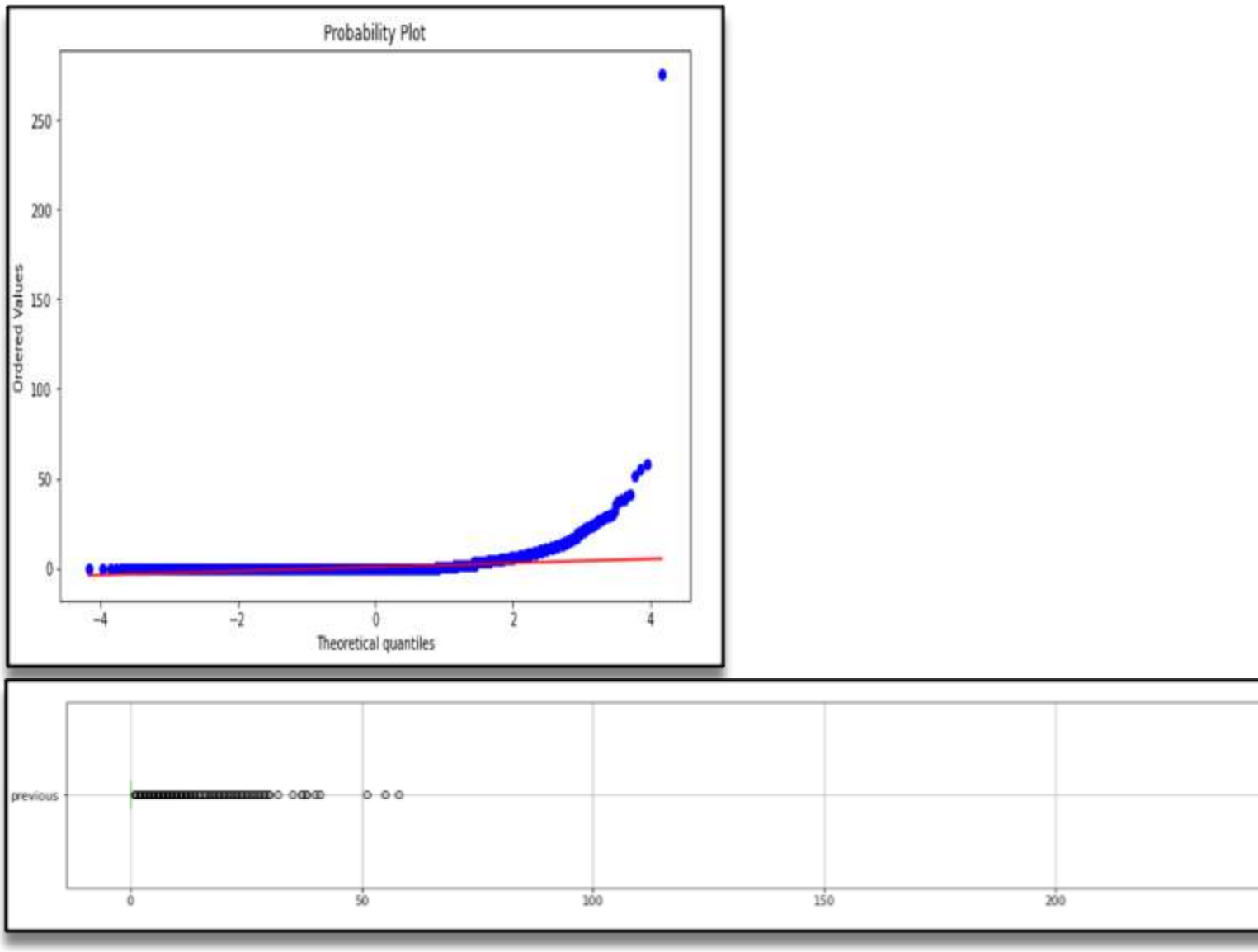
I don’t see any outliers on this feature so we can use it without any preprocessing.

"Pdays" Q-Q Plots and Box Plot.

