

PROJECT REPORT

Team Id	PNT2022TMID46202
Project Name	Smart Lender-Applicant Credibility Prediction for Loan Approval

Smart Lender-Applicant Credibility Prediction for Loan Approval

IBM-Project-29566-1660127178

TEAM MEMBER

KAMALI.A

RAJESHWARI.V

TAMILARASI.S

KAVIYARASI.A

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

1. INTRODUCTION

1.1 Project Overview

Today a lot of people/companies are applying for bank loans. The core business part of every bank is the distribution of loans. The main objective of the banking sector is to give their assets in safe hands. But the banks or the financial companies take a very long time for the verification and validation process and even after going through such a regress process there is no surety that whether the applicant chosen is deserving or not. To solve this problem, we have developed a system in which we can predict whether the applicant chosen will be a deserving applicant for approving the loan or not. The system predicts on the basis of the model that has been trained using machine learning algorithms. We have even compared the accuracy of different machine learning algorithms. We got a percentage of accuracy ranging from 75-85% but the best accuracy we got was from Logistic Regression i.e., 88.70% The system includes a user interface web application where the user can enter the details required for the model to predict. The drawback of this model is that it takes into consideration many attributes but in real life sometimes the loan application can also be approved on a single strong attribute, which will not be possible using this system.

1.2 Purpose

Despite the fact that our banking system has many products to sell, the main source of income for a bank is its credit line. So, they can earn from interest on the loans they credit. The credit risk is defined as the likelihood that borrowers will fail to meet their loan obligations [5]. To predict whether the borrower will be good or bad is a very difficult task for any bank or organization. The banking system uses a manual process for checking whether a borrower is a defaulter or not. No doubt the manual process will be more accurate and effective, but this process cannot work when there are a large number of loan applications at the same time. If there occurs a time like this, then the decision-making process will take a very long time and also lots of manpower will be required. If we are able to do the loan prediction it will be very helpful for applicants and also for the employees of banks.

2. LITERATURE SURVEY

2.1 Existing problem

Account firms and banks need to automatize the credit qualification activity (continuously) essentially dependent on data given by customers when rounding out an online structure. Sex, Marital Status, Education, Number of Dependents, Salary, Loan Amount, Credit History, and different subtleties are incorporated. To digitize this interaction, they made an issue to group the client sections that can apply for a credit sum, permitting them to focus on these clients explicitly. They have introduced a fractional informational collection for this situation.

2.2 References

- [1] Ashwini S. Kadam, Shraddha R Nikam, Ankita A. Aher, Gayatri V. Shelke, Amar S. Chandgude. "Prediction for Loan Approval using Machine Learning Algorithm", Apr 2021 International Research Journal of Engineering and Technology (IRJET)
- [2] Mohammad Ahmad Sheikh, Amit Kumar Goel, Tapas Kumar. "An Approach for Prediction of Loan Approval using Machine Learning Algorithm", 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020
- [3] X.FrencisJensy, V.P.Sumathi, Janani Shiva Shri, "An exploratory Data Analysis for Loan Prediction based on nature of clients", International Journal of Recent Technology and Engineering (IJRTE), Volume-7 Issue-4S, November 2018
- [4] J. Tejaswini¹, T. Mohana Kavya, R. Devi Naga Ramya, P. Sai Triveni Venkata Rao Maddumala. "ACCURATE LOAN APPROVAL PREDICTION BASED ON MACHINE LEARNING APPROACH" Vol 11, www.jespublication.com, page 523 Issue 4, April/ 202
- [5] "Prediction for Loan Approval using Machine Learning Algorithm" International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072
- [6] Vaidya, "Predictive and probabilistic approach using logistic regression: Application to prediction of loan approval," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-6.doi: 10.1109/ICCCNT.2017.8203946
- [7] M. Bayraktar, M. S. Aktaş, O. Kalipsız, O. Susuz and S. Bayracı, "Credit risk analysis with classification Restricted Boltzmann Machine," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018, pp. 1-4.doi: 10.1109/SIU.2018.840

