PREDICTING PERSONAL LOAN APPROVAL USING MACHINE LEARNING

CONTENTS

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.

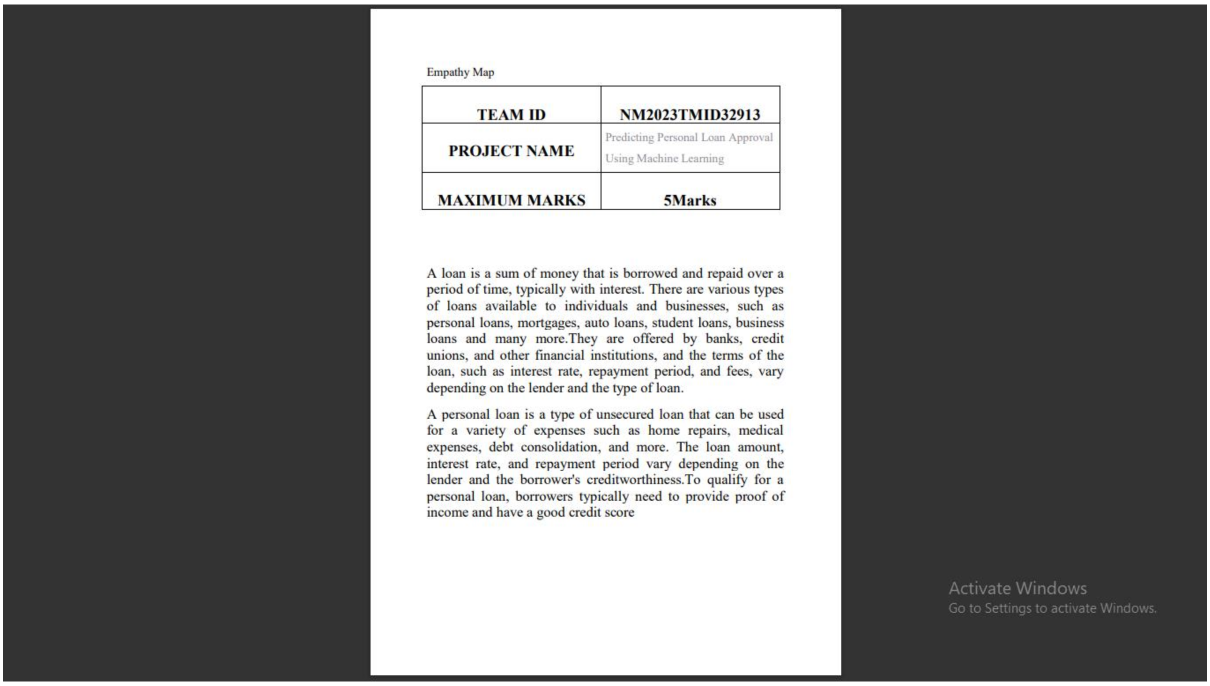
1.2Purpose

It is a classification problem where we have to predict whether a loan would be approved or not. In these kinds of problems, we have to predict discrete values based on a given set of independent variables.

It is done by predicting if the loan can be given to that person on the basis of various parameters like credit score, income, age, marital status, gender, etc. The prediction model not only helps the applicant but also helps the bank by minimizing the risk and reducing the number of defaulters.

PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map

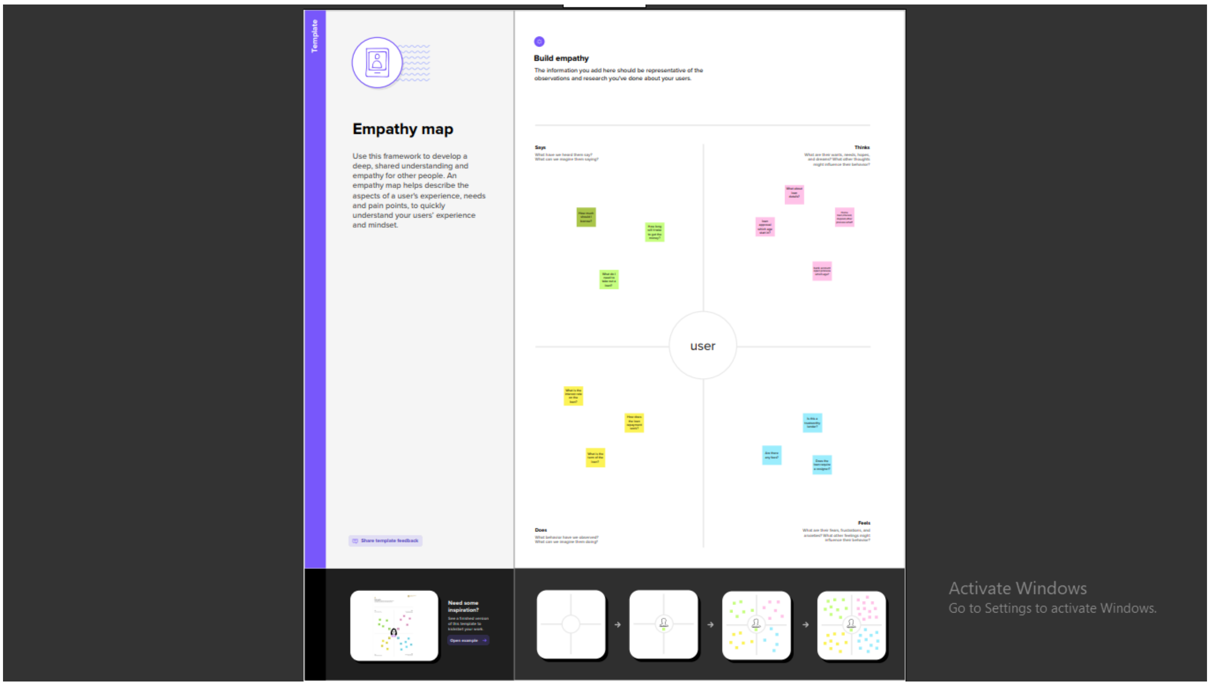


Empathy Map

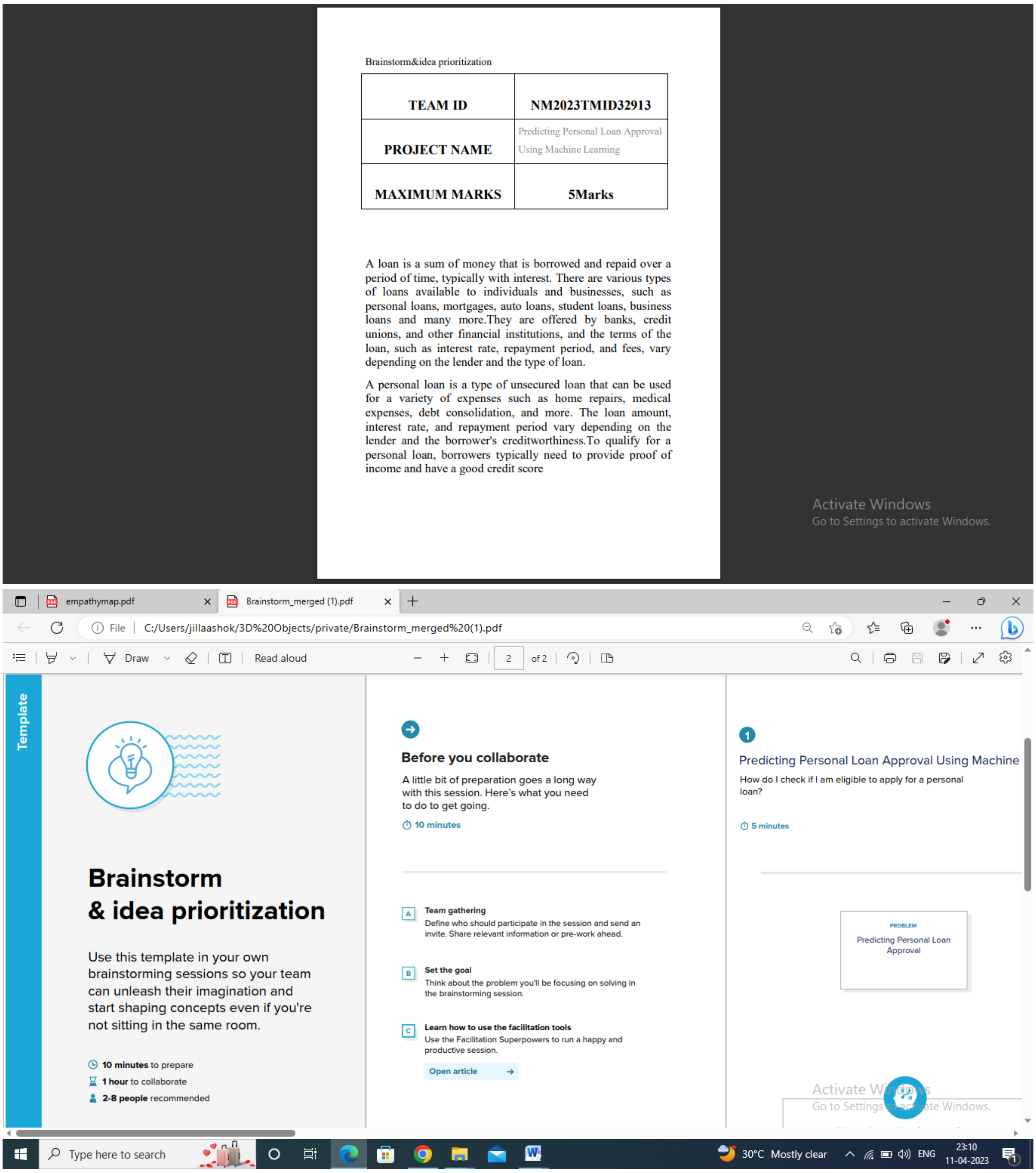
TEAM ID	NM2023TMID32913
PROJECT NAME	Predicting Personal Loan Approval Using Machine Learning
MAXIMUM MARKS	5Marks

