Credit Scoring Model

Big Data and Analytics mini project report

Electronics and Telecommunication

By

Shubham Shirsekar (40) [BE-ET-2]

Shailey Shah (29) [BE-ET-2]

Sharvin Ramkumar (33) [BE-ET-2]

Ajinkya Shelke (34) [BE-ET-2]

Under the guidance of

Ruchi Chauhan (Professor)

Department of Electronics and Telecommunication Engineering,

Atharva College of Engineering, Malad (W)

University of Mumbai

2021-22

Abstract

Having a good credit score has become one of the important parameters for everyone these days as the banks and other money lending organizations consider the customer's credit score rating before lending them any money. Hence maintaining a good credit score has become essential for everyone. Considering credit scores the banks decide to lend money to anyone.. The system will be working on various algorithms & parameters. Big data analytics has revolutionized the way credit scores are calculated. As a large number of variables are needed to be considered while calculating the credit score without the use of big data technologies it becomes difficult to get a comprehensive view. Using big data technologies one can quickly analyse and calculate the credit score of multiple numbers of prospective borrowers and choose the best one in terms of risk factor. Using Jupyter notebook and python we have developed a code to model a credit score with the help of the borrower's financial history.

Keywords *— Big data, Credit scoring, Jupyter notebook, Python, Logistic regression, data analysis, modelling.*

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

A credit scoring model is a tool that is typically used in the decision-making process of accepting or rejecting a loan. A credit scoring model is the result of a statistical model which, based on information about the borrower (e.g. age, number of previous loans, etc.), allows one to distinguish between "good" and "bad" loans and give an estimate of the probability of default. A credit score is a number between 300–850 that depicts a consumer's creditworthiness. The higher the score, the better a borrower looks to potential lenders. A credit score is based on credit history: number of open accounts, total levels of debt, and repayment history, and other factors. Lenders use credit scores to evaluate the probability that an individual will repay loans in a timely manner. Quantifying the risk involved in lending is one of the crucial steps. Nowadays even a single person's financial history consists of many things and hence it has become unfeasible to simply calculate a credit score while looking at a person's income tax returns or other transaction history. Without the help of big data technologies one may never get a comprehensive overview of a borrower's financial capability to return a loan. Big data technologies can handle the large amount of data generated from one's financial history and further analysis can be done on that data to refine and calculate a credit score. Machine learning models can be trained and tested on datasets to automate and further refine this process and automate it. Credit score has become a vital part of an individual's life or even for a business without one it is often difficult to find a good line of credit and one may have no choice but to accept a loan with unreasonably high interest rate. From the borrower's perspective it helps them to open up avenues for credit that may not be available without a credit score. Then they can make an informed decision with all the information in front of them.

Motivation

No one can argue the importance of a credit score in today's world. However with so much riding on the credit score it becomes of paramount importance to get an accurate credit score. An ambiguous credit score may hinder the borrower's borrowing capability and hurt his future prospects gravely. Hence we tried to model a program that will test the accuracy of calculated credit score by taking into account previous accurately calculated credit scores as training models. Getting a perfect credit score is not feasible and neither can any lender blindly follow the credit score to decide the risk factor of the money being repaid on time. There may be other unfortunate circumstances that may hinder a borrower with a perfect credit score from repaying the loan on time. No one can predict the future but having a credit score can definitely help in the process of securing a loan as it gives the lender a baseline to begin.

Problem Statement

Access to credit can make or break a company or an individual's future prospects. Lending institutions have made it mandatory nowadays to have a credit score to secure any amount of loan from them. Therefore it is indispensable in the modern world to secure a loan without having a credit score. Furthermore since this score is of so much importance it is necessary to make sure that it is accurate. To calculate a credit score a large amount of data is needed to be taken as input for which big data technologies are needed to process and analyse.

Objectives

The objectives are as follows:

- To calculate credit score with the help of financial history of the borrower
- To analyse the data and classify it as needed
- To check the accuracy of the calculated credit score
- To use machine learning techniques to automate and refine the score

Scope

- The credit score modelling project will aim to calculate an accurate credit score with the help of the financial history data available.
- This model can be used by lending institutions as well as banks to assess the risk involved in lending to a particular borrower.
- Quantify the risk involved in lending and give a baseline to decide whether to sanction the loan or not.

Chapter 2
Review of literature

Title	Author	Source	Description
Use of Machine Learning Techniques to Create a Credit Score Model for Airtime Loans	Bernard, Dushimimana, Yvonne Wambui, Timothy Lubega, Patrick E. McSharry	Journal of Risk and Financial Management, August 2020	Airtime lending default rates are typically lower than those experienced by banks and microfinance institutions (MFIs) but are likely to grow as the service is offered more widely. In this paper, credit scoring techniques are reviewed, and that knowledge is built upon to create an appropriate machine learning model for airtime lending.
Experimental analysis of machine learning methods for credit score classification	Diwakar Tripathi, Damodar Reddy Edla, Annushree Bablani, Alok Kumar Shukla, B. Ramachandra Reddy	Progress in Artificial Intelligence, February 2021	Credit scoring concerns an emerging empirical model to assist financial institutions in the financial decision-making process. Credit risk analysis plays a vital role in the decision-making process; statistical and machine learning approaches are utilized to estimate the risk associated with a credit applicant.
A QUANTITATIVE THEORY OF THE	Satyajit Chatterjee Dean Corbae	NATIONAL BUREAU OF	Our theory is founded on the premise that an

CREDIT SCORE	Kyle P. Dempsey José-Víctor Ríos-Rull	ECONOMIC RESEARCH, August 2020	individual's true propensity to repay — i.e., the individual's true type — is hidden from her creditors, and it is the presence of this persistent hidden information that makes an individual's history of actions relevant for lenders. Our theory is dynamic: at any point in time, lenders use a person's observable history of actions to perform a Bayesian update of her type; individuals understand this and choose actions mindful of the consequence any action has on the future beliefs of lenders.
A Machine Learning Framework for Improving Classification Performance on Credit Approval	Pulung Hendro Prastyo, Septian Eko Prasetyo, Shindy Arti	International Journal on Informatics for Development, June 2021	An evaluation tool that is usually used in the decision -making process is the credit scoring model that aims to refuse or accept loan requests .

