PREDICTION OF THE OUTCOME OF A PROMOTION CAMPAIGN IN THE BANKING SECTOR

With the use of Machine Learning

ASIF RANA
Email-asifranaar01@gmail.com
Linked in- https://www.linkedin.com/in/asif-rana-ar/

Abstract

A classifier model is being created from a dataset which recorded the promotion outcome conducted by a bank. There were 17 attributes in the dataset and a total of 45,211 records. Some feature engineering was required before starting the training phase of building the model. Two machine learning classifier algorithms were selected to compare and find out the model which was able to perform better in the training and testing phase. The Random Forest classifier was able to produce the best result with an accuracy of 86.3% and precision of 86.9%.

Introduction

Machine learning can be used to predict a classification problem statement. In this task a machine leaning model has been developed to predict the outcome of a promotional campaign. The dataset is a record of a promotional campaign conducted by the Portuguese Bank. This dataset was collected from the UCI repository. Initially, the raw dataset had a total of 45,211 records and 17 attributes including the classifier attribute.

In the first stage, the dataset was first imported into a pandas data frame and some basic exploration on the dataset was carried out to get a high level overview of the dataset. In the initial stage of the exploration, it was discovered that the columns had a combination of categorical and numerical values. Moreover, the scale of the numeric columns was different which meant some form of standardization of the numeric values were required. Some machine learning models can not deal with categorical values; which means label encoding to the categorical attributes was required.

In the next stage, ten attributes were selected which could give us more understanding about the distribution of the attribute independently. A range of graphical tools were utilized together with some metric values to make the understanding much more succinct. Later, the relationship between the attributes were explored to identify the level of correlation between the attributes. If there were high correlation between the attributes, there is a risk of getting into the 'curse of dimensionality'. During the exploration stage, it was discovered that the classifier column-'y' is highly imbalanced. This was dealt with implementing a down sampling to balance the distribution of the categories of the classifier attribute.

After the data preparation stage, three suits of data were generated. Each dataset had different percentage of training and test dataset. Finally, we get into training the classifier model. Here, two classifier models are being created- Random Forest Classifier and Gaussian Naive Bayes Classifier. The parameters of the model were adjusted to fine tune it with the training dataset. In the end, the performance of the two classifier models are recorded and visualized.

Problem Formulation, Data Acquisition and Preparation

The model of predicting the outcome of promotion campaign for a banking sector was based on the dataset from UCI repository at https://archive.ics.uci.edu/ml/datasets/Bank+Marketing. From the collection of the dataset file named 'bank-full.csv' was used as the training and testing dataset for the machine learning model. The classifier column is the column named 'y' which stored the outcome of the promotion conducted by the bank. The values of the classifier column are either 'yes' or 'no'. There were other 16 attributes which were recorded in the dataset which were being used to build the classifier model.

The csv dataset was read into a pandas data frame and some insights about the dataset were gathered. In total the dataset had 45,211 records which meant there was a large dataset to start with. The next stage was the data preparation stage. In the dataset there were no null values but some of the attributes had 'unknown' recorded as values. The instance with 'unknown' as its value were kept as it is. Since 'unknown' could bring some valuable insights in the further analysis. Moreover, 'unknown' could also be representing a separate category similar to 'others'.

In the dataset the 'pdays' attribute had a dummy value of -1 to represent the clients who were not previously contacted. This dummy value of -1 was converted to a value of 0. Since the -1 might influence the result of further analysis. Further, there were a total of 7 attributes which had numeric values. Each attribute had different scale of measurement which meant there are needed to be standardized before further analysis could be carried out. In this case normalization of the numeric values was used as the choice of standardization. After dealing with the attributes with numeric values, now it was time to focus on the attributes with categorical values. There were 10 categorical attributes which required to be encoded into numeric values. Since the machine learning models build used in this task cannot deal with categorical values. The encoding was carried out after the data exploration stage so that the visualization carried out in task-2 can be interpreted easily.

Data Exploration

Exploring the Attributes

There were 17 attributed in the dataset including the classifier attributes. Out of these attributes 10 attributes were selected for visualization to get more insights about the dataset. The selection was made based on the criteria which will give us valuable insight on the distribution of the attributes.

i) **age-** Since the age of the clients is a continue value, a box plot was used to represent the distribution. Moreover, through box plot we will be able to comment on the skewness of the distribution and identify the presence of extreme outliers. The box plot was generated using the seaborn library. Analyzing the plot in figure 2.1.1, we can see that the distribution for the age of the client is highly positively skewed with many outliers with high standard deviation. A distribution plot was also generated to confirm this observation as shown in figure 2.1.2