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



2.2 Ideation & Brainstorming Map

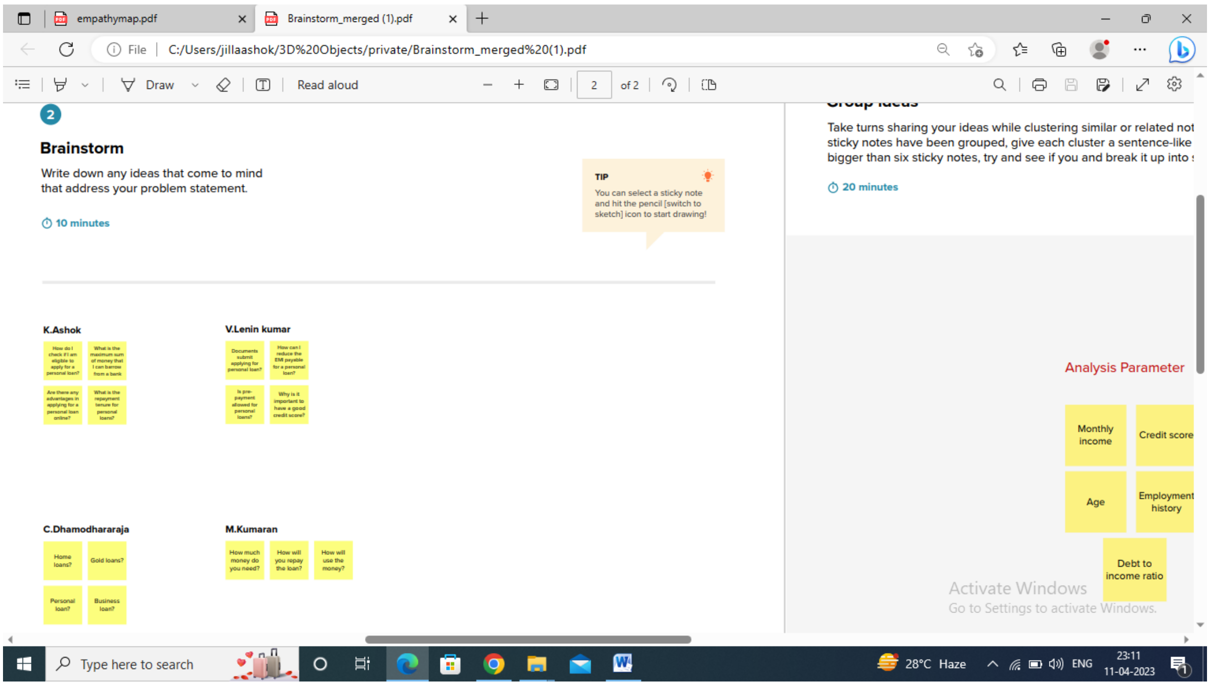


Brainstorm&idea prioritization

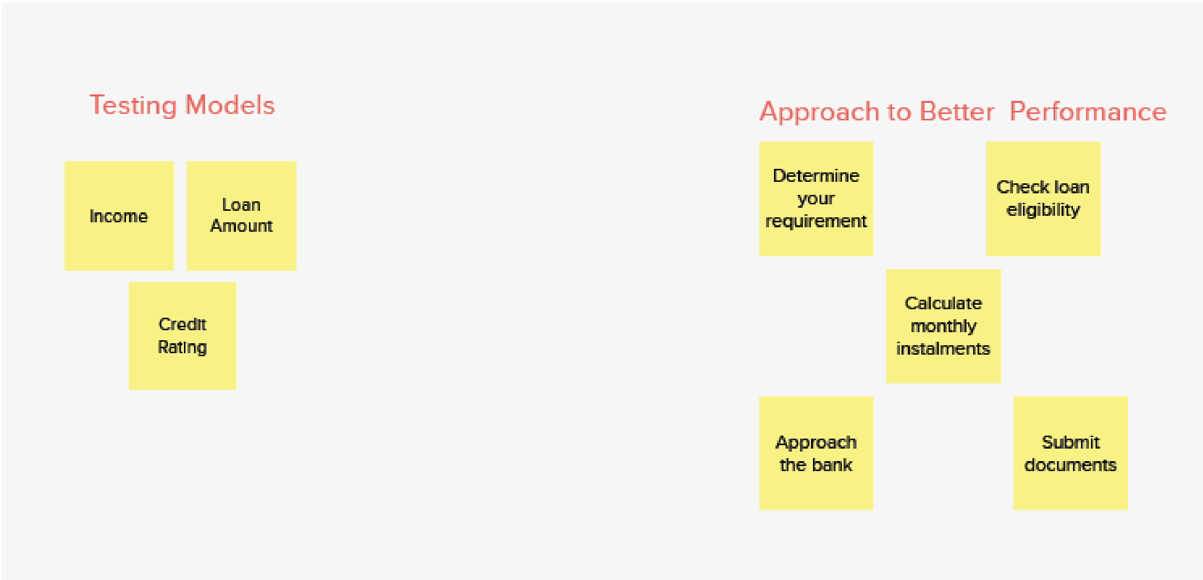
TEAM ID	NM2023TMID32913
PROJECT NAME	Predicting Personal Loan Approval Using Machine Learning
MAXIMUM MARKS	5Marks

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.

