TEAM ID: PNT2022TMID35425

SMART LENDER-APPLICANT PREDICTION FOR LOAN APPROVAL

Team Leader

AJAYKUMAR K.R.

Team Members

HARISH J.M.
MADHAVAN P
SANJAY S

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

As the needs of people are increasing, the demand for loans in banks is also frequently getting higher every day. The number of applications is increasing every day for loan approval. There are some bank policies that they have to consider while selecting an applicant for loan approval.

Banks typically process an applicant's loan after screening and verifying the applicant's eligibility, which is a difficult and time-consuming process. In some cases, some applicants default and banks lose capital. The machine learning approach is ideal for reducing human effort and effective decision making in the loan approval process by implementing machine learning tools that use classification algorithms to predict eligible loan applicants.This work's primary objective is to predict whether the loan approval to a specific individual is safe or not.

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 Problem:

Commercial banks need an effective and quick way to shortlist and approve the loan application based on the credibility of the loan applicant. Today's banking sector faces this major problem of loan non repayment which they have approved to their customers. Generally, the major profit of the bank comes directly from the loan's interest. The loan company grants its customer loan after an intensive process of verification and validation. However, they still don't have assurance if the applicant will be able to repay the loan with no difficulties.

2.2 References:

[1] M. Sheikh, A. Goel, T. Kumar, "An Approach for Prediction of Loan Approval using Machine Learning Algorithm," International Conference on Electronics and Sustainable Communication Systems (ICESC), (2020).

[2] S. M S, R. Sunny T, "Loan Credibility Prediction System Based on Decision Tree Algorithm," International Journal of Engineering Research & Technology (IJERT) Vol. 4 Issue 09, (2015).

[3] A. Kumar, I. Garg and S. Kaur, "Loan Approval Prediction based on Machine Learning Approach," IOSR Journal of Computer Engineering, (2016).

[4] Dr K. Kavitha, "Clustering Loan Applicants

2.3 Problem Statement Definition

SBBI bank needs an effective and quick way to shortlist and approve the loan application based on the credibility of the loan applicant Today's banking sector face this major problem of loan non repayment which they have approved to their customers. Generally, the major profit of the bank comes directly from the loan's interest. The loan companies grants its customer loan after an intensive process of verification and validation. However, they still don't have assurance if the applicant will be able to repay the loan with no difficulties.

LITERATURE SURVEY

TITLE 1: Loan Prediction System Using Machine Learning

AUTHOR: Shinde, Yash Patil , Ishan Kotian , Abhinav Shinde and Reshma Gulwani

DESCRIPTION: As the needs of people are increasing, the demand for loans in banks is also frequently getting higher every day. Banks typically process an applicant's loan after screening and verifying the applicant's eligibility, which is a difficult and time-consuming process. In some cases, some applicants default and banks lose capital. The machine learning approach is ideal for reducing human effort and effective decision making in the loan approval process by implementing machine learning tools that use classification algorithms to predict eligible loan applicants

TITLE 2: Bank Loan Prediction System using Machine Learning

AUTHOR: Anshika Gupta, Vinay Punt

DESCRIPTION: With the advancement in technology, there are so many enhancements in the banking sector also. The number of applications is increasing every day for loan approval. There are some bank policies that they have to consider while selecting an applicant for loan approval. Based on some parameters, the bank has to decide which one is best for approval. It is tough and risky to check out manually every person and then recommended for loan approval. In this work, we use a machine learning technique that will predict the person who is reliable for a loan, based on the previous record of the person whom the loan amount is accredited before. This work's primary objective is to predict whether the loan approval to a specific individual is safe or not

TITLE 3: Loan Prediction Using Machine Learning and Its Deployement On Web Application

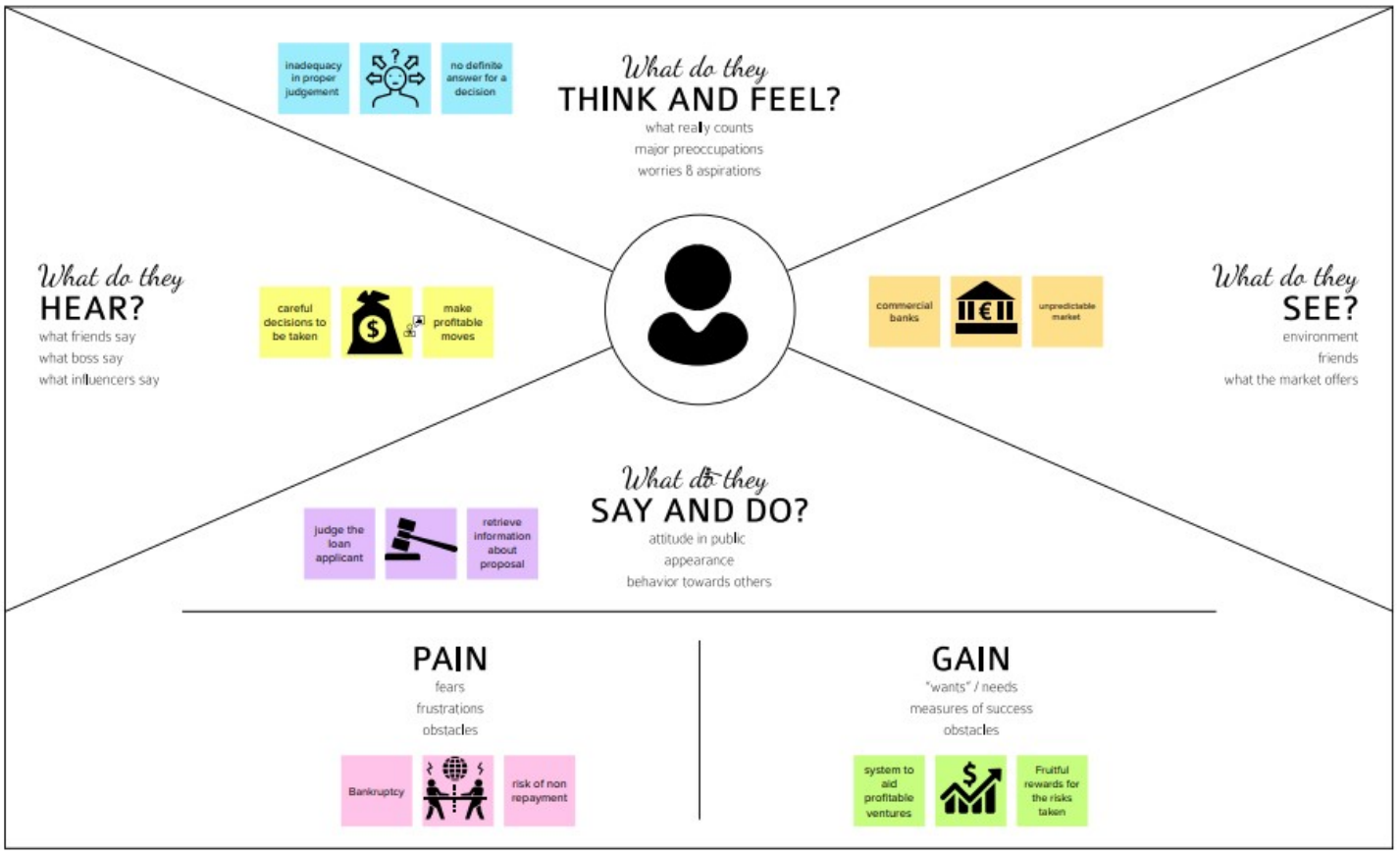
AUTHOR: C N Sujatha, N Karthik

DESCRIPTION: Loan prediction is one of the most important and most prominent research areas in the field of banking and insurance sectors. In the modern environment identifying and analyzing the patterns of the obtained sample dataset plays a vital role in this era. The loan prediction involves the application of various machine learning algorithms. There are some prediction systems in the market using deep learning and so on. But those are limited with certain features and cannot assist the users beyond those limits. The loan prediction project is developed using machine learning algorithms such as logistic regression. The Python programming language is used for the implementation of the code which has been developed in Colab and the html pages are developed for deployment of website using Visual Studio code. The proposed system can deliver high accuracy results and moderate loss for training and validate data. Finally, the results show the model implemented with high accuracy. Further, this work can be extended in order to improve the focus where the high accuracy can be obtained.

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.

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

2

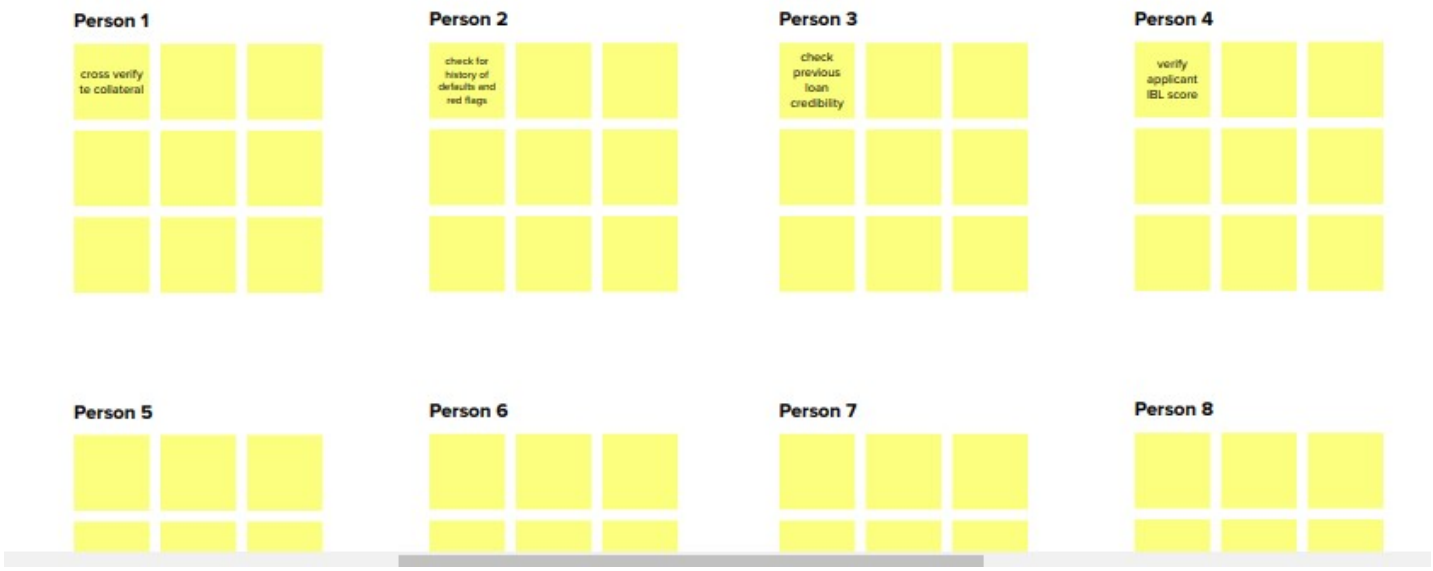
Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!



TIP

Get dataset of accepted and rejected applicants

TIP

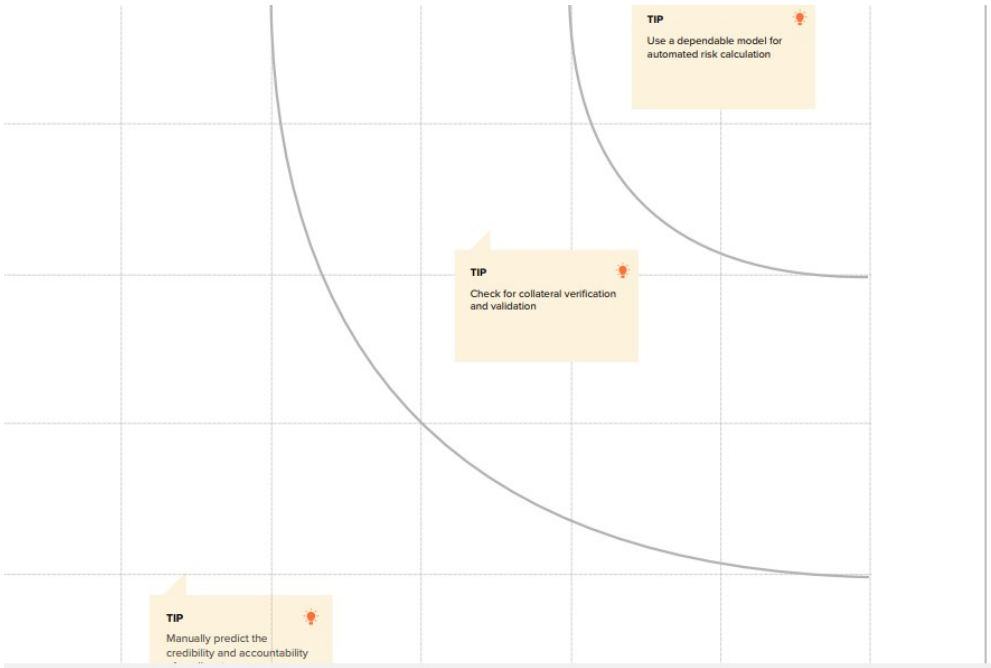
Verify the attributes of the dataset and check for correlation

TIP

Represent te dataset in feature space and split into train and test set

TIP

Train the model and test and optimize the model using the dataset



them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

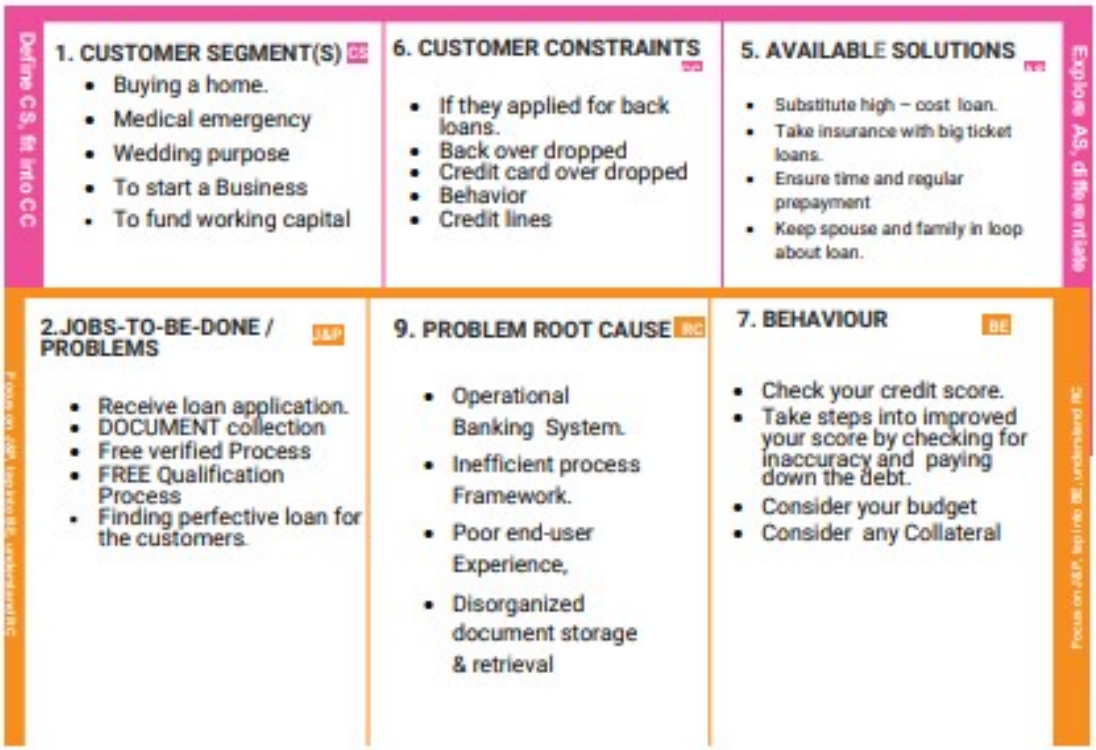
3.3 Proposed Solution

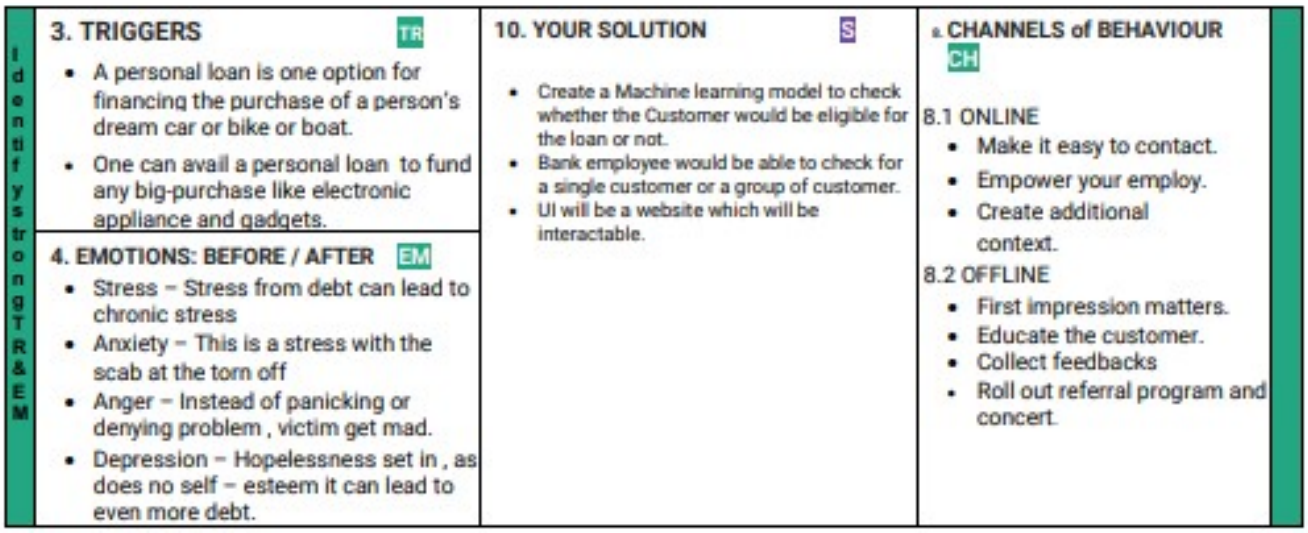
In the suggested system, we aggregate datasets from several sources to generate a generalized dataset and employ four machine learning algorithms on the same dataset, including Random forest, kNN, Decision tree, and XGBoost. The dataset we gathered for predicting supplied data is divided into 7:3 training 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,

Project Title:Smart Lender | Project Design Phase-I - Solution Fit Template | Team ID: PNT2022TMID35425





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 besimple 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

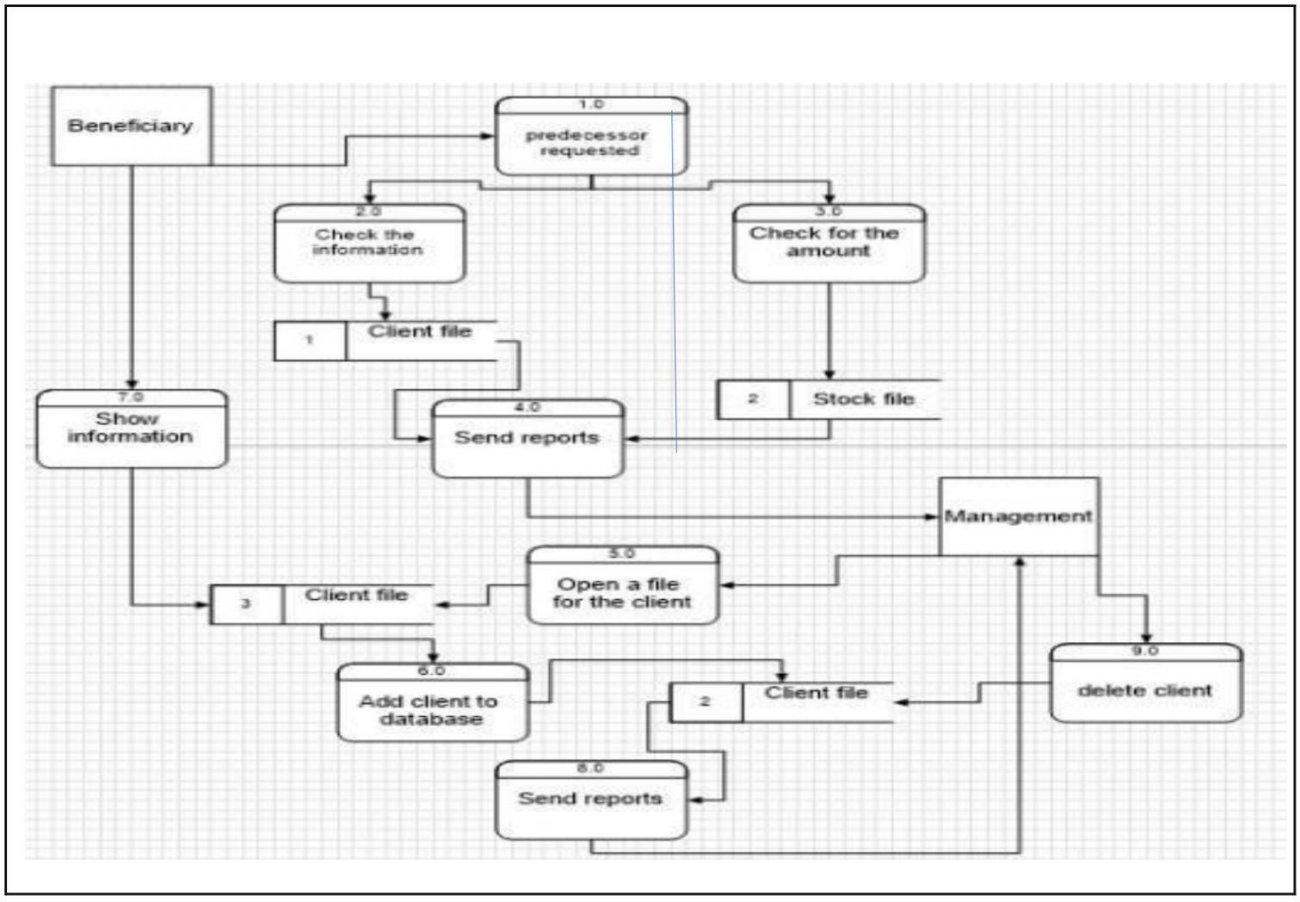
The user interfaceof the application interactive must be available in all the time when the user enter

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.

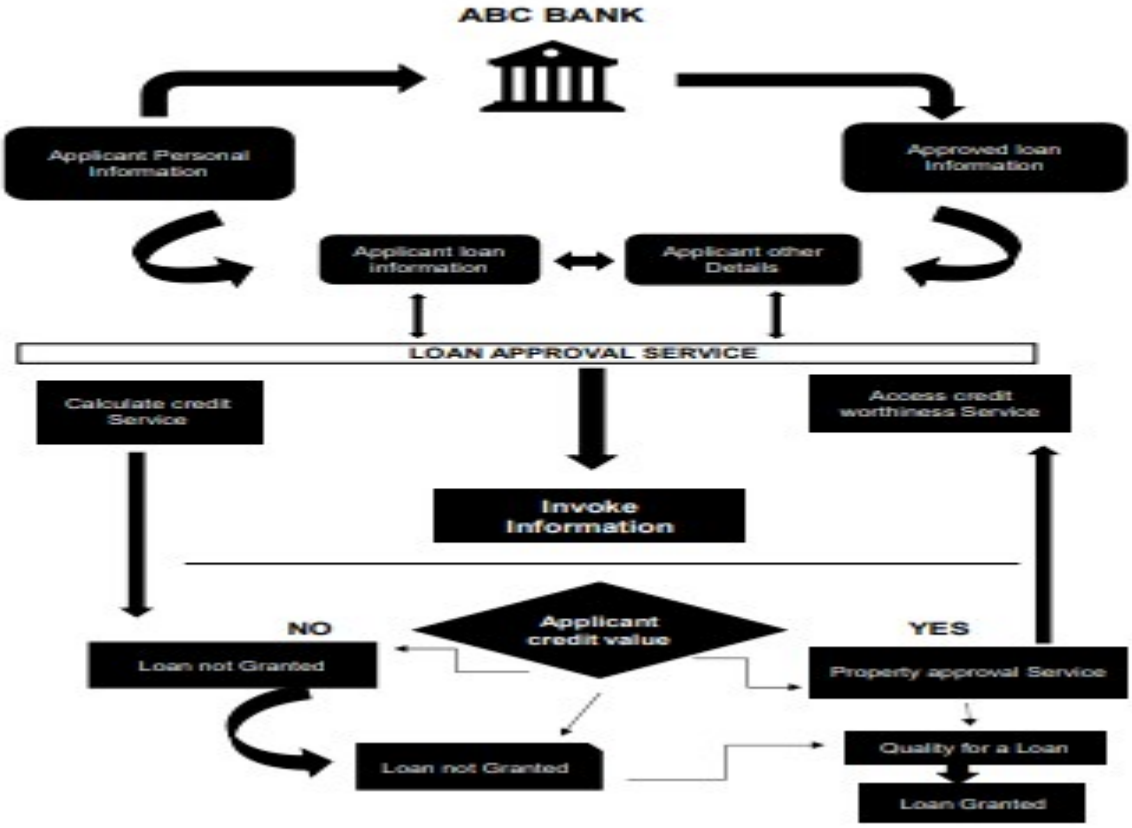


5.2 Solution And Technical Architecture

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.

Solution Architecture diagram:



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.

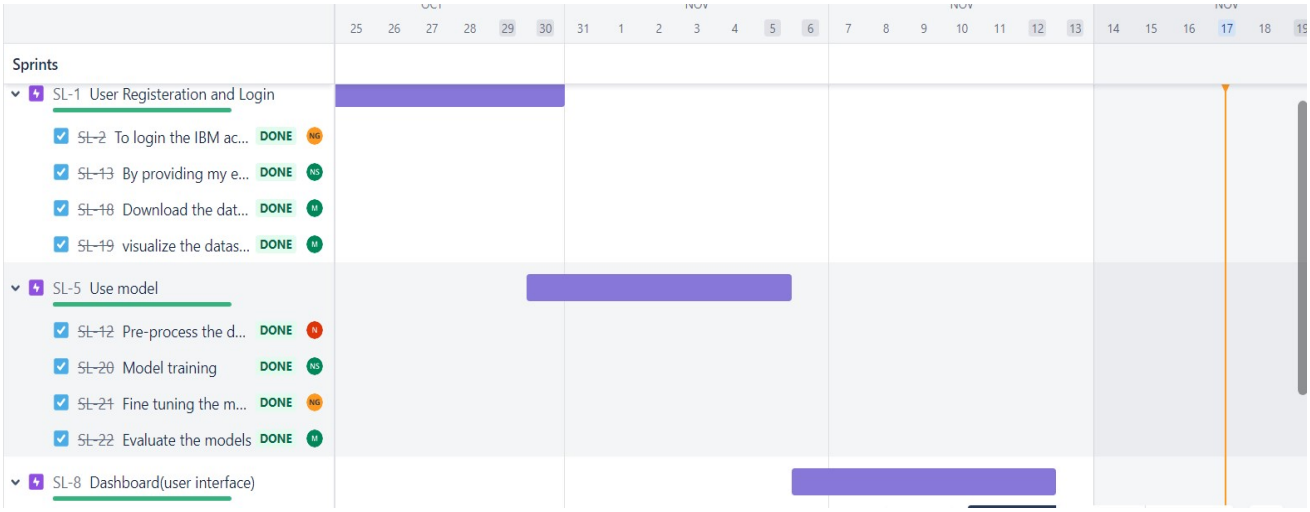
In Sprint 3 ,which involve to create the functional requirement of Dashboard (User Interface).It performs the task such as To integrate the model with flask,To create a dashboard as like User Interface,As a user able to fill the application and access the application on the user interface,To fill the 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

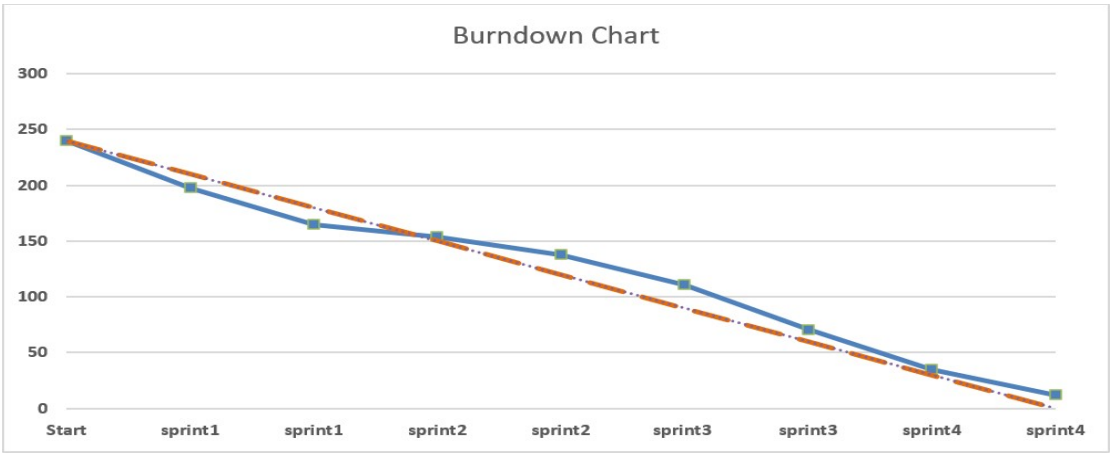
Jira

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

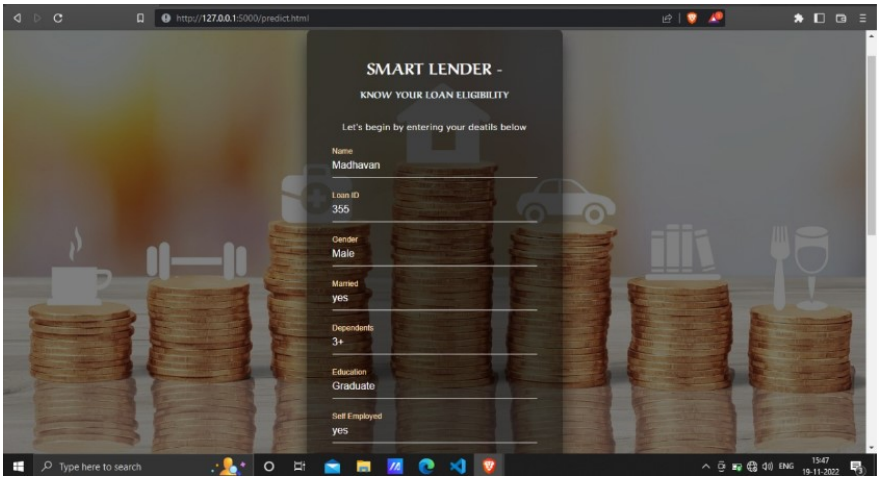
```
data.info()

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()
```

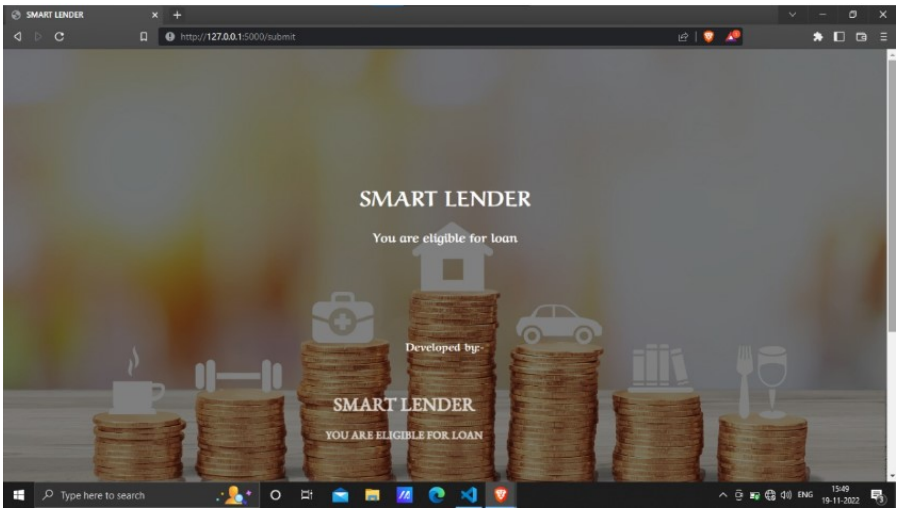
Missing values in the column "Loan monthly payment" indicate that consumers did not make loan payments. In this case, instead of the mean or median, the missing values should be imputed with zero. The original data has a category target variable. It is divided into four categories, numbered A through 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))
```

Front end Feature:





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" />
    <title>Loan Prediction</title>
      <link rel="stylesheet" type="text/css"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <link rel="stylesheet" type="text/css"
href="https://codepen.io/skjha5993/pen/bXqWpR.css">
  </head>
  <body>
    <header class="speaker-form-header">
      <center><h1>SMART LOAN PREDICTION</h1></center>

    </header>
    {% block content %}
       <div class="container">
    <form action="{{ url_for('predict') }}" method="post" class="speaker-form">
      <div class="form-row">
        <label for="full-name">Name</label>
        <input id="full-name" name="full-name" class="form-control" type="text" required/>
      </div>
      <div >

      </div>
      <br><br>
      <div class="form-row">

        <button class="btn btn-primary float-right">Submit</button>
      </div>
    </form>
       </div>
    {% endblock %}
```

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

predict.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <style>
      .header
  { width:
  500px;height:
  50px; margin:
  50px;


  justify-content: center;
  align-items: center;
}
    </style>
    <meta charset="UTF-8" />
    <title>Loan Prediction</title>

          <link rel="stylesheet" type="text/css"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <link rel="stylesheet" type="text/css"
href="https://codepen.io/skjha5993/pen/bXqWpR.css">
  </head>
  <body>
    <div class="header">
      <h1>SMART LOAN PREDICTION</h1>
      <p>Welcome {{project_name}}!!!</p>

    </div>
    <div class="container">
    <form action="{{ url_for('evaluate') }}" method="post" class="speaker-form">

      <div class="form-row">
        <label for="Gender">Gender</label>
        <select class="form-control"  id="Gender" name="Gender" required>
          <option value="1">Male</option>
```

```
    <option value="0">Female</option>
  </select>
</div><br>
<div class="form-row">
  <label for="Married">Married</label>
  <select class="form-control"  id="Married" name="Married" required>
    <option value="1">Yes</option>
    <option value="0">No</option>
  </select>
</div><br>
<div class="form-row">
  <label for="Dependents">Dependents</label>
  <input
        class="form-control"
    id="Dependents"
    name="Dependents"
    type="number"
    min="0"
    max="3"
    placeholder="No of Dependents on you....."
    required
  />
</div><br>
<div class="form-row">
  <label for="Education">Education</label>
  <select class="form-control"  id="Education" name="Education" required>
    <option value="0">Graduate</option>
    <option value="1">Not Graduate</option>
  </select>
</div><br>
<div class="form-row">
  <label for="Self Employed">Self Employed</label>
  <select class="form-control"  id="Self Employed" name="Self Employed" required>
    <option value="1">Yes</option>
    <option value="0">No</option>
  </select>
</div><br>
<div class="form-row">
  <label for="Applicant Income">Applicant Income</label>
  <input
        class="form-control"
    id="Applicant Income"
```

```
    name="Applicant Income"
    type="number"
    min="0"
    placeholder="Your Income....."
    required
   />
</div><br>
<div class="form-row">
  <label for="Co Applicant Income">Co Applicant Income</label>
  <input
        class="form-control"
    id="Co Applicant Income"
    name="Co Applicant Income"
    type="number"
    min="0"
    placeholder="Your Co Applicant Income....."
    required
   />
</div><br>
<div class="form-row">
  <label for="Loan Amount">Loan Amount</label>
  <input
        class="form-control"
    id="Loan Amount"
    name="Loan Amount"
    type="number"
    min="0"
    placeholder="Enter the Loan Amount....."
    required
   />
</div><br>
<div class="form-row">
  <label for="Loan Amount Term">Loan Amount Term</label>
  <input
        class="form-control"
    id="Loan Amount Term"
    name="Loan Amount Term"
    type="number"
    min="0"
    placeholder="Enter the Loan Amount Term in days....."
    required
   />
```

```
    </div><br>
    <div class="form-row">
      <label for="Credit History">Credit History</label>
      <select class="form-control"  id="Credit History" name="Credit History" required>
        <option value="1">Yes</option>
        <option value="0">No</option>
      </select>
    </div><br>
    <div class="form-row">
      <label for="Property Area">Property Area</label>
      <select class="form-control"  id="Property Area" name="Property Area" required>
        <option value="0">Urban</option>
        <option value="1">Semiurban</option>
        <option value="2">Rural</option>
      </select>
    </div><br>
    <!-- <div class="form-row">
      <label for="abstract">Abstract</label>
      <textarea id="abstract" name="abstract"></textarea>
      <div class="instructions">Describe your talk in 500 words or less</div>
    </div>
    <div class="form-row">
      <label class="checkbox-label" for="available">
        <input
          id="available"
          name="available"
          type="checkbox"
          value="is-available"
        />
        <span>I'm actually available the date of the talk</span>
      </label>
    </div> -->
    <div class="form-row">
      <button class="btn btn-primary float-right">Submit</button>
    </div><br><br>
  </form>
    </div>
  </body>
</html>
```

main.py

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

# NOTE: you must manually set API_KEY below using information retrieved from your
IBM Cloud account.
API_KEY = "gty1PYR_T522sN6_r51HL2g88kxNxhyQXGVp5uPGmGFC"
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 ' + mltoken}

app = Flask(__name__)
model = pickle.load(open(r'rdf.pkl','rb'))
@app.route("/", methods=['GET', 'POST'])
def home():
    return render_template("home.html")

@app.route("/predict",methods=['POST','GET'])
def predict():
    if request.method == 'POST':
        project_name=request.form['full-name']
        print(project_name)
    return render_template("predict.html",project_name=project_name)

@app.route("/success",methods=['POST','GET'])
def evaluate():
    input_feature = [int(x) for x in request.form.values()]
    print(input_feature)
    # input_feature=[np.array(input_feature)]
    print(input_feature)
    names = ['Gender', 'Married', 'Dependents', 'Education', 'Self Employed', 'Applicant
Income', 'Coapplicant Income', 'Loan Amount', 'Loan_Amount_Term', 'Credit_History',
'Property_Area']

    # NOTE: manually define and pass the array(s) of values to be scored in the next line
    payload_scoring = {"input_data": [{"fields": [names],
                                       "values": [input_feature]}]}
```

```
    response_scoring = requests.post(
        'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/a05131f3-dcb8-46cd-bf08-
1c2ecf28cc86/predictions?version=2022-11-13',
        json=payload_scoring,
        headers={'Authorization': 'Bearer ' + mltoken})
    predictions = response_scoring.json()
    prediction = predictions['predictions'][0]['values'][0][0]
    print("Scoring response") print(response_scoring.json())
    print(prediction)

    # data = pandas.DataFrame(input_feature, columns=names)
    # print(data)
    # prediction=model.predict(data)
    # print(prediction)
    # prediction = int(prediction)
    # print(type(prediction))
    loan=1
    if (prediction == 0):
        loan=0
        return render_template("success.html",result = "Loan will Not be
Approved",loan=loan)
    else:
        return render_template("success.html",result = "Loan will be Approved",loan=loan)
    return render_template("success.html")


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

Loan_Prediction.ipynb

{"cells": [{"metadata": {}, "cell_type": "code", "source": "# Authors: RAGHUL R,PRABHAKARAN S ,RAGHUL D ,RAAHULL R ", "execution_count": null, "outputs": []},
{"metadata": {}, "cell_type": "code", "source": "import os, types\nimport pandas as pd\nfrom botocore.client import Config\nimport ibm_boto3\n\ndef __iter__(self): return 0\n\n# @hidden_cell\n# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.\n# You might want to remove those credentials before you share the notebook.\nclient_5158bfd5065b40c4b6cf7e02a60cf879 = ibm_boto3.client(service_name='s3',\n ibm_api_key_id='Rob46tTNo97O_Wdw9cPUe7whW_akOBfAuD9qWugyZBTB',\n ibm_auth_endpoint=\"https://iam.cloud.ibm.com/oidc/token\",\n config=Config(signature_version='oauth'),\n endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')\n\nbody = client_5158bfd5065b40c4b6cf7e02a60cf879.get_object(Bucket='ibmsmartlender-donotdelete-pr-fn1gcvrcmxp1mg',Key='test.csv')['Body']\n# add missing __iter__ method, so pandas accepts body as file-like object\nif not hasattr(body, \"__iter__\"): body.__iter__ = types.MethodType(__iter__, body)\n\ntest = pd.read_csv(body)\ntest.head()", "execution_count": null, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "body = client_5158bfd5065b40c4b6cf7e02a60cf879.get_object(Bucket='ibmsmartlender-donotdelete-pr-fn1gcvrcmxp1mg',Key='train.csv')['Body']\n# add missing __iter__ method, so pandas accepts body as file-like object\nif not hasattr(body, \"__iter__\"): body.__iter__ = types.MethodType(__iter__, body)\n\ntrain= pd.read_csv(body)\ntrain.head()", "execution_count": 30, "outputs": [{"output_type": "execute_result", "execution_count": 30, "data": {"text/plain": " Gender Married Dependents Education Self_Employed ApplicantIncome \\\n0 1 1 0 0 0 8.699515 \n1 1 1 0 0 0 7.992269 \n2 1 1 2 0 0 8.740337 \n3 1 1 0 0 0 7.641564 \n4 1 0 0 0 0 8.334712 \n\n CoapplicantIncome LoanAmount Loan_Amount_Term Credit_History \\\n0 2250.0 5.579730 360.0 1 \n1 2900.0 4.875197 360.0 1 \n2 1695.0 5.347108 360.0 1 \n3 3150.0 4.852030 360.0 1 \n4 0.0 4.584967 360.0 0 \n\n Property_Area Loan_Status \n0 1 0 \n1 1 1 \n2 1 1 \n3 1 1 \n4 1 0 ", "text/html": " \n<style scoped>\n .dataframe tbody tr th:only-of-type {\n vertical-align: middle;\n }\n\n .dataframe tbody tr th {\n vertical-align: top;\n }\n\n .dataframe thead th {\n text-align: right;\n }\n</style>\n<table border=\"1\" class=\"dataframe\">\n <thead>\n <tr style=\"text-align: right;\">\n <th></th>\n <th>Gender</th>\n <th>Married</th>\n <th>Dependents</th>\n <th>Education</th>\n <th>Self_Employed</th>\n <th>ApplicantIncome</th>\n <th>CoapplicantIncome</th>\n <th>LoanAmount</th>\n <th>Loan_Amount_Term</th>\n <th>Credit_History</th>\n

<th>Property_Area</th>\n <th>Loan_Status</th>\n </tr>\n </thead>\n <tbody>\n <tr>\n <th>0</th>\n <td>1</td>\n <td>1</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>8.699515</td>\n <td>2250.0</td>\n <td>5.579730</td>\n <td>360.0</td>\n <td>1</td>\n <td>1</td>\n <td>0</td>\n </tr>\n <tr>\n <th>1</th>\n <td>1</td>\n <td>1</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>7.992269</td>\n <td>2900.0</td>\n <td>4.875197</td>\n <td>360.0</td>\n <td>1</td>\n <td>1</td>\n <td>1</td>\n </tr>\n <tr>\n <th>2</th>\n <td>1</td>\n <td>1</td>\n <td>2</td>\n <td>0</td>\n <td>0</td>\n <td>8.740337</td>\n <td>1695.0</td>\n <td>5.347108</td>\n <td>360.0</td>\n <td>1</td>\n <td>1</td>\n <td>1</td>\n </tr>\n <tr>\n <th>3</th>\n <td>1</td>\n <td>1</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>7.641564</td>\n <td>3150.0</td>\n <td>4.852030</td>\n <td>360.0</td>\n <td>1</td>\n <td>1</td>\n <td>1</td>\n </tr>\n <tr>\n <th>4</th>\n <td>1</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>8.334712</td>\n <td>0.0</td>\n <td>4.584967</td>\n <td>360.0</td>\n <td>0</td>\n <td>1</td>\n <td>0</td>\n </tr>\n </tbody>\n</table>\n "}, "metadata": {}}]}, {"metadata": {}, "cell_type": "code", "source": "import pandas as pd\nimport numpy as np\nfrom sklearn.preprocessing import MaxAbsScaler\nfrom sklearn.tree import DecisionTreeClassifier\nfrom sklearn.ensemble import RandomForestClassifier\nfrom sklearn.neighbors import KNeighborsClassifier\nfrom sklearn.ensemble import GradientBoostingClassifier\nfrom sklearn.metrics import confusion_matrix\nfrom sklearn.metrics import classification_report\nfrom sklearn.model_selection import cross_val_score\nfrom sklearn.metrics import f1_score\nimport pickle", "execution_count": 31, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "scaler = MaxAbsScaler()", "execution_count": 32, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "train_y = train.iloc[:,-1]\ntrain_x = train.drop('Loan_Status',axis=1)\n\ntest_y = test.iloc[:,-1]\ntest_x = test.drop('Loan_Status',axis=1)\nx = pd.concat([train_x,test_x],axis=0)\ny = pd.concat([train_y,test_y],axis=0)", "execution_count": 33, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "train_x = scaler.fit_transform(train_x)\ntest_x = scaler.transform(test_x)\n\ndt = DecisionTreeClassifier()\ndt.fit(train_x,train_y)\ny_pred = dt.predict(test_x)\nprint(\"**** Decision Tree Classifier ****\")\nprint('Confusion Matrix')\nprint(confusion_matrix(test_y,y_pred))\nprint('Classification Report')\nprint(classification_report(test_y,y_pred))", "execution_count": 34, "outputs": [{"output_type": "stream", "text": "**** Decision Tree Classifier ****\nConfusion Matrix\n[[23 20]\n [16 64]]\nClassification Report\n precision recall f1-score support\n\n 0 0.59 0.53 0.56 43\n 1 0.76 0.80 0.78 80\n\n accuracy 0.71 123\n macro avg 0.68 0.67 0.67 123\nweighted avg 0.70 0.71 0.70 123\n\n", "name": "stdout"}]}, {"metadata": {"scrolled": true}, "cell_type": "code", "source": "", "execution_count": null, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "pip install ibm_watson_machine_learning\n", "execution_count": 35, "outputs": [{"output_type": "stream", "text": "Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)\nRequirement already

satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.11.0)\nRequirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2022.9.24)\nRequirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9)\nRequirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3)\nRequirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7)\nRequirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3)\nRequirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.26.0)\nRequirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)\nRequirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)\nRequirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)\nRequirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (0.10.0)\nRequirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)\nRequirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.8.2)\nRequirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2021.3)\nRequirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.20.3)\nRequirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (1.15.0)\nRequirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (2.0.4)\nRequirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm_watson_machine_learning) (3.3)\nRequirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm_watson_machine_learning) (3.6.0)\nRequirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)\nNote: you may need to restart the kernel to use updated packages.\n", "name": "stdout"}]}, {"metadata": {}, "cell_type": "code", "source": "from ibm_watson_machine_learning import APIClient\nimport json",

"execution_count": 36, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "wml_credentials = {\n   \"apikey\" : \"gty1PYR_T522sN6_r51HL2g88kxNxhyQXGVp5uPGmGFC\",\n   \"url\" : \"https://us-south.ml.cloud.ibm.com\"\n}", "execution_count": 37, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "wml_client = APIClient(wml_credentials)", "execution_count": 38, "outputs": []}, {"metadata": {"scrolled": true}, "cell_type": "code", "source": "wml_client.spaces.list()", "execution_count": 39, "outputs": [{"output_type": "stream", "text": "Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50\n------------------------------------ ------------ ------------------------\nID   NAME   CREATED\nbfdc6002-40bf-49da-b75a-6a88c850d1b7 loan-prediction 2022-11-13T09:43:10.710Z\n------------------------------------ ------------ ------------------------\n", "name": "stdout"}]}, {"metadata": {}, "cell_type": "code", "source": "SPACE_ID = \"bfdc6002-40bf-49da-b75a-6a88c850d1b7\"", "execution_count": 40, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "wml_client.set.default_space(SPACE_ID)", "execution_count": 41, "outputs": [{"output_type": "execute_result", "execution_count": 41, "data": {"text/plain": "'SUCCESS'"}, "metadata": {}}]}, {"metadata": {"scrolled": true}, "cell_type": "code", "source": "wml_client.software_specifications.list(100)", "execution_count": 16, "outputs": [{"output_type": "stream", "text": "------------------------------------ ------------------------ ----------\nNAME   ASSET_ID   TYPE\ndefault_py3.6   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base\nkernel-spark3.2-scala2.12   020d69ce-7ac1-5e68-ac1a-31189867356a base\npytorch-onnx_1.3-py3.7-edt   069ea134-3346-5748-b513-49120e15d288 base\nscikit-learn_0.20-py3.6   09c5a1d0-9c1e 4473-a344-eb7b665ff687 base\nspark-mllib_3.0-scala_2.12   09f4cff0-90a7-5899-b9ed-1ef348aebdee base\npytorch-onnx_rt22.1-py3.9   0b848dd4-e681 5599-be41-b5f6fccc6471 base\nai-function_0.1-py3.6   0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base\nshiny-r3.6   0e6e79df-875e-4f24-8ae9-62dcc2148306 base\ntensorflow_2.4-py3.7-horovod   1092590a-307d-563d-9b62-4eb7d64b3f22 base\npytorch_1.1-py3.6   10ac12d6-6b30-4ccd 8392-3e922c096a92 base\ntensorflow_1.15-py3.6-ddl   111e41b3-de2d-5422-a4d6-bf776828c4b7 base\nautoai-kb_rt22.2-py3.10   125b6d9a-5b1f-5e8d-972a-b251688ccf40 base\nruntime-22.1-py3.9   12b83a17-24d8-5082-900f-0ab31fbfd3cb base\nscikit-learn_0.22-py3.6   154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base\ndefault_r3.6   1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base\npytorch-onnx_1.3-py3.6   1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base\nkernel-spark3.3-r3.6   1c9e5454-f216-59dd-a20e-474a5cdf5988 base\npytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f base\ntensorflow_2.1-py3.6   1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base\nspark-mllib_3.2   20047f72-0a98-58c7-9ff5-a77b012eb8f5 base\ntensorflow_2.4-py3.8-horovod   217c16f6-178f-56bf-824a-b19f20564c49 base\nruntime-22.1-py3.9-cuda   26215f05-08c3-5a41-a1b0-da66306ce658 base\ndo_py3.8   295addb5-9ef9-547e-9bf4-92ae3563e720 base\nautoai-ts_3.8-py3.8   2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base\ntensorflow_1.15-py3.6   2b73a275-7cbf-420b-a912-eae7f436e0bc base\nkernel-spark3.3-py3.9   2b7961e2-e3b1-5a8c-a491-482c8368839a base\npytorch_1.2-py3.6   2c8ef57d-2687-4b7d-

acce-01f94976dac1 base\nspark-mllib_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base\npytorch-onnx_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base\nspark-mllib_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base\nspark-mllib_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base\nautoai-ts_rt22.2-py3.10 396b2e83-0953-5b86-9a55-7ce1628a406f base\nxgboost_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base\npytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base\npytorch-onnx_rt22.2-py3.10 40e73f55-783a-5535-b3fa-0c8b94291431 base\ndefault_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base\nautoai-ts_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base\nautoai-obm_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base\npmml-3.0_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base\nspark-mllib_2.4-r_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base\nxgboost_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base\npytorch-onnx_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base\nautoai-ts_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base\nspark-mllib_2.4-scala_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base\nspark-mllib_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base\nautoai-obm_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base\nspss-modeler_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base\ncuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base\nautoai-kb_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base\npytorch-onnx_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base\nspark-mllib_2.3-r_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base\ntensorflow_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base\nspss-modeler_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base\npytorch-onnx_1.2-py3.6 692a6a4d-2c4d-45ff-a1ed-b167ee55469a base\nspark-mllib_2.3-scala_2.11 7963efe5-bbec-417e-92cf-0574e21b4e8d base\nspark-mllib_2.4-py37 7abc992b-b685-532b-a122-a396a3cdbaab base\ncaffe_1.0-py3.6 7bb3dbe2-da6e-4145-918d-b6d84aa93b6b base\npytorch-onnx_1.7-py3.7 812c6631-42b7-5613-982b-02098e6c909c base\ncuda-py3.6 82c79ece-4d12-40e6-8787-a7b9e0f62770 base\tensorflow_1.15-py3.6-horovod 8964680e-d5e4-5bb8-919b-8342c6c0dfd8 base\nhybrid_0.1 8c1a58c6-62b5-4dc4-987a-df751c2756b6 base\npytorch-onnx_1.3-py3.7 8d5d8a87-a912-54cf-81ec-3914adaa988d base\ncaffe-ibm_1.0-py3.6 8d863266-7927-4d1e-97d7-56a7f4c0a19b base\nspss-modeler_17.1 902d0051-84bd-4af6-ab6b-8f6aa6fdeabb base\ndo_12.10 9100fd72-8159-4eb9-8a0b-a87e12eefa36 base\ndo_py3.7 9447fa8b-2051-4d24-9eef-5acb0e3c59f8 base\nspark-mllib_3.0-r_3.6 94bb6052-c837-589d-83f1-f4142f219e32 base\ncuda-py3.7-opence 94e9652b-7f2d-59d5-ba5a-23a414ea488f base\nlp-py3.8 96e60351-99d4-5a1c-9cc0-473ac1b5a864 base\ncuda-py3.7 9a44990c-1aa1-4c7d-baf8-c4099011741c base\nhybrid_0.2 9b3f9040-9cee-4ead-8d7a-780600f542f7 base\nspark-mllib_3.0-py38 9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418 base\nautoai-kb_3.3-py3.7 a545cca3-02df-5c61-9e88-998b09dc79af base\nspark-mllib_3.0-py39 a6082a27-5acc-5163-b02c-6b96916eb5e0 base\nruntime-22.1-py3.9-do a7e7dbf1-1d03-5544-994d-e5ec845ce99a base\ndefault_py3.8 ab9e1b80-f2ce-592c-a7d2-4f2344f77194 base\ntensorflow_rt22.1-py3.9 acd9c798-

6974-5d2f-a657-ce06e986df4d base\nkernel-spark3.2-py3.9 ad7033ee-794e-58cf- 812e-a95f4b64b207 base\nautoai-obm_2.0 with Spark 3.0 af10f35f-69fa-5d66-9bf5-acb58434263a base\ndefault_py3.7_opence c2057dd4-f42c-5f77-a02f-72bdbd3282c9 base\ntensorflow_2.1-py3.7 c4032338-2a40-500a-beef-b01ab2667e27 base\ndo_py3.7_opence cc8f8976-b74a-551a-bb66-6377f8d865b4 base\nspark-mllib_3.3 d11f2434-4fc7-58b7-8a62-755da64fdaf8 base\nautoai-kb_3.0-py3.6 d139f196-e04b-5d8b-9140-9a10ca1fa91a base\nspark-mllib_3.0-py36 d82546d5-dd78-5fbb-9131-2ec309bc56ed base\nautoai-kb_3.4-py3.8 da9b39c3-758c-5a4f-9cfd-457dd4d8c395 base\nkernel-spark3.2-r3.6 db2fe4d6-d641-5d05-9972-73c654c60e0a base\nautoai-kb_rt22.1-py3.9 db6afe93-665f-5910-b117-d879897404d9 base\ntensorflow_rt22.1-py3.9-horovod dda170cc-ca67-5da7-9b7a-cf84c6987fae base\nautoai-ts_1.0-py3.7 deef04f0-0c42-5147-9711-89f9904299db base\ntensorflow_2.1-py3.7-horovod e384fce5-fdd1-53f8-bc71-11326c9c635f base\ndefault_py3.7 e4429883-c883-42b6-87a8-f419d64088cd base\do_22.1 e51999ba-6452-5f1f-8287-17228b88b652 base\nautoai-obm_3.2 eae86aab-da30-5229-a6a6-1d0d4e368983 base\ntensorflow_rt22.2-py3.10 f65bd165-f057-55de-b5cb-f97cf2c0f393 base\ndo_20.1 f686cdd9-7904-5f9d-a732-01b0d6b10dc5 base\npytorch-onnx_rt22.2-py3.10-edt f8a05d07-e7cd-57bb-a10b-23f1d4b837ac base\nscikit-learn_0.19-py3.6 f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9 base\ntensorflow_2.4-py3.8 fe185c44-9a99-5425-986b-59bd1d2eda46 base\n_____ _____ ___\n", "name": "stdout"}]}, {"metadata": {}, "cell_type": "code", "source": "import sklearn\nsklearn.__version__", "execution_count": 17, "outputs": [{"output_type": "execute_result", "execution_count": 17, "data": {"text/plain": "'1.0.2'"}, "metadata": {}}]}, {"metadata": {}, "cell_type": "code", "source": "MODEL_NAME = 'Model_building_SL'\nDEPLOYMENT_NAME = 'loan-prediction'\nDEMO_MODEL = dt", "execution_count": 19, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')\n", "execution_count": 20, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "model_props = {\n   wml_client.repository.ModelMetaNames.NAME: MODEL_NAME, \n   wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0', \n wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid \n}", "execution_count": 21, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "model_details = wml_client.repository.store_model(\n   model=DEMO_MODEL, \n meta_props=model_props, \n   training_data=train_x, \n   training_target=train_y\n)", "execution_count": 23, "outputs": [{"output_type": "stream", "text": "Deployment creation failed. Error: 400. {\"trace\":\"-x2w0kl9hcma4\",\"errors\":[{\"code\":\"space_lacks_compute\",\"message\":\"Space: bfdc6002-40bf-49da-b75a-6a88c850d1b7 is not associated with a WML instance which is mandatory for create_deployment operation\"}]}\n", "name": "stderr"}]}, {"metadata": {}, "cell_type": "code", "source": "model_details", "execution_count": 24, "outputs": [{"output_type": "execute_result", "execution_count": 24, "data": {"text/plain": "{'entity': {'hybrid_pipeline_software_specs': [],\n 'label_column': 'Loan_Status',\n 'schemas': {'input': [{'fields': [{'name': 'f0', 'type': 'float'},\n   {'name': 'f1', 'type': 'float'},\n   {'name': 'f2',

'type': 'float'},\n {'name': 'f3', 'type': 'float'},\n {'name': 'f4', 'type': 'float'},\n {'name': 'f5', 'type': 'float'},\n {'name': 'f6', 'type': 'float'},\n {'name': 'f7', 'type': 'float'},\n {'name': 'f8', 'type': 'float'},\n {'name': 'f9', 'type': 'float'},\n {'name': 'f10', 'type': 'float'}],\n 'id': '1',\n 'type': 'struct'}],\n 'output': []},\n 'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',\n 'name': 'runtime-22.1-py3.9'},\n 'type': 'scikit-learn_1.0'},\n 'metadata': {'created_at': '2022-11-13T09:52:02.094Z',\n 'id': '2a0df4f2-a693-4aca-8ec5-08a2db4ef3ce',\n 'modified_at': '2022-11-13T09:52:05.819Z',\n 'name': 'Model_building_SL',\n 'owner': 'IBMid-6610045XS5',\n 'resource_key': 'e7668676-1edf-4e44-a413-67287f8cc8be',\n 'space_id': 'bfdc6002-40bf-49da-b75a-6a88c850d1b7'},\n 'system': {'warnings': []}}"}, "metadata": {}}]}, {"metadata": {}, "cell_type": "code", "source": "model_id = wml_client.repository.get_model_id(model_details)\nmodel_id", "execution_count": 25, "outputs": [{"output_type": "execute_result", "execution_count": 25, "data": {"text/plain": "'2a0df4f2-a693-4aca-8ec5-08a2db4ef3ce'"}, "metadata": {}}]}, {"metadata": {}, "cell_type": "code", "source": "deployment_props = {\n wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME, \n wml_client.deployments.ConfigurationMetaNames.ONLINE: {}\n}", "execution_count": 26, "outputs": []}, {"metadata": {}, "cell_type": "code", "source": "deployment = wml_client.deployments.create(\n artifact_uid=model_id, \n meta_props=deployment_props \n)", "execution_count": 42, "outputs": [{"output_type": "stream", "text": "\n\n###\n\nSynchronous deployment creation for uid: '2a0df4f2-a693-4aca-8ec5-08a2db4ef3ce' started\n\n###\n\n\ninitializing\nNote: online_url is deprecated and will be removed in a future release. Use serving_urls instead.\n\nready\n\n\n--\nSuccessfully finished deployment creation, deployment_uid='9038b222-47c3-4cde-8960-db3a05b2fa59'\n--\n\n\n", "name": "stdout"}]}, {"metadata": {}, "cell_type": "markdown", "source": "Copyright \u00a9 2020, 2021, 2022 IBM. This notebook and its source code are released under the terms of the MIT License."}], "metadata": {"kernelspec": {"name": "python3", "display_name": "Python 3.9", "language": "python"}, "language_info": {"name": "python", "version": "3.9.13", "mimetype": "text/x-python", "codemirror_mode": {"name": "ipython", "version": 3}, "pygments_lexer": "ipython3", "nbconvert_exporter": "python", "file_extension": ".py"}}, "nbformat": 4, "nbformat_minor": 4}

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-3111-1658502171

DEMO VIDEO LINK:

https://drive.google.com/file/d/1wJD2cK-uHNUC8OU7Xe03OPYEQcByU6qV/view?usp=share_link