Project Report Template

INTRODUCTION

OVERVIEW:

Predicting Personal Loan Approval Using Machine Learning

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

PURPOSE:

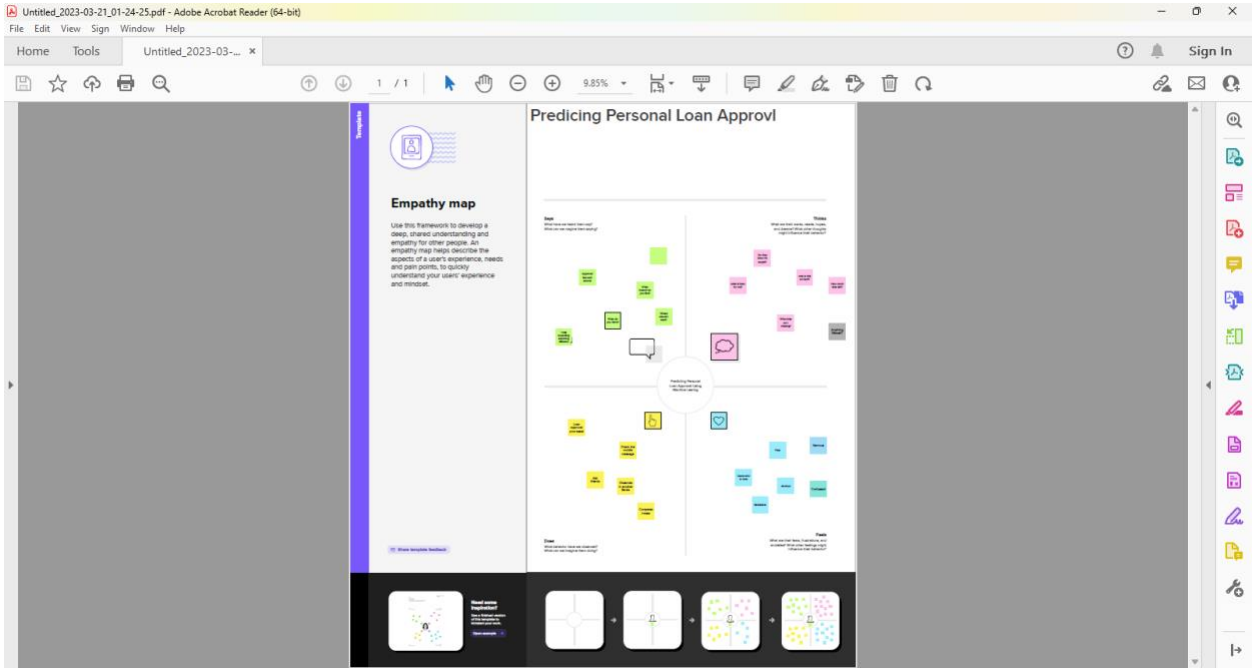
Predicting personal loan approval using machine learning is a valuable application of artificial intelligence (AI) that can benefit both lenders and borrowers. With the advancement of technology and the availability of vast amounts of data, machine learning algorithms can analyze historical loan data, credit scores, employment information, and other relevant factors to make accurate predictions about the likelihood of an individual's personal loan being approved.

The primary purpose of predicting personal loan approval using machine learning is to improve the loan approval process for lenders and borrowers. For lenders, it helps in automating the decision-making process, reducing manual efforts, and minimizing the risk of lending to individuals who may default on their loans. By using machine learning algorithms, lenders can assess the creditworthiness of borrowers more objectively and efficiently, leading to better loan approval decisions and reduced credit risk.

