

#### Vivekanand Education Society’s Institute of Technology

**Hashu Advani memorial Complex Collector’s Colony R C Marg, Chembur, Mumbai 400074**

#### DEPARTMENT OF INFORMATION TECHNOLOGY



##### MINI PROJECT REPORT ON

**"Loan Prediction"**

#### T.E. (Information Technology)

*SUBMITTED BY*

#### Mr. \_\_\_\_\_

*UNDER THE GUIDANCE OF*

#### PROF. Asha Bharambe

**(Academic Year: 2021-2022)**

#### Mumbai University

**Vivekanand Education Society’s Institute Of Technology, Mumbai**

#### DEPARTMENT OF INFORMATION TECHNOLOGY

****



****

***Certificate***

This is to certify that project entitled

#### ”Loan Prediction”

##### Mr/Miss. Your Name ( Roll No. xxx )

have satisfactorily carried out the project work, under the head - DS Using Python Lab at Semester VI of TE-IT in Information Technology as prescribed by the Mumbai University.

**Prof. Guide Name**

**Prof. Asha Bharambe**

**External Examiner**

**Dr.(Mrs.) Shalu Chopra**

**H.O.D**

**Dr.(Mrs.) J.M.Nair**

**Principal**

Date: 25/04/2022

Place: VESIT, Chembur

### LO Mapping

LO1: Understand the concept of Data science process and associated terminologies to solve real-world problems

LO2: Analyze the data using different statistical techniques and visualize the out- come using different types of plots.

LO3: Analyze and apply the supervised machine learning techniques like Classifi- cation, Regression or Support Vector Machine on data for building the models of data and solve the problems.

LO4: Apply the different unsupervised machine learning algorithms like Clustering or Association to solve the problems.

LO5: Design and Build an application that performs exploratory data analysis us- ing Apache Spark.

LO6: Design and develop a data science application that can have data acquisi- tion, processing, visualization and statistical analysis methods with supported machine learning technique to solve the real-world problem

### Declaration

I declare that this written submission represents my ideas in my own words and where other’s ideas or words have been included, I have adequately cited and ref- erenced the original source. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

- - - - - - - - - - -

**(Signature)**

Mr/Ms

**(Name of the Student and Roll No.)**

T.E. INFT

**Contents**

1. **Introduction 1**

[1.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1](#_gjdgxs)

[1.2 Literature Survey . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1](#_30j0zll)

[1.3 Problem Definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1](#_1fob9te)

[1.4 Objectives . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_3znysh7) 2

[1.5 Proposed Solution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_2et92p0) 2

[1.6 Technology Used . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_tyjcwt) 2

1. **Pre-processing 3**

[2.1 Dataset Description . . . . . . . . . . .. . . . . . . . . . . . . . . . . . .](#_3dy6vkm) 3

2.2 Handling Missing Data [. . . . . . . . . .. . . . . . . . . . . . . . . . . . .](#_3dy6vkm) 4

2.3 Handling Outliers [. . . . . . . . . . . .. . . . . . . . . . . . . . . . . . .](#_3dy6vkm) 4

2.4 Feature Scaling [. . . . . . . . . . . .. . . . . . . . . . . . . . . . . . .](#_3dy6vkm) 5

1. **EDA and Visualization 7**

[3.1 Measures of Central Tendency . . . . . . . . . . . . . . . . . . . . . . . .](#_1t3h5sf) 7

[3.2 Univariate Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_1t3h5sf) 7

[3.3 Bivariate Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_1t3h5sf) 9

[3.4 Correlation Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . .](#_1t3h5sf) 11

1. **Data Modeling 14**
2. **Hypothesis Testing 20**
3. **Result and analysis 21**
4. **CONCLUSION 23**

**List of Figures**

3.1 Distribution plot of LoanAmount . . . . . . . . . . . . . . . . . . . . . 8

3.2 Distribution plot of ApplicantIncome . . . . . . . . . . . . . . . . . . . . 8

3.3 Distribution plot of CoapplicantIncome . . . . . . . . . . . . . . . . . . . 8

3.4 Count plot of all attributes . . . . . . . . . . . . . . . . . . . . . . . . . . 9

3.5 Count plot of Education based on Approval of loan . . . . . . . . . . . . 10

3.6 Relation Between the Male or female Applicant's income, Loan taken and Self employment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

3.7 Boxplots for relation between Property Area, Amount of Loan and Education qualification . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

3.8 Relation between Credit History and Loan status . . . . . . . . . . . . . . 12

3.9 Heat Map showing correlation between all attributes . . . . . . . . . . . . 13

4.1 Representation of logistic regression . . . . . . . . . . . . . . . . . . . . 14

4.2 Representation of Random Forest Classification . . . . . . . . . . . . . . 15

4.3 Representation of KNN Classification . . . . . . . . . . . . . . . . . . . . 16

4.4 Representation of Naive Bayesian Classification. . . . . . . . . . . . . . . 16

4.5 Representation of SVM Classification . . . . . . . . . . . . . . . . . . . . 17

4.6 Representation of Gradient Boosting Classification . . . . . . . . . . . . . 18

6.1 Bar chart showing precedence of importance of features . . . . . . . . . . 21

8.1 Score of our submission . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

8.2 Rank of our submission . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

**List of Tables**

2.1 Data Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

3.1 Measures of Central Tendency of Dataset . . . . . . . . . . . . . . . . . 7

3.2 Relation between Gender and Approval of loan . . . . . . . . . . . . . . 9

3.3 Relation between Number of Dependants and Approval of loan . . . . . . 10

3.4 Relation between Credit history and Approval of loan . . . . . . . . . . . 10

3.5 Relation between Self employed and Approval of loan . . . . . . . . . . . 10

4.1 Summary of Data modeling . . . . . . . . . . . . . . . . . . . . . . . . . 18

**Abstract**

Housing Finance companies in the industry provide loans to their customers for buying homes. They can earn from the interest of those loans which they credit. By predicting the loan defaulters, the financial firm can reduce its Non-performing Assets and in the case of housing finance companies, they can use this to find their target demographic of customers who can readily acquire and pay loans for houses. This makes the study of this phenomenon very important. It is essential to study the nature of the different methods and their comparison. A very important approach in predictive analytics is used to study the problem of predicting loan defaulters (i) Collection of Data, (ii) Data Cleaning and (iii) Performance Evaluation.

