**Presentation On Bank Lending Project**

**Presented By :**

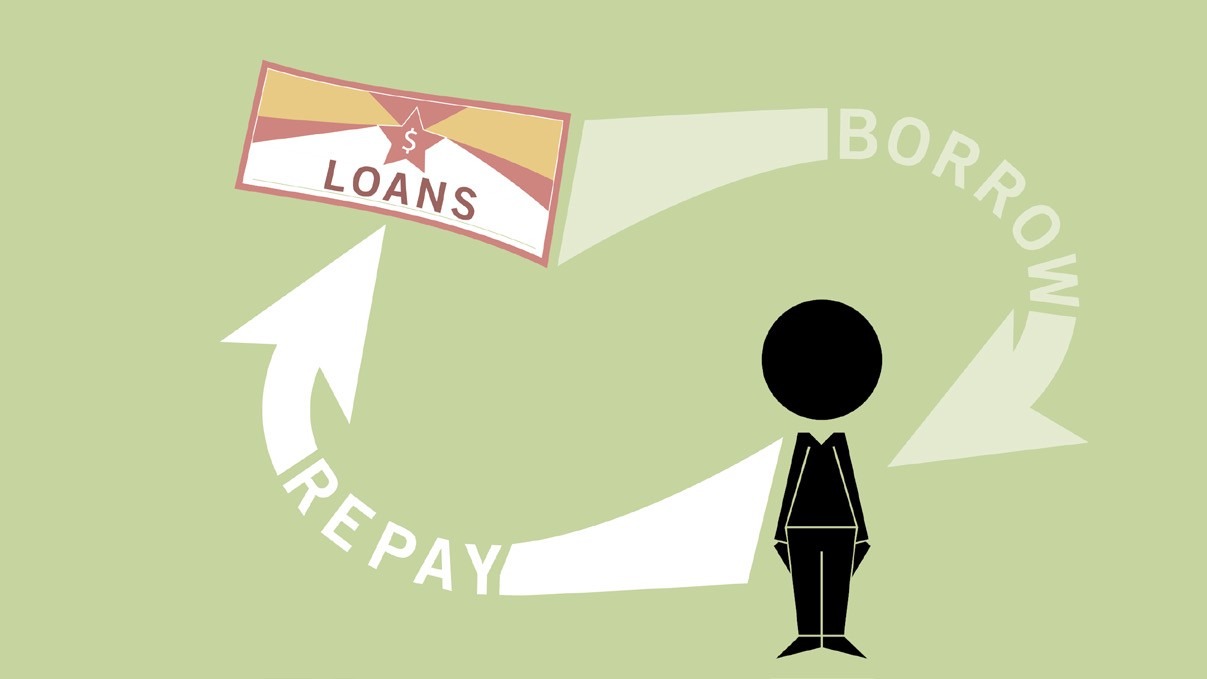
**1)Sanyoja Sagare** Batch DSP29

**2)Pooja Singh** Batch DSP29

**3)Ameya Soman** Batch DSP29

**4)Linotta Demel.N** Batch DSP29

**5)Ankit Potdar** Batch DSP27



**Introduction:**

The two most critical questions in the lending industry are:

1) How risky is the borrower?

2) Given the borrower’s risk, should we lend him/her?

Check for details of the customers in order to know if he will pay the loan or not