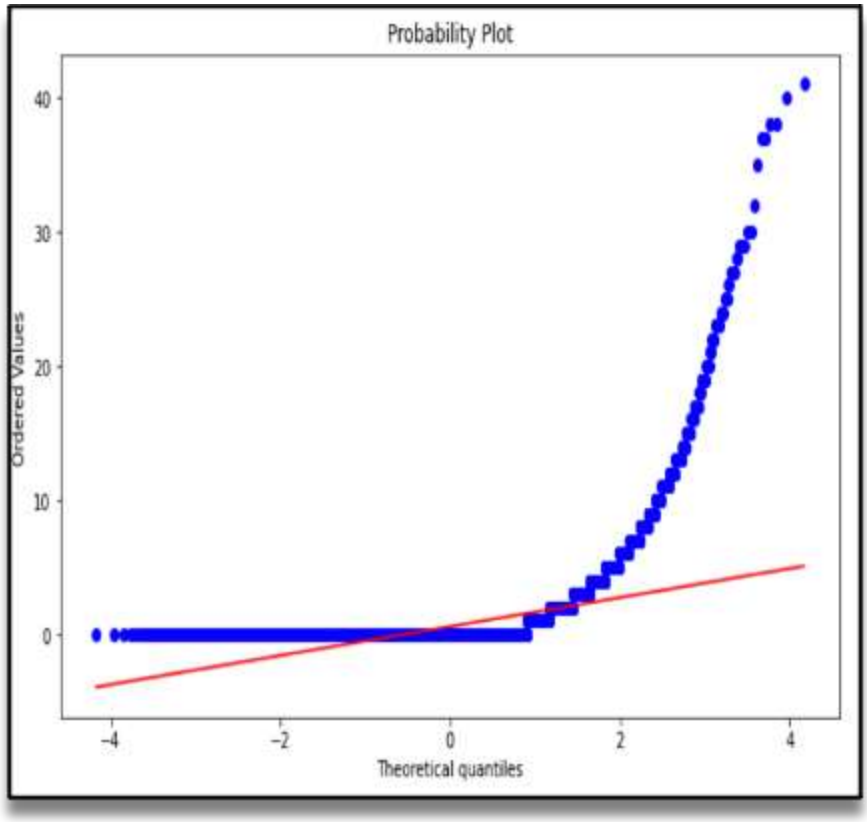


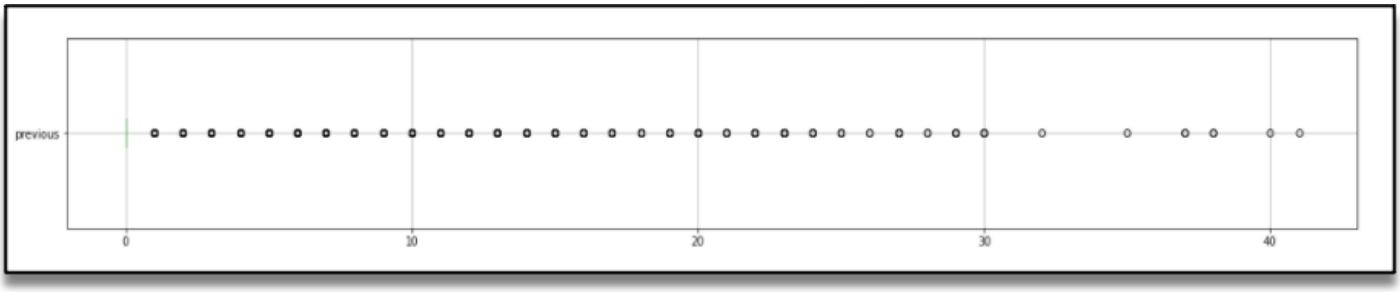
Number of days that passed by after the client was last contacted from a previous campaign (-1 means client was not previously contacted). We have to treat feature by using label encoding, because have -1 in 36940 values to mean client was not previously contacted.

"Previous" Q-Q Plots and Box Plot



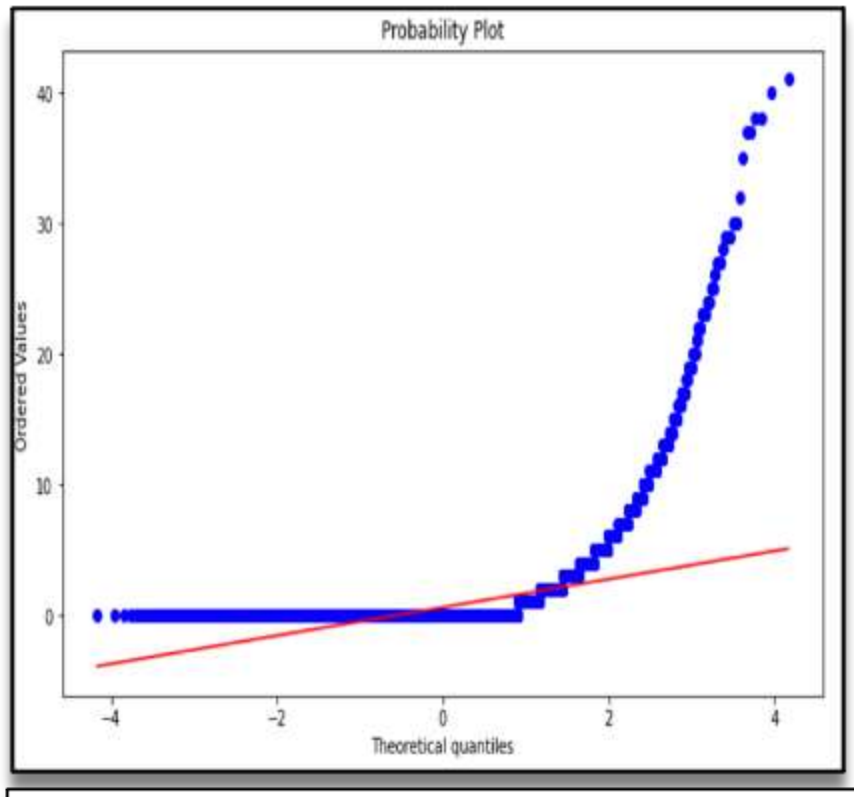
Number of contacts performed before this campaign and for Particular client. Here we can see some outliers. We will clean them all.