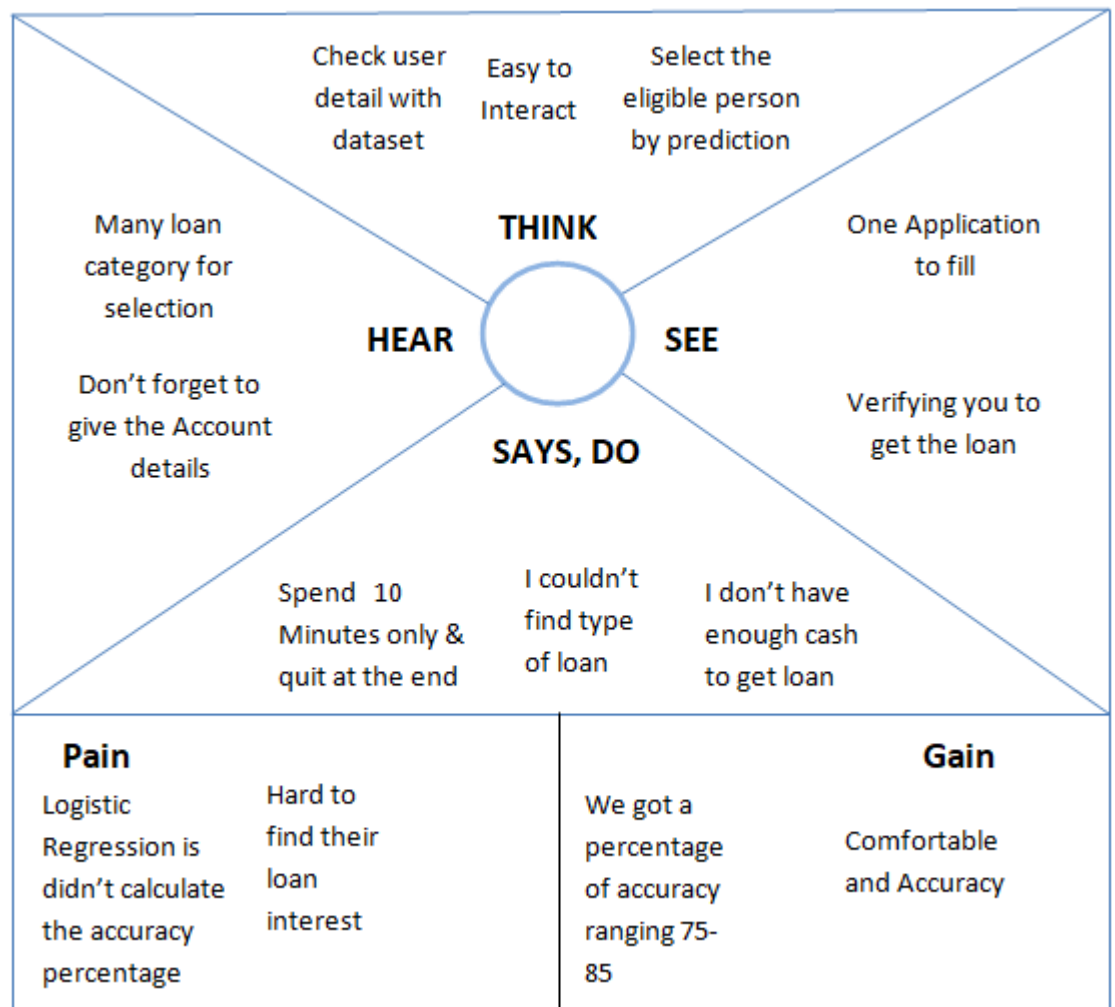
S.no	Title	Author Name	Journal Name & Year	Methodology	Advantages	Disadvantages
1.	Loan Approval prediction	Shubham Nalawade, Suraj Andhe, Siddhesh Parab	IRJET & Apr,2022	Logistic Regression, Naive Bayes	We got a percentage of accuracy ranging 75-85	It takes into consideration many attributes in real life.
2.	Loan Approval with Machine Learning :A Survey	Ms.Kathe Rutika Pramod, Arhad pooja prakash, Mr.Ghorpade Dinesh B	IJCRT & 6 June 2021	DT is a Supervised Learning Algorithm	A key important approach in predictive analytics used.	Logistic Regression is didn't calculate the accuracy percentage.
3.	Application of Data Mining Tools in CRM for selected Bank	Dilleep B.Desai, Dr.R.V.Kulkarni	IJCSIT & 2013	Machine Learning Logic Regression	This system can Automatically calculate the weight of each Features taking part.	This model that is emphasize different weight to each factor but in real life.
4.	Loan Prediction by using Machine Learning Models	Pidikiti Supriya, K.Vikas, Namburi Vimala Kumari	IJET & Mar-Apr 2019	Logic Regression Machine Learning Decision Tree Python	Data mining is the Process of analyzing data from different Perspectives.	Sometimes loan can be approved on the basis of single strong Factor only ,which is not possible in this system.
5.	Developing Prediction Model of Loan Risk in Banks using Data mining	J.H.Aboodyda,MA. Tarig	MLAIJ & 2016	Machine Learning Logic Regression	Loan processing and on new test data same features are processed with respect to their data.	It emphasize the different weight to each of the user in the real life.

