MACHINE LEARNING ENGINEER

NANO DEGREE

BANK MARKETING CAMPAIGN

By

K. Guru Tejaswini

Proposal

Domain Back Ground:

Usually, banks hold a data about their customers and this sort of data is very useful to judge the bank like working of bank, and also about the customers of bank. In order to hold that huge amount of data to explore we use the machine learning algorithms to predict the required results about the data**.** In banks, usually, the selected customers are contacted via: personal contact, telephone, cellular, mail, email and any other contacts to advertise the new product/service or propose an offer. This type of marketing is known as direct marketing.

Historically, the name and identification of the term direct marketing was first suggested in 1967 by Lester Wunderman, which is why he is considered as the father of direct marketing.

In addition, some of the banks and financial-services companies may depend only on strategy of mass marketing for promoting a new service or product to their customers. In this strategy, a single communication message is broadcasted to all customers through media such as television, radio or advertising firm, etc... In this approach, companies do not set up a direct relationship to their customers for new-product offers. In fact, many of the customers are not interested or just don't respond to this kind of sales promotion. Accordingly, banks, financial-services companies and other companies are shifting away from mass marketing strategy because of its ineffectiveness, and they are now targeting most of their customers by direct marketing for specific product and service offers.

Analysis of campaign includes a lot of things to be classified and it can better be solved by using machine learning techniques one can refer the sample example for campaign analysis using machine learning is https://www.aarki.com/blog/usingmachine-learning-to-predict-campaign-performance by this link .

The main benefit of solving this kind of problem with the help of machine learning lie in the essence of machine learning which has various techniques and approaches to deal with a complex in an easy way by classifying the things, tuning the parameters, performing evaluation metrics and many more So,

I believe that the ml has power to deal with this problem. The similar which was solved using machine learning was as follows: https://nycdatascience.com/blog/studentworks/machine-learning/machine-learning-retail-bankmarketing-data
PROBLEM STATEMENT:

All bank marketing campaigns are dependent on customers data. The size of these data sources make it impossible for a human analyst to extract interesting information that helps in the decision-making process. Data mining models help us achieve this. The purpose is to increase the campaign effectiveness by identifying significant characteristics that affect the success (the deposit subscribed by the client) based on a handful of algorithms that we will test (e.g. Logistic Regression, Gaussian Naive Bayes, Decision Trees and others). The experiments will demonstrate the performance of models by statistical metrics like accuracy, sensitivity, precision, recall,etc

METRICS FOR EVALUATION:

The evaluation metrics proposed are appropriate given the context of the data, the problem statement, and the intended solution. The performance of each classification model is evaluated using three statistical measures; classification accuracy, sensitivity and specificity. It is using true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

The percentage of Correct/Incorrect classification is the difference between the actual and predicted values of variables. True Positive (TP) is the number of correct predictions that an instance is true, or in other words; it is occurring when the positive prediction of the classifier coincided with a positive prediction of target attribute. True Negative (TN) is presenting a number of correct predictions that an instance is false, (i.e.) it occurs when both the classifier, and the target attribute suggests the absence of a positive prediction. The False Positive (FP) is the number of incorrect predictions that an instance is true. Finally, False Negative (FN) is the number of incorrect predictions that an instance is false.

The evaluation metrics was used to determine the performance of a model. An important aspects of **evaluation metrics** is their capability to discriminate among model results. ... But, creating and selecting a model which gives high accuracy on out of sample data.

The following I considered to be important suitable for my solution:

Classification accuracy is defined as the ratio of the number of correctly classified cases and is equal to the sum of TP and TN divided by the total number of cases (TN + FN + TP + FP).

Precision is defined as the number of true positives (TP) over the number of true positives plus total number of false positives.

Recall is defined as the number of true positives (TP) over the number of true positives plus the number of false negatives (FN)

Sensitivity refers to the rate of correctly classified positive and is equal to TP divided by the sum of TP and FN. Sensitivity may be referred as a True Positive Rate.

