Smart Lender - Applicant Credibility Prediction for Loan Approval

1. INTRODUCTION

1.1 Project Overview

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.

1.2 Purpose

To predict the approval of loan and to avoid the credit risk, "Applicant Credibility Prediction for Loan Approval" is very crucial and useful. 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.

2. LITERATURE SURVEY

2.1 Existing Problem

credit risk is one of the main functions of the banking community. It makes difficulty for the bank loan provider to find the credit risk and it takes much time for approving the loan and it is difficult to do it by manually.

2.2 References

[1] Sudhamathy G and Jothi Venkateswaran “Analytics Using R for Predicting Credit Defaulters”, IEEE international conference on advances in computer applications (ICACA),978-1-5090-3770-4, 2016.

[2] M. Sudhakar, and C.V.K. Reddy, “Two Step Credit Risk Assessment Model for Retail Bank Loan Applications UsingDecision Tree Data Mining Technique”, IJARCET, vol. 5, no.3, pp. 705-718, 2016

[3] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, "Data Pre-processing for Supervised Leaning", International Journal of Computer Science, 2006, Vol 1 N. 2, pp 111–117.

[4] Vivek Bhambri “Application of Data Mining in Banking Sector”, International Journal of Computer Science and Technology Vol. 2, Issue 2, June 2011

[5] Dileep B. Desai, Dr. R.V.Kulkarni “A Review: Application of Data Mining Tools in CRM for Selected Banks”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (2) , 2013, 199 – 201.

[6] Dr. Madan Lal Bhasin, “Data Mining: A Competitive Tool in the Banking and Retail Industries”, The Chartered Accountant October 2006

[7] Frawley, W. J., Piatetsky-Shapiro, G., and Matheus, C. J. (1992). Knowledge discovery in databases: An overview. AI Magazine, 13(3):57.

2.3 Problem Statement Definition

Banks are making major part of profits through loans. Loan approval is a very important process for banking organizations. It is very difficult to predict the possibility of payment of loan by the customers because there is an increasing rate of loan defaults and the banking authorities are finding it more difficult to correctly access loan requests and tackle the risks of people defaulting on loans.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

Empathy map is used to know the insight of user. It is a visualized perspections of user's emotions or feelings about the problem. It makes easy to understand the thought of the user. Emapthy map is used to represent user thought through a chart. user's pain and gain can be noted in empathy map. In empathy map there is the answer for the following questions:

What do the user thing and feel?

What do the user see?

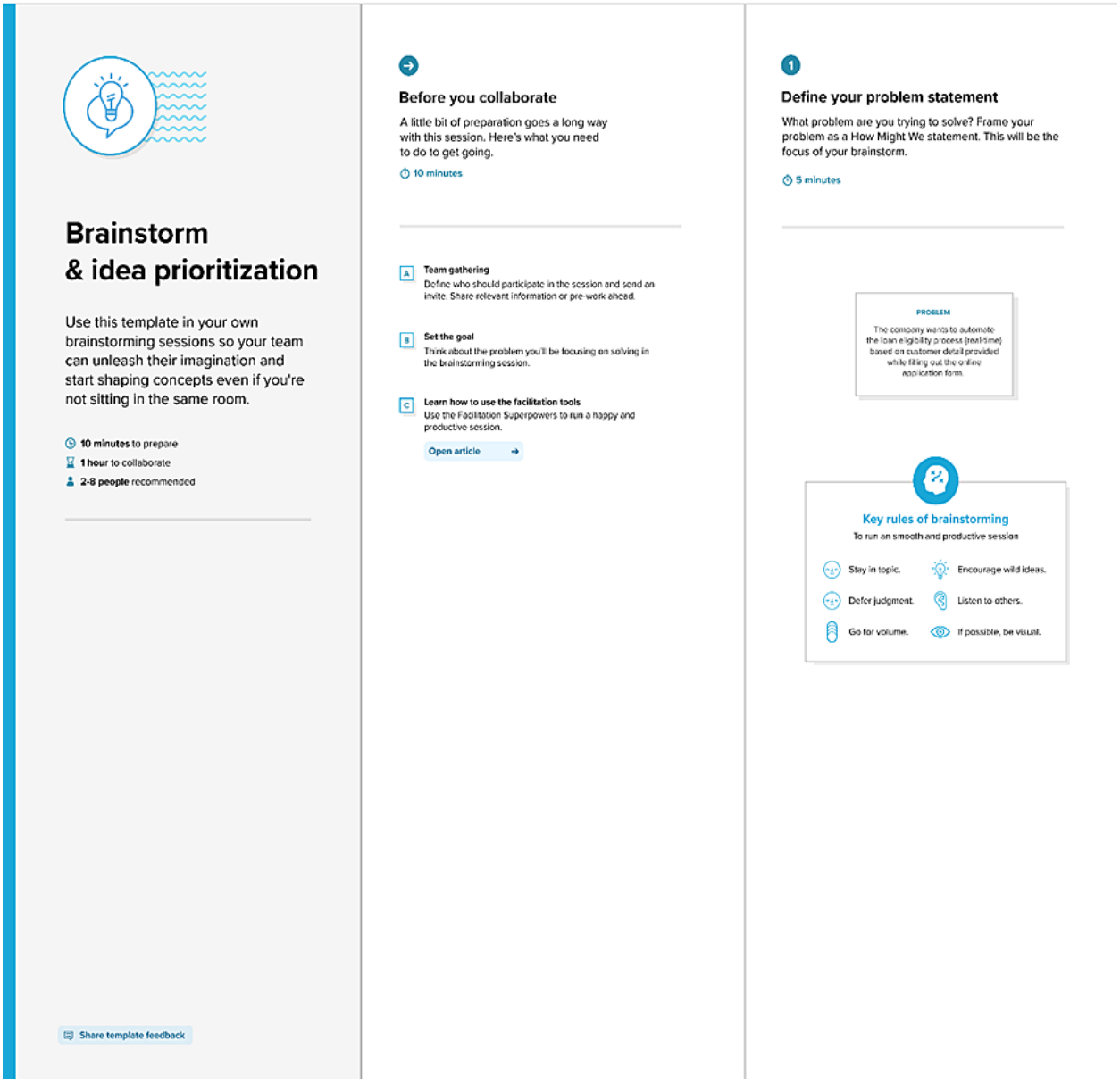
What do the user say and do?

What do the user hear?