Fig 2.1 Literature survey

Chapter 3
Requirement Analysis

Software Requirements:

- Jupyter Notebook
- Python 3.6 (or later)
- Windows 7 (or later)
- VS Code editor

Dataset Required:

- Financial history

Description:

- Jupyter Notebook:
  The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. It is also known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document. Computational notebooks have been around for decades, but Jupyter in particular has exploded in popularity over the past couple of years.
- Python:
  Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
- Windows OS:
  Microsoft Windows, commonly referred to as Windows, is a group of several proprietary graphical operating system families, all of which are developed and marketed by Microsoft.
- VS Code editor:
  Visual Studio Code is an integrated development environment made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Fig 3.1 Company logos

Chapter 4
Report on Mini Project
Existing Systems for calculating credit scores

- **FICO:**
  FICO Scores are calculated using many different pieces of credit data in your credit report. This data is grouped into five categories: payment history (35%), amounts owed (30%), length of credit history (15%), new credit (10%) and credit mix (10%). Developed by Fair Isaac Corporation, these are widely used. FICO Scores consider both positive and negative information in your credit report. The percentages in the chart reflect how important each of the categories is in determining how your FICO Scores are calculated. The importance of these categories may vary from one person to another.
- **CIBIL:**
  CIBIL stands for TransUnion CIBIL Limited, an Indian company that has access to your credit information. This information refers to all financial transactions where you have borrowed or repaid money. CIBIL scores can range between 300 and 900.
- **Credit Karma:**
  Online free credit scoring program. They work with Equifax and TransUnion, two of the three major credit bureaus, to give our members access to their scores for free. On top of showing free credit scores, their members can see free personalized offers and recommendations on their profiles to help them make their next financial move, their best move.

Implementation

Part 1 - Data Processing: Cleaning and Transforming Raw Data into the Understandable Format

Part 2 - Profiling: Data profiling is the process of examining the data available from an existing information source (e.g. a database or a file) and collecting statistics or informative summaries about that data.

Part 3 - Pearson's Correlations: Pearson’s Correlation is bounded in [-1, 1]. If it is positive, the two variables tend to be high or low together. If it is negative, the two variables tend to be opposite of each other. If it is zero or close to zero they don’t affect each other. Sx, Sy are the standard deviations of the X, Y series respectively. Standard Deviation (σ) is a measure of the spread of the distribution.

Part 4 - Histogram Plots

Part 5 - Logistic Regression Model: In statistics, logistic regression(or logit regression), is a regression model where the dependent variable is categorical. This article covers the case of a binary dependent variable—that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose etc.
Part 4 - Optimal Cut-off Using Youden's Index
Part 5 - Principal Components Analysis

