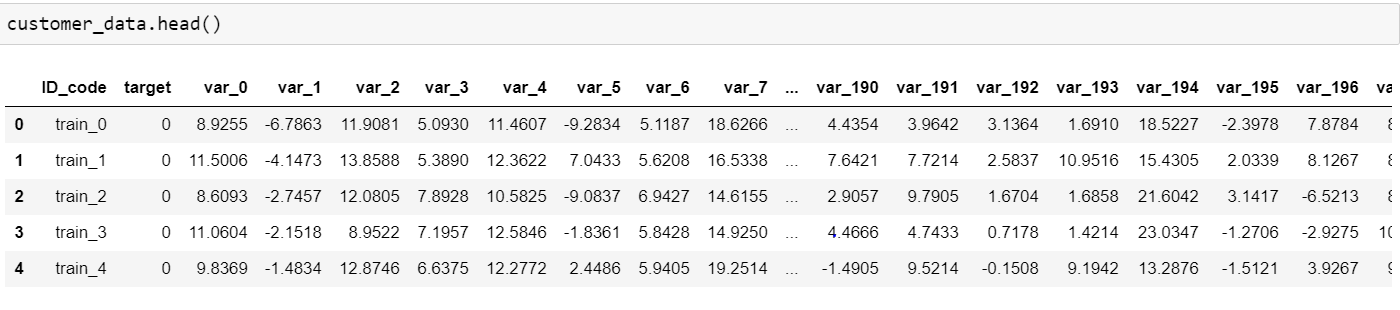
**Santander Customer Transaction Prediction**

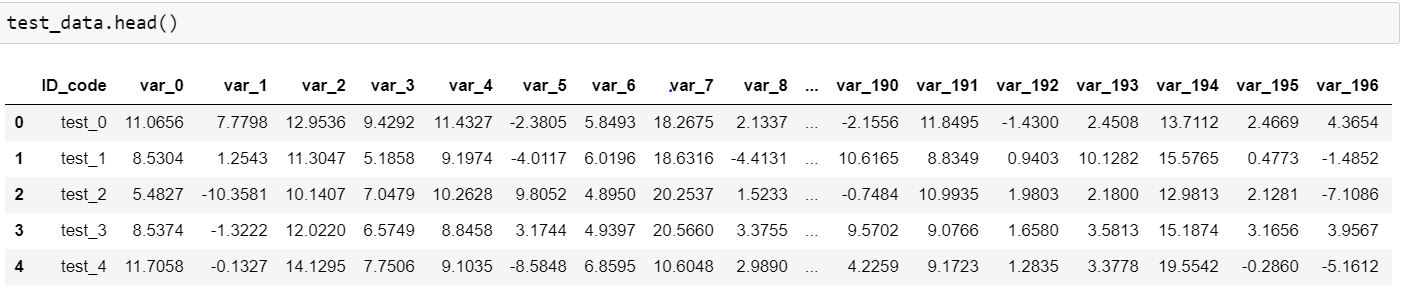
In Santander Customer Transaction Prediction ,we have a binary classification task. In this challenge, we need to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted.

**1) Data**

We are provided with Train and test data which have 200k samples each and we have 200 anonimyzed numerical columns**.**

A look at the training and testing data:



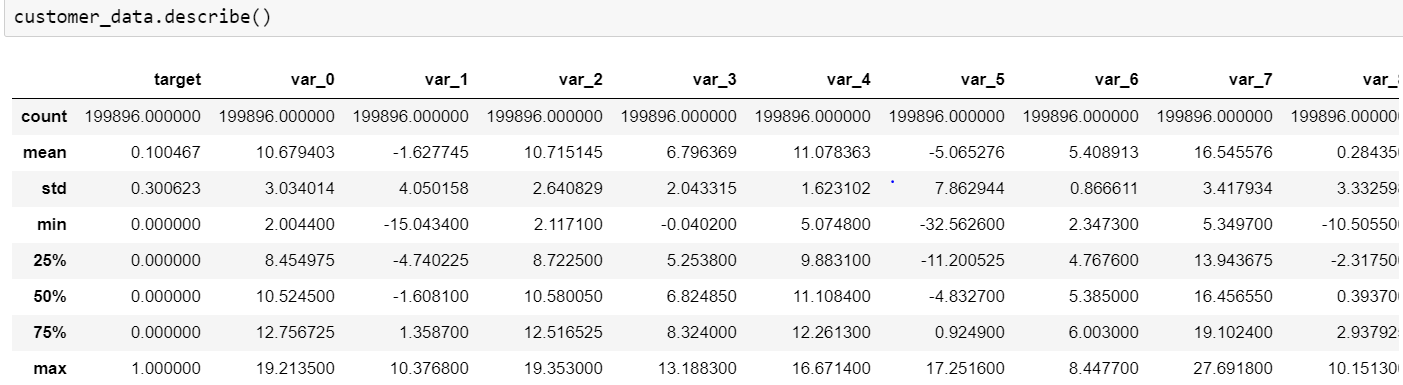
Testing data:  