**Problem Statement:**

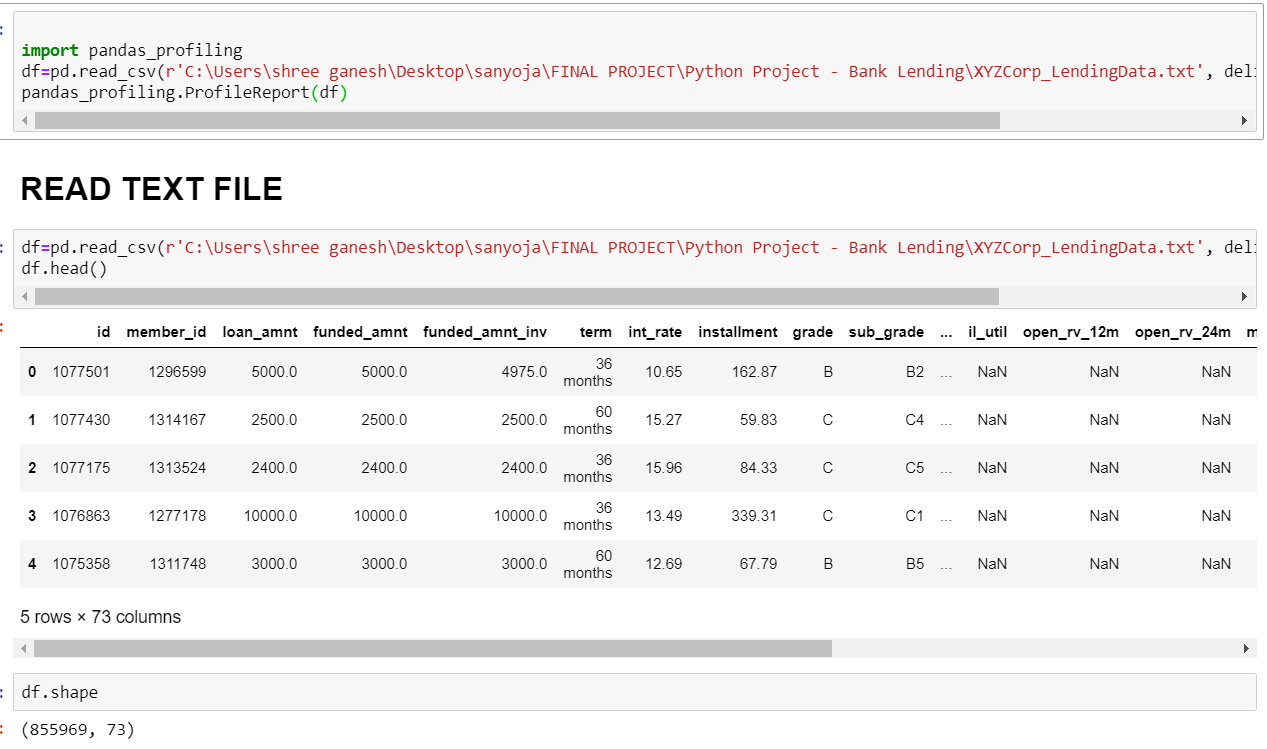
We have built a data model to predict the probability of default, and choose a cut-off based on what you feel is suitable.

**Steps:**

* Read data,
* Visualization
* include relevant columns

1. From where we got the data
2. The relevant variables selected and why we did not select the other variables

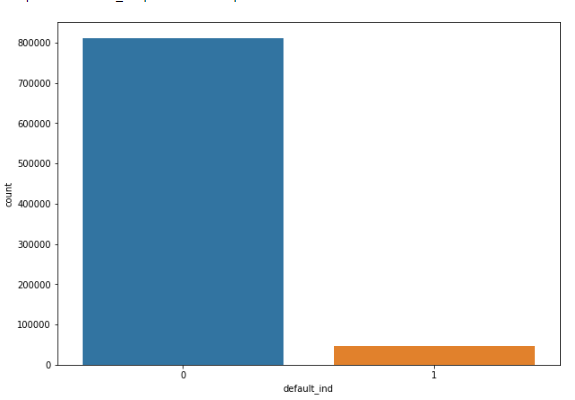
* Missing value imputation
* Label Encoding
* Split - train and cross validation sets
* Model Building
* Predict

