A
Project Report
On

"CREDIT DECISION-MAKING SYSTEM"

Submitted in partial fulfillment of
the requirements for the 8th Semester Sessional Examination of

BACHELOR OF TECHNOLOGY
IN

Computer Science and Engineering

By

LALATENDU NAYAK (1901060145)
DINESH KUMAR GOUDA (1901060323)
KAUSTUV PATRA (1901060103)

Under the esteemed guidance of

Mr. Murali Kr. Senapaty

SCHOOL OF ENGINEERING AND TECHNOLOGY
Department of Computer Science and Engineering
GIET University, GUNUPUR – 765022

2023-24

GIET UNIVERSITY

GUNUPUR - 765022, ODISHA, www.giet.edu

CERTIFICATE

This is to certify that the project work entitled "CREDIT DECISION-MAKING SYSTEM" is done by Lalatendu Nayak- (1901060145), Dinesh Kumar Gouda- (1901060323) , Kaustuv patra- (1901060103) in partial fulfillment of the requirements for the 8th Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2023-24. This work is submitted to the department as a part of evaluation of 8th Semester Major Project-2.

Mr. Murali Kr. Senapaty	*Dr.Sachikanta Dash*	*Dr. (Mrs) Bandita sahu*
Project Supervisor	*Project Co-ordinator*	HoD, CSE

ACKNOWLEDGEMENT

We express our sincere gratitude to our project guide **Prof Mr. Murali Kr. Senapaty** of Computer science and engineering for giving me an opportunity to accomplish the project. Without his active support and guidance, this project report has been successfully completed.

We also thank Dr. **Bandita Sahu**, Head of the department of computer science, Prof. (Dr.) **Sanjay Kumar Kuanar**, Dy. Dean, SOET and **Dr. Sachikanta Dash**, Project Coordinator for their consistent support, guidance and help.

Lalatendu Nayak (1901060145)
Dinesh Kumar Gouda (1901060323)
Kaustuv Patra (1901060103)

ABSTRACT

Taking out loans from banks has become very common in today's world. Banks' main business is lending money. The primary source of benefit is the interest on the loan. However, because the bank has limited funds to distribute to a limited number of people, determining who the loan can be given to and who would be a better choice for the bank is standard procedure. Credit firms issue a loan after a lengthy period of authentication and confirmation. They are also concerned about whether the borrower will be able to repay the loan without difficulty. Many researchers have been exploring systems for determining loan acceptance in recent years. Machine learning can bring an additional reliable predictive modeling method to the banking business, which is still needed. The primary aim of this paper is to determine if a loan granted to some organization or a specific person would be accepted.

Keywords: Classification, Exploratory Data Analysis, Loan, Loan Approval, Machine Learning, Prediction, Python.

TABLE OF CONTENT

Chapter - 01

INTRODUCTION

The two most pressing issues in the banking sector are:

1) How risky is the borrower?

2) Should we lend to the borrower given the risk?

The response to the first question dictates the borrower's interest rate. Interest rate, among other things (such as time value of money), tests the riskiness of the borrower, that is the higher the interest rate, the riskier the borrower. We will then decide whether the applicant is suitable for the loan based on the interest rate. Lenders (investors) make loans to creditors in return for the guarantee of interest-bearing repayment. That is, the lender only makes a return (interest) if the borrower repays the loan. However, whether he or she does not repay the loan, the lender loses money. Banks make loans to customers in exchange for the guarantee of repayment. Some would default on their debts, unable to repay them for a number of reasons. The bank retains insurance to minimize the possibility of failure in the case of a default. The insured sum can cover the whole loan amount or just a portion of it.

Banking processes use manual procedures to determine whether or not a borrower is suitable for a loan based on results. Manual procedures were mostly effective, but they were insufficient when there were a large number of loan applications. At that time, making a decision would take a long time. As a result, the loan prediction machine learning model can be used to assess a customer's loan status and build strategies. This model extracts and introduces the essential features of a borrower that influence the customer's loan status. Finally, it produces the planned performance (loan status). These reports make a bank manager's job simpler and quicker.

In this case, we are going to analyze the previous records of the customer related with the bank and also some personal records in order to predict. We are predicting here whether the customer will be able to pay the debt or not? So that the bank can achieve minimum risks about giving loans to its customers. We are going to use some ML techniques to solve this particular problem.

Chapter – 02

LITERATURE SURVEY

Rajiv Kumar and Vinod Jain, in their research paper [1] used the Python programming language to implement the logistic tree, decision tree, and random forest algorithms. They chose the Decision tree method as the most efficient after comparing the evaluation of three kinds of machine learning approaches in terms of prediction accuracy. They didn't fill in the blanks or properly categorize the data however, this can be fixed by filling in the blanks and properly categorizing the data. Pidikiti Supriya and Myneedi Pavani, in their research paper [2] claim to have pre-processed the data to remove inconsistencies in the dataset. They've also compiled a list of Correlating Characteristics that were found to make people more likely to repay their debts. To divide the dataset into training and testing operations, the 80:20 rule was used. The Python platform's corplot and boxplot are used to find the correlation between attributes. However, they haven't utilized any other method to compare accuracy results other than a decision tree. This may be avoided by training datasets with multiple algorithms and comparing their efficiency. Kumar Arun and Garg Ishan, in their research paper [3] tested a total of six different machine learning approaches, including neural networks, support vector machines, random forests, decision trees, linear models, and Adaboost. There are four sections to this study. (i) Gathering of data (ii) Model evaluation using ML on the collected information (iii) System training using the most feasible model (iv) After the system has been trained on the most promising model, it is put to the test. R programming language was used to create this system. They didn't represent the data results for easier comprehension and comparison, but this problem can be solved by offering data visualization in the form of graphs or other matrix forms. Authors in [4]. Initially, the information was cleansed. The next step was exploratory data analysis and feature engineering. They had done visualization through graphs. For loan prediction, four models are used. Decision Tree, Naive Bayes, Support Vector Machines, and Logistic Regression methods are the four methods. They determined confidently showing the Naive Bayes model is very capable of delivering superior results to other models after thoroughly studying positive attributes and constraints. Authors in [5] said a set of data was obtained from the banking sector. The data set is in the ARFF (AttributeRelation File Format) format, which Weka understands. They used exploratory data analysis to solve the challenge of granting or rejecting loan requests, as well as short-term loan projection. In their research, they did an exploratory data analysis. For prediction, two machine learning classification models are used Decision Tree and Random Forest. In their analysis, they chose the random forest method.

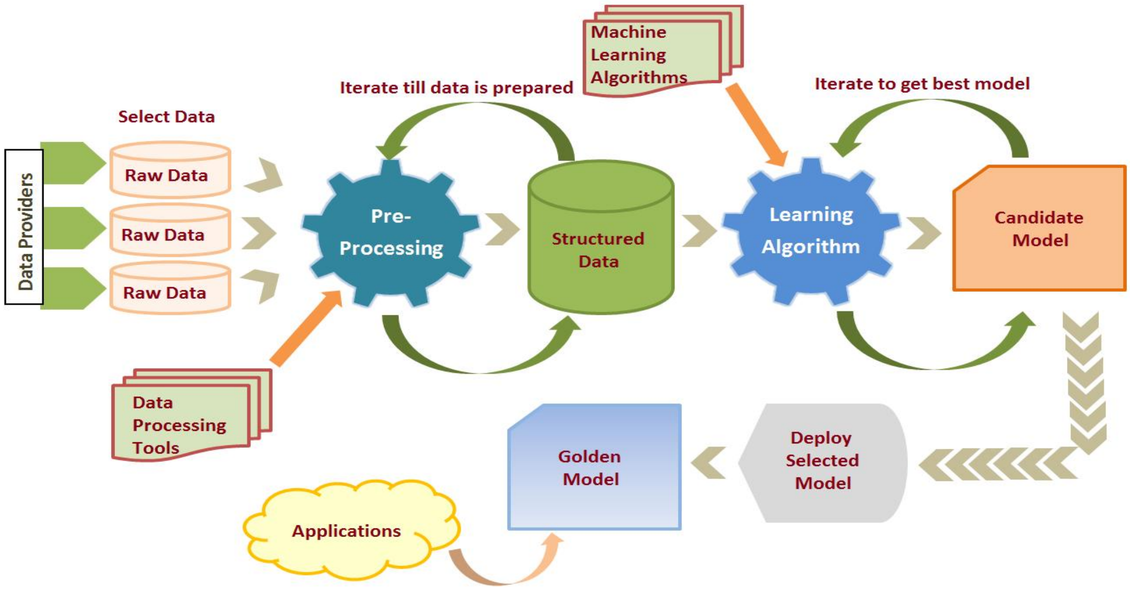
Chapter – 03

OBJECTIVE

- The objective of the problem is to pick "whether the customer is able to pay the debt or not".
- The main objective of this project is to predict whether assigning the loan to particular person will be safe or not.
- In this model we will predict the loan data by using some machine learning classification algorithms.
- So, we will build a model that will help them identify the potential customers who have a higher probability of purchasing the loan.

Chapter – 04

METHODOLOGY



Methodology used in this project: -

1. Collecting the relevant data
2. Data preprocessing
3. Applying ML algorithms
4. Choosing the best Model
5. Model deployment

Chapter – 05

DATA COLLECTION AND DATASET DESCRIPTION

We have got the loan data set through Kaggle [14]. The redundant and identical entries were deleted once the dataset was normalized. There is a chance that the data received possibly involve some null values, which could cause inconsistencies. Data must have been pre-processed to boost the algorithm's efficiency. Outliers must be eliminated, and variable conversion must be performed. The dataset gathered for forecasting loan default customers is divided into two groups: training and testing. Our data set includes a total of 13 columns. The characteristics are as follows:

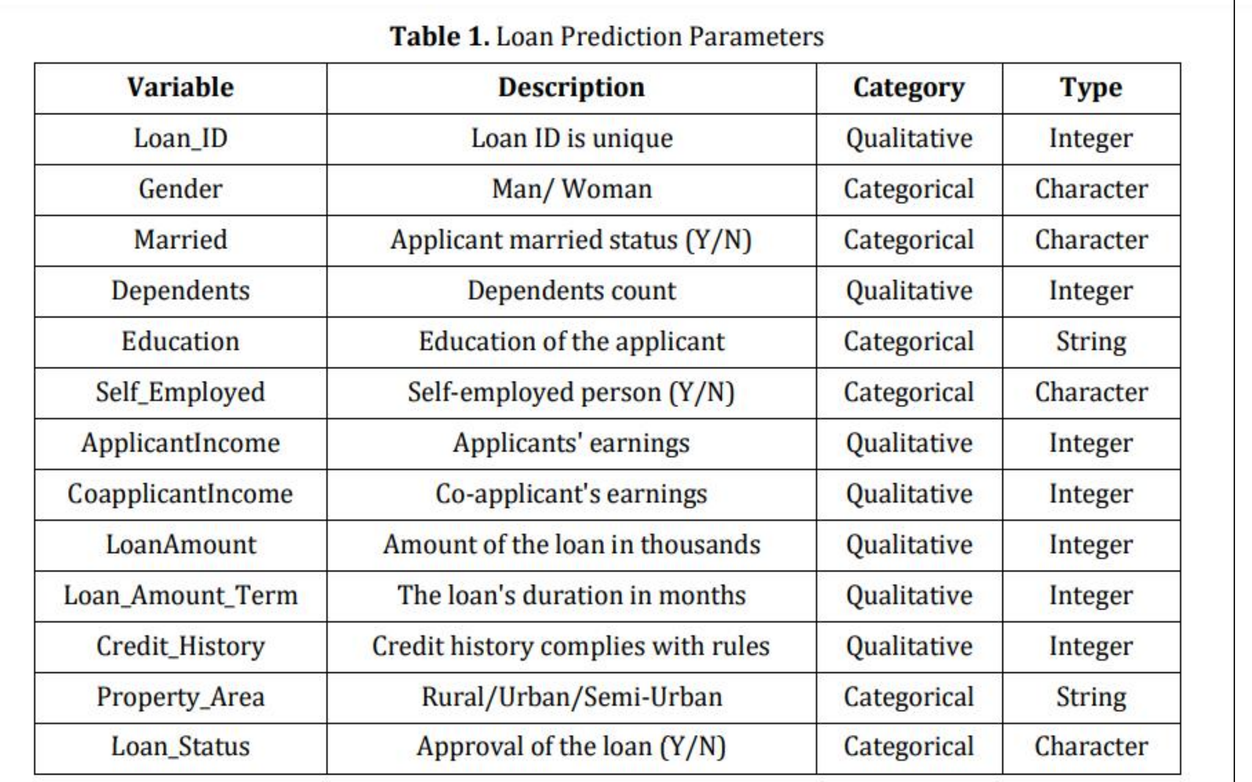


Table 1. Loan Prediction Parameters

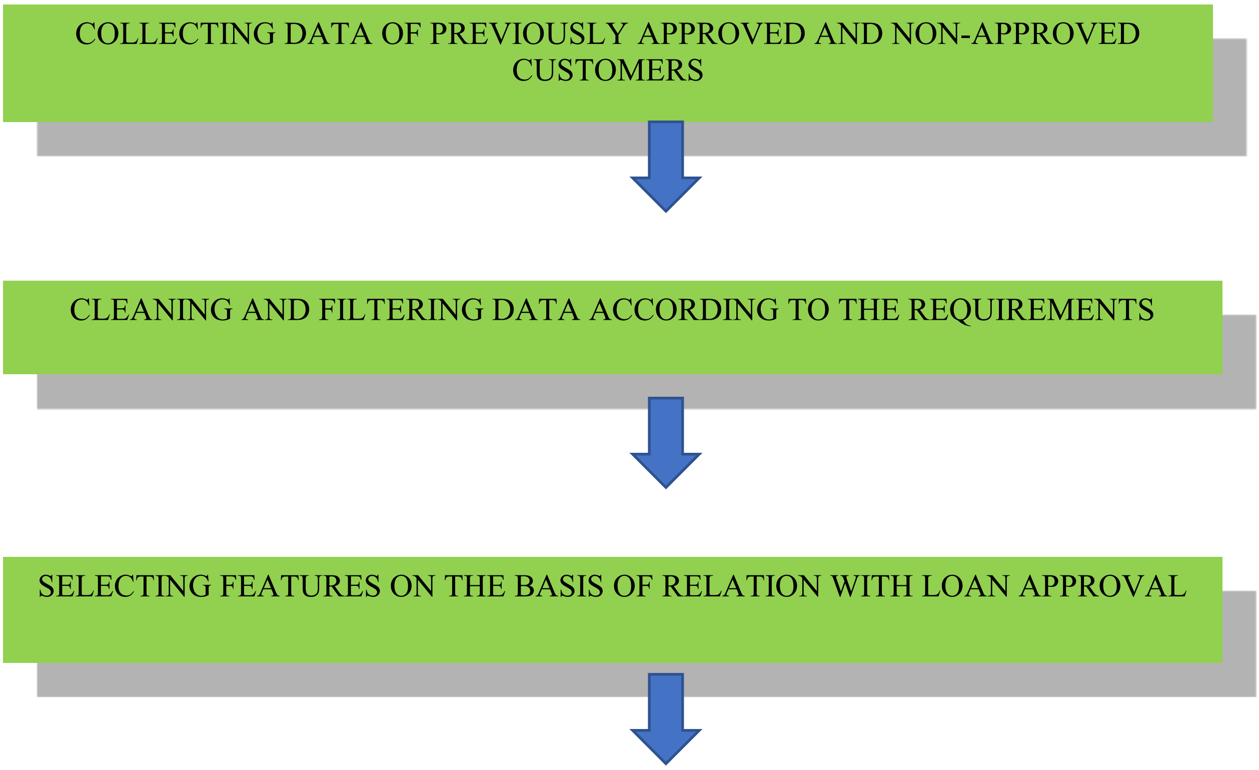
Variable	Description	Category	Type
Loan_ID	Loan ID is unique	Qualitative	Integer
Gender	Man/ Woman	Categorical	Character
Married	Applicant married status (Y/N)	Categorical	Character
Dependents	Dependents count	Qualitative	Integer
Education	Education of the applicant	Categorical	String
Self_Employed	Self-employed person (Y/N)	Categorical	Character
ApplicantIncome	Applicants' earnings	Qualitative	Integer
CoapplicantIncome	Co-applicant's earnings	Qualitative	Integer
LoanAmount	Amount of the loan in thousands	Qualitative	Integer
Loan_Amount_Term	The loan's duration in months	Qualitative	Integer
Credit_History	Credit history complies with rules	Qualitative	Integer
Property_Area	Rural/Urban/Semi-Urban	Categorical	String
Loan_Status	Approval of the loan (Y/N)	Categorical	Character

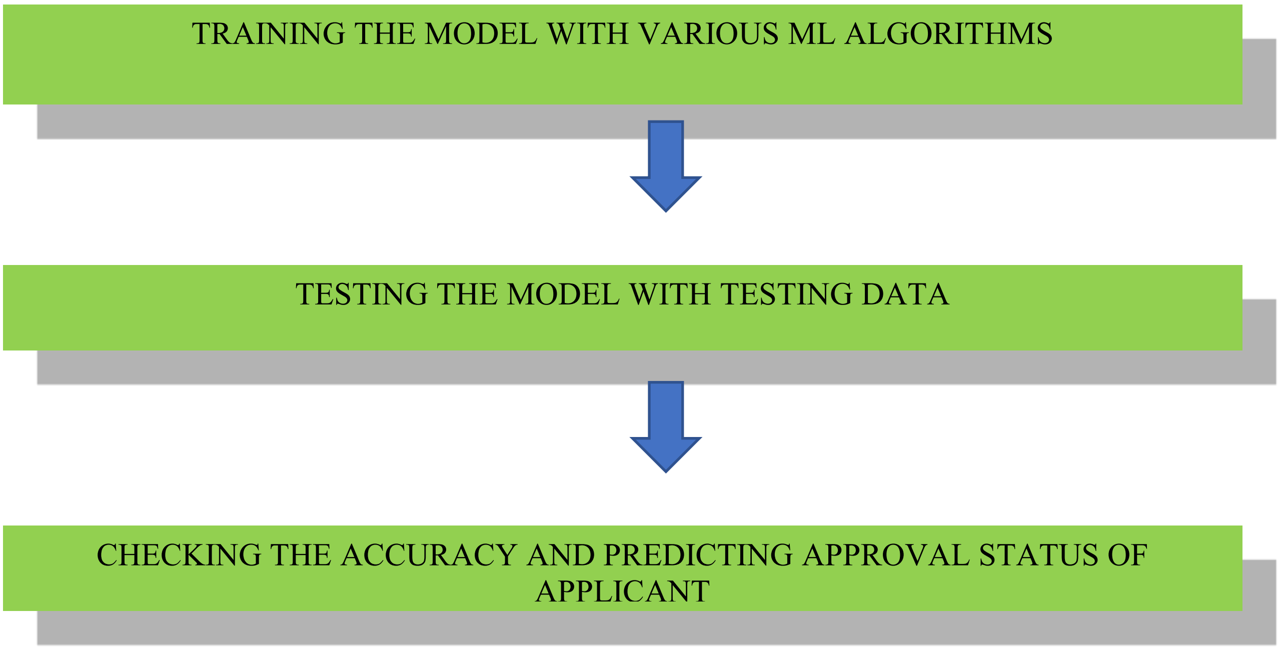
As we can see from the above table here the response variable is Loan Status, and the remaining variables/factors determine whether the loan would be approved or not. Here Loan_Status is only dependent column and rest 12 columns are independent columns.

Chapter – 06

WORKING OF THE MODEL

Based on the data provided by the borrower, an organization must automate the loan qualifying method (in real-time). Data such as Loan Amount, Gender, Marital Status, Income, Credit History, Education, Number of Dependents, and a few other details while completing a request form. As shown in Table 1. To make things simple, they created a system that allows them to identify types of applicants, who are qualified for a loan amount and approach them specifically. Since we need to classify everything before determining if the loan status is Yes or No, therefore this is considered as a classification issue. The system can quickly determine if a loan application is likely to be granted or rejected. Figure. 1, shows the working of the proposed model step by step.


