LOAN APPROVAL PREDICTION

Submitted in partial fulfillment of the requirements

of the degree of

B. E. Information Technology

By

Suraj Andhe 02

Shubham Nalawade 61

Siddhesh Parab 65

Supervisor:

Prof. Amruta Sankhe

Guide

Department of Information Technology
Atharva College of Engineering
Malad West Mumbai

University of Mumbai
2021-2022

AET'S

ATHARVA COLLEGE OF ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"Loan Approval Prediction"** is a bonafide work of **"Suraj Andhe" (02), "Shubham Nalawade" (61)** and **"Siddhesh Parab" (65)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Information Technology

(Name and sign)

Supervisor/Guide

(Name and sign)

External Examiner

College Seal

(Name and sign)

Internal Examiner

(Name and sign)

Head of Department

(Name and sign)

Principal

Project Report Approval for B.E.

This project report entitled ***Loan Approval Prediction*** by ***Suraj Andhe, Shubham Nalawade*** and ***Siddhesh Parab*** is approved for the degree of ***B.E. in Information Technology.***

Examiners

1.--

2.--

Date:

Place:

Declaration

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

Suraj Andhe (02)

Shubham Nalawade (61)

Siddhesh Parab (65)

Date:

ABSTRACT

Today a lot of people/companies are applying for bank loans. The core business part of every bank is the distribution of loans. The main objective of the banking sector is to give their assets in safe hands. But the banks or the financial companies take a very long time for the verification and validation process and even after going through such a regress process there is no surety that whether the applicant chosen is deserving or not.

The enhancement in the banking sector lots of people are applying for bank loans but the bank has its limited assets which it has to grant to limited people only, so finding out to whom the loan can be granted which will be a safer option for the bank is a typical process. Machine Learning (ML) techniques are very useful in predicting outcomes for large amounts of data.

To solve this problem, we have developed a system in which we can predict whether the applicant chosen will be a deserving applicant for approving the loan or not. The system predicts on the basis of the model that has been trained using machine learning algorithms. We have even compared the accuracy of different machine learning algorithms.

We got a percentage of accuracy ranging from 75-85% but the best accuracy we got was from Logistic Regression i.e., 88.70% The system includes a user interface web application where the user can enter the details required for the model to predict. The drawback of this model is that it takes into consideration many attributes but in real life sometimes the loan application can also be approved on a single strong attribute, which will not be possible using this system.

Key Words: *Machine Learning, Loan Approval Prediction, Web Application, Bank, Algorithms, Random Forest, Naïve Bayes, Logistic Regression, K Nearest Neighbor, Decision Tree.*

LIST OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Today many banks/financial companies approve loans after a regress process of verification and validation but still there is no surety whether the chosen applicant is the deserving right applicant out of all applicants. Through this system we can predict whether that particular applicant is safe or not and the whole process of validation of features is automated by machine learning technique.

1.1 Description

Nowadays, Banks play a vital role in the market economy. The success or failure of an organization largely depends on the industry's ability to evaluate credit risk. Before giving the credit loan to borrowers, the bank decides whether the borrower is bad (defaulter) or good (non defaulter).The prediction of borrower status i.e. in future borrower will be defaulter or non defaulter is a challenging task for any organization or bank. Basically the loan defaulter prediction is a binary classification problem Loan amount; customers history governs his credit ability for receiving loan. The problem is to classify the borrower as defaulter or non defaulter. However developing such a model is a very challenging task due to increasing demands for loans.Banks struggle a lot to get an upper hand over each other to enhance overall business due to tight competition. Credit Risk assessment is a crucial issue faced by Banks nowadays which helps them to evaluate if a loan applicant can be a defaulter at a later stage so that they can go ahead and grant the loan or not. This helps the banks to minimize the possible losses and can increase the volume of credits.

Loan Prediction is very helpful for employees of banks as well as for the applicant also. The aim of this project is to provide a quick, immediate and easy way to choose the deserving applicants. It can provide special advantages to the bank.

1.2 Problem Statement

With the enhancement in the banking sector lots of people are applying for bank loans but the bank has its limited assets which it has to grant to limited people only, so finding out to whom the loan can be granted which will be a safer option for the bank is a typical process.

So in this project we try to reduce this risk factor behind selecting the safe person so as to save lots of bank efforts and assets. This is done by mining the Big Data of the previous records of the people to whom the loan was granted before and on the basis of these records/experiences the system was trained using the machine learning model which gives the most accurate result.

1.3 Motivation

Despite the fact that our banking system has many products to sell, the main source of income for a bank is its credit line. So, they can earn from interest on the loans they credit. Commercial loans have always been a big part of the banking industry, and lenders are always aiming to reduce their credit risk. The profit or loss of a bank is largely influenced by loans, i.e., whether the customers repay the loans or default on them. The banks need to decide whether he/she is a good(non-defaulter) or bad(defaulter) before giving the loans to the borrowers. Among the most important problems to be addressed in commercial loan lending is the borrowers' creditworthiness. The credit risk is defined as the likelihood that borrowers will fail to meet their loan obligations.

To predict whether the borrower will be good or bad is a very difficult task for any bank or organization. The banking system uses a manual process for checking whether a borrower is a defaulter or not. No doubt the manual process will be more accurate and effective, but this process cannot work when there are a large number of loan applications at the same time. If there occurs a time like this, then the decision-making process will take a very long time and also lots of manpower will be required. If we are able to do the loan prediction it will be very helpful for applicants and also for the employees of banks. So, the task is to classify the borrower as good or bad i.e., whether the borrower will be able to pay the debts back or not.

1.4 Proposed Solution

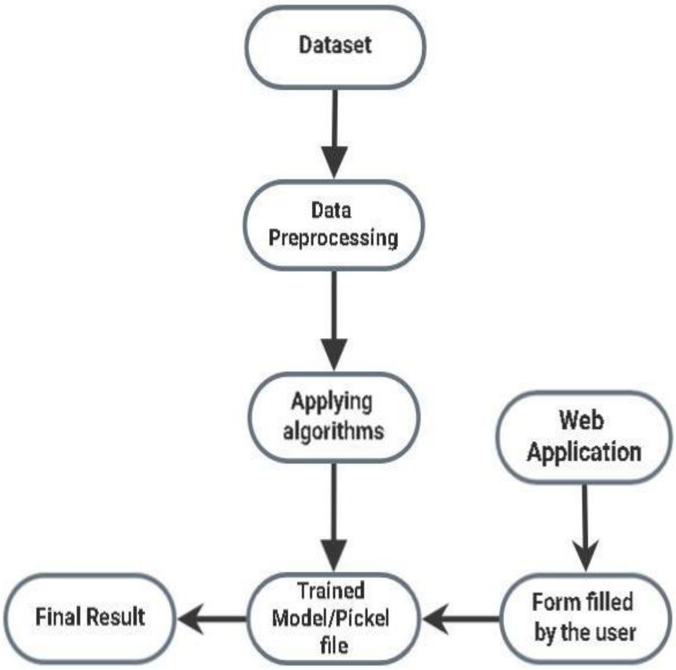


Fig (1.4) : Block Diagram

The proposed system includes a web application with a model trained by using machine learning algorithms deployed in it. There are a total 11 fields in the form which the user needs to fill. The dataset that we have used for training the model also includes 11 attributes. This dataset is pre-processed before using it for training the model. The pre-processing is done by replacing the null values in the dataset with mean and mode method and replacing the string values with 1 and 0 using label encoder. Then the dataset was divided into two parts: train and test. 90% of the dataset is used for training purposes and 10% is used for testing the accuracy that the model will give for different algorithms. After splitting the dataset different algorithms were applied and each of them gave different accuracy. The best we got was from Logistic Regression i.e., 88%. Once the model is trained a pickle file is created of the model. When the client wants to predict his/her loan approval the client has to first fill a form by visiting our web application.

After filling the form, the user has to just click on the MAKE PREDICTION button and depending on the pickle file or the model that we have trained it will give the result as whether the loan of the customer will be approved or not. As we have also done the comparison of different machine learning algorithms in terms of their accuracy. The web application also includes a bar plot graph of the comparison of algorithms, insights of the dataset that we have used for training the model. This system will make it easier for the banks or organizations to do the job of loan approval prediction.

1.5 Scope of Project

- With the help of this system it will become easy for the banks in the future to take the decision regarding loan approval or rejection.

- To provide Quick, immediate and easy way to choose the deserving applicants.

- It will ease the efforts of the bank as loan applications are increasing day by day

Chapter 2

Literature Review

Table 2 : Literature Review

Sr No	Title	Authors	Description	Accuracy

1	An Approach for Prediction of Loan Approval using Machine Learning Algorithm [1] IEEE(2020)	Mohammad Ahmad Sheikh;
		Amit Kumar Goel;
		Tapas Kumar.
			In our banking system, banks have many products to sell but the main source of income of any bank is on its credit line.	81.11%
				(Logistic Regression)

2	AzureML Based Analysis and Prediction Loan Borrowers Creditworthy [2] IEEE(2020)	Khaldoon Alshouiliy, Ali AlGhamdi, Dharma P. Agrawal.	In this research, our aim is to analyze LendingClub dataset to make it well understood dataset features.	77.00% Two class decision jungles 81.00% Two classes decision
3	Loan Default Forecasting using Data Mining IEEE [3] (2020)	Bhoomi Patel, Harshal Patil, Shree Jaswal.	Assessment of default on a debt is a crucial process that should be done banks to help them to assess if a loan applicant can be a defaulter	84.03% Gradient Boosting 83.51% Random Forest 49.63% Logistic Regression
4	Application of Machine Learning in Credit Risk Assessment: A Prelude to Smart Banking [4] IEEE (2019)	Sazzad Hossain, Hossain Arif , Samiul Islam	An ever-changing economy as the rate of loan defaults are gradually increasing, authorities are finding it difficult to assess loan requests and tackle the risks of loan defaulters	85.1% Logistic Regression 84.5% SVM 85.4% Random Forest
5	Prediction of loan status in commercial bank using machine learning classifier IEEE [5] (2018)	G. Arutjothi; C. Senthamarai	The Banking Industry always needs a more accurate predictive modeling system for many issues. Predicting credit defaulters is a difficult task for the banking industry.	75.08% (K nearest Neighbor)