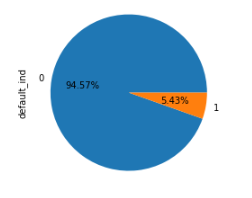


**Visualisation:**

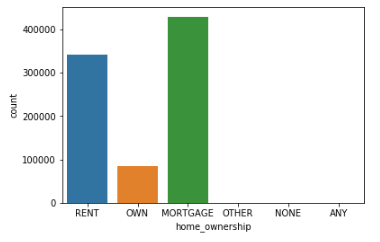
**Defaulter Information:**

* Total number of defaulters and non defaulters
* 0=non defaulter
* 1= defaulter

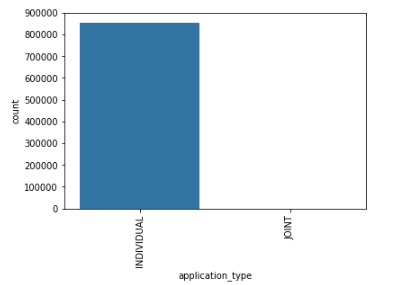




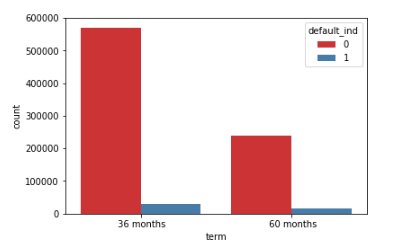
**Home ownership:**



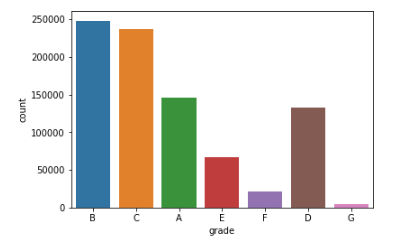
**Application Type:**



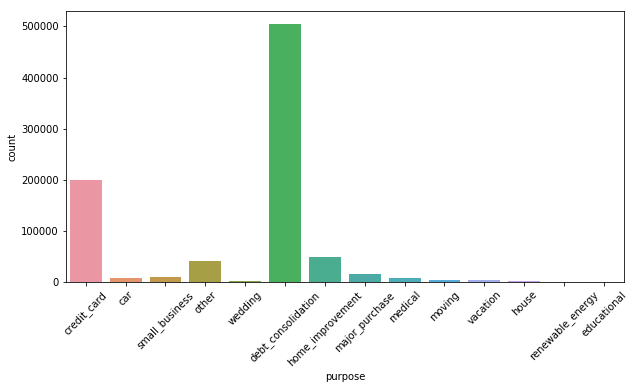
**Term:**



**Grade:**



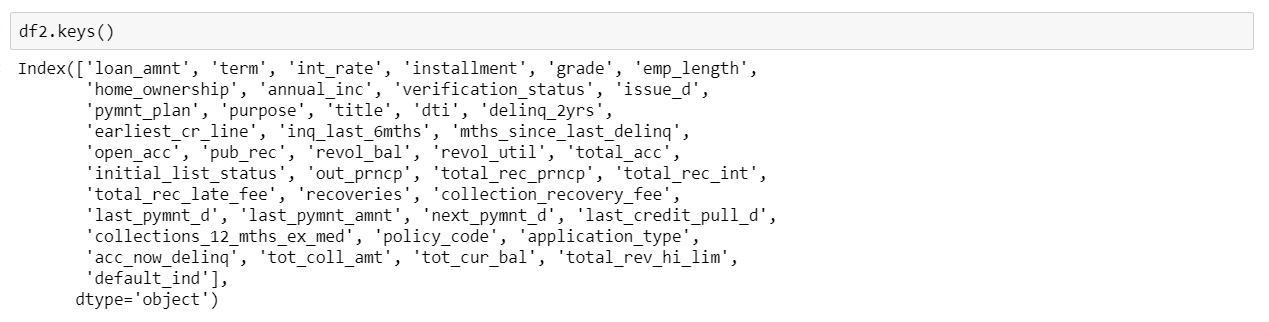
**Purpose:**

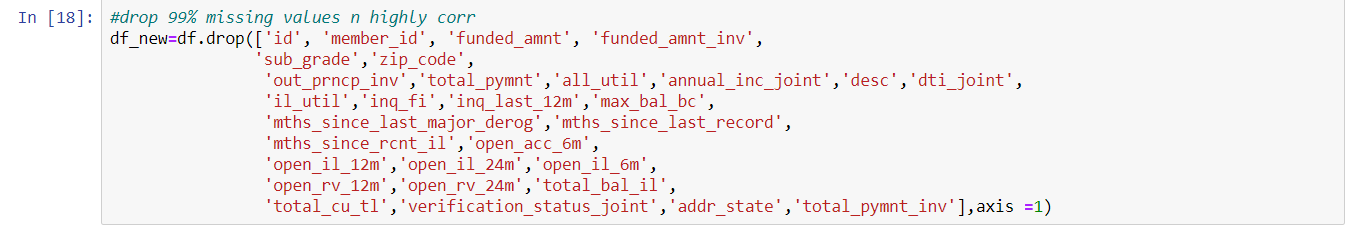


**Variable selection:**

The basis of selection of variables

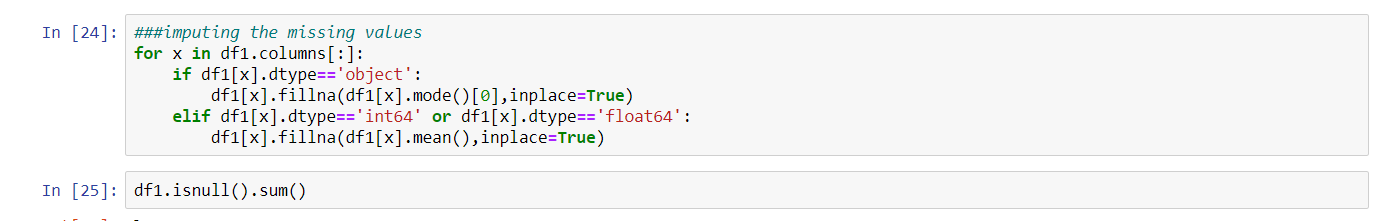
* Missing Values, High Correlation, Dropping From Joint Account.
* Selected variables



