PROJECT REPORT

ON

**XYZ Corporation Lending Data Project**

Submitted towards the partial fulfillment of the criteria for award of Post Graduate Data Science Degree by Imarticus Learning

*Submitted By:*

*Mr. Sanjay Sarkar*

*Mr. Rishabh Tiwari*

*Mr. Victor Ravindra*

*Mr. Yogesh Thakare*

Course and Batch: PGA-06 June 2021



# Abstract

People often save their money in the banks which offer security but with lower interest rates. XYZ Lending Club operates an online lending platform that enables borrowers to obtain a loan and investors to purchase notes backed by payments made on loans.

In general, whenever an individual/corporation applies for a loan from a bank (or any loan issuer), their credit history undergoes a rigorous check to ensure that whether they are capable enough to pay off the loan and in this industry it is referred to as credit-worthiness.

It is transforming the banking system to make credit system more affordable and investing more rewarding. But this comes with a high risk of borrowers defaulting the loans.

Based on the amount of risk that the issuer is willing to take (plus some other factors) they decide on a cutoff of that score and use it to take a decision regarding whether to pass the loan or not. This is a way of managing credit risk. The whole process collectively is referred to as underwriting. Hence there is a need to classify each borrower as defaulter or not using the data collected when the loan has been approved.

The XYZ Corp. have decided to a build or set a model which will take information of lenders regarding their current financial standing, previous credit history and some other variables as input and output a metric which gives a measure of the risk that the issuer will potentially take on issuing the loan. The measure is generally in the form of a probability and is the risk that the person will default on their loan (called the probability of default) in the future or not and will also help the them in deciding whether to sanction a loan or not in future.

# Acknowledgements

We are using this opportunity to express our gratitude to everyone who supported us throughout the course of this group project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. We are sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, we were fortunate to have great teachers who readily shared their immense knowledge in data analytics and guided us in a manner that the outcome resulted in enhancing our data skills.

We wish to thank, all the faculties, as this project utilized knowledge gained from every course that formed the PGA program.

Also, immensely thankful to Prof. Nikita Tandel who seamlessly trained us in Python and guided us to give our best performance.

We certify that the work done by us for conceptualizing and completing this project is original and authentic.

Date: February 25, 2021 Sanjay Sarkar

Place: Mumbai Rishabh Tiwari

Victor Ravindra

Yogesh Thakare

# Certificate of Completion

I hereby certify that the Project titled “**XYZ Corporation Lending Data**” was undertaken and completed under the supervision of Prof. Nikita Tandel. By Sanjay Sarkar, Rishab Tiwari, Victor Ravindra and Yogesh Thakare from the batch of PGA-6 (Feb 2021)

Date: February 25, 2021

Place – Mumbai

**Table of Contents**

[Abstract 2](#_bookmark0)

[Acknowledgements](#_bookmark1) 3

[Certificate of Completion](#_bookmark2) 4

[**CHAPTER 1: INTRODUCTION** 6](#_bookmark3)

* 1. [Title & Objective of the study 6](#_bookmark4)
  2. [Need of the Study 6](#_bookmark5)
  3. [Business or Enterprise under study 6](#_bookmark6)
  4. [Business Model of Enterprise](#_bookmark7) 7
  5. [Data Sources](#_bookmark8) 7
  6. [Tools & Techniques](#_bookmark9) 7

[**CHAPTER 2: DATA PREPARATION AND UNDERSTANDING**](#_bookmark10) 9

* 1. [Data Extraction and Cleaning 9](#_TOC_250019)
     1. [Missing value analysis 9](#_TOC_250018)
     2. [Checking duplicate records 12](#_TOC_250017)
     3. [Categorical an Numerical variables………………………………………………………12](#_TOC_250016)
     4. [Handling Outliers………………………………………………………………..................13](#_TOC_250015)
  2. [Data Visualization……………………………………………………………………………….15](#_TOC_250014)
  3. [Feature selection………………………………………………………………………………...24](#_TOC_250011)
  4. Imputing Missing Values………………………………………………………………………….26
  5. [Encoding………………………………………………………………………………………...27](#_TOC_250010)

[**CHAPTER 3: FITTING MODELS TO DATA**](#_bookmark14) ………30

* 1. [Data Partition](#_bookmark15) 30
  2. [Feature Scaling 30](#_TOC_250007)

**CHAPTER 4: MODEL BUILDING**…………………………………………………………………31

* + 1. [Logistic Regression 32](#_TOC_250005)
    2. [Decision Tree Classification 36](#_TOC_250004)
    3. Extra-Trees classifier…………………………………………………………………………..38
    4. Random-Forest classifier……………………………………………………………………….39
    5. AdaBoost classifier…………………………………………………………………………….40
    6. Gradient-Boosting classifier…………………………………………………………………….41
    7. XGBOOST classifier…………………………………………………………………………...42
    8. Artificial Neural Network (ANN) …..43
    9. Ensemble Models………………………………………………………………………………47
  1. Mutual-info-classification………………………………………………………………………………………49
  2. Select-KBEST……………………………………………………………………………………..51
  3. SMOTE (Gradient-Boosting)……………………………………………………………………...54
  4. Feature Selection ………………………………………………………………………………….57
  5. Pre-Processing……………………………………………………………………………………..58
  6. Logistic Regression ( new-dataset)……………………………………………………………...…61
  7. Decision Tree ( new-dataset)………………………………………………………………..……..63
  8. Gradient-Boosting classifier ( new-dataset)…………………………………………….………….64
  9. Artificial Neural Network (ANN) ……………………………………………………………….66

[CHAPTER 5: FINAL MODEL](#_bookmark16) …67

[CHAPTER 6: CONCLUSION](#_bookmark16) 69

# CHAPTER 1: INTRODUCTION

## Title & Objective of the study report

‘**XYZ Corporation Lending Data Project’** is the project we are working upon which falls under the BFSI domain (Banking Financial services and Insurance sector). The text files contain complete loan data for all loans issued by XYZ Corp. through 2007-2015. The primary purpose of working on this project is to predict the probability of default, whether the customer will default the loan or not by using the past data. That means, given a set of new predictor variables, we need to predict the target variable as 1 -> Defaulter or 0 -> Non- Defaulter.

## Need of the Study

In this project, the main purpose is to predict whether a borrower will default or not, so that investors can avoid such borrowers using manual investing feature provided by lending club. This, however, does not necessarily lead to highest return on investment (ROI) because by completely avoiding potential defaults, one is also avoiding riskier loans that may lead to higher ROI even though they’ll default at some point in the future. In order to maximize ROI, one needs to optimize ROI instead. In this project, we work on the simpler problem that is to predict loan defaults.

## Business or Enterprise under study

XYZ Corporation Lending Data is under the study. Data of Loans issued by XYZ Corp. through 2007-2015 is used for analysis. The data contains the indicator of default, payment information, credit history, etc.

## Business Model of Enterprise

Selecting the relevant variables from the dataset and arranging their values in order of importance to create a models to predict the probability of default of an individual in the future by performing different types of algorithms on the data.

* 1. **Data Sources**

XYZ Corp Lending Data- Data contains the information about the status of the loan defaulter. The dataset contains the information like age, gender, annual income, grade of the customer paying capacity

Data Set Description: Contains 855969 rows and 73 columns

The response variable is ‘default\_ind’ with ‘0’ for Non-Defaulter and ‘1’ for Defaulter.

* 1. **Tools & Techniques Tools:** Jupyter Notebook.

**Techniques:** Logistic Regression, Decision Tree Classification, Artificial Neural Networks, Gradient Boosting Classifier.

# CHAPTER 2: DATA PREPARATION AND UNDERSTANDING

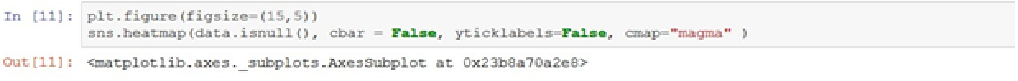
Firstly we imported the required Libraries and then the following sequence of steps was followed to perform data preprocessing.

## Phase I – Data Extraction and Cleaning:

* + - **Missing Value Analysis and Treatment**

After printing the shape of the data, we found that the dataset consists of 855969 observations and 73 variables.

The initial step was to check the missing values in each variable and for a better view, we plotted a heatmap of the dataset for visualizing the missing values as shown below:

Chart, timeline

Description automatically generated with medium confidence

It is evident from the above heatmap that our dataset contains a lot of missing values and we cannot use feature that has so many missing values.

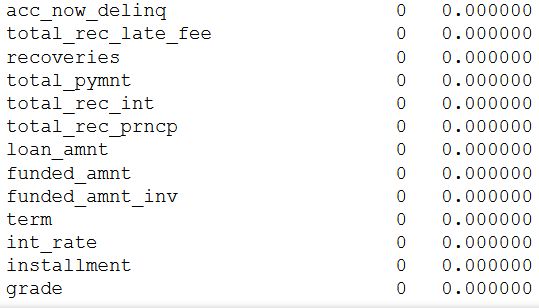
List of missing values:

Text, table

Description automatically generated

Text

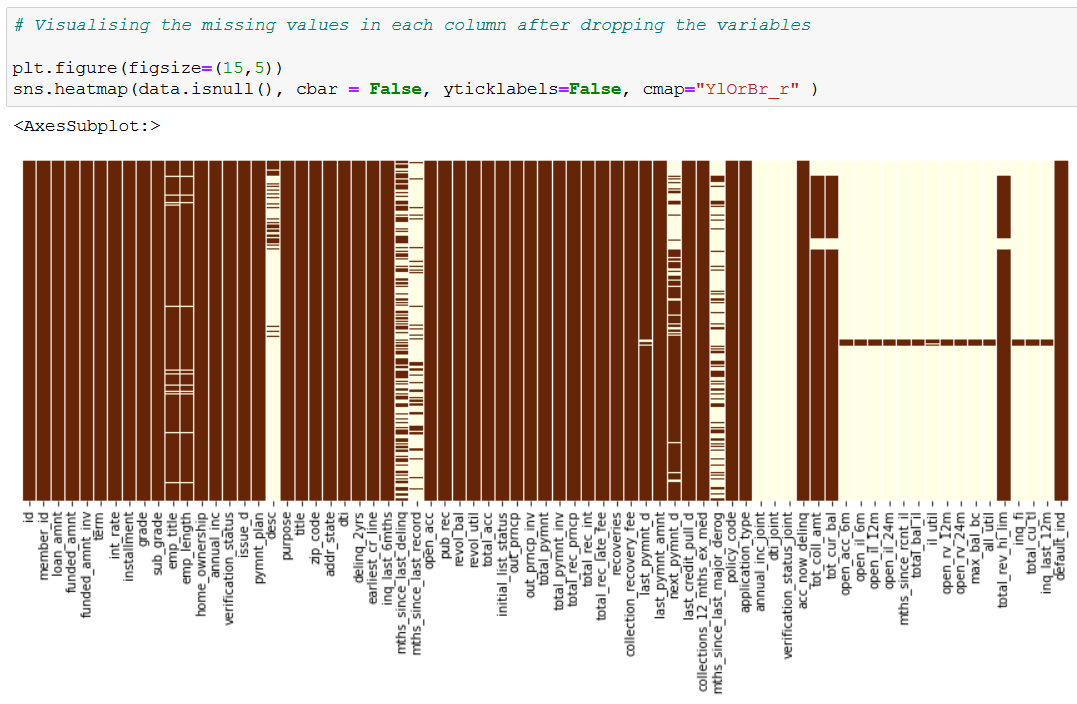
Description automatically generated



Above heatmap shows the intensity of values that are missing in every columns. All the light colored columns represents the amount of missing values present in that specific column.

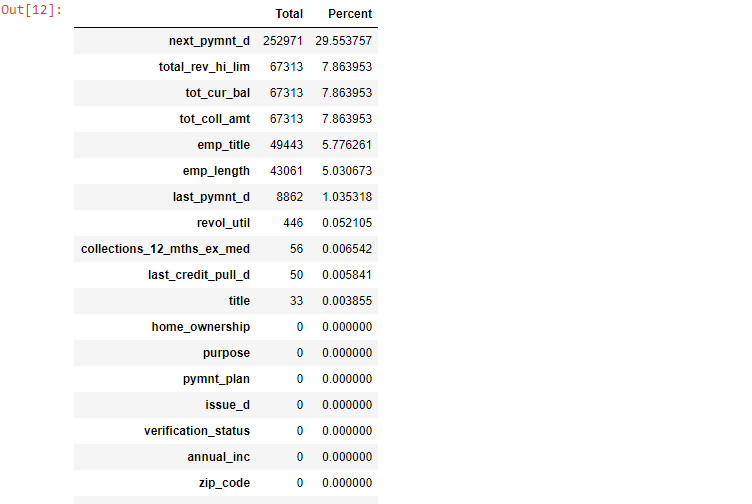
Firstly, setting a threshold of 50%, i.e. dropping the columns which have more than or equal to 50% missing values. We are then left with **52 variables**.

Then visualizing the missing values in each column after dropping the variables, we get the following heatmap:



By comparing the above two heatmaps, it is clearly seen that the amount of missing values have been reduced drastically.

* **List of variables available after dropping the missing values total and percentage wise**

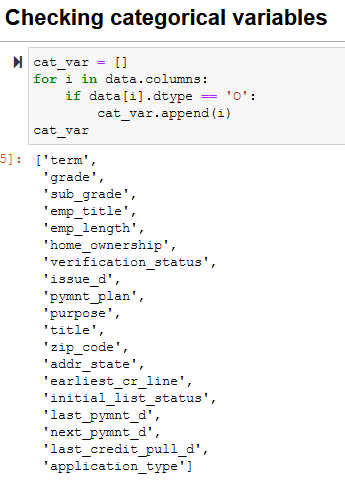


* Also the dataset does not consists any duplicate records.



