**SANTANDER CUSTOMER TRANSACTION PREDICTION USING R AND PYTHON**

**Prepared by**: **MOULEESHWARAN.S**

**Prepared on: 30.08.2019**

**INTRODUCTION**

**Background:**

At Santander, mission is to help people and businesses prosper. We are always looking for ways to help our customers understand their financial health and identify which products and services might help them achieve their monetary goals.

Our data science team is continually challenging our machine learning algorithms, working with the global data science community to make sure we can more accurately identify new ways to solve our most common challenge, binary classification problems such as:

* + is a customer satisfied?
  + Will a customer buy this product?
  + Can a customer pay this loan?

**Problem Statement:**

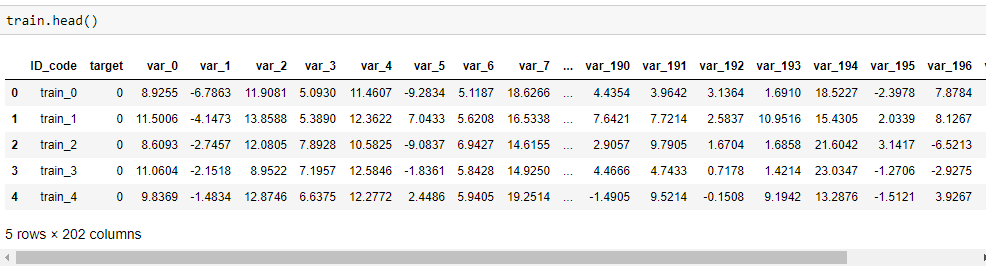
In this challenge, we need to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted.

**Data set:**

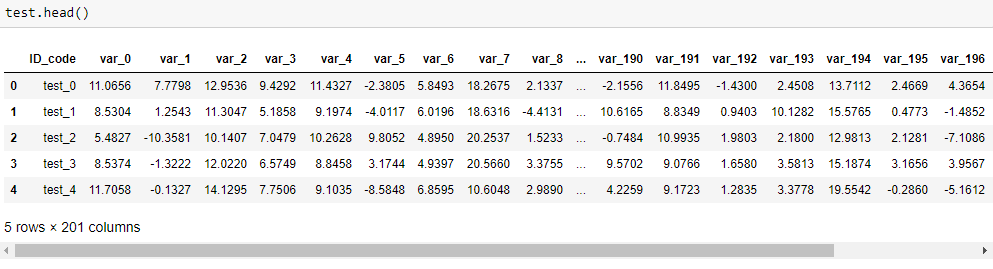
We have two types of dataset:

* One is train data set which contains target variable.
* Another is test data set which does not contain target variable.

**TRAIN DATASET:**



**TEST DATASET:**

****

**Our goal:**

We need to predict the target variable in the test dataset with the help of building ML algorithms by using train dataset.

Train data set contains target variable which has two observations

* + 0 indicates no which means customer does not make transaction
  + 1 indicates yes which means customer makes transactions irrespective to amount

**METHODS AND ANALYSIS**

**LOADING REQUIRED LIBRARIES FOR PYTHON AS WELL AS R**

1. **For python**

* import os
* import pandas as pd
* import numpy as np
* import matplotlib.pyplot as plt
* import seaborn as sns
* import statsmodels.api as sm
* from random import randrange, uniform

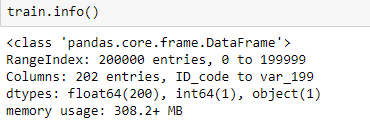
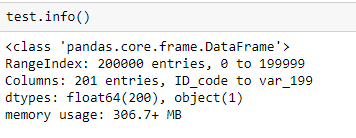
1. **For Rscript**

* library(gridExtra)
* library(grid)
* library(ggplot2)
* library(caret)
* library(rpart)
* library (DataCombine)
* library (ROSE)
* library(e1071)
* library(xgboost)