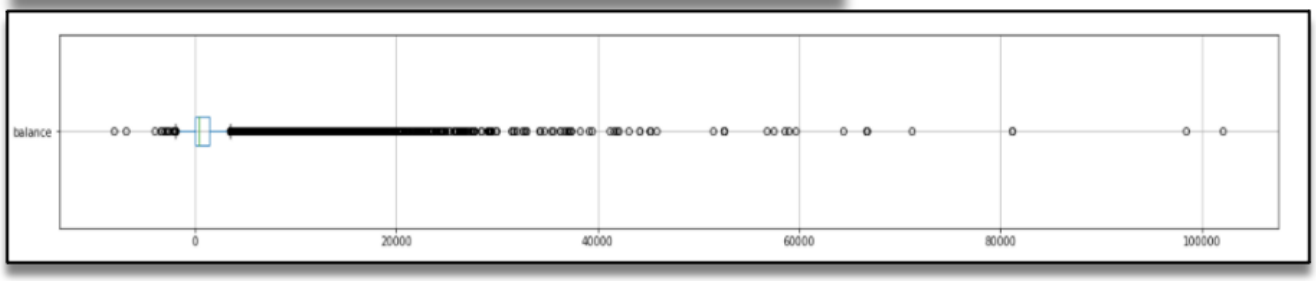




It looks perfect now.

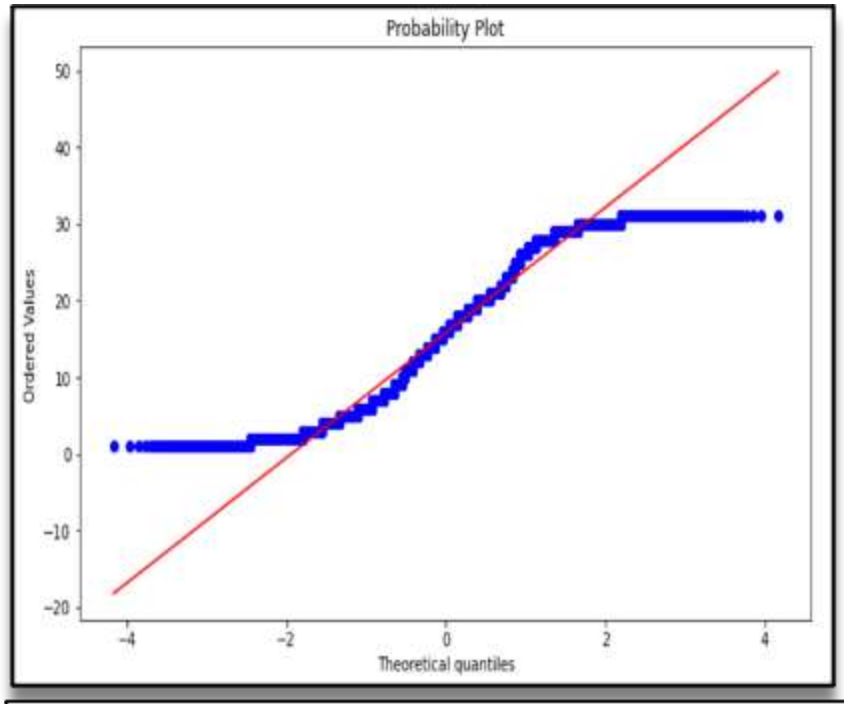
"Balance" Q-Q Plots and Box Plot.

