TEAM ID: PNT2022TMID10667

Smart Lender - Applicant Credibility Prediction for Loan Approval

Team Leader

Manjula.S

Team Members

Monica Jain.D

Nashiha Sulthana.G

Nishanthi.S

Nishi Gilbert.G

CHAPTER -1
INTRODUCTION

1. INTRODUCTION

Banks make the majority of their income through loans. Loan approval is a critical step for financial institutions. It is extremely difficult to estimate the probability of loan repayment by customers due to a growing incidence of loan defaults, and banking authorities are finding it increasingly difficult to appropriately access loan requests and address the dangers of individuals defaulting on loans. Many scholars have focused on loan approval system prediction in recent years. Machine learning is a powerful tool for predicting outcomes from massive amounts of data. A large amount of a bank's assets are directly derived from interest earned on loans made. Lending loans has significant risks, including the borrower's inability to repay the loan within the time frame specified. It is known as "credit risk." The worthiness of an applicant for loan acceptance or rejection was determined by a numerical score known as a "credit score." As a result, the use of various Machine Learning approaches that properly identify people to lend to and assist banks in identifying loan defaulters for much-reduced credit risk.To anticipate client loan acceptance, four algorithms are used: the Random Forest method, the Decision Tree algorithm, the KNN algorithm, and the XGBoost algorithm. All four methods will be run on the same dataset to select the approach with the highest accuracy for deploying the model. We will now create a bank loan prediction system listing machine learning techniques, so that the system will automatically identify the most qualified people to authorize the loan.

1.1 Project Overview

Loan Prediction is extremely beneficial to both bank employees and applicants. The goal is to give a quick, straightforward approach to choose qualified applicants. They have a presence in all urban, suburban, and rural regions. After that firm or bank checks the consumer's loan eligibility,the customer applies for a loan or not.Based on the criteria loan approval or rejection will be provided to the applicants.

1.3 Purpose

One of the most significant financial instruments is the loan. Every bank is attempting to come up with successful marketing techniques to get clients to apply for loans. Some consumers, meanwhile, behave badly once their applications are accepted. The solution is for banks to develop techniques for anticipating client behaviour. The banking industry frequently uses machine learning algorithms, which perform well for this purpose. Here, I'll be utilising machine learning models to anticipate lending behaviour. Finding the most appropriate model and relevant attribute identification.

CHAPTER-2
LITERATURE SURVEY

2.1 Existing System

Anomaly detection is based on individual behavior profiling and works by identifying any variation from the norm. It has three drawbacks when used to identify online banking fraud. First, for an individual, previous behavior data are frequently insufficient for profiling his/her behavior pattern. Second, due to the heterogeneous nature of transaction data, there is no standard handling of multiple attribute values, which might be a barrier to the model's development and future use. Third, the transaction data are extremely skewed, making efficient use of the label information difficult. Anomaly detection is frequently plagued by weak generalization and a high false alarm rate.

2.2 References

1.Arun Kumar, Ishan Garg and Sanmeer Kaur, Loan Approval Prediction based on Machine Learning Approach.

2. Mohamed El Mohadab, Belaid Bouikhalene and Said Safi, "Predicting rank for scientific research papers using supervised learning", Applied Computing and Informatics, vol. 15, pp. 182-190, 2019.

3. K. Hanumantha Rao, G. Srinivas, A. Damodhar and M. Vikas Krishna, "Implementation of Anomaly Detection Technique Using Machine Learning Algorithms", International Journal of Computer Science and Telecommunications, vol. 2, no. 3, June 2011.

4. J.R. Quinlan, Induction of decision trees, Machine learning Springer, vol. 1, no. 1, pp. 81-106, 1086.

5. S.S. Keerthi and E.G. Gilbert, Convergence of a generalize SMO algorithm for SVM classifier design, Machine Learning, Springer, vol. 46, no. 1, pp. 351-360, 2002.

6. J.M. Chambers, "Computational methods for data analysis" in Applied Statistics, Wiley, vol. 1, no. 2, pp. 1-10.

7. Kumar Arun, Garg Ishan, Kaur Sanmeet, May-Jun. 2016. Loan Approval Prediction based on Machine Learning Approach, IOSR Journal of Computer Engineering (IOSR-JCE)
8. Wei Li, Shuai Ding, Yi Chen, and Shanlin Yang, Heterogeneous Ensemble for Default Prediction of Peer-to-Peer Lending in China, Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei University of Technology, Hefei 2009, China

2.3 Problem Statement Definition

An Banks provides various forms of loans like as housing loans, personal loans,
business loans, and so on throughout the country. These businesses can be found in rural, semi-urban, and urban locations. After a consumer applies for a loan, these firms determine whether or not the customer is eligible for the loan.Using a machine learning technique. As a result, the consumer will complete out an online loan application form. This form includes information such as Sex, Applicant Income, Loan Amount, Details of Dependents, Loan Amount, Loan Amount Team, Applicant Credit History and ers.oth

LITERATURE SURVEY

TITLE 1: Improving Information Quality in Loan Approval Processes for Fair Lending and Fair Pricing

AUTHOR: M. Cary Collins

DESCRIPTION: Bank data management on loan approval processes has great room for improvements of information quality and data problems prevention especially with regards to fair lending and fair pricing practices. They first reviewed briefly typical data collection protocols deployed at many financial institutions for loan approval and loan pricing. Federal regulations mandate portions of these data protocols. While discussing the data capture and analysis for fair lending, they illustrated some initial key steps currently needed for improving information quality to all parties involved.

TITLE 2: Loan Credibility Prediction System Based on Decision Tree Algorithm

AUTHOR: Sivasree M S, Rekha Sunny T

DESCRIPTION: Data mining techniques are becoming very popular nowadays because of the wide availability of huge quantity of data and the need for transforming such data into knowledge. Data mining techniques are implemented in various domains such as retail industry, biological data analysis, intrusion detection, telecommunication industry and other scientific applications. Techniques of data mining are also be used in the banking industry which help them compete in the market well equipped. In this paper, they introduced a prediction model for the bankers that will help them predict the credible customers who have applied for a loan. Decision Tree Algorithm is being applied to predict the attributes relevant for credibility. A prototype of the model has been described in this paper which can be used by the organizations for making the right decisions to approve or reject the loan request from the customers.

TITLE 3: Loan Approval Prediction based on Machine Learning Approach

AUTHOR: Kumar Arun, Garg Ishan, Kaur Sanmeet

DESCRIPTION: With the enhancement in the banking sector, lots of people apply for bank loans but the bank has its limited assets which it grants to only limited people , so finding out to whom the loan can be granted is a typical process for the banks. So, in this paper , they tried to reduce this risk by selecting the safe person so as to save lots of bank efforts and asselt. The main goal of this paper is to predict if loan assignment to a specific person will be safe or not. This paper has into four sections (i) Collection of data (ii) Comparing the machine learning models on collected data (iii) Training the system on most promising.