6	Predictive and Probabilistic approach using Logistic Regression IEEE [6] (2017)	Ashlesha Vaidya	This paper adheres to logistic regression as a machine learning tool in order to approach to the problem of loan approval prediction	Logistic Regression (Accuracy Not Mentioned)
7	Loan Prediction using Decision Tree and Random Forest IRJET [7] (2020)	Kshitiz Gautam, Arun Pratap Singh , Keshav Tyagi , Mr. Suresh Kumar	The aim of this paper is to find the nature or background or credibility of the client that is applying for the loan. We use exploratory data analysis techniques to deal with the problem of approving or rejecting the loan request or in short loan prediction.	62.12% Decision Tree 85.75% Random Forest
8	Prediction of Loan Approval using Machine Learning Algorithm: A Review Paper IRJET [8] (2021)	Kathe Rutika Pramod , Panhale Sakshi Dattatray , Avhad Pooja Prakash , Ghorpade Dinesh B. , Dapse Punam Laxman	Previous research in this era has shown that there are so many methods to study the problem of controlling loan default. But as the right predictions are very important for the maximization of profits	81.1% Decision Tree

Chapter 3

System Analysis

3.1 Functional Requirements

The entire project is designed using Pycharm IDE. Application is written using Python language since it has a great support for libraries which are used for machine learning and computer vision. This language is also easy to implement and code. The project uses pandas library for the purpose of data analysis of the given dataset. The numpy is used for the mathematical operations required in the financial calculations. While the Sci-kit or sklearn library is used for training the model i.e machine learning purpose. Not only that the bar plots and graphs shown in the webpage(under insights) are also created using this library along with the classification, regression and clustering part. For the frontend part, this system uses HTML5, CSS and Javascript along with bootstrap which is CSS library for a modern and perceptive GUI. Flask was used to create a pickle file of the model trained and this pickle file was further used for prediction.

3.2 Non Functional Requirements

3.2.1 Performance

The setup of the system should not only be smooth but it should also be lag free. It should predict the result accurately when the applicant fills the form for applying the loan. The UI of the webpage should be simple and understandable enough for every person engaging on it. The predicting model should work perfectly in order to get the desired outcome based on the required details filled by the user. Thus, the entire working should be well functioned to avoid any kind of inconvenience to the user

3.2.2 Technical Requirements

There is no such technical requirement since it has to be accessible to every customer and so this webpage can be opened on any laptop/desktop or mobile.

3.2.3 Scalability

This project is highly scalable as users can use it virtually from home or anywhere which makes it a much better procedure than the traditional banking loan process of waiting for long hours in the verification and approval/rejection process. In future this system can be improved by working on more dataset as we have currently worked on home loan dataset only. We can work on other data sets like vehicle loan, Education loan, business loan and gold loan. Even for smooth verification, we can add image option to verify the authenticity of the proofs enclosed with the application(example: Bank statements, identity proof, Address proof,etc)

3.3 Specific Requirements

3.3.1 Hardware Requirements

Table 3.3.1 : Hardware Requirements

1.	System Processor	Core i5 1035G1
2.	Hard Disk	1TB
3.	Max Processor Speed	3.60 GHz
4.	Memory	8 GB
5.	Input	Keyboard, Mouse

3.3.2 Software Requirements

- Pycharm
- Jupyter Notebook
- Python and its Libraries
- Flask
- Machine Learning

3.4 Use-Case Diagram

In UML diagram there are three actors i.e user/customer, trained dataset and system which individually perform their roles. The user has the role of interacting with the system during the loan application process.The user will first apply for his role by giving his details as mentioned in the web application.The input details further goes to system as well as trained dataset for giving the end result. The input taken by trained model goes through the algorithms and predicts the best possible accuracy which will later help the model to classify whether the loan applied will be approved or rejected.The system has the major role of handling the input details into dataset. Further it has to be cleaned and processed which includes the modifying of incorrect data and convert it into desired form which will be easily accessible. Later the data which is taken has to go through certain validations for verifying if the data which is given by customer is logical/valid or not. After this, there has to be model selections includes selecting the best model and helping out the model to give out the best accuracy of result prediction.

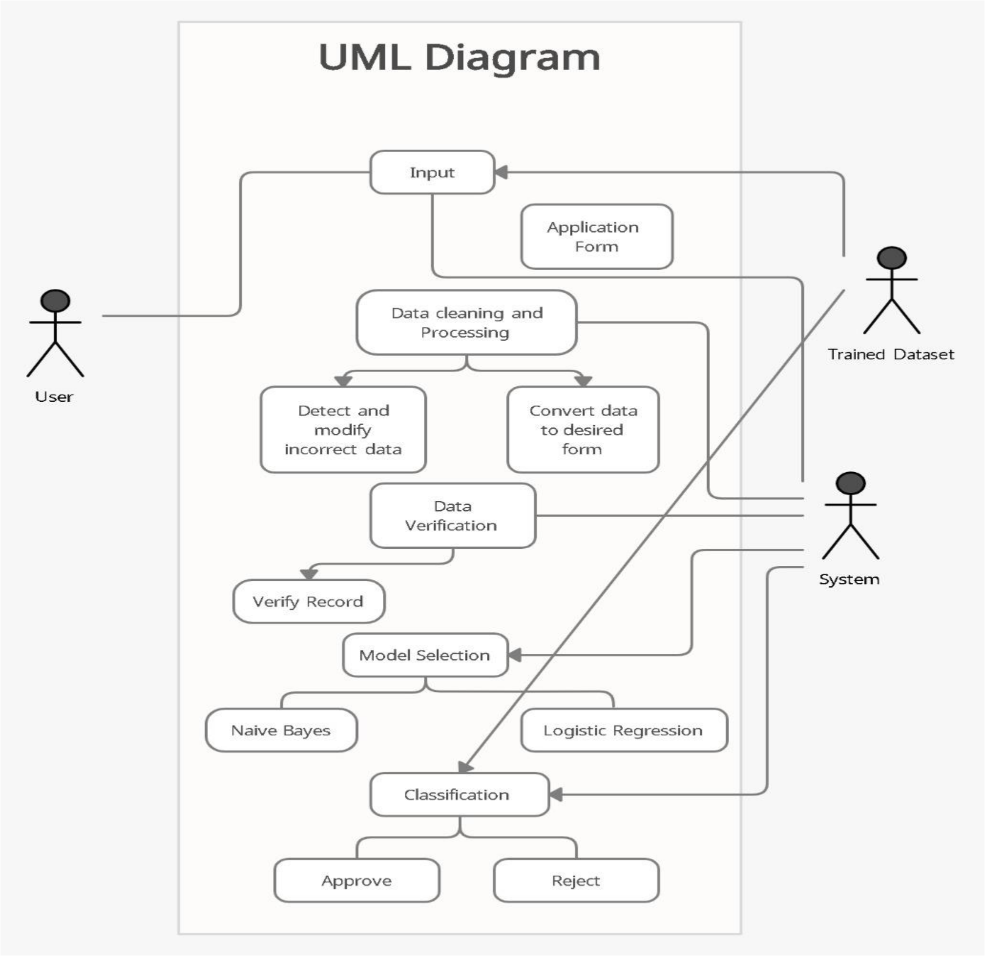


Fig (3.4)

Chapter 4

Analysis Modeling

4.1 Data Modeling

The ER diagram of our project has 4 entities: bank, customer,account and loan.The bank stores the information about the customer which has attributes name, id and address.The bank will itself have its own identity which includes the branch name, id and address where it is located. Not just that it has accounts of its customers which includes account no. and type of account.After verifying all the details the bank will then disburse the loan of the customer which will have the attributes consisting of loan id and amount of loan provided to the customer.

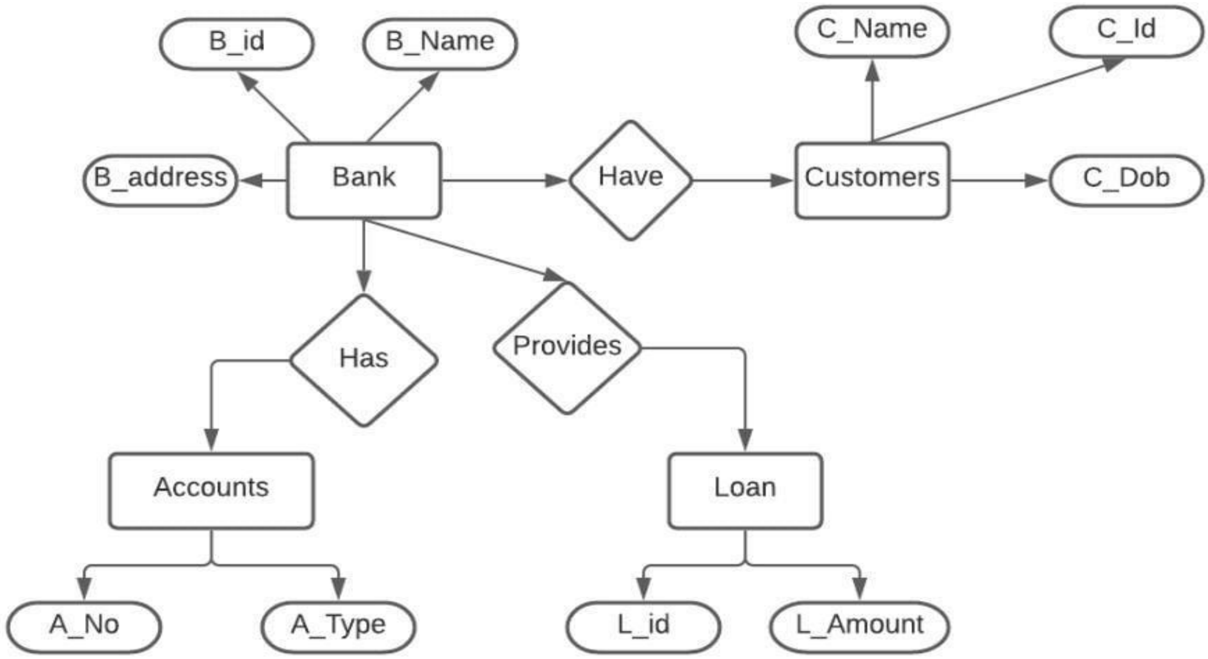


Fig (4.1) : ER Diagram

4.2 Sequence Diagram

The sequence diagram consists of all the processes that will undergo to get the final result in a particular sequence.It will begin with deploying the loan prediction app on the web so that customer can access it, fill the details and apply for the loan. After filling details the app will the Request result from the ML model which will return the output as "approved or rejected" to the app which will further display it for the user as the final step.

