GOVERNMENT OF TAMILNADU

Naan Muthalvan - Project-Based Experiential Learning

A Review of Liver Patient Analysis Methods Using Machine Learning

Submitted by

Team ID-NM2023TMID22556

K.VISHNUPRIYA-(20326ER038)

A.SIVASAKTHI - (20326ER035)

H.VIJAYALAKSHMI - (20326ER036)

S.VINOTHINI-(20326ER037)

Under the guidance of

Mrs. J. SUKANYA, MCA., M.Phil.,

Assistant Professor

PG and Research Department of Computer Science

M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR WOMEN

(Affiliated To Mother Teresa Women's University, Kodaikanal)

Reaccredited with "A" Grade by NAAC

DINDIGUL-624001.

APRIL - 2023

M.V.MUTHIAH GOVERNMENT ARTS COLLEG FOR WOMEN

(Affiliated to Mother Teresa Women's University, Kodaikanal)

Reaccredited with "A" Grade by NAAC

Dindigul - 624 001

PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of the project entitled, **"REAL TIME SIGN LANGUAGE DETECTION**" done **by Ms.A.SIVASAKTHI -(20326ER035), Ms.H.VIJAYALAKSHMI-(20326ER036),Ms.S.VINOTHINI-(20326ER036)**and Ms.**K.VISHNUPRIYA- (20326ER038)**. This is submitted in partial fulfillment for the award of the degree of **Bachelor of Science in Computer Science in M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR OMEN,DINDIGUL** during the period of December 2022 to April 2023.

Project Mentor(s) **Head of the Department**

Submitted for viva-voce Examination held on 11.4.2023

TABLE OF CONTENTS

ABSTRACT:

Personal loans are a popular form of borrowing in the financial industry, and the approval or rejection of personal loan applications is typically determined by financial institutions based on various factors. In recent years, machine learning algorithms have gained significant attention for their potential in predicting personal loan approval decisions with high accuracy. This paper presents a comprehensive review of existing research and techniques used for predicting personal loan approval using machine learning.

Furthermore, the paper discusses feature engineering techniques that are commonly employed to extract relevant features from the loan data, such as feature selection , and handling missing data. Finally, the paper highlights some of the key challenges and limitations of using machine learning for personal loan approval prediction, such as issues related to data quality, bias, and interpretability. It also provides suggestions for future research directions in this area, including the integration of alternative data soures , the use of explainable AI techniques, and the development of fair lending models to ensure non-discriminatory loan approval decisions.

1.INTRODUCTION

1.1 Overview

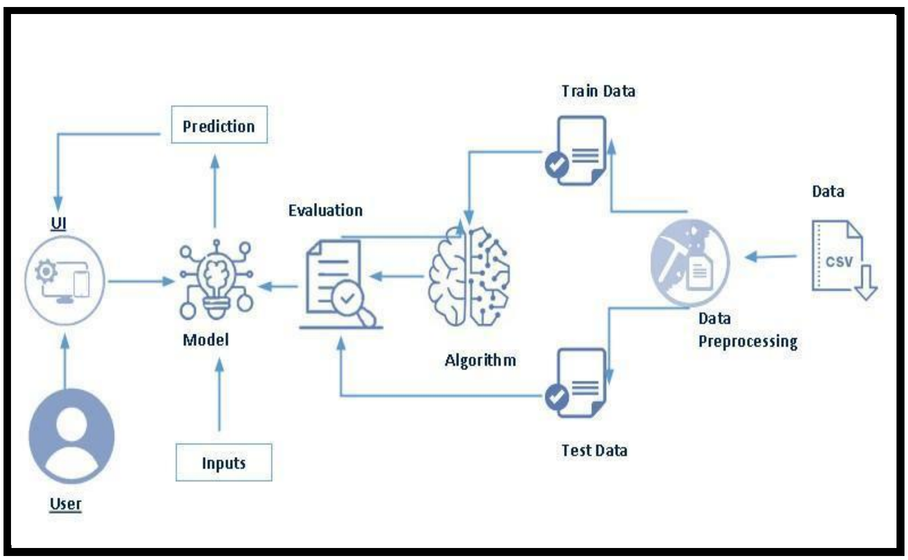
A loan is a sum of money that is borrowed and repaid over a period of time, typically with interest. There are various types of loans available to individuals and businesses, such as personal loans, mortgages, auto loans, student loans, business loans and many more.They are offered by banks, credit unions, and other financial institutions, and the terms of the loan, such as interest rate, repayment period, and fees, vary depending on the lender and the type of loan.

A personal loan is a type of unsecured loan that can be used for a variety of expenses such as home

repairs, medical expenses, debt consolidation, and more. The loan amount, interest rate, and repayment period vary depending on the lender and the borrower's creditworthiness.To qualify for a personal loan, borrowers typically need to provide proof of income and have a good credit score.

Predicting personal loan approval using machine learning analyses a borrower's financial data and credit history to determine the likelihood of loan approval. This can help financial institutions to make more informed decisions about which loan applications to approve and which to deny.

TechnicalArchitecture:



1.2 Purpose:

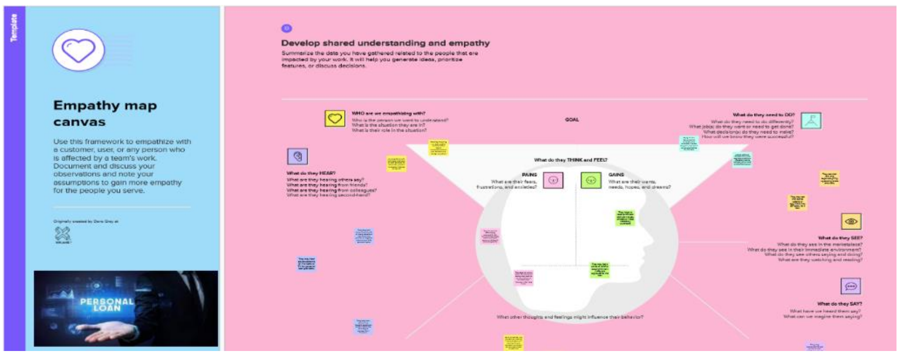
The use of a machine learning model for predicting personal loan approval can have several potential benefits for financial institutions, borrowers, and other stakeholders. Some potential use cases and benefits of using such a model include:

- Improved Loan Approval Decision-making: A well-trained machine learning model can analyze a wide range of relevant data points and provide accurate predictions on loan approval decisions. This can help financial institutions make more informed and data-driven loan approval decisions, reducing the risk of human bias or subjective judgments. It can also help streamline the loan approval process and reduce the time and effort involved in manual reviews, leading to increased efficiency.
- Enhanced Risk Assessment: By analyzing historical loan data and identifying patterns and trends, a machine learning model can assess the credit risk associated with each loan application. This can help lenders better understand the creditworthiness of borrowers and assess the risk of default, which can ultimately lead to more prudent lending decisions and reduced credit risk exposure.
- Increased Loan Approval Rates: By accurately predicting loan approval outcomes, a machine learning model can help financial institutions identify borrowers who are likely to be approved for a loan, increasing the chances of successful loan applications. This can help borrowers gain access to credit and achieve their financial goals, while also benefiting lenders by expanding their customer base.
- Enhanced Customer Experience: By using a machine learning model to automate and streamline the loan approval process, financial institutions can provide borrowers with a faster, more convenient, and transparent loan application experience. This can lead to increased customer satisfaction and loyalty, as borrowers appreciate a seamless and efficient loan application process.
- Reduced Loan Defaults: By accurately assessing credit risk and making informed loan approval decisions, a machine learning model can help lenders mitigate the risk of loan defaults. This can result in reduced financial losses associated with defaulting loans and improved overall loan portfolio performance.
- Better Compliance and Transparency: Using a machine learning model for loan approval can help financial institutions ensure compliance with relevant regulations, such as fair lending practices, anti-discrimination laws, and consumer protection laws. The model can provide a transparent.

