**LOAN DEFAULTER SYSTEM**

**INDUSTRIAL TRAINING PROJECT REPORT**

**BY**

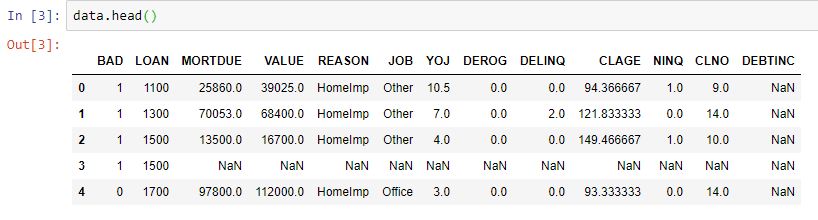
**SAYAN BANERJEE**

**DEPT : CSE  
ENROLLMENT NO : 12016002002185  
UNIVERSITY ROLL NO : 10400116102**

**DATA DESCRIPTION**

**Feature description:**

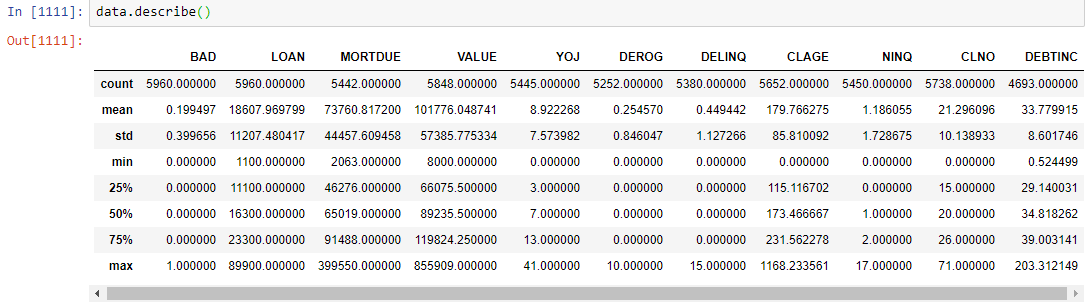
* “BAD”(Output Feature) : 1 = client defaulted on loan, 0 = loan repaid (Boolean)
* “LOAN” : Amount of the loan request (Number)
* “MORTDUE” : Amount due on existing mortgage (Number)
* “VALUE”: Value of current property (Number)
* “REASON”: “DebtCon” = debt consolidation “HomeImp” = home improvement (Categorical)
* “JOB” : Six occupational categories (Categorical)
* “YOJ” : Years at present job (Number)
* “DEROG” : Number of major derogatory reports (Number)
* “DELINQ” : Number of delinquent credit lines (Number)
* “CLAGE” : Age of oldest trade line in months (Number)
* “NINQ” : Number of recent credit lines (Number)
* “CLNO” : Number of credit lines (Number)
* “DEBTINC” : Debt-to-income ratio (Number)

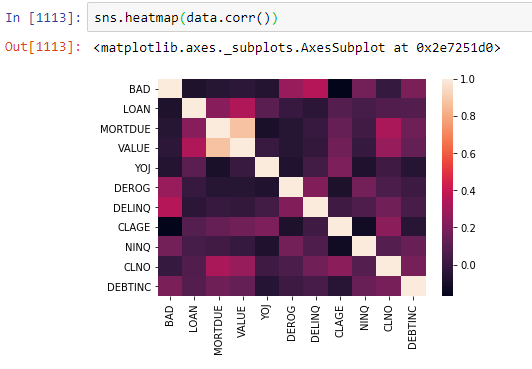
**First 5 rows of the dataset :  
Shape of data :**  
C:\Users\USER\Desktop\sc\shape.JPG

The dataset has 13 features with 5960 columns

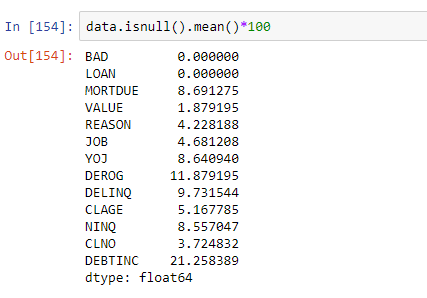
**EXPLORATORY DATA ANALYSIS**

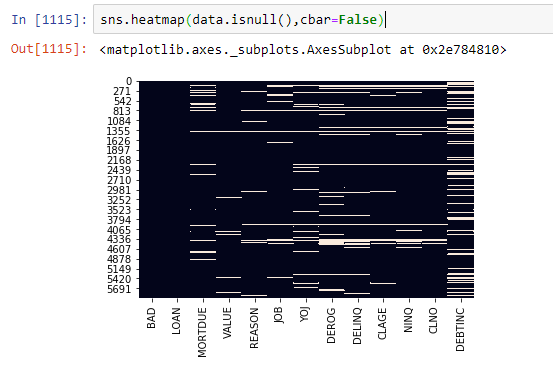
**Exploratory data analysis** (**EDA**) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

