NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA

Introduction to Machine Learning(18CS3751)

on

Loan Prediction using Machine Learning

Submitted in partial fulfilment of the requirement for the award of Degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

Tarun Kumar Arcot	**1NT18CS174**
Rakshith R	**1NT19CS413**

Department of Computer Science and Engineering

(Accredited by NBA Tier-1)

2021-2022

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM
, APPROVED BY AICTE & GOVT.OF KARNATAKA)

Department of Computer Science and Engineering

(Accredited by NBA Tier-1)

CERTIFICATE

This is to certify that the Internship Report on **"Python Chat Bot using Machine Learning"** is an authentic work carried out by **Tarun Kumar Arcot (1NT18CS174),Rakshith R(1NT19CS413)** bonafide students of **Nitte Meenakshi Institute of Technology**, Bangalore in partial fulfilment for the award of the degree of ***Bachelor of Engineering*** in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belagavi during the academic year ***2021-2022.*** It is certified that all corrections and suggestions indicated during the internal assessment has been incorporated in the report.

Internal Guide	**Signature of HOD**
Dr Vani Vasudevan Professor, CSE, NMIT Bangalore	Dr. Sarojadevi H Professor, Head, Dept. CSE, NMIT Bangalore

DECLARATION

We hereby declare that

(i) The project work is our original work

(ii) This Project work has not been submitted for the award of any degree or examination at any other university/College/Institute.

(iii) This Project Work does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

(iv) This Project Work does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

a) their words have been re-written but the general information attributed to them has been referenced;

b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.

(v) This Project Work does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

NAME	USN	Signature
Tarun Kumar Arcot	1NT18CS174	
Rakshith R	1NT19CS413	

Date: 17/01/2022

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. I express my sincere gratitude to our Principal **Dr. H. C. Nagaraj**, Nitte Meenakshi Institute of Technology for providing facilities.

We wish to thank our HoD**, Dr. Sarojadevi H.** for the excellent environment created to further educational growth in our college. We also thank him for the invaluable guidance provided which has helped in the creation of a better project.

I hereby like to thank ***Dr Vani Vasudevan, Professor, CSE,NMIT Bangalore*** on his periodic inspection, time to time evaluation of the project and help to bring the project to the present form. Thanks to our Departmental Project coordinators. We also thank all our friends, teaching and non-teaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the project.

Signature

Tarun Kumar Arcot	**1NT18CS174**	
Rakshith R	**1NT19CS413**	

Date: 17/01/2021

ABSTRACT

In today's world loan disbursement is one of the major business lines for the banks. However, Loan disbursement can be a risky business if the banks are unable to take an informed decision on whom to give the loan and whom not. Banks ask for various information and documents to ascertain the credibility of the client.

In this project an attempt has been made to use machine learning algorithms like Logistic regression, naive bayes, Decision tree and random forest to predict whether the next client who applies for the loan will repay the loan in time or not and thus whether the bank should disperse the loan. Instead of human taking the decision which may be subjective, machine learning algorithms can help take accurate decision based on multiple information provided by the customer. In today's world of AI & ML, decision of whom to give loan is decided by the bank based on various past data trends and by using mathematical algorithm. ML has reduced the errors considerably in disbursement of loans and thus improve loan recovery in time, hence higher profitability to the banks.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

The improvements in the fields of inter-networking and information technology have been intricate in executing an Artificial Intelligent (AI) and Machine Learning (ML) systems. These systems are drawing nearer of human activities, for example, choice emotionally supportive networks, robotics, natural language processing, and so forth. Indeed, machine learning algorithms have helped in forecasting the future trends. Machine learning and deep learning algorithms and models process an immense amount of data to enable faster, smarter, and better business decisions. As such, machine learning forecasting for the financial industry holds incredible potential for banks, the historical custodians of vast stores of data. However, the technology's direct impact is still marginal as only a few institutions have capitalized on the technology's extensive potential. Indeed, even in the machine learning fields, there are some strategies and algorithms that can improve the accuracy of the loan prediction.