* **Filtering Categorial and Numerical Variables**

Categorical variables:



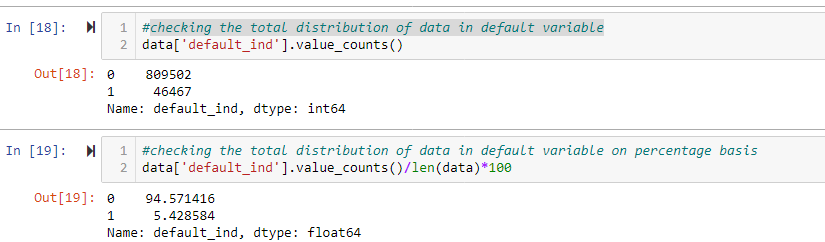
Numerical variables:



* + - **Handling Outliers**

Outlier Treatment was not done because of the following reasons:

* + - * Presence of Clusters in the outliers.
      * Less number of outliers as compared to the huge number of observations whose effect will be negligible.
      * Lack of Domain knowledge.
* **checking the total distribution of data in default variable by total counts and percentage distribution:**





## Exploratory Data Analysis:

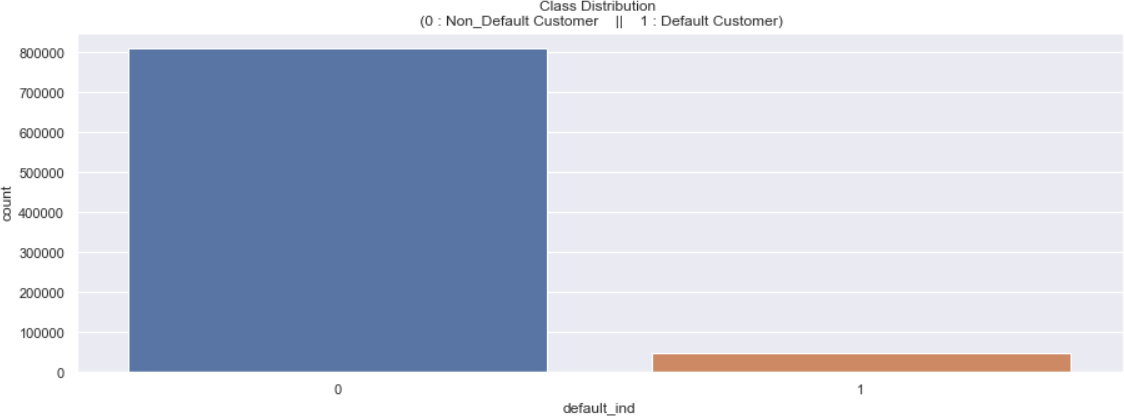
EDA is the process of performing initial investigations on data to discover patterns, to test hypothesis and to check assumptions with the help of descriptive statistics and graphical representations.

The response variable in this data is ‘**default\_ind**’ which indicates that the customer will

Default (‘1’) or Non-Default (‘2’)

* + - Plot showing the count of the Default customers and Non-default customers in

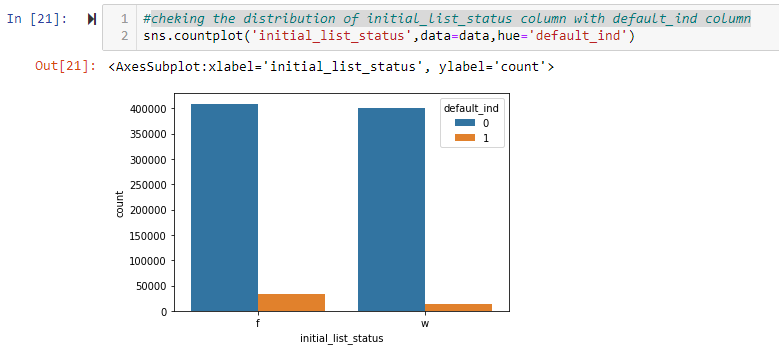
‘**default\_ind**’ variable.



Non-Default Customer: 94.57 % of the dataset. Default Customer: 5.43 % of the dataset.

From the above graph, we gain that the dataset is highly unbalanced.

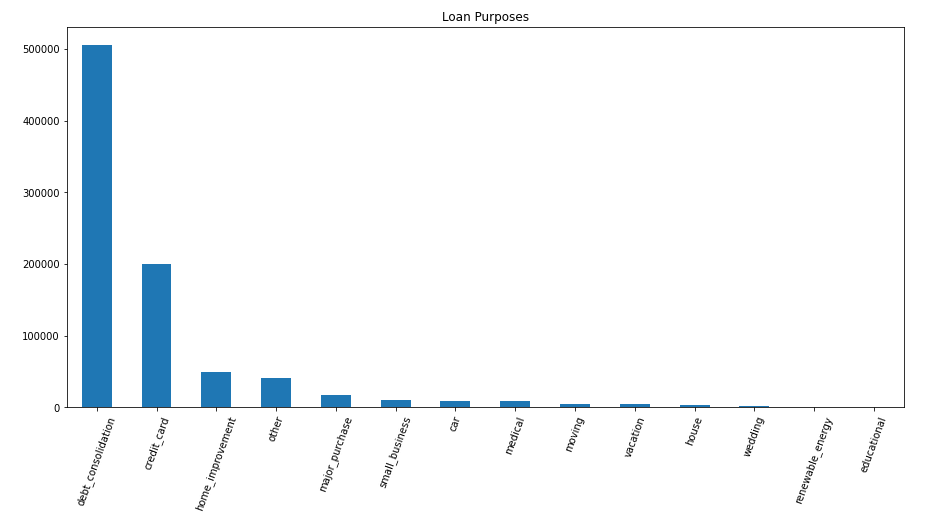
* cheking the distribution of initial\_list\_status column with default\_ind column



* + - Plot showing the distribution of ‘term’ variable.

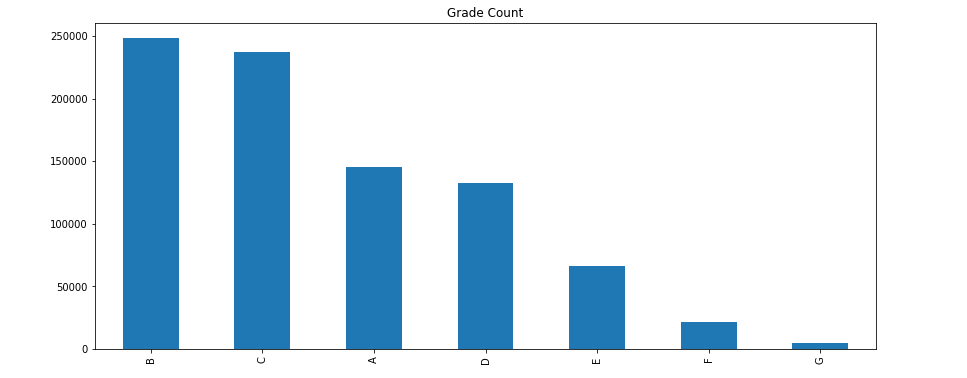


**# cheking the loan purpose of customer**

****

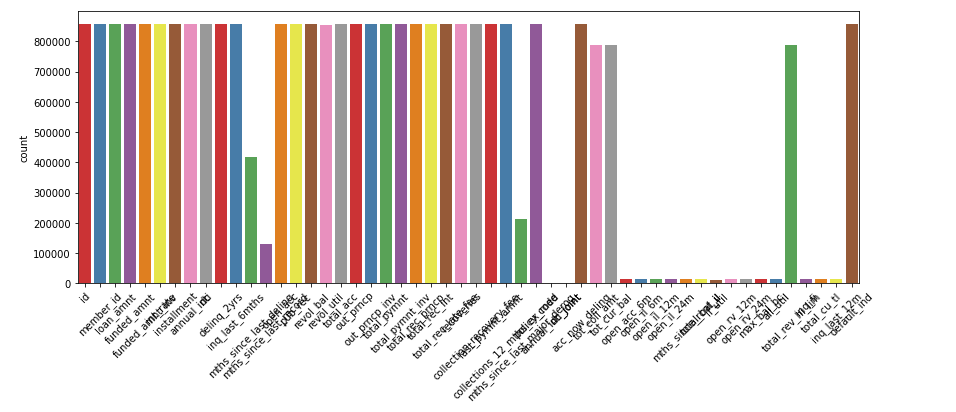
From the above figure, it is observed that huge loans are taken for debt consolidation.

**# Checking for grade count**

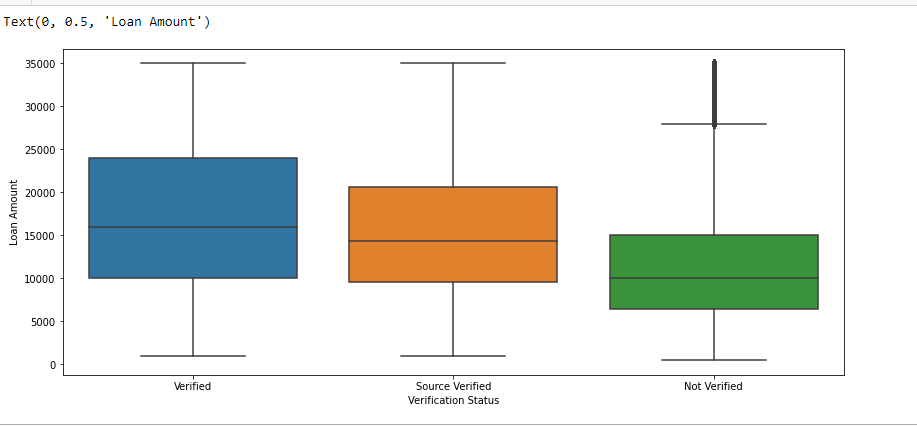
****

It appears that B and C are the dominant grades.

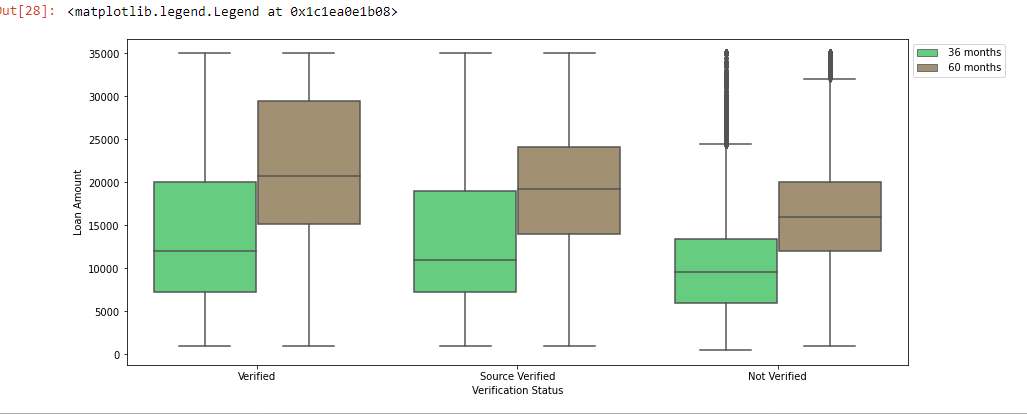
**# Using countplot to visualize the frequency distribution of a single discrete/categorical variable**

****

**#checking verification status w.r.t loan amount**

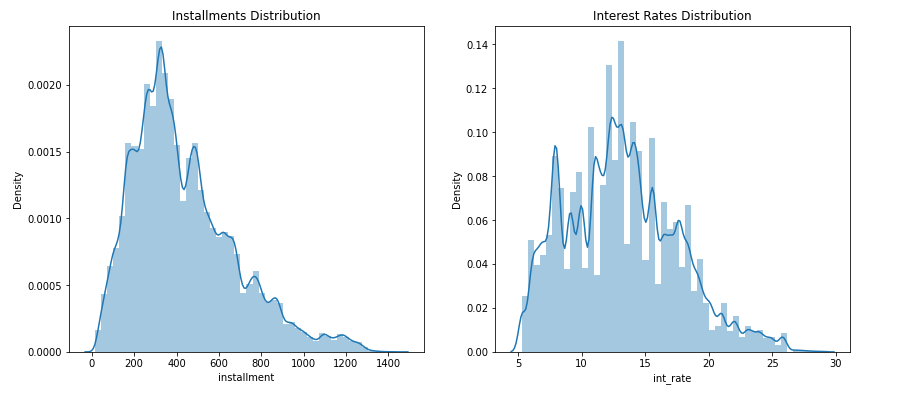


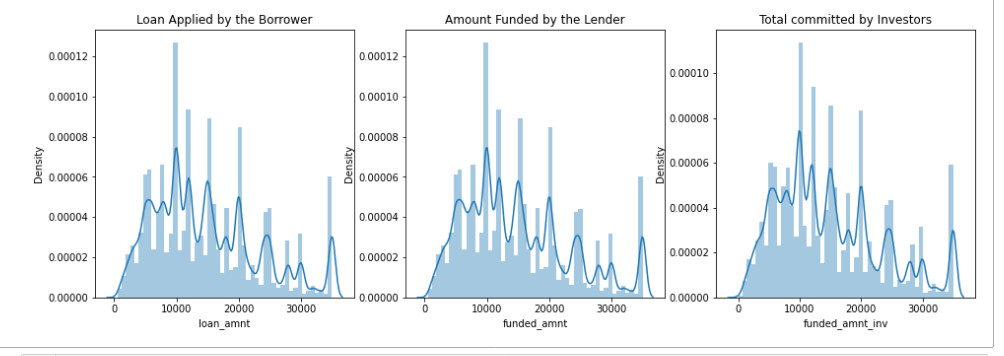
**# checking loan amount w.r.t verification status**