  
**Data description:**

  
**Correlation Heatmap:**

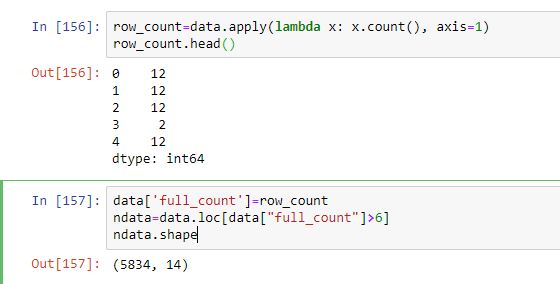
**MISSING VALUE TREATMENT**

**Percentage of missing values** **:**  


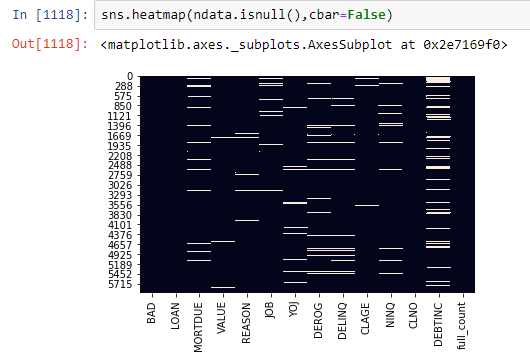
**Heatmap for missing values :**

In this heatmap we can see that there are certain rows in which more than half of the features contain missing values. We will drop those rows.

To drop those rows we have calculated the the number of not null values in each row. Only those rows which have more than 50% missing values are kept in the dataset.

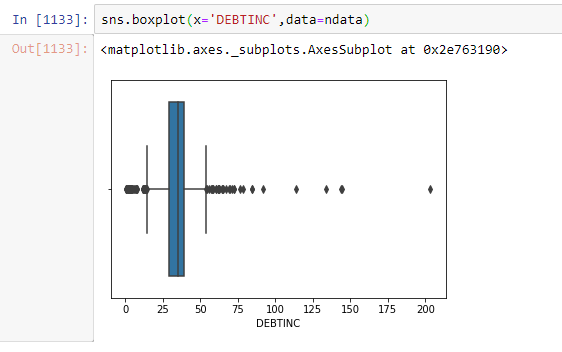


**Heatmap after dropping the rows :**



The dataset still has missing values which we can fill by median or mean based on the presence of outliers.

For visualizing outliers in a particular feature we use boxplot from seaborn.  
  
**Boxplot of feature ‘DEBTINC’ :**



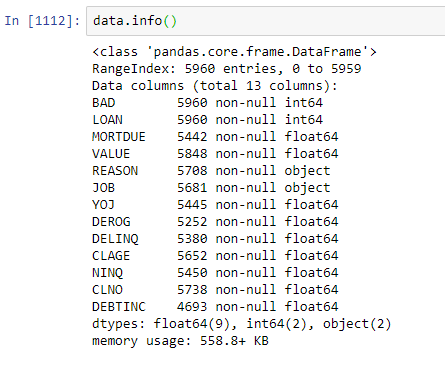
To find a value that can we fill in place of the missing values, we have created a function that will find the replacement using the mean of outliers and the mean of inliers.



This way we have replaced null values of all the continuous features of the dataset

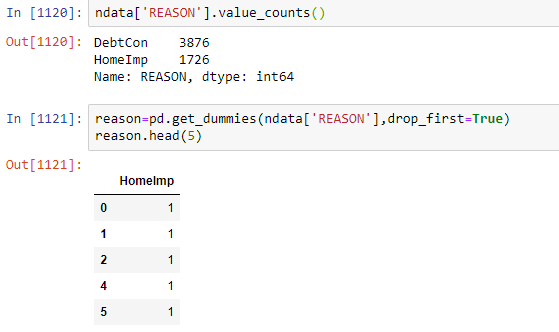
**TREATMENT OF CATEGORICAL VALUES**

A machine learning model can work on only numeric data. So, we have to convert the categorical features into numeric forms.