3.2 Ideation & Brainstorming

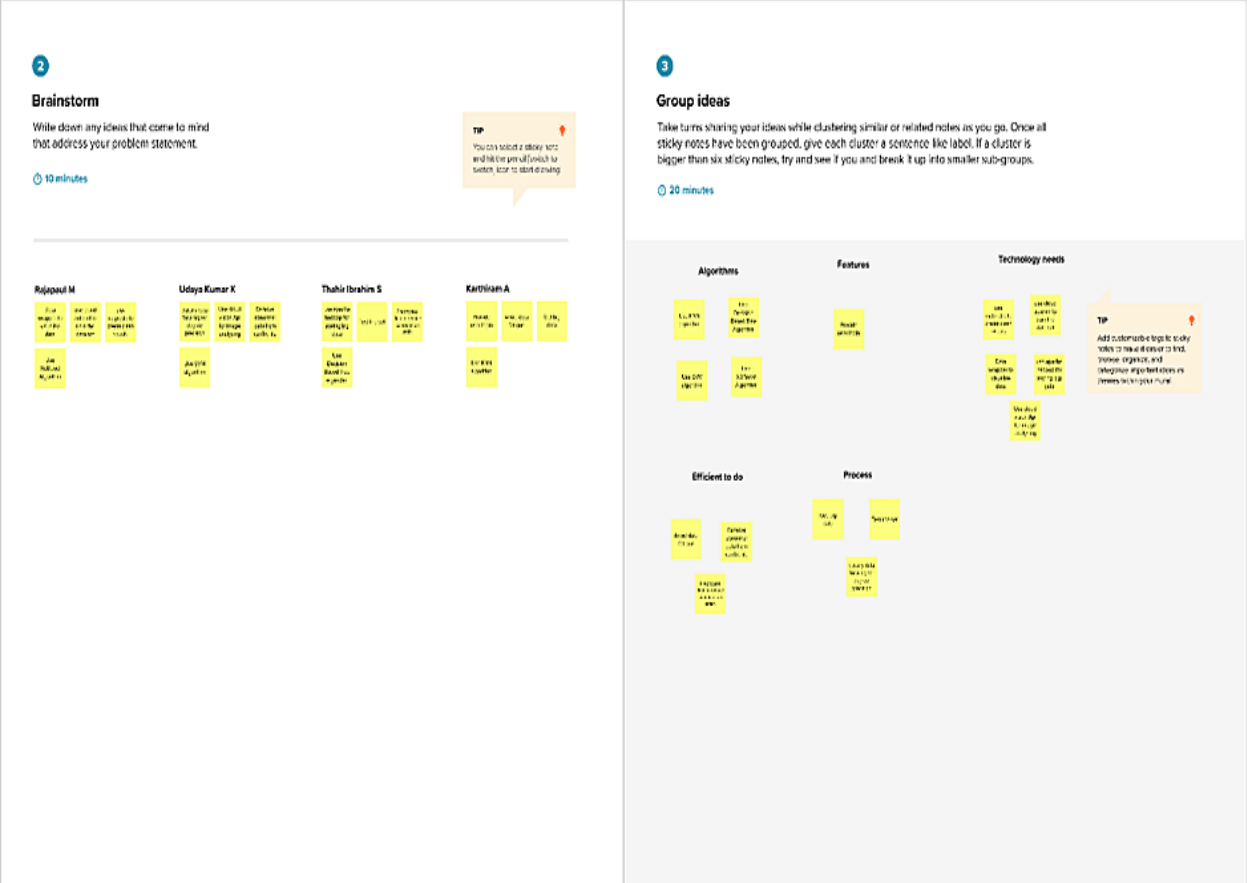
In brainstorming, a group discussion is made to produce ideas to solve the problem. The persons who indulge in brainstorming gives their opinion or approach or idea for the problem and finally the efficient ideas will be taken for the problem.Great brainstorms are ones that set the stage for fresh and generative thinking through simple guidelines and an open and collaborative environment. Use this when you're just kicking-off a new project and want to hit the ground running with big ideas that will move your team forward.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping

Everyone move their ideas into the "group sharing space" within the template and have the team silently read through them. As a team, sort and group them by thematic topics or similarities. Discuss and answer any questions that arise. Encourage "Yes, and…" and build on the ideas of other people along the way. This is the process in grouping the ideas.



Step-3: Idea Prioritization

The idea given by everyone is priorized in this step. Each person will give two icons to vote which idea should your team focus on.

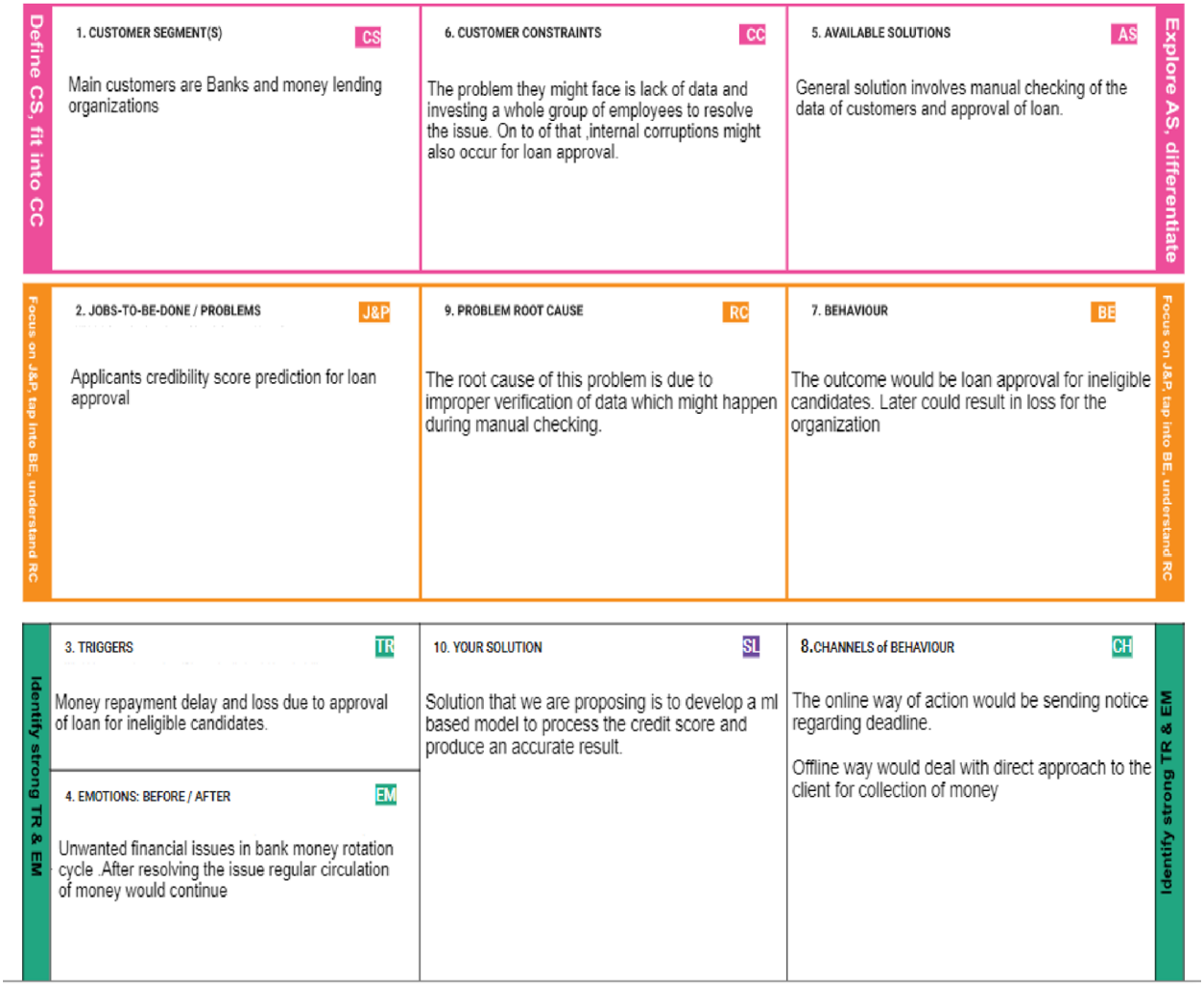
3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Many 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.	Idea / Solution description	An efficient Decision Tree is formulated with Decision Tree Induction Algorithm. It produces a model with the most relevant 6 attributes (Job, age, Income, Education, Marital Status, Existing loan). Attribute with rank-1 is placed as the root node of the Decision tree, other attributes from Rank-2 to Rank-6 constitute the intermediate nodes.
3.	Novelty / Uniqueness	Preprocessing of data such as normalization and scaling is not required which reduces the effort in building a model. Any missing value present in the data does not affect a decision tree which is why it is considered a flexible algorithm.
4.	Social Impact / Customer Satisfaction	By using Applicant Credibility Prediction for Loan Approval,the process of loan approval will be more efficient and transparent. Using various algorithms such as Decision tree , Xgboost,Random forest we can easily manage large data of customers and predict loan credibility.

5.	Business Model (Revenue Model)	Credit risk modelling is a technique used by lenders to assess the risk associated with extending credit to a specific application by reviewing a variety of factors, including the applicant's and co-income, applicant's educational background, credit history, and work status. The indicator of a borrower's creditworthiness is credit risk. We can use machine learning algorithms to forecast whether a specific application will be granted a loan or not with the aid of historical data patterns for loans supplied for the applicants.
6.	Scalability of the Solution	This process can be implemented in various banking sector and can be of good use. Numerous instances of computer glitches, content errors, and most crucially, the weight of features, have been resolved in automated prediction systems. As a result, in the near future, the aforementioned "software" may be designed to be more secure, dependable, and dynamically weighted.

3.4 Proposed Solution Fit

Problem solution fit used to the match between the problem and proposed solution. Problem-solution fit precedes product-market fit. The first indication of whether an idea will be successful is typically based on finding a problem-solution fit.



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User interaction	User must be able to view the home page of the website
FR-2	User input	User must be able to enter all his details in the fields.
FR-3	Data verification	The data entered by the user should be in correct format as required by the trained model
FR-4	Retrieving Values	The application must be able to retrieve the values present in the fields
FR-5	Predicting	The most accurate model is chosen for prediction
FR-6	Displaying the result	The prediction result is displayed in the User Interface.

