PROJECT REPORT

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.

1.2 PURPOSE

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.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

In reference [1], Typical data collection protocols deployed at many financial institutions for loan approval and loan pricing are reviewed. Key steps involved in improving information quality for all parties involved are discussed.

In reference [2], Decision Tree Induction Data Mining Algorithm is applied to predict the attributes relevant for credibility. A prototype of

the model is described which can be used by the organizations in making the right decision to approve or reject the loan request of the customers.

In reference [3], The approach in this study is a hybrid under sampling method that combines the clustering, the stochastic sensitivity measure and the radial basis function neural networks. A real loan default data from a P2P company in China is used to valid the performance of our method.

2.2 REFERENCES

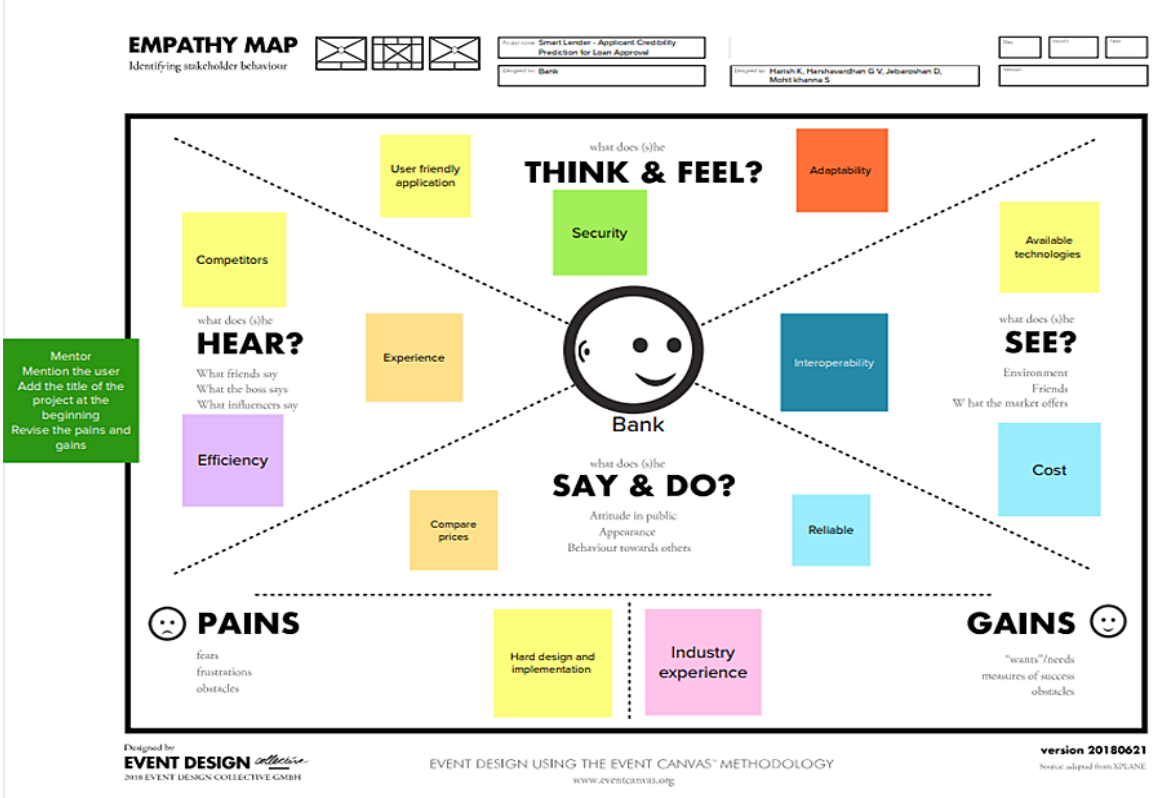
[1] M. Cary Collins, "Improving Information Quality in Loan Approval Processes for Fair Lending and Fair Pricing" 2013.
[2] Sivasree M S, Rekha Sunny T, "Loan Credibility Prediction System Based on Decision Tree Algorithm" 2015.
[3] Y. -Q. Chen, J. Zhang and W. W. Y. Ng, "Loan Default Prediction Using Diversified Sensitivity Under sampling," 2018.

2.3 PROBLEM STATEMENT DEFINITION

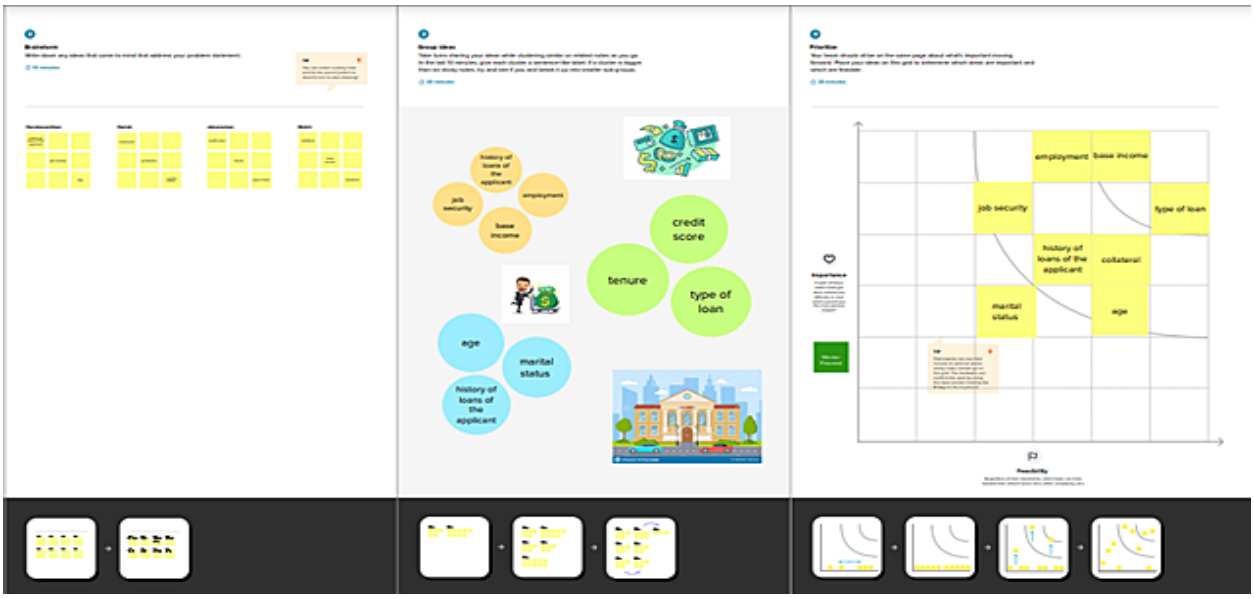
Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependants, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers. It is a classification problem where we have to predict whether a loan would be approved or not.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING

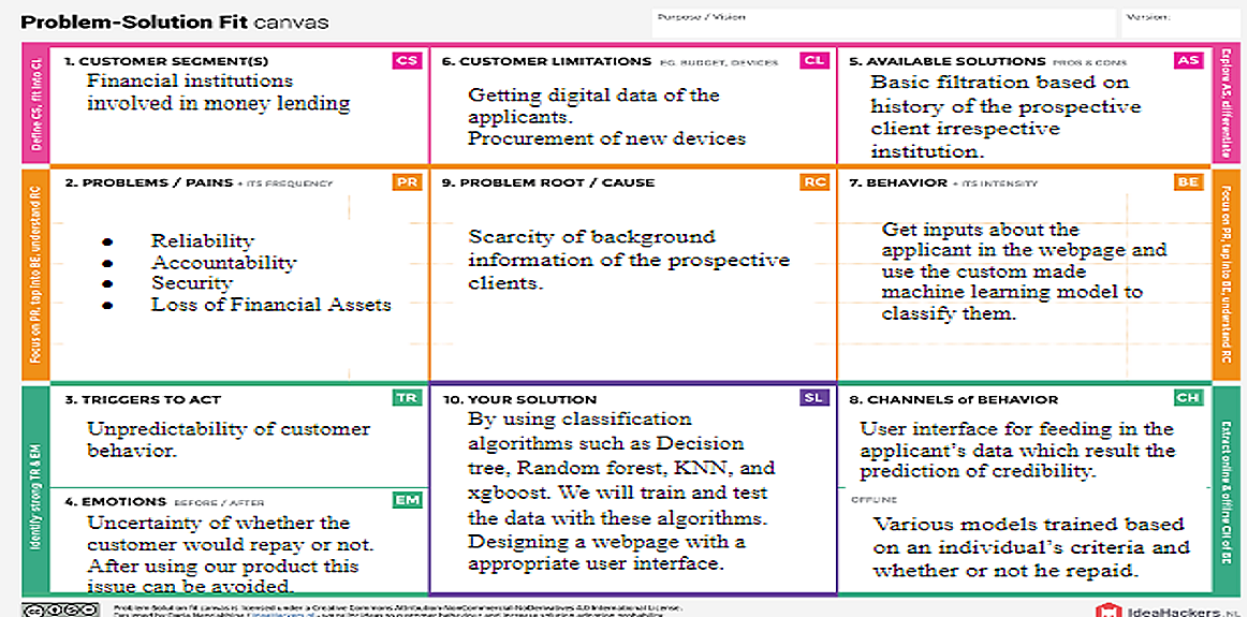


3.3 PROPOSED SOLUTION

This solution template relates the current situation to a desired result of this project and also describes the benefits acquired when the desired result is achieved

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Our objective is to automate the loan eligibility process (real-time) based on customer detail provided while filling out online application forms. These details are Marital Status, Education, number of Dependants, Income, Loan Amount, Credit History, and others.
2.	Idea / Solution description	Understanding the dataset and deciding on the number of features. By using classification algorithms such as Decision tree, Random forest, KNN, and xgboost. We will train and test the data with these algorithms. Designing a webpage with a appropriate user interface.
3.	Novelty / Uniqueness	Creating a interactive user interface. Including various features that are meaningful according to the applicant.
4.	Social Impact / Customer Satisfaction	As the global economy depends on the banking system, their main problem is money supply which can be solved by our solution. This will ultimately help RBI to control the money supply in the market.
5.	Business Model (Revenue Model)	Many banks are demanding automated system for loan approval which can significantly increase their revenue by reducing labour cost ,time and increasing the efficiency of the system
6.	Scalability of the Solution	We can extend this at the people end by suggesting them the opportunities available to them to apply various types of loans.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 Functional requirements

FR NO.	FUNTIONAL REQUIREMENT	SUB REQUIREMENT
FR-1	Home page	Access to all other pages. Smart Lender application Credibility is shown
FR-2	User Registration	Enter email id and other personal details for registration
FR-3	User login	User email id and password for login
FR-4	Loan approval form	Credibility details should be entered for prediction
FR-5	Result	Display shows whether loan is approved or rejected

4.2 Non-Functional requirements

FR NO.	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
NFR-1	Usability	The interface we provide with must be easy to use.
NFR-2	Performance	It must produce highly accurate predictions.
NFR-3	Reliability	Predictions must stay consistent.
NFR-4	Availability	Must be ready to use whenever necessary.
NFR-5	Scalability	Must have very good prospective updates for the future.
NFR-6	Security	Individual accounts must be secure.

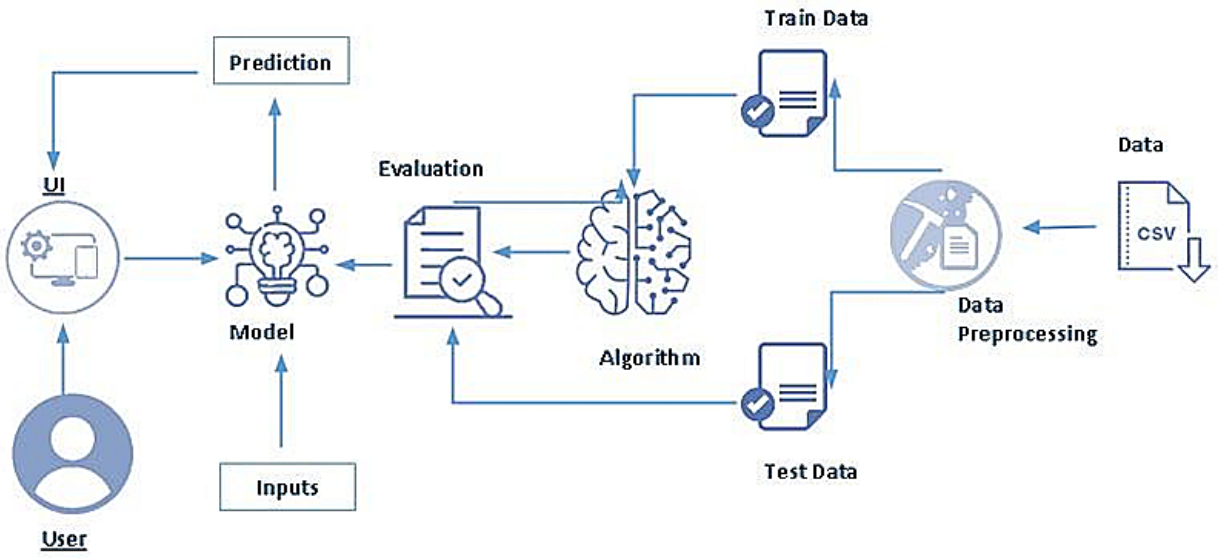
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

1. The user can register in website by using Email.
2. The user can Login by using Email and password as Registered in the respective website.
3. The user will provide personal and financial details.
4. User should upload the scanned documents.
5. Then it will goes to approval process.
6. Finally they will get loan closure certificate.



Data Flow Diagram Level 1

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

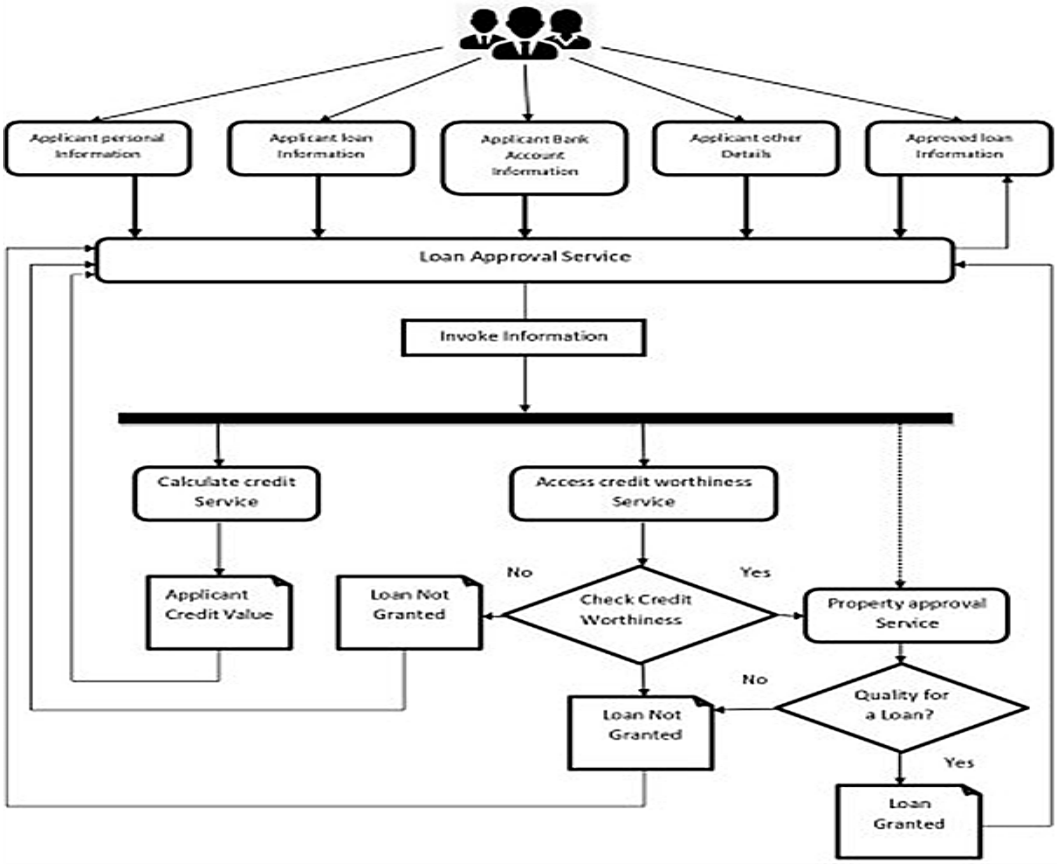
Solution Architecture:

a. The essential objective in the financial business is to put their assets in safe hands. Thus, the framework needs to check the archives actually and ought to guarantee that as it were able individuals get the advance.
b. The model ought to be prepared to create results with palatable precision, later which it produces precise outcomes regarding whether a borrower- ought to be loaned cash or not with next to no drawn-out manual work.

c. The clients can obtain the outcomes in the solace of their home.

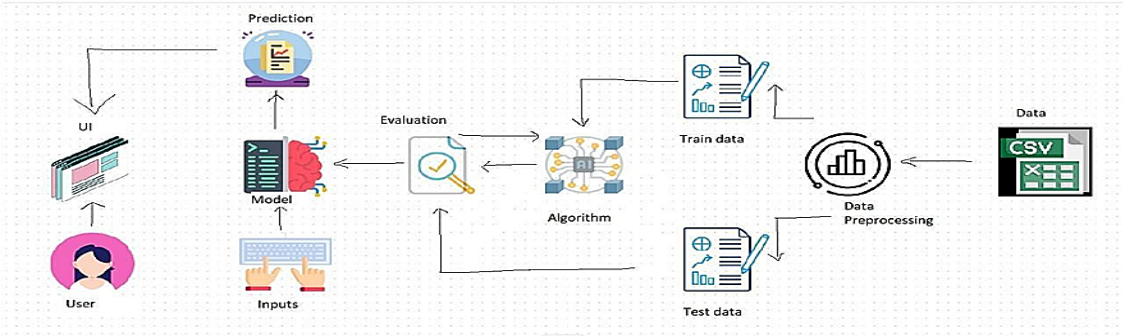
d. The framework ought to diminish chance to both the bank and the client.

e. The model can expect results and is rapidly versatile to an extensive variety of inputs. Likewise, this system saves the financial business and its staff a huge measure of time.



Solution Architecture Diagram

Technical Architecture:



5.3 User Stories

Sprint	User Story Number	User Story / Task
Sprint-1	USN-1	Download the Dataset
Sprint-1	USN-2	Visualize the Dataset
Sprint-1	USN-3	Pre-Process the Dataset
Sprint-1	USN-4	KNN Model
Sprint-2	USN-5	Decision Tree Model
Sprint-2	USN-6	Random Forest Model
Sprint-2	USN-7	XgBoost Model
Sprint-2	USN-8	Fine-Tune the Model
Sprint-2	USN-9	Evaluate and select the best Model
Sprint-3	USN-10	Model Integration with Flask
Sprint-3	USN-11	User should be able to see the interface.
Sprint-3	USN-12	Select the Type of Loan

Sprint-3	USN-13	Fill the required Applications
Sprint-4	USN-14	Register on the IBM Cloud
Sprint-4	USN-15	Train and test the ML Model
Sprint-4	USN-16	Deploy the website on IBM Cloud

6. PROJECT PLANNING AND SCHEDULING

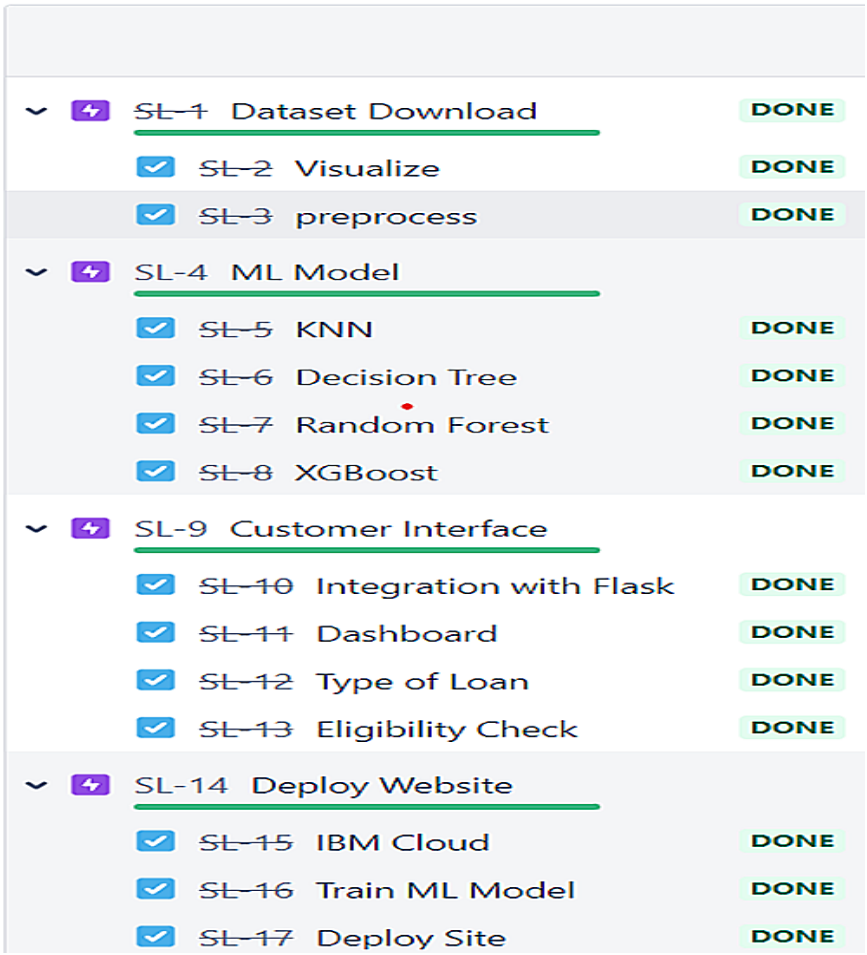
6.1 Sprint Planning and Estimation

Sprint Number	Planned Tasks
Sprint 1	● Downloading the Dataset ● Visualizing the Dataset ● Pre-Processing the Dataset
Sprint 2	● KNN Model ● Decision Tree Model ● Random Forest ● XGboost Algorithm ● Fine-tuning the Model ● Evaluating the best Model
Sprint 3	● Model Integration with Flask ● Interface Development ● Loan types Formulated ● Prediction mechanism finalized
Sprint 4	● Register on IBM Cloud ● Train the Model ● Deploy the Model and perform prediction operations on IBM Cloud

6.2 Sprint Delivery Schedule

Sprint Number	Start Date	End Date
Sprint 1	24 Oct 2022	29 Oct 2022
Sprint 2	31 Oct 2022	05 Nov 2022
Sprint 3	07 Nov 2022	12 Nov 2022
Sprint 4	14 Nov 2022	19 Nov 2022

6.3 Reports From Jira



7. CODING & SOLUTIONING

7.1 Feature 1

Home Interface

Users are presented with an introductory home interface Webpage which has direct links to all our associated web pages.

```
        <!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 -PNT2022TMID52913</h2><br>
```

```
        <h3>Team members</h3>
        <p>HARSHAVARDHAN G V</p>
        <p>MOHIT KHANNA</p>
        <p>JEBAROSHAN</p>
        <p>HARISH K</p>


      </div>

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

7.2 Feature 2

Separate Logins for potential Loan lenders and Bank Employees with their own registered IDs and Passwords

Bank Login:

```
<!DOCTYPE html>
<html>
<head>
  <title>LogIn Page</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
</head>
<style>
   .group{
      padding-top: 100px;
   }
</style>
<body>



<div class="container">
  <div class="row">
    <div style="width: 40%; margin: 25px auto;">
```

```
        <div class="group">
   <h3 style="text-align: center;">Bank Login Page</h3>
   <form method="POST" action="bank1.php">
    <div class="form-group">
     <label>Bank user ID:</label><input type="text" name="BankUserName" class="form-control" autofocus
placeholder="Enter the Bank User ID" required>
    </div>
    <div class="form-group">
     <label>Bank Email ID:</label><input type="email" name="bankemail" class="form-control" autofocus
placeholder="Enter the Bank Email ID" required>
    </div>
        <div class="form-group">
     <label>Password:</label><input type="Password" name="Password" class="form-control" autofocus
placeholder="Password" required>
    </div>

    <label>Enter Captcha:</label>
    <div class="form-row">
     <div class="form-group col-md-6">
      <input type="text" class="form-control" read-only id="capt" required>
     </div>
     <div class="form-group col-md-6">
      <input type="text" class="form-control" id="textinput" required>
     </div>
    </div>

  <div class="form-group">
     <button onclick="validcap()" name="Submit" class="btn btn-lg btn-success btn-block">Submit</button>
    </div>
   </form>
   <h6>Captcha not visible <img src="refresh.png" width="40px" onclick="cap()"></h6>

  </div>
  </div>
</div>
</div>
<script type="text/javascript">
 function cap(){
  var alpha = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'
```

```
,'W','X','Y','Z','1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i',
'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z', '!','@','#','$','%','^','&','*','+'];
var a = alpha[Math.floor(Math.random()*71)];
var b = alpha[Math.floor(Math.random()*71)];
var c = alpha[Math.floor(Math.random()*71)];
var d = alpha[Math.floor(Math.random()*71)];
var e = alpha[Math.floor(Math.random()*71)];
var f = alpha[Math.floor(Math.random()*71)];

var final = a+b+c+d+e+f;
document.getElementById("capt").value=final;
}
function validcap(){
var stg1 = document.getElementById('capt').value;
var stg2 = document.getElementById('textinput').value;
if(stg1==stg2){
alert("Form is validated Successfully");
return true;
}else{
alert("Please enter a valid captcha");
return false;
}
}
</script>
</body>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"></script>
</html>
```

Customer Login:

```
<!DOCTYPE html>
<html>
<head>
```

```
  <title>LogIn Page</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
</head>
<style>
    .group{
        padding-top: 100px;
    }
</style>
<body>
    <div class="container">
  <div class="row">
    <div style="width: 40%; margin: 25px auto;">
            <div class="group">
    <h3 style="text-align: center;">Login Page</h3>
    <form action="login1.php" method="POST" >
      <div class="form-group">
        <label>UserName:</label><input type="text" name="username" class="form-control" autofocus
placeholder="Enter your username or gmail ID" required="">
      </div>
      <div class="form-group">
        <label>Password:</label><input type="Password" name="password" class="form-control" autofocus
placeholder="Password" required="">
      </div>

      <label>Enter Captcha:</label>
      <div class="form-row">
        <div class="form-group col-md-6">
          <input type="text" class="form-control" readonly id="capt" required="">
        </div>
        <div class="form-group col-md-6">
          <input type="text" class="form-control" id="textinput" required="">
        </div>
      </div>

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

```
  </form>
  <h6>Captcha not visible <img src="refresh.png" width="40px" onclick="cap()"></h6>
  <p>New Here?<a href="register.html">Register</a> </p>
 </div>
 </div>
</div>
</div>
<script type="text/javascript">
 function cap(){
  var alpha = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'
            ,'W','X','Y','Z','1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i',
            'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z', 'l','@','#','$','%','^','&','*','+'];
            var a = alpha[Math.floor(Math.random()*71)];
            var b = alpha[Math.floor(Math.random()*71)];



           var c = alpha[Math.floor(Math.random()*71)];
            var d = alpha[Math.floor(Math.random()*71)];
            var e = alpha[Math.floor(Math.random()*71)];
            var f = alpha[Math.floor(Math.random()*71)];

            var final = a+b+c+d+e+f;
            document.getElementById("capt").value=final;
           }
           function validcap(){
            var stg1 = document.getElementById('capt').value;
            var stg2 = document.getElementById('textinput').value;
            if(stg1==stg2){
             // alert("Form is validated Successfully");
              return true;
            }else{
              alert("Please enter a valid captcha");
              return false;
            }
           }
</script>
</body>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
```

```
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"></script>
</html>
```

8. Testing

8.1 User Acceptance Testing and Test-Cases

Sprint 1 test Cases:

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Expected Result	Actual Result	Status	TC for Automation(Y/N)
LoginPage_TC_OO 1	Functional	Predict page	Verify wheather loan details entered by the user is correctly received	1.Enter URL and click go 2.Click on predict button 3.Enter the details shown on the page 4.Click submit	Prediction result page should appear	Working as expected	Pass	N

Sprint 2 Test Cases:

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)
LoginPage_TC_OO 1	UI	Home Page	Verify wheather user is able to see the landing page	1.Enter URL and click go	Prediction result page should appear	Working as expected	Pass		
LoginPage_TC_OO 2	Functional	Predict Page	Verify wheather loan details entered by the user is correctly received	1.Enter URL and click go 2.Click on predict button 3.Enter the details shown on the page 4.Click submit	Application should show below UI elements: • applicant_income • coapplicant_income • credit_history • dependents • education • loan_amount • loan_amt_term • married • property_area	Working as expected	Pass	Steps are clear to follow	N

Sprint 3 Test Cases:

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)
LoginPage_TC_OO 1	UI	Home Page	Verify wheather user is able to see the landing page	1.Enter URL and click go	Prediction result page should appear	Working as expected	Pass		
LoginPage_TC_OO 2	Functional	Predict Page	Verify wheather loan details entered by the user is correctly received	1.Enter URL and click go 2.Click on predict button 3.Enter the details shown on the page 4.Click submit	Application should show below UI elements: • applicant_income • coapplicant_income • credit_history • dependents • education • loan_amount • loan_amt_term • married • property_area	Working as expected	Pass	Steps are clear to follow	N
naturesPage_TC_OO	Functional	Features page	Verify whether the use can see features page at the click of the button	1.Enter URL and click go 2.Click on predict button 3.Enter the details shown on the page 4.Click Go 5. View Features Page	1. User should see Features page 2. User should be able to go to our Github page when they click the "GITHUB" button 3. User should be able to navigate to the different sections of the page, including features, about and predict sections	Working as expected	Pass		N

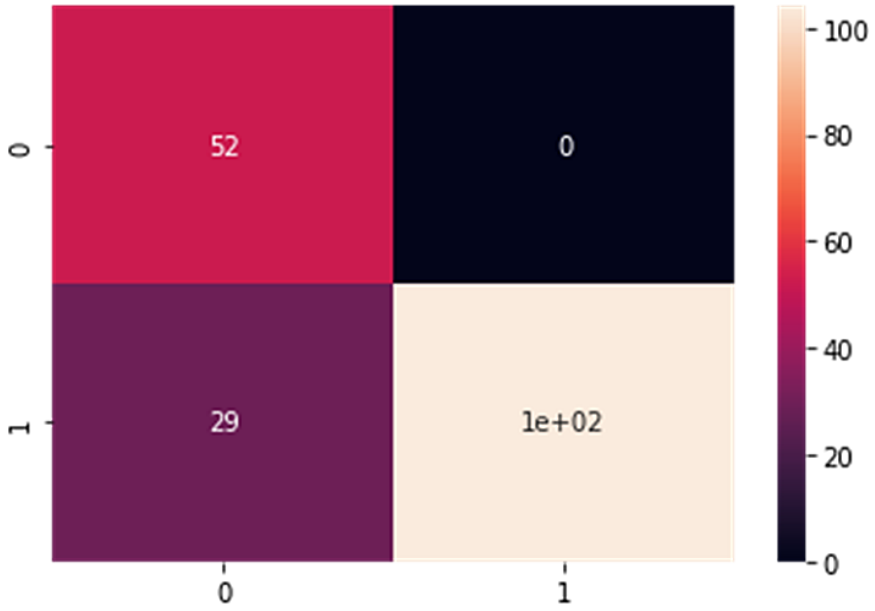
Sprint 4 test cases:

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)
LoginPage_TC_OO 1	UI	Home Page	Verify wheather user is able to see the landing page	1.Enter URL and click go	Prediction result page should appear	Working as expected	Pass		
PredictPage_TC_O O2	Functional	Predict Page	Verify wheather loan details entered by the user is correctly received	1.Enter URL and click go 2.Click on predict button 3.Enter the details shown on the page 4.Click submit	Application should show below UI elements: • applicant_income • coapplicant_income • credit_history • dependents • education • loan_amount • loan_amt_term • married • property_area	Working as expected	Pass	Steps are clear to follow	N
eaturesPage_TC_OO	Functional	Features page	Verify whether the use can see features page at the click of the button	1.Enter URL and click go 2.Click on predict button 3.Enter the details shown on the page 4.Click Go 5. View Features Page	1. User should see Features page 2. User should be able to go to our Github page when they click the "GITHUB" button 3. User should be able to navigate to the different sections of the page, including features, about and predict sections	Working as expected	Pass		N
Predictpage_TC_O O_02	Functional	Predict Page	Verify whether the user can go to predict page when they click predict button	1.Enter URL and click go 2.Click on predict button 3.Enter the details shown on the page 4.Click Go 5. View Features Page 6. Click Predict button 7. View predict page	1. User should be taken to predict page when they click predict button	Working as expected	Pass		N
Predictpage_TC_O O2	Functional	Predict Page	Verify wheather loan details entered by the user is correctly received	1.Enter URL and click go 2.Click on predict button 3.Enter the details shown on the page 4.Click submit	Application should show below UI elements: • applicant_income • coapplicant_income • credit_history • dependents • education • loan_amount • loan_amt_term • married • property_area • self_employed • gender	Working as expected	Pass	Steps are clear to follow	N
Registerpage__TC_ _001	Functional	Register page	Verify whether use can see register page and enter details	1. Enter the URL and click register 2. User should enter first name, last name and email address			Pass		N
SubmitPage__TC__ 01	Functional	Submit page	Verify whether use is able to see the submit page which displays the result of the prediction	1. Enter the URL and click register 2. User should enter first name, last name and email address 3. User should go to features page 4. User should go to predict section and click PREDICT button 5. User should enter details in predict page 6. User should click SUBMIT button	User should see results of prediction on submit page	Working as expected	Pass		N

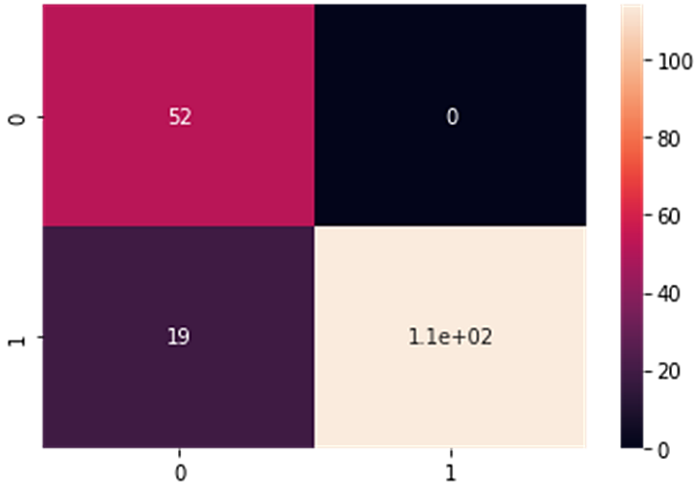
9.RESULTS

9.1 Performance Metrics

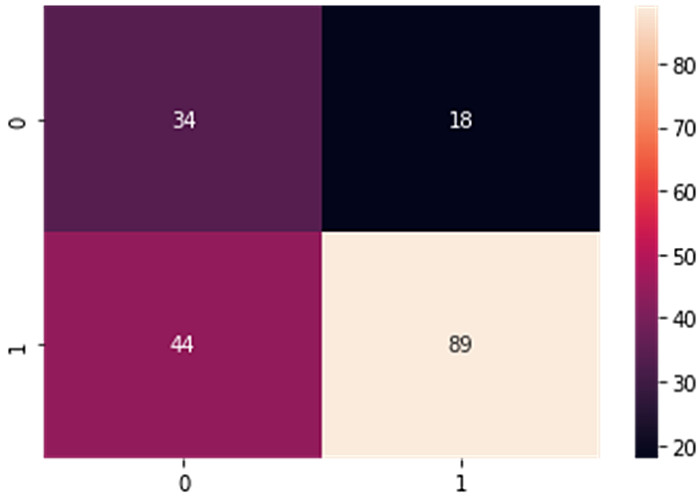
Decision Tree:



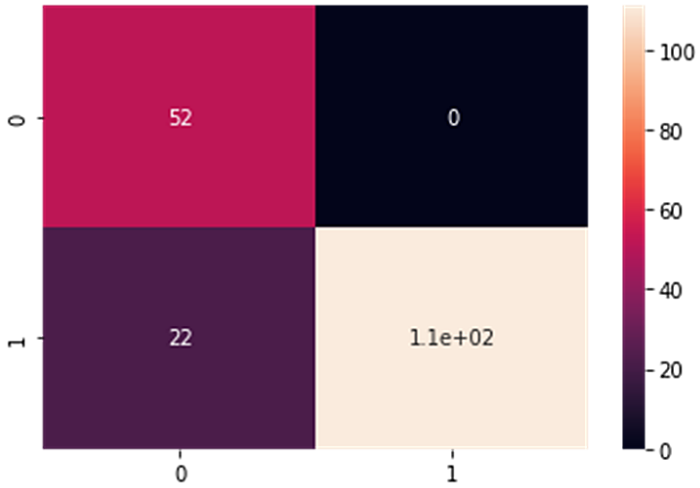
Random Forest:



KNN:



XgBoost:



10. ADVANTAGES & DISADVANTAGES

***Advantages*:**

i. With our solution, the prospective customers can be better analysed.
ii. Funds allocated to be lent by the Bank would find themselves with a trustworthy customer.
iii. The process of judging a candidate is made exponentially easier.
iv. This system not only benefits the Banks by finding themselves reliable customers but also facilitates the latter with timely Loans
v. Sets a standard for Financial History of an Individual.
vi. Our chosen model – Random Forest is robust to rarely occurring outliers.
vii. Has a Low risk of Over fitting.
viii. Works very well on large datasets

Disadvantages:

i. Given the small dataset, the model may fail to predict the right decisions for customers with parameters of high variance.
ii. For more accurate and real time results, we would need a dataset which contains millions of data points.

11. CONCLUSION

a. After the Submission of test data, our model returned an accuracy of 92%.
b. Feature engineering assisted in improving the accuracy of our model.
c. Random forest worked better than all other decision tree, regression models.

12. FUTURE SCOPE

- Higher accuracy can be exploited by increasing the size of the dataset by generating synthetic data which can be obtained by scaling the applicant income , co-applicant income and loan amount columns and adding it to the existing dataset and testing our model on the new dataset. Although, synthetic data generation is a very tedious and difficult process, it can help achieve a better accuracy.
- User Interface can be updated with more features.
- Our Website can be integrated with social media platforms to give it a higher reach.
- Security of our customers accounts can be further increased by using 2-way logins.

13. Appendix

Source Code

Model Selection and Execution

```
import numpy as np
import pandas as pd
import pickle
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import sklearn
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier,RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RandomizedSearchCV
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier
import imblearn
from imblearn.under_sampling import RandomUnderSampler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import scale
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import
accuracy_score,classification_report,confusion_matrix,f1_score
```

```
df=pd.read_csv('Loan_dataset.csv')
df
df=df.drop(columns=["Loan_ID"],axis=1)
sns.distplot(df.ApplicantIncome)
plt.pie(df.Property_Area.value_counts(),[0,0,0],labels=['Semi urban','Urban','Rural'])
plt.pie(df.Education.value_counts(),[0,0],labels=['Graduate','Not Graduate'])
plt.scatter(df.ApplicantIncome,df.LoanAmount)
sns.heatmap(df.corr(),annot=True)
plt.plot(df.LoanAmount,df.ApplicantIncome,df.CoapplicantIncome)
plt.plot(df.Loan_Amount_Term,df.ApplicantIncome,df.CoapplicantIncome)
x=df.iloc[:,:-1]
y=df.Loan_Status
x_scale=pd.DataFrame(scale(x),columns=x.columns)
x_scale.head()
sns.countplot(df.Loan_Status)
rus=RandomUnderSampler(sampling_strategy=1)
x_res,y_res=rus.fit_resample(x,y)
ax=y_res.value_counts().plot.pie(autopct='%.2f')
_=ax.set_title("under-sampling")
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=10)
dmodel=DecisionTreeClassifier(random_state=100)
dmodel.fit(x_res,y_res)
ypredd=dmodel.predict(xtest)
Rmodel=RandomForestClassifier(n_estimators=100)
Rmodel.fit(x_res,y_res)
ypredR=Rmodel.predict(xtest)
kmodel=KNeighborsClassifier()
kmodel.fit(x_res,y_res)
ypredk=kmodel.predict(xtest)
xmodel=XGBClassifier(eval_metric='mlogloss',n_estimators=100,random_state=100)
xmodel.fit(x_res,y_res)
ypredx=xmodel.predict(xtest)
print("Decision Tree Model Testing Accuracy")
print(accuracy_score(ytest,ypredd))
from sklearn.metrics import confusion_matrix
cf = confusion_matrix(ytest, ypredd)
import seaborn as sns
sns.heatmap(cf, annot=True)
print("Random Forest Model Testing Accuracy")
print(accuracy_score(ytest,ypredR))
from sklearn.metrics import confusion_matrix
```

```
cf = confusion_matrix(ytest, ypredR)
import seaborn as sns
sns.heatmap(cf, annot=True)
print("KNN Model Testing Accuracy")
print(accuracy_score(ytest,ypredk))
from sklearn.metrics import confusion_matrix
cf = confusion_matrix(ytest, ypredk)
import seaborn as sns
sns.heatmap(cf, annot=True)
print("Xgboost Model Testing Accuracy")
print(accuracy_score(ytest,ypredx))
from sklearn.metrics import confusion_matrix
cf = confusion_matrix(ytest, ypredx)
import seaborn as sns
sns.heatmap(cf, annot=True)
print(classification_report(ytest,ypredR))
pickle.dump(Rmodel,open('Rmodel.pkl','wb'))
pickle.dump(x_scale,open('scale.pkl','wb'))
import joblib
joblib.dump(xmodel,r"C:\Users\HP\Desktop\Rmodel.joblib")
```

GitHub

https://github.com/IBM-EPBL/IBM-Project-19400-1659697313

Project Demo Link

https://photos.app.goo.gl/wHo5apw9ZeBSrxkc9