2.3 Problem Statement Definition

Banks are making major part of profits through loans. Loan approval is a very important process for banking organizations. It is very difficult to predict the possibility of payment of loan by the customers because there is an increasing rate of loan defaults and the banking authorities are finding it more difficult to correctly access loan requests and tackle the risks of people defaulting on loans. Machine Learning has eased today's world by developing these prediction models .

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

KAMALI.A

- The system predicts the basis of model that has been trained using **Machine learning** Algorithm. We have even compared the accuracy of different Machine Learning Algorithm. Accuracy ranging from **75-85%**.
- The data is collected from the Kaggle for studying and prediction **Logistic Regression** models have been performed and the different measures of performance are computed.

TAMILARASI.S

- **Decision Tree** algorithm in **Machine Learning** methods which efficiently performs both Classification and Regression task.
- The data is collected from the Kaggle for studying and prediction **Logistic Regression** models have been performed and the different measures of performance are computed.

RAJESHWARI.V

- In **Machine Learning** the Decision Tree algorithm the work proves that the R package is an efficient visualizing tool that applies data mining techniques.
- Using **R package** customer's data analysis can be done and depends on that bank can sanction or reject the Loan.

KAVIYARASI.A

- Machine Learning algorithm we using the **XGBoost**, this algorithm only works with the quantitative variable.
- In this project we will be using the fine techniques of **Machine Learning - Decision Tree** algorithm to build this prediction model for Loan assessment.

Best of Three:

- Machine Learning algorithm we using the **XGBoost**, this algorithm only works with the quantitative variable.

- The data is collected from the Kaggle for studying and prediction **Logistic Regression** models have been performed and the different measures of performance are computed.
- **Decision Tree** algorithm in **Machine Learning** methods which efficiently performs both Classification and Regression task.

3.3 Proposed Solution

Machine learning was used to predict loan acceptance. The prediction method begins with data pre-processing, filling the missing values, experimental data analysis. After evaluating model on test dataset, each of these algorithms obtained a precision rate between 70% and 80%. Although here it can be concluded with certainty that the Support Vector Machine model is very efficient and produces superior results than other models.

3.4 Problem Solution fit

The prediction method begins with data pre-processing, filling the missing values, experimental data analysis. After evaluating model on test dataset, each of these algorithms obtained a precision rate between 70% and 80%. Although here it can be concluded with certainty that the Support Vector Machine model is very efficient and produces superior results than other models.

1. It generates random forests and then uses these random forests to seek the solutions.
2. It is an ensemble learning in which significant number of classifiers are used to solve a
3. Random forest analyzes each tree for prediction rather than just one to avoid overfitting issues
4. The greater the number of trees, the greater the accuracy in problem solving.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

The functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	User can Registration through Form Registration through Gmail
FR-2	User Confirmation	The user can Confirmation via Email Confirmation via OTP
FR-3	User Verification	The user is verified with the previous dataset using Machine Learning.
FR-4	User Conformation	After user get tested by the Dataset ,If He eligible to get loan send conformation through Gmail
FR-5	Authentication & Authorization	The user is verified by using Authentication control protocol by Login details.

4.2 Non-Functional requirements

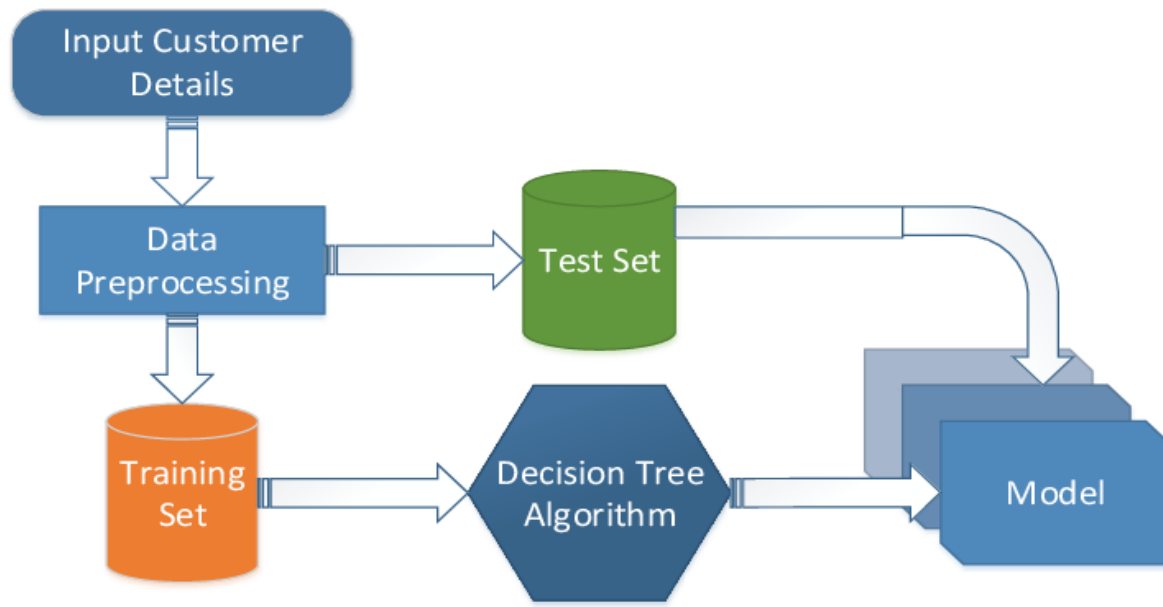
The non-functional requirements of the proposed solution.

