

# **Analysis of Customer for Financial Loan using Machine Learning**

Minor project report submitted in partial fulfilment of the requirement  
for the degree of Bachelor of Technology

in

**Information Technology**

By

BHAVYA GUPTA (191543)

KALPIT BANSAL (191537)

**UNDER THE SUPERVISION OF**

MR. PRAVEEN MODI



Department of Computer Science & Engineering and Information  
Technology

**Jaypee University of Information Technology, Wagnaghat,  
173234, Himachal Pradesh, INDIA**

## TABLE OF CONTENT

<b>Title</b>	<b>Page No.</b>
<b>Declaration</b>	<b>I</b>
<b>Certificate</b>	<b>II</b>
<b>Acknowledgement</b>	<b>III</b>
<b>Abstract</b>	<b>IV</b>
<b>Chapter-1 (Introduction)</b>	<b>1-2</b>
<b>Chapter-2 (Feasibility Study, Requirements Analysis and Design)</b>	<b>3-5</b>
<b>Chapter-3 (Implementation)</b>	<b>6-19</b>
<b>Chapter-4 (Results)</b>	<b>20-21</b>
<b>References</b>	<b>22</b>

# DECLARATION

I hereby declare that, this project has been done by me under the supervision of Mr. **Praveen Modi, Asst. Prof. in CSE & IT**, Jaypee University of Information Technology. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

## **Supervised by:**

**Mr. Praveen Modi**

Asst. Prof. (Grade-I)

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology

## **Submitted by:**

**Bhavya Gupta (191543)**

**Kalpita Bansal (191537)**

Computer Science & Engineering Department

Jaypee University of Information Technology

## **CERTIFICATE**

This is to certify that the work which is being presented in the project report titled **“Analysis of Customer for Financial Loan using Machine Learning”** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by “Bhavya Gupta – 191543, Kalpit Bansal – 191537” during the period from January 2022 to May 2022 under the supervision of Mr. Praveen Modi, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Bhavya Gupta (191543)

Kalpita Bansal (191537)

The above statement made is correct to the best of my knowledge.

Mr. Praveen Modi

Asst. Prof. (Grade-I)

Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat,

## ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I really grateful and wish my profound my indebtedness to Supervisor **Mr. Praveen Modi, Asst. Prof. (Grade-I)**, Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of “**Machine Learning**” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Mr. Praveen Modi**, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Bhavya Gupta (191543)

Kalpita Bansal (191537)

## ABSTRACT

Banks are making a major part of profits through loans. Though a lot of people are applying for loans, so it's hard to select the genuine applicant, who will repay the loan. While doing the process manually, a lot of misconception may happen to select the genuine applicant. We are creating a machine learning model which will help the finance companies to distinguish between persons in various categories like according to the age - youth, middle-age or old, their marital status i.e., whether they are single or married, or according to their education whether they are educated or not etc. So, there are various aspects into which the finance company look into before automating the loan approval process. So, the user who wants to apply for a loan fills an online form and then the company looks into the following aspects and then decides whether the person is eligible for the loan or not.

So basically, we will develop a machine learning system for such a problem which will help companies to decide whether a particular person is eligible for the loan or not. In this we have used machine learning algorithms used for solving regression problems i.e., in which the output variables are categorical type. We had split our data in 80 – 20 ratio that is 80% training data and 20% test data, after which we scaled our dataset using Standard Scaler for better analysis as our dataset had different variables with different range. While preprocessing, we used Label Encoder so that the categorical values in our dataset gets changed to 0 or 1 for our machine to understand. We have used mainly four algorithms in our model i.e., Logistic Regression, Random Forest, Decision Tree and Naïve Bayes algorithm and have kept random state=0 so that our result /accuracy does not gets changed in every cycle and remains same every time we run our dataset, then we found accuracy using each of them. We found that the accuracy using Decision tree was somewhat 61.78%, using Naïve Bayes, it was 83.73%, by using Logistic Regression it was around 82.92% and by using Random Forest it was 78.86%. Hence according to our result and dataset, we can conclude that Naïve Bayes is the best algorithm because of its highest accuracy among others, to predict Loan Status of customers applying for loans in banks.

# **Chapter 01:**

## **INTRODUCTION**

### **1.1 Introduction-**

The core business of banks is to provide loans to people as the main profit comes directly from the loan's interest. The loan companies grant a loan after an intensive process of verification and validation. However, they still don't have surety whether the person would be able to pay the loan amount or not. Though a lot of people are applying for loans, so it's hard to select the genuine applicant, who will repay the loan. By predicting the loan defaulters, the bank can reduce its Non- Performing Assets. While doing the process manually, a lot of misconception may happen to select the genuine applicant.

