

**SMART LENDER - APPLICANT CREDIBILITY
PREDICTION FOR LOAN APPROVAL**

TEAM ID :PNTIMD03672

HEAMANTH R

KISHORE KUMAR R

MOHAMED MUSTAQ M

SHANKARA NARAYANAN M

RAMKUMARAN S

1.1.

PROJECT OVERVIEW

The expenses required for people's daily life is increasing day by day and amount required to build a house or assets or start a business is very high. The savings alone isn't enough to do the necessary. The easiest way to get the required funds is to apply for a loan. But process of getting a loan is a very time consuming process. The Application has to go through a lot of stages and it's still not necessary that it will be approved. We have created this project to make it simple for the people and minimise the time required for loan approval.

Bank credit is the total amount of funds a person or business can borrow from a financial institution. Credit approval is determined by a borrower's credit rating, income, collateral, assets, and pre-existing debt. The process of bank credit risk evaluation is recognized at banks across the globe. Credit risk is one of the main functions of the banking community. The prediction of credit defaulters is one of the difficult tasks for any bank. But by forecasting the loan defaulters, the banks definitely may reduce their loss by reducing their non-profit assets, so that recovery of approved loans can take place without any loss and it can play as the contributing parameter of the bank statement. This makes the study of this loan approval prediction important. Machine Learning techniques are very crucial and useful in the prediction of these types of data. Decision tree, Random Forest, KNN, and xgboost algorithms are used. We will train and test the data with these algorithms. From this best model is selected and saved in pkl format. We have also done Flask integration and IBM deployment.

1.2.

PURPOSE

- Knowledge of Machine Learning Algorithms.
- Knowledge of Python Language with Machine Learning
- You'll be able to understand the problem to classify if it is a regression or a classification kind of problem.
- You will be able to know how to pre-process/clean the data using different data preprocessing techniques.
- Applying different algorithms according to the dataset and based on visualization.
- Real-time Analysis of Project
- Building ease of User Interface (UI)
- Navigation of ideas towards other projects (creativity)
- Knowledge of building ML models.
- How to build web applications using the Flask framework.

2. LITERATURE SURVEY

2.1.EXISTING PROBLEM

- Human errors often occur during manual verification hence, lacking in accuracy. A lot of capital investment is also involved for the labour
- Low accuracy of manual credibility inspection which leads to misinterpretation of fraudulent loan applicants as credible ones and vice versa.
- It is a time consuming process. A lot of labour is required for this task.

2.2.

REFERENCES

- [1] B. P. Lohani, M. Trivedi, R. J. Singh, V. Bibhu, S. Ranjan and P. K. Kushwaha, "Machine Learning Based Model for Prediction of Loan Approval," 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), 2022, pp. 465-470, doi: 10.1109/ICIEM54221.2022.9853160.
- [2] Soni PM, Varghese Paul, "Algorithm For the Loan Credibility Prediction System", International Journal of Recent Technology and Engineering (IJRTE), Volume-8, Issue-1S4, June 2019.
- [3] Okfalisa, R. Fitriani and Y. Vitriani, "The Comparison of Linear Regression Method and K-Nearest Neighbors in Scholarship Recipient," 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2018, pp. 194-199, doi: 10.1109/SNPD.2018.8441068.
- [4] Kathe Rutika Pramod, Panhale Sakshi Dattatray, Avhad Pooja Prakash, "An Approach For Prediction Of Loan Approval using Machine Learning Algorithm", International Journal of Creative Research Thoughts (IJCRT), Volume 9, Issue 6, June 2021.
- [5] Ambika, & Biradar, Santosh. (2021). Survey on Prediction of Loan Approval Using Machine Learning Techniques. International Journal of Advanced Research in Science, Communication and Technology. 449-454.

10.48175/IJARSCT-1165.

[6] Yash Divate, Prashant Rana, Pratik Chavan, "Loan Approval Prediction Using Machine Learning", International Research Journal of Engineering and Technology (IRJET), Volume: 08 Issue: 05, May 2021.

[7] Anant Shinde, Yash Patil, Ishan Kotian, Abhinav Shinde, Reshma Gulwani, "Loan Prediction System Using Machine Learning", International Conference on Automation, Computing and Communication 2022 (ICACC-2022), Volume 44, May 2022.

[8] Q. Du, N. Li, S. Yang, D. Sun and W. Liu, "Integrating KNN and Gradient Boosting Decision Tree for Recommendation," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2021, pp. 2042-2049, doi: 10.1109/IAEAC50856.2021.9390647.

2.3.

PROBLEM STATEMENT DEFINITION

- The prediction of credit defaulters is one of the difficult tasks for any bank.
- Machine Learning techniques are very crucial and useful in the prediction of these types of data. Classification algorithms such as Decision tree, Random forest, KNN, and xgboost can be utilized to serve this purpose.
- A model must be trained using a dataset to predict the credibility of an applicant accurately.

3.3 PROPOSED SOLUTION

Proposed Solution Template:

Project team shall fill the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Howard is a businessman. He wants to build his own house. He doesn't have enough savings to build a house so he needs a personal loan from the bank. He wants to know whether his credit is enough to get his loan approved.
2.	Idea / Solution Description	Users can get knowledge about the loan process from the app and also apply for loans from the App itself.
3.	Novelty / Uniqueness	An automated customer support system will help the user and guide them through the loan approval process.
4.	Social Impact / Customer Satisfaction	Customers can use the Loan amount to improve their business which in turn impacts the economy of the country.
5.	Business Model (Revenue Model)	Can monetize features like viewing multiple banks or applying for multiple banks or we can also have subscriptions once we hit a certain user rate.
6.	Scalability of the Solution	This will provide access to people across the country to approach the banks, and also helps users in remote locations to access the bank's loan approval process.

3.4 PROBLEM SOLUTION FIT

Define CS, fit into CC Focus on J&P, tap into BE, understand RC	1. CUSTOMER SEGMENT(S) CS Who is your customer? 1. People who need loan 2. People who have Bank Account 3. People who needs to get their Loan approved.	6. CUSTOMER CONSTRAINTS CC What constraints prevent your customers from taking action or limit their choices of solutions? 1. Complicated Process 2. Not easily accessible 3. Too much of formalities	5. AVAILABLE SOLUTIONS AS Which solutions are available to the customers when they face the problem? or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? 1. Pre-existing Bank Loan Applying Apps 2. Websites to check credit score 3. Loan Agents	Explore AS, differen Focus on J&P, tap into BE, understand RC
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. 1. Data availability problem 2. Accurate information	9. PROBLEM ROOT CAUSE RC What is the real reason that this problem exists? What is the back story behind the need to do this job? 1. Credit worthiness of an individual or an organisation is highly important before sanctioning the loan 2. We need to verify the borrower before sanctioning the loan	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? 1. Customer should be aware of his credibility to apply for the loan. 2. Customer should do their research on repayments of the loan and choose their affordable rate of interest on their repayments	

Identify strong TR & CH	3. TRIGGERS TR What triggers customers to act? 1. Poor communication between Bank and the Customer during loan process 2. Lack of open information about credibility available to the user.	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior. 1. Users can get knowledge about the loan process from the app and also apply for loans from the app itself. 2. An automated customer support system will help the user and guide them through the loan approval process.	8. CHANNELS of BEHAVIOR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 1. Apply for loan, track their loan process, check their credibility 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. 1. Submit the hardcopy of the required documents	

4. REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	User registers on the website using his google account or Email ID.
FR-2	User Confirmation	Through one time verification and using captcha etc.
FR-3	User Requirement	User need to update their bank details to check their credibility
FR-4	Profile Updation	Users can update their profile when it's required.
FR-5	User Support	User Support ISD provided by an user friendly Interface
FR-6	User Authentication	By OTP or verification code the user gets authenticated and OTP is used for mobile number registration.