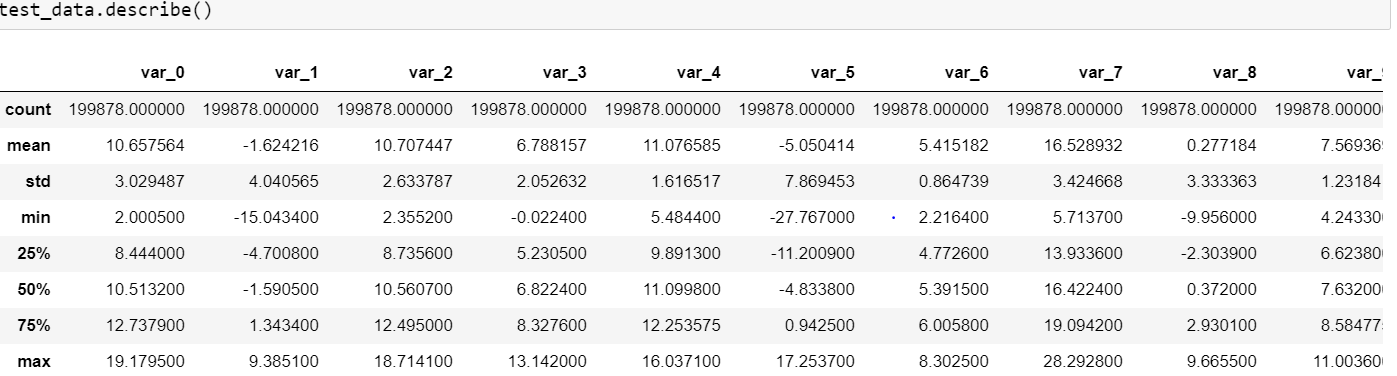
We have 200 numerical variables in both sets , named from var\_0 to var\_199 and one **target, ID column**.

**2) EDA:**

We take a look at the data provided to us by organizing ,plotting and summarizing the data.  
By doing same, we can get an idea how the data is distributed and is there any pattern observed in data.

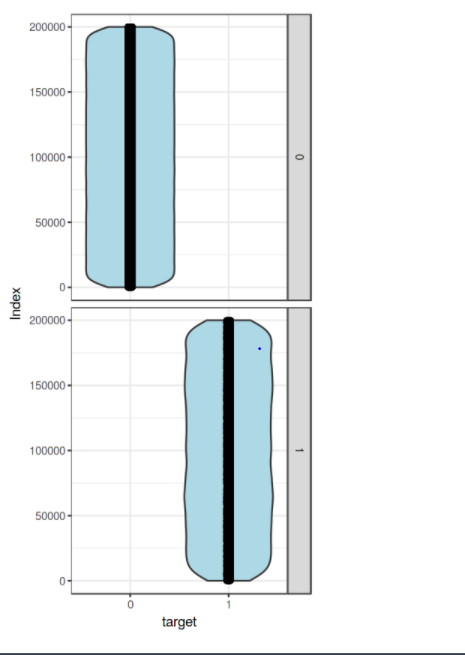
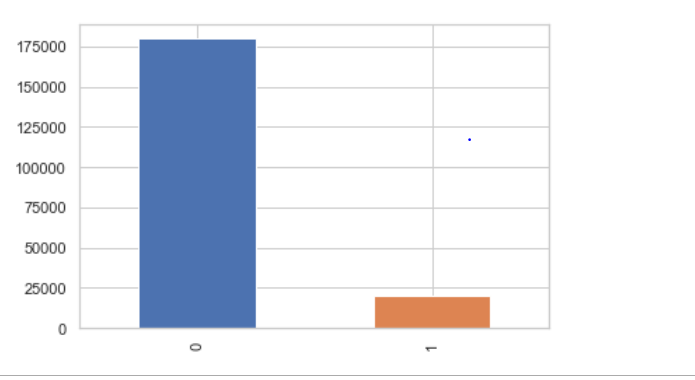
**2.1) Describe the data:**

* We use .describe() in python to get a look at the numerical data composition.



* Standard Deviation in both train and test is quite significant.
* Mean and other measures are close.

**2.2) Target Distribution:**

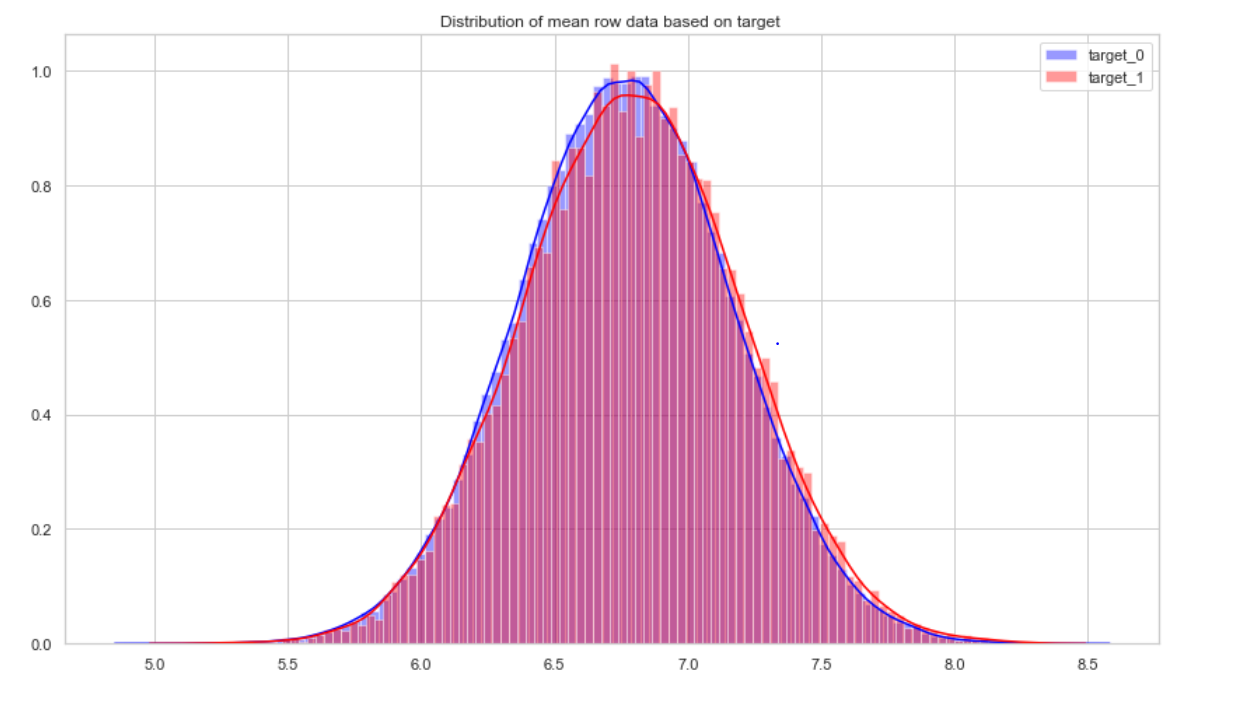
We visualize the **Target** distribution:  


