

# Imarticus Learning



**Project on**

---

## **CREDIT RISK ANALYSIS**

---

**Data Science Pro Degree Batch - 27**



**Submitted By:**

1. Nidhi Patel
2. Sanyukta Shenoy
3. Wasim Akram

## ACKNOWLEDGEMENT

Apart from our efforts, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show our greatest appreciation to our project guide **Ms. Nikita Tandel**. We can't thank her enough for the tremendous support and help. We feel motivated and encouraged every time we attend her Lectures. Without her encouragement and guidance, this project would not have materialised.

We express our deep thanks to the high authority of Data Science Pro Degree **Mr. Arun Upadhyay** and other faculty member for extending their support.

## INDEX

SR.NO	TOPIC	PAGE NUMBER
1.	ABSTRACT	3
2.	INTRODUCTION	4
3.	PROPOSED ANALYSIS AND OBJECTIVE	6
4.	PURPOSE OF THE STUDY	7
5.	ORGANISATION DESCRIPTION	8
6.	TECHNOLOGY USED FOR PREDICTION	10
7.	VISUALISATION	11
8.	MODEL DESCRIPTION	17
9.	PREPROCESSING OF DATA	18
10.	MODEL BUILDING	23
11.	COMPARISION OF MODELS	28
12.	CONCLUSION	29

## ABSTRACT

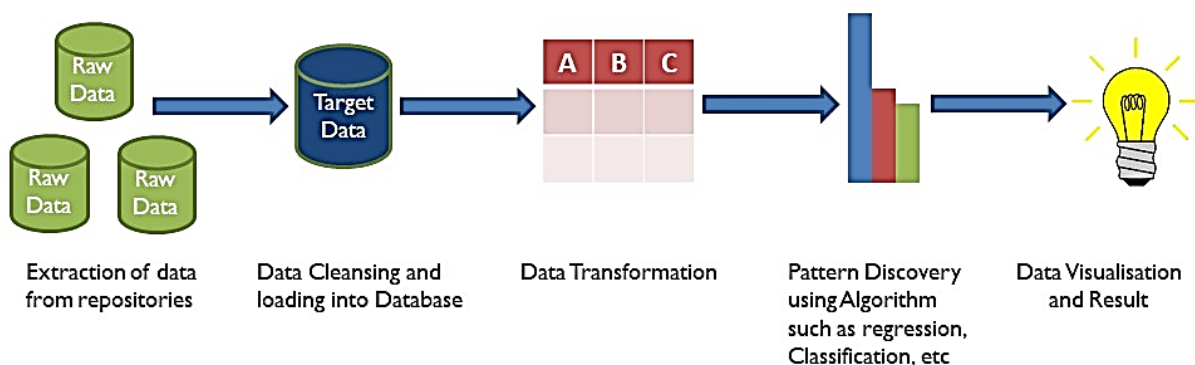
Bank databases is a very interesting field of research, which mainly focus on analysis and understand Credit Risk for bank.

All business including the business of banking requires top line growth in terms of volumes of business to increase the bottom line of profit growth. Credit risk is the possibility of a loss resulting from a borrower's failure to repay a loan or meet contractual obligations. Credit risk also describes the risk that a bond issuer may fail to make payment when requested or that an insurance company will be unable to pay a claim. Hence, it is very important for any Businesses to quickly analyze data used to assess a customer's risk profile.

So, the goal of credit risk management in banks is to maintain credit risk exposure within proper and acceptable parameters to classify each borrower as defaulter or not, based on the loan issued by XYZ Co-operation through 2007-2015.

## INTRODUCTION

In Data Science the amount of data being generated and stored is growing very rapidly, due in large part to the continuing advances in computer technology. This presents tremendous opportunities for those who can unlock the information embedded within this data, but also introduces new challenges. Data Science can be used to extract useful knowledge from the data that surround us. Those that can master this technology and its methods can derive great benefits and gain a competitive advantage.



### Proposed Architecture for Credit Risk Analysis

Credit Risk is new trend in the data science and knowledge Discoveries in Databases field which focuses in mining and discovering the useful information such as our example consider for the Financial Domain to predict whether the borrower will repay loan or not.

The challenges faced by the bank today includes:

- Managing quality of loan assets.
- Boosting the credit flow to all productive sectors of the economy.
- To create an interest rate environment that supports revival of investment demand at the same time ensuring consistent growth in bank's profitability.

The biggest and most difficult to manage for the Bank is Credit Risk, Credit risk can be described as the possibility of losses being incurred on account of deterioration in the quality of the borrowers in a portfolio. The basic function of commercial banks are accepting deposits and lending. So, to overcome this we can use the following details which is collected while opening an account such as:

1. Employment details such as job specifications, name and address of the employer, length of service, etc.
2. Provide details about source of income and annual income.
3. Details of assets owned such as house, vehicle, etc.
4. Other personal details such as qualification, marital status, etc.

Any Business including the business of banking requires top line growth in terms of volumes of business to increase the bottom line of profit growth, In this context, quality and value advances are of utmost importance to ensure sustainable business growth. Hence there is an urgent and immediate need for the bank to concentrate on quality of credit with profitability.

People often save their money in the banks which offer security but with lower interest rates. 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. It is transforming the banking system to make credit more affordable and investing more rewarding. But this comes with a high risk of borrowers defaulting the loans. Hence there is a need to classify each borrower as defaulter or not using the data collected from bank named as XYZ Co-operation through the year 2007-2015.

## **PROPOSED ANALYSIS AND OBJECTIVE**

### **1. TITLE**

## **CREDIT RISK ANALYSIS**

### **2. OBJECTIVE**

The objective of Credit Risk Analysis project is to put ourselves in the shoes of a loan issuer and manage credit risk by using the past data and deciding whom to give the loan to in the future. Model has to analyzing the XYZ Co-operation Data to analyze and detect defaulters by building models on data from June 2007 to May 2015 and testing it on data from June 2015 to December 2015. Based on the accuracy of each model we can determine how good a model is for predicting that a person applying for loan will default or not.

## PURPOSE OF THE STUDY

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

0 represents – Not Defaulter

1 represents – Defaulter



## ORGANISATION DESCRIPTION

**Name of the Organisation:** XYZ Co-operation Bank

XYZ Corporation Lending Data is used under the study. Data of Loans issued by XYZ Co-operation through the year 2007-2015 is used for analysis. The data contains the indicator of default, payment information, credit history and many more variables.

### Database Description:

**Rows:** 855696 **Columns:** 73

No	LoanStatNew	Description
1	Id	A unique assigned ID for the loan listing.
2	member_id	A unique Id for the borrower member.
3	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 reflected in this value.
4	funded_amnt	The total amount committed to that loan at that point in time.
5	funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
6	Term	The number of payments on the loan. Values are in months and can be either 36 or 60.
7	int_rate	Interest Rate on the loan
8	Instalment	The monthly payment owed by the borrower if the loan originates.
9	Grade	XYZ corp. assigned loan grade
10	sub_grade	XYZ assigned assigned loan subgrade
11	emp_title	The job title supplied by the Borrower when applying for the loan.
12	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.
13	home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
14	annual_inc	The self-reported annual income provided by the borrower during registration.
15	verification_status	Was the income source verified
16	issue_d	The month which the loan was funded
17	pymnt_plan	Indicates if a payment plan has been put in place for the loan
18	Desc	Loan description provided by the borrower
19	Purpose	A category provided by the borrower for the loan request.
20	Title	The loan title provided by the borrower
21	zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.
22	addr_state	The state provided by the borrower in the loan application
23	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.
24	delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
25	earliest_cr_line	The month the borrower's earliest reported credit line was opened
26	inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
27	mths_since_last_delinq	The number of months since the borrower's last delinquency.
28	mths_since_last_record	The number of months since the last public record.
29	open_acc	The number of open credit lines in the borrower's credit file.
30	pub_rec	Number of derogatory public records
31	revol_bal	Total credit revolving balance

No	LoanStatNew	Description
32	revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
33	total_acc	The total number of credit lines currently in the borrower's credit file
34	initial_list_status	The initial listing status of the loan. Possible values are – W, F
35	out_prncp	Remaining outstanding principal for total amount funded
36	out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
37	total_pymnt	Payments received to date for total amount funded
38	total_pymnt_inv	Payments received to date for portion of total amount funded by investors
39	total_rec_prncp	Principal received to date
40	total_rec_int	Interest received to date
41	total_rec_late_fee	Late fees received to date
42	Recoveries	post charge off gross recovery
43	collection_recovery_fee	post charge off collection fee
44	last_pymnt_d	Last month payment was received
45	last_pymnt_amnt	Last total payment amount received
46	next_pymnt_d	Next scheduled payment date
47	last_credit_pull_d	The most recent month XYZ corp. pulled credit for this loan
48	collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
49	mths_since_last_major_derog	Months since most recent 90-day or worse rating
50	policy_code	publicly available policy_code=1 new products not publicly available policy_code=2
51	application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
52	annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
53	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
54	verified_status_joint	Indicates if the co-borrowers' joint income was verified by XYZ corp., not verified, or if the income source was verified
55	acc_now_delinq	The number of accounts on which the borrower is now delinquent.
56	tot_coll_amt	Total collection amounts ever owed
57	tot_cur_bal	Total current balance of all accounts
58	open_acc_6m	Number of open trades in last 6 months
59	open_il_6m	Number of currently active installment trades
60	open_il_12m	Number of installment accounts opened in past 12 months
61	open_il_24m	Number of installment accounts opened in past 24 months
62	mths_since_rcnt_il	Months since most recent installment accounts opened
63	total_bal_il	Total current balance of all installment accounts
64	il_util	Ratio of total current balance to high credit/credit limit on all install acct
65	open_rv_12m	Number of revolving trades opened in past 12 months
66	open_rv_24m	Number of revolving trades opened in past 24 months
67	max_bal_bc	Maximum current balance owed on all revolving accounts
68	all_util	Balance to credit limit on all trades
69	total_rev_hi_lim	Total revolving high credit/credit limit
70	inq_fi	Number of personal finance inquiries
71	total_cu_tl	Number of finance trades
72	inq_last_12m	Number of credit inquiries in past 12 months
73	Default_ind	Current status of the loan