**Keywords - *Loan Prediction, Big Data, Machine Learning, Logistic Regression, SVM, Decision Tree, Naïve Bayes, Random Forest***

# Chapter 1

# Introduction

## Introduction

Finance raising and lending for real estate, consumer, mortgage and companies‘ loans is the central part of almost every bank‘s business model. Lending money to inappropriate customers forms the major source of credit risk. The major share of the bank‘s assets comes directly from the profit derived from the bank‘s loans. The banking companies‘ face, however, a dual challenge to distinguish the possible deliberate defaulters from the applicants and the biased nature of few bank employees who have been at the instigation of developers of defaulting companies for many years. The primary goal of the banking community is to safely invest their capital. In the current scenario, many NBFCs and banks approve loans after a clear verification and authentication process, however, it remains uncertain whether the candidate selected is the worthy correct of all the applicants. In the case of housing finance companies, they can use this to find their target demographic of customers who can readily acquire and pay loans for houses. Through this method, we can predict whether or not that particular applicant is secure and the machine learning technique automates the entire process of authentication.

## Literature Survey

The main objective of the research paper [1] is to predict whether assigning the loan to a particular person will be safe or not. This paper is divided into four sections (i)Data Collection (ii) Comparison of machine learning models on collected data (iii) Training of system on most promising model (iv) Testing

The paper [2] Extending credits to corporates and individuals for the smooth functioning of growing economies like India is inevitable. As an increasing number of customers apply for loans in the banks and non- banking financial companies (NBFC), it is really challenging for banks and NBFCs with limited capital to devise a standard resolution and safe procedure to lend money to its borrowers for their financial needs. In addition, in recent times NBFC inventories have suffered a significant downfall in terms of the stock price. It has contributed to a contagion that has also spread to other financial stocks, adversely affecting the benchmark in recent times. In this paper, an attempt is made to condense the risk involved in selecting the suitable person who could repay the loan on time thereby keeping the bank’s non performing assets (NPA) on the hold. This is achieved by feeding the past records of the customer who acquired loans from the bank into a trained machine learning model which could yield an accurate result.

## Problem Definition

Link to the problem statement of the competition: [https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii](https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/#About)

Predict Loan Eligibility for Dream Housing Finance company:

Dream Housing Finance company deals in all kinds of home loans. They have presence across all urban, semi urban and rural areas. Customer first applies for a home loan and after that the company validates the customer eligibility for the loan.

Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application forms. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have provided a dataset to identify the customer segments that are eligible for loan amount so that they can specifically target these customers.

The *Business Goal* of this solution will be to process the data entered by users real time, and identify customer segments that are eligible for loan amount, so that they can specifically target these customers.

## Objectives

The first objective is to build a machine learning model that will predict the Loan Granted Status of a user with the highest accuracy. This will be done by building multiple ML models and comparing their performance.

The second objective is to help the housing finance companies with customer segmentation to find out which customers will successfully acquire and pay back loans, so that they can focus on marketing to these customers.

## Proposed Solution

First step would be to pre-process the data, separate the data set into Training and Testing data. After which we clean data by removing null and nan values, handling outliers. Then using data visualization techniques to look for the necessary factors that we need for prediction.

This is a Classification problem and can be solved by using any Classification Algorithm. Algorithms that we will use for predictions are

* Logistic Regression
* Random Forest
* K-Nearest Neighbors
* Gaussian Naive Bayes Classifier
* SVM Classifier
* Gradient Boosting Classifier

Then use the algorithm with highest accuracy

## Technology Used

Programming Language: **Python**

Modules Used: **pandas, numpy, seaborn, sklearn**

# Chapter 2

# Pre-processing

## 

## 2.1 Dataset Description

**Dataset**: Loan Prediction Dataset

**Data Description:**

The dataset contains details like Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount and Credit History of the customers for whom loan eligibility is known as 'Loan\_Status'.

| **Variable** | **Data Type** | **Description** |
| --- | --- | --- |
| Loan\_ID | object | Unique Loan ID |
| Gender | object | Male/ Female |
| Married | object | Applicant married (Y/N) |
| Dependents | object | Number of dependents |
| Education | object | Applicant Education (Graduate/ Under Graduate) |
| Self\_Employed | object | Self employed (Y/N) |
| ApplicantIncome | int64 | Applicant income |
| CoapplicantIncome | int64 | Coapplicant income |
| LoanAmount | float64 | Loan amount in thousands |
| Loan\_Amount\_Term | float64 | Term of loan in months |
| Credit\_History | float64 | credit history meets guidelines |
| Property\_Area | object | Urban/ Semi Urban/ Rural |
| Loan\_Status | object | (Target) Loan approved (Y/N) |

*Table 2.1 Data Description*

**Numerical Attributes:**

ApplicantIncome, CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, Credit\_History

**Categorical Attributes:**

Nominal Attributes: Gender, Married, Self\_Employed, Loan\_Status

Ordinal Attributes: Dependents, Education, Property\_Area

## 2.2 Handling Missing Data

**train.isnull().sum()**

Loan\_ID 0

Gender 13

Married 3

Dependents 15

Education 0

Self\_Employed 32

ApplicantIncome 0

CoapplicantIncome 0

LoanAmount 22

Loan\_Amount\_Term 14

Credit\_History 50

Property\_Area 0

Loan\_Status 0

**test.isnull().sum()**

Loan\_ID 0

Gender 11

Married 0

Dependents 10

Education 0

Self\_Employed 23

ApplicantIncome 0

CoapplicantIncome 0

LoanAmount 5

Loan\_Amount\_Term 6

Credit\_History 29

Property\_Area 0

Imputing the null values of numerical attributes using the median of all data in that attribute, because median is not affected by the presence of outliers in a dataset.

train.LoanAmount = train.LoanAmount.fillna(train.LoanAmount.median())

test.LoanAmount = test.LoanAmount.fillna(test.LoanAmount.median())

train.Loan\_Amount\_Term = train.Loan\_Amount\_Term.fillna(train.Loan\_Amount\_Term.median())

test.Loan\_Amount\_Term = test.Loan\_Amount\_Term.fillna(test.Loan\_Amount\_Term.median())

Imputing the null values of categorical attributes using the most occurring value of that column, that is, the mode of the data.

train.Gender = train.Gender.fillna(train.Gender.mode()[0])

test.Gender = test.Gender.fillna(test.Gender.mode()[0])

train.Married = train.Married.fillna(train.Married.mode()[0])

train.Dependents = train.Dependents.fillna(train.Dependents.mode()[0])

test.Dependents = test.Dependents.fillna(test.Dependents.mode()[0])

train.Self\_Employed = train.Self\_Employed.fillna(train.Self\_Employed.mode()[0])

test.Self\_Employed = test.Self\_Employed.fillna(test.Self\_Employed.mode()[0])

train.Credit\_History = train.Credit\_History.fillna(train.Credit\_History.mode()[0])

test.Credit\_History = test.Credit\_History.fillna(test.Credit\_History.mode()[0])

Now, there are no missing values left in the dataset.