****

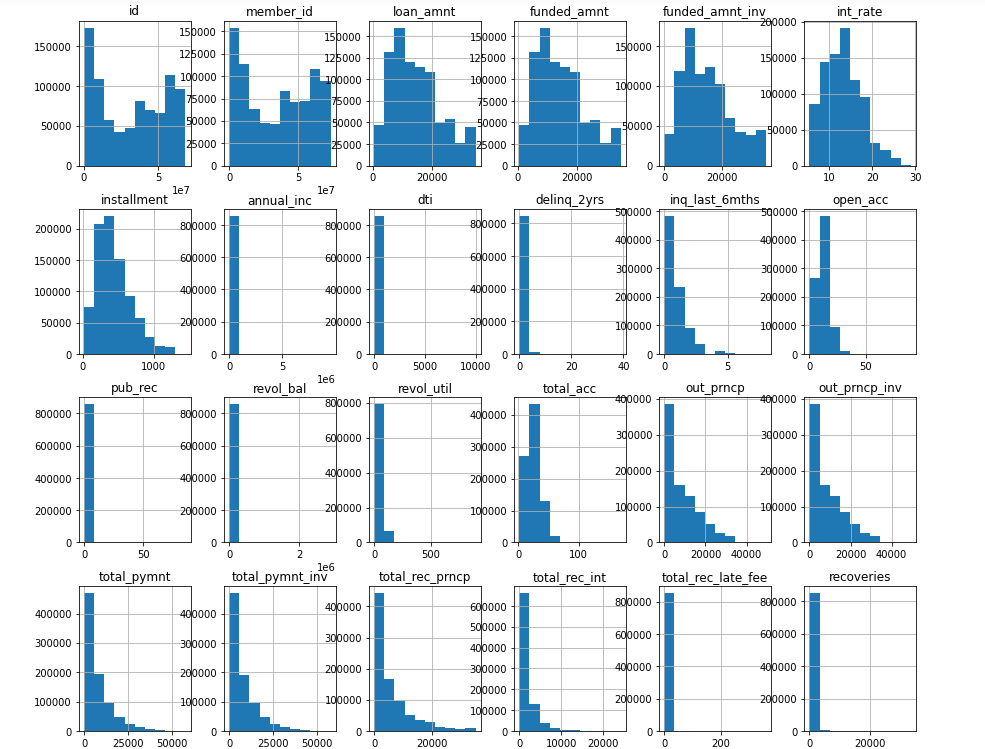
* + - Plot showing the distribution of loan amount, funded amount, Installments distribution

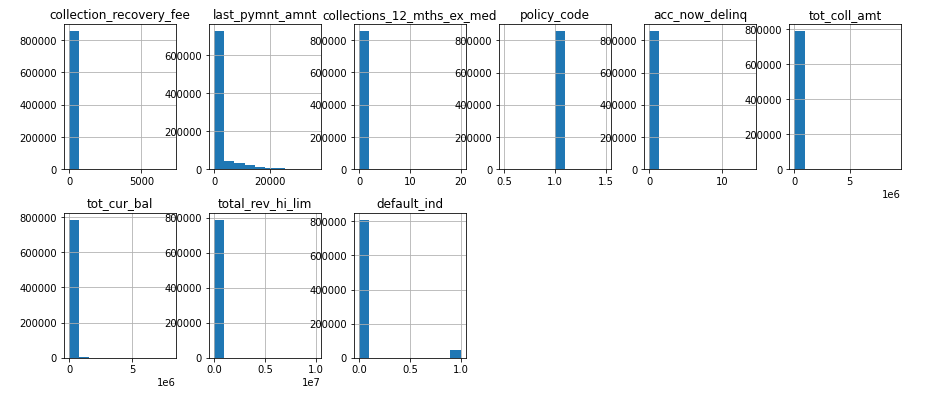
and total committed by investor.





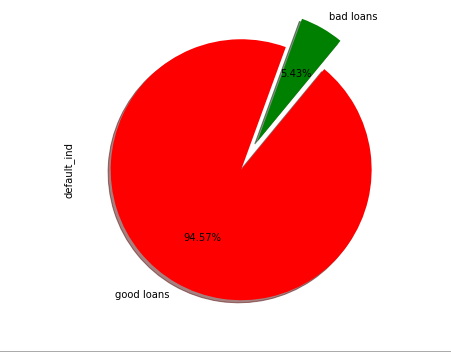
**#plotting histogram of all features**

****



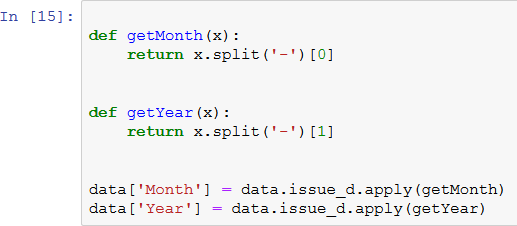
Above histograms shows the variations in the symmetry of the variable, some are symmetric that is some are almost normally distributed and some are asymmetric that is they are skewed.

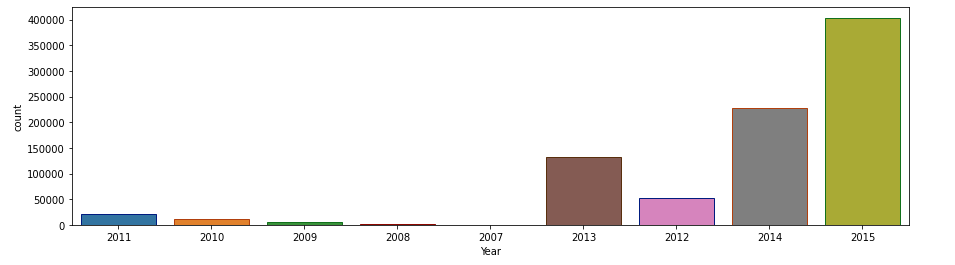
**# Pie chart on Loan Status**

****

Above pie chart show the distribution of the dependent variable

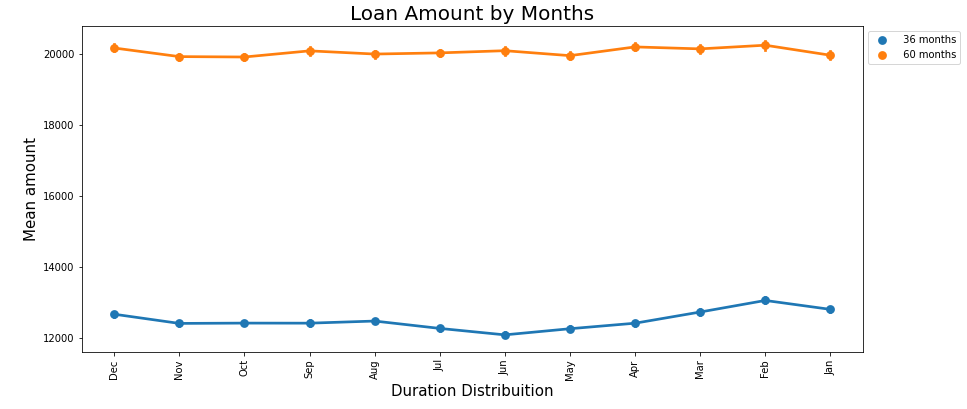
* **A function is created that will split the ‘issue\_d’ variable which is nothing but the month in which the loan was funded.**





An exponential rise is observed in the number of applications for loan over a period of years.

#**Loan Amount by Months and term**

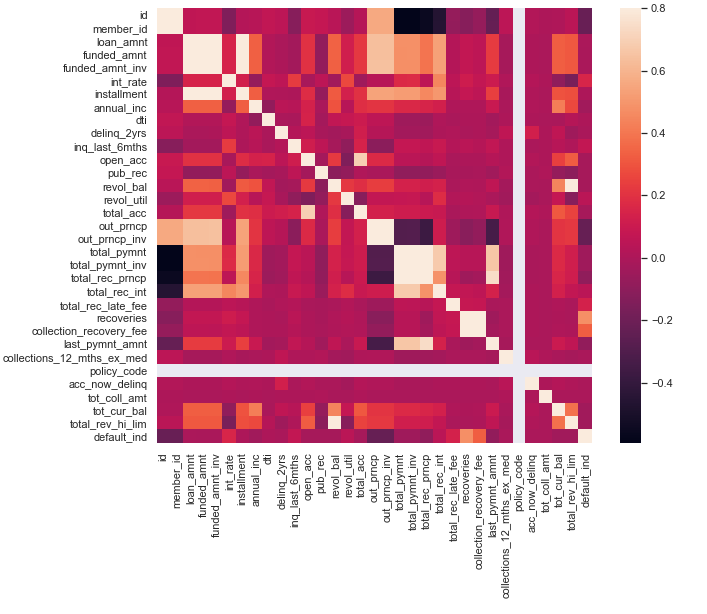
****

Above graph shows the distribution of loan amount on basis of months and term.

**# Bar plot on emp\_length**

****

**# plotting correlation matrix**

****

**Correlation matrix show the correlation between the variables light shade show strong positive correlation and dark side showing strong negative correlation.**

**Feature Selection:**

**After having a closer look at the data (and studying frequency tables), we can remove the following irrelevant variables, reasons described:**

1. policy\_code is always == 1

2. payment\_plan has only 10 y and 887372 n

3. id and member\_id are all unique, which is a bit misleading. So every record is a unique customer

4. application\_type is 'INDIVIDUAL' for 99.94% of the records

5. acc\_now\_delinq is 0 for 99.5% of the records

6. emp\_title not needed here, because it's a categorical varibale with (290912 level)

7. zip\_code not needed for this level of analysis,

8. title can be removed as well because it's a categorical varibale with (61000 level).

9. earliest\_cr\_line variable because it's a date varibale with (697 level)

10. addr\_state variable has 51 level

11. next\_pymnt\_d variable because it's a date varibale with (3 level) and it contains 29% Missing info.

12. last\_pymnt\_d variable because it's a date varibale with (51 level)

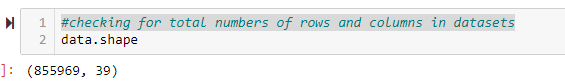
13. last\_credit\_pull\_d variable because it's a date varibale with (102 level)

14. sub\_grade because it has 35 levels

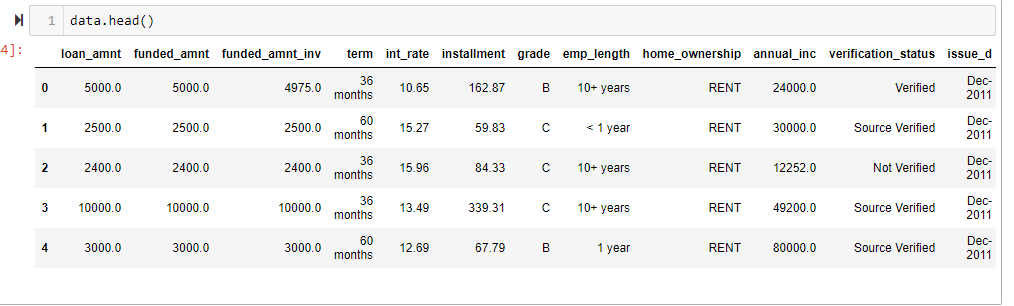
**After dropping the above columns, we are left with 39 variables of whose missing values will further be treated.**

****

**#checking for total numbers of rows and columns in datasets**

****

**# Checking the contents of dataset by displaying the whole dataset:**

****

* The remaining missing values present are treated by using **Mean** and **Mode**.
* **Missing values treatment with Mean:**

The missing values of the following variables are treated with mean:

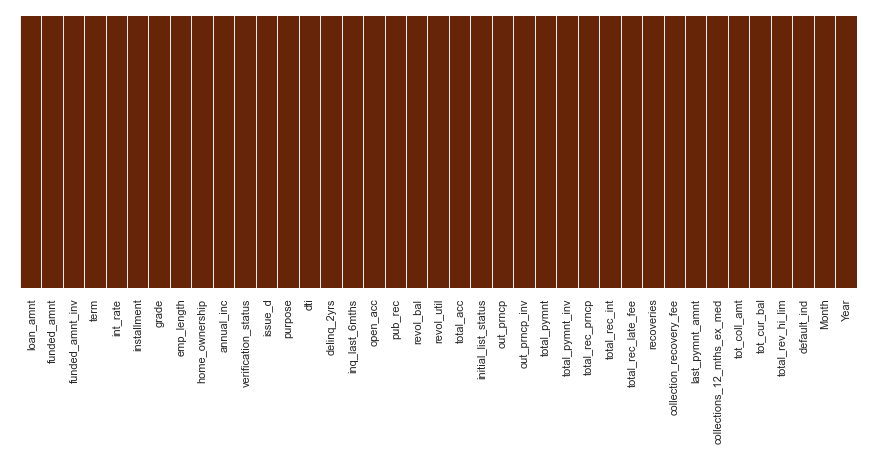
* + - ths\_since\_last\_delinq
    - collections\_12\_mths\_ex\_med
    - total\_rev\_hi\_lim
    - revol\_util

**While the missing values of the following variables are treated with Mode:**

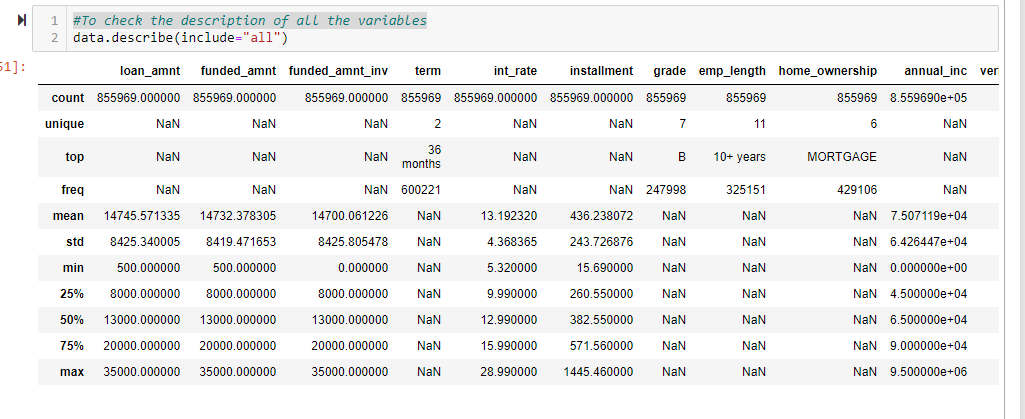
* + - term
    - emp\_length
    - verification\_status
    - last\_pymnt\_d
    - next\_pymnt\_d
    - last\_credit\_pull\_d



After the complete treatment of the missing values, it is evident from the below heat map that the dataset is now clean and ready for EDA.



