

SMART LENDER - **APPLICANT CREDIBILITY** **PREDICTION FOR LOAN** **APPROVAL**

TEAM ID:

PNT2022TMID01198

TEAM MEMBERS:

Vijay.S

Vishal.R

Sanjay.B

Raja Tharun Devar

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code and GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview:

The primary source of revenue for the banking industry and source of financial risk for banks is a loan. The interest collected on loans disbursed directly accounts for significant amounts of a bank's assets. The ability of the borrower to repay the loan within the allotted period is one of the major hazards associated with the lending of loans. It's known as "credit risk." A candidate's credit score served as the basis for determining whether or not to approve a loan on them. Thus, the purpose of this study is to describe the use of various machine learning approaches that effectively identify who to lend money to and assist banks in identifying loan defaulters for significantly lower credit risk.

The fact that our banking system offers a wide range of goods, a bank's credit line is its primary source of income. As a result, they are able to profit from interest on the loans they credit. Lenders always seek to lower their credit risk since commercial loans have historically represented a significant portion of the banking sector. The function that banks play in the modern market economy is substantial. Loans, or whether clients repay or don't return them, significantly impact a bank's profitability. Before granting loans to borrowers, banks must determine if they are good (non-defaulters) or bad (defaulters). The credit worthiness of the borrowers is one of the most critical issues in commercial loan financing.

The probability that borrowers may default on their loan commitments is referred to as credit risk. For any bank or institution, determining whether a borrower will be good or bad is a very difficult process. The banking system employs a manual

procedure to determine whether or not a borrower has defaulted. The manual method will undoubtedly be more precise and efficient, but it will not be able to handle a high volume of loan applications at once. When a situation like this arises, it will take a very long time to make decisions and a lot of labour will be needed.

1.2 Purpose:

Loans account for a large portion of bank profits. For financial companies, the loan approval process is crucial. Because loan defaults are occurring more frequently and it is becoming more challenging for banking authorities to properly access loan requests and address the dangers of people defaulting on loans, it is very difficult to forecast if clients will be able to pay back the loan. Numerous scholars have been focusing on loan approval system prediction in recent years. For vast amounts of data, the machine learning technique is highly helpful in predicting outcomes. Four algorithms, including Random Forest, Decision Tree, Naive Bayes, and Logistic Regression, are employed in this study to forecast whether or not clients would be approved for loans. The same information will be analysed for all four methods, and the most accurate algorithm will be chosen to deploy the model. From this point forward, we create a machine learning-based bank loan prediction system that chooses the qualified applicants for loan approval on its own.

2. LITERATURE SURVEY

2.1 Existing problem:

Dream Housing Finance company deals in all kinds of home loans. They have a presence across all urban, semi-urban and rural areas. The customer first applies for a home loan and after that, the company validates the customer eligibility for the loan.

The company wants to automate the loan eligibility process (real-time) based on customer detail provided while filling out online application forms. These details are Gender, Marital Status, Education, number of Dependents, Income, Loan Amount, Credit History, and others.

To automate this process, they have provided a dataset to identify the customer segments that are eligible for loan amounts so that they can specifically target these customers.

2.2 References:

Paper 1: An Approach For Prediction Of Loan Approval

Publication year: May-June 2021

Author name : Ms. Kathe Rutika Pramod

Journal name: International Journal of Creative Research Thoughts (IJCRT)

Summary: In our banking system banks have many products to sell but main source of income of any banks is on its credit line they can earn from interest of those loans which they credits a bank's profit or a loss depends to a large extent on loans the customers are paying back the loan defaulting by predicting the loan defaulters the bank can reduce its non performing assets. the maximization of profits, it is essential to study the nature of the different methods and their comparison.

Methodology used: The prediction model which is constructed using three different training algorithms to train a supervised two layer feedforward network. The results show that the training algorithm improves the design of loan default prediction model.

Paper 2: Loan Approval Predictions**Publication year:** 04 | Apr 2022**Author name:** M. A. Sheikh, A. K. Goel and T. Kumar**Journal name:** International Research Journal of Engineering and Technology (IRJET)

Summary: 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. 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.

2.3 Problem Statement Definition:

Dream Housing Finance company deals in all home loans. They have a presence across all urban, semi-urban and rural areas. Customers first apply for a home loan after that company validates the customer's eligibility for a loan. The company wants to automate the loan eligibility process (real-time) based on customer detail provided while filling out the online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and others. To automate this process, they have given a problem to identify the customer segments, that are eligible for loan amounts so that they can specifically target these customers.

One of the most important factors which affect our country's economy and financial condition is the credit system governed by the banks. The process of bank credit risk evaluation is recognized at banks across the globe. "As we know credit risk evaluation is very crucial, there is a variety of techniques are used for risk level calculation. In addition, credit risk is one of the main functions of the banking community.

The prediction of credit defaulters is one of the difficult tasks for any bank. But by forecasting the loan defaulters, the banks definitely may reduce their loss by reducing their non-profit assets, so that recovery of approved loans can take place without any loss and it can play as the contributing parameter of the bank statement. This makes the study of this loan approval prediction important. Machine Learning techniques are very crucial and useful in the prediction of these types of data.

It is a classification problem where we have to predict whether a loan would be approved or not. In these kinds of problems, we have to predict discrete values based on a given set of independent variables. Classification can be of two types:

- **Binary Classification:-** In this, we have to predict either of the two given classes. For example: classifying the “gender” as male or female, predicting the “result” as to win or loss, etc.

- **MultiClass Classification:-** Here we have to classify the data into three or more classes. For example: classifying a “movie’s genre” as comedy, action, or romantic, classifying “fruits” like oranges, apples, pears, etc.

Loan prediction is a very common real-life problem that each retail bank faces at least once in its lifetime. If done correctly, it can save a lot of man-hours at the end of a retail bank. Although this course is specifically built to give you a walkthrough of the Loan Prediction problem, you can always refer to the content to get a comprehensive overview to solve a classification problem.