2.Problem Definition and Design Thinking:

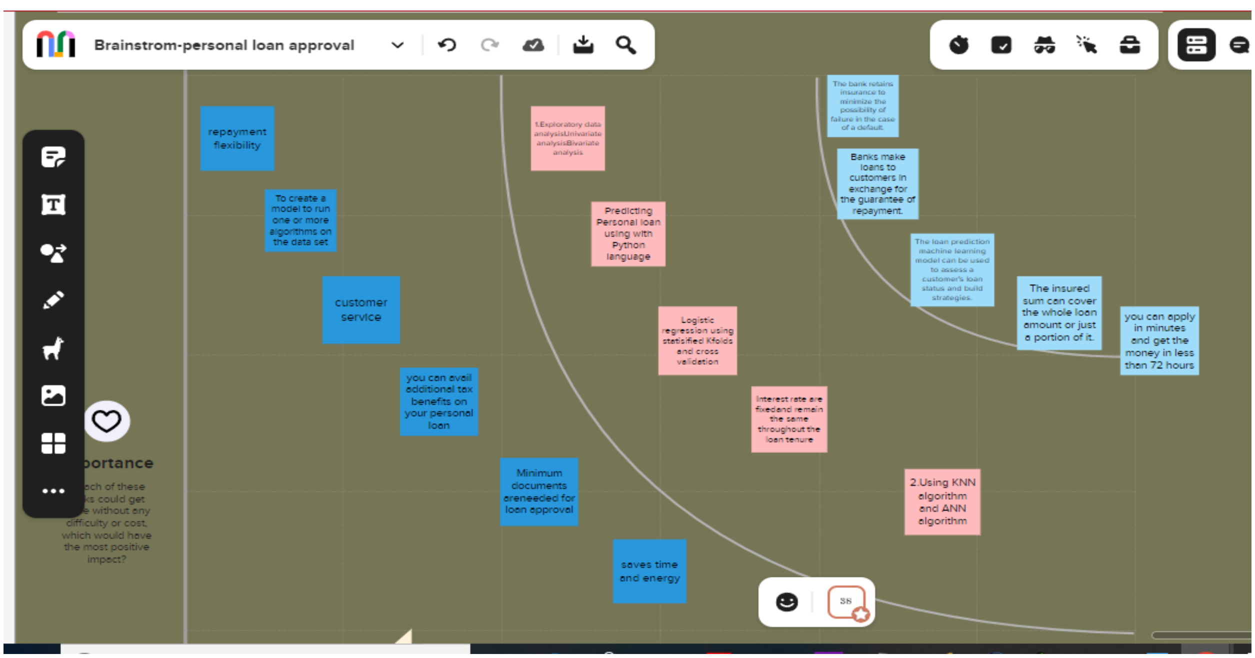
2.1 Empathy Map

Empathy maps are tools used to understand and empathize with the thoughts, feelings, and experiences of a particular group of people. In this case, we can create an empathy map to predict the thoughts, feelings, and experiences of an individual who is applying for a personal loan and awaiting approval.

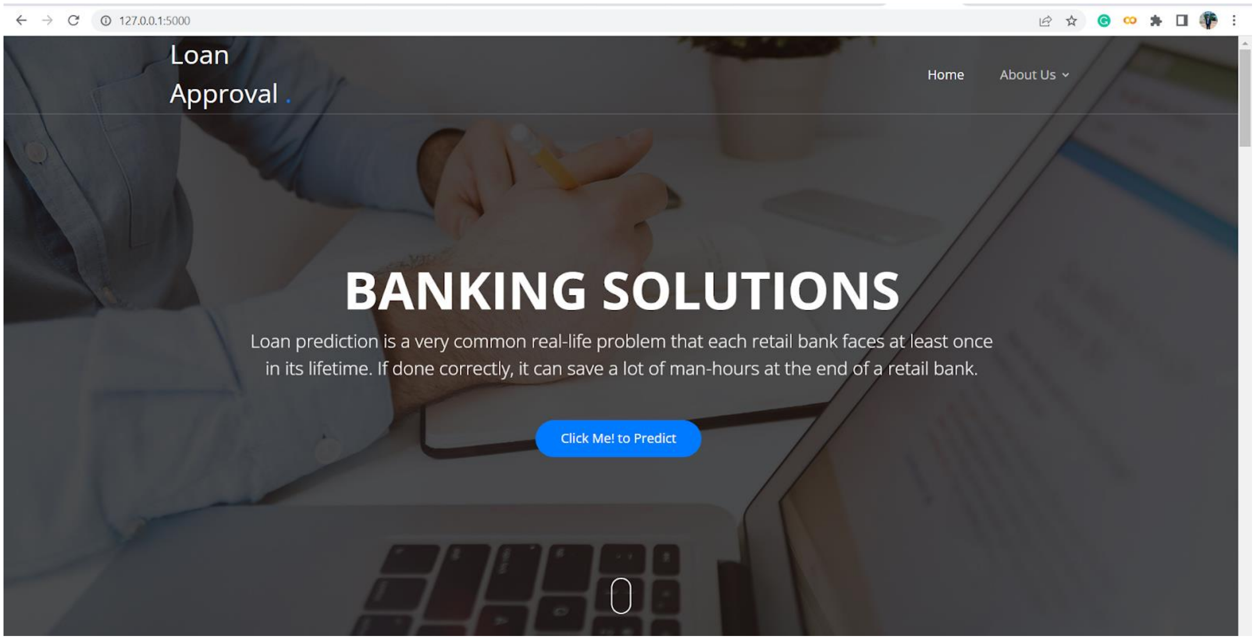


2.2 Ideation and Brainstorming map

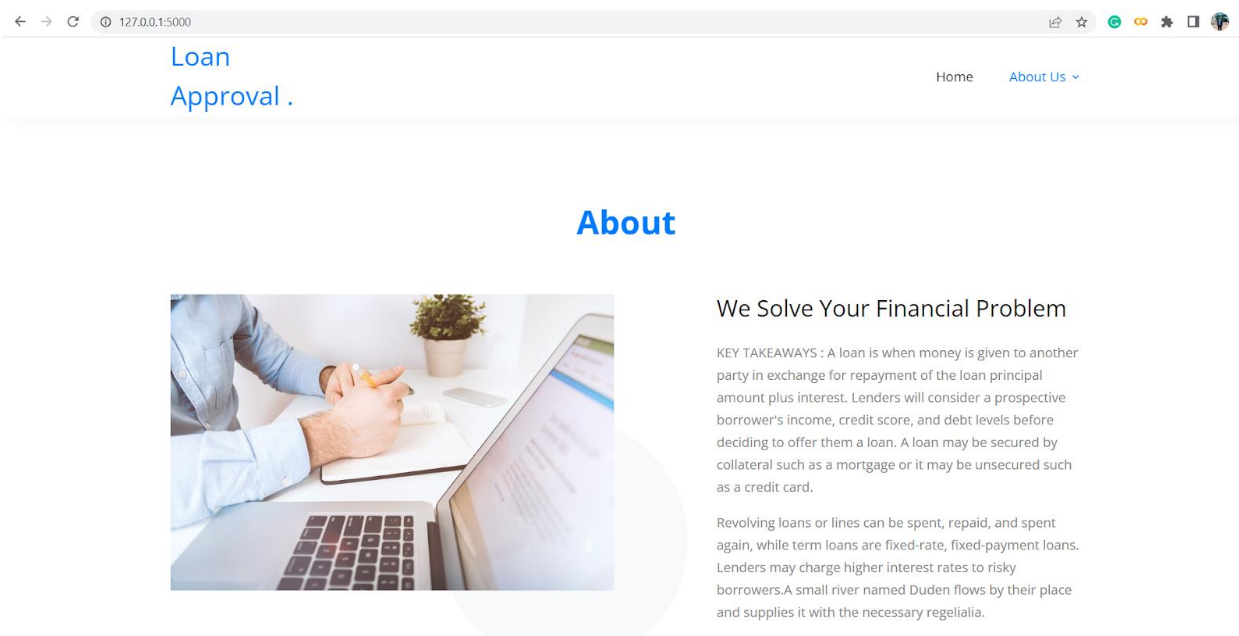
Using this ideation and brainstorming map, you can visually organize your thoughts, stimulate creativity, and generate a wide range of ideas related to your central idea or problem statement.