Machine Learning is a subclass of Artificial Intelligence (AI) that majorly focuses on developing applications that input the data learned from it and improve their performance without being programmed to do so. The Machine Learning algorithms used to train the model use the technique of pattern and feature recognition from the profusely available data. Using this data, the predictions and decisions are made. The accuracy rates can be elevated and the performance can be enhanced by using some of the best algorithms and training them well. Hence, we are creating a machine learning model which will help the finance companies to distinguish between persons in various categories like according to the age - youth, middle-age or old, their marital status i.e., whether they are single or married, or according to their education whether they are educated or not etc. But as the right predictions are very important for the maximization of profits, it is essential to study the nature of the different methods and their comparison. So basically, we will develop a machine learning system for such a problem which will help companies to decide whether a particular person is eligible for the loan or not. We have used four techniques to predict the accuracy of our predictive model – Logistic regression, Random Forest, Naïve bayes and Decision tree.

## **1.2 Objective –**

Analysing the customer on the basis of various parameters like age, property etc to check whether he is eligible for loan or not which can reduce the risk and minimize the no. of defaulters.

## **1.3 Motivation –**

Due to the increase in number of frauds in today's time, it's very difficult to judge who is genuine and who is fraud. Bank frauds are more common in these times and hence building a model which could help in finding loan defaulters will be of great profitability for banks. Machine Learning is a subclass of Artificial Intelligence (AI) that recognizes some hidden patterns in the given data to construct different models. Our system uses various machine learning algorithms to predict whether a person would be eligible for loan or not. Hence, this project is carried out with the major motivation of analyzing several algorithms in order to further refine the accuracy of the model.

## **1.4 Language Used –**

Python 3.8

## **1.5 Technical Requirements -**

### **(Hardware) –**

Python IDLE and minimum of 8GB RAM along with CPU which must be a minimum of 6th generation (Intel Core i5 processor).

### **(Software) –**

We have used Google colab for the implementation of our project and python libraries like sklearn, NumPy, matplotlib and pandas.



## **1.6 Deliverables/Outcomes-**

Model building through Machine Learning and Data Mining is a complex task. Considering a large dataset and numerous features, parameters of the training algorithms and testing it takes a while and a lot of effort to build a model. After building the model we will be able to predict that a person would be eligible for the loan or not. This will be helpful in checking for loan defaulters and can help the banks in recognizing between genuine and fraud persons.

# **Chapter 02:**

## **Feasibility Study, Requirements Analysis and Design**

### **2.1 Feasibility Study**

#### **• 2.1.1 Problem Definition –**

In today's time, the expenses and needs of people have increased which makes it quite difficult for a middle-class family to live a comfortable life, around 77% of people takes loan from the bank either for education or house finance or their personal reasons to make end meet. Therefore, due to the increase in number of people taking loans, it has become important for banks to check who is a genuine person and who is fraud.