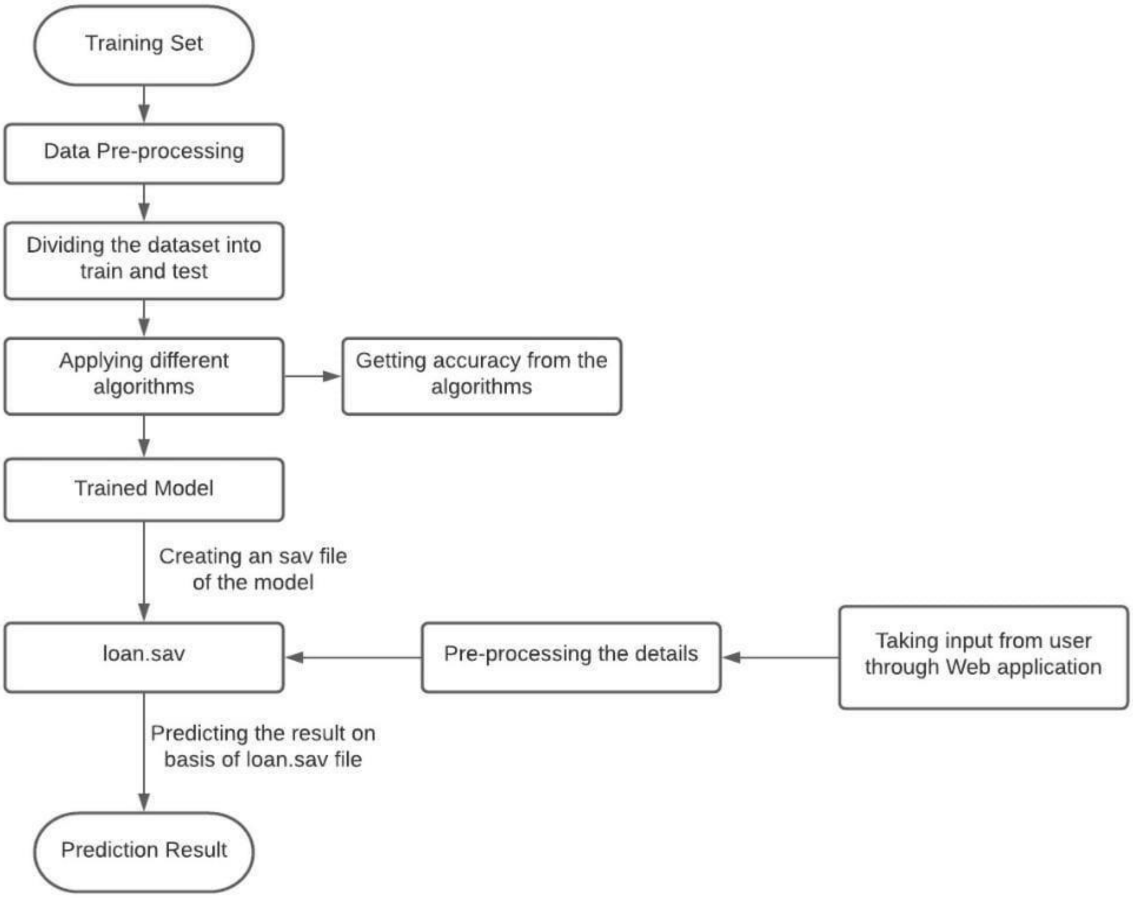

Training Set
Data Pre-processing
Dividing the dataset into train and test
Applying different algorithms
Getting accuracy from the algorithms
Trained Model
Creating an sav file of the model
loan.sav
Pre-processing the details
Taking input from user through Web application
Predicting the result on basis of loan.sav file
Prediction Result

Fig (4.2) : Sequence Diagram

4.3 Functional Modeling

Data flow diagram is the overall flow of the project. Firstly the loan data set is taken and it is analyzed and gets splitted into train and test model for further processing.Next is we do the data cleaning and preprocessing of it by removing the null values and visualizing it.The M.L algorithms are further applied on the model for getting the accuracy of each algorithm.Later a comparative study is done on the basis of accuracy result of each algorithm and finally the highest accuracy algorithm is chosen for getting the best possible accurate prediction. After this, the model is deployed on the web app in which the customer inputs his details while applying for a loan and later gets the desired result as approved or rejected on the basis on prediction model.

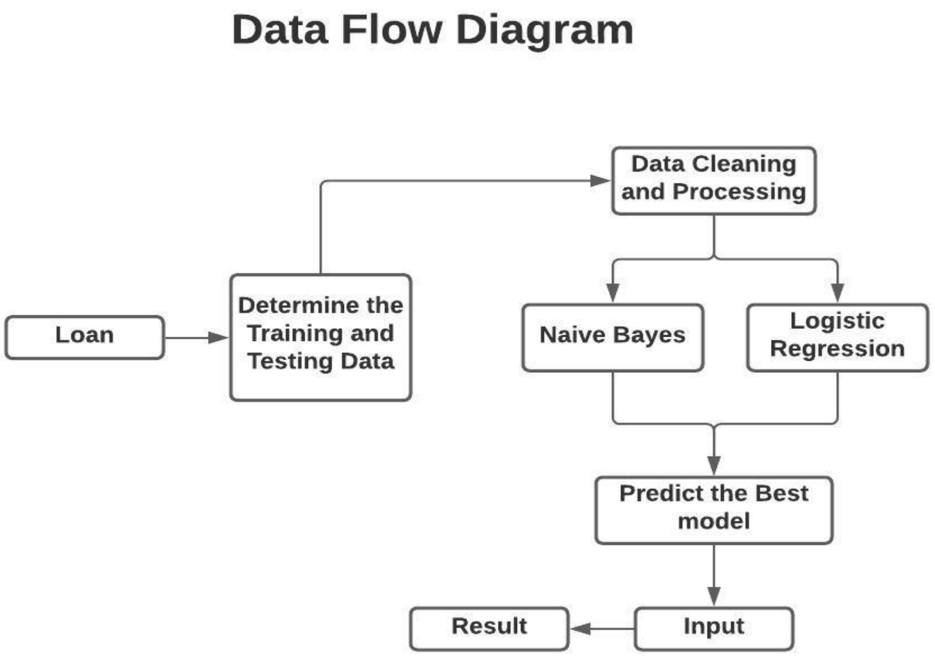


Fig (4.3) : Data Flow Diagram

4.4 Timeline Chart

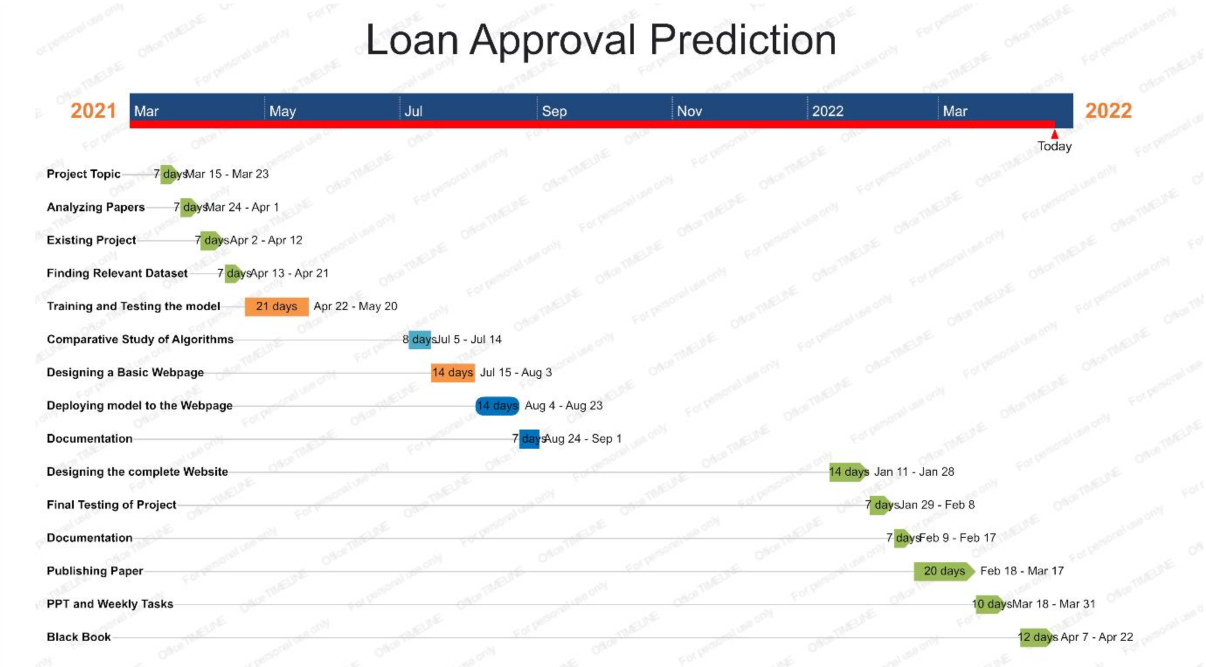


Fig (4.4)