4.2 Non-Functional Requirement

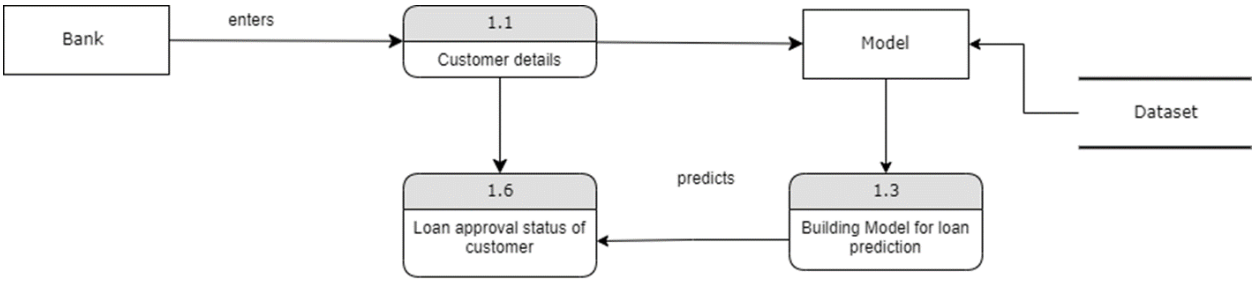
Following are the non-functional requirements of the proposed solution.

FR NO	Non-Functional Requirement	Description
NFR-1	Usability	The system must be easy to use. The user must be able to enter their data easily.
NFR-2	Security	Users' data must not be misused.
NFR-3	Reliability	The system should function without crashing.
NFR-4	Performance	The system must be able to withstand erroneous data and provide suitable messages.
NFR-5	Availability	The system must be available at all times.
NFR-6	Scalability	The system should be open for future

		developments, such as creating a login and storing data.

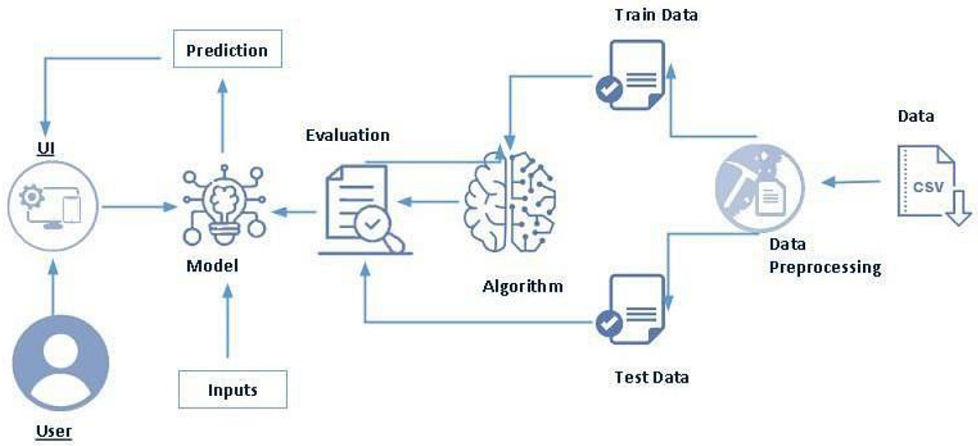
5. PROJECT DESIGN

5.1 Data Flow Diagram



5.2 Solution and Technical Architecture

Technical Architecture :



5.2 User Stories

User Type	Functional requirement	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Bank employee, Customer, Web user	User Interface	USN-1	As a user, I can view the web application and I am able to interact with it.	I can access the website	High	Sprint-1
		USN-2	As a user, I will be able to enter the details of the borrower whom I will be lending the loan.	I can enter the details in the given parameters list	High	Sprint-1
		USN-3	As a user, I can modify the data that I've entered before	I can modify the parameters list to an extent	Low	Sprint-2
		USN-4	As a user, I can ask for the data's prediction score at any time.	I can get some prediction score for any input	High	Sprint-1
		USN-5	As a user, I can request for mail transcript of the prediction along with the details given at the time to a specified email id.	I can request and receive an email transcript of the results	Medium	Sprint-2
Administrator		USN-6	As an admin, I can look at the past prediction stored in the cloud	I can view past predictions	Medium	Sprint-2
		USN-7	As an admin, I can tweak the ML model	I can modify ML model	High	Sprint-1

		USN-8	As an admin,I can modify the dataset and change the attributes	I can modify the dataset attributes	Low	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning And Estimation

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	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-1	Dataset	USN-5	Visualizing the dataset	2	Low	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-1	Dataset	USN-6	Pre-process the dataset	3	Medium	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-1	Machine Learning Model	USN-7	KNN model building	5	High	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-2	Machine LearningModel	USN-8	DecisionTree model building	5	High	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-2	Machine LearningModel	USN-9	Naive Bayes modelbuilding	5	High	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-2	Machine LearningModel	USN-10	Fine Tuning the model	3	Low	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-2	Machine LearningModel	USN-11	Evaluation and savingof the models	5	High	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-3	User Interface	USN-12	Model Integration withflask	5	High	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A

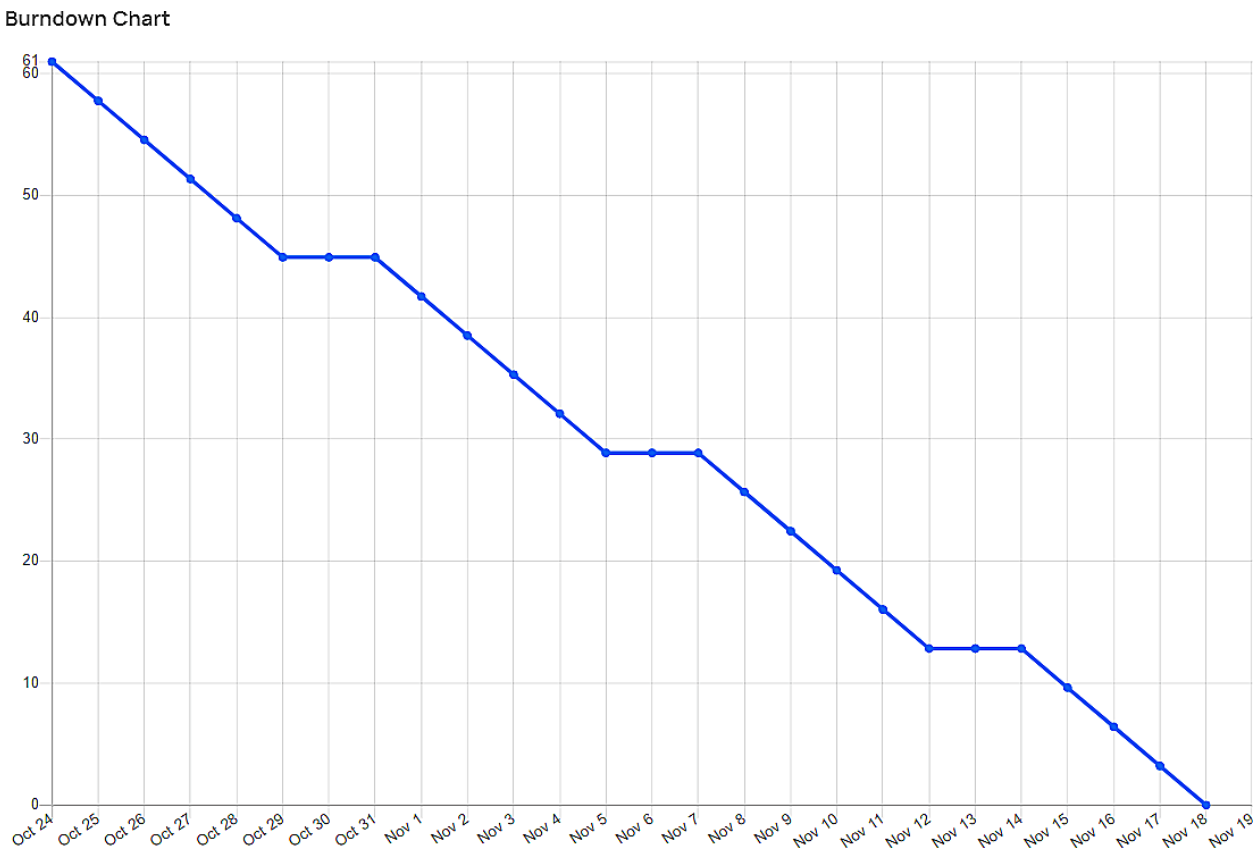
Sprint-3	UserInterface	USN-1	As a user, I should be able to access dashboard.	3	Medium	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-3	UserInterface	USN-2	Select the type of loan	3	Medium	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-3	UserInterface	USN-3	Fill theapplication and checkthe eligibility of loan approval	5	High	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-4	Deploying website	USN-13	Registeron IBM Cloud	3	Low	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-4	Deploying website	USN-14	Train the ML model on IBM Cloud	5	Medium	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A
Sprint-4	Deploying website	USN-15	Deploy the website on IBM Cloud	8	High	Udaya Kumar k Thahir Ibrahim S Rajapaul M Karthiram A