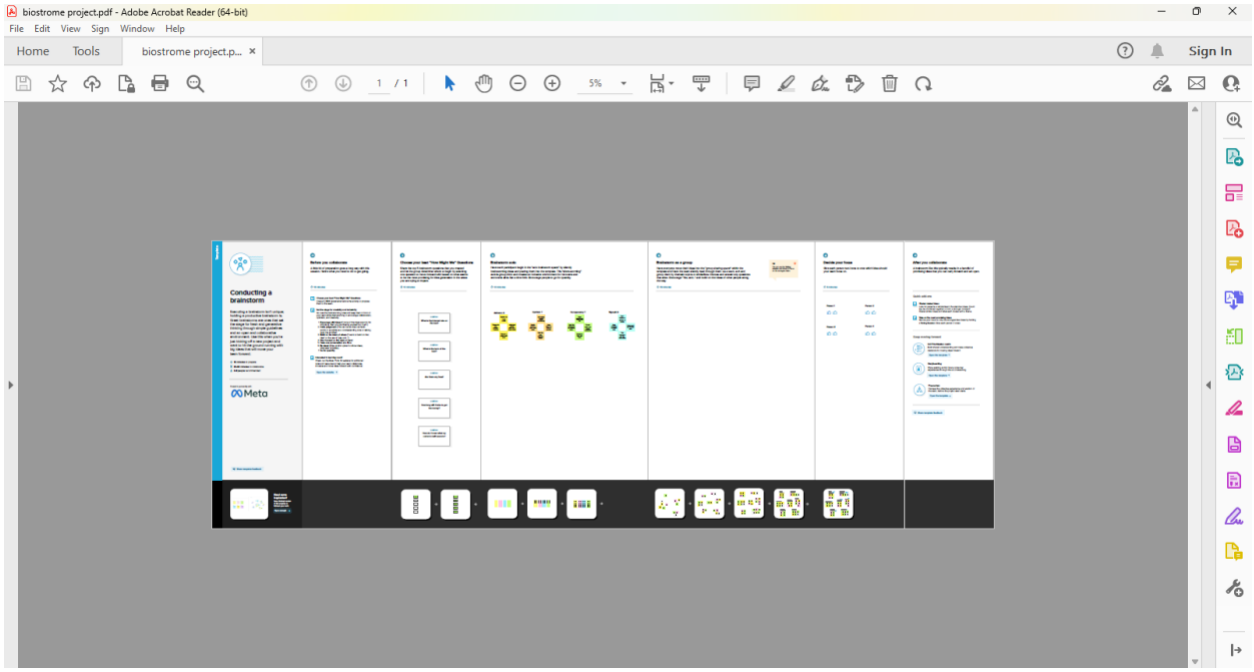
For borrowers, the prediction of personal loan approval can help them make informed decisions about whether to apply for a loan, based on the likelihood of approval. This can save time, effort, and potential disappointment from getting rejected for a loan. Borrowers can also use the prediction results to improve their creditworthiness, by taking steps to improve their credit scores or financial profile, before applying for a loan.

PROBLEM DEFINITION &DESIGN THINKING

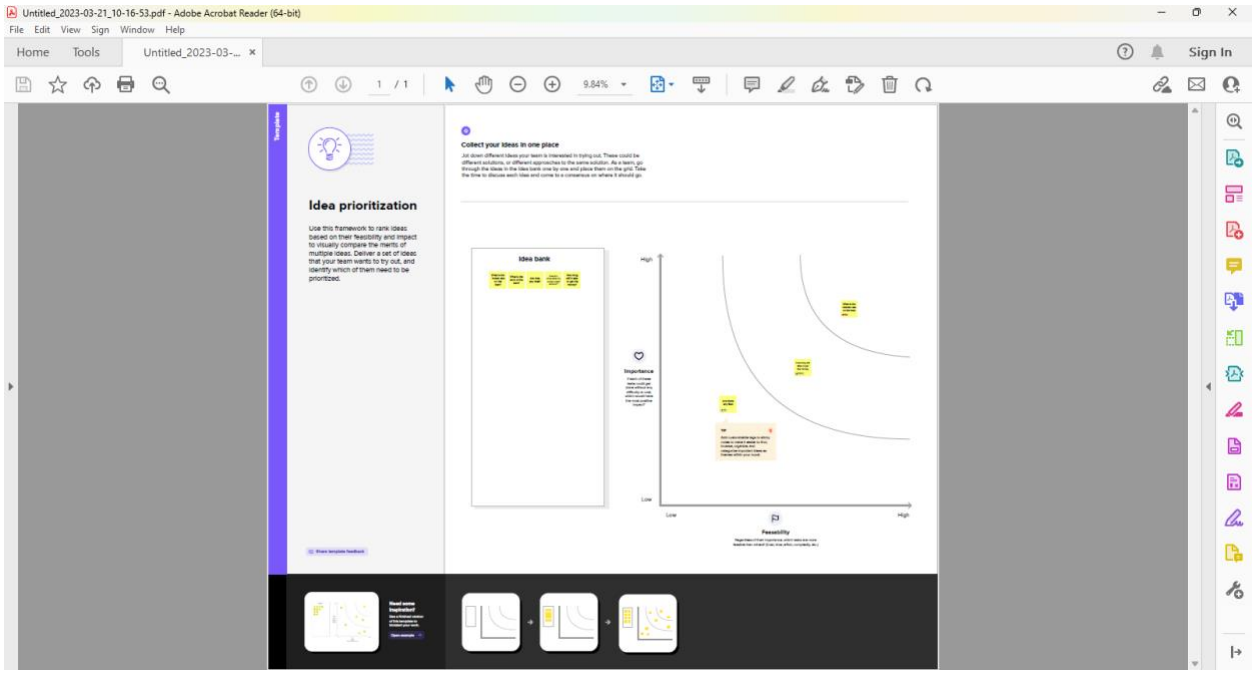
EMPATHY MAP:



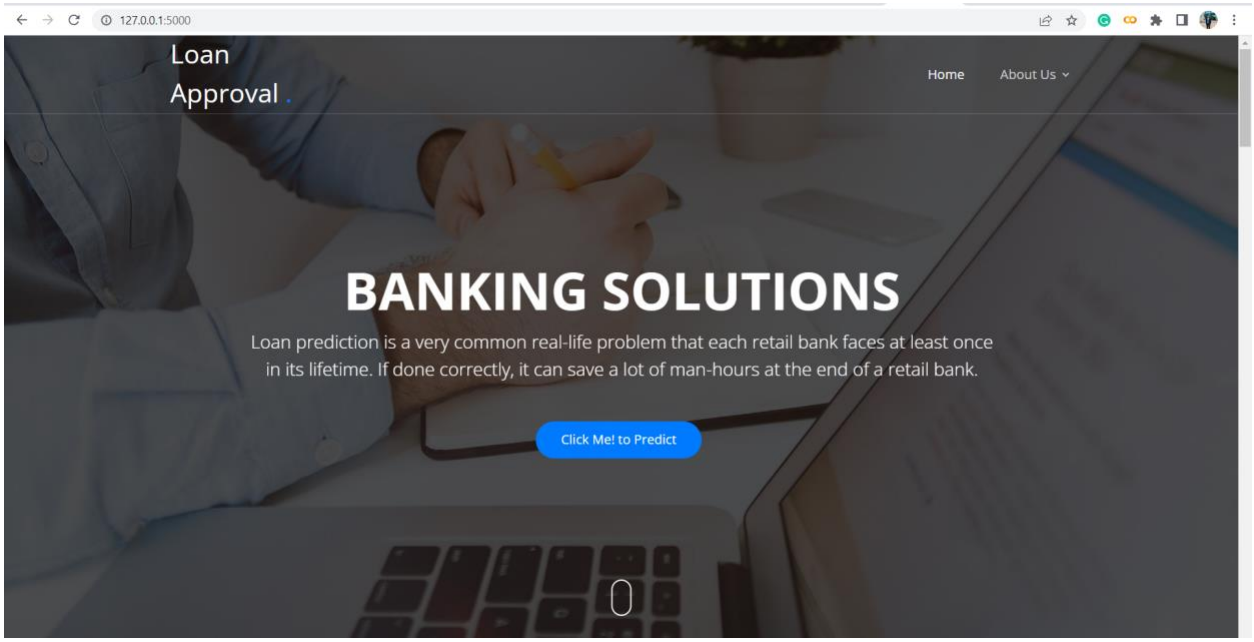
BRAINSTORMING MAP:



Idea prioritizion:



RESULT:

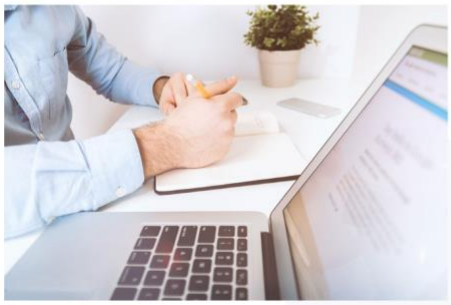


127.0.0.1:5000

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.