Flowchart

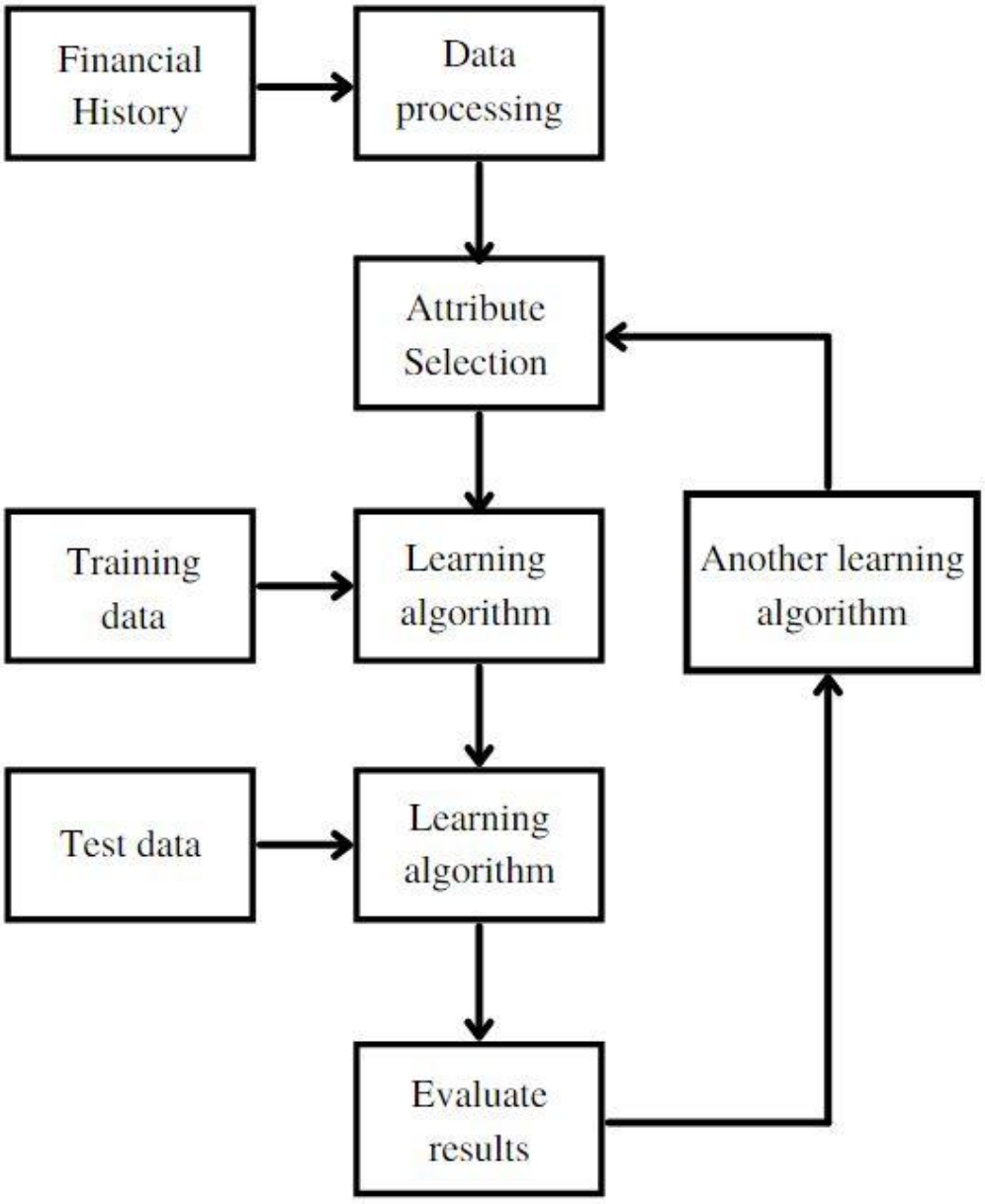

Fig 4.1 Flowchart of the system

Algorithms

- **Data Processing:**
  Cleaning and Transforming Raw Data into an understandable Format. Removing outliers and ambiguous data is important to improve the accuracy of the system and make it reliable.
- **Data profiling:**
  Data profiling is the process of examining the data available from an existing information source (e.g. a database or a file) and collecting statistics or informative summaries about that data.
- **Pearson's correlation:**
  In statistics, the Pearson correlation coefficient — also known as Pearson's r, the Pearson product-moment correlation coefficient, the bivariate correlation, or colloquially simply as the correlation coefficient — is a measure of linear correlation between two sets of data.
- **Logistic Regression:**
  In statistics, logistic regression(or logit regression), is a regression model where the dependent variable is categorical. This article covers the case of a binary dependent variable—that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose etc.
- **ROC Curve:**
  An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate.
- **Youden’s Index:**
  Youden's index integrates sensitivity and specificity information under circumstances that emphasize both sensitivity and specificity, with a value that ranges from 0 to 1.

Dataset

The raw dataset is in the file "CreditScoring.csv" which contains 4455 rows and 14 columns. Credit status of the individual which is classified as good (above 690) or bad (below 690) is included in the first column. The second column indicates the seniority of the individual which is basically the experience level of the person. Next column shows the type of home ownership of the individual. There are three types here - rent, owner, priv, parents, and ignore. The time column indicates the time period for which the loan is requested. Next column has the age. The sixth column comments on the marital status of the individual. Next column indicates whether complete financial records are present from the beginning of professional career or not. The eighth column describes the type of job which may be freelance, fixed or part time. Next column

includes the monthly expenses. The tenth column shows the monthly income. Next column includes the value of the assets held by the individual if any. The twelfth column displays the debt undertaken by the individual if any. Next column displays the amount of credit requested by the individual. The fourteenth column indicates the cost of the service or product for which the credit is requested. The next column calculated the ratio of the amount and price column. The last column indicates the savings by the individual.

Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn as sklearn
import seaborn as sns
%matplotlib inline
from scipy import stats
data = pd.read_csv("CreditScoring(After_Preprocessing).csv")
data.head()
## Dataset Dimensions
data.shape
# Checking if any Missing Values are there in the Dataset
# No Missing Values are there in the Dataset
data.isnull().any()
## checking if any categorical Features are there in the Dataset
categorical_data = data.select_dtypes(exclude=[np.number])
print ("There are {} categorical Columns in Dataset".format(categorical_data.shape[1]))
# Name of all the Categorical Features Present in the Dataset
categorical_data.any()
from sklearn.preprocessing import LabelEncoder
model = LabelEncoder()
data['Status'] = model.fit_transform(data['Status'].astype('str'))
data['Home'] = model.fit_transform(data['Home'].astype('str'))
data['Marital'] = model.fit_transform(data['Marital'].astype('str'))
data['Job'] = model.fit_transform(data['Job'].astype('str'))
data['Records'] = model.fit_transform(data['Records'].astype('str'))
#Checking Data Types of the Features for Confirmation
data.dtypes
# Summary of the Data
data.describe()
```

```
## Value Counts of 'GOOD' Status and 'BAD' Status
## 'GOOD': 3197 and 'BAD': 1249
data.Status.value_counts()
# As per the Dataset there are approximately 72% GOOD Score and 28% BAD Score instances
status_count = data.Status.value_counts()/len(data)
status_count
 Correlation between all the Features
# Correlation Plot
corr = data.corr()
sns.heatmap(corr)
## Correlation Values of all the Features with respect to Target Variable 'Status'
## Top 10 Values
print (corr['Status'].sort_values(ascending=False)[:10], '\n')

## Last 5 Values
print (corr['Status'].sort_values(ascending=False)[-5:])
## Visualising Correlation Matrix with actual Correlation Values
cmap=sns.diverging_palette(5, 250, as_cmap=True)

def magnify():
    return [dict(selector="th",
                 props=[("font-size", "7pt")]),
            dict(selector="td",
                 props=[('padding', "0em 0em")]),
            dict(selector="th:hover",
                 props=[("font-size", "15pt")]),
            dict(selector="tr:hover td:hover",
                 props=[('max-width', '200px'),
                        ('font-size', '15pt')])
]

corr.style.background_gradient(cmap, axis=1)\
    .set_properties(**{'max-width': '80px', 'font-size': '10pt'})\
    .set_caption("Hover to magify")\
    .set_precision(3)\
    .set_table_styles(magnify())
num_bins = 10

data.hist(bins = num_bins, figsize=(20,15))
plt.savefig("Data_Histogram_Plots")
```

