**Smart Lender - Applicant Credibility Prediction for Loan Approval**

**Industry Mentor(s) Name**: Sowjanya, Sandeep Doodigani

**Faculty Mentor(s) Name**: Dr. Ponsy R K Sathia Bhama

**Team Members:** Bhavani Venkata Karthik K (2019503511)

Amaresh Saddish (2019503004)

Tanooj Cheekati (2019503056)

Shreyas Karthik Ramesh (2019503050)

**Table Of Contents**

1. **INTRODUCTION** 
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   1. Feature 1
   2. Feature 2
   3. Feature 3
   4. Feature 4
8. **TESTING** 
   1. Test Cases
   2. User Acceptance Testing
9. **RESULTS**
   1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

**Introduction**

* 1. **Overview**

Big Data's main advantage is its ability to analyse a set of data in order to draw out valuable information. Volume, Velocity, and Variety are the attributes used to describe the data collected by various organisations and businesses. Sometimes, the precision and accuracy of the data depend not only on how accurate the data is but also on when it is delivered. Big data has the greatest potential in the financial industry. These data can be quickly transformed into decision-making tools with added value by using efficient analysis techniques.

The answer to the query of how to analyse this data is machine learning. Artificial intelligence offers features that cost less in terms of both time and resources when used. Additionally, it provides novel classification methods and new insights into data. Machine learning is currently being actively used in banking and financial institutions for a variety of purposes, including trading, portfolio management, fraud detection, and loan approval. Artificial intelligence is used in the financial industry to create "robo-advisors," intelligent portfolio systems that can adjust to the risks and objectives of the user. To make decisions very quickly, decision trading support systems, also known as algorithmic trading, are used. Machine learning has brought promising new approaches to analyse user behaviour and spot fraudulent transactions for fraud detection. Loan providers, especially big businesses like banks and insurance companies, train machine learning algorithms using a tonne of consumer data and financial lending outcomes. which can be used to decide wisely when it comes to insurance and lending. In this study, we focus on the latter: the decision-making process for loan approval.

According to a data analysis perspective, determining whether to approve a loan is a "good" or "bad" risk depends on the analysis and classification of a set of loan applicant data. A set of data is divided into two classes using the binary classification method, which is a straightforward classification case. When we want to predict a specific outcome that can only take two distinct values, we typically use this type of classification. Typical examples include detecting spam, diagnosing illnesses, looking for credit card fraud, or in our case, approving loans. Binary classification is a very basic problem, despite being fairly straightforward. Different paradigms are applied when learning binary classifiers, such as: Support Vector Machines, Decision Trees, K-nearest neighbourhood, Bayesian Classification, Logistic regression, Neural Networks, and more recently, deep learning.

* 1. **Purpose**

To develop a smart lending loan approval or rejection application that can automatically approve or reject a loan based on the user’s credit score. This increases the efficiency of the loan approval process and makes it less cumbersome and more economical.

**LITERATURE SURVEY**

**2.1 Existing Problem**

Currently the process is performed manually by dedicated experts who review credit history and application and manually approve or reject a loan. This is inefficient for a multitude of reasons. It makes the whole process cumbersome, complex and prone to mistakes. It is also not very economical or convenient. By automating this process with the help of Data Science and Machine Learning, all the existing problems can be easily overcome.

**2.2 References**

[1] V. Singh, A. Yadav, R. Awasthi and G. N. Partheeban, "Prediction of Modernized Loan Approval System Based on Machine Learning Approach," 2021 International Conference on Intelligent Technologies (CONIT), 2021, pp. 1-4, doi: 10.1109/CONIT51480.2021.9498475.

[2] H. Ramachandra, G. Balaraju, R. Divyashree and H. Patil, "Design and Simulation of Loan Approval Prediction Model using AWS Platform," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 53-56, doi: 10.1109/ESCI50559.2021.9397049.

[3] P. Tumuluru, L. R. Burra, M. Loukya, S. Bhavana, H. M. H. CSaiBaba and N. Sunanda, "Comparative Analysis of Customer Loan Approval Prediction using Machine Learning Algorithms," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), 2022, pp. 349-353, doi: 10.1109/ICAIS53314.2022.9742800.

[4] S. Barua, D. Gavandi, P. Sangle, L. Shinde and J. Ramteke, "Swindle: Predicting the Probability of Loan Defaults using CatBoost Algorithm," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1710-1715, doi: 10.1109/ICCMC51019.2021.9418277.

[5] V. Singh, A. Yadav, R. Awasthi and G. N. Partheeban, "Prediction of Modernized Loan Approval System Based on Machine Learning Approach," 2021 International Conference on Intelligent Technologies (CONIT), 2021, pp. 1-4, doi: 10.1109/CONIT51480.2021.9498475.

[6] M. Alaradi and S. Hilal, "Tree-Based Methods for Loan Approval," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-6, doi: 10.1109/ICDABI51230.2020.9325614.

[7] D. P. Rajesh, M. Alam, M. Tahernezhadi, C. Vikram and P. N. Phaneendra, "Real Time Data Science Decision Tree Approach to Approve Bank Loan from Lawyer’s Perspective," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 921-929, doi: 10.1109/ICMLA51294.2020.00150.

[8] K. Sirisong, P. Palangsantikul, P. Arpasat, S. Intarasema and S. Tumswadi, "Analysis of a Bank’s Lending Approval System using Process Mining," 2021 19th International Conference on ICT and Knowledge Engineering (ICT&KE), 2021, pp. 1-4, doi: 10.1109/ICTKE52386.2021.9665411.

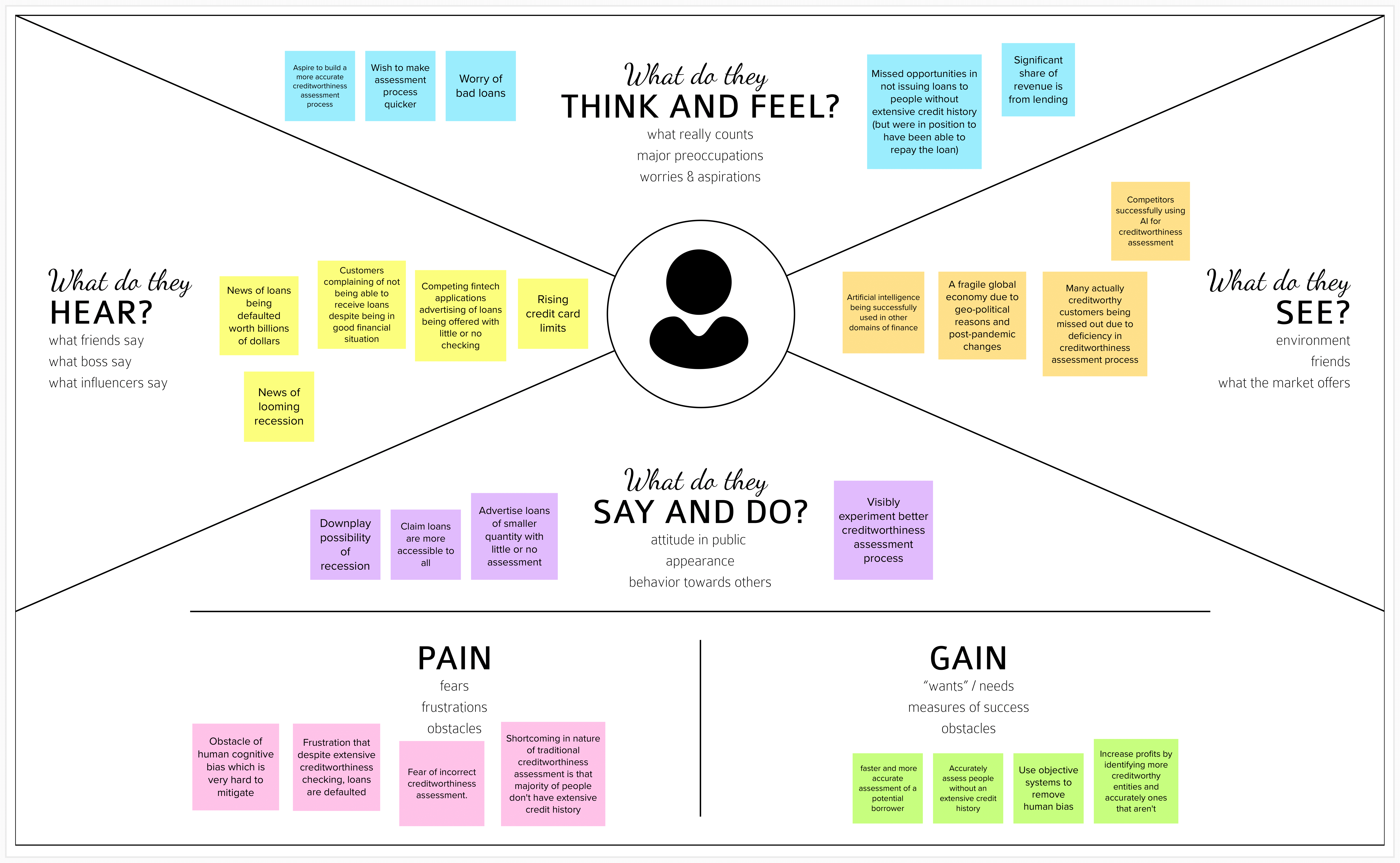
**2.3 PROBLEM STATEMENT DEFINITION**

* To accurately predict the credibility of a candidate for loan approval.
* Developing a loan approval system that can provide a possible reason for rejection
* Develop a loan approval system that can predict loan approval based on expert data.

