Project Report

on

Santander Customer Transaction Prediction

By:
Shivam Choudhary

Table of Contents:

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 Details:
Provided with an anonymized dataset containing 200 numeric feature variables, the binary target column, and a string ID-code column.

Problem Analysis:
This is a binary classification problem under supervised machine learning algorithm. The task is to predict the value of target column in the test set.

Tools:
Python Jupyter Notebook, R Studio.

Techniques:
Logistic Regression, Decision Tree, Random Forest, Naïve Bayes.

Exploratory Data Analysis

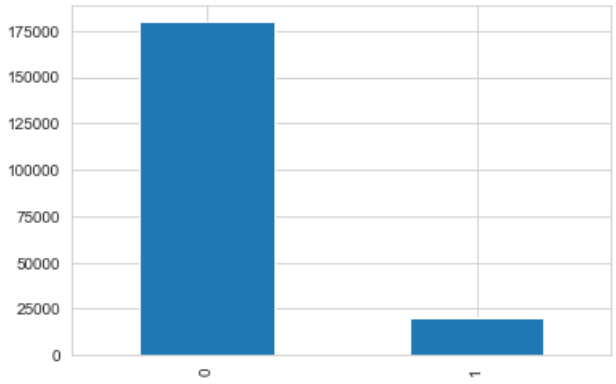
Shape of data:
200000 observations with 202 columns in train data and 200000 observations with 201 columns in test data.

Variable data types in train and test data:
Id_code and target is object type, 200 anonymous variables of float type.

Missing Value Analysis:
First, we check if there is any empty cell in any column or not. In the Train and Test data, there is no empty cell in any variable, therefore, there is no need to impute any value or remove any row.

Graphs:
Plotting Graphs to know the nature of data.

Target Column:
```
0: 179902
1:  20098
```



The Dataset is Highly Imbalanced.

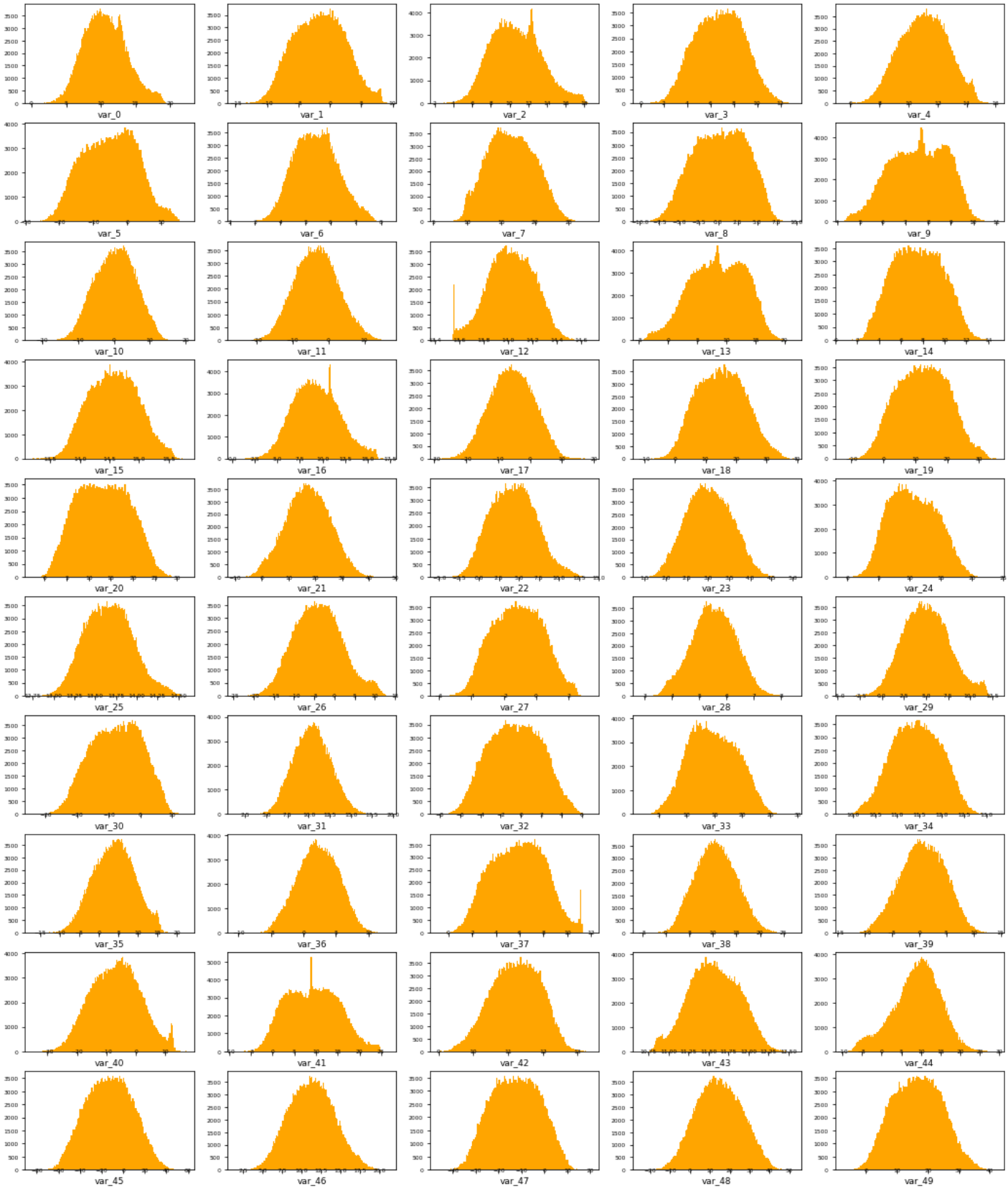
Distribution of Numerical Variable:

Train Dataset:

Allmost all features of Train dataset follow normal distribution.

NOTE: Showing only 50 variables in report, the plot of rest of the variables can be seen in Graphs folder attached with project with file name Distribution of Train.

Test Dataset:



Allmost all features of Test dataset follow normal distribution.

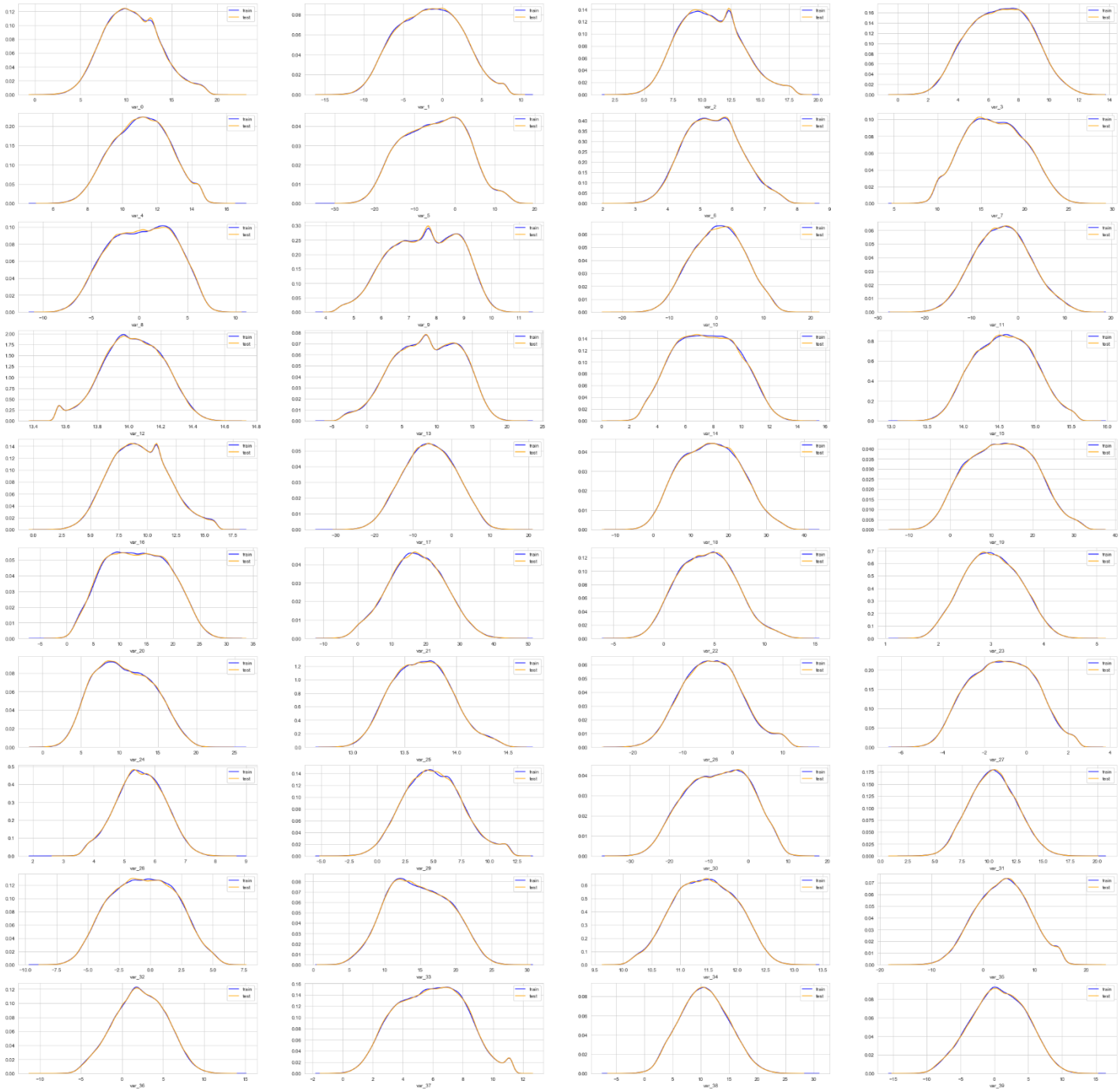
NOTE: Showing only 50 variables in report, the plots of rest of the variables can be seen in Graphs folder attached with project with file name Distribution of Test.