Specificity refers to the rate of correctly classified negative and is equal to the ratio of TN to the sum of TN and FP

ANALYSIS:

Data Exploration The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns are based on phone calls. Often, more than one contact to the same client was required, in order to check if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

I found that in my data set there are no missing values but there are unknown values during my analysis of data.

Input variable:

age (numeric) 2. job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unk n own') 3. marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed) 4. education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','profession al.course','university.degree','unknown') 5. default: has credit in default? (categorical: 'no','yes','unknown') 6. housing: has housing loan? (categorical: 'no','yes','unknown') 7. loan: has personal loan? (categorical: 'no','yes','unknown') 8. contact: contact communication type (categorical: 'cellular','telephone') 9. month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec') 10. day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri') 11. duration: last contact duration, in seconds (numeric). Important note: 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. 12. campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact) 13. pdays: number of days that passed by after the client was last contacted from a previous campaign

(numeric; 999 means client was not previously contacted) 14. previous: number of contacts performed before this campaign and for this client (numeric) 15. poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

Output variable (desired target): 1. y - has the client subscribed a term deposit? (binary: 'yes','no')

DATA EXPLORATION:

I explore the data by visualizing it in the following graphs:

Here I would like to import the data into a variable called the “full_data” and it has all the attributes and now I would like to see the relationship between the attributes of my data so now I have the following things in data exploration module:

Here I would the distribution of age of the customers in the data which we will be useful to predict my outcome yes or no.

I used to explore the data by statistical representation using describe method it will gives the statistical view of data like showing the min,max,standard deviation of data attributes which have numbers as the data.So, used the describe() to find the statistical summary about the data and the statistical summary of data is as follows:

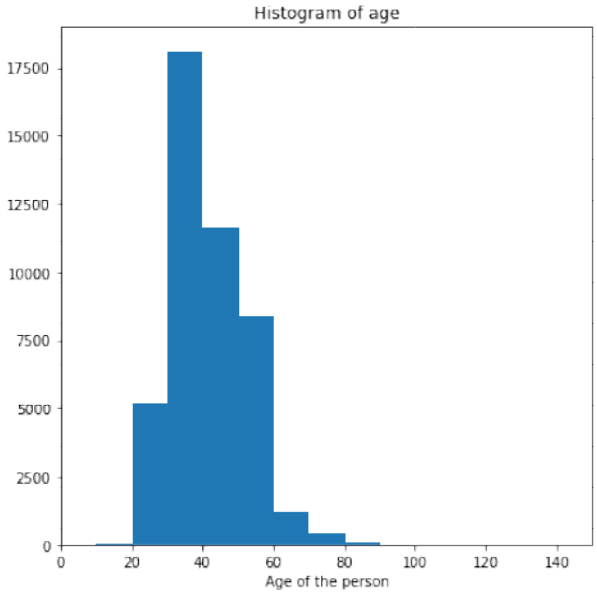
	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

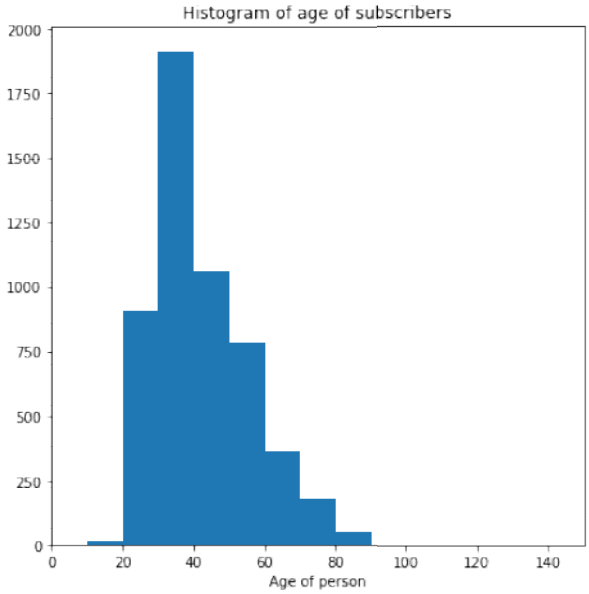
checking whether the data set contain the null values