FR No	Non-Functional Requirement	Description
NFR-1	Usability	This system is very comfortable to use for Applicant
NFR-2	Security	It is very Secured to Access by Authentication control.
NFR-3	Reliability	This is completely Reliable for the applicant and management.
NFR-4	Performance	It has very high performance by Machine Learning.
NFR-5	Availability	The applicant can access this system by Mobile Phone ,PC and any Browser ,etc..
NFR-6	Scalability	Many applicant can access this system at the same time.It is very scalable.

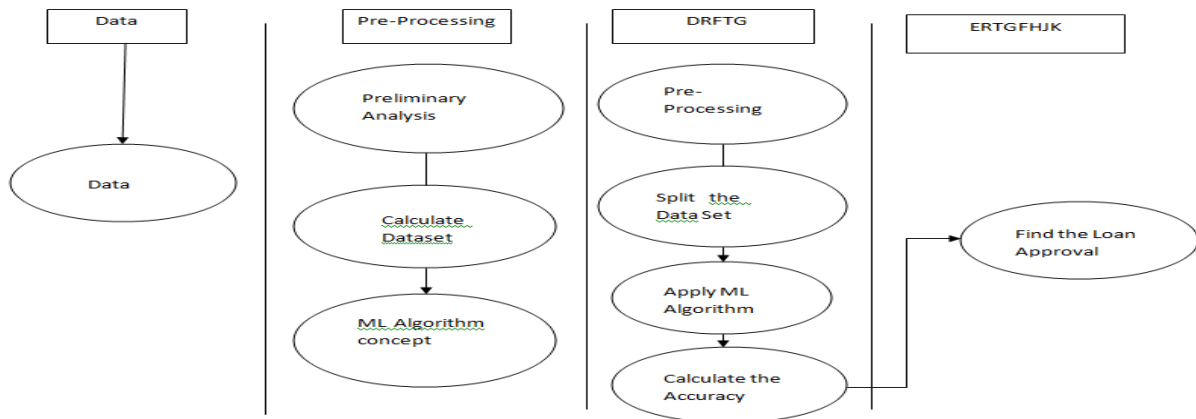
5.PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically.



5.2 Solution & Technical Architecture



5.3 User Stories

A customer journey map is a visual representation of the customer journey (also called the buyer journey or user journey). It helps you tell the story of your customers' experiences with your brand across all touchpoints. Whether your customers interact with you via social media, email, livechat or other channels, mapping the customer journey out visually helps ensure no customer slips through cracks.

Problems	If a client arrives by car, will they be able to park at our building	Is the queuing system easy to use	Does the employee pitch additional services at an inopportune moment	Can the employee briefly describe in simple terms	How do support staff behave if they don't know the answer to a custom	Do we respond to feedback, or does it gather dust	Doesn't our customer's email box look like a graveyard of sales offers
Ideas/ Operation	Can the customer be sure they will be accepted if there is not much time left before the bank closes	Can clients sit comfortably or do they have to stand while waiting in line	Does the bank employee express themselves in an understandable language	Is it possible to delay the signing so the client can read the contract at their own pace	When is support available, given that our customers may be in different time zones	you can check bank review sites and forums and search social media for clients' opinion.	A person in a difficult situation has taken out a loan, it makes no sense to promote favorable insurance plans or a new loan to them.