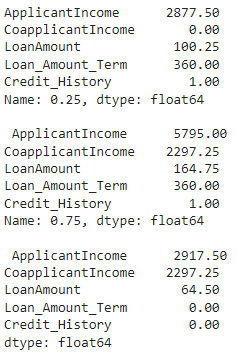
## 2.3 Handling Outliers

Q1 = train.quantile(0.25)

Q3 = train.quantile(0.75)

IQR = Q3 - Q1

print(Q1,"\n\n",Q3,"\n\n",IQR)



low\_lim = Q1 - 1.5 \* IQR

up\_lim = Q3 + 1.5 \* IQR

outlier = []

for x in train['LoanAmount']:

if ((x> up\_lim['LoanAmount']) or (x<low\_lim['LoanAmount'])):

outlier.append(x)

print(len(outlier))

Since there is a sizable number of outliers in the dataset, we do not drop them from the dataset.

## 2.4 Feature Scaling

For algorithms like Logistic regression, it is important to convert categorical attributes into numerical counterparts so that the classification becomes easier to calculate predictions using numbers.

**Converting categorical variable Gender(Male,Female) to numerical variables(0,1)**

sex = pd.get\_dummies(train['Gender'] , drop\_first = True )

train.drop(['Gender'], axis = 1 , inplace =True)

train = pd.concat([train , sex ] , axis = 1)

**Converting categorical variables Dependents(1,2,3+) to numerical variables(1,2,3)**

rpl = {'0':'0', '1':'1', '2':'2', '3+':'3'}

train.Dependents = train.Dependents.replace(rpl).astype(int)

test.Dependents = test.Dependents.replace(rpl).astype(int)

**Converting categorical variable Self\_Employed(Yes,No) to numerical variables(1,0)**

self\_Employed = pd.get\_dummies(train['Self\_Employed'] ,prefix = 'employed' ,drop\_first = True )

train.drop(['Self\_Employed'], axis = 1 , inplace =True)

train = pd.concat([train , self\_Employed ] , axis = 1)

self\_Employed = pd.get\_dummies(test['Self\_Employed'] , prefix = 'employed' ,drop\_first = True )

test.drop(['Self\_Employed'], axis = 1 , inplace =True)

test = pd.concat([test , self\_Employed ] , axis = 1)

**Converting categorical variable Married(Yes,No) to numerical variables(1,0)**

married = pd.get\_dummies(train['Married'] , prefix = 'married',drop\_first = True )

train.drop(['Married'], axis = 1 , inplace =True)

train = pd.concat([train , married ] , axis = 1)

married = pd.get\_dummies(test['Married'] , prefix = 'married', drop\_first = True )

test.drop(['Married'], axis = 1 , inplace =True)

test = pd.concat([test , married ] , axis = 1)

**Converting categorical variable Education to numerical variables**

train['Education'] = train['Education'].map( {'Graduate': 0, 'Not Graduate': 1} ).astype(int)

test['Education'] = test['Education'].map( {'Graduate': 0, 'Not Graduate': 1} ).astype(int)

**Converting Categorical Property\_Area to Numerical**

train['Property\_Area'] = train['Property\_Area'].map( {'Urban': 0, 'Semiurban': 1 ,'Rural': 2 } ).astype(int)

test['Property\_Area'] = test['Property\_Area'].map( {'Urban': 0, 'Semiurban': 1 ,'Rural': 2 } ).astype(int)

**Converting categorical target Loan Status variables to numerical variables**

train['Loan\_Status'] = train['Loan\_Status'].map( {'N': 0, 'Y': 1 } ).astype(int)

# Chapter 3

# EDA and Visualization

## Measures of Central Tendency

|  | **ApplicantIncome** | **CoapplicantIncome** | **LoanAmount** | **Loan\_Amount\_Term** | **Credit\_History** |
| --- | --- | --- | --- | --- | --- |
| **count** | 614.000000 | 614.000000 | 614.000000 | 614.000000 | 614.000000 |
| **mean** | 5403.459283 | 1621.245798 | 145.752443 | 342.410423 | 0.855049 |
| **std** | 6109.041673 | 2926.248369 | 84.107233 | 64.428629 | 0.352339 |
| **min** | 150.000000 | 0.000000 | 9.000000 | 12.000000 | 0.000000 |
| **25%** | 2877.500000 | 0.000000 | 100.250000 | 360.000000 | 1.000000 |
| **50%** | 3812.500000 | 1188.500000 | 128.000000 | 360.000000 | 1.000000 |
| **75%** | 5795.000000 | 2297.250000 | 164.750000 | 360.000000 | 1.000000 |
| **max** | 81000.000000 | 41667.000000 | 700.000000 | 480.000000 | 1.000000 |

*Table 3.1 Measures of Central Tendency of Dataset*

Skewness:

ApplicantIncome 6.539513

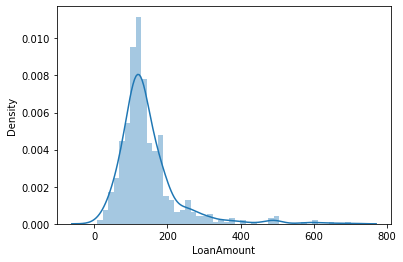
CoapplicantIncome 7.491531

LoanAmount 2.743053

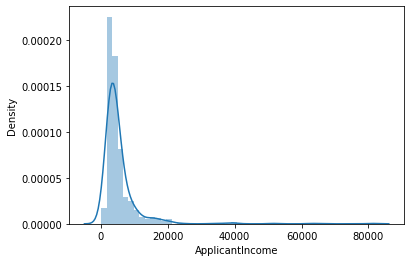
Loan\_Amount\_Term -2.402112

Credit\_History -2.021971

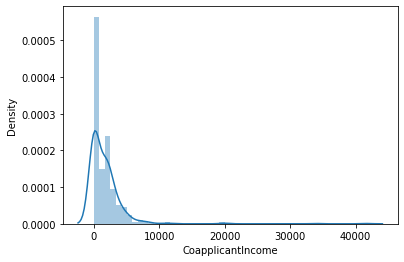
## Univariate Analysis



*Fig. 3.1 Distribution plot of LoanAmount*

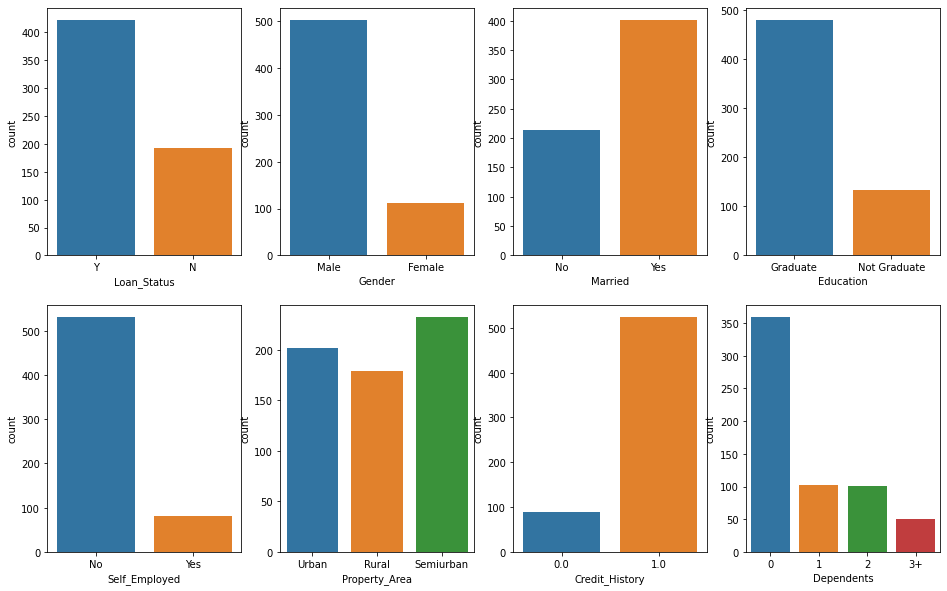


*Fig. 3.2 Distribution plot of ApplicantIncome*



*Fig. 3.3 Distribution plot of CoapplicantIncome*

This shows that the given data is right skewed for LoanAmount, ApplicantIncome and CoapplicantIncome.



*Fig. 3.4 Count plot of all attributes*

These graphs show the categorical value distribution of the variables in the dataset.

## Bivariate Analysis

**Relationship between Gender and Approval of loan**

| **Loan\_Status** | **N** | **Y** | **All** |
| --- | --- | --- | --- |
| **Gender** |  |  |  |
| **Female** | 37 | 75 | 112 |
| **Male** | 155 | 347 | 502 |
| **All** | 192 | 422 | 614 |

*Table 3.2 Relation between Gender and Approval of loan*

Approval of loan based on gender:

* Males: 69.325%
* females: 66.964%

Inference : Male Applicants are significantly more than female applicants

**Relationship between Number of Dependants and Approval of loan**

| **Loan\_Status** | **N** | **Y** | **All** |
| --- | --- | --- | --- |
| **Dependents** |  |  |  |
| **0** | 113 | 247 | 360 |
| **1** | 36 | 66 | 102 |
| **2** | 25 | 76 | 101 |
| **3+** | 18 | 33 | 51 |
| **All** | 192 | 422 | 614 |

*Table 3.3 Relation between Number of Dependants and Approval of loan*

Inference : The applicants with the highest number of dependents are least in number whereas applicants with no dependents are greatest among these.

**Relationship between Credit history of customer and Approval of loan**

| **Loan\_Status** | **N** | **Y** | **All** |
| --- | --- | --- | --- |
| **Credit\_History** |  |  |  |
| **0.0** | 82 | 7 | 89 |
| **1.0** | 110 | 415 | 525 |
| **All** | 192 | 422 | 614 |

*Table 3.4 Relation between Credit history and Approval of loan*

Approval of load based on customer’s credit history:

* Without credit history: 7.865%
* With credit history: 68.729%  
  It is clearly seen that customer with previous credit history is more likely to get the loan approved

**Relationship between Self employed customer and Approval of loan**

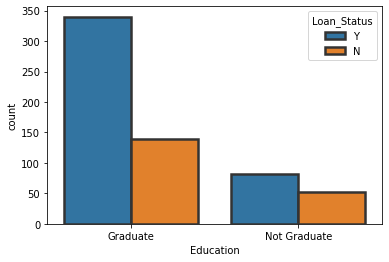
| **Loan\_Status** | **N** | **Y** | **All** |
| --- | --- | --- | --- |
| **Self\_Employed** |  |  |  |
| **No** | 166 | 366 | 532 |
| **Yes** | 26 | 56 | 82 |
| **All** | 192 | 422 | 614 |

*Table 3.5 Relation between Self employed and Approval of loan*

Approval of load based on customer’s employment status:

* Not Self employed: 68.292%
* Self employed: 68.729%

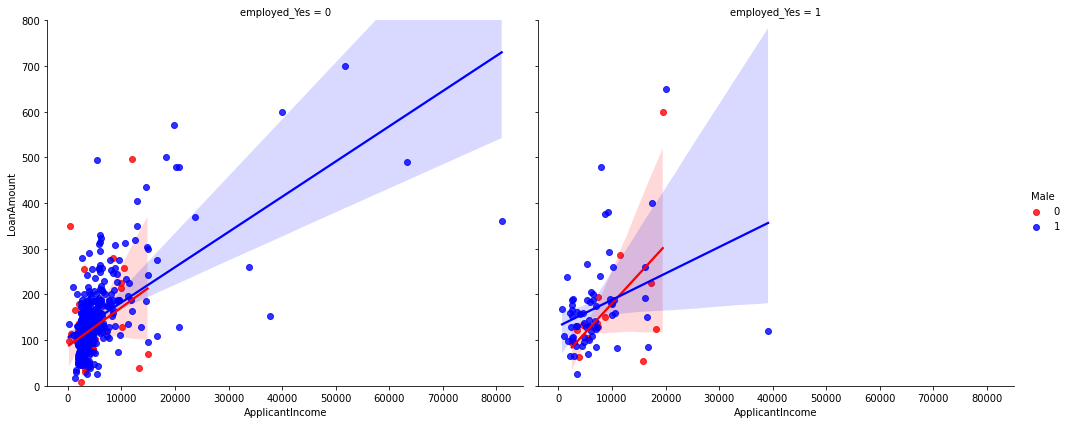
**Relationship between Education and Approval of loan**