## TECHNOLOGY USED FOR PREDICTION

**For Coding:**

**Anaconda Navigator Jupyter Notebook:**

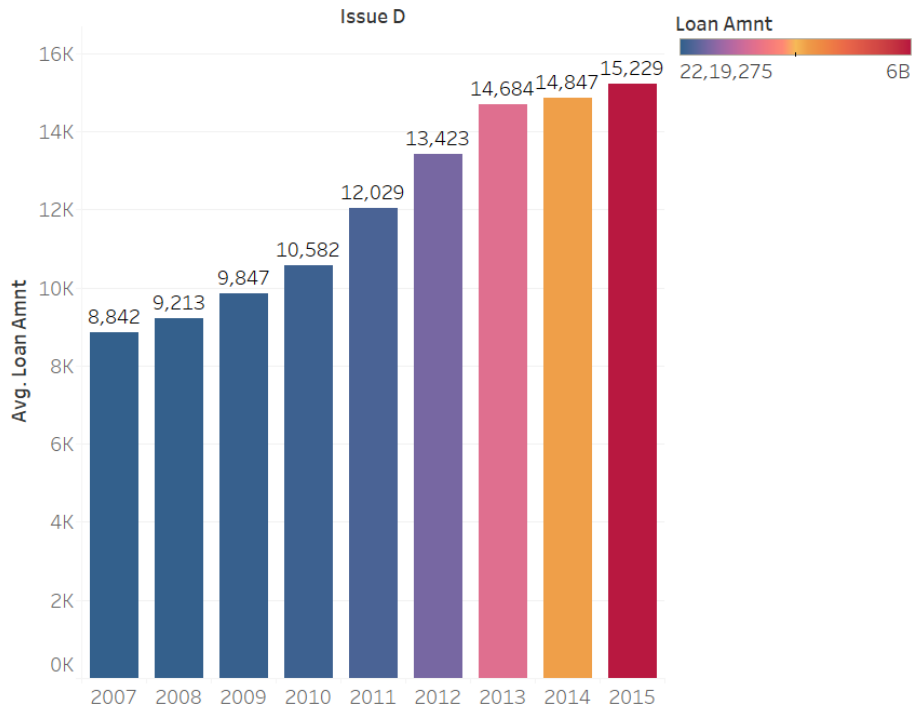


**For Visualisation:**



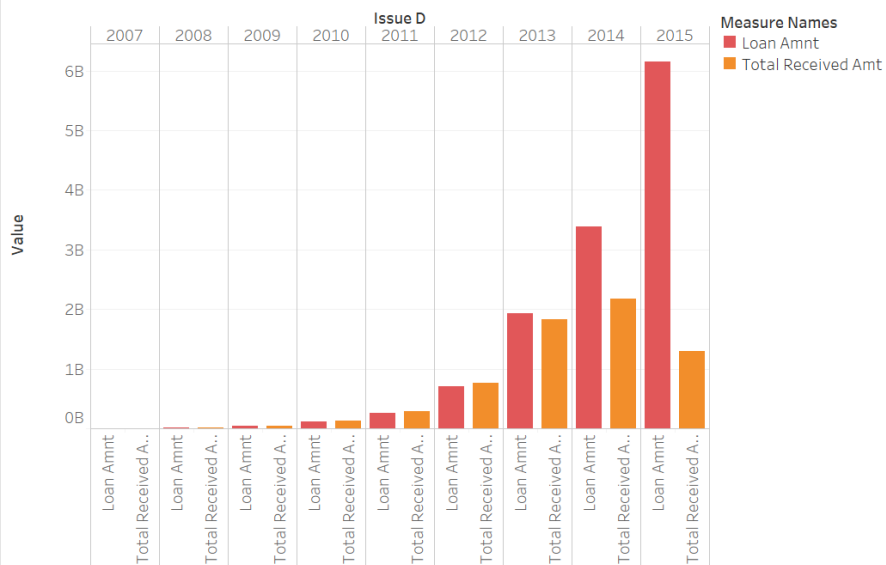
## VISUALISATION

### AVERAGE LOAN AMOUNT GIVEN EVERY YEAR



Average of Loan Amnt for each Issue D Year. Color shows sum of Loan Amnt. The marks are labeled by average of Loan Amnt.

### LOAN AMOUNT GIVEN VS RECEIVED EVERY YEAR



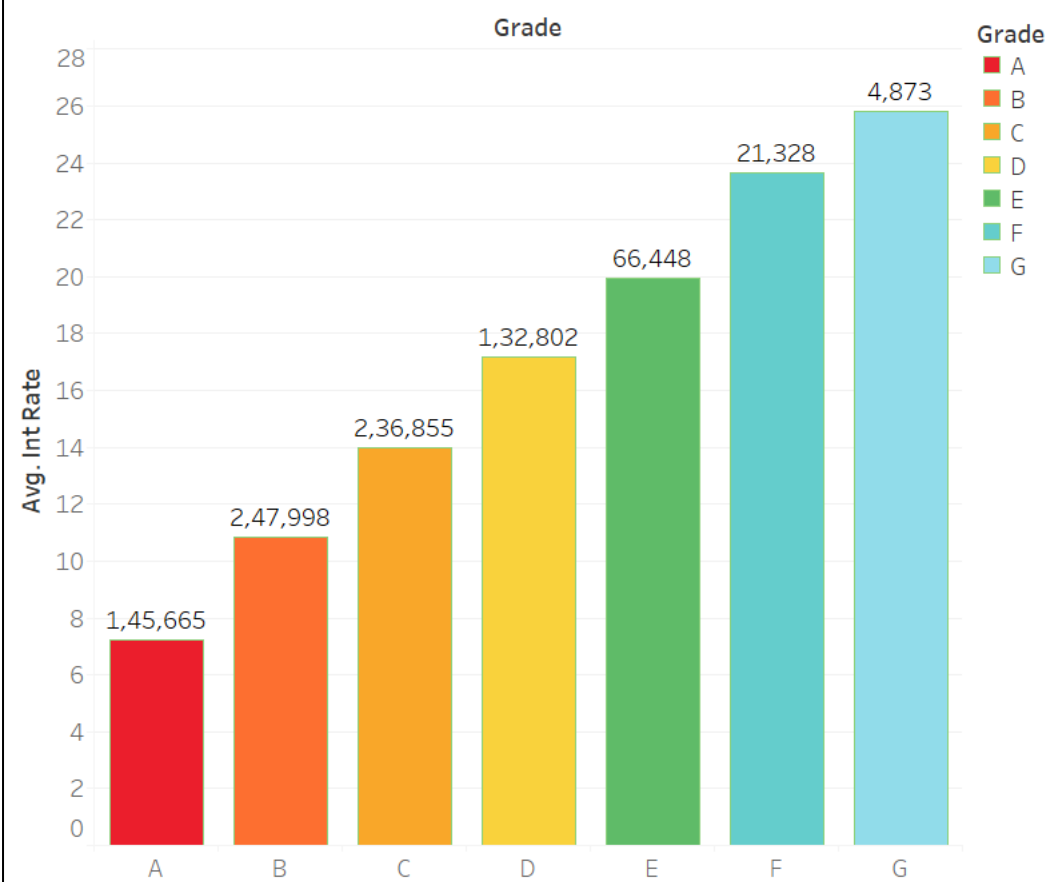
Loan Amnt and Total Received Amt for each Issue D Year. Color shows details about Loan Amnt and Total Received Amt.

## **AVERAGE INTEREST RATE PER PURPOSE**

Purpose	Avg. Int Rate	Loan Amnt
car	12	76,042,850
credit_card	12	3,067,696,575
debt_consolidati..	14	7,790,583,700
educational	12	2,215,600
home_improvem..	13	713,535,300
house	16	52,326,250
major_purchase	13	191,623,025
medical	15	73,836,625
moving	16	40,508,050
other	15	404,905,200
renewable_ener..	15	5,448,900
small_business	16	150,736,000
vacation	14	28,288,325
wedding	14	24,005,550

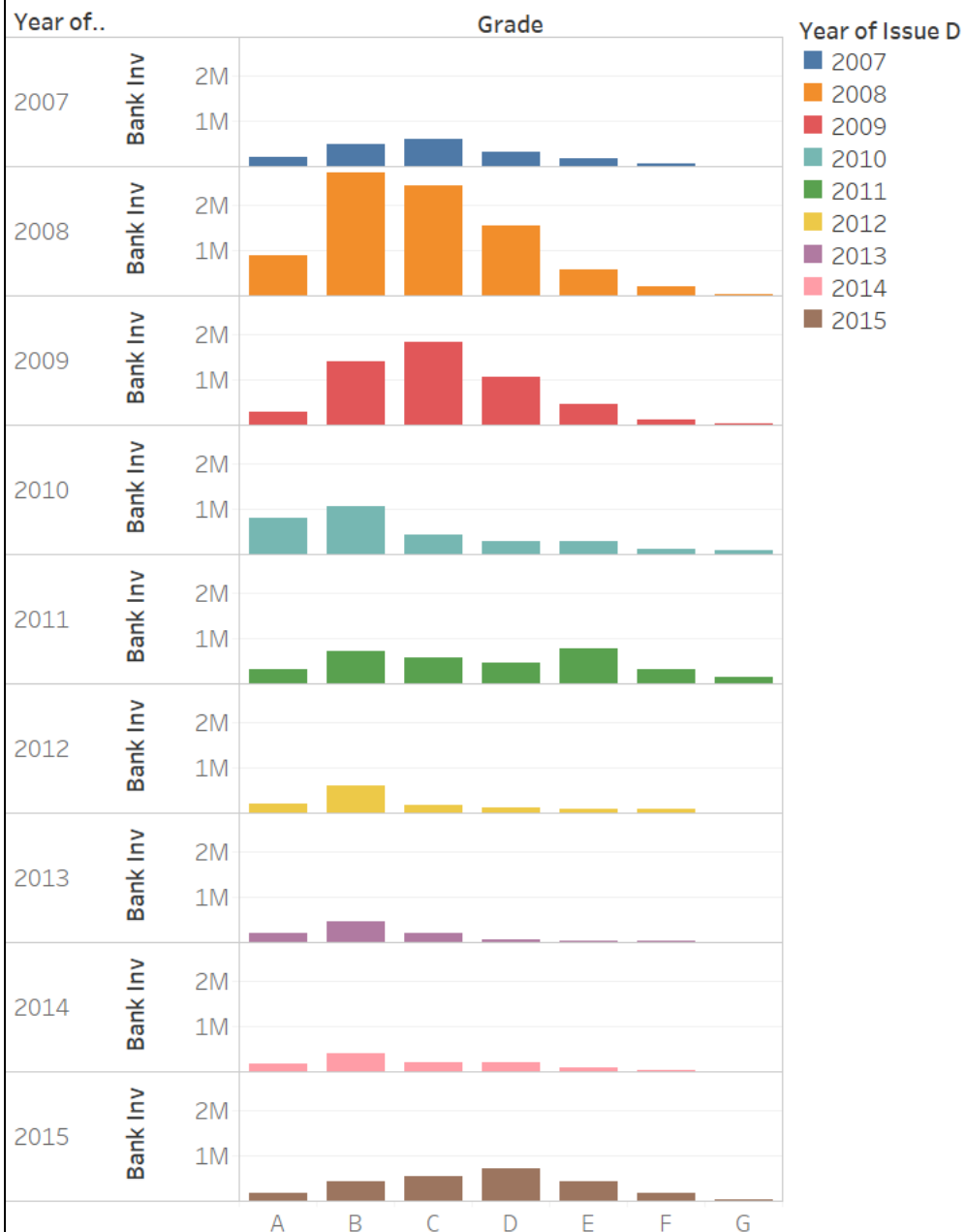
Avg. Int Rate and Loan Amnt broken down by Purpose.

