PROJECT REPORT

SMART LENDER - Applicant Credibility Prediction For Loan Approval

SUBMITTED BY

TEAM ID: PNT2022TMID21336

ABHIJITH VS- 917719D002
AADIL KHAN A- 9177D001
SASI KUMAR M- 917719D082
SHASHI VISHNU M- 917719D089

1.INTRODUCTION

1.1 Project Overview

The Banking Sector across the world provides loans to the citizens and the interest given in return for providing loans are the main source of income for the Bank. The loan repaying capacity of an individual is determined by the factors such as Cibil Score, Repaying Capacity etc. of the applicant. The management of the bank must be able to predict whether the applicant, whether they are an individual or a group of individuals, are able to repay the loan interest in time. If the loan is not repaid in time, it will lead to a dent in the economy of the country. Also we have to consider the factors such as Assets and Liabilities of the applicant. By considering all these factors we must be able to predict the credibility of the applicant. In this project we will build a machine learning model to determine whether the applicant is able to repay the lending company or not.

1.2 Purpose

The main objectives of this project includes'

- Have basic knowledge about Machine Learning
- Knowledge about Python Programming Language
- At the end of the project have knowledge about cloud computing.
- Building HTML pages for UI (user interface)
- Training the given dataset with different algorithms

2.LITERATURE SURVEY

2.1 Existing Problem

In the existing system, we only have a cibil score to determine the repaying capacity of the applicant. We are not able to reassure you whether the details provided are true. Also there may be frauds and some false information provided in the application of the applicant. Also the existing solutions don't consider Gender and Marital Status which are quite important for deciding the Credibility of the client. With the currently available resources we need manual verification of the details, which is very time consuming. Also the model is not very accurate and has many flaws, because of which we don't get desirable results. The current banking system is usually more manual than automatic which leads to some errors that may be misused by the applicant. To avoid this it is necessary to have a machine learning approach that deals with this particular problem.

2.2 References

IEEE Papers:

- **Prediction of Modernized Loan Approval System Based on Machine Learning Approach**
  Author: Vishal Singh, Ayushman Yadav, Rajat Awasthi
  Year: 2021
- **Predictive And Probabilistic Approach Using Logistic Regression**
  Author: Ashelsha Vaidhya

Year: 2017

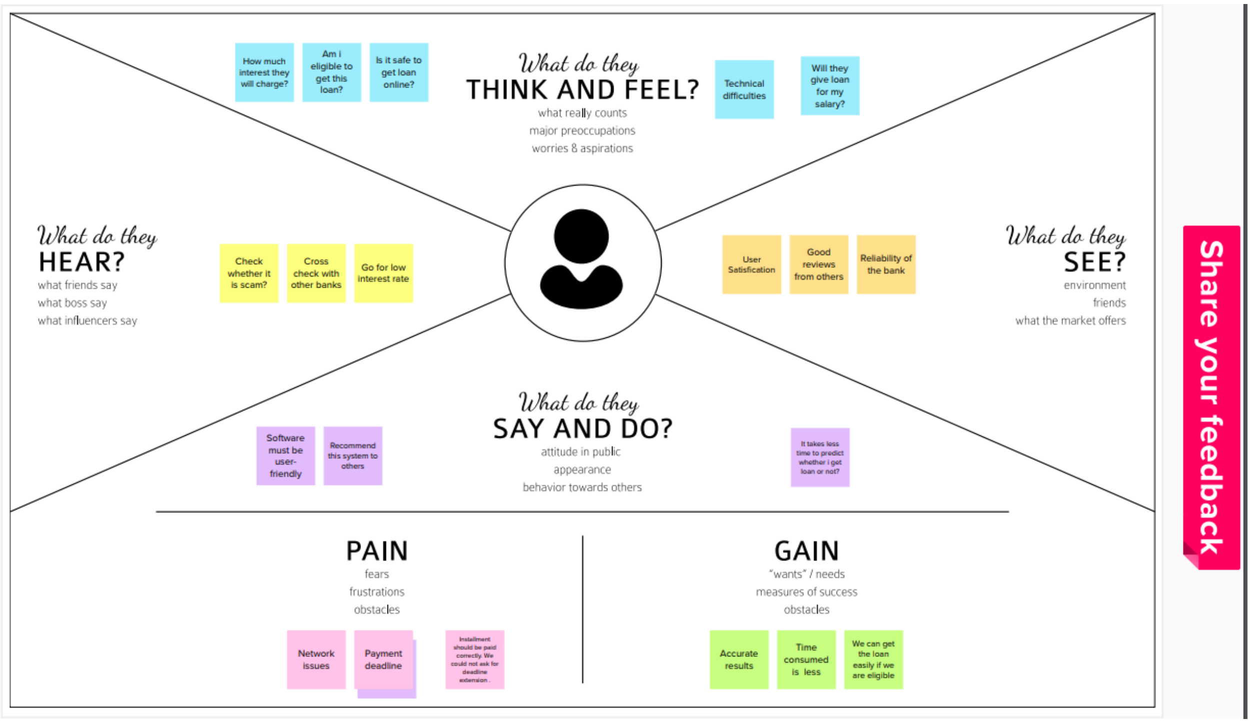
- **Bank Loan Prediction System using Machine Learning**
  Author: Anshika Gupta , Vinay Pant , Sudhanshu Kumar, Pravesh Kumar Bansal
  Year: 2020
- **Loan Delinquency Prediction**
  Author: Kathe Rutika Pramod, Panhale Sakshi Dattatray
  Year: 2020
- **Analysis Of Loan Availability Using Machine Learning Techniques**
  Author: Sharayu Dosalwar, Ketki Kinkar, Rahul Sannat, Dr Nitin Pise
  Year: 2022
- **Algorithm For The Loan Credibility Prediction System**
  Author: Soni P M, Varghese Paul
  Year: 2022

2.3 Problem Statement Definition

Loans are the core business of the banking sector. The main profit comes from the interest repayment of the applicant. The banking industry must ensure that the applicant has submitted proper documentation about their need for the loan and also are they able to repay the loan without any delay within the stipulated time period. To ensure this the Credibility of the applicant must be thoroughly checked. The loans include Housing, Educational, Vehicular, Personal loans etc. These loans have different interest rates and repayment periods. These loans are applied by people in all sectors which include Rural, Semi-Urban and Urban areas. Taking into consideration the following factors we must develop a machine learning algorithm to find out the Credibility of Loan Applications using a given Dataset and deploy the application in the IBM cloud.

3.IDEATION AND PROPOSED SOLUTION

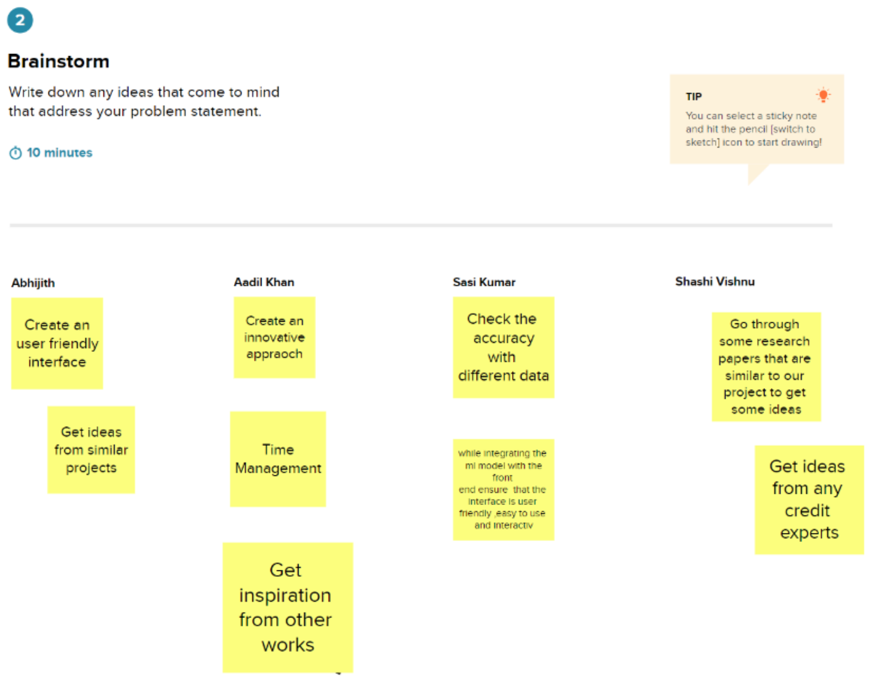
3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

During the Brainstorming sessions we came up with several ideas on how to implement the ideas to create a successful model for the project. We collected and validated the ideas of every team member and segregated them. The main objective is to create a user friendly application with an appealing User Interface. The other objectives included

- Check the accuracy with different datasets
- Time Management
- Get inspiration from other research experts
- Collect inputs about loan schemes from banks
- Go through the research papers for better understanding of the problem statement
- Front end and Back End development



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!

Abhijith

Create an user friendly interface

Get ideas from similar projects

Aadil Khan

Create an innovative appraoch

Time Management

Get inspiration from other works

Sasi Kumar

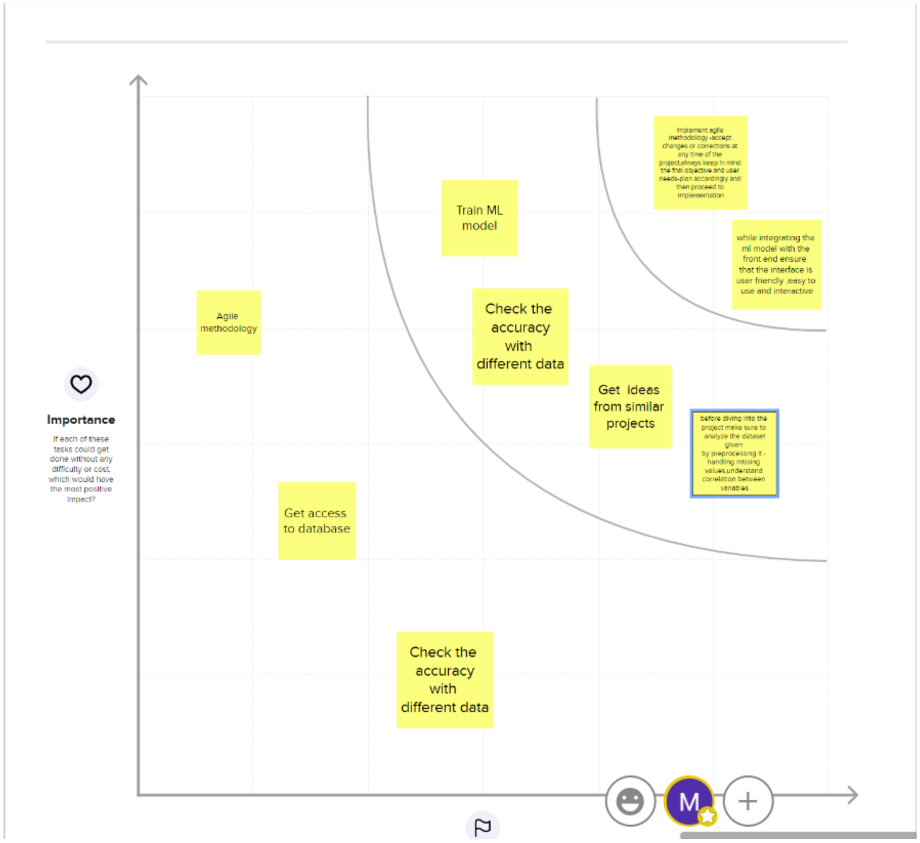
Check the accuracy with different data

while integrating the ml model with the front
end ensure that the interface is user friendly ,easy to use and interactiv

Shashi Vishnu

Go through some research papers that are similar to our project to get some ideas

Get ideas from any credit experts



Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
M
+

Implement agile methodology -accept changes or corrections at any time of the project,always keep in mind the final objective and user needs-plan accordingly and then proceed to implementation

Train ML model

while integrating the ml model with the front end ensure that the interface is user friendly ,easy to use and interactive

Agile methodology

Check the accuracy with different data

Get Ideas from similar projects

before diving into the project make sure to analyze the dataset given
by preprocessing it - handling missing values,understand correlation between variables

Get access to database

Check the accuracy with different data

3.3 Proposed Solution

IDEA:

A machine learning algorithm must be deployed which considers the following characteristics which include Gender, Educational Qualification, Loan amount, Repaying capacity, Interest rates, Number of dependencies etc.. These characteristics are used to determine whether an individual is eligible for the loan amount to be sanctioned.

We can use several machine learning algorithms like KNN, Decision Tree, Xgboost, Random forest to calculate the ML model accurately. By combining these algorithms we can get a single score to improve the accuracy of the ML model.

SOCIAL IMPACT :

Banking sector plays an important role in the country's economy. The success of this industry depends on the interest received from the clients. So if we have a model that is based on machine learning we will be able to predict the accurate data and provide sanctions to the loans which in turn will produce interest at the correct time and will boost the economy of the country.

BENEFITS :

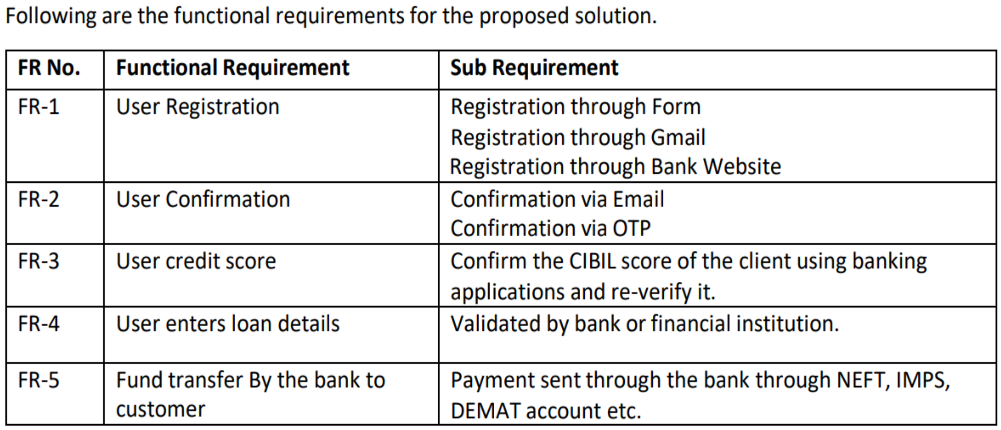
By using the ML algorithm we are able to reduce human error and bias. As the process is automated we are able to sanction based on the results provided by the automated machine. By doing this we can largely contribute toward the development of the banking sector.

3.4 Problem Solution Fit

The banking sector faces the issue of properly sanctioning loans to its customers. To address these issues we have introduced an automated method which reduces human error and bias, and sanctions loans only on merit basis. By using univariate, bi-variate and multivariate analysis we are able to visualize and analyze the dataset and provide conclusive and accurate results for the decision to be taken by the banking experts.

4.REQUIREMENT ANALYSIS

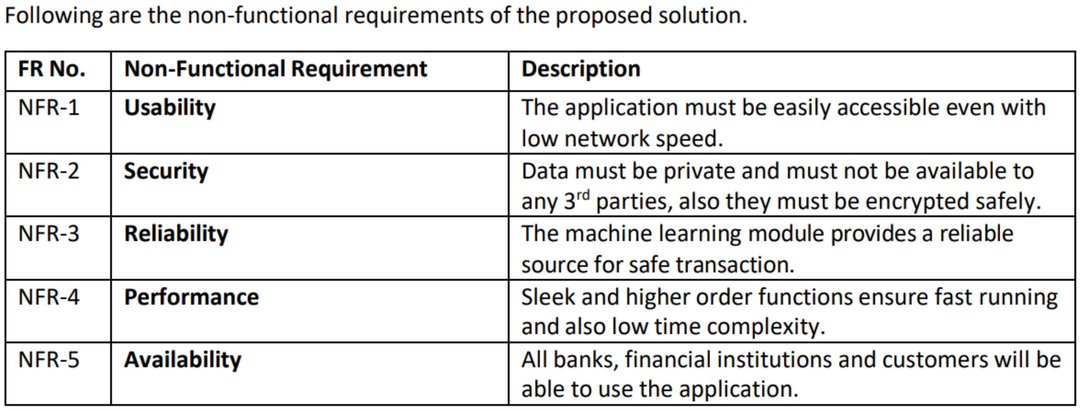
4.1 Functional Requirements



Following are the functional requirements for the proposed solution.

FR No.	Functional Requirement	Sub Requirement
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Bank Website
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User credit score	Confirm the CIBIL score of the client using banking applications and re-verify it.
FR-4	User enters loan details	Validated by bank or financial institution.
FR-5	Fund transfer By the bank to customer	Payment sent through the bank through NEFT, IMPS, DEMAT account etc.

4.2 Non-Functional Requirements

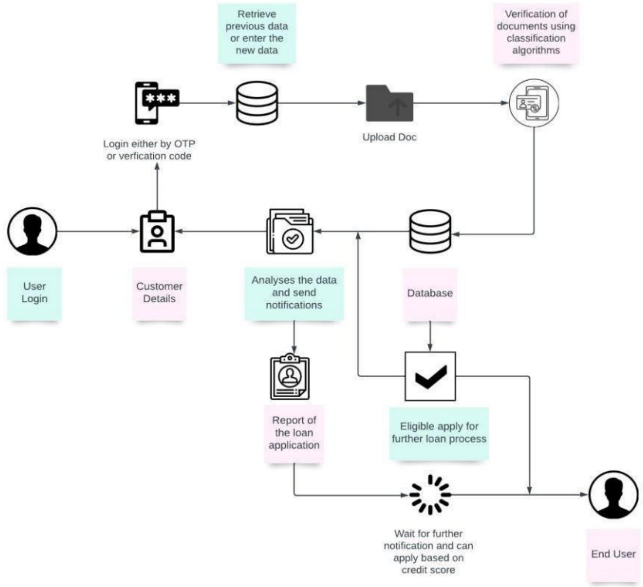


Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application must be easily accessible even with low network speed.
NFR-2	**Security**	Data must be private and must not be available to any 3rd parties, also they must be encrypted safely.
NFR-3	**Reliability**	The machine learning module provides a reliable source for safe transaction.
NFR-4	**Performance**	Sleek and higher order functions ensure fast running and also low time complexity.
NFR-5	**Availability**	All banks, financial institutions and customers will be able to use the application.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

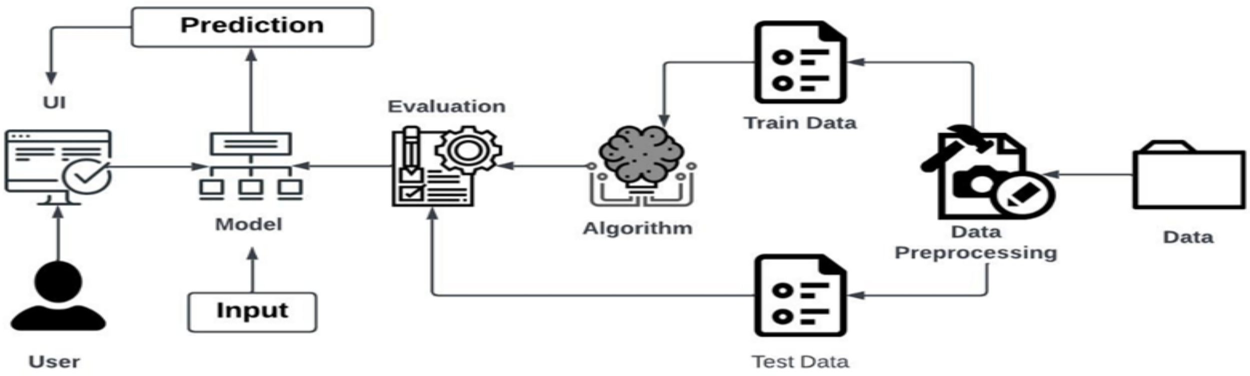

Retrieve previous data or enter the new data
Verification of documents using classification algorithms
Login either by OTP or verfication code
Upload Doc
User Login
Customer Details
Analyses the data and send notifications
Database
Report of the loan application
Eligible apply for further loan process
Wait for further notification and can apply based on credit score
End User

5.2 Solution and Technical Architecture

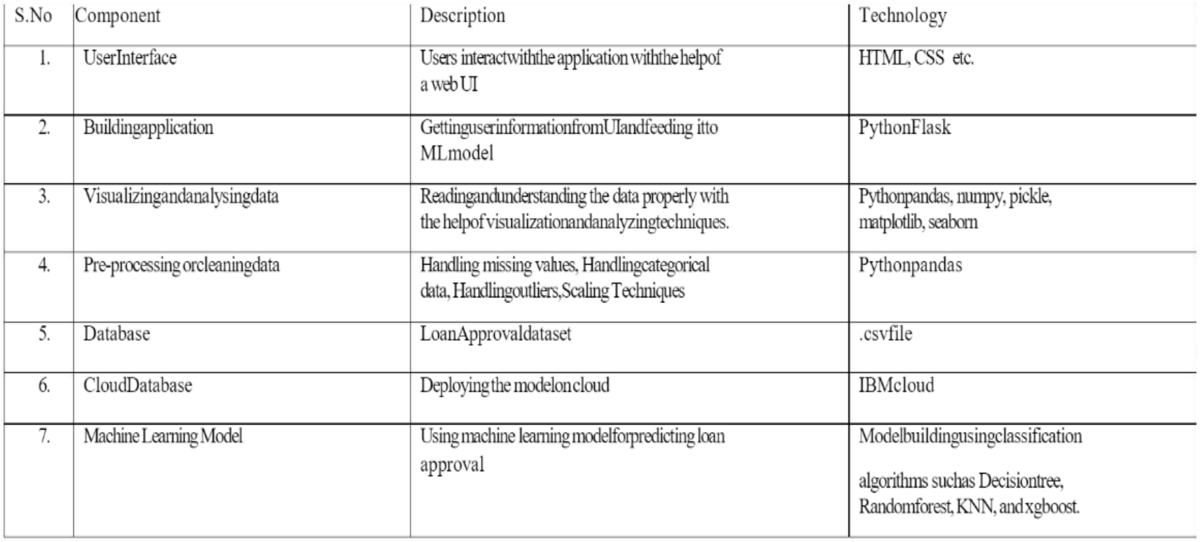
Solution Architecture

A complicated process with several sub-processes, solution architecture connects business issues with technological solutions. Its goals are to:

- Track down the most effective technological remedy for current company issues.
- Explain to project stakeholders the structure, traits, behavior, and other features of the software.
- Specify the features, stages of development, and requirements for the solution.
- Offer guidelines for how the solution is created, managed, and delivered.

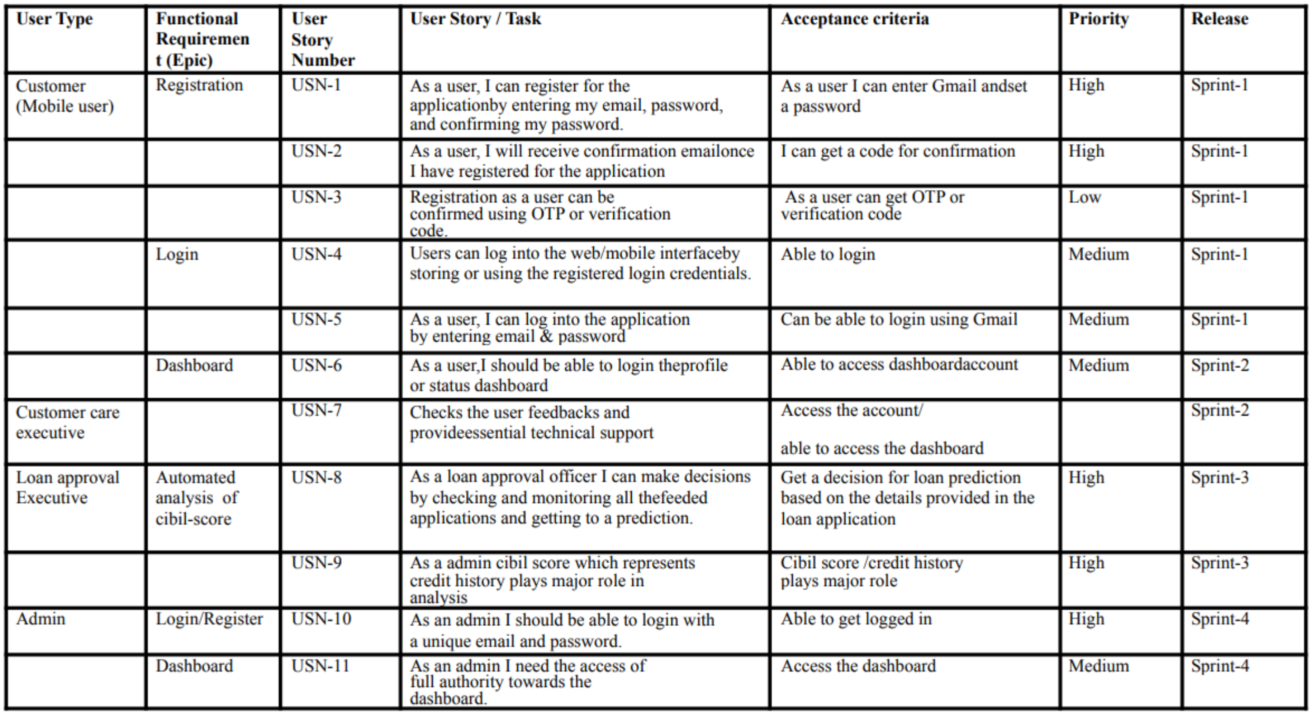


Technical Architecture



S.No	Component	Description	Technology
1.	UserInterface	Users interactwiththe application withthe helpof a web UI	HTML, CSS etc.
2.	Buildingapplication	GettinguserinformationfromUIandfeeding itto MLmodel	PythonFlask
3.	Visualizingandanalysingdata	Readingandunderstanding the data properly with the helpof visualizationandanalyzingtechniques.	Pythonpandas, numpy, pickle, matplotlib, seaborn
4.	Pre-processing orcleaningdata	Handling missing values, Handlingcategorical data, Handlingoutliers,Scaling Techniques	Pythonpandas
5.	Database	LoanApprovaldataset	.csvfile
6.	CloudDatabase	Deployingthe modeloncloud	IBMcloud
7.	Machine Learning Model	Using machine learning modelforpredicting loan approval	Modelbuildingusingclassification algorithms suchas Decisiontree, Randomforest, KNN, andxgboost.

5.3 User Stories



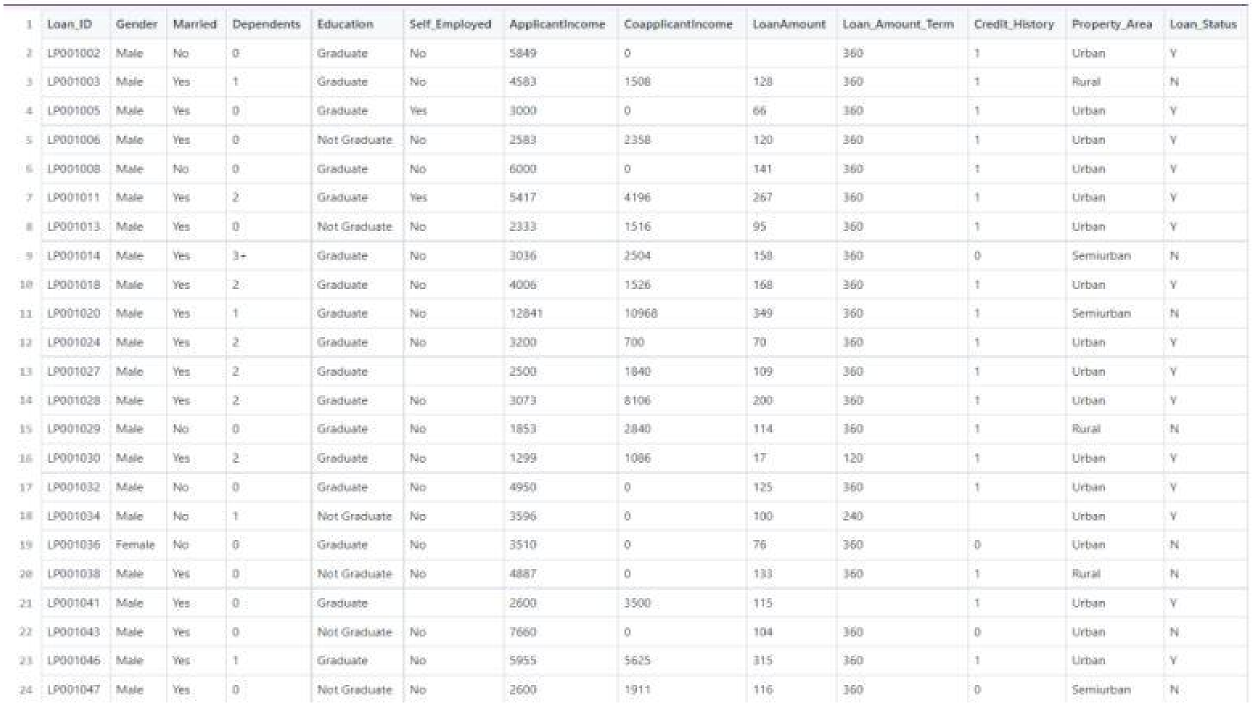
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the applicationby entering my email, password, and confirming my password.	As a user I can enter Gmail andset a password	High	Sprint-1
		USN-2	As a user, I will receive confirmation emailonce I have registered for the application	I can get a code for confirmation	High	Sprint-1
		USN-3	Registration as a user can be confirmed using OTP or verification code.	As a user can get OTP or verification code	Low	Sprint-1
	Login	USN-4	Users can log into the web/mobile interfaceby storing or using the registered login credentials.	Able to login	Medium	Sprint-1
		USN-5	As a user, I can log into the application by entering email & password	Can be able to login using Gmail	Medium	Sprint-1
	Dashboard	USN-6	As a user,I should be able to login theprofile or status dashboard	Able to access dashboardaccount	Medium	Sprint-2
Customer care executive		USN-7	Checks the user feedbacks and provideessential technical support	Access the account/ able to access the dashboard		Sprint-2
Loan approval Executive	Automated analysis of cibil-score	USN-8	As a loan approval officer I can make decisions by checking and monitoring all thefeeded applications and getting to a prediction.	Get a decision for loan prediction based on the details provided in the loan application	High	Sprint-3
		USN-9	As a admin cibil score which represents credit history plays major role in analysis	Cibil score /credit history plays major role	High	Sprint-3
Admin	Login/Register	USN-10	As an admin I should be able to login with a unique email and password.	Able to get logged in	High	Sprint-4
	Dashboard	USN-11	As an admin I need the access of full authority towards the dashboard.	Access the dashboard	Medium	Sprint-4

6.1 SPRINT PLANNING AND ESTIMATION:

SPRINT 1:

A sprint is an Agile methodology that helps you to complete a set amount of work in a timeboxed period. In this, we have done our data preprocessing and loading a dataset, leading to splitting datasets into train sets and test sets. This process is explained as follows.

1) Dataset:



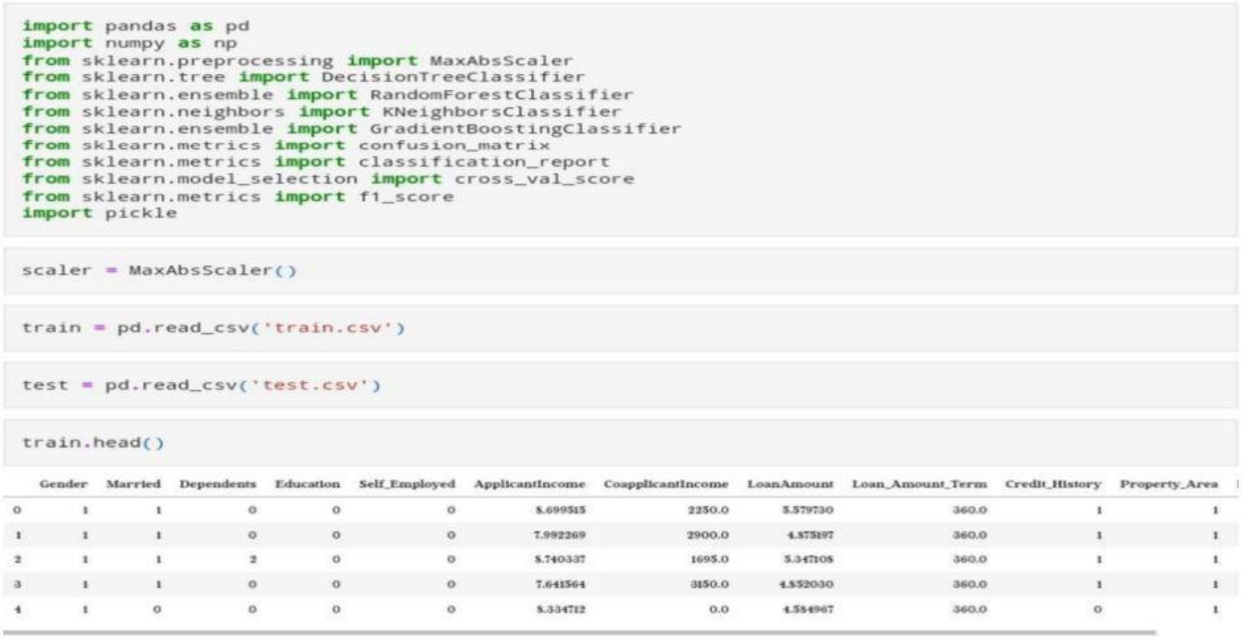
1	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
2	LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
3	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
4	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
5	LP001006	Male	Yes	0	Not Graduate	No	2583	2358	120	360	1	Urban	Y
6	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
7	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
8	LP001013	Male	Yes	0	Not Graduate	No	2333	1516	95	360	1	Urban	Y
9	LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	Semiurban	N
10	LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Urban	Y
11	LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	Semiurban	N
12	LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Urban	Y
13	LP001027	Male	Yes	2	Graduate		2500	1840	109	360	1	Urban	Y
14	LP001028	Male	Yes	2	Graduate	No	3073	8106	200	360	1	Urban	Y
15	LP001029	Male	No	0	Graduate	No	1853	2840	114	360	1	Rural	N
16	LP001030	Male	Yes	2	Graduate	No	1299	1086	17	120	1	Urban	Y
17	LP001032	Male	No	0	Graduate	No	4950	0	125	360	1	Urban	Y
18	LP001034	Male	No	1	Not Graduate	No	3596	0	100	240		Urban	Y
19	LP001036	Female	No	0	Graduate	No	3510	0	76	360	0	Urban	N
20	LP001038	Male	Yes	0	Not Graduate	No	4887	0	133	360	1	Rural	N
21	LP001041	Male	Yes	0	Graduate		2600	3500	115		1	Urban	Y
22	LP001043	Male	Yes	0	Not Graduate	No	7660	0	104	360	0	Urban	N
23	LP001046	Male	Yes	1	Graduate	No	5955	5625	315	360	1	Urban	Y
24	LP001047	Male	Yes	0	Not Graduate	No	2600	1911	116	360	0	Semiurban	N

UNDERSTANDING THE FEATURES OF DATASET:
ANALYSIS OF CATEGORICAL DATA:
THE CHOICE OF AN BETTER DATASET TO TRAIN:
HANDLING THE DATASET WITH NULL VALUES
TESTING THE DATASET

SPRINT 2:

MODEL BUILDING:

Setting up methods for data collection, understanding and paying attention to what is significant in the data to address the questions you are posing, and finding a simulation, statistical, or mathematical model to gain understanding and make predictions are all part of the model-building Process.



```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MaxAbsScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
from sklearn.metrics import f1_score
import pickle
```

```
scaler = MaxAbsScaler()
```

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

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

```
train.head()
```

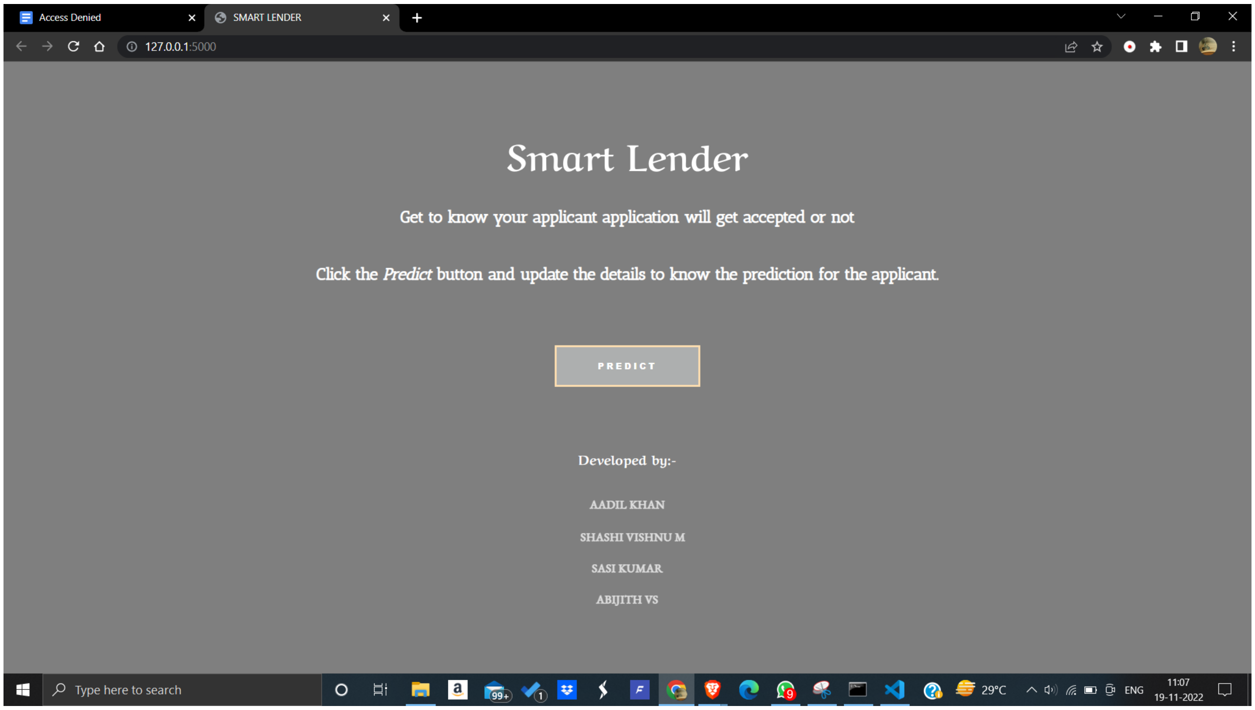
	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area
0	1	1	0	0	0	8.699515	2250.0	5.579730	360.0	1	1
1	1	1	0	0	0	7.992269	2900.0	4.875197	360.0	1	1
2	1	1	2	0	0	8.740337	1695.0	5.347108	360.0	1	1
3	1	1	0	0	0	7.641564	3150.0	4.852030	360.0	1	1
4	1	0	0	0	0	8.334712	0.0	4.584967	360.0	0	1

Testing & Training

The train/test approach is a way to gauge how accurate your model is. Because the data set is divided into two sets—a training set and a testing set—this technique is known as train/test.

SPRINT 3:

- **Web UI**
  HTML
  CSS



Smart Lender

Get to know your applicant application will get accepted or not

Click the *Predict* button and update the details to know the prediction for the applicant.

PREDICT

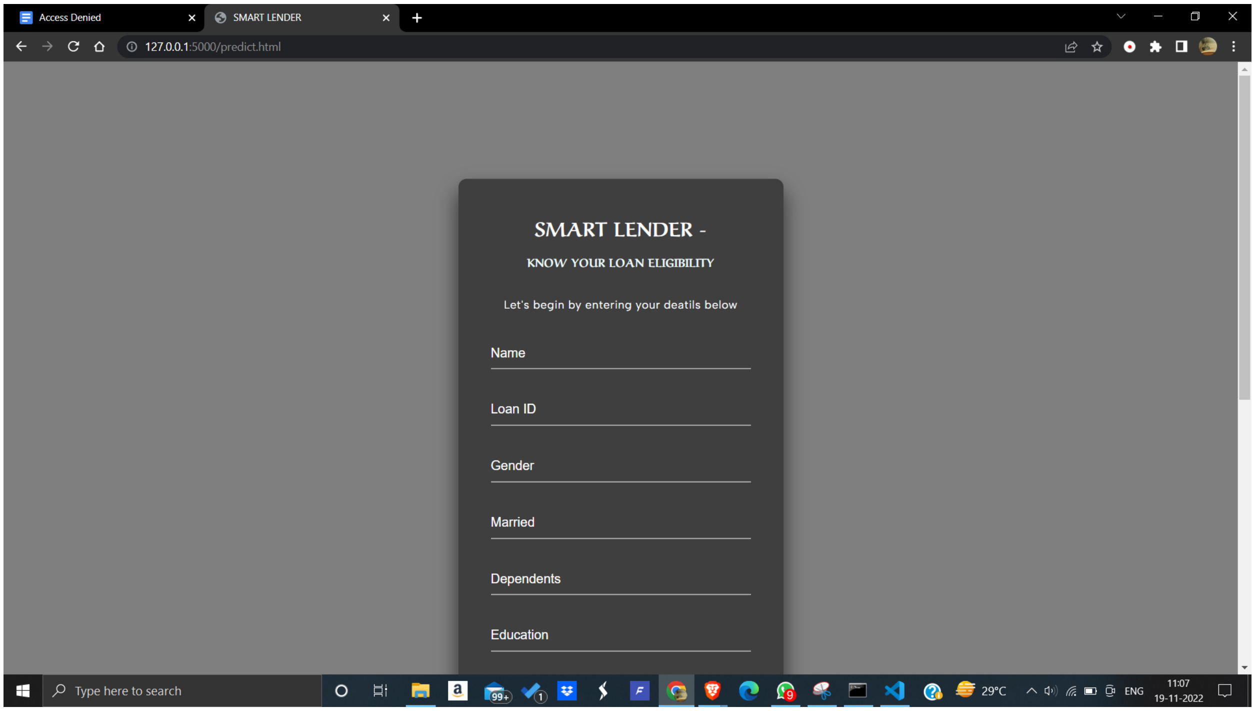
Developed by:-

AADIL KHAN

SHASHI VISHNU M

SASI KUMAR

ABIJITH VS



SMART LENDER -

KNOW YOUR LOAN ELIGIBILITY

Let's begin by entering your deatils below

Name

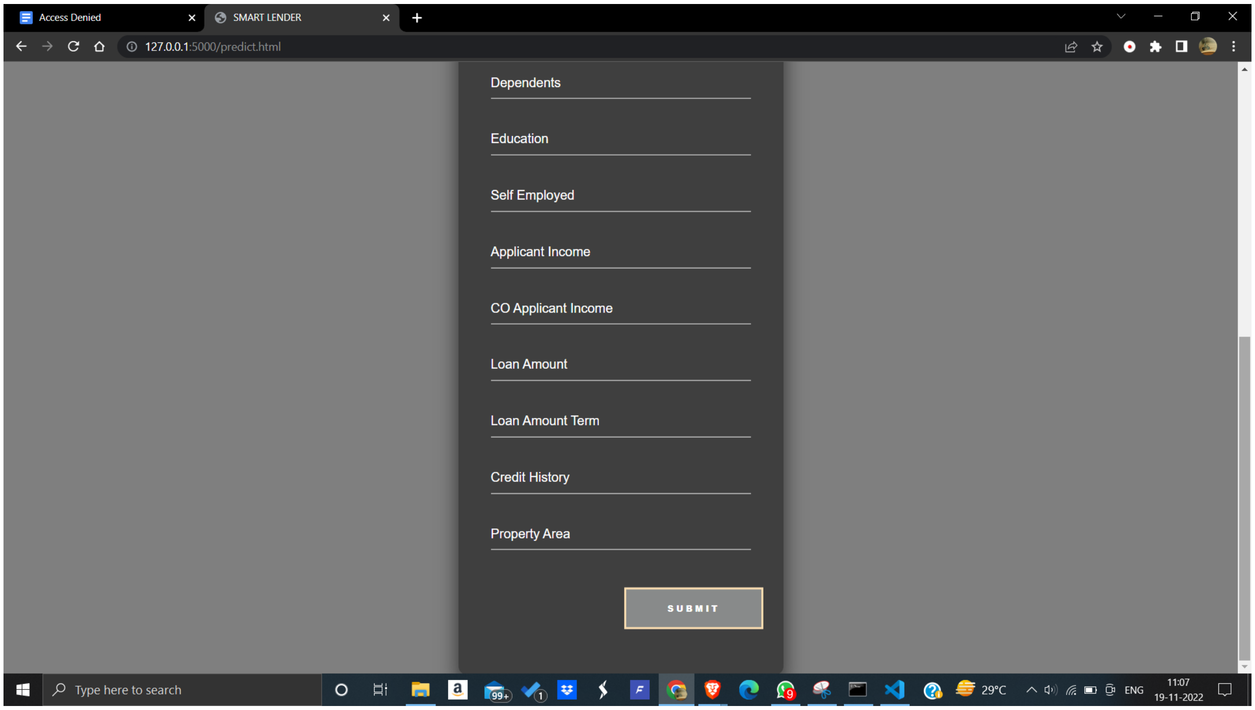
Loan ID

Gender

Married

Dependents

Education



Dependents

Education

Self Employed

Applicant Income

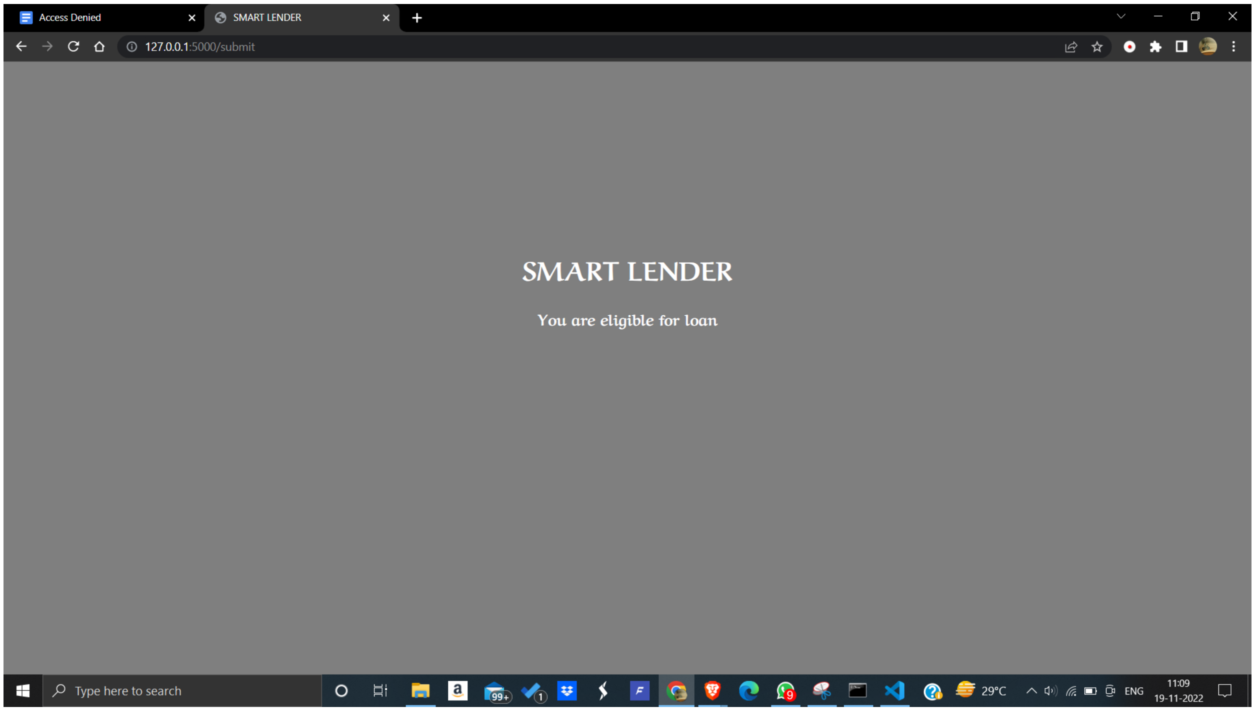
CO Applicant Income

Loan Amount

Loan Amount Term

Credit History

Property Area

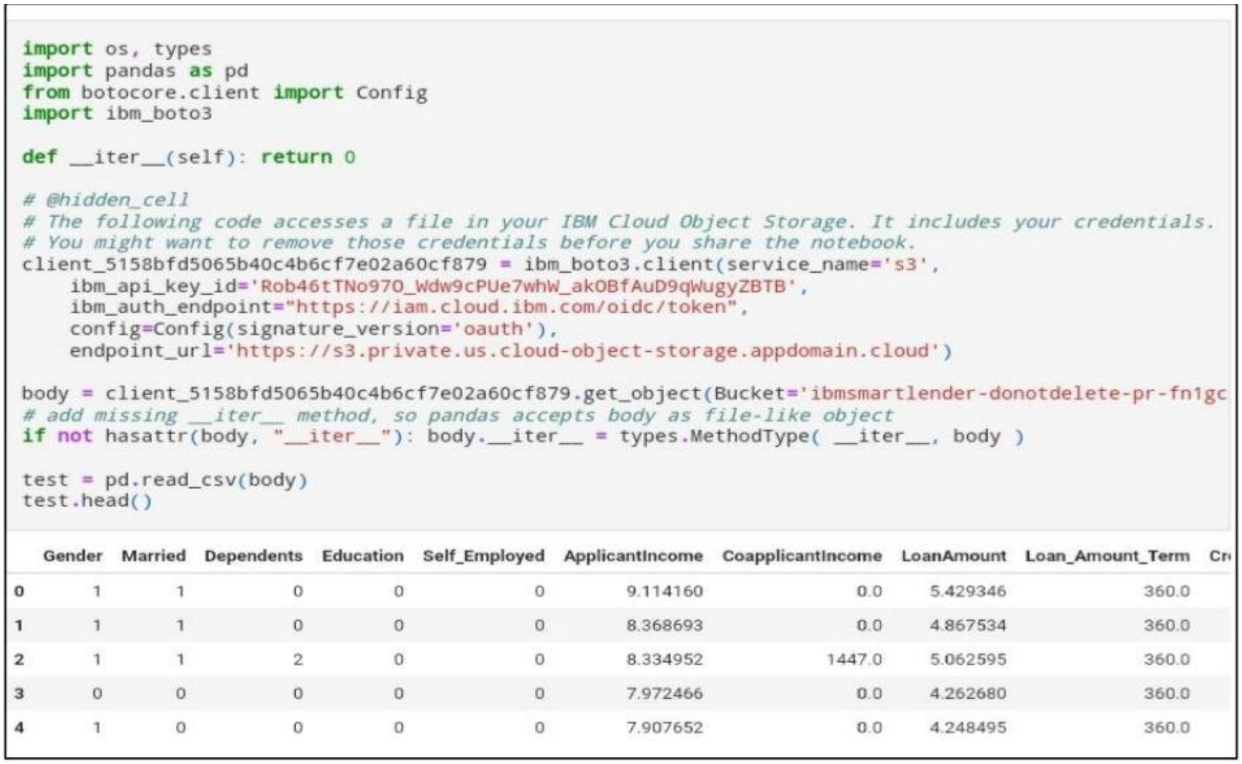


SMART LENDER

You are eligible for loan

SPRINT 4:

Cloud deployment is the process of deploying an application through one or more hosting models—software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS)—that leverage the cloud. This includes architecting, planning, implementing, and operating workloads on the cloud.



```python
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_5158bfd5065b40c4b6cf7e02a60cf879 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='Rob46tTNo970_Wdw9cPUe7whW_akOBfAuD9qWugyZBTB',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

body = client_5158bfd5065b40c4b6cf7e02a60cf879.get_object(Bucket='ibmsmartlender-donotdelete-pr-fn1gc
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

test = pd.read_csv(body)
test.head()
```

```
   Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  Cr
0       1        1           0          0              0         9.114160                0.0    5.429346             360.0
1       1        1           0          0              0         8.368693                0.0    4.867534             360.0
2       1        1           2          0              0         8.334952             1447.0    5.062595             360.0
3       0        0           0          0              0         7.972466                0.0    4.262680             360.0
4       1        0           0          0              0         7.907652                0.0    4.248495             360.0
```

7.1 CODING & SOLUTIONING

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df = pd.read_csv('loan_prediction.csv')
df.head(10)
df.describe()
df.isnull().any()
df.drop('Loan_ID',axis=1,inplace=True)
df.Property_Area.unique()
plt.figure(figsize=(15,7))
df['ApplicantIncome'].hist(bins=25)
plt.show()
df.boxplot(column='ApplicantIncome',figsize=(15,7))
df.boxplot(column='ApplicantIncome', by = 'Education',figsize=(15,7))
df['Property_Area'].value_counts()
```

Analysis of Categorical Values

```
df['Loan_Status'].value_counts()['Y']
pd.crosstab(df ['Credit_History'], df ['Loan_Status'], margins=True)
plt.figure(figsize=(15,7))
df['ApplicantIncome'].hist(bins=20)
plt.show()
df['ApplicantIncome'] = np.log(df['ApplicantIncome'])
plt.figure(figsize=(15,7))
df['ApplicantIncome'].hist(bins=25)
plt.show()
def percentageConvert(ser):
 return ser/float(ser[-1])
tabs = pd.crosstab(df ["Credit_History"], df ["Loan_Status"], margins=True).apply(percentageConvert, axis=1)
tabs
app_loan = tabs['Y'][1]
print(f'{app_loan*100:.2f} % applicants got their loans approved')
```

So this is a good data set to train with

```
df['Self_Employed'].fillna('No',inplace=True)
#df['TotalIncome'] = df['AppplicantIncome'] + df['CoapplicantIncome']
#df['TotalIncome_log'] = np.log(df['TotalIncome'])
#plt.figure(figsize=(15,7))
#df['TotalIncome'].hist(bins=25)
#plt.show()
#plt.figure(figsize=(15,7))
```

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

```
df['LoanAmount'] = np.log(df['LoanAmount'])
plt.figure(figsize=(15,7))
df['LoanAmount'].hist(bins=25)
plt.show()
plt.figure(figsize=(15,7))
df['ApplicantIncome'].hist(bins=20)
plt.show()
df['ApplicantIncome'] = np.log(df['ApplicantIncome'])
plt.figure(figsize=(15,7))
df['ApplicantIncome'].hist(bins=25)
plt.show()
plt.figure(figsize=(15,7))
df['Loan_Amount_Term'].hist(bins=20)
plt.show()
#df['ApplicantIncome'] = np.log(df['ApplicantIncome'])
#plt.figure(figsize=(15,7))
#df['ApplicantIncome'].hist(bins=25)
#plt.show()
#df.drop('LoanAmount',axis=1,inplace=True)
#df.drop('TotalIncome',axis=1,inplace=True)
df.head()
```

Now to Handle with null values

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

```
cat=['Gender','Married','Dependents','Education','Self_Employed','Credit_History','Property_Area']
target = ['Loan_Status']
all_cols = ['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
 'ApplicantIncome', 'CoapplicantIncome', 'Loan_Amount_Term',
 'Credit_History', 'Property_Area', 'Loan_Status', 'TotalIncome_log',
 'LoanAmount_log']
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
for var in cat:
 le = LabelEncoder()
 df[var]=le.fit_transform(df[var].astype('str'))
print('Done encoding Catergorical Values')
for tar in target:
 oe = OneHotEncoder()
 df[tar]=le.fit_transform(df[tar].astype('str'))
print('Done encoding Target Value')
df.head(5)
from sklearn.model_selection import train_test_split
train, test = train_test_split(df,test_size=0.2,random_state=42)
test.to_csv('test.csv',encoding='utf-8',index=False)
train.to_csv('train.csv',encoding='utf-8',index=False)
```

8.ADVANTAGES

- The customer can predict their eligibility from any part of the world and at any time so it provides user convenience
- Eligible applicant will be sanctioned loan without any delay
- Minimal documentation is required and there is no physical submission of documents
- Whole process will be automated,so human error will be avoided
- Time period for loan sanctioning will be reduced and more Accurate prediction for loan eligibility will be given.
- The customer can contact bank at any time in case of any queries and we had also provided the detailed procedure for applying loan and customer can also provide the ratings.

9.DISADVANTAGES

- The customer can contact the lender only through online using email or call them in case of any queries
- The bank should externally connect to database and use this software in real time we had provided only the feature
- There may be some risk associated with security of the customers as they are providing all their details in online
- The Accuracy of prediction can also be improved

10.CONCLUSION

The analysis has started from data preprocessing ,handling missing value, exploratory analysis and different models were build like Decision tree model,KNN model,Xgboost model and Random Forest model and there performance were evaluated , as a result the Random Forest model is selected as the best model for predicting the loan approval status of the customer after evaluating its performance ,as it got 91% accuracy in prediction.This application is then tested and it functions properly and it also meets all the requirements of the bank in selecting the trust worthy person to provide loan.

11. Future scope

In future,payment option can be included in this application for exchanging money between the lender and borrower and bank can verify the customer document online using AI which makes the process of verification simpler and could be made more secure,trustworthy and dynamic weight conformation and in near future this module can be integrated with the module of automated processing system.

GitHub and Project Demo Link

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-23661-1659891274/

Demo Link: https://www.youtube.com/watch?v=O3AMVR3DEss