1.1 Motivation:

- In today's world of AI & ML, decision of whom to give loan is decided by the bank based on various past data trends and by using mathematical algorithm. ML has reduced the errors considerably in disbursement of loans and thus improve loan recovery in time, hence higher profitability to the banks.
- In this project, an attempt has been made to use some of Machine Learning Algorithms like Decision tree classification, Logistics regression, Naive Bayes, and Support vector Machines to support the Bank's decision on whether to give loan to a new client. This way we are moving away from human based decision making which could be ambiguous to a model-based decision making which is robust.

1.2 Problem Domain:

- It uses machine learning algorithms which come under supervised learning.
- The algorithms belong to even classification and regression.

1.3 Objectives:

The algorithm should be able to predict the loan eligibility so that the bank manager who is working for the bank can predict the whether the client will be able to repay the loan in the specified time.

The algorithms should be to build up the model by learning and responding accurately to the dataset, by using training dataset to check the predictions on the test dataset.

Also, by successfully parsing the training dataset and forecast the dataset. Algorithms work with test data to predict the output. Market products and enable their purchases.

CHAPTER 2

Data Source and Data Quality

2.1Data Source

1. The dataset is taken from the Kaggle website. The dataset used is train.csv and test.csv.

2. Train will be used to develop a model and test its accuracy while test will be dataset having missing values and used to evaluate the model

3. The origin of the dataset is Kaggle is from https://www.kaggle.com/vikasukani/loan-eligibility-prediction-machine-learning/data

4. The column in the dataset used are:-

Loan ID----------> Unique Loan ID.

Gender ----------> Male/ Female

Married ----------> Applicant married (Y/N)

Dependents ----------> Number of dependents

Education ----------> Applicant Education (Graduate/ Under Graduate)

Self Employed ---------> Self-employed (Y/N)

Applicant Income -------> Applicant income

Co-applicant Income -----> Co-applicant income

Loan Amount ----------> Loan amount in thousands

Loan Amount Term ------> Term of a loan in months

Credit History --------> Credit history meets guidelines

Property Area ---------> Urban/ Semi-Urban/ Rural

Loan Status ----------> Loan approved (Y/N)

2.2 Data Quality: -

- In the csv file obtained from the Kaggle, there are several fields not having any value.
- Some of the columns/attributes have string values which needs to be converted into numerical value for the purpose of training and testing algorithm.

Gender	**Married**	**Education**	**Self Employed**	**Property Area**	**Loan Status**	**Loan ID**

- The single value attribute in the dataset is: -

Gender	**Married**	**Loan Status**	**Education**	**Credit History**

- The continuous value attributes in the dataset are: -

Applicant Income	**Property Area**	**Loan ID**	**Dependents**	**Co-applicant Income**	**Loan Amount**	**Loan Amount Term**

- Total no of rows/ customer details was 614/13. Whereas some of the attributes had lesser number of rows.
- The y attribute (dependent variable) is loan status, this is the attribute which will help the organization to take the decision of taking the loan or not.
- The x attributes (other columns) are the independent variables.
- There are many outliers.

CHAPTER 3

Data Pre-processing

Before the data pre-processing is done, we need to import various libraries which will support this project. These are: -

Pandas – Library for data analysis, manipulation, and filtering.

NumPy – Used to perform mathematical and logical operations on arrays

Matplotlib – Comprehensive library for creating static, interactive and animated visualization

Scikit learn – Most useful library with efficient tools for machine learning and statistical modelling including range of supervised and unsupervised learning algorithm.

3.1 Data Cleaning

3.2 Data transformation

3.3 Data Reduction

3.1 Data cleaning: -

- To view the dataset, we have the following: -

```
dataset.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Hist
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	

- We will see the number of values in each column/attribute in the dataset:-

```
[73] dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

As we can see above there are missing data in some of the columns/attributes

- To see the total number of rows and columns in the dataset we can use the command:

```
[72] dataset.shape
     (614, 13)
```