**Observation:**

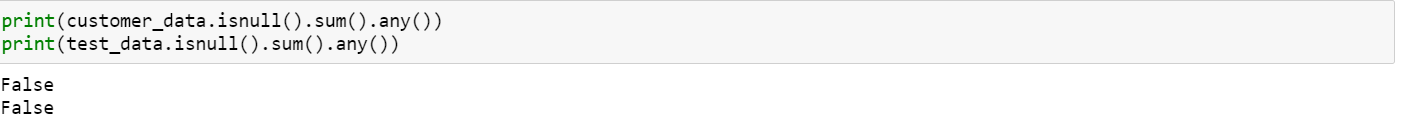
-We are having a unbalanced data, where 90% of the data is no. of customers who will not make a transaction & 10 % of the data are those who will make a transaction.

-From the violin plots, it seems that there is no relationship between the target a nd index of the data frame, it is more dominated by zero compare to one's.

-From the jitter plots with violin plots, we can observe that target looks uniform ly distributed over the indexes of the data frame.

**2.3) Distribution:**Get an idea of this data distribution, we review in the training dataset that we will work with, we review the histogram of the mean values of each record based on the binary target variable.  
  
As we can see that there is a small variation in the mean of all feature that could explain the target variable.

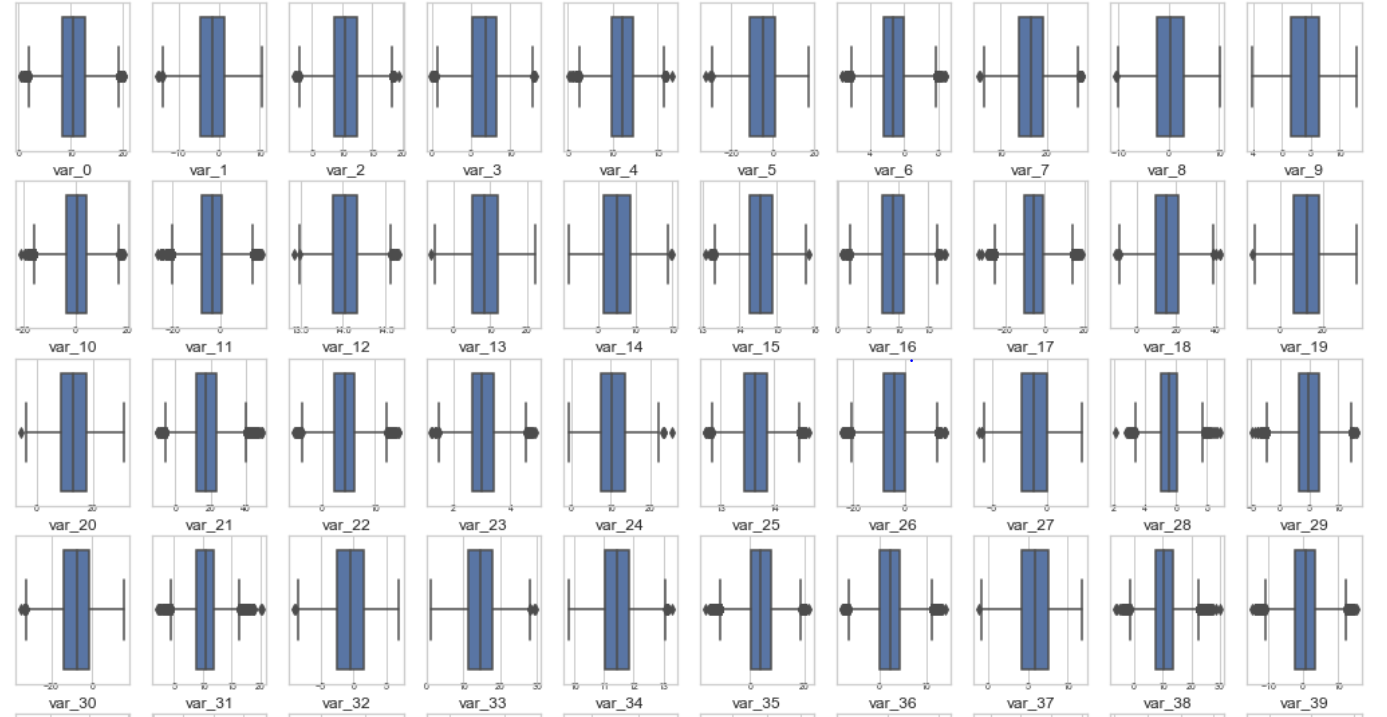
**2.4) Missing Value Analysis:**

We look for possible null values in the dataframe and if found any we will be filling them if number is significant.  


As we can see there is no missing values in both train as well as in test data.