```
plt.show()
list( data.columns )
X = list( data.columns )
X.remove( 'Status' )
X
Y = data['Status']
credit_data = pd.get_dummies( data[X], drop_first = True )
len( credit_data.columns )
### Splitting Dataset into 'Training' and 'Testing' Dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( data[X], Y, test_size = 0.3, random_state = 42 )
## Using 'statsmodel.api' you can use R-style formulas together with pandas data frames to fit your models
import statsmodels.api as sm
logit = sm.Logit( y_train, sm.add_constant( X_train ) )
lg = logit.fit()
lg.summary()
def get_significant_vars( lm ):
    var_p_vals_df = pd.DataFrame( lm.pvalues )
    var_p_vals_df['vars'] = var_p_vals_df.index
    var_p_vals_df.columns = ['pvals', 'vars']
    return list( var_p_vals_df[var_p_vals_df.pvals <= 0.05]['vars'] )
significant_vars = get_significant_vars( lg )
significant_vars
from sklearn import metrics
def get_predictions( y_test, model ):
    y_pred_df = pd.DataFrame( { 'actual': y_test,
                                "predicted_prob": lg.predict( sm.add_constant( X_test ) ) } )
    return y_pred_df
y_pred_df = get_predictions( y_test, lg )
## Calculating predicted probability for the Target Variable 'Status(Default Classes)'
y_pred_df[0:10]
y_pred_df['predicted'] = y_pred_df.predicted_prob.map( lambda x: 1 if x > 0.5 else 0)
y_pred_df[0:10]
import matplotlib.pylab as plt
%matplotlib
def draw_cm( actual, predicted ):
    cm = metrics.confusion_matrix( actual, predicted, [1,0] )
    sns.heatmap(cm, annot=True,  fmt='.2f', xticklabels = ["No Default", "Default"] , yticklabels = ["No Default", "Default"] )
```

```
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()
draw_cm( y_pred_df.actual, y_pred_df.predicted )
## Finding Overall Accuracy of the Model
print( 'Total Accuracy : ',np.round( metrics.accuracy_score( y_test, y_pred_df.predicted ), 2 ) )
print( 'Precision : ',np.round( metrics.precision_score( y_test, y_pred_df.predicted ), 2 ) )
print( 'Recall : ',np.round( metrics.recall_score( y_test, y_pred_df.predicted ), 2 ) )

cm1 = metrics.confusion_matrix( y_pred_df.actual, y_pred_df.predicted, [1,0] )

sensitivity = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Sensitivity : ', round( sensitivity, 2) )

specificity = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Specificity : ', round( specificity, 2 ) )
sns.distplot( y_pred_df[y_pred_df.actual == 1]["predicted_prob"], kde=False, color = 'b' )
sns.distplot( y_pred_df[y_pred_df.actual == 0]["predicted_prob"], kde=False, color = 'g' )
auc_score = metrics.roc_auc_score( y_pred_df.actual, y_pred_df.predicted_prob  )
round( float( auc_score ), 2 )
def draw_roc( actual, probs ):
    fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
                                      drop_intermediate = False )
    auc_score = metrics.roc_auc_score( actual, probs )
    plt.figure(figsize=(6, 4))
    plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score )
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver operating characteristic example')
    plt.legend(loc="lower right")
    plt.show()

    return fpr, tpr, thresholds
thresholds[0:10]
fpr[0:10]
tpr[0:10]
y_pred_df['predicted_new'] = y_pred_df.predicted_prob.map( lambda x: 1 if x > 0.29 else 0)
```

```
draw_cm( y_pred_df.actual, y_pred_df.predicted_new )
cm = metrics.confusion_matrix( y_pred_df.actual, y_pred_df.predicted_new, [1,0] )
cm_mat = np.array( cm )
cm_mat[1, 0]
cm_mat[0, 1]
def get_total_cost( actual, predicted ):
    cm = metrics.confusion_matrix( actual, predicted, [1,0] )
    cm_mat = np.array( cm )
    return cm_mat[0,1] * 2 + cm_mat[0,1] * 1
get_total_cost( y_pred_df.actual, y_pred_df.predicted_new )
cost_df = pd.DataFrame( columns = ['prob', 'cost'])
idx = 0
for each_prob in range( 20, 50):
    cost = get_total_cost( y_pred_df.actual,
                           y_pred_df.predicted_prob.map(
            lambda x: 1 if x > (each_prob/100)  else 0) )
    cost_df.loc[idx] = [(each_prob/100), cost]
    idx += 1
cost_df.sort_values( 'cost', ascending = True )[0:5]
y_pred_df['predicted_final'] = y_pred_df.predicted_prob.map( lambda x: 1 if x > 0.20 else 0)
draw_cm( y_pred_df.actual, y_pred_df.predicted_final )
print( 'Total Accuracy : ',np.round( metrics.accuracy_score( y_test, y_pred_df.predicted_final ), 2
) )
print( 'Precision : ',np.round( metrics.precision_score( y_test, y_pred_df.predicted_final ), 2 ) )
print( 'Recall : ',np.round( metrics.recall_score( y_test, y_pred_df.predicted_final ), 2 ) )

cm1 = metrics.confusion_matrix( y_pred_df.actual, y_pred_df.predicted_final, [1,0] )

sensitivity = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Sensitivity : ', round( sensitivity, 2) )

specificity = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Specificity : ', round( specificity, 2 ) )
```

Screenshots of the output with description

(1) Data Preprocessing

```
data = pd.read_csv("CreditScoring(After_Preprocessing).csv")
data.head()
```

✓ 0.2s

	Status	Seniority	Home	Time	Age	Marital	Records	Job	Expenses	Income	Assets	Debt	Amount	Price	Finrat	Savings
0	good	9	rent	60	30	married	no_rec	freelance	73	129	0	0	800	846	94.562648	4.200000
1	good	17	rent	60	58	widow	no_rec	fixed	48	131	0	0	1000	1658	60.313631	4.980000
2	bad	10	owner	36	46	married	yes_rec	freelance	90	200	3000	0	2000	2985	67.001675	1.980000
3	good	0	rent	60	24	single	no_rec	fixed	63	182	2500	0	900	1325	67.924528	7.933333
4	good	0	rent	36	26	single	no_rec	fixed	46	107	0	0	310	910	34.065934	7.083871

Fig 4.2 Data preprocessing

```
# Checking if any Missing Values are there in the Dataset
# No Missing Values are there in the Dataset
data.isnull().any()
```

✓ 0.2s

```
Status       False
Seniority    False
Home         False
Time         False
Age          False
Marital      False
Records      False
Job          False
Expenses     False
Income       False
Assets       False
Debt         False
Amount       False
Price        False
Finrat       False
Savings      False
dtype: bool
```

Fig 4.3 Check for missing values

Here we can see that the data is pre processed to remove any outliers and bad data. Also data is checked to see if there are any missing values.

(2) Data Exploration