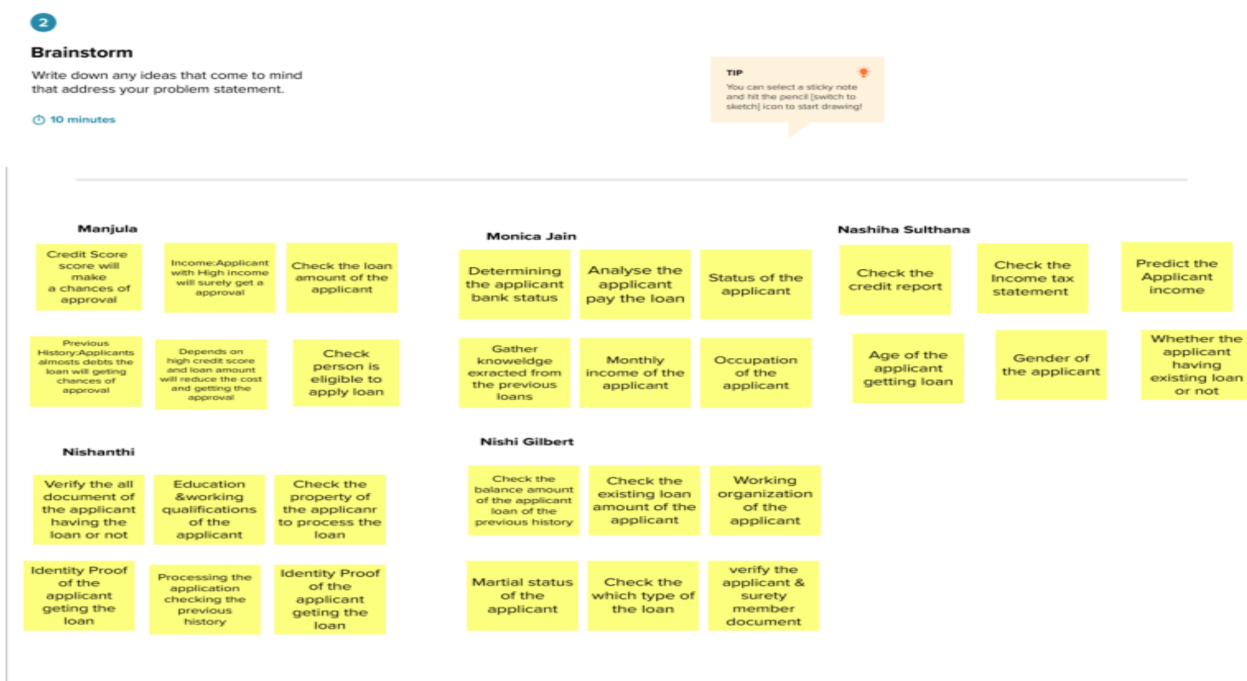
CHAPTER-3
IDEATION AND PROPOSED SOLUTION

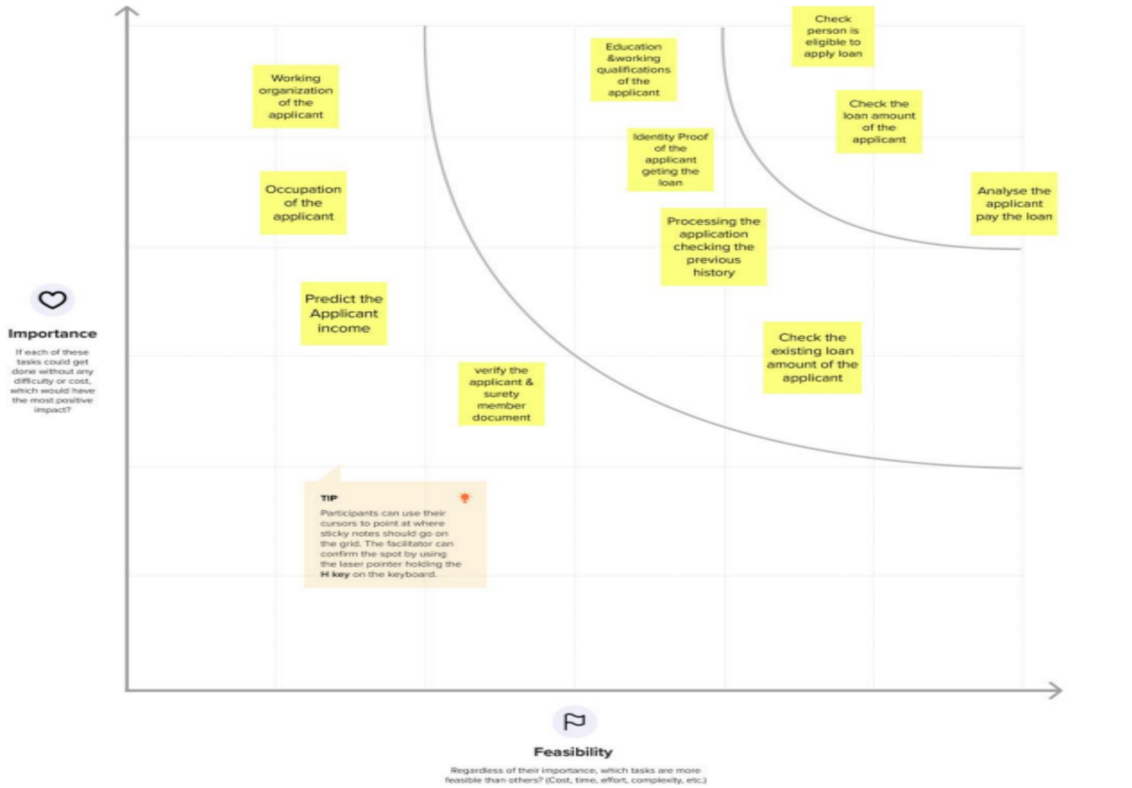
3.1 Empathy Map Canvas

The Empathy Map Canvas helps teams develop deep, shared understanding and empathy for other people. People use it to help them improve customer experience, to navigate organizational politics, to design better work environments, and a host of other things.

3.2 Ideation And Brainstroming

Brainstroming is the term which breaks any idea that come to our mind which addresses our problem statement.It deals to discuss the ides to the team members and gather the ideas from them.Each team members reveals their ideas about bank loan prediction such as check the loan amount of the applicant,Occupation of the applicant,Gender of the applicant,Marital status of the applicant,Identity proof of the applicant.It involves to collect group ideas,Ideas prioritization.





3.3 Proposed Solution

In the suggested system, we aggregate datasets from several sources to generate a generalized dataset and employ four machine learning algorithms on the same dataset, including Random forest, kNN, Decision tree, and XGBoost. The dataset we gathered for predicting supplied data is divided into 7:3 training and test sets. The data model generated using machine learning algorithms is applied to the training set, and the test set prediction is done using the method with the highest performance based on the maximum test result from the four algorithms.