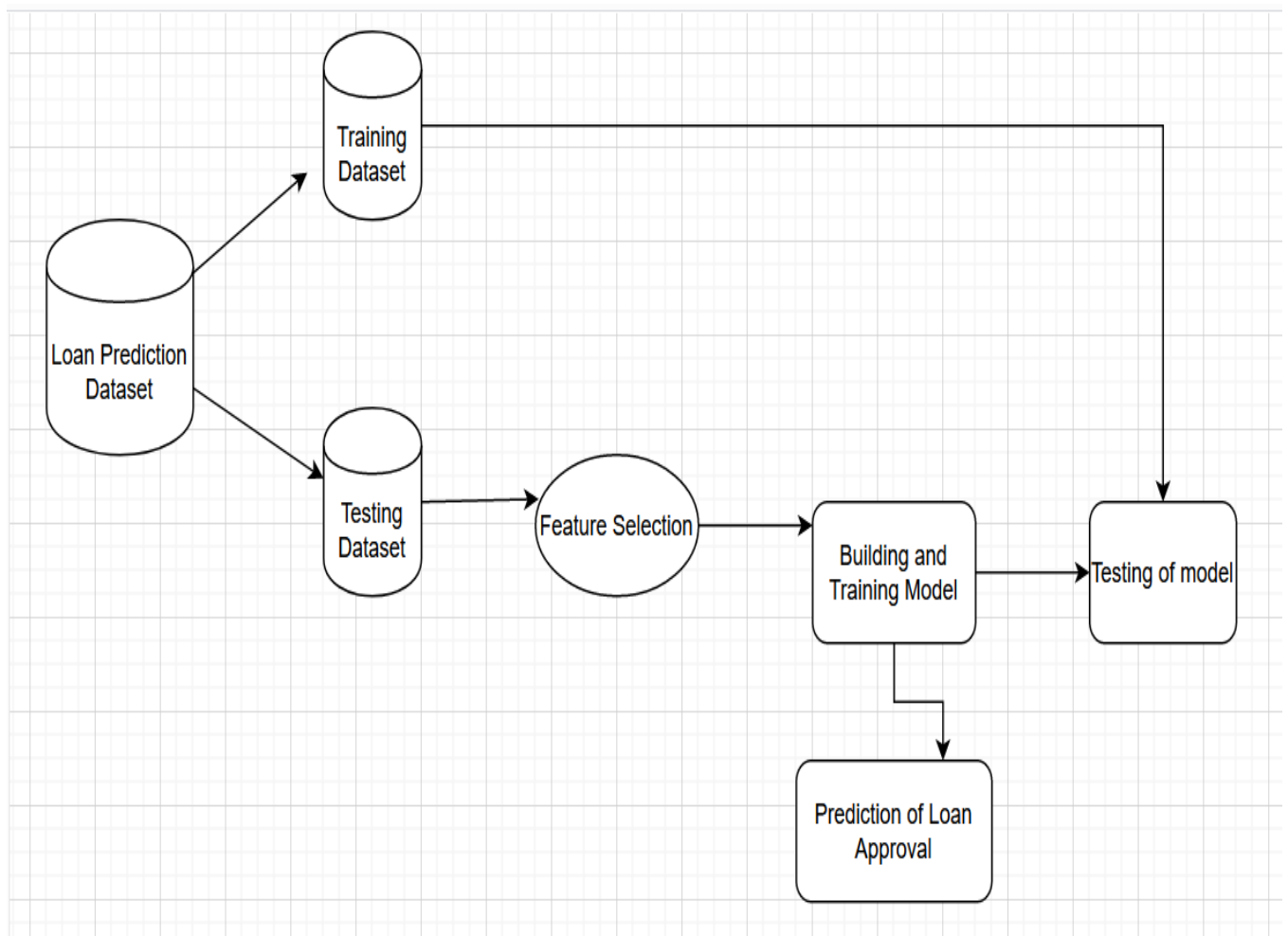
#### **• 2.1.2 Problem Analysis –**

Analyzing the customers for financial loan refers to analyzing the customers on some criteria basis which a financial institution evaluates to decide the eligibility of a customer for a particular loan i.e., whether he is eligible for loan or not.

#### **• 2.1.3 Solution –**

We would build a model using Python libraries and machine learning algorithms like Logistic Regression, Decision Tree, Random Forest and Naïve bayes to after splitting and training our dataset to predict whether a person would be eligible for loan or not. The experiments with the Loan Prediction dataset have concentrated on distinguishing person eligible for loan with value 1 from the person who is not eligible having value 0.

## 2.3 Data-Flow Diagram (DFD)



**Figure – 1: Data Flow diagram**

**Table:1- Literature Survey**

<b>TITLE</b>	<b>AUTHOR &amp; YEAR</b>	<b>DATASETS</b>	<b>TECHNIQUES</b>	<b>RESULT</b>
Survey on Ensemble model for loan prediction.	Aachal Goel and Ranpreet Kaur [Jan- Feb, 2016]	Used Mackey dataset, Sunspot's dataset and Stock Price	Bagging, Boosting, Stacking, Adaboost	In this we compare several models and choose the best, it enhances performance and accuracy.
Factors that affect loan giving decisions of banks	Rhishab Mukherjee Et al. [April, 2021]	Loan Status dataset with 13 features in total.	Logistic Regression, Decision trees, Random Forest, XGBoost	In this a model built with two independent variables got the highest accuracy of 83%.
Analysis of loan availability using ML techniques	Sharayu Dosalwar Et al. [Sept, 2021]	Data from previous customers of various banks who had loans approved.	Logistic Regression, Decision trees, Random Forest, XGBoost, KNN, SVM, Naive Bayes	It was seen that Logistic Regression gave best accuracy of 78.5%.