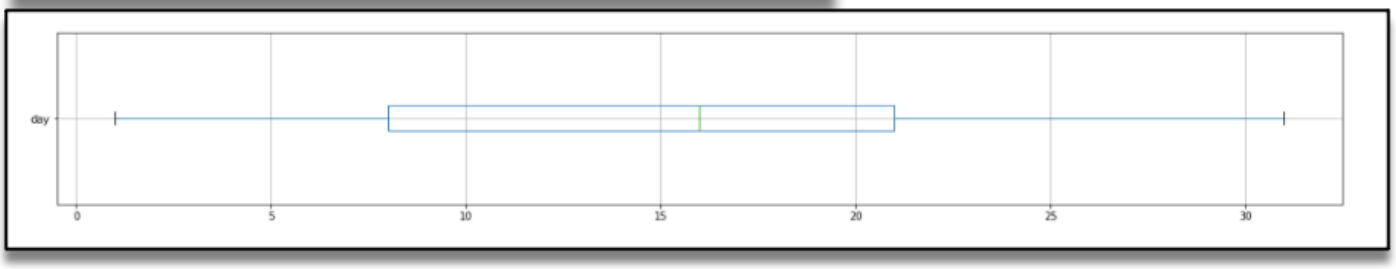




All is clear he. We can proceed it without any changes.

"Day" Q-Q Plots and Box Plot.

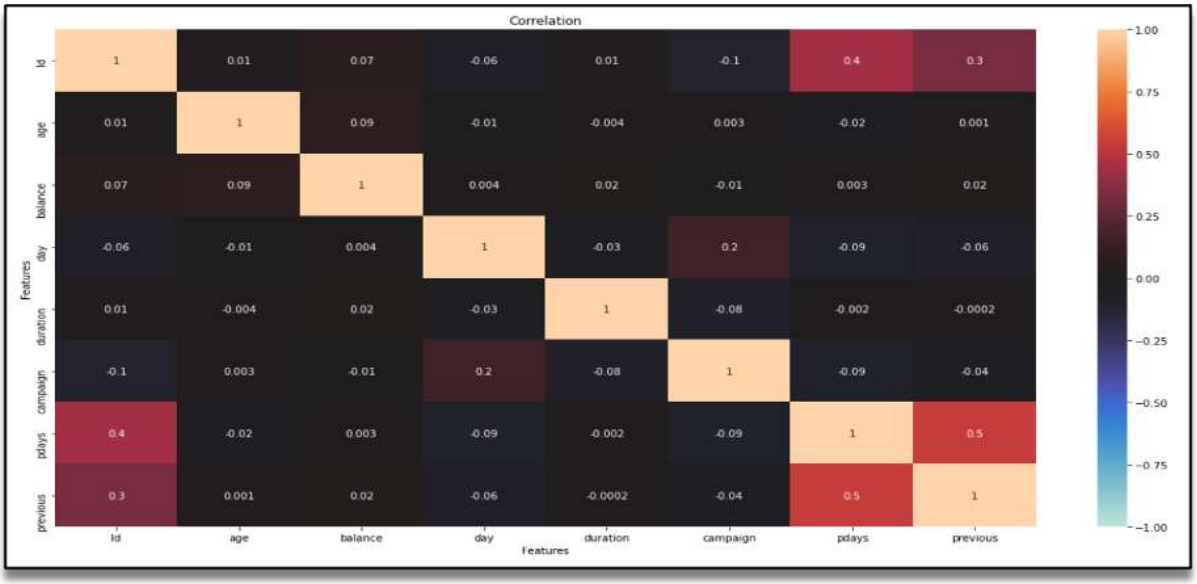




All is clear he. We can proceed it without any changes.

Correlation.

Correlation shows the relationship between variables in the dataset



Data Preprocessing

When building a machine learning model, it is important to preprocess the data to have an efficient model.
We will need to change our 'pdays' to categorical data.
ML models are require all input and output values should to be numerical. So if our dataset have categorical data, we must have to encode it into the numbers before fit and evaluate a model. There are several methods available.Here I have used One-hot Encoding
Another data preprocessing method is to rescale our numerical columns; this helps to normalize our data within a particular range. Sklearn preprocessing StandardScaler() was used here.

	duration	balance	campaign	day	age	previous
0	-0.416121	-0.437933	-0.56945	-1.298959	0.288558	-0.307445
1	-0.707438	-0.446799	-0.56945	-1.298959	-0.747420	-0.307445
2	-0.645290	0.047080	-0.56945	-1.298959	0.571098	-0.307445
3	-0.233563	-0.447127	-0.56945	-1.298959	-0.747420	-0.307445
4	-0.462732	-0.371601	-0.56945	-1.298959	-0.559061	-0.307445
5	-0.159763	-0.300671	-0.56945	-1.298959	-1.218320	-0.307445
6	-0.140341	-0.358794	-0.56945	-1.298959	0.006019	-0.307445
7	-0.470500	-0.319389	-0.56945	-1.298959	-1.124140	-0.307445
8	1.005504	-0.445485	-0.56945	-1.298959	1.136177	-0.307445
9	-0.726859	-0.424141	-0.56945	-1.298959	1.607076	-0.307445

Output of data set after do the scaling.