Who does the problem affect?	Applicants who apply loan in bank
What are the boundaries of the problem?	Applicants who apply loan in bank they faces eligibility related issues.
What is the issue?	Applicants don’t know what are eligibility criteria to a particular loan.
Where does the issue occur?	While applicants trying to apply for loan with required eligibility
Where does the issue occur?	This issue occurs among applicant who don’t have any idea “What are the eligibility criteria to apply loan”
Why is it important that we fix the problem?	So that applicants and also bank don’t waste their time in applying a particular loan which they are not eligible.
What solution to solve this issue?	A web application to predict whether he/she is eligible for the loan by checking

	the details of the applicants with eligibility criteria.
What methodology use to solve this issue?	Machine Learning algorithms are used to predict the eligibility with the help of past data's and say whether he/she is eligible or not.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map canvas is a more in-depth version of the original empathy map, which helps identify and describe the user's needs and pain points. And this is valuable information for improving the user experience.

Teams rely on user insights to map out what is important to their target audience, what influences them, and how they present themselves. This information is then used to create personas that help teams visualize users and empathize with them as individuals, rather than just as a vague marketing demographic or account number.

An empathy map canvas helps brands provide a better experience for users by helping teams understand the perspectives and mindset of their customers. Using a template to create an empathy map canvas reduces the preparation time and standardizes the process so you create empathy map canvases of similar quality.

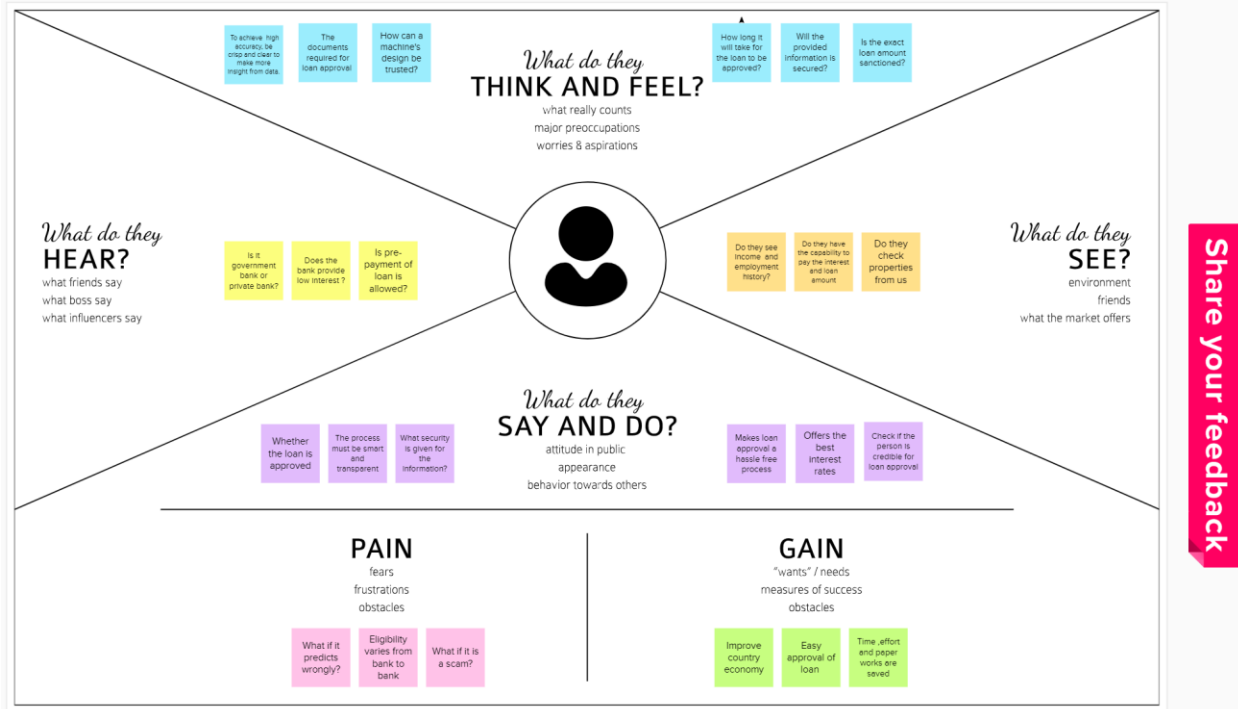
Empathy maps are divided into segments, which are typically defined by questions that teams work to answer one by one to complete the map. Using MURAL's template allows you to add color-coded sticky notes to help categorize answers visually.

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.



3.2 Ideation & Brainstorming:

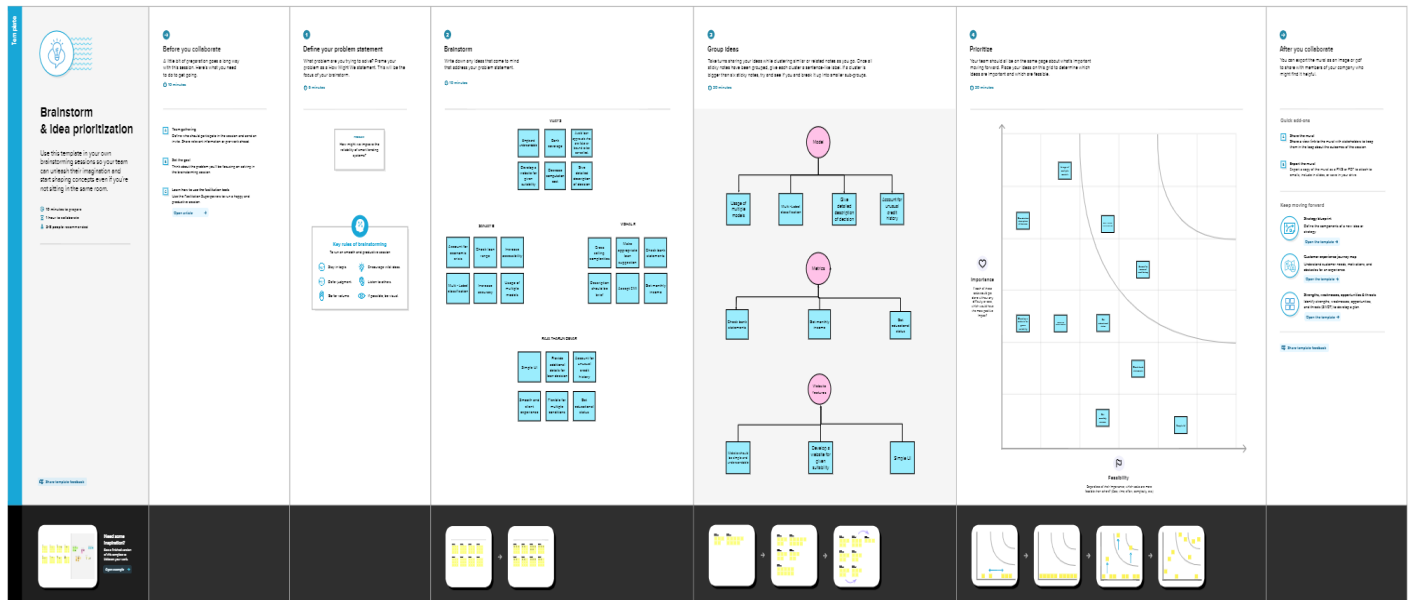
Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

Brainstorming is usually conducted by getting a group of people together to come up with either general new ideas or ideas for solving a specific problem or dealing with a specific situation.