## AVERAGE INTEREST RATE PER GRADE



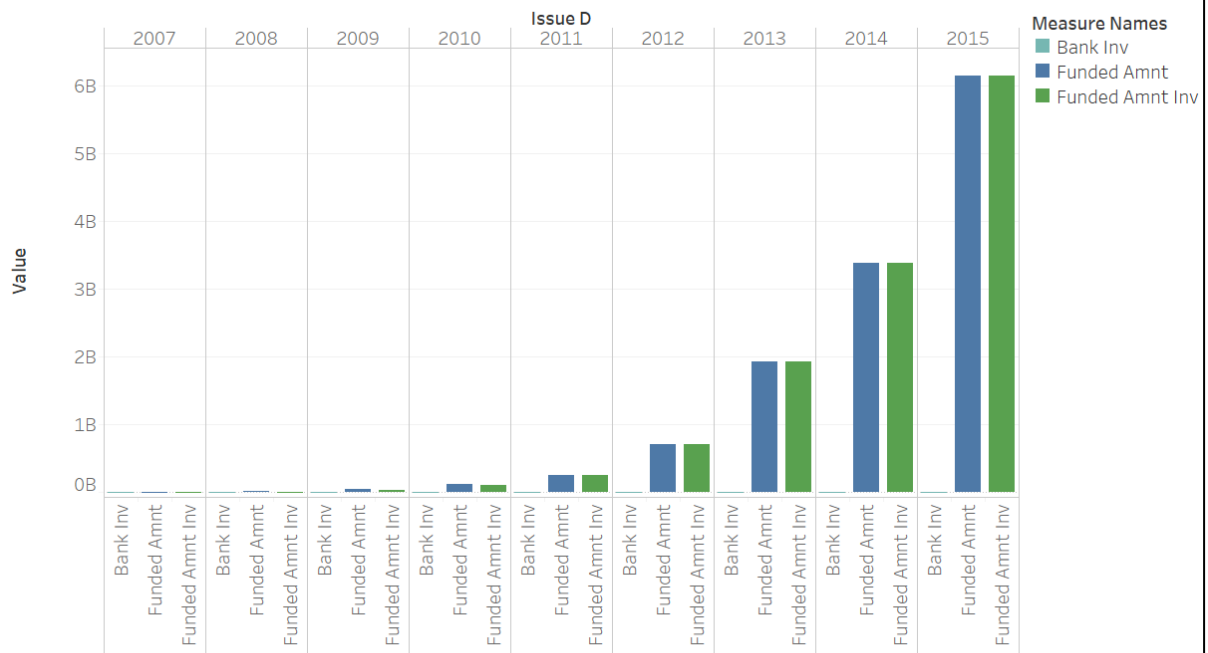
Average of Int Rate for each Grade. Color shows details about Grade. The marks are labeled by count of Grade.

## INVESTMENT OF BANK ON DIFFERENT GRADES



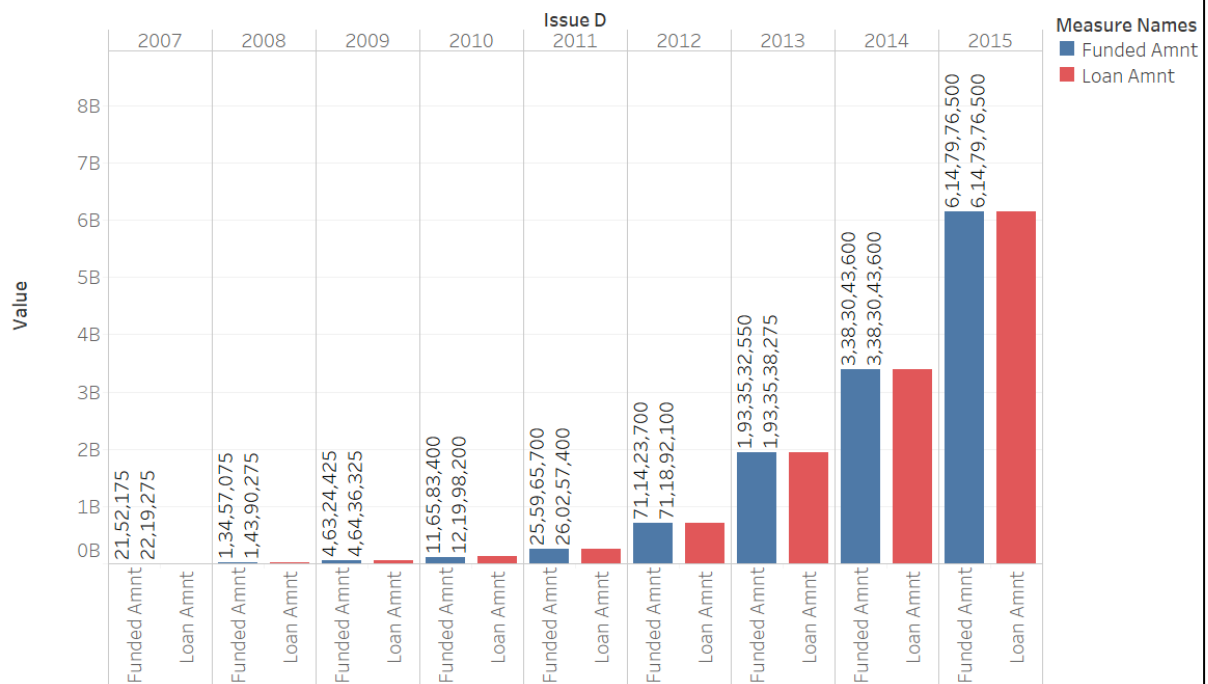
Sum of Bank Inv for each Grade broken down by Issue D Year. Color shows details about Issue D Year.

## INVESTMENT OF BANK VS INVESTOR



Bank Inv, Funded Amnt and Funded Amnt Inv for each Issue D Year. Color shows details about Bank Inv, Funded Amnt and Funded Amnt Inv.

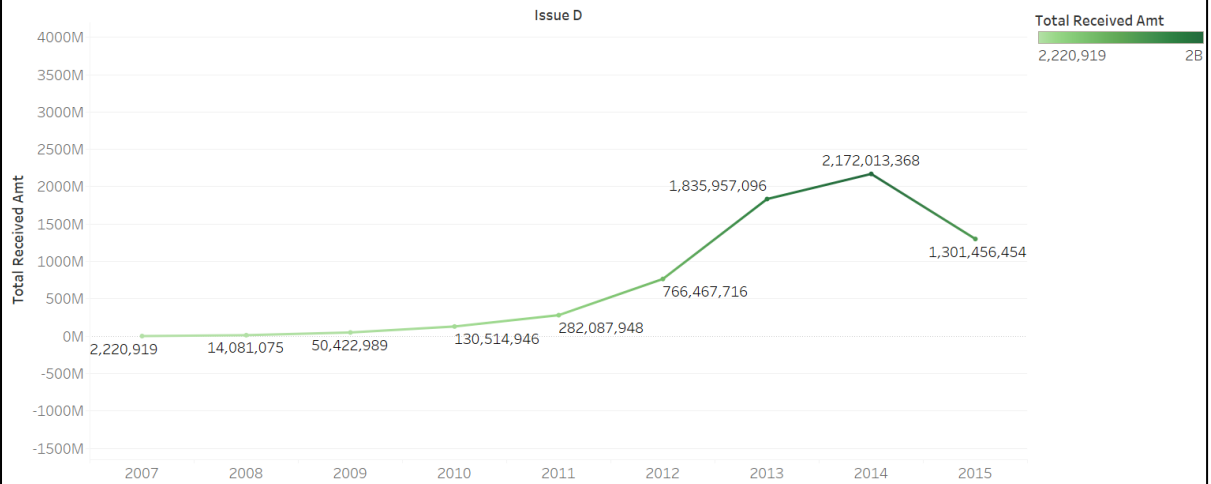
## FUNDED AMOUNT VS LOAN AMOUNT



Funded Amnt and Loan Amnt for each Issue D Year. Color shows details about Funded Amnt and Loan Amnt. The marks are labeled by Funded Amnt and Loan Amnt.