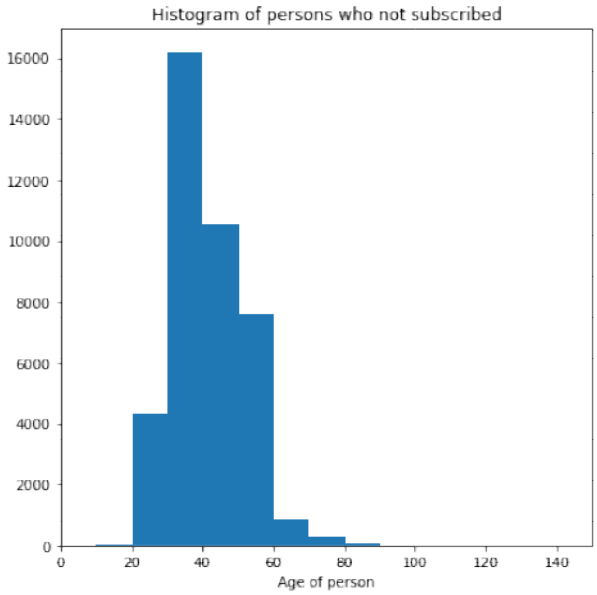
```
full_data.isnull().sum()
```

The distribution of the age in the data of customers are as follows:

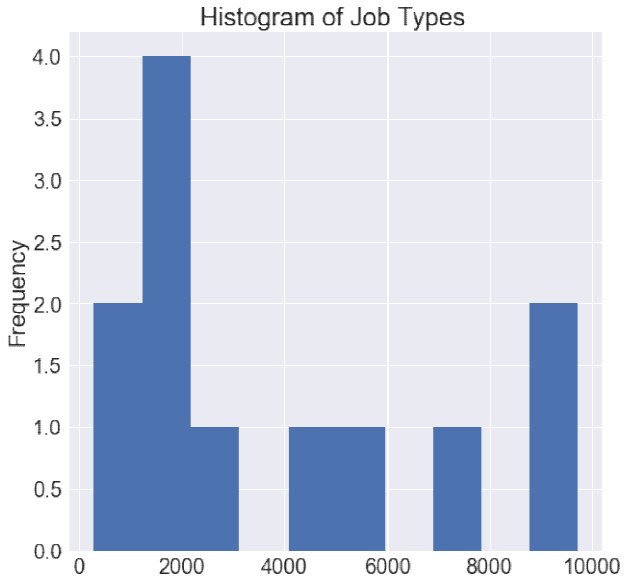
Here I would like to predict the histogram of age of customers and the histogram of age of subscribers and the age of non subscribers and the visualization is as follows:

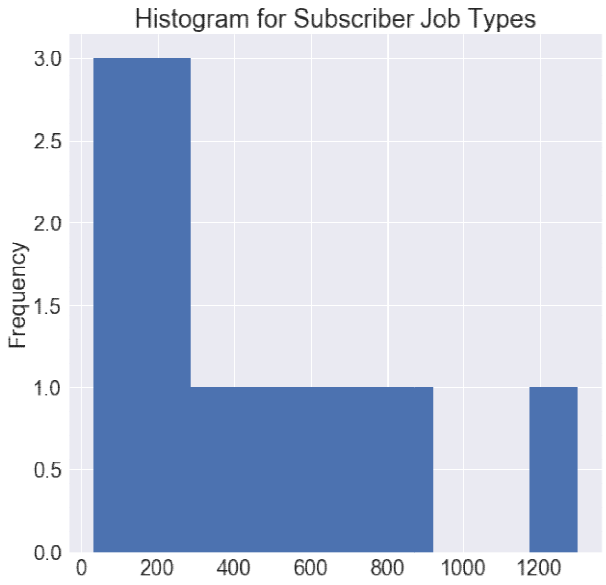

Histogram of age
17500
15000
12500
10000
7500
5000
2500
0
0
20
40
60
80
100
120
140
Age of the person