* Non selected varaibles

|  |  |
| --- | --- |
| Id | Mths\_since\_last\_major\_derog |
| Member\_id | Mths\_since\_last\_record |
| Funded\_amnt | Mths\_since\_rcnt\_il |
| Funded\_amnt\_inv | Open\_acc\_6m |
| Sub\_grade | Open\_il\_12m |
| Zip\_code | Open\_il\_24m |
| Out\_prncp\_inv | Open\_il\_6m |
| Total\_pymnt | Open\_rv\_12m |
| All\_util | Open\_rv\_24m |
| Annual\_inc\_joint | Total\_bal\_il |
| Desc | Total\_cu\_tl |
| Dti\_joint | Verification\_status\_joint |
| Il\_util | Addr\_state |
| Inq\_fi | Total\_pymnt\_inv |
| Inq\_last\_12m | Emp\_title |
| Max\_bal\_bc |  |

**Imputing Missing Values:**



* the ways of imputing: mode and mean

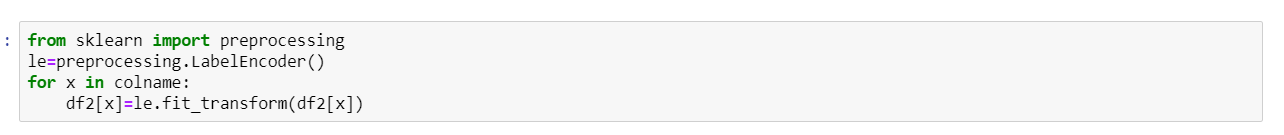
**Data Encoding:**

* **Manual Encoding**:



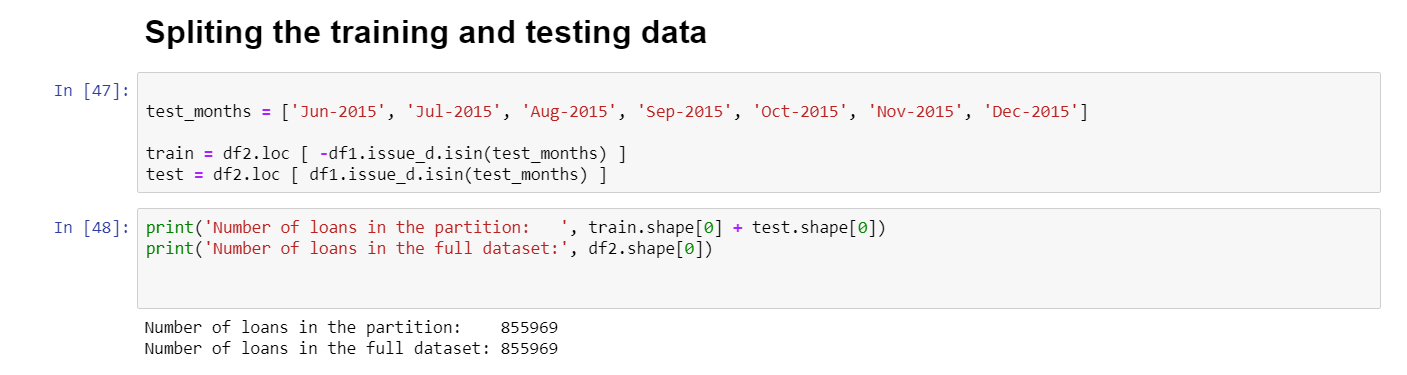


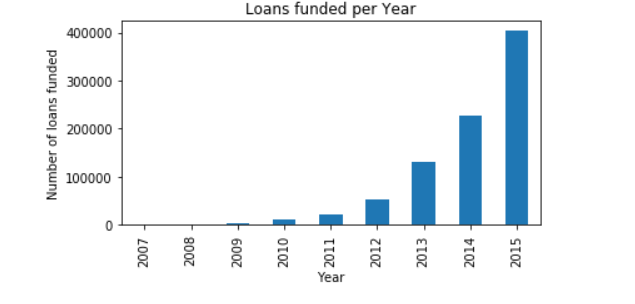
* **Label Encoding:**

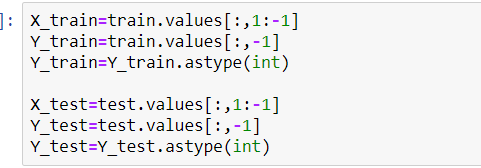


**Splitting Train and Test Data:**

* Basis of splitting the Data



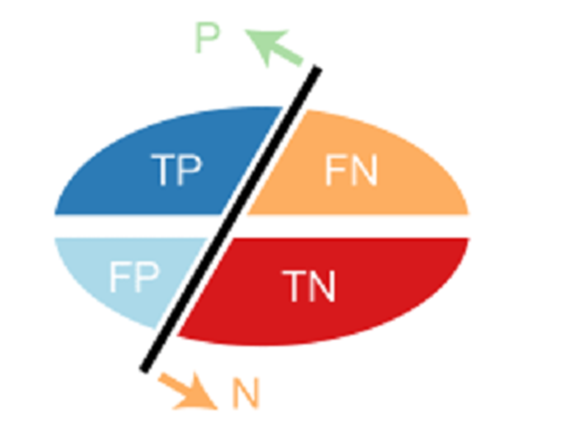
****

****

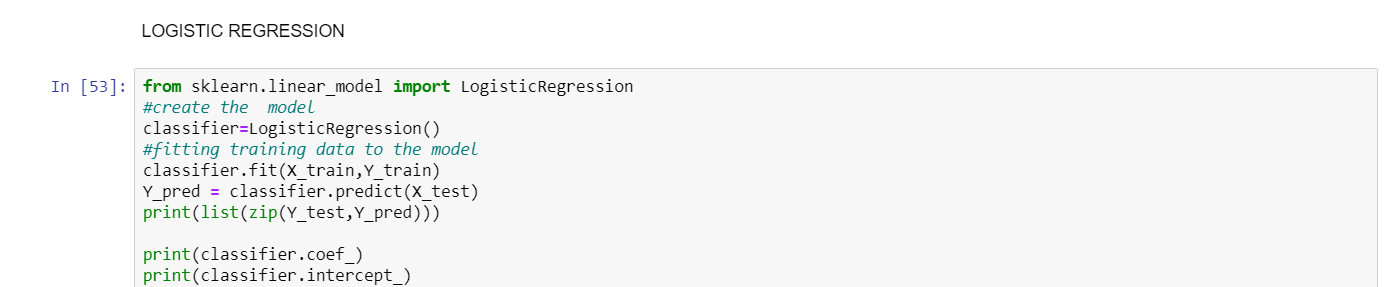
**Scaling:**

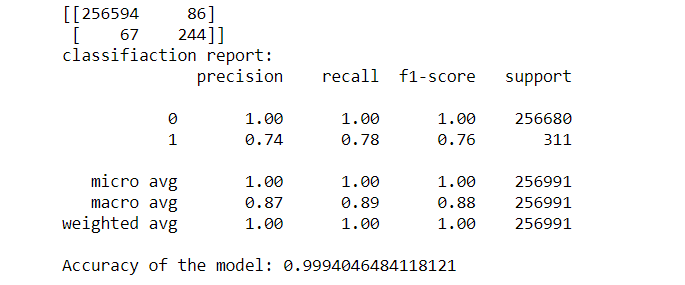
**We have our training and testing separate we put scalar on training data and the same scales are used to transform to both training and testing data**

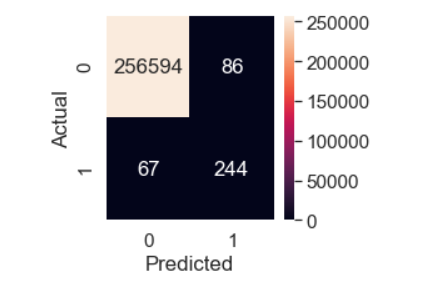
**Model Building:**

****

* **Logistic Regression:**





****

**‘0’ - Paid**

**‘1’-Not Paid (Defaulter)**

|  |  |  |