**IDEATION AND PROPOSED SOLUTION**

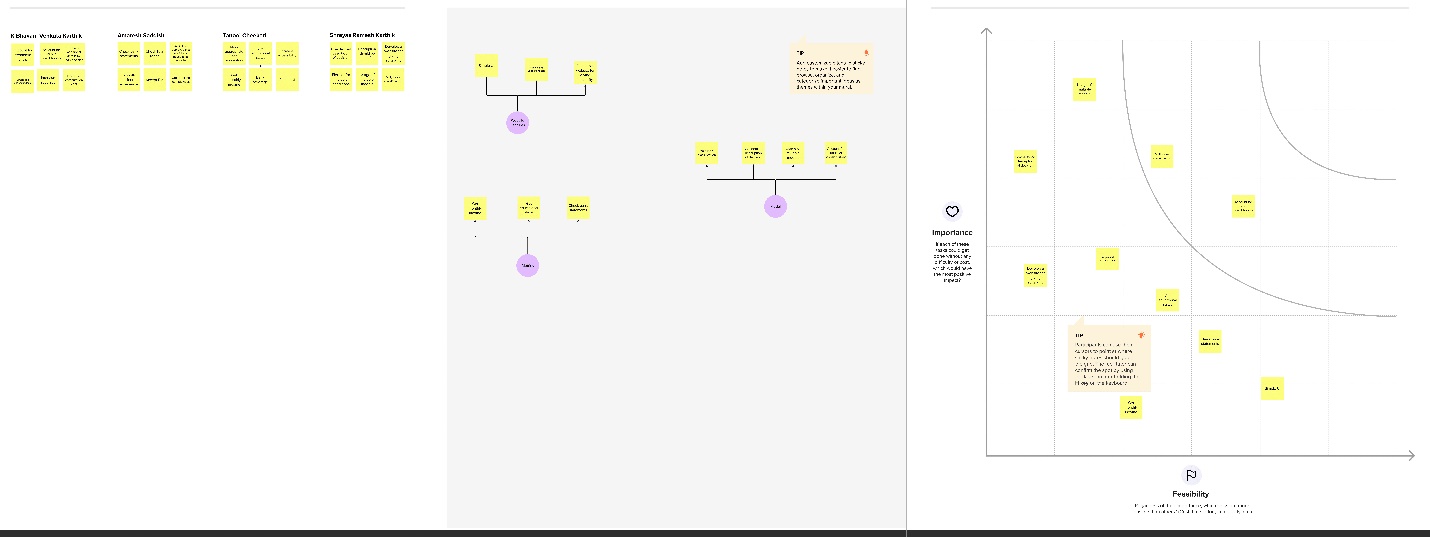
**3.1 EMPATHY MAP CANVAS**

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user‟s behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user‟s perspective along with his or her goals and challenges.



**Fig. 1 Empathy Map**

**3.2 IDEATION AND BRAINSTORMING**

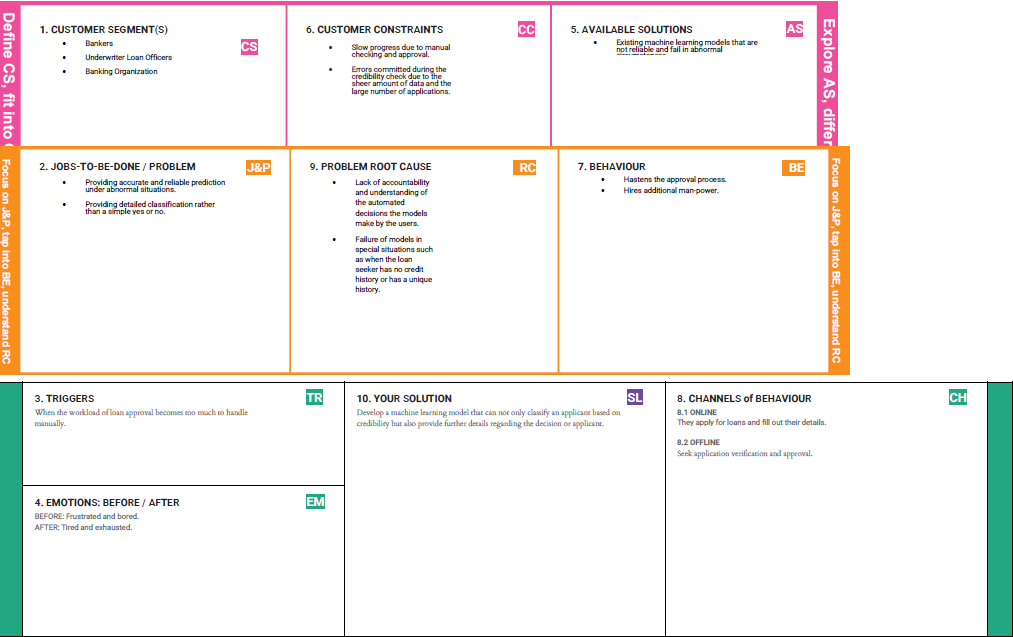
****

**Fig. 2 Brainstorming Ideas**

**3.3 PROPOSED SOLUTION**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | To accurately predict the credibility of a candidate for loan approval. |
|  | Idea / Solution description | Develop a machine learning model that can not only classify an applicant based on credibility but also provide further details regarding the decision or applicant. |
|  | Novelty / Uniqueness | The model will perform multi-label classification and provide additional parameters supporting the loan approval decision that it will give. |
|  | Social Impact / Customer Satisfaction | The product will help predict and automate loan approvals reliably and understand the decision as well. It will also account for uncommon circumstances and diverse situations. |
|  | Business Model (Revenue Model) | Freemium Model |
|  | Scalability of the Solution | The model can be adapted by any banking organization with ease. Classification will be generalized as much as possible but the process can be very customized. |

**3.4 PROBLEM SOLUTION FIT**

****

**Fig. 3 Problem Solution Fit**

**REQUIREMENT ANALYSIS**

**4.1 FUNCTIONAL REQUIREMENT**

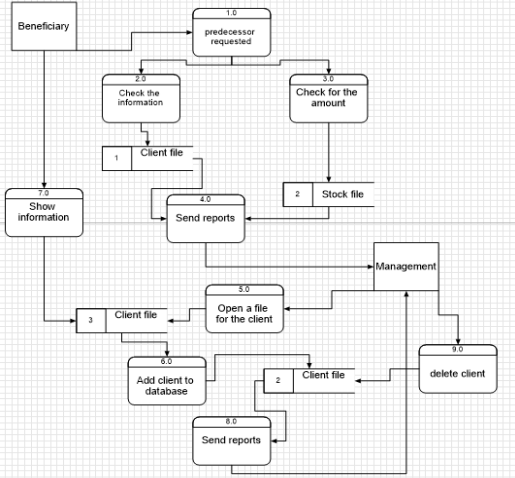
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form  Registration through Gmail  Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 | User Application | Filling of application  Modification of application  Verification of application |
| FR-4 | Loan Issuance | Checking status of loan  Loan Approval  Loan Rejection |
| FR-5 | Credit history analysis | Credit score auditing  Income auditing |
| FR-6 | User management | Choosing appropriate loan program for users  Categorising users according to credit history. |

**4.2 NON- FUNCTIONAL REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Simple and understandable UI.  Easy to navigate  Smooth and seamless  Easy to comprehend |
| NFR-2 | **Security** | Restricted access to data.  Login verification  Registration verification  Upholding privacy of user |
| NFR-3 | **Reliability** | Backup to prevent data loss  Negation of data loss due to lag. |
| NFR-4 | **Performance** | Web based application.  Requires minimum Intel Pentium 4 processor, 4 GB RAM, 1280x1024 screen with application window size 1024x680 |
| NFR-5 | **Availability** | Platform independent support. |
| NFR-6 | **Scalability** | Can operate efficiently across multiple devices with varying hardware and software specifications. |

**PROJECT DESIGN**

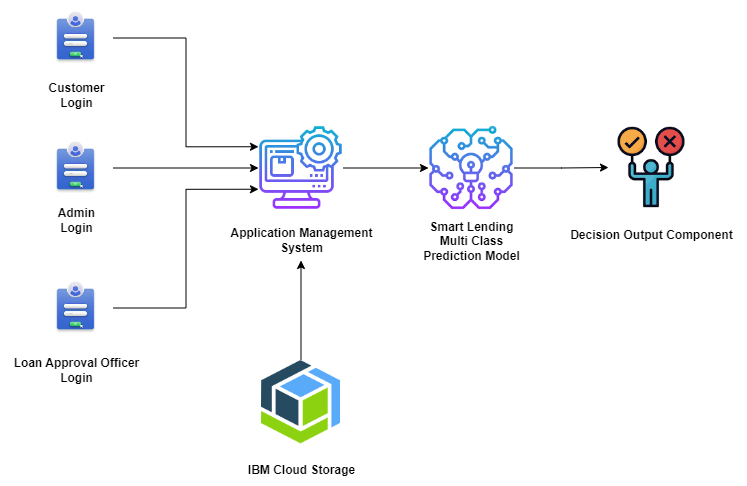
**5.1 DATA FLOW DIAGRAM**



**Fig. 4 Data Flow DIagram**

**5.2 SOLUTION AND TECHNICAL ARCHITECTURE**

An application should be developed which is able to take the details of the loan applicant, store them and utilize a machine learning model to predict the eligibility for loan approval based on the user application details and credit history. The user should be able to know his/her eligibility upon giving the required details to the application.