**#checking the description of all the variables**

****

* 1. **ENCODING**

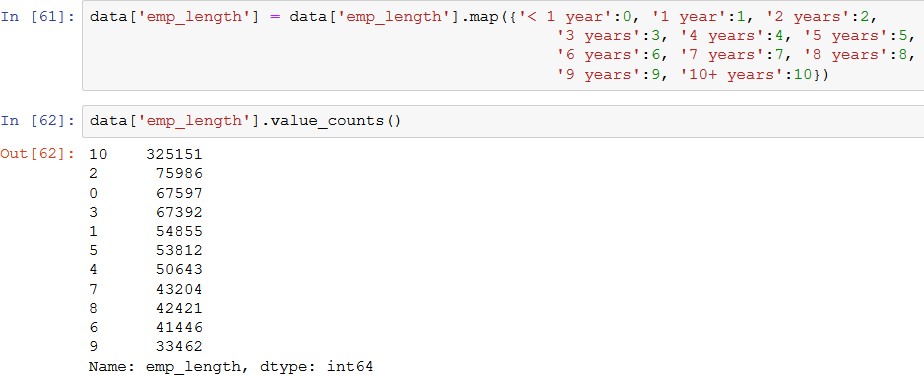
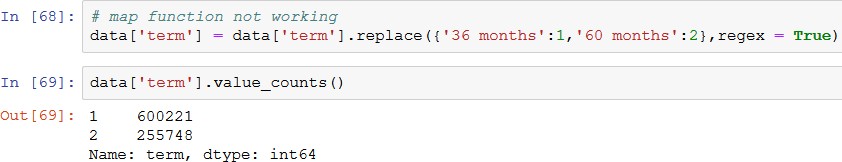
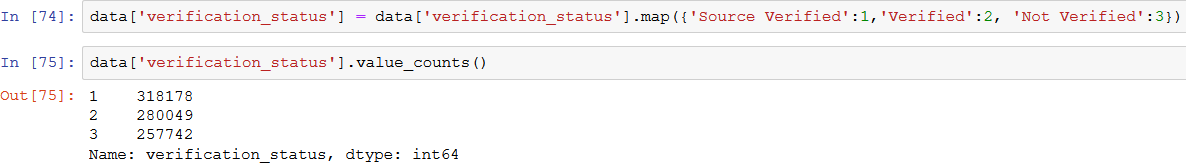
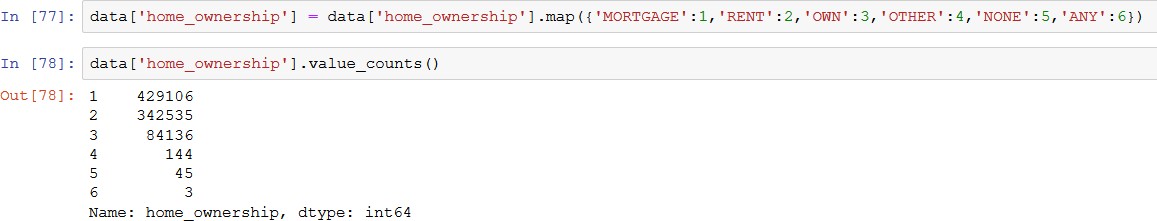
**Label Encoding:**

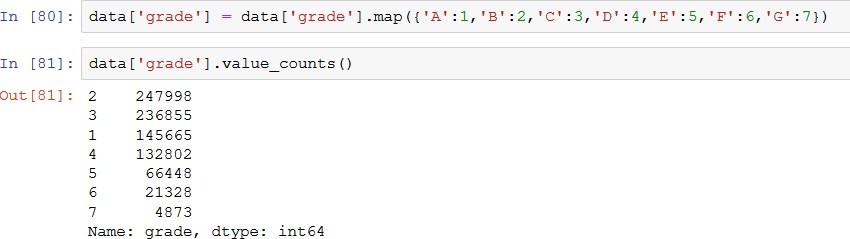
The SciKit Learn library in Python consists of two encoders which are used to convert categorical data or text data into numbers which will help our model to understand. The two encoders are **Label Encoder** and **One Hot Encoder.**

By importing the LabelEncoder class from the sklearn library, a categorical data or text data can be converted to numbers, fit and transform the respective categorical variable data and then replace the existing text data with the new encoded data.

Now when the data has been encoded into numbers, the model might get confused into thinking that a column has data with some kind of order or hierarchy. Therefore, to overcome this One Hot Encoder is used.

**MANUAL LABEL ENCODING**

* + - Employee length in years has 10 levels. The possible values we can assign is from 0 to 10 with 0 indicating less than one year and 10 indicating experience of ten or more years.
    - Similarly the term which consists of 2 levels (36months and 60 months) are label encoded with 1 and 2 respectively.
    - Initial list status which indicates whether the loan is an individual application or a joint application with two co-borrowers. Replacing f and w with 1 and 2 respectively.
    - Verification status with 3 levels: Source Verified, Verified and Not Verified replaced with 1, 2 and 3 respectively.
    - Home ownership which has 6 levels such as ‘Mortgage’,’ Rent’, ‘Own’, ‘Other’, ‘None’ and ‘Any’ have been label encoded as well.
    - 7 levels of Grades which was assigned by XYZ Corp also needed label encoding as well as the purpose variable with 14 levels provided by the borrower for the loan request. Grade:

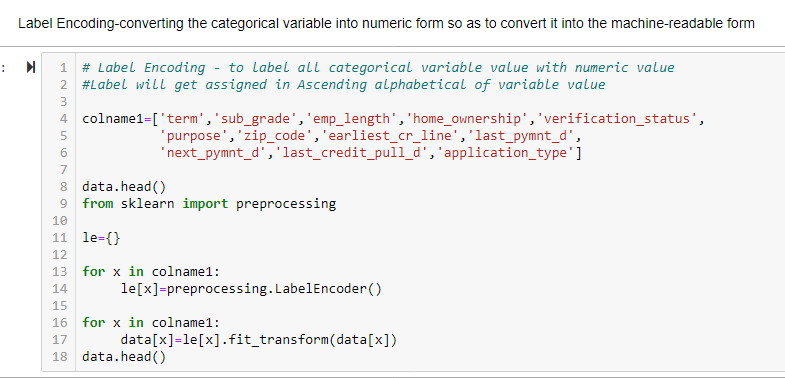


Purpose:



# Method 2

**Label Encoding-converting the categorical variable into numeric form so as to convert it into the machine-readable form.**

****

The final data is prepared and we are left with 37 variables.

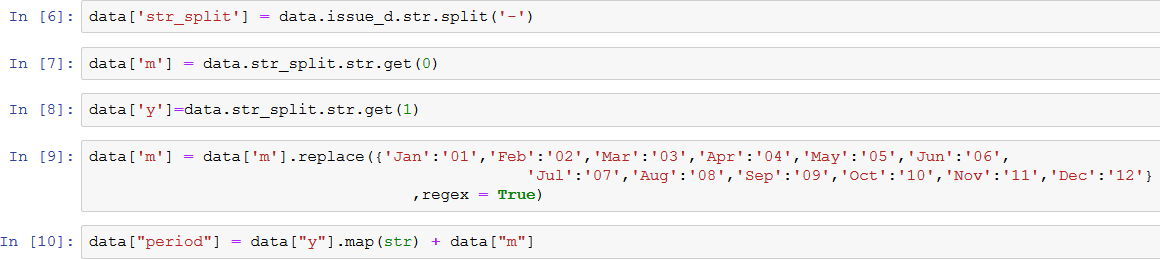
# CHAPTER 3: FITTING MODELS TO DATA

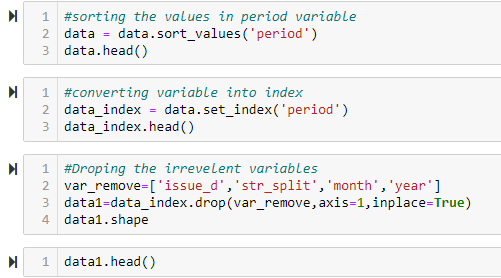
## Data Partition:

The data is divided based on the ‘issue\_d’ variable from which the records from **June-2007 to May-2015** will go into Training data while the records from **June-2015 to Dec-2015** will fall in the Testing data.

So to treat the date column i.e. ‘issue\_d’, Split the column into two different columns and replace the values as per the requirement. Then with the help of map function join the split columns and merge them one with a different name (‘period’). Followed by sorting the

‘period’ column and making it an index for slicing according to the requirement.

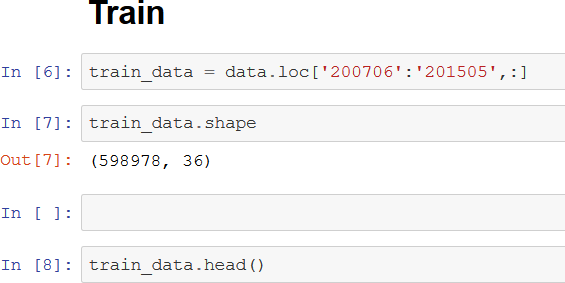


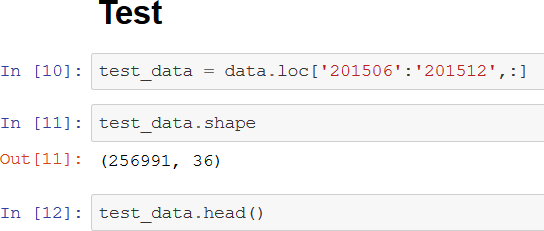


Followed by dropping the irrelevant columns such as ‘issue\_d’, ‘str\_split’, ‘m’ and ‘y’, we are

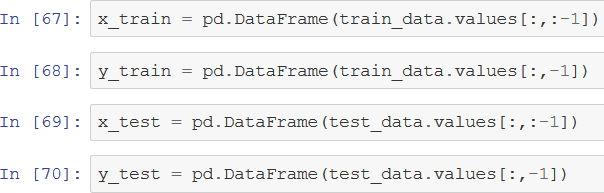
left with 36 variables.

**# Slicing the data into train and test**





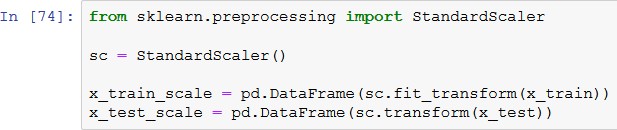
Creating the x\_train, y\_train, x\_test and y\_test dataframes:



## 

## Feature Scaling

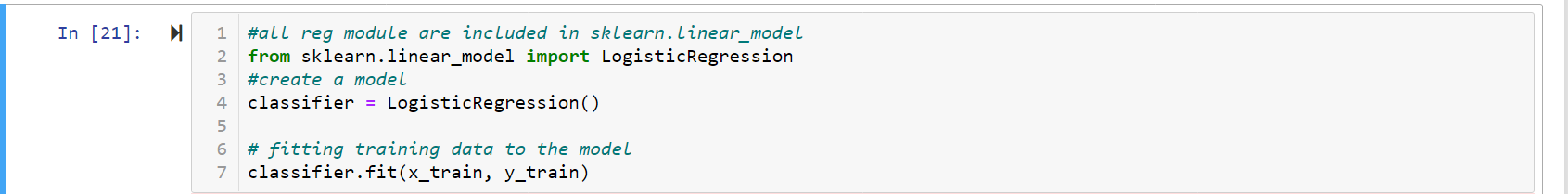
Feature scaling involves rescaling the features so as to limit the range of variables so that they can be compared on common grounds. Using the sklearn library and importing the StandardScaler class, we can use feature scaling.

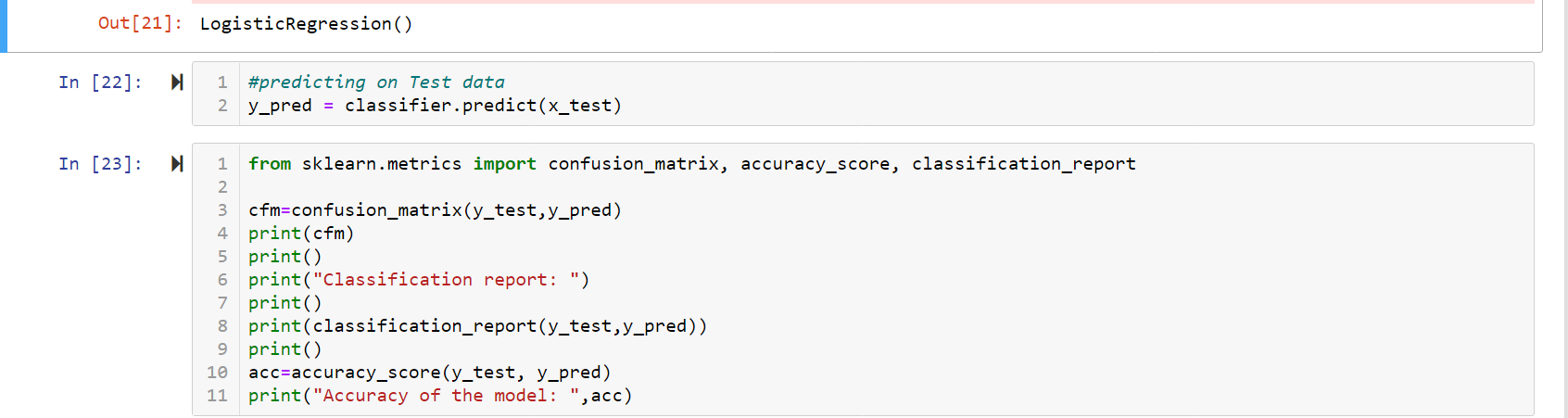


# Chapter 4: Model Building

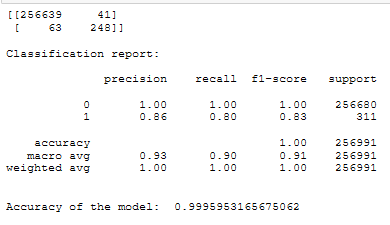
# We used different types of Machine leaning and Deep learning techniques

## 4.1.1 Logistic Regression



****

From the sklearn library using the ‘Logistic Regression’ class, we created a logistic regression model and following results were interpreted:

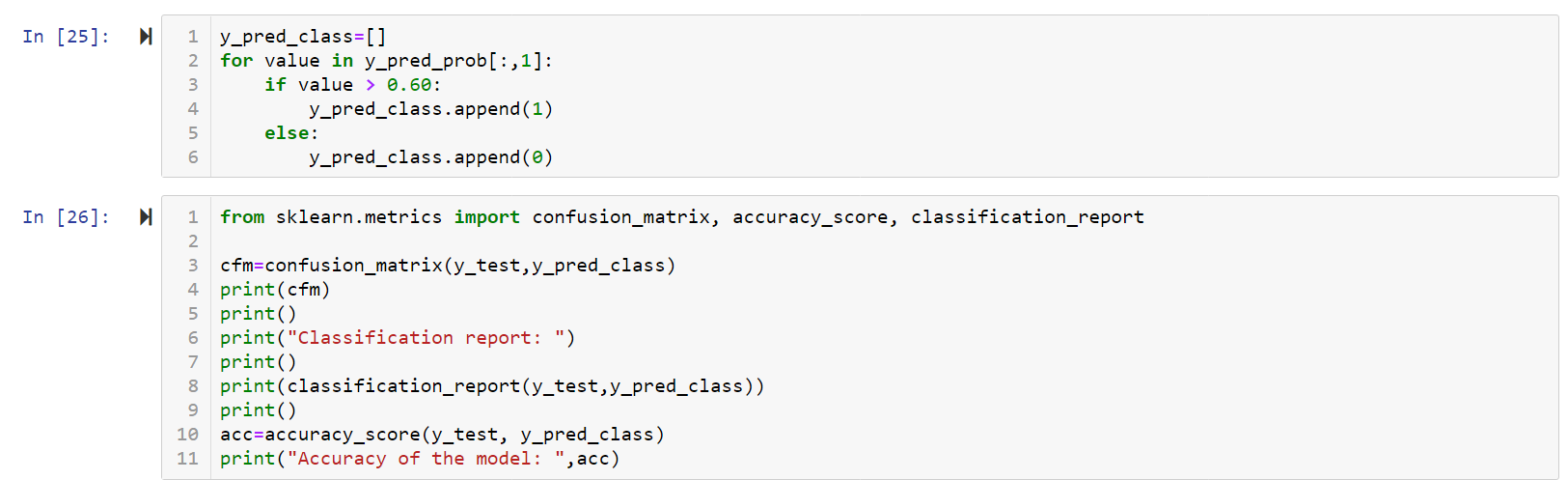


Referring to the above confusion matrix, we can clearly see that the **Type I** error is **41** while the **Type II** error is **63**.