Distribution of all numerical features per class:

There are number of features with significant different distribution for the two target values.

NOTE: Showing only 50 variables in report, the plot of rest of the variables can be seen in Graphs folder attached with project with file name Distribution of Train per class.

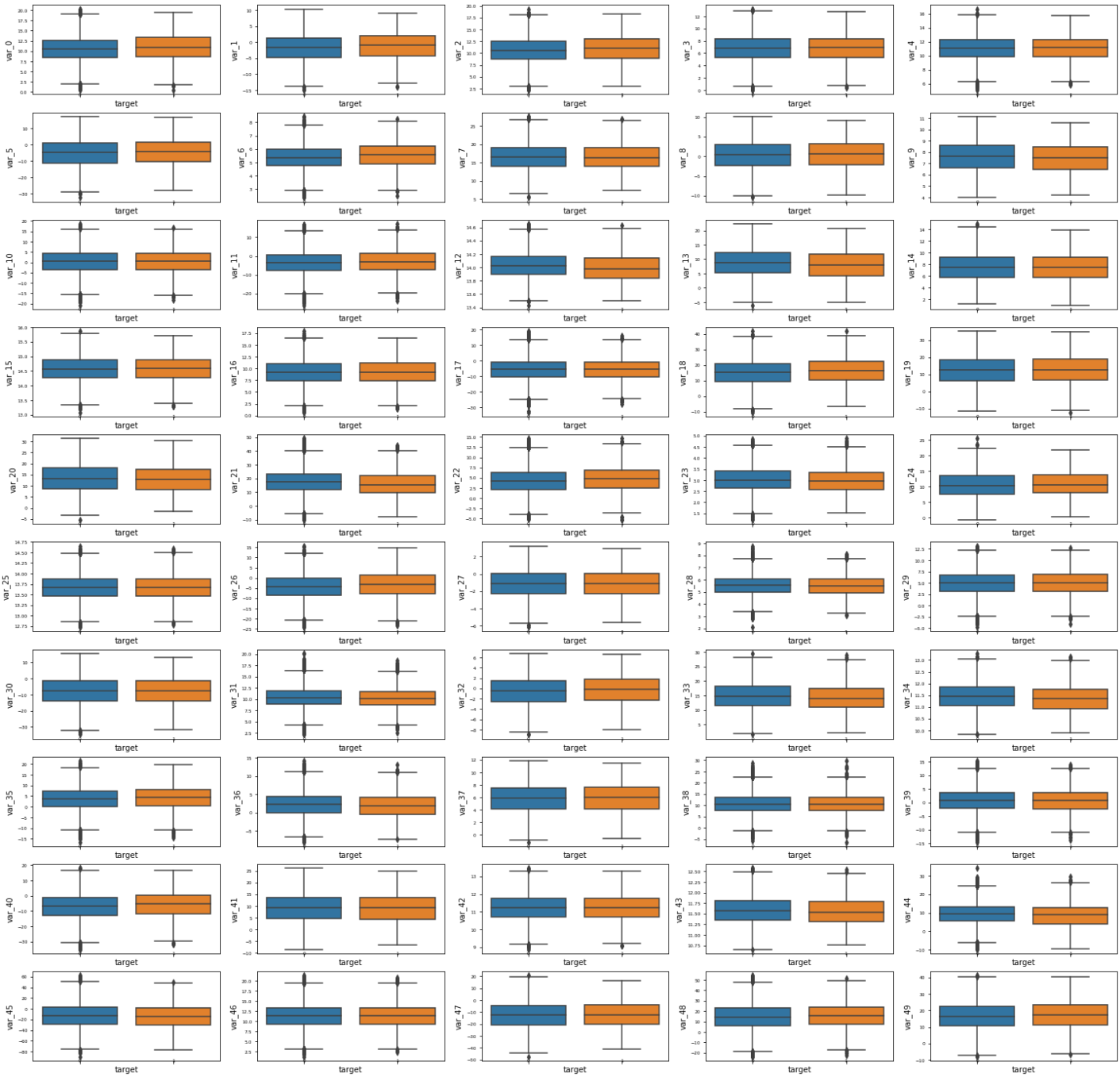
Distribution of numerical features in train and test data:



The train and test seems to be well balanced with respect to distribution of the numeric variables.

NOTE: Showing only 40 variables in report, the plots of rest of the variables can be seen in Graphs folder attached with project with file name Distribution of Train and Test dataset.