*Fig. 3.5 Count plot of Education based on Approval of loan*

## Correlation Analysis

**Relation Between the Male or female Applicant's income , Loan taken and Self employment:**

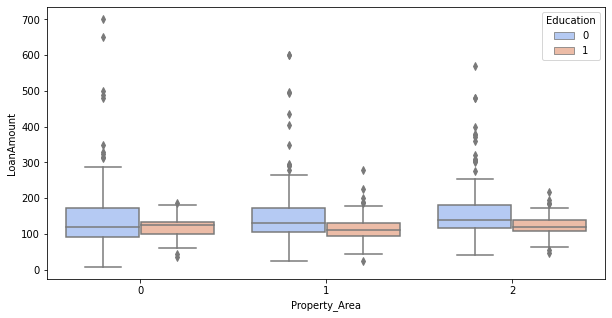


*Fig. 3.6 Relation Between the Male or female Applicant's income , Loan taken and Self employment*

Conclusions from Above Graph

1. The male applicants take more loans than females.
2. The males are higher in the "NOT self employed" category.
3. The amount is still larger in the income range (0 to 20000).
4. Also we observe that the majority of applicants are NOT self employed.
5. Highest Loan amount taken is by the female applicant of about 700 which is NOT self employed.
6. The majority of income taken is about 0-200 with income in the range 0-20000.
7. The line plot shows that with increase in income the amount of loan increases with almost same slope for the case of women in both the cases but a slightly lesser slope in the case of men in Self- Employed category as compared to non-self employed.

**Boxplots for relation between Property Area, Amount of Loan and Education qualification:**

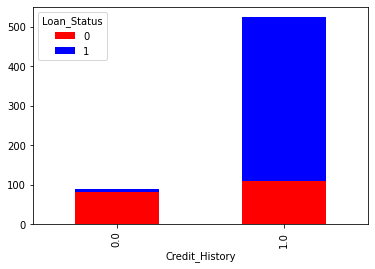


*Fig. 3.7 Boxplots for relation between Property Area, Amount of Loan and Education qualification*

Conclusions from the box plot

1. In the Urban area the non graduates take slightly more loan than graduates.
2. In the Rural and semi urban area the graduates take more amount of Loan than non graduates
3. The higher values of Loan are mostly from Urban area
4. The semi urban area and rural area both have one unusual Loan amount close to zero.

**Relation between Credit History and Loan status:**

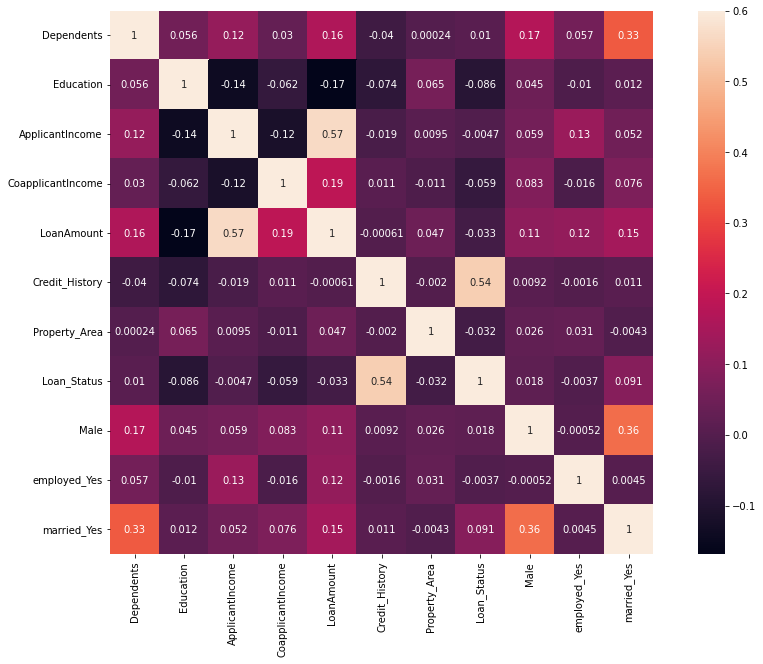


*Fig. 3.8 Relation between Credit History and Loan status*

The credit history vs Loan Status indicates:

1. The good credit history applicants have more chances of getting Loan.
2. With better credit History the Loan amount given was greater too.
3. But many were not given loan in the range 0-100
4. The applicants with poor credit history were handled in the range 0-100 only.

**Correlation Heat Map:**



*Fig. 3.9 Heat Map showing correlation between all attributes*

High positive correlation between:

* LoanAmount and ApplicantIncome
* Credit\_History and Loan\_Status

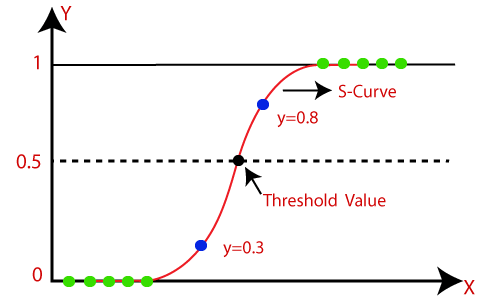
# Chapter 4

# Data Modeling

Problem of predicting if the loan claim will be approved or not, is a Classification problem. Hence we try different Classification algorithms and compare their performances to choose the best one for our dataset.

**Logistic Regression**

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.