**Fig. 5 Solution Architecture**

**5.3 USER STORIES**

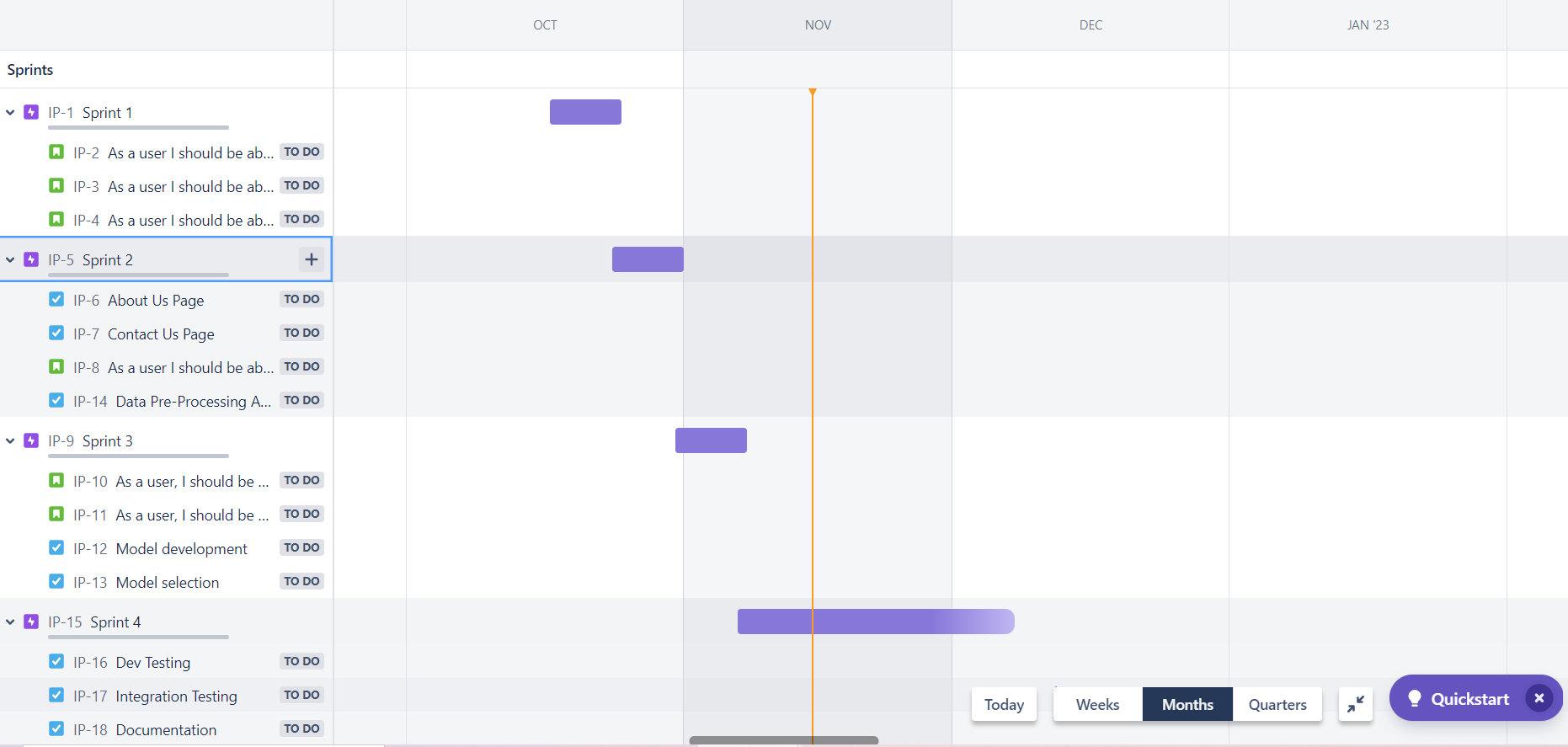
| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| --- | --- | --- | --- | --- | --- | --- |
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
|  |  | USN-3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low | Sprint-1 |
|  |  | USN-4 | As a user, I can register for the application through Gmail | I can receive confirmation email & click confirm | Medium | Sprint-1 |
|  | Login | USN-5 | As a user, I can log into the application by entering email & password | Able to login | High | Sprint-1 |
|  | Dashboard | USN-6 | As a user, I should be able to access the dashboard with everything I am allowed to use. | Access the dashboard | Medium | Sprint-1 |
| Customer (Web user) | Registration | USN-7 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN-8 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
|  |  | USN-9 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low | Sprint-1 |
|  |  | USN-10 | As a user, I can register for the application through Gmail | I can receive confirmation email & click confirm | Medium | Sprint-1 |
|  | Login | USN-11 | As a user, I can log into the application by entering email & password | Able to login | High | Sprint-1 |
|  | Dashboard | USN-12 | As a user, I should be able to access the dashboard with everything I am allowed to use. | Access the dashboard | Medium | Sprint-1 |
| Loan Approval Officer | Register | USN-13 | As a loan approval officer, I should be able to register myself as one using unique email and password. | I can access my account | Medium | Sprint-2 |
|  | Login | USN-14 | As a loan approval officer I should be able to login myself as one using unique email and password. | Access loan approval dashboard | Medium | Sprint-2 |
|  | Automated analysis of credit history | USN-15 | As a loan approval officer, I can access the dashboard where I feed application for loan prediction. | I can access the dashboard for loan application prediction. | High | Sprint-3 |
|  |  | USN-16 | As a loan approval officer, I can get a decision followed by some details for the decision when I feed an application for loan prediction. | Get a decision for loan prediction with details regarding the decision. | High | Sprint-3 |
| Admin | Register | USN-17 | As an admin, I should be able to register myself as one using unique email and password. | I can access my account | Medium | Sprint-4 |
|  | Login | USN-18 | As an admin I should be able to login myself as one using unique email and password. | Able to login | Medium | Sprint-4 |
|  | Dashboard | USN-19 | As a admin, I should be able to access the dashboard with everything I am allowed to use. | Access the dashboard | Medium | Sprint-4 |