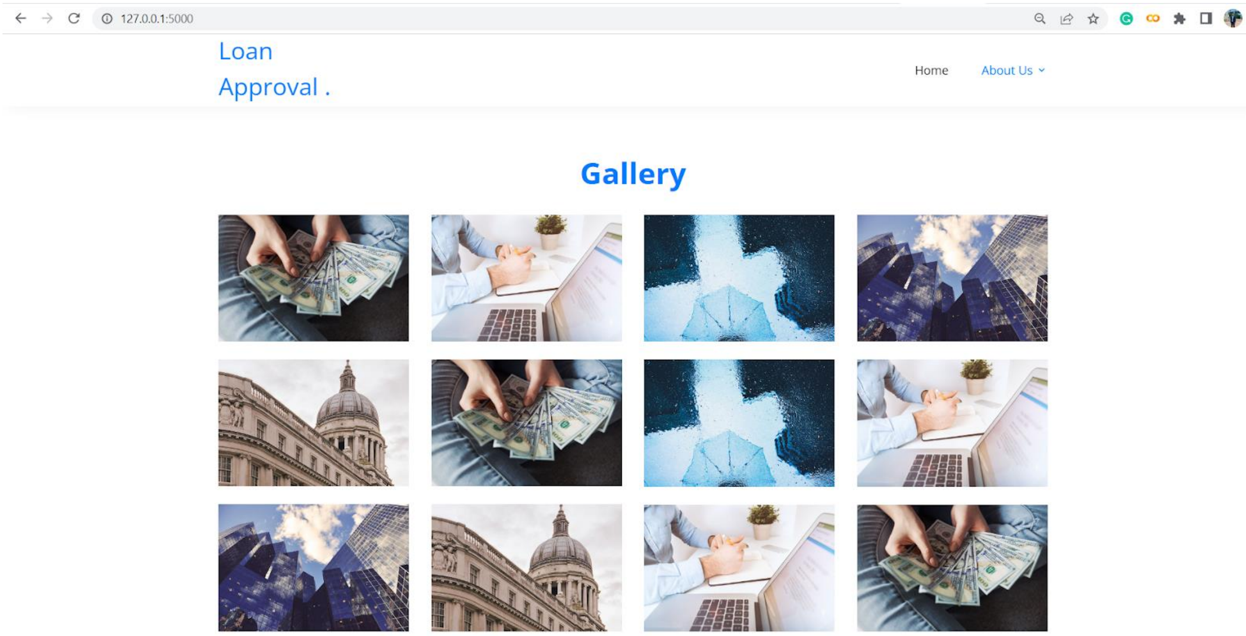
3.Result

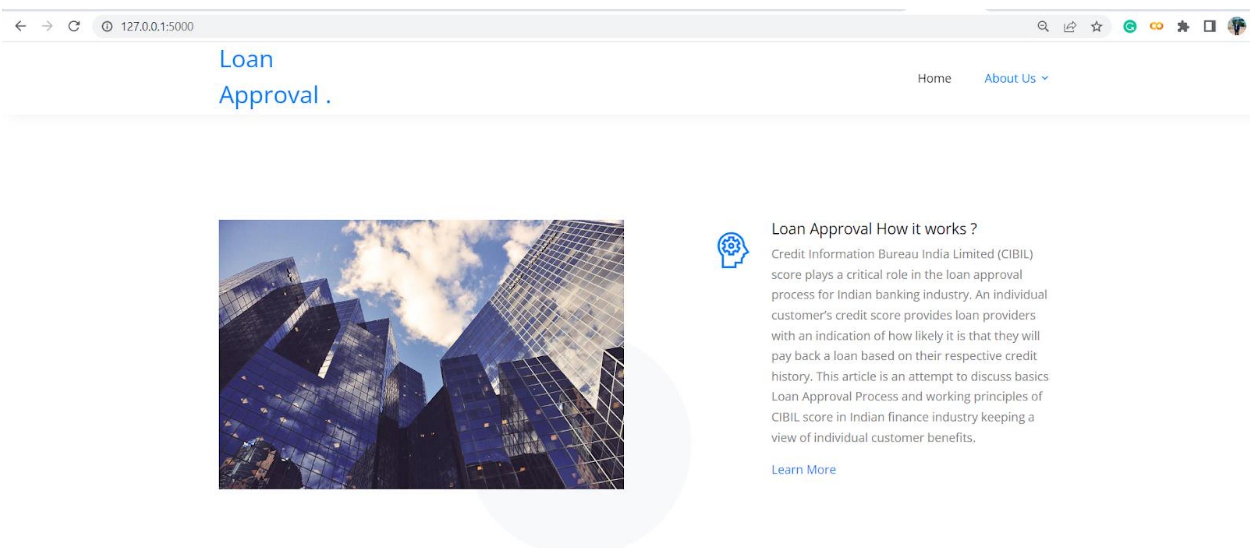


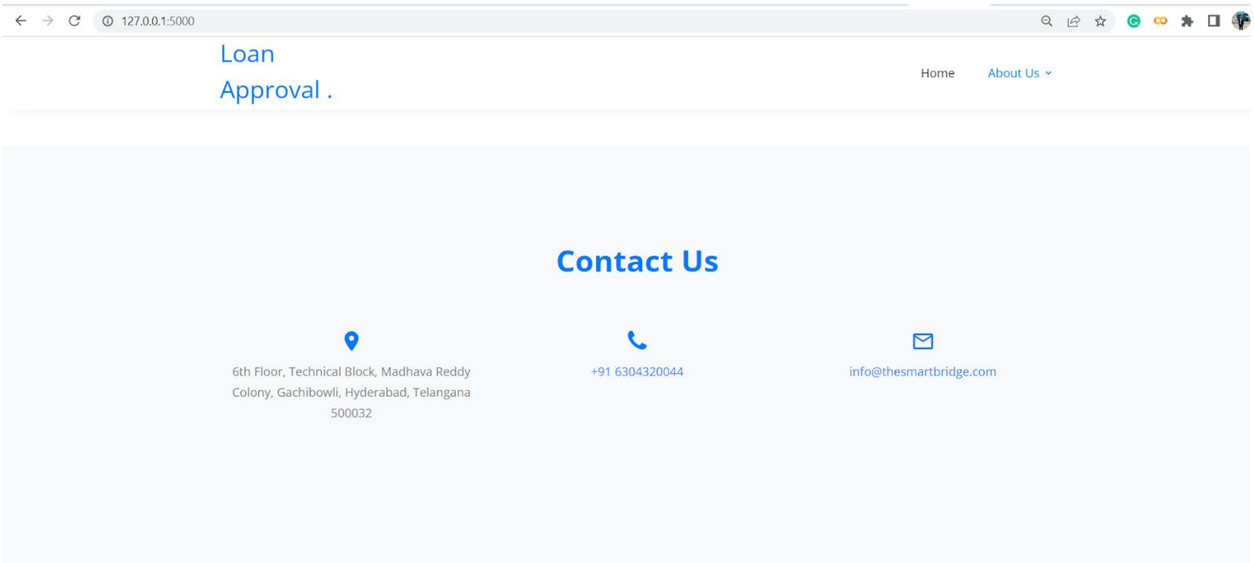
127.0.0.1:5000
Loan
Approval .
Home
About Us
BANKING SOLUTIONS
Loan prediction is a very common real-life problem that each retail bank faces at least once in its lifetime. If done correctly, it can save a lot of man-hours at the end of a retail bank.
Click Me! to Predict