Next, we will combine our tables. Frame with numerical columns which we scaled and normalize, and our categorical frame without original numerical data.

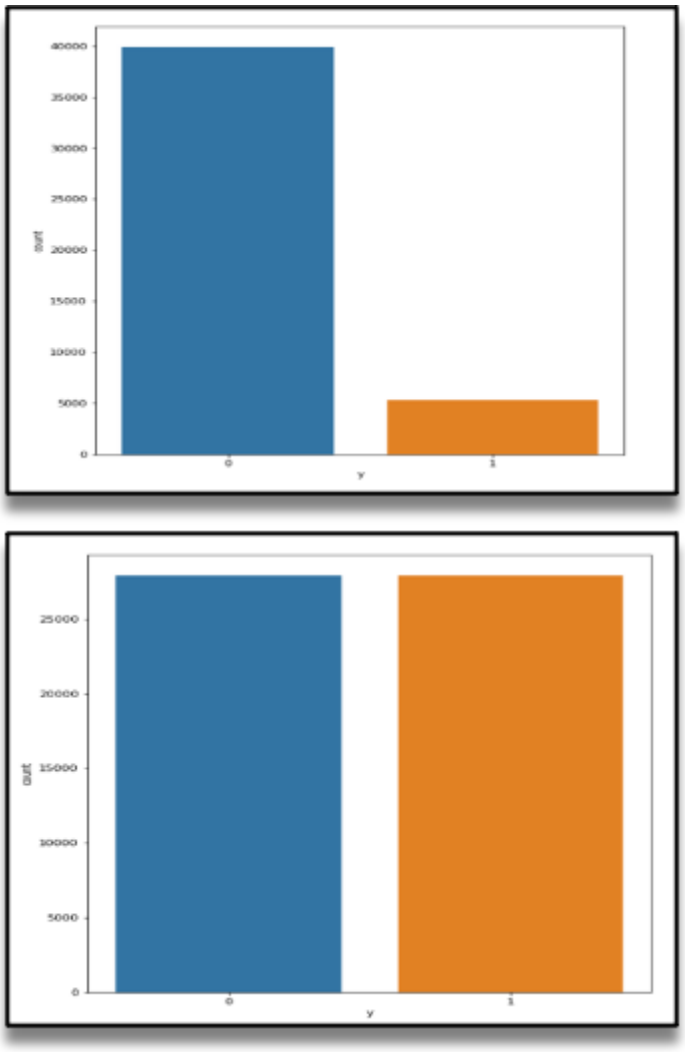
	duration	balance	campaign	day	age	previous	Id	y	job_admin
0	-0.416121	-0.437933	-0.569450	-1.298959	0.288558	-0.307445	1002	0	0
1	-0.707438	-0.446799	-0.569450	-1.298959	-0.747420	-0.307445	1003	0	0
2	-0.645290	0.047080	-0.569450	-1.298959	0.571098	-0.307445	1004	0	0
3	-0.233563	-0.447127	-0.569450	-1.298959	-0.747420	-0.307445	1005	0	0
4	-0.462732	-0.371601	-0.569450	-1.298959	-0.559061	-0.307445	1006	0	0
...	...	...	...	...	...	...	...	...	...
45187	2.792246	-0.176545	0.076029	0.143069	0.947817	-0.307445	46207	1	0
45188	0.768566	0.120308	-0.246711	0.143069	2.831415	-0.307445	46208	1	0
45189	3.374879	1.429217	0.721507	0.143069	2.925595	1.308769	46209	1	0
45190	0.970546	-0.228100	0.398768	0.143069	1.512897	-0.307445	46210	0	0
45191	0.399565	0.528151	-0.246711	0.143069	-0.370701	5.618675	46211	0	0

45192 rows × 610 columns

To proceed in building our prediction model, we have to specify our dependent and independent variablels. Here we can place 'duration' for more realistic model.

By using below codes i have divide the data set into 30% for testing and 70% for training by using train*test*split from sklearn.model_selection. It is reasonable to always split the dataset into train and test set when building a machine learning model because it helps us to evaluate the performance of the model.