```
# Summary of the Data
data.describe()
```

✓ 0.1s

	Status	Seniority	Home	Time	Age	Marital	Records	Job	Expenses	Income	Assets	Debt	Amount	Price	Finrat	Savings
count	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000	4446.000000
mean	0.719073	7.991453	2.862348	46.453441	37.084121	1.503599	0.172964	0.610886	55.601439	140.629780	5354.948943	342.257085	1038.763383	1462.480432	72.616409	3.860083
std	0.449502	8.176370	1.308943	14.647979	10.986366	0.891838	0.378259	0.960975	19.520839	80.177896	11534.328183	1244.694549	474.747952	628.555171	20.390595	3.726292
min	0.000000	0.000000	0.000000	6.000000	18.000000	0.000000	0.000000	0.000000	35.000000	1.000000	0.000000	0.000000	100.000000	105.000000	6.702413	-8.160000
25%	0.000000	2.000000	2.000000	36.000000	28.000000	1.000000	0.000000	0.000000	35.000000	90.000000	0.000000	0.000000	700.000000	1116.250000	60.030020	1.615385
50%	1.000000	5.000000	2.000000	48.000000	36.000000	1.000000	0.000000	0.000000	51.000000	124.000000	3000.000000	0.000000	1000.000000	1400.000000	77.096757	3.120000
75%	1.000000	12.000000	4.000000	60.000000	45.000000	2.000000	0.000000	1.000000	72.000000	170.000000	6000.000000	0.000000	1300.000000	1691.500000	88.460263	5.195688
max	1.000000	48.000000	5.000000	72.000000	68.000000	4.000000	1.000000	3.000000	180.000000	959.000000	300000.000000	30000.000000	5000.000000	11140.000000	100.000000	33.250000

Fig 4.4 Data exploration

Here we can see the data exploration stage where we can find the maximum and minimum values along with mean and median values. Additionally top 20%, 50% and 75% values can also be found.

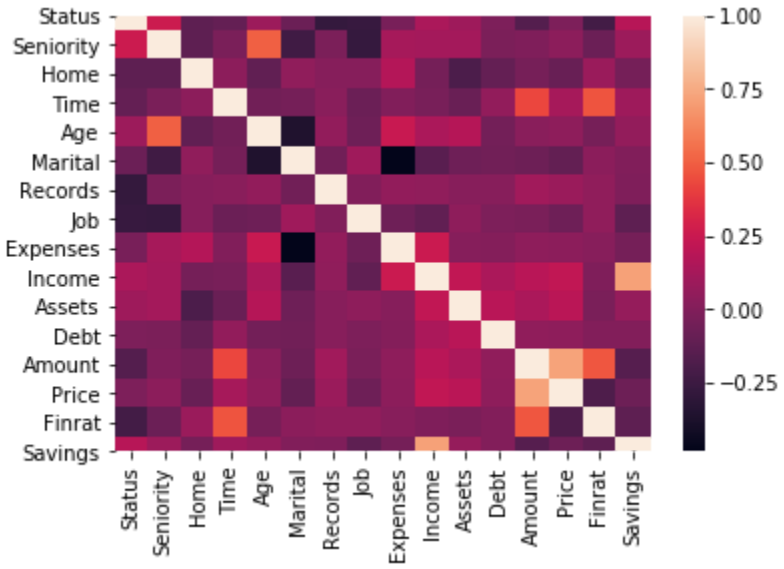

Fig 4.5 Pearson's correlation with feature selection

Here we can see the output obtained from Pearson's correlation. As per the Correlation Matrix and the Correlation Values Calculated, we can say that No Feature in the Dataset is highly Correlated with the Target Attribute 'Status'. Hence, there is no need for Feature Selection as of now and the Classification Model Such as 'Linear Regression ' and 'Logistic Regression' etc, can be performed directly on all the Features of the Dataset.

	Status	Seniority	Home	Time	Age	Marital	Records	Job	Expenses	Income	Assets	Debt	Amount	Price	Finrat	Savings
Status	1	0.26	-0.123	-0.101	0.0953	-0.073	-0.278	-0.268	-0.0296	0.144	0.0972	-0.0104	-0.154	-0.0101	-0.222	0.186
Seniority	0.26	1	-0.121	-0.0219	0.506	-0.229	-0.0161	-0.271	0.126	0.121	0.124	-0.0193	-0.00764	0.0411	-0.0734	0.0932
Home	-0.123	-0.121	1	0.0415	-0.108	0.0519	0.0199	0.0155	0.176	-0.0393	-0.183	-0.0981	-0.0323	-0.087	0.0886	-0.0412
Time	-0.101	-0.0219	0.0415	1	-0.0518	-0.0393	0.0283	-0.0768	-0.00172	-0.0299	-0.083	0.0583	0.431	0.13	0.469	0.102
Age	0.0953	0.506	-0.108	-0.0518	1	-0.358	0.0598	-0.0614	0.248	0.148	0.181	-0.0462	0.029	0.0487	-0.0345	0.062