COLLECTING DATA OF PREVIOUSLY APPROVED AND NON-APPROVED CUSTOMERS
CLEANING AND FILTERING DATA ACCORDING TO THE REQUIREMENTS
SELECTING FEATURES ON THE BASIS OF RELATION WITH LOAN APPROVAL


TRAINING THE MODEL WITH VARIOUS ML ALGORITHMS
TESTING THE MODEL WITH TESTING DATA
CHECKING THE ACCURACY AND PREDICTING APPROVAL STATUS OF APPLICANT

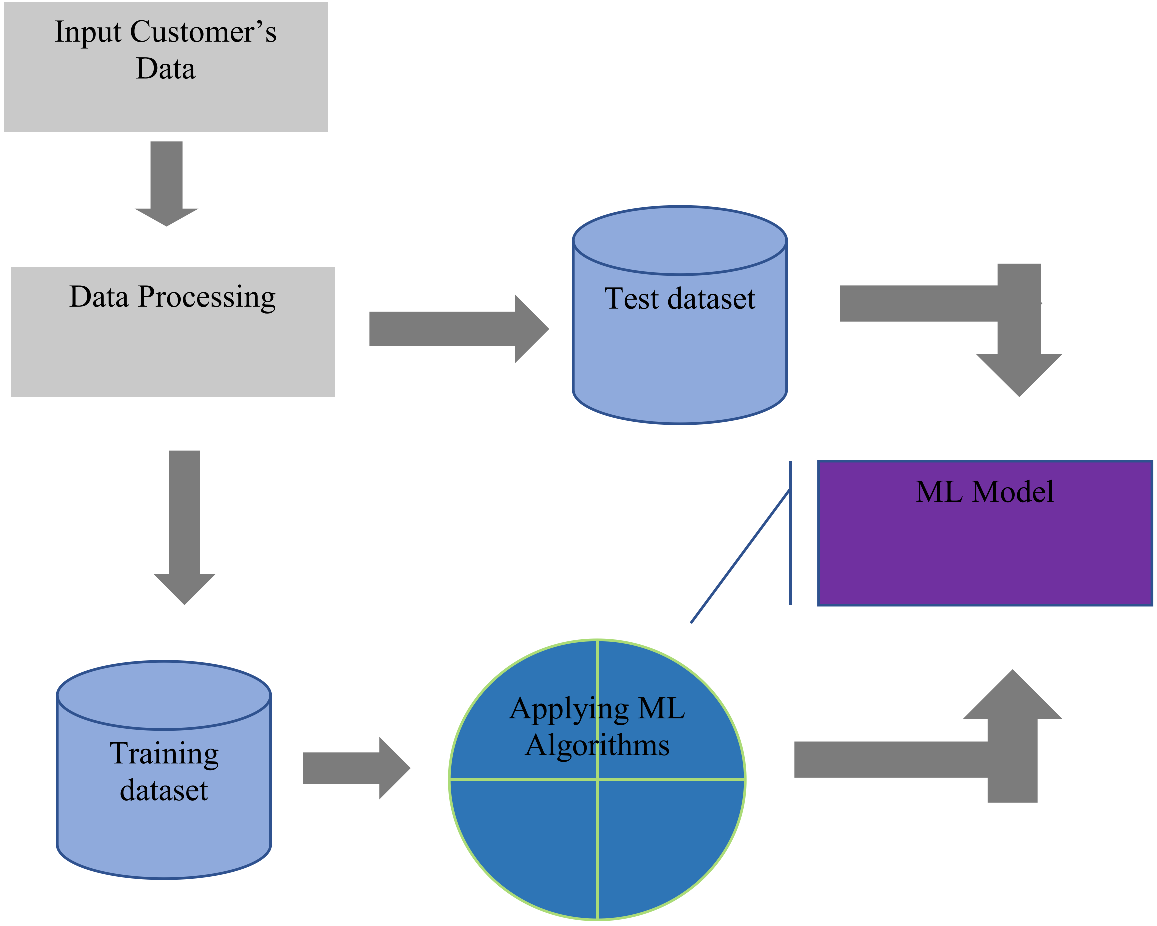
(Figure -01: Proposed Methodology**)**

Chapter – 07

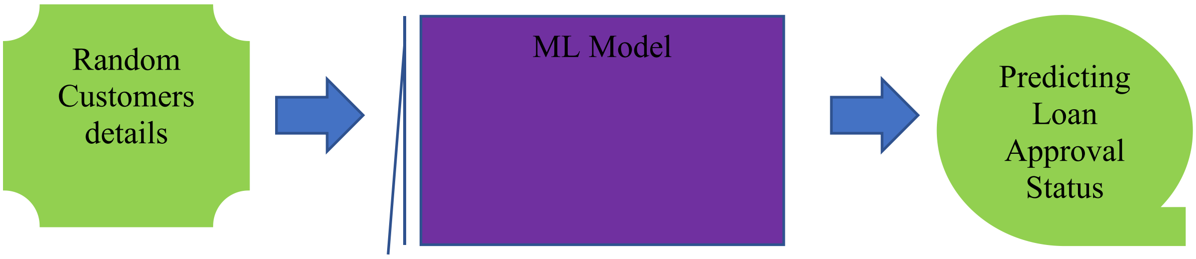
ARCHITECTURE OF THE PROPOSED MODEL

Our project makes use of a variety of algorithms to help us achieve a precise result. The Python programming language, which is among the most often used and popular languages in AI and ML because it comes with all of the necessary tools and libraries has been used in our project. It has several libraries, like pandas for the filtering process, matplotlib for plotting the data, data visualization, and exploratory data analysis. We have also used sklearn which is Scikit-learn which includes several clustering, regression, and classification algorithms that are commonly used in AI and machine learning. NumPy is used to deal with the multidimensional array and data structures.

Seaborn library has been used for data visualization. The model then applies this technique to pre-defined data set including all the information about our customers. In a linear pattern, the algorithms are executed one after the other. The data is then analyzed, segregated, and provided into the model to train it. As shown in Figure. 2. After each algorithm, the precision rate is displayed. We have trained our model with many algorithms to get a precise result. The Random Forest Algorithm, Decision Tree Algorithm, Logistic Regression Algorithm, SVM, and K Neighbors algorithm will all be used, with an 90% training set and a 10% testing set. We have discovered that logic regression, SVM, and Naive Bayes have superior precision. Following the testing procedure, the model predicts if the current candidate based on the conclusion is a good candidate for getting a loan acceptance, it draws from the training data sets. As a result, the better we are in determining the capable borrower, the more beneficial it is to the organization.



(Figure -02: Proposed Model's Architecture**)**



(Figure -03: Proposed Model's Implementation**)**

Chapter - 08

MACHINE LEARNING AND CONCEPTS

SUPPORT VECTOR MACHINE ALGORITHM

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.

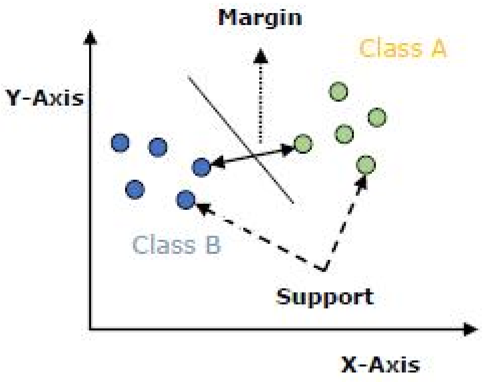
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

SVM can be of two types:

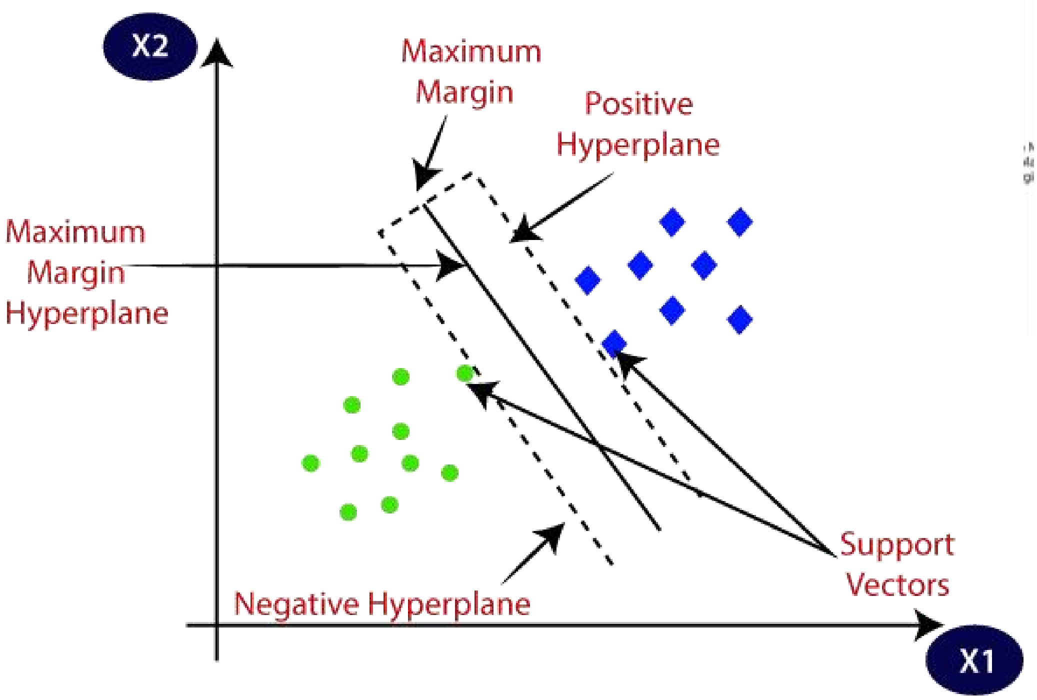
- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Working of SVM

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).



Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Terms related to SVM

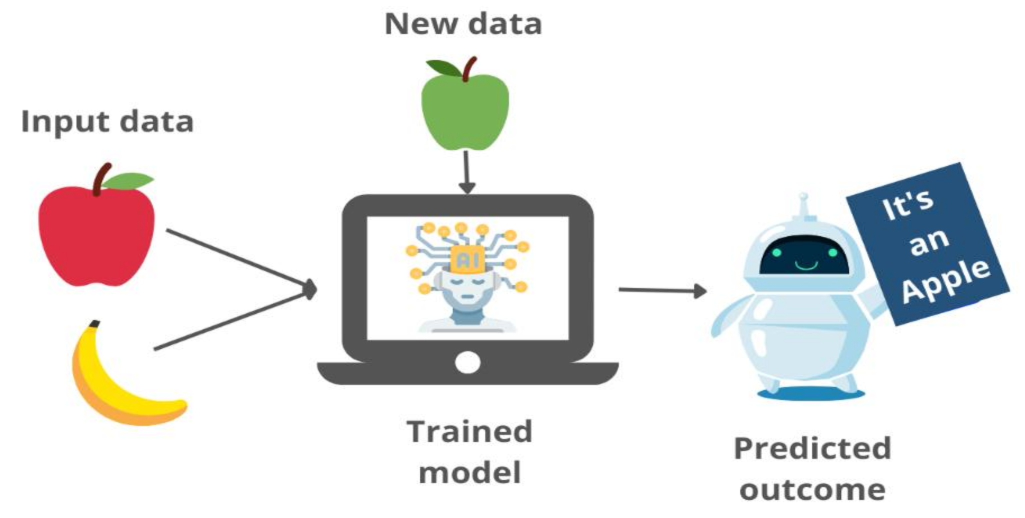
Support Vectors − Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.

Hyperplane − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

Example: -

As we can see in the below figure there is two types of labelled data i.e. apple and banana feeding to the algorithm. So when a new data will be given to it , it will predict whether the new data is an apple or banana.