CHAPTER 5

DESIGN

5.1 Architectural Design

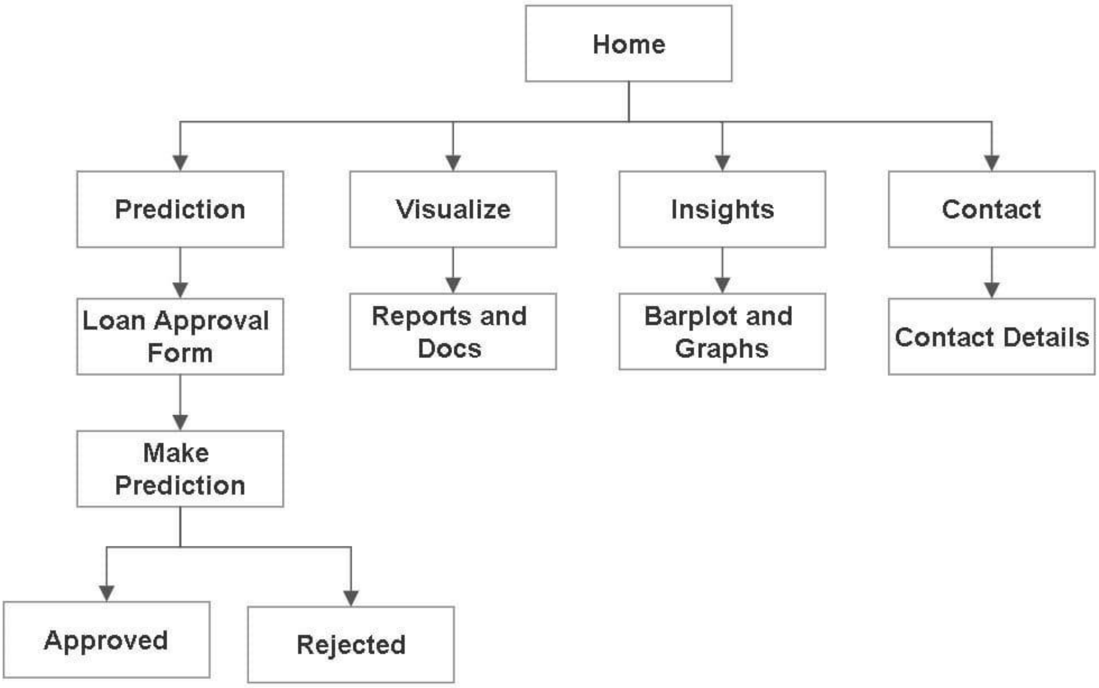


Fig (5.1)

This diagram shows the design implementation of our project. First thing that appears after launching this application are 4 buttons i.e Prediction, Visualize, insights and Contact Details.

Prediction : After clicking this, a form appears wherein customers who wish to apply for a loan can check their eligibility status by filling out all the required personal details which the banks require for checking the credibility of the customer. After filling the details, the customer will have to wait for the response from the webpage if the application is approved further or rejected.

Visualize : It includes the report in the form of graphical representation of the attributes of training dataset on the basis of which the model is been trained.

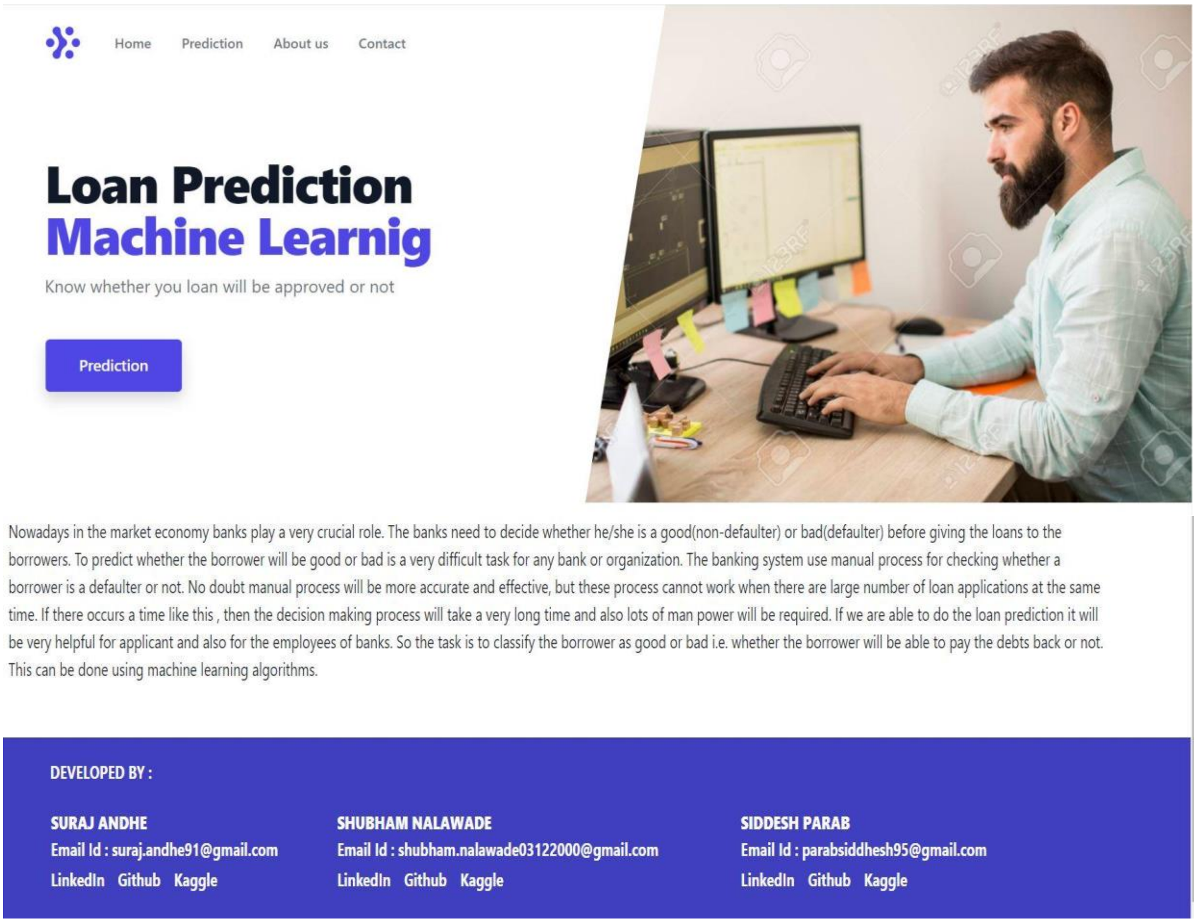
Insights : It includes the barplot and graphical representation of accuracy of the different machine learning algorithms that we applied on our dataset.

Contact Details: It contains all the possible ways of reaching out the webpage admins for any inconvenience.

5.2 User Interface Design

5.2.1 Home Page

First, we have our home page where we get information about out system, details of the developers of the system and also a button to go the prediction page.



Loan Prediction
Machine Learnig

Know whether you loan will be approved or not

Nowadays in the market economy banks play a very crucial role. The banks need to decide whether he/she is a good(non-defaulter) or bad(defaulter) before giving the loans to the borrowers. To predict whether the borrower will be good or bad is a very difficult task for any bank or organization. The banking system use manual process for checking whether a borrower is a defaulter or not. No doubt manual process will be more accurate and effective, but these process cannot work when there are large number of loan applications at the same time. If there occurs a time like this , then the decision making process will take a very long time and also lots of man power will be required. If we are able to do the loan prediction it will be very helpful for applicant and also for the employees of banks. So the task is to classify the borrower as good or bad i.e. whether the borrower will be able to pay the debts back or not. This can be done using machine learning algorithms.

DEVELOPED BY :

SURAJ ANDHE
Email Id : suraj.andhe91@gmail.com
LinkedIn Github Kaggle

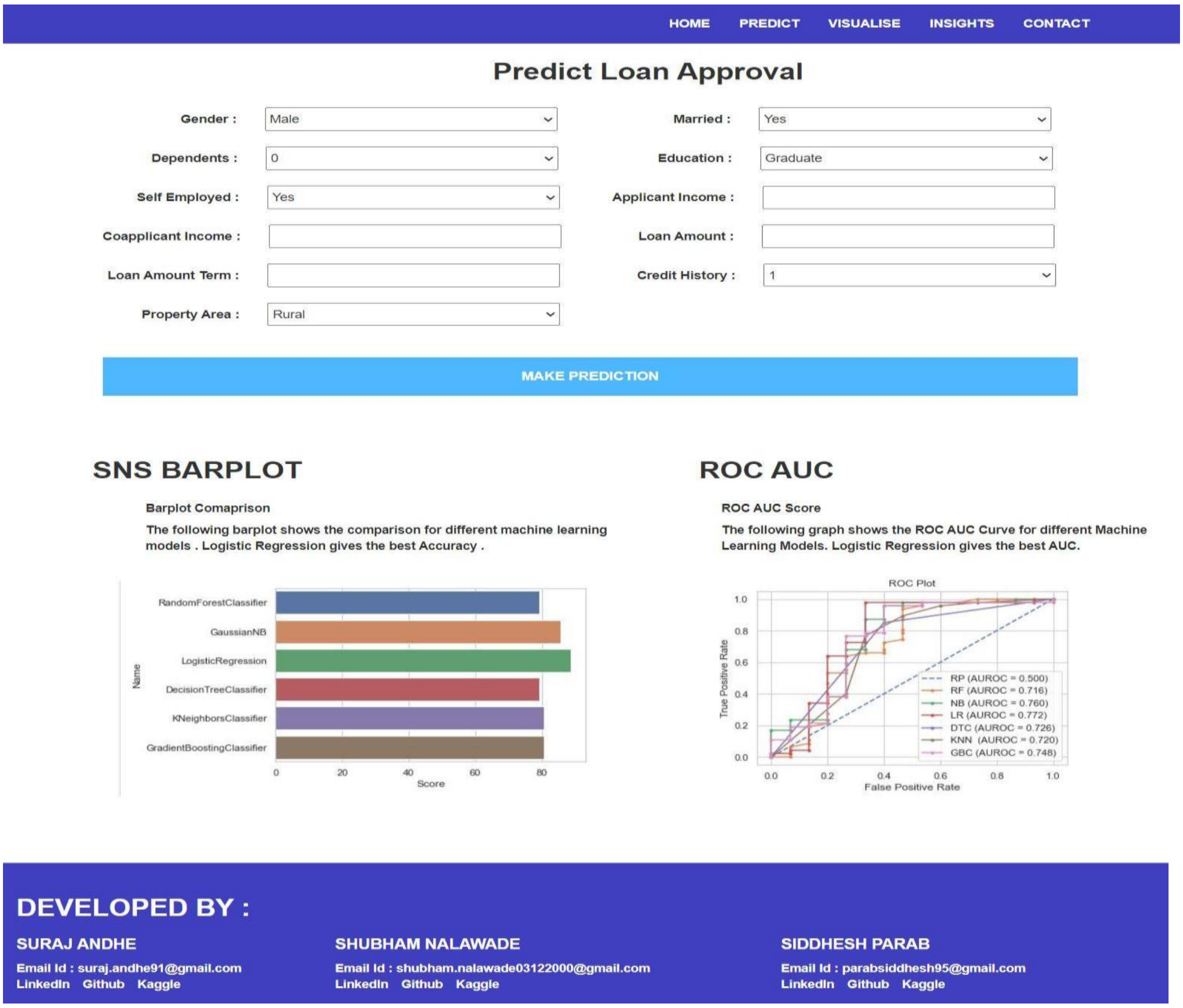
SHUBHAM NALAWADE
Email Id : shubham.nalawade03122000@gmail.com
LinkedIn Github Kaggle

SIDDESH PARAB
Email Id : parabsiddhesh95@gmail.com
LinkedIn Github Kaggle

Fig (5.2.1)

5.2.2 Prediction Page

The next is the prediction page where the user can fill the form to check whether he/she is eligible for loan approval or not. It also includes comparison of different algorithms in terms of accuracy in graphical representation.