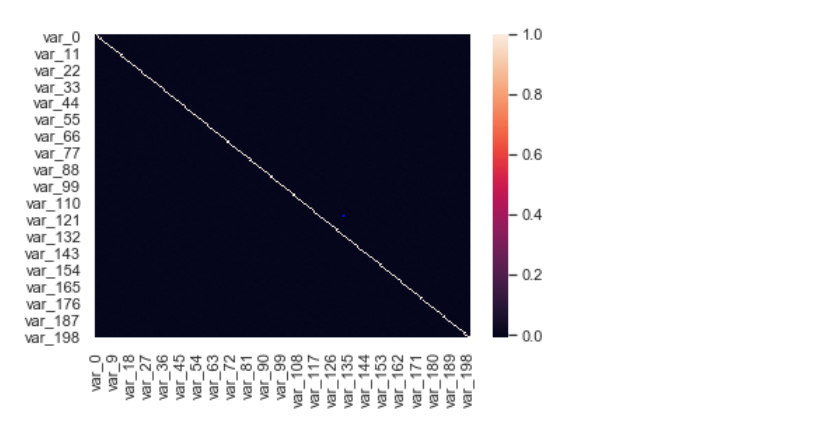
**2.5) Outlier Analysis:**

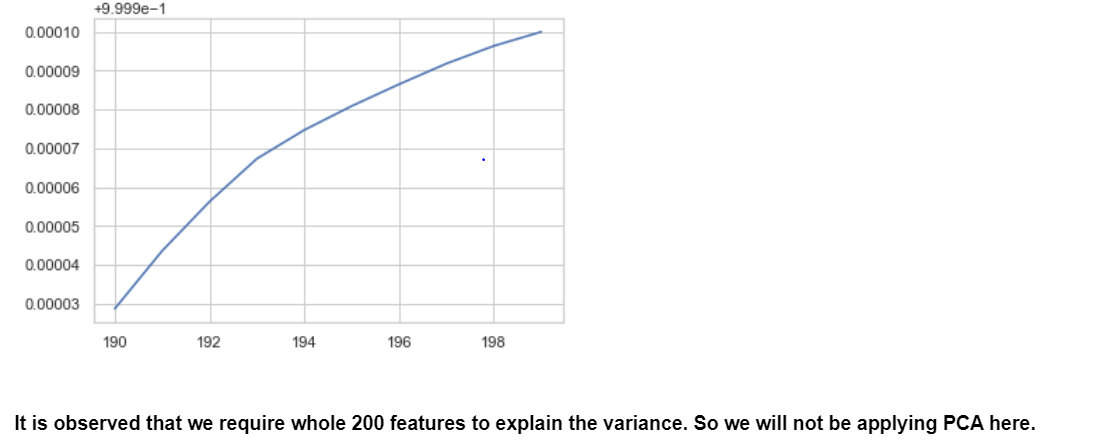
Outliers in the data may occur due to poor measurement quality or some external reasons. As they may effect in our prediction modelling we have to deal with it. In a simple way we can detect outliers by plotting box plots of the different variables in the data set. We used boxplot method to identify the outliers. It helps by defining the upperlimit and lower limit beyond which any data lying is considered to be an outlier.



Although, I have only posted here till var\_39 but outliers are present in all columns. This can affect the model so we will be removing them.  
We calculate the 25 and 75 percentile , and found min and max , and remove all the points less than min and greater than max.

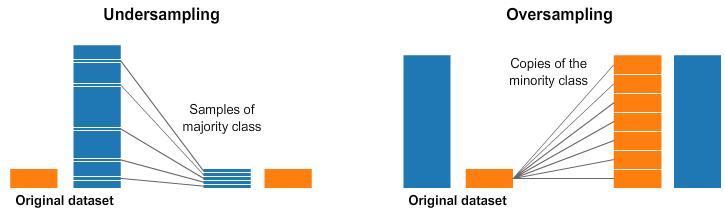
**2.5 Feature Selection:**

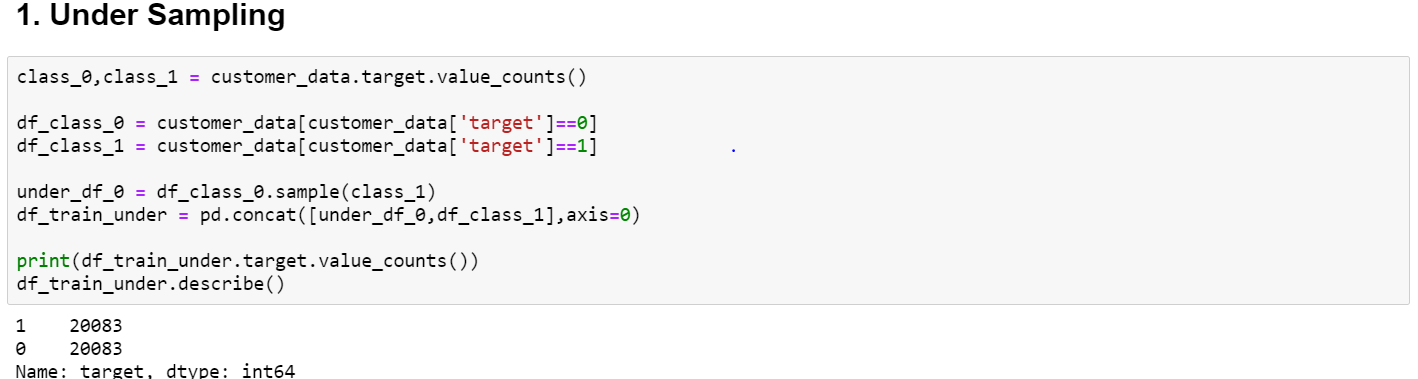
Feature selection is very important for modelling . Every dataset have unwanted and good features and both features affect the performance of model. In the classification modelling feature selection is about selecting the independent variables which will be helpful in predicting the target variable. It is also know as Dimensionality Reduction. For numerical data we can use correlation plot.  
  