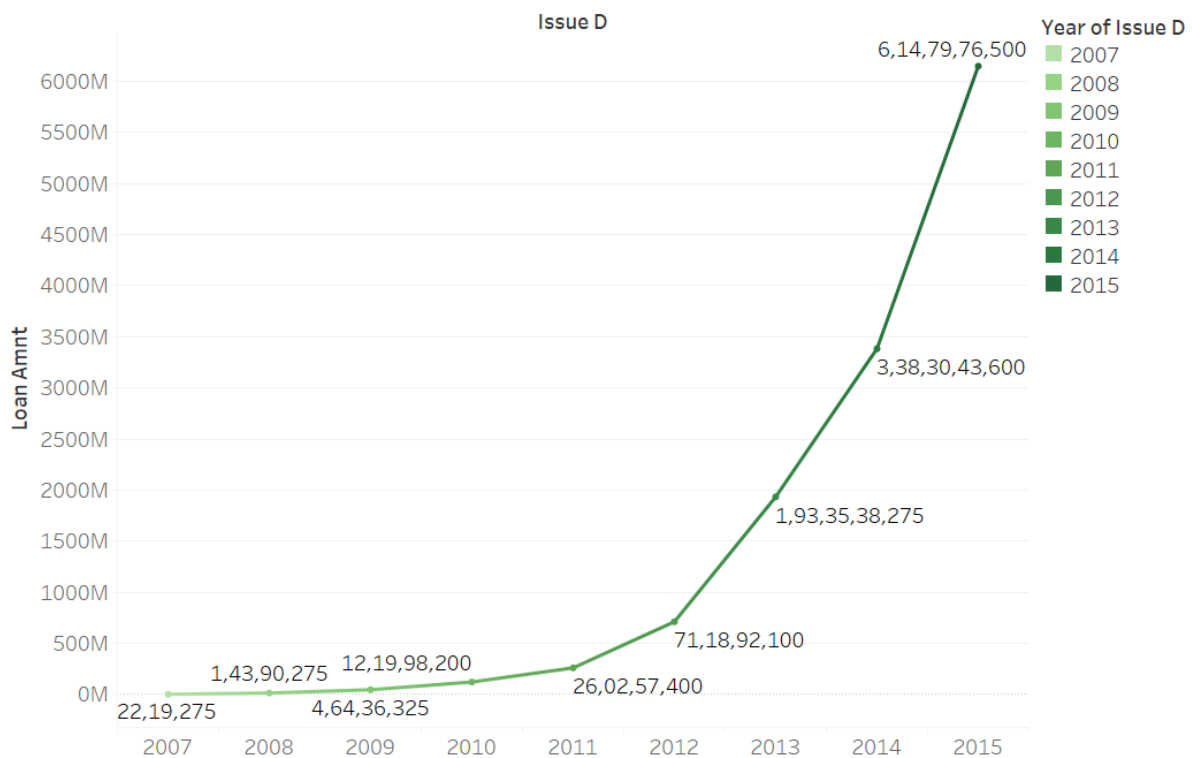


## TOTAL AMOUNT RECOVERED EVERY YEAR



The trend of sum of Total Received Amt for Issue D Year. Color shows sum of Total Received Amt. The marks are labeled by sum of Total Received Amt.

## TOTAL AMOUNT OF LOAN GIVEN EVERY YEAR



The trend of sum of Loan Amnt for Issue D Year. Color shows details about Issue D Year. The marks are labeled by sum of Loan Amnt.

## MODELS DESCRIPTION

### MODEL 1 – LOGISTIC REGRESSION

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

### MODEL 2 – DECISION TREE

*Decision Tree Classifier is a simple and widely used classification technique. It applies a straight forward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receive an answer, a follow-up question is asked until a conclusion about the class label of the record is reached. The decision tree classifiers organized a series of test questions and conditions in a tree structure.*

### MODEL 3 – ADAPTIVE BOOST CLASSIFIER (ADABOOST)

*Boosting is an ensemble technique that attempts to create a strong classifier from a number of weak classifiers. AdaBoost was the first really successful boosting algorithm developed for binary classification. It is the best starting point for understanding boosting.*

### MODEL 4 – EXTREMELY RANDOMIZED TREES (EXTRA TREES CLASSIFIER)

*An “extra trees” classifier, otherwise known as an “Extremely randomized trees” classifier, is a variant of a random forest. Unlike a random forest, at each step the entire sample is used and decision boundaries are picked at random, rather than the best one. In real world cases, performance is comparable to an ordinary random forest, sometimes a bit better.*

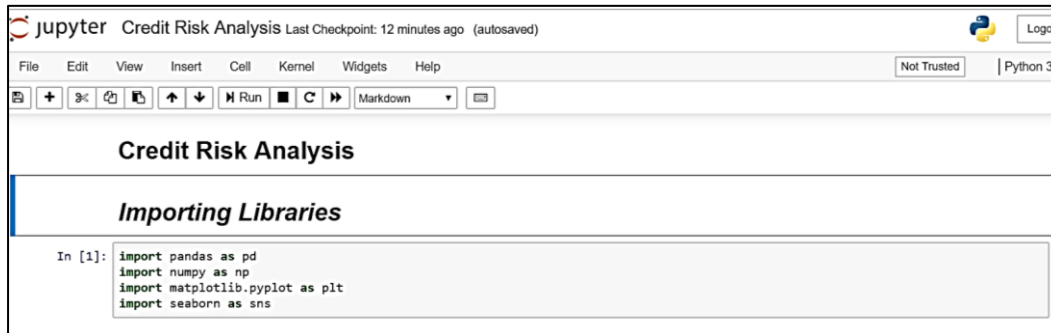
## CROSS VALIDATION TECHNIQUE

### K-FOLD CROSS VALIDATION TECHNIQUE

*Cross-validation is a statistical method used to estimate the skill of machine learning models .It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.*

## DATA EXTRACTION AND CLEANING

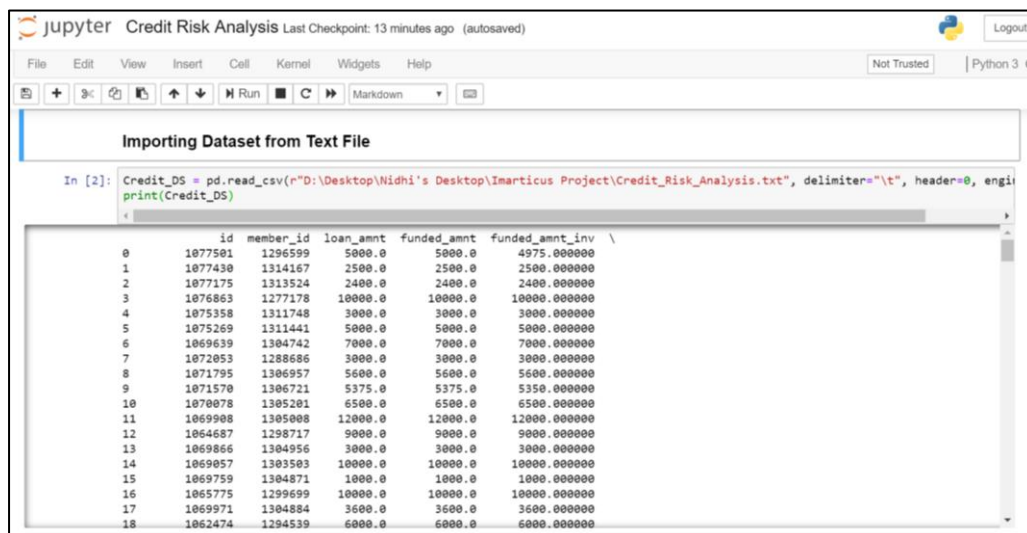
### 1. Importing the Libraries



```

In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
  
```

### 2. Importing the Dataset

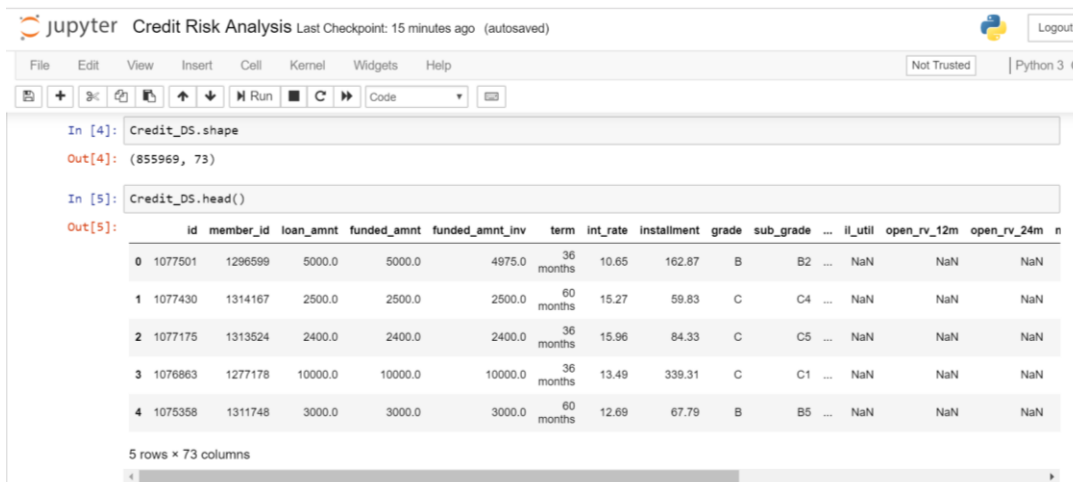


```

In [2]: Credit_DS = pd.read_csv(r"D:\Desktop\Nidhi's Desktop\Imarticus Project\Credit_Risk_Analysis.txt", delimiter="t", header=0, engine='python')
print(Credit_DS)
  
```

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	
0	1077501	1296599	5000.0	5000.0	4975.000000	
1	1077430	1314167	2500.0	2500.0	2500.000000	
2	1077175	1313524	2400.0	2400.0	2400.000000	
3	1076863	1277178	10000.0	10000.0	10000.000000	
4	1075358	1311748	3000.0	3000.0	3000.000000	
5	1075269	1311441	5000.0	5000.0	5000.000000	
6	1069639	1304742	7000.0	7000.0	7000.000000	
7	1072053	1288686	3000.0	3000.0	3000.000000	
8	1071795	1306957	5600.0	5600.0	5600.000000	
9	1071570	1306721	5375.0	5375.0	5350.000000	
10	1070078	1305201	6500.0	6500.0	6500.000000	
11	1069908	1305008	12000.0	12000.0	12000.000000	
12	1064687	1298717	9000.0	9000.0	9000.000000	
13	1069866	1304956	3000.0	3000.0	3000.000000	
14	1069057	1303503	10000.0	10000.0	10000.000000	
15	1069759	1304871	1000.0	1000.0	1000.000000	
16	1065775	1299699	10000.0	10000.0	10000.000000	
17	1069971	1304884	3600.0	3600.0	3600.000000	
18	1062474	1294539	6000.0	6000.0	6000.000000	