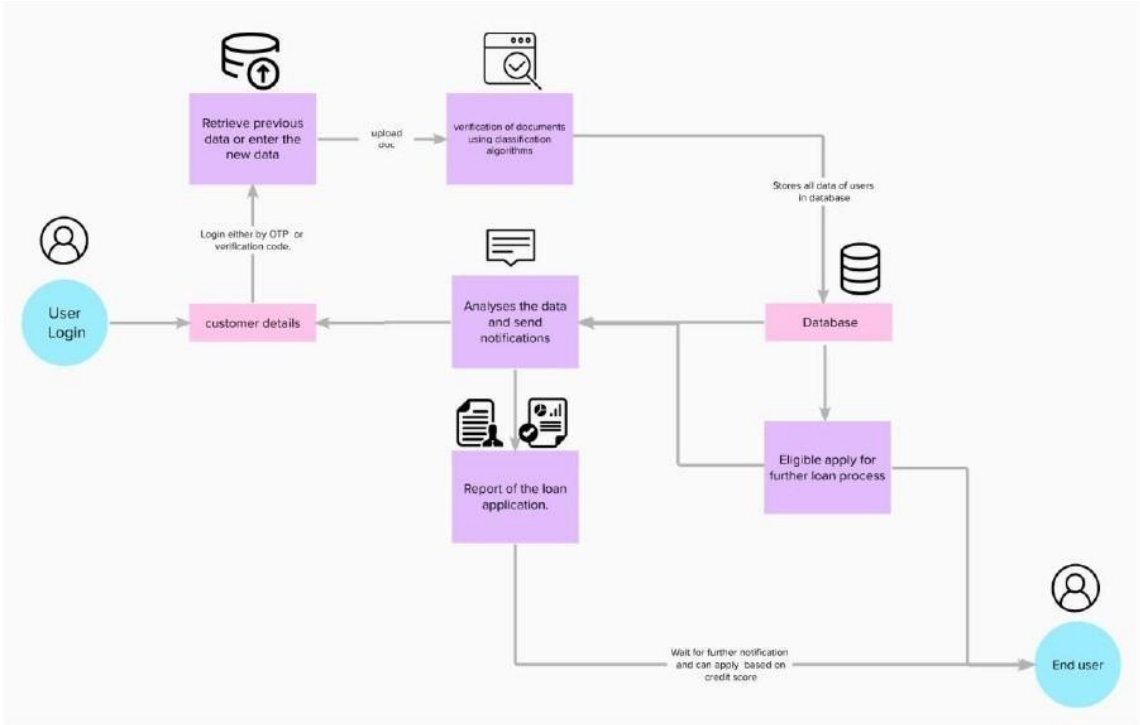
4.2. NON FUNCTIONAL REQUIREMENT

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Cibil score of the user is assessed using the app and predicts the eligibility for users to avail for loan approval
NFR-2	Security	Verification is done by using One Time Passwords and it uses enhanced security features over to update documents of user .
NFR-3	Reliability	Bugs will be detected and Fixes will be provided using App updates
NFR-4	Performance	The collected data is stored in a database for data retrieval.It will be able to handle traffic efficiently
NFR-5	Availability	Users can access this using any device with any operating system since it is platform independent .
NFR-6	Scalability	The loan eligibility checking will be accurate as it parses efficiently through all user documents.