Marital	-0.073	-0.229	0.0519	-0.0393	-0.358	1	-0.0535	0.106	-0.484	-0.141	-0.0619	-0.0537	-0.0716	-0.102	0.0327	0.000178
Records	-0.278	-0.0161	0.0199	0.0283	0.0598	-0.0535	1	-0.000477	0.0583	0.0535	0.0215	0.0229	0.111	0.0848	0.0511	-0.00334
Job	-0.268	-0.271	0.0155	-0.0768	-0.0614	0.106	-0.000477	1	-0.0637	-0.111	0.0454	-0.0102	-0.0257	-0.0627	0.0517	-0.123
Expenses	-0.0296	0.126	0.176	-0.00172	0.248	-0.484	0.0583	-0.0637	1	0.255	0.0189	0.0148	0.0495	0.0407	0.0246	-0.0381
Income	0.144	0.121	-0.0393	-0.0299	0.148	-0.141	0.0535	-0.111	0.255	1	0.219	0.145	0.191	0.218	-0.00341	0.718
Assets	0.0972	0.124	-0.183	-0.083	0.181	-0.0619	0.0215	0.0454	0.0189	0.219	1	0.192	0.144	0.195	-0.0251	0.067
Debt	-0.0104	-0.0193	-0.0981	0.0583	-0.0462	-0.0537	0.0229	-0.0102	0.0148	0.145	0.192	1	0.0509	0.0439	0.00861	0.00499
Amount	-0.154	-0.00764	-0.0323	0.431	0.029	-0.0716	0.111	-0.0257	0.0495	0.191	0.144	0.0509	1	0.725	0.474	-0.149
Price	-0.0101	0.0411	-0.087	0.13	0.0487	-0.102	0.0848	-0.0627	0.0407	0.218	0.195	0.0439	0.725	1	-0.181	-0.0704
Finrat	-0.222	-0.0734	0.0886	0.469	-0.0345	0.0327	0.0511	0.0517	0.0246	-0.00341	-0.0251	0.00861	0.474	-0.181	1	-0.123
Savings	0.186	0.0932	-0.0412	0.102	0.062	0.000178	-0.00334	-0.123	-0.0381	0.718	0.067	0.00499	-0.149	-0.0704	-0.123	1

Fig 4.6 Correlation Matrix

Visualising Correlation Matrix with actual Correlation Values.

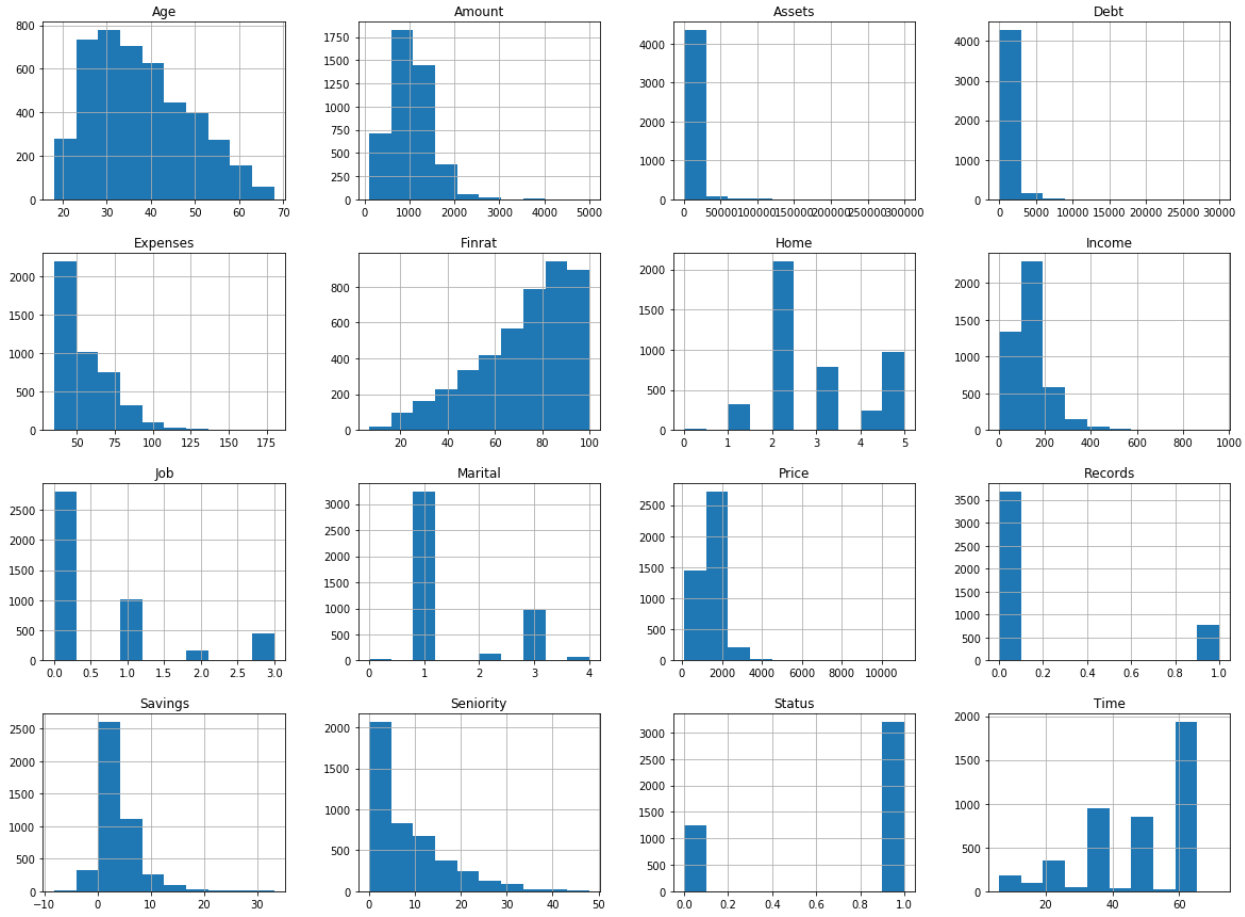

Fig 4.7 Histogram representation of data

Histogram Plots of all the Features in the Dataset

Logit Regression Results

Dep. Variable:	Status	No. Observations:	3112
Model:	Logit	Df Residuals:	3096
Method:	MLE	Df Model:	15
Date:	Sun, 10 Oct 2021	Pseudo R-squ.:	0.2336
Time:	12:59:55	Log-Likelihood:	-1405.2
converged:	True	LL-Null:	-1833.5
Covariance Type:	nonrobust	LLR p-value:	7.074e-173

	coef	std err	z	P>|z|	[0.025	0.975]
const	4.6164	0.578	7.987	0.000	3.484	5.749
Seniority	0.0808	0.009	9.337	0.000	0.064	0.098
Home	-0.1420	0.038	-3.784	0.000	-0.216	-0.068
Time	0.0062	0.005	1.354	0.176	-0.003	0.015
Age	-0.0098	0.005	-1.860	0.063	-0.020	0.001
Marital	-0.1995	0.062	-3.241	0.001	-0.320	-0.079
Records	-1.6714	0.119	-14.088	0.000	-1.904	-1.439
Job	-0.4610	0.047	-9.763	0.000	-0.554	-0.368
Expenses	-0.0132	0.003	-3.936	0.000	-0.020	-0.007
Income	0.0062	0.002	3.653	0.000	0.003	0.010
Assets	3.206e-05	7.84e-06	4.088	0.000	1.67e-05	4.74e-05
Debt	-0.0001	4.03e-05	-3.575	0.000	-0.000	-6.52e-05
Amount	-0.0004	0.000	-0.881	0.378	-0.001	0.000
Price	-0.0003	0.000	-1.230	0.219	-0.001	0.000
Finrat	-0.0269	0.006	-4.307	0.000	-0.039	-0.015
Savings	-0.0032	0.035	-0.090	0.928	-0.071	0.065

Fig 4.8 Logistic regression results

Logistic regression results

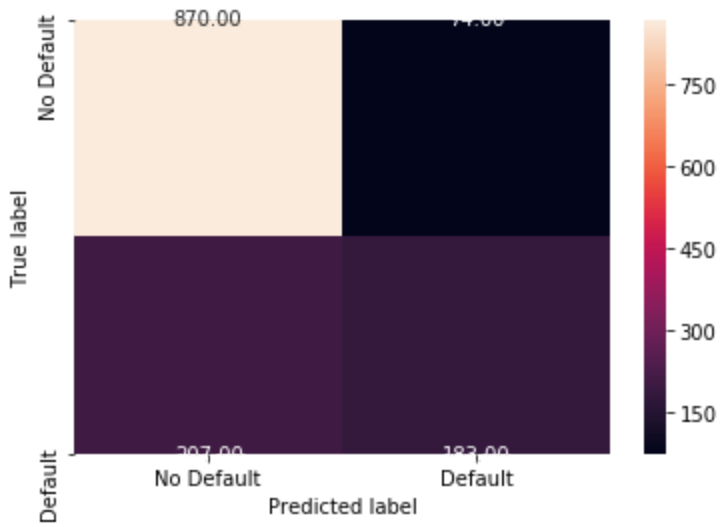

Fig 4.9 Predicted probability in graphical representation

Predicted probability for the Target Variable