### 3. View the Shape and head of the Data



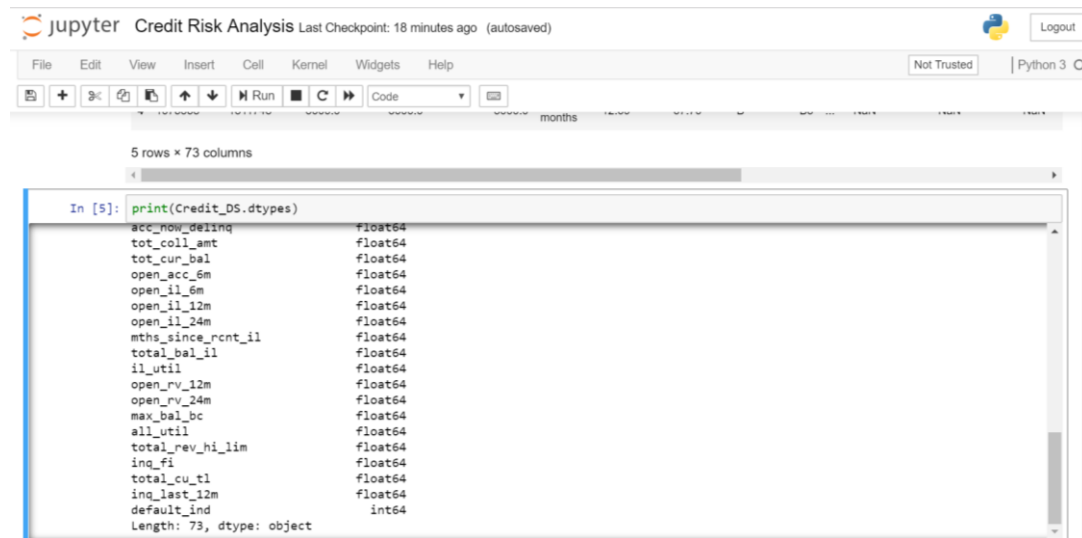
```

In [4]: Credit_DS.shape
Out[4]: (855969, 73)

In [5]: Credit_DS.head()
Out[5]:
   id  member_id  loan_amnt  funded_amnt  funded_amnt_inv  term  int_rate  installment  grade  sub_grade  ...  il_util  open_rv_12m  open_rv_24m
0  1077501    1296599     5000.0      5000.0         4975.0  36 months    10.65         162.87      B      B2      ...    NaN      NaN      NaN
1  1077430    1314167     2500.0      2500.0         2500.0  60 months    15.27         59.83      C      C4      ...    NaN      NaN      NaN
2  1077175    1313524     2400.0      2400.0         2400.0  36 months    15.96         84.33      C      C5      ...    NaN      NaN      NaN
3  1076863    1277178    10000.0     10000.0        10000.0  36 months    13.49        339.31      C      C1      ...    NaN      NaN      NaN
4  1075358    1311748     3000.0      3000.0         3000.0  60 months    12.69         67.79      B      B5      ...    NaN      NaN      NaN
  
```

5 rows x 73 columns

## 4. Datatype of Datasets



Jupyter Credit Risk Analysis Last Checkpoint: 18 minutes ago (autosaved)

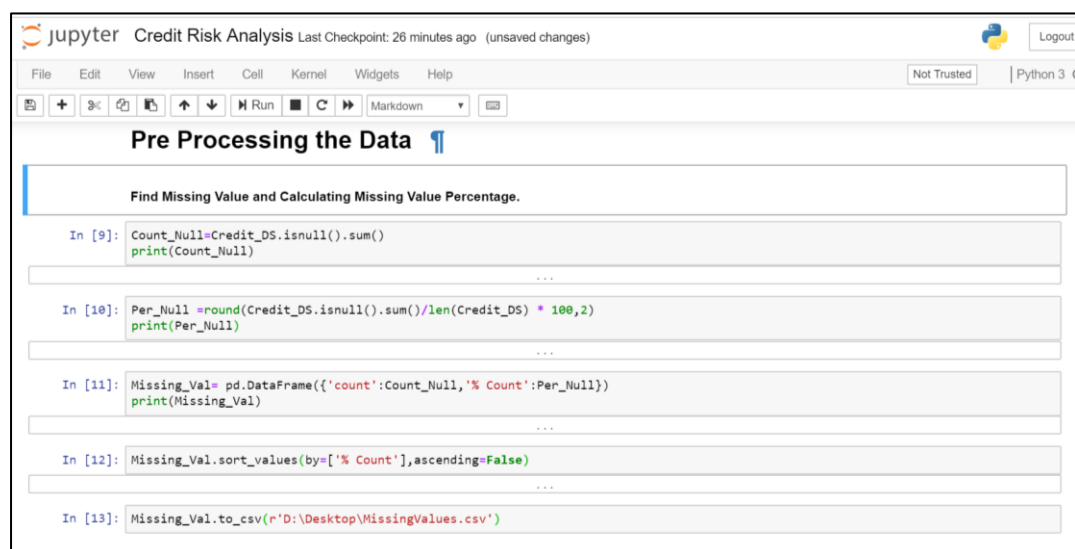
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

5 rows x 73 columns

```
In [5]: print(Credit_DS.dtypes)
acc_now_delinq      float64
tot_coll_amt        float64
tot_cur_bal         float64
open_acc_6m         float64
open_il_6m          float64
open_il_12m         float64
open_il_24m         float64
mths_since_rcnt_il  float64
total_bal_il        float64
il_util             float64
open_rv_12m         float64
open_rv_24m         float64
max_bal_bc          float64
all_util            float64
total_rev_hi_lim     float64
inq_fi              float64
total_cu_tl         float64
inq_last_12m        float64
default_ind         int64
Length: 73, dtype: object
```

## Treating Missing Values

### 5. Checking Missing Value and Deleting all the Column having Missing Value more than 50%



Jupyter Credit Risk Analysis Last Checkpoint: 26 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

### Pre Processing the Data

Find Missing Value and Calculating Missing Value Percentage.

```
In [9]: Count_Null=Credit_DS.isnull().sum()
print(Count_Null)

...

In [10]: Per_Null =round(Credit_DS.isnull().sum()/len(Credit_DS) * 100,2)
print(Per_Null)

...

In [11]: Missing_Val= pd.DataFrame({'count':Count_Null,'% Count':Per_Null})
print(Missing_Val)

...

In [12]: Missing_Val.sort_values(by=['% Count'],ascending=False)

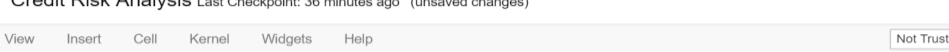
...

In [13]: Missing_Val.to_csv(r'D:\Desktop\MissingValues.csv')
```

### Output:

Column Name	count	% Count
annual_inc_joint	855527	99.95
dti_joint	855529	99.95
verification_status_joint	855527	99.95
il_util	844360	98.64
mths_since_rcnt_il	843035	98.49
open_acc_6m	842681	98.45
open_il_6m	842681	98.45
open_il_12m	842681	98.45
open_il_24m	842681	98.45
total_bal_il	842681	98.45
open_rv_12m	842681	98.45
open_rv_24m	842681	98.45
max_bal_bc	842681	98.45
all_util	842681	98.45
inq_fi	842681	98.45
total_cu_tl	842681	98.45
inq_last_12m	842681	98.45
Desc	734157	85.77
mths_since_last_record	724785	84.67
mths_since_last_major_derog	642830	75.1
mths_since_last_delinq	439812	51.38
next_pymnt_d	252971	29.55
tot_coll_amt	67313	7.86
tot_cur_bal	67313	7.86
total_rev_hi_lim	67313	7.86
emp_title	49443	5.78
emp_length	43061	5.03
last_pymnt_d	8862	1.04
revol_util	446	0.05
last_credit_pull_d	50	0.01
collections_12_mths_ex_med	56	0.01
Id	0	0
member_id	0	0
loan_amnt	0	0
funded_amnt	0	0
funded_amnt_inv	0	0

term	0	0
int_rate	0	0
installment	0	0
grade	0	0
sub_grade	0	0
home_ownership	0	0
annual_inc	0	0
verification_status	0	0
issue_d	0	0
pymnt_plan	0	0
purpose	0	0
title	33	0
zip_code	0	0
addr_state	0	0
dti	0	0
delinq_2yrs	0	0
earliest_cr_line	0	0
inq_last_6mths	0	0
open_acc	0	0
pub_rec	0	0
revol_bal	0	0
total_acc	0	0
initial_list_status	0	0
out_prncp	0	0
out_prncp_inv	0	0
total_pymnt	0	0
total_pymnt_inv	0	0
total_rec_prncp	0	0
total_rec_int	0	0
total_rec_late_fee	0	0
recoveries	0	0
collection_recovery_fee	0	0
last_pymnt_amnt	0	0
policy_code	0	0
application_type	0	0
acc_now_delinq	0	0
default_ind	0	0



**Deleting Variable having Missing Values More than 50%**

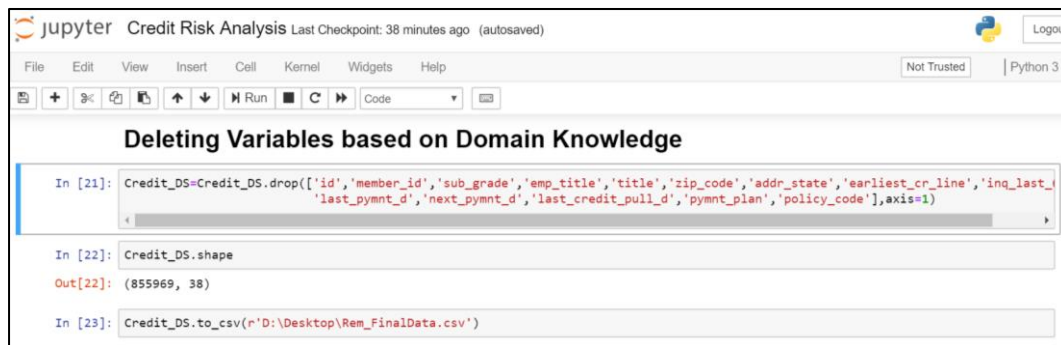
```
In [14]: Credit_DS = Credit_DS.loc[:,Credit_DS.isnull().sum()/len(Credit_DS) < .50 ]

In [15]: Credit_DS.shape

Out[15]: (855969, 52)
```

```
In [*]: Credit_DS.to_csv(r'D:\Desktop\Rem_Data.csv')
```