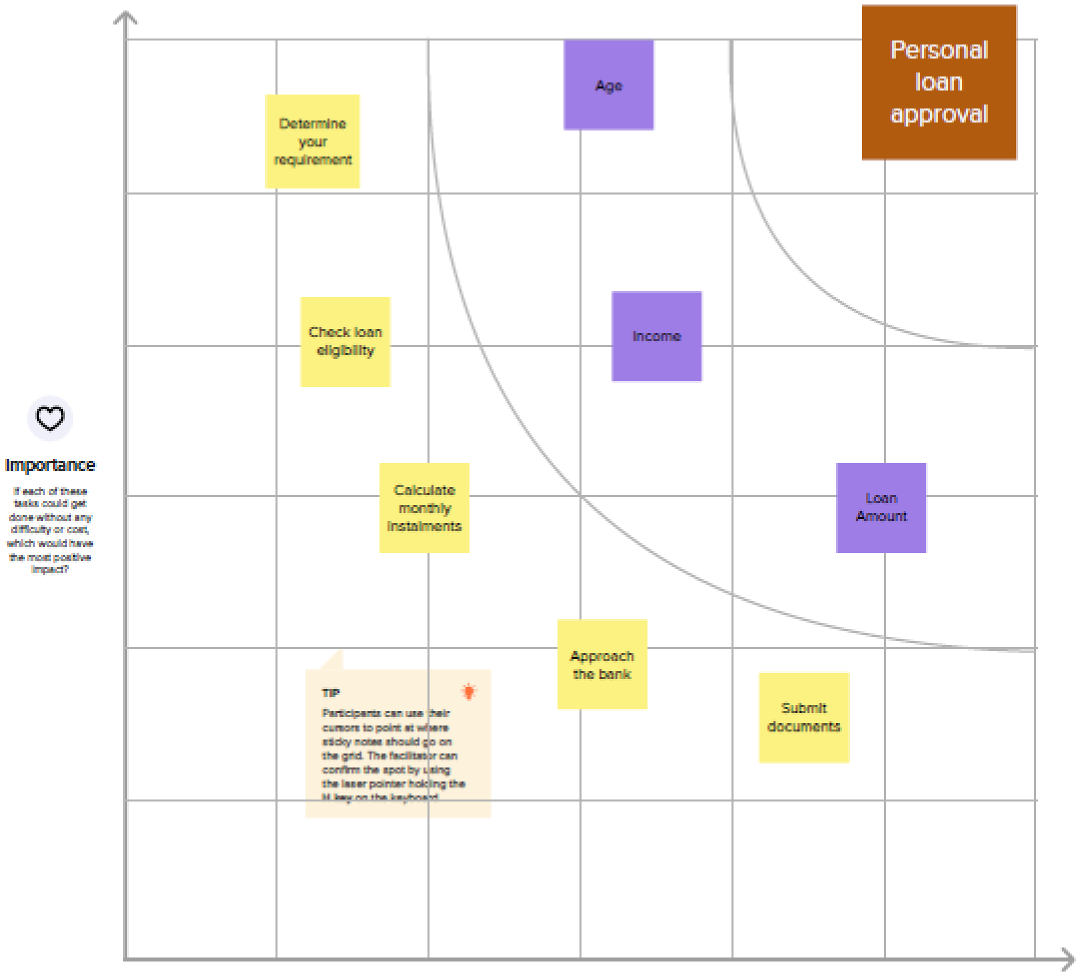
A personal loan is a type of unsecured loan that can be used for a variety of expenses such as home repairs, medical expenses, debt consolidation, and more. The loan amount, interest rate, and repayment period vary depending on the lender and the borrower's creditworthiness.To qualify for a personal loan, borrowers typically need to provide proof of income and have a good credit score



empathymap.pdf
Brainstorm_merged (1).pdf
File | C:/Users/jillaashok/3D%20Objects/private/Brainstorm_merged%20(1).pdf
Draw
Read aloud
2
of 2
2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!
K.Ashok
V.Lenin kumar
C.Dhamodhararaja
Home loans?
Gold loans?
Personal loan?
Business loan?
M.Kumaran
Take turns sharing your ideas while clustering similar or related not sticky notes have been grouped, give each cluster a sentence-like bigger than six sticky notes, try and see if you and break it up into
20 minutes
Analysis Parameter
Monthly income
Credit score
Age
Employment history
Debt to income ratio
Activate Windows
Go to Settings to activate Windows.
Type here to search
28°C Haze
ENG
23:11
11-04-2023