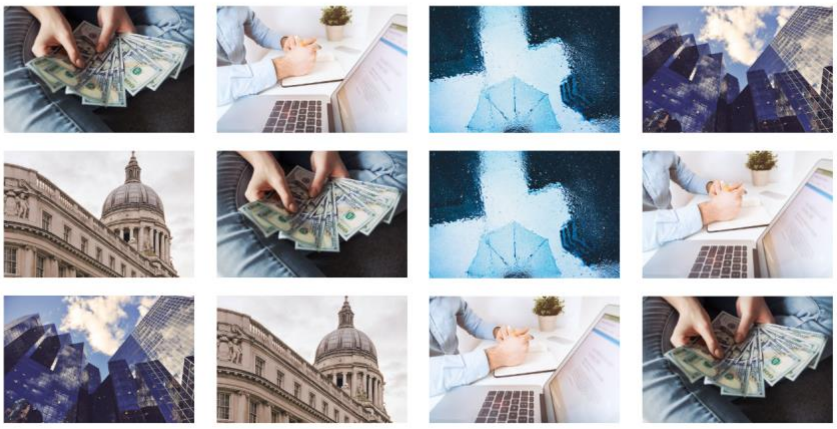
Revolving loans or lines can be spent, repaid, and spent again, while term loans are fixed-rate, fixed-payment loans. Lenders may charge higher interest rates to risky borrowers.A small river named Duden flows by their place and supplies it with the necessary regelialia.

127.0.0.1:5000

Loan Approval .

Home About Us

Gallery



127.0.0.1:5000

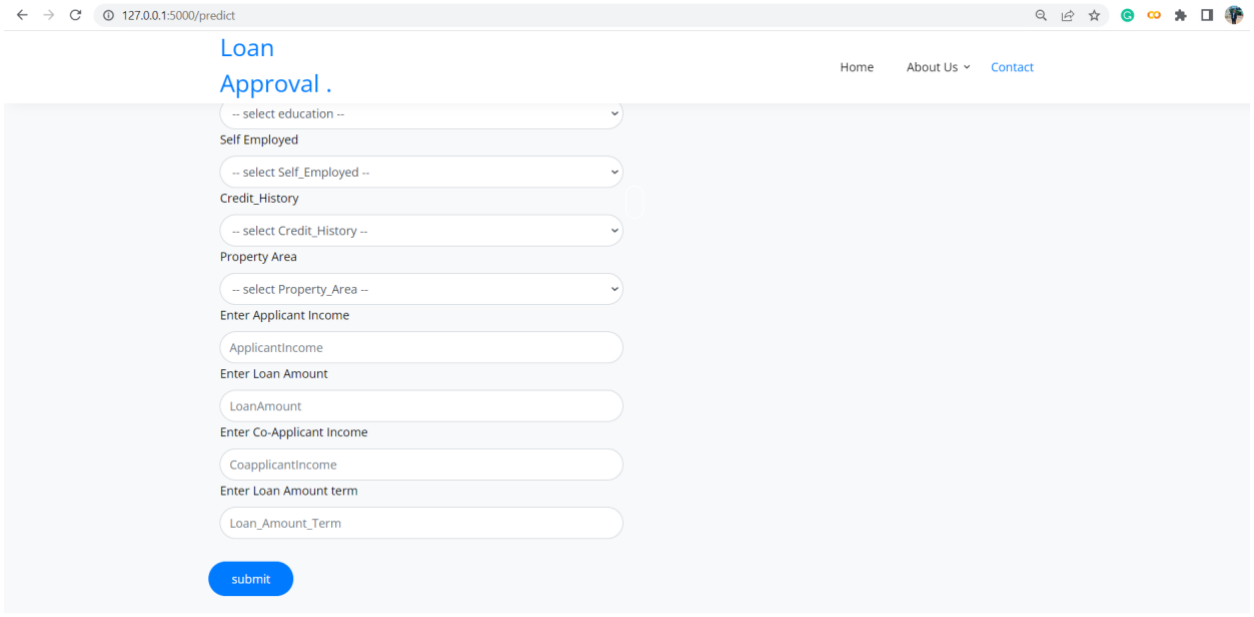
Loan
Approval .