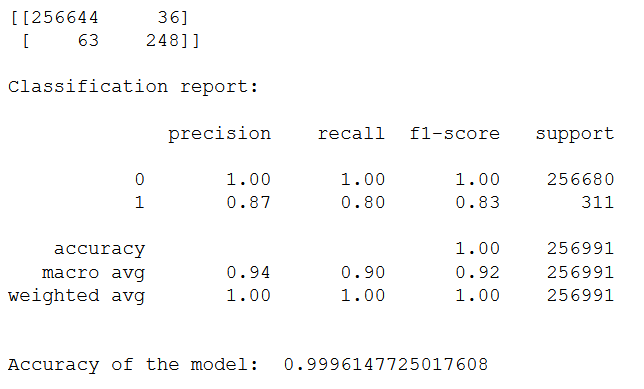
Since the data is unbalanced, we would not focus on the accuracy of the model but instead tune the model for less Type I and Type II errors.

**4.1.2 TUNING THE LOGISTIC REGRESSION MODEL:**

Adjusting the threshold level of the probabilities to 0.60:

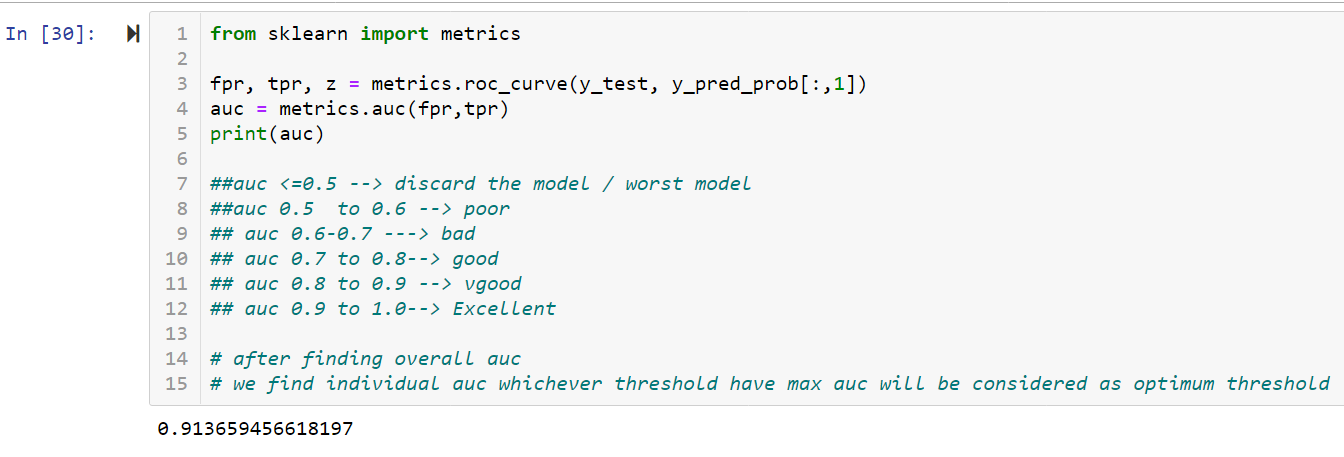


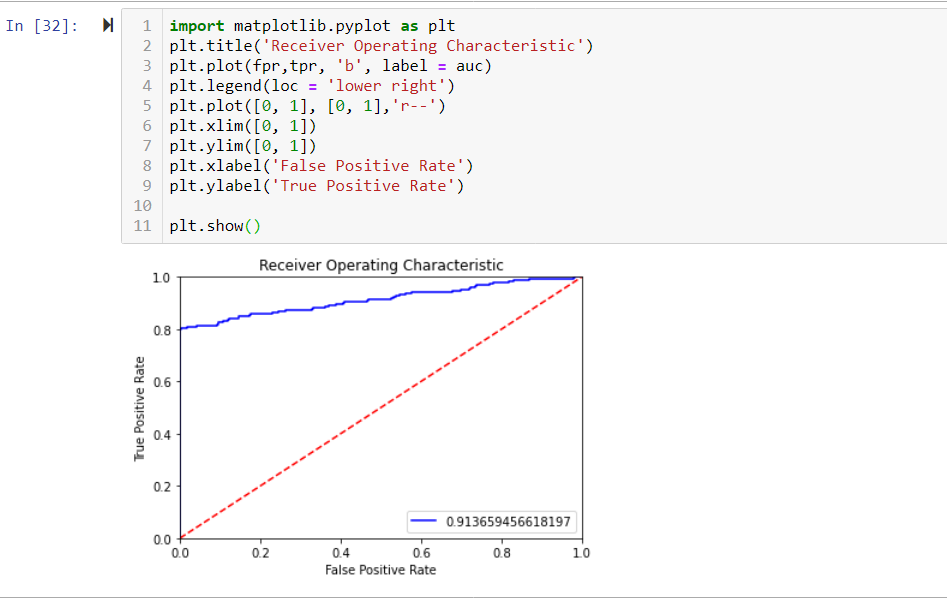
After tuning the model, we get the following results:

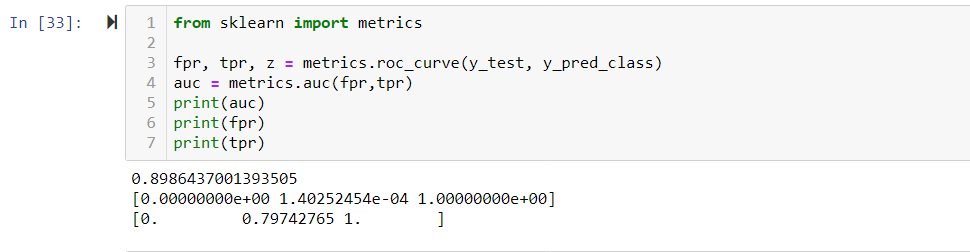


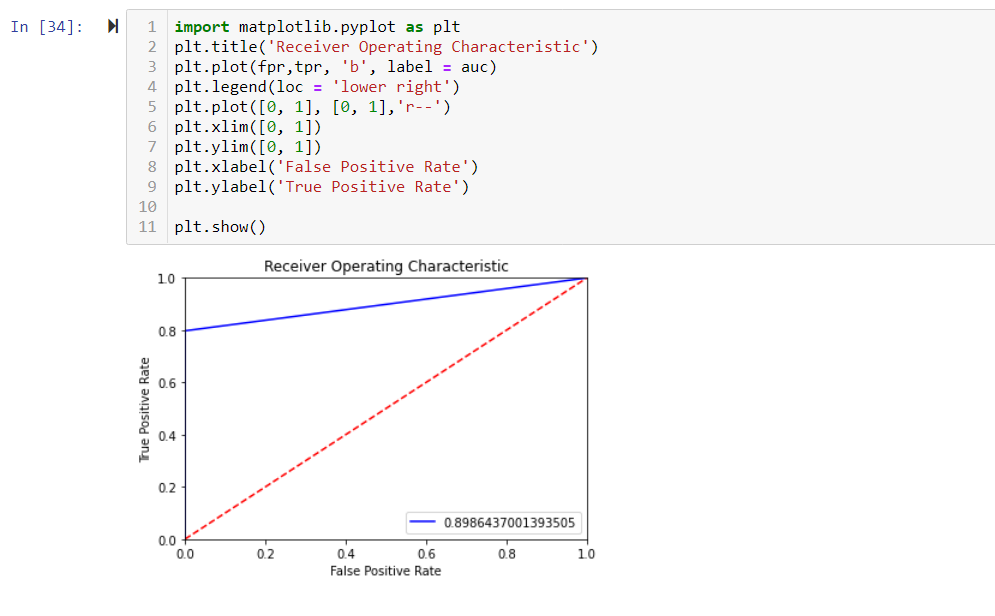
Now the **Type I** error has decreased to **36** after tuning while the Type II error is still the same.

# 4.1.3 Determing AUC - ROC:







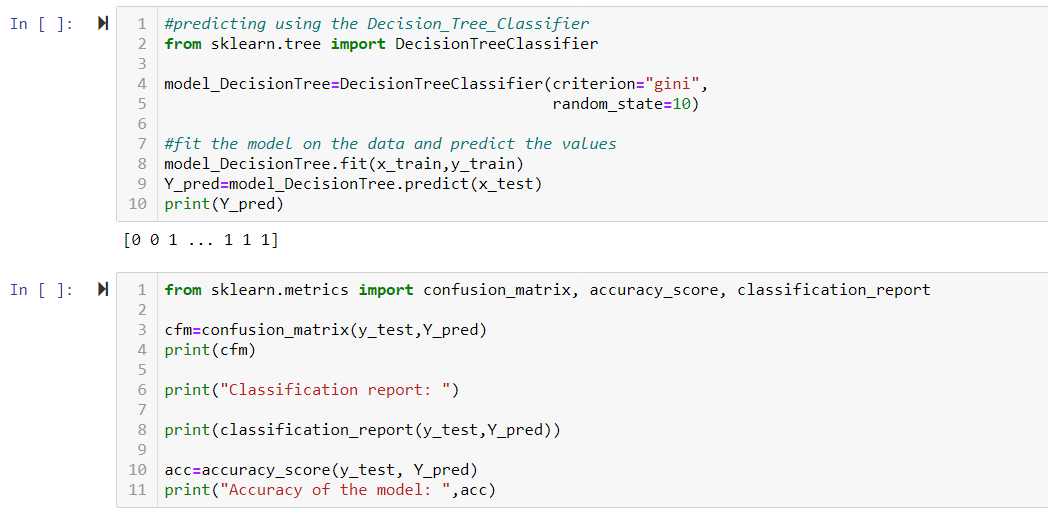


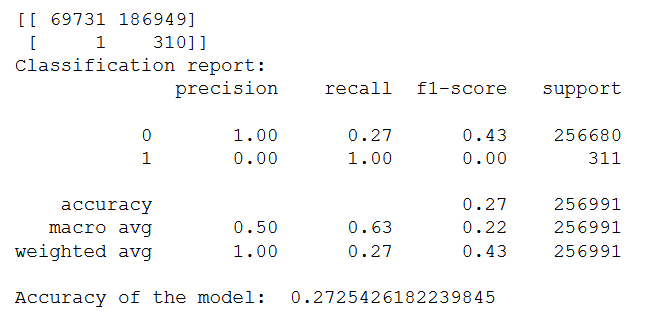


## Decision Tree Classification

Training the model on the train set and then predicting on the test set using ‘**Gini**’ for

Splitter selection and using the ‘DecisionTreeClassifier’ class.

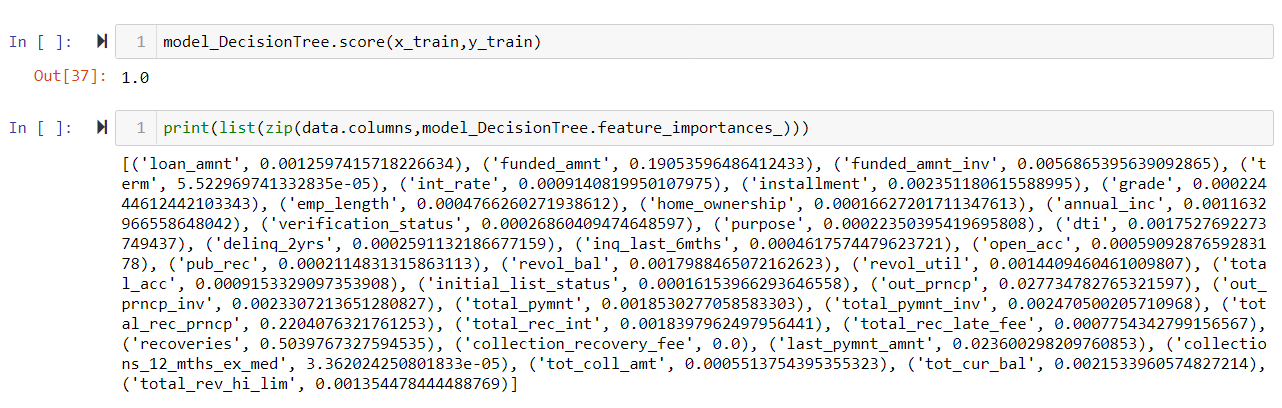




From the confusion matrix we can conclude that Type II error is 1 & Type I error is very high i.e 186949, F1 score for class 1 is very low which is 0.00 and the accuracy of the model is 0.2725.