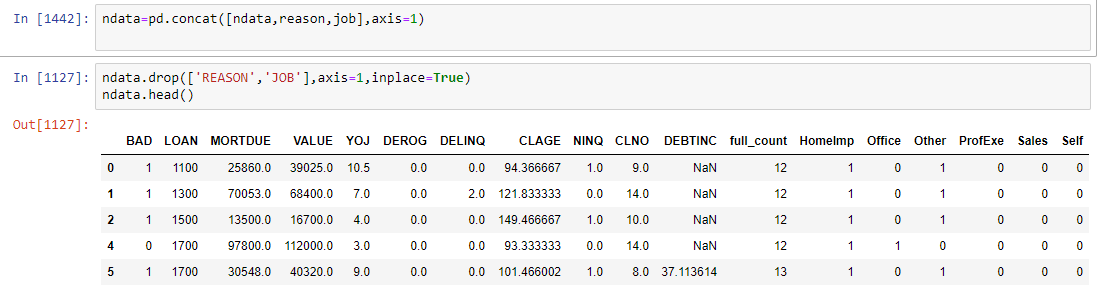
**Data information:**

There are two features with categorical values : “REASON” and “JOB”.

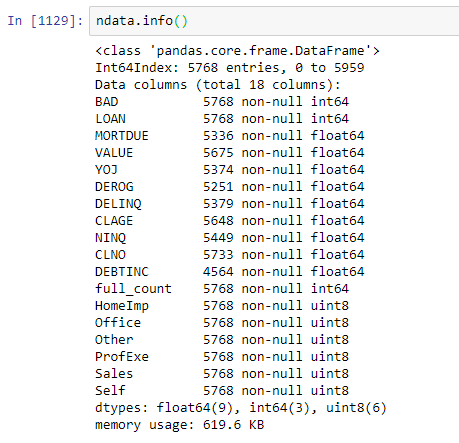
To convert them into numeric types we have used the get\_dummies function in pandas. For example, in the feature ‘Reason’ we can create dummy dataframe like this-



We did the same for other categorical features and then we have concatenated the dummy features with the actual dataset and have dropped the original features.

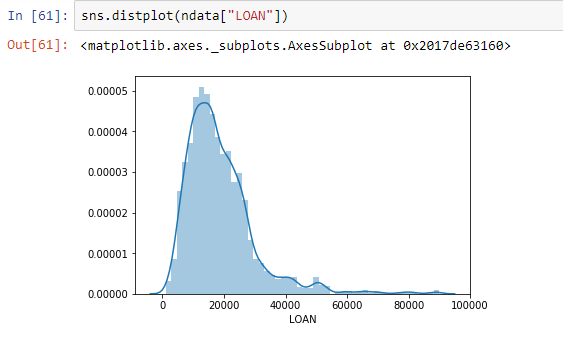


After treating the categorical values,

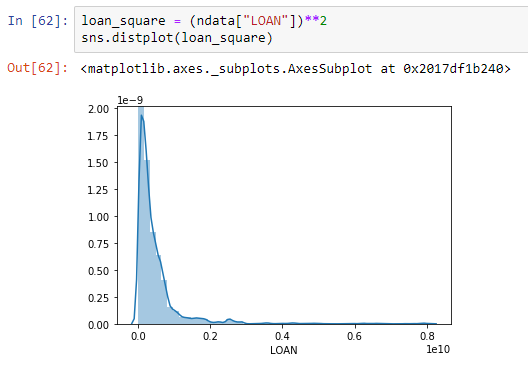


**TREATMENT OF SKEWED FEATURES**

We need to normalize the data as much as possible before giving it to our model. To do that we need to check the distribution of each feature and find out whether the graph is left skewed or right skewed. If the graph is left skewed, we can try to fix it by using squares and cubes. If the graph is right skewed, we can try to fix it by using squareroots, cuberoots, reciprocals or logarithms.  
  
For example, the distplot of the feature “LOAN” is



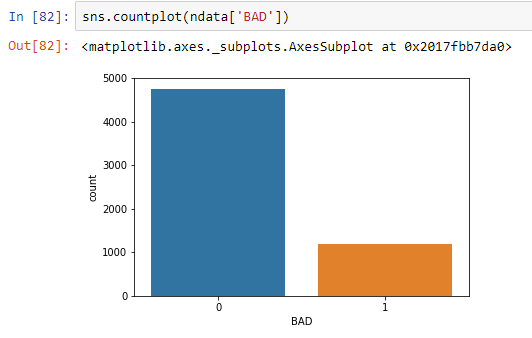
This feature is left skewed since the peak is shifted towards left.