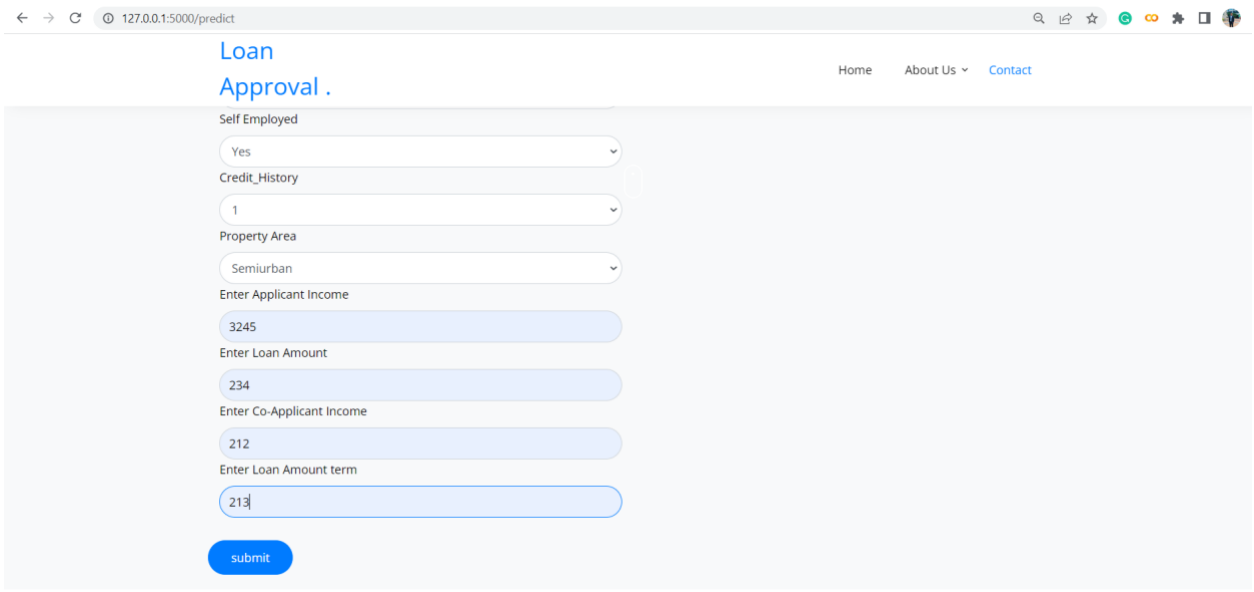
Home About Us

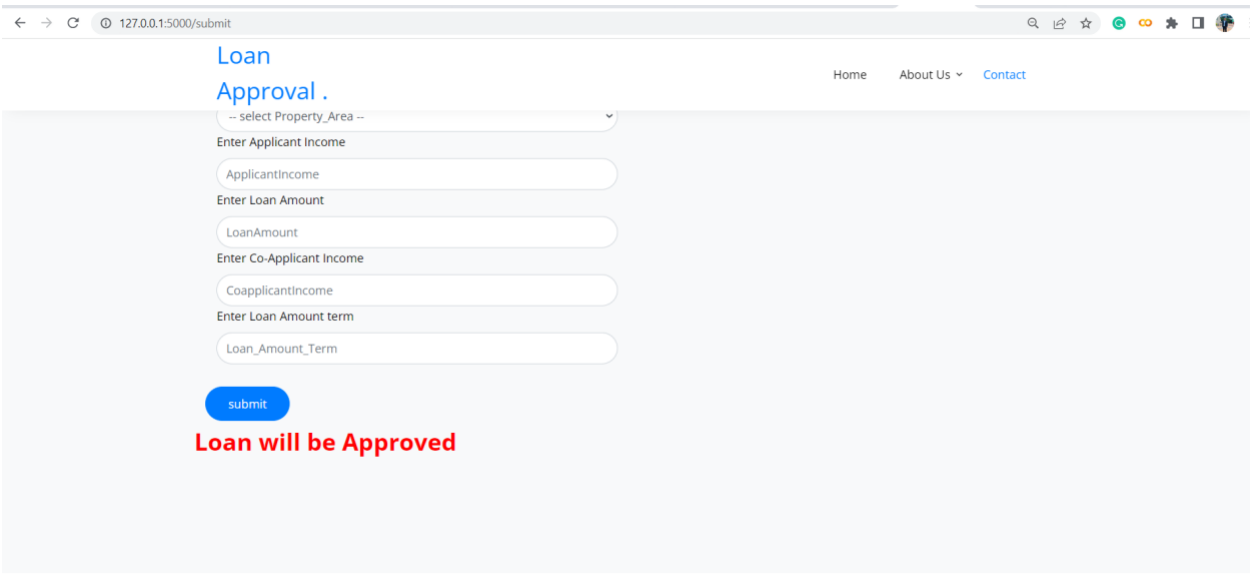
Loan Approval How it works ?