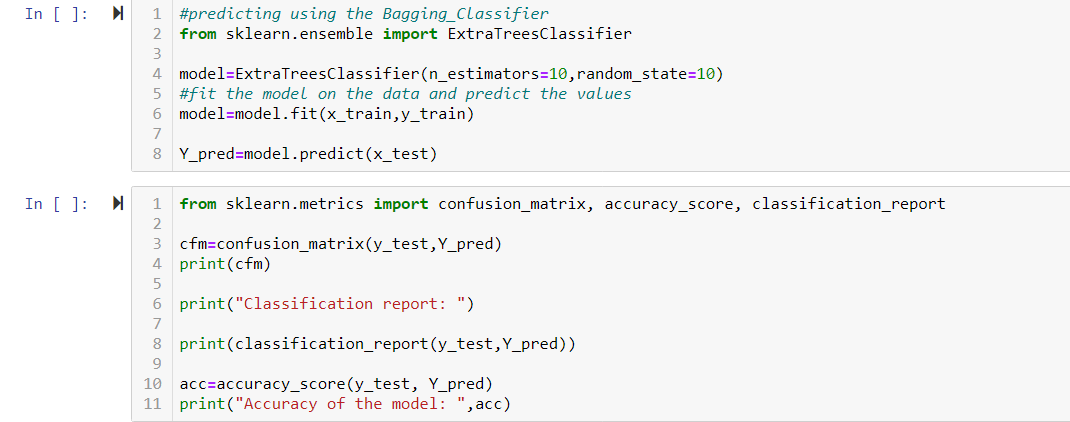
In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

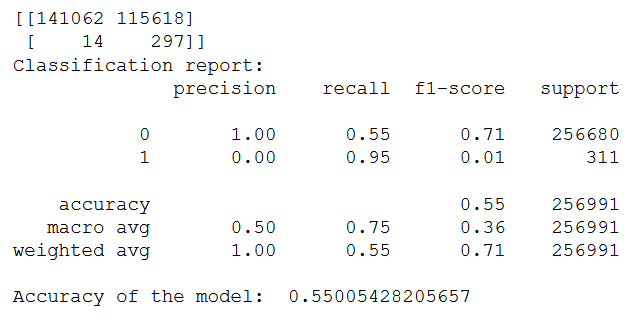
**#Using feature importance method to check for significant variables.**



* We performed feature importance method to determine the list of significant variables in dataset.
  + 1. **Bagging Technique: ExtraTreesClassifier**

Further training the model on the train set on the basis of Bagging techniques that is using Extra tree classifier and then predicting on the test set to check for the desired output.

****

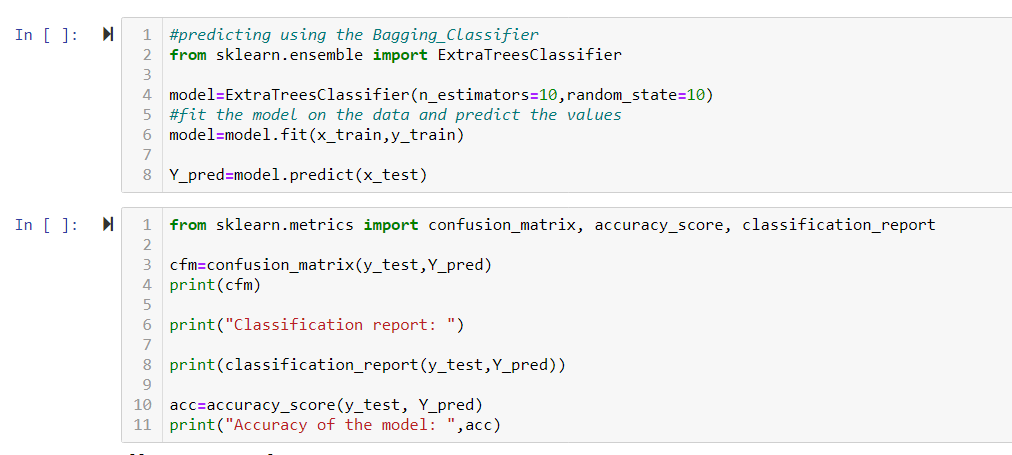


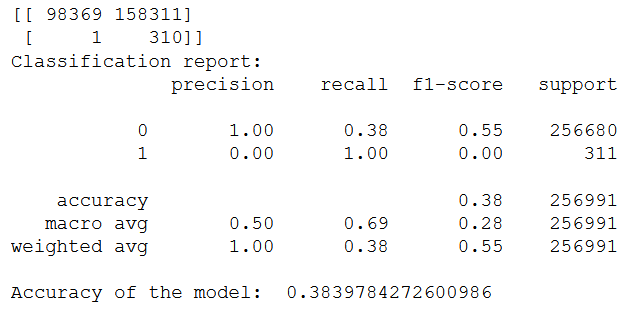
From the confusion matrix we can conclude that Type II error is 14 & Type I error is very high i.e. 115618 and the accuracy of the model is 0.5500.

In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

* + 1. **Bagging Technique: Random Forest Classifier**

Further training the model on the train set on the basis of another bagging techniques that is using Random Forest Classifier and then predicting on the test set to check for the desired output.



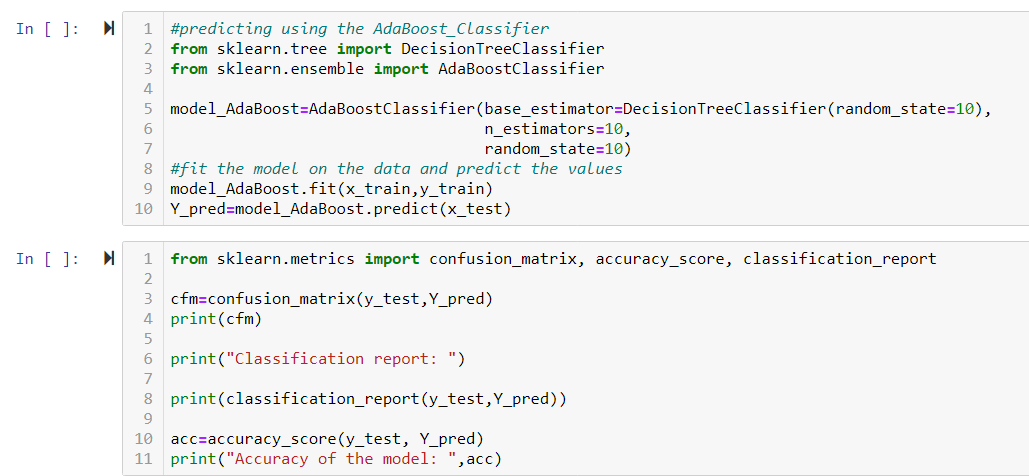


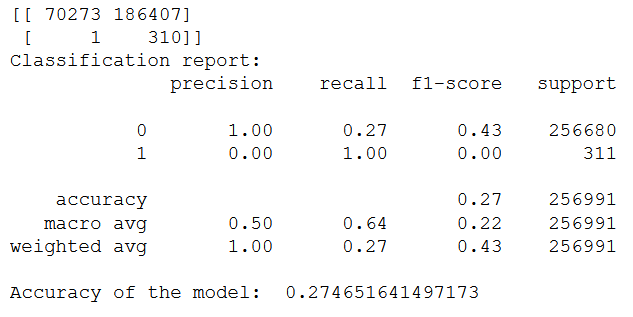
From the confusion matrix we can conclude that Type II error is 1 & Type I error is very high i.e 158311 and the accuracy of the model is 0.3839.

In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

* + 1. **Boosting Technique : AdaBoost Classifier**

Further training the model on the train set on the basis of boosting techniques that is using AdaBoost Classifier and then predicting on the test set to check for the desired output.



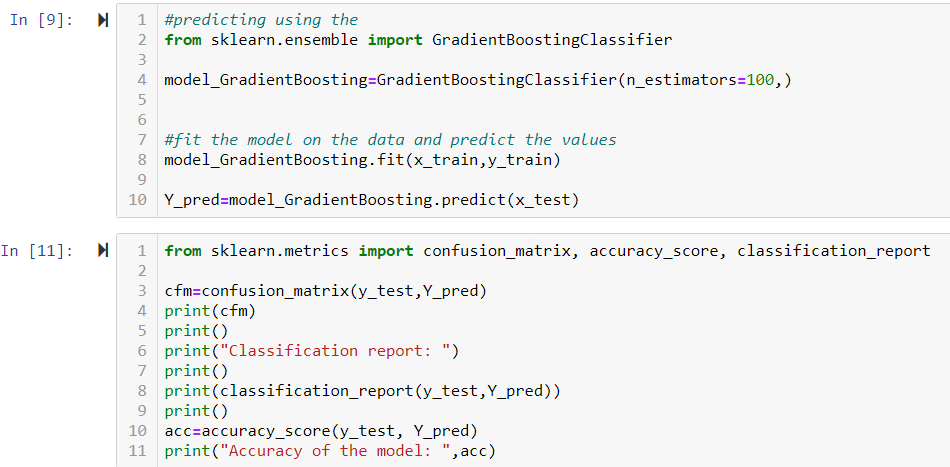


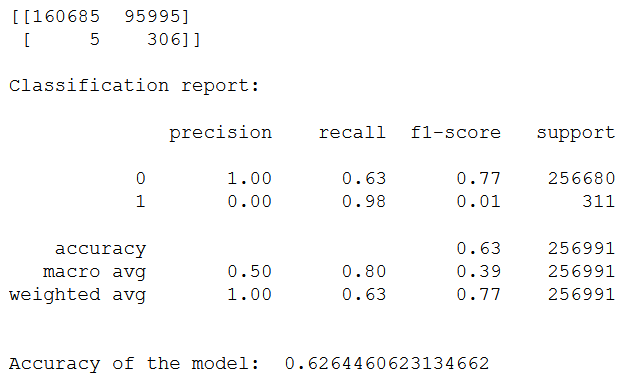
From the confusion matrix we can conclude that Type II error is 1 & Type I error is very high i.e. 186407 and the accuracy of the model is 0.2746.

In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

* + 1. **Boosting Technique : Gradient Boosting Classifier**

Further training the model on the train set on the basis of boosting techniques that is using Gradient Boosting Classifier and then predicting on the test set to check for the desired output.



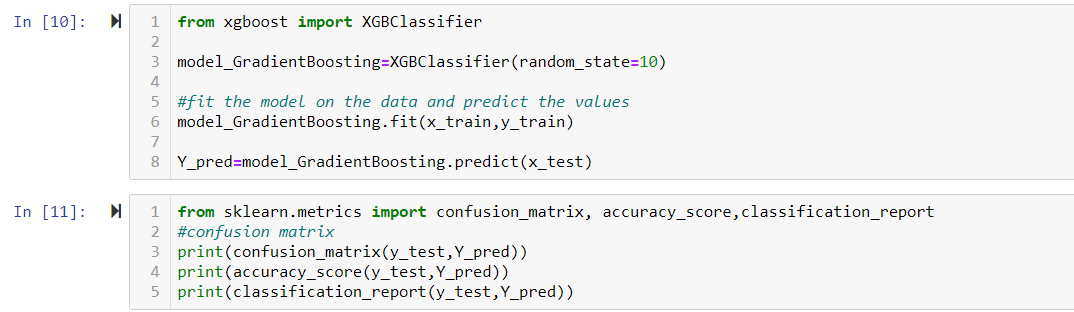


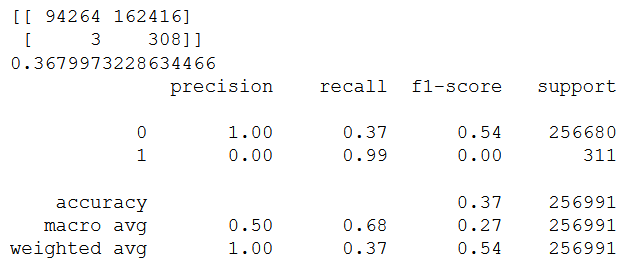
From the confusion matrix we can conclude that Type II error is 1 & Type I error is very high i.e. 95995 and the accuracy of the model is 0.6264.

In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

* + 1. **XGBOOST Classifier:**

Training the model on the train set using XGBoost Classifier and then predicting on the test set to check for the desired output.



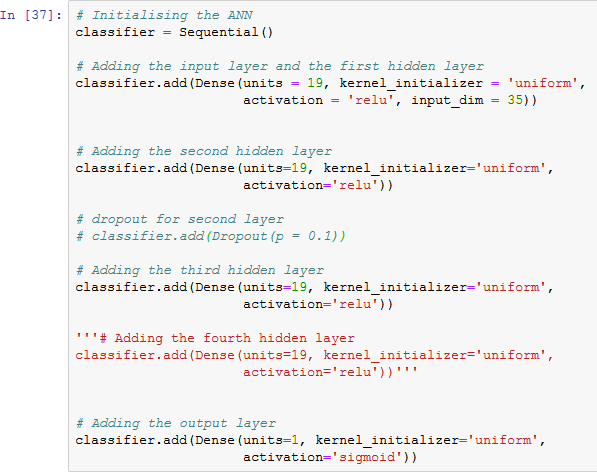


From the confusion matrix we can conclude that Type II error is 1 & Type I error is very high i.e. 162416 and the accuracy of the model is 0.3679.