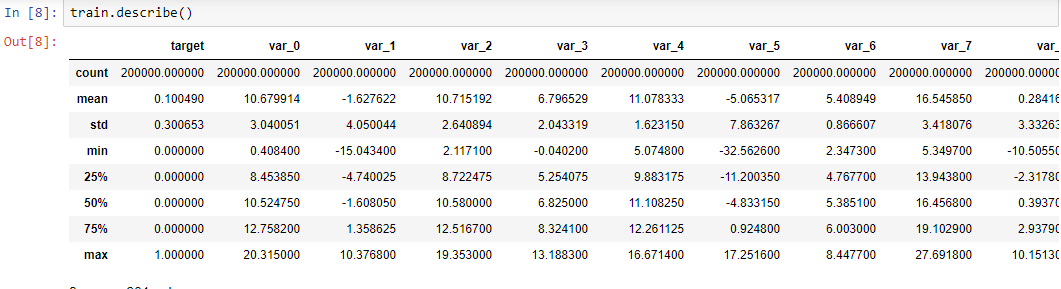
Here I have mentioned some of the basic libraries required for some operations. If need we need to call the required libraries for performing the operations

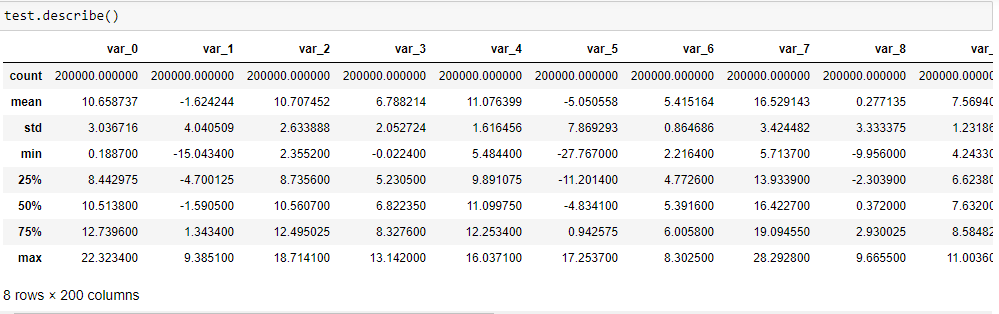
**EXPLORATORY DATA ANALYSIS:**

We need to get more information about our data.



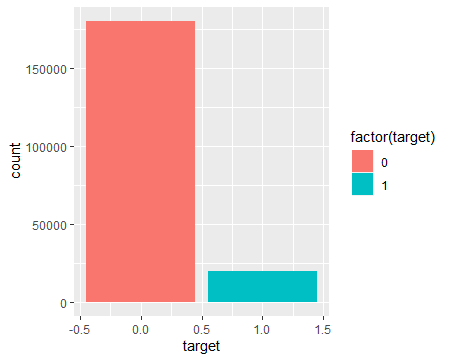
Above diagram shows the dimensions, data types and memory consumed by the data





Above diagram shows the summary of the dataset

Now let us see the distribution of target value with diagrammatic illustrations

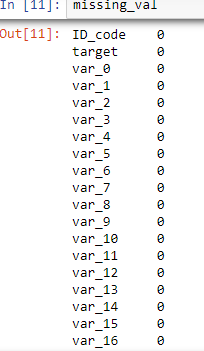
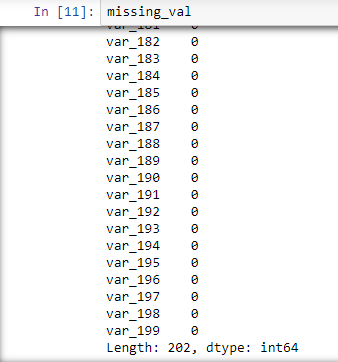


Above diagram clearly shows that about 10% of the dataset have value of 1(yes) and remaining are 0. our dependent variable is biased.

**MISSING VALUE ANALYSIS:**

We need to check whether there are any missing values present in the dataset

Missing values can be occurred due to human error or if the field marked as an optional. There is no use of that data hence we need to do either drop the variable or observation or impute with any one of the methods like mean, mode and KNN imputation for numerical variables and median method for categorical variables.