3.4 Problem Solution Fit

The problem-Solution Fit basically implies that you identified a problem with your consumer and that the solution you devised genuinely solves the problem.Problem solution fit deals to have customer segments,Jobs to be done/Problems,Triggers, Customer Constraints,Problem root cause,General Solutions,Behavior and Available solutions,

Identify Strong TR & EM	3. TRIGGERS TR The applicant who l. nding th. loan based on the CIBIL credit score to get the loan approval based on the CIBIL score is low applicant will not get the loan 4. EMOTIONS BEFORE / AFTER EM The applicant will feel frustrated or tensioned whether the loan application will get approved to get the loan for the urgent	10. YOUR SOLUTION SL Our solution for the project is to get a detailed overview to apply for the loan whether the applicant is eligible or not it helps the applicant easily apply and get the loan in the bank without wasting the time it helps to reduce the frustration or tension of the applicant	8. CHANNELS of BEHAVIOUR CH **8.1 ONLINE** The online application for the customer is easy to apply for loans based on the each time the customer can use the same id to process the loan whenever they want. **8.2 OFFLINE** The loan approval process traditional way it takes lot of steps and each step from filling the application and verifying need to spend lot of time.	Identify Strong TR & EM

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 Functional Requirement

A functional requirement document specifies the functionality of a system or one of its subsystems. It also relies on the type of programme, expected users, and the system on which the software is run.Functional user requirements may be high-level declarations of what the system should perform, but functional system requirements should also specify the system services in depth.

Functional Requirements are:

- ◊ User Click the Predict Button
- ◊ User Fill the Application
- ◊ Message generated

User Fill the Application

User will follow the steps to apply the bank loan

Message generated

If the loan approvedor rejected the message will begenerated

4.2 Non-Funcctional Rerquirements

A non-functional requirement (NFR) is one that defines criteria for judging the functioning of a system rather than particular behaviors. They differ from functional requirements, which describe precise behavior or functions. The system design includes a thorough plan for accomplishing functional requirements. Because non-functional needs are frequently architecturally significant, the plan for accomplishing them is outlined in the system architecture.

Non-functional Requirements are:

- ◊ Usability
- ◊ Security
- ◊ Reliability
- ◊ Performance
- ◊ Availability
- ◊ Scalability

Usability

The application of the user interfaceshould be simple and easy to use

Security

The data given by the user must be more secure would prevent from the unauthorized access to save the user detail to prevent from the attack

Reliability

The users can access the website without any Problem

Performance

The user should not be wait for more time duringthe registration or login or performing any other activity in the application shouldbe efficient

Availability

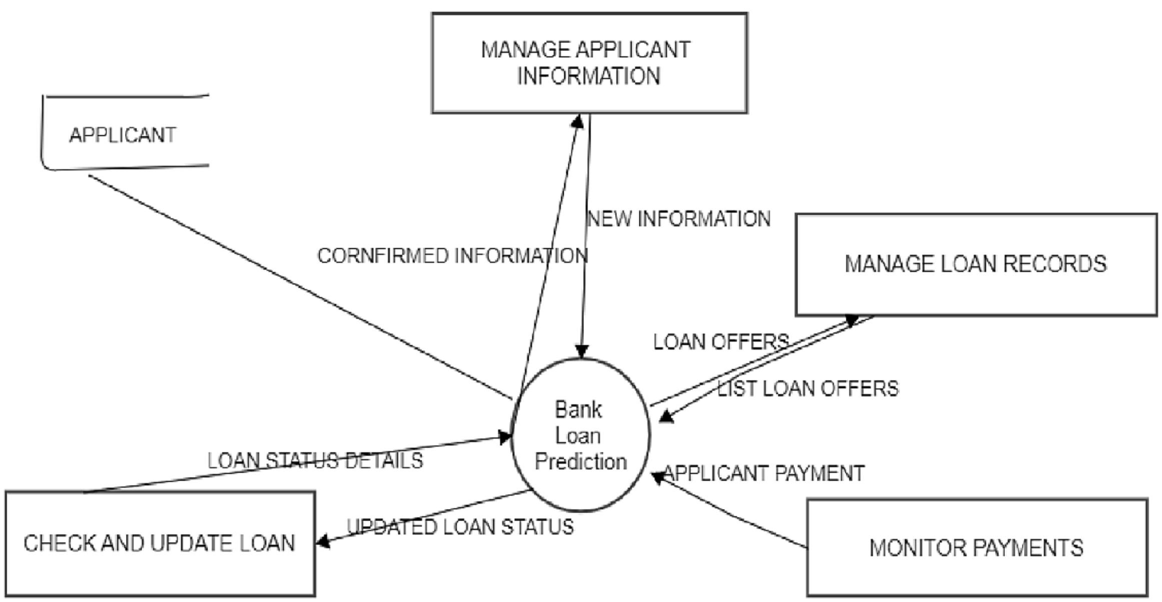
The user interfaceof the application interactive must be available in all the time when the user enter

CHAPTER-5

PROJECT DESIGN

5.1 Data Flow Diagram

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 systemrequirement graphically. It shows how data entersand leaves the system, what changes the information, and where data is stored.Geneerally it shows clear view of thethe system requirements.The bank loan prediction deals to Manage the loan records,Monitor payments,Manage Applicant Information,Check and update the loan.



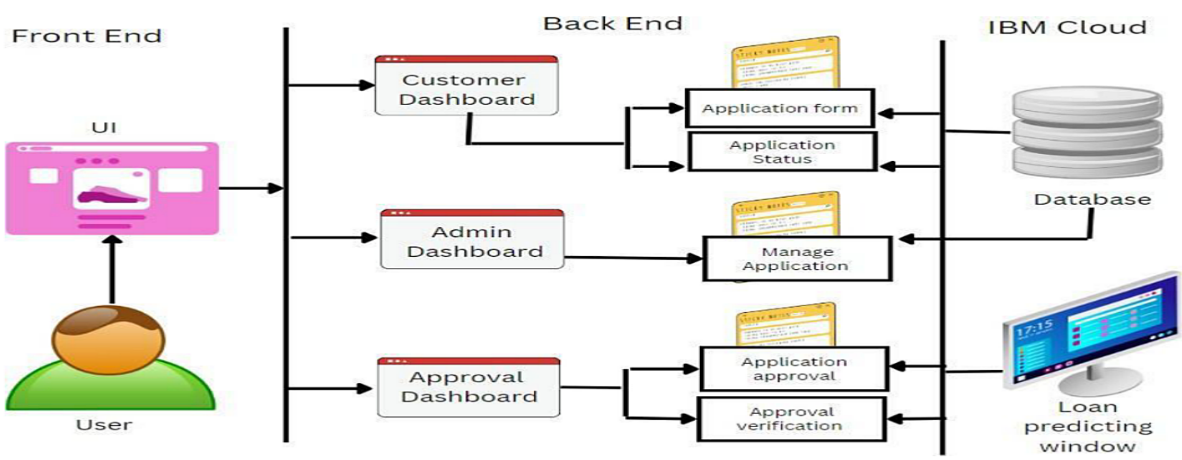
5.2 Solution And Technical Architecture

Solution Architecture

The process of designing solutions based on predetermined procedures, rules, and best practises with the goal of ensuring that the generated solution fits inside the corporate architecture in terms of information architecture, system portfolios, integration needs, and other factors. It may therefore be defined as a set of roles, procedures, and documentation aimed at addressing specific business goals, requirements, or challenges through the design and development of applications and information systems.