- Let us now describe the dataset: -

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

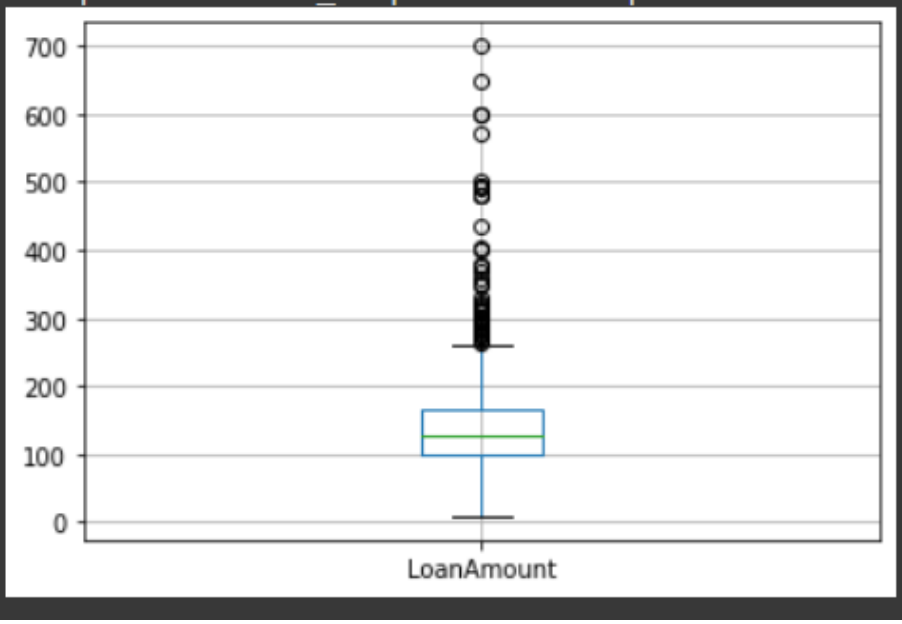
	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
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

This describes the various statistical parameters of the attributes in the dataset.

- Boxplots for Loan Amount

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

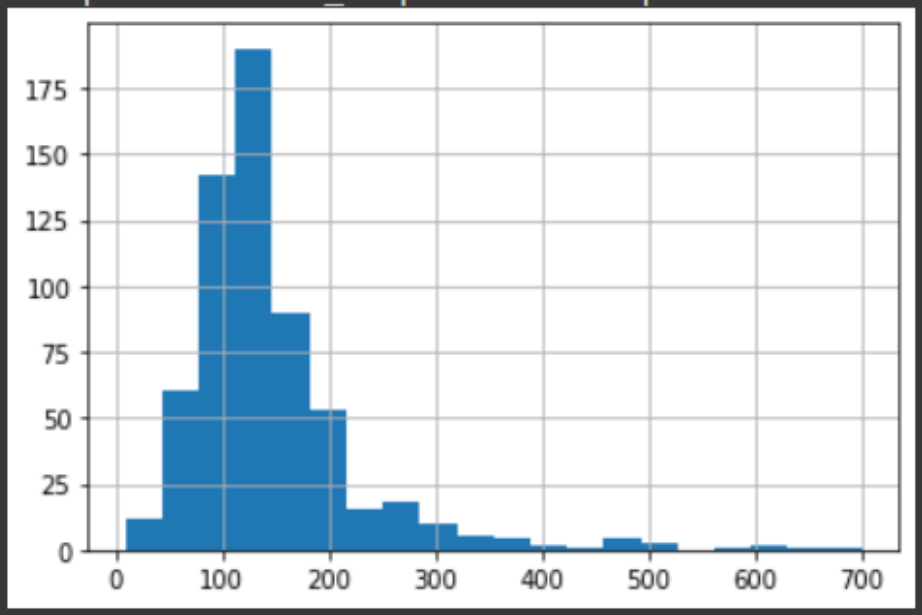
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4e0d22c110>
```



There are outliers in the dataset as shown in the graph. Even loan amount is right skewed

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

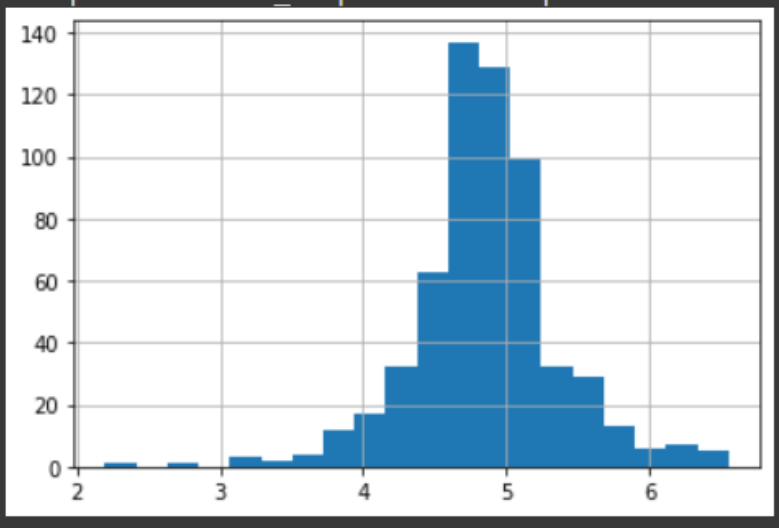
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4e0d236b50>
```