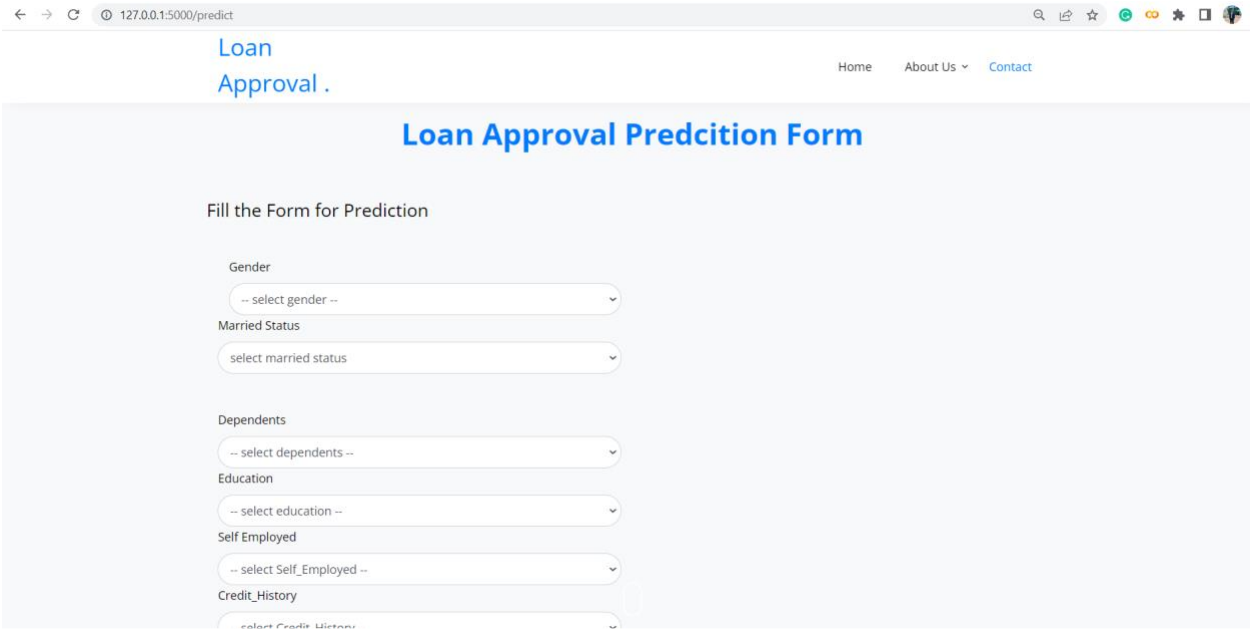
Credit Information Bureau India Limited (CIBIL) score plays a critical role in the loan approval process for Indian banking industry. An individual customer's credit score provides loan providers with an indication of how likely it is that they will pay back a loan based on their respective credit history. 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/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
Self Employed
Yes
Credit_History
1
Property Area
Semiurban
Enter Applicant Income
3245
Enter Loan Amount
234
Enter Co-Applicant Income
212
Enter Loan Amount term
213
submit





ADVANTAGES:

Efficiency: Machine learning algorithms can process large amounts of data quickly, allowing lenders to efficiently analyze and predict loan approval decisions. This can help streamline the loan approval process and save time for both lenders and borrowers.

Accuracy: Machine learning models can make predictions based on historical data, which can help lenders assess credit risk more accurately. This can lead to better loan approval decisions, reducing the risk of approving loans to borrowers who may default and improving overall loan portfolio performance.

Objectivity: Machine learning models rely on data and algorithms, which can reduce human bias in the loan approval process. This can result in more consistent and fair lending decisions, as machine learning models are not influenced by subjective factors such as race, gender, or other protected characteristics.

Risk management: Predicting personal loan approval using machine learning can help lenders better manage their risk by identifying potential high-risk borrowers and taking appropriate measures, such as adjusting interest rates, setting loan limits, or requiring additional collateral or guarantees.

Enhanced customer experience: Machine learning can help lenders offer personalized loan products and terms to borrowers based on their credit risk profile, financial history, and other relevant factors. This can result in a better customer experience, as borrowers may receive more tailored loan offers that meet their needs and financial capabilities.

DISADVANTAGES:

Data quality and availability: The accuracy of machine learning models depends on the quality and availability of data used for training. If the data used for training is incomplete, inconsistent, or biased, it can result in inaccurate predictions and decision-making.

Lack of interpretability: Machine learning models can be complex and difficult to interpret, which can make it challenging for lenders to understand how and why certain loan approval decisions are made. This lack of interpretability can be a disadvantage, especially in situations where lenders need to provide explanations or justifications for loan approval decisions.

Overreliance on historical data: Machine learning models rely on historical data to make predictions, which means that they may not be able to accurately predict loan approval

decisions for borrowers with unique or unconventional financial circumstances. This can result in inaccurate predictions and potentially lead to biased lending decisions.