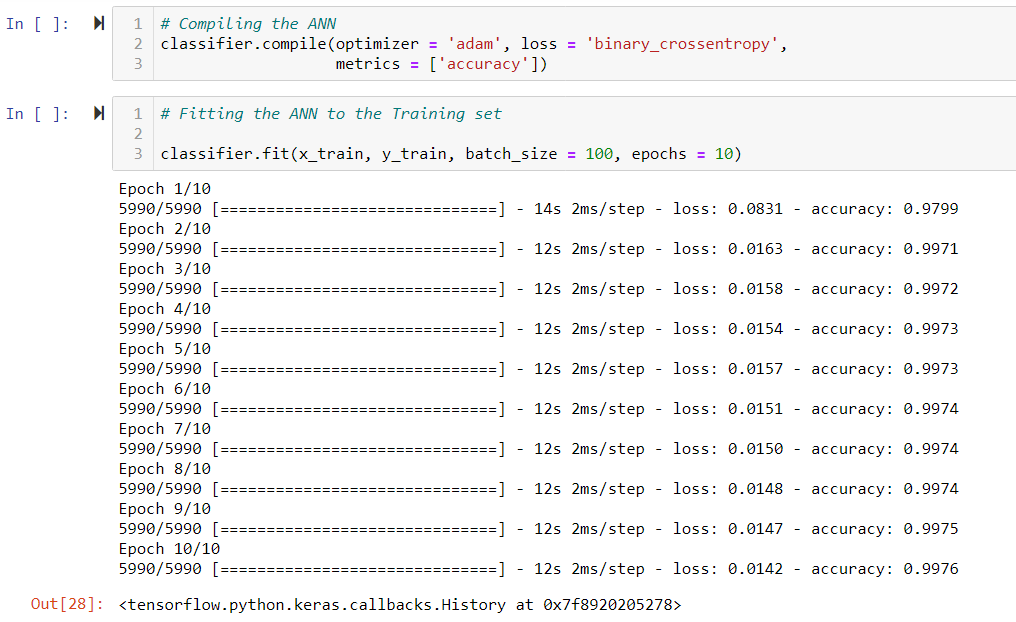
In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

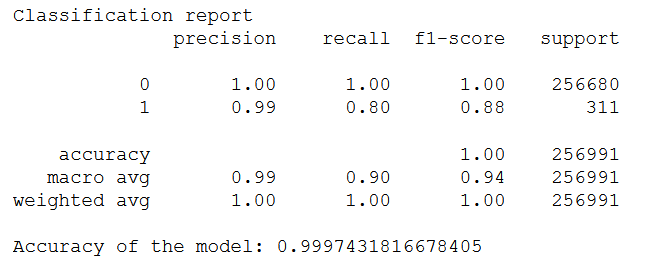
* + 1. **Artificial Neural Networks (ANN)**
       - **Deep learning on Unbalanced Dataset**

After importing all the keras libraries and packages for deep learning, we create the following layers:



Then compiling and fitting the ANN to the training set with batch size of 100 and 10 epochs.

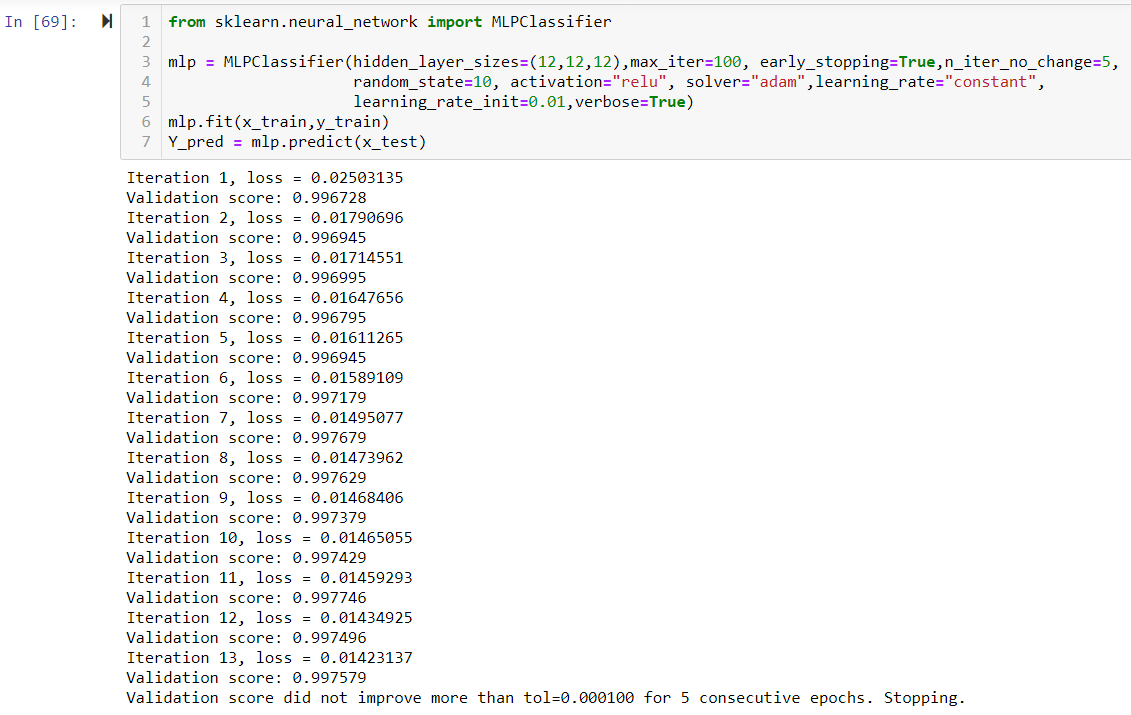


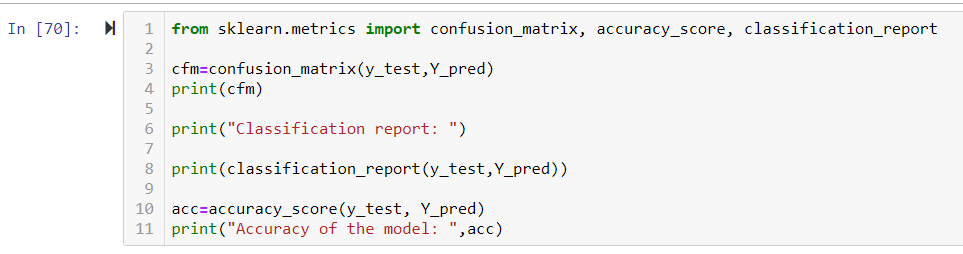


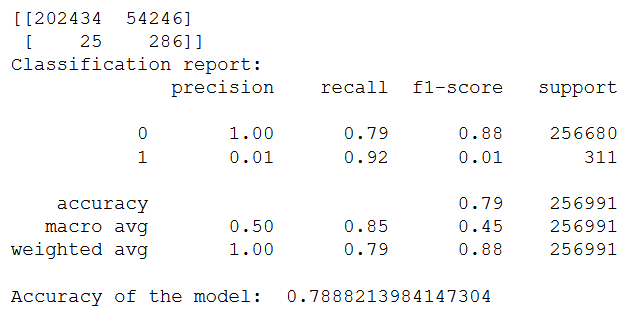
After predicting the test results, we came to conclusion that the accuracy of the model is 0.9997. We have a significant Type 1 Error.

## Artificial Neural Networks (ANN) using MLP classifier

Training the model on the train set using MLP classifier and then predicting on the test set to check for the desired output.







From the confusion matrix we can conclude that Type II error is 25 & Type I error is very high i.e. 54246 and the accuracy of the model is 0.7888.

In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

## Ensemble models using Voting classifier :

Training the model on the train set by selecting the top 3 models which are concluding by giving almost proper accuracy they are Logistic Regression, Gradient Boosting Classifier and

MLP Classifier and then predicting on the test set to check for the desired output.

## 

## 

## 

## 

## 

## 

From the confusion matrix we have concluded, the overall accuracy achieved by using all the 3 algorithms,it predicted the output based on the highest of voting and in output we found that that Type II error is 24 & Type I error is very high i.e. 51080 and the accuracy of the model is 0.8011.

In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

## FEATURE SELECTION TECHNIQUES :

## 1. mutual\_info\_classif

## SelectKBest

## We used these feature selection techniques such that we can select the significant variables and predict a better output based on the selection techniques.

## 

## 

## Futher we plotted a bar graph consisting all top 25 best selected variables.

## # Using the selectKboost techniques to predict the significant variables

## 

## # Based on mutual\_info\_classif , SelectKBest selected Top 25 Variables

## 

## Further we have used this variables and created a dataframe out of itand we further splitted the observation, considering the period variable as index which was created earlier from issue date variable and splitting the dataset based on train and test and we applied scaling techniques on the train and test data test and ran the model to get the desired output.

## 

## 

## 

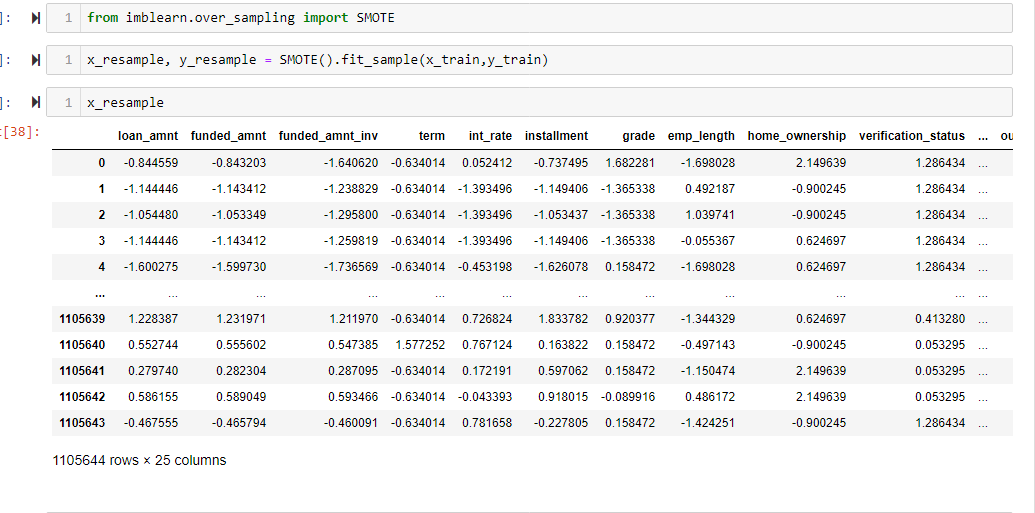
Based on the mutual classifier techniques, the output achieved we can conclude that in the confusion matrix Type II error is 3 & Type I error is very high i.e. 130027 and the accuracy of the model is 0.4940.

In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

**5.1 SMOTE- Synthetic Minority Oversampling Techniques:**

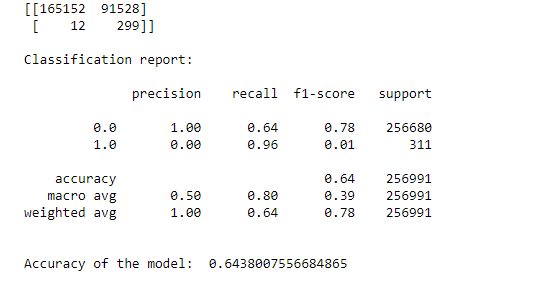
Smote means oversampling of the minority class, it generally adds synthetic data in the minority class to such extent that the observations of minority class matches the majority class.

-Smote techniques are generally used to overcome the over fitting problem.

****

**Using Gradient boosting classifier**

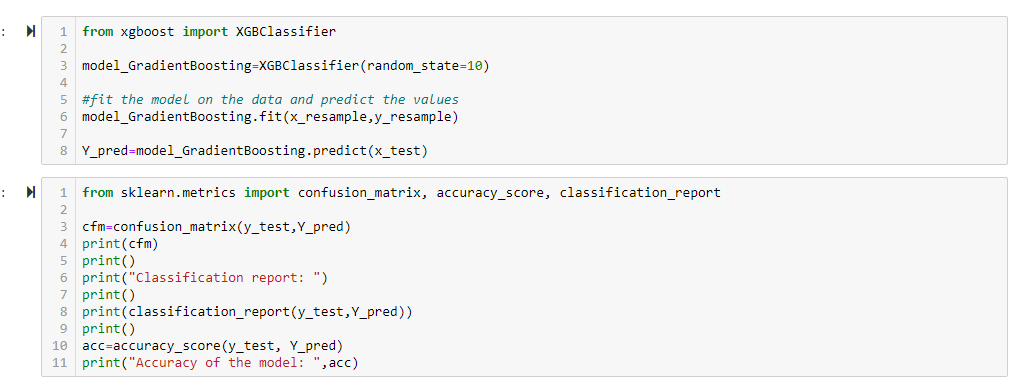
****

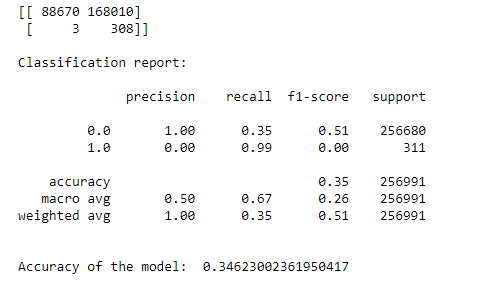
****

Based on the Gradient boosting techniques after scaling with smote technique, the output achieved we can conclude that in the confusion matrix Type II error is 12& Type I error is very high i.e. 91528 and the accuracy of the model is 0.6438.

In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

**Using XG Boost technique**

****

****

Based on the XGBoost techniques after scaling with smote technique, the output achieved we can conclude that in the confusion matrix Type II error is 3 & Type I error is very high i.e. 168010 and the accuracy of the model is 0.3462.

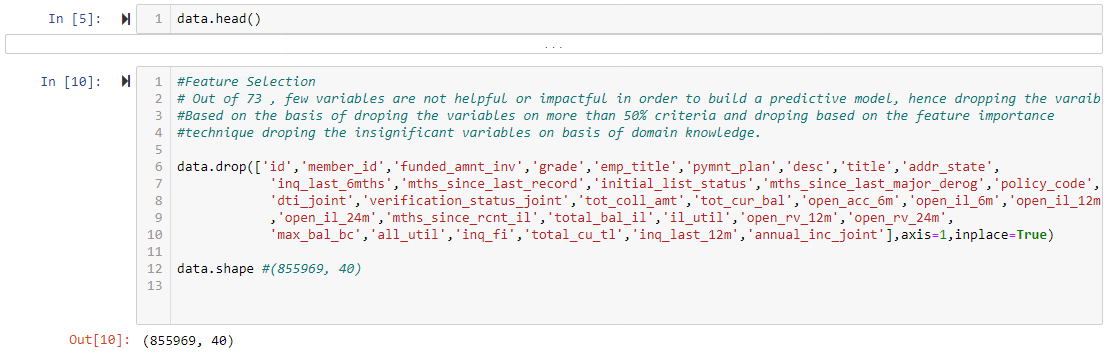
In this model, the Type II error is low but the Type I error is extremely high which is not acceptable. We have a significant Type 1 Error.