- Normalizing the Loan Amount attribute in the dataset we get the histogram graph which looks more normalized

```
[79] dataset['LoanAmount_log']=np.log(dataset['LoanAmount'])
     dataset['LoanAmount_log'].hist(bins=20)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4e12185650>
```



- Let’s find out how many missing values are there:-

```
[80] 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
     LoanAmount_log        22
     dtype: int64
```

- Filling the missing values with NA. (Note: - For alphabetic valued attributes use mode and for numeric it is mean)

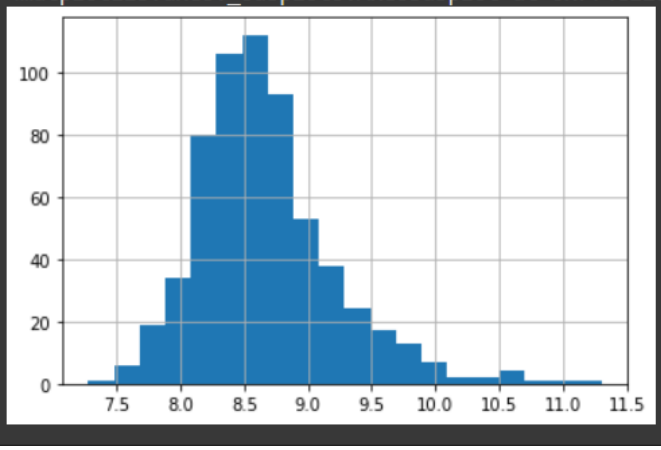
```
[83] dataset['Gender'].fillna(dataset['Gender'].mode()[0],inplace=True)

[84] dataset['Married'].fillna(dataset['Married'].mode()[0],inplace=True)

[85] dataset['Dependents'].fillna(dataset['Dependents'].mode()[0],inplace=True)

[86] dataset['Self_Employed'].fillna(dataset['Self_Employed'].mode()[0],inplace=True)

[87] dataset.LoanAmount = dataset.LoanAmount.fillna(dataset.LoanAmount.mean())
     dataset.LoanAmount_log = dataset.LoanAmount_log.fillna(dataset.LoanAmount.mean())

[88] dataset['Loan_Amount_Term'].fillna(dataset['Loan_Amount_Term'].mode()[0],inplace=True)

[89] dataset['Credit_History'].fillna(dataset['Credit_History'].mode()[0],inplace=True)
```

- The data set is normalized,

```
[90] 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
LoanAmount_log       0
dtype: int64
```

- To get the total income and total income log do the following

```
[91] dataset['TotalIncome']= dataset['ApplicantIncome']+ dataset['CoapplicantIncome']
     dataset['TotalIncome_log'] =np.log(dataset['TotalIncome'])

[92] dataset['TotalIncome_log'].hist(bins=20)

<matplotlib.axes._subplots.AxesSubplot at 0x7f4e1209d650>
```