```
Total Accuracy :  0.79
Precision :  0.81
Recall :  0.92
Sensitivity :  0.92
Specificity :  0.47
```

Fig 4.10 Accuracy of the model 1

Overall Accuracy of the Model

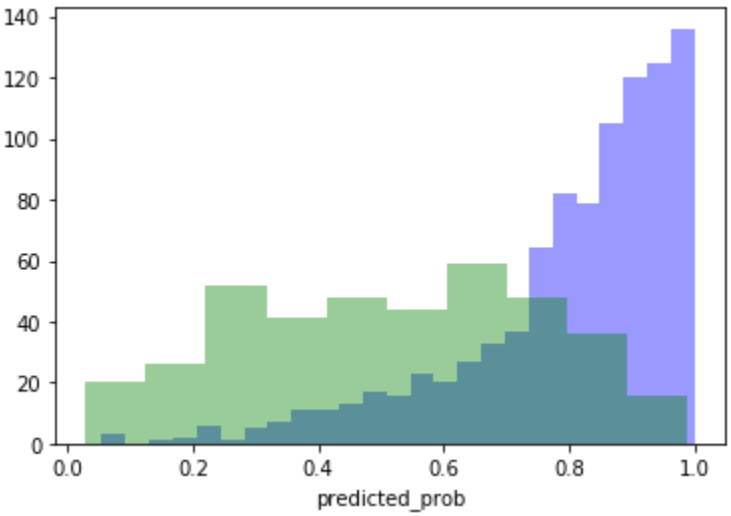

Fig 4.11 Probability distribution plots

Predicted Probability distribution Plots for Defaults(BAD) and Non Defaults(GOOD)

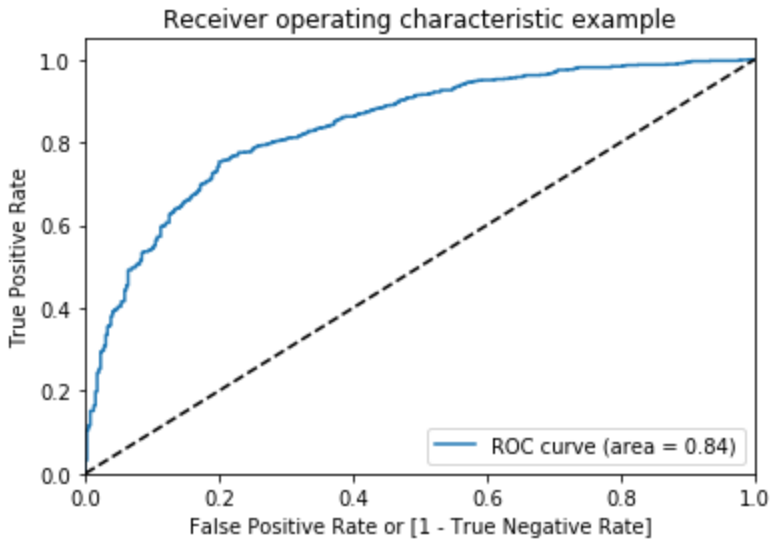

Fig 4.12 ROC curve

Plotting ROC Curve

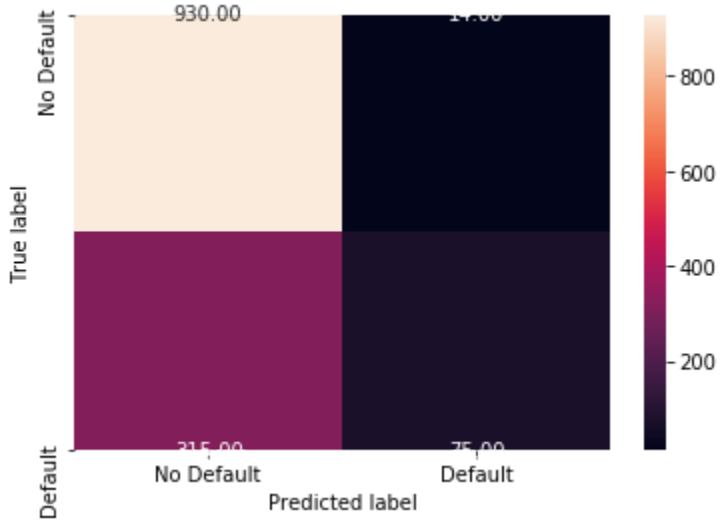

Fig 4.13 Optimal cutoff using Youden's index

Optimal cutoff using Youden's index

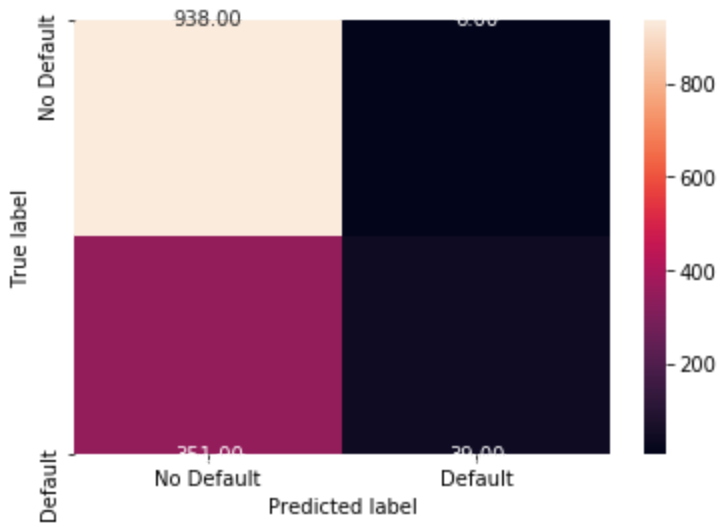

Fig 4.14 Optimal cut-off using cost

Optimal Cut-off Probability using Cost