5. PROJECT DESIGN

5.1. DATAFLOW DIAGRAMS



5.2 SOLUTION & TECHINICAL ARCHITECTURE

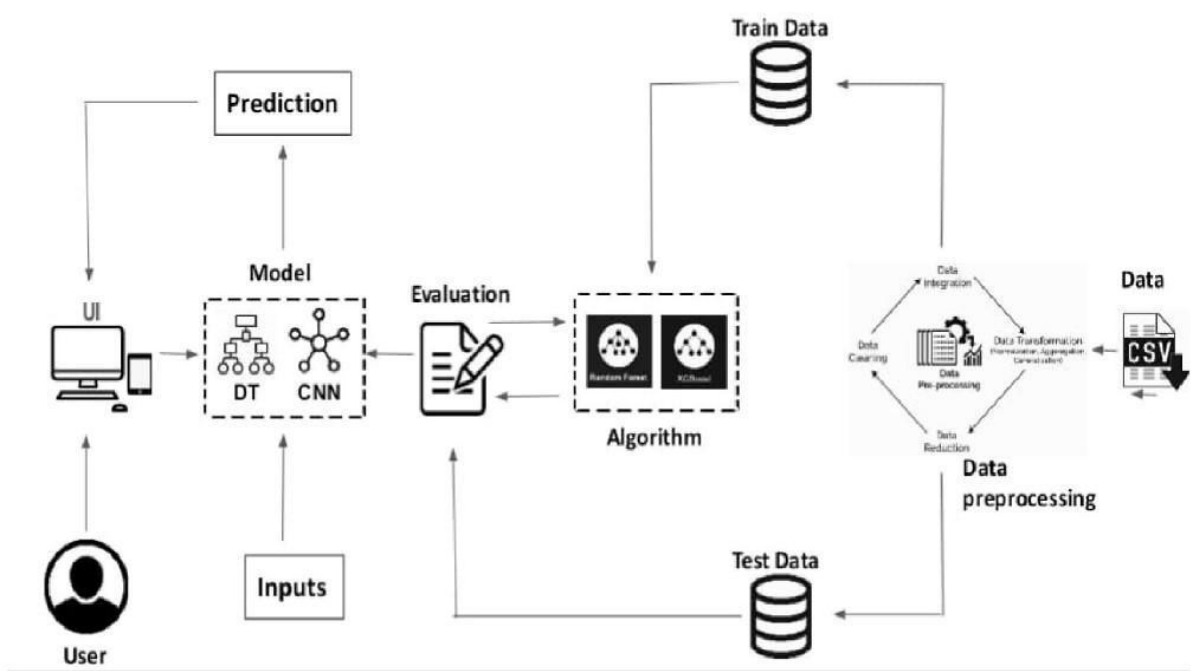


FIG 5.2.1 SOLUTION ARCHITECTURE

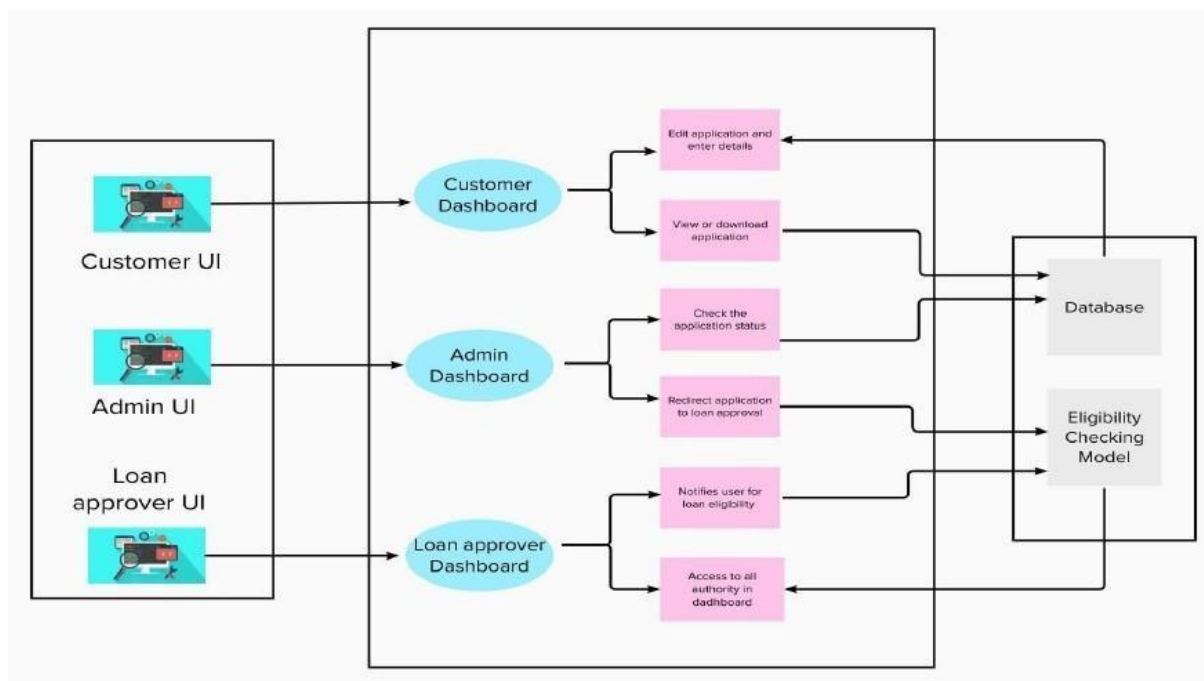


FIG 5.2.2 TECHNICAL ARCHITECTURE

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 application by using my email and password	As a user I can enter Gmail and set a password	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can get a code for confirmation	High	Sprint-1
		USN-3	As a user my verification will be done by confirming verification code sent to my mail	As a user can get OTP or verification code	Low	Sprint-1
	Login	USN-4	Login by the user into the interface will be doneby using the login credentials.	Being able to login	Medium	Sprint-1
		USN-5	As a user, I can enter my mail and password to login	Able to login using mail id	Medium	Sprint-1
	Dashboard	USN-6	As a user,I should be use the dashboard in my profile	Able to access dashboard account	Medium	Sprint-2
Customer care executive		USN-7	Analyses the feedbacks and provide technical support	Send Support request	Medium	Sprint-2
Bank Approval Officer	Cibil-score Checking	USN-8	As a Bank approval officer I can make decisions by anaysing the applicant details	Make a decision for loan approval based on the details provided in the loan application	High	Sprint-3
		USN-9	As a approval officer, cibil score of the applicant plays major role in analysis.	Cibil score plays the major role	High	Sprint-3
Admin	Login/Register	USN-10	As an admin I will be able to login with my email and password.	Able to get logged in	High	Sprint-4
	Dashboard	USN-11	As an admin can control and acces the Dashboard	Access and control the dashboard	Medium	Sprint-4

6. PROJECT PLANNING AND SCHEDULING

6.1. SPRINT PLANNING ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by using my email and password	2	High	MOHAMED MUSTAQ M HEAMANTH R
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	SHANKARA NARAYANAN M KISHORE KUMAR R
Sprint-1	Login	USN-3	As a user, I can enter my mail and password to login	2	Low	MOHAMED MUSTAQ M RAMKUMARAN S
Sprint-2	Dashboard	USN-4	As a user, I should be able to use the dashboard in my profile	3	High	RAMKUMARAN S HEAMANTH R
Sprint-2	Document Upload	USN-5	As a user, I can upload my documents	2	Low	KISHORE KUMAR R MOHAMED MUSTAQ M
Sprint-3	Verify User details	USN-6	As a Bank Administrator, I can view the user details	3	Medium	SHANKARA NARAYANAN M MOHAMED MUSTAQ M
Sprint-3	Cibil score verification	USN-7	As a Bank Approval administrator, I can verify the cibil score details	3	High	HEAMANTH R SHANKARA NARAYANAN M
Sprint-4	Document Validation	USN-8	As a Loan Administrator, I can verify all the documents and proofs uploaded by the user.	3	High	RAMKUMARAN S SHANKARA NARAYANAN M
Sprint-4	Loan Approval Process	USN-9	As a Bank approval officer I can make decisions by analyzing the applicant details	3	High	RAMKUMARAN S KISHORE KUMAR R
Sprint-4		USN-10	As an approval officer, cibil score of the applicant plays a major role in the analysis.	2	Medium	KISHORE KUMAR R HEAMANTH R

6.2. SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	5	6 Days	24 Oct 2022	29 Oct 2022	5	29 Oct 2022
Sprint-2	5	6 Days	31 Oct 2022	05 Nov 2022	5	05 Nov 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	6	12 Nov 2022
Sprint-4	8	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

7 CODING AND SOLUTIONING

7.1 VISUALIZING THE DATASET

DATASET USED FOR TRAINING

1	Loan_ID	Gender	Married st	Dependen	Education	Self_Empl	ApplicantI	Coapplicant	LoanAmou	Loan_Amc	Credit_His	Property_Area
2	LP001015	Male	Yes	0	Graduate	No	5720	0	110	360	1	Urban
3	LP001022	Male	Yes	1	Graduate	No	3076	1500	126	360	1	Urban
4	LP001031	Male	Yes	2	Graduate	No	5000	1800	208	360	1	Urban
5	LP001035	Male	Yes	2	Graduate	No	2340	2546	100	360		Urban
6	LP001051	Male	No	0	Not Gradu	No	3276	0	78	360	1	Urban
7	LP001054	Male	Yes	0	Not Gradu	Yes	2165	3422	152	360	1	Urban
8	LP001055	Female	No	1	Not Gradu	No	2226	0	59	360	1	Semiurban
9	LP001056	Male	Yes	2	Not Gradu	No	3881	0	147	360	0	Rural
10	LP001059	Male	Yes	2	Graduate		13633	0	280	240	1	Urban
11	LP001067	Male	No	0	Not Gradu	No	2400	2400	123	360	1	Semiurban
12	LP001078	Male	No	0	Not Gradu	No	3091	0	90	360	1	Urban
13	LP001082	Male	Yes	1	Graduate		2185	1516	162	360	1	Semiurban
14	LP001083	Male	No	3+	Graduate	No	4166	0	40	180		Urban
15	LP001094	Male	Yes	2	Graduate		12173	0	166	360	0	Semiurban
16	LP001096	Female	No	0	Graduate	No	4666	0	124	360	1	Semiurban

This is the dataset taken for training the model. It has 11 attributes namely Gender, Married, Dependents, Education, Self Employed, Applicant Income, Coapplicant Income, Loan Amount, Loan Amount Term, Credit History, Property Area.