  
  
From the colour of the graph we can see that there isn't much correlation between the variables. So, we have to keep all the 200 columns.

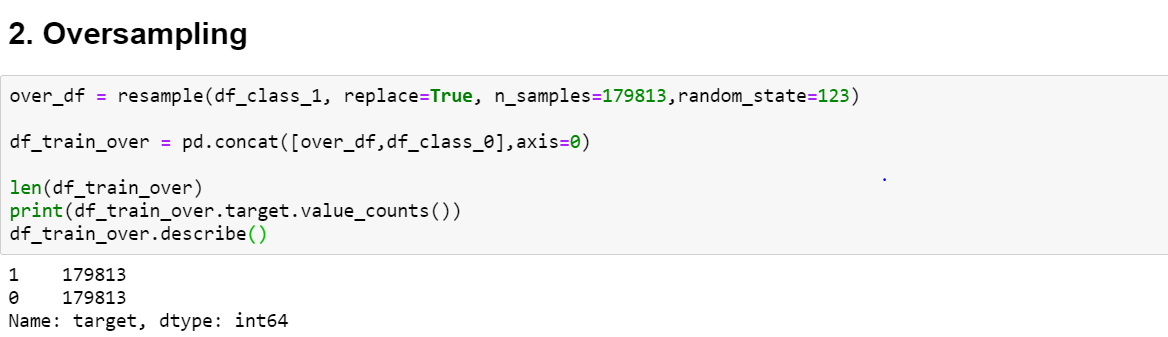
To re-confirm the same I also used PCA on dummy training dataset and plot variance ratio.  


**2.5 Unbalanced Data and Resampling:**

Note we are dealing with a data set very unbalanced, where there is only **10%** of records categorized with target 1, so those customers who have made a financial transaction.  
  
To develop a binary classification model we need to have more balanced data since most machine learning algorithms work best when the number of samples in each class is almost the same. This is because most algorithms are designed to maximize accuracy and reduce error, so we'll try to do this in this section before to predict models fit better.

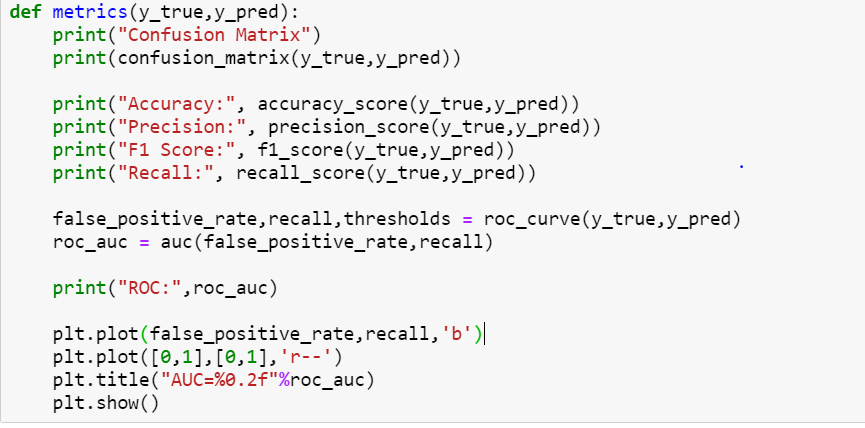
How we have a large dataset with 200,000 records we could undersampling in the data with the balanced target variable. Initially we will test a resampling in a 1:1 ratio but depending on the results we can use other proportions. Keep in mind that with undersampling we might be removing information that may be valuable. This could lead to a lack of fit and poor generalization of the test set.  
  




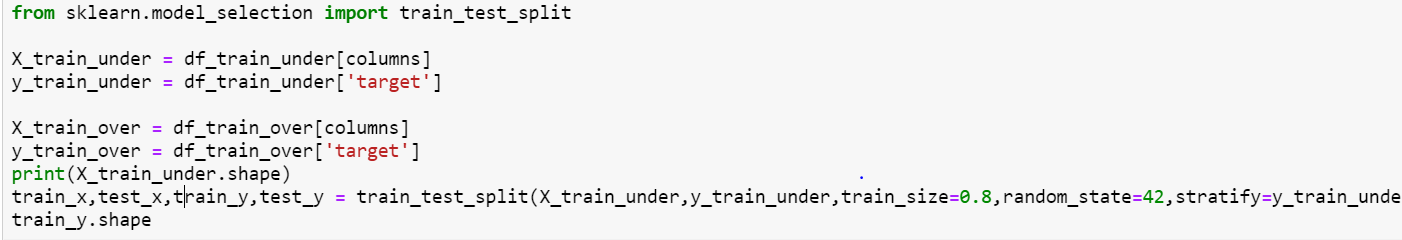
  
  
I have performed both under and oversampling and will be testing the models on both of these data to verify whether this makes any difference.  
Point to be not over here is that it might be possible that in oversampling some data gets replicated.

**3) Modelling:**

After all early stages of preprocessing, then model the data. So, we have to select best model for this project with the help of some metrics.

In order to automate the performance measures of the different models, we will factor a function to measure the metrics and be able to make comparisons between the different algorithms applied.  
  