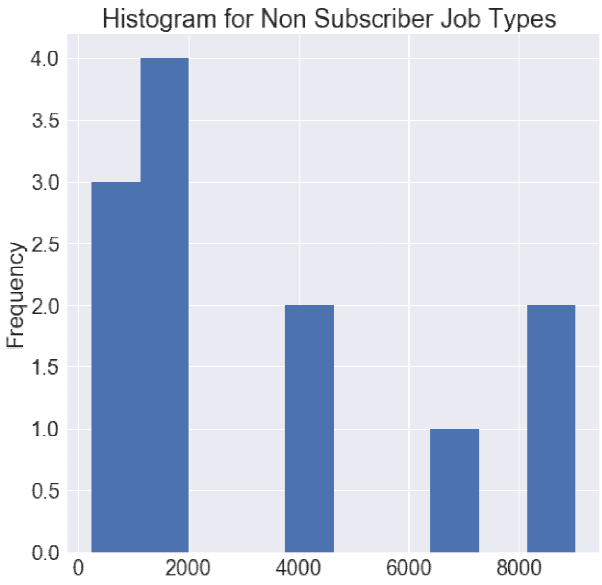

Histogram of age of subscribers
2000
1750
1500
1250
1000
750
500
250
0
0
20
40
60
80
100
120
140
Age of person


Histogram of persons who not subscribed
16000
14000
12000
10000
8000
6000
4000
2000
0
0
20
40
60
80
100
120
140
Age of person

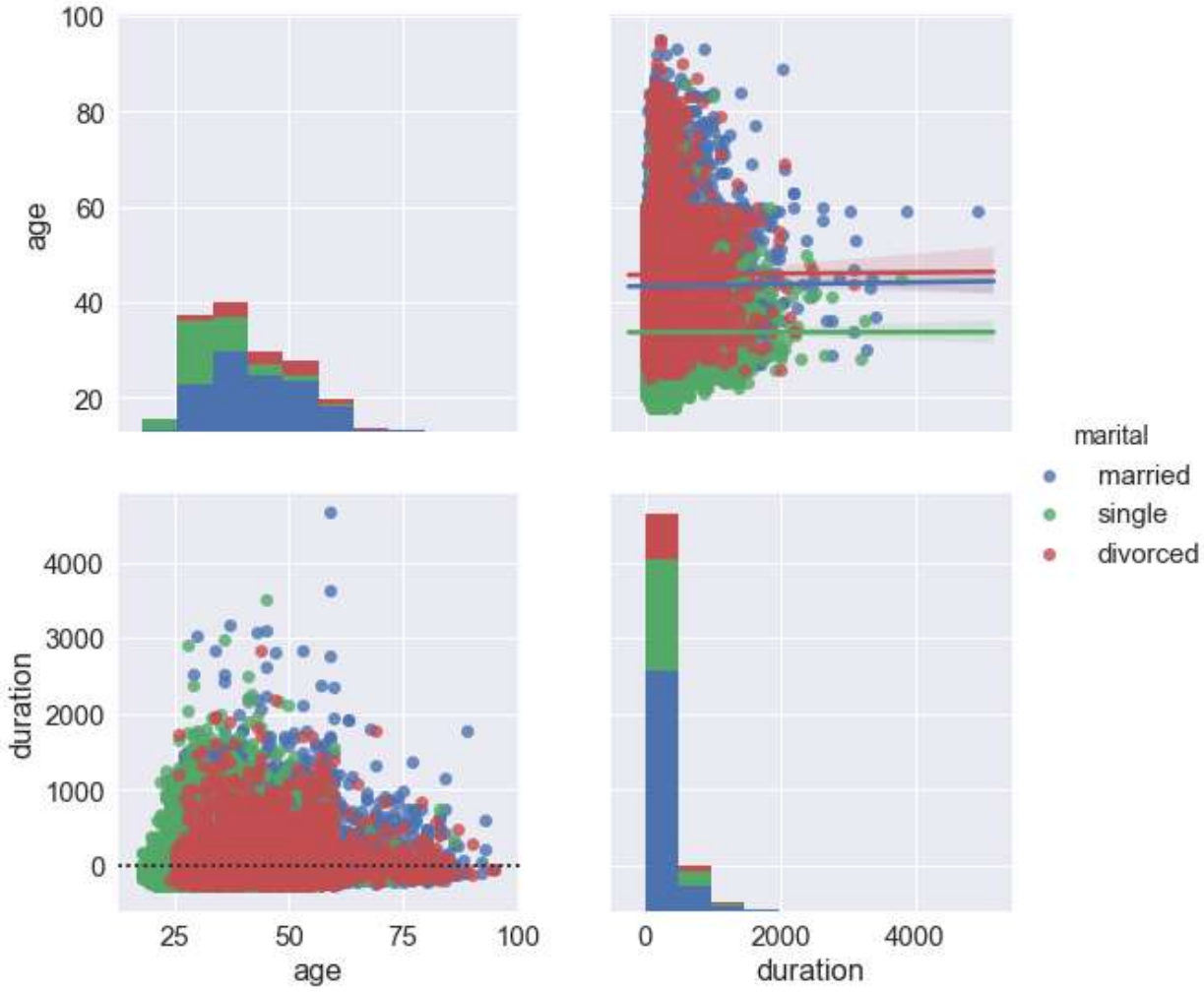
The realtion between the job types will indeed a great phenomenon to predict about the person data in a bank so I used to plot this graph