## 6. Deleting Variable based on Domain Knowledge



```

jupyter Credit Risk Analysis Last Checkpoint: 38 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help
Not Trusted Python 3

Deleting Variables based on Domain Knowledge

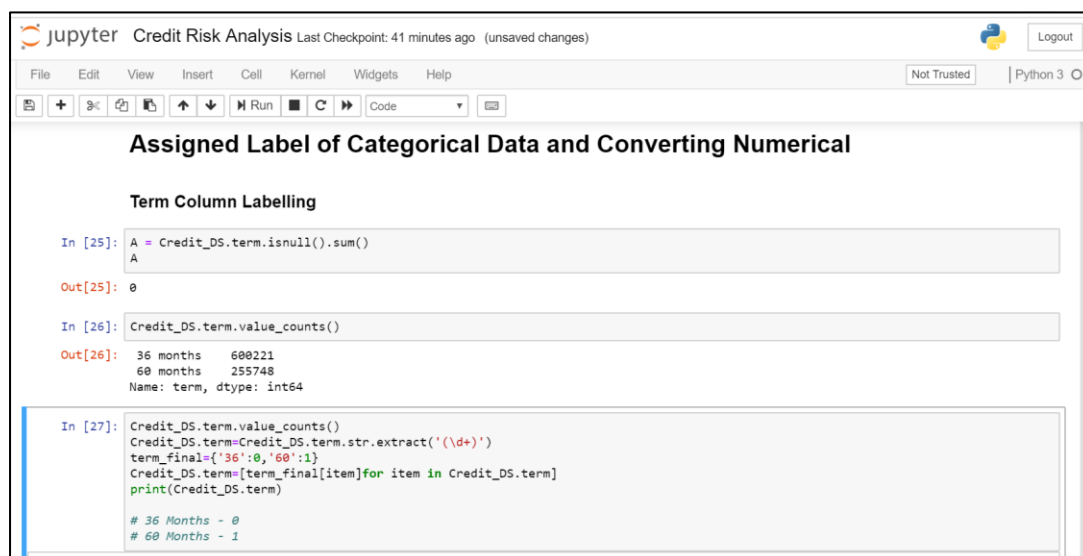
In [21]: Credit_DS=Credit_DS.drop(['id','member_id','sub_grade','emp_title','title','zip_code','addr_state','earliest_cr_line','inq_last_12m','last_pymnt_d','next_pymnt_d','last_credit_pull_d','pymnt_plan','policy_code'],axis=1)

In [22]: Credit_DS.shape
Out[22]: (855969, 38)

In [23]: Credit_DS.to_csv(r'D:\Desktop\Rem_FinalData.csv')
  
```

## 7. Treating Missing Value of Remaining variable and Converting Categorical Variable to Numeric by Label Encoding Method.

For Example:



```

jupyter Credit Risk Analysis Last Checkpoint: 41 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help
Not Trusted Python 3

Assigned Label of Categorical Data and Converting Numerical

Term Column Labelling

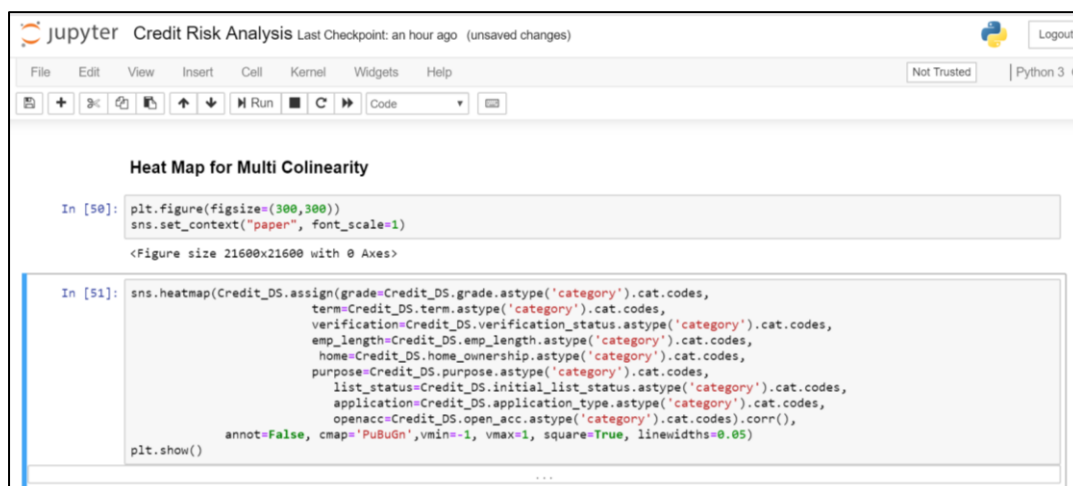
In [25]: A = Credit_DS.term.isnull().sum()
A
Out[25]: 0

In [26]: Credit_DS.term.value_counts()
Out[26]: 36 months    600221
        60 months    255748
        Name: term, dtype: int64

In [27]: Credit_DS.term.value_counts()
Credit_DS.term=Credit_DS.term.str.extract('(\d+)')
term_final={'36':0,'60':1}
Credit_DS.term=[term_final[item]for item in Credit_DS.term]
print(Credit_DS.term)

# 36 Months - 0
# 60 Months - 1
  
```

## 8. To Check there is no Much Multi – Collinearity using Heat Map



```

jupyter Credit Risk Analysis Last Checkpoint: an hour ago (unsaved changes)

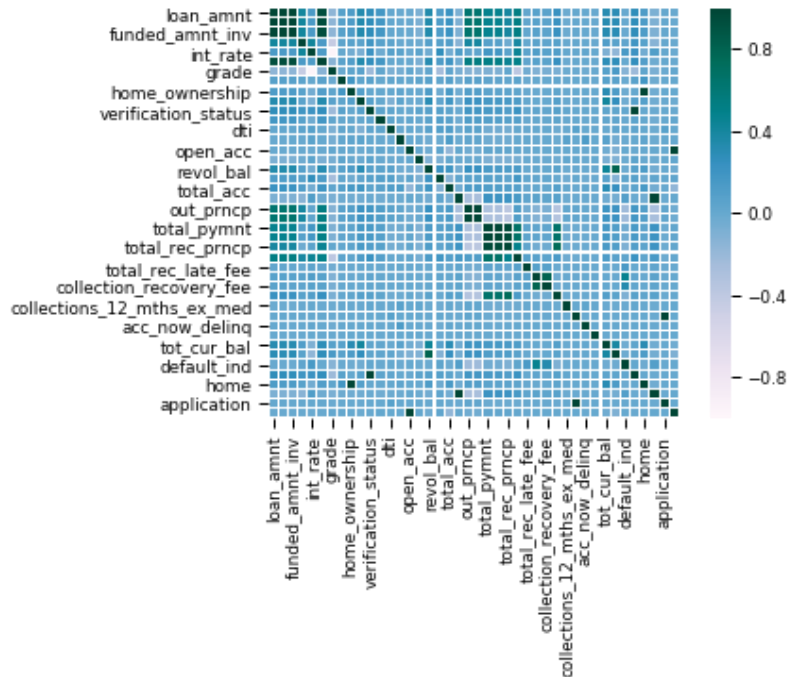
File Edit View Insert Cell Kernel Widgets Help
Not Trusted Python 3

Heat Map for Multi Colinearity

In [50]: plt.figure(figsize=(300,300))
sns.set_context("paper", font_scale=1)

<Figure size 21600x21600 with 0 Axes>

In [51]: sns.heatmap(Credit_DS.assign(grade=Credit_DS.grade.astype('category').cat.codes,
term=Credit_DS.term.astype('category').cat.codes,
verification=Credit_DS.verification_status.astype('category').cat.codes,
emp_length=Credit_DS.emp_length.astype('category').cat.codes,
home=Credit_DS.home_ownership.astype('category').cat.codes,
purpose=Credit_DS.purpose.astype('category').cat.codes,
list_status=Credit_DS.initial_list_status.astype('category').cat.codes,
application=Credit_DS.application_type.astype('category').cat.codes,
openacc=Credit_DS.open_acc.astype('category').cat.codes).corr(),
annot=False, cmap='PuBuGn',vmin=-1, vmax=1, square=True, linewidths=0.05)
plt.show()
  
```



## 9. Splitting the Data into Testing and Training Data using issue\_d Column

```

jupyter Credit Risk Analysis Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Split Train and Test data with issue_d date.

In [53]: Credit_DS.issue_d=pd.to_datetime(Credit_DS.issue_d,infer_datetime_format=True)
col_name='issue_d'
print(Credit_DS[col_name].dtype)

datetime64[ns]

In [54]: split_data="2015-06-01"
Training=Credit_DS[Credit_DS['issue_d']<split_data]
Training.shape

Out[54]: (598978, 38)

In [55]: Testing=Credit_DS.loc[Credit_DS['issue_d']>="2015-06-01",:]
Testing.shape

Out[55]: (256991, 38)

```

## 10. Scaling the Data

```

jupyter Credit Risk Analysis Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Scaling The Data

In [59]: from sklearn.preprocessing import StandardScaler
scaler= StandardScaler()
scaler.fit(X_train)
X_train=scaler.transform(X_train)
X_test=scaler.transform(X_test)
print(X_train)
print(X_test)

[[-1.14444640e+00 -1.14341202e+00 -1.13988120e+00 ... 1.22446777e-03
  4.79105551e-03  2.72648272e-02]
 [-1.44433354e+00 -1.44362136e+00 -1.43672608e+00 ... 1.22446777e-03
  4.79105551e-03  2.72648272e-02]
 [-1.45632902e+00 -1.45562974e+00 -1.44871981e+00 ... 1.22446777e-03
  4.79105551e-03  2.72648272e-02]
 ...
 [-1.84807551e-01 -1.82742115e-01 -1.77384177e-01 ... -1.76378776e-02
  -7.29853126e-01 -3.85913557e-01]
 [-3.04762407e-01 -3.02825853e-01 -2.97321501e-01 ... -1.76378776e-02
  -5.61200123e-01 -1.23366454e-01]
 [ 6.54876440e-01  6.57844050e-01  6.62177089e-01 ... -1.76378776e-02
  -7.35913223e-01  3.05460480e-01]]
 [[ 1.25465072  1.25826274  1.26186371 ... -0.01763788  2.03008646