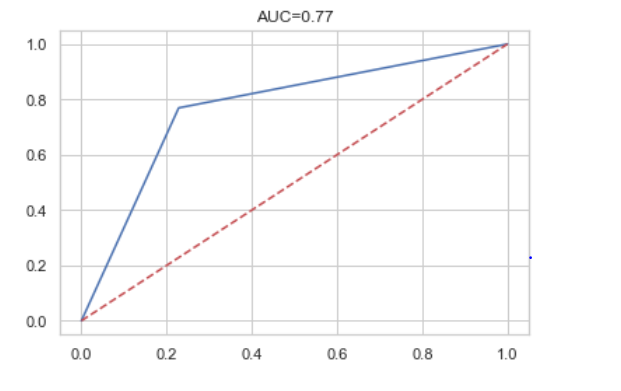
In modelling we first have to split the clean dataset to train-set and test-set and then develop different models and evaluate them by metrics.



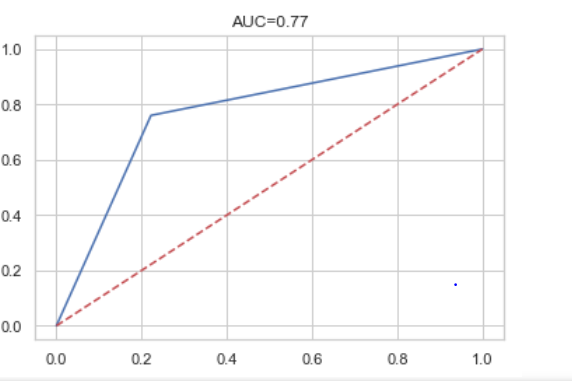
**3.1 Logistic Regression:**It is used to explain the relationship between one dependent binary variable and one or more independent variable. If the target variable is categorical variable then go for logistic regression. Logistic regression is used only for the classification model.The binary classification of events can be performed from a logistic regression model where the expression is used:

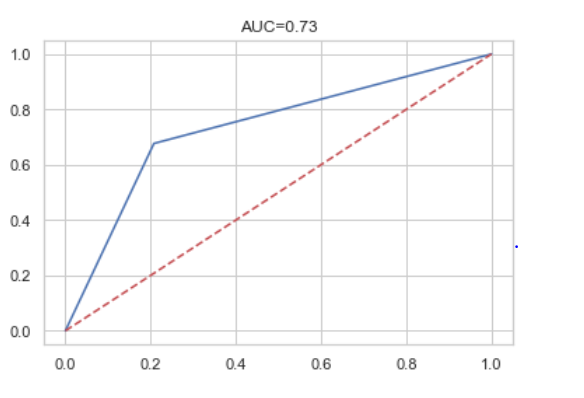
F(x)=1 / (1+e∑−wixiF(x))

In scikit-learn the constructor with which you can create a logistic regression model is **LogisticRegression**.  
We take this first model as a reference for its easy implementation and in which we can see how the other models behave.  
  
 **1) Training**

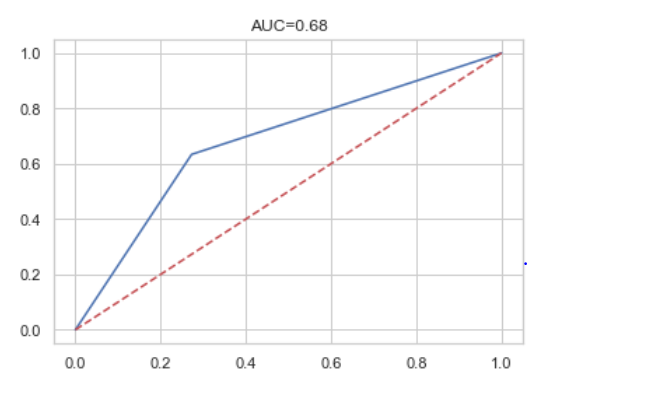


**2) Testing:**



We get a **medium performance** of the model but with very **low overfitting** between training and validation.  
We will take this performance as a base reference to compare it with other models.  
 **3.2) Random Forest:**We don´t test with the origin decision tree since being a dataset with discrete and non-categorical variables, it is difficult to achieve acceptable performance and stability, so it´s not good for unbalanced classification problems because the generated trees they will be very biased.In scikit-learn the constructor with which you can create a Random Forest model is RandomForestClassifier. This constructor requires more parameters than the decision tree because it is to be told the number of tree models to use, for which the parameter can be used n\_estimators. On the other hand, as selecting the data to be used for each submodel it is a good idea to fix the seed to ensure that the results are repeatable. With this in mind we can create a model for the resampled training and validation dataset.  
**UnderSampling:  
1) Training**

**2) Testing:**



**3.3) Naïve Bayes:**

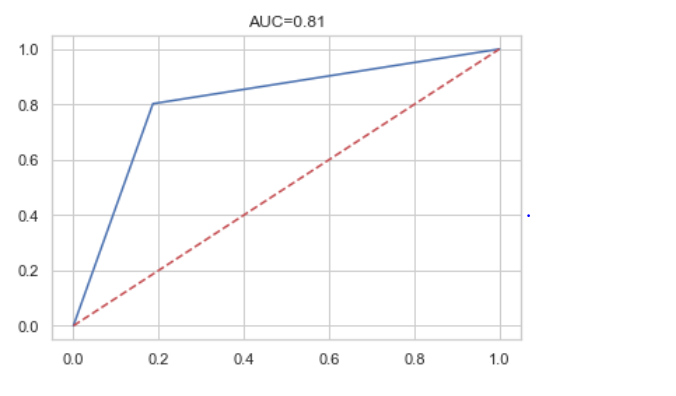
Naive Bayes classifiers are a family of simple "probabilistic classifier" based on applying bayes theorm with strong independence assumptions between the features. This is only use for the classification purpose. It is the one of the supervised machine learning algorithm which works based on the probability. Basically this allow us to predict a class for a set of features or predictors using the probability. So it is called as probabilistic algorithm for classifier.