6.PROJECT PLANNING & 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 entering my email, password, and	2	High	4

			confirming my password			
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	4
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	4
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	4
Sprint-1		USN-5	As a user, I can log into the application by entering email & password	1	High	4
	Dashboard					

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022

Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

// Code Starts Here

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import GradientBoostingClassifier

#Read CSV data
data = pd.read_csv("../input/train_u6lujuX_CVtuZ9i (1).csv")

#preview data
data.head()

#Check missing values
data.isnull().sum()

# percent of missing "Gender"
```

```

print('Percent of missing "Gender" records is %.2f%%'
      %((data['Gender'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by gender :")

print(data['Gender'].value_counts())

sns.countplot(x='Gender', data=data, palette = 'Set2')

print("Number of people who take a loan group by marital status :")

print(data['Married'].value_counts())

sns.countplot(x='Married', data=data, palette = 'Set2')

# percent of missing "Dependents"

print('Percent of missing "Dependents" records is %.2f%%'
      %((data['Dependents'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by dependents :")

print(data['Dependents'].value_counts())

sns.countplot(x='Dependents', data=data, palette = 'Set2')

//Code End Here

```

7.2 Feature 2

// Code Starts Here

```

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.ensemble import GradientBoostingClassifier

#Read CSV data

data = pd.read_csv("../input/train_u6lujuX_CVtuZ9i (1).csv")

#preview data

data.head()

```



```

#Check missing values

data.isnull().sum()

# percent of missing "Gender"

print('Percent of missing "Gender" records is %.2f%%'
      %((data['Gender'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by gender :")

print(data['Gender'].value_counts())

sns.countplot(x='Gender', data=data, palette = 'Set2')

print("Number of people who take a loan group by marital status :")

print(data['Married'].value_counts())

sns.countplot(x='Married', data=data, palette = 'Set2')

# percent of missing "Dependents"

print('Percent of missing "Dependents" records is %.2f%%'
      %((data['Dependents'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by dependents :")

print(data['Dependents'].value_counts())

sns.countplot(x='Dependents', data=data, palette = 'Set2')

# percent of missing "LoanAmount"

print('Percent of missing "LoanAmount" records is %.2f%%'
      %((data['LoanAmount'].isnull().sum()/data.shape[0])*100))

ax = data["LoanAmount"].hist(density=True, stacked=True, color='teal', alpha=0.6)

data["LoanAmount"].plot(kind='density', color='teal')

ax.set(xlabel='Loan Amount')

plt.show()

# percent of missing "Loan_Amount_Term"

print('Percent of missing "Loan_Amount_Term" records is %.2f%%'
      %((data['Loan_Amount_Term'].isnull().sum()/data.shape[0])*100))