# Chapter 03: IMPLEMENTATION

## 3.1 Data Set Used in the Minor Project –

Loan Prediction from Kaggle having 614 rows and 13 columns. The main goal of using this dataset is to find whether a person is eligible for loan or not. This dataset also consists of missing values so that it is needed to be pre-processed for the prediction. The experiments with the Loan Prediction dataset have concentrated on distinguishing person eligible for loan with value 1 from the person who is not eligible having value 0.

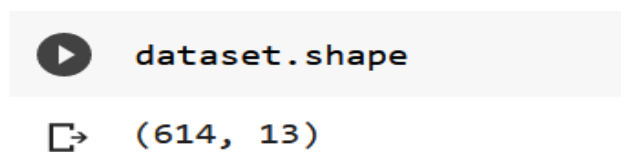
## 3.2 Data Set Features –

### 3.2.1 Types of Data

Attributes	Type
Loan_Id	object
Gender	object
Married	object
Dependents	object
Education	object
Self_Employed	object
ApplicantIncome	int64
CoapplicantIncome	float64
LoanAmount	float64
Loan_Amount_Term	float64
Credit_History	float64
Property_Area	object
Loan_Status	object

- **Loan ID:** This attribute defines the individual unique id of a person.
- **Gender:** This attribute shows the gender of the person using the below given format: Male and Female
- **Married:** This attribute talks about the marital status of a person using the below given format: No- not married  
Yes- married
- **Dependents:** Number of dependents i.e., 0,1,2 of a person
- **Education:** Applicant Education (Graduate/ Not Graduate)
- **Self Employed:** Self-employed (Yes/No)
- **Applicant Income:** Income of applicant
- **Co-Applicant Income:** Income of co-applicant
- **Loan Amount:** Amount of loan one wants to take in thousands
- **Loan Amount Term:** period of loan in months
- **Credit History:** in format 1 or 0
- **Property Area:** property area defined by urban, semi urban or rural
- **Loan Status:** Whether loan is approved or not ,1- approved and 0- not approved.