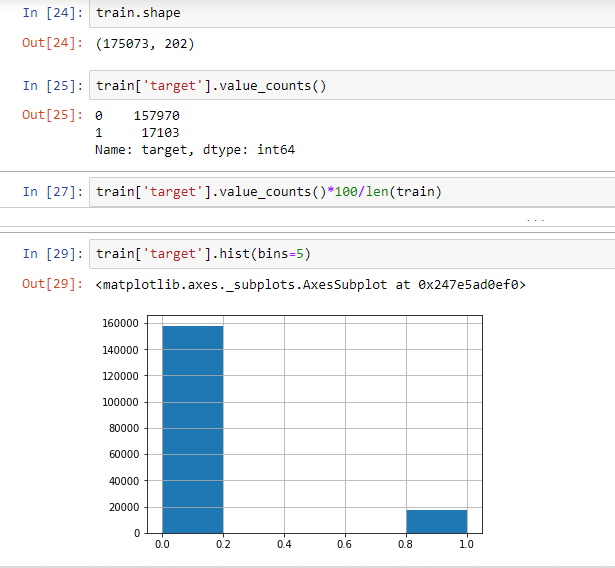
In our dataset there is no missing values. So, we are free from the use of missing value analysis.

We can simply jump to outlier analysis.

**OUTLIER ANALYSIS**

Outliers are inconsistent data which is not lying under the range of other variables. outliers may lead the model towards inconsistent variable and biased towards it in our data set outliers are detected and removed.it makes our dataset small and doesn’t affect our whole dataset.

A box plot is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagrams. Outliers may be plotted as individual points.