3.2 Data Transformation:

- Dividing the dataset into independent and independent variables

```
[ ]  X=dataset.iloc[:,np.r_[1:5,9:11,13:15]].values
     y=dataset.iloc[:,12].values
```

```
X
```

```
array([['Male', 'No', '0', ..., 1.0, 146.41216216216213, 5849.0],
       ['Male', 'Yes', '1', ..., 1.0, 4.852030263919617, 6091.0],
       ['Male', 'Yes', '0', ..., 1.0, 4.189654742026425, 3000.0],
       ...,
       ['Male', 'Yes', '1', ..., 1.0, 5.53338948872752, 8312.0],
       ['Male', 'Yes', '2', ..., 1.0, 5.231108616854587, 7583.0],
       ['Female', 'No', '0', ..., 0.0, 4.890349128221754, 4583.0]],
      dtype=object)
```

```
y
```

```
array(['Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'Y', 'Y',
       'N', 'Y', 'Y', 'Y', 'N', 'N', 'Y', 'N', 'Y', 'N', 'N', 'N', 'Y',
       'Y', 'Y', 'N', 'Y', 'N', 'N', 'N', 'Y', 'N', 'Y', 'N', 'Y', 'Y',
       'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y',
       'N', 'N', 'N', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'N', 'N',
       'N', 'N', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'N', 'N',
       'N', 'Y', 'Y', 'Y', 'N', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'N', 'N', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y',
       'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'N',
       'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'N', 'Y', 'N', 'N', 'N', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'N', 'N', 'Y', 'Y',
       'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'Y', 'N', 'Y', 'N', 'N', 'N',
       'Y', 'N', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'N', 'Y', 'Y',
       'N', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'N', 'N', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'N',
       'Y', 'Y', 'Y', 'Y', 'N', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'Y',
       'Y', 'Y', 'Y', 'N', 'N', 'Y', 'Y', 'N', 'Y', 'N', 'N', 'N', 'N',
       'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'Y',
       'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N',
       'N', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'N', 'Y', 'Y', 'Y',
       'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y',
       'N', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
       'N', 'Y', 'N', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y',
       'N', 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'N', 'Y', 'N', 'Y', 'Y', 'Y',
       'Y', 'N', 'N', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y',
       'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'N', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'N', 'N', 'N', 'Y',
       'Y', 'N', 'Y', 'Y', 'Y', 'N', 'N', 'N', 'Y', 'N', 'Y', 'N', 'Y',
       'N', 'N', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'Y', 'N', 'Y', 'Y',
       'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'N', 'N', 'N', 'Y', 'N', 'Y', 'Y',
       'Y', 'Y', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y',
       'Y', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'Y',
       'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y',
       'N', 'Y', 'Y', 'N', 'Y', 'Y', 'N', 'N', 'Y', 'Y', 'N', 'N', 'N',
       'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'N',
       'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y',
       'N', 'Y', 'N', 'Y', 'N', 'Y', 'Y', 'N', 'N', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'N', 'N', 'N', 'Y', 'N',
       'Y', 'N', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'N',
       'N', 'Y', 'Y', 'Y', 'N', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'N',
       'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'N', 'Y', 'Y', 'Y', 'Y', 'Y',
       'Y', 'Y', 'N'], dtype=object)
```

- Splitting the dataset into the 2(train and test dataset)-

```
[ ]  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)
```

```
[ ] 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)
```

```
print(x_train)
```

```
[['Male' 'Yes' '0' ... 1.0 4.875197323201151 5858.0]
 ['Male' 'No' '1' ... 1.0 5.278114659230517 11250.0]
 ['Male' 'Yes' '0' ... 0.0 5.003946305945459 5681.0]
 ...
 ['Male' 'Yes' '3+' ... 1.0 5.298317366548036 8334.0]
 ['Male' 'Yes' '0' ... 1.0 5.075173815233827 6033.0]
 ['Female' 'Yes' '0' ... 1.0 5.204006687076795 6486.0]]
```

- Converting the characters (Male or Female, Graduated or not Graduated, and so on(to binary value(1 or 0, and so on). So, we need to use label encoder.

```
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[:, i] = labelEncoder_x.fit_transform(x_train[:, 7])
```

```
[ ] x_train
```

```
array([[1, 1, 0, ..., 1.0, 4.875197323201151, 5858.0],
       [1, 0, 1, ..., 1.0, 5.278114659230517, 11250.0],
       [1, 1, 0, ..., 0.0, 5.003946305945459, 5681.0],
       ...,
       [1, 1, 3, ..., 1.0, 5.298317366548036, 8334.0],
       [1, 1, 0, ..., 1.0, 5.075173815233827, 6033.0],
       [0, 1, 0, ..., 1.0, 5.204006687076795, 6486.0]], dtype=object)
```

```
[ ] labelencoder_y=LabelEncoder()
    y_train=labelencoder_y.fit_transform(y_train)
```

```
[ ] y_train
```

```
y_train
```

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

```
[ ]  for i in range (0, 5):
       x_test[:, i]= labelEncoder_x.fit_transform(x_test[:,i])
```

```
[ ]  x_test[:, 7] = labelEncoder_x.fit_transform(x_test[:, 7])
```

```
[ ]  labelencoder_y=LabelEncoder()
     y_test=labelencoder_y.fit_transform(y_test)
```

- We have different values in the dataset with different ranges. So it is important to scale the dataset so that it is easy for analysis and prediction.