Fitting the SVM classifier to the training set:

Now the training set will be fitted to the SVM classifier. To create the SVM classifier, we will import **SVC** class from **Sklearn.svm** library. Below is the code for it:

```
from sklearn.svm import SVC
classifier = SVC(kernel='linear', random_state=0)
classifier.fit(x_train, y_train)
```

In the above code, we have used **kernel='linear'**, as here we are creating SVM for linearly separable data. However, we can change it for non-linear data. And then we fitted the classifier to the training dataset(x_train, y_train)

Predicting the test set Results :

Now, we will predict the output for test set. For this, we will create a new vector y_pred.

Below is the code for it:

```
y_pred= classifier.predict(x_test)
```

Pros and Cons of SVM Classifier

Pros of SVM classifiers

SVM classifiers offers great accuracy and work well with high dimensional space. SVM classifiers basically use a subset of training points hence in result uses very less memory.

Cons of SVM classifiers

They have high training time hence in practice not suitable for large datasets. Another disadvantage is that SVM classifiers do not work well with overlapping classes.

LOGISTIC 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 predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

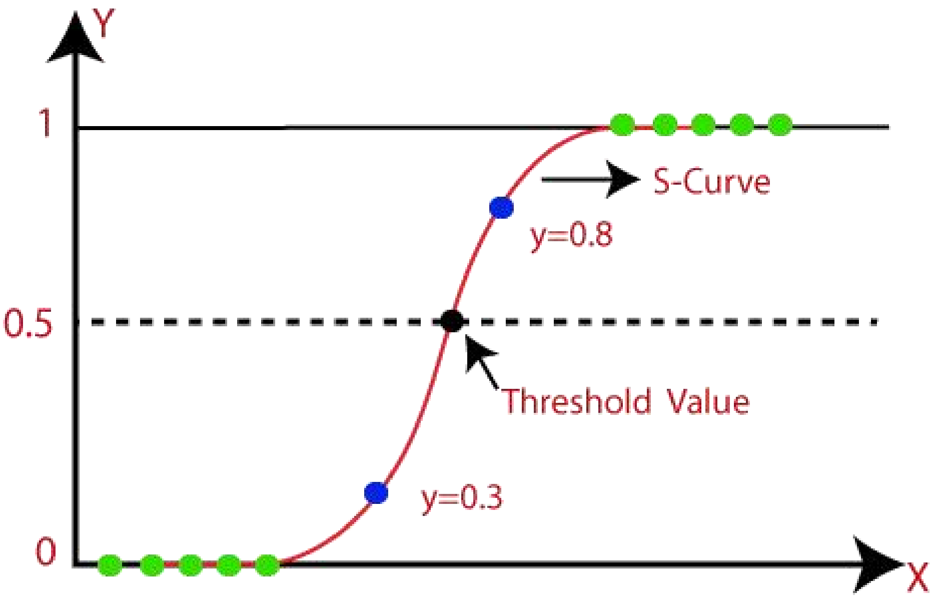
Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



Types of Logistic Regression

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types –

Binary or Binomial

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

Multinomial

In such a kind of classification, dependent variable can have 3 or more possible **unordered** types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

Ordinal

In such a kind of classification, dependent variable can have 3 or more possible **ordered** types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

DECISION TREE

Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. 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 decisions or the test are performed on the basis of features of the given dataset.

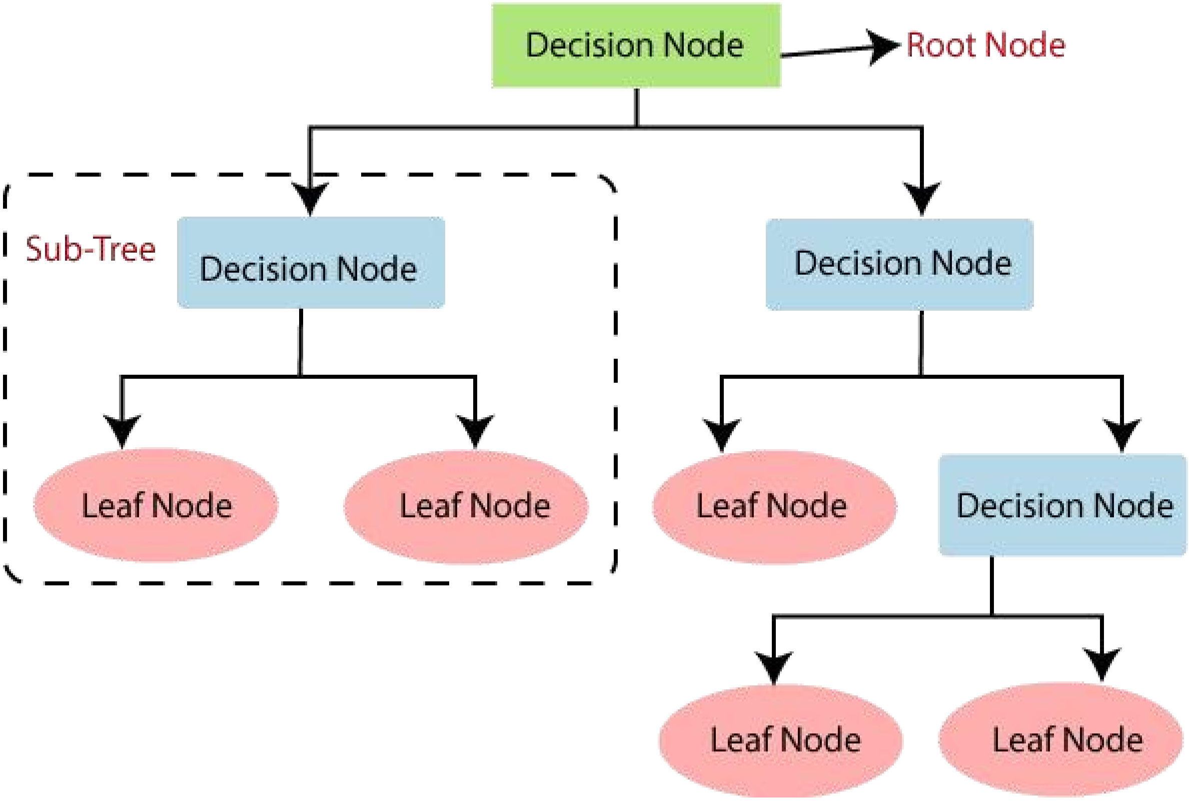
It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**

A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Below diagram explains the general structure of a decision tree:



Types of decision Tree

Classification decision trees − In this kind of decision trees, the decision variable is categorical. The above decision tree is an example of classification decision tree.

Regression decision trees − In this kind of decision trees, the decision variable is continuous.

NAIVE BAYES CLASSIFIER

Naive Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.

It is mainly used in text classification that includes a high-dimensional training dataset.

Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Some popular examples of Naive Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Why is it called Naive Bayes?

The Naive Bayes algorithm is comprised of two words Naïve and Bayes, which can be described as:

- **Naive**: It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes Theorem.

Bayes Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as: P(A|B) = [P(B|A) P(A)] / P(B)

Where, 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.

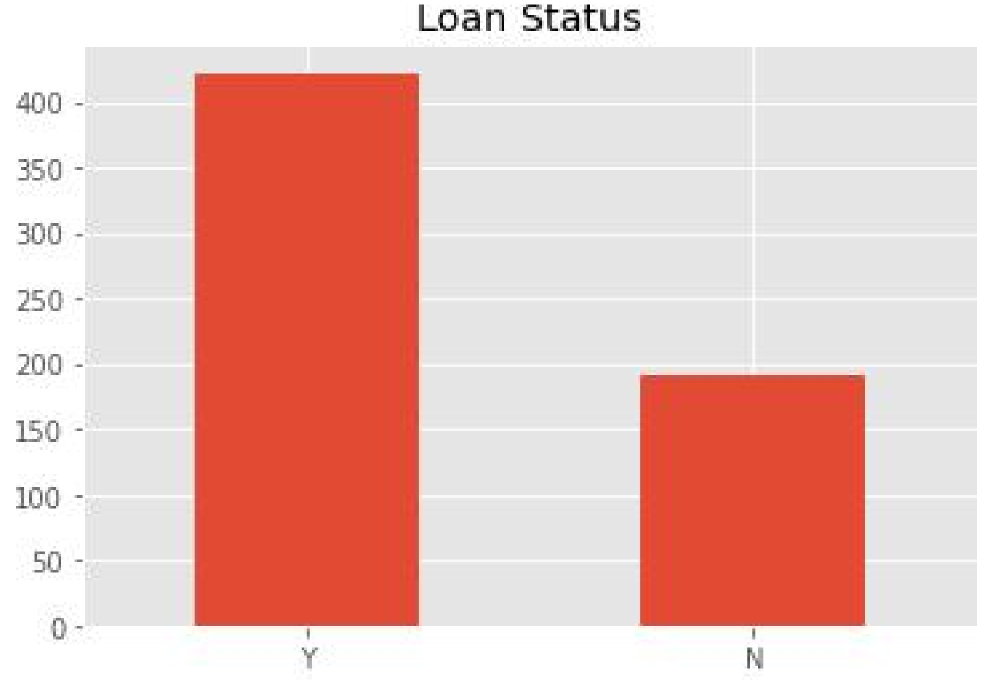
P(B) is Marginal Probability: Probability of Evidence.

Chapter - 09

EXPLARATORY DATA ANALYSIS

UNIVARIATE ANALYSIS

Here we can consider Loan status as target variable. We will start first with an independent variable which is our target variable as well. We will analyze this categorical variable using a bar chart as shown below. The bar chart shows that loan of 422 (around 69 %) people out of 614 was approved.



PREDICTOR VARIABLE

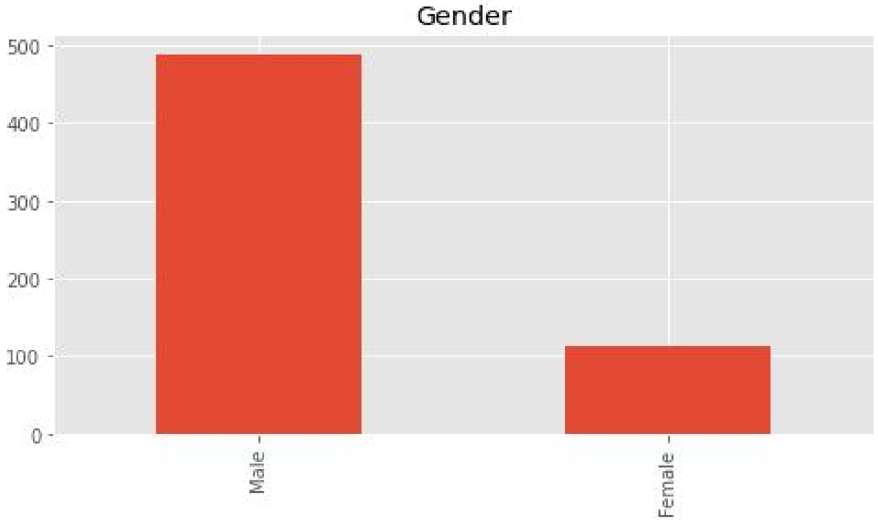
There are 3 types of Independent Variables: Categorical, Ordinal & Numerical.