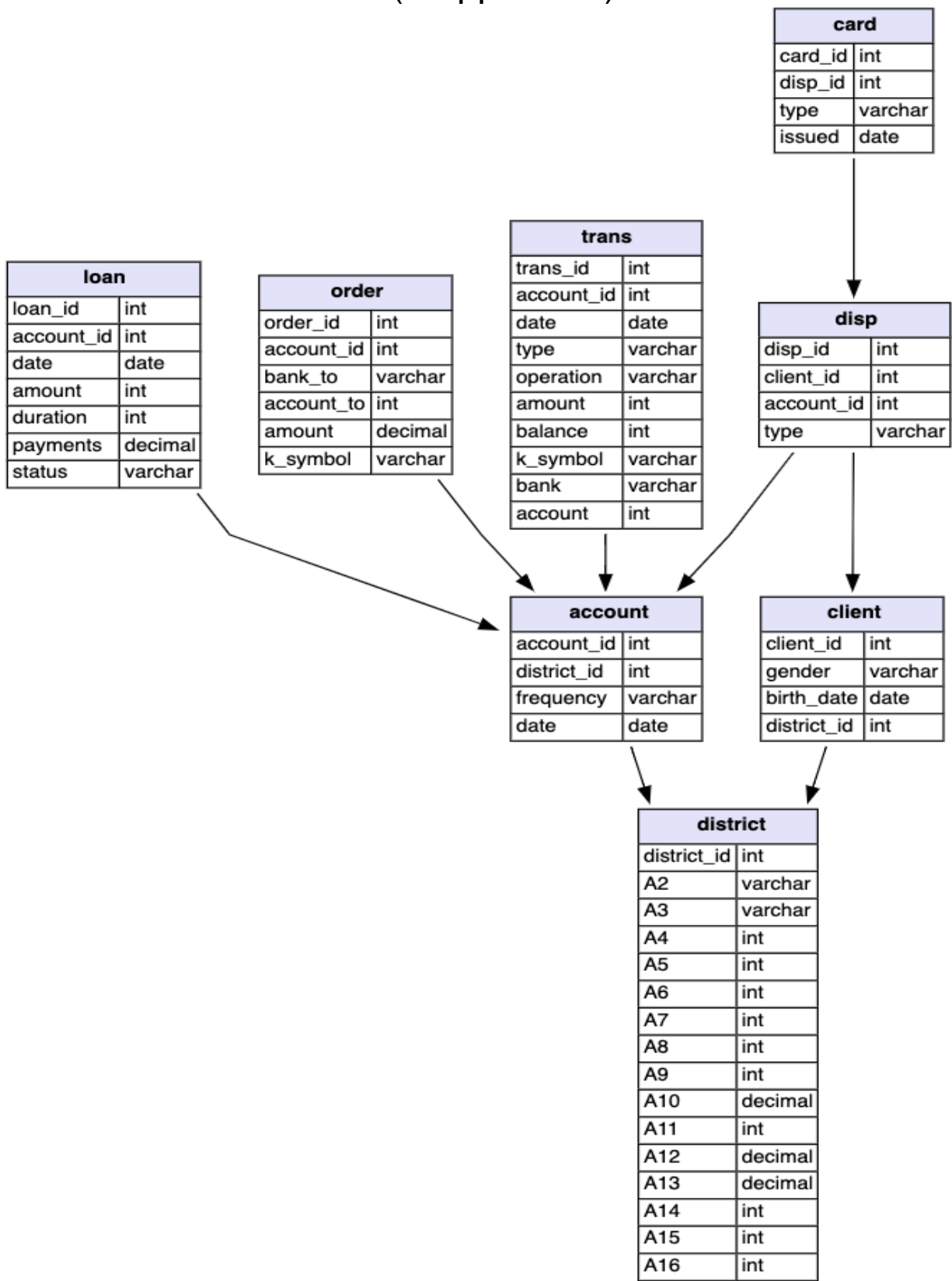
```

```

print("Number of people who take a loan group by loan amount term :")
print(data['Loan_Amount_Term'].value_counts())
sns.countplot(x='Loan_Amount_Term', data=data, palette = 'Set2')
# percent of missing "Credit_History"
print('Percent of missing "Credit_History" records is %.2f%%'
      %((data['Credit_History'].isnull().sum()/data.shape[0])*100))
print("Number of people who take a loan group by credit history :")
print(data['Credit_History'].value_counts())
sns.countplot(x='Credit_History', data=data, palette = 'Set2')
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import numpy as np
import pandas as pd
sc = StandardScaler()
data=pd.read_csv(r"C:\Users\ELCOT\Downloads\Dataset\Churn_Modelling.csv")
x = data.iloc[:, 0:2]
x_scaled = sc.fit_transform(x)
y=data['Geography']
x_train, x_test, y_train, y_test = train_test_split(x_scaled, y, test_size =
0.1,random_state = 0)
x_train
x_train.shape
x_test
x_test.shape

```

7.3 Database Schema (if Applicable)



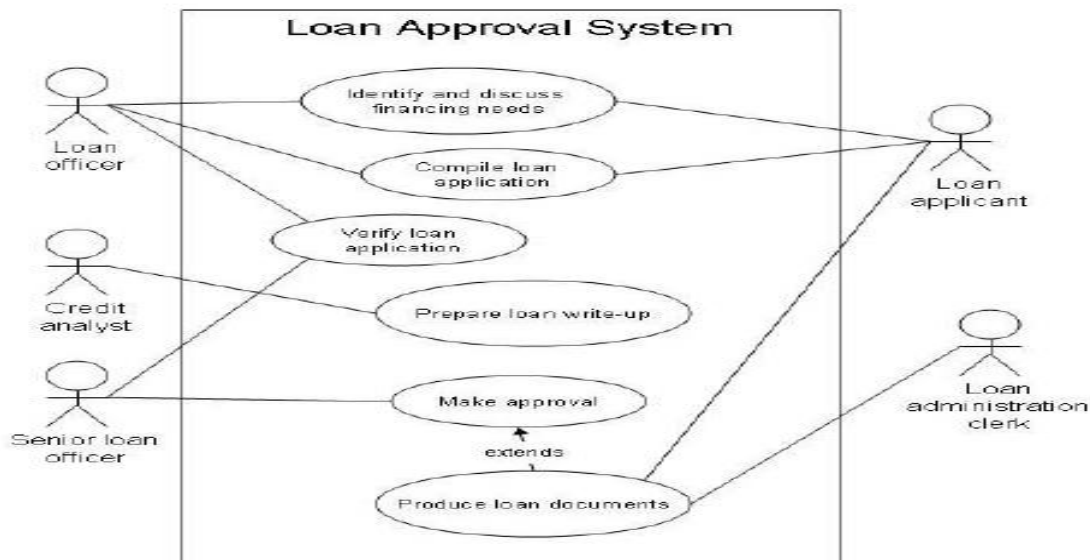
8. TESTING

8.1 Test Cases

Using decision tree algorithm, the outcomes of all applicant can be stored in any file. In this milestone you will start the project development and expected to perform the coding & solutioning, acceptance testing, performance testing based as per the sprint and submit them. In this activity you are expected to develop & submit the developed code by testing it.

Algorithm:

1. Import all the required python modules
2. Import the database for both TESTING and TRAINING.
3. Check any NULLVALUES are exists
4. If NULLVALUES exists ,fill the table with corresponding coding
5. Exploratory Data Analysis for all ATTRIBUTES from the table
6. Plot all graphs using MATPLOTLIB module
7. Build the DECISIONTREE MODEL for the coding
8. Send that output to CSV FILE



8.2 User Acceptance Testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

1. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3 .Test Case Analysis

This report shows the number of test cases that have passed,failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource	3	0	0	3

Shipping				
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics




Performance matrices	Condition output
Human interference cut down	good
Reduction of wastage	good
Economical efficiency	better
reliability	excellent

10. Advantages and Disadvantages:-

Advantages:

- ✚ The user can be remote at any time.
- ✚ The user interference is not required .
- ✚ Reduces over irrigation.
- ✚ Reliability is high.
- ✚ Enhances the process of irrigation.
- ✚ Reduce wastage of resources.
- ✚ Improves lifestyle of farmers.
- ✚ Makes the progression to be easy.
- ✚ Improves ground water level in a periodical manner.
- ✚ Improved yield for farmers.
- ✚ Attracts most of the people to involve in this schema.
- ✚ Since the agriculture improves, human life also improves.