Technical Architecture:

In Front End ,it involves to create the User Interface by using HTML,CSS.In Back End, it contains Customer Dashboard,Admin Dashboard,Approval Dashboard,Customer Dashboard connect to the Application Form and Application Status.Admin Dashboard connects to the manage application.Approval Dashboard connects to the Application Approval and Application Verification.In IBM Cloud which which contains Database and Loan Predicting Windows.Database connects to the Customer Dashboard,Admin Dashboard and Loan Predicting Window connects to the Approval Dashboard.



5.3 User Stories

It handles tasks such as logging into the IBM account in Sprint 1. Download the dataset and visualise it. It performs activities such as pre-processing the dataset in sprint 2. Model the algorithm Decision Tree modelbuilding, Knn modelbuilding, Random Forest modelbuilding, Xgboost modelbuilding, and then assess the models. In Sprint 3, it completes tasks such as integrating the model with Flask and Finally it deploy our project on IBM Cloud.

- To design a dashboard similar to the User Interface, As a user, you may fill out the application and access it through the user interface.
- You can also fill out the application and check for available sources.
- It conducts tasks such as registering all team members to IBM Cloud in sprint 4.
- On the IBM Cloud, train the model.
- Install the website on IBM Cloud.
- The user applies for the loan (the loan can be checked by the user).

CHAPTER-6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning and Estimation

Sprint Planning

A sprint is essentially a predetermined length of time in which a development team needs to perform a specified amount of work. Sprints are often scheduled to last two weeks, although they can last as little as one week or as long as a month.The limited time span of a sprint forces developers to focus on sending out tiny, incremental improvements rather than massive, sweeping ones. As a result, significantly less debugging is necessary, and clients may have a more smooth experience with the programme.Generally it is used to create product backlog and contains sprint 1,2,3,4.Each performs some specific tasks to do so.

Sprint-1

In Sprint 1 which involve to create the functional requirement of User Registration and Login and Dataset. It performs the task such as To login the IBM account,Download the dataset and visualize the dataset.

Sprint-2

In sprint 2 ,which involves to create the functional requirent of use model. It performs the tasks such as Pre-process the dataset,Model the algorithm Decision Tree model building,Knn model Random Forest model and Xgboost model and then evaluate the models.

Sprint-3

In Sprint 3,which involve to create the functional requirement of Dashboard (User Interface).It perform the task such as To integrate the model with flask,To create a dashboard as like User

Interface,As a user able to fill the application and accessthe application on the user interface,To fill the application and check for theavailability sources.

Sprint- 4

In Sprint 4 ,which involve to create the functional requirement of Deployed the websitein IBM Cloud. It performs the task such as Register all the team membersto IBM Cloud,Train the model on IBM Cloud,Deploy the website on IBM Cloud,User apply for the loan (user can check the loan eligibility or not).

Sprint Estimation

Sprint Estimation is part of the Sprint Turnover process, which happens at the end of the last sprint but before the next sprint starts. It make sure to check our JIRA for issues that were thrown out of the previous sprint or issues that emerged during the sprint time. To ensure that this process runs well or not.

Velocity:

Calculate the team's average velocity (AV) periteration unit (story points per day) .

AV= Sprint Duration/Velocity=20/10=2

Sprint-1 = 20/9=2.2

Sprint-2 = 20/6=3.33

Sprint-3 =20/6 = 3..33

Sprint-3 = 19/6 = 3.16

Total Velocity= 79/2 7= 2.92

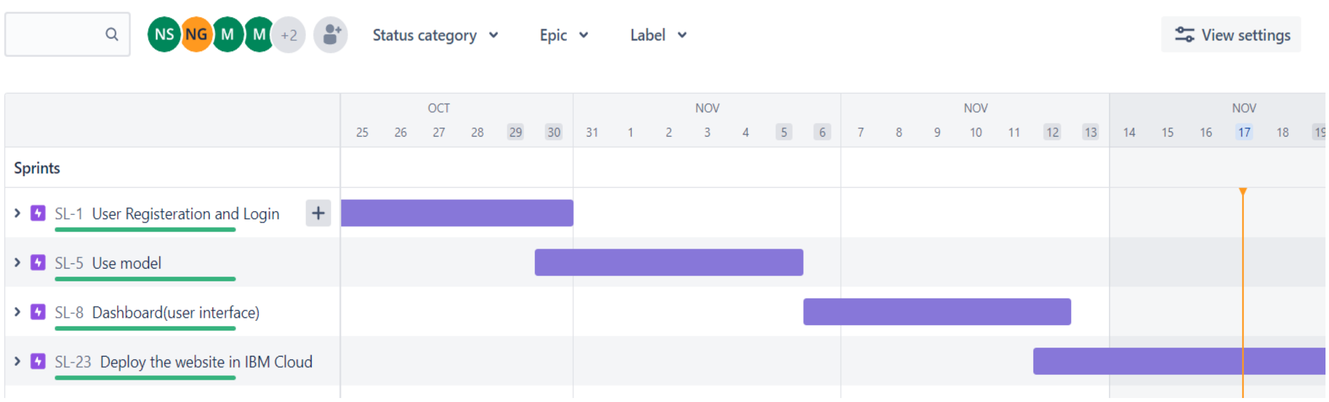
6.2 Sprint Delivery Schedule

In Sprint 1 ,which involve to create the functional requirement of User Registration and Login and Dataset . It performs the task such as To login the IBM account,Download the dataset and visualize the datase.Total duration required to complete sprint 1 was 9 days.

In sprint 2 ,which involves to create the functional requirent of use mode.It performs the tasks such as Pre-process the dataset,Model the algorithm Decision Tree modelbuilding,building of Knn model,Random Forest model,Decision Tree model,Xgboost model.and then evaluate the models.Total duration required to complete sprint 2 was 6 days.

In Sprint 3 ,which involve to create the functional requirement of Dashboard (User Interface).It performs the task such as To integrate the model with flask,To create a dashboard as like User Interface,As a user able to fill the application and access the application on the user interface,To fill the application and check for the availability sourcesToatal duration required to complete sprint 3 was 6 days.