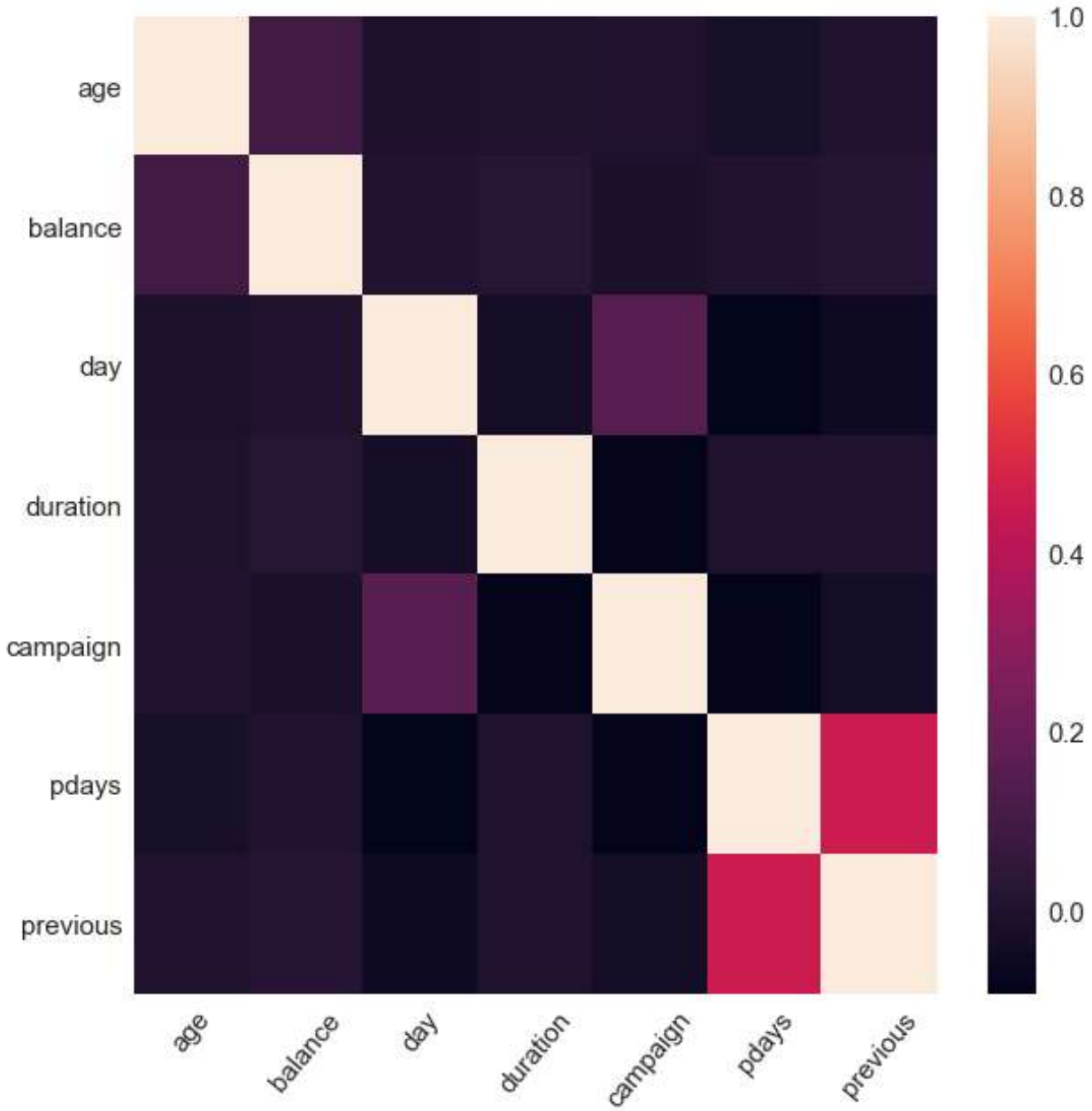




Here, I would like to predict the output of my data using the martial status of a person and age of person.



By using this heatmap we can able to visualize the things like age,martial status and more which will be important to my data set which I have taken.



Preprocessing of data was a crucial step to process the data into a required format.

I used the following algorithms in my model of data:

1.Gaussian naïve_bayes:

In machine learning, **naive Bayes classifiers** are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s, and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines.] t also finds application in automatic medical diagnosis

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

2.Decision trees:

A **decision tree** is a **decision** support tool that uses a **tree**-like model of **decisions** and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

3.Ada boost :

AdaBoost, short for “Adaptive Boosting”, is the first practical boosting algorithm proposed by Freund and Schapire in 1996. It focuses on classification problems and aims to convert a set of weak classifiers into a strong one

4. Random forest:

Random forests or **random decision forests** are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of **decision** trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression).

5.Logistic regression:

Like all **regression** analyses, the **logistic regression** is a predictive analysis. **Logistic regression** is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables

BENCH MARK MODEL:

I have choosen the bench mark model as XGBoost as it was a good ensemble algorithm to deal with the bench mark model.

The results which I obtained are as follows:

```
Accuracy and error is
```

```
Naïve Bayes: 0.867581 (0.009588)
Decision Tree: 0.874032 (0.008565)
RandomForest: 0.904519 (0.007161)
AdaBoost: 0.904593 (0.006638)
```

III. Methodology