After finding the square, the graph looks like  


**TREATMENT OF CLASS IMBALANCE**

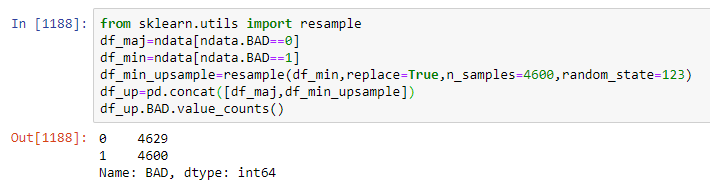
Class imbalance is a **problem** in machine learning where the total number of one **class** of data (positive) is far less or far more than the total number of another **class** of data (negative) in the output feature. If we don’t treat this, there might be a biasness in the model towards the class occurring more.

**Countplot of the feature “BAD” :**



There is a clear class imbalance in this feature where the number of 0s is far more than the number of 1s.

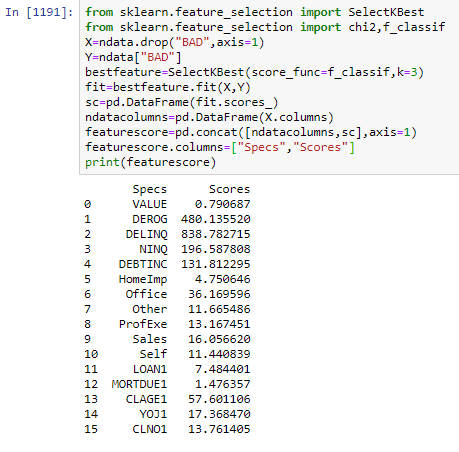
To solve this problem we can either increase the number of minority instances or decrease number of the majority instances with the help of resample method in sklearn.



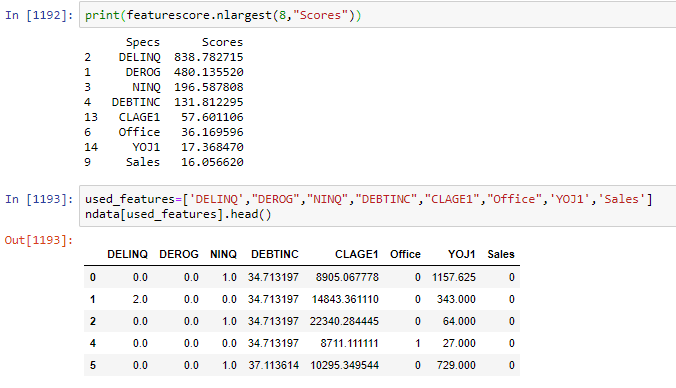
Here we have upsampled the instances with 1s in the output feature.

**FEATURE SELECTION**

We have selected the features that should be used for training the model with the help of their correlation and score with respect to the output feature.



To train the model we have chosen the first 8 features with highest score.

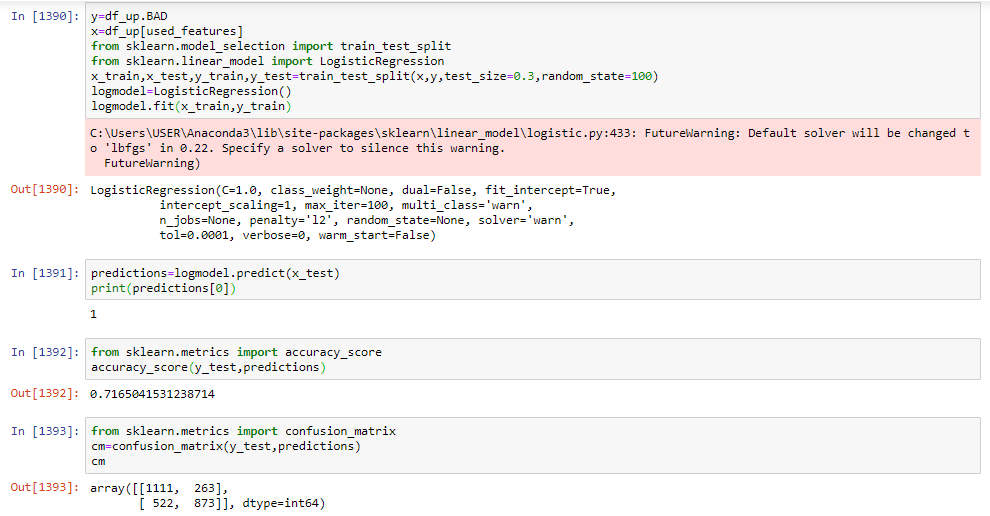


**MODEL BUILDING**

The classification algorithms used to build a predictive model for the output feature (“BAD”) of the dataset are as follows –