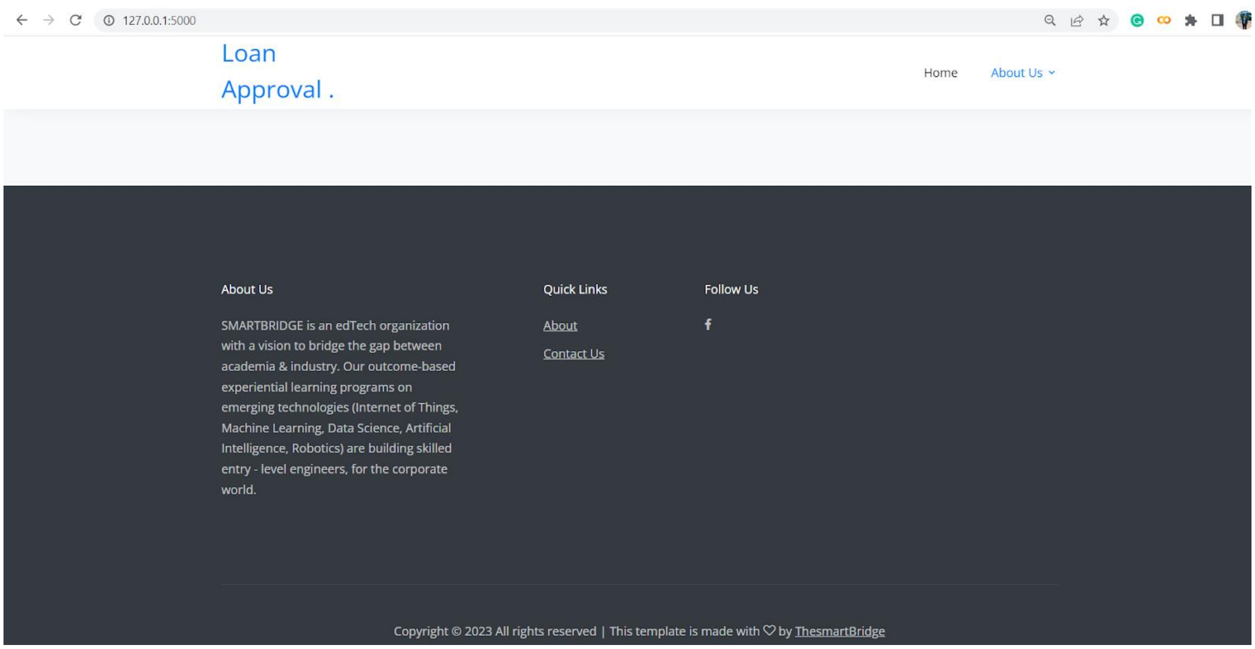


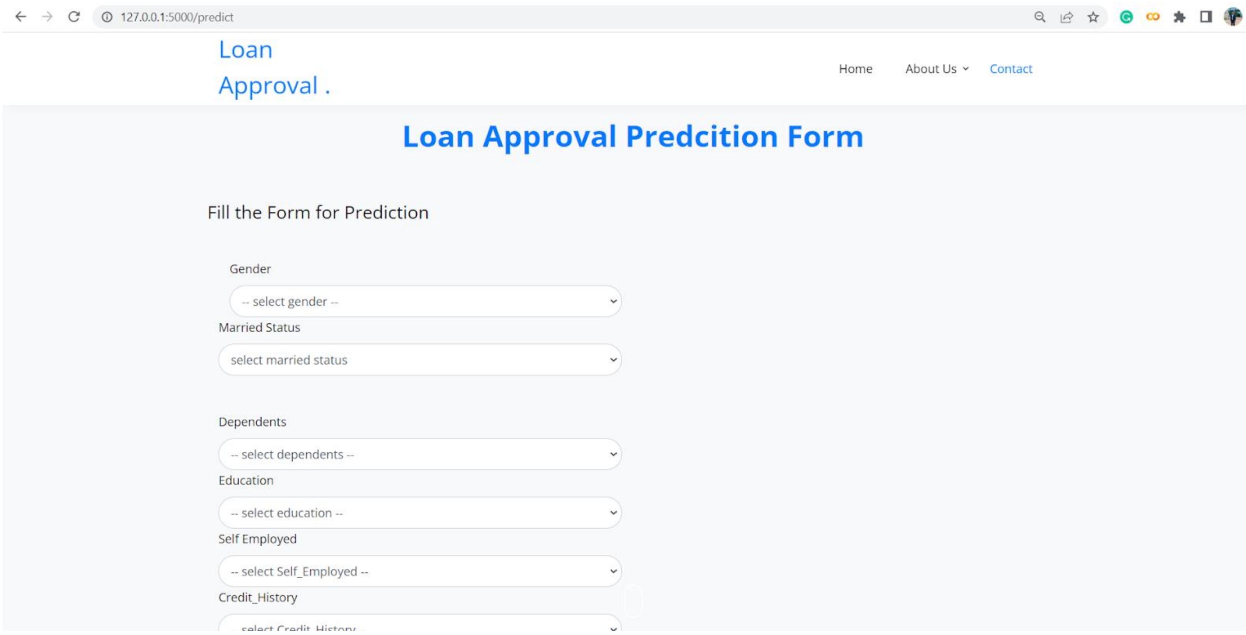
127.0.0.1:5000
Loan
Approval .
Home
About Us
About
We Solve Your Financial Problem
KEY TAKEAWAYS : A loan is when money is given to another party in exchange for repayment of the loan principal amount plus interest. Lenders will consider a prospective borrower's income, credit score, and debt levels before deciding to offer them a loan. A loan may be secured by collateral such as a mortgage or it may be unsecured such as a credit card.
Revolving loans or lines can be spent, repaid, and spent again, while term loans are fixed-rate, fixed-payment loans. Lenders may charge higher interest rates to risky borrowers.A small river named Duden flows by their place and supplies it with the necessary regelialia.