Data Preprocessing We will prepare the data by splitting feature and target/label columns and also check for quality of given data and perform data cleaning. To check if the model I created is any good, I will split the data into `training` and `validation` sets to check the accuracy of the best model. We will split the given `training` data in two ,60% of which will be used to train our models and 40% we will hold back as a `validation` set. There are several non-numeric columns that need to be converted. Many of them are simply yes/no, e.g. housing. These can be reasonably converted into 1/0 (binary) values. Other columns, like profession and marital, have more than two values, and are known as categorical variables. The recommended way to handle such a column is to create as many columns as possible values (e.g. profession_admin, profession_blue-collar, etc.), and assign a 1 to one of them and 0 to all others. These generated columns are sometimes called dummy variables, and we will use the pandas.get_dummies() function to perform this transformation. Several Data preprocessing steps like preprocessing feature columns, identifying feature and target columns, data cleaning and creating training and validation data splits were followed and can be referenced for details in attached jupyter notebook.

I have taken the validation size as 0.40 as training data when I we go with the testing set was 0.60 which I found very important when we come to the accuracy point of view and if we will change the validation size as high or low the accuracy will differ.

IMPLEMENTATION:

Here, I implemented the model by extracting the data and visualizing the data and then preprocessing of given data to desired model and then through benchmark model I found the results and trained on various models and at last by implementing the various models and tuning the parameters and by tuning the model I concluded by improving the accuracy of model.