**PROJECT PLANNING AND SCHEDULING**

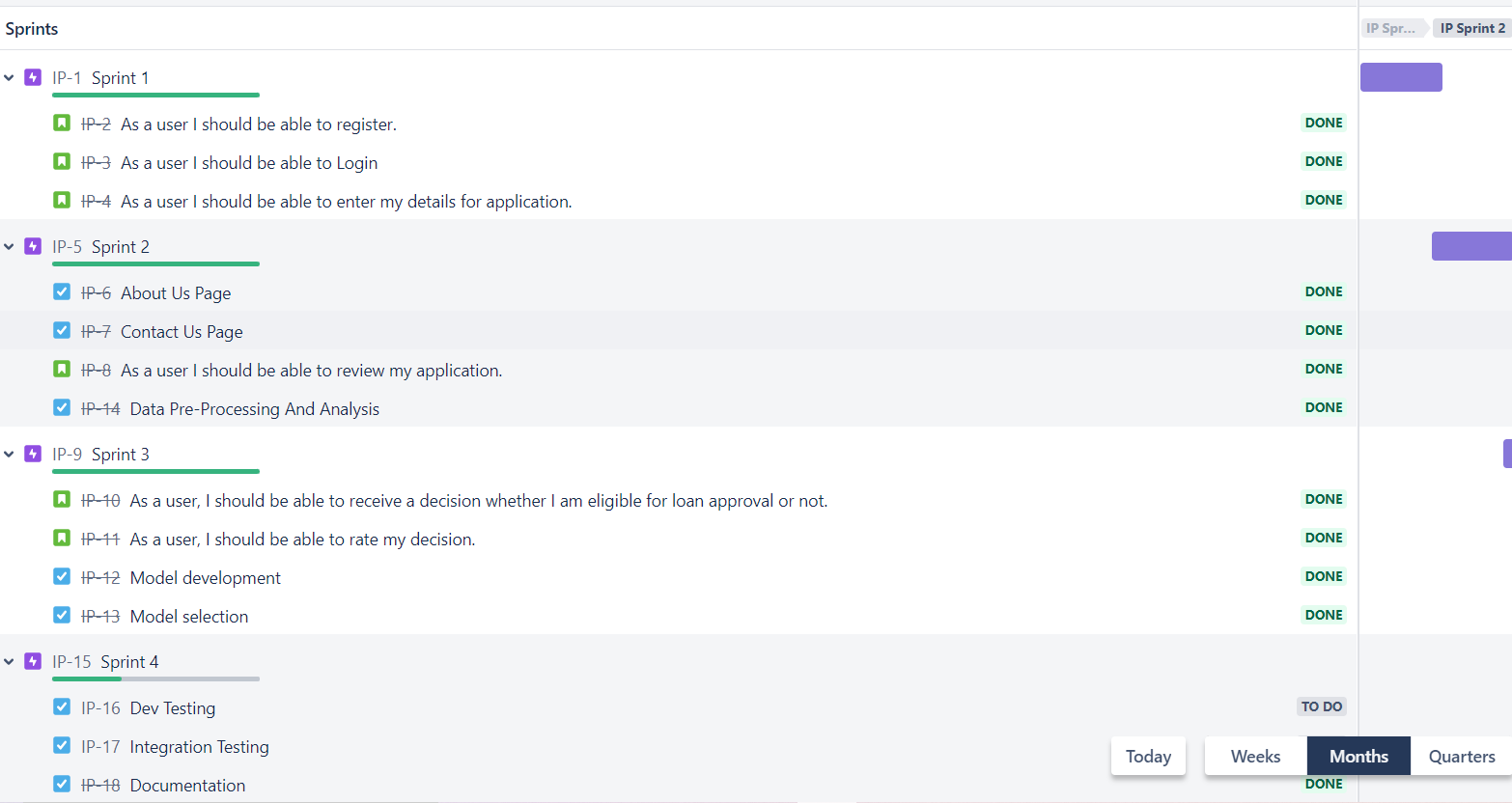
* 1. **SPRINT PLANNING AND ESTIMATION**

| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 3 | High | Amaresh Saddish, Tanooj Cheekati |
| Sprint-1 |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | 3 | High | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-1 |  | USN-3 | As a user, I can register for the application through Facebook | 1 | Low | Amaresh Saddish, Tanooj Cheekati |
| Sprint-1 |  | USN-4 | As a user, I can register for the application through Gmail | 2 | Medium | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-1 | Login | USN-5 | As a user, I can log into the application by entering email & password | 3 | High | Amaresh Saddish, Tanooj Cheekati |
| Sprint-1 | Dashboard | USN-6 | As a user, I should be able to access the dashboard with everything I am allowed to use. | 2 | Medium | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-1 | Registration | USN-7 | As a user, I can register for the application by entering my email, password, and confirming my password. | 3 | High | Amaresh Saddish, Tanooj Cheekati |
| Sprint-1 |  | USN-8 | As a user, I will receive confirmation email once I have registered for the application | 3 | High | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-1 |  | USN-9 | As a user, I can register for the application through Facebook | 1 | Low | Amaresh Saddish, Tanooj Cheekati |
| Sprint-1 |  | USN-10 | As a user, I can register for the application through Gmail | 2 | Medium | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-1 | Login | USN-11 | As a user, I can log into the application by entering email & password | 3 | High | Amaresh Saddish, Tanooj Cheekati |
| Sprint-1 | Dashboard | USN-12 | As a user, I should be able to access the dashboard with everything I am allowed to use. | 2 | Medium | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-2 | Register | USN-13 | As a loan approval officer, I should be able to register myself as one using unique email and password. | 5 | Medium | Amaresh Saddish, Tanooj Cheekati |
| Sprint-2 | Login | USN-14 | As a loan approval officer I should be able to login myself as one using unique email and password. | 5 | Medium | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-3 | Automated analysis of credit history | USN-15 | As a loan approval officer, I can access the dashboard where I feed application for loan prediction. | 10 | High | Amaresh Saddish, Tanooj Cheekati |
| Sprint-3 |  | USN-16 | As a loan approval officer, I can get a decision followed by some details for the decision when I feed an application for loan prediction. | 15 | High | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-4 | Register | USN-17 | As an admin, I should be able to register myself as one using unique email and password. | 2 | Medium | Amaresh Saddish, Tanooj Cheekati |
| Sprint-4 | Login | USN-18 | As an admin I should be able to login myself as one using unique email and password. | 2 | Medium | K Bhavani Venkata Karthik, Shreyas Karthik Ramesh |
| Sprint-4 | Dashboard | USN-19 | As a admin, I should be able to access the dashboard with everything I am allowed to use. | 2 | Medium | Amaresh Saddish, Tanooj Cheekati |

* 1. **SPRINT DELIVERY SCHEDULE**



**Fig. 6 Sprint Delivery Schedule**

**6.2 REPORTS FROM JIRA**  


**Fig. 7 Project Report**

**CODING AND SOLUTIONING**

**7.1 FEATURE 1-SIGN UP**

**Register.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Register</title>

<link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">

<link rel="stylesheet" type="text/css" href="static/register.css">

<script >

function check(x)

{

var number=/^([0-9]{10})+$/;

if(x.value.match(number)){

alert("Valid email address!");

document.myform.mon.focus();

return true;

}

else{

alert("Please enter only your 10 digit mobile number");

document.myform.mon.focus();

return false;

}

}

function ValidateEmail(input) {

var validRegex = /^[a-zA-Z0-9.!#$%&'\*+/=?^\_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)\*$/;

if (input.value.match(validRegex)) {

alert("Valid email address!");

document.myform.email.focus();

return true;

} else {

alert("Invalid email address!");

document.myform.email.focus();

return false;

}

}

</script>

</head>

<body>

<div class="container">

<form name="myform" method="post" class="form-signup" onsubmit="return check(document.myform.mon)" onsubmit="return ValidateEmail(document.myform.email)">

<h1 class="reg">Register</h1>

<p>Create your account</p>

<div class="form-group">

<input type="text" class="form-control" name="name" placeholder="Enter your name" required >

</div>

<div class="form-group">

<input type="email" class="form-control" name="email" placeholder="Enter your emailID" required>

</div>

<div class="form-group">

<input type="user name" class="form-control" name="username" placeholder="Enter your username" required>

</div>

<div class="form-group">

<input type="password" class="form-control" name="password" placeholder="Enter your password" required>

</div>

<div class="form-group">

<input type="text" class="form-control" name="mon" placeholder="Enter your mobile number" required >

</div>

<div class="form-group">

<label>

<input type="checkbox">

I accept the <a href="terms.html">Terms and conditions</a>

</label>

</div>

<input type="submit" class="btn btn-success btn-block" name="" value="submit">

</form>

</div>

</body>

</html>

**7.2 FEATURE 2-LOGIN**

**Login.html**

<!DOCTYPE html>

<html>

<head>

<title>LogIn Page</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<style>

.group{

padding-top: 100px;

}

</style>

<body>

<div class="container">

<div class="row">

<div style="width: 40%; margin: 25px auto;">

<div class="group">

<h3 style="text-align: center;">Login Page</h3>

<form action="login1.php" method="POST" >

<div class="form-group">

<label>UserName:</label><input type="text" name="username" class="form-control" autofocus placeholder="Enter your username or gmail ID" required="">

</div>

<div class="form-group">

<label>Password:</label><input type="Password" name="password" class="form-control" autofocus placeholder="Password" required="">

</div>

<label>Enter Captcha:</label>

<div class="form-row">

<div class="form-group col-md-6">

<input type="text" class="form-control" readonly id="capt" required="">

</div>

<div class="form-group col-md-6">

<input type="text" class="form-control" id="textinput" required="">

</div>

</div>

<div class="form-group">

<button onclick="validcap()" name="save" class="btn btn-lg btn-success btn-block" >Submit</button>

</div>

</form>

<h6>Captcha not visible <img src="refresh.png" width="40px" onclick="cap()"></h6>

<p>New Here?<a href="register.html">Register</a> </p>

</div>

</div>

</div>

</div>

<script type="text/javascript">

function cap(){

var alpha = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'

,'W','X','Y','Z','1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i',

'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z', '!','@','#','$','%','^','&','\*','+'];

var a = alpha[Math.floor(Math.random()\*71)];

var b = alpha[Math.floor(Math.random()\*71)];

var c = alpha[Math.floor(Math.random()\*71)];

var d = alpha[Math.floor(Math.random()\*71)];

var e = alpha[Math.floor(Math.random()\*71)];

var f = alpha[Math.floor(Math.random()\*71)];

var final = a+b+c+d+e+f;

document.getElementById("capt").value=final;

}

function validcap(){

var stg1 = document.getElementById('capt').value;

var stg2 = document.getElementById('textinput').value;

if(stg1==stg2){

// alert("Form is validated Succesfully");

return true;

}else{

alert("Please enter a valid captcha");

return false;

}

}

</script>

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

**7.3 FEATURE 3 – BANK LOGIN**

**Bank login.html**

<!DOCTYPE html>

<html>

<head>

<title>LogIn Page</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<style>

.group{

padding-top: 100px;

}

</style>

<body>

<div class="container">

<div class="row">

<div style="width: 40%; margin: 25px auto;">

<div class="group">

<h3 style="text-align: center;">Bank Login Page</h3>

<form method="POST" action="bank1.php">

<div class="form-group">

<label>Bank user ID:</label><input type="text" name="BankUserName" class="form-control" autofocus placeholder="Enter the Bank User ID" required>

</div>

<div class="form-group">

<label>Bank Email ID:</label><input type="email" name="bankemail" class="form-control" autofocus placeholder="Enter the Bank Email ID" required>

</div>

<div class="form-group">

<label>Password:</label><input type="Password" name="Password" class="form-control" autofocus placeholder="Password" required>

</div>

<label>Enter Captcha:</label>

<div class="form-row">

<div class="form-group col-md-6">

<input type="text" class="form-control" readonly id="capt" required>

</div>

<div class="form-group col-md-6">

<input type="text" class="form-control" id="textinput" required>

</div>

</div>

<div class="form-group">

<button onclick="validcap()" name="Submit" class="btn btn-lg btn-success btn-block">Submit</button>

</div>

</form>

<h6>Captcha not visible <img src="refresh.png" width="40px" onclick="cap()"></h6>

</div>

</div>

</div>

</div>

<script type="text/javascript">

function cap(){

var alpha = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'

,'W','X','Y','Z','1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i',

'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z', '!','@','#','$','%','^','&','\*','+'];

var a = alpha[Math.floor(Math.random()\*71)];

var b = alpha[Math.floor(Math.random()\*71)];

var c = alpha[Math.floor(Math.random()\*71)];

var d = alpha[Math.floor(Math.random()\*71)];

var e = alpha[Math.floor(Math.random()\*71)];

var f = alpha[Math.floor(Math.random()\*71)];

var final = a+b+c+d+e+f;

document.getElementById("capt").value=final;

}

function validcap(){

var stg1 = document.getElementById('capt').value;

var stg2 = document.getElementById('textinput').value;

if(stg1==stg2){

alert("Form is validated Succesfully");

return true;

}else{

alert("Please enter a valid captcha");

return false;

}

}

</script>

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

**7.4 FEATURE 4 – LOAN APPROVAL OR REJECTION**

**Approve.html**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Loan approva status</title>

<link rel="stylesheet" href="static/approve.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<h1>LOAN APPROVAL STATUS</h1>

<h2>{{prediction\_text}}</h2>

<img src="static/approve.jpg" width="150px" height="150px">

<h3>Please provide your feedback</h3>

<div class="container">

<div class="post">

<div class="text">Thanks for rating us!</div>

<div class="edit">EDIT</div>

</div>

<div class="star-widget">

<input type="radio" name="rate" id="rate-5">

<label for="rate-5" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-4">

<label for="rate-4" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-3">

<label for="rate-3" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-2">

<label for="rate-2" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-1">

<label for="rate-1" class="fas fa-star"></label>

<form action="#">

<header></header>

<div class="textarea">

<textarea cols="30" placeholder="Describe your experience.."></textarea>

</div>

<div class="btn">

<button type="submit">Post</button>

</div>

</form>

</div>

</div>

<script>

const btn = document.querySelector("button");

const post = document.querySelector(".post");

const widget = document.querySelector(".star-widget");

const editBtn = document.querySelector(".edit");

btn.onclick = ()=>{

widget.style.display = "none";

post.style.display = "block";

editBtn.onclick = ()=>{

widget.style.display = "block";

post.style.display = "none";