As you rememeber our data a bit imbalanced. This can affect our prediction. I will do oversampling here.



Its applied-on training set.
Now we a finally ready to do modeling and prediction. It always very important to preprocess data perfectly before jump to next step, we can see perfect result of our work

Machine Learning Models and Predictions

I will first compare the model performance of the following 3 machine learning models using default hyperparameters:

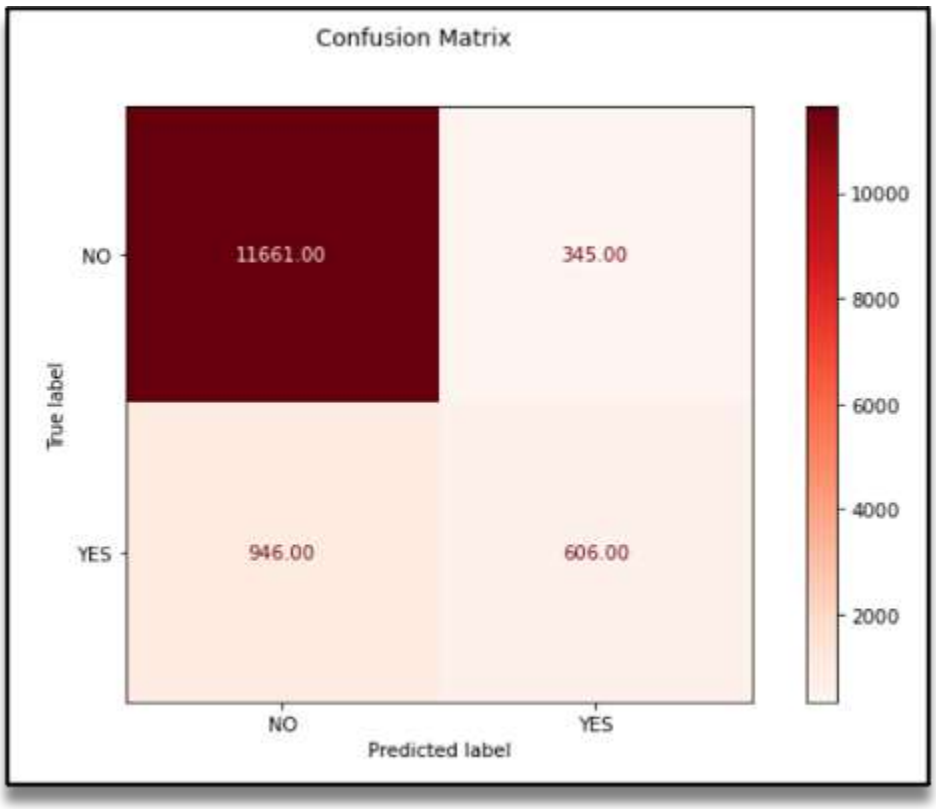
• Logistic Regression
• Decision Tree
• K Nearest Neighbors (KNN)

First, we will load libraries which we will use for ML and plots with reports

Logistic Regression

Logistic regression is a traditional machine learning model that fits a linear decision boundary between the positive and negative samples. Logsitic regression uses a line (Sigmoid function) in to predict if the dependent variable is true or false based on the independent variables. One advantage of logistic regression is the model is interpretable — we know which features are important for predicting positive or negative. Take note that the modeling is sensitive to the scaling of the features, so that is why we scaled the features above. We can fit logistic regression using the following code from scikit-learn

	precision	recall	f1-score	support
0	0.92	0.97	0.95	12006
1	0.64	0.39	0.48	1552
accuracy			0.90	13558
macro avg	0.78	0.68	0.72	13558
weighted avg	0.89	0.90	0.89	13558