DESCRIPTION OF THE DATASET

df.describe()					
	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

UNIVARIATE ANALYSIS

```
#plotting the using distplot
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['ApplicantIncome'], color='r')
plt.subplot(122)
sns.distplot(df['Credit_History'])
plt.show()
```

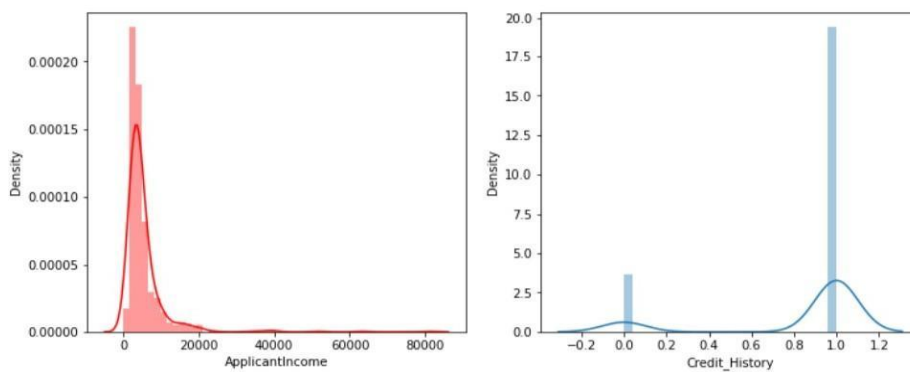


FIG. CODE & OUTPUT OF UNIVARIATE ANALYSIS

BIVARIATE ANALYSIS

```
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(df['Married'], hue=df['Gender'])
plt.subplot(132)
sns.countplot(df['Self_Employed'], hue=df['Education'])
plt.subplot(133)
sns.countplot(df['Property_Area'], hue=df['Loan_Amount_Term'])
```

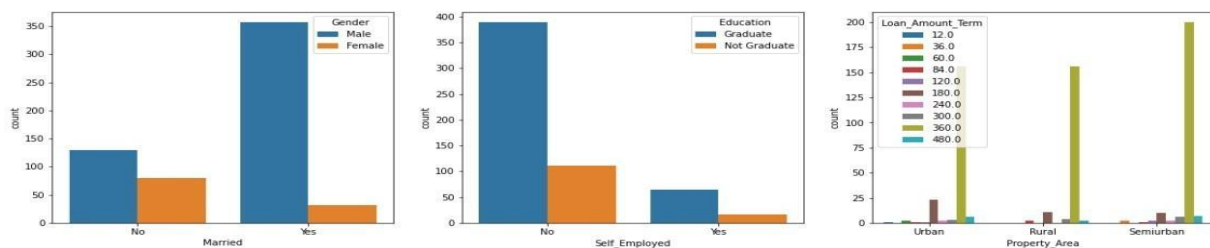


FIG . CODE & OUTPUT OF BIVARIATE ANALYSIS

MULTIVARIATE ANALYSIS

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

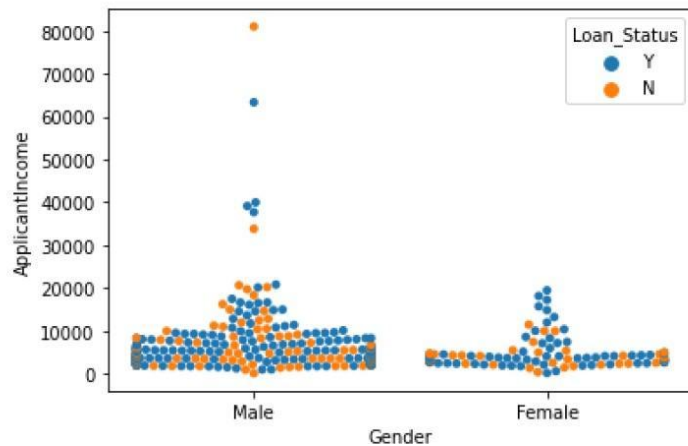


FIG. CODE & OUPUT MULTIVRIATE ANALYSIS

7.2 DATA PREPROCESSING

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Loan_ID               614 non-null   object
1   Gender                601 non-null   object
2   Married               611 non-null   object
3   Dependents            599 non-null   object
4   Education             614 non-null   object
5   Self_Employed         582 non-null   object
6   ApplicantIncome       614 non-null   int64
7   CoapplicantIncome     614 non-null   float64
8   LoanAmount            592 non-null   float64
9   Loan_Amount_Term      600 non-null   float64
10  Credit_History        564 non-null   float64
11  Property_Area         614 non-null   object
12  Loan_Status           614 non-null   object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

FIG. INFORMATION ABOUT DATASET

```

df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])
df['Married'] = df['Married'].fillna(df['Married'].mode()[0])
#replacing + with space for filling the nan values
df['Dependents']=df['Dependents'].replace('3+',3)
df['Dependents'] = df['Dependents'].fillna(df['Dependents'].mode()[0])
df['Self_Employed'] = df['Self_Employed'].fillna(df['Self_Employed'].mode()[0])
df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount']. mode()[0])
df['Loan_Amount_Term'] = df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mode()[0])
df['Credit_History'] = df['Credit_History'].fillna(df['Credit_History'].mode()[0])

```

FIG. REMOVING NULL VALUES IN DATASET

```

#Balancing the dfset by using smote
from imblearn.combine import SMOTETomek
smote = SMOTETomek (0.95)
y = df['Loan_Status']
x = df.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())

```

```

1    422
0    192
Name: Loan_Status, dtype: int64
1    352
0    330
Name: Loan_Status, dtype: int64

```

FIG. BALANCING THE DATASET

```

sc=StandardScaler()
x_bal_scaled=sc.fit_transform(x_bal)
x_bal_scaled = pd.DataFrame(x_bal_scaled,columns=x.columns)

```

FIG. SCALING THE DATASET

7.3. TRAINING THE MODELS

```
train,test = train_test_split(final_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=final_df.drop(["Loan_Status"],axis=1)
y=final_df.Loan_Status
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

FIG. SPLITTING THE DATASET

```
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 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
```

FIG. IMPORTING ML METHODS

Comparing the Models

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

***DecisionTreeClassifier***
Confusion matrix
[[53 18]
 [12 56]]
Classification report

```

	precision	recall	f1-score	support
0	0.82	0.75	0.78	71
1	0.76	0.82	0.79	68
accuracy			0.78	139
macro avg	0.79	0.79	0.78	139
weighted avg	0.79	0.78	0.78	139

```
score
0.7841726618705036
```

FIG. DECISION TREE CLASSIFIER

```

randomforest(x_train, x_test, y_train, y_test)

***RandomForestClassifier***
Confusion matrix
[[48 23]
 [ 3 65]]
Classification report

```

	precision	recall	f1-score	support
0	0.94	0.68	0.79	71
1	0.74	0.96	0.83	68
accuracy			0.81	139
macro avg	0.84	0.82	0.81	139
weighted avg	0.84	0.81	0.81	139

```

score
0.8129496402877698

```

FIG. RANDOM FOREST CLASSIFIER

```

KNN(x_train, x_test, y_train, y_test)

***KNeighborsClassifier***
Confusion matrix
[[50 21]
 [21 47]]
Classification report

```

	precision	recall	f1-score	support
0	0.70	0.70	0.70	71
1	0.69	0.69	0.69	68
accuracy			0.70	139
macro avg	0.70	0.70	0.70	139
weighted avg	0.70	0.70	0.70	139

```

score
0.697841726618705