| --- | --- | --- |
|  | **0** | **1** |
| **0** | **TP** | **FN** |
| **1** | **FP** | **TN** |

Now,  
**Classification Rate/Accuracy:**  
Accuracy = (TP + TN) / (TP + TN + FP + FN)=

(256594+244) /(256594+244+84+68)= 0.99940854

**Recall:** Recall gives us an idea about when it’s actually yes, how often does it predict yes.  
Recall(0)=TP / (TP + FN)=256594/(256594+86)=1

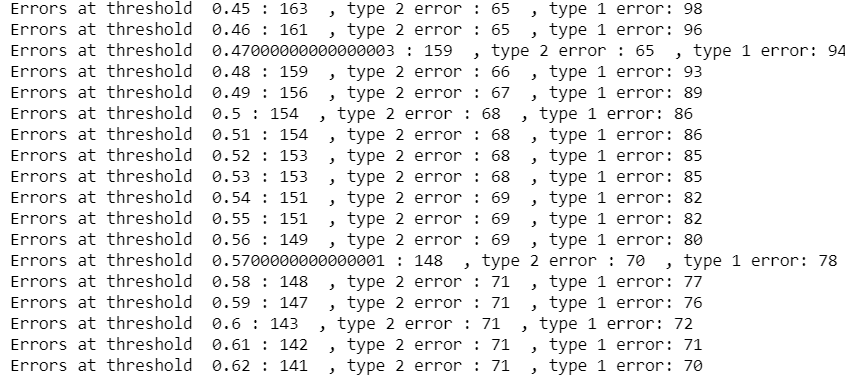
Recall(1)=TN / (TN + FP)=244/(244+67)=0.78

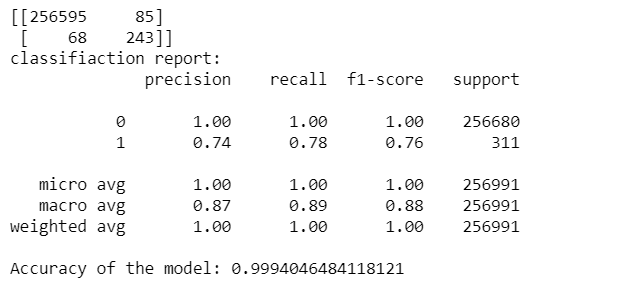
**Precision:** Precsion tells us about when it predicts yes, how often is it correct.  
Precision(0) = TP / (TP + FP)=256594/ (256594+67)=1