### 3.2.2 Number of Attributes, fields, description of the dataset –



```
dataset.shape
```

```
(614, 13)
```

**Figure-3: Number of attributes in dataset**

✓  
0s

```
[6] dataset.describe()
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

**Figure-4: Description of dataset**

### 3.3 Design of Problem Statement-

The aim of this project is to get the most significant results for predicting whether a person is eligible for the loan or not using various Machine Learning techniques. To achieve this aim, we have used techniques like Logistic Regression, Random Forest, Naive Bayes and Decision tree on the dataset to get the results. These results will be helpful in future for banks and will increase their profitability by checking for loan defaulters. Firstly, Feature Sampling of the dataset will be done for the optimum results. Secondly, pre-processing of the dataset will be done. That includes the data set being cleaned of its missing and null values using various techniques coded in Python. Then moving on to the third step standardization of the dataset will be done after that splitting of the dataset into training and testing dataset. The model will be then trained on the training set and tested using a test set. After the tuning of the hyperparameters of the dataset we will get the best parameters to get the best results. We will then retrain the model using those parameter values and will observe the accuracy using each technique of machine learning.

### 3.4 Algorithm / Pseudo code of the Project Problem –

**Step- 1:** Collect the data of customers who want to take loan.

**Step- 2:** Data Cleaning by checking for any null values.

**Step- 3:** Data Preprocessing and scaling of data.

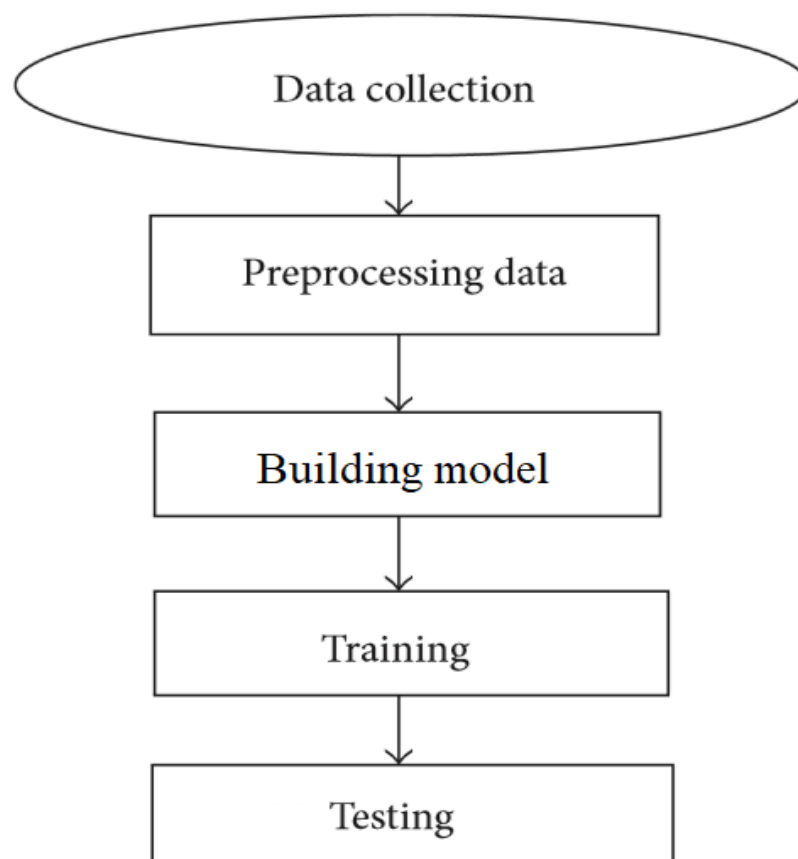
**Step- 4:** Splitting of that data into train and test data in ratio 80-20.

**Step- 5:** Training the data and building the machine learning model.

**Step- 6:** Evaluating Performance metrics i.e., accuracy, precision, recall, f-score and above used models.

**Step- 7:** Finally predicting the result.

### 3.5 Flow graph of the Minor Project Problem –



**Figure – 5: Flow graph**



## 3.6 Screen shots of the various stages of the Project

### 3.6.1 Libraries Used:

The following libraries have been used for the implementation of the model and for training the data.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

**Figure 6: Libraries Used**

### 3.6.2 Loading the Dataset:

✓  
0s

[2] dataset=pd.read\_csv("Loan\_Prediction.csv")

Printing the first 5 rows of the dataframe

+ Code

+ Text

✓  
0s

[3] dataset.head()

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	LP001002	Male	No	0	Graduate	No	5849	0.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0

**Figure 7: Loading the dataset**

### 3.6.3 Data Cleaning:

The data needs to be cleaned because of big quantities, missing data, and noisy data present in real-life data. These data need to be cleaned to overcome certain issues and make accurate predictions. Noise and missing values are very common when it comes to cleaning collected data and they must be filled in or dropped out in order to obtain a correct and reliable result. To make data more understandable, transformation shifts the format of the data from one type to another.

```
✓ 0s dataset.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
dtype: int64
```

**Figure 8: Checking Null Values**

```
✓ 0s [8]
dataset['Gender'].fillna(dataset['Gender'].mode()[0],inplace=True)
dataset['Married'].fillna(dataset['Married'].mode()[0],inplace=True)
dataset['Dependents'].fillna(dataset['Dependents'].mode()[0],inplace=True)
dataset['Self_Employed'].fillna(dataset['Self_Employed'].mode()[0],inplace=True)
dataset.LoanAmount = dataset.LoanAmount.fillna(dataset.LoanAmount.mean())
dataset['Loan_Amount_Term'].fillna(dataset['Loan_Amount_Term'].mode()[0],inplace=True)
dataset['Credit_History'].fillna(dataset['Credit_History'].mode()[0],inplace=True)
```

**Figure 9: Handling those null or missing values by filling them**

```
dataset.isnull().sum()
```

```
Loan_ID      0
Gender        0
Married       0
Dependents    0
Education     0
Self_Employed 0
ApplicantIncome 0
CoapplicantIncome 0
LoanAmount    0
Loan_Amount_Term 0
Credit_History 0
Property_Area 0
Loan_Status   0
dtype: int64
```

**Figure 10: Checking null values again after filling them**

### 3.6.4 Using crosstabs:

It is a table which shows relationship between two or more variables, it is basically used for summarizing the data and is computed by `crosstab()` function.

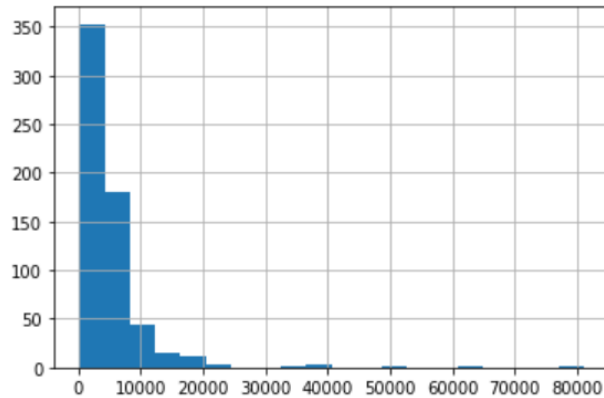
```
[ ] pd.crosstab(dataset['Credit_History'],dataset['Loan_Status'],margins=True)
```

Loan_Status	N	Y	All
Credit_History			
0.0	82	7	89
1.0	110	415	525
All	192	422	614

**Figure 11: Credit history affect another applicant**