Disadvantages:-

-  work for the people is reduced .
-  sensors and the components should be maintained .
-  there may be a threat of damaging sensors by animals present in the field .

11. CONCLUSION

The interaction of expectation begins from cleaning and handling of information, attribution of missing qualities, exploratory investigation of informational collection and afterward model structure to assessment of model and testing on test information. On Data set, the best-case precision acquired on the first informational collection is 0.811. The accompanying ends are reached after examination that those candidates whose credit score rating was most noticeably awful will neglect to get advance endorsement, because of a higher likelihood of not repaying the credit sum. More often than not, those candidates who have top level salary and requests for lower measure of advance are bound to get affirmed which bodes well, bound to take care of their credits. Some other trademark like gender and conjugal status appears to be not to be mulled over by the organization.

12. FUTURE SCOPE

We hope that this project is able to tackle the problems present in the real and could be developed further more in the process of automation on feeding pest killer, insect killer sprays, and feeding fertilizer for the land,etc...

13. APPENDIX

Source Code

// Code Starts Here

```
#Import packages

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn import svm

#Read CSV data

data = pd.read_csv("../input/train_u6lujuX_CVtuZ9i (1).csv")
```



```
#preview data

data.head()

#Preview data information

data.info()

#Check missing values

data.isnull().sum()

# percent of missing "Gender"

print('Percent of missing "Gender" records is %.2f%%'
      %((data['Gender'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by gender :")

print(data['Gender'].value_counts())

sns.countplot(x='Gender', data=data, palette = 'Set2')

# percent of missing "Married"

print('Percent of missing "Married" records is %.2f%%'
      %((data['Married'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by marital status :")

print(data['Married'].value_counts())

sns.countplot(x='Married', data=data, palette = 'Set2')

# percent of missing "Dependents"

print('Percent of missing "Dependents" records is %.2f%%'
      %((data['Dependents'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by dependents :")

print(data['Dependents'].value_counts())

sns.countplot(x='Dependents', data=data, palette = 'Set2')

# percent of missing "Self_Employed"

print('Percent of missing "Self_Employed" records is %.2f%%'
      %((data['Self_Employed'].isnull().sum()/data.shape[0])*100))
```

```

print("Number of people who take a loan group by self employed :")

print(data['Self_Employed'].value_counts())

sns.countplot(x='Self_Employed', data=data, palette = 'Set2')