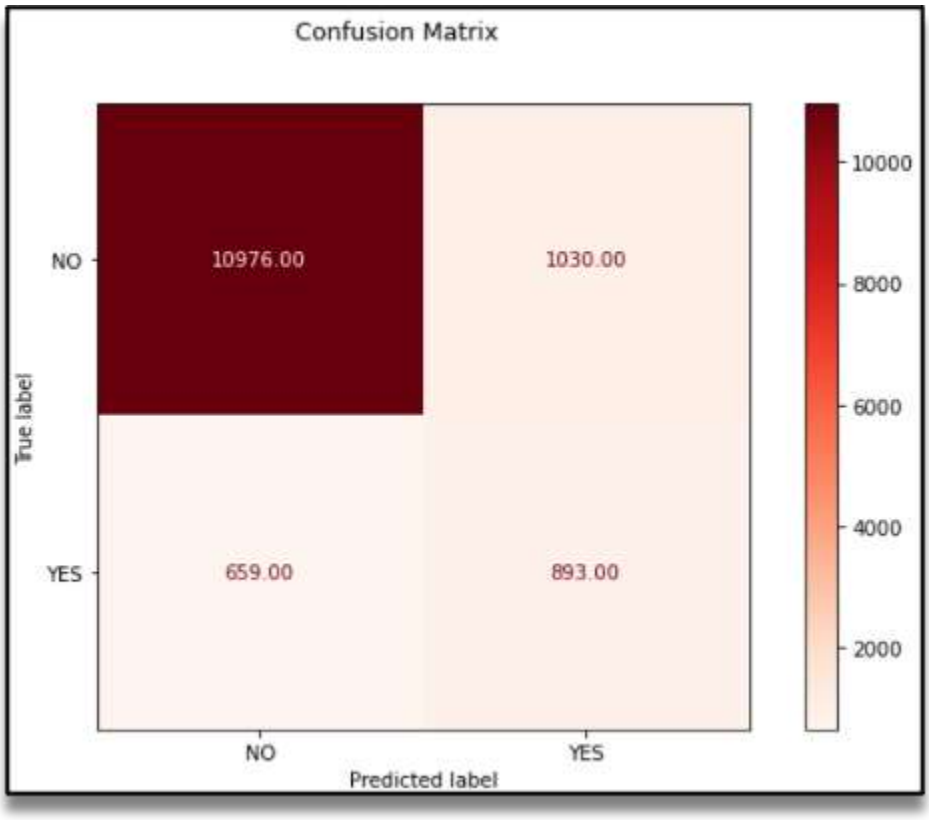


As you can see our accuracy is 0.90. From above code output we can see the overall prediction accuracy of the model. But we can't evaluate the model by looking overall prediction accuracy only. So have to do the study with comparing to the classification report also.

Decision Tree

This machine learning models is tree-based methods. The simplest tree-based method is known as a decision tree. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules gotten from training data. In Decision Trees, for predicting a class label for a record we start from the root of the tree. One advantage of tree-based methods is that they have no assumptions about the structure of the data and are able to pick up non-linear effects if given sufficient tree depth. We can fit decision trees using the following code.

	precision	recall	f1-score	support
0	0.94	0.91	0.93	12006
1	0.46	0.58	0.51	1552
accuracy			0.88	13558
macro avg	0.70	0.74	0.72	13558
weighted avg	0.89	0.88	0.88	13558

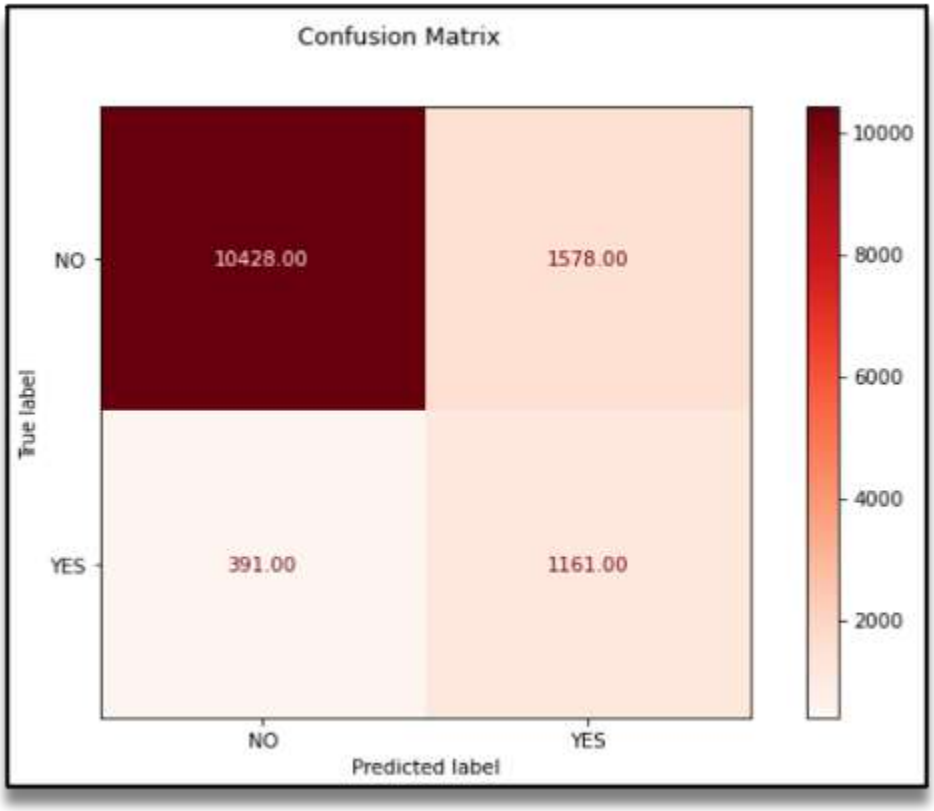


Our accuracy with this model 0.88. Let’s evaluate the model by looking overall prediction accuracy only. So have to do the study with comparing to the classification report as well.