Outlier Analysis:



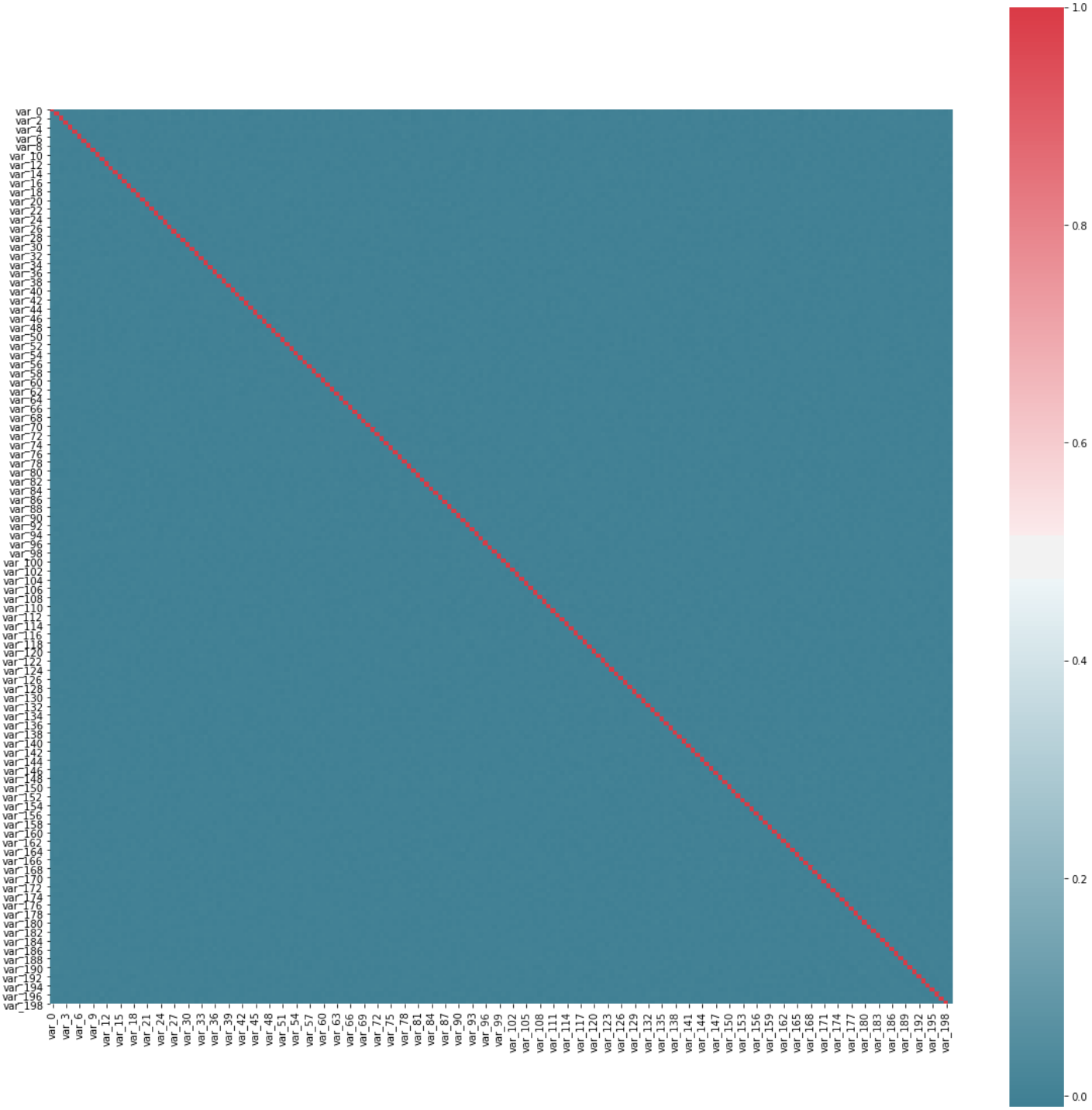
By plots, we can see that there are very few outliers presrent in our dataset.
Using Box Plot method, we tested continuous numerical variables for outliers. After detecting outliers, we replaced outliers with Median value of that column.

NOTE: Showing only 50 variables in report, the plots of rest of the variables can be seen in Graphs folder attached with project with file name boxplot of Train set.

Feature Selection:

Some variables contains same information, so it's better to remove such variables.
Correlation among numerical variables:
Minimum correlation among variables is -0.009844361358419677

Maximum correlation between variables is 0.009713658349534146



All 200 Numerical variables are not correlated with each other. Therefore, no variable is removed here.

Feature Scalling:

We have scaled the data using standarization so that difference between means of variable should not be very large as it effects the model.

Sampling:

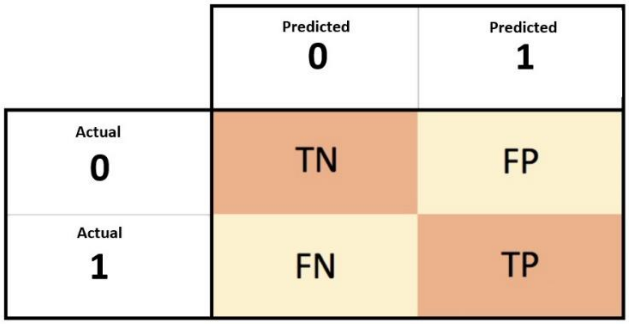
We have used Stratified sampling method to divide Test data in Test_data and Training_set datasets. Training_set consists of 70% of Historical data while Test_data have remaining 30% data.