# percent of missing "LoanAmount"

print('Percent of missing "LoanAmount" records is %.2f%%'
      %((data['LoanAmount'].isnull().sum()/data.shape[0])*100))

ax = data["LoanAmount"].hist(density=True, stacked=True, color='teal', alpha=0.6)

data["LoanAmount"].plot(kind='density', color='teal')

ax.set(xlabel='Loan Amount')

plt.show()

# percent of missing "Loan_Amount_Term"

print('Percent of missing "Loan_Amount_Term" records is %.2f%%'
      %((data['Loan_Amount_Term'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by loan amount term :")

print(data['Loan_Amount_Term'].value_counts())

sns.countplot(x='Loan_Amount_Term', data=data, palette = 'Set2')

# percent of missing "Credit_History"

print('Percent of missing "Credit_History" records is %.2f%%'
      %((data['Credit_History'].isnull().sum()/data.shape[0])*100))

print("Number of people who take a loan group by credit history :")

print(data['Credit_History'].value_counts())

sns.countplot(x='Credit_History', data=data, palette = 'Set2')

train_data = data.copy()

train_data['Gender'].fillna(train_data['Gender'].value_counts().idxmax(), inplace=True)

train_data['Married'].fillna(train_data['Married'].value_counts().idxmax(), inplace=True)

train_data['Dependents'].fillna(train_data['Dependents'].value_counts().idxmax(),
                               inplace=True)

```

```

train_data['Self_Employed'].fillna(train_data['Self_Employed'].value_counts().idxmax(),
inplace=True)

train_data["LoanAmount"].fillna(train_data["LoanAmount"].mean(skipna=True),
inplace=True)

train_data['Loan_Amount_Term'].fillna(train_data['Loan_Amount_Term'].value_counts().
idxmax(), inplace=True)

train_data['Credit_History'].fillna(train_data['Credit_History'].value_counts().idxmax(),
inplace=True)

#Check missing values

train_data.isnull().sum()

train_data

#Convert some object data type to int64

gender_stat = {"Female": 0, "Male": 1}

yes_no_stat = {'No' : 0, 'Yes' : 1}

dependents_stat = {'0':0, '1':1, '2':2, '3+':3}

education_stat = {'Not Graduate' : 0, 'Graduate' : 1}

property_stat = {'Semiurban' : 0, 'Urban' : 1, 'Rural' : 2}


train_data['Gender'] = train_data['Gender'].replace(gender_stat)
train_data['Married'] = train_data['Married'].replace(yes_no_stat)
train_data['Dependents'] = train_data['Dependents'].replace(dependents_stat)
train_data['Education'] = train_data['Education'].replace(education_stat)
train_data['Self_Employed'] = train_data['Self_Employed'].replace(yes_no_stat)
train_data['Property_Area'] = train_data['Property_Area'].replace(property_stat)

#Preview data information

data.info()

data.isnull().sum()

```

```

#Separate feature and target
x = train_data.iloc[:,1:12]
y = train_data.iloc[:,12]

#make variabel for save the result and to show it

classifier = ('Gradient Boosting','Random Forest','Decision Tree','K-Nearest
Neighbor','SVM')

y_pos = np.arange(len(classifier))

score = []

clf = GradientBoostingClassifier()
scores = cross_val_score(clf, x, y,cv=5)
score.append(scores.mean())

print('The accuration of classification is %.2f%%' %(scores.mean()*100))

clf = RandomForestClassifier(n_estimators=10)
scores = cross_val_score(clf, x, y,cv=5)
score.append(scores.mean())

print('The accuration of classification is %.2f%%' %(scores.mean()*100))

clf = DecisionTreeClassifier()
scores = cross_val_score(clf, x, y,cv=5)
score.append(scores.mean())

print('The accuration of classification is %.2f%%' %(scores.mean()*100))

clf = KNeighborsClassifier()
scores = cross_val_score(clf, x, y,cv=5)
score.append(scores.mean())

print('The accuration of classification is %.2f%%' %(scores.mean()*100))

```

```
clf = svm.LinearSVC(max_iter=5000)
scores = cross_val_score(clf, x, y, cv=5)
score.append(scores.mean())
print('The accuration of classification is %.2f%%' %(scores.mean()*100))
plt.barh(y_pos, score, align='center', alpha=0.5)
plt.yticks(y_pos, classifier)
plt.xlabel('Score')
plt.title('Classification Performance')
plt.show()
```

//Code End Here

Source Code Link : <https://github.com/IBM-EPBL/IBM-Project-29566-1660127178>

GitHub Link : <https://github.com/IBM-EPBL/IBM-Project-29566-1660127178>

Project Demo video Link : <https://bit.ly/30Dd0oF>

Thank You