Predict Loan Approval

Gender :	Male	Married :	Yes
Dependents :	0	Education :	Graduate
Self Employed :	Yes	Applicant Income :	
Coapplicant Income :		Loan Amount :	
Loan Amount Term :		Credit History :	1
Property Area :	Rural		

SNS BARPLOT

Barplot Comaprison

The following barplot shows the comparison for different machine learning models . Logistic Regression gives the best Accuracy .

ROC AUC

ROC AUC Score

The following graph shows the ROC AUC Curve for different Machine Learning Models. Logistic Regression gives the best AUC.

DEVELOPED BY :

SURAJ ANDHE
Email Id : suraj.andhe91@gmail.com
LinkedIn Github Kaggle

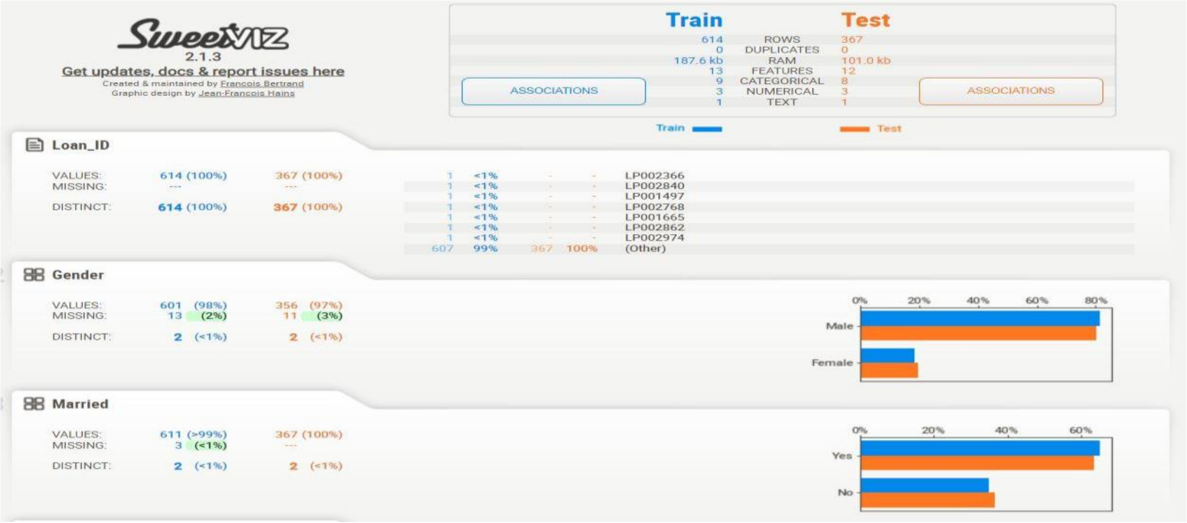
SHUBHAM NALAWADE
Email Id : shubham.nalawade03122000@gmail.com
LinkedIn Github Kaggle

SIDDHESH PARAB
Email Id : parabsiddhesh95@gmail.com
LinkedIn Github Kaggle

Fig (5.2.2)

5.2.3 Visualise Page

This page gives the report or analysis of the dataset that we have used to train the model.



SweetVIZ
2.1.3
Get updates, docs & report issues here
Created & maintained by Francois Bertrand
Graphic design by Jean-Francois Hains

Train		Test
614	ROWS	367
0	DUPLICATES	0
187.6 kb	RAM	101.0 kb
13	FEATURES	12
9	CATEGORICAL	8
3	NUMERICAL	3
1	TEXT	1

ASSOCIATIONS
ASSOCIATIONS
Train
Test

Loan_ID

VALUES:	614 (100%)	367 (100%)
MISSING:	---	---
DISTINCT:	614 (100%)	367 (100%)

1	<1%	-	-	LP002366
1	<1%	-	-	LP002840
1	<1%	-	-	LP001497
1	<1%	-	-	LP002768
1	<1%	-	-	LP001665
1	<1%	-	-	LP002862
1	<1%	-	-	LP002974
607	99%	367	100%	(Other)

Gender

VALUES:	601 (98%)	356 (97%)
MISSING:	13 (2%)	11 (3%)
DISTINCT:	2 (<1%)	2 (<1%)

0%
20%
40%
60%
80%
Male
Female

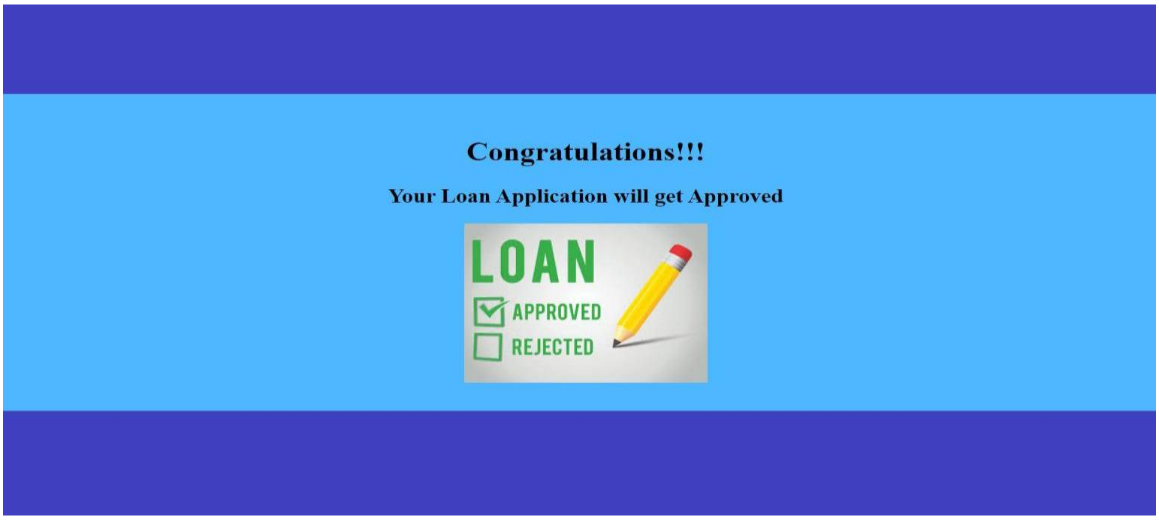
Married

VALUES:	611 (>99%)	367 (100%)
MISSING:	3 (<1%)	---
DISTINCT:	2 (<1%)	2 (<1%)

Fig (5.2.3)

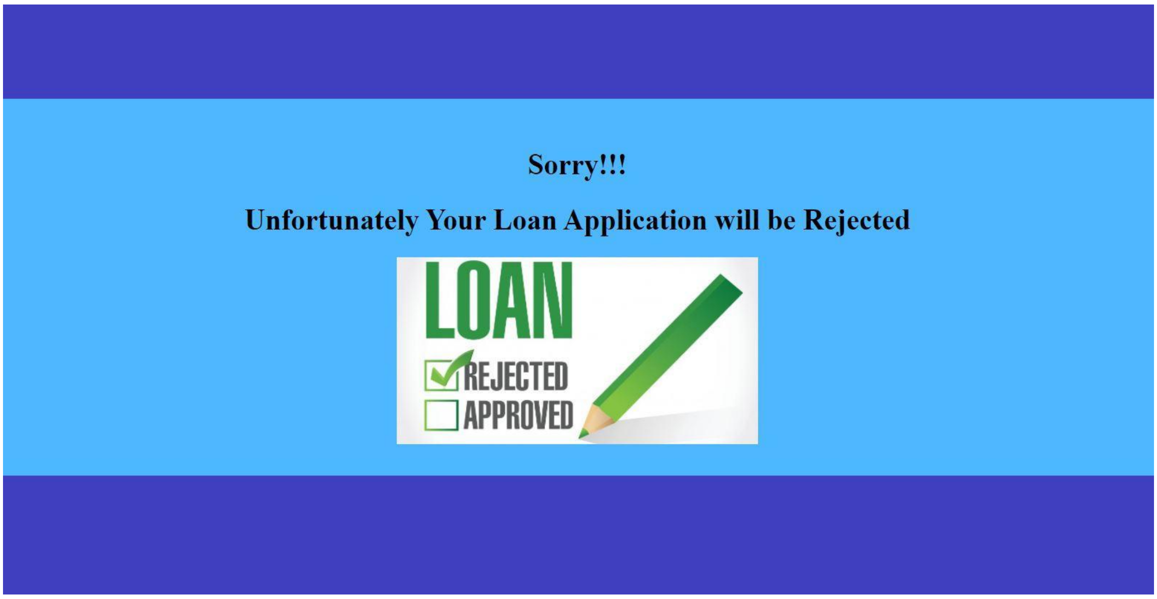
5.2.4 Result Page

The last is the result page where it shows the result of whether the loan application is approved or not.



Congratulations!!!

Your Loan Application will get Approved



Sorry!!!

Unfortunately Your Loan Application will be Rejected

Fig (5.2.4)

CHAPTER 6
IMPLEMENTATION

6.1 Algorithms

Following are the most popular Supervised Machine learning algorithms used in this Project:

6.1.1 Naïve Bayes

It is mainly used in text classification that includes a high- dimensional training dataset. The 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 [7].