```

FIG. K NEIGHBOUR CLASSIFIER

```

xgboost(x_train, x_test, y_train, y_test)

***Gradient BoostingClassifier***
Confusion matrix
[[46 25]
 [ 4 64]]
Classification report

```

	precision	recall	f1-score	support
0	0.92	0.65	0.76	71
1	0.72	0.94	0.82	68
accuracy			0.79	139
macro avg	0.82	0.79	0.79	139
weighted avg	0.82	0.79	0.79	139

```

score
0.7913669064748201

```

FIG. GRADIENT BOOST CLASSIFIER

Finalising the ML Model

The RandomForestClassifier model is finalised as the model based on the comparison scores and it is trained and exported as pkl file.

```
from sklearn.model_selection import cross_val_score
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
yPred = rf.predict(x_test)
f1_score(yPred,y_test, average='weighted')
cv = cross_val_score(rf,x,y,cv=5)
np.mean(cv)

0.7998331769367115

pickle.dump(rf,open('rdf.pkl','wb'))
```

FIG. EXPORTING THE TRAINED MODEL

7.4. Deploying the model in IBM Cloud

```
!pip install -U ibm-watson-machine-learning

Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.255)
Collecting ibm-watson-machine-learning
  Downloading ibm_watson_machine_learning-1.0.257-py3-none-any.whl (1.8 MB)
    | 1.8 MB 31.6 MB/s eta 0:00:01
```

FIG. INSTALLING NECESSARY PACKAGES

Here the required library of IBM Watson Machine Learning is getting installed.

```

In [20]: from ibm_watson_machine_learning import APIClient
import json

In [21]: wml_credentials = {
    "apikey" : "yJjRKquYkWeXxU3CGuvFCQq1f29M8lpidj6PZ8B0RQgY",
    "url" : "https://eu-gb.ml.cloud.ibm.com"
}

In [22]: wml_client = APIClient(wml_credentials)

In [24]: wml_client.spaces.list()

Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
-----
ID                                NAME                                CREATED
99b1a4a9-7cc5-4852-9388-f8907fe20de7  sl_223_ibm_space  2022-11-16T10:44:29.321Z
-----

In [25]: SPACE_ID= "99b1a4a9-7cc5-4852-9388-f8907fe20de7"

In [26]: wml_client.set.default_space(SPACE_ID)

Out[26]: 'SUCCESS'

```

FIG. AUTHENTICATION AND SPACE SETTING

Using the unique API key generated in IBM Cloud and mentioning our server location. Using the API credentials a new space is created in IBM Watson. The space has its unique Space id.

```

In [28]: import sklearn
sklearn.__version__

Out[28]: '1.0.2'

In [29]: MODEL_NAME = 'Model_building_SL_223_IBM'
DEPLOYMENT_NAME = 'Smart-Lender_223_IBM'
DEMO_MODEL = rf

In [30]: software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')

In [31]: model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}

In [32]: model_details = wml_client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
    training_data=x_train,
    training_target=y_train
)

```

FIG. IMPORTING MODEL AND SETTING IT UP

Downloading the required ML model. Looking for the version that is being supported by IBM and downloading the correct version. Creating a new deployment space for the model. To set up the model requirements and link it to the deployment space. Saving the model to the space by mentioning the attributes of the model.


```
In [33]: model_details
```

```
Out[33]: {'entity': {'hybrid_pipeline_software_specs': [],
  'label_column': 'Loan_Status',
  'schemas': {'input': [{'fields': [{'name': 'Gender', 'type': 'int64'},
    {'name': 'Married', 'type': 'int64'},
    {'name': 'Dependents', 'type': 'int64'},
    {'name': 'Education', 'type': 'int64'},
    {'name': 'Self_Employed', 'type': 'int64'},
    {'name': 'ApplicantIncome', 'type': 'int64'},
    {'name': 'CoapplicantIncome', 'type': 'int64'},
    {'name': 'LoanAmount', 'type': 'int64'},
    {'name': 'Loan_Amount_Term', 'type': 'int64'},
    {'name': 'Credit_History', 'type': 'int64'},
    {'name': 'Property_Area', 'type': 'int64'}]},
    'id': '1',
    'type': 'struct'}],
  'output': []},
  'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
    'name': 'runtime-22.1-py3.9'},
  'type': 'scikit-learn_1.0'},
  'metadata': {'created_at': '2022-11-16T10:48:43.132Z',
    'id': '03542d22-55b9-4830-af6f-c000da875e4e',
    'modified_at': '2022-11-16T10:48:46.959Z',
    'name': 'Model_building_SL_223_IBM',
    'owner': 'IBMid-6620042VBA',
    'resource_key': 'cdb1c157-cfd2-4271-a4a5-9f28198439ca',
    'space_id': '99b1a4a9-7cc5-4852-9388-f8907fe20de7'},
  'system': {'warnings': []}}
```

FIG. MODEL DETAILS

```
In [34]: model_id = wml_client.repository.get_model_id(model_details)
         model_id
```

```
Out[34]: '03542d22-55b9-4830-af6f-c000da875e4e'
```

```
In [37]: deployment_props = {
         wml_client.deployments.ConfigurationMetaNames.NAME: DEPLOYMENT_NAME,
         wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
       }
```

```
In [38]: deployment = wml_client.deployments.create(
         artifact_uid=model_id,
         meta_props=deployment_props
       )
```

```
=====
Synchronous deployment creation for uid: '03542d22-55b9-4830-af6f-c000da875e4e' started
=====

initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.

ready

-----
Successfully finished deployment creation, deployment_uid='04569608-2735-48f9-991a-fb61d5a0c2af'
-----
```

FIG. DEPLOYMENT ON IBM CLOUD

To set the configuration of the deployment. Giving the name for the deployment in IBM Watson. Deploying the model in IBM Cloud using model id. An id is created for the model using which the model can be accessed online

Flask Application

```
100, 10 minutes ago | 1 author (100)
import numpy as np
import os
from PIL import Image
from flask import Flask, request, render_template, url_for
from werkzeug.utils import secure_filename, redirect
from gevent.pywsgi import WSGIServer
from flask import send_from_directory
from joblib import Parallel, delayed
import joblib
import pandas as pd
from scipy.sparse import issparse
import pickle
import requests
```

FIG. IMPORTING THE REQUIRED LIBRARIES

```
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "yJjRKquYkWeXxU3CGuvFC0Q1f29M8lpidj6PZ8B0RQeY"
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__)
```

FIG. LINKING WITH MODEL ON IBM CLOUD

The program in above serves as the backend for our Web page API and linking our Machine Learning model with it. We have a home landing page. From that you will be directed to the predict page where you can give the inputs. These input received from that page is then sent to out ML model to do the prediction and the output will be displayed at the next web page. It is the connection between the Frontend and backend.

```

@app.route('/')
def index():
    return render_template('home.html')

@app.route('/predict')
def predict():
    return render_template('predict.html')

@app.route('/result', methods=['GET', 'POST'])
def upload():
    if request.method == "POST":
        lend_data = request.form.get('lend')
        data = [[request.form.get('gender'), request.form.get('married'), request.form.get('dep'), request.form.get('edu'), request.form.get('co_inco'), request.form.get('cai'), request.form.get('la'), request.form.get('lat'), request.form.get('ch'), request.form.get('pa')]
        a = ''
        lend_data = int(lend_data)
        data_list = data
        payload_scoring = {"input_data": [{"fields": ['gender', 'married', 'depend', 'education', 'self_emp', 'applicant_income', 'co_inco', 'co_inco', 'cai', 'la', 'lat', 'ch', 'pa']}]}
        response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/20aab57d-c8a0-48e0-bf79-b48bac4f7de3/pred')
        print("response_scoring")
        prediction = response_scoring.json()
        num = prediction['predictions'][0]['values'][0][0]
        if(num==0):
            if(lend_data==1):
                a = 'It is not advisable to provide loan for this applicant.'
            else:
                a = 'Your loan application will be Rejected.'
        else:
            if(lend_data==1):
                a = 'This applicant can be provided with the loan amount requested.'
            else:
                a = 'Your loan application will be successful.'
        return render_template('submit.html', num=a)
    You, 22 hours ago • Sprint 4 Uploaded -
if __name__ == '__main__':
    app.run(debug=True, threaded=False)