```

## BUILDING MULTIPLE MODELS

### Model – 1 – Logistic Regression

```
jupyter Credit Risk Analysis Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
In [73]: from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit(X_train,Y_train)
Y_pred = classifier.predict(X_test)
print(list(zip(Y_test,Y_pred)))

In [75]: print(classifier.coef_)
print(classifier.intercept_)

[[ 2.17493176e+00  1.38620321e+01  9.19229856e+00  1.16776380e-01
  1.10154297e+00  1.14982128e+00  5.96673189e-01  2.84121707e-02
 -8.47874498e-02 -1.13756355e-01 -5.75222289e-02 -4.93696376e-02
 -4.78896509e-03 -3.10269478e-02 -4.66993186e-02 -9.40237162e-02
 -7.06279119e-02 -1.75248375e-02  3.69442189e-02  2.28298089e-01
 -1.14244187e+01 -1.13331845e+01 -6.36379793e+00 -7.51858035e+00
 -1.00731504e+01  3.38741019e+00  1.95857252e-01  2.14780391e+01
  6.74274310e+00 -1.49043250e+01 -3.58587715e-02  0.00000000e+00
 -6.03376045e-02 -6.29534876e-01  6.33526057e-02  4.99645617e-02]]
[-6.15456559]
```

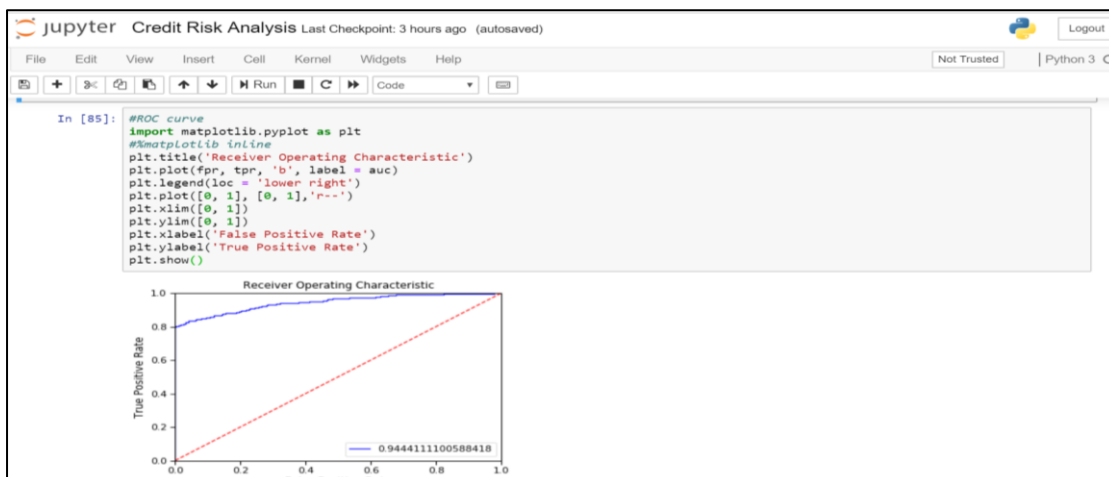
```
jupyter Credit Risk Analysis Last Checkpoint: 2 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3
In [81]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
cfm = confusion_matrix(Y_test, Y_pred)
print(cfm)
print("Classification Report :")
print(classification_report(Y_test, Y_pred))
acc = accuracy_score(Y_test, Y_pred)
print("Accuracy of the model: ",acc)

[[256633  47]
 [ 63  248]]
Classification Report :
              precision    recall  f1-score   support

      0       1.00      1.00      1.00     256680
      1       0.84      0.80      0.82       311

   micro avg       1.00      1.00      1.00     256991
   macro avg       0.92      0.90      0.91     256991
  weighted avg       1.00      1.00      1.00     256991

Accuracy of the model:  0.9995719694464008
```





## Model – 2 – Decision Tree

jupyter Credit Risk Analysis Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3

**Decision Tree Model**

```
In [76]: from sklearn.tree import DecisionTreeClassifier
model_DT = DecisionTreeClassifier(random_state=10,min_samples_leaf=100,max_depth=25,criterion='gini')
#default criterion is gini
model_DT.fit(X_train,Y_train)
```

```
Out[76]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=25,
max_features=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=100, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort=False, random_state=10,
splitter='best')
```

jupyter Credit Risk Analysis Last Checkpoint: an hour ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3

**Evaluation of Model**

```
In [78]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

cfm = confusion_matrix(Y_test, Y_pred)
print(cfm)

print("Classification report: ")

print(classification_report(Y_test, Y_pred))

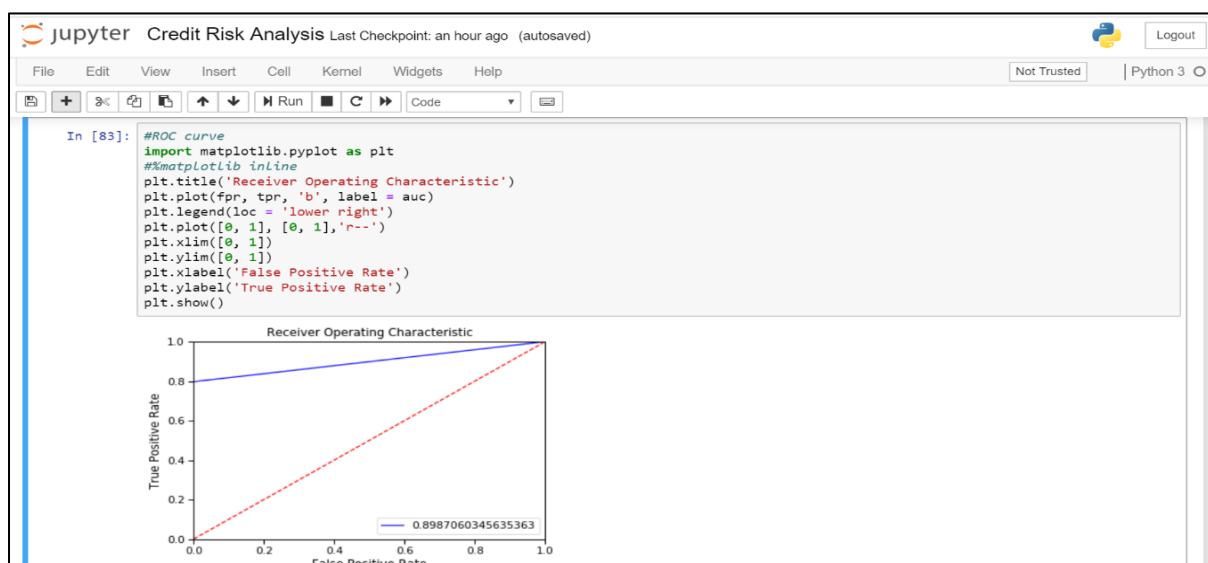
acc = accuracy_score(Y_test, Y_pred)
print("Accuracy of the model: ", acc)
```

```
[[256676  4]
 [ 63 248]]
Classification report:
              precision    recall  f1-score   support

      0       1.00      1.00      1.00     256680
      1       0.98      0.80      0.88       311

   micro avg       1.00      1.00      1.00     256991
   macro avg       0.99      0.90      0.94     256991
  weighted avg       1.00      1.00      1.00     256991

Accuracy of the model:  0.9997392904809896
```



## MODEL 3 – ADAPTIVE BOOST (ADABOOST)

jupyter Credit Risk Analysis Last Checkpoint: an hour ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

ADA Booster Classifier

```
In [88]: from sklearn.ensemble import AdaBoostClassifier
ada_model=AdaBoostClassifier()
ada_model.fit(X_train,Y_train)
Y_pred=ada_model.predict(X_test)
from sklearn.metrics import confusion_matrix, accuracy_score,classification_report
cm=confusion_matrix(Y_test,Y_pred)
print(cm)
print("Classification report: ")
print(classification_report(Y_test,Y_pred))
acc=accuracy_score(Y_test, Y_pred)
print("Accuracy of the model: ",acc)
```

```
[[248532  8148]
 [    63   248]]
Classification report:
      precision    recall  f1-score   support

      0       1.00      0.97      0.98     256680
      1       0.03      0.80      0.06        311

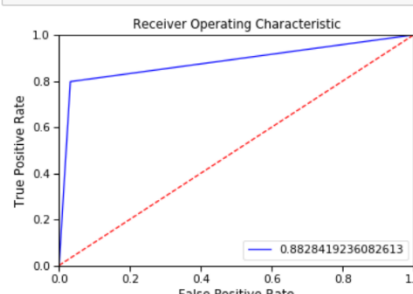
 micro avg       0.97      0.97      0.97     256991
 macro avg       0.51      0.88      0.52     256991
 weighted avg     1.00      0.97      0.98     256991

Accuracy of the model:  0.9680494647672486
```