}

return false;

}

</script>

</body>

</html>

**Reject.html**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Loan approval status</title>

<link rel="stylesheet" href="static/reject.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<h1>LOAN APPROVAL STATUS</h1>

<h2>{{prediction\_text}}</h2>

<img src="static/reject.jpg" width="200px" height="200px">

<h3>Please provide your feedback</h3>

<div class="container">

<div class="post">

<div class="text">Thanks for rating us!</div>

<div class="edit">EDIT</div>

</div>

<div class="star-widget">

<input type="radio" name="rate" id="rate-5">

<label for="rate-5" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-4">

<label for="rate-4" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-3">

<label for="rate-3" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-2">

<label for="rate-2" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-1">

<label for="rate-1" class="fas fa-star"></label>

<form action="#">

<header></header>

<div class="textarea">

<textarea cols="30" placeholder="Describe your experience.."></textarea>

</div>

<div class="btn">

<button type="submit">Post</button>

</div>

</form>

</div>

</div>

<script>

const btn = document.querySelector("button");

const post = document.querySelector(".post");

const widget = document.querySelector(".star-widget");

const editBtn = document.querySelector(".edit");

btn.onclick = ()=>{

widget.style.display = "none";

post.style.display = "block";

editBtn.onclick = ()=>{

widget.style.display = "block";

post.style.display = "none";

}

return false;

}

</script>

</body>

</html>

**Prediction.html**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="static/prediction.css">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">

<link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">

<title>prediction</title>

</head>

<body>

<script>

function valid(){

var Ai=document.getElementById("ApplicantIncome").value;

var Co=document.getElementById("CoapplicantIncome").value;

var LA=document.getElementById("LoanAmount").value;

var LT=document.getElementById("Loan\_Amount\_Term").value;

if(Ai > 100000000000000000000000000000000000000){

alert("Applicant income is too large enter a valid number")

return false;

}

if(Co > 100000000000000000000000000000000000000){

alert("Coapplicant income is too large enter a valid number")

return false;

}

if(LA > 100000000000000000000000000000000000000){

alert("Loan Amount is too large enter a valid number")

return false;

}

if(LT > 100000000000000000000000000000000000000){

alert("loan amount term is too large enter a valid number")

return false;

}

var name=document.getElementById("Name").value;

var letters=/^[a-zA-Z]\*$/;

if(!name.match(letters)){

alert("Name must contain only alphabets")

return false;

}

var num=/^[0-9]+$/;

if(!Ai.match(num)){

alert("Enter only valid numbers alphabets are not allowed ")

return false;

}

if(!Co.match(num)){

alert("Enter only valid numbers alphabets are not allowed ")

return false;

}

if(!LA.match(num)){

alert("Enter only valid numbers alphabets are not allowed ")

return false;

}

if(!LT.match(num)){

alert("Enter only valid numbers alphabets are not allowed ")

return false;

}

var mo=document.getElementById("mon").value;

var mn=/^[0-9]{10}$/;

if(!mo.match(mn)){

alert("Please enter only 10 digit mobile number")

return false;

}

}

</script>

<section class="text-gray-600 body-font">

<div class="container px-5 py-24 mx-auto">

<div class="flex flex-col text-center w-full mb-20">

<h1 class="Heading">LOAN ELIGIBILITY PREDICTION</h1><br>

<p class="fill">Fill the form for prediction</p>

</div>

<div>

</div>

<div class="mb-3">

<a class="btn btn-primary" href="./" id="back" role="button">Back</a></div>

<form action='/prediction.html' method="post" onsubmit="return valid()">

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Name</label>

<input type="text" class="form-control" id="Name" name="Name" placeholder="Enter your Name" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Email ID</label>

<input type="email" class="form-control" id="email" name="email" placeholder="Enter your Email ID" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Mobile Number</label>

<input type="text" class="form-control" id="mon" name="mon" placeholder="Enter your Mobile number" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Gender</label>

<select class="form-select" id="gender" name="gender" aria-label="Default select example" required >

<option selected>-- select gender --</option>

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Married status</label>

<select class="form-select" id="married" name="married" aria-label="Default select example" required >

<option selected>-- select married status --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Dependents</label>

<select class="form-select" id="dependents" name="dependents" aria-label="Default select example" required>

<option selected>-- select dependents --</option>

<option value="0">0</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3+">3+</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Education</label>

<select class="form-select" id="education" name="education" aria-label="Default select example" required>

<option selected>-- select education --</option>

<option value="Graduate">Graduate</option>

<option value="Not Graduate">Not Graduate</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Self\_Employed</label>

<select class="form-select" id="employed" name="employed" aria-label="Default select example" required>

<option selected>-- select Self\_Employed --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Credit\_History</label>

<select class="form-select" id="credit" name="credit" aria-label="Default select example" required >

<option selected >-- select Credit\_History --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Property\_Area</label>

<select class="form-select" id="proparea" name="proparea" aria-label="Default select example" required>

<option selected>-- select Property\_Area --</option>

<option value="Semiurban">Semiurban</option>

<option value="Urban">Urban</option>

<option value="Rural">Rural</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter ApplicantIncome</label>

<input type="text" class="form-control" id="ApplicantIncome" name="ApplicantIncome" placeholder="ApplicantIncome" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter CoapplicantIncome</label>

<input type="text" class="form-control" id="CoapplicantIncome" name="CoapplicantIncome" placeholder="CoapplicantIncome" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Purpose of loan</label>

<select class="form-select" id="pur" name="pur" aria-label="Default select example" required>

<option selected>-- select the purpose of loan --</option>

<option value="person">Personal loan</option>

<option value="Bussiness">Bussiness loan</option>

<option value="Education">Education loan</option>

<option value="Home">Home loan</option>

<option value="Other">other</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter LoanAmount</label>

<input type="text" class="form-control" id="LoanAmount" name="LoanAmount" placeholder="LoanAmount" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter Loan\_Amount\_Term</label>

<input type="text" class="form-control" id="Loan\_Amount\_Term" name="Loan\_Amount\_Term" placeholder="Loan\_Amount\_Term" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter Adhar Number</label>

<input type="text" class="form-control" id="Adhar" name="Adhar" placeholder="Adhar Number" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter PAN card ID</label>

<input type="text" class="form-control" id="PAN " name="PAN " placeholder="PAN card ID" required>

</div>

<div class="mb-3">

<label for="property document" class="form-label">Property Document</label><br><input type="file" required >

</div>

<div class="mb-3">

<label for="Govt ID proof" class="form-label">Govet ID proof</label><br><input type="file" required>

</div>

<div class="mb-3">

<input type="checkbox" required>

I accept the <a href="terms.html">Terms and conditions</a>

</div>

<br><br>

<div class="mb-3">

<input type="submit" class="but" value="PREDICT">

</div>

</form>

</div>

</section>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf" crossorigin="anonymous"></script>

</body>

</html>

**Model.ipynb**

import numpy as np

import pandas as pd

import pickle

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

import sklearn

from sklearn.preprocessing import LabelEncoder

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from xgboost import XGBClassifier

from sklearn.ensemble import RandomForestClassifier

import imblearn

from imblearn.under\_sampling import RandomUnderSampler

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import scale

from sklearn.preprocessing import MaxAbsScaler

from sklearn.metrics import accuracy\_score,classification\_report,confusion\_matrix,f1\_score

## Reading The Dataset

df=pd.read\_csv('Loan\_dataset.csv')

df

df.head()

df.info()

df.shape

df=df.drop(columns=["Loan\_ID"],axis=1)

## Uni-Variate Analysis

sns.distplot(df.ApplicantIncome)

sns.countplot(df.Property\_Area)

sns.countplot(df.Gender)

sns.countplot(df.Education)

sns.countplot(df.Self\_Employed)

sns.countplot(df.Married)

sns.histplot(df.LoanAmount)

sns.displot(df.CoapplicantIncome)

plt.pie(df.Property\_Area.value\_counts(),[0,0,0],labels=['Semi urban','Urban','Rural'])

## Bivariate Analysis

sns.countplot(df['Married'],hue=df['Gender'])

sns.countplot(df['ApplicantIncome'],hue=df['CoapplicantIncome'])

sns.countplot(df['LoanAmount'],hue=df['Property\_Area'])

sns.countplot(df['Education'],hue=df['Self\_Employed'])

sns.barplot(df.ApplicantIncome,df.CoapplicantIncome)

sns.countplot(df['LoanAmount'],hue=df['Loan\_Amount\_Term'])

plt.scatter(df.ApplicantIncome,df.LoanAmount)

sns.countplot(df['Dependents'],hue=df['Gender'])

## Multi variate Analysis

sns.heatmap(df.corr(),annot=True)

plt.plot(df.LoanAmount,df.Loan\_Amount\_Term,df.ApplicantIncome)

df.plot.line()

df.hist()

plt.plot(df.LoanAmount,df.ApplicantIncome,df.CoapplicantIncome)

plt.plot(df.Loan\_Amount\_Term,df.ApplicantIncome,df.CoapplicantIncome)

## Descriptive Analysis

df.describe()

df.std()

df.mean()

df.mode()

## Data Pre-Processing

## Check for Null Values

df.isnull().any()

df.isnull().sum()

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.isnull().any()

df.isnull().sum()

## Handling Categorical Values

df.head()

le=LabelEncoder()

df.Gender=le.fit\_transform(df.Gender)

df.Married=le.fit\_transform(df.Married)

df.Education=le.fit\_transform(df.Education)

df.Self\_Employed=le.fit\_transform(df.Self\_Employed)

df.Property\_Area=le.fit\_transform(df.Property\_Area)

df.Loan\_Status=le.fit\_transform(df.Loan\_Status)

df.Dependents=le.fit\_transform(df.Dependents)

df.head()