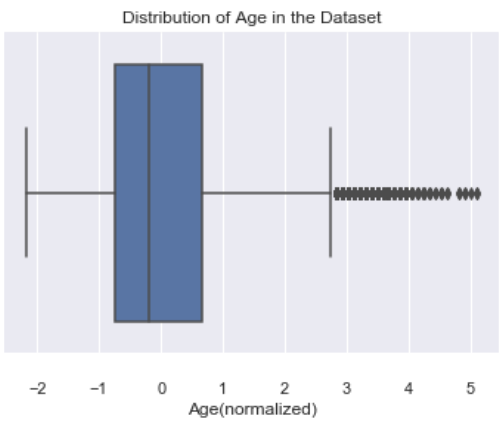

figure – 2.1.1

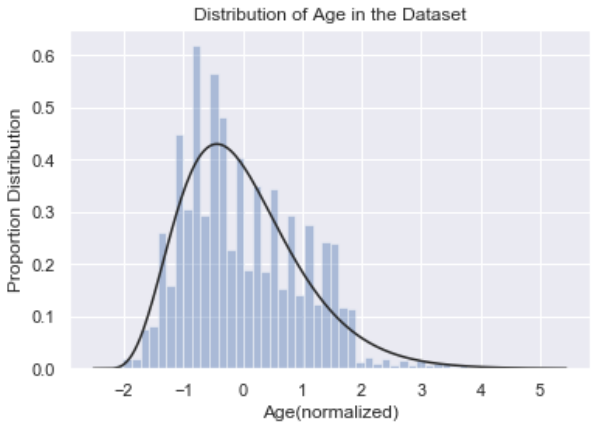

figure – 2.1.2

ii) **Job-** The job attribute is categorical with a total of 12 categories. To visualize the distribution a bar chart was being plot (figure – 2.2) with each category represented by a different color and the percentage share of each category at the top of each bar to make the comparisons between the job categories easier. In figure – 2.2 we can see that blue-collar job category had the highest percentage share with a value of 21.5%, the unknown category accounted for only 0.6% in the sample. This means that the categories of the job category were able to represent most of the job sector the clients of the bank were involved with.

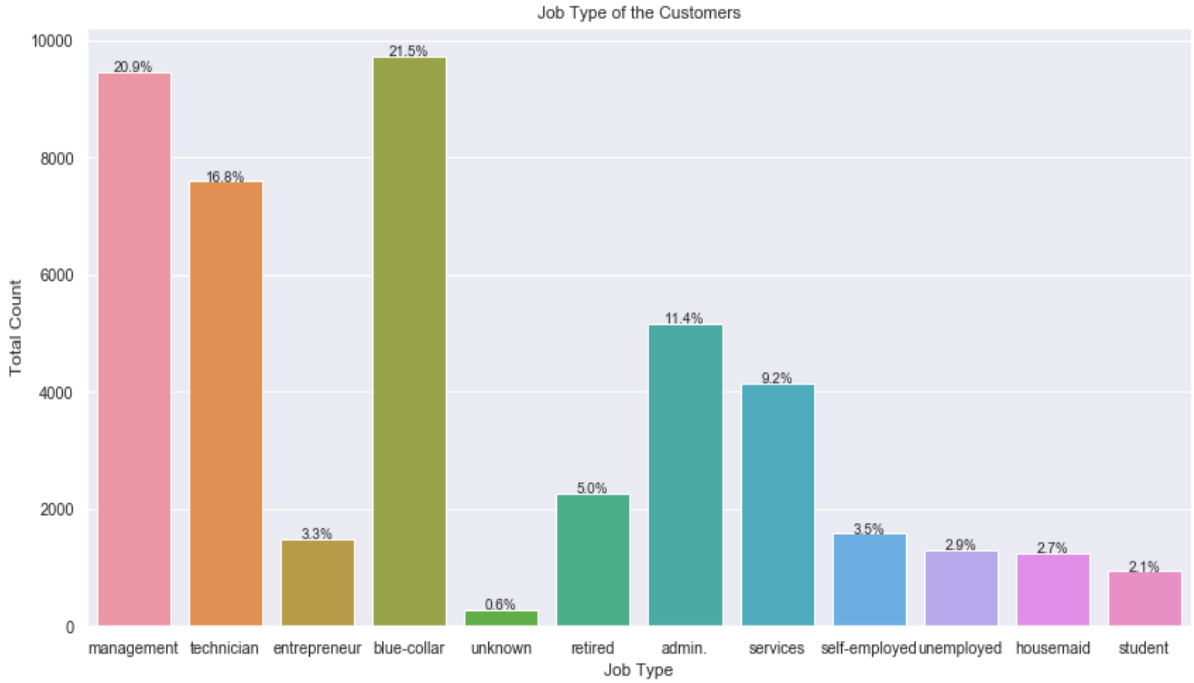

figure – 2.2

iii) **marital-** This is a categorical attribute with a total of 3 categories. To visualize the distribution a bar chart was used as shown in figure – 2.3.1 as well as a pie chart shown in figure- 2.3.2. Analyzing both of the graphs provided below we can see that most of the clients were married with a percentage share of 60.2%.