jupyter Credit Risk Analysis Last Checkpoint: an hour ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
In [91]: #ROC curve
import matplotlib.pyplot as plt
#%matplotlib inline
plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b', label = auc)
plt.legend(loc = 'lower right')
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.show()
```



Receiver Operating Characteristic

True Positive Rate

False Positive Rate

0.8828419236082613

## MODEL 4 – EXTREMELY RANDOMIZED TREES (EXTRA TREES CLASSIFIER)

jupyter Credit Risk Analysis Last Checkpoint: a minute ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

### Extra Tree Classifier

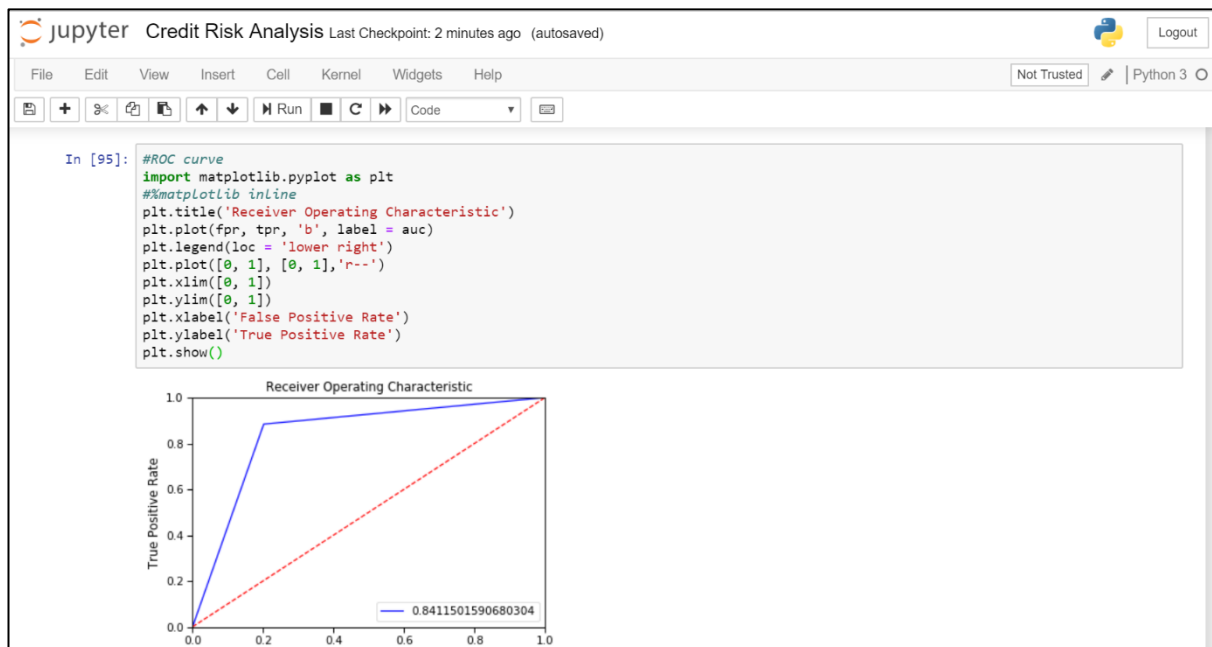
```
In [92]: from sklearn.ensemble import ExtraTreesClassifier
model=ExtraTreesClassifier(4,random_state=100)
model=model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)
from sklearn.metrics import confusion_matrix, accuracy_score,classification_report
cm=confusion_matrix(Y_test,Y_pred)
print(cfm)
print("Classification report: ")
print(classification_report(Y_test,Y_pred))
acc=accuracy_score(Y_test, Y_pred)
print("Accuracy of the model: ",acc)
```

```
[[204845  51835]
 [    36    275]]
Classification report:
              precision    recall  f1-score   support

      0       1.00      0.80      0.89    256680
      1       0.01      0.88      0.01      311

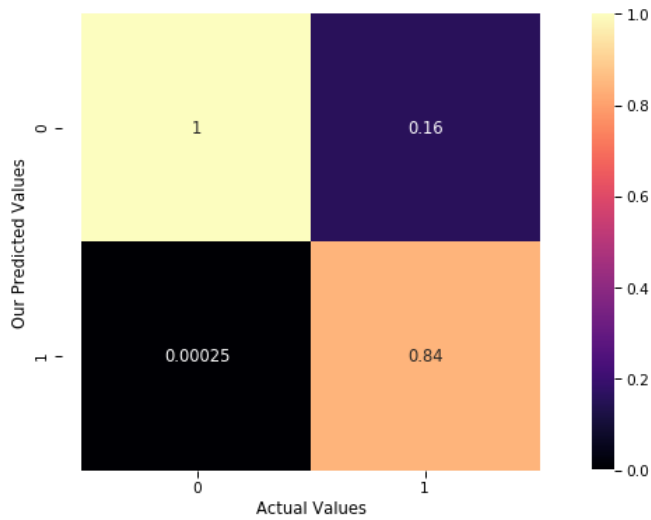
   micro avg       0.80      0.80      0.80    256991
   macro avg       0.50      0.84      0.45    256991
  weighted avg       1.00      0.80      0.89    256991

Accuracy of the model: 0.7981602468568938
```

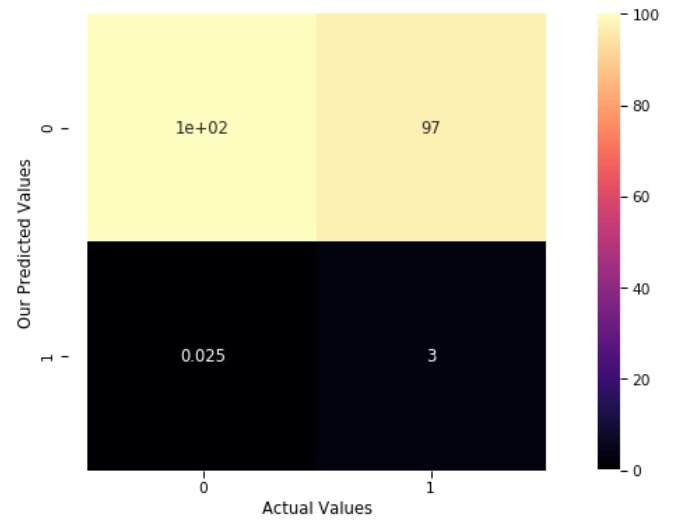


## CONFUSION MATRIX

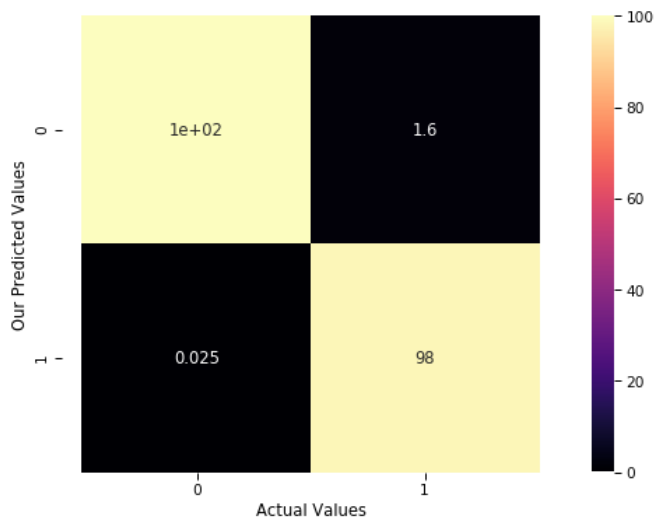
### Model 1 – Logistic Regression



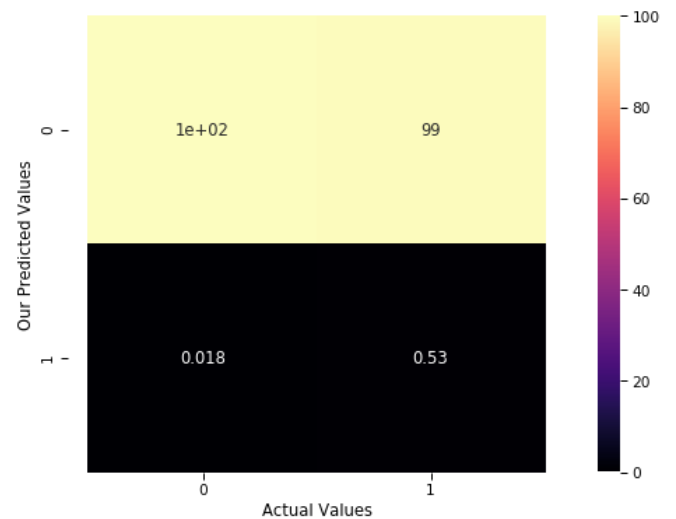
### Model 3 – Ada Boost Classifier



### Model 2 – Decision Tree

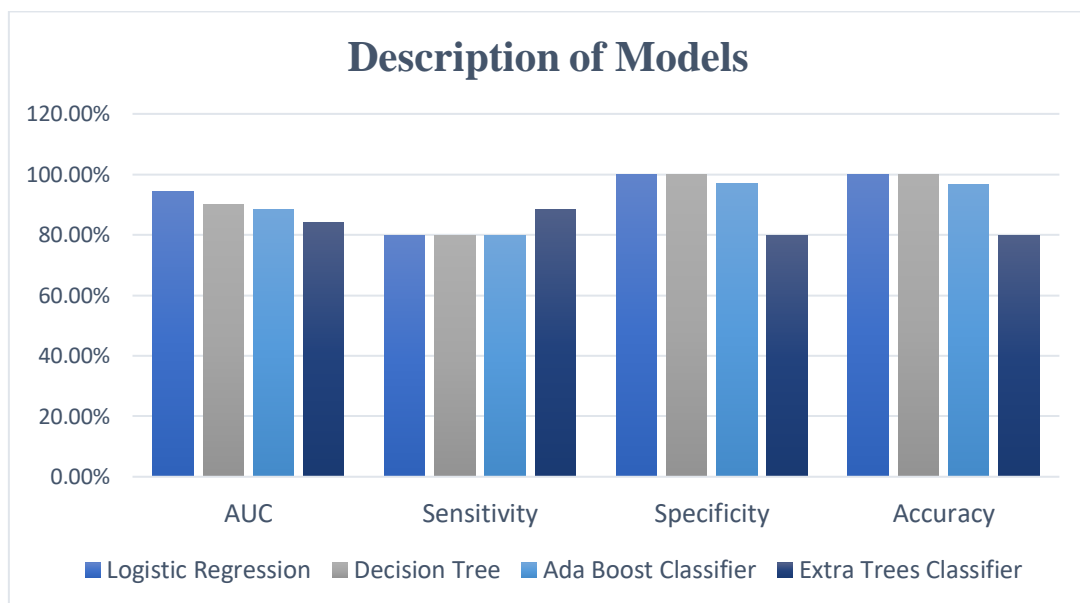


### Model 4 – Extra Trees Classifier



## COMPARISION OF AUC, FPR, TPR

Sr.no	Model Name	AUC	Sensitivity	Specificity	Accuracy
1	Logistic Regression	94.44%	0.797428	0.999817	99.96%
2	Decision Tree	89.87%	0.797428	0.999984	99.97%
3	Ada Boost Classifier	88.28%	0.797428	0.968256	96.80%
4	Extra Trees Classifier	84.11%	0.884244	0.798056	79.82%



## CONCLUSION

Credit Risk Analysis is a very crucial part of the banking sector and it plays an important role in the growth of the bank's profit. Using analysing techniques one can predict or analyse that a person applying for loan will repay the loan or not. So, multiples algorithms have been implemented to analyse a defaulter. The best model selected out of all models that have been tested is Logistic Regression model with an accuracy of 99.957%, which is cross verified with K-fold cross validation technique having the same accuracy of 99.956%. Further insights gained using visualizations from Tableau is that the bank is really working hard to come up from crisis of loss of revenue. To overcome this crisis, credit risk analysis will help them to grow and earn profit.