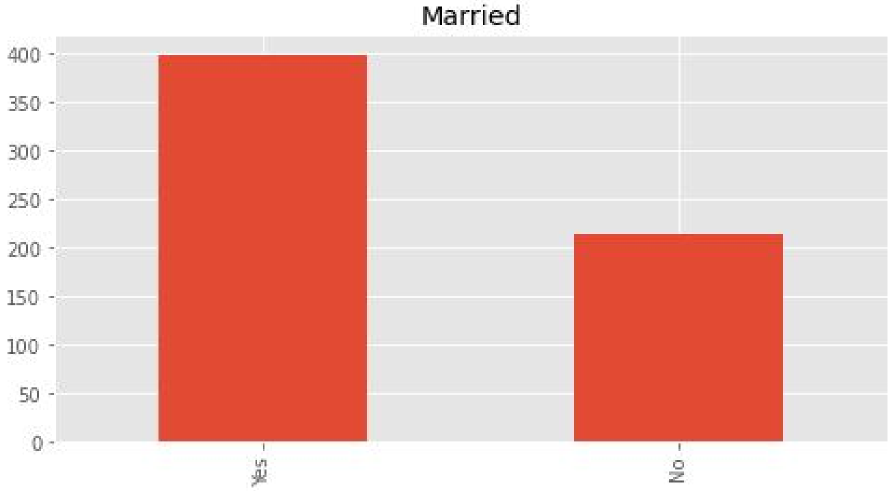
Categorical Features

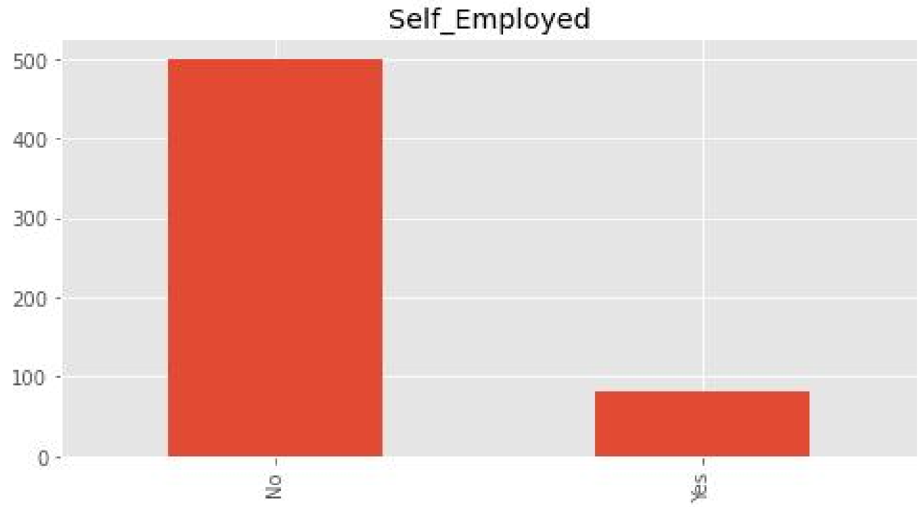
- Gender
- Marital Status
- Employment Type
- Credit History

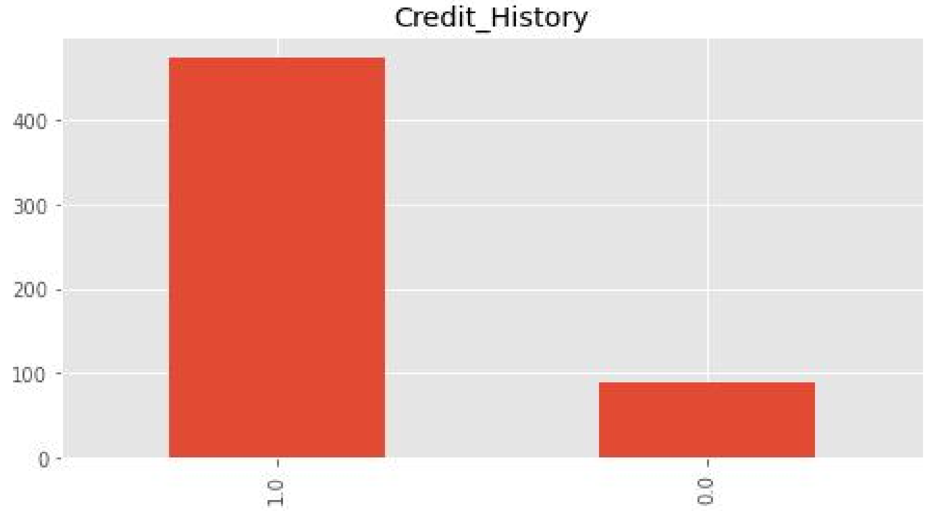
It can be observed from the below bar plots that in our observed data

1. 80% of loan applicants are male
2. Nearly 70% are married
3. 75% of loan applicants are graduates
4. Nearly 85-90% loan applicants are self-employed
5. The loan has been approved for more than 65% applicants







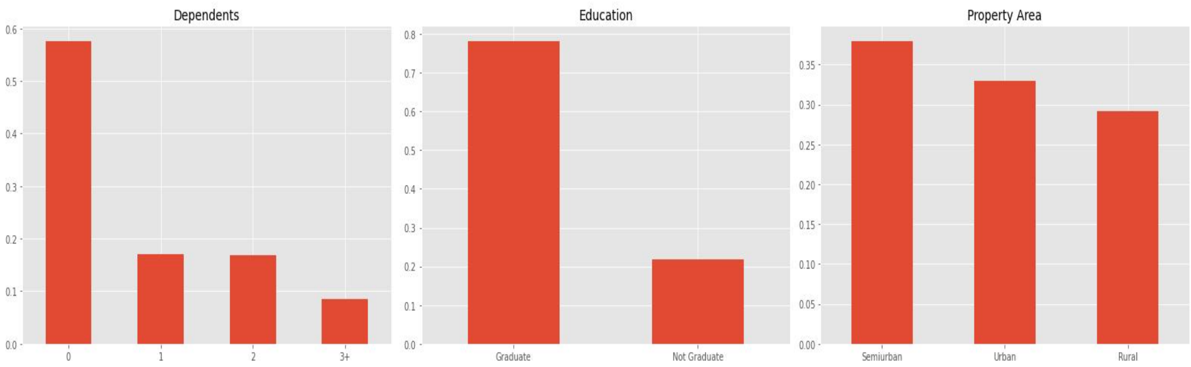


Ordinal Features

- Number of Dependents
- Education Level
- Property or Area Background

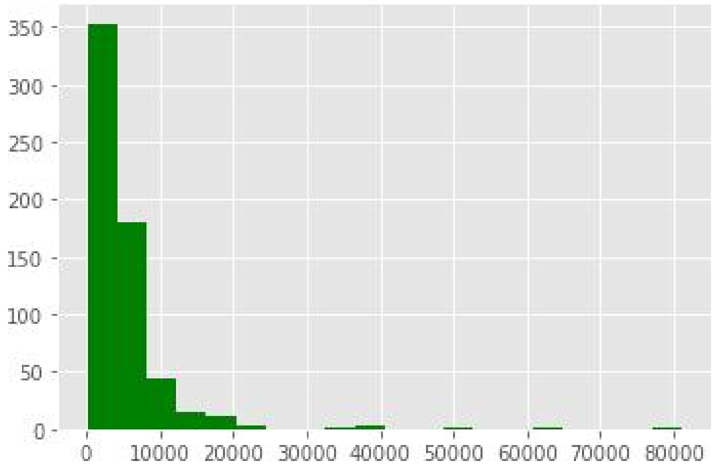
From the visual Analysis we can clearly understand that

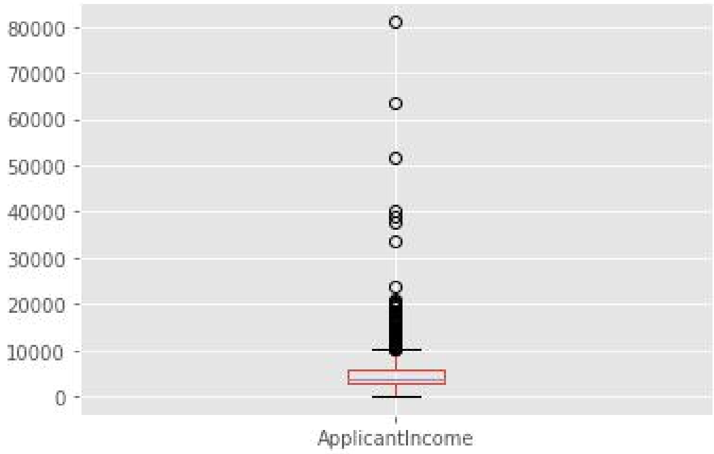
1. Almost 58% of the applicants have no dependents
2. Around 80% applicants are graduate
3. Highest number of applicants are from semi-urban area, followed by urban areas