```

FIG. LINK MODEL AND WEB PAGES WITH FLASK CODE

HTML CODE

```

<div class="Team_content">

    <h1>Smart Lender - Applicant Credibility Prediction For Loan Approval </h1>
    <p> Predit your loan eligibility here</p><br><br>
    <a href="#" onclick="req()" class="btn">PREDICT</a>
    <br><br>
    <h2>Team ID -PNT2022TMID036723</h2><br>
    <h3>Team members</h3>
    <p>Heamanth R</p>
    <p>Mohamed Mustaq </p>
    <p>Ram Kumaran </p>
    <p>Shankara Narayanan</p>
    <p>Kishore Kumar </p>

</div>

```

FIG. HOME PAGE

```

<div class="container px-5 py-24 mx-auto">
  <div class="flex flex-col text-center w-full mb-20">
    <h1 class="Heading">LOAN ELIGIBILITY PREDICTION</h1><br>
    <p class="fill">Fill the form for prediction</p>
  </div>
  <div>

  </div>
  <div class="mb-3">
    <a class="btn btn-primary" href="." id="back" role="button">Back</a></div>

  <form action="/prediction.html" method="post" onsubmit="return valid()">
    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">Name</label>
      <input type="text" class="form-control" id="Name" name="Name" placeholder="Enter your Name" required >
    </div>
    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label"> Email ID</label>
      <input type="email" class="form-control" id="email" name="email" placeholder="Enter your Email ID" required >
    </div>
    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">Mobile Number</label>
      <input type="text" class="form-control" id="mon" name="mon" placeholder="Enter your Mobile number" required>
    </div>
    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label"> Gender</label>
      <select class="form-select" id="gender" name="gender" aria-label="Default select example" required >
        <option selected>-- select gender --</option>
        <option value="Male">Male</option>
        <option value="Female">Female</option>
      </select>
    </div>
  </form>

```

FIG. PREDICT PAGE

```

<div class="row">
  <div style="width: 40%; margin: 25px auto;">
    <div class="group">
      <h3 style="text-align: center;">Login Page</h3>
      <form action="login1.php" method="POST" >
        <div class="form-group">
          <label>UserName:</label><input type="text" name="username" class="form-control" autofocus placeholder="Enter your user"
        </div>
        <div class="form-group">
          <label>Password:</label><input type="Password" name="password" class="form-control" autofocus placeholder="Password"
        </div>

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

```

FIG. LOGIN PAGE

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Loan approval status</title>
    <link rel="stylesheet" href="../static/approved.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@5.15.3/css/all.min.css"/>
  </head>
  <body>
    <h1> STATUS</h1>

    
    <h2>{{prediction_text}}</h2>
  </body>
</html>

```

FIG. APPROVE PAGE

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Loan approval status</title>
    <link rel="stylesheet" href="../static/reject.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@5.15.3/css/all.min.css"/>
  </head>
  <body>
    <h1> STATUS</h1>

    
    <h2>{{prediction_text}}</h2>
  </body>
</html>

```

FIG. REJECT PAGE

WEB PAGE DESIGN :

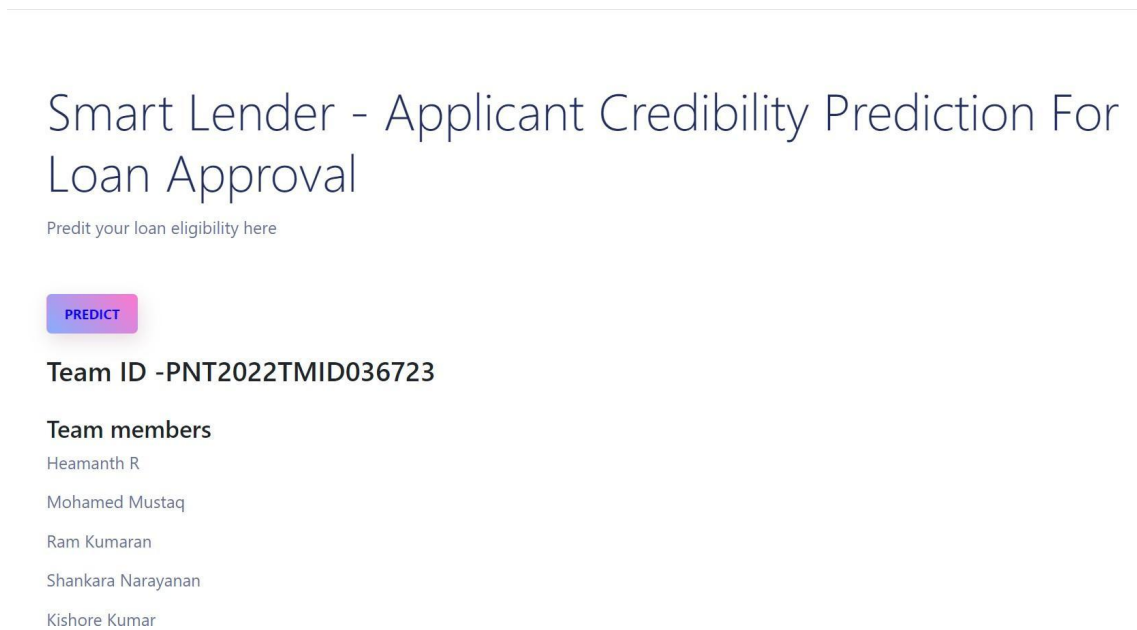


FIG. HOME PAGE

LOAN ELIGIBILITY PREDICTION

Name Enter your Name

Email ID Enter your Email ID

Mobile Number Enter your Mobile numbe

Gender -- select gender --



Married status -- select married status --



Enter LoanAmount LoanAmount

Enter Loan_Amount_Term Loan_Amount_Term

Enter Adhar Number Adhar Number

Enter PAN card ID PAN card ID

Property Document

No file chosen

Govet ID proof

No file chosen

☐ I accept the Terms and conditions

PREDICT

FIG . PREDICTION PAGE

8. TESTING

8.1. TEST CASES

Lender/ Applicant	Gender	Marital Status	Dependants	Education	Self Employed	Applicant Income	Co- applicant Income	Loan Amount	Loan Amount Term	Credit History	Property Area	Loan Status
Applicant	Male	No	0	Graduate	No	54170	0	168000	1080	Yes	Urban	Y
Lender	Male	No	0	Graduate	Yes	69500	0	175000	1080	Yes	Semi-urban	Y
Lender	Male	Yes	0	Graduate	No	26980	20340	212000	580	No	Semi-urban	N
Applicant	Male	Yes	2	Graduate	No	11757	0	187000	780	No	Rural	N
Lender	Female	Yes	0	Graduate	No	23300	44860	1000000	360	Yes	Semi-urban	N
Applicant	Female	Yes	2	Graduate	No	14866	0	700000	1500	Yes	Urban	Y
Applicant	Male	Yes	1	Graduate	No	153800	41250	300000	1000	Yes	Urban	Y
Lender	Female	No	0	Graduate	No	10000	16660	225000	500	No	Rural	N
Lender	Male	Yes	0	Graduate	No	48600	83000	1250000	2360	Yes	Semi-urban	Y