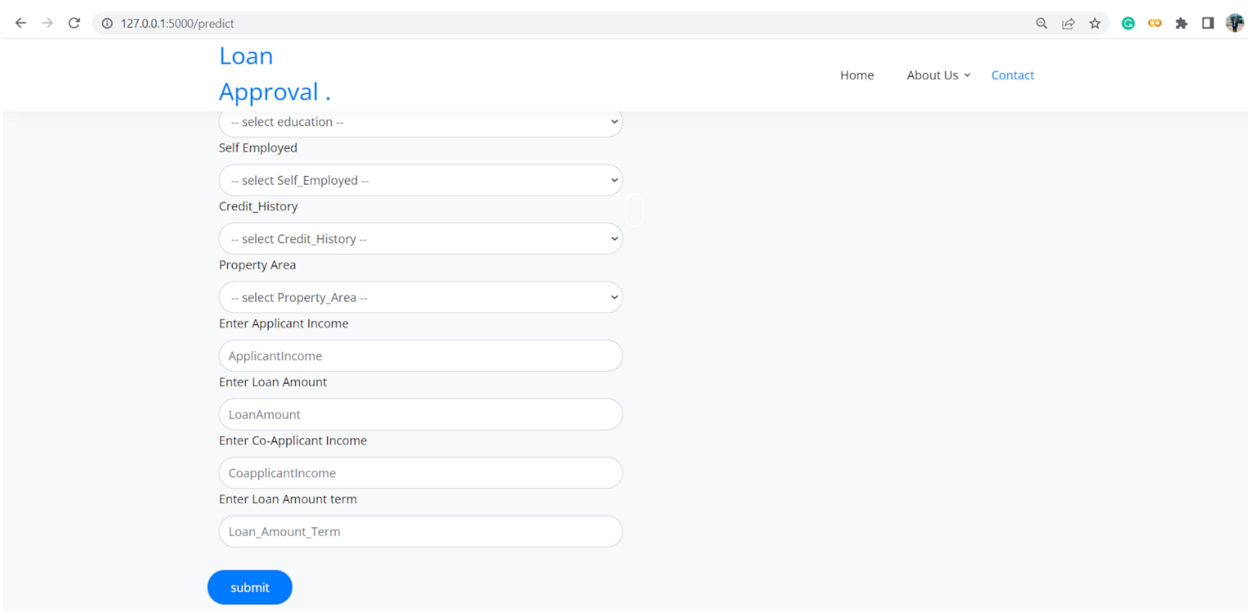

127.0.0.1:5000
Loan
Approval .
Home
About Us
Gallery

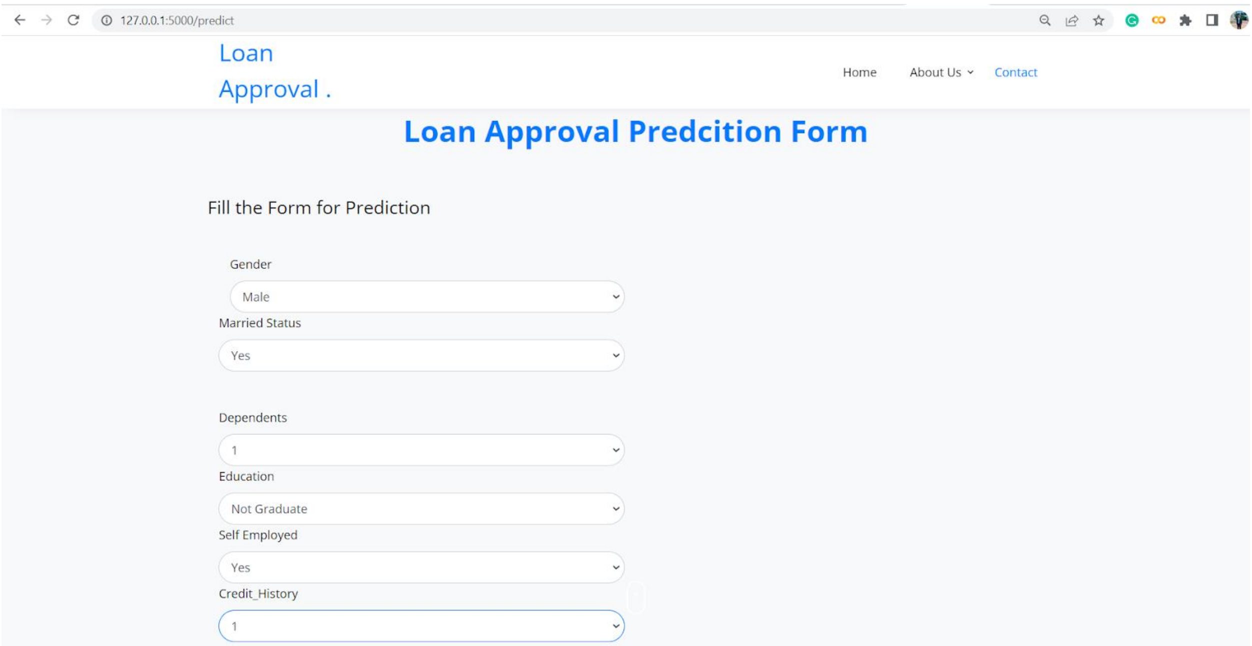

127.0.0.1:5000
Loan
Approval .
Home
About Us
Loan Approval How it works ?
Credit Information Bureau India Limited (CIBIL) score plays a critical role in the loan approval process for Indian banking industry. An individual customer's credit score provides loan providers with an indication of how likely it is that they will pay back a loan based on their respective credit history. This article is an attempt to discuss basics Loan Approval Process and working principles of CIBIL score in Indian finance industry keeping a view of individual customer benefits.
Learn More

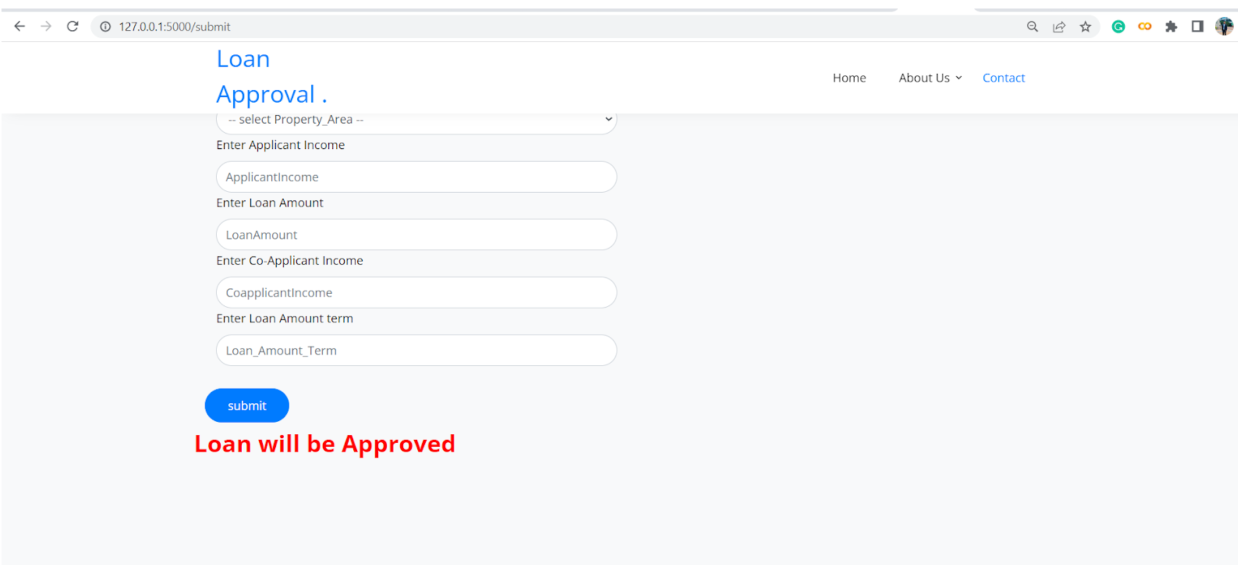

127.0.0.1:5000
Loan
Approval .
Home
About Us
Contact Us
6th Floor, Technical Block, Madhava Reddy Colony, Gachibowli, Hyderabad, Telangana 500032
+91 6304320044
info@thesmartbridge.com