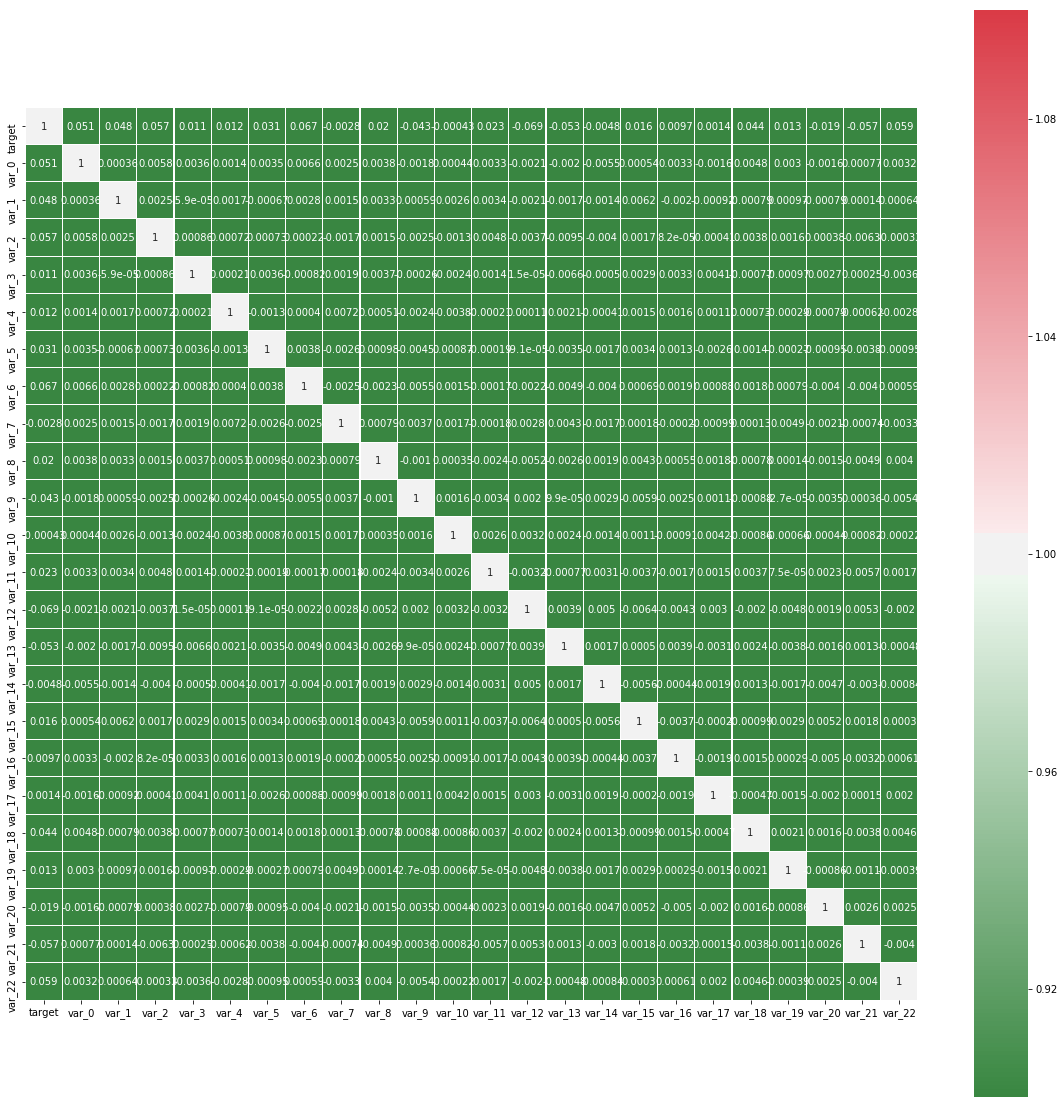


Above represents the shape and distribution of target value after performing outlier analysis.

**FEATURE SELECTION:**

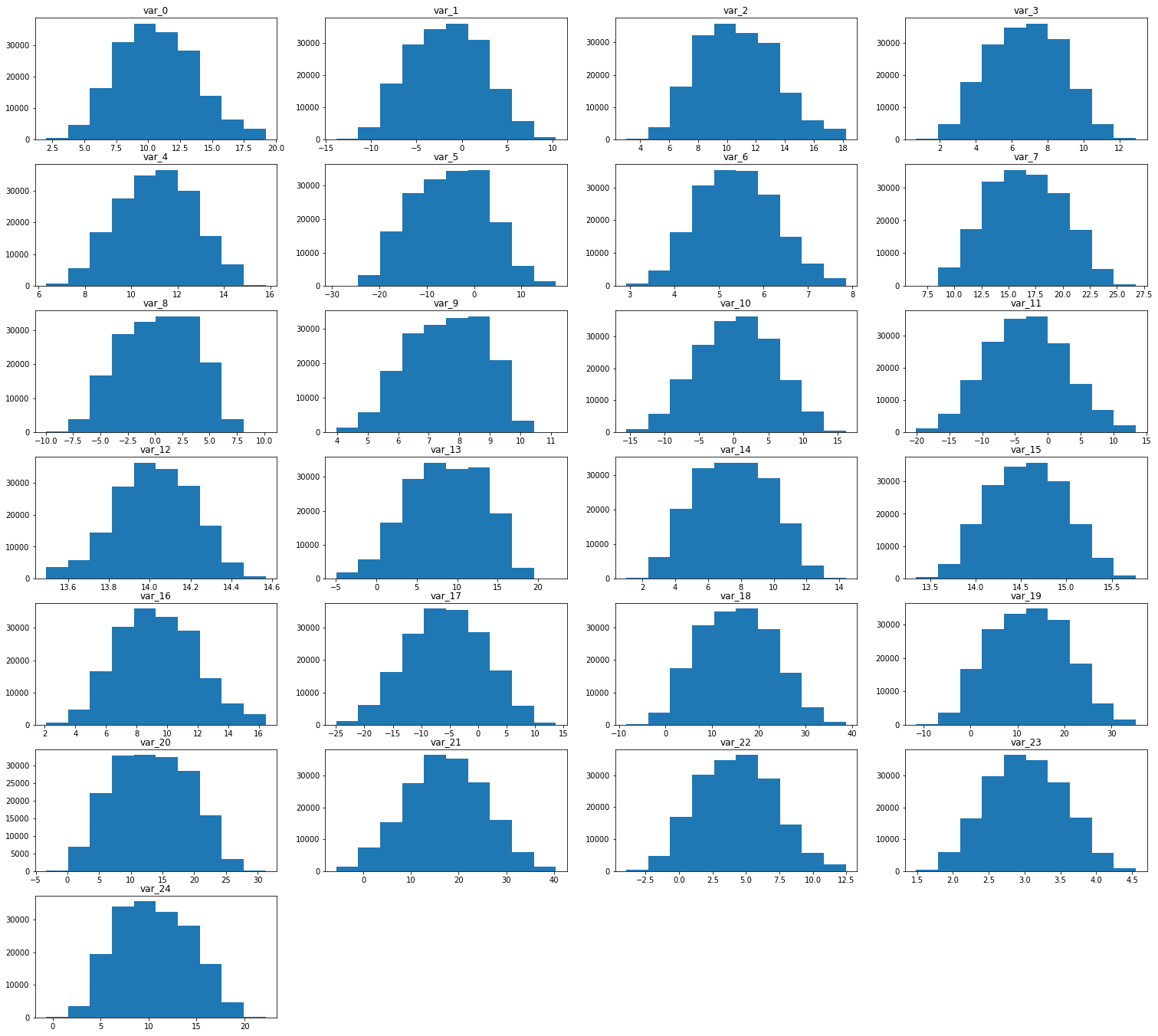
Some have sensitive data which might enhance the prediction of the target variable and some variable can contain noise which have mere any impact on the prediction. We can bring to use various plots and heat maps which will show us the importance of the variable and whether the act as a duplicate to other variable in which case only one of them will suffice for our model development. Co-relation analysis determines whether the correlation between two variables is positive, negative or have no correlation at all. Say if correlation between two variables is 1, we can drop one of them as both of the variable will fetch the same output.