I did the implementation of my model using the various steps like after analysis of data I set my validation size as 0.4 and I obtained the things like training and testing size of data. Later I went with the benchmark model as XGboost. I have chosen it because of it was the best ensemble machine learning algorithm and by going through the testing size data and training size data I used to tune the parameters and I got the accuracy as 0.89 and I would to improve it so I went through the techniques like k-fold because I thought that the k-fold cross validation is a procedure used to estimate the skill of the model on new data. I have chosen the algorithms like

Adaboost which was the most significant boosting algorithm to the model of my data and I used the algorithms like decision trees because it will definitely indeed a super algorithm with various parameters to tune with and I tuned min_depth and max_child_weight for my data set and naïve_bayes was a better algorithm so I have used it and without using the parameters to tune the model I got the accuracy score as 0.86

And I used the random forest algorithm as it bests suit to my model of data due to large data set and finally I used the k-fold techniques because it will split the data into k-groups and take the the group as a whole data set so I thought of using it my implementation model. Anyone easily get the different results only changing the size of validation set which lead to change in training and testing data and this leads to finally change in the accuracy of results and also by tuning the parameters to a different level one can able to adjust the results according to their requirement

Finally, I feel difficulty in mainly tuning the parameters like which parameter to use in order to improve my result as in training of decision tree I found difficulty because it has a lot of parameters to tune in order to get accurate results and at splitting the data into training and testing of data I found difficulty due to imbalance size of data as the more the training data the more the algorithm works to obtain the correct results

and as very high of training data leads to failure of the model. So, this section was very important

Refinement:

We will prepare the data by splitting feature and target/label columns and also check for quality of given data and perform data cleaning. To check if the model we created is any good, we will split the data into training and validation sets to check the accuracy of the best model. We will split the given training data in two ,60% of which will be used to train our models and 40% we will hold back as a validation set.

Parameters which I used for tuning the model are:

max_depth:to control overfitting of data by [3,4,5,6,7,8,9]

min_child_weight:minimum sum of weights required by [2,3,4,5,6,7]

Gamma: minimum loss required to make a split.

RESULTS:

The results which I obtained by tuning the parameters are as follows:

```
For tuning of the parameters {'max_depth':
5, 'min_child_weight': 6} score was
0.9301197923676505
For tuning of gamma the score was
0.9300976464981106
Robustness of model:
```

Model robustness can be understood as - If a model has a testing error(on a new test set) equal to the training error, then the model is said to be robust i.e the model generalises well and doesn't overfit.

In order to explain robustness for logistic regression models, let us take an example of a binary classification problem with output having two levels- 1 and 0.

Training set - Number of 1s - 1000, Number of 0s- 500.
Test pet - Number of 1s - 500, Number of 0s- 250.