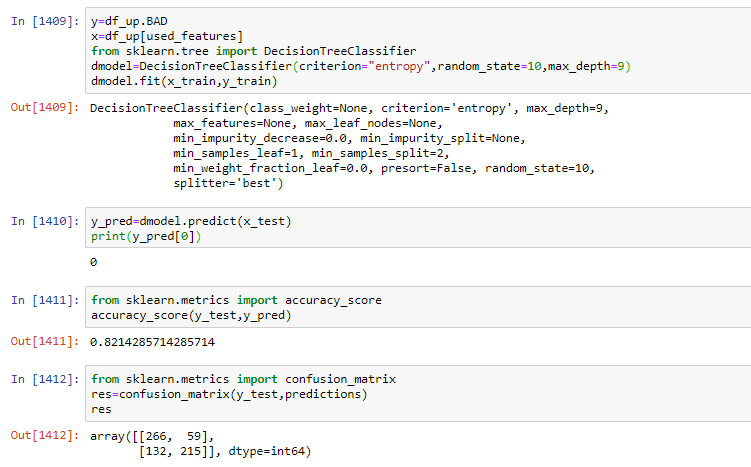
**1. Logistic Regression :**

**Logistic regression** is a statistical **model** that in its basic form uses a **logistic** function to **model** a binary dependent variable, although many more complex extensions exist. In **regression analysis**, **logistic regression** (or **logit regression**) is estimating the parameters of a **logistic model** (a form of binary**regression**).

****

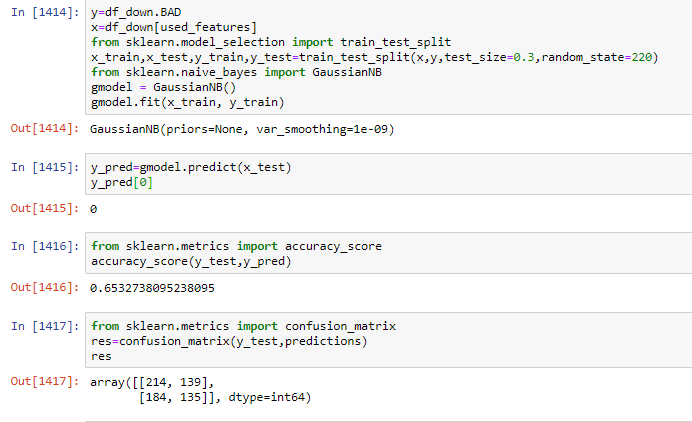
Logistic Regression gives us 71.65% accuracy

**2. Decision Tree :**

A **decision tree** is a **decision** support tool that uses a **tree**-like graph or **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.

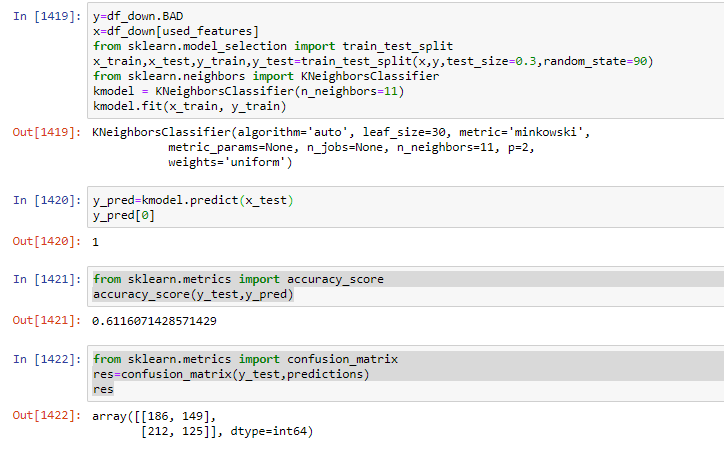
Decision Tree gives us 82.14% accuracy.

**3. Naïve Bayes Classifier** **:**

The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach you would come up with if you wanted to model a predictive modeling problem probabilistically.