8.2. USER ACCEPTANCE TESTING

IBM Platform was used to test the website. Inputs had been taken from the users who have tested this website and have done modifications to satisfy everyone's needs. The users found the interface very easy to use. The Web pages were colourful and attractive. There was no unnecessary details in the web page. It was clean and simple that any new user could master it. The data input format was also simple. The user need not enter any unit. He could simply enter the value. The prediction time is fairly low at an average time of 3 seconds. This delay primarily varies depending on the internet connectivity. The model has been hosted in the IBM cloud. Thus with the API available, the model can be accessed remotely from any system provided IBM access key is given. The model predicts the loan status in a more accurate manner. We have two provisions. An applicant can also use this website to predict his loan application acceptance probability. Also a banker/lender can also use this to verify whether the applicant can be provided with the loan amount requested. The users are satisfied with the predicted results as they are easier to interpret. Various inputs have been given by the users to test the consistency of the model. The model proved itself and all the users accepted the model as a reliable and convenient option.

9. RESULTS

9.1. PERFORMANCE METRICS

Better performance in speed and accuracy have been shown by The RandomForestClassifier ML model that we have used here compared to other models. We have compared the performance metrics of 4 models and selected this as the best for the application. The model performed well for all the test cases. The API developed also performed good with no glitches or lag found during the testing phase.

10. ADVANTAGES & DISADVANTAGES

10.1. Advantages

The previous manually assessed loan applicant's datasets are used to train this model. So it assesses new applications more accurately. It takes a lot of parameters as input for prediction, which makes the model more effective in prediction. Since the dataset is balanced the model trained is also balanced and produce more accurate, unbiased results. The user interface is simple and elegant, hence making it easier for the end user to utilise it. It serves as a boon to both the lender and loan applicant in accessing the loan application. It saves a lot of time and manual labour involved in this process. With this website's prediction values in hand, the applicant can have a confidence in applying for a loan amount. And it is the same in case of the lender, he can confidently lend money to an applicant.

10.2. Disadvantages

This model must extensively reach every person, so that they can make use of this. Massive implementation of this model in all banks might have practical difficulties. Some banks will have some privacy policies which may not allow such implementation in their system. Some banks might need some extra checks before providing loan to a person. In that case they must remodify the model. So as of now this can be a basic gatepass for the lenders to process a loan application.

11. CONCLUSION

The performance of our dataset is really good with the use of The RandomForestClassifier ML model. The model is fast and consumes less resources. The API developed is also simple and user friendly. By using this model, we could access the credibility of a loan applicant provided the required input data. This saves the time and prevents money landing on fraudulent hands. The model is not 100% accurate but it performs sufficiently well. It can be concluded that the output of this model can be taken as a very important and basic guideline in deciding the credibility of the applicant. Some high priority ground check is unavoidable. So we can proceed to that ground check once we receive a green sign from this model.

12. FUTURE SCOPE

The yet-to-be done updates on this project is to include few more features in model training to study the effect on the prediction. A long history of data (dataset of more than 3 years) can be used for training for increased accuracy. The application can be upgraded such that the input values are fetched directly from the application file and then fed to the model rather than the user entering it manually. A login systems for banks can be developed, so that each bank can have its own login hence making their applicants data more secure.

13. APPENDIX

13.1 SOURCE CODE

HOME PAGE:

```
<!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">
  <!-- CSS only -->
  <link rel="stylesheet" href="../../static/styles.css">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.c
ss" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9
BvlWTRi" crossorigin="anonymous">
  <title>Smart Lender</title>
</head>
<body>
  <nav class="navbar navbar-expand-lg bg-light ">
    <div class="container-fluid">
      <a class="navbar-brand p-3" href="/">Smart Lender</a>
      <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse " id="navbarNav">
        <ul class="navbar-nav me-auto p-2" >
          <li class="nav-item p-2">
            <a class="nav-link " aria-current="page"
href="/">Home</a>
          </li>
          <li class="nav-item p-2">
            <a class="nav-link" href="#">About</a>
          </li>
          <li class="nav-item p-2">
            <a class="nav-link" href="#">Procedure</a>
          </li>
          <li class="nav-item p-2">
            <a href="login" class="nav-link ">Login</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>

  <!-- JavaScript Bundle with Popper -->

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.
```

```

min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6
V8Qbsw3" crossorigin="anonymous"></script>
<script src="../../static/index.js"></script>
</body>
</html>

```

```

<!-- #0C699 #ff6961 #A020F0 -->
<div class="Team_content">

    <h1>Smart Lender - Applicant Credibility Prediction For Loan
Approval </h1>
    <p> Predit your loan eligibility here</p><br><br>
    <a href="#" onclick="req()" class="btn">PREDICT</a>
    <br><br>
    <h2>Team ID -PNT2022TMID036723</h2><br>
    <h3>Team members</h3>
    <p>Heamanth R</p>
    <p>Mohamed Mustaq </p>
    <p>Ram Kumaran </p>
    <p>Shankara Narayanan</p>
    <p>Kishore Kumar </p>

</div>

```

Prediction Page:

```

<!doctype html>
<html lang="en">
    <head>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <!-- Bootstrap CSS -->
        <link rel="stylesheet" href="../../static/styles.css">
        <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap
.min.css" rel="stylesheet"
integrity="sha384-e0JMYsd53ii+sc0/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpb
zKgwra6" crossorigin="anonymous">
        <link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css"
rel="stylesheet">

        <title>prediction</title>
    </head>
    <body>
        <script>
            function valid(){

```

```

var Ai=document.getElementById("ApplicantIncome").value;
var Co=document.getElementById("CoapplicantIncome").value;
var LA=document.getElementById("LoanAmount").value;
var LT=document.getElementById("Loan_Amount_Term").value;
if(Ai > 1000000000000000000000000000000000000){
    alert("Applicant income is too large enter a valid number")
    return false;
}
if(Co > 1000000000000000000000000000000000000){
    alert("Coapplicant income is too large enter a valid number")
    return false;
}
if(LA > 1000000000000000000000000000000000000){
    alert("Loan Amount is too large enter a valid number")
    return false;
}
if(LT > 1000000000000000000000000000000000000){
    alert("loan amount term is too large enter a valid number")
    return false;
}
var name=document.getElementById("Name").value;
var letters=/^[a-zA-Z]*$/;
if(!name.match(letters)){
    alert("Name must contain only alphabets")
    return false;
}
var num=/^[0-9]+$/;
if(!Ai.match(num)){
    alert("Enter only valid numbers alphabets are not allowed ")
    return false;
}
if(!Co.match(num)){
    alert("Enter only valid numbers alphabets are not allowed ")
    return false;
}
if(!LA.match(num)){
    alert("Enter only valid numbers alphabets are not allowed ")
    return false;
}
if(!LT.match(num)){
    alert("Enter only valid numbers alphabets are not allowed ")
    return false;
}
var mo=document.getElementById("mon").value;
var mn=/^[0-9]{10}$/;
if(!mo.match(mn)){
    alert("Please enter only 10 digit mobile number")
    return false;
}
}
</script>
<section class="text-gray-600 body-font">
<div class="container px-5 py-24 mx-auto">
    <div class="flex flex-col text-center w-full mb-20">
        <h1 style="color:blueviolet;">LOAN ELIGIBILITY PREDICTION</h1><br>