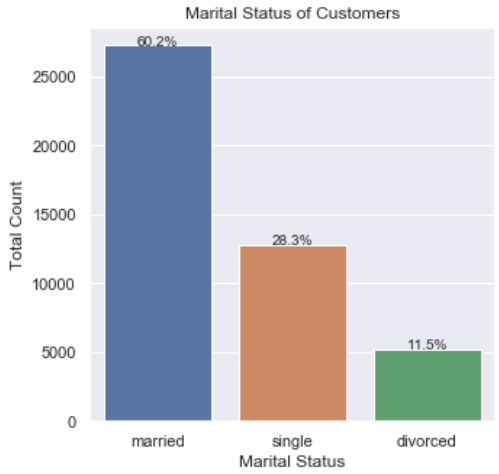

figure- 2.3.1

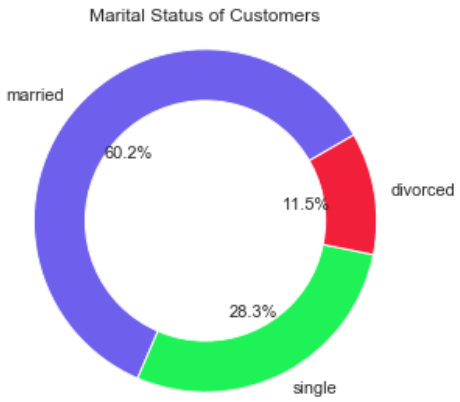

figure- 2.3.2

iv) **education-** This attribute has 4 categories. To visualize the distribution a bar chart was used as shown in figure – 2.4.1 as well as a pie chart shown in figure- 2.4.2. Both graphs show that most of the clients had an education till secondary. About 4.1% of the clients did not have their education level recorded.

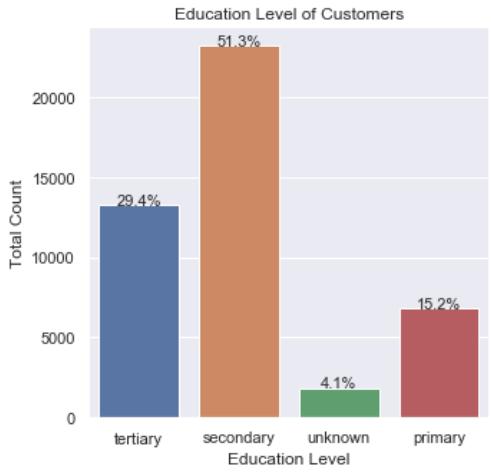

figure- 2.4.1

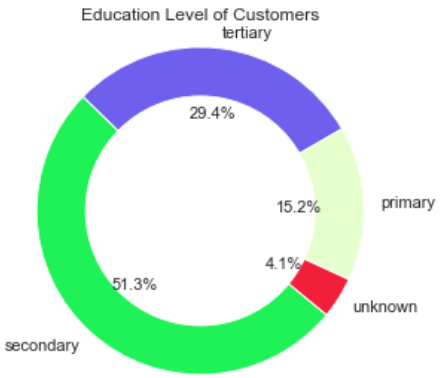

figure- 2.4.2

v) **default-** This attribute had the largest amount of disproportionate distribution as shown in figure 2.5. The bar chart shown below shows that only 1.8% of the clients has a previous default history meaning that in the promotion most of the clients were financially stable.

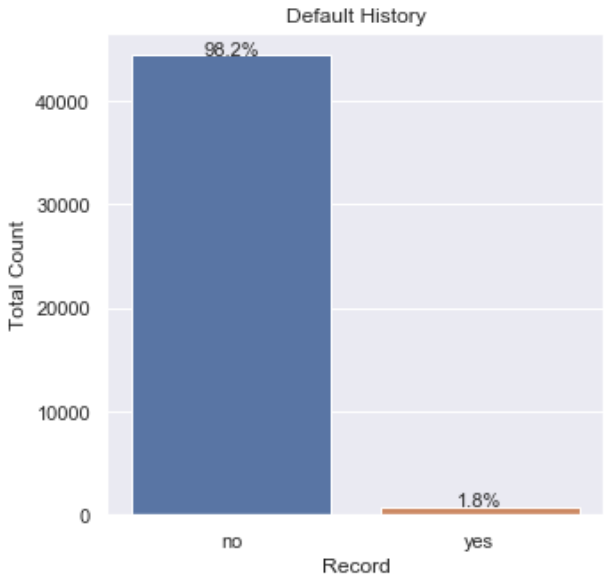

figure- 2.5

vi) **balance-** The balance of the clients of the bank who participated in the promotion was plotted using a distribution plot as shown in figure- 2.6. The plot shows that the distribution for balance is normally distributed.