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

CHAPTER 4

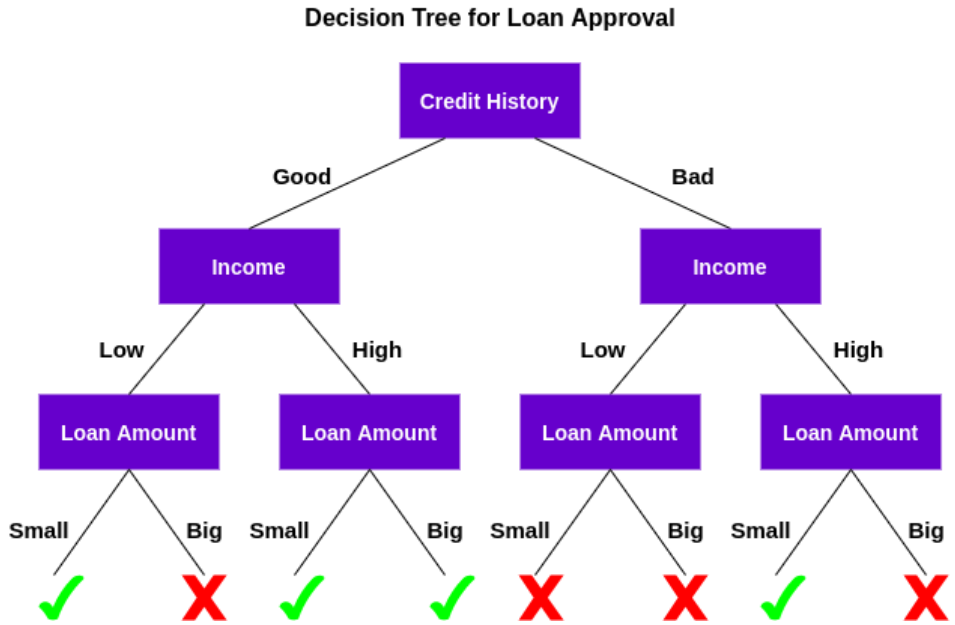
Machine Learning Methods

The algorithms used are: -

1. Decision Tree Classifier
2. Naïve Bayes Classifier
3. Logistic Regression
4. Random forest
5. Support Vector Machine
6. KNN Algorithm

1. Decision tree Classifier

- Decision Tree is a Supervised Machine Learning Algorithm that uses a set of rules to make decisions, similarly to how humans make decisions.
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules, and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The example is



- There are two popular techniques for ASM, which are:

  - Information Gain- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.

$$Information\ Gain = Entropy(S) - [(Weighted\ Avg) * Entropy(each\ feature)]$$

Gini Index-Gini index is a measure of impurity or purity used while creating a decision tree in the CART (Classification and Regression Tree) algorithm.

$$Gini\ index = 1 - \sum j * \text{Pj2}$$

Were,

S= Total number of samples

P(yes)= probability of yes

P(no)= probability of no

- Python code:

```
from sklearn.tree import DecisionTreeClassifier
DTClassifier=DecisionTreeClassifier(criterion='entropy',random_st
ate=0)
```

```
DTClassifier.fit(x_train,y_train)
```

2. Naïve Bayes Classifier-

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features.
- Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem

$$P(A|B) = \frac{(P(B|A)P(A))}{P(B)}$$

Were,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

- Python Code:

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

3. Logistics Regression-

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic Regression is much like the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- The diagram is as follows:

- The equation for logistics equation: -

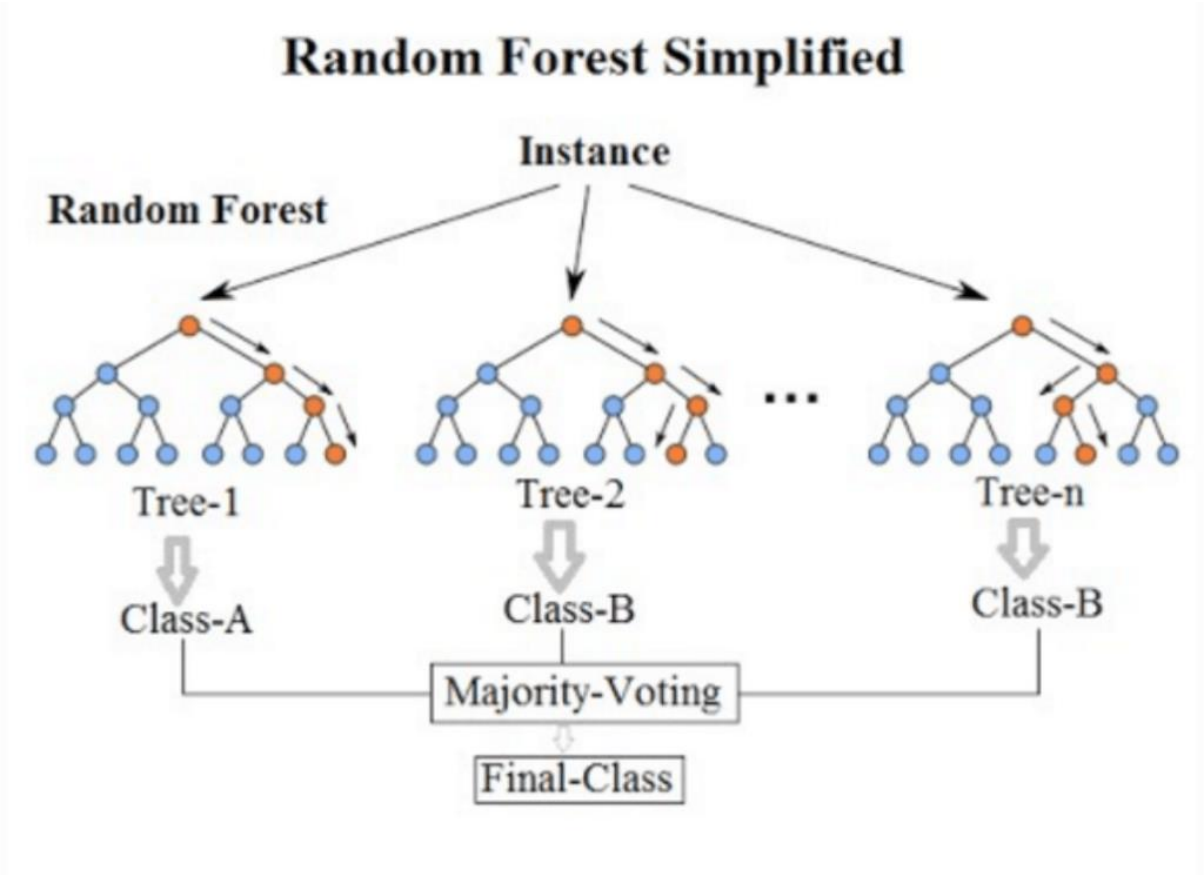
$$\log\left[\frac{y}{1-y}\right] = b0 + b1x1 + b2x2 + b3x3 + \cdots + bnxn$$

- Python Code: -

```
from sklearn.naive_bayes import GaussianNB
NBClassifier = GaussianNB()
NBClassifier.fit(x_train,y_train)
logistic_model = LogisticRegression()
```

4. Random Forest-

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
- Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.
- The diagram is: -