## Spliting into dependent and independent data

df.head()

x=df.iloc[:,:-1]

y=df.Loan\_Status

x.head()

y.head()

## Scaling The Data

scaler = MaxAbsScaler()

x\_sc=scaler.fit\_transform(x)

x\_sc

## Balancing The Dataset

sns.countplot(df.Loan\_Status)

rus=RandomUnderSampler(sampling\_strategy=1)

x\_res,y\_res=rus.fit\_resample(x,y)

ax=y\_res.value\_counts().plot.pie(autopct='%.2f')

\_=ax.set\_title("under-sampling")

## Splitting Data Into Train and Test

xtrain,xtest,ytrain,ytest=train\_test\_split(x,y,test\_size=0.3,random\_state=10)

xtrain.head()

xtest.head()

ytrain.head()

ytest.head()

xtrain.shape

xtest.shape

ytrain.shape

ytest.shape

## Model Building

## Decision Tree Model

dmodel=DecisionTreeClassifier(random\_state=100)

dmodel.fit(x\_res,y\_res)

ypredd=dmodel.predict(xtest)

ypred2d=dmodel.predict(xtrain)

## Random Forest Model

Rmodel=RandomForestClassifier(n\_estimators=100,max\_depth=12,max\_features=3)

Rmodel.fit(x\_res,y\_res)

ypredR=Rmodel.predict(xtest)

ypred2R=Rmodel.predict(xtrain)

## KNN Model

kmodel=KNeighborsClassifier()

kmodel.fit(x\_res,y\_res)

ypredk=kmodel.predict(xtest)

ypred2k=kmodel.predict(xtrain)

## Xgboost Model

xmodel=XGBClassifier(eval\_metric='mlogloss',n\_estimators=100,random\_state=100)

xmodel.fit(x\_res,y\_res)

ypredx=xmodel.predict(xtest)

ypred2x=xmodel.predict(xtrain)

## Compare The Model

print("Decision Tree Model Testing Accuracy")

print(accuracy\_score(ytest,ypredd))

print("Decision Tree Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2d))

print("Random Forest Model Testing Accuracy")

print(accuracy\_score(ytest,ypredR))

print("Random Forest Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2R))

print("KNN Model Testing Accuracy")

print(accuracy\_score(ytest,ypredk))

print("KNN Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2k))

print("Xgboost Model Testing Accuracy")

print(accuracy\_score(ytest,ypredx))

print("Xgboost Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2x))

## Evaluating Performance Of The Model And Saving The Model

## Random Forest Model is Selected

print("Random Forest Model Testing Accuracy")

print(accuracy\_score(ytest,ypredR))

print("Random Forest Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2R))

y=Rmodel.predict([[1,1,1,1,0,4583.0,1508.0,128.0,360.0,1.0,0]])

print(y)

y1=Rmodel.predict([[1,0,0,0,0,5849,0.0,146.412162,360.0,1.0,2]])

print(y1)

y2=Rmodel.predict([[1,0,0,1,0,678,987,90,24,1.0,2]])

print(y2)

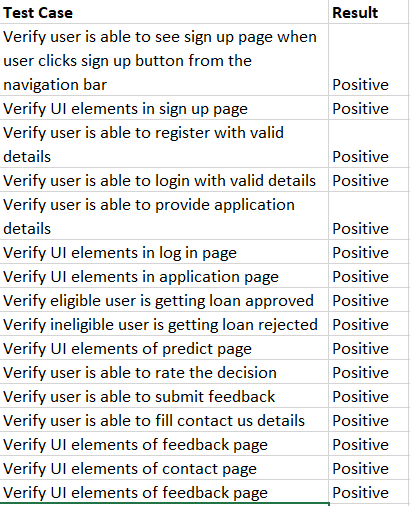
f1\_score(ypredR,ytest,average='weighted')

pd.crosstab(ytest,ypredR)

print(classification\_report(ypredR,ytest))

**TESTING**

**8.1 TEST CASES**

****

**8.2 USER ACCEPTANCE TESTING**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Feature Type** | **Component** | **Test Scenario** | **Steps To Execute** |
| SignUpPage\_TC\_001 | Feature | Sign up page | Verify user is able to see sign up page | 1. Enter Url 2. Click sign up 3. Fill details |
| SignUpPage\_TC\_002 | UI | Sign up page | Verify sign up page UI elements | 1. Enter Url 2. Click sign up 3. Check UI elements match requirements. |
| LoginPage\_TC\_001 | Feature | Login page | Verify user is able to see login page | 1. Enter Url 2. Click login up 3. Fill details |
| LoginPage\_TC\_002 | UI | Login page | Verify login page UI elements | 1. Enter Url 2. Click login up 3. Check UI elements match requirements. |
| ApplicationPage\_TC\_001 | Feature | Application page | Verify user is able to see application page | 1. Enter Url 2. Click application 3. Fill details |
| ApplicationPage\_TC\_002 | UI | Application page | Verify application page UI elements | 1. Enter Url 2. Click application 3. Check UI elements match requirements. |
| PredictionPage\_TC\_001 | Feature | Approve page | Verify user is able to see prediction approval page | 1. Enter Url 2. Prediction 3. Check prediction |
| PredictionPage\_TC\_002 | Feature | Reject page | Verify user is able to see prediction rejection page | 1. Enter Url 2. Prediction 3. Check prediction |
| PredictionPage\_TC\_003 | UI | Prediction page | Verify prediction page UI elements | 1. Enter Url 2. Click predict 3. Check UI elements match requirements. |
| RatingPage\_TC\_001 | Feature | Rating page | Verify user is able to see rating page | 1. Enter Url 2. Prediction 3. Check prediction 4. Give Rating |
| RatingPage\_TC\_002 | UI | Rating page | Verify rating page UI elements | 1. Enter Url 2. Click predict 3. Check UI elements match requirements. |
| FeedbackPage\_TC\_001 | Feature | Feedback page | Verify user is able to see feedback page | 1. Enter Url 2. Go to feedback page 3. Fill details |
| FeedbackPage\_TC\_002 | UI | Feedback page | Verify feedback page UI elements | 1. Enter Url 2. Click feedback 3. Check UI elements match requirements. |
| ContactUsPage\_TC\_001 | Feature | Contact us page | Verify user is able to see contact us page | 1. Enter Url 2. Go to contact us page 3. Fill details |
| ContactUsPage\_TC\_002 | UI | Contact us page | Verify contact us page UI elements | 1. Enter Url 2. Click contact us 3. Check UI elements match requirements. |

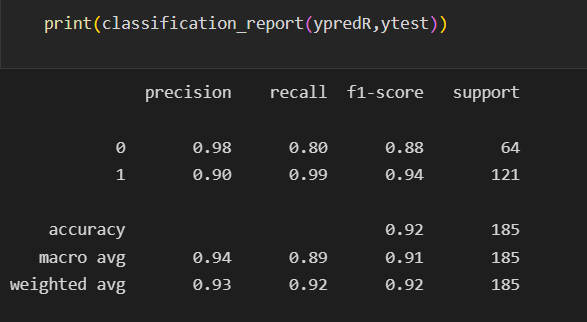
**RESULTS**

**9.1 PERFORMANCE METRICS**

There are numerous metrics used to assess the capability of a machine learning model. Precision, recall, fi-score, support, accuracy, macro average and weighted average have been used to assess the performance of the machine learning application.

Precision is one indicator of a machine learning model's performance – the quality of a positive prediction made by the model. Precision refers to the number of true positives divided by the total number of positive predictions (i.e., the number of true positives plus the number of false positives). The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The F1 score is defined as the harmonic mean of precision and recall.

Support may be defined as the number of samples of the true response that lies in each class of target values. Accuracy is one metric for evaluating classification models. It is the fraction of predictions our model got right. Macro average is the simple mean of scores of all classes. Weighted average or weighted sum ensemble is an ensemble machine learning approach that combines the predictions from multiple models, where the contribution of each model is weighted proportionally to its capability or skill.



**Fig. 8 Performance Metrics**

**ADVANTAGES AND DISADVANTAGES**

**10.1 ADVANTAGES**

* The process is now entirely automated.
* Process is streamlined and efficient.
* Prediction have a high degree of accuracy, avoiding errors.
* Predictions are swift and almost instant, reducing time taken to perform the eligibility process.
* Process is decentralized and economical.

**10.2 DISADVANTAGES**

* May give unexpected result for anomalous credit history.

**CONCLUSION**

Loan eligibility determination is a sensitive, vital yet cumbersome process. A lot of resources are drained to ensure the process is continuous without any interruption or error when done manually. With the help of machine learning, the entire process has been fully automated and made more resourceful, economical, convenient and efficient.

**FUTURE SCOPE**

* Decentralized integration with banks as an API service.
* Detection of loan frauds.