Numerical Features

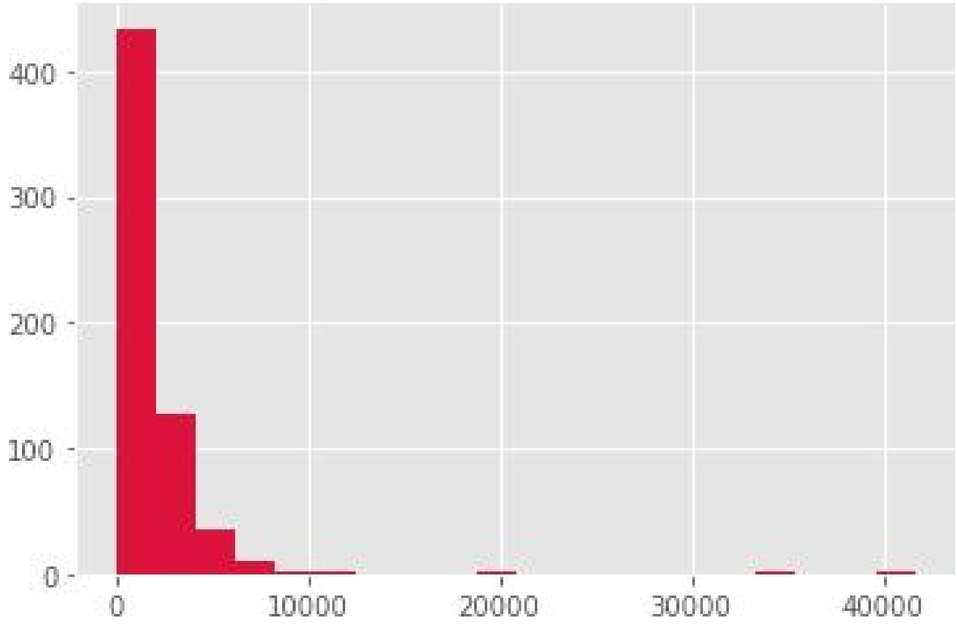
- The applicant’s income

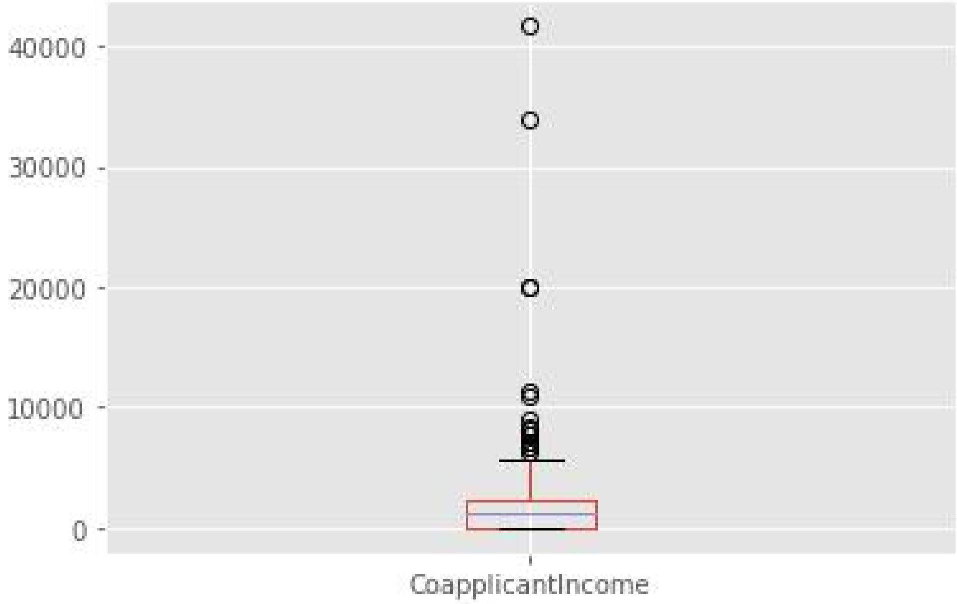




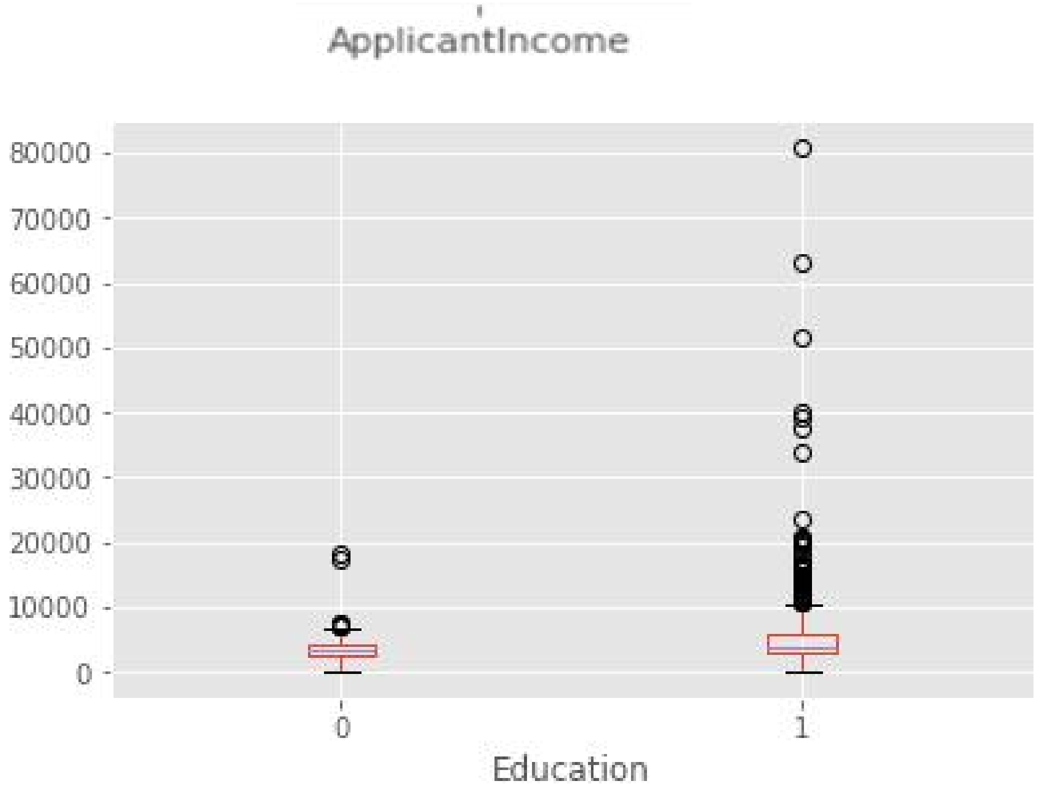
It can be inferred that most of the data in Applicant income is towards left which means it is not normally distributed. The boxplot confirms the presence of outliers. This can be attributed to income disparity in the society.

- The co-applicant's income





Co-applicant’s Income is lesser than applicant’s Income and is within the 5000–15000, again with some outliers.



We can see that there are higher number of graduates with very high incomes, which are appearing to be the outliers.

Chapter – 10

DATA PREPROCESSING

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

Data Cleaning

Data cleaning is particularly done as part of data pre-processing to clean the data by filling missing values, smoothing the noisy data, resolving the inconsistency, and removing outliers.

1. Missing values

Here are a few ways to solve this issue:

- Ignore those tuples

This method should be considered when the dataset is huge and numerous missing values are present within a tuple.

- Fill in the missing values

There are many methods to achieve this, such as filling in the values manually, predicting the missing values using regression method, or numerical methods like attribute mean.

Handling Missing data

Handling missing data is very important as many machine learning algorithms do not support data with missing values. If you have missing values in the dataset, it can cause errors and poor performance with some machine learning algorithms.

Here is the list of common missing values you can find in your dataset.

- N/A
- null
- Empty
- ?
- none
- empty
- -
- NaN

Mean/Median Imputation

This one of the common techniques is to use the mean or median of the non-missing observations. This strategy can be applied to a feature that has numeric data.

```
# filling missing values with medians of the columns
data = data.fillna(data.median())
```

Most Common value imputation

This method is replacing the missing values with the maximum occurred value in a column/feature. This is a good option for handling categorical columns/features.

```
# filling missing values with medians of the columns
```

```
data['column_name'].fillna(data['column_name'].value_counts().idxmax().inplace=True)
```

2.Noisy Data

It involves removing a random error or variance in a measured variable. It can be done with the help of the following techniques:

- Binning

It is the technique that works on sorted data values to smoothen any noise present in it. The data is divided into equal-sized bins, and each bin/bucket is dealt with independently. All data in a segment can be replaced by its mean, median or boundary values.

- Regression

This data mining technique is generally used for prediction. It helps to smoothen noise by fitting all the data points in a regression function. The linear regression equation is used if there is only one independent attribute; else Polynomial equations are used.

- Clustering

Creation of groups/clusters from data having similar values. The values that don't lie in the cluster can be treated as noisy data and can be removed.

3.Removing outliers

Clustering techniques group together similar data points. The tuples that lie outside the cluster are outliers/inconsistent data.

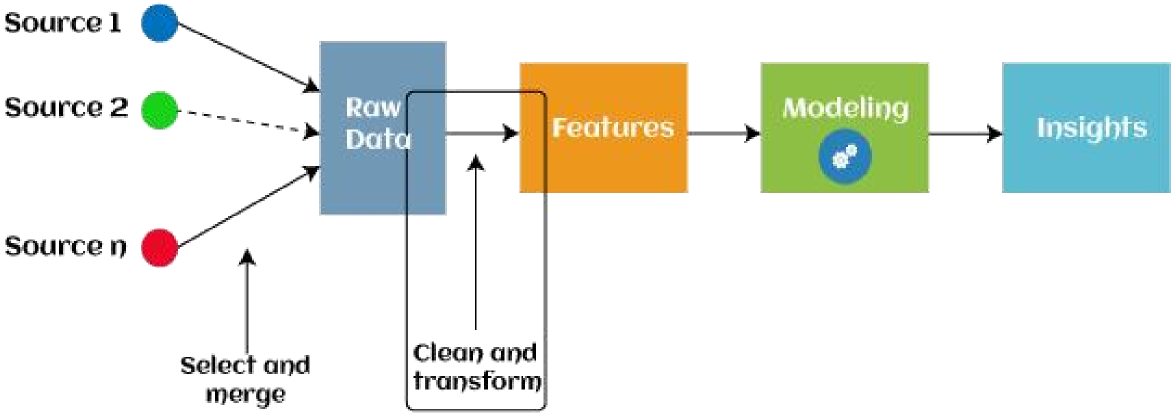
Chapter – 11

FEATURE ENGINEERING AND FEATURE SELECTION

Feature engineering is the process of selecting variable/features when creating a predictive model.

Feature engineering has two goals:

- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the **performance** of machine learning models.



Feature Creation: Feature creation is finding the most useful variables to be used in a predictive model. The process is subjective, and it requires human creativity and intervention. The new features are created by mixing existing features using addition, subtraction, and ration, and these new features have great flexibility.

Feature Selection: Feature selection is a way of selecting the subset of the most relevant features from the original features set by removing the redundant, irrelevant, or noisy features.

Feature Extraction: Feature extraction is an automated feature engineering process that generates new variables by extracting them from the raw data. The main aim of this step is to reduce the volume of data so that it can be easily used and managed for data modelling.