```

```
</div>
<div>

</div>
```

```
<form action="/prediction.html" method="post" onsubmit="return valid()">
  <div class="frm">
    <label for="exampleFormControlInput1"
class="form-label">Name</label>
    <input type="text" class="input-form reg-input-form" id="Name"
name="Name" placeholder="Enter your Name" required >
  </div>
  <div class="frm">
    <label for="exampleFormControlInput1" class="form-label"> Email
ID</label>
    <input type="email" class="input-form reg-input-form" id="email"
name="email" placeholder="Enter your Email ID" required >
  </div>
  <div class="frm">
    <label for="exampleFormControlInput1" class="form-label">Mobile
Number</label>
    <input type="text" class="input-form reg-input-form" id="mon"
name="mon" placeholder="Enter your Mobile number" required>
  </div>
  <div class="frm">
    <label for="exampleFormControlInput1" class="form-label"> Gender</label>
    <select class="input-form reg-input-form" id="gender" name="gender"
aria-label="Default select example" required >
      <option selected>-- select gender --</option>
      <option value="Male">Male</option>
      <option value="Female">Female</option>
    </select>
  </div>
  <div class="frm">
    <label for="exampleFormControlInput1" class="form-label"> Married
status</label>
    <select class="input-form reg-input-form" id="married" name="married"
aria-label="Default select example" required >
      <option selected>-- select married status --</option>
      <option value="Yes">Yes</option>
      <option value="No">No</option>
    </select>
  </div>
  <div class="frm">
    <label for="exampleFormControlInput1"
class="form-label">Dependents</label>
    <select class="input-form reg-input-form" id="dependents"
name="dependents" aria-label="Default select example" required>
      <option selected>-- select dependents --</option>
      <option value="0">0</option>
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3+">3+</option>
    </select>
  </div>
  <div class="frm">
    <label for="exampleFormControlInput1"
```

```

class="form-label">Education</label>
  <select class="input-form reg-input-form" id="education" name="education"
aria-label="Default select example" required>
    <option selected>-- select education --</option>
    <option value="Graduate">Graduate</option>
    <option value="Not Graduate">Not Graduate</option>
</select>
</div>
<div class="frm">
  <label for="exampleFormControlInput1"
class="form-label">Self_Employed</label>
  <select class="input-form reg-input-form" id="employed" name="employed"
aria-label="Default select example" required>
    <option selected>-- select Self_Employed --</option>
    <option value="Yes">Yes</option>
    <option value="No">No</option>
</select>
</div>
<div class="frm">
  <label for="exampleFormControlInput1"
class="form-label">Credit_History</label>
  <select class="input-form reg-input-form" id="credit" name="credit"
aria-label="Default select example" required >
    <option selected >-- select Credit_History --</option>
    <option value="Yes">Yes</option>
    <option value="No">No</option>

</select>
</div>
<div class="frm">
  <label for="exampleFormControlInput1"
class="form-label">Property_Area</label>
  <select class="input-form reg-input-form" id="proparea" name="proparea"
aria-label="Default select example" required>
    <option selected>-- select Property_Area --</option>
    <option value="Semiurban">Semiurban</option>
    <option value="Urban">Urban</option>
    <option value="Rural">Rural</option>
</select>
</div>
<div class="frm">
  <label for="exampleFormControlInput1" class="form-label">Enter
ApplicantIncome</label>
  <input type="text" class="input-form reg-input-form" id="ApplicantIncome"
name="ApplicantIncome" placeholder="ApplicantIncome" required>

</div>
<div class="frm">
  <label for="exampleFormControlInput1" class="form-label">Enter
CoapplicantIncome</label>
  <input type="text" class="input-form reg-input-form"
id="CoapplicantIncome" name="CoapplicantIncome"
placeholder="CoapplicantIncome" required>
</div>
<div class="frm">
  <label for="exampleFormControlInput1" class="form-label">Purpose of
loan</label>

```

```

    <select class="input-form reg-input-form" id="pur" name="pur"
aria-label="Default select example" required>
    <option selected>-- select the purpose of loan --</option>
    <option value="person">Personal loan</option>
    <option value="Bussiness">Bussiness loan</option>
    <option value="Education">Education loan</option>
    <option value="Home">Home loan</option>
    <option value="Other">other</option>
</select>
</div>
<div class="frm">
    <label for="exampleFormControlInput1" class="form-label">Enter
LoanAmount</label>
    <input type="text" class="input-form reg-input-form" id="LoanAmount"
name="LoanAmount" placeholder="LoanAmount" required>
</div>
<div class="frm">
    <label for="exampleFormControlInput1" class="form-label">Enter
Loan_Amount_Term</label>
    <input type="text" class="input-form reg-input-form"
id="Loan_Amount_Term" name="Loan_Amount_Term"
placeholder="Loan_Amount_Term" required>
</div>
<div class="frm">
    <label for="exampleFormControlInput1" class="form-label">Enter Adhar
Number</label>
    <input type="text" class="input-form reg-input-form" id="Adhar"
name="Adhar" placeholder="Adhar Number" required >
</div>
<div class="frm">
    <label for="exampleFormControlInput1" class="form-label">Enter PAN card
ID</label>
    <input type="text" class="input-form reg-input-form" id="PAN " name="PAN
" placeholder="PAN card ID" required>
</div>
<div class="frm">
<label for="property document" class="form-label">Property
Document</label><br><input type="file" required >
</div>
<div class="frm">
<label for="Govt ID proof" class="form-label">Govet ID
proof</label><br><input type="file" required>
</div>
<div class="frm">
    <input type="checkbox" required>
    I accept the <a href="terms.html">Terms and conditions</a>
    </div>
<br><br>
<div class="frm">
<input type="submit" class="loginbtn" value="PREDICT">
</div>
    </form>

</div>
</section>
<script

```

```

src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-JEW9xMcG8R+ph31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLts3qm9Ekf" crossorigin="anonymous"></script>

</body>
</html>

```

RESULT:

APPROVAL:

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Loan approval status</title>
    <link rel="stylesheet" href="../static/approved.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>
  </head>
  <body>
    <h1> STATUS</h1>

    
    <h2>{{prediction_text}}</h2>

```

REJECTION:

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Loan approval status</title>
    <link rel="stylesheet" href="../static/reject.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>
  </head>
  <body>
    <h1> STATUS</h1>

    
    <h2>{{prediction_text}}</h2>

```

13.2. GITHUB & PROJECT DEMO LINK

GITHUB REPO LINK : <https://github.com/IBM-EPBL/IBM-Project-36054-1660292222>

DEMO VIDEO LINK

: <https://drive.google.com/file/d/15rTRleRo7nwEbgOpQ2Jr8CPMOSgKrwfl/view>