**APPENDIX**

**13.1 SOURCE CODE**

**Register.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Register</title>

<link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">

<link rel="stylesheet" type="text/css" href="static/register.css">

<script >

function check(x)

{

var number=/^([0-9]{10})+$/;

if(x.value.match(number)){

alert("Valid email address!");

document.myform.mon.focus();

return true;

}

else{

alert("Please enter only your 10 digit mobile number");

document.myform.mon.focus();

return false;

}

}

function ValidateEmail(input) {

var validRegex = /^[a-zA-Z0-9.!#$%&'\*+/=?^\_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)\*$/;

if (input.value.match(validRegex)) {

alert("Valid email address!");

document.myform.email.focus();

return true;

} else {

alert("Invalid email address!");

document.myform.email.focus();

return false;

}

}

</script>

</head>

<body>

<div class="container">

<form name="myform" method="post" class="form-signup" onsubmit="return check(document.myform.mon)" onsubmit="return ValidateEmail(document.myform.email)">

<h1 class="reg">Register</h1>

<p>Create your account</p>

<div class="form-group">

<input type="text" class="form-control" name="name" placeholder="Enter your name" required >

</div>

<div class="form-group">

<input type="email" class="form-control" name="email" placeholder="Enter your emailID" required>

</div>

<div class="form-group">

<input type="user name" class="form-control" name="username" placeholder="Enter your username" required>

</div>

<div class="form-group">

<input type="password" class="form-control" name="password" placeholder="Enter your password" required>

</div>

<div class="form-group">

<input type="text" class="form-control" name="mon" placeholder="Enter your mobile number" required >

</div>

<div class="form-group">

<label>

<input type="checkbox">

I accept the <a href="terms.html">Terms and conditions</a>

</label>

</div>

<input type="submit" class="btn btn-success btn-block" name="" value="submit">

</form>

</div>

</body>

</html>

**Login.html**

<!DOCTYPE html>

<html>

<head>

<title>LogIn Page</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<style>

.group{

padding-top: 100px;

}

</style>

<body>

<div class="container">

<div class="row">

<div style="width: 40%; margin: 25px auto;">

<div class="group">

<h3 style="text-align: center;">Login Page</h3>

<form action="login1.php" method="POST" >

<div class="form-group">

<label>UserName:</label><input type="text" name="username" class="form-control" autofocus placeholder="Enter your username or gmail ID" required="">

</div>

<div class="form-group">

<label>Password:</label><input type="Password" name="password" class="form-control" autofocus placeholder="Password" required="">

</div>

<label>Enter Captcha:</label>

<div class="form-row">

<div class="form-group col-md-6">

<input type="text" class="form-control" readonly id="capt" required="">

</div>

<div class="form-group col-md-6">

<input type="text" class="form-control" id="textinput" required="">

</div>

</div>

<div class="form-group">

<button onclick="validcap()" name="save" class="btn btn-lg btn-success btn-block" >Submit</button>

</div>

</form>

<h6>Captcha not visible <img src="refresh.png" width="40px" onclick="cap()"></h6>

<p>New Here?<a href="register.html">Register</a> </p>

</div>

</div>

</div>

</div>

<script type="text/javascript">

function cap(){

var alpha = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'

,'W','X','Y','Z','1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i',

'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z', '!','@','#','$','%','^','&','\*','+'];

var a = alpha[Math.floor(Math.random()\*71)];

var b = alpha[Math.floor(Math.random()\*71)];

var c = alpha[Math.floor(Math.random()\*71)];

var d = alpha[Math.floor(Math.random()\*71)];

var e = alpha[Math.floor(Math.random()\*71)];

var f = alpha[Math.floor(Math.random()\*71)];

var final = a+b+c+d+e+f;

document.getElementById("capt").value=final;

}

function validcap(){

var stg1 = document.getElementById('capt').value;

var stg2 = document.getElementById('textinput').value;

if(stg1==stg2){

// alert("Form is validated Succesfully");

return true;

}else{

alert("Please enter a valid captcha");

return false;

}

}

</script>

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

**Bank login.html**

<!DOCTYPE html>

<html>

<head>

<title>LogIn Page</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<style>

.group{

padding-top: 100px;

}

</style>

<body>

<div class="container">

<div class="row">

<div style="width: 40%; margin: 25px auto;">

<div class="group">

<h3 style="text-align: center;">Bank Login Page</h3>

<form method="POST" action="bank1.php">

<div class="form-group">

<label>Bank user ID:</label><input type="text" name="BankUserName" class="form-control" autofocus placeholder="Enter the Bank User ID" required>

</div>

<div class="form-group">

<label>Bank Email ID:</label><input type="email" name="bankemail" class="form-control" autofocus placeholder="Enter the Bank Email ID" required>

</div>

<div class="form-group">

<label>Password:</label><input type="Password" name="Password" class="form-control" autofocus placeholder="Password" required>

</div>

<label>Enter Captcha:</label>

<div class="form-row">

<div class="form-group col-md-6">

<input type="text" class="form-control" readonly id="capt" required>

</div>

<div class="form-group col-md-6">

<input type="text" class="form-control" id="textinput" required>

</div>

</div>

<div class="form-group">

<button onclick="validcap()" name="Submit" class="btn btn-lg btn-success btn-block">Submit</button>

</div>

</form>

<h6>Captcha not visible <img src="refresh.png" width="40px" onclick="cap()"></h6>

</div>

</div>

</div>

</div>

<script type="text/javascript">

function cap(){

var alpha = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'

,'W','X','Y','Z','1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i',

'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z', '!','@','#','$','%','^','&','\*','+'];

var a = alpha[Math.floor(Math.random()\*71)];

var b = alpha[Math.floor(Math.random()\*71)];

var c = alpha[Math.floor(Math.random()\*71)];

var d = alpha[Math.floor(Math.random()\*71)];

var e = alpha[Math.floor(Math.random()\*71)];

var f = alpha[Math.floor(Math.random()\*71)];

var final = a+b+c+d+e+f;

document.getElementById("capt").value=final;

}

function validcap(){

var stg1 = document.getElementById('capt').value;

var stg2 = document.getElementById('textinput').value;

if(stg1==stg2){

alert("Form is validated Succesfully");

return true;

}else{

alert("Please enter a valid captcha");

return false;

}

}

</script>

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

**Approve.html**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Loan approva status</title>

<link rel="stylesheet" href="static/approve.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<h1>LOAN APPROVAL STATUS</h1>

<h2>{{prediction\_text}}</h2>

<img src="static/approve.jpg" width="150px" height="150px">

<h3>Please provide your feedback</h3>

<div class="container">

<div class="post">

<div class="text">Thanks for rating us!</div>

<div class="edit">EDIT</div>

</div>

<div class="star-widget">

<input type="radio" name="rate" id="rate-5">

<label for="rate-5" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-4">

<label for="rate-4" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-3">

<label for="rate-3" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-2">

<label for="rate-2" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-1">

<label for="rate-1" class="fas fa-star"></label>

<form action="#">

<header></header>

<div class="textarea">

<textarea cols="30" placeholder="Describe your experience.."></textarea>

</div>

<div class="btn">

<button type="submit">Post</button>

</div>

</form>

</div>

</div>

<script>

const btn = document.querySelector("button");

const post = document.querySelector(".post");

const widget = document.querySelector(".star-widget");

const editBtn = document.querySelector(".edit");

btn.onclick = ()=>{

widget.style.display = "none";

post.style.display = "block";

editBtn.onclick = ()=>{

widget.style.display = "block";

post.style.display = "none";

}

return false;

}

</script>

</body>

</html>

**Reject.html**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Loan approval status</title>

<link rel="stylesheet" href="static/reject.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<h1>LOAN APPROVAL STATUS</h1>

<h2>{{prediction\_text}}</h2>

<img src="static/reject.jpg" width="200px" height="200px">

<h3>Please provide your feedback</h3>

<div class="container">

<div class="post">

<div class="text">Thanks for rating us!</div>

<div class="edit">EDIT</div>

</div>

<div class="star-widget">

<input type="radio" name="rate" id="rate-5">

<label for="rate-5" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-4">

<label for="rate-4" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-3">

<label for="rate-3" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-2">

<label for="rate-2" class="fas fa-star"></label>

<input type="radio" name="rate" id="rate-1">

<label for="rate-1" class="fas fa-star"></label>

<form action="#">

<header></header>

<div class="textarea">

<textarea cols="30" placeholder="Describe your experience.."></textarea>

</div>

<div class="btn">

<button type="submit">Post</button>

</div>

</form>

</div>

</div>

<script>

const btn = document.querySelector("button");

const post = document.querySelector(".post");

const widget = document.querySelector(".star-widget");

const editBtn = document.querySelector(".edit");

btn.onclick = ()=>{

widget.style.display = "none";

post.style.display = "block";

editBtn.onclick = ()=>{

widget.style.display = "block";

post.style.display = "none";

}

return false;

}

</script>

</body>

</html>

**Prediction.html**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="static/prediction.css">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">

<link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">

<title>prediction</title>

</head>

<body>

<script>

function valid(){

var Ai=document.getElementById("ApplicantIncome").value;

var Co=document.getElementById("CoapplicantIncome").value;

var LA=document.getElementById("LoanAmount").value;

var LT=document.getElementById("Loan\_Amount\_Term").value;

if(Ai > 100000000000000000000000000000000000000){

alert("Applicant income is too large enter a valid number")

return false;

}

if(Co > 100000000000000000000000000000000000000){

alert("Coapplicant income is too large enter a valid number")

return false;

}

if(LA > 100000000000000000000000000000000000000){

alert("Loan Amount is too large enter a valid number")

return false;

}

if(LT > 100000000000000000000000000000000000000){

alert("loan amount term is too large enter a valid number")

return false;

}

var name=document.getElementById("Name").value;

var letters=/^[a-zA-Z]\*$/;

if(!name.match(letters)){

alert("Name must contain only alphabets")

return false;

}

var num=/^[0-9]+$/;

if(!Ai.match(num)){

alert("Enter only valid numbers alphabets are not allowed ")

return false;

}

if(!Co.match(num)){

alert("Enter only valid numbers alphabets are not allowed ")

return false;

}

if(!LA.match(num)){

alert("Enter only valid numbers alphabets are not allowed ")

return false;

}

if(!LT.match(num)){

alert("Enter only valid numbers alphabets are not allowed ")

return false;

}

var mo=document.getElementById("mon").value;

var mn=/^[0-9]{10}$/;

if(!mo.match(mn)){

alert("Please enter only 10 digit mobile number")

return false;

}

}

</script>

<section class="text-gray-600 body-font">

<div class="container px-5 py-24 mx-auto">

<div class="flex flex-col text-center w-full mb-20">

<h1 class="Heading">LOAN ELIGIBILITY PREDICTION</h1><br>

<p class="fill">Fill the form for prediction</p>

</div>

<div>

</div>

<div class="mb-3">

<a class="btn btn-primary" href="./" id="back" role="button">Back</a></div>

<form action='/prediction.html' method="post" onsubmit="return valid()">

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Name</label>

<input type="text" class="form-control" id="Name" name="Name" placeholder="Enter your Name" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Email ID</label>