Naïve Bayes classifier gave us 65.32% accuracy

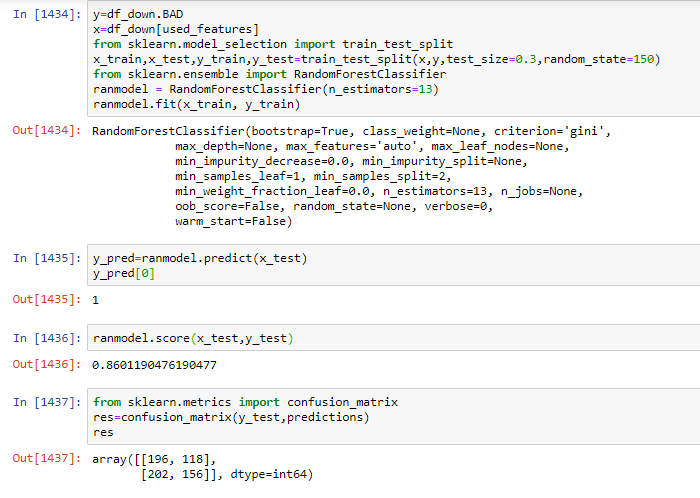
**4. K Nearest Neighbor :**

It **works** based on minimum distance from the query instance to the training samples to determine the **K**-**nearest neighbors**. The data for **KNN** algorithm consist of several multivariate attributes name that will be used to classify.

KNN classifier gives us 61.16% accuracy

**5. Random Forest Classifier :**

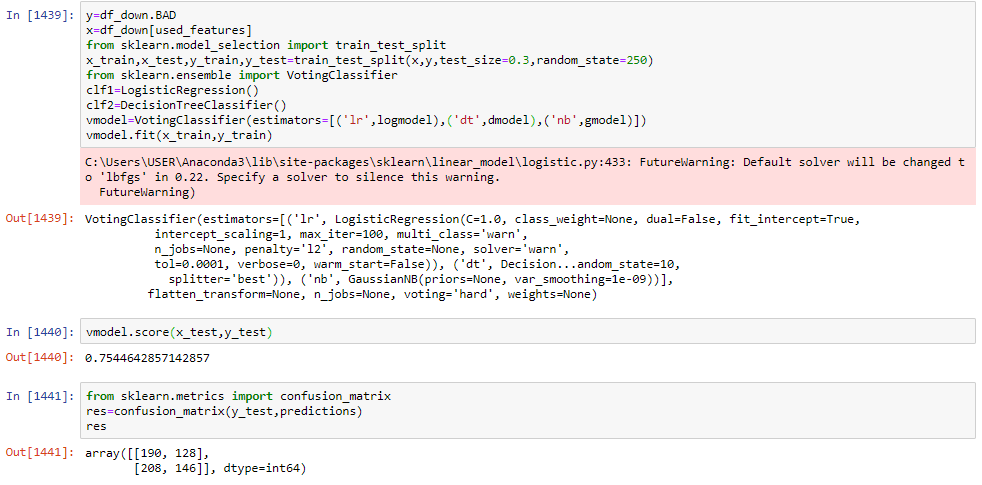
Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.



Random Forest classifier gives us 86.01% accuracy.

**6. Voting Classifiers :**

It is a form of ensemble learning where we use heterogenous learners and performance is the measure of their mean.



Here we have used Logistic Regression, Decision Tree and Naïve Bayes Classifiers

The accuracy is 75.44%

Among all the classification models, Random Forest gave us the best accuracy.

**MODEL EVALUATION**

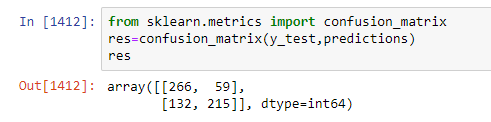
A **confusion matrix** is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It helps us to know that how many times the samples were correctly classified

It is represented as,

[[ True, +ve False, +ve ],

[ False, -ve True, –ve ]]

The diagonal spaces of True +ve and True–ve tells us how many times our samples were correctly classified. And the other ones tells us the number of times the algorithm messed up.



Like this, we have to compare the confusion matrix of all the other models and observe the best model whose number of times samples classified were much greater in number than the others.

**FUTURE SCOPE**

* The dataset can be cleaned more for obtaining a normal distribution curve in each feature.
* The predictive model can be improved by implementing more forms of ensemble learning with different values for splitting and random state.
* An application can be built which will take different features of a client as input and give the predicted output based the classification model.