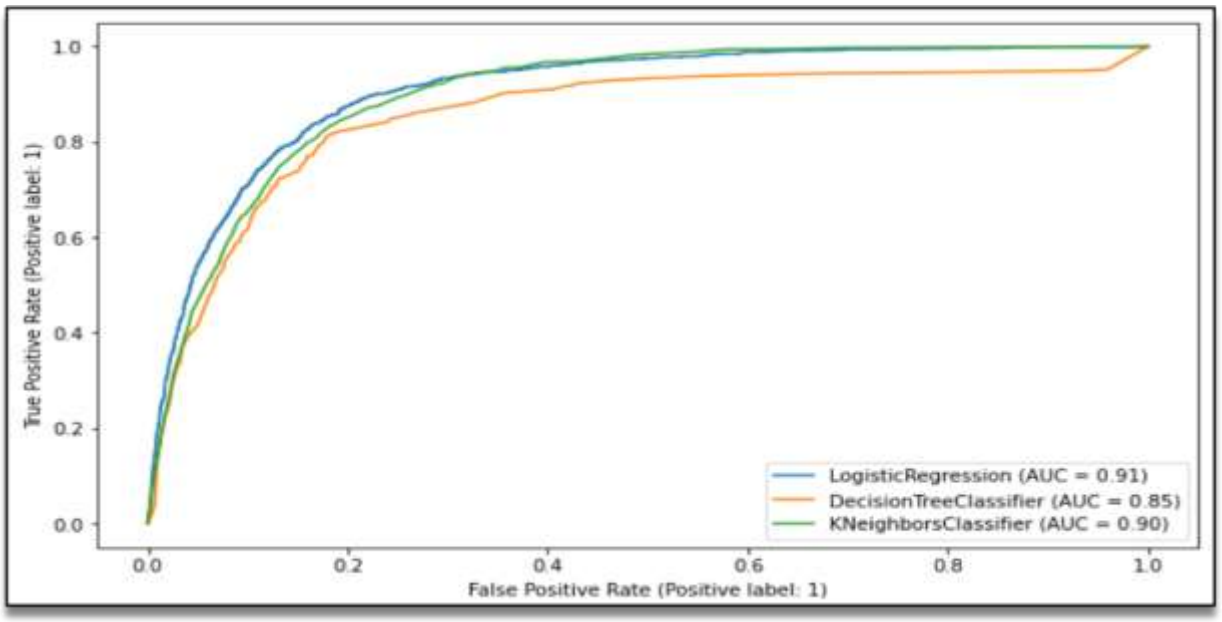
K Nearest Neighbors (KNN)

KNN is one the simplest machine learning models. KNN looks at the k closest datapoints and probability sample that has positive labels. This model is very easy to understand, versatile, and you don’t need an assumption for the data structure. KNN is also good for multivariate analysis. A caveat with this algorithm is being sensitivity to K and takes a long time to evaluate if the number of trained samples is large. We can fit KNN using the following code from scikit-learn.

	precision	recall	f1-score	support
0	0.96	0.87	0.91	12006
1	0.42	0.75	0.54	1552
accuracy			0.85	13558
macro avg	0.69	0.81	0.73	13558
weighted avg	0.90	0.85	0.87	13558



Our accuracy with this model 0.85. A bit lower than previous one. Let’s evaluate the model by looking overall prediction accuracy only. So have to do the study with comparing to the classification report as well.



AUC (Area under the ROC Curve): It provides an aggregate measure of performance across all possible classification.

Conclusion

Model	Accuracy	Specificity	Sensitivity	Negative predictive value	Precision
Logistic Regression	0.90	0.92	0.64	0.97	0.39
Decision Tree	0.88	0.94	0.46	0.91	0.58
K Nearest Neighbors	0.85	0.96	0.42	0.87	0.75

We were able to analyse bank marketing dataset, I built different models which help us to analyse the dataset properly, I classify the dataset according to the data preparing
description. Here I showed various plots for easy reading and understanding. Result of my
classification I present in the following table. I can see that obtain result of model mostly
are similar. But in my opinion the best one is Logistic regression model. It can predict the chances that customer will purchase a product across all possible classification with score 0.91