```
dataset['ApplicantIncome'].hist(bins=20)
```

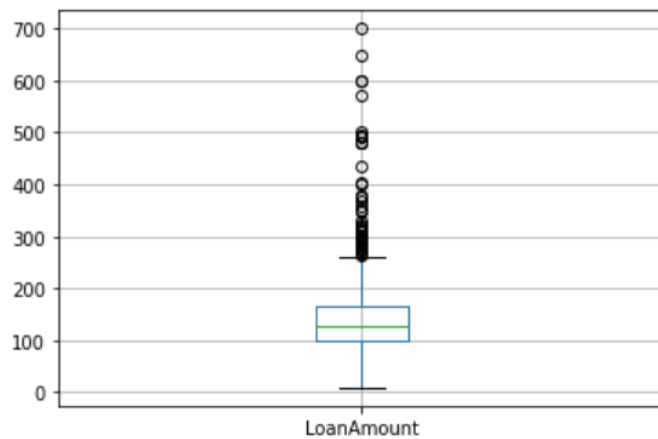
<matplotlib.axes.\_subplots.AxesSubplot at 0x7f7b98ea4c50>



**Figure-12: Histogram for Applicant Income**

```
dataset.boxplot(column='LoanAmount')
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f7b9881d6d0>



**Figure-13: Boxplot for column Loan Amount**

### 3.6.5 Training Testing Split:

To train a model and test it again we need two sets of data, one called the training set used to train the model and other is the testing set used to test the model. However, the ideal split ratio is not fixed anywhere, it majorly depends on the data set and personal choice, problem statements etc. but to take a safe approach and avoid the problem of overfitting, we have used 80-20 split i.e., of the total data we have 80% as training data and 20% will be as the testing data. To do this we have import `train_test_split()` from `sklearn` library.

```
✓ [24] from sklearn.model_selection import train_test_split
      x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2, random_state=0)
```

**Figure 14: Train test split**

### 3.6.6 Data Preprocessing:

Data needs to be preprocessed for better analysis and accurate results, we have used Label Encoder to preprocess the data, Label Encoder converts all the categorical values to binary values i.e., 0 and 1 which makes it understandable by machine.

```
[ ] from sklearn.preprocessing import LabelEncoder
    labelencoder_x = LabelEncoder()

[ ] for i in range(0,5):
    x_train[:,i]=labelencoder_x.fit_transform(x_train[:,i])

[ ] x_train[:,7]=labelencoder_x.fit_transform(x_train[:,7])

[ ] x_train

array([[1, 1, 0, ..., 1.0, 5858.0, 267],
       [1, 0, 1, ..., 1.0, 11250.0, 407],
       [1, 1, 0, ..., 0.0, 5681.0, 249],
       ...,
       [1, 1, 3, ..., 1.0, 8334.0, 363],
       [1, 1, 0, ..., 1.0, 6033.0, 273],
```

**Figure- 15: Data Preprocessing**

### 3.6.7 Standardization of Data:

Standardization is the technique mostly used on the dataset before performing the Machine Learning Algorithms to standardize the range of the data. We import Standard Scaler function from Sklearn which basically scales in a certain range and its quite important since we have different variables with different range, so for better analysis, it's good to scale the data. The data we used after preprocessing is standardized with the help of Python Sklearn Library.

```
✓ [37] from sklearn.preprocessing import StandardScaler  
      ss=StandardScaler()  
      x_train=ss.fit_transform(x_train)  
      x_test=ss.fit_transform(x_test)
```

**Figure 16: - Standardization of data**

### 3.6.8 Methods and Techniques:

After the Preprocessing of the data, now it's time to implement the machine learning method to the dataset. This will be done by using Python Sklearn library. We are applying various Machine Learning techniques to implement our model starting with the importing of the data set and its preprocessing. We would use four techniques namely Logistic Regression, Random Forest, Decision Tree and Naïve Bayes to check and compare the accuracy and build our model.

