horizontal line

CAPSTONE PROJECT REPORT

**PREDICTION OF LOAN DEFAULTER**

**PGPDSE – FT Bangalore July21**



***Submitted by***

**GROUP 4**

**Saurabh Kunal**

**Shubham Mishra**

**Mohith R**

**Yugendhar K**

**Siraj Ahmed**

Mentor:

VIKAS CHANDRA

# Table of Contents

Sl NO 1.1

1.2

2

2.1

2.2

3

4

4.1

Topic

Industry Review - Current practices, Background Research Literature Survey

Data set and Domain Basic Statistics of Dataset Data Dictionary

Variable categorization

Pre-Processing Data Analysis

Statistics about the numerical features in the dataset

Page No 3

4-5

4

4

5-8

9

9

10-12

5

5.1

5.2

6

6.1

Project Justification Project Statement, Complexity involved Project Statement

Complexity Involved Data Exploration (EDA)

Relation between variables - Univariate Analysis for numerical features

13

13

13

14

14-16

6.2

6.3

7

8

9

10

11

Relation between variables - Analysis for Categorical features Relation between variables- Multi-variate Analysis

Presence of outliers and its treatment Statistical significance of variable Class imbalance and its treatment Feature Importance

Base model Performance

17-19

20

21-22

23

24

25

26

# Industry Review - Current practices, Background Research

The loan is one of the most important products of the financial institutes. All the institutes are trying to figure out effective business strategies to persuade more customers to apply for loans.

In the lending industry, investors provide loans to borrowers in exchange for the promise of repayment with interest. If the borrower repays the loan, then the lender would make profit from the interest.

However, there are some customers who are not able to pay off the loan after their application are approved**, then** the lender loses money. Therefore, lenders face the problem of predicting the risk of a borrower being unable to repay a loan so, many financial institutions take several variables into account when approving a loan. Determining whether a given borrower will fully pay off the loan or cause it to be charged off (not fully pay off the loan) is difficult. In this study, loan behaviors are analyzed with several machine learning models. The dataset that used in this paper is from Lending Club, a website that connects borrowers and investors over the Internet.

# Literature Survey - Publications, Application, past and undergoing research

Benkler, Yochai (2002). “Coase’s Penguin, or, Linux and” The Nature of the Firm””. In: The Yale Law Journal. Chen, Dongyu and Chaodong Han (2012).

“A Comparative Study of online P2P Lending in the USA and China”. In: Journal of Internet Banking and Commerce. Freedman, Seth and Ginger Zhe Jin (2008).

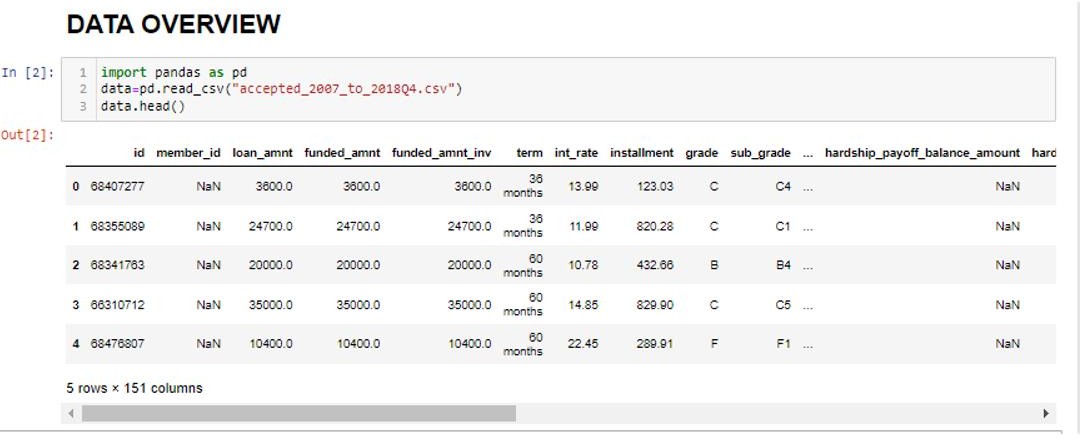
“Do Social Networks Solve Information Problems for Peer-to-Peer Lending? Evidence from Prosper.com”. In: Working Paper. Ge, Ruyi et al. (2017).

“Predicting and Deterring Default with Social Media Information in Peer-to-Peer Lending”. In: Journal of Management Information Systems. Guo, Yanhong et al. (2015).

“Statistical Classification Methods in Consumer Credit Scoring: A Review”. In: Journal of the Royal Statistical Society. Series A (Statistics in Society). Iyer, Rajkamal et al. (2009).

“Screening in New Credit Markets Can Individual Lenders Infer Borrower Creditworthiness in Peer-to-Peer Lending?” In: Working Paper Series. MIT Sloan School of Management, Harvard Kennedy School and NBER. Ke, Guolin et al. (2017).

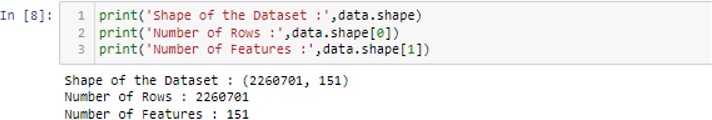
“LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: Advances in Neural Information Processing Systems 30. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 3146– 3154. url: <http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-> boosting decision-tree.pdf. Kim, Aleum and Sung-Bae Cho (2019).



We are reading the dataset ‘accepted\_2007\_to\_2018Q4.csv ’and displaying all the first five rows and columns of the dataset using head() .

# Basic Statistics of Dataset

Using. shape function we get to know the number of rows and columns in dataset



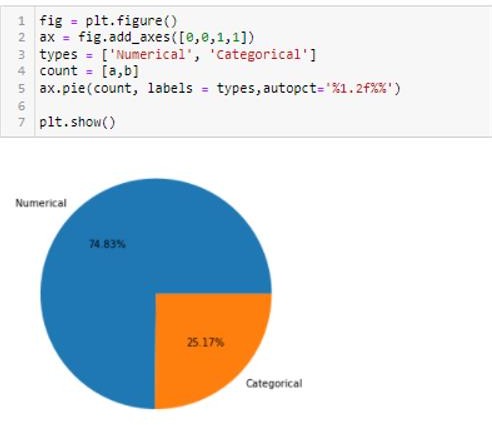
# Data Dictionary

**This shows data description of each and every attribute used**

**Columns Description**

|  |  |
| --- | --- |
| Loan\_amount | The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value |
| Term | The number of payments on the loan. Values are in months and can be either 36 or 60. |
| Int\_Rate | Interest Rate on the loan |
| installment | The monthly payment owed by the borrower if the loan originates. |
| grade | LC assigned loan grade |
| Emp\_length | The job title supplied by the Borrower when applying for the loan. |
| home\_ownership | The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER |
| annual\_inc | The self-reported annual income provided by the borrower during registration. |
| verification\_status | Indicates if income was verified by LC, not verified, or if the income source was verified |
| purpose | A category provided by the borrower for the loan request. |
| dti | A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income. |
| pub\_rec | Number of derogatory public records |
| revol\_bal | Total credit revolving balance |
| revol\_util | Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit. |
| mort\_acc | Number of mortgage accounts. |
| pub\_rec\_bankruptcies | Number of public record bankruptcies |
| target | Targer column With values default and fully paid |
| application\_type\_Joint App | Indicates whether the loan is an individual application or a joint application with two co-borrowers |
| credit\_history | Duration of last Credit |
| creditline\_ratio | ratio between open credit lines and total credit lines |
| avg\_fico | Average of Fico range high and low  The upper boundary range the borrower’s FICO at loan origination belongs to.  The lower boundary range the borrower’s FICO at loan origination belongs to. |
| balance\_income | Balance income is the money used by borrower for other expenses other than the loan.  A person with a high balance income is more likely to repay a loan |

# Variable categorization (count of numeric and categorical)



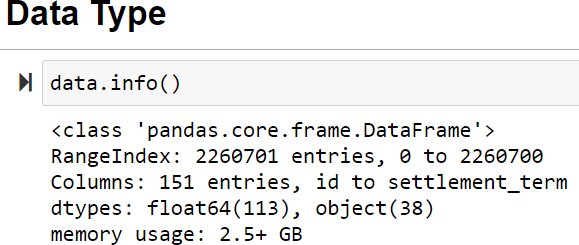
Numerical variables **count -113**

Categorical variables count –38

From the above pie chart we can observe that 74.83% data is numerical and only 25.17 % is categorical data in the data set

# Pre-Processing Data Analysis (count of missing/ null values, redundant columns, etc.)

Now, we know the shape of the data and also, regarding type of the features that are present in the data. let’s get into the information regarding the data

There is function called info() in pandas which gives us the concise summary regarding the dataset. It includes the index dtypes, column dtypes, null values and memory usage.

From the info () we get to know that,

-There are null values/missing values present in the dataset

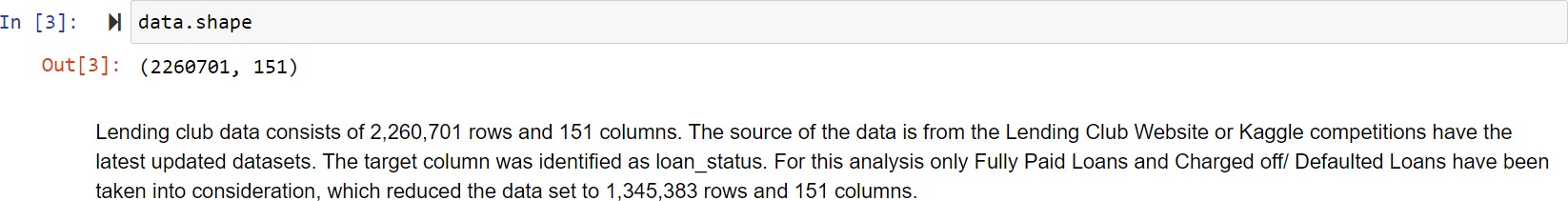
-We have range of index from 0 to 1687859

-Total memory usage is 296.2+ MB

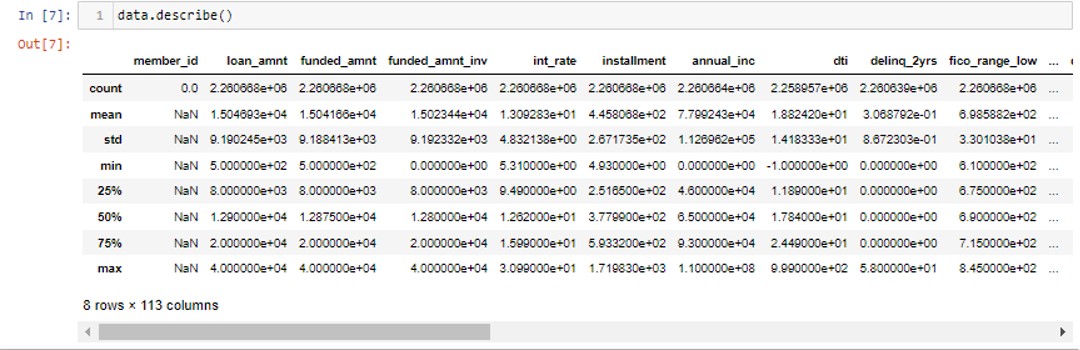
-There are 15 floating datatypes and 8 object. i.e., categorical

From .info function we can see that there is a range of index from 0 to 2260700. We see memory

Usage as 2.5 GB. There are 113 float and 38 object data type.

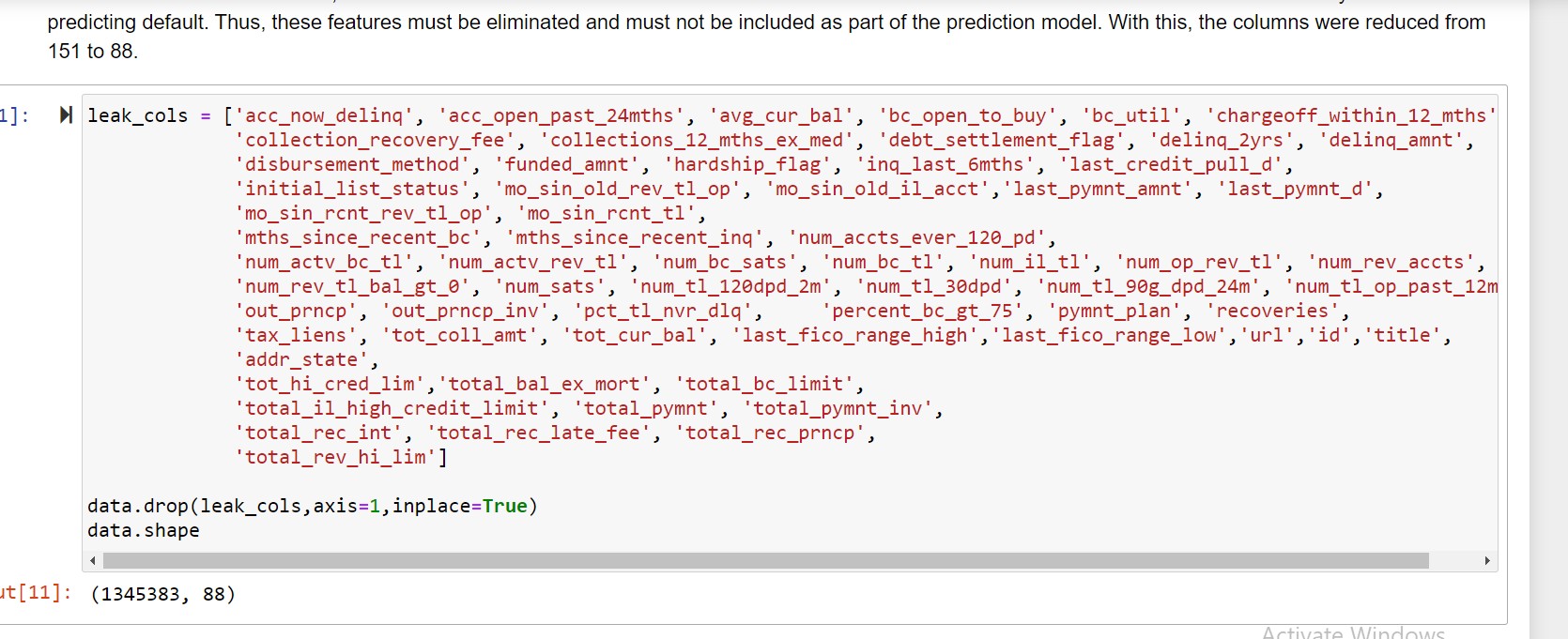


# Statistics about the numerical features in the dataset



* Through five-point summary we get to know the count, mean, std, min, max values of numerical features.

# Redundant columns



* 1. **NULL VALUES**



*-The last row of the data set is having all null values, which can be removed.*

*-And the rest of the null values were also dropped as the null value percentage is less than 6% of the total data*.

From above we see columns where there are 70% null values . We kept a threshold value of 70% for missing data, which meant removing columns with more than 70% null values.Since majority of the features had more than 90% missing data imputations wouldn't be very helpful. This reduced the number of features from 88 to 46

# Project Justification Project Statement, Complexity involved, Project Outcome

* 1. **Project Statement: -**

Determining whether a given borrower will fully pay of the loan or will default.

Within this project, we intend to build a machine learning algorithm for the purpose of correctly identifying if a person, given with certain characteristics, has a high likelihood to default on a loan. There is a certain methodology that needs to be followed in order to properly load effective predictors - data cleaning, exploration, and feature engineering.

# Complexity Involved:

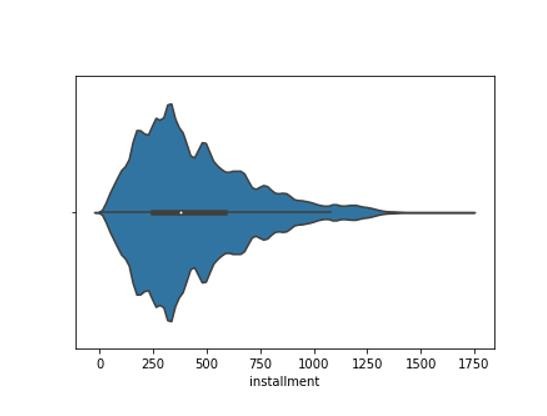
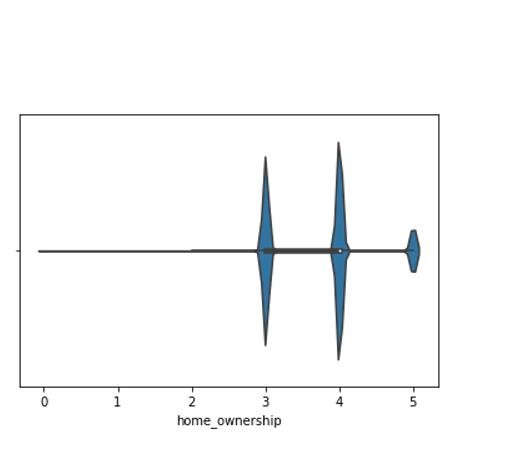
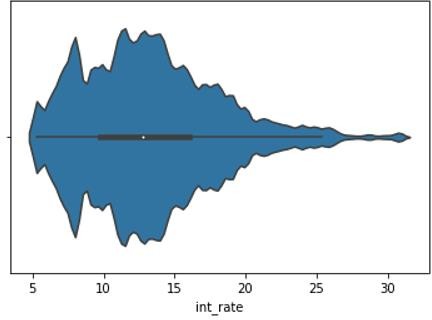
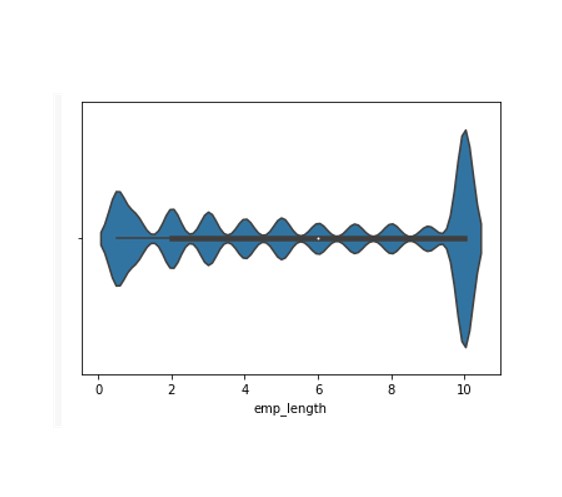
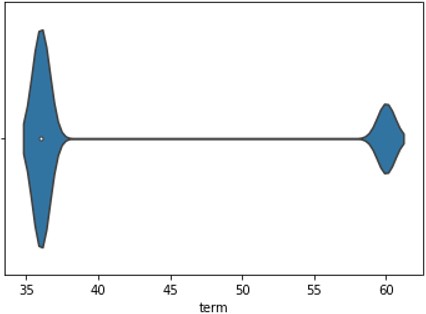
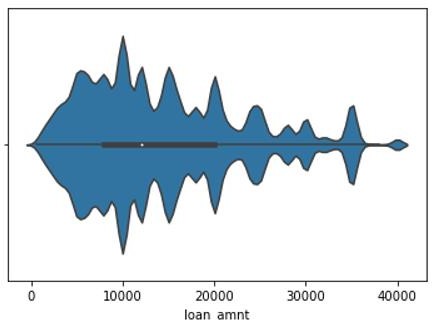
From heatmap we don’t see any strong correlation of any predictors with the target variable which is complexity.



# Data Exploration (EDA)

* 1. **Relation between variables -** Univariate Analysis for numerical features

# Numeric Feature:

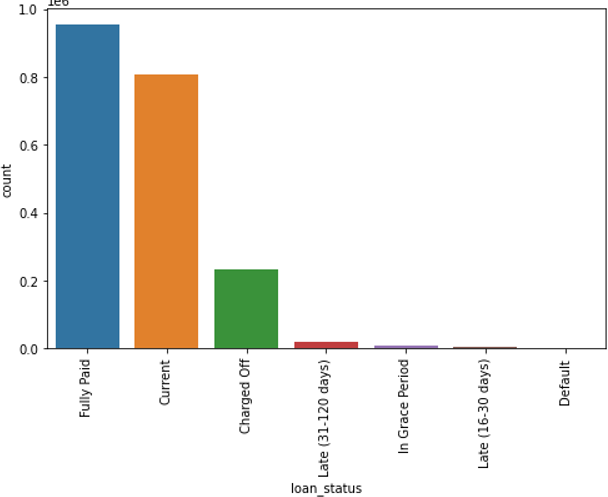


Above is the plot of some numerical columns with distribution and skewness .We see high skewness among numerical columns and will be transforming the numerical features

* 1. **Relation between variables –**

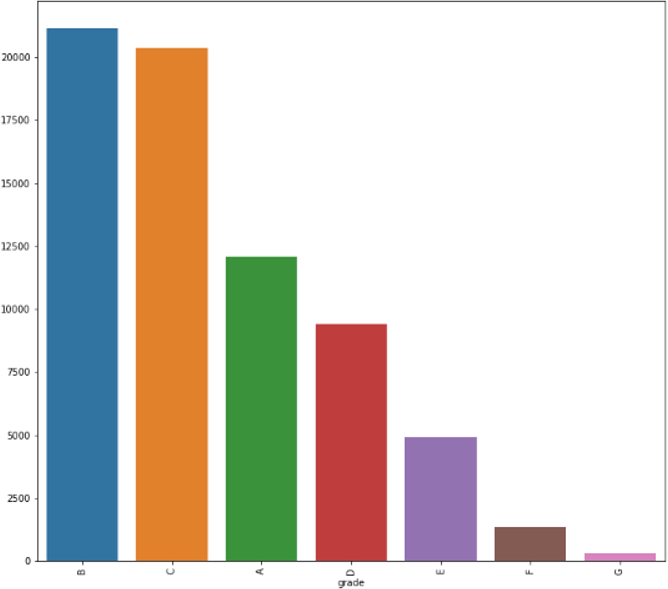
Analysis for Categorical features

Feature: **loan\_status**



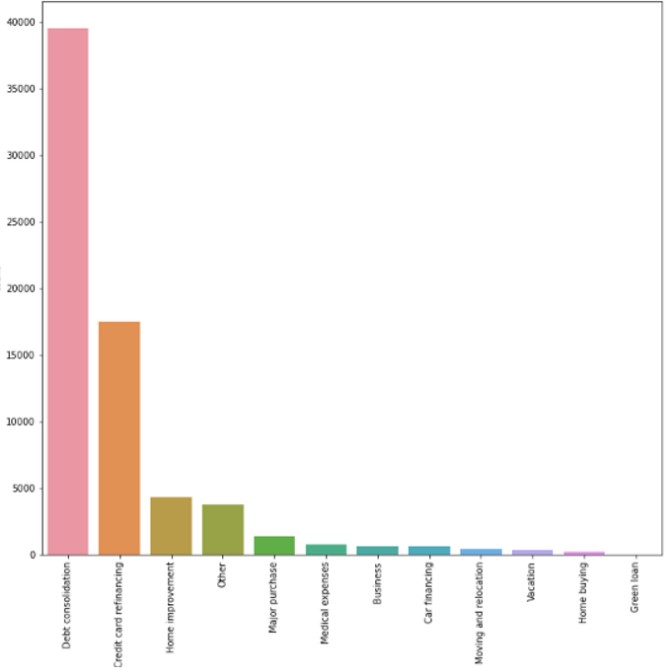
We see maximum count of full paid loans. This is target column and has been converted to integer 1 and 0. Fully paid has been converted to 0 and Default and charged off has been converted to 1

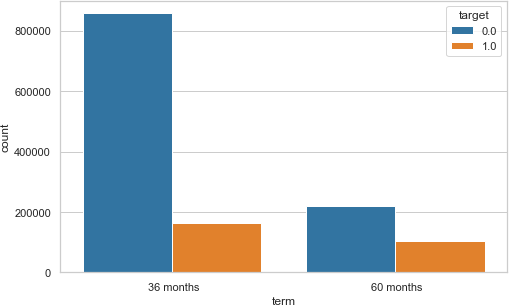
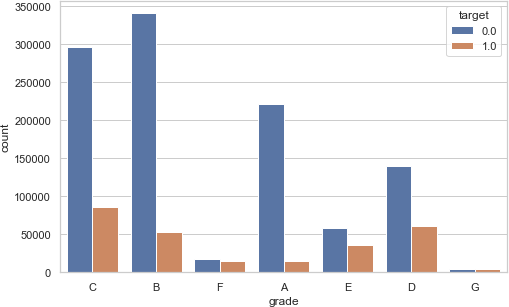
Feature : **Grade**

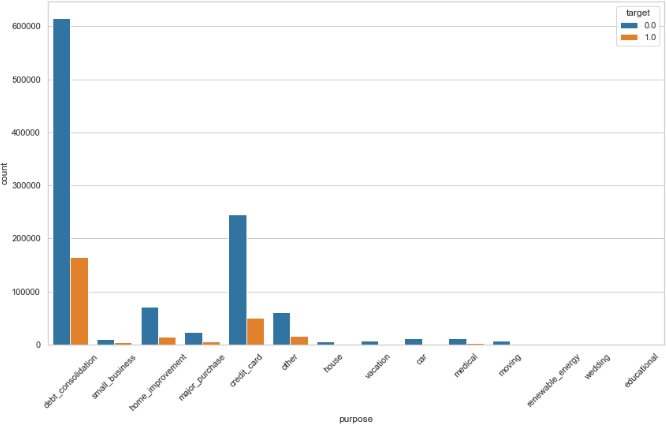
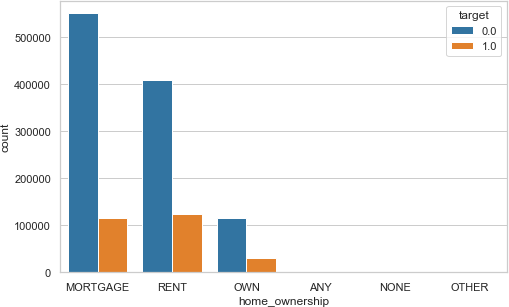


We see maximum count of Grade B followed by Grade C,A,D,E,F and G .

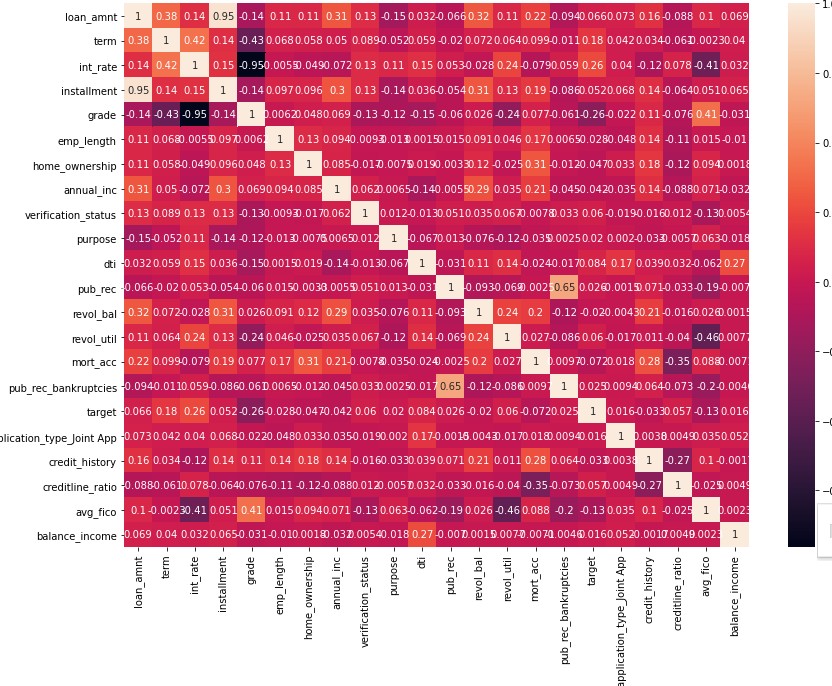
Feature : **Purpose**







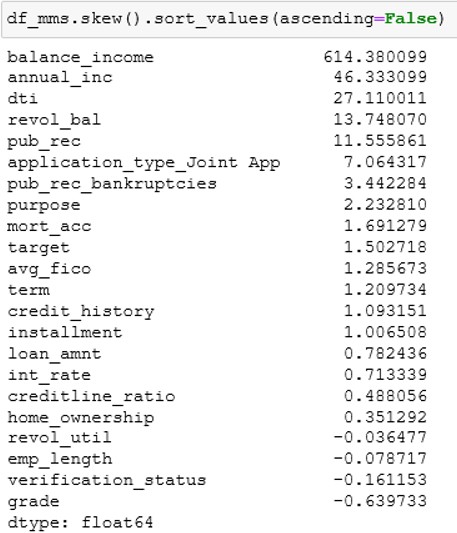
**6.2** Relation between variables- Multi-variate Analysis:



* We see there is no correlation between target and other predictors
* Loan \_amt and installment is highly correlated having value 0.95
* Pub\_rec\_bankrupties and pub\_rec has high collinearity with value 0.65
* Avg\_fico and revol\_util has moderate correlation of -0.46
* Avg\_fico and grade have moderate correlation of 0.41
* All other attributes are having weak correlation as evident from above graph

# Presence of outliers and its treatment

Outlier is a datapoint that differs significantly from other observations. An outlier may be due to variability in the measurement or it may indicate experimental error. An outlier can cause serious problems in statistical analyses.

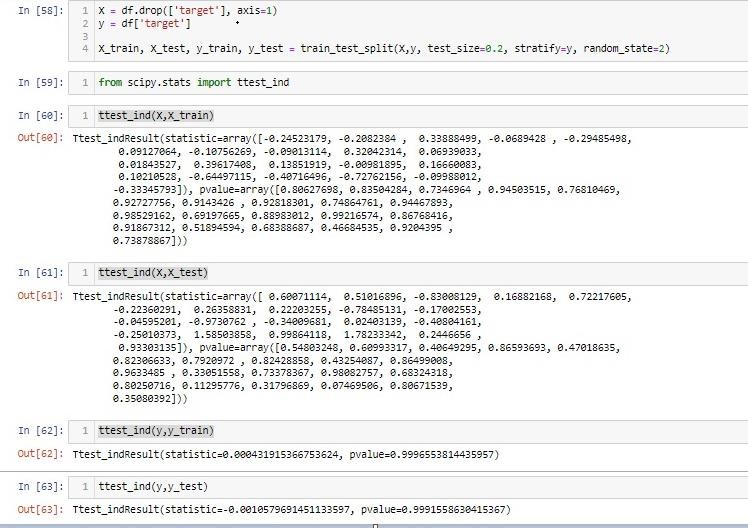


As our dataset is highly skewed, it has lot of outliers present. So, removing outliers through IQR/Z score method doesn’t reduce the skewness and outliers at all.

Hence power transformation is the best method to deal with outliers for this dataset. After applying power transformer we still observed very high skewness on few columns and further log transformation was applied to reduce the skewness and now skewness of all the columns is between 0 and 2.

# Statistical significance of variable:

**Numerical Feature**



Since p-value of all columns is above .05 we can conclude that train and test data is true representation of overall data

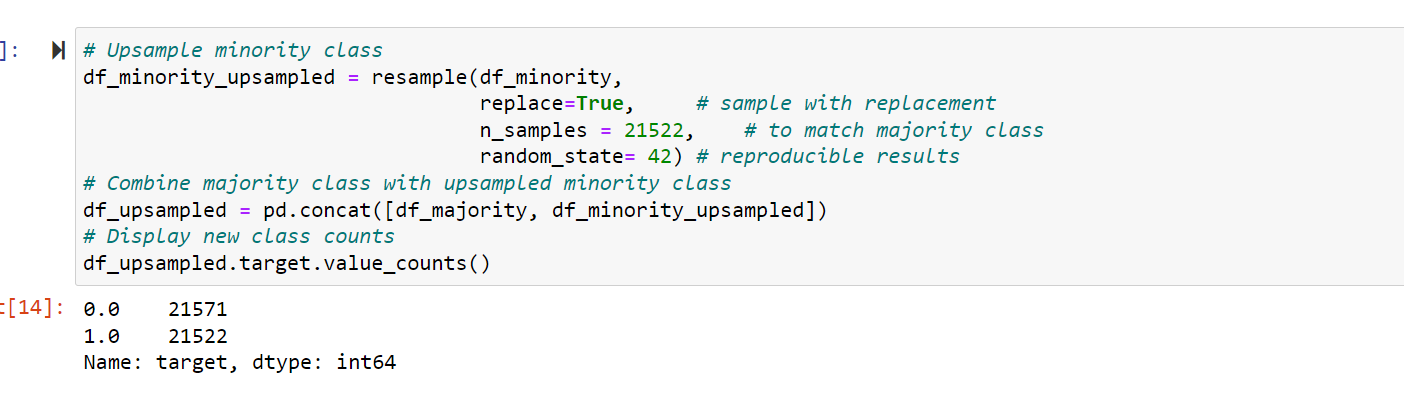
# Class imbalance and its treatment:

Since our sample is too big, we have considered random sampling technique as below:

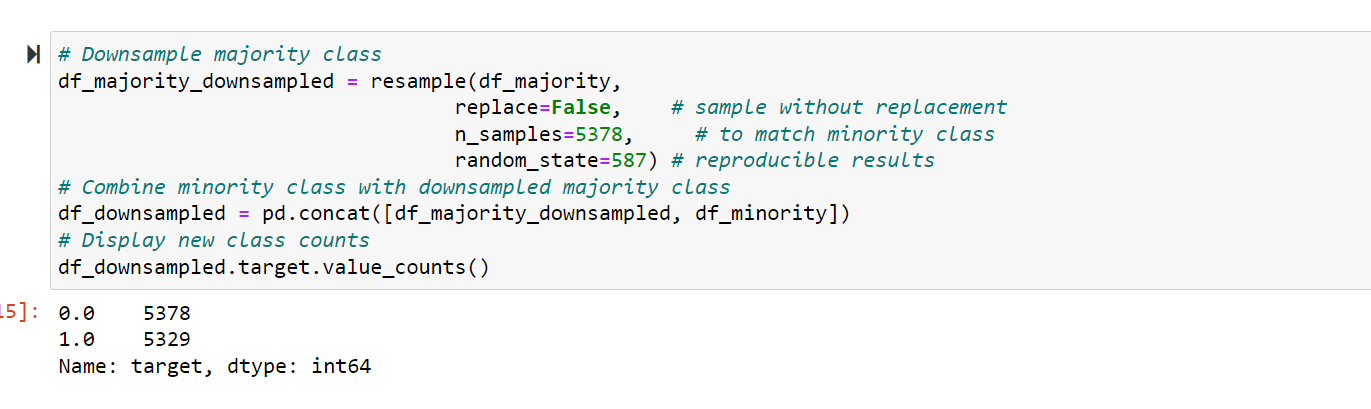
**Random Oversampling:** Random oversampling involves randomly duplicating examples from the minority class and adding them to the training dataset. Examples from the training dataset are selected randomly with replacement. This means that examples from the minority

class can be chosen and added to the new “more balanced” training dataset multiple times; they are selected from the original training dataset, added to the new training dataset, and then

returned or “replaced” in the original dataset, allowing them to be selected again.



**Random Under sampling**: Random under sampling involves randomly selecting examples from the majority class to delete from the training dataset. This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated until the desired class distribution is achieved, such as an equal number of examples for each class. This approach may be more suitable for those datasets where there is a class imbalance although a sufficient number of examples in the minority class, such a useful model can be fit.



# Feature Engineering:

We used domain understanding and literature survey to reduce the number of attributes by using the relations between them to create new variables and drop the parent variables.

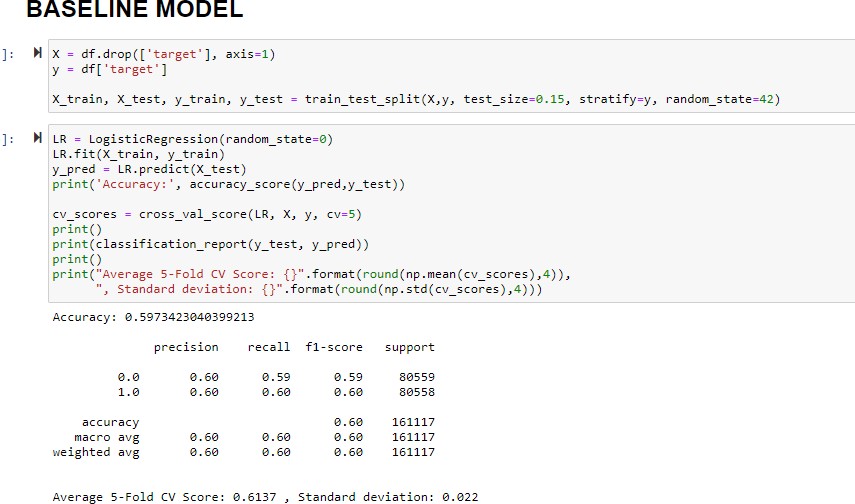
This helped us manage the huge dataset we were working on. EXAMPLES:

* To find the credit history or duration of a credit line we use earliest credit line open date and issue date of loan. *Since we found the relation between the 2 features, we can drop the parent.*
* Credit line ratio is the ratio between open credit lines and total credit lines.
* Balance income is the money used by borrower for other expenses other than the loan.

A person with a high balance income is more likely to repay a loan**.**

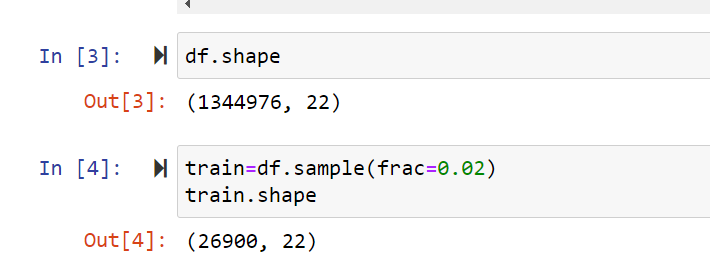


# Base model Performance:



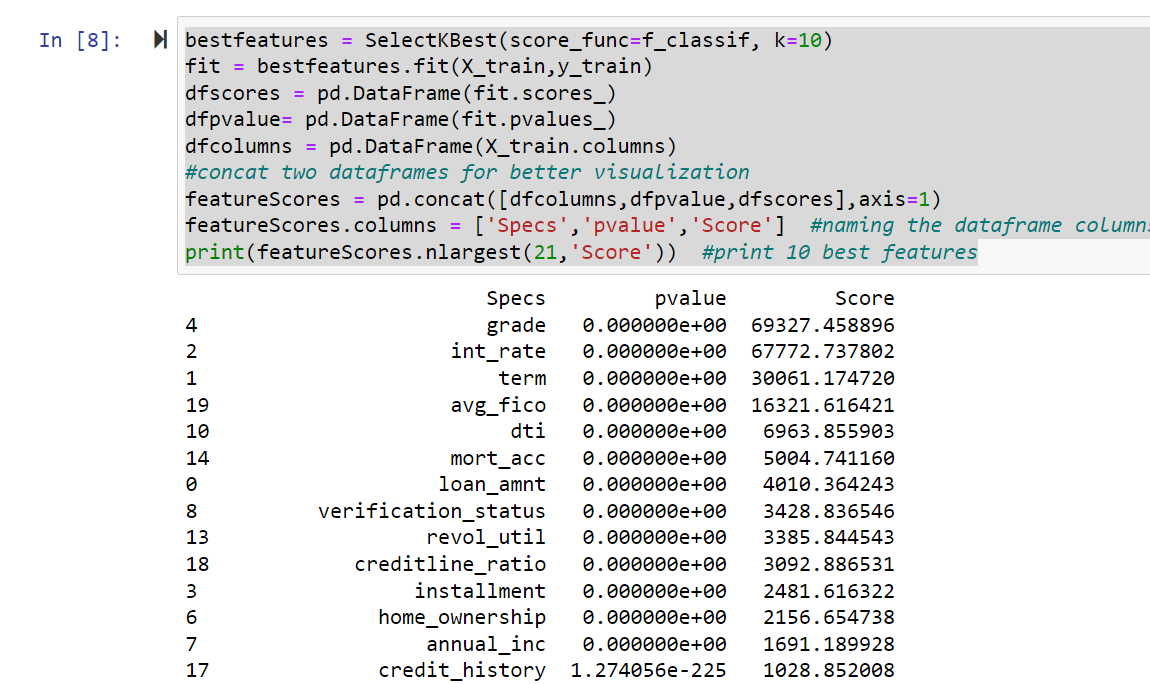
We have created baseline model using LogisticRegression and see average accuracy using 5 cross validations as 61 %.

**12 .Data Sample Taken:**

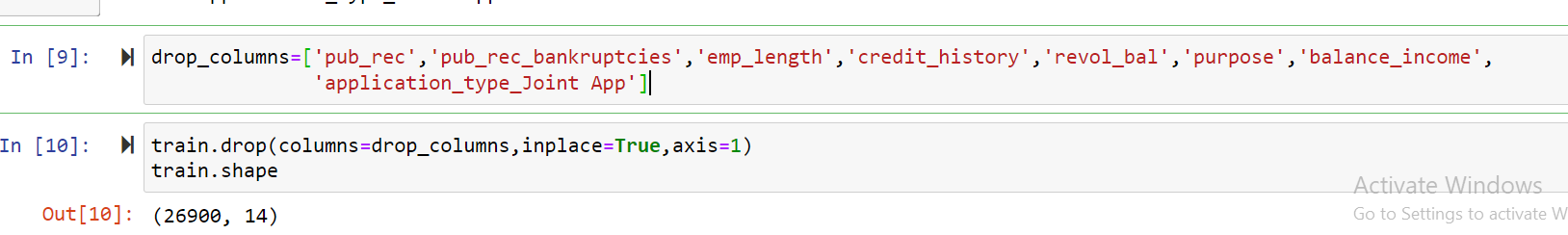


We have taken .02 % of entire sample and above is the number of rows after filtering data

1. **Best features Taken using KBest score:**



Below columns dropped based on K score .Updated number of columns is 14 after dropping below columns.



**14. Models used along with their performance**

**Metrics Considered:** F1 Score, Recall

## Logistic Regression Performance:

|  |  |  |
| --- | --- | --- |
| **Data with Normal Sampling** | **Data with Over Sampling** | **Data with Under Sampling** |
| Train accuracy 80.3 % | Train accuracy 64.10% | Train accuracy 64.36% |
| Test accuracy 80.12 % | Test accuracy 66.50% | Test accuracy 66.76% |
| Train AUC score 69.23 | Train AUC score 70.06 | Train AUC score 70.22 |
| Test AUC score 70.28 | Test AUC score 70.79 | Test AUC score 70.71 |
| F1 score .76 | F1 score .65 | F1 score .68 |

* 1. Naive Bayes

|  |  |  |
| --- | --- | --- |
| **Data with Normal Sampling** | **Data with Over Sampling** | **Data with Under Sampling** |
| Train accuracy 69.81 % | Train accuracy 64.88% | Train accuracy 64.58% |
| Test accuracy 71.17 % | Test accuracy 65.25% | Test accuracy 68.09% |
| Train AUC score 69.81 | Train AUC score 70.44 | Train AUC score 70.26 |
| Test AUC score 71.17 | Test AUC score 71.24 | Test AUC score 71.33 |
| F1 score .76 | F1 score .65 | F1 score .68 |

* 1. KNN

|  |  |  |
| --- | --- | --- |
| **Data with Normal Sampling** | **Data with Over Sampling** | **Data with Under Sampling** |
| Train accuracy 80.74 % | Train accuracy 100% | Train accuracy 67.79% |
| Test accuracy 79.92 % | Test accuracy 99.83% | Test accuracy 67.79% |
| Train AUC score 75.87 | Train AUC score 1 | Train AUC score 74.58 |
| Test AUC score 67.42 | Test AUC score 99.57 | Test AUC score 72.60 |
| F1 score .80 | F1 score 1 | F1 score .68 |

* 1. Decision Tree

|  |  |  |
| --- | --- | --- |
| **Data with Normal Sampling** | **Data with Over Sampling** | **Data with Under Sampling** |
| Train accuracy 80.43 % | Train accuracy 88.02 % | Train accuracy 64.09% |
| Test accuracy 80.26 % | Test accuracy 84.05% | Test accuracy 57.58% |
| Train AUC score 71.22 | Train AUC score 95.80 | Train AUC score 70.36 |
| Test AUC score 69.47 | Test AUC score 94.23 | Test AUC score 70.74 |
| F1 score .80 | F1 score .84 | F1 score .58 |

* 1. Random Forest

|  |  |  |
| --- | --- | --- |
| **Data with Normal Sampling** | **Data with Over Sampling** | **Data with Under Sampling** |
| Train accuracy 80.58 % | Train accuracy 75.05 % | Train accuracy 71.09% |
| Test accuracy 80.27 % | Test accuracy 71.45% | Test accuracy 64.63% |
| Train AUC score 77.17 | Train AUC score 83.40 | Train AUC score 79.46 |
| Test AUC score 69.46 | Test AUC score 81.14 | Test AUC score 75.09 |
| F1 score .80 | F1 score .71 | F1 score .65 |

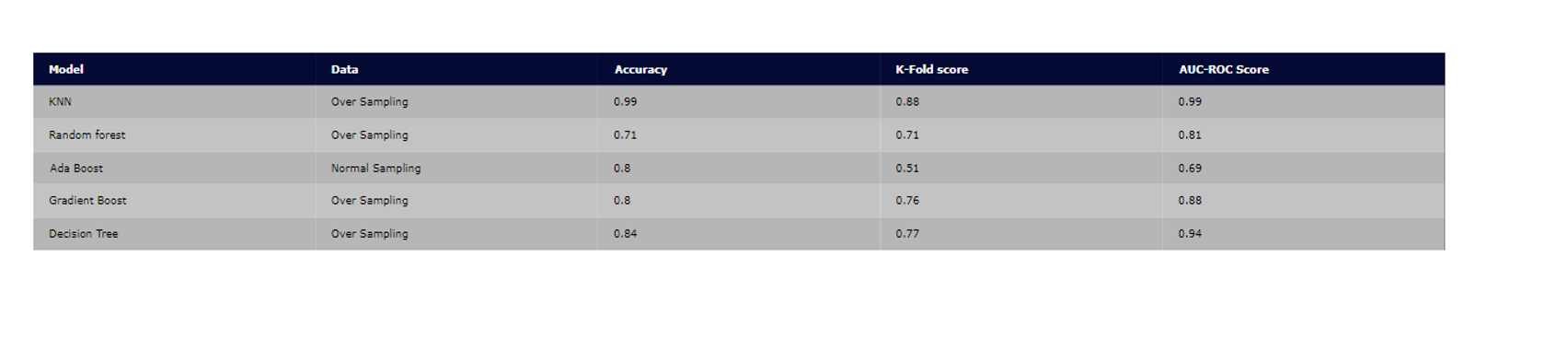
* 1. AdaBoost

|  |  |  |
| --- | --- | --- |
| **Data with Normal Sampling** | **Data with Over Sampling** | **Data with Under Sampling** |
| Train accuracy 80.31 % | Train accuracy 67.06 % | Train accuracy 66.09% |
| Test accuracy 80.10 % | Test accuracy 65.24% | Test accuracy 62.63% |
| Train AUC score 71.55 | Train AUC score 73.25 | Train AUC score 71.46 |
| Test AUC score 69.45 | Test AUC score 72.19 | Test AUC score 70.09 |
| F1 score .80 | F1 score .65 | F1 score .63 |

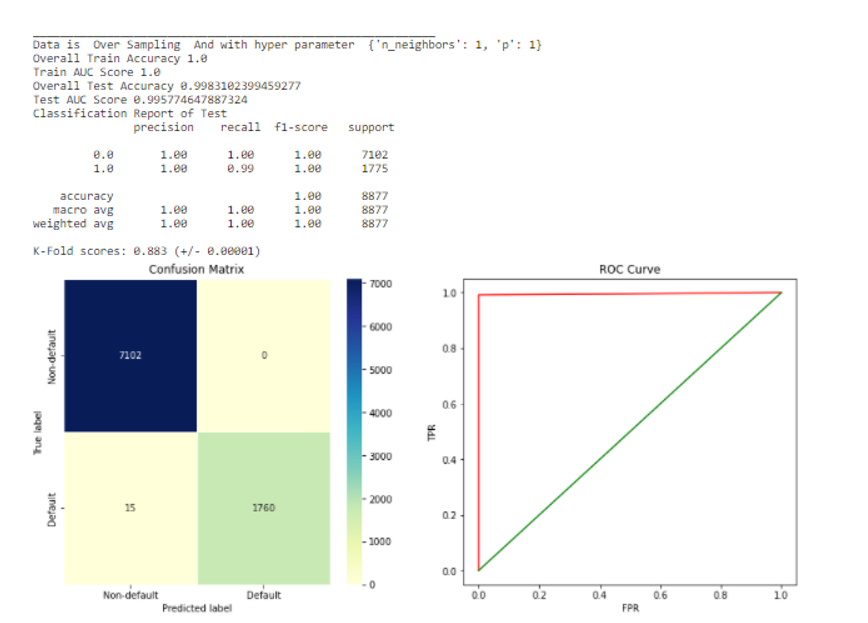
* 1. Gradient Boosting

|  |  |  |
| --- | --- | --- |
| **Data with Normal Sampling** | **Data with Over Sampling** | **Data with Under Sampling** |
| Train accuracy 85.71 % | Train accuracy 83.06 % | Train accuracy 80.09% |
| Test accuracy 78.35 % | Test accuracy 80.24% | Test accuracy 68.63% |
| Train AUC score 86.20 | Train AUC score 91.54 | Train AUC score 88.75 |
| Test AUC score 66.51 | Test AUC score 88.99 | Test AUC score 79.86 |
| F1 score .78 | F1 score .81 | F1 score .69 |

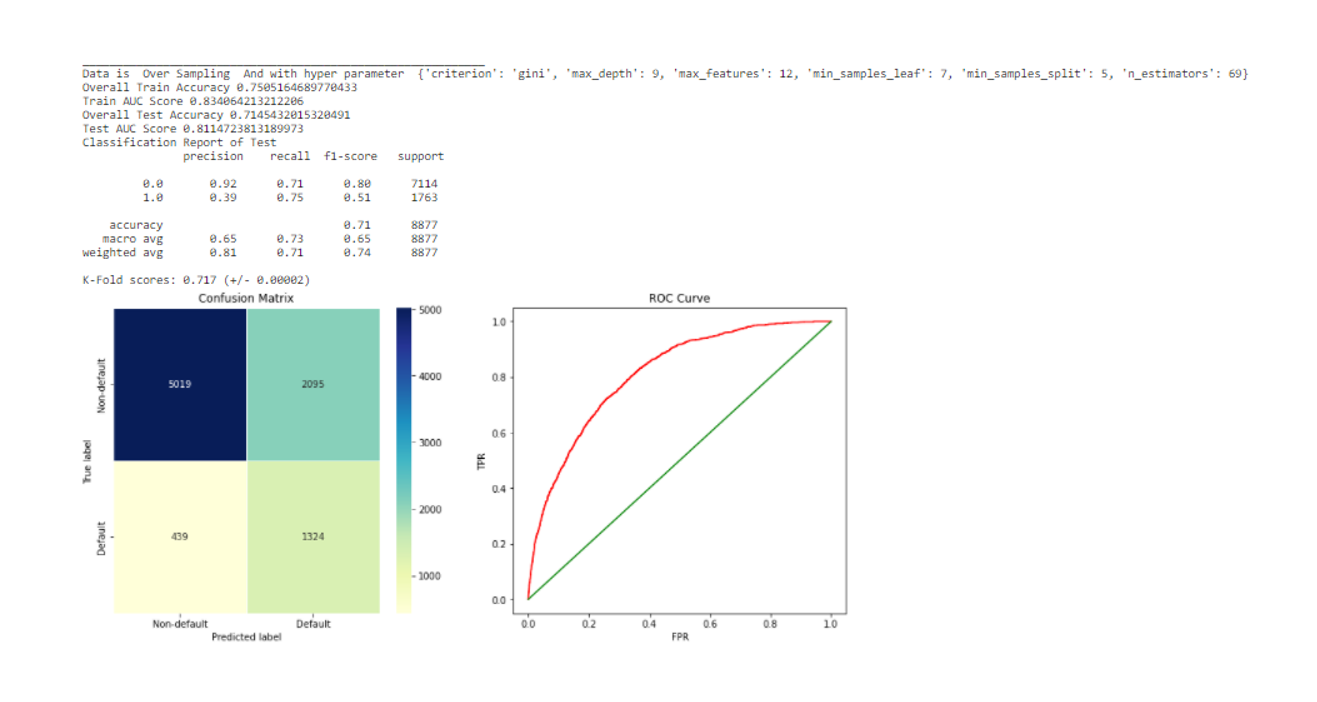
**14. Models with best performance and corresponding graphs**



**KNN Over sampling**



**Random Forest Over Sampling**



**Ada Boost Normal Sampling**