From the below plot, we can clearly know that the variables are not correlated to each other.



**FEATURE SCALING:**

We can convert each variable on a single comparison number starting from 0 to 1. This can be applied on the dataset which contains multiple variables and huge size of data like our current dataset. Variables containing different measure scale can cause anomalies which while doing a comparison may not be interpreted in the correct manner. This step will reduce unwanted variation either within or between variables. It also brings all of the variables into proportion with one another. Normalization was used to get all the variables in line to a single range.



The above plot shows the distribution of data for the first 25 columns

**MODEL DEVELOPMENT**

In our dataset, I am going to use different models in both languages. They are mentioned below as follows:

* **IN PYTHON:**

1. Decision tree
2. Random forest
3. KNN Imputation
4. Logistic Regression
5. Naive Bayes
6. SGDClassifier

* **IN RSCRIPT:**

1. Logistic Regression
2. Decision tree
3. Naive Bayes
4. XGBoost

**SAMPLING METHODS**:

Before training the model, we need to treat the imbalanced data.so that it may give accurate results. We can achieve it using sampling methods. There are many different methods for drawing samples from data; the ideal one depends on the data set and situation. Sampling can be based on probability an approach that uses random numbers that correspond to points in the data set to ensure that there is no correlation between points chosen for the sample. They are

1. Simple random sampling
2. Stratified sampling
3. Over sampling
4. Under sampling

**IN PYTHON** - I had used stratified sampling method for treating the imbalanced dependent variable for all models

**In R SCRIPT** – I had used random over sampling, stratified sampling and non-sampled data based on the different models

**DECISION TREE:**

It is very powerful tool for both classification and prediction. It has two most used algorithm methods.

1. C5.0
2. CART

**C5.0** uses entropy (Information Gain) as major splitting criteria and **CART** uses Gini index as prominent criteria

In my model **C5.0** is used and the evaluation metrics are mentioned below.

**IN PYTHON:**

After sampling I had split the whole dataset into train data and test data for predicting the results of my trained data. Here is my result of the trained data using my model.

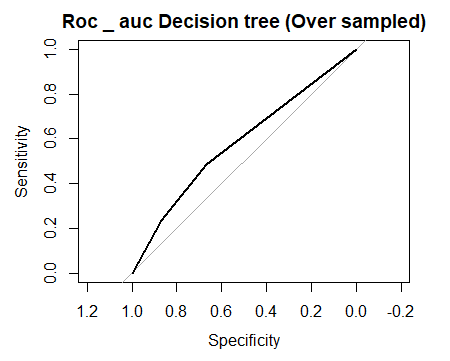
Accuracy =58.71

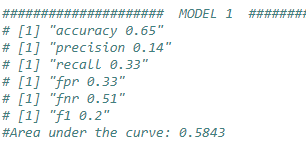
FNR = 41.62

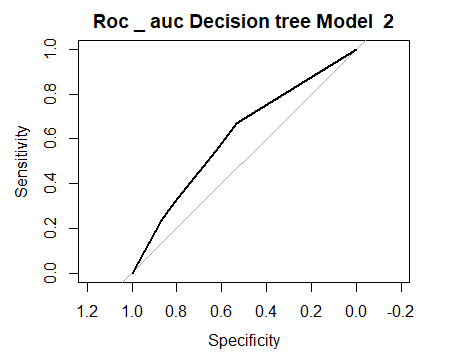
Recall = 0.58

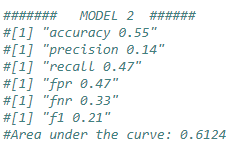
Precision = 0.58

**IN R SCRIPT:**

 I had used over sampling method (MODEL 1) and manually initialising the depth of the tree (MODEL 2) for evaluating my test data. Below is the result of my trained data.