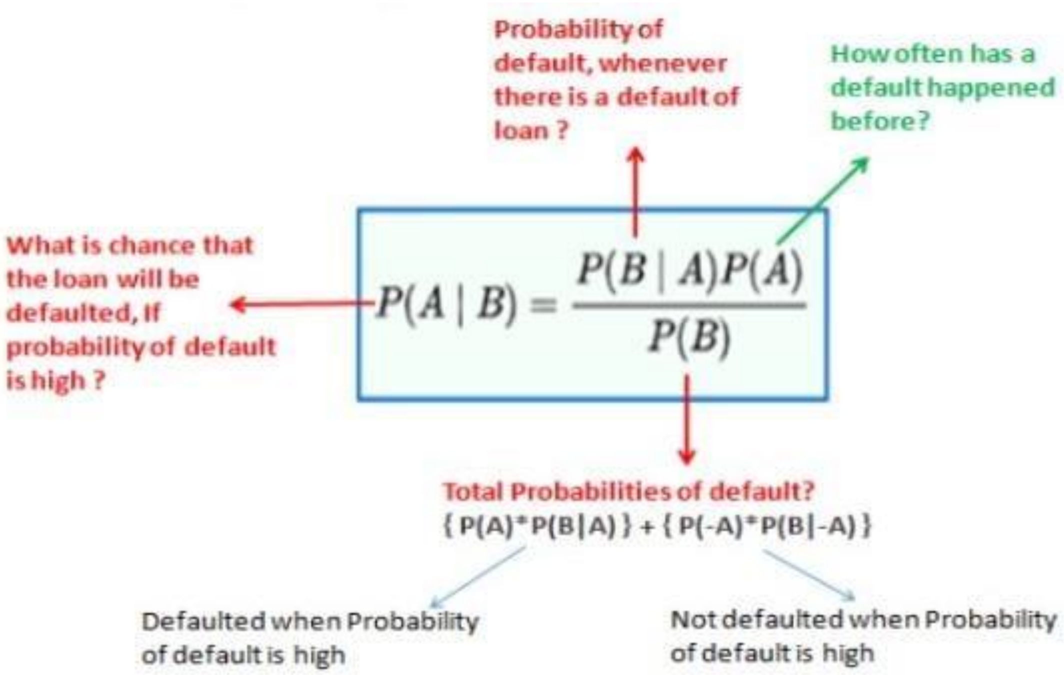

Probability of default, whenever there is a default of loan ?
How often has a default happened before?
What is chance that the loan will be defaulted, if probability of default is high ?
$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$
Total Probabilities of default?
{ P(A)*P(B|A) } + { P(-A)*P(B|-A) }
Defaulted when Probability of default is high
Not defaulted when Probability of default is high

Fig (6.1.1) [9]

6.1.2 Logistic Regression

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 [8].

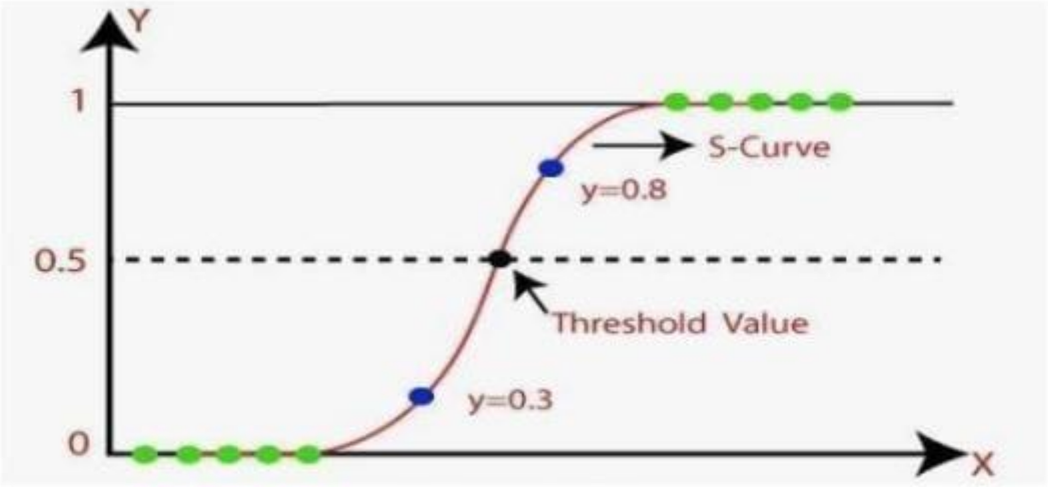


Fig (6.1.2) [10]

6.1.3 Random Forest

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. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting [9].

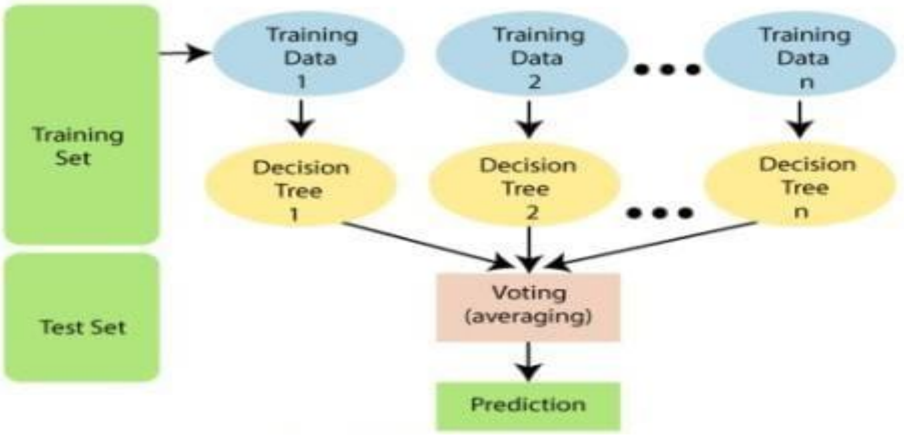


Fig (6.1.3) [11]

6.1.4 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 [10].

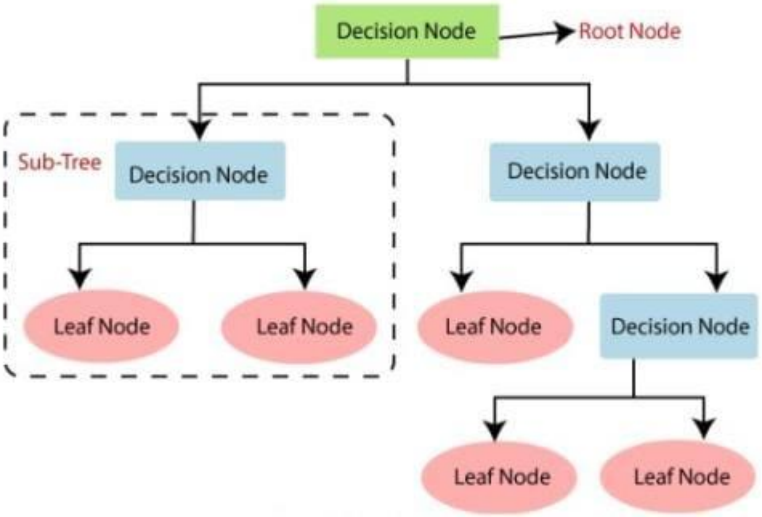


Fig (6.1.4) [12]

6.1.5 K Nearest Neighbours

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems [11].

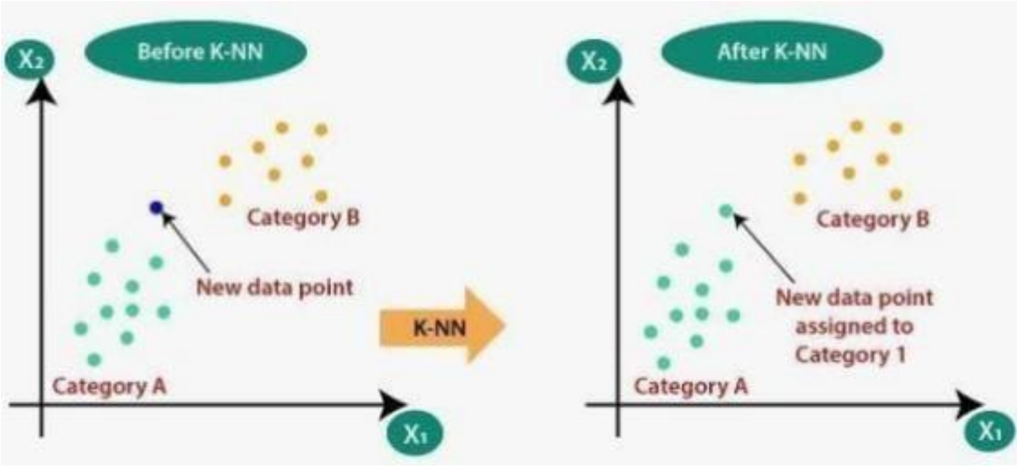
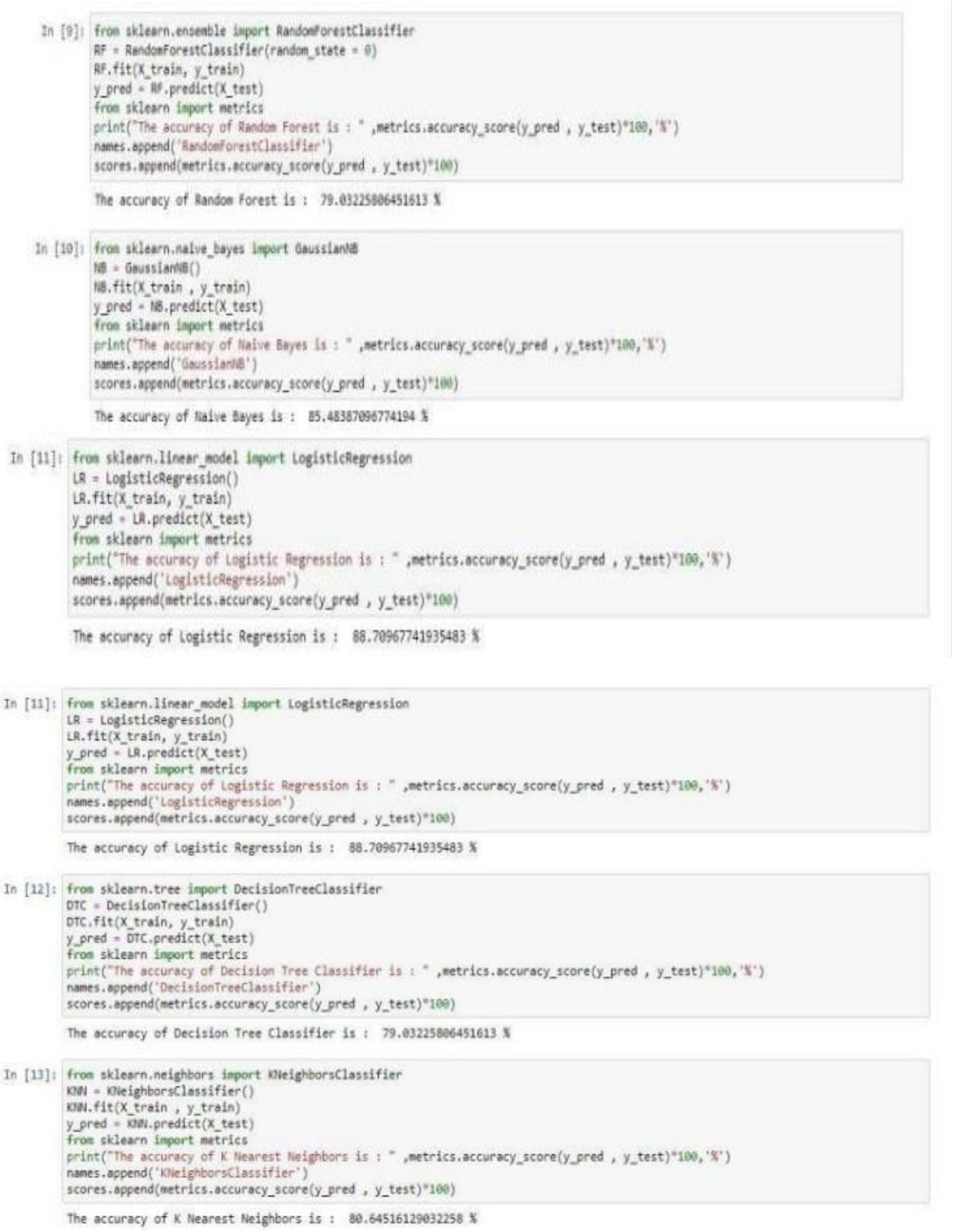