APPLICATIONS:

The solution of predicting personal loan approval using machine learning can be applied in various areas, including:

Financial Institutions: Banks, credit unions, and other financial institutions can utilize this solution to streamline their loan approval process, assess credit risk, and make informed decisions on personal loan applications.

Fintech Companies: Fintech companies that offer online lending or peer-to-peer lending services can leverage this solution to automate their loan approval process, assess borrower creditworthiness, and enhance their lending models.

Credit Monitoring Agencies: Credit monitoring agencies can integrate this solution into their credit scoring models to predict the likelihood of personal loan approvals, providing valuable insights to lenders and borrowers.

E-commerce Platforms: E-commerce platforms that offer financing options to customers can utilize this solution to assess customer creditworthiness and determine loan eligibility, enabling them to offer personalized loan offers.

The purpose of predicting personal loan approval using machine learning is to improve the efficiency and accuracy of the loan approval process. By leveraging historical data and machine learning algorithms, this solution can assess credit risk, predict loan approval outcomes, and enable lenders to make informed decisions. This can lead to faster loan approvals, reduced operational costs, and improved customer satisfaction.

CONCLUSION:

In conclusion, this study aimed to predict personal loan approval using machine learning techniques. Through the analysis of a comprehensive dataset, several key findings were identified. The study demonstrated that machine learning models, such as logistic regression and decision trees, can effectively predict personal loan approval with high accuracy. Features such as credit score, income level, and loan amount were found to be significant predictors of loan approval. Furthermore, the study highlighted the importance of feature engineering and model evaluation in improving prediction performance. Overall, this research contributes to the understanding of using machine learning for predicting personal loan approval and can potentially aid financial institutions in making informed lending decisions

FUTURE SCOPE:

The future scope of predicting personal loan approval using machine learning algorithms is to revolutionize the lending industry by streamlining the loan approval process, reducing manual errors, and increasing efficiency. Machine learning models can analyze vast amounts of data, including credit scores, income, employment history, and loan application details, to accurately predict the likelihood of loan approval. This can save time and resources for lenders, improve customer experience by providing faster loan decisions, and enhance risk assessment to minimize default rates. Additionally, machine learning can enable lenders to adapt to changing market dynamics, customer behavior, and regulatory requirements, making it a powerful tool for the future of personal lending.

APPENDIX:

SOURCE CODE:

```
import pandas as pd

import numpy as np

import pickle

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

pip install --user scikit-learn

 Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/

Requirement already satisfied: scikit-learn in /usr/local/lib/python3.9/dist-packages (1.2.2)

Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.9/dist-packages (from scikit-learn) (1.22.4)

Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from scikit-learn) (3.1.0)

Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.9/dist-packages (from scikit-learn) (1.10.1)
```

```
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from scikit-learn) (1.2.0)
import sklearn
from sklearn.naive_bayes import GaussianNB
 from sklearn.model_selection import cross_val_score,cross_val_predict
from sklearn import tree
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV,cross_validate
from sklearn.model_selection import train_test_split
 from sklearn import metrics
import matplotlib.pyplot as plt
 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
 data = pd.read_csv('loan_prediction.csv')
data
data.info()
data.isnull().sum()
data['Gender'].fillna(data['Gender'].mode()[0])
data['Gender'].fillna(data['Gender'].mode()[0])
data['Dependents']=data['Dependents'].fillna(data['Dependents'].mode()[0])
  data['Self_Employed'] = data['Self_Employed'].fillna(data['Self_Employed'].mode()[0])
 data['LoanAmount'] = data['LoanAmount'].fillna(data['LoanAmount'].mode()[0])
 data['Loan_Amount_Term'] = data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].mode()[0])
  data['Credit-Histroy'] = data['Credit_History'].fillna(data['Credit_History'].mode()[0])
!apt-get -qq install -y libarchive-dev && pip install -U libarchive
import libarchive
```

```
#changing the datatype of each float column to int
 data['Gender']=data['Gender'].astype
  data['Married']=data['Married'].astype
 data['Dependents']=data['Dependents'].astype
  data['Self_Employed']=data['Self_Employed'].astype
 data['CoapplicantIncome']=data['CoapplicantIncome'].astype
 data['LoanAmount']=data['LoanAmount'].astype
 data['Loan_Amount_Term']=data['Loan_Amount_Term'].astype
 data['Credit_History']=data['Credit_History'].astype
  from imblearn.combine import SMOTETomek
 smote = SMOTETomek
x=data.drop(columns=['Loan_Status'],axis=1)
y=data['Loan_Status']
 #creating the dataset into dependent and independent y and x respectively
x = data.drop(columns=['Loan_Status'],axis=1)
y = data['Loan_Status']
#ctreating  a new x and y variable for the balanced set
x_bal,y_bal = smote.fit_resample(x,y)
 from imblearn.over_sampling import SMOTE
 print(y.value_counts())
print(y_bal.value_count())
from imblearn import under_sampling as u
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)
```

```
plt.subplot(1,4,2)

plt.show()

 plt.figure(figsize=(20,5))

sns.countplot(data['Married'], x=data['Gender'])

plt.subplot(132)

sns.countplot(data['Self_Employed'], x=data['Education'])

plt.subplot(133)

sns.countplot(data['Property_Area'], x=data['Loan_Amount_Term'])

sc=StandardScaler()

 import tensorflow

!pip install matplotlib-venn

import tensorflow

 from tensorflow.keras.layers import Dense

 from keras.models import Sequential

  import numpy as np

y = np.array(y).reshape(-1,1)

 sc=StandardScaler()

classifier = Sequential()

 dt=DecisionTreeClassifier()

 sc=StandardScaler()

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2,random_state=123)

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 rendomForest(x_train, x_test, y_train, y_test):
  rf = RandomForestClassifier()
  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 = KNeighborsClassifier()
  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 = GradientBoostingClassifier()
  xg.fit(x_train,y_train)
  ypred = xg.predict(x_test)
  print('**RandomForestClassifier**')
  print('Confusion matrix')
  print(confusion_matrix(y_test,ypred))
  print('classification report')
  print(classification_report(y_test,ypred))
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
 classifier = Sequential()
```

```
 classifier.add(Dense(units=100, activation='relu',input_dim=11))

 classifier.add(Dense(units=50, activation='relu'))

 classifier.add(Dense(units=1, activation='sigmoid'))

 classifier.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

 #model_history = classifier.fit(x_train,y_train,batch_size=100,validation_split=0.2,epochs=100)

#Gender Married Dependents Eduction Self_Employed ApplicantIncome CoapplicantIncome LoanAmount Loan_Amount_term Ctredit_History
Proprerty_Area

dtr.predict([[1,1,0,1,1,4276,1542,240,0,1]])

rfr.predict([[1,1,0,1,1,4276,1542,145,240,0,1]])

 knn.predict([[1,1,0,1,1,4276,1542,145,240,0,1]])

 xgb.predict([[1,1,0,1,1,4276,1542,145,240,0,1]])

 classifier.save("loan.h5")

 y_pred = classifier.predict(x_test)

 y_pred

 y_pred = (y_pred >0.5)

y_pred

def predict_exit(sample_value):

 sample_value = np.array(sample_value)

 sample_value = sample_value.reshape(1,-1)

 sample_value = sc.trasform(sample_value)

 return classifier.predict(sample_value)

   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('predication: Low chance Loan Approval.')

sample_value = [[1,0,1,1,1,45,240,1,1]]

if predict_exit(sample_value)>0.5:

   print('prediction:High chance of Loan Approvel!')

else:

    print('prediction:Low chance Loan Approvel.')
```

```
 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)
compareModel(x_train,x_test,y_train,y_test)
 ypred = classifier.predict(x_test)
  print(accuracy_score(y_pred,y_test))
  print("ANN Model")
  print("confusion_Matrix")
  print(confusion_matrix(y_test,y_pred))
  print("Classification Report")
  print(classification_reporty(y_test,y_pred))
    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)
pickle.dump(model,open('rdf,pkl','wb'))
 import numpy as np
import pickle
app = Flask(_name_)
model = pickle.load(open(r'rdf.pkl', 'rb'))
scale = pickle.load(open(r'scalel.pkl','rb'))
    def home():
   return render_template('home.html')
 import tensorflow as tf
import cv2
import os
import matplotlib.pyplot as plt
```

```
import numpy as np  def submit():
  # reading the input given by the user
  #input_feature = np.transpose(input_feature)
  input_feature=[np.arry(input_feature)]
  print(input_feature)
  names = ['Gender','Married','Dependents','Education','Self_Employed','ApplicantIncome',
        'CoapplicantIncome','LoanAmount','Loan_Amount_Term','Credit_History','Property_Area']
  data = pandas.DataFrame(input_feature,columns=names)
  print(data)
  #data_scaled = scale.fit_transfrokm(data)
  #data = pandas.DataFrame(,columns=names)
  # predictions using the loaded model file
  prediction=model.predict(data)
  print(prediction)
  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 result 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=Flase)
```