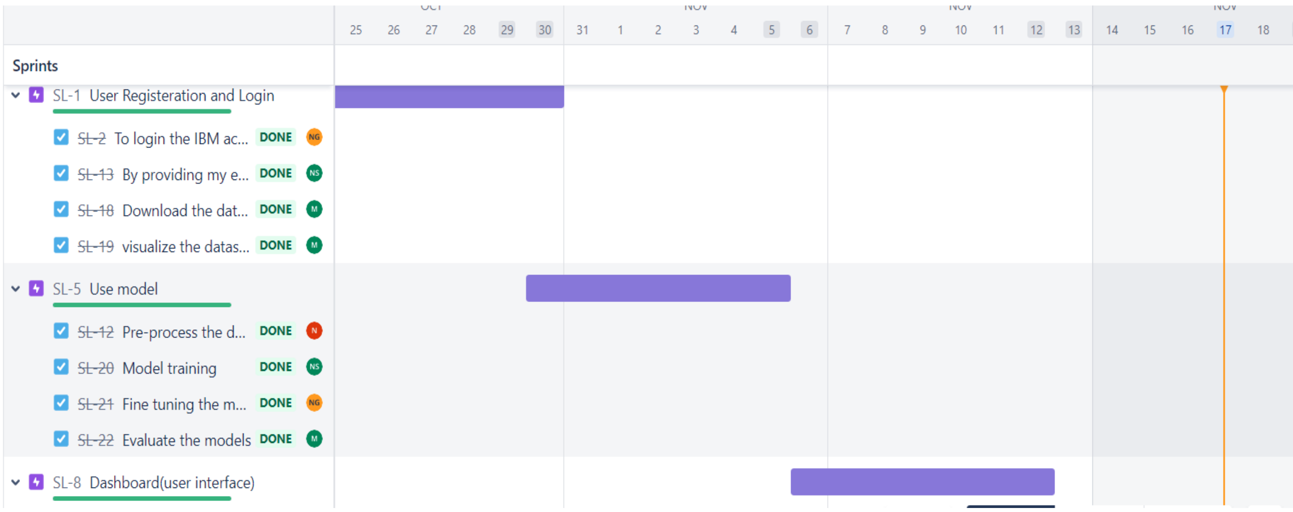
In Sprint 4 ,which involve to create the functional requirement of Register,Deployed the website in IBM Cloud.It performs the task such as Register all the team members to IBM Cloud,Train the model on IBM Cloud,Deploy the website on IBM Cloud,User apply for the loan (user can check theloan eligibility or not).Total duration required to complete sprint 4 was 6 days



6.3 Reports from JIRA

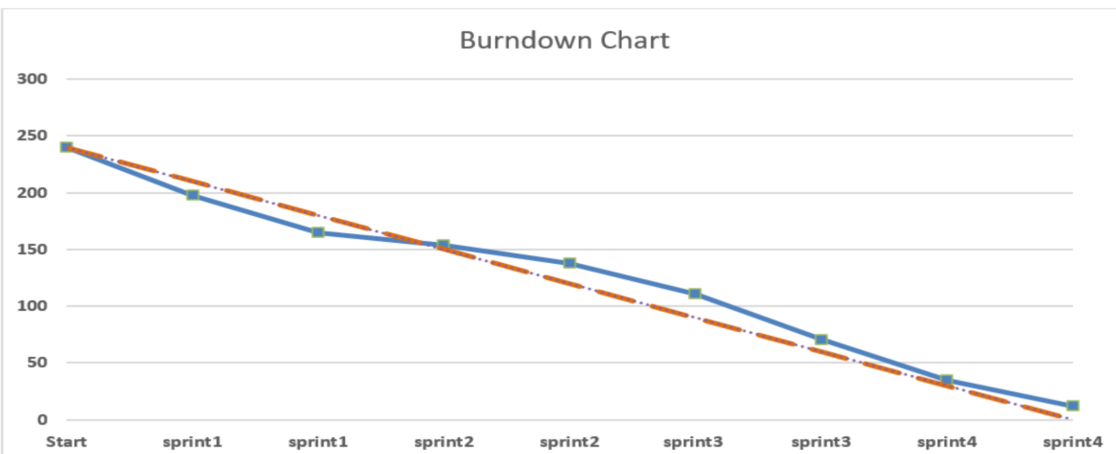
Jira

Jira Software is part of a solution family that assists teams of all sizes with job management. Jira was initially intended to be a bug and problem tracker. Jira, on the other hand, has evolved into a robust task management platform for various sorts of applications, ranging from requirements and test case management to agile development.



Burndown chart

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum.



CHAPTER-7

CODING & SOLUTIONING

FEATURE:

Feature Engineering

```
data.info()

data.isnull().sum()

data['Gender'].fillna(data['Gender'].mode()[0], inplace=True)

data['Married'].fillna(data['Married'].mode()[0], inplace=True)

data['Dependents'].fillna(data['Dependents'].mode()[0], inplace=True)

data['Self_Employed'].fillna(data['Self_Employed'].mode()[0], inplace=True)

data['LoanAmount'].fillna(data['LoanAmount'].mode()[0], inplace=True)

data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].mode()[0], inplace=True)

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

data.info()
```

Missing values in the column "Loan monthly payment" indicate that consumers did not make loan payments. In this case, instead of the mean or median, the missing values should be imputed with zero. The original data has a category target variable. It is divided into four categories, numbered A through D. To make the prediction, I must encode the category variable as 1 or 0, representing binary classes.By using the alforithm in machine learning is able to predict the loan approval.

Random Forest Algorithm

```
def randomForest(x_train,x_test,y_train,y_test):
    rf = RandomForestClassifier()
    rf.fit(x_train,y_train)
    pred_test = rf.predict(x_test)
    print('Confusion Matrix')
    print(confusion_matrix(y_test,pred_test))
    print('Classification Report')
    print(classification_report(y_test,pred_test))
    print('Score')
    print(rf.score(x_test,y_test))
```

Comparison of Randim Forest Algorithm Vs KNN Vs Decision Tree:

CHAPTER-8

TESTING

User Acceptance Testing

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

Defect AnalysiS: This report shows the number of resolved or closed bugs ateach severity level, and how they were resolved.

UAT Report Submission and usage of tools

	Total StoryPoints	**Duration**	**Sprint Start Date**	**Sprint End Date(Planned)**	**Story Point Completed(as an planned enddate)**	**Sprint Release Date(Actual)**
Sprint-1	10	9 Days	21 Oct2022	30 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	11	6 Days	06 Nov2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	19	6 Days	12 Nov 2022	19 Nov 2022	19	19 Nov 2022

CHAPTER-9
RESULTS

9.1 Performance Metrics

FIS Financial View, for example, compiles useful indicators and KPIs and then helps organize and explain them so you can react to trends, uncover performance possibilities, and monitor financial health. In bank laon prediction ,the upside of the framework is that we present the prerequisites as a calculation, and while confirming the subtleties, we decide the necessities that have beenendorsed and that meet the rerequisites of the unlawful client.

Decision Tree

Decision trees may be used to forecast numerical values (regression) as well as categorise data. The decision tree which hold ,

Performance metrices of decision tree:

Confusion Matrix

[[49 14]

[20 58]]

Classification Report

	precision	recall	f1-score	support
0	0.71	0.78	0.74	63
1	0.81	0.74	0.77	78
accuracy			0.76	141
macro avg	0.76	0.76	0.76	141
weighted avg	0.76	0.76	0.76	141

Score:0.7588652482269503

Random Forest

In a random forest, the machine learning algorithm predicts a value or category by combining the results from a number of decision trees.The random forest algorithm is a bagging technique extension that use both bagging and feature randomization to produce an uncorrelated forest of decision trees.