Fig (6.1.5) [13]

6.2 Working of the Project



```
In [9]: from sklearn.ensemble import RandomForestClassifier
        RF = RandomForestClassifier(random_state = 0)
        RF.fit(X_train, y_train)
        y_pred = RF.predict(X_test)
        from sklearn import metrics
        print("The accuracy of Random Forest is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
        names.append('RandomForestClassifier')
        scores.append(metrics.accuracy_score(y_pred , y_test)*100)

        The accuracy of Random Forest is :  79.03225806451613 %

In [10]: from sklearn.naive_bayes import GaussianNB
         NB = GaussianNB()
         NB.fit(X_train , y_train)
         y_pred = NB.predict(X_test)
         from sklearn import metrics
         print("The accuracy of Naive Bayes is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
         names.append('GaussianNB')
         scores.append(metrics.accuracy_score(y_pred , y_test)*100)

         The accuracy of Naive Bayes is :  85.48387096774194 %

In [11]: from sklearn.linear_model import LogisticRegression
         LR = LogisticRegression()
         LR.fit(X_train, y_train)
         y_pred = LR.predict(X_test)
         from sklearn import metrics
         print("The accuracy of Logistic Regression is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
         names.append('LogisticRegression')
         scores.append(metrics.accuracy_score(y_pred , y_test)*100)

         The accuracy of Logistic Regression is :  88.70967741935483 %
```

```
In [11]: from sklearn.linear_model import LogisticRegression
         LR = LogisticRegression()
         LR.fit(X_train, y_train)
         y_pred = LR.predict(X_test)
         from sklearn import metrics
         print("The accuracy of Logistic Regression is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
         names.append('LogisticRegression')
         scores.append(metrics.accuracy_score(y_pred , y_test)*100)

         The accuracy of Logistic Regression is :  88.70967741935483 %
```

```
In [12]: from sklearn.tree import DecisionTreeClassifier
         DTC = DecisionTreeClassifier()
         DTC.fit(X_train, y_train)
         y_pred = DTC.predict(X_test)
         from sklearn import metrics
         print("The accuracy of Decision Tree Classifier is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
         names.append('DecisionTreeClassifier')
         scores.append(metrics.accuracy_score(y_pred , y_test)*100)

         The accuracy of Decision Tree Classifier is :  79.03225806451613 %

In [13]: from sklearn.neighbors import KNeighborsClassifier
         KNN = KNeighborsClassifier()
         KNN.fit(X_train , y_train)
         y_pred = KNN.predict(X_test)
         from sklearn import metrics
         print("The accuracy of K Nearest Neighbors is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
         names.append('KNeighborsClassifier')
         scores.append(metrics.accuracy_score(y_pred , y_test)*100)

         The accuracy of K Nearest Neighbors is :  80.64516129032258 %
```

Fig (6.2)

CHAPTER 7

TEST CASES

7.1 Test Cases

Table 7.1 : Test Cases

Test Case	Test Scenario	Test Steps	Expected Results	Actual Results	Pass/Fail
1.	Predict	Go to the Website & click on Predict button	A loan approval form will appear that contains applicant income, loan amount, etc.	As Expected	Pass
2.	Visualize	First click on predict button , then a page will appear where on the top there will be option of visualize in the menu bar, select that	Takes you to the report section which contains detail about each attribute	As Expected	Pass
3.	Insights	First click on predict button , then a page will appear where on the top there will be option of Insights in the menu bar, select that	Displays the graphs and shows which ML algorithm performs best	As Expected	Pass
4.	Contact	Click on Contact button	Shows the details of candidates who have created the project	As Expected	Pass
5.	Submit	Press the submit button after filling the loan approval form	Displays the decision whether the candidates loan should be approved or not	As Expected	Pass

7.2 Type of Testing Used

White Box Testing

The box testing approach of software testing consists of black box testing and white box testing. We are discussing here white box testing which also known as glass box is testing, structural testing, clear box testing, open box testing and transparent box testing. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs. It is based on inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform and verify the application along with the requirements and identify the bugs and sends it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application.

CHAPTER 8
RESULT ANALYSIS AND DISCUSSION

8.1 Comparison of Algorithms

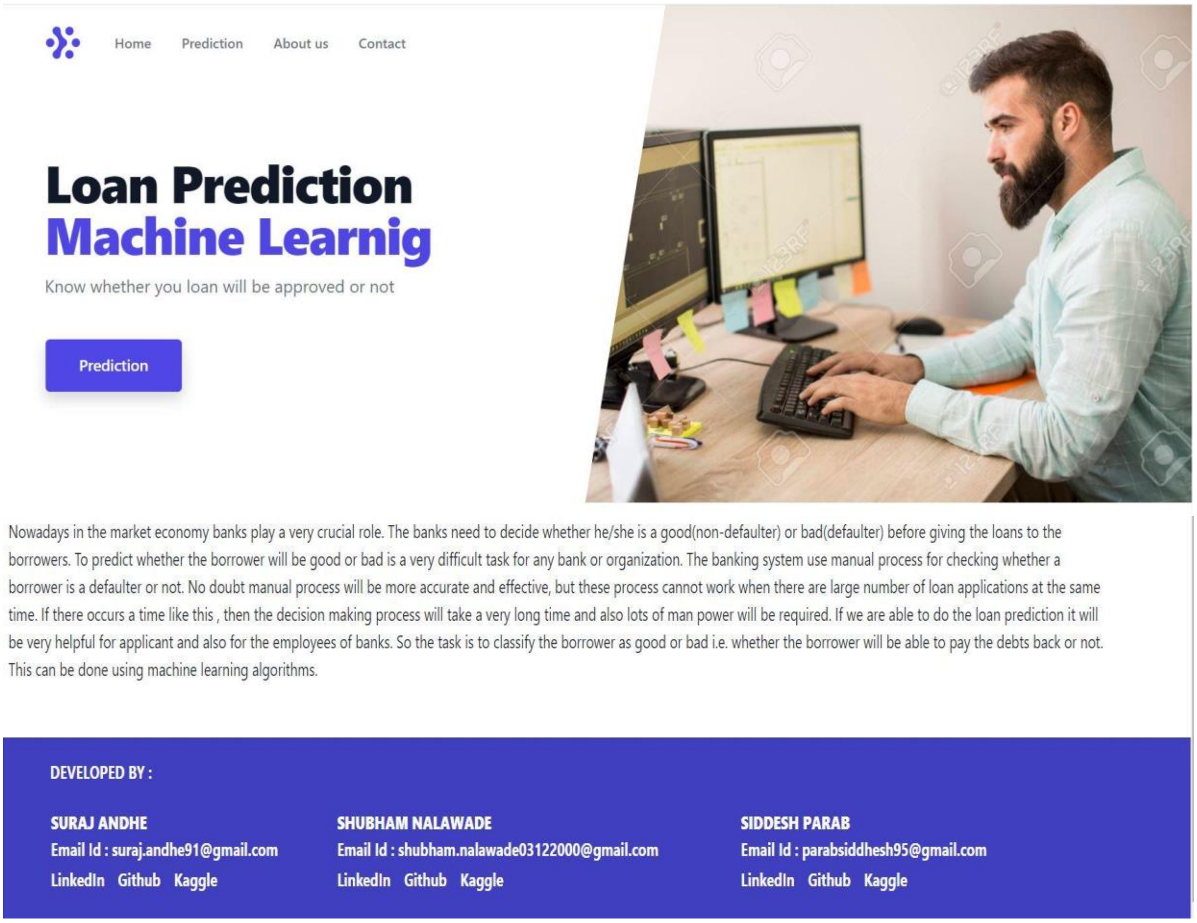
So we have successfully compared different machine learning algorithms for the Property Loan dataset they are Random Forest , Naive Bayes , Logistic Regression and K Nearest Neighbors.

The Logistic Regression algorithm gave the best accuracy (88.70%) .

Table 8.1 : Comparison of Algorithm

Sr. No.	Algorithm	Accuracy
1.	Random Forest	79.03%
2.	Naive Bayes	85.48%
3.	Decision Tree	79.03%
4.	**Logistic Regression**	**88.70%**
5.	K Nearest Neighbour	80.64%

8.2 Web Application



Loan Prediction
Machine Learnig

Know whether you loan will be approved or not

Nowadays in the market economy banks play a very crucial role. The banks need to decide whether he/she is a good(non-defaulter) or bad(defaulter) before giving the loans to the borrowers. To predict whether the borrower will be good or bad is a very difficult task for any bank or organization. The banking system use manual process for checking whether a borrower is a defaulter or not. No doubt manual process will be more accurate and effective, but these process cannot work when there are large number of loan applications at the same time. If there occurs a time like this , then the decision making process will take a very long time and also lots of man power will be required. If we are able to do the loan prediction it will be very helpful for applicant and also for the employees of banks. So the task is to classify the borrower as good or bad i.e. whether the borrower will be able to pay the debts back or not. This can be done using machine learning algorithms.