6.2 Sprint Delivery Schedule

Sprint	**Total StoryPoints**	**Duration**	**Sprint StartDate**	**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	05 Nov 2022
Sprint-2	18	6 Days	31 Oct 2022	05 Nov 2022	18	08 Nov 2022
Sprint-3	16	6 Days	07 Nov 2022	12 Nov 2022	16	14 Nov 2022

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

6.3 Report From JIRA



7. CODING AND SOLUTIONING

7.1 Feature 1

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
import seaborn as sns
%matplotlib inline
df = pd.read_csv('loan_prediction.csv')
df.head(10)

df.describe()

df.isnull().any()

df.drop('Loan_ID',axis=1,inplace=True)

df.Property_Area.unique()

plt.figure(figsize=(15,7))
df['ApplicantIncome'].hist(bins=25)
plt.show()

df.boxplot(column='ApplicantIncome',figsize=(15,7))
df.boxplot(column='ApplicantIncome', by = 'Education',figsize=(15,7))

plt.figure(figsize=(15,7))
df['LoanAmount'].hist(bins=25)
plt.show()

df.boxplot(column='LoanAmount',figsize=(15,7))
df.boxplot(column='LoanAmount', by = 'Gender',figsize=(15,7))

df['Property_Area'].value_counts()



df['Loan_Status'].value_counts()['Y']

pd.crosstab(df ['Credit_History'], df ['Loan_Status'], margins=True)

def percentageConvert(ser):
    return ser/float(ser[-1])

tabs = pd.crosstab(df ["Credit_History"], df ["Loan_Status"],
margins=True).apply(percentageConvert, axis=1)
tabs

app_loan = tabs['Y'][1]
```

```
print(f'{app_loan*100:.2f} % applicants got their loans approved')

df['Self_Employed'].fillna('No',inplace=True)

plt.figure(figsize=(15,7))
df['LoanAmount'].hist(bins=20)
plt.show()

df['LoanAmount'] = np.log(df['LoanAmount'])
plt.figure(figsize=(15,7))
df['LoanAmount'].hist(bins=25)
plt.show()
plt.figure(figsize=(15,7))
df['ApplicantIncome'].hist(bins=20)
plt.show()
df['ApplicantIncome'] = np.log(df['ApplicantIncome'])
plt.figure(figsize=(15,7))
df['ApplicantIncome'].hist(bins=25)
plt.show()

plt.figure(figsize=(15,7))
df['Loan_Amount_Term'].hist(bins=20)
plt.show()

df.head()

df['Gender'].fillna(df['Gender'].mode()[0],inplace=True)
df['Married'].fillna(df['Married'].mode()[0],inplace=True)
df['Dependents'].fillna(df['Dependents'].mode()[0],inplace=True)
df['LoanAmount'].fillna(df['LoanAmount'].mean(), inplace=True)
df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mean(),
inplace=True)
df['ApplicantIncome'].fillna(df['ApplicantIncome'].mean(),
inplace=True)
df['CoapplicantIncome'].fillna(df['CoapplicantIncome'].mean(),
inplace=True)

df['Gender'].fillna(df['Gender'].mode()[0], inplace=True)
df['Married'].fillna(df['Married'].mode()[0], inplace=True)
df['Dependents'].fillna(df['Dependents'].mode()[0], inplace=True)
df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mode()[0],
```

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

df.isnull().any()
df.head()

cat=['Gender','Married','Dependents','Education','Self_Employed','Cred
it_History','Property_Area']
target = ['Loan_Status']
all_cols = ['Gender', 'Married', 'Dependents', 'Education',
'Self_Employed',
       'ApplicantIncome', 'CoapplicantIncome', 'Loan_Amount_Term',
       'Credit_History', 'Property_Area', 'Loan_Status',
'TotalIncome_log',
       'LoanAmount_log']

from sklearn.preprocessing import LabelEncoder,OneHotEncoder
for var in cat:
    le = LabelEncoder()
    df[var]=le.fit_transform(df[var].astype('str'))

print('Done encoding Catergorical Values')
for tar in target:
    oe = OneHotEncoder()
    df[tar]=le.fit_transform(df[tar].astype('str'))
print('Done encoding Target Value')
df.head(5)

from sklearn.model_selection import train_test_split

train, test = train_test_split(df,test_size=0.2,random_state=42)
test.to_csv('test.csv',encoding='utf-8',index=False)
train.to_csv('train.csv',encoding='utf-8',index=False)
```

7.2 Feature 2

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MaxAbsScaler
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn.metrics import f1_score
import pickle


scaler = MaxAbsScaler()

train = pd.read_csv('train.csv')

test = pd.read_csv('test.csv')

train.head()

train_y = train.iloc[:,-1]

train_x = train.drop('Loan_Status',axis=1)

test_y = test.iloc[:,-1]

test_x = test.drop('Loan_Status',axis=1)

x = pd.concat([train_x,test_x],axis=0)
y = pd.concat([train_y,test_y],axis=0)

train_x = scaler.fit_transform(train_x)

test_x = scaler.transform(test_x)

def decisionTree(train_x,test_x,train_y,test_y):
    dt = DecisionTreeClassifier()
    dt.fit(train_x,train_y)
    y_pred = dt.predict(test_x)
    print("**** Decision Tree Classifier ****")
    print('Confusion Matrix')
```

```
    print(confusion_matrix(test_y,y_pred))
    print('Classification Report')
    print(classification_report(test_y,y_pred))

def randomForest(train_x,test_x,train_y,test_y):
    rf = RandomForestClassifier()
    rf.fit(train_x,train_y)
    y_pred = rf.predict(test_x)
    print("**** Random Forest Classifier ****")
    print('Confusion Matrix')
    print(confusion_matrix(test_y,y_pred))
    print('Classification Report')
    print(classification_report(test_y,y_pred))


def knn(train_x,test_x,train_y,test_y):
    knn = KNeighborsClassifier()
    knn.fit(train_x,train_y)
    y_pred = knn.predict(test_x)
    print("**** KNeighbour Classifier ****")
    print('Confusion Matrix')
    print(confusion_matrix(test_y,y_pred))
    print('Classification Report')
    print(classification_report(test_y,y_pred))



def xgboost(train_x,test_x,train_y,test_y):
    xg = GradientBoostingClassifier()
    xg.fit(train_x,train_y)
    y_pred = xg.predict(test_x)
    print("**** Gradient Boosting Classifier ****")
    print('Confusion Matrix')
    print(confusion_matrix(test_y,y_pred))
    print('Classification Report')
    print(classification_report(test_y,y_pred))


decisionTree(train_x,test_x,train_y,test_y)

randomForest(train_x,test_x,train_y,test_y)
```

```
knn(train_x,test_x,train_y,test_y)

xgboost(train_x,test_x,train_y,test_y)
rf = RandomForestClassifier()
rf.fit(train_x,train_y)
ypred = rf.predict(test_x)
f1_score(ypred,test_y,average='weighted')

cv = cross_val_score(rf,x,y,cv=5)