<input type="email" class="form-control" id="email" name="email" placeholder="Enter your Email ID" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Mobile Number</label>

<input type="text" class="form-control" id="mon" name="mon" placeholder="Enter your Mobile number" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Gender</label>

<select class="form-select" id="gender" name="gender" aria-label="Default select example" required >

<option selected>-- select gender --</option>

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Married status</label>

<select class="form-select" id="married" name="married" aria-label="Default select example" required >

<option selected>-- select married status --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Dependents</label>

<select class="form-select" id="dependents" name="dependents" aria-label="Default select example" required>

<option selected>-- select dependents --</option>

<option value="0">0</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3+">3+</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Education</label>

<select class="form-select" id="education" name="education" aria-label="Default select example" required>

<option selected>-- select education --</option>

<option value="Graduate">Graduate</option>

<option value="Not Graduate">Not Graduate</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Self\_Employed</label>

<select class="form-select" id="employed" name="employed" aria-label="Default select example" required>

<option selected>-- select Self\_Employed --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Credit\_History</label>

<select class="form-select" id="credit" name="credit" aria-label="Default select example" required >

<option selected >-- select Credit\_History --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Property\_Area</label>

<select class="form-select" id="proparea" name="proparea" aria-label="Default select example" required>

<option selected>-- select Property\_Area --</option>

<option value="Semiurban">Semiurban</option>

<option value="Urban">Urban</option>

<option value="Rural">Rural</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter ApplicantIncome</label>

<input type="text" class="form-control" id="ApplicantIncome" name="ApplicantIncome" placeholder="ApplicantIncome" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter CoapplicantIncome</label>

<input type="text" class="form-control" id="CoapplicantIncome" name="CoapplicantIncome" placeholder="CoapplicantIncome" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Purpose of loan</label>

<select class="form-select" id="pur" name="pur" aria-label="Default select example" required>

<option selected>-- select the purpose of loan --</option>

<option value="person">Personal loan</option>

<option value="Bussiness">Bussiness loan</option>

<option value="Education">Education loan</option>

<option value="Home">Home loan</option>

<option value="Other">other</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter LoanAmount</label>

<input type="text" class="form-control" id="LoanAmount" name="LoanAmount" placeholder="LoanAmount" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter Loan\_Amount\_Term</label>

<input type="text" class="form-control" id="Loan\_Amount\_Term" name="Loan\_Amount\_Term" placeholder="Loan\_Amount\_Term" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter Adhar Number</label>

<input type="text" class="form-control" id="Adhar" name="Adhar" placeholder="Adhar Number" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter PAN card ID</label>

<input type="text" class="form-control" id="PAN " name="PAN " placeholder="PAN card ID" required>

</div>

<div class="mb-3">

<label for="property document" class="form-label">Property Document</label><br><input type="file" required >

</div>

<div class="mb-3">

<label for="Govt ID proof" class="form-label">Govet ID proof</label><br><input type="file" required>

</div>

<div class="mb-3">

<input type="checkbox" required>

I accept the <a href="terms.html">Terms and conditions</a>

</div>

<br><br>

<div class="mb-3">

<input type="submit" class="but" value="PREDICT">

</div>

</form>

</div>

</section>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf" crossorigin="anonymous"></script>

</body>

</html>

**Model.ipynb**

import numpy as np

import pandas as pd

import pickle

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

import sklearn

from sklearn.preprocessing import LabelEncoder

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from xgboost import XGBClassifier

from sklearn.ensemble import RandomForestClassifier

import imblearn

from imblearn.under\_sampling import RandomUnderSampler

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import scale

from sklearn.preprocessing import MaxAbsScaler

from sklearn.metrics import accuracy\_score,classification\_report,confusion\_matrix,f1\_score

## Reading The Dataset

df=pd.read\_csv('Loan\_dataset.csv')

df

df.head()

df.info()

df.shape

df=df.drop(columns=["Loan\_ID"],axis=1)

## Uni-Variate Analysis

sns.distplot(df.ApplicantIncome)

sns.countplot(df.Property\_Area)

sns.countplot(df.Gender)

sns.countplot(df.Education)

sns.countplot(df.Self\_Employed)

sns.countplot(df.Married)

sns.histplot(df.LoanAmount)

sns.displot(df.CoapplicantIncome)

plt.pie(df.Property\_Area.value\_counts(),[0,0,0],labels=['Semi urban','Urban','Rural'])

## Bivariate Analysis

sns.countplot(df['Married'],hue=df['Gender'])

sns.countplot(df['ApplicantIncome'],hue=df['CoapplicantIncome'])

sns.countplot(df['LoanAmount'],hue=df['Property\_Area'])

sns.countplot(df['Education'],hue=df['Self\_Employed'])

sns.barplot(df.ApplicantIncome,df.CoapplicantIncome)

sns.countplot(df['LoanAmount'],hue=df['Loan\_Amount\_Term'])

plt.scatter(df.ApplicantIncome,df.LoanAmount)

sns.countplot(df['Dependents'],hue=df['Gender'])

## Multi variate Analysis

sns.heatmap(df.corr(),annot=True)

plt.plot(df.LoanAmount,df.Loan\_Amount\_Term,df.ApplicantIncome)

df.plot.line()

df.hist()

plt.plot(df.LoanAmount,df.ApplicantIncome,df.CoapplicantIncome)

plt.plot(df.Loan\_Amount\_Term,df.ApplicantIncome,df.CoapplicantIncome)

## Descriptive Analysis

df.describe()

df.std()

df.mean()

df.mode()

## Data Pre-Processing

## Check for Null Values

df.isnull().any()

df.isnull().sum()

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.isnull().any()

df.isnull().sum()

## Handling Categorical Values

df.head()

le=LabelEncoder()

df.Gender=le.fit\_transform(df.Gender)

df.Married=le.fit\_transform(df.Married)

df.Education=le.fit\_transform(df.Education)

df.Self\_Employed=le.fit\_transform(df.Self\_Employed)

df.Property\_Area=le.fit\_transform(df.Property\_Area)

df.Loan\_Status=le.fit\_transform(df.Loan\_Status)

df.Dependents=le.fit\_transform(df.Dependents)

df.head()

## Spliting into dependent and independent data

df.head()

x=df.iloc[:,:-1]

y=df.Loan\_Status

x.head()

y.head()

## Scaling The Data

scaler = MaxAbsScaler()

x\_sc=scaler.fit\_transform(x)

x\_sc

## Balancing The Dataset

sns.countplot(df.Loan\_Status)

rus=RandomUnderSampler(sampling\_strategy=1)

x\_res,y\_res=rus.fit\_resample(x,y)

ax=y\_res.value\_counts().plot.pie(autopct='%.2f')

\_=ax.set\_title("under-sampling")

## Splitting Data Into Train and Test

xtrain,xtest,ytrain,ytest=train\_test\_split(x,y,test\_size=0.3,random\_state=10)

xtrain.head()

xtest.head()

ytrain.head()

ytest.head()

xtrain.shape

xtest.shape

ytrain.shape

ytest.shape

## Model Building

## Decision Tree Model

dmodel=DecisionTreeClassifier(random\_state=100)

dmodel.fit(x\_res,y\_res)

ypredd=dmodel.predict(xtest)

ypred2d=dmodel.predict(xtrain)

## Random Forest Model

Rmodel=RandomForestClassifier(n\_estimators=100,max\_depth=12,max\_features=3)

Rmodel.fit(x\_res,y\_res)

ypredR=Rmodel.predict(xtest)

ypred2R=Rmodel.predict(xtrain)

## KNN Model

kmodel=KNeighborsClassifier()

kmodel.fit(x\_res,y\_res)

ypredk=kmodel.predict(xtest)

ypred2k=kmodel.predict(xtrain)

## Xgboost Model

xmodel=XGBClassifier(eval\_metric='mlogloss',n\_estimators=100,random\_state=100)

xmodel.fit(x\_res,y\_res)

ypredx=xmodel.predict(xtest)

ypred2x=xmodel.predict(xtrain)

## Compare The Model

print("Decision Tree Model Testing Accuracy")

print(accuracy\_score(ytest,ypredd))

print("Decision Tree Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2d))

print("Random Forest Model Testing Accuracy")

print(accuracy\_score(ytest,ypredR))

print("Random Forest Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2R))

print("KNN Model Testing Accuracy")

print(accuracy\_score(ytest,ypredk))

print("KNN Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2k))

print("Xgboost Model Testing Accuracy")

print(accuracy\_score(ytest,ypredx))

print("Xgboost Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2x))

## Evaluating Performance Of The Model And Saving The Model

## Random Forest Model is Selected

print("Random Forest Model Testing Accuracy")

print(accuracy\_score(ytest,ypredR))

print("Random Forest Model Training Accuracy")

print(accuracy\_score(ytrain,ypred2R))

y=Rmodel.predict([[1,1,1,1,0,4583.0,1508.0,128.0,360.0,1.0,0]])

print(y)

y1=Rmodel.predict([[1,0,0,0,0,5849,0.0,146.412162,360.0,1.0,2]])

print(y1)

y2=Rmodel.predict([[1,0,0,1,0,678,987,90,24,1.0,2]])

print(y2)

f1\_score(ypredR,ytest,average='weighted')

pd.crosstab(ytest,ypredR)

print(classification\_report(ypredR,ytest))

**13.2 GITHUB LINK**

https://github.com/IBM-EPBL/IBM-Project-5389-1658761514

**13.3 DEMO VIDEO LINK**

https://drive.google.com/file/d/149UdoCUNItAryoKIMRmd5aDSFBTItdyg/view?usp=share\_link