Precision(1)= TN / (TN + FN)=244/ (244+86)=0.74

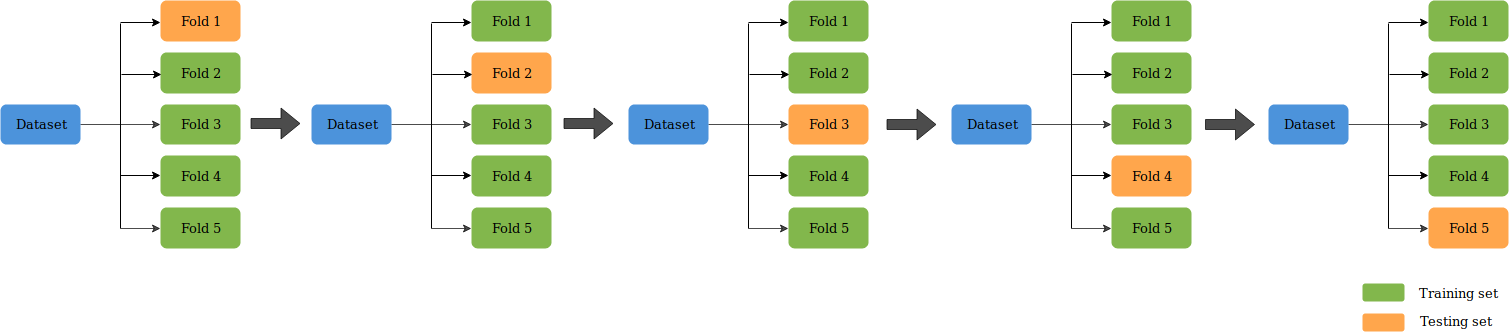
**F-measure:**  
Fmeasure=(2\*Recall\*Precision)/(Recall+Presision)=0.76