Preformance matrices of Random forest algorithm:

Confusion Matrix

[[44 18]

[6 72]]

Classification Report

	precision	recall	f1-score	support
0	0.88	0.71	0.79	62
1	0.80	0.92	0.86	78
accuracy			0.83	140
macro avg	0.84	0.82	0.82	140
weighted avg	0.84	0.83	0.83	140

Score

0.8285714285714286

K-Nearest Neighbors algorithm

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

Performance matrices of KNN algorithm:

Confusion Matrix

[[40 27]

[26 50]]

Classification Report

	precision	recall	f1-score	support
0	0.61	0.60	0.60	67
1	0.65	0.66	0.65	76
accuracy			0.63	143
macro avg	0.63	0.63	0.63	143
weighted avg	0.63	0.63	0.63	143

Score

0.6293706293706294

XGboost

XGBoost, or Extreme Gradient Boost, is a machine learning technique used to create gradient boosting decision trees. When it comes to unstructured data, such as photos and unstructured text data, ANN models (Artificial neural network) appear to be at the top of the list when it comes to prediction.

Performance matrices of Xgboost algorithm:

Confusion Matrix

[[53 16]

 [25 44]]

Classification Report

	precision	recall	f1-score	support
0	0.68	0.77	0.72	69
1	0.73	0.64	0.68	69
accuracy			0.70	138
macro avg	0.71	0.70	0.70	138
weighted avg	0.71	0.70	0.70	138

score

0.7028985507246377

Evaluating Performance Of The Models:

When compared alla the other algorithms Random Forest Algorithm has the high est accuracy of 0.8285714285714286. By using this algorithm ,we obtain the prediction for the loan approval or rejection.

F1-Score:

0.7833417327163604

Mean:

0.8228181529673121

CHAPTER-10
ADVANTAGES & DISADVANTAGES

Advantages

Various sources to generate a generalised dataset and apply four machine learning algorithms to the dataset, including Random forest, Logistic regression, and Decision tree.

- The advantage of the framework is that we show the requirements as a calculation, and while checking the subtleties, we determine the demands that have been approved and fulfil the requirements of the illicit customer.
- The framework is rated higher than high even out information. The shown structure is similar to a good memory.
- The risk of spreading to the necessary framework is minimal.
- Slight changes in information have little effect on the hyper plant.
- Performance and accuracy of the algorithms can be calculated and compared.
- Class imbalance can be dealt with machine learning approaches

Disadvantages

- They provided a mathematical model and did not employ machine learning methods.
- The problem of class imbalance was not addressed, and appropriate measures were not adopted.
- Existing frameworks typically fail. Computations are undeniably difficult because many of the outcomes are linked.

CHAPTER-11

CONCLUSION

The analysis begins with data cleansing and missing value processing, followed by exploratory analysis, model creation, and model evaluation. When we receive a better accuracy score and other performance indicators on the public test set, we will have the best accuracy. This paper can assist in predicting whether or not an applicant will be approved for a bank loan. When a consumer suffers a calamity, for example, the calculation cannot predict the outcome. This assessment paper can be used to determine whether a customer is capable.

APPENDIX

Source Code

Home.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>Welcome</title>

</head>

<body>

    <h1>Welcome To Loan Predictor</h1>

    <div class="Section_top">

        <div class="content">

            <marquee behavior="scroll" direction="left" color="white">Check Your Loan Applicant
will get accepted or not</marquee>

          <h3>

                Loan is a tedious process, rather than going to bank and getting rejected

                  We made it simple that you can get your loan approval status by our machine learning
model,

                We just need some of your information
```

```
            </h3>

            <h3></h3>

            <h3>Click "Predict" Button and update some of your Information</h3>

            <h3></h3>

            <a type="button" class="btn btn-primary" href="predict.html">Predict</a>

        </div>

    </div>

</body>

</html>

<style>

body,html{

    margin: 0;

    padding: 0;

    width: 100%;

    height: 100vh;

    font-family: sans-serif;

}

h1{

    text-align: center;

}

.Section_top{
```

```
    width: 100%;

    height: 100%;

    overflow: hidden;

    position: relative;

    background-image: url("https://w0.peakpx.com/wallpaper/34/887/HD-wallpaper-loan-3d-icon-white-background-3d-symbols-loan-finance-icons-3d-icons-loan-sign-business-3d-icons.jpg");

    background-position: center;

    background-repeat: no-repeat;

    background-size: cover;

    text-align: center;

    justify-content: center;

    animation: change 10s infinite ease-in-out;

}

.content{

    padding:20px;

    background:rgb(0,0,0,0.5);

    position: absolute;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

    text-transform: uppercase;
```

```
    color:white;
}
.content h3{
    text-transform:none;
    padding-top:20px;
    text-align:center;
    font-size:22px;
    color:white;
}
.content h1{
    color: blue;
    font-size: 60px;
    letter-spacing: 7px;
}
@keyframes hue {
  from {
        background-position:100% 200%;
  }
  to {
        background-position:200% 100%;
  }
```

```
}
.content a{
    background: #85c1ee;
    padding: 10px 24px;
    text-decoration: none;
    font-size: 18px;
    border-radius: 20px;
}
.content a:hover{
    color: #fff;
    background-size:400% 400%;
    background: linear-gradient(92deg, blue);
animation: hue 10s infinite linear;
}
@keyframes change{
    100%
    {
     background-image: url("https://img.freepik.com/free-photo/hands-agent-client-shaking-hands-after-signed-contract-buy-new-apartment_1150-14835.jpg?w=1060&t=st=1666641204~exp=1666641804~hmac=eab7c6e1eeb5e8e1000e10e4dab8c39be134266f416bb6e710d9d4ebbaef3c87");
    }
}
```

```
</style>
```

predict.html

```
<!DOCTYPE html>

<html lang="en">

<style type="text/css">

  *{

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  font-family: 'Montserrat', sans-serif;

}

body{

  background: grey;

  padding: 0 10px;

}

.wrapper{

  max-width: 500px;

  width: 100%;

  background: #fff;

  margin: 20px auto;

  box-shadow: 4px 4px 25px #000;
```

```
  padding: 30px;
}
.wrapper .title{
  font-size: 20px;
  font-weight: 700;
  margin-bottom: 25px;
  color: #264653;
  text-transform: uppercase;
  text-align: center;
}
.wrapper .form{
  width: 100%;
}
.wrapper .form .inputfield{
  margin-bottom: 15px;
  display: flex;
  align-items: center;
}

.wrapper .form .inputfield label{
   width: 200px;
```

```
  color: #757575;
  margin-right: 10px;
 font-size: 14px;
}

.wrapper .form .inputfield .input,
.wrapper .form .inputfield .textarea{
 width: 100%;
 outline: none;
 border: 1px solid #d5dbd9;
 font-size: 15px;
 padding: 8px 10px;
 border-radius: 3px;
 transition: all 0.3s ease;
}
.wrapper .form .inputfield .textarea{
 width: 100%;
 height: 125px;
 resize: none;
}
.wrapper .form .inputfield .custom_select{
```

```
  position: relative;
  width: 100%;
  height: 37px;
}
.wrapper .form .inputfield .custom_select:before{
  content: "";
  position: absolute;
  top: 12px;
  right: 10px;
  border: 8px solid;
  border-color: #d5dbd9 transparent transparent transparent;
  pointer-events: none;
}
.wrapper .form .inputfield .custom_select select{
  -webkit-appearance: none;
  -moz-appearance:   none;
  appearance:        none;
  outline: none;
  width: 100%;
  height: 100%;
  border: 0px;
```

```
  padding: 8px 10px;

  font-size: 15px;

  border: 1px solid #d5dbd9;

  border-radius: 3px;

}

.wrapper .form .inputfield .input:focus,

.wrapper .form .inputfield .textarea:focus,

.wrapper .form .inputfield .custom_select select:focus{

  border: 1px solid #264653;

}

.wrapper .form .inputfield .btn{

  width: 100%;

   padding: 8px 10px;

  font-size: 15px;

  border: 0px;

  background: #264653;

  color: #fff;

  cursor: pointer;

  border-radius: 3px;

  outline: none;

  text-align: center;
```

```
}
.wrapper .form .inputfield .btn:hover{
  background: #ffb5a7;
}
.wrapper .form .inputfield:last-child{
  margin-bottom: 0;
}
.wrapper .form .inputfield .btn{
  width: 100%;
   padding: 8px 10px;
  font-size: 15px;
  border: 0px;
  background: #264653;
  color: #fff;
  cursor: pointer;
  border-radius: 3px;
  outline: none;
}
</style>
<head>
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Loan Predictor App</title>

</head>

<body>

<br>

<center>

  <H1 style="color:#264653;"> Bank Loan</H1>

</center>

<div class="wrapper">

    <div class="title">

      Loan Approval Prediction Form

      <br><br><h2><b>{{prediction_text}}</b></h2>

    </div>

    <form action='/predict' method='POST'>

      <div class="form">

         <div class="inputfield">

            <label>Gender</label>

            <div class="custom_select">

              <select name="genderBox">

                <option value="">Select</option>

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

```
        <option value="Female">Female</option>
      </select>
    </div>
</div>
<div class="inputfield">
    <label>Married</label>
    <div class="custom_select">
      <select name="maritalBox">
        <option value="">Select</option>
        <option value="Yes">Yes</option>
        <option value="No">No</option>
      </select>
    </div>
</div>
 <div class="inputfield">
    <label>Dependents</label>
    <div class="custom_select">
     <select name="dependents">
      <option selected disabled hidden>Select</option>
      <option value="0">0</option>
      <option value="1">1</option>
```

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

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

    </select>

   </div>

   </div>

   <div class="inputfield">

     <label>Education</label>

     <div class="custom_select">

       <select name="educationBox">

         <option value="">Select</option>

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

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

       </select>

     </div>

   </div>

  <div class="inputfield">

     <label>Self Employed</label>

     <div class="custom_select">

       <select name = "employmentBackground">

         <option value="">Select</option>

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

```
        <option value="No">No</option>
      </select>
    </div>
</div>
<div class="inputfield">
  <label>Applicant Income (Monthly)</label>
  <input type="number" class="input" name="applicantIncomeBox" min="0">
</div>
<div class="inputfield">
  <label>Co Applicant Income</label>
  <input type="number" class="input" name="coApplicantIncomeBox" min="0">
</div>
<div class="inputfield">
  <label>Loan  Amount</label>
  <input type="number" class="input" name="laonAmtBox" min="0">
</div>
<div class="inputfield">
    <label>Loan Amount Term</label>
    <input type="number" class="input" name="laonAmtTermBox" min="0">
</div>
<div class="inputfield">
```

```
    <label>Credit History</label>
    <div class="custom_select">
      <select name = "CHBox">
        <option value="">Select</option>
        <option value="Yes">Yes</option>
        <option value="No">No</option>
      </select>
    </div>
</div>
<div class="inputfield">
    <label>Property Area</label>
    <div class="custom_select">
      <select name = "propertyAreaBox">
        <option value="">Select</option>
        <option value="Rural">Rural</option>
        <option value="SemiUrban">Semi Urban</option>
        <option value="Urban">Urban</option>
      </select>
    </div>
</div>
<div class="inputfield">
```

```
        <button type="submit" class="btn">Submit</button>

        </div>

      </form>

    </div>

</div>

</body>

</html>
```

app.py

```
from flask import Flask, render_template, requestimport requests

API_KEY = "_C1hxAGknUXtQEw2nUiKae4TxLkfB0ty6lZfiPPYzm7h"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":

API_KEY, "grant_type": 'urn:IBM:params:oauth:grant-type:apikey'})

Mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + ml token}

app = Flask(__name__)

@app.route('/')

def Home():

    return render_template('home.html')

@app.route('/predict.html')

@app.route('/predict',methods = ['POST','GET'])

def predict():
```

```
if request.method == "POST":
    gender = request.form['genderBox']
    married = request.form['maritalBox']
    dependents = request.form['dependents']
    education = request.form['educationBox']
    employment = request.form['employmentBackground']
    applicant_income = request.form['applicantIncomeBox']
    coapplicant_income = request.form['coApplicantIncomeBox']
    loan_amount = request.form['laonAmtBox']
    loan_amount_term = request.form['laonAmtTermBox']
    credit_history = request.form['CHBox']
    prop_area = request.form['propertyAreaBox']
    if gender == 'Male':
        gender = 1
    else:
        gender = 0
    if married == 'Yes':
        married = 1
    else:
        married = 0
    if dependents == '0':
```

```
    dependents = 0
elif dependents == '1':
    dependents = 1
elif dependents == '2':
    dependents = 2
else:
    dependents = 3
if education == 'Graduate':
    education = 0
else:
    education = 1
if employment == 'Yes':
    employment = 1
else:
    employment = 0
if credit_history=='Yes':
    credit_history=1
else:
    credit_history=0
if prop_area == 'Rural':
    prop_area = 0
```

```
        elif prop_area == 'Semiurban':

            prop_area = 1

        else:

            prop_area = 3

x=[[gender,married,dependents,education,employment,applicant_income,coapplicant_income,lo
an_amount, loan_amount_term,credit_history,prop_area]]

payload_scoring = {"input_data": [{"fields": [[gender,married,dependents, education,
employment,applicant_income,coapplicant_income,loan_amount,
loan_amount_term,credit_history,prop_area]], "values":x}]}

response_scoring=requests.post('https://ussouth.ml.cloud.ibm.com/ml/v4/deployments/12295bb
3-75b2-41c1-8bbb-a41a43b8ad19/predictions?version=2022-11-15', json=payload_scoring,

        headers={'Authorization': 'Bearer ' + mltoken})

        print("Scoring response")

        prediction=response_scoring.json()

        print(response_scoring.json())

        if(prediction=="N"):

            prediction="No"

        else:

            prediction="Yes"

                return render_template("predict.html", prediction_text="Congratulations Your Loan
Status is {}".format(prediction))

    else:

          return render_template("predict.html")

if __name__ == "__main__":
```

```
 app.run(debug=True)
```

Loan_Prediction.ipynb

loan_prediction.iynb

```
import pandas as pd

import numpy as np

import pickle

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

import sklearn

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.model_selection import RandomizedSearchCV

import warnings

warnings.filterwarnings('ignore')

import imblearn

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score

from sklearn.model_selection import cross_val_score
```

```
data=pd.read_csv('loan_prediction.csv')

data.head()

data.shape

data.info()

data.isnull().sum()

data.info()

data.head()

sns.distplot(data.ApplicantIncome)

sns.countplot(data.Property_Area)

sns.countplot(data.Gender)

sns.countplot(data.Education)

sns.countplot(data.Self_Employed)

sns.countplot(data.Married)

plt.figure(figsize=(12,5))

plt.subplot(121)

sns.histplot(data['ApplicantIncome'], color='r')

plt.subplot(122)

sns.histplot(data['Credit_History'])

plt.show()

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

sns.countplot(data['LoanAmount'],hue=data['Property_Area'])
```

```
sns.countplot(data['Education'],hue=data['Self_Employed'])

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

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

plt.figure(figsize=(18,4))

plt.subplot(1,4,1)

sns.countplot(data['Gender'])

plt.subplot(1,4,2)

sns.countplot(data['Education'])

plt.show()

plt.figure(figsize=(20,5))

plt.subplot(131)

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

plt.subplot(132)

sns.countplot(data['Self_Employed'], hue=data['Education'])

plt.subplot(133)

sns.countplot(data['Property_Area'], hue=data['Loan_Amount_Term'])

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

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

data.plot.line()

data.hist()

sns.swarmplot(data['Gender'],data['ApplicantIncome'],hue=data['Loan_Status'])
```

```
data.describe()

data.info()

data.isnull().sum()

data['Gender'].fillna(data['Gender'].mode()[0], inplace=True)

data['Married'].fillna(data['Married'].mode()[0], inplace=True)

data['Dependents'].fillna(data['Dependents'].mode()[0], inplace=True)

data['Self_Employed'].fillna(data['Self_Employed'].mode()[0], inplace=True)

data['LoanAmount'].fillna(data['LoanAmount'].mode()[0], inplace=True)

data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].mode()[0], inplace=True)

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

data.isnull().sum()

data.info()

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

data['Gender'] = le.fit_transform(data['Gender'])

data['Dependents'] = le.fit_transform(data['Dependents'])

data['Loan_Status'] = le.fit_transform(data['Loan_Status'])

data['Married'] = le.fit_transform(data['Married'])

data['Self_Employed'] = le.fit_transform(data['Self_Employed'])

data['Education'] = le.fit_transform(data['Education'])

data['Property_Area'] = le.fit_transform(data['Property_Area'])
```

```
data.head()

data['Gender']=data['Gender'].astype('int64')

data['Married']=data['Married'].astype('int64')

data['Dependents']=data['Dependents'].astype('int64')

data['Self_Employed']=data['Self_Employed'].astype('int64')

data['CoapplicantIncome']=data['CoapplicantIncome'].astype('int64')

data['LoanAmount']=data['LoanAmount'].astype('int64')

data['Loan_Amount_Term']=data['Loan_Amount_Term'].astype('int64')

data['Credit_History']=data['Credit_History'].astype('int64')

data.info()

from imblearn.combine import SMOTETomek

smote=SMOTETomek(0.95)

y=data['Loan_Status']

x=data.drop(columns=["Loan_ID",'Loan_Status'],axis=1)

x_bal,y_bal=smote.fit_resample(x,y)

print(y.value_counts())

print(y_bal.value_counts())

sc=StandardScaler()

x_bal_scaled=sc.fit_transform(x_bal)

x_bal_scaled = pd.DataFrame(x_bal,columns=x.columns)

x_bal_scaled
```

```
df=pd.concat([x_bal_scaled,y_bal],axis=1)

df

train,test = train_test_split(df, test_size=0.33, random_state=42)

train.to_csv('train.csv',encoding='utf-8',index=False)

test.to_csv('test.csv',encoding='utf-8',index=False)

x=df.drop(["Loan_Status"],axis=1)

y=df.Loan_Status

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

def decisionTree(x_train,x_test,y_train,y_test):

  dt = DecisionTreeClassifier()

  dt.fit(x_train,y_train)

  pred_test = dt.predict(x_test)

  pred_test

  print('***DecisionTreeClassifier***')

  print('Confusion Matrix')

  print(confusion_matrix(y_test,pred_test))

  print('Classification Report')

  print(classification_report(y_test,pred_test))

  print('Score')

  print(dt.score(x_test,y_test))

  decisionTree(x_train,x_test,y_train,y_test)
```

```
def randomForest(x_train,x_test,y_train,y_test):
    rf = RandomForestClassifier()
    rf.fit(x_train,y_train)
    pred_test = rf.predict(x_test)
    print("**** Random Forest Classifier ****")
    print('Confusion Matrix')
    print(confusion_matrix(y_test,pred_test))
    print('Classification Report')
    print(classification_report(y_test,pred_test))
    print('Score')
    print(rf.score(x_test,y_test))
randomForest(x_train,x_test,y_train,y_test)
def KNN(x_train,x_test,y_train,y_test):
    Knn = KNeighborsClassifier()
    Knn.fit(x_train,y_train)
    pred_test = Knn.predict(x_test)
    print("**** KNeigborsClassifier ****")
    print('Confusion Matrix')
    print(confusion_matrix(y_test,pred_test))
    print('Classification Report')
    print(classification_report(y_test,pred_test))
```

```
    print('Score')
    print(Knn.score(x_test,y_test))
KNN(x_train,x_test,y_train,y_test)
def xgboost(x_train,x_test,y_train,y_test):
    xg= KNeighborsClassifier()
    xg.fit(x_train,y_train)
    pred_test = xg.predict(x_test)
    print("**** GradientBoostingClassifier****")
    print('Confusion Matrix')
    print(confusion_matrix(y_test,pred_test))
    print('Classification Report')
    print(classification_report(y_test,pred_test))
    print("score")
    print(xg.score(x_test,y_test))
xgboost(x_train,x_test,y_train,y_test)
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
pred_test = rf.predict(x_test)
f1_score(pred_test,y_test,average='weighted')
cv=cross_val_score(rf,x,y,cv=5)
np.mean(cv)  pickle.dump(rf,open('rdf.pkl','wb')
```

Githhub Link: https://github.com/IBM-EPBL/IBM-Project-23299-1659877556