```
Total Accuracy :  0.73
Precision :  0.73
Recall :  0.99
Sensitivity :  0.99
Specificity :  0.1
```

Fig 4.15 New adjusted accuracy

New adjusted accuracy

Chapter 5
Advantages of using big data for credit scoring

Credit scoring has to take into account a large data set as input and calculate the score according to that. Use of big data makes the task of handling this huge amount of data a piece of cake and also analysing this data becomes easier with the help of big data technologies. Our model can be used to get a baseline for the credit score of any loan applicant. The user just needs to enter their financial history as prompted and they will get an idea of the range that their score will fall in. The accuracy of this model can be improved by using more machine learning techniques. Also since this model is automated it is quicker to calculate a credit score using our model.

- **Quick**
- **Easy to use**
- **Simple to understand**
- **Automated**
- **Gives a starting point**

Chapter 6
Disadvantages and Limitations of our code

Since our code undertakes a very basic and simple methodology the accuracy is not up to the mark. We only applied logistic regression once which can be improved to improve the accuracy and get a more precise model. Also including better machine learning models can help refine it even more. Further training this model using a larger dataset may help improve the accuracy and get faster results as well.

- **Only one dataset is used**
- **Poor accuracy**
- **More regression models can be included**
- **More machine learning models can be included**

Chapter 7
Applications of credit scoring model

- **Financial Institutions**
- **Banks**
- **Private lenders**
- **NBFC**
- **Borrower**
- **Businesses seeking to apply for loans/debts**

Chapter 8
Results and Discussion

We can see that using this basic modelling code we get an accuracy of around 73% and a precision of 73%. There are 5 categorical columns in the datasets. Majority of the data type is int however there are also two float type datas included. After calculating the Pearson's correlation matrix we found out that no feature in the dataset is highly correlated with the target attribute "status". Hence we directly performed logistic regression on all the features of the dataset. Histograms were used to plot all the features in the data set to get a graphical view of the data for better understanding. Now the dataset was split into Training and Testing datasets. A logistic regression model was applied to the training dataset and the summary of the results for the same was obtained. Now we remove extra features and significant features are identified. Status, time, age, amount, price, and savings are removed. Now the model is tested and accuracy is calculated. The overall accuracy of the model comes out to be 79% with a precision of 81%. Now we predict probability distribution plots for the defaults and non defaults to understand them better graphically. Next the ROC curve is plotted, which has an area of 84%. Now we try to find the optimal cutoff probability in two ways, one using the standard method and another time using Youden's index. Now optimal cutoff probability is identified using the cost of the loan. Finally the accuracy is calculated again now it is reduced but the quadrants that contribute to the cost are minimised.

Chapter 9
Conclusion

We have successfully modelled a credit scoring system and checked its accuracy with a test data set which came out to be 73%. This accuracy can be improved by using more than one regression algorithm and using decision tree modelling to further refine the score. Dataset used in this code included just 4446 rows of data, more such data can be collected to train this model to be more accurate with better precision. However this is a good starting point as the dataset is not too big and one can easily test and try many different algorithms on this dataset to generate a variety of results suitable for their applications. Insights can be generated and both the lender and the borrower can be empowered with this credit score. Big data technologies played a key role in handling this data and generating insights from it as well. Without the help of these technologies it would've been difficult to generate any meaningful insight from this huge amount of data.

References

- https://github.com/vikrantpailkar/Credit_Score_Analysis
- https://github.com/max-fitzpatrick/Credit-scoring-model
- https://www.myfico.com/credit-education/whats-in-your-credit-score
- https://www.forbes.com/sites/robertberger/2017/01/06/which-credit-score-do-lenders-actually-use/?sh=44c2fe1d30b1
- https://www.creditkarma.com/
- https://www.myfico.com/credit-education/whats-in-your-credit-score
- https://www.doughroller.net/credit/quizzle-vs-credit-karma-vs-credit-sesame/
- https://www.bajajfinserv.in/insights/everything-you-need-to-know-about-your-credit-score
- https://www.kaggle.com
- https://www.upgrad.com/blog/big-data-project-ideas-beginners/
- https://www.mdpi.com/1911-8074/13/8/180
- https://www.financialexpress.com
- https://link.springer.com/article/10.1007/s13748-021-00238-2
- https://www.nber.org/papers/w27671
- http://ejournal.uin-suka.ac.id/saintek/ijid/article/view/2384