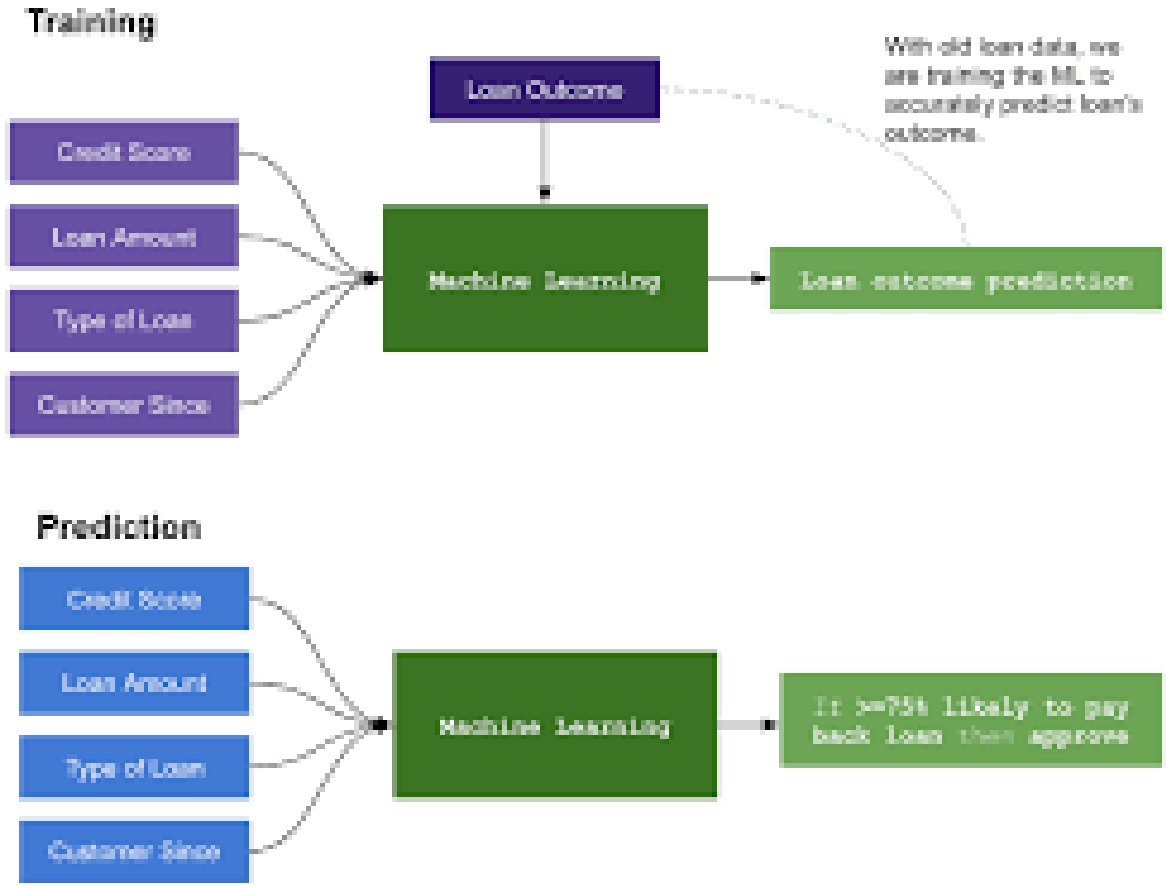


- Python code-

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 600)
rf.fit(x_train,y_train)
```

5. Support Vector Machine –

- 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.
- The graph is:



- Python Code:

```
from sklearn import svm
from sklearn.metrics import accuracy_score
classifier = svm.SVC(kernel='linear')
classifier.fit(x_train,y_train)
x_train_prediction = classifier.predict(x_train)
```

CHAPTER 5

RESULTS And Discussion

The accuracy of each algorithm is:

Decision tree classifier = 0.4146341463414634

Naïve Bayes Classifier = 0.8130081300813008

Logistic Regression = 0.8085539714867617

Random forest classifier = 1.0

Support Vector machine = 0.8044806517311609

K Nearest Neighbour = 0.835030549898167

- From this accuracy test we see that random forest has the highest accuracy of 100%.
- This dataset can use random forest for predicting the output

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this project, we see that we have seen following process which is included as the machine learning process: -

1. Data pre-processing
    a. Data Cleaning
    b. Data Transformation
2. Machine learning algorithms
    a. Decision tree classifier
    b. Naïve Bayes
    c. Random Forest
    d. Support vector machines
    e. Logistic Regression

There are various other classification and regression algorithms which can be used in the dataset.

BIBLIOGRAPHY

https://www.javatpoint.com/machine-learning

https://colab.research.google.com/?utm_source=scs-index

Introduction to Machine Learning by Ethem Alpaydin (z-lib.org)

Machine Learning - Tom Mitchell

https://deepnote.com/@rhishab-mukherjee/Loan-Prediction-Project-TermPaper-VPSOpiywSu6FZeN2fK8fug