*Fig. 4.1 Representation of logistic regression*

Code:

from sklearn.linear\_model import LogisticRegression

logmodel = LogisticRegression()

logmodel.fit(X\_train , y\_train)

pred\_l = logmodel.predict(X\_test)

acc\_l = accuracy\_score(y\_test , pred\_l)\*100

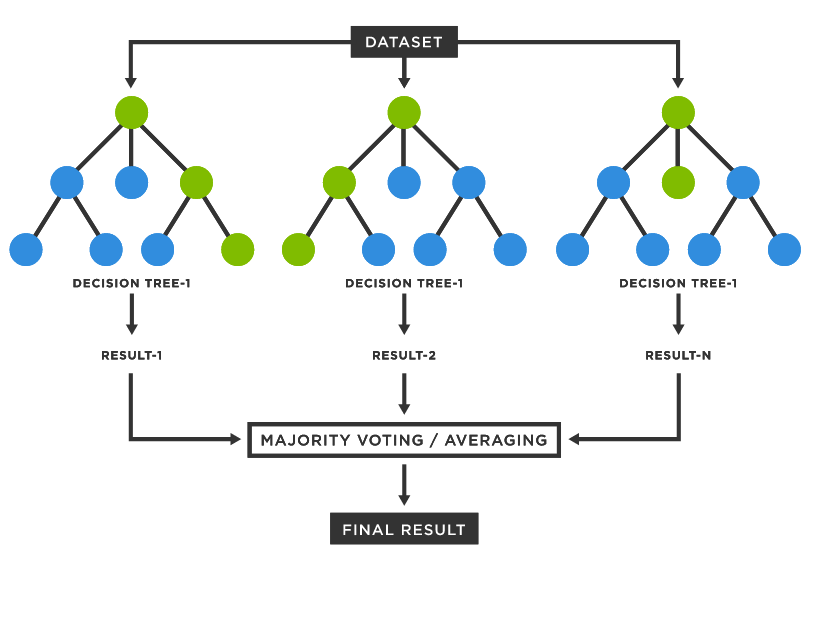
acc\_l

Output:

80.0

**Random Forest**

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

****

*Fig. 4.2 Representation of Random Forest Classification*

Code:

random\_forest = RandomForestClassifier(n\_estimators= 100)

random\_forest.fit(X\_train, y\_train)

pred\_rf = random\_forest.predict(X\_test)

acc\_rf = accuracy\_score(y\_test , pred\_rf)\*100

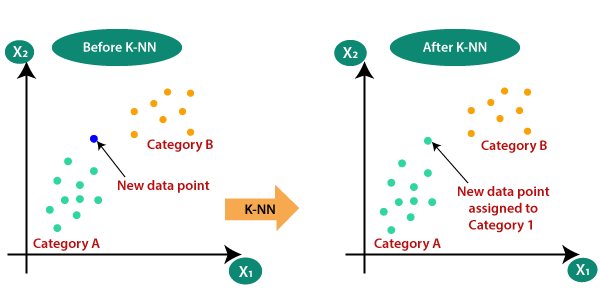
acc\_rf

Output:

76.75675675675676

**K-Nearest Neighbors**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.

****

*Fig. 4.3 Representation of KNN Classification*

Code:

knn = KNeighborsClassifier(n\_neighbors = 3)

knn.fit(X\_train, y\_train)

pred\_knn = knn.predict(X\_test)

acc\_knn = accuracy\_score(y\_test , pred\_knn)\*100

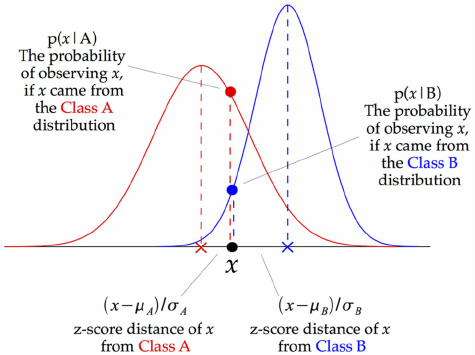
acc\_knn

Output:

61.08108108108108

**Gaussian Naive Bayes Classifier**

Gaussian Naive Bayes is an extension of naive Bayes. Other functions can be used to estimate the distribution of the data, but the Gaussian (or Normal distribution) is the easiest to work with because you only need to estimate the mean and the standard deviation from your training data.



*Fig. 4.4 Representation of Naive Bayesian Classification*

Code:

gaussian = GaussianNB()

gaussian.fit(X\_train, y\_train)

pred\_gb = gaussian.predict(X\_test)

acc\_gb = accuracy\_score(y\_test , pred\_gb)\*100

acc\_gb

Output:

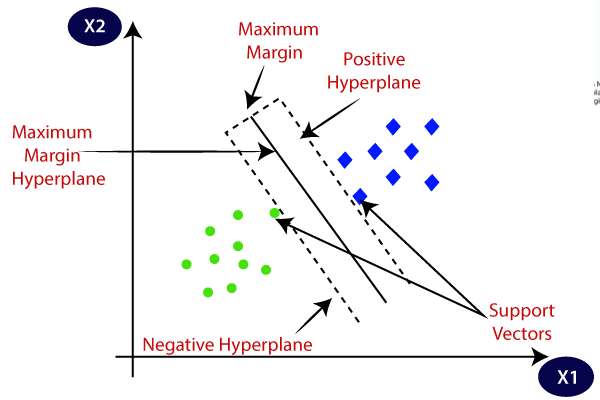
78.91891891891892

**SVM Classifier**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



*Fig. 4.5 Representation of SVM Classification*

Code:

svc = SVC()

svc.fit(X\_train, y\_train)

pred\_svm = svc.predict(X\_test)

acc\_svm = accuracy\_score(y\_test , pred\_svm)\*100

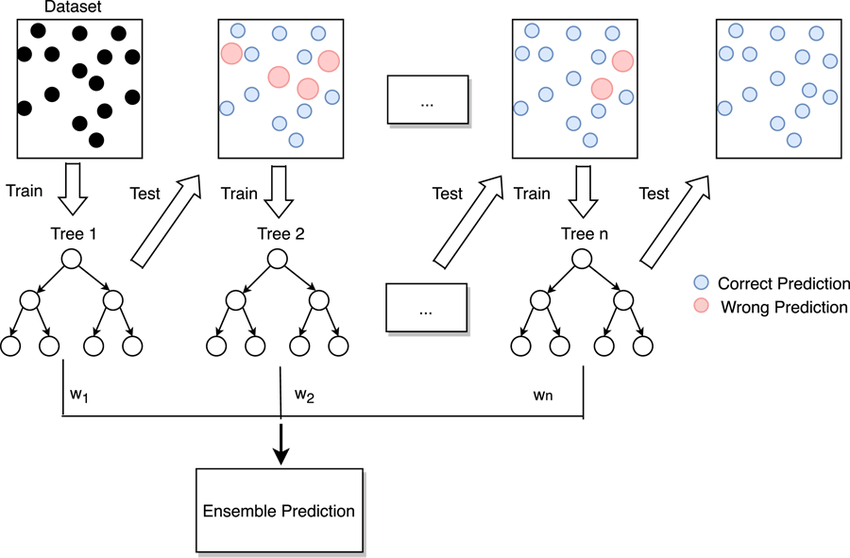
acc\_svm

Output:

71.89189189189189

**Gradient Boosting Classifier**

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets



*Fig. 4.6 Representation of Gradient Boosting Classification*

Code:

gbk = GradientBoostingClassifier()

gbk.fit(X\_train, y\_train)

pred\_gbc = gbk.predict(X\_test)

acc\_gbc = accuracy\_score(y\_test , pred\_gbc)\*100

acc\_gbc

Output:

77.83783783783784

**Summary of Data modeling**

| **Model** | **Score** |
| --- | --- |
| Logistic Regression | 80.000000 |
| Gaussian Naive Bayes Classifier | 78.918919 |
| Gradient Boosting Classifier | 77.837838 |
| Random Forest | 76.756757 |
| SVM | 71.891892 |
| K- Nearest Neighbor | 61.081081 |

*Table 4.1 Summary of Data modeling*

The Highest Accuracy among Classifiers is shown by Logistic Regression => **80.00%**

# Chapter 5

# Hypothesis Testing

Machine learning models are chosen based on their mean performance. The algorithm with the best mean performance is expected to be better than those algorithms with worse mean performance. But what if the difference in the mean performance is caused by a statistical fluke? The solution is to use a statistical hypothesis test to evaluate whether the difference in the mean performance between any two algorithms is real or not.

The number of folds and repeats of the Paired Student's T-Test can be configured to achieve a good sampling of model performance that generalizes well to a wide range of problems and algorithms. Specifically two-fold cross-validation with five repeats, so-called 5×2-fold cross-validation. The MLxtend library by Sebastian Raschka provides an implementation via the paired\_ttest\_5x2cv() function.

Null Hypothesis: There is no difference between performance of Logistic Regression and Naive Bayes Classifier

Alternate Hypothesis: There is significant difference between performance of Logistic Regression and Naive Bayes Classifier

Hence, H0 : d = 0

H1 : d != 0

from mlxtend.evaluate import paired\_ttest\_5x2cv

t, p = paired\_ttest\_5x2cv(estimator1=logmodel, estimator2=gaussian, X=X, y=y, scoring='accuracy', random\_seed=1)

print('P-value: %.3f, t-Statistic: %.3f' % (p, t))

**P-value: 0.731, t-Statistic: -0.364**

if p <= 0.05:

print('There is significant difference between performance of Logistic Regression and Naive Bayes Classifier')

else:

print('There is no difference between performance of Logistic Regression and Naive Bayes Classifier')

**There is no difference between performance of Logistic Regression and Naive Bayes Classifier**

As there is no significant difference between performance of Logistic Regression and Naive Bayes Classifier, we can choose any one of these models to proceed with our classification. We choose the Logistic Regression model to proceed.

# Chapter 6

**Result and Analysis**

**Performance metrics for the model using Logistic Regression:**

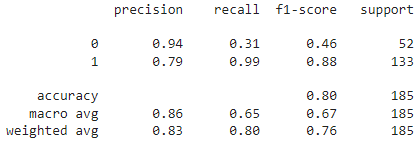
Accuracy : 0.8

Confusion Matrix:

**[[ 16 36]**

**[ 1 132]]**

Classification Report:

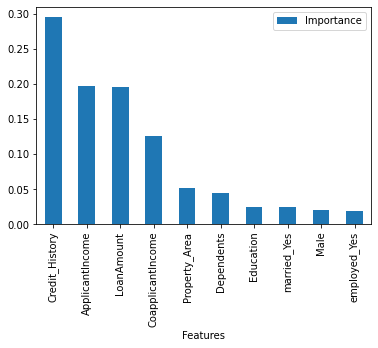


**Features important for the classification:**

importances = pd.DataFrame({'Features':X\_train.columns,'Importance':np.round(random\_forest.feature\_importances\_,3)})

importances = importances.sort\_values('Importance',ascending=False).set\_index('Features')

importances.plot.bar()



*Fig. 6.1 Bar chart showing precedence of importance of features*

Credit History has maximum importance and Employment has the least.

Inference:

The Loan status has better relation with features such as Credit History, Applicant's Income, Loan Amount needed by them, Family status(Dependents) and Property Area which are generally considered by the loan providing organizations. These factors are hence used to make correct decisions to provide loan status or not. This data analysis hence gives a realization of features and the relation between them from the older decision examples hence giving a learning to predict the class of the unseen data.

**Prediction over unseen dataset using Logistic Regression for final submission:**

df\_test = test.drop("Loan\_ID",axis=1)

df\_test.head()

p\_log = logmodel.predict(df\_test)

predict\_combine = np.zeros((df\_test.shape[0]))

for i in range(0, test.shape[0]):

temp = p\_log[i]

if temp==1:

predict\_combine[i] = 1

predict\_combine = predict\_combine.astype('int')

predict\_combine = ['Y' if p == 1 else 'N' for p in predict\_combine]

final\_submission = pd.DataFrame({

"Loan\_ID": test["Loan\_ID"],

"Loan\_Status": predict\_combine

})