### 3.6.8.1 Random Forest:

Random forest is a supervised machine learning algorithm which is used to solve both regression as well as classification type of problems. It is based on the model of ensemble learning which means combining multiple classifiers to solve a complex problem and improve its accuracy. It is more like decision tree, but the main difference is it combines multiple decision tree to solve a problem and then takes the average of them to predict the result. More the number of trees in forest means greater the accuracy and lessen the chance of overfitting. The advantage of random forest is that it provides better accuracy even with the large dataset even if it is of high dimensionality. This is done by using RandomForestClassifier () from Python Sklearn.ensemble.

```
✓ [59] from sklearn.ensemble import RandomForestClassifier
0s      clf = RandomForestClassifier(n_estimators = 100)
      clf.fit(x_train, y_train)
      y_pred = clf.predict(x_test)

✓ [60] print("The accuracy of Random Forest is: ", metrics.accuracy_score(y_test, y_pred)*100)
0s

The accuracy of Random Forest is: 77.23577235772358
```

**Figure- 17: Accuracy using Random Forest**

### 3.6.8.2 Logistic Regression:

Logistic Regression is a supervised machine learning algorithm which is used to solve classification problems and predicts the output of categorical dependent variable and where the output comes out as a discrete value i.e., either 0 or 1 or true or false etc. This algorithm is very simple and significant as it can provide probability and classify new data using continuous as well as discrete values. In this we fit an “S” shaped curve which has range between 0 and 1 and this curve basically talks about the likelihood of an event one is predicting and, in the curve, we use the concept of threshold value which is basically set to 0.5 i.e., values above it will be taken as 1 and below it will be taken as 0. The activation function used in this is known as Sigmoid function.

```
✓ [52] from sklearn.linear_model import LogisticRegression
0s classifier = LogisticRegression(random_state = 0)
classifier.fit(x_train, y_train)

LogisticRegression(random_state=0)

✓ [53] y_pred=classifier.predict(x_test)
0s y_pred

array([1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1])

✓ [54] print ("The accuracy of Logistic Regression is: ", metrics.accuracy_score(y_test, y_pred)*100)
0s

The accuracy of Logistic Regression is: 82.92682926829268
```

**Figure- 18: Accuracy using Logistic Regression**



### 3.6.8.3 Decision Tree:

It is a supervised machine learning algorithm used to solve classification as well as regression problems. It is basically a tree-structured classifier and the internal nodes of tree is represented by the features of the dataset and the branches of the tree is represented by decision rules and the leaf nodes represents the result/outcome. It is known as decision tree as it starts from the root node and grows like a tree where leaf node represents the outcome and in order to build the tree we take help of CART algorithm i.e., Classification and regression tree algorithm. It is very easy to understand and use decision tree algorithm as it mimics the human thinking ability and displays it in form of graphical tree representation. One of the best advantages of decision tree is that there is less requirement of data cleaning as compared to other algorithms however there maybe the issue of overfitting which can be resolved using random forest algorithm.

```
[38] from sklearn.tree import DecisionTreeClassifier
      DTClassifier = DecisionTreeClassifier(criterion='entropy',random_state=0)
      DTClassifier.fit(x_train,y_train)

      DecisionTreeClassifier(criterion='entropy', random_state=0)

y_pred=DTClassifier.predict(x_test)
y_pred

array([1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1,
       1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1,
       1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1,
       0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0])

[40] from sklearn import metrics
      print('The accuracy of decision tree is: ',metrics.accuracy_score(y_pred,y_test)*100)

The accuracy of decision tree is: 61.78861788617886
```

**Figure – 19: Accuracy using Decision Tree**

### 3.6.8.4 Naive Bayes:

Naïve bayes is a supervised machine learning algorithm which is used to solve binary as well as multiclass classification problems, being one of the most simple and effective classifiers, it helps in building machine learning model which can make quick predictions. Naive bayes predicts on the basis of the probability factor of the situation and hence is called probabilistic classifier. It is based on the principle of bayes theorem and hence it is called naïve bayes. Some of the applications of it are text classification, spam filtering etc.

```
[45] from sklearn.naive_bayes import GaussianNB
      NBClassifier = GaussianNB()
      NBClassifier.fit(x_train,y_train)

      GaussianNB()

[46] y_pred=NBClassifier.predict(x_test)

      y_pred

      array([1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1,
            1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1])

[47] print('The accuracy of Naive Bayes is: ',metrics.accuracy_score(y_pred,y_test)*100)

      The accuracy of Naive Bayes is:  83.73983739837398
```