DEVELOPED BY :

SURAJ ANDHE
Email Id : suraj.andhe91@gmail.com
LinkedIn Github Kaggle

SHUBHAM NALAWADE
Email Id : shubham.nalawade03122000@gmail.com
LinkedIn Github Kaggle

SIDDESH PARAB
Email Id : parabsiddhesh95@gmail.com
LinkedIn Github Kaggle

Fig (8.2)

CHAPTER 9
CONCLUSION

Conclusion

For the purpose of predicting the loan approval status of the applied customer, we have chosen the machine learning approach to study the bank dataset. We have applied various machine learning algorithms to decide which one will be the best for applying on the dataset to get the result with the highest accuracy. Following this approach, we found that apart from the logistic regression, the rest of the algorithms performed satisfactory in terms of giving out the accuracy. The accuracy range of the rest of the algorithms were from 75% to 85%. Whereas the logistic regression gave us the best possible accuracy (88.70%) after the comparative study of all the algorithms.

We also determined the most important features that influence the loan approval status. These most important features are then used on some selected algorithms and their performance accuracy is compared with the instance of using all the features. This model can help the banks in figuring out which factors are important for the loan approval procedure. The comparative study makes us clear about which algorithm will be the best and ignores the rest, based on their accuracy.

APPENDIX

model.py :

```
import pandas as pd

import numpy as np

import pickle

import seaborn as sns

from matplotlib import pyplot as plt

import matplotlib

import joblib

df = pd.read_csv("train.csv")

df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount'].mean())

df['Loan_Amount_Term'] = df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mean())

df['Credit_History'] = df['Credit_History'].fillna(df['Credit_History'].mean())

df['Gender'] = df["Gender"].fillna(df['Gender'].mode()[0])

df['Married'] = df["Married"].fillna(df['Married'].mode()[0])

df['Dependents'] = df["Dependents"].fillna(df['Dependents'].mode()[0])

df['Self_Employed'] = df["Self_Employed"].fillna(df['Self_Employed'].mode()[0])


df['Total_Income'] = df['ApplicantIncome'] + df['CoapplicantIncome']

df['Total_Income_Log'] = np.log(df['Total_Income'])

df['LoanAmountLog'] = np.log(df['LoanAmount'])


from sklearn.preprocessing import LabelEncoder

cols = ['Gender',"Married","Education",'Self_Employed',"Property_Area","Loan_Status","Dependents"]

le = LabelEncoder()

for col in cols:

    df[col] = le.fit_transform(df[col])
```

```
cols=['Loan_ID','ApplicantIncome','CoapplicantIncome','LoanAmount','Total_Income']
df = df.drop(columns=cols , axis =1)
x = df.drop(columns=['Loan_Status'], axis=1)
y = df['Loan_Status']
from sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split(x, y , test_size = 0.1 , random_state = 0)
from sklearn.preprocessing import StandardScaler
ss = StandardScaler()
X_train = ss.fit_transform(X_train)
X_test = ss.fit_transform(X_test)
names = []
scores = []

from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier(random_state = 0)
RF.fit(X_train, y_train)
y_pred = RF.predict(X_test)
from sklearn import metrics
print("The accuracy of Random Forest is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
names.append('RandomForestClassifier')
scores.append(metrics.accuracy_score(y_pred , y_test)*100)

from sklearn.naive_bayes import GaussianNB
NB = GaussianNB()
NB.fit(X_train , y_train)
y_pred = NB.predict(X_test)
from sklearn import metrics
print("The accuracy of Naive Bayes is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
```

```
names.append('GaussianNB')

scores.append(metrics.accuracy_score(y_pred , y_test)*100)


from sklearn.linear_model import LogisticRegression

LR = LogisticRegression()

LR.fit(X_train, y_train)

y_pred = LR.predict(X_test)

from sklearn import metrics

print("The accuracy of Logistic Regression is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')

names.append('LogisticRegression')

scores.append(metrics.accuracy_score(y_pred , y_test)*100)


from sklearn.tree import DecisionTreeClassifier

DTC = DecisionTreeClassifier()

DTC.fit(X_train, y_train)

y_pred = DTC.predict(X_test)

from sklearn import metrics

print("The accuracy of Decision Tree Classifier is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')

names.append('DecisionTreeClassifier')

scores.append(metrics.accuracy_score(y_pred , y_test)*100)


from sklearn.neighbors import KNeighborsClassifier

KNN = KNeighborsClassifier()

KNN.fit(X_train , y_train)

y_pred = KNN.predict(X_test)

from sklearn import metrics

print("The accuracy of K Nearest Neighbors is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')

names.append('KNeighborsClassifier')
```

```
scores.append(metrics.accuracy_score(y_pred , y_test)*100)


from sklearn.ensemble import GradientBoostingClassifier
gbc = GradientBoostingClassifier()
gbc.fit(X_train , y_train)
y_pred = gbc.predict(X_test)
from sklearn import metrics
print("The accuracy of K GradientBoostingClassifier is : " ,metrics.accuracy_score(y_pred , y_test)*100,'%')
names.append('GradientBoostingClassifier')
scores.append(metrics.accuracy_score(y_pred , y_test)*100)
print(names)
print(scores)
pickle.dump(LR, open('mod.pkl', 'wb'))
```

app.py :

```
import math
from flask import Flask, render_template, request
import pickle
import numpy as np
model = pickle.load(open('model.pkl', 'rb'))
app = Flask(__name__)
@app.route('/')
def man():
    return render_template('index.html')
@app.route('/Home')
def h1():
    return render_template('Home.html')
@app.route('/Report')
def report():
    return render_template('Report.html')
@app.route('/predict', methods=['POST'])
def home():
    Gender = request.form['Gender']
    Married = request.form['Married']
    Dependents = request.form['Dependents']
    Education = request.form['Education']
    Self_Employed = request.form['Self_Employed']
    ApplicantIncome = request.form['ApplicantIncome']
    CoapplicantIncome = request.form['CoapplicantIncome']
    LoanAmount = request.form['LoanAmount']
    Loan_Amount_Term = request.form['Loan_Amount_Term']
    Credit_History = request.form['Credit_History']

    Property_Area = request.form['Property_Area']

    TotalIncome = int(ApplicantIncome) + int(CoapplicantIncome)

    LoanAmountLog = LoanAmount

    arr = np.array([[Gender,Married,Dependents,Education,Self_Employed,Loan_Amount_Term,Credit_History,Property_Area,TotalIncome,LoanAmountLog]])

    pred = model.predict(arr)

    print(arr)

    return render_template('result.html',data=pred)

if __name__ == "__main__":

    app.run(debug=True)
```

LITERATURE CITED

1. M. A. Sheikh, A. K. Goel and T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 490-494, doi: 10.1109/ICESC48915.2020.9155614.

2. K. Alshouiliy, A. AlGhamdi and D. P. Agrawal, "AzureML Based Analysis and Prediction Loan Borrowers Creditworthy," 2020 3rd International Conference on Information and Computer Technologies (ICICT), 2020, pp. 302-306, doi: 10.1109/ICICT50521.2020.00053.

3. B. Patel, H. Patil, J. Hembram and S. Jaswal, "Loan Default Forecasting using Data Mining," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154100.

4. S. Z. H. Shoumo, M. I. M. Dhruba, S. Hossain, N. H. Ghani, H. Arif and S. Islam, "Application of Machine Learning in Credit Risk Assessment: A Prelude to Smart Banking," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), 2019, pp. 2023-2028, doi: 10.1109/TENCON.2019.8929527.

5. G. Arutjothi ; C. Senthamarai ,” Prediction of loan status in commercial bank using machine learning classifier” 2018 International Conference Sustainable Systems(ICISS)

6. Ashlesha Vaidya , “Predictive and Probabilistic approach using Logistic Regression” 2017 8th International Conference on Computing, Communication and Networking Technologies

7. Kshitiz Gautam, Arun Pratap Singh, Keshav Tyagi, Mr. Suresh Kumar “Loan Prediction using Decision Tree and Random Forest” 2020 International Research Journal of Engineering and Technology (IRJET)

8. Kathe Rutika Pramod , Panhale Sakshi Dattatray , Avhad Pooja Prakash , Ghorpade Dinesh B. , Dapse Punam Laxman “ Prediction of Loan Approval using Machine Learning Algorithm : A Review Paper” 2021 International Research Journal of Engineering and Technology (IRJET)

9. https://blog.floydhub.com/naive-bayes-for-machine-learning/

10. https://www.javatpoint.com/logistic-regression-in-machine-learning

11. https://www.javatpoint.com/machine-learning-random-forest-algorithm

12. https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

13. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learnin g

PUBLICATION

Published in International Research Journal of Engineering and Technology (IRJET)

Paper Title: Loan Approval Prediction

Date of Publication: 13th April 2022

Volume: 09 **Issue:** 04

Link: https://www.irjet.net/archives/V9/i4/IRJET-V9I4118.pdf

ACKNOWLEDGEMENT

It gives us great pleasure in presenting this project synopsis report titled: “Loan Approval Prediction”. We express our gratitude to our project guide Prof. Amruta Sankhe, who provided us with all the guidance and encouragement and made the lab available to us at any time. We also would like to deeply express our sincere gratitude to the Project coordinators. We are eager and glad to express our gratitude to the Head of the Information Technology Dept. Prof. Deepali Maste, for her approval of this project. We are also thankful to her for providing us the needed assistance, detailed suggestions and encouragement to do the project. We would like to deeply express our sincere gratitude to our respected principal Prof. Dr. Shrikant Kallurkar and the management of Atharva College of Engineering for providing such an ideal atmosphere to build up this project with well-equipped library with all the utmost necessary reference materials and up to date IT Laboratories. We are extremely thankful to all staff and the management of the college for providing us all the facilities and resources required.