**LOGISTIC REGRESSION:**

 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. For using this model there should be no outliers.it is also predictive analysis. It mainly identifies relationships between the data.

Logistic Regression is a supervised machine learning algorithm where the predicted output is categorical.

**IN PYTHON:**

Data is fragmented to train and test data using under sampling methods. In our dataset we got,

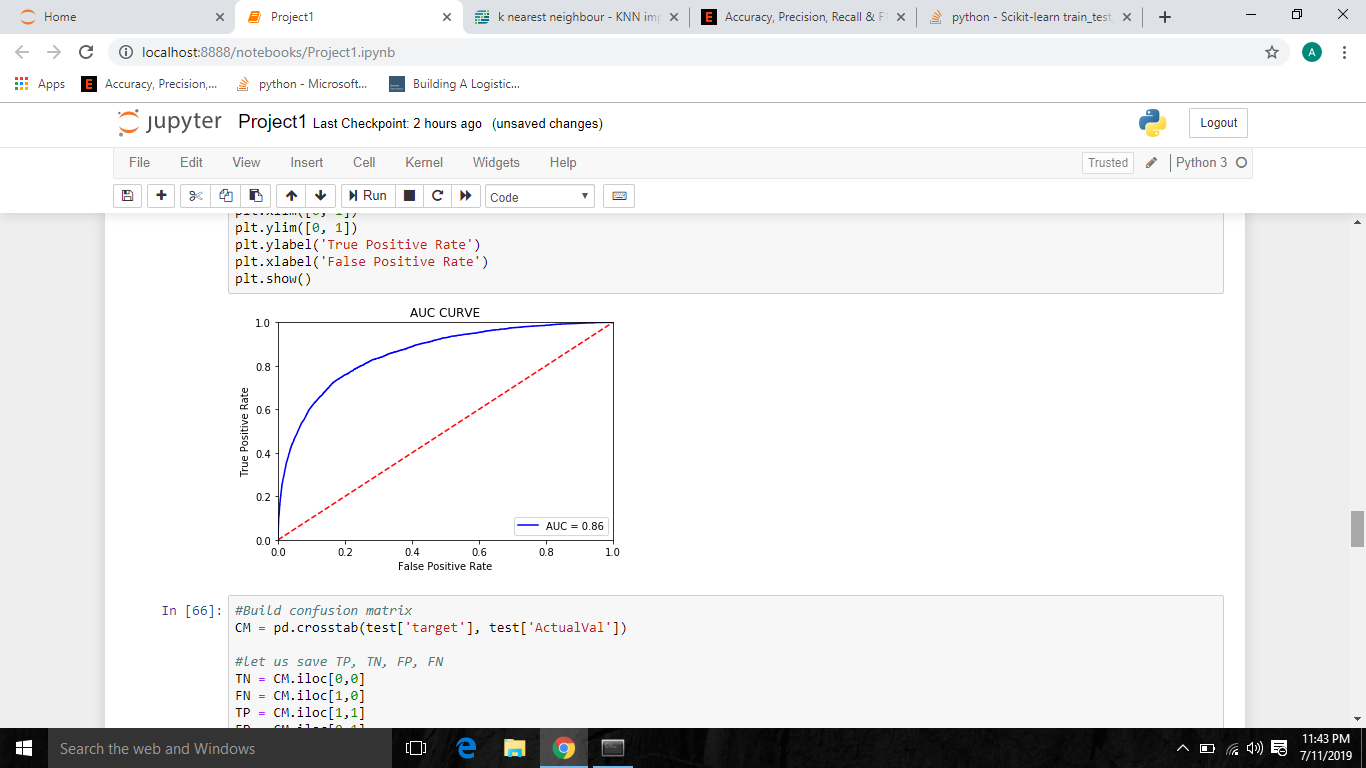
Accuracy = 76.98

FNR = 23.03

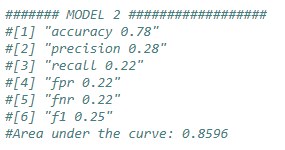
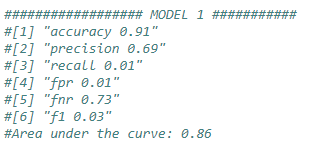
Recall = 0.77

precision = 0.76

AUC = 0.86



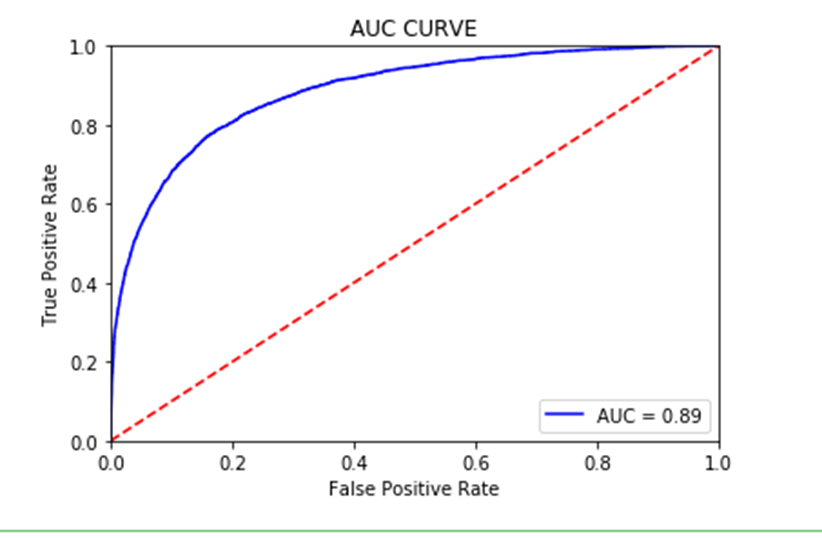
**IN R SCRIPT:**

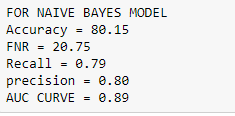
Here I have used stratified sample data for logistic base model **(MODEL 1)** and over sampling method for **MODEL 2**. Output of the trained model is mentioned below

**NAIVE BAYES:**

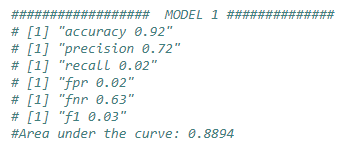
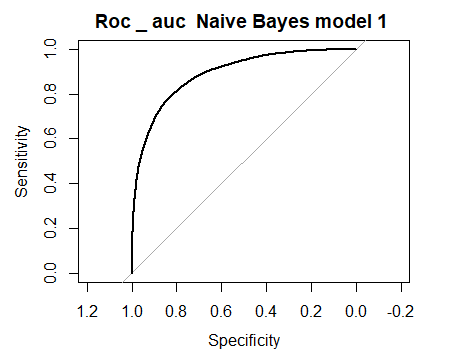
It is classification and practical learning algorithm.it classifies based on the probability. It mainly works on Bayes theorem to predict the probability of the dataset.

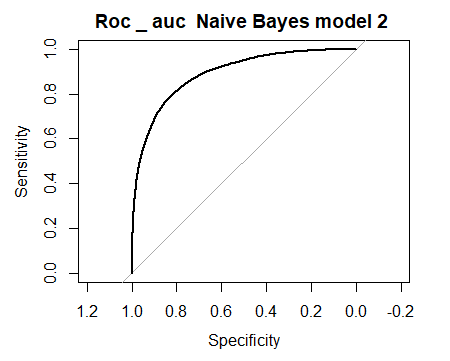
**IN PYTHON:**

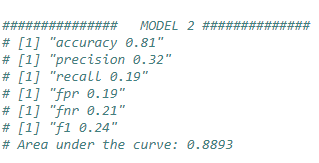




**IN R SCRIPT:**

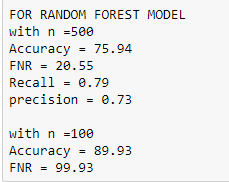
** MODEL 1 – STRATIFIED SAMPLING METHOD**

**MODEL2 – OVER SAMPLING METHOD:**

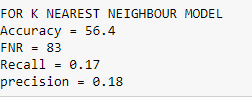


*Below mentioned models are used only in python only*

**RANDOM FOREST:**

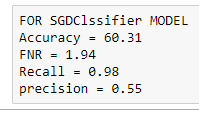
Random forests  are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean prediction (regression) of the individual trees. In our dataset we got,