Naive Bayes used for below conditions  
-> if the dataset is too large  
-> if all the categorical parameters are independent

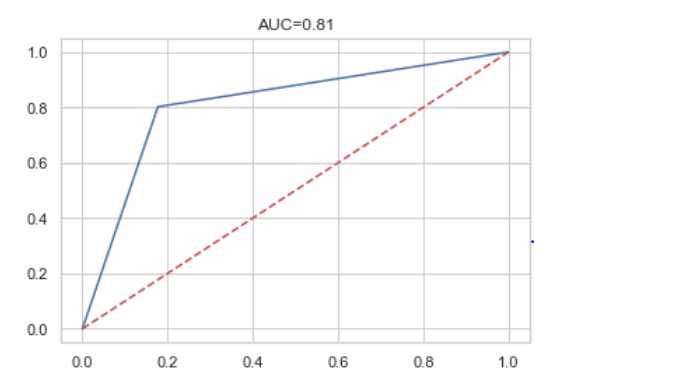
Although Naive Bayes has three variants

1. GaussianNB : used for normal classification
2. BernoulliNB & MultinomialNB : these two are used for text claasification.

**UnderSampling:****1) Training**



**2) Testing:**



**4) Result:**

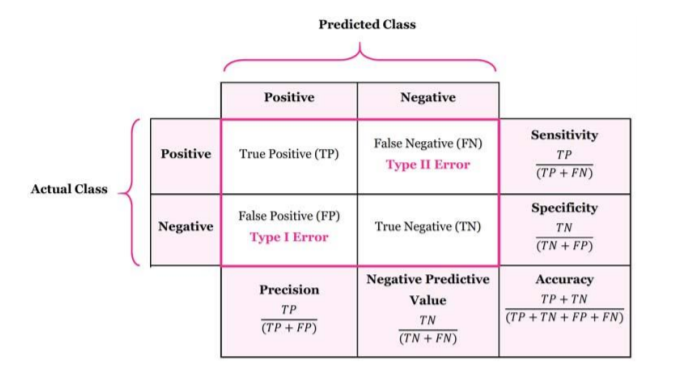
Now, we have a three models for predicting the target variable, but we need to decide which model better for this project. There are many metrics used for model evaluation. Classification accuracy may be misleading if we have an imbalanced dataset or if we have more than two classes in dataset.

For classification problems, the confusion matrix used for evaluation. But, in our case the data is imbalanced. So, roc\_auc\_score is used for evaluation.

In this project, we are using two metrics for model evaluation as follows:

1. **Confusion Matrix:**

In machine learning, confusion matrix is one of the easiest ways to summarize the performance of your algorithm. At times, it is difficult to judge the accuracy of a model by just looking at the accuracy because of problems like unequal distribution. So, a better way to check how good your model is, is to use a confusion matrix.



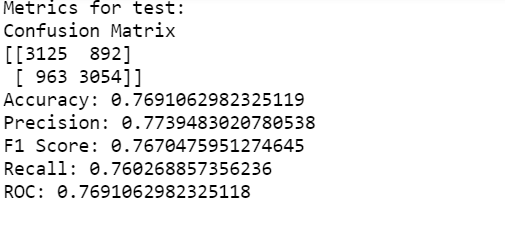
1. **Receiver operating characteristics (ROC)\_Area under curve(AUC) Score:**

It is a metric that computes the area under the Roc curve and also used metric for imbalanced data. Roc curve is plotted true positive rate or Recall on y axis against false positive rate or specificity on x axis. The larger the area under the roc curve better the performance of the model.

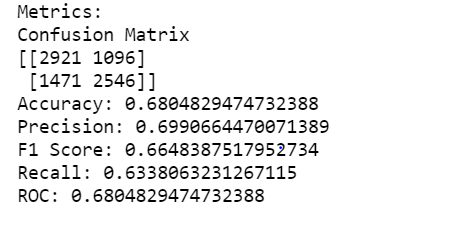
Now we will see the metrics for each models:

**Undersampling:**

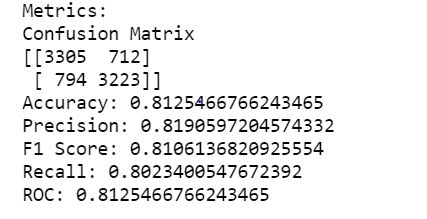
1. **Logistic Regression:**



1. **Random Forest:**

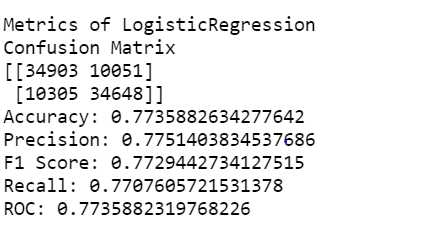


1. **Naïve Bayes:**

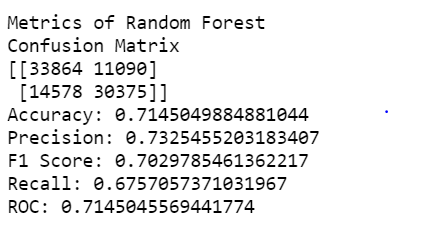


**Oversampling:**

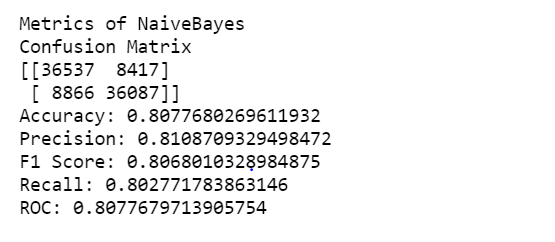
1. **Logistic Regression:**



1. **Random Forest:**



1. **Naïve Bayes:**



We can see that Naïve Bayes in both Undersampling and Oversampling is outperforming rest of the models. So ,we will be using Naïve Bayes model to predict the final values.