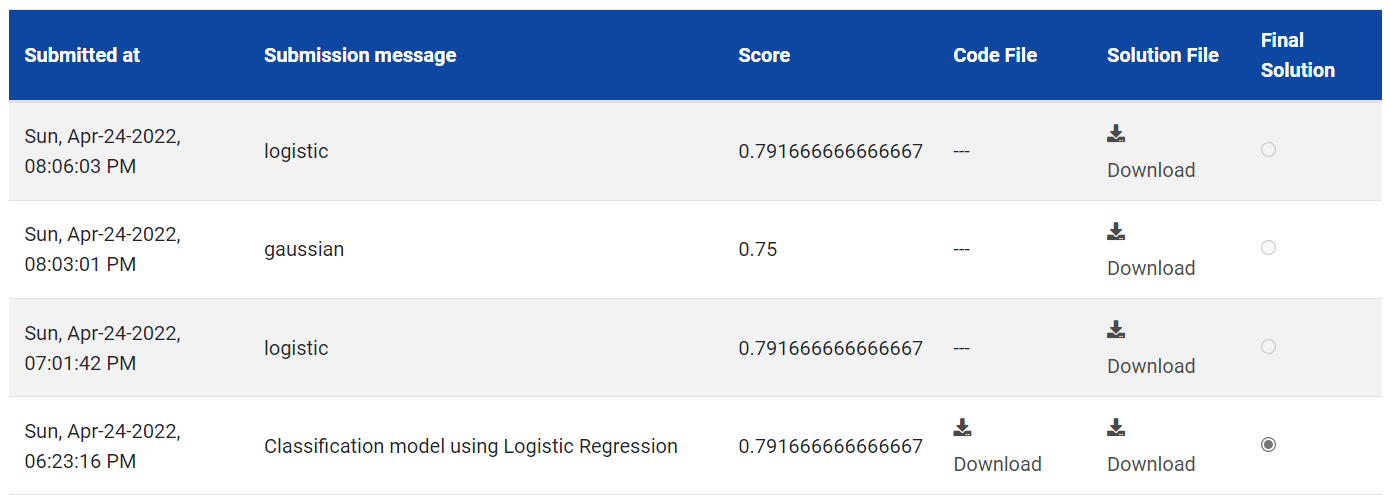
final\_submission.to\_csv("logistic\_regression\_results.csv", encoding='utf-8', index=False)

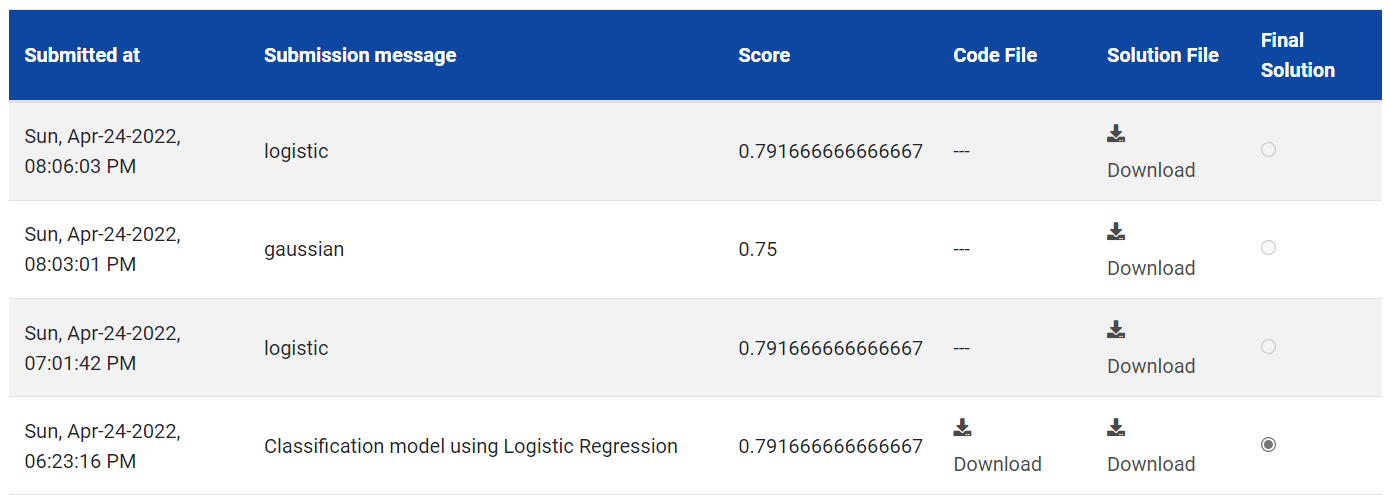
# Chapter 8

# CONCLUSION

Thus, we have created a Classification ML Model to predict if a user is eligible for a loan, based on the details provided, using the Logistic Regression Classifier Algorithm.

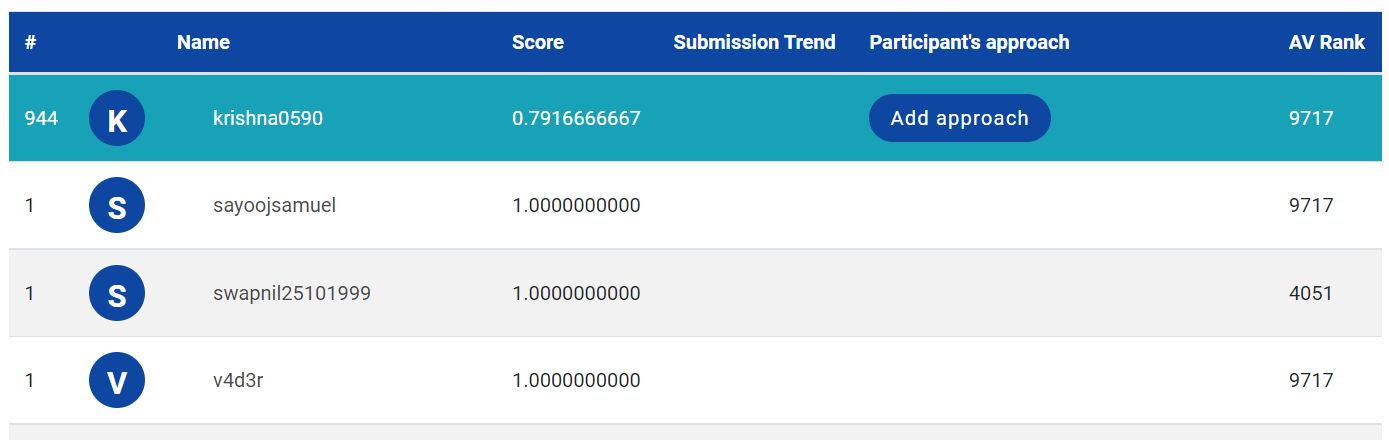
Score received on submission: **0.791666666666667**





*Fig. 8.1 Score of our submission*

Overall ranking of the submission: **944**



*Fig. 8.2 Rank of our submission*

# References

* + 1. Kumar Arun, Garg Ishan, Kaur Sanmeet, ― Loan Approval Prediction based on Machine Learning Approach‖, IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 18, Issue 3, pp. 79-81, Ver. I (May-Jun. 2016)
    2. E. Chandra Blessie, R. Rekha, ― Exploring the Machine Learning Algorithm for Prediction the Loan Sanctioning Process (2019)
    3. M. A. Sheikh, A. K. Goel and T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 490-494, doi: 10.1109/ICESC48915.2020.9155614.