**K-NEAREST NEIGHBOUR(KNN):**



KNN is simple algorithm that stores all available cases and classifies new cases based on a similarity measure. It is a lazy algorithm because it takes more time to train the model. In our data set we got,

**SGD CLASSIFIER:**

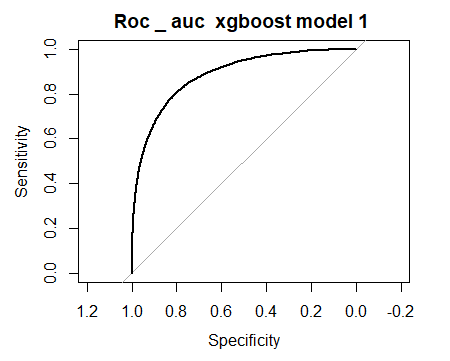
Stochastic Gradient Descent trains the model by randomly taking set of data in the whole dataset for each iteration. By using that information, it trains our model.it is very simple and efficient to use. In our dataset we got,

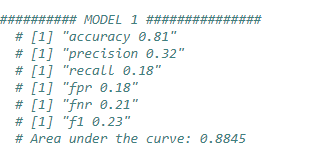
*Below mentioned model is used only in R*

**XGBOOST:**

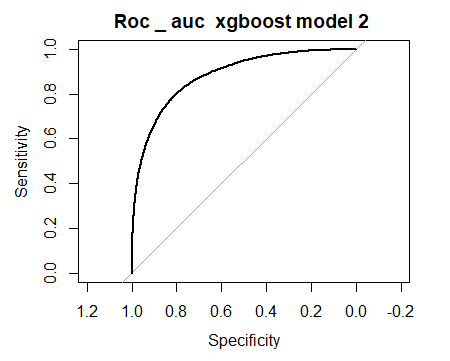
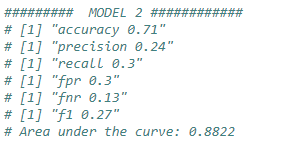
XGBOOST is the decision tree-based ensemble algorithm which provides gradient boosting framework.

**MODEL 1 – STRATIFIED SAMPLING METHOD**





**MODEL 2 – OVER SAMPLING METHOD**



**MODEL EVALUATION**

Evaluation of model can be different for classification and regression problems.

**For classification problems, the error metrics are**

1. *ACCURACY*
2. *PRECISION*
3. *RECALL*
4. *FALSE NEGATIVE RATE*
5. *FALSE POSTIVE RATE*

**For regression problems ,the error metrics are**

1. *MSE(Mean Square Error)*
2. *RMSE(Root Mean Square Error)*
3. *MAPE(Mean Absolute Percentage Error)*

In our dataset, we are going to predict that whether the customer makes transaction or not. For that apart from accuracy our important error metrics must be false negative rate. Because if our model predicted potential customers wrongly means the client going to lose the customers.

I had mentioned all the predicted test cases in a tabluar format in both Python and R

By comparing all the models in the tabular format,we came with final decision with the help of error metrics.



We need to select the model with low FNR and high auc curve with acceptable accuracy.

**IN PYTHON**

I am selecting **LOGISTIC REGRESSION** algorithm because it has low FNR value and acceptable accuracy.

Error metrics values are mentioned below:

* Accuracy = 76.98
* FNR = 23.03
* Recall = 0.77
* precision = 0.76
* AUC curve = 0.86

**IN R SCRIPT**

I am selecting **NAIVE BAYES** algorithm because it has low FNR value, high AUC value, high f1 score and acceptable accuracy.

Error metrics values are mentioned below:

* Accuracy = 0.81
* Precision = 0.32
* Recall = 0.18
* FNR = 0.21
* F1 = 0.23
* AUC curve = 0.88

REFERENCES:

* Edwisor (learning.edwisor.com)
* Kaggle (Kaggle.com)
* GitHub(github.com)
* Google (rbloggers, towardsdatascience, mediam and Wikipedia