Chapter – 12

MODEL TRAINING AND EVALUTION

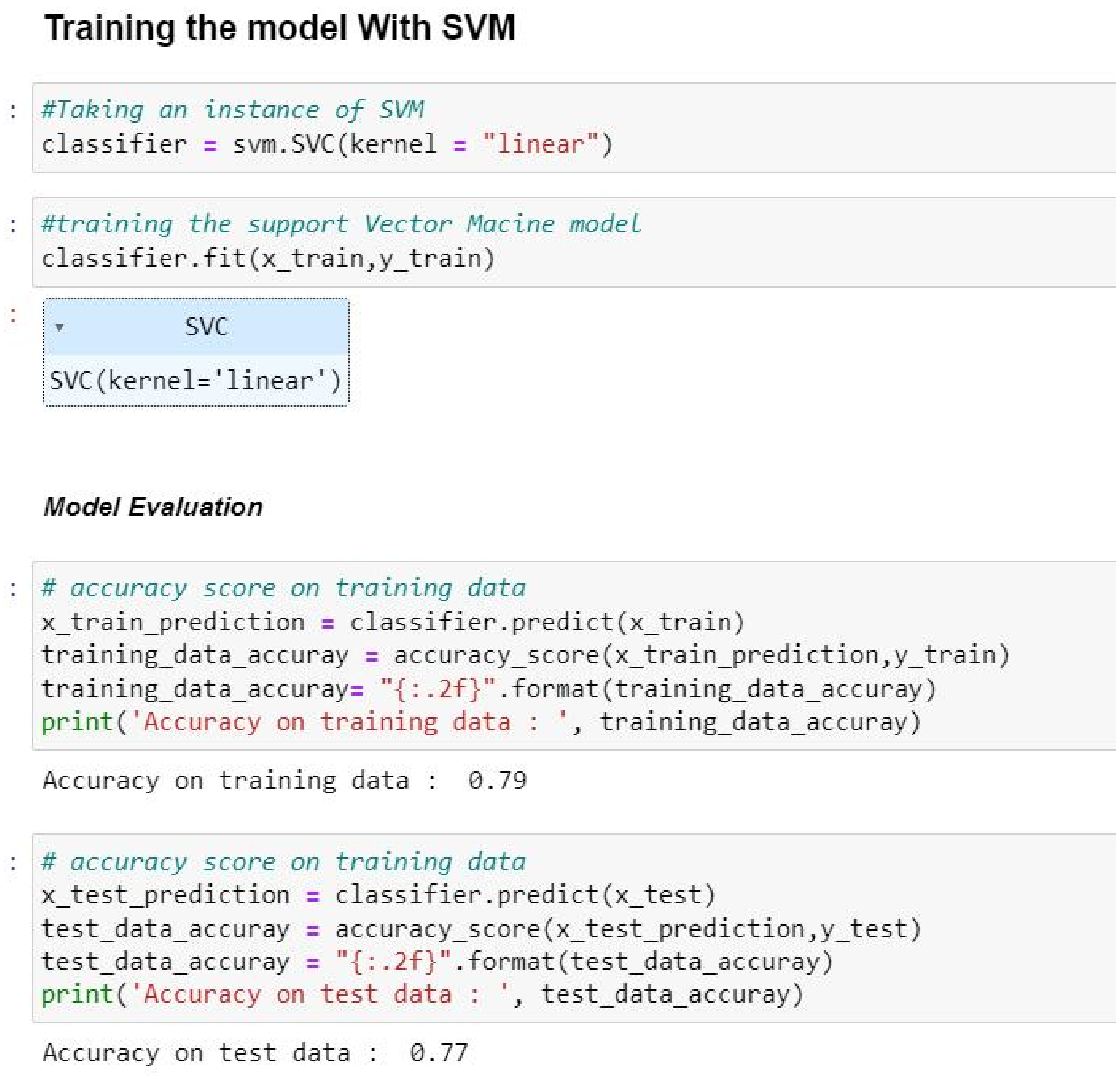
Machine Learning Model does not require hard-coded algorithms. We feed a large amount of data to the model and the model tries to figure out the features on its own to make future predictions.

Here we have used four ML algorithm to make our predictive model

- SVM
- Logistic Regression
- Decision Tree
- Naïve Baye’s

After testing the accuracy of training and testing dataset we got maximum accuracy in Logistic regression. So, it can be considered as final model.

RESULTS AND SCREENSHOTS



Training the model With SVM

```
#Taking an instance of SVM
classifier = svm.SVC(kernel = "linear")
```

```
#training the support Vector Macine model
classifier.fit(x_train,y_train)
```

```
▾ SVC
SVC(kernel='linear')
```

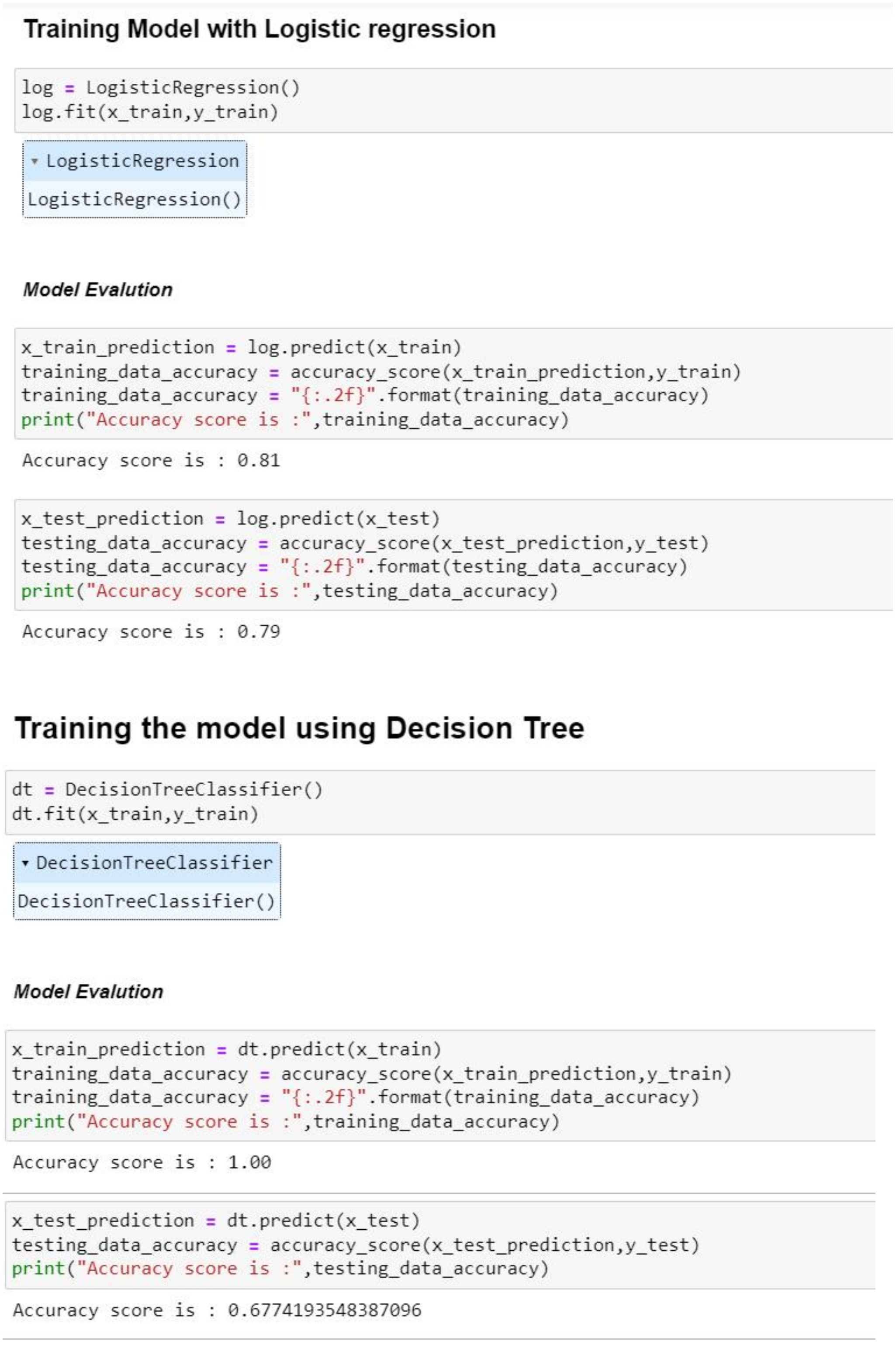
Model Evaluation

```
# accuracy score on training data
x_train_prediction = classifier.predict(x_train)
training_data_accuray = accuracy_score(x_train_prediction,y_train)
training_data_accuray= "{:.2f}".format(training_data_accuray)
print('Accuracy on training data : ', training_data_accuray)
```

```
Accuracy on training data :  0.79
```

```
# accuracy score on training data
x_test_prediction = classifier.predict(x_test)
test_data_accuray = accuracy_score(x_test_prediction,y_test)
test_data_accuray = "{:.2f}".format(test_data_accuray)
print('Accuracy on test data : ', test_data_accuray)
```

```
Accuracy on test data :  0.77
```



Training Model with Logistic regression

```
log = LogisticRegression()
log.fit(x_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

Model Evalution

```
x_train_prediction = log.predict(x_train)
training_data_accuracy = accuracy_score(x_train_prediction,y_train)
training_data_accuracy = "{:.2f}".format(training_data_accuracy)
print("Accuracy score is :",training_data_accuracy)
```

```
Accuracy score is : 0.81
```

```
x_test_prediction = log.predict(x_test)
testing_data_accuracy = accuracy_score(x_test_prediction,y_test)
testing_data_accuracy = "{:.2f}".format(testing_data_accuracy)
print("Accuracy score is :",testing_data_accuracy)
```

```
Accuracy score is : 0.79
```

Training the model using Decision Tree

```
dt = DecisionTreeClassifier()
dt.fit(x_train,y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