**With different threshold:**

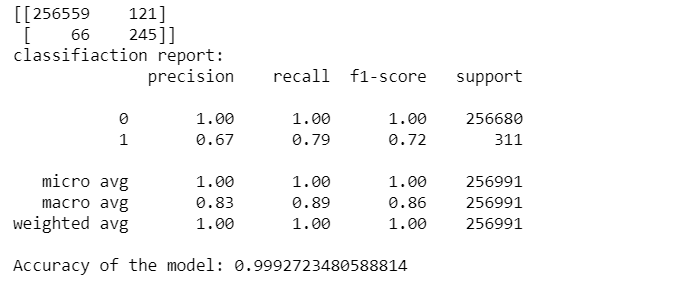




**Cross Validation:**

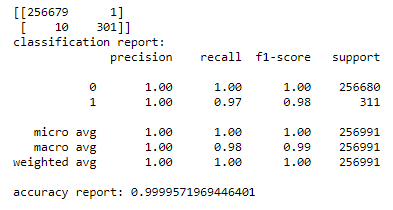






**Decision Tree:**

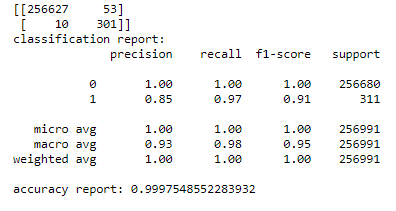




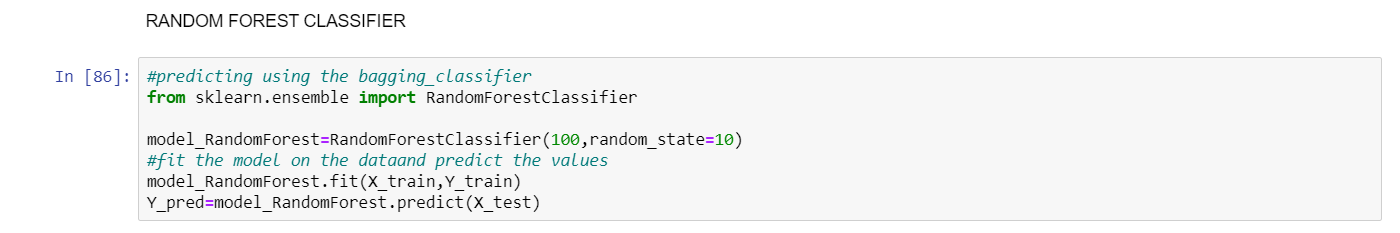
**Gradient Boosting Model:**

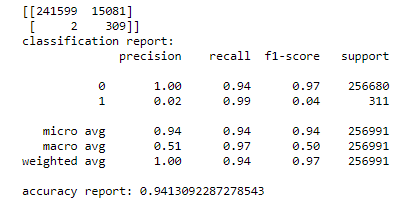
(Introduction On Gradient):



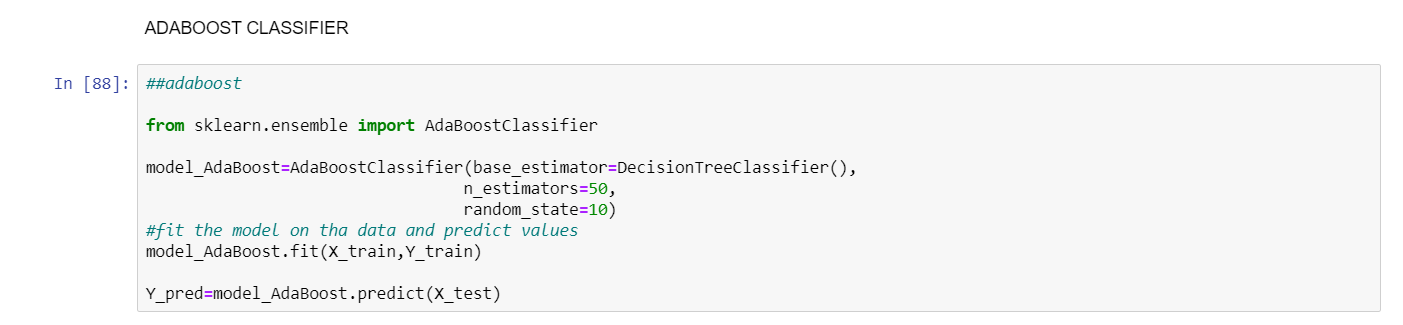


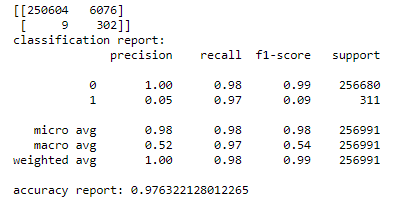
**Random Forest:**





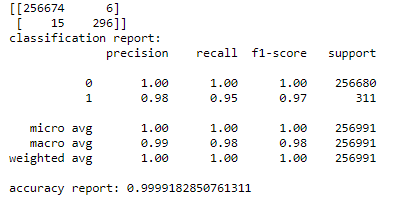
**Adaboost:**





**Bagging Classifier**





**Comparison Of Models**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model/Feature: | Accuracy score | Type I  Error | Type II  Error | F1 score | | Recall | |
| 0 | 1 | 0 | 1 |
| Logistic | 0.9994046484118121 | 86 | 67 | 1.00 | 0.76 | 1.00 | 0.78 |
| Cross Validation | 0.9992723480588814 | 121 | 66 | 1.00 | 0.72 | 1.00 | 0.79 |
| Decision Tree | 0.9999571969446401 | 1 | 10 | 1.00 | 0.98 | 1.00 | 0.97 |
| Gradient Model | 0.9997548552283932 | 53 | 10 | 1.00 | 0.91 | 1.00 | 0.97 |
| Random Forest | 0.9325073640711153 | 17344 | 1 | 0.97 | 0.03 | 0.93 | 1.00 |
| Adaboost | 0.9763143456385632 | 6078 | 9 | 0.99 | 0.09 | 0.98 | 0.97 |
| Bagging Classifier | 0.9999182850761311 | 6 | 15 | 1.00 | 0.97 | 1.00 | 0.95 |

**Conclusion**

Decision Tree is Best Among All