127.0.0.1:5000
Loan
Approval .
Home
About Us
About Us
SMARTBRIDGE is an edTech organization with a vision to bridge the gap between academia & industry. Our outcome-based experiential learning programs on emerging technologies (Internet of Things, Machine Learning, Data Science, Artificial Intelligence, Robotics) are building skilled entry - level engineers, for the corporate world.
Quick Links
About
Contact Us
Follow Us
Copyright © 2023 All rights reserved | This template is made with ♡ by ThesmartBridge


127.0.0.1:5000/predict
Loan
Approval .
Home
About Us
Contact
Loan Approval Predcition Form
Fill the Form for Prediction
Gender
-- select gender --
Married Status
select married status
Dependents
-- select dependents --
Education
-- select education --
Self Employed
-- select Self_Employed --
Credit_History


127.0.0.1:5000/predict
Loan
Approval .
Home
About Us
Contact
-- select education --
Self Employed
-- select Self_Employed --
Credit_History
-- select Credit_History --
Property Area
-- select Property_Area --
Enter Applicant Income
ApplicantIncome
Enter Loan Amount
LoanAmount
Enter Co-Applicant Income
CoapplicantIncome
Enter Loan Amount term
Loan_Amount_Term
submit


127.0.0.1:5000/predict
Loan Approval .
Home
About Us
Contact
Loan Approval Predcition Form
Fill the Form for Prediction
Gender
Male
Married Status
Yes
Dependents
1
Education
Not Graduate
Self Employed
Yes
Credit_History
1


127.0.0.1:5000/submit
Loan Approval .
Home
About Us
Contact
-- select Property_Area --
Enter Applicant Income
ApplicantIncome
Enter Loan Amount
LoanAmount
Enter Co-Applicant Income
CoapplicantIncome
Enter Loan Amount term
Loan_Amount_Term
submit
Loan will be Approved

4.ADVANTAGES AND DISADVANTAGES

Advantages:

- Increased Efficiency: Automating the loan approval process with a machine learning model can save time and effort compared to manual reviews. The model can analyze large amounts of data quickly and provide predictions, enabling faster decision-making and reducing the time borrowers have to wait for loan approval.
- Improved Accuracy: A well-trained machine learning model can accurately assess credit risk and predict loan approval outcomes based on historical data. This can help financial institutions make more accurate and consistent loan approval decisions, reducing the risk of human bias or subjective judgments.
- Enhanced Risk Management: By accurately assessing credit risk, a machine learning model can help lenders identify high-risk loan applications and mitigate the risk of defaults. This can lead to better risk management practices, reduced credit risk exposure, and improved overall loan portfolio performance.

Disadvantages:

- Lack of Complete Information: Predicting loan approval requires access to comprehensive and accurate data, including the borrower's financial history, credit score, employment information, and more. However, the information provided may not always be complete or up-to-date, which can result in inaccurate predictions.
- Changing Economic Conditions: Economic conditions, such as interest rates, inflation rates, and job market trends, can affect loan approval decisions. However, these conditions can change over time and may not always be accounted for in the prediction model, leading to inaccurate predictions.
- Bias and Fairness Issues: Personal loan approval prediction models may inadvertently contain bias, such as racial, gender, or socio-economic bias, if the data used to train the model is biased. This can result in unfair treatment of certain groups of borrowers, leading to potential discrimination and unethical practices.

5.APPLICATIONS

1. Risk Assessment: Lenders can use prediction models to assess the risk associated with a loan application by analyzing various factors, such as the borrower's credit history, income, employment stability, and other relevant financial information. This can help lenders determine the likelihood of a borrower defaulting on a loan and make informed decisions about loan approval.
2. Loan Underwriting: Prediction models can aid in the loan underwriting process, where lenders evaluate the creditworthiness of a borrower and determine the terms of the loan, such as interest rate, loan amount, and repayment period. Prediction models can provide insights into the borrower's credit risk profile and help streamline the underwriting process.
3. Automated Decision Making: Prediction models can be integrated into automated loan approval systems, where loan applications are processed and approved or denied automatically based on predefined criteria. This can help lenders expedite the loan approval process, reduce manual effort, and improve operational efficiency.
4. Portfolio Management: Prediction models can be used to assess and manage the risk of an existing loan portfolio by monitoring the creditworthiness of borrowers over time. Lenders can use these models to identify potential delinquencies or defaults, proactively manage risk, and take appropriate actions, such as loan restructuring or collection strategies.
5. Customer Relationship Management: Prediction models can also help lenders personalize their interactions with borrowers by providing insights into their credit risk profiles. This can enable lenders to offer customized loan products, pricing, and repayment options based on the borrower's creditworthiness, financial needs, and preferences, leading to better customer relationship management.
6. Fraud Detection: Prediction models can be used to detect potential loan application fraud or misrepresentation by analyzing patterns and anomalies in the loan application data. This can help lenders identify suspicious loan applications and mitigate the risk of fraudulent loans.

6.CONCLUSION

predicting personal loan approval using prediction models can offer several potential benefits, such as improving risk assessment, streamlining loan underwriting, enabling automated decision-making, enhancing portfolio management, supporting customer relationship management, and aiding fraud detection. These models can provide lenders with insights into borrowers' creditworthiness, financial history, and other relevant factors, helping them make informed decisions about loan approvals.

However, it's crucial to acknowledge the limitations and potential disadvantages of loan approval prediction models, such as incomplete information, changing economic conditions, bias and fairness issues, dynamic borrower behavior, legal and regulatory constraints, and the subjective nature of human judgment. Lenders should use these models responsibly and in compliance with relevant laws and regulations, ensuring fair and ethical lending practices.

Loan approval prediction models should be considered as tools to aid in decision-making and not as standalone determinants of loan approvals. Human judgment and oversight should always be exercised to ensure that borrowers are treated fairly and transparently, and that potential biases are identified and addressed. Responsible and ethical use of prediction models, in conjunction with human judgment, can help lenders make informed and responsible loan approval decisions while upholding fairness, transparency, and compliance with relevant regulations.

7.FUTURE SCOPE

- Enhanced Accuracy: As technology continues to advance, prediction models may become more accurate in assessing credit risk, leveraging sophisticated algorithms and machine learning techniques. This can lead to improved loan approval decisions, reduced default rates, and better risk management for lenders.
- Big Data and Alternative Data Sources: The availability of big data and alternative data sources, such as social media, transactional data, and behavioral data, can provide lenders with more comprehensive and diverse information to assess creditworthiness. This can help lenders make more accurate loan approval predictions, especially for borrowers with limited credit history or no credit score.
- Explainable AI and Fairness: The development of explainable artificial intelligence (AI) models can provide greater transparency and insights into the factors influencing loan approval decisions. This can help lenders identify and address potential biases and ensure fairness in lending practices, promoting ethical and responsible use of prediction models.
- Real-time Decision Making: With the increasing availability of real-time data, lenders may be able to make loan approval decisions in near real-time, allowing for faster and more efficient loan processing. This can benefit borrowers by reducing processing times and improving the overall customer experience.
- Personalized Loan Products: Prediction models may enable lenders to offer more personalized loan products and terms based on the borrower's individual credit risk profile, financial needs, and preferences. This can lead to more tailored loan offerings, better customer engagement, and improved borrower satisfaction.
- Integration with Blockchain Technology: The use of blockchain technology in the lending process can provide enhanced security, transparency, and efficiency. Prediction models can potentially leverage blockchain-based smart contracts and digital identities to streamline loan approvals, reduce fraud, and enhance trust in the lending process.
- Regulatory Compliance: Prediction models may evolve to incorporate and adapt to changing regulatory requirements, such as anti-discrimination laws, consumer protection laws, and privacy regulations. This can help lenders ensure compliance with relevant regulations and mitigate legal and reputational risks.