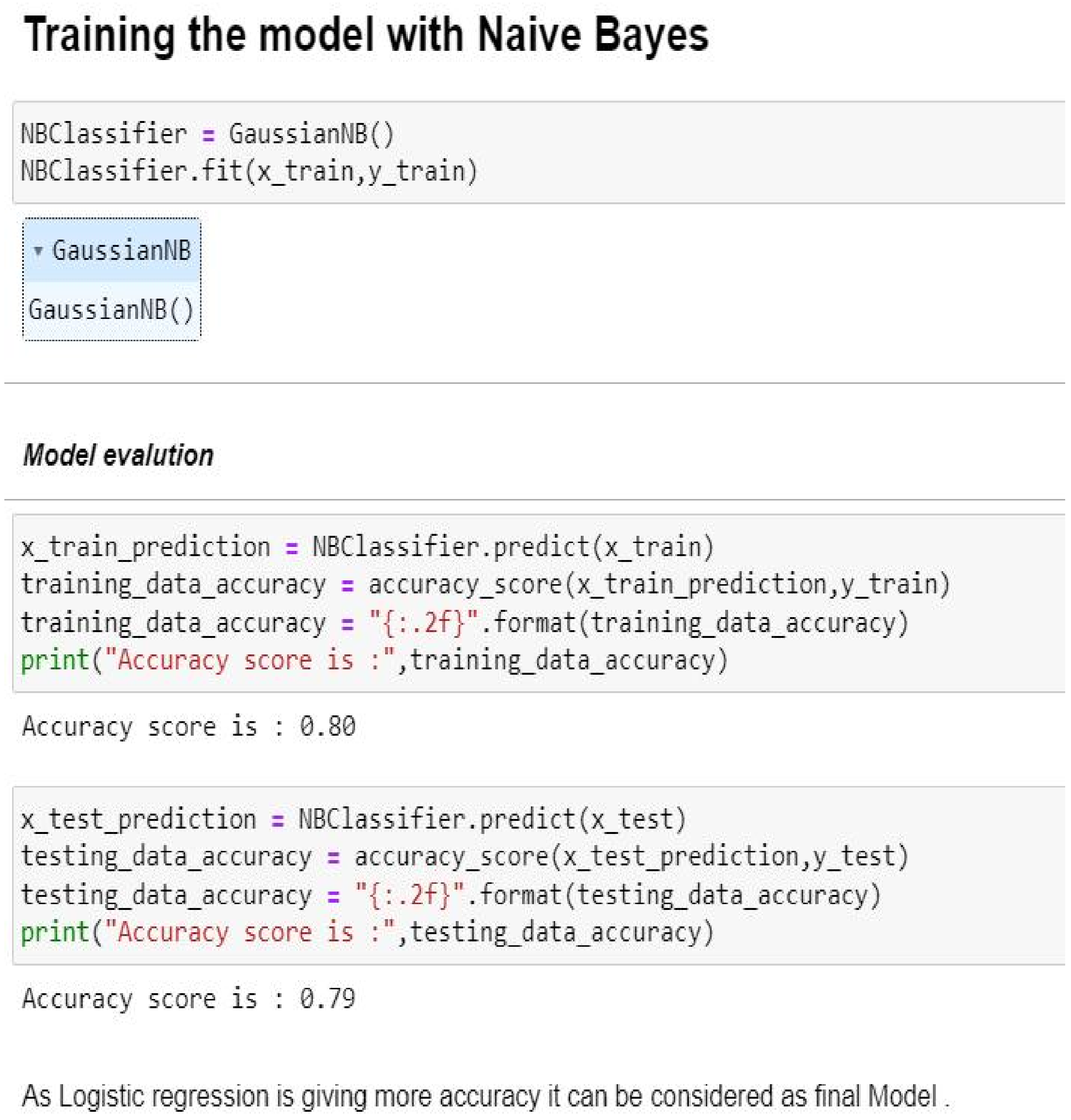
Model Evalution

```
x_train_prediction = dt.predict(x_train)
training_data_accuracy = accuracy_score(x_train_prediction,y_train)
training_data_accuracy = "{:.2f}".format(training_data_accuracy)
print("Accuracy score is :",training_data_accuracy)
```

```
Accuracy score is : 1.00
```

```
x_test_prediction = dt.predict(x_test)
testing_data_accuracy = accuracy_score(x_test_prediction,y_test)
print("Accuracy score is :",testing_data_accuracy)
```

```
Accuracy score is : 0.6774193548387096
```



Training the model with Naive Bayes

```
NBClassifier = GaussianNB()
NBClassifier.fit(x_train,y_train)
```

```
▾ GaussianNB

GaussianNB()
```

Model evalution

```
x_train_prediction = NBClassifier.predict(x_train)
training_data_accuracy = accuracy_score(x_train_prediction,y_train)
training_data_accuracy = "{:.2f}".format(training_data_accuracy)
print("Accuracy score is :",training_data_accuracy)
```

```
Accuracy score is : 0.80
```

```
x_test_prediction = NBClassifier.predict(x_test)
testing_data_accuracy = accuracy_score(x_test_prediction,y_test)
testing_data_accuracy = "{:.2f}".format(testing_data_accuracy)
print("Accuracy score is :",testing_data_accuracy)
```

```
Accuracy score is : 0.79
```

As Logistic regression is giving more accuracy it can be considered as final Model .

Chapter – 13

DEPLOYMENT

This system can be used for making a predictive system so that it can predict whether the loan will approved or not in future .

Steps to making a predictive system:

Taking input

```
Input= ("LP001091","Male","Yes",1,"Graduate","No",4166,3369,201,360,0,"Urban")
ipt = list(input)
```

Removing non-required columns i.e., Loan_Amount_Term and Loan_ID

```
del ipt[0]
del ipt[8]
```

Converting into numerical columns

```
if ipt[0]=="Male":
   ipt[0]=1
else:
   ipt[0]=0
if ipt[1]=="Yes":
   ipt[1]=1
else:
   ipt[1]=0
if ipt[3]=="Graduate":
   ipt[3]=1
else:
   ipt[3]=0
if ipt[4]=="Yes":
   ipt[4]=1
else:
    ipt[4]=0
if ipt[9]=="Urban":
    ipt[9]=2
elif ipt[9]=="Semiurban":
    ipt[9]=1
else:
    ipt[9]=0
```

Changing input to NumPy array

```
input_array = np.asarray(ipt)
```

Rreshaping the array as we predicting for only one instance

```
reshape_data = input_array.reshape(1,-1)
```

Standardizing the input data

```
scaler = StandardScaler()
#fitting the all data to scalar
scaler.fit(reshape_data)
standardize_data = scaler.transform(reshape_data)
```

Final prediction

```
pred = log.predict(reshape_data)
```

Decision making System

```
if(pred[0]==1):
    print("Loan Approved")
else:
    print("Loan not Approved")
```

Chapter – 14

CONCLUSION

This system would be able to determine the status of the loan whether it would get approved or denied swiftly in real-time. Displays accuracy with various algorithms. We have compared the Logistic regression algorithm to three other algorithms, SVM, and decision tree and Naïve Baye's. However, of all the algorithms, Logistic regression has the highest accuracy. Also, it can fill the missing values of the datasets, treat categorical values, scalability problems, overfitting problems, and provide a good visualization of the data using a confusion matrix. Applicants who have a poor credit history are likely to be rejected, especially to the risk of not repaying the loan. Applicants with high income who request low-interest loans have a stronger chance of being accepted, which is logical because they have a strong chance to repay their debts. Few essential characteristics, such as marital status and gender, appear to be overlooked by the organization, but the number of dependents is taken into consideration. The libraries are used professionally and are sufficient for now because we chose the Python programming language, but many aspects require additional exploration. Many areas of our project are left unexplored and might be studied and explored further.

For further research, applicants' Age, past health records, as well as the type of occupation they have will be utilized to evaluate the ambiguity factor of paying debts, and possible defaults of corporate loans for businesses and startups can be forecasted. Another method could be developed to forecast defaulters on different types of loans as well. We used a medium-sized data set to train our model, which may have influenced the outcome; therefore, a big and well-defined data set is required for more accurate results. This paperwork could be expanded to a higher level in the future, allowing the software to be improved to make it more dependable, secure, and accurate.

Chapter – 15

REFERENCES

[1] Rajiv Kumar, Vinod Jain, Prem Sagar Sharma- Prediction of Loan Approval using Machine LearningInternational Journal of Advanced Science and Technology Vol. 28, No. 7, (2019).

[2] Pidikiti Supriya, Myneedi Pavani, Nagarapu Saisushma- Loan Prediction by using Machine Learning. International Journal of Engineering and Techniques - Volume 5 Issue 2, Mar-Apr 2019

[3] Kumar Arun, Garg Ishan, Kaur Sanmeet- Loan Approval Prediction based on Machine Learning Approach- IOSR Journal of Computer Engineering, p-ISSN: 2278-8727 PP 18-21.

[4] E. Chandra Blessie, R. Rekha- Exploring the Machine Learning Algorithm for Prediction the Loan Sanctioning Process- (IJITEE) ISSN: 2278-3075, Volume-9 Issue-1, November 2019.

[5] Kshitiz Gautam, Arun Pratap Singh, Keshav Tyagi - Loan Prediction using Decision Tree and Random Forest- (IRJET) Volume: 07 Issue: 08 | Aug 2020.

[6] Sujoy Barua, Divya Gavandi- Swindle: Predicting the Probability of Loan Defaults using CatBoost Algorithm- ICCMC 2021.

[7] Bhoomi Patel, Harshal Patil, Jovita Hembram- Loan Default Forecasting using Data Mining- (INCET) Belgaum, India. Jun 2020.

[8] Sourav Kumar, Amit Kumar Goel- Prediction of Loan Approval using Machine Larning TechniqueInternational Journal of Advanced Science and Technology Vol. 29, pp. 4152 – 4161 (2020).

[9] Soni PM, Varghese Paul- Algorithm For the Loan Credibility Prediction System-(IJRTE) Volume-8, June 2019.

[10] X. Francis Jency, V.P.Sumathi, Janani Shiva Sri-An Exploratory Data Analysis for Loan Prediction Based on Nature of the Clients, -(IJRTE) Vol. 7, pp. 176-179, No. 48, 2018.

[11] S. Vimala, K. C. Sharmili - Prediction of Loan Risk using NB and Support Vector Machinel, vol. 4, no. 2, pp. 110-113 (ICACT 2018).

[12] K. Ulaga Priya, S. Pushpa- Exploratory analysis on prediction of loan privilege for customers using random forest- International Journal of Engineering & Technology, 7 (2018) 339-341.

[13] Deepak Ishwar Gouda, Ashok Kumar, Anil Manjunatha Madivala-Loan Approval Prediction Based On Machine Learning- e-ISSN: 2395-0056 Volume: 08 Issue: 01 | Jan 2021 (IRJET).

[14] https://www.kaggle.com/burak3ergun/loan-data-set

[15] A. Ibrahim, R. L., M. M., R. O. and G. A., "Comparison of the CatBoost Classifier with other Machine Learning Methods", International Journal of Advanced Computer Science and Applications, vol. 11, no. 11, 2020. Available: 10.14569/ijacsa.2020.0111190.

[16] J. Tejaswini, M. Kavya, D. Ramya, S. Triveni and V. Rao Maddumala, "ACCURATE LOAN APPROVAL PREDICTION BASED ON MACHINE LEARNING APPROACH", Journal of Engineering Sciences, vol. 11, no. 0377-9254, 2020.

[17] A. Gupta, V. Pant, S. Kumar and P. Kumar Bansal, "Bank Loan Prediction System using Machine Learning", Ieeexplore.ieee.org, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9336801. [Accessed: 08- Apr- 2022].

[18] A. Vaidya, "Predictive and probabilistic approach using logistic regression: Application to prediction of loan approval", Ieeexplore.ieee.org, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/8203946. [Accessed: 08- Apr- 2022].

[19] M. Ahmad Sheikh, A. Kumar Goel and T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm", Ieeexplore.ieee.org, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9155614/authors#authors. [Accessed: 08- Apr- 2022].

[20] V. Singh, A. Yadav and R. Awasthi, "Prediction of Modernized Loan Approval System Based on Machine Learning Approach", Ieeexplore.ieee.org, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9498475/authors#authors. [Accessed: 08- Apr- 2022].