Models:

For our Santander Customer Transaction Prediction project, we evaluated our models using the number of AUC, Precision, Recall, accuracies and FNR metrics.

Confusion Matrix:

It is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values. It is extremely useful for measuring Recall, Precision, Specificity, Accuracy and most importantly AUC-ROC Curve.



- **True Positive:** You predicted positive and it's true.
- **True Negative:** You predicted negative and it's true.
- **False Positive:** (Type 1 Error) You predicted positive and it's false.
- **False Negative:** (Type 2 Error) You predicted negative and it's false.
- **Accuracy:** Overall, how often is the classifier correct?
- **Recall:** (True positive rate) Out of all the positive classes, how much we predicted correctly. It should be high as possible.
- **Precision:** Out of all the classes, how much we predicted correctly. It should be high as possible.
- **False Negative Rate:** When it's Yes, how often does it predict no?
- **False Positive Rate:** When it's no, how often does it predict yes?
- **True Negative Rate:** (Specificity)When it's no, how often does it predict no?
- **ROC Curve:** It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class.

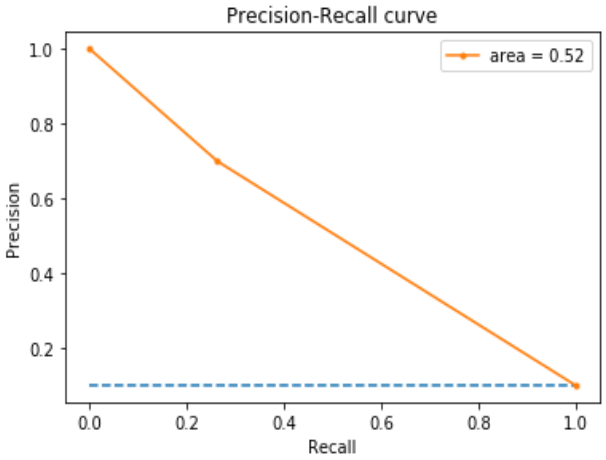
Logistic Regression:

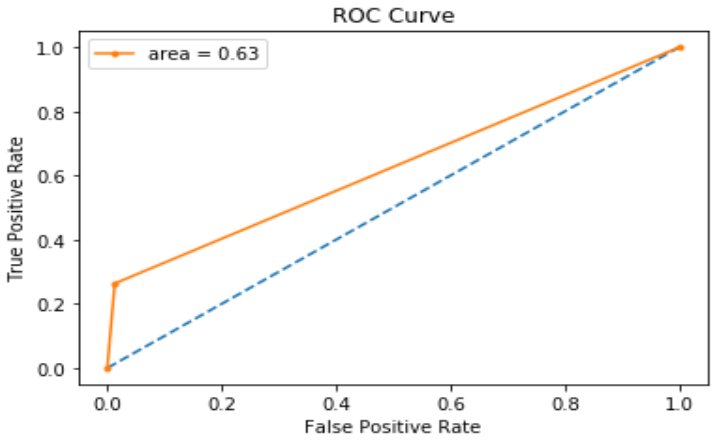
Logistic Regression is used when the dependent variable(target) is categorical. It is bias towards classes which have large number of instances. It tends to only predict the majority class data. The features of the minority class are treated as noise and are often ignored. Thus, there is a high probability of misclassification of the minority class as compared to the majority class.

Code in Python:

		Predicted	
		0	1
Actual	0	53161	695
	1	4525	1619

- **True Positive:** 1619
- **True Negative:** 53161
- **False Positive:** 695
- **False Negative:** 4525
- **Accuracy:** 91.30%
- **Recall:** 26.35%
- **Precision:** 69.96%
- **False Negative Rate:** 73.65%
- **AUC-ROC Curve:** 63.00%
- **AUC-PR Curve:** 52.00%