1. **Feature selection- Selections of new significant variables.**

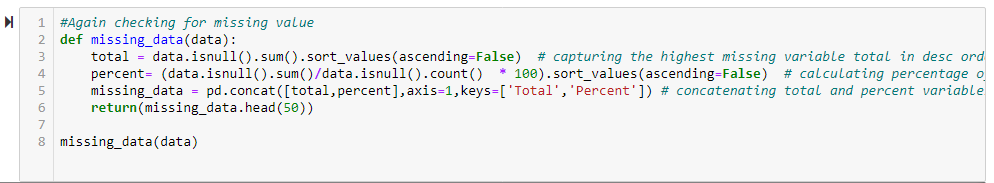
* using new feature selection to decrease the type-1 error

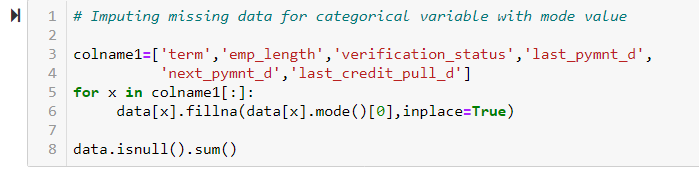
## 6.1.1 Phase II - Feature Engineering

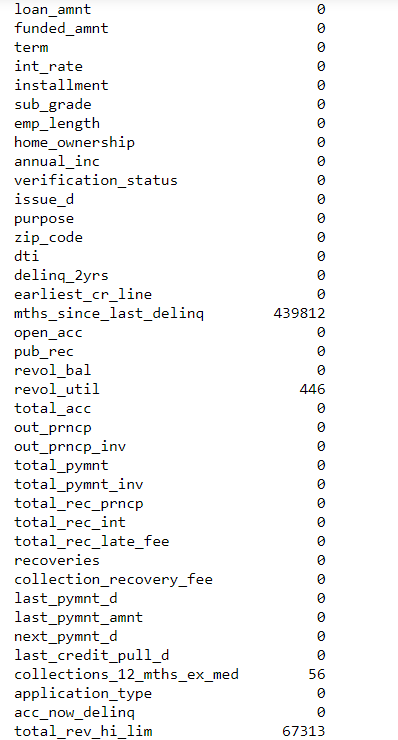
After building the Logistic Regression, Decision Tree and the ANN on dataset as well as applying tuning method and feature selection technique, we created a new dataframe with different variable selections to check the effect on the model and also decrease the errors.

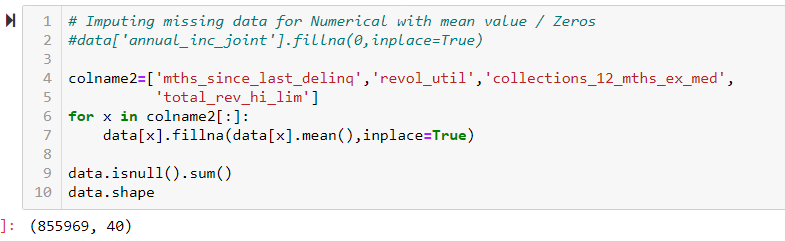


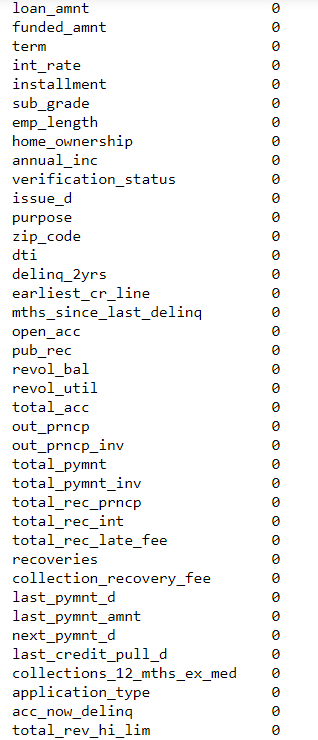
**# Checking for missing values and imputing missing values with mean for numercal variables and mode for categorical variables.**





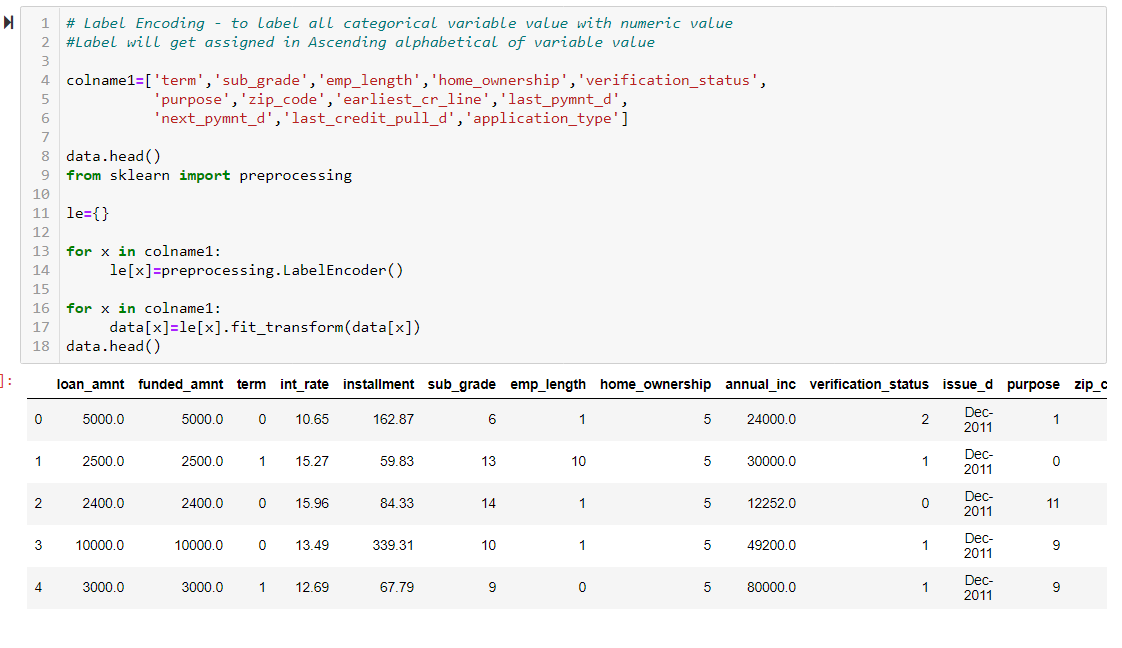






#**Using Label Encoding technique**

# Label Encoding will label all categorical variable value with numeric value and Label will get assigned in Ascending alphabetical of variable value



#After imputing the missing data for categorical variable with mode and for numerical variable with mean value/zeros, we split the dataset into Train and Test.



## Logistic Regression on the new dataset

## 

## 

Here the Type I error has reduced from 41 to 34 as compared to the logistic regression model on the previous dataset.

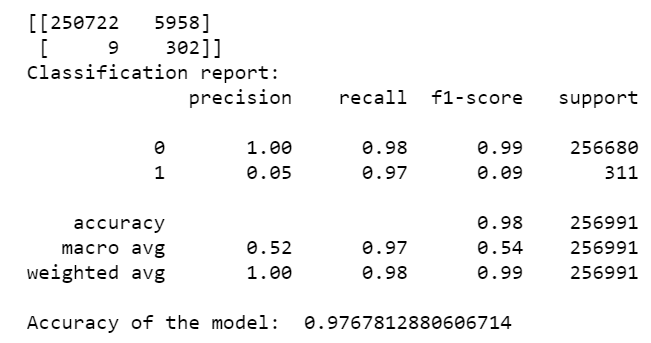
## # tuning the model

## 

## 

The Type I error has decreased from 39 to 34 as compared to the previously tuned model.

## Decision Tree Classification on the new dataset



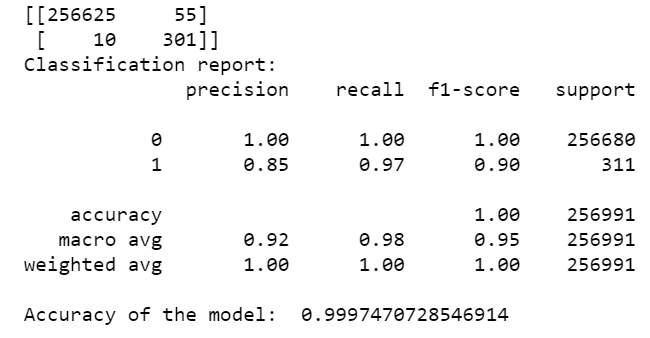
Observing the previous Decision Tree model and the current one, the Type I error has drastically reduced from 183708 to 5958, while the Type II error has increased from 1 to 9.

## Gradient Boosting Classifier

Using the ‘sklearn.ensemble’ library and importing ‘GradientBoostingClassifier’, we build a

model as shown:



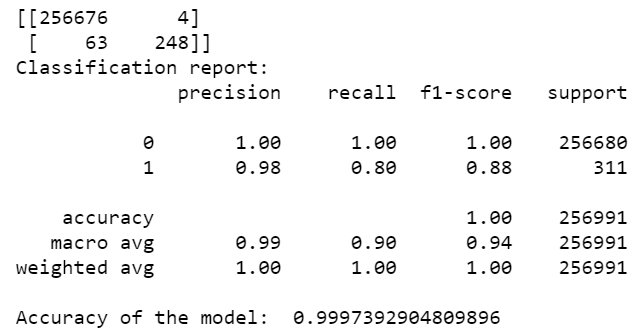


Type I error - 55 Type II error –10

Comparing the gradient boosting model with the other models, we observe that this is the most accurate model with minimum error

## ANN on the new dataset

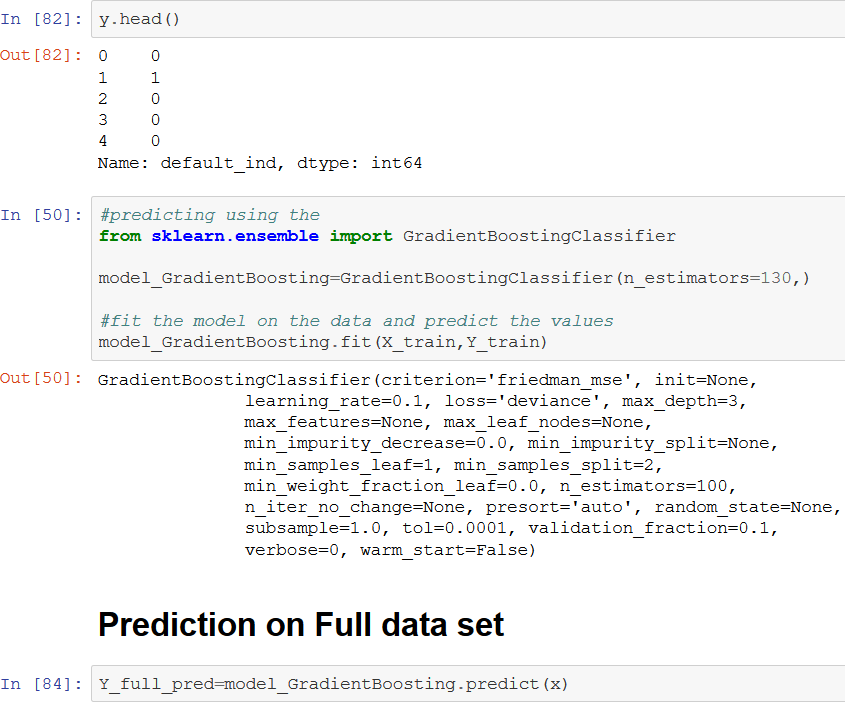
## On Unbalanced data

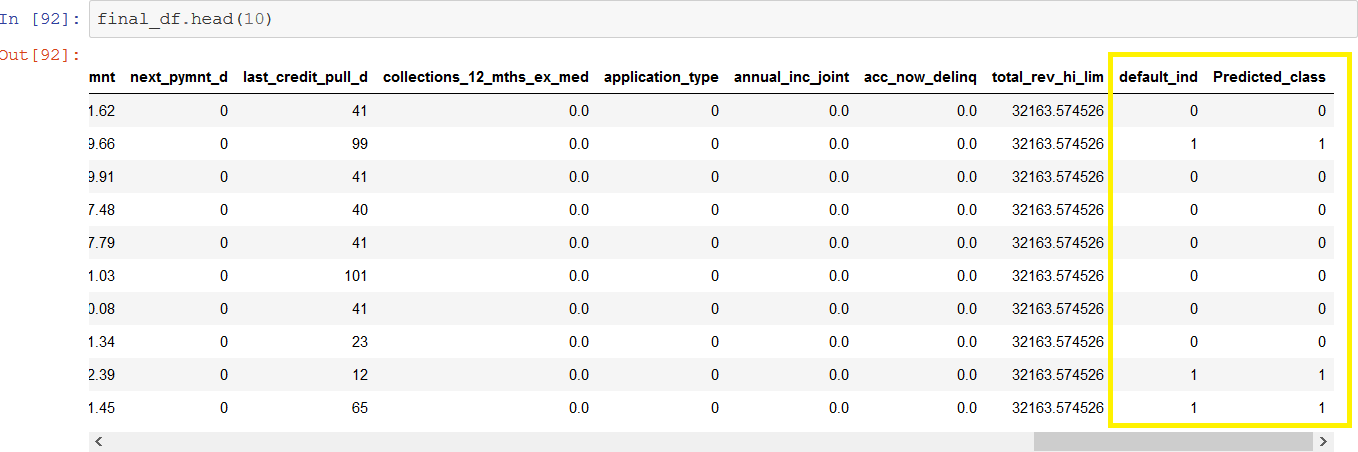


**CHAPTER 5: FINAL MODEL**

Now that we know that Gradient Boosting with tuning is our best model, we will now

perform prediction on the whole dataset which consists of around 8.55 lacs observations and then concatenated the predicted variable to the dataset for final submission to the client for comparing the actual and predicted values.





**CHAPTER 6: CONCLUSION**

Out of all the algorithms used, Gradient Boosting Classifier gave us the least Type II error as 10 after tuning the model. Since our priority for accuracy was based on Type II error, Gradient Boosting model with tuning was preferred. But after examining all the models, ANN is the best model because despite of not tuning, it gave minimum error.

If in case of some cases where Type I error is more significant, ANN is the best suited model.