Participants in a brainstorming session are encouraged to freely toss out whatever ideas may occur to them. The thinking is that by generating a large

number of ideas, the brainstorming group is likely to come up with a suitable solution for whatever issue they are addressing.

The lines between ideation and brainstorming have become a bit more blurred with the development of several brainstorming software programs, such as Brightidea and Ideawake. These software programs are designed to encourage employees of companies to generate new ideas for improving the companies' operations and, ultimately, bottom-line profitability.



3.3 Proposed Solution:

Your proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result.

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement	A bank is a financial institution licensed to receive deposits and make loans needs a way to verify and trust the customer details and their

		documents for getting loan because they need an trustable customer with proper assets ,cash flow, documents and background who can repay the loan amount and interest on time.
2.	Idea/Solution description	<p>- The customer only need to enter the details,the loan approval status is then predicted automatically and quickly.</p> <p>-The property documents of the customer need to be submitted and the customer should agree to the terms and conditions of the bank.</p> <p>- The loan approval will also depend on the CIBIL score of customer.Provide captcha security .</p> <p>- Automatic calculation of interest rate and repayment date based on loan amount.</p> <p>-Varies efficient machine learning algorithms can be used to predict the loan eligibility of the customer.</p>
3.	Novelty/Uniqueness	<p>-Provide customer ratings and reviews for understanding the customer.</p> <p>-Adding digital signature of the customer on</p>

		<p>agreement of the terms and conditions.</p> <p>-Provides data security. The customer details will not be shared to the third party.</p> <p>- Instant Loan approval status.</p>
4.	Social Impact	<p>- Easy and fast loan approval process for the customer.</p> <p>- Approves Loan to a trustable person.</p> <p>- Bank can find a genuine person to provide loan.</p> <p>-Secure storage of customer details.</p>
5.	Business Model	<p>-Can generate revenue through advertisement.</p> <p>-Can collaborate with many companies.</p> <p>- Can charge the processing fees and service fees from customer</p> <p>-Can give pre payment option.</p> <p>- Can generate revenue by referencing.</p>
6.	Scalability Of Solution	<p>-It can be provided as software as a service.</p> <p>- Both borrower and Lender can use this software.</p> <p>- Any type of customer can predict their loan approval without any discrimination.</p>

		<p>- Can use this software anytimeand anywhere.</p> <p>- This system is easily scalableand efficient.</p> <p>- Easy and user friendly software for all.</p>
--	--	---

3.4 Problem Solution fit:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.

Project Title: Smart Lender

Project Design Phase-I - Solution Fit

Team ID: PNT2022TMID01198

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Our customer segment mainly consists of Bank employees who deal with evaluating the credibility of the account holders for loan approval.	6. CUSTOMER CONSTRAINTS CC Possible constraints imposed may be : -> Getting a clear understanding about using the application. -> Access to details of the account holder for feeding in parameters. -> System requirements such as fast internet, hardware specifications etc.	5. AVAILABLE SOLUTIONS AS 1. Manual examination of eligibility of borrower by exploring various documentations of the borrower. 2. Creation of functions with various conditional statements using backend (database) languages. 3. Creation of models using Machine Learning.	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Examining the eligibility of the customer of a bank who requests a loan before fulfilling their demands.	9. PROBLEM ROOT CAUSE RC 1. Manual evaluation is a monotonous and heavy time-consuming process. 2. Manual evaluation could be unreliable because of accuracy issues.	7. BEHAVIOUR BE In order to arrive at a conclusion, bank employees take their time carefully analysing various details of individuals and to verify that the individual is who they say they are, they may also interview them in numerous ways.	
Identify strong TR & EM	3. TRIGGERS TR Consumption of unusual amount of time, producing results of low accuracy could be some of the factors that push the customers to address the problem.	10. YOUR SOLUTION SL Quantitative analysis of credibility of a bank customer for loan approval using various Machine Learning classification algorithms such as Linear Regression, Decision Tree etc. This Machine Learning model can be integrated with an appropriate user interface in order to deploy it as an application.	8. CHANNELS of BEHAVIOUR CH 8.1 ONLINE Customers may access appropriate databases in order to fetch details of the borrower, or to verify the authenticity of the details given. 8.2 OFFLINE Customers may feed in the details and get a credibility score as output from the ML model.	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER EM Before - Customers may feel confused about the eligibility of the borrower with results of low standards from Manual evaluation After - Customers may now be more confident in making decisions of loan approval as they have arrived where they are through proper statistical analysis.			

4. REQUIREMENT ANALYSIS

4.1 Functional requirement:

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behavior under specific conditions.

Following are the functional requirements of the proposed solution

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Home Page	<ul style="list-style-type: none">• Smart Lender Applicant Credibility description• Information about Credibility details required for loan approval• if new user , REGISTER• if already exists, SIGN IN
FR-2	User Registration	Enter Mail Id and other personal details required for Registering
FR-3	User login	User Mail Id and Password for Login
FR-4	Loan Approval form	Credibility details should be entered for prediction
FR-5	Result	if Approved - It display the information about what is done to be next. if Not Approved - It display the information about what rejection criteria you are not eligible for the loan.

4.2 Non-Functional requirements:

FR No	Non-Functional Requirement	Description
NFR-1	Usability	If the customer is eligible, he/she should be able to receive the acknowledgement receipt for loan application within 7 days from the bank. If not, then he/she should be intimated with the reason for rejection.

NFR-2	Security	Check if the customer has any fraudulent history
NFR-3	Reliability	Customer's financial status plays a crucial role. So the customer should have a good credit score.
NFR-4	Performance	By training the model using different ML algorithms, the performance of the system can be increased.
NFR-5	Availability	The loan will be available easily to those persons who have high income and to those who assure to repay the high sum within short period of time.
NFR-6	Scalability	The customer should be between 21 to 60 age. And based on customer's capacity to handle their expenses

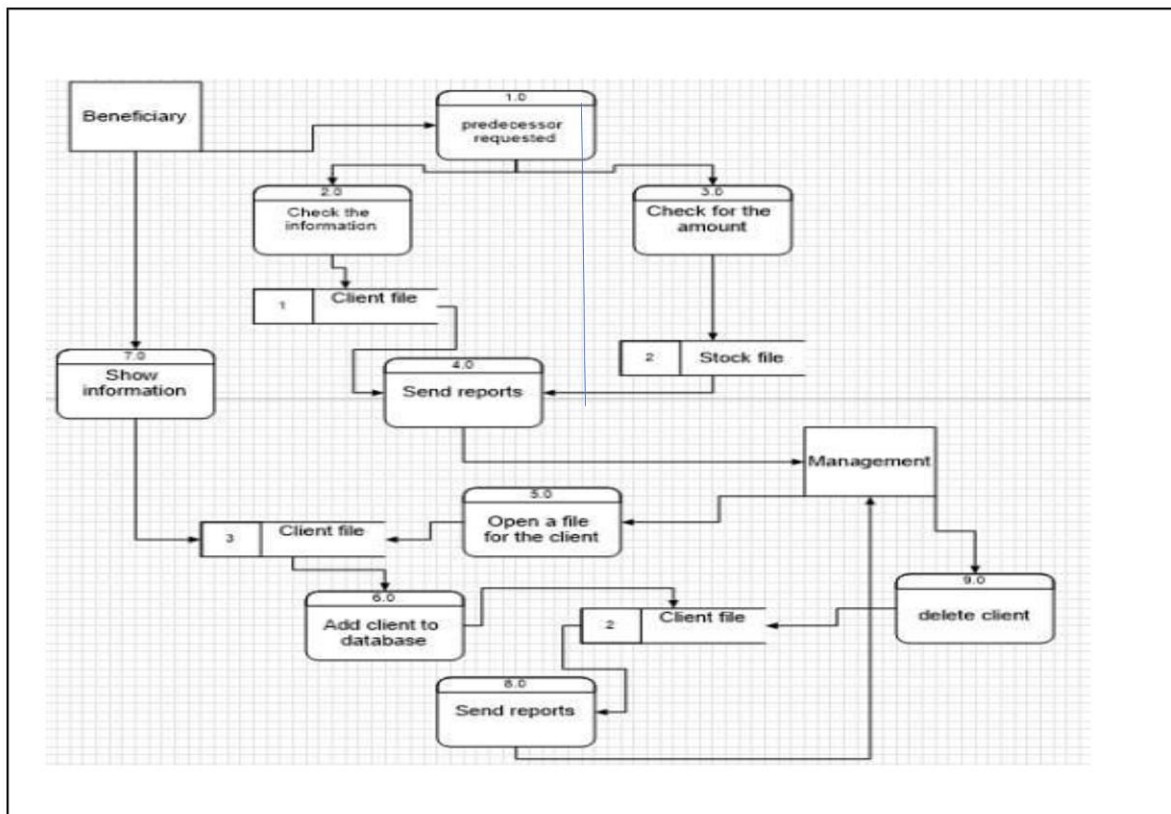
5. PROJECT DESIGN

5.1 Data Flow Diagrams:

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

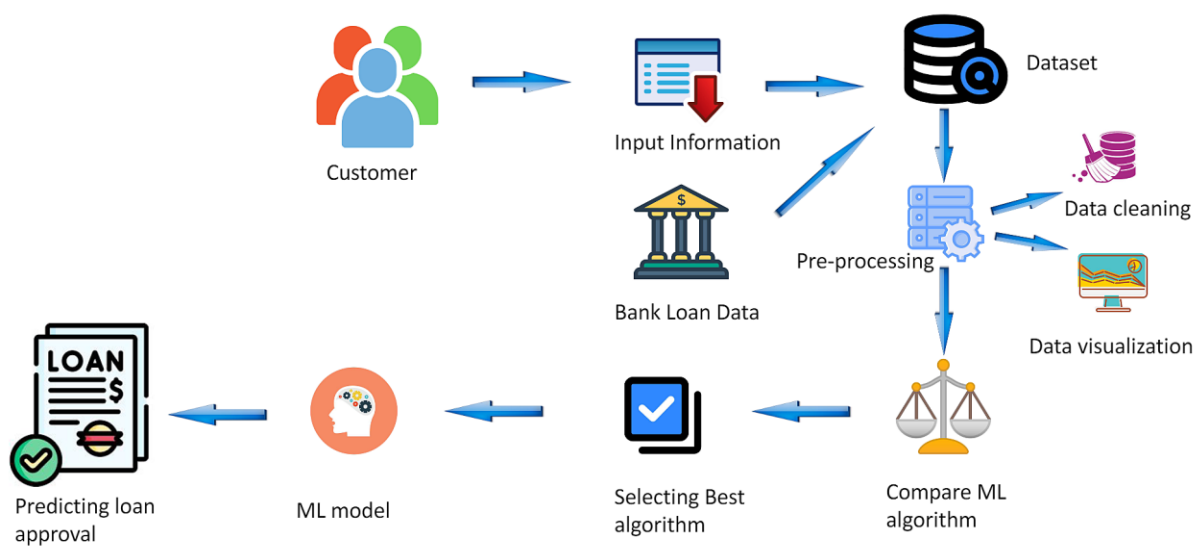
Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO.

That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.



5.2 Solution & Technical Architecture:

Solution Architecture



**Project Design Phase-II
Technology Stack (Architecture & Stack)**

Date	03 October 2022
Team ID	PNT2022TMID01198
Project Name	Project - Smart Lender - Applicant Credibility Prediction for Loan Approval
Maximum Marks	4 Marks

Technical Architecture:

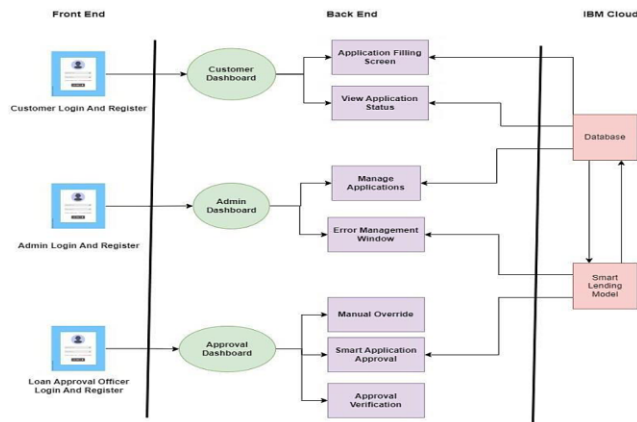


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Customer, Admin and Loan Approval Officers will be able to register login and use the functionalities available to them	HTML, CSS, Javascript
2	Application Filing	Available to customers only, this screen allows customers to fill details in an online application for loan approval.	Javascript
3.	View Application Status	View pertinent information relevant to the	Javascript

		applications. Can be accessed by admin only.	
4.	Manage Applications	Ability to modify applications as and when needed. Can be accessed by admin only	Javascript
5.	Error Management Window	Ability to rectify or troubleshoot errors. Can be accessed by admin only	Javascript
6.	Manual Override	Ability to change loan approval decision manually. Available only to loan approval officer	Javascript
7.	Smart Application Approval	Performs automated loan approval decision and gives details supporting decision	Javascript, Python
8.	Application Verification	Verification of application and documents provided by applicant	Javascript
9.	Cloud Database	Data Types: String, Number, Float, Int, Bool Configuration abort-slave-event-count 0 allow-suspicious-udfs FALSE archive ON auto-increment-increment 1 auto-increment-offset 1 autocommit TRUE automatic-sp-privileges TRUE avoid-temporal-upgrade FALSE back-log 80 based /home/jon/bin/mysql-8 0/. ... tmpdir transaction-alloc-block-size transaction-isolation READ transaction-prealloc-size transaction-read-only transaction-write-set-extraction updatable-views-with-limit validate-user-plugins verbose wait-timeout /tmp 8192 REPEATABLE4096 FALSE XXHASH64 YES TRUE TRUE 28800	MySQL Online

10.	File Storage	Network File System (NFS)	IBM Cloud File Storage
11.	Machine Learning Model	Automated decision making for loan approval and giving details for decision.	CNN, DNN
12.	Infrastructure (Server / Cloud)	Default	Flask

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask is used to host the website. Scikit, numpy and tensorflow are all open source python machine learning frameworks.	Scikit, Numpy, Tensorflow, Flask
2.	Security Implementations	OpenSSL is a program and library that supports many different cryptographic operations, including: Symmetric key encryption. Public/private key pair generation. Public key encryption. Hash functions.	OpenSSL Encryption
3.	Scalable Architecture	Since the application servers can be deployed on many machines. Also, the database does not make longer connections with every client – it only requires connections from a smaller number of application servers. It improves data integrity	3 Tier Architecture
4.	Availability	Decentralized storage and distribution along-with web application approach make the service highly available.	IBM Cloud file storage, MySQL Online
5	Performance	Long term header expiration.	AJAX, CDN

5.3 User Stories:

A user story is a short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system.

User stories were historically written on index cards or sticky notes, stored in a shoe box, and arranged on walls or tables to facilitate planning and discussion. Nowadays, they might just as easily be stored in a Jira issue.

User stories are designed to strongly shift the focus from writing about features to discussing them. In fact, these discussions are more important than whatever text is written.

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	Able to login	High	Sprint-1

			email & password			
	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	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	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

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Planning and Estimation are essential in software projects to achieve predictability, reduce the risks involved, and set a basic expectation for all stakeholders. Planning brings a lot of focus on preparation and forecasting whereas Estimation is a process to forecast project-related variables i.e., effort, scope, schedule, etc.

Planning: Planning is required irrespective of the project management methodologies that the team follows, whether it is Waterfall or Agile. Planning gives the project team a perspective on how to meet the objective in a systematic way and helps project stakeholders to keep a tab on the project progress and investments done.

Estimation: Schedule, Scope, Cost, and Effort are the four major variables that typically control Software projects. Any changes in any of these variables can have an effect on a project. Estimation is a process to forecast these variables to develop or maintain software based on the information specified by the client.

Planning and Estimation in Agile projects bring a lot of focus on preparation and forecasting. Both these activities are done keeping business context in mind and measurable value delivery is committed to the client. Therefore, it is recommended to have required planning and estimation in Agile from the start of the project, in order to ensure better risk coverage and higher predictability.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dataset	USN-4	Downloading the dataset	1	High	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-1		USN-5	Visualizing the dataset	2	Low	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-1		USN-6	Pre-process the dataset	3	Medium	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-1	Machine Learning Model	USN-7	KNN model building	5	High	s.vijay r.vishal b.sanjay raja tharun devar

Sprint-2		USN-8	Decision Tree model building	5	High	s.vijay r.vishal b.sanjay raja tharun devar
----------	--	-------	------------------------------	---	------	--

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-9	Naive Bayes model building	5	High	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-2		USN-10	Fine Tuning the model	3	Low	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-2		USN-11	Evaluation and saving of the models	5	High	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-3	Customer User Interface	USN-12	Model Integration with flask	5	High	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-3		USN-1	As a user, I should be able to access the dashboard.	3	Medium	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-3		USN-2	Select the type of loan	3	Low	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-3		USN-3	Fill the application and check the eligibility of	5	High	s.vijay r.vishal b.sanjay raja tharun devar

			loan approval			
Sprint-4	Deployed the website	USN-13	Register on IBM Cloud	3	Low	s.vijay r.vishal b.sanjay raja tharun devar
Sprint-4		USN-14	Train the ML model on IBM Cloud	5	Medium	s.vijay r.vishal b.sanjay raja tharun devar

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4		USN-15	Deploy the website on IBM Cloud	8	High	s.vijay r.vishal b.sanjay raja tharun devar

6.2 Sprint Delivery Schedule:

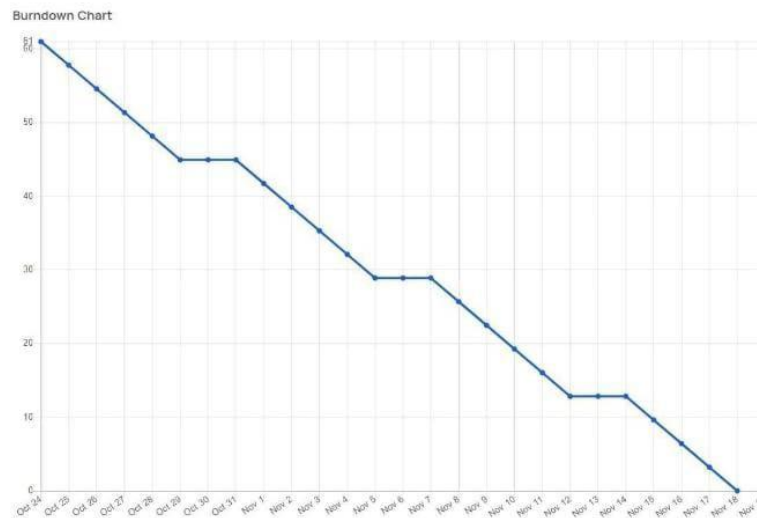
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	11	6 Days	24 Oct 2022	29 Oct 2022	11	29 Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	05 Nov 2022	18	05 Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	16	12 Nov 2022

Sprint-4	16	6 Days	14 Nov 2022	19 Nov 2022	16	19 Nov 2022
----------	----	--------	-------------	-------------	----	-------------

6.3. Reports from JIRA:

BurndownChart:

A burndown chart is a graphical representation of work left to do over time. It is often used in agile software development methodologies such as Scrum. However, burndown charts can be applied to any project containing measurable progress over time.



7. CODING & SOLUTIONING

7.1. Feature-1:

- i. Smart Lender - Applicant Credibility Prediction for Loan Approval
- ii. Languages : Python

- iii. Tools/IDE : Anaconda
- iv. Libraries :Recommendations

```
from flask import render_template,Flask,request
import numpy as np
import pickle
from sklearn.preprocessing import scale
app= Flask(__name__, template_folder='templates')
```

```
model = pickle.load(open("Rfmodel.pkl","rb"))
```

```
@app.route('/')
def home():
    return render_template('home.html')
@app.route('/login.html')
def login():
    return render_template('login.html')
@app.route('/procedure.html')
def procedure():
    return render_template('procedure.html')
@app.route('/bank login.html')
def bank():
    return render_template('bank login.html')
@app.route('/About.html')
def about():
    return render_template('About.html')
@app.route('/terms.html')
def terms():
    return render_template('terms.html')
@app.route('/register.html')
def register():
    return render_template('register.html')
@app.route('/contact.html')
def contact():
    return render_template('contact.html')
@app.route('/home.html')
def home1():
    return render_template('home.html')
@app.route('/prediction.html')
def formpg():
```

```

    return render_template('prediction.html')
@app.route('/rating.html')
def rat():
    return render_template('rating.html')
@app.route('/prediction.html', methods = ['POST'])
def predict():
    if request.method == 'POST':
        name=request.form['Name']
        gender=request.form['gender']
        married=request.form['married']
        dependents=request.form['dependents']
        education=request.form['education']
        employed=request.form['employed']
        credit=request.form['credit']
        proparea=request.form['proparea']
        ApplicantIncome=float(request.form['ApplicantIncome'])
        CoapplicantIncome=float(request.form['CoapplicantIncome'])
        LoanAmount=float(request.form['LoanAmount'])
        Loan_Amount_Term=float(request.form['Loan_Amount_Term'])
    if gender == 'Male':
        gender = 1
    else:
        gender = 0

    if married == 'Yes':
        married = 1
    else:
        married = 0

    if education == 'Graduate':
        education = 0
    else:
        education = 1

    if employed == 'Yes':
        employed = 1
    else:
        employed = 0

    if dependents == '3+':
        dependents = 3
    if credit == 'Yes':
        credit = 1
    else:
        credit = 0

```

```

if proparea == 'Urban':
    proparea = 2
elif proparea == 'Rural':
    proparea = 0
else:
    proparea = 1

features =
[gender,married,dependents,education,employed,ApplicantIncome,CoapplicantIncome,LoanAmount,Loan_Amount_Term
,credit,proparea]

con_features = [np.array(features)]

prediction = model.predict(con_features)
print(prediction)
if prediction==1:
    return render_template('approve.html',prediction_text='Congratulations! '+name+' You are eligible for loan')
else:
    return render_template('reject.html',prediction_text='Sorry '+name+' You are not eligible for loan')

if __name__ == "__main__":
    app.run(debug=False)

```

7.2. Feature-2:

Index

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Loan Prediction</title>
    <link rel="stylesheet" href="static/home.css">
</head>
<body>
    <div class="container">
        <div class="navbar">

```

```

        <nav>
            <ul>
                <li><a href="home.html">Home</a></li>
                <li><a href="About.html">About</a></li>
            <li><a href="procedure.html">Procedure</a></li>
                <li><a href="contact.html">Contact Us</a></li>
                <li><a href="login.html">User login</a></li>
            <li><a href="bank login.html">Bank login</a></li>
            </ul>
        </nav>

    </div>
    <div class="content">

        <h1>Smart Lender - Applicant Credibility Prediction For Loan Approval </h1>
        <p> Predit your loan eligibility here</p><br><br>
        <a href="prediction.html" class="btn">PREDICT</a>
        <br><br>
        <h2>Team ID -PNT2022TMID01198</h2><br>
        <h3>Team members</h3>
        <p>Vijay S </p>
        <p>Sanjay B</p>
        <p>Vishal R</p>
        <p>Raja tharun devar</p>

    </div>

</div>
</body>
</html>

```

Chance:

```

<!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>

<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>

```

NO CHANCE :

```

<!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>
    
    <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>

```

DATABASE SCHEMA

Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	LoanAmount_Term	Credit_History	Property_Area	Loan_Status
LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N

LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Urban	Y
LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Urban	Y
LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	Semiurban	N
LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Urban	Y
	LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	Semiurban
	LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Urban
	LP001027	Male	Yes	2	Graduate		2500	1840	109	360	1	Urban
	LP001028	Male	Yes	2	Graduate	No	3073	8106	200	360	1	Urban
	LP001029	Male	No	0	Graduate	No	1853	2840	114	360	1	Rural

	LP0010 30	Male	Yes	2	Grad uate	No	1299	1086	17	120	1	Urban
	LP0010 32	Male	No	0	Grad uate	No	4950	0	125	360	1	Urban
	LP0010 34	Male	No	1	Not Grad uate	No	3596	0	100	240		Urban
	LP0010 36	Fema le	No	0	Grad uate	No	3510	0	76	360	0	Urban
	LP0010 38	Male	Yes	0	Not Grad uate	No	4887	0	133	360	1	Rural

8.TESTING

1.Purpose of Document:

The purpose of this document is to briefly explain the test coverage and open issues of Applicant Credibility Prediction for Loan Approval project at the time of the release to user acceptance testing (UAT).

2. Defect Analysis:

This report shoes the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	0	2	1	3
Duplicate	1	0	0	0	1
External	0	0	1	0	1
Fixed	0	1	1	0	2
Not Reproduced	0	1	0	0	1
Skipped	0	0	0	0	0
Won't Fix	0	1	0	0	1
Totals	1	3	4	1	9

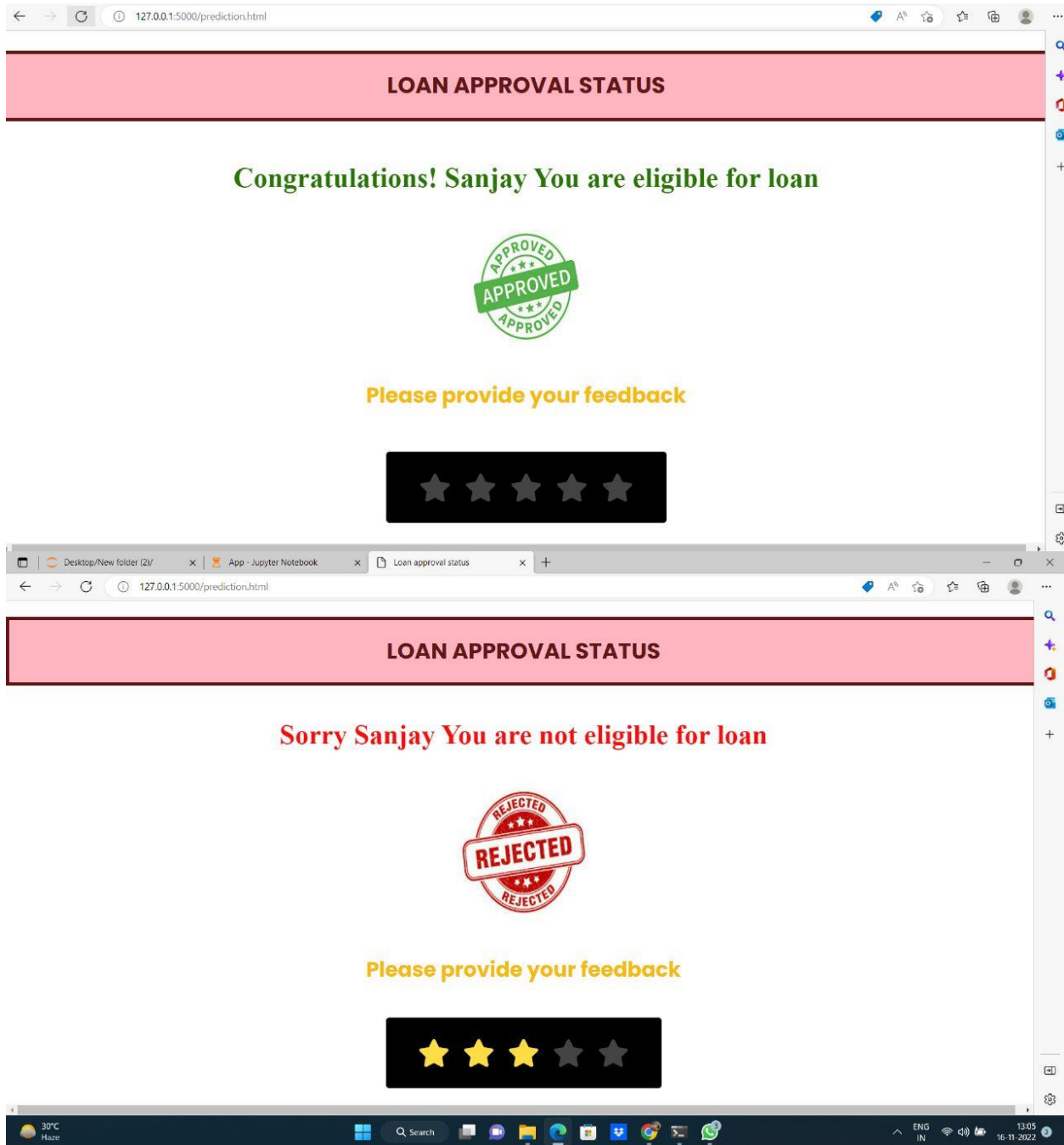
3. Test Case Analysis:

This report shows the number of test cases that have passed and untested.

Section	Total Cases	Not Tested	Fail	Pass
View Home Page	10	0	3	7
Enter the scores	20	0	3	17
Click Submit button	2	0	0	2
Image displayed	10	0	4	6
Selecting from Drop down	5	0	0	5
Final Report Output	30	0	10	20
Version Control	5	0	2	3

9.RESULTS

9.1 Performance metrics:



10. ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

- It is user-friendly.
- It is fast efficient and reliable.
- It tells quickly whether he is eligible for loan or not.

- Error percentage is minimum.
- Avoid data redundancy and inconsistency

DISADVANTAGES:

- It requires internet connection
- System will provide inaccurate answers if the data entered is incorrect
- Other factors such as change in policy should be updated periodically, otherwise it may provide inaccurate answers.

11. CONCLUSION:

We have selected the machine learning technique to analyse the bank dataset in order to forecast the loan approval status of the applied client. We tested a number of machine learning algorithms to see which one would work the best on the dataset to get the most accurate results. Using this strategy, we discovered that, aside from logistic regression, the other algorithms delivered accurate results in a suitable manner.

The remaining algorithms' accuracy ranged from 75% to 85%. The logistic regression, however, provided us with the highest level of accuracy (88.70%) after comparison of all the methods.

We also identified the key elements that have the greatest impact on the loan approval status. The performance accuracy of these most crucial characteristics applied to a few chosen algorithms is then compared to the case where all features were employed. The banks may use this model to determine what elements are crucial for the loan approval process. Based on their accuracy, the comparison research clearly identifies which algorithm will be the best and disregards the others.

12. FUTURE SCOPE:

The future scope of this project is very broad, few of them are:

- This can be implemented in less time.
- This can be accessed anywhere at anytime.
- The scope of this project is a web application that allows user to enter the data and get prediction of the chances of getting their loan.
- The database can be implemented for this system in such a way that the details of the user saved and referred for future use.

13. APPENDIX:

13.1.Source Code:

```
"source": [  
    "from flask import render_template,Flask,request\n",  
    "import numpy as np\n",  
    "import pickle \n",  
    "from sklearn.preprocessing import scale\n",  
    "app= Flask(__name__, template_folder='templates')\n",  
    "\n",  
    "model = pickle.load(open(\"Rfmodel.pkl\",\"rb\"))\n",  
    "\n",  
    "\n",  
    "\n",  
    "@app.route('/')\n",  
    "def home():\n",  
    "    return render_template('home.html')\n",  
    "@app.route('/login.html')\n",  
    "def login():\n",  
    "    return render_template('login.html')\n",  
    "@app.route('/procedure.html')\n",  
    "def procedure():\n",  
    "    return render_template('procedure.html')\n",  
    "@app.route('/bank login.html')\n",  
    "def bank():\n",  
    "    return render_template('bank login.html')\n",  
    "@app.route('/About.html')\n",  
    "def about():\n",  
    "    return render_template('About.html')\n",  
    "@app.route('/terms.html')\n",  
    "def terms():\n",  
    "    return render_template('terms.html')\n",  
    "@app.route('/register.html')\n",  
    "def register():\n",  
    "    return render_template('register.html')\n",  
    "@app.route('/contact.html')\n",  
    "def contact():\n",  
    "    return render_template('contact.html')\n",  
    "@app.route('/home.html')\n",  
    "def home1():\n",  
    "    return render_template('home.html')\n",  
    "@app.route('/prediction.html')\n",  
    "def formpg():\n",
```



```

"    return render_template('prediction.html')\n",
"@app.route('/rating.html')\n",
"def rat():\n",
"    return render_template('rating.html')\n",
"@app.route('/prediction.html', methods = ['POST'])\n",
"def predict():\n",
"    if request.method=='POST':\n",
"        name=request.form['Name']\n",
"        gender=request.form['gender']\n",
"        married=request.form['married']\n",
"        dependents=request.form['dependents']\n",
"        education=request.form['education']\n",
"        employed=request.form['employed']\n",
"        credit=request.form['credit']\n",
"        proparea=request.form['proparea']\n",
"        ApplicantIncome=float(request.form['ApplicantIncome'])\n",
"        CoapplicantIncome=float(request.form['CoapplicantIncome'])\n",
"        LoanAmount=float(request.form['LoanAmount'])\n",
"        Loan_Amount_Term=float(request.form['Loan_Amount_Term'])\n",
"        if gender == 'Male':\n",
"            gender = 1\n",
"        else:\n",
"            gender = 0\n",
"\n",
"        if married == 'Yes':\n",
"            married = 1\n",
"        else:\n",
"            married = 0\n",
"\n",
"        if education == 'Graduate':\n",
"            education = 0\n",
"        else:\n",
"            education = 1\n",
"\n",
"        if employed == 'Yes':\n",
"            employed = 1\n",
"        else:\n",
"            employed = 0\n",
"\n",
"        if dependents == '3+':\n",
"            dependents = 3\n",
"        if credit == 'Yes':\n",
"            credit = 1\n",
"        else:\n",
"            credit = 0\n",

```

```

" if proparea == 'Urban':\n",
"     proparea = 2\n",
" elif proparea == 'Rural':\n",
"     proparea = 0\n",
" else:\n",
"     proparea = 1     \n",
"     \n",
"\n",
"     \n",
"\n",
" features =
[gender,married,dependents,education,employed,ApplicantIncome,CoapplicantIncome,L
oanAmount,Loan_Amount_Term,credit,proparea]\n",
"     \n",
"     con_features = [np.array(features)]\n",
"     \n",
"\n",
"\n",
"     prediction = model.predict(con_features)\n",
"     print(prediction)\n",
"     if prediction==1:\n",
"         return render_template('approve.html',prediction_text ='Congratulations!
'+name+' You are eligible for loan')\n",
"     else:\n",
"         return render_template('reject.html',prediction_text ='Sorry '+name+' You are not
eligible for loan')\n",
"     \n",
"     \n",
"if __name__ == '__main__':\n",
"     app.run(debug=False)"
]
},
{
"cell_type": "code",
"execution_count": null,
"id": "18c6b1b5",
"metadata": {},
"outputs": [],
"source": []
},
{
"cell_type": "code",
"execution_count": null,
"id": "c249c156",
"metadata": {},

```

```

"outputs": [],
"source": []
},
],
"metadata": {
"kernel_spec": {
"display_name": "Python 3 (ipykernel)",
"language": "python",
"name": "python3"
},
"language_info": {
"codemirror_mode": {
"name": "ipython",
"version": 3
},
"file_extension": ".py",
"mimetype": "text/x-python",
"name": "python",
"nbconvert_exporter": "python",
"pygments_lexer": "ipython3",
"version": "3.9.12"
}
},
"nbformat": 4,
"nbformat_minor": 5
}

```

13.2. Github & Project Demo Link:

Github link : <https://github.com/IBM-EPBL/IBM-Project-2412-1658471312>

Project Demo link:
https://drive.google.com/file/d/1dwECvcvQFCX5UchMuBczB2EyYsa041WU/view?usp=share_link