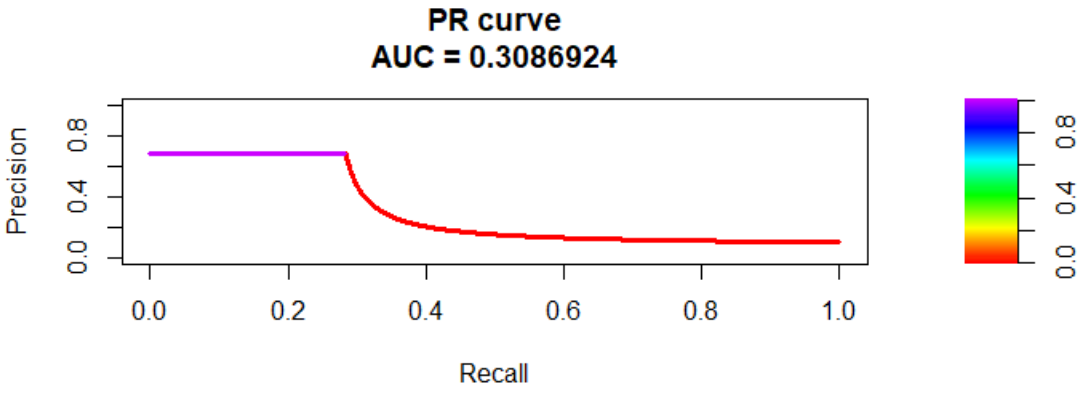


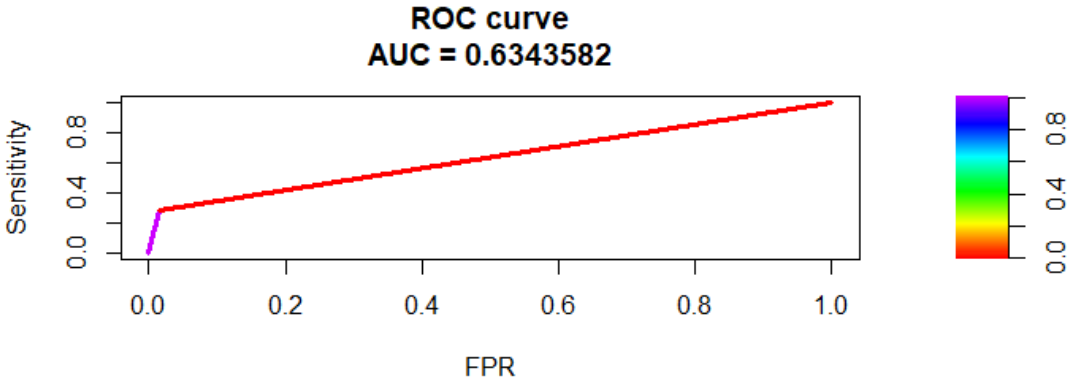


Code in R:

		Predicted	
		0	1
Actual	0	53193	778
	1	4322	1707

- **True Positive:** 1707
- **True Negative:** 53193
- **False Positive:** 778
- **False Negative:** 4322
- **Accuracy:** 91.50%
- **Recall:** 28.31%
- **Precision:** 68.69%
- **False Negative Rate:** 71.68%
- **AUC-ROC Curve:** 63.43%
- **AUC-PR Curve:** 30.86%





Result:

Accuracy is good by Logistic Regression but have high False Negative Rate. Precision, Recall and AUC-ROC are not much higher as Compared to Naïve Bayes.

Random Forest:

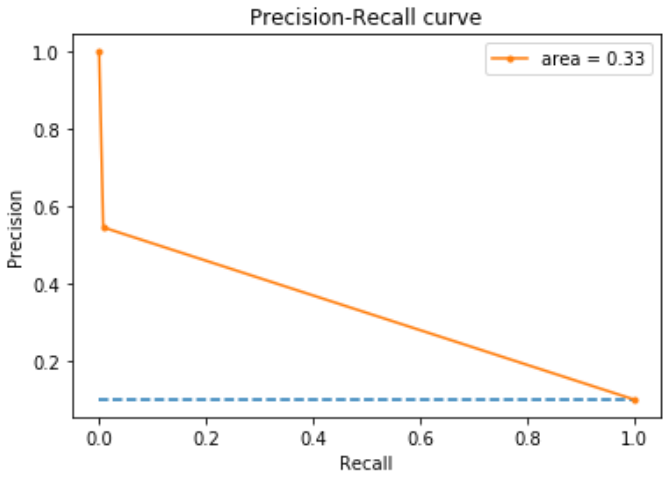
Random forest is slow in generating predictions because it has multiple decision trees. Whenever it makes a prediction, all the trees in the forest must make a prediction for the same given input and then perform voting on it. This whole process is time-consuming.

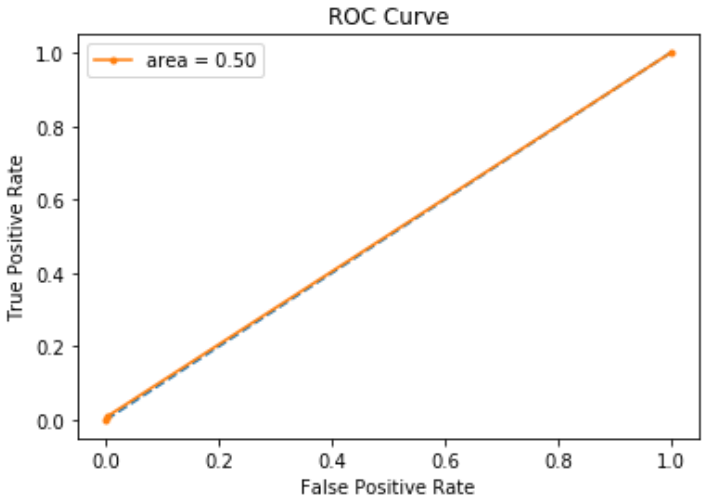
Code in Python:

		Predicted	
		0	1
Actual	0	53817	39
	1	6097	47

- **True Positive:** 47
- **True Negative:** 53817
- **False Positive:** 39
- **False Negative:** 6097
- **Accuracy:** 89.77%
- **Recall:** 0.76%

- **Precision:** 54.65%
- **False Negative Rate:** 99.23%
- **AUC-ROC Curve:** 50.00%
- **AUC-PR Curve:** 33.00%

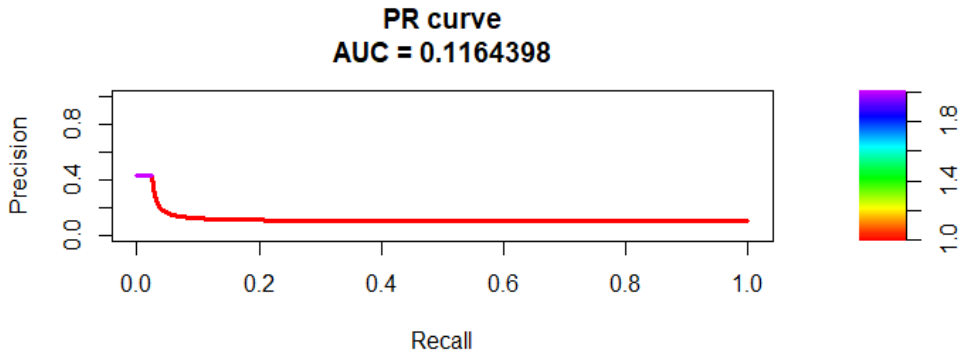


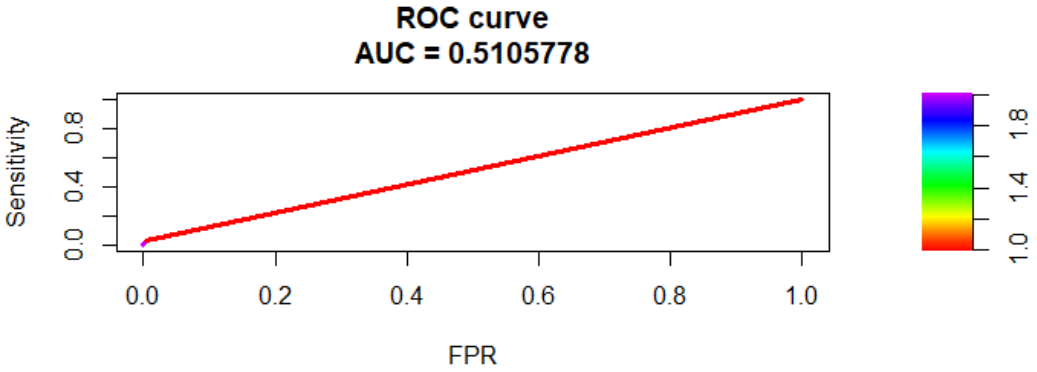


Code in R:

		Predicted	
		0	1
Actual	0	53770	201
	1	5879	150

- **True Positive:** 150
- **True Negative:** 53770
- **False Positive:** 201
- **False Negative:** 5879
- **Accuracy:** 89.87%
- **Recall:** 2.48%
- **Precision:** 42.73%
- **False Negative Rate:** 97.51%
- **AUC-ROC Curve:** 51.05%
- **AUC-PR Curve:** 11.64%





Result:

Random Forest Model have AUC 0.5, it means model has no class separation capacity. Recall and Precision value is also very low and False negative rate is almost 100%. Therefore, not selecting this model. Random Forest is very time consuming, so the observations are made on 10 number of trees only. Selecting a greater number of trees consumes a lot of time.

Naïve Bayes:

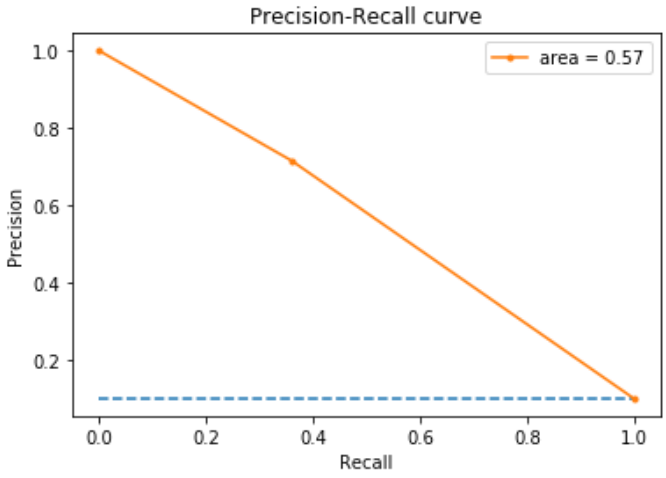
Naive Bayes model is easy to build and particularly useful for very large data sets. Naïve Bayes is Efficient, not biased by outliers, works on nonlinear problems, probabilistic approach. Based on the assumption that features have same statistical relevance.