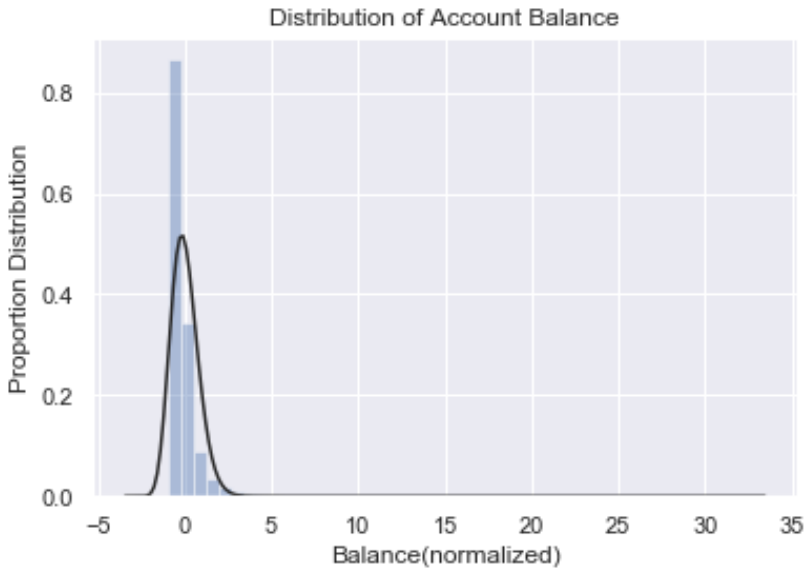

figure- 2.6

vii) **housing-** The distribution of the housing attribute was represented using a bar chart as shown in figure- 2.7. Analyzing the plot we can see that the distribution for housing is very close to even distribution between 'yes' (55.6%) and 'no'(44.4%).

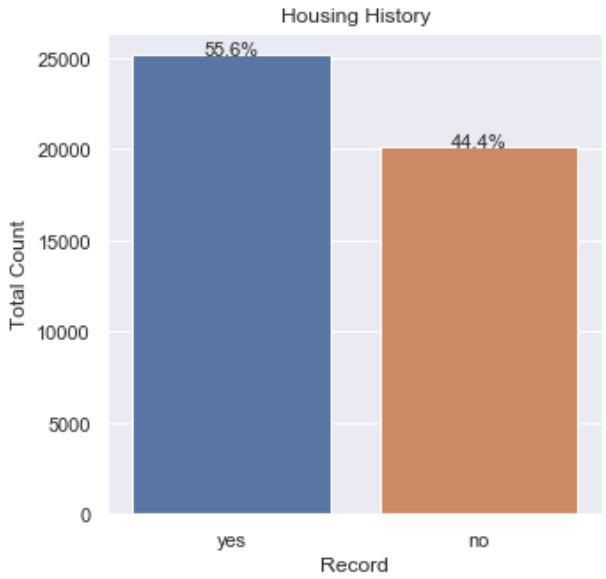

figure- 2.7

viii) **loan-** The loan attribute was visualized using a bar chart as shown in figure 2.8. The graph shows that the distribution is biased towards 'no'(84%) compared to 'yes' (16%).

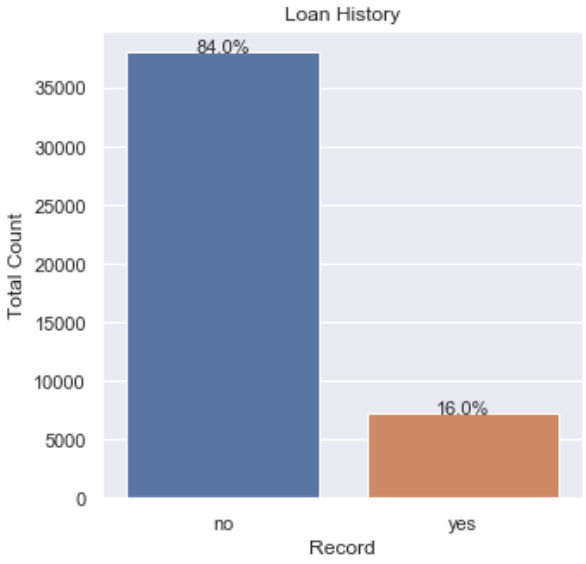

figure- 2.8

ix) **poutcome-** The poutcome represents the outcome of the previous promotion of the clients. The distribution is represented using a bar chart and a pie chart as shown in figure- 2.8.1 and figure-2.8.2. Analyzing both plots, we can see that the distribution is biased to the 'unknown' category. This finding gives us an indication that most of the clients in this

dataset could be involved in the promotion for the first time or that the record of the promotions was not properly recorded by the bank.

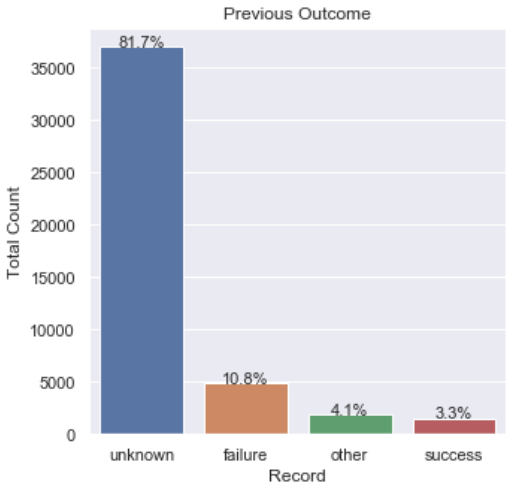

figure- 2.9.1

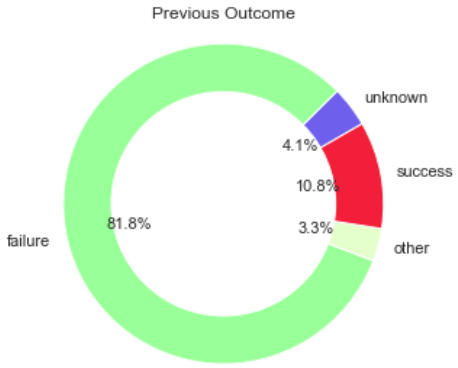

figure- b2.9.2

x) **y-** This is the classifier attribute. Its distribution is represented using a bar chart as shown in figure- 2.10. Analyzing the plot, it can be confirmed that the classifier is biased towards 'no' with a percentage share of 88.3% and 'yes' is only having a share of 11.7%. This mean if a classifier is built with this biased dataset the classifier model will suffer from overfitting. Due to overfitting the classifier might perform poorly in the testing phase. To overcome this concern, the dataset needs to go through either oversampling of the 'yes' records in the 'y' attribute or under sampling the 'no' records in the 'y' attribute.

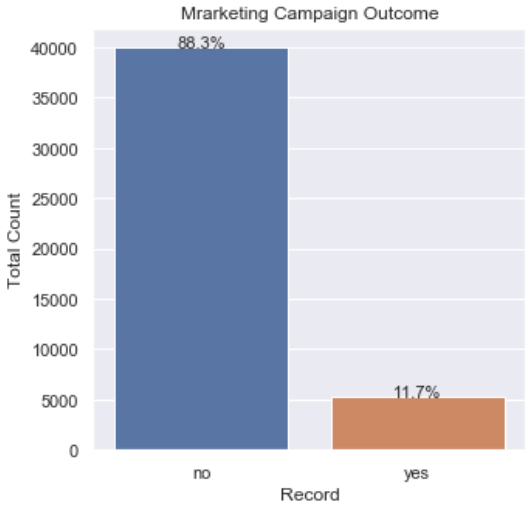

figure- 2.1

10 attributes were selected to compare the relationship between each other. The comparison was carried out using a heatmap. Before the graph was being plot the columns with the categorical values were needed to be label encoded into numerical values. The heat map generated is shown in figure- 3.

Analyzing the heatmap in figure- 3, it can be concluded that the classifier attribute 'y' is positively correlated with the 'balance' column and 'education'. Moreover, the 'y' column is negatively correlated with the 'housing' and 'performance' column. The relation between the 'y' and the columns- 'age', 'job', 'marital', 'default', 'loan' is less significant.

Further findings between the non-classifier columns are listed below-

Columns	**Positive Correlation with**		**Negative Correlation with**	
Strength	**Strong**	**Weak**	**Strong**	**Weak**
age	balance	y	marital	education, housing
job	education	marital		housing
marital	education	marital		Loan
education		balance, y		housing
default		loan		balance
balance		y		Loan, housing
housing				y, poutcome
Loan				y
performance				y

The table shown above gives us a short summary about the scale of the relationship between the attributes. Most of the correlation which existed between the attributes were weak. There were some attributes which showed the presence of slightly stringer correlation; but there was no attribute which were very highly correlated with one another.

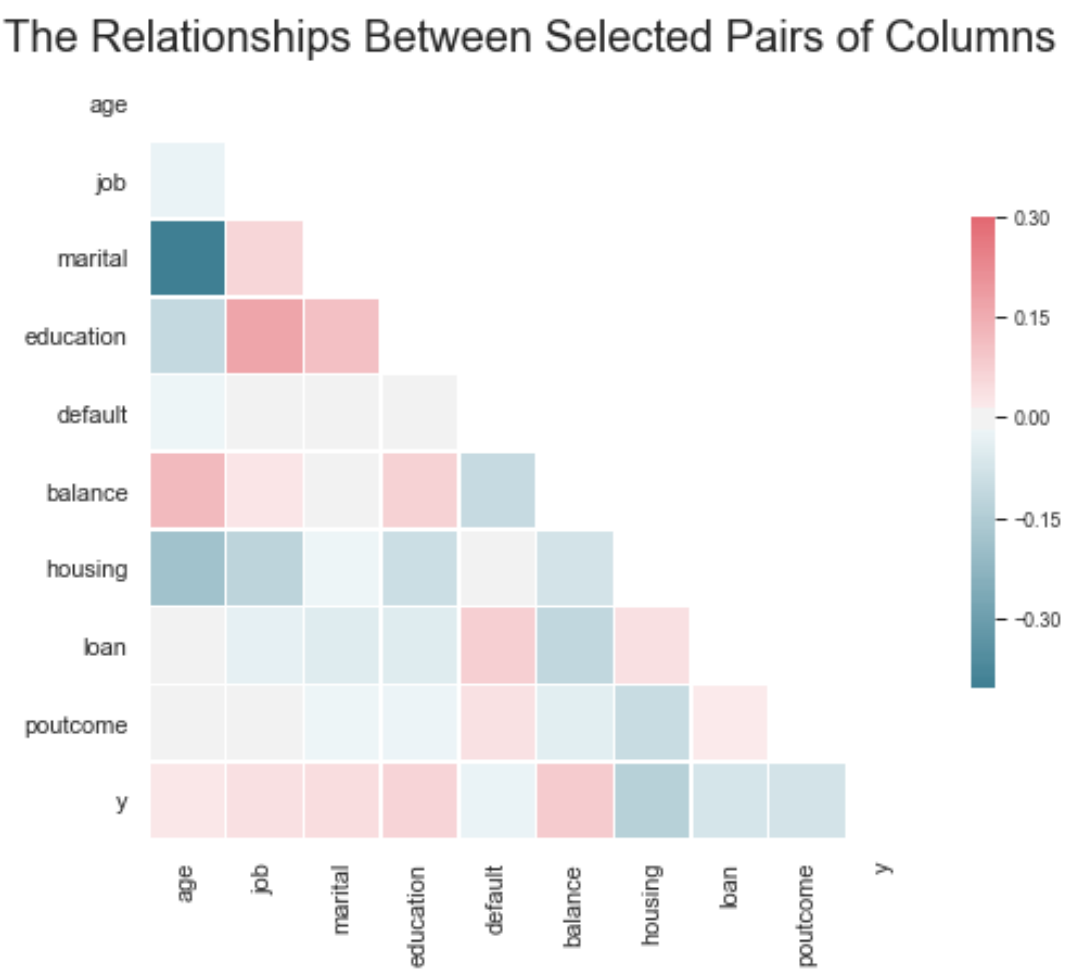

figure- 3

In the heatmap shown in figure- 3, we came to know that the balance of a client is highly correlated with the outcome of the promotion (y-column). A question arises, does clients with higher balance in their account is more likely to accept the promotional campaign? To get the answer of this question, the dataset was divided into two separate data set. The split was made based on the normalized value of the balance. If the balance was less than the median value of 0.30028, it was assigned to the lower balance group else it was assigned to the higher balance group.

Later, the count of 'yes' was recorded for both groups and the result was plotted into a pie chart as shown in figure- 4. As shown in the figure- 4, we can see that 61.3% of the clients who has responded to the promotion had a higher bank balance. Where only 38.7% of the clients who had a lower bank balance has accepted the promotion.

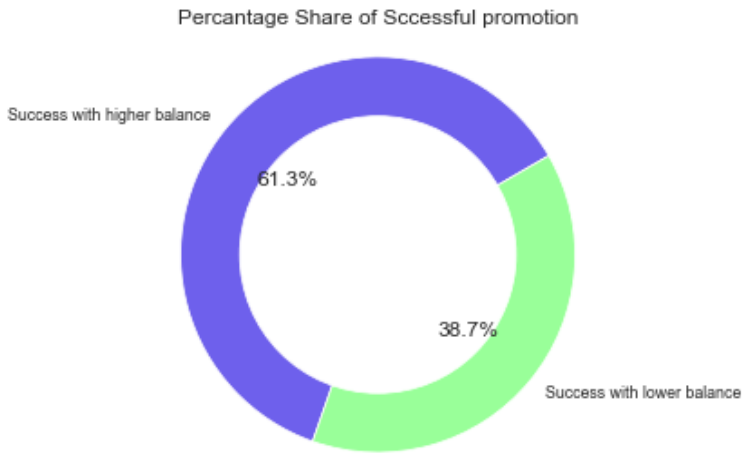

figure- 4

Data Modelling

During the data exploration stage, it was discovered that the classifier column ‘y’ has imbalanced distribution as shown in figure- 2.10. Building a classifier model with an imbalanced dataset might result into a biased machine learning model as it will try to overfit the model. To take care of this issue there were two viable option which are either apply under sampling to the dominant ‘no’ records or apply oversampling to the ‘yes’ records.

In this case under sampling of the ‘no’ records were carried out. The reason behind this reason is that there was a large imbalance between the two categorical values and in order to implement over sampling we needed to copy the ‘no’ records about 8 times to match the count of the ‘yes’ records. Copying records repeatedly multiple times might make our model less representable to the population. Moreover, the number of records with ‘yes’ as a value for the y-column is 5,289 which means if we apply under sampling to the ‘no’ records to match the count of that of the ‘yes’ we would still have enough data to create a representable classifier model.

After the dataset went through under sampling, the dataset was used to create 3 separate suites of samples.

- Suite1: 50% for training and 50% for testing
- Suite2: 60% for training and 40% for testing
- Suite3: 80% for training and 20% for testing

In order to ensure reproducibility of this splitting, the random_state was set to 42 while generating each of the suites. random_state is a parameter for the splitting function train_test_split in seaborn which controls the shuffling of the dataset before splitting it. Ensuring the reproducibility of the dataset was important while building and evaluating the machine learning models.

The two classification algorithms selected to build the machine learning model for predicting the result of a bank promotional campaign are Random Forest and Gaussian Naive Bayes algorithms. Both the models were made using the machine learning algorithms in 'scikit-learn' package.

The parameters of the Random Forest classifier model are –

(bootstrap= False, min_samples_split= 5, n_estimators=30)

Bootstrap was set to false because we had a large dataset (10,578 records) which means there was no requirement for bootstrapping. Since if bootstrap is being implementing, there is a high probability of selection for the same record multiple times and this might affect the performance of the model.

min_samples_split was set to 5. This parameter determines the minimum number of samples required to split an internal node. By default min_samples_split is set to 2, the reason of increasing this parameter to 5 is that we want to reduce the depth of the tree and make as less number of internal nodes as possible since if the depth of the tree is high there is a possibility of overfitting the model.

n_estimators determines the minimum number of trees in the random forest. By default, it is set to 100 which was then reduced to 30. Since we want to limit the number of trees in the random forest to prevent the model from overfitting.

The performance metrics for the Random forest Classifier-

Dataset	Accuracy	Precision	Recall	F1 Score
Suite1	0.853	0.864	0.835	0.849
Suite2	0.856	0.872	0.83	0.851
Suite3	0.863	0.869	0.834	0.851

Confusion Metrix-

Suite1	TP	TN
TP	2146	438
TN	397	2308

Suite2	TP	TN
TP	1699	36
TN	316	1856

Suite3	TP	TN
TP	834	190
TN	142	950

In case of the parameters of the Gaussian Naive Bayes model the two default parameters were kept unchanged. Since the parameters were able to adapt to the dataset provided and were able to produce optimal result without further tweaking required.

The performance metrics for the Gaussian Naive Bayes Classifier-

Dataset	Accuracy	Precision	Recall	F1 Score
Suite1	0.765	0.799	0.756	0.777
Suite2	0.766	0.804	0.756	0.779
Suite3	0.76	0.806	0.749	0.776

Confusion Metrix-

Suite1	TP	TN
TP	1901	683
TN	569	2136

Suite2	TP	TN
TP	1480	580
TN	418	1754

Suite3	TP	TN
TP	707	317
TN	192	900

The performance metrics for both of the models were recorded in a csv file which was then imported into a pandas data frame for visualization as shown in figure- 5 and figure- 6. After analyzing both the graphs, we could see that the performance of the Random Forest classifier (figure-6) is better than the Gaussian Classifier (figure-5). This is because for the Random Forest classifier, all the performance metrics are higher than the Gaussian classifier. Which means Random Forest was able to create a better model in the training phase.

After comparing the performance graphs for both of the classifier models, it can be concluded that the Random Forest classifier should be used as the classifier model for predicting the result of a promotional campaign conducted by a bank. Random forest classifier model with data suite-3 was able to generate a high accuracy of 0.863, precision- 0.869, recall-0.834 and F1 score- 0.851

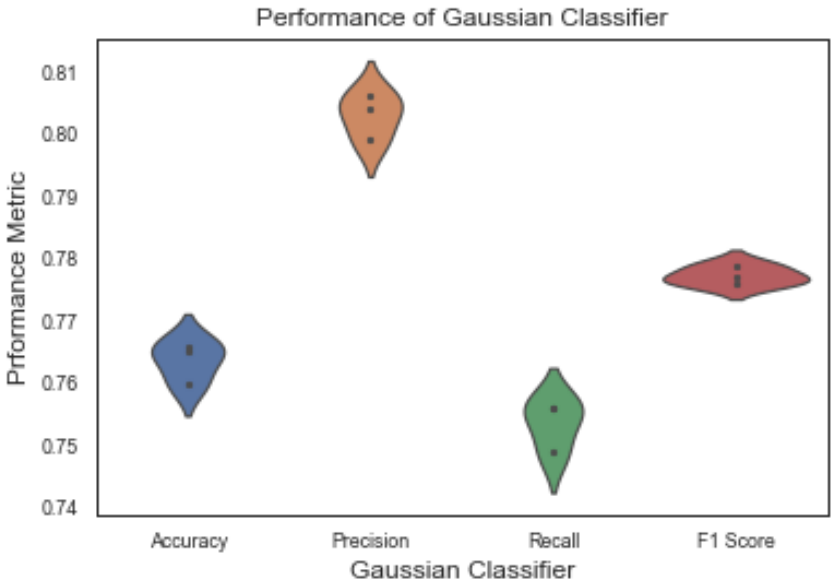

figure- 5

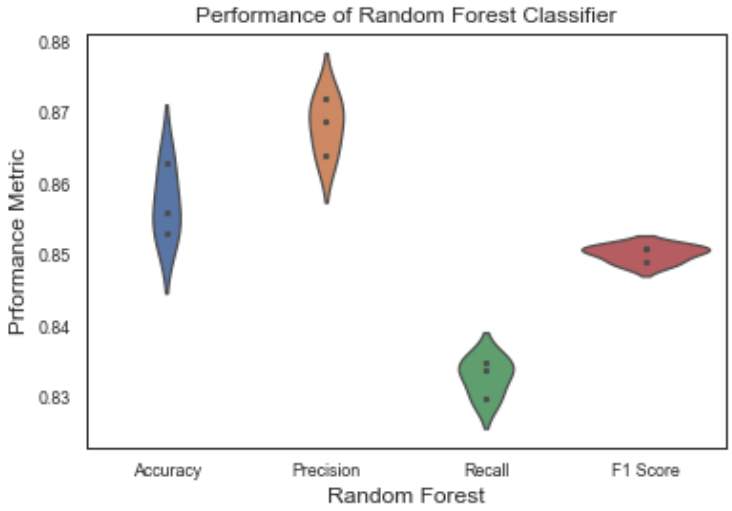

figure- 6

Discussion and Conclusions

An opensource data repository was selected to create a machine learning classification model. The dataset referred to the marketing campaign conducted by a Portuguese bank. In total there were 45,211 records and 17 attributes including the classifier attribute. The dataset comprises of numeric and categorical values, which required some form of feature engineering such as standardization and label encoding. Moreover, the dataset was highly imbalanced with a 9:1 ratio between the classifier values. This means, to prevent the machine learning model from overfitting it is required to make the dataset

balanced. The dataset was balanced through underfitting the ‘no’ category records to match with the ‘yes’ category. After all these preprocessing stages, there were a total of 10,578 to build the classifying model.

Two machine learning models were created- Random Forest Classifier, Gaussian Classifier. Some fine tuning of the parameters of the classifiers were made to fit the model to our dataset. This was done so that we can achieve optimal performance from the classifier. Out of the two models, the model which was implemented with Random Forest had the best performance between the two. The model was tested with several suits of dataset each with a different percentage split between the training and testing dataset. Out of the three suits the 80% training and 20% testing combination of datasets resulted with an accuracy of 86.3% and precision of 86.9%.

The performance of this classifier model shows us the real-life implication of machine learning. If this kind of machine learning model is being deployed by the bank for future marketing campaigns, it will allow them to target their promotional campaigns to the clients who are more likely to respond to the campaign. Thus, saving time and money for the banks. During the phase of development of the machine learning model a lot of time and thoughts was invested to understand the distribution of each attribute and the relation between the attributes. The findings from this exploration was used to create a classifier model which will have a better performance.