**Figure – 20: Accuracy using Naive Bayes**

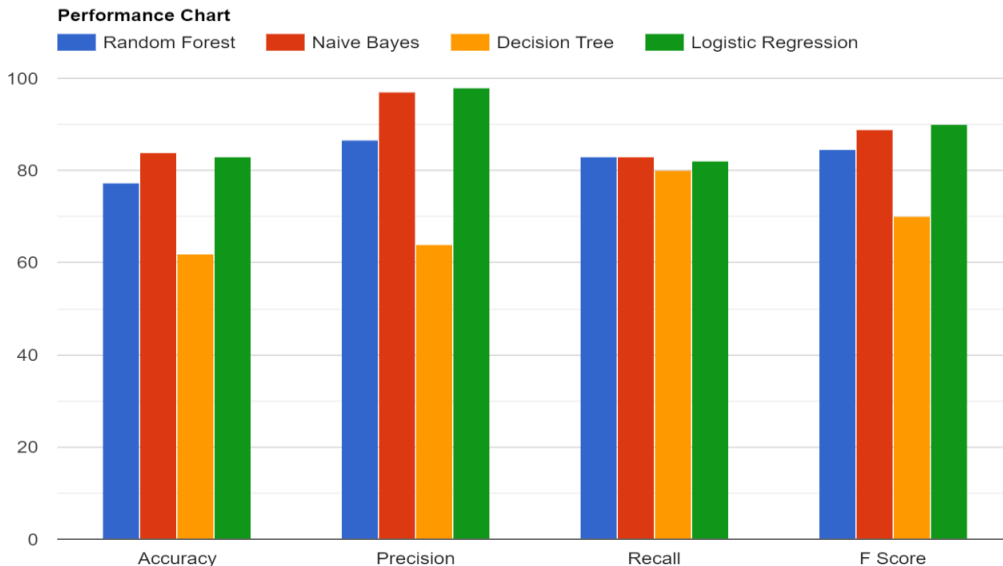
# Chapter 04: RESULTS

## 4.1 Discussion on the Results Achieved –

The goal of our project was to achieve the most significant results of whether a person is eligible for loan or not. These predictions were made by using various ML algorithms like Naïve Bayes, Logistic Regression, Decision tree and Random Forest. The modelling was done by using Python 3.8 and its necessary libraries. Code was performed by all the team members on their respective system but main results output was performed on Intel(R) Core (TM) i5-1035G1 with CPU @ 1.00GHz 1.19 GHz and RAM 8GB.

**Table 3: - Tabular representation of results**

AI	Random Forest	Naïve Bayes	Decision Tree	Logistic Regression
Accuracy	77.23%	83.73%	61.78%	82.92%
Precision	86.66%	97.77%	63.33%	97.77%
Recall	82.97%	83.01%	80.28%	82.24%
F Score	84.78%	89.79%	70.80%	89.34%



**Figure-21: Visual representation of different algorithms**

## **4.2 Application of the Minor Project –**

Its major application is in finding loan fraudsters, it can directly affect banks profitability by checking who is eligible for loan and who is not and hence preventing frauds which are very common in today's time.

## **4.3 Limitation of the Minor Project –**

The limitation of our project is that it emphasizes different weights to each factor but in real life sometimes loan can be approved on the basis of only one single strong factor only, which is however not possible through this system.

## **4.4 Future Work –**

In future, we are expecting to compare and hybridize on other machine learning algorithms to obtain the most optimum result with the same dataset. Optimum result defines the results in which the accuracy and precision of the model is significant than obtained results till now.

# References

- [1]. Anchal Goyal, Ranpreet Kaur, “A survey on Ensemble Model for Loan Prediction”,In 2016 International Journal of Engineering Trends and Applications (IJETA) – Volume 3 Issue 1, Jan-Feb 2016
  
- [2]. Rhishab Mukherjee, Vanshika Madan, “Factors that affect loan giving decision of banks” In April,2021
  
- [3]. Sharayu Dosalwar Et al.,” Analysis of loan availability using ML techniques", In 2021 International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) - Volume 9, Issue 1, September 2021
  
- [4]. <https://www.mdpi.com/2073-4441/10/11/1536/htm>