8.APPENDIX

A. SOURCE CODE

```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier,RandomForestClassi
fier
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,confusio
n_matrix,f1_score
#importing the dataset which is in csv file
data=pd.read_csv('/content/sample_data/train_u6lujuX_CVtuZ9i.csv')
data
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	L
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	3
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	3
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	3
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	3
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	3
...	...	...	...	...	...	...	...	...	...	..
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	3
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	1
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	3
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	3
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	3

614 rows × 13 columns

```
data.info()
```

	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

```
#finding the sum of null values in each column
data.isnull().sum()

data['Gender']=data['Gender'].fillna(data['Gender'].mode()[0])
data['Married']=data['Married'].fillna(data['Married'].mode()[0])
#replacing+with space for filling the non values
data['Dependents']=data['Dependents'].str.replace('+','')
data['Dependents']=data['Dependents'].fillna(data['Dependents'].mode()[0])
data['Self_Employed']=data['Self_Employed'].fillna(data['Self_Employed'].m
ode()[0])
data['Loan_Amount_Term']=data['Loan_Amount_Term'].fillna(data['Loan_Amount
_Term'].mode()[0])
data['Credit_History']=data['Credit_History'].fillna(data['Credit_History'
].mode()[0])
```

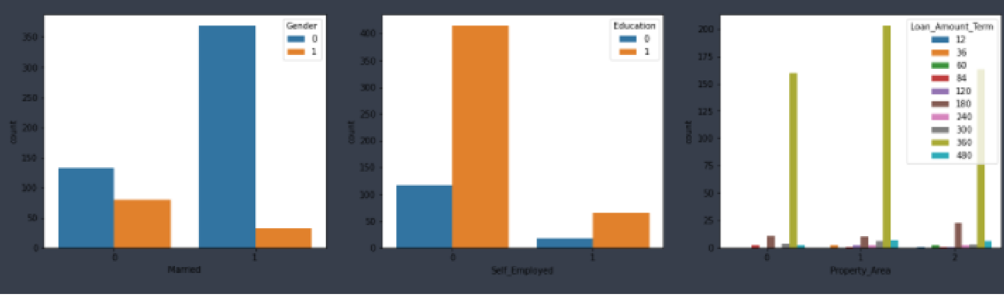
```
#changing the datype of each float column to int
data['Gender'] = data['Gender'].astype('object')
data['Married'] = data['Married'].astype('object')
data['Dependents'] = data['Dependents'].astype('object')
data['Self_Employed'] = data['Self_Employed'].astype('object')
data['CoapplicantIncome'] = data['CoapplicantIncome'].astype('float64')
data['LoanAmount'] = data['LoanAmount'].astype('float64')
data['Loan_Amount_Term'] = data['Loan_Amount_Term'].astype('float64')
data['Credit_History'] = data['Credit_History'].astype('object')
```

```python
#Balancing the dataset by using smote
from imblearn.combine import SMOTETomek



smote = SMOTETomek(0.90)


#dividing the dataset into dependent and independent y and x respectively
y=data['Loan_Status']
x=data.drop(columns=['Loan_Status'],axis=1)



#creating a new x and y varialbles for the balanced set
x_bal , y_bal = smote.fit_resample(x,y)



#printing the values of y before balancing the data and after
print(y.value_counts())
print(y.value_counts())



data.describe()



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


```
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(data['Gender'])
plt.subplot(1,4,2)
sns.countplot(data['Education'])
plt.show()
```

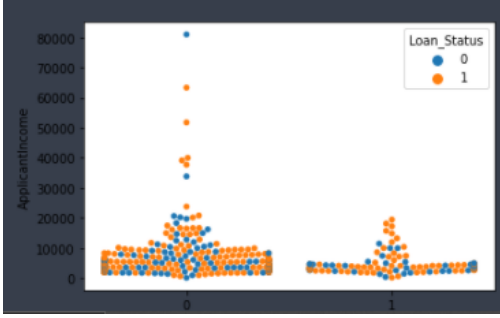

```
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(data['Married'],hue=data['Gender'])
plt.subplot(132)
sns.countplot(data['Self_Employed'],hue=data['Education'])
plt.subplot(133)
sns.countplot(data['Property_Area'],hue=data['Loan_Amount_Tera'])
```

```
error or misinterpretation.
  warnings.warn(
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From v
ersion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(

<AxesSubplot:xlabel='Property_Area', ylabel='count'>
```




```
#visualaized based gender and income what would be the application status
sns.swarmplot(data=['Gender'], data=['ApplicantIncome'], hue = data['Loan_
Status'], palette="deep")
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
C:\Users\HP\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 67.1% of the points cannot be placed; you may want to de
crease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\HP\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 33.0% of the points cannot be placed; you may want to de
crease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)

<AxesSubplot:xlabel='Gender', ylabel='ApplicantIncome'>
```




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


X_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_s
tate=42)
```

```python
def decisionTree(x_train,x_test,y_train,y_test):
  dt=DecisionTreeClassifier()
  dt.fit(x_train,y_train)
  yPred=dt.predict(x_test)
  print('***DecisionTreeClassifier***')
  print('Confusion matrix')
  print(confusion_matrix(y_test,yPred))
  print('Classification report')
  print(classification_report(y_test,yPred))


def randomForest(x_train,x_test,y_train,y_test):
  rf=DecisionTreeClassifier()
  rf.fit(x_train,y_train)
  yPred=rf.predict(x_test)
  print('***RandomForestClassifier***')
  print('Confusion matrix')
  print(confusion_matrix(y_test,yPred))
  print('Classification report')
  print(classification_report(y_test,yPred))




def KNN(x_train,x_test,y_train,y_test):
  knn=DecisionTreeClassifier()
  knn.fit(x_train,y_train)
  yPred=knn.predict(x_test)
  print('***KNeighborsClassifier***')
  print('Confusion matrix')
  print(confusion_matrix(y_test,yPred))
  print('Classification report')
  print(classification_report(y_test,yPred))


def xgboost(x_train,x_test,y_train,y_test):
  xg=DecisionTreeClassifier()
  xg.fit(x_train,y_train)
  yPred=xg.predict(x_test)
  print('***GradientBoostingClassifier***')
  print('Confusion matrix')
  print(confusion_matrix(y_test,yPred))
  print('Classification report')
  print(classification_report(y_test,yPred))
```

```
#Importing the keras libraries and packages
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

#Initialising the ANN
classifier = Sequential()
#Adding the second hidden layer
classifier.add(Dense(units=100,activation='relu',input_dim=11))
#Adding the second hidden layer
classifier.add(Dense(units=50,activation='relu'))
#Adding the output layer
classifier.add(Dense(units=1,activation='sigmoid'))
#compiling the ANN
classifier.compile(optimizer='adam',loss='binary_crossentropy',metrics=['a
ccuracy'])
#Fitting the ANN to the Training set
model_history=classifier.fit(X_train, y_train, batch_size=100, validation_
split=0.2, epochs=100)