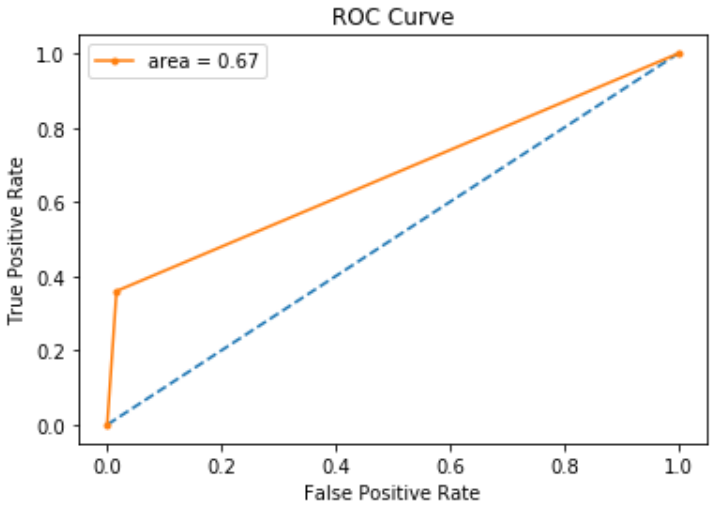
Code in Python:

		Predicted	
		0	1
Actual	0	52979	877
	1	3932	2212

- **True Positive:** 2212
- **True Negative:** 52979
- **False Positive:** 877
- **False Negative:** 3932
- **Accuracy:** 91.98%

- **Recall:** 36.00%
- **Precision:** 71.60%
- **False Negative Rate:** 63.99%
- **AUC-ROC Curve:** 67.18%
- **AUC-PR Curve:** 57.00%

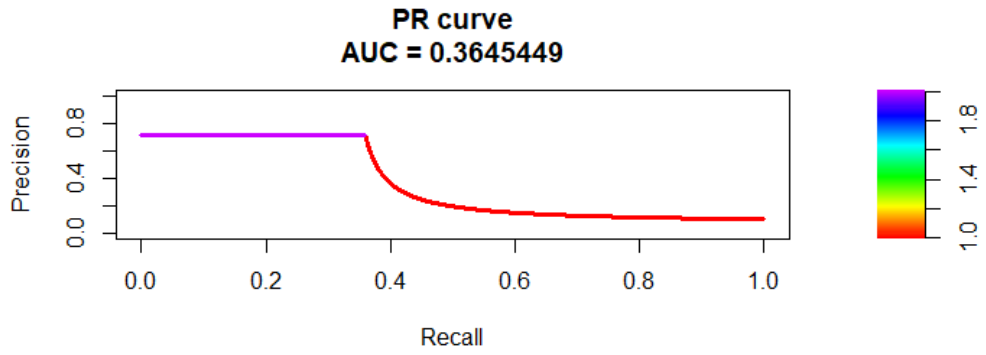


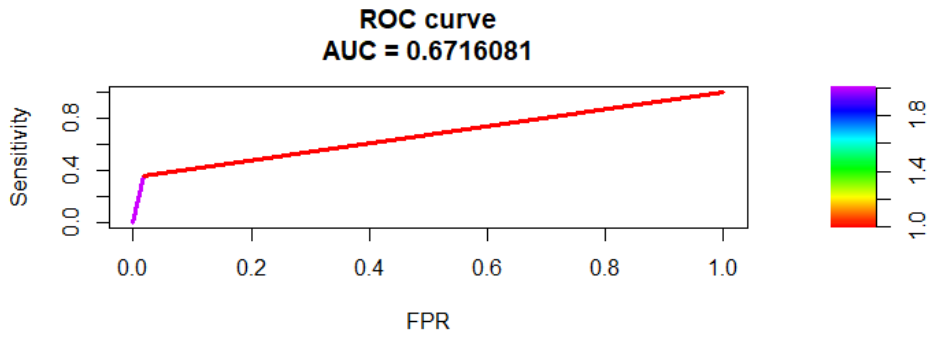


Code in R:

		Predicted	
		0	1
Actual	0	53087	884
	1	3861	2168

- **True Positive:** 2168
- **True Negative:** 53087
- **False Positive:** 884
- **False Negative:** 3861
- **Accuracy:** 92.08%
- **Recall:** 35.96%
- **Precision:** 71.03%
- **False Negative Rate:** 64.04%
- **AUC-ROC Curve:** 67.16%
- **AUC-PR Curve:** 36.45%





Result:

Naïve Bayes has the Highest Accuracy in all the Models we have implemented till now with High AUC-ROC, Precision and Recall as compared to other Models and Low False Negative Rate.

Conclusion:

A classification problem on an unbalanced dataset with no missing values. Variables are not Correlated with each other. After implementing various Machine Learning algorithm selected Naïve Bayes for this problem statement as it gave best Accuracy and Highest Precision, Recall and AUC-ROC among all Models with Lowest False Negative Rate. Test dataset is predicted using this model. We predicted that out of 200000 customers only 7627 will make transaction in future.

Visualizing Predicted Values of test: