PROJECT REPORT

Loan Default Prediction Using Machine Learning

Submitted towards the partial fulfillment of the criteria for award of

Genpact Data Science Prodegree

by Imarticus

Submitted By:

Dhananjay Mishra

Mumal Tanwar

Neha Nirwan

Course and Batch: DSP 03 2020

Abstract

The loan is one of the most important products of the financial institutes. The major profit all the financial institute or banks earn through interest earned through loans. All the financial institutes are trying to figure out effective business strategies to persuade more customers to apply the loan in their institute. However, there are some customers who are not able to pay off the loan after their application is approved. Therefore, 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. If the lender is too strict, fewer loans get approved, which means there is less interest to collect. But if they are too lax, they end up approving loans that default. In this study, we have used more than 8 lakh records and 73 different parameters to study the behaviors of a customers whose loans were approved and some of them were defaulters and some of them were not, which will help the institute in future to predict the whether the customer will be defaulter or not. We have used different techniques to solve clean the data, transform the data, and used different machine learning algorithms like logistic regression, decision tree, random forest, SVC etc., to predict the outcomes and after using ML algorithm we have used several cross validation technique and hyper parameter techniques to make our ML mode better, after that we have tested the machine learning model on a separate data which we have not used to create ML model to verify the result which we have obtained from the training data. In the end we have compared the result of all the ML models and select one best out of all the ML models.

Acknowledgement

We are using this opportunity to express my 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. I am 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 Jitendra Gautam as our mentor. He has readily shared his immense knowledge in data analytics and guide 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 DSP program.

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

Date: 26 December 2020

Dhananjay Mishra

Mumal Tanwar

Place: Jaipur

Neha Nirwan

Certificate of Completion

I hereby certify that the project titled "**XYZ Corporation Lending Data Project**" was undertaken and completed under my supervision by Dhananjay, Mumal and Neha from the batch of DSP 03(Jaipur) (March 2020)

Mentor: Jitendra Gautam

Date: 26 December 2020

Place – Jaipur

Table of Contents

List of Figures

CHAPTER 1: INTRODUCTION

1.1Title & Objective of the study

The '**XYZ Corporation Lending Project'** is the project under the BFSI domain. In this project we need to predict that after taking the loan whether the customer will be able to return the loan or he will be a defaulter (person who cannot return the loan) using the past records of the corporation. In this work we will use of the variables which are provided by the organization in their past records. The past record provided by the organization contains data from 2007 to 2015 with more than 8 lakh records.

1.2Need of the Study

The bank/loan providing corporation run their business on the interest they get from the loan which they provide to the individual customers or the other organization. But when the bank/loan providing corporation provides the loan to their clients there is a risk involved in this, that whether the client will be able to pay back the loan or not. The bank/loan providing corporation verifies a lot of documents and inquire about their client but sometimes the client who takes the loan either runs away or won't pay back the loan, in such cases the bank/loan providing corporation suffers a huge loss and this can also lead to the closing of the bank/loan providing corporation. Thus, it is very important for the bank/loan providing corporation to be confident that the client will repay the loan amount.
For such purpose we will make a machine learning model which will help us to predict that the customer will be able to pay back the loan or not if provided by using certain variables.

1.3 Business or Enterprise under study

XYZ Corporation is under study. The dataset provided by the corporation contains information of their previous customers from year 2007 to 2015. The dataset contains different variables like income, loan amount, credit history, purpose of loan, defaulter or not.

1.4 Data Sources

The data source is provided by XYZ Corporation for the year 2007 to 2015 which contains more than 8 lakhs of records and 73 different variables which will help to predict whether the client will be defaulter or not.

Out of 73 variables 72 variables are independent variables like income of client, loan amount, purpose of the loan, employment title, employment length, credit history etc. and one variable is dependent variable which is having two values 0 for the one who were non-defaulter and 1 for those who were defaulter.

1.5 Tools & Techniques

Tools: Jupyter Notebook, Python

Techniques: Logistic Regression, Support Vector Classifier, Decision Tree, Random Forest, Gradient Boosting,XG Boost

CHAPTER 2: DATA PREPARATION AND UNDERSTANDING

A machine learning model is generally broken down into following steps: Data acquiring, data cleaning, data exploring, building models and finally presenting the result of the model. There are various techniques involved in each of the step.

Fig-1 Steps of ML model

2.1 Data Extraction and Cleaning:

- **Missing Value Analysis and Treatment**

  This is generally one of the foremost and important step a data scientist perform i.e., to check whether there is any missing value in the dataset or not. Because if there is any missing values we need to handle those missing values either by removing those records or by removing those variables or we can replace them with different techniques depending on the data type and type of the missing values and the relation they have with other variables.

  Generally, there is a threshold value provided if the null values in any of the variables is more than that threshold we remove that variable. In our case we have taken that threshold as 50%. Any variable having more than 50% null values we will remove that variable. In our data set the fig-2 shows all the variables which are having more than 50% null values, we will directly remove these variables because if we try to fill these variables then there are chances that the values we replace might not be relevant or they might misguide the machine learning model.

	SUM	PERCENT
desc	733715	85.761759
mths_since_last_delinq	439609	51.384585
mths_since_last_record	724444	84.678099
mths_since_last_major_derog	642512	75.101312
annual_inc_joint	855527	100.000000
dti_joint	855527	100.000000
verification_status_joint	855527	100.000000
open_acc_6m	842337	98.458260
open_il_6m	842337	98.458260
open_il_12m	842337	98.458260
open_il_24m	842337	98.458260
mths_since_rcnt_il	842690	98.499521
total_bal_il	842337	98.458260
il_util	844004	98.653111
open_rv_12m	842337	98.458260
open_rv_24m	842337	98.458260
max_bal_bc	842337	98.458260
all_util	842337	98.458260
inq_fi	842337	98.458260
total_cu_tl	842337	98.458260
inq_last_12m	842337	98.458260

Fig -2 Data frame of variables having more than 50% null value

In Fig-3 we can see all the variables which are having less than 50% null values. For these variables we will replace them by appropriate technique, which we will see soon.

	Sum	Percent
emp_title	49387	5.772699
emp_length	43005	5.026726
title	32	0.003740
revol_util	446	0.052132
last_pymnt_d	8792	1.027671
next_pymnt_d	252970	29.568909
last_credit_pull_d	50	0.005844
collections_12_mths_ex_med	56	0.006546
tot_coll_amt	67313	7.868016
tot_cur_bal	67313	7.868016
total_rev_hi_lim	67313	7.868016

Fig-3 Variables having less than 50% null values

We have used median and mode to replace the missing values depending on the data type of the variables. If the variable is of int64 or float64 we have used mean to replace the missing values and if the variable is of object type data, we have used mode to replace the variable.

Variables where we have used median to replace missing values-

```
#tot_coll_amt
#Replacing the NaN with median
data['tot_coll_amt'].fillna(data.tot_coll_amt.median(), inplace=True)
```

```
#revol_util
#Replacing the NaN with median
data['revol_util'].fillna(data['revol_util'].median(),inplace=True)
```

```
#tot_cur_bal
#Replacing the NaN with median
data['tot_cur_bal'].fillna(data['tot_cur_bal'].median(),inplace=True)
```

```
#total_rev_hi_lim
#Replacing the NaN with median
data['total_rev_hi_lim'].fillna(data['total_rev_hi_lim'].median(),inplace=True)
```

```
#Filling null of 'last_credit_pull_d','last_pymnt_d','next_pymnt_d'
for i in ['last_credit_pull_d','last_pymnt_d','next_pymnt_d']:
    med = data_copy1[i].mode()[0]
    data_copy1[i].fillna(med,inplace=True)
```

Fig-4 Variables where we have removed missing values with median

We have replaced the collections_12_mths_ex_med with mode

```
#collections_12_mths_ex_med
#Replace the NaN with mode
data['collections_12_mths_ex_med'].fillna(data.collections_12_mths_ex_med.mode()[0], inplace=True)
```

Fig- 5 Variables where we have replaced missing values with mode

```
data_copy1['emp_length'] = np.where(data_copy1['emp_length']=='10+ years','10 years',data_copy1['emp_length'])
data_copy1['emp_length'] = np.where(data_copy1['emp_length']=='< 1 year','0 year',data_copy1['emp_length'])
data_copy1['emp_length'].fillna('0 year',inplace=True)
data_copy1['emp_length'] = data_copy1['emp_length'].str.split(" ",expand=True)
data_copy1['emp_length'] = data_copy1['emp_length'].astype('float64')
```

Fig- 6 Variables where we have replaced missing values with 0

In the above figure the emp_length was in object type data but we have changed that into int64 type and filled the null value with 0.

- **Handling Outliers**

  In this dataset we have not handled the outliers, the following are the reasons for that
  - There were less number of outliers as compared to the number of records
  - Lack of domain knowledge
- **Feature Extraction**

  There were some variables which were not so important for the machine learning model and can decrease the accuracy of the model so we have removed those variables and we have also used correlation plot to check the multi-collinearity and removed some of the variables which are highly correlated among themselves.

```
#We will now remove the unwanted variables
unneccesary_variables = ['member_id','sub_grade','pymnt_plan','title','zip_code','initial_list_status'
                         ,'policy_code']
data_copy1.drop(unneccesary_variables,axis=1,inplace=True)
```

```
#Droping the variables which are correlated
correlated_variables_removed = ['funded_amnt','funded_amnt_inv','total_pymnt_inv','total_rec_prncp','next_pymnt_d_year'
                                ,'grade','last_pymnt_d_month','out_prncp','installment']
data_copy1.drop(correlated_variables_removed,axis=1,inplace=True)
```

Fig -7 Removal of the variables which are not important and are having multi-collinearity

2.2 Data Dictionary:

LoanStatNew	Description
addr_state	The state provided by the borrower in the loan application
annual_inc	The self-reported annual income provided by the borrower during registration.
annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
collection_recovery_fee	post charge off collection fee
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
Desc	Loan description provided by the borrower
Dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested loan, divided by the borrower's self-reported monthly income.
dti_joint	A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested loan, divided by the co-borrowers' combined self-reported monthly income
earliest_cr_line	The month the borrower's earliest reported credit line was opened
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
emp_title	The job title supplied by the Borrower when applying for the loan.
funded_amnt	The total amount committed to that loan at that point in time.
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
Grade	XYZ corp. assigned loan grade
home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
Id	A unique assigned ID for the loan listing.
initial_list_status	The initial listing status of the loan. Possible values are – W, F
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
Installment	The monthly payment owed by the borrower if the loan originates.

int_rate	Interest Rate on the loan
issue_d	The month which the loan was funded
last_credit_pull_d	The most recent month XYZ corp. pulled credit for this loan
last_pymnt_amnt	Last total payment amount received
last_pymnt_d	Last month payment was received
loan_amnt	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 reflectedin this value.
loan_status	Current status of the loan
member_id	A unique Id for the borrower member.
mths_since_last_delinq	The number of months since the borrower's last delinquency.
mths_since_last_major _derog	Months since most recent 90-day or worse rating
mths_since_last_record	The number of months since the last public record.
next_pymnt_d	Next scheduled payment date
open_acc	The number of open credit lines in the borrower's credit file.
out_prncp	Remaining outstanding principal for total amount funded
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
policy_code	publicly available policy_code=1 new products not publicly available policy_code=2
pub_rec	Number of derogatory public records
Purpose	A category provided by the borrower for the loan request.
pymnt_plan	Indicates if a payment plan has been put in place for the loan
Recoveries	post charge off gross recovery
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.
sub_grade	XYZ assigned loan subgrade
Term	The number of payments on the loan. Values are in months and can be either 36 or 60.
Title	The loan title provided by the borrower
total_acc	The total number of credit lines currently in the borrower's credit file
total_pymnt	Payments received to date for total amount funded
total_pymnt_inv	Payments received to date for portion of total amount funded by investors
total_rec_int	Interest received to date
total_rec_late_fee	Late fees received to date

total_rec_prncp	Principal received to date
verified_status_joint	Indicates if the co-borrowers' joint income was verified by XYZ corp., not verified, or if the income source was verified
zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.
open_acc_6m	Number of open trades in last 6 months
open_il_6m	Number of currently active installment trades
open_il_12m	Number of installment accounts opened in past 12 months
open_il_24m	Number of installment accounts opened in past 24 months
mths_since_rcnt_il	Months since most recent installment accounts opened
total_bal_il	Total current balance of all installment accounts
il_util	Ratio of total current balance to high credit/credit limit on all install acct
open_rv_12m	Number of revolving trades opened in past 12 months
open_rv_24m	Number of revolving trades opened in past 24 months
max_bal_bc	Maximum current balance owed on all revolving accounts
all_util	Balance to credit limit on all trades
total_rev_hi_lim	Total revolving high credit/credit limit
inq_fi	Number of personal finance inquiries
total_cu_tl	Number of finance trades
inq_last_12m	Number of credit inquiries in past 12 months
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
tot_coll_amt	Total collection amounts ever owed
tot_cur_bal	Total current balance of all accounts
verification_status	Was the income source verified

2.3 Exploratory Data Analysis:

EDA is a phenomenon under data analysis used for gaining a better understanding of data aspects like:

1. What are the main features of data
2. What variables have what kind of relationships with other variables
3. Identifying the important patterns inside the data

We will now see some of the important plots –

1. Defualt_ind –

```
sns.countplot(data['default_ind'])
plt.xlabel('Default or Not',fontsize=10)
plt.ylabel('Count',fontsize=10)
plt.title('Count of Defauler and Non-Defaulter',fontsize=15)
plt.show()
```

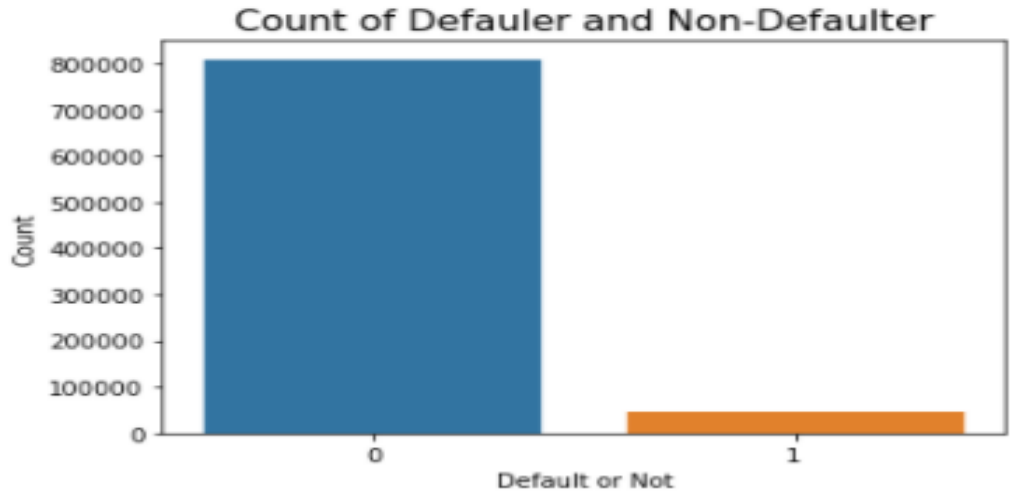

Fig-8 Countplot of variable default_ind

This plot shows the count/frequency of the default_ind variable which is the output variable and tells us how many records were defaulters and how many were not defaulters.
So, it is clearly visible that the dataset is imbalanced.

2. Loan amount

```
plt.figure(figsize=(8,5))
sns.distplot(a=data['loan_amnt'],kde=False,bins=8,color='red')
plt.title('Count of customer on basis of Loan Amount')
plt.ylabel('Count')
plt.show()
```

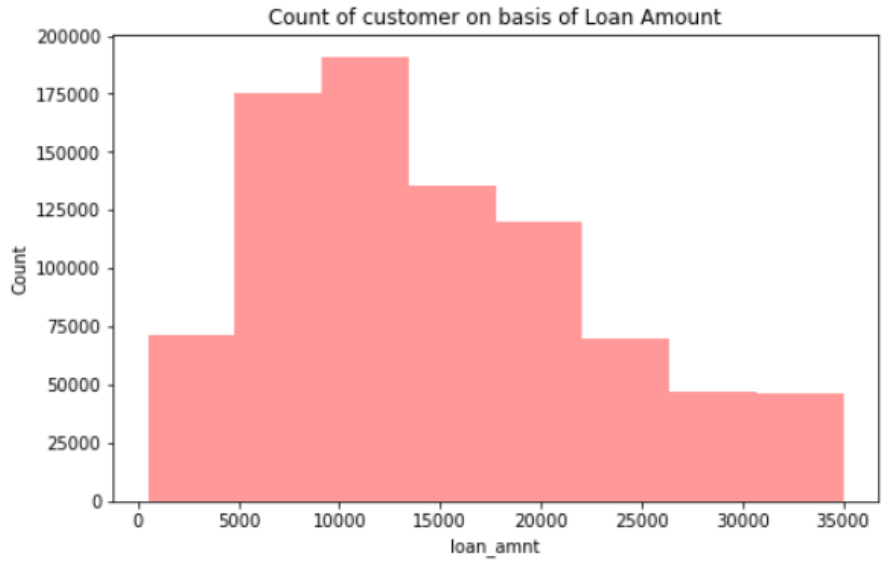

Fig-9 Distplot of variable loan_amnt

- The highest loan amount is of 35000. The distribution of the variable is slightly right skewed.
- From this histogram we can see that there are high number of customers who wants a loan amount of 5000 to 15000.
- There is almost same number of customers who have applied for loan amount of 0 to 5000 and 21000 to 26000.
- There are few customers who have applied for the loan where amount is more than 26000.

3. Term

```
count = data['term'].value_counts()
plt.figure(figsize=(5,5))
plt.pie(x=count,labels=['36 months','60 months'],autopct='%1.2f%%',explode=[0,0.1])
plt.title('Term')
plt.show()
```

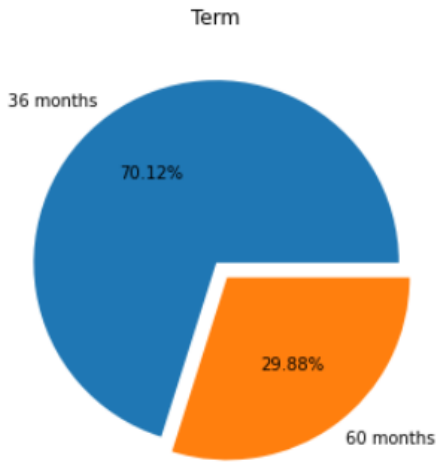

Fig-10 Pie Chart of variable Term

More than 70% of the records belong to the 36 month term and less than 30% belong to 60 months term.

4.Interest Rate

```
plt.hist(data['int_rate'])
plt.xlabel('Interest Rate',fontsize=10)
plt.ylabel('Count',fontsize=10)
plt.title('Frequency of Interest Rate',fontsize=15)
plt.show()
```

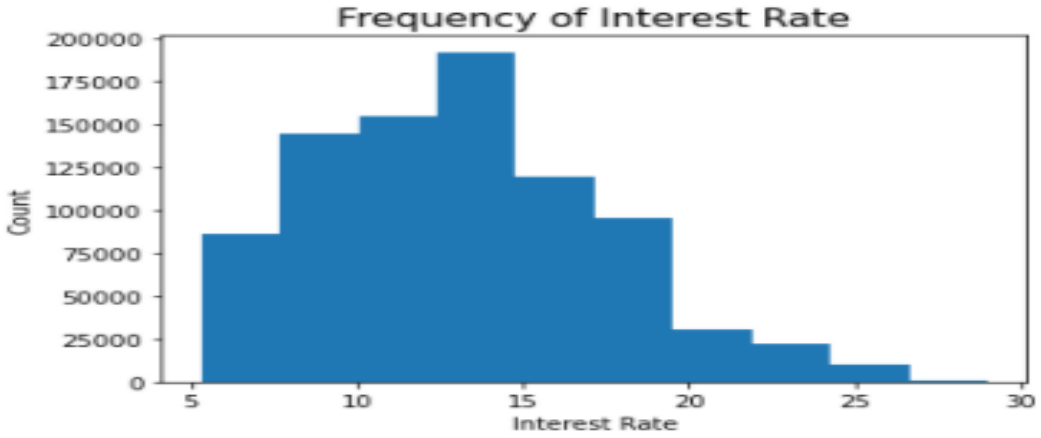

Fig-11 Histogram of variable int_rate

5.Grade

```
sns.countplot(data['grade'])
plt.xlabel('Grades',fontsize=10)
plt.ylabel('Count',fontsize=10)
plt.title('Count of Every Grade',fontsize=15)
plt.show()
```

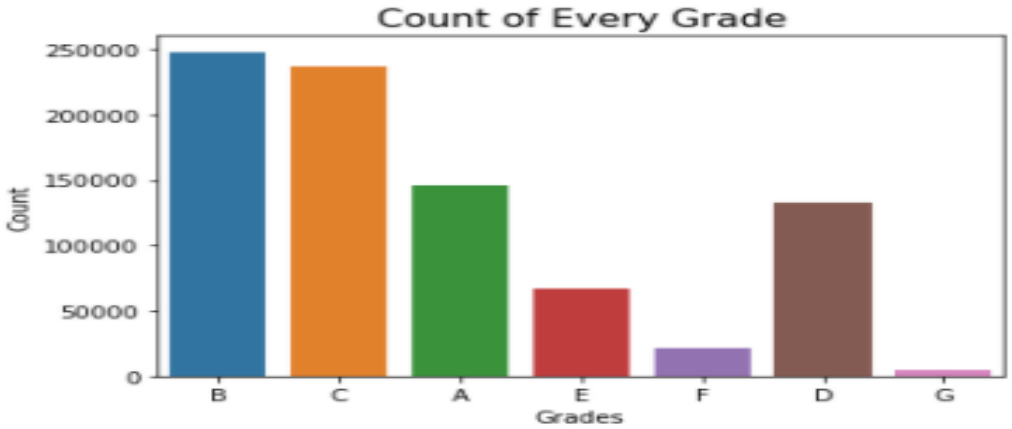

Fig -12 Countplot for variable Grade

6.Employment Length

```
plt.figure(figsize=(10,6))
sns.countplot(data['emp_length'])
plt.xlabel('Employment Length',fontsize=10)
plt.ylabel('Count',fontsize=10)
plt.title('Count of Employment Length',fontsize=15)
plt.show()
```

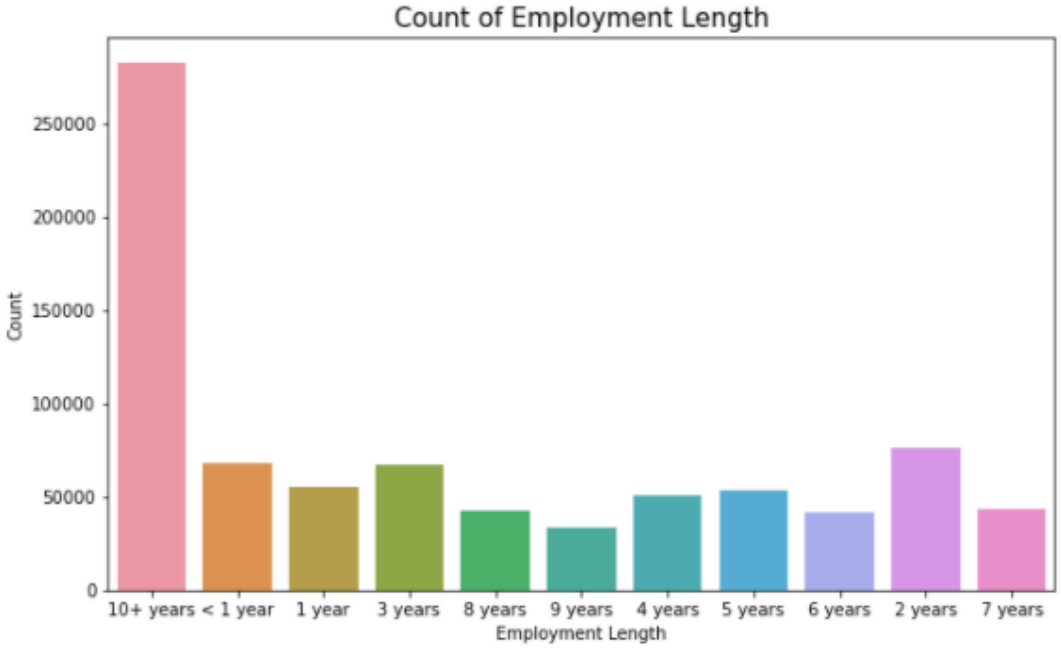

Fig-13 Countplot for variable emp_length

- Most of the records are having more than 10+ years of experience.
- Least records belong to the 9 years of experience.

7.Home Ownership –

```
order = ['MORTGAGE','RENT','OWN','OTHER','NONE','ANY']
sns.countplot(data['home_ownership'],order=order)
plt.xlabel('Home Ownership',fontsize=10)
plt.ylabel('Count',fontsize=10)
plt.title('Count of every category of Home Ownership',fontsize=15)
plt.show()
```

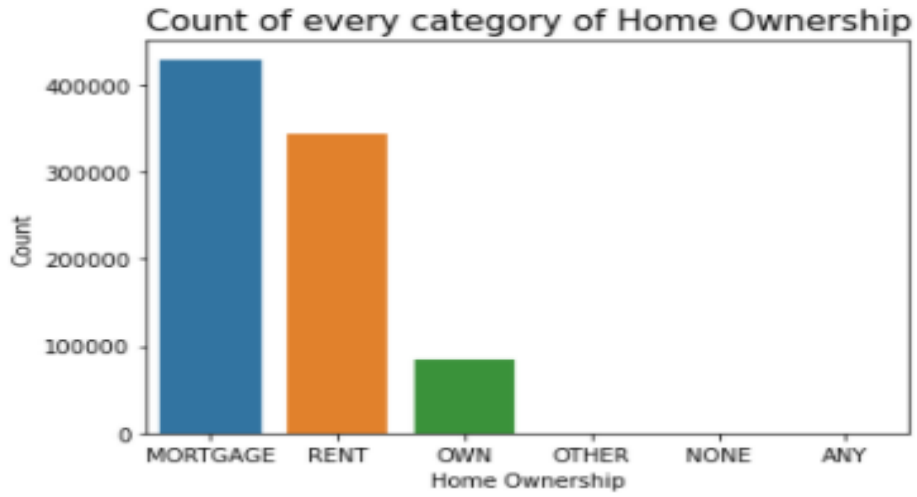

Fig-14 Countplot for variable home_ownership

- Most of the records are having Mortgage type home ownership.
- There are very few records which are having Other, None and Any type of home ownership.

8.Annual Income –

```
plt.hist(data['annual_inc'])
plt.xlabel('Annual Income',fontsize=10)
plt.ylabel('Count',fontsize=10)
plt.title('Frequency of Annual Income',fontsize=15)
plt.show()
```

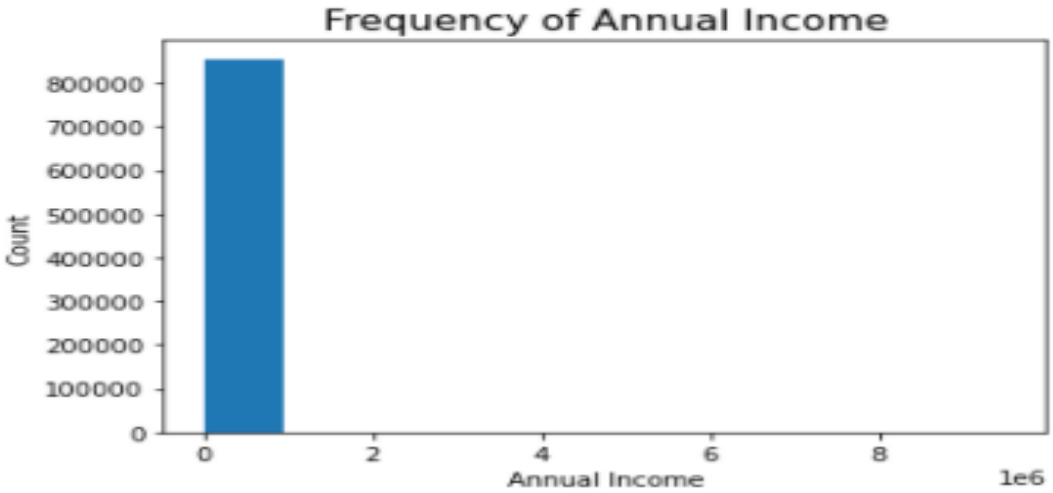

Fig-15 histogram for variable annual_inc

- Almost all the records are having annual income of less than 100000.
- There are few records which are having more than 100000.

9.Loan amount vs Term

```
plt.figure(figsize=(8,5))
sns.boxplot(x='loan_amnt',y='term',data=data)
plt.title('Loan Amount vs Term')
plt.show()
```

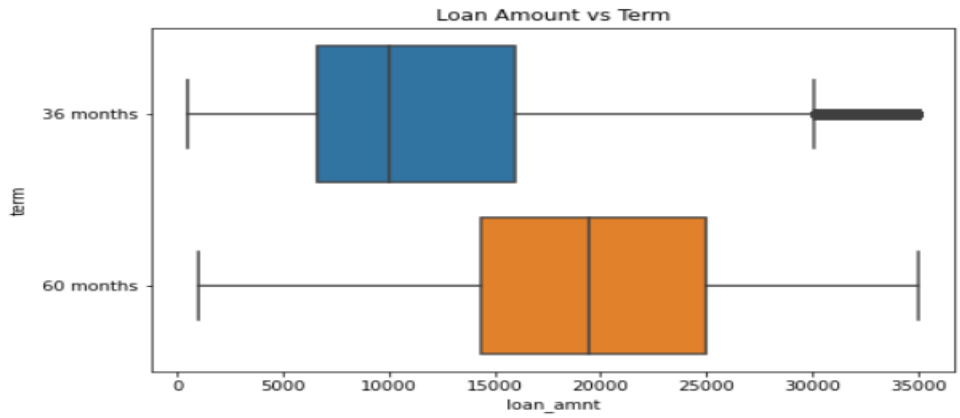

Fig -16 Boxplot between variable loan_amnt and term

- The median is higher for the customers who have taken a loan for 60 months is, they have median around 19000 and while who have taken it for 36 months is having median of around 10000.
- There are some outliers in the loan amount who have opted for 36 months.

10.Interest Rate vs Grade-

```
#We will see that is there any relation of grade and interest rate
sns.catplot(x='grade',y='int_rate',data=data)
plt.title('Grade vs Interest Rate')
plt.show()
```

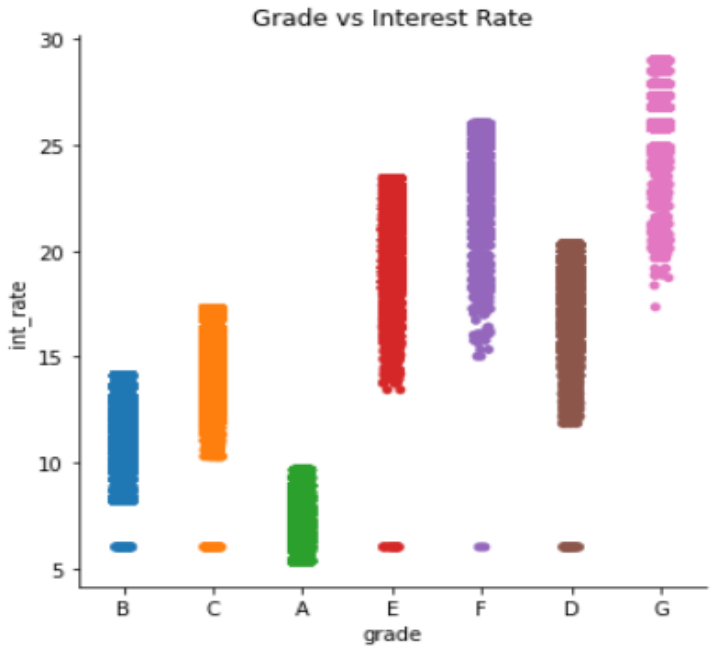

Fig-17 Catplot between variable grade and int_rate

- As the grade increases, the interest rate is also increasing.
- For the records which belongs to the grade G, the interest rate vary from 18 to 30%.
- For the grade A the interest rate vary from 5% to 10%.

11.Defaulter vs Annual income

```
#We will now check is there any relation betweem annualincome and loan defaulter
sns.catplot(x='default_ind',y='annual_inc',data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0x1b4abc16fa0>
```

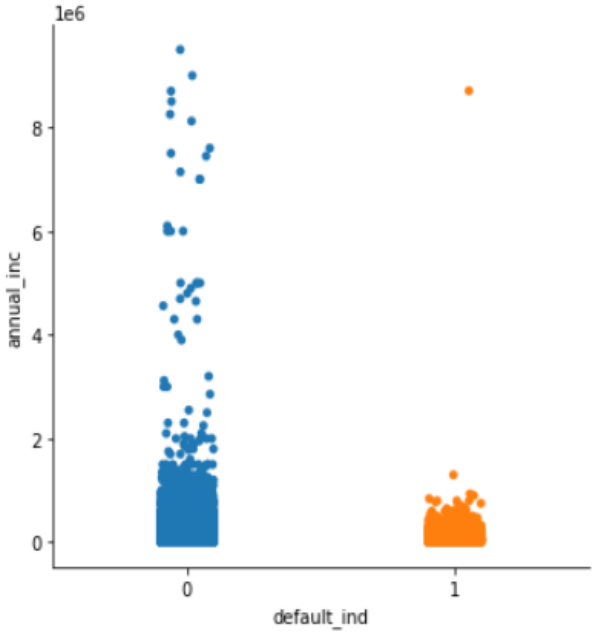

Fig-18 Catplot between variable default_ind and annual_inc

- There are very few records which are having income more than 2000000.
- There is only one record which is defaulter and is having income more than 2000000.
- All the defaulters have income less than 1000000.

Chapter 3 – Machine Learning

Logistic Regression – It is one of the most foremost and basic algorithms used for solving classification problems. It is a statistical machine learning algorithm that classifies the data and creates a logarithmic line between the different classes. Logistic regression uses logit function known as sigmoid function. The equation for the sigmoid function is –

$$f(x) = \frac{1}{1 + e^{-x}}$$

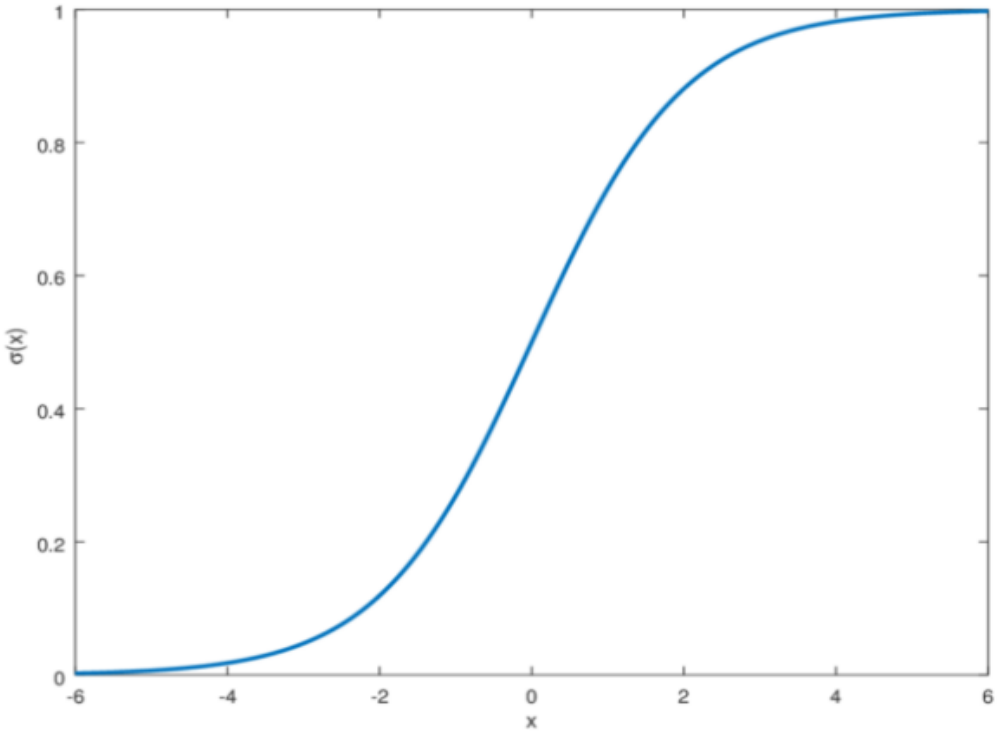

Fig 19– The diagram of sigmoid function

In our problem there is, we are having the binary classification case (the output variable is having two type of categories) where output is having 1 and 0 category, where 0 is for the person who are non-defaulters, and 1 is for the defaulter.

We have trained the different model using different parameters and got different accuracy, but now we have selected the best model which is showing the best output.

```
[[256678      0]
 [    64    247]]
-------------------------------------------------
0.9997509621034363
-------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    256678
           1       1.00      0.79      0.89       311

    accuracy                           1.00    256989
   macro avg       1.00      0.90      0.94    256989
weighted avg       1.00      1.00      1.00    256989
```

Fig 20– Output for the logistic regression

3.2 Decision Tree – It is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility. Decision tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

Decision Tree algorithm divides a dataset into smaller groups of data based on certain conditions using different techniques like 'gini','entropy' etc., until they get the decision node which tells the label. The DT algorithm uses a tree like structure to make a decision because of which its name is decision tree.

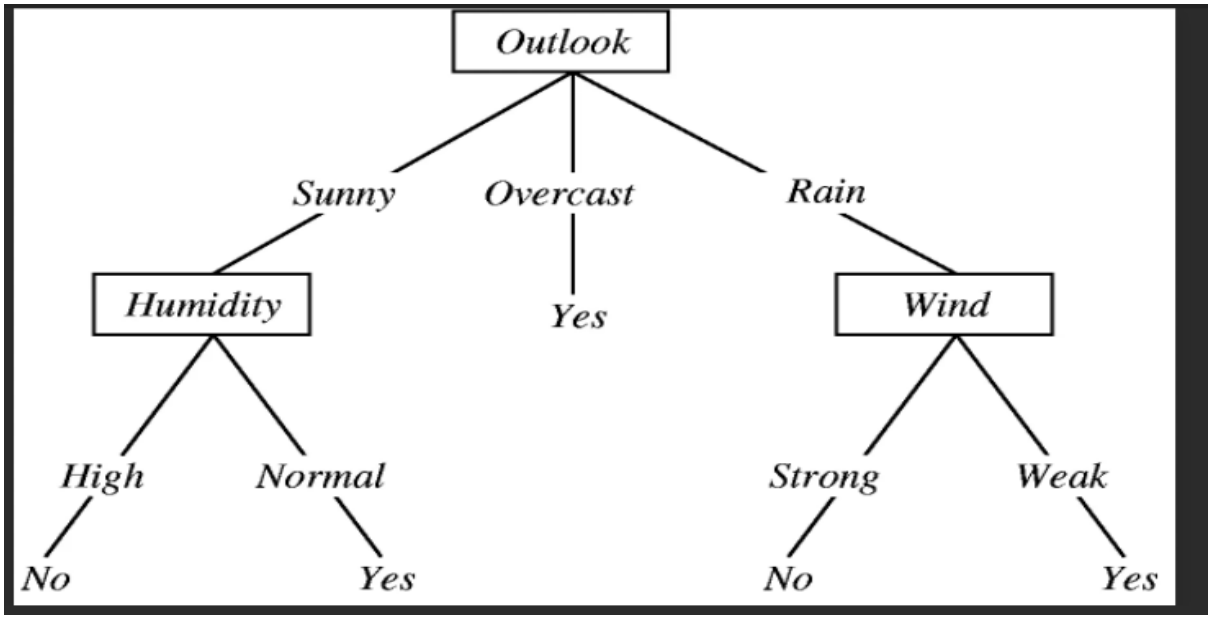

Fig -21 Sample of Decision Tree Diagram

```
[[220337  36341]
 [    19    292]]
------------------------------------------------
0.8585153450147671
------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      0.86      0.92    256678
           1       0.01      0.94      0.02       311

    accuracy                           0.86    256989
   macro avg       0.50      0.90      0.47    256989
weighted avg       1.00      0.86      0.92    256989
```

Fig 22– Accuracy, Precision, Recall and F-1 Score of decision tree before tuning

As we can see in the figure above there were total 256678 records which belong to the non-defaulter and 311 cases belong to defaulters. Out of 256678 records our model was able to predict 220337 cases correctly but it has done a false prediction for 36341 cases. For defaulter case model gave false prediction for 19 records and correct prediction for 292 records.

Since the result was not satisfied we have used the Kfold and RandomSeachCV to tune our model and tried to improve the result.

```
[[256535    143]
 [    64    247]]
------------------------------------------------
0.9991945180533018
------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    256678
           1       0.63      0.79      0.70       311

    accuracy                           1.00    256989
   macro avg       0.82      0.90      0.85    256989
weighted avg       1.00      1.00      1.00    256989
```

Fig-23 Accuracy, Precision, Recall and F-1 Score of decision tree after tuning

After cross validation and applying hyper parameter tuning, false prediction for the non-defaulters decreased to 143 from 36341 but the false prediction for the defaulter increased to 64.

3.3 Random Forest - A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to

combine multiple decision trees in determining the final output rather relying on individual decision tree.

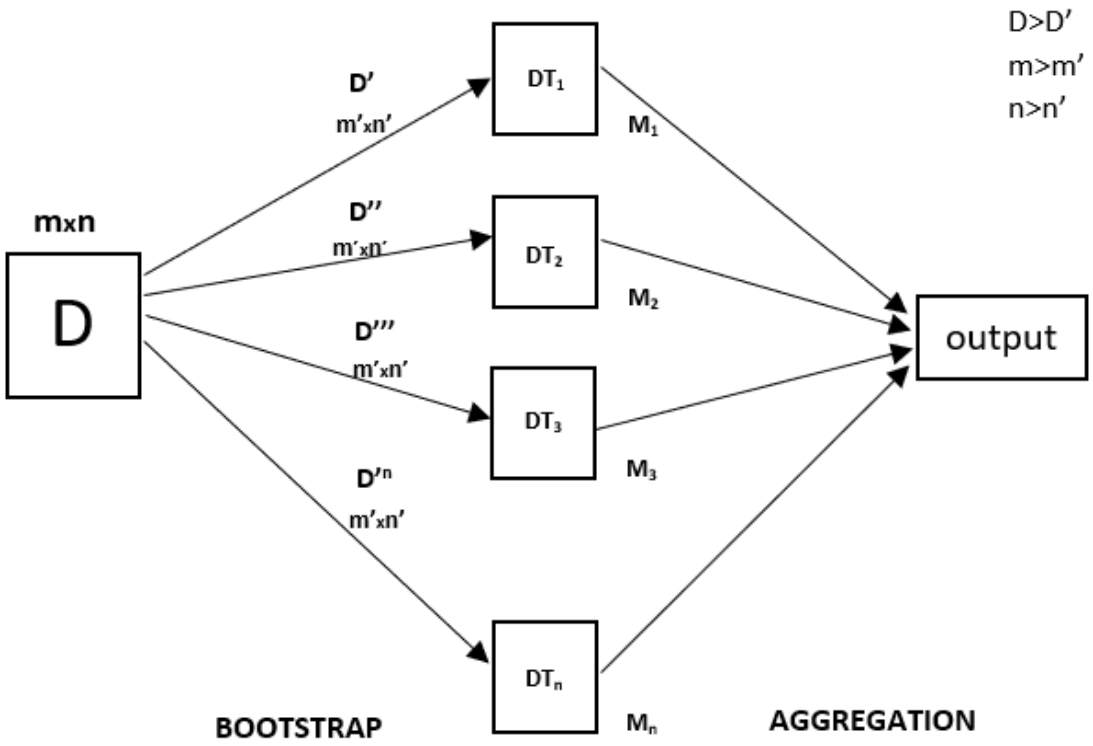

Fig-24 Diagram showing working of Random Forest

```
[[231376  25302]
 [     8    303]]
-------------------------------------------------
0.9015132943433376
-------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      0.90      0.95    256678
           1       0.01      0.97      0.02       311

    accuracy                           0.90    256989
   macro avg       0.51      0.94      0.49    256989
weighted avg       1.00      0.90      0.95    256989
```

Fig-25 Accuracy, Precision, Recall, F-1 score of Random Forest before tuning

From the above figure we can see that the model has performed very well when it comes to prediction of defaulters, it has false prediction of just 8 records out of 311. But it predicted 25302 false prediction for non-defaulters out of 256678. To improve the performance we have tuned our model.

```
[[248451   8227]
 [    58    253]]
-------------------------------------------------
0.9677612660464067
-------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      0.97      0.98    256678
           1       0.03      0.81      0.06       311

    accuracy                           0.97    256989
   macro avg       0.51      0.89      0.52    256989
weighted avg       1.00      0.97      0.98    256989
```

Fig-26 Accuracy, Precision, Recall, F-1 score of Random Forest after tuning

After tuning the model there was not much improvement in the model, false prediction for defaulters increased to 58.

3.4 KNN -This algorithm is used to solve the classification model problems. K-nearest neighbor or K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line. Therefore, larger k value means smother curves of separation resulting in less complex models. Whereas, smaller k value tends to over fit the data and resulting in complex models.

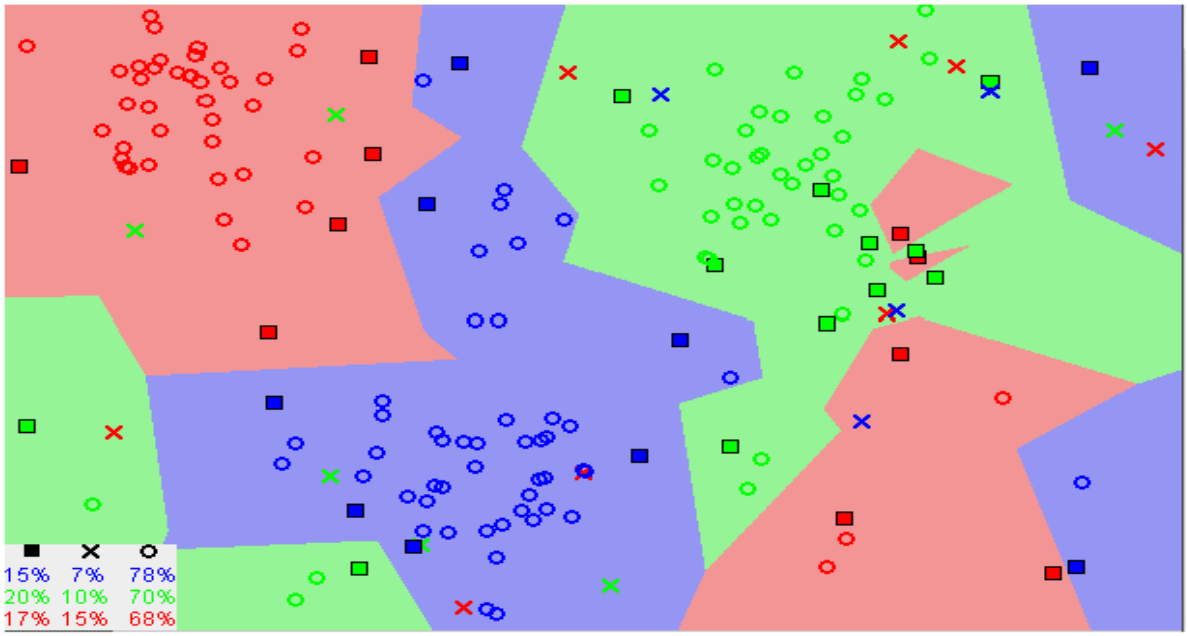

Fig-27 KNN showcase

```
[[256594     84]
 [   141    170]]
-------------------------------------------------
0.9991244761448934
-------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    256678
           1       0.67      0.55      0.60       311

    accuracy                           1.00    256989
   macro avg       0.83      0.77      0.80    256989
weighted avg       1.00      1.00      1.00    256989
```

Fig-28 Accuracy, Precision, Recall, F-1 score of KNN model

In the fig above there are the performance measurements of the machine learning model built using K-Nearest neighbor algorithm. In this model there was a good prediction of non – defaulters i.e., 84 false prediction out of 256678, but the model performed very poor when we classified defaulters. There was around half wrong classification. When we did cross validation and hyper parameter to improve the accuracy, precision, recall there was not much improvement in the model.

3.5 SVM – Support vector machine is a supervised algorithm used to solve classification and regression problems. When we are solving classification problem we use svc library from sklearn and we call it as Support Vector Classifier. A support-vector classifier set a hyperplane in a high- or infinite-dimensional space, which can be used for classification and regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the, generalization error of the classifier.

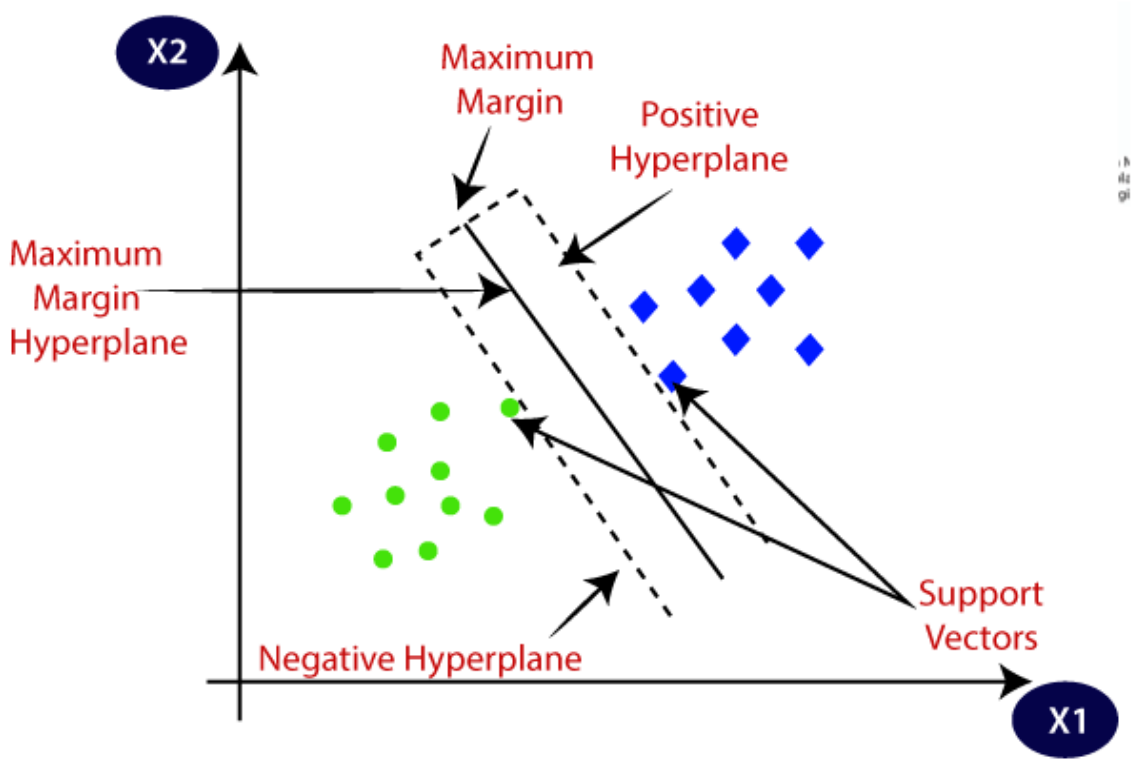

Fig-29 Working of SVC

```
[[256669     11]
 [    63    248]]
-------------------------------------------------
0.9997120521730333
-------------------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    256680
           1       0.96      0.80      0.87       311

    accuracy                           1.00    256991
   macro avg       0.98      0.90      0.94    256991
weighted avg       1.00      1.00      1.00    256991
```

Fig-30 Accuracy, Precision, Recall, F-1 score of SVM model

From the following figure we can conclude that our model was able to perform a better accuracy for non-defaulters, there was a good accuracy for the defaulters case as well. It was able to predict 248 true prediction out of 311 for defaulters, but for 256680 non-defaulters there were just 11 false prediction. The accuracy of our model is 99.9%.

3.6 XG Boost – XG Boost is a decision-tree based ensemble machine learning algorithm that uses a gradient boosting framework. When it comes to small-to-medium structure data decision-tree based algorithms are considered best in class.

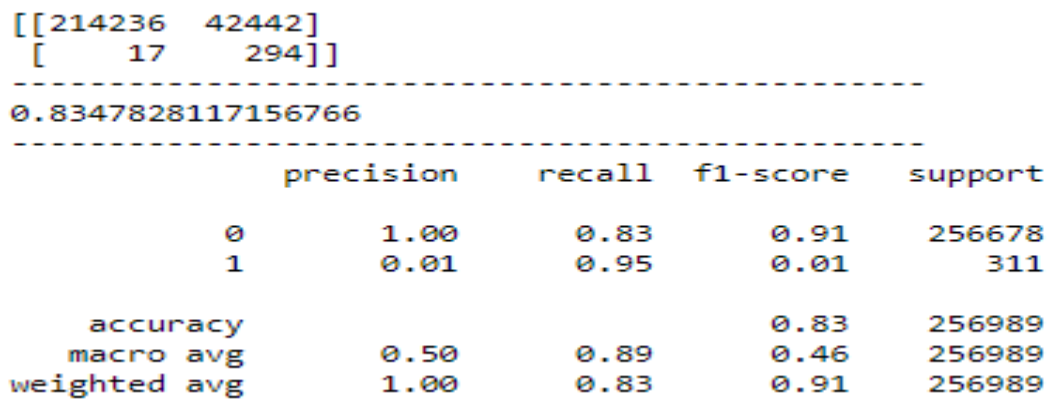

Fig-31 Accuracy, Precision, Recall, F-1 score of XG Boost model

In the given figure we can see the accuracy comparison for both the classes, the false prediction for the defaulter will be 17 out of 311 and false prediction for the non-defaulter is 42442. To improve the accuracy of the model we have used KFold cross validation and hyper parameter but there was not much improvement in the model.

CHAPTER 4: KEY FINDING

1. The customer will be a defaulter or not will not depend on just one variable.
2. The relationship of the dependent variables is very strong with int_rate, out_prncp_inv,recevories, collection_recovery_fee, last_pymnt_d_year.
3. The imbalance in the target category of loan repayment in the dataset, was due to the fact that 82 out of 100 loans were repaid. This indicates money could be lent continuously (always predicting that the borrower would repay) and be correct about 82.07% of the time that the loan was repaid.
4. The cross-validation scores and ROC curves suggest the Logistic Regression is the best model
5. If we look at the confusion matrix, though, we see a big problem. The model can predict who are going to pay off the loan with a good accuracy of 99% but cannot predict who are going to default. The true positive rate of default (0 predicting 0) is almost 0. Since our main goal is to predict defaulter's, we have to do something about this. The reason this is happening could be because of high imbalance in our dataset and the algorithm is putting everything into 1.
6. We cannot perform grid search as the running time is very high due to the size of the data we are having.

Chapter 5: Recommendation and Conclusion

We have successfully built a machine learning algorithm to predict the people who might default on their loans.
Also, we might want to look on other techniques or variables to improve the prediction power of the algorithm. The main disadvantage we have is that the records whose term period has not been completed till the data has been assumed that they will not be a defaulter i.e., they will fall under class 0. This might affect the prediction of the class as their actual value can be something else.

Algorithm_Use	Overall Accuracy	Precision Value	Recall Value	F1-Score	Type I Error	Type II Error	Total Error
Logistic Regression	0.9998	1	0.7942	0.8853	0	64	64
Decision Tree	0.8585	0.008	0.9389	0.0158	36341	19	36360
Tuned Decision Tree	0.9992	0.6333	0.7942	0.7047	143	64	207
Random Forest	0.9015	0.0118	0.9743	0.0234	25302	8	25310
Tuned Random Forest	0.9678	0.0298	0.8135	0.0576	8227	58	8285
KNN	0.9991	0.6693	0.5466	0.6018	84	141	225
SVC	0.9994	0.7352	0.7588	0.7468	85	75	160
XG Boost	0.8348	0.0069	0.9453	0.0137	42442	17	42459

Fig- 32 Comparison table

In this comparison table we have mentioned all the algorithms we have used during this project and their results including accuracy, precision, recall, errors.

From this we can conclude that, the Logistic Regression model is giving the best performance among these 7 models. As the accuracy, precision value is highest and total error is least in the model in comparison to the other ML models we have built. In this we have tried changing the threshold value but there was not much affect in the performance of the model so we are using the same threshold value which is provided for default.

Business Insights and Recommendations:

Among the customers who are having high salary there was only one record who was not able to pay back the loan and was in the category of the defaulters. So the customers who are having less income have slight high chance of being in the category of the defaulter.

All the people from the states ME, ND are in the category of the non-defaulters, the maximum people who are in the category are from CA, FL states, so organization need to take care of the customers who belong to these two particular states.

The customers who have applied for the joint application type is having 100% success rate in returning the loan. So the organization can trust the applicant who have applied for joint application type.

The records who fall under the category of the non-defaulter are having less mean interest i.e., around 12%-13% while those who fall under default category are having mean interest of 18%-19%. So, the customers who are having high rate of interest they need to be taken care by the organization.

The records who fall under the category of the non-defaulter are having term period of 36 months while those who fall under default category are having term period of 60 months. So, the customers who have opted for 60 month term period they need to be taken care by the organization and that can reduce the number of defaulters.

CHAPTER 6: REFERENCES

XG Boost Algorithm: Long May She Reign! | by Vishal Morde | Towards Data Science

Support Vector Machine (SVM) Algorithm - Javatpoint

Support-vector machine - Wikipedia

Machine Learning Basics with the K-Nearest Neighbors Algorithm | by Onel Harrison | Towards Data Science

K-Fold Cross Validation. Evaluating a Machine Learning model can… | by Krishni | Data Driven Investor | Medium