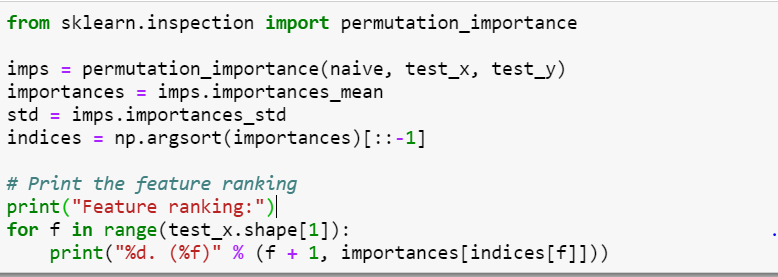
**5) Feature Importance:**

Feature importance is the methodology of scoring based on how much effective they are in predicting target variable.

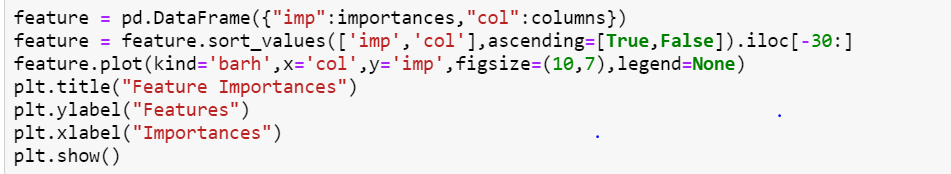
The naive bayes classifers don't offer an intrinsic method to evaluate feature importances. Naïve Bayes methods work by determining the conditional and unconditional probabilities associated with the features and predict the class with the highest probability. Thus, there are no coefficients computed or associated with the features you used to train the model (compare with its documentation).

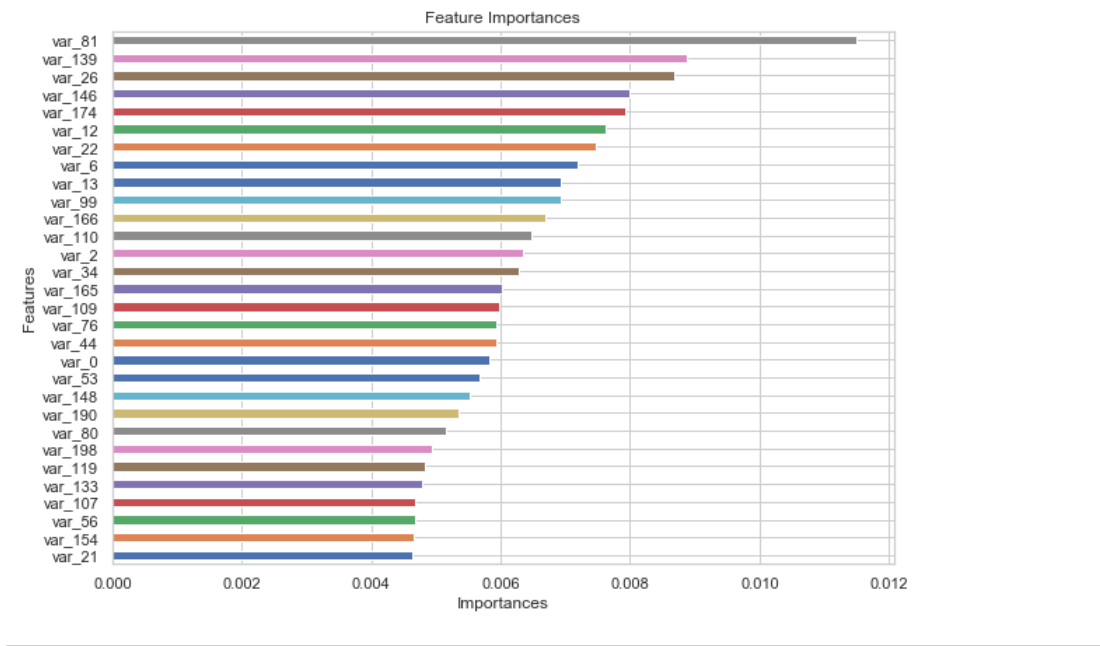
That being said, there are methods that you can apply post-hoc to analyze the model after it has been trained. One of these methods is the Permutation Importance and it, conveniently, has also been implemented in scikit-learn.

Permutation feature importance is a model inspection technique that can be used for any fitted estimator when the data is tabular. The permutation\_importance function calculates the feature importance of estimators for a given dataset. The n\_repeats parameter sets the number of times a feature is randomly shuffled and returns a sample of feature importances.



Plotting the top 30 features:





**SUMMARY**

This project can help the companyto help us identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted.

With classification, banks can classify potential customers and assign them with significant future value in order to invest company resources on them.

Some of the various ways in which classification can helps the banking institutitute are –

* Identification of customers based on their profitability.
* Segmenting customers based on their usage of banking services.
* Strengthening relationships with their customers.
* Providing appropriate schemes and services that appeal to specific customers.
* Analyzing customer segments to implement and improve services.