#Gender married Dependents Education Self_Employed ApplicantIncome
CoapplicantIncome LoanAmount Loan_Amount_tree Credit_history Property_Area
dtr.predict([[1,1,0,1,1,4276,1542,145,240,0,1]])


#Gender married Dependents Education Self_Employed ApplicantIncome
CoapplicantIncome LoanAmount Loan_Amount_tree Credit_history Property_Area
xgb.predict([[1,1,0,1,1,4276,1542,145,240,0,1]])


classifier.save("loan.h5")

#predicting the test set results
y_pred=classifier.predict(x_test)


y_pred

y_pred=(y_pred>0.5)
y_pred


def predict_exit(sample_value):
#convert list to numpy array
sample_value=np.array(sample_value)
```

```
#Reshape because sample_value contains only 1 record
sample_value=sample_value.reshape(1,-1)

#Feature scaling
sample_value=sc.transform(sample_value)
return classifier.predict(sample_value)


#predictions
#value order 'Creditscore','Age','Tenure','Balance','Numofproducts','HasCr
Card','IsActiveMember','France','Germany','Spain','Female','Male'.
sample_value=[[1,1,0,1,1,4276,1542,145,240,0,1]]
if predict_exit(sample_value)>0.5:
     print('Prediction:High chance of loan Approval.')
else:
     print('Prediction:low chance of loan Approval.')


#predictions
#value order 'Creditscore','Age','Tenure','Balance','Numofproducts','HasCr
Card','IsActiveMember','France','Germany','Spain','Female','Male'.
sample_value=[[1,0,1,1,1,45,14,45,240,1,1]]
if predict_exit(sample_value)>0.5:
     print('Prediction:High chance of loan Approval.')
else:
     print('Prediction:low chance of loan Approval.')


def compareModel(x_train,x_test,y_train,y_test):
   decisionTree(x_train,x_test,y_train,y_test)
   print('-'*100)
   RandomForest(x_train,x_test,y_train,y_test)
   print('-'*100)
   XGB(x_train,x_test,y_train,y_test)
   print('-'*100)
   KNN(x_train,x_test,y_train,y_test)
   print('-'*100)


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

```
------------------------------------------------------------------
1.0
0.8088888888888889
Random Forest
Confusion_Matrix
[[ 78  29]
 [ 14 104]]
Classification Report
              precision    recall  f1-score   support

           0       0.85      0.73      0.78       107
           1       0.78      0.88      0.83       118

    accuracy                           0.81       225
   macro avg       0.81      0.81      0.81       225
weighted avg       0.81      0.81      0.81       225

------------------------------------------------------------------
```

```
1.0
0.7822222222222223
Decision Tree
Confusion_Matrix
[[83 24]
 [25 93]]
Classification Report
              precision    recall  f1-score   support

           0       0.77      0.78      0.77       107
           1       0.79      0.79      0.79       118

    accuracy                           0.78       225
   macro avg       0.78      0.78      0.78       225
weighted avg       0.78      0.78      0.78       225

------------------------------------------------------------------------------------------
```

```
yPred=classifier.predict(x_test)
 print(accuracy_score(yPred,y_test)
 print("ANN Model")
 print('Confusion matrix')
 print(confusion_matrix(y_test,yPred))
 print('Classification report')
 print(classification report(y_test,yPred))
```

```
8/8 [==============================] - 0s 4ms/step
0.6844444444444444
ANN Model
Confusion_Matrix
[[63 44]
 [27 91]]
Classification Report
              precision    recall  f1-score   support

           0       0.70      0.59      0.64       107
           1       0.67      0.77      0.72       118

    accuracy                           0.68       225
   macro avg       0.69      0.68      0.68       225
weighted avg       0.69      0.68      0.68       225
```

```
from sklearn.model_selection import cross_val_score
#Random forest model is selected
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)
```

```
0.9691629955947136
0.8222222222222222
Random Forest
Confusion_Matrix
[[ 77  30]
 [ 10 108]]
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.72      0.79       107
           1       0.78      0.92      0.84       118

    accuracy                           0.82       225
   macro avg       0.83      0.82      0.82       225
weighted avg       0.83      0.82      0.82       225

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    0.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   2 out of   2 | elapsed:    0.0s remaining:    0.0s
```

```
#saving the model by using pickle function
pickle.dump(model,open('rdf.pkl','wb'))
```

Html Coding:

```
<!DOCTYPE html>
<html>
<head>
  <title>Loan Approval Form</title>
</head>
<body>
  <h1>Personal Loan Approval</h1>
  <form id="loanForm">
    <label for="income">Income:</label>
    <input type="text" id="income" name="income" required>
    <br>
    <label for="creditScore">Credit Score:</label>
    <select id="creditScore" name="creditScore" required>
      <option value="">-- Select --</option>
      <option value="excellent">Excellent</option>
      <option value="good">Good</option>
      <option value="fair">Fair</option>
      <option value="poor">Poor</option>
    </select>
    <br>
    <label for="loanAmount">Loan Amount:</label>
    <input type="text" id="loanAmount" name="loanAmount" required>
    <br>
    <input type="submit" value="Submit">
  </form>
  <script>
    document.getElementById('loanForm').addEventListener('submit',
function(event) {
      event.preventDefault(); // prevent form submission

      // Get input values
      var income = parseFloat(document.getElementById('income').value);
      var creditScore = document.getElementById('creditScore').value;
      var loanAmount =
parseFloat(document.getElementById('loanAmount').value);

      // Perform loan approval logic (this is a simplified example)
      var approvalStatus = '';
      if (income >= 50000 && creditScore === 'excellent' && loanAmount <=
50000) {
        approvalStatus = 'Approved';
      } else {
        approvalStatus = 'Not Approved';
      }
```

```
      // Display result
      alert('Loan Approval Status: ' + approvalStatus);
    });
  </script>
</body>
</html>
```

pycharm coding:

```
from flask import flask, render_template, request
import numpy as np
import pickle
app=flask(_name_)
model=pickle.load(open(r'rdf.pkl','rb'))
scale=pickle.load(open(r'scale1.pkl','rb'))
@app.route('/')
Def home:
     Return render_template('home.html')

@app.route('/') # rendering the html template
def home():
   return render_template('home.html')

@app.route('/submit',methods=["POST","GET"])# route to show the prediction
in a web UI
def submit():
   # reading the inputs givenby the user
   input_feature=[int(x) for x in request.form.values()]
   #input feature=np.transpose(input_feature)
   input feature=[np.array(input_feature)]
   print(input_feature)

names=['Gender','Married','Dependents','Education','Self_Employed','Applic
antIncome','CoapplicantIncome','LoanAmount','Loan_amount_Term','Credit_his
tory','Property_Area']
   data=pandas.DataFrame(input_feature,columns=names)
   print(data)


   #data_scaled=scale.fit_transform(data)
   #data=pandas.DataFrame(,columns=names)

   #predictions using the loaded modelfile
   prediction=model.predict(data)
   print(prediction=int(prediction)
```

```
   print(type(prediction))

   if(prediction==0):
       return render_template("output.html",result="Loan will not be
Approved")
   else:
      return render_template("output.html",result="Loan will be Approved")
   #showing the prediction results in a UI
if_name_=="_main_":
   # app.run(host='0.0.0.0',port=8000,debug=True)        #running the app
port=int(os.environ.get('PORT',5000))
app.run(debug=False)
```