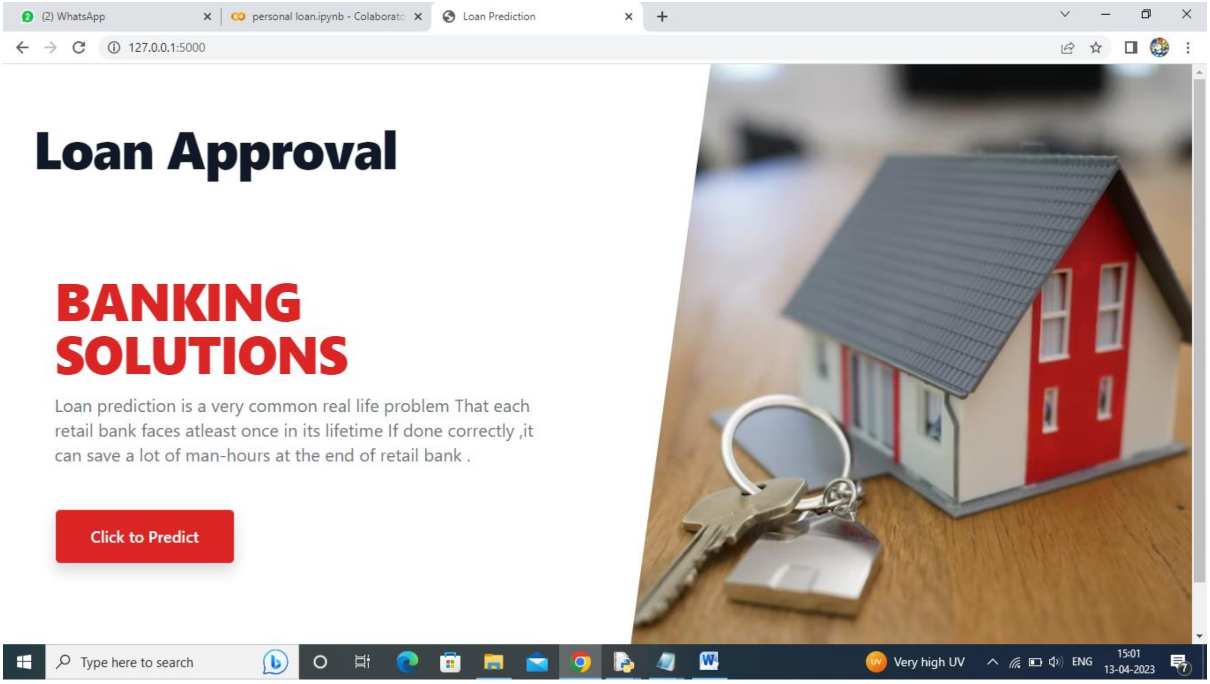


Testing Models
Income
Loan Amount
Credit Rating
Approach to Better Performance
Determine your requirement
Check loan eligibility
Calculate monthly instalments
Approach the bank
Submit documents

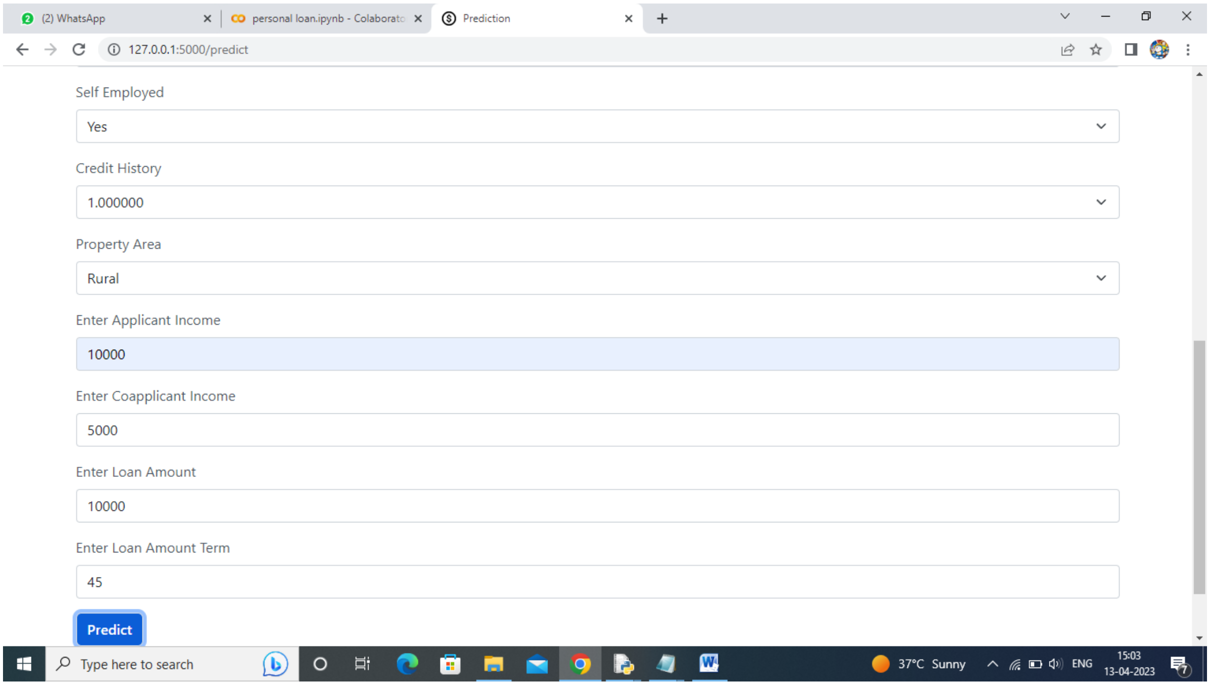

Personal loan approval
Age
Determine your requirement
Check loan eligibility
Income
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Calculate monthly instalments
Loan Amount
Approach the bank
Submit documents
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard!

RESULT