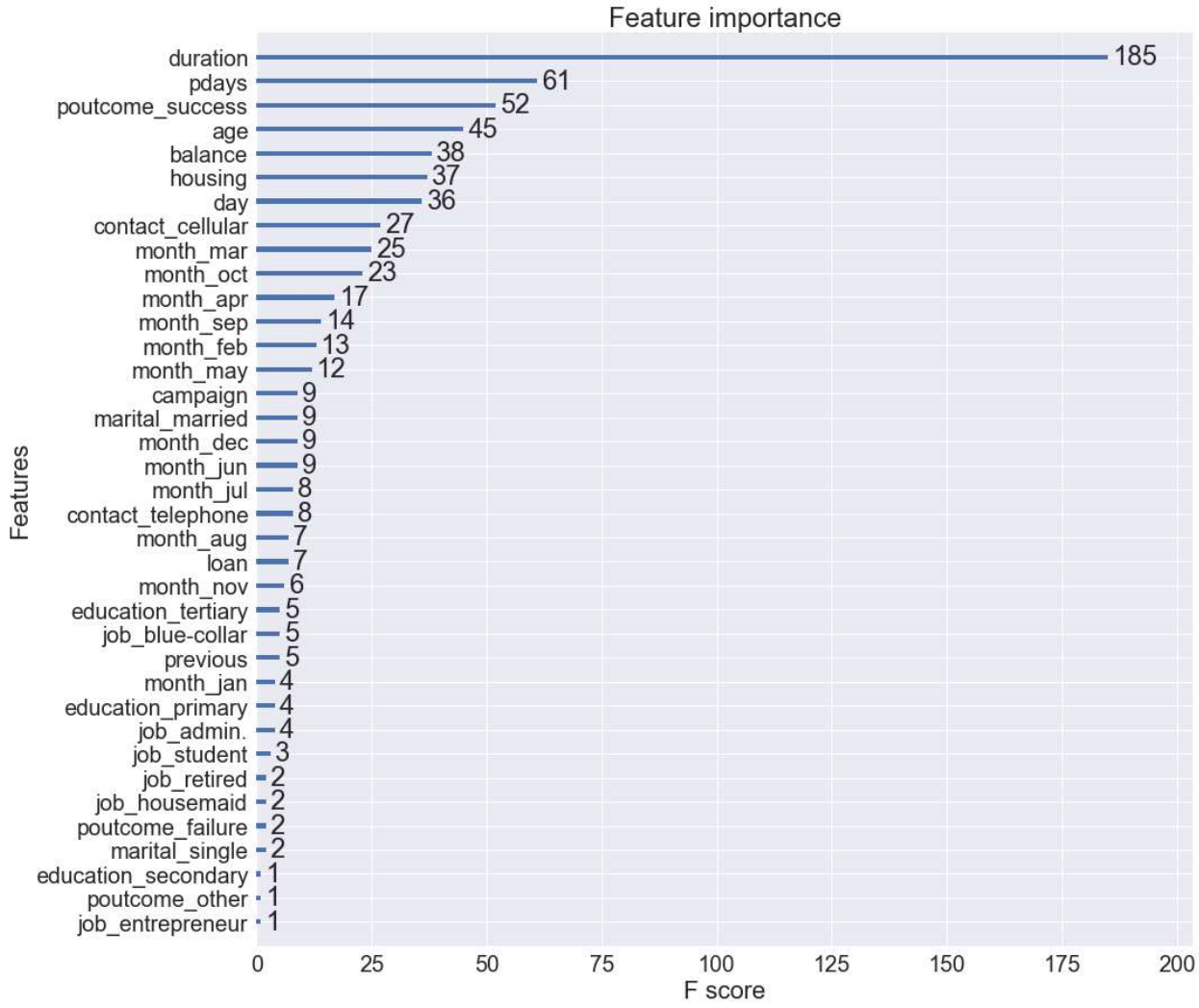
Now after training on the training set, the logistic classifier predicts all 1s as 1s and 0s as 0s perfectly but while making prediction on the test set classifies 250 1s as 0s and 200 0s as 1s.

Such a model will not be called robust since this model failed to generalise to the new dataset.

The model which I have choosen was the robust model.

The model can be trusted to new and unseen data.

Feature importance:



Machine learning works on a simple rule – if you put garbage in, you will only get garbage to come out. By garbage here, I mean noise in data.

This becomes even more important when the number of features are very large. You need not use every feature at your disposal for creating an algorithm. You can assist your algorithm by feeding in only those features that are really important. So, I used to go with the feature importance.

Conclusion:

After tuning the models I used to get the results as the following:

However, after tuning the model and by tuning the parameters I got an improvement in the accuracy score from0.89 to 0.90 which was greater value before tuning the parameters.

I particularly felt difficult when I found the validation_size and to tune the model with different parameters because finding the most necessary parameters for the model is the essence of learning of machine learning and I found it very crucial step and however I succeed by finding the correct scores as an improvement.

Justfication:

I would to justify my results by observation of results from unoptimised model of XGboost to the optimized XGboost model._XGBoost is a tool that serves only one purpose - build GBM for classification/regression. It certainly has its internal hacks for better performance but they are not relevant when it comes to usage. Although, the developers give some explanations on what parameters are the best for certain purposes.

Improvement:

The results can be improved by tuning more parameters like sub sample tuning and gamma tuning etc inorder to get more accurate results.