np.mean(cv)
pickle.dump(rf,open('rdf.pkl','wb'))
pickle.dump(scaler,open('scale.pkl','wb'))
```

8. TESTING

8.1 Test Cases

For checking the loan application, We have two testcase

- Eligible
- Not Eligible

This is based on the training and testing the model we used in our application.
This eligibility can be checked by using the details entered by the users. This includes the details like

- Gender
- Status
- Dependants
- Education
- Employ
- Income
- Co-income(additional income)
- Loan amount
- Loan amount term(in days)
- Credit history
- Property area(type of location)

8.2 User Acceptance Testing

1. Purpose of Document
The purpose of this document is to briefly explain the test coverage and open issues of the project - Smart Lender - Applicant Credibility Prediction for Loan Approval at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1 (High)	Severity 2 (Moderate)	Severity 3 (Low)	Subtotal
By Design	1	3	2	6
Duplicate	1	0	3	4
External	2	3	0	5
Fixed	4	6	4	14
Not Reproduced	0	0	1	1
Totals	8	12	10	30

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	16	0	0	16
Security	2	0	0	2
Exception Reporting	3	0	0	3
Final Report Output	4	0	0	4
Version Control	1	0	0	1

9.RESULT

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values
1.	Metrics	**Classification Model:** Confusion Matrix – [[18 25] [2 75]], Accuracy Score – 79% & Classification Report –

	precision	recall	f1-score	support
0	0.90	0.42	0.57	43
1	0.76	0.97	0.85	80
accuracy			0.78	123
macro avg	0.83	0.70	0.71	123
weighted avg	0.81	0.78	0.75	123

In [14]:

```
def randomForest(train_x,test_x,train_y,test_y):
    rf = RandomForestClassifier()
    rf.fit(train_x,train_y)
    y_pred = rf.predict(test_x)
    print("**** Random Forest Classifier ****")
    print('Confusion Matrix')
    print(confusion_matrix(test_y,y_pred))
    print('Classification Report')
    print(classification_report(test_y,y_pred))
```

In [18]:
```
randomForest(train_x,test_x,train_y,test_y)
```

```
**** Random Forest Classifier ****
Confusion Matrix
[[18 25]
 [ 2 78]]
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.42      0.57        43
           1       0.76      0.97      0.85        80

    accuracy                           0.78       123
   macro avg       0.83      0.70      0.71       123
weighted avg       0.81      0.78      0.75       123
```

In [22]:
```
f1_score(ypred,test_y,average='weighted')
```

Out[22]: 0.7977251407129455

In [23]:
```
cv = cross_val_score(rf,x,y,cv=5)
```

In [24]:
```
np.mean(cv)
```

Out[24]: 0.7915367186458749

10.ADVANTAGES AND DISADVANTAGES

ADVANTAGES	DISADVANTAGES
1. Optimization of Loan LifeCycle. 2. Digital Lendingtechnology thrives onprocess speed 3. Easy capture of Applicant information 4. Quicker Decision Making 5. Consistency 6. Comfort acrossDevices 7. Perfect for first timeborrowers 8. Compliance with Rules andRegulations 9. Power of Analytics	1. Strict eligibility criteria 2. One of the major disadvantages of a bank loanis that bankscan be cautious about lending to small businesses 3. Lengthy application process 4. Not suitable for ongoingexpenses 5. Secured loanscarry risk

11.CONCLUSION

In this study, we present a method for predicting whether a consumer will repay a loan using supervised learning techniques. Different algorithms were used to forecast consumer loans. Decision Tree Classifier, Random Forest, KNN, and SVM were used to achieve the best results. Random forest has the highest accuracy when compared to the other three methods. It can be confidently concluded from a correct study of the part's advantages and limitations that the product might be a very effective component. This application is operating properly and satisfies any requirements set forth by the Banker. In many different systems, this area is frequently blocked. There are several instances of content faults and computer glitches, and the biggest weight of choice is mounted in a machine-driven.

12.FUTURE SCOPE

Smart lender platforms have a promising future because there is a sizable untapped lending market at around 20 lakh crores. These lending platforms have a great deal of potential to offer cutting-edge solutions and revolutionary technology for the borrowing and lending industry because they are technology-based. The generation of today is significantly better educated, digitally savvy, and inclined to spend money on a variety of life activities. Platforms that offer instant approval and paperless loans with little time and personal interaction are anticipated to grow significantly as the habit of consumerization spreads because they give the general public the opportunity for financial inclusion while also producing higher returns for lenders to encourage them.

Submit.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SMART LENDER</title>
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Aref+Ruqaa+Ink:wght@700&display=swap');
        @import url('https://fonts.googleapis.com/css2?family=EB+Garamond&display=swap');

        body {
```

```
    color: white;
    font-family: 'Aref Ruqaa Ink', serif;
    background: rgb(20, 30, 48);
    background: linear-gradient(to right,rgb(255, 84, 84) 20%,rgb(65, 65, 255) 50%);
}

.output {
    margin-top: 15%;
}


/*--- Footer ---*/

.footer {
    margin-top: 28vh;
}

.nav-link {
    font-weight: bold;
    font-size: 14px;
    text-transform: uppercase;
    text-decoration: none;
    color: #031D44;
    padding: 20px 0px;
    /* margin: 0px 20px;*/

    display: inline-block;
    position: relative;
    opacity: 0.75;
}

#d {
    margin-top: -40px;
    font-family: 'EB Garamond', serif;
    letter-spacing: 0.5px;
}
```

```
        #p {
            /* margin-top: -50px;*/
            font-family: 'EB Garamond', serif;
            letter-spacing: 0.5px;
        }

        .nav-link:hover {
            opacity: 1;
        }

        .nav-link::before {
            transition: 300ms;
            height: 3px;
            content: "";
            position: absolute;
            background-color: #031D44;
        }

        .nav-link-fade-up::before {
            width: 100%;
            bottom: 5px;
            opacity: 0;
        }

        .nav-link-fade-up:hover::before {
            bottom: 10px;
            opacity: 1;
        }

        p {
            color: white;
            font-family: 'Aref Ruqaa Ink', serif;
            letter-spacing: 0.5px;
        }
    </style>
</head>
```

```
<body>
   <main class="output">
      <center>
         <h1>SMART LENDER</h1>
         <h3>{{prediction_text}}</h3>
      </center>
   </main>
   <center>
      <footer  style="margin-bottom: 2rem;">

         <div>
            <p>Developed by:-</p> <br>
<h3>Karthiram</h3>
<h3>   Thahir Ibrahim </h3>
<h3>   Udaya Kumar</h3>
<h3>Rajapaul</h3>




         </div>
      </footer>
   </center>
</body>

</html>
```

Predict.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>SMART LENDER</title>
   <style>
```

```
        @import
url('https://fonts.googleapis.com/css2?family=Aref+Ruqaa+Ink:wght@700&display=swap');
        @import url('https://fonts.googleapis.com/css2?family=Albert+Sans&display=swap');
        @import url('https://fonts.googleapis.com/css2?family=EB+Garamond&display=swap');

        html {
            height: 100%;
        }

        body {
            margin: 0;
            padding: 0;
            font-family: sans-serif;
            /* background: linear-gradient(#141e30, #243b55);*/
            background: rgb(20, 30, 48);
            background: radial-gradient(circle, #daf3ff 5%, #161e24 95%);
        }

        .login-box {

            width: 400px;
            padding: 40px;
margin: auto;           justify-content: center;
            align-items: center;
            background: rgba(0, 0, 0, .5);
            box-sizing: border-box;
            box-shadow: 0 15px 25px rgba(0, 0, 0, .6);
            border-radius: 10px;
        }

        ::placeholder {
            color: aliceblue;
        }

        .login-box h2 {
            margin: 0 0 30px;
            padding: 0;
```

```
    color: #fff;
    text-align: center;
}

.fon {
    color: #fff;
    text-align: center;
    font-family: 'Albert Sans', sans-serif;
}

.login-box .user-box {
    position: relative;
}

.login-box .user-box input {
    width: 100%;
    padding: 10px 0;
    font-size: 16px;
    color: #fff;
    margin-bottom: 30px;
    border: none;
    border-bottom: 1px solid #fff;
    outline: none;
    background: transparent;
}

.login-box .user-box label {
    position: absolute;
    top: 0;
    left: 0;
    padding: 10px 0;
    font-size: 16px;
    color: #fff;
    pointer-events: none;
    transition: .5s;
}
```

```
.login-box .user-box input:focus~label,
.login-box .user-box input:valid~label {
    top: -20px;
    left: 0;
    color: #03e9f4;
    font-size: 12px;
}

/*--- Button */

.container,
.container:before,
.container:after {
    box-sizing: border-box;
    padding: 0;
    margin: 0;
    font: 300 1em/1.5 'Open Sans', 'Helvetica Neue', Arial, sans-serif;
    text-decoration: none;
    color: #111;
}

.btn {
    background: rgb(236, 240, 241);
}

.container {
    min-width: 500px;
    margin: 5% auto;
    text-align: center;
}


button:hover {
    cursor: pointer
}

button {
```

```
    background: transparent;
    outline: none;
    position: relative;
    border: 3px solid #81C6E8;
    padding: 15px 50px;
    overflow: hidden;
}

/*button:before (attr data-hover)*/
button:hover:before {
    opacity: 1;
    transform: translate(0, 0);
}

button:before {
    content: attr(data-hover);
    position: absolute;
    top: 1.1em;
    left: 0;
    width: 100%;
    text-transform: uppercase;
    letter-spacing: 3px;
    font-weight: 800;
    font-size: .8em;
    opacity: 0;
    transform: translate(-100%, 0);
    transition: all .3s ease-in-out;
}

/*button div (button text before hover)*/
button:hover div {
    opacity: 0;
    transform: translate(100%, 0)
}

button div {
    text-transform: uppercase;
```

```
    letter-spacing: 3px;
    font-weight: 800;
    font-size: .8em;
    transition: all .3s ease-in-out;
}


/*--- Footer ---*/

.footer {
    margin-top: 203vh;
}

.nav-link {
    font-weight: bold;
    font-size: 14px;
    text-transform: uppercase;
    text-decoration: none;
    color: #031D44;
    padding: 20px 0px;
    /* margin: 0px 20px;*/

    display: inline-block;
    position: relative;
    opacity: 0.75;
}

#d {
    margin-top: -40px;
    font-family: 'EB Garamond', serif;
    letter-spacing: 0.5px;
}

#p {
    margin-top: -50px;
    font-family: 'EB Garamond', serif;
    letter-spacing: 0.5px;
```

```
        }

        .nav-link:hover {
            opacity: 1;
        }

        .nav-link::before {
            transition: 300ms;
            height: 3px;
            content: "";
            position: absolute;
            background-color: #031D44;
        }

        .nav-link-fade-up::before {
            width: 100%;
            bottom: 5px;
            opacity: 0;
        }

        .nav-link-fade-up:hover::before {
            bottom: 10px;
            opacity: 1;
        }

        p {
            color: white;
            font-family: 'Aref Ruqaa Ink', serif;
            letter-spacing: 0.5px;
        }
    </style>
</head>

<body>

    <div class="login-box">
        <h2 style="text-transform: uppercase; font-family: 'Aref Ruqaa Ink', serif;">Smart lender -
```

```
<br> <span
            style="font-size:14px ; color:azure">Know your
            Loan eligibility</span> </h2>
        <p class="fon" style="font-size: 14px ;">Let's begin by entering your deatils below</p>
<br>
        <form action="/submit" method="post">
            <div class="user-box">
                <input type="text" name="" required="" onfocus="this.placeholder='Enter your name'"
                    onblur="this.placeholder="">
                <label>Name</label>
            </div>
            <div class="user-box">
                <input type="text" name="Loan_ID" required="" onfocus="this.placeholder='Enter
your Loan ID'"
                    onblur="this.placeholder="">
                <label>Loan ID</label>
            </div>
            <div class="user-box">
                <input list="gender" type="data-list" name="Gender" required=""
onchange="resetIfInvalid(this);"
                    onfocus="this.placeholder='Enter your Gender'" onblur="this.placeholder="">
                <label>Gender</label>
                <datalist id="gender" name="gender">
                    <option value="Male"></option>
                    <option value="female"></option>
                </datalist>
            </div>
            <div class="user-box">
                <input list="married" type="text" name="Married" required=""
onchange="resetIfInvalid(this);"
                    onfocus="this.placeholder='Enter your Marital Status'"
onblur="this.placeholder="">
                <label>Married</label>
                <datalist id="married" name="married">
                    <option value="yes"></option>
                    <option value="no"></option>
                </datalist>
```

```
        </div>
        <div class="user-box">
            <input list="dep" type="text" name="Dependents" required=""
onchange="resetIfInvalid(this);"
                onfocus="this.placeholder='Enter your Dependents'" onblur="this.placeholder="">
            <label>Dependents</label>
            <datalist id="dep" name="dep">
                <option value="0"></option>
                <option value="1"></option>
                <option value="2"></option>
                <option value="3+"></option>
            </datalist>
        </div>
        <div class="user-box">
            <input list="edu" type="text" name="Education" required=""
onchange="resetIfInvalid(this);"
                onfocus="this.placeholder='Enter your Educational Qualification'"
onblur="this.placeholder="">
            <label>Education</label>
            <datalist name="edu" id="edu">
                <option value="Graduate"></option>
                <option value="Non-Graduate"></option>
            </datalist>
        </div>
        <div class="user-box">
            <input list="emp" type="text" name="Self_Employes" required=""
onchange="resetIfInvalid(this);"
                onfocus="this.placeholder='Are you self employed?'" onblur="this.placeholder="">
            <label>Self Employed</label>
            <datalist name="emp" id="emp">
                <option value="yes"></option>
                <option value="no"></option>
            </datalist>
        </div>
        <div class="user-box">
            <input type="number" name="ApplicantIncome" required=""
                onfocus="this.placeholder='Enter your Income in Dollars'"
```

```
onblur="this.placeholder="">
            <label>Applicant Income</label>
        </div>
        <div class="user-box">
            <input type="number" name="CoaaplicantIncome" required=""
                onfocus="this.placeholder='Enter your CO Applicant Income in Dollars'"
onblur="this.placeholder="">
            <label>CO Applicant Income</label>
        </div>
        <div class="user-box">
            <input type="number" name="LoanAmount" required=""
                onfocus="this.placeholder='Enter your Loan Amount in Dollars'"
onblur="this.placeholder="">
            <label>Loan Amount</label>
        </div>
        <div class="user-box">
            <input list="term" type="text" name="Loan_Amount_Term" required=""
onchange="resetIfInvalid(this);"
                onfocus="this.placeholder='Enter the loan amount term'"
onblur="this.placeholder="">
            <label>Loan Amount Term</label>
            <datalist name="term" id="term">
                <option value="480"></option>
                <option value="360"></option>
                <option value="300"></option>
                <option value="240"></option>
                <option value="180"></option>
                <option value="120"></option>
                <option value="84"></option>
                <option value="60"></option>
                <option value="36"></option>
                <option value="12"></option>
            </datalist>
        </div>
        <div class="user-box">
            <input list="credit" type="text" name="Credit_History" required=""
onchange="resetIfInvalid(this);"
```

```
                onfocus="this.placeholder='Enter your Credit History'"
onblur="this.placeholder=''">
            <label>Credit History</label>
            <datalist name="credit" id="credit">
                <option value="yes"></option>
                <option value="no"></option>
            </datalist>
        </div>
        <div class="user-box">
            <input list="prop" type="text" name="Property_Area" required=""
onchange="resetIfInvalid(this);"
                onfocus="this.placeholder='Enter your area of the property'"
onblur="this.placeholder=''">
            <label>Property Area</label>
            <datalist name="prop" id="prop">
                <option value="Urban"></option>
                <option value="Rural"></option>
                <option value="Semi-Rural"></option>
            </datalist>
        </div>

        <div class="container">
            <a href="submit.html">
                <button class="btn" data-hover="PREDICT" onclick="submit.html">
                    <div>SUBMIT</div>
                </button>
            </a>
        </div>
    </form>
</div>

<center>
    <footer  style="margin-bottom: 2rem;">

        <div>
            <p>Developed by:-</p> <br>
<h3>Karthiram</h3>
```

```
<h3>   Thahir Ibrahim </h3>
<h3>   Udaya Kumar</h3>
<h3>Rajapaul</h3>




        </div>
      </footer>
   </center>

</body>
<script>
   function resetIfInvalid(el) {
      //just for beeing sure that nothing is done if no value selected
      if (el.value == "")
         return;
      var options = el.list.options;
      for (var i = 0; i < options.length; i++) {
         if (el.value == options[i].value)
            //option matches: work is done
            return;
      }
      //no match was found: reset the value
      el.value = "";
   }
</script>

</html>
```

Index.html:

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SMART LENDER</title>
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Aref+Ruqaa+Ink:wght@700&display=swap');

        @import url('https://fonts.googleapis.com/css2?family=EB+Garamond&display=swap');
        @import url('https://fonts.googleapis.com/css2?family=Antic+Slab&display=swap');

        html {
            user-select: none;
        }

        body {
            margin-top: 7%;
            color: white;
        }

        html {
            background: rgb(20, 30, 48);
            background: linear-gradient(to right,white,red,orange,pink)
        }

        h1 {
            font-size: 45px;
            font-family: 'Aref Ruqaa Ink', serif;
        }

        h3 {
            font-size: 20px;
            font-family: 'Antic Slab', serif;
        }

        h6 {
            font-size: 20px;
            font-family: 'Antic Slab', serif;
```

```
}

/* ~~~~~~~ BUTTON ~~~~~~~ */

.container,
.container:before,
.container:after {
    box-sizing: border-box;
    padding: 0;
    margin: 0;
    font: 300 1em/1.5 'Open Sans', 'Helvetica Neue', Arial, sans-serif;
    text-decoration: none;
    color: #111;
}

.btn {
    background: rgb(236, 240, 241);
}

.container {
    min-width: 500px;
    margin: 5% auto;
    text-align: center;
}


button:hover {
    cursor: pointer
}

button {
    background: transparent;
    outline: none;
    position: relative;
    border: 3px solid #031D44;
    padding: 15px 50px;
    overflow: hidden;
```

```
}

/*button:before (attr data-hover)*/
button:hover:before {
    opacity: 1;
    transform: translate(0, 0);
}

button:before {
    content: attr(data-hover);
    position: absolute;
    top: 1.1em;
    left: 0;
    width: 100%;
    text-transform: uppercase;
    letter-spacing: 3px;
    font-weight: 800;
    font-size: .8em;
    opacity: 0;
    transform: translate(-100%, 0);
    transition: all .3s ease-in-out;
}

/*button div (button text before hover)*/
button:hover div {
    opacity: 0;
    transform: translate(100%, 0)
}

button div {
    text-transform: uppercase;
    letter-spacing: 3px;
    font-weight: 800;
    font-size: .8em;
    transition: all .3s ease-in-out;
}
```

```
/*--- Footer ---*/

.footer {
    margin-top: 10px;
}

.nav-link {
    font-weight: bold;
    font-size: 14px;
    text-transform: uppercase;
    text-decoration: none;
    color: #031D44;
    padding: 20px 0px;
    /* margin: 0px 20px;*/

    display: inline-block;
    position: relative;
    opacity: 0.75;
}

#d {
    margin-top: -40px;
    font-family: 'EB Garamond', serif;
    letter-spacing: 0.5px;
}

#p {
    margin-top: -50px;
    font-family: 'EB Garamond', serif;
    letter-spacing: 0.5px;
}

.nav-link:hover {
    opacity: 1;
}

.nav-link::before {
```

```
            transition: 300ms;
            height: 3px;
            content: "";
            position: absolute;
            background-color: #031D44;
        }

        .nav-link-fade-up::before {
            width: 100%;
            bottom: 5px;
            opacity: 0;
        }

        .nav-link-fade-up:hover::before {
            bottom: 10px;
            opacity: 1;
        }

        p {
            color: white;
            font-family: 'Aref Ruqaa Ink', serif;
            letter-spacing: 0.5px;
        }
    </style>

</head>

<body>
    <main>
        <center>
            <h1>Smart Lender</h1>
            <h3>Get to know your applicant application will get accepted or not</h3>
            <h6>Click the <em><b> Predict </b></em> button and update the details to know the
prediction for the
                applicant.
            </h6>
```

```
        <div class="container">
            <a href="predict.html">
                <button class="btn" data-hover="Loan Predictor" onclick="predict.html">
                    <div>Predict</div>
                </button>
            </a>
        </div>

        <footer>
            <div class="footer">
                <p>Developed by:-</p> <br>
                <h3>Karthiram</h3>
                <h3>   Thahir Ibrahim </h3>
                <h3>   Udaya Kumar</h3>
                <h3>Rajapaul</h3>




            </div>
        </footer>
    </center>


  </main>
</body>

</html>
```

App.py:

```
from flask import render_template,Flask,request
import numpy as np
import pickle

app= Flask(__name__, template_folder ='templates')

model = pickle.load(open(r'./rdf.pkl','rb'))
scale = pickle.load(open(r'./scale.pkl','rb'))

@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict.html')
def formpg():
    return render_template('predict.html')
@app.route('/submit',methods = ['POST'])
def predict():

loan_num,gender,married,depend,education,self_emp,applicant_income,co_income,loan_amount
,loan_term,credit_history,property_area = [x for x in request.form.values()]
    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 self_emp == 'Yes':
        self_emp = 1
```

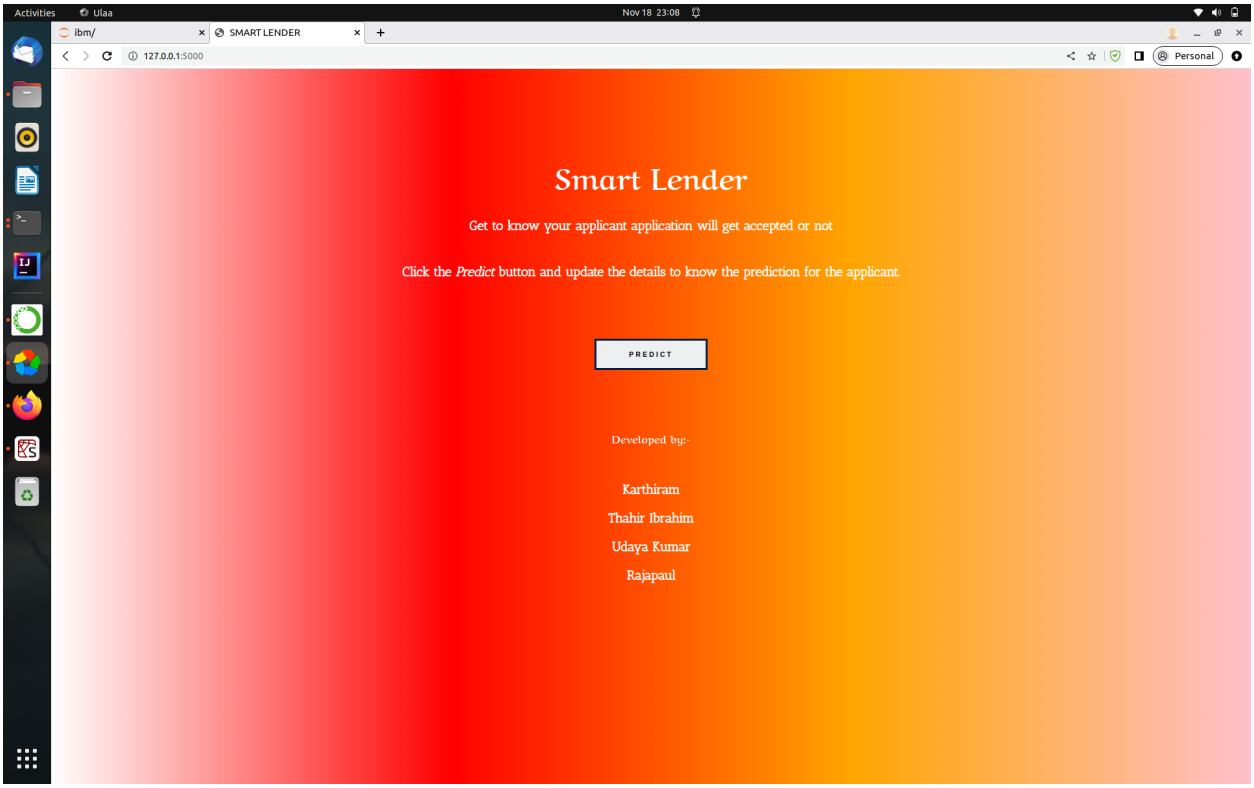
```
    else:
        self_emp = 0

    if depend == '3+':
        depend = 3

    applicant_income = int(applicant_income)
    applicant_income = np.log(applicant_income)
    loan_amount = int(loan_amount)
    loan_amount = np.log(loan_amount)

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

    if property_area == 'Urban':
        property_area = 2

    elif property_area == 'Rural':
        property_area = 0
    else:
        property_area = 1

    features =
[gender,married,depend,education,self_emp,applicant_income,co_income,loan_amount,loan_ter
m,credit_history,property_area]
    con_features = [np.array(features)]
    scale_features = scale.fit_transform(con_features)
    prediction = model.predict(scale_features)
    print(prediction)
    if prediction == 0:
        return render_template('submit.html', prediction_text ='You are eligible for loan')
    else:
        return render_template('submit.html',prediction_text = 'Sorry you are not eligible for loan')
```

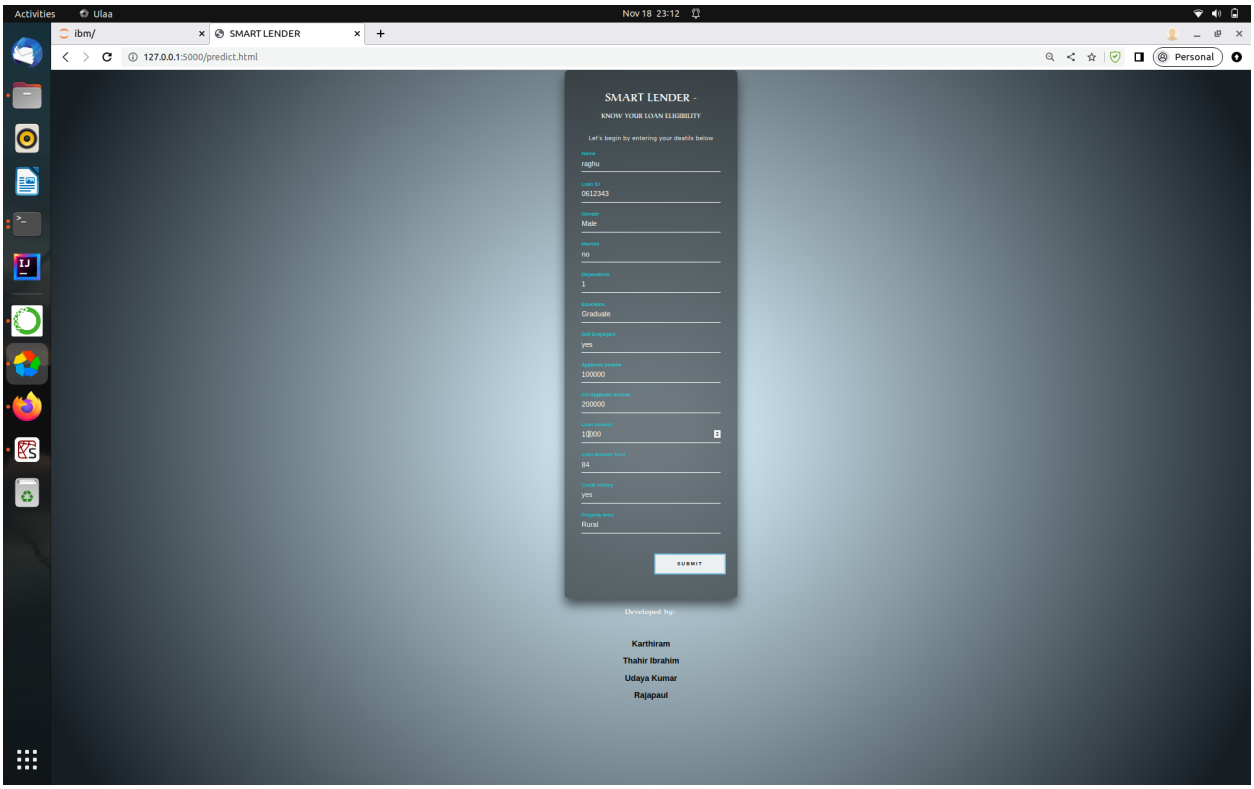
```
if __name__ == "__main__":
    app.run(debug=True)
```

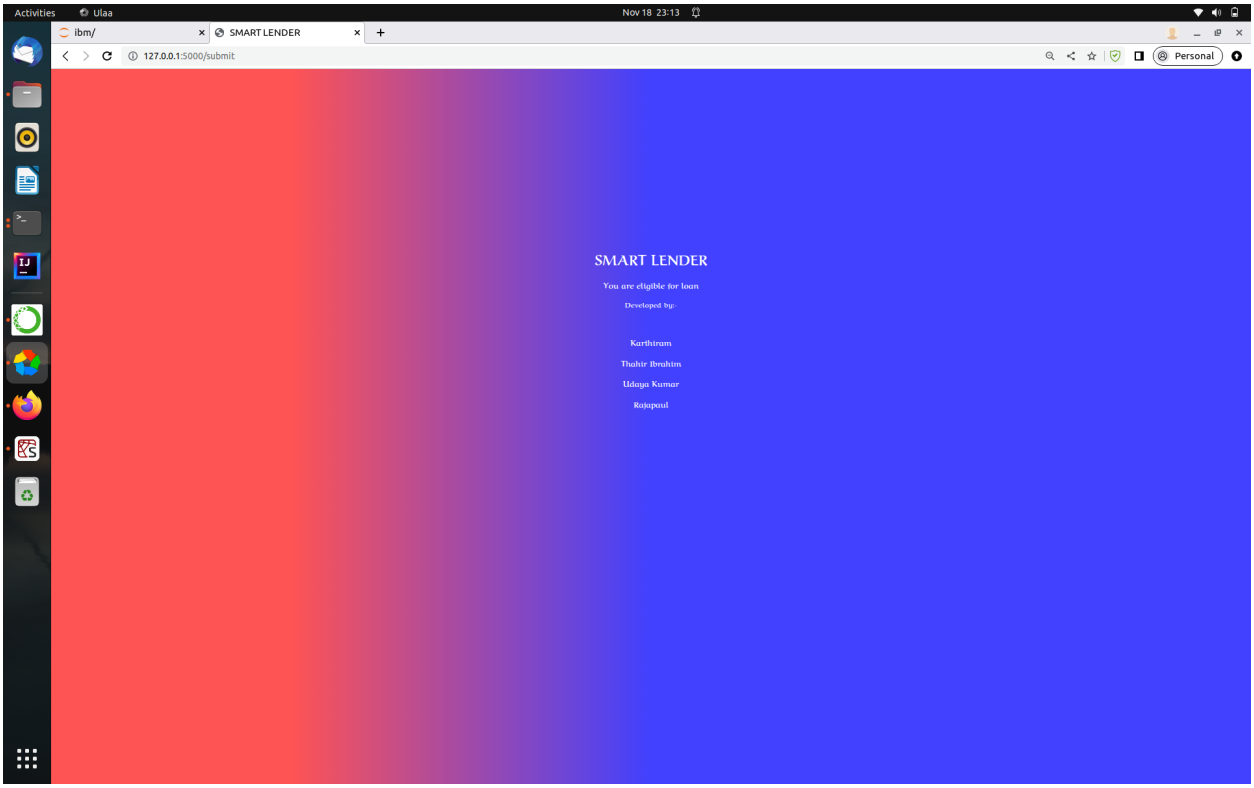
Github Link:

https://github.com/IBM-EPBL/IBM-Project-32790-1660212069

Demo Screenshot:




Activities
Ulaa
Nov 18 23:12
ibm/
SMART LENDER
127.0.0.1:5000/predict.html
Personal
SMART LENDER -
KNOW YOUR LOAN ELIGIBILITY
Let's begin by entering your deatils below
Name
raghu
Loan ID
0612343
Gender
Male
Married
no
Dependents
1
Education
Graduate
Self Employed
yes
Applicant Income
100000
CO Applicant Income
200000
Loan Amount
10000
Loan Amount Term
84
Credit History
yes
Property Area
Rural
SUBMIT
Developed by:-
Karthiram
Thahir Ibrahim
Udaya Kumar
Rajapaul


Activities
Ulaa
Nov 18 23:13
ibm/
SMART LENDER
127.0.0.1:5000/submit
Personal
SMART LENDER
You are eligible for loan
Developed by:-
Karthiram
Thahir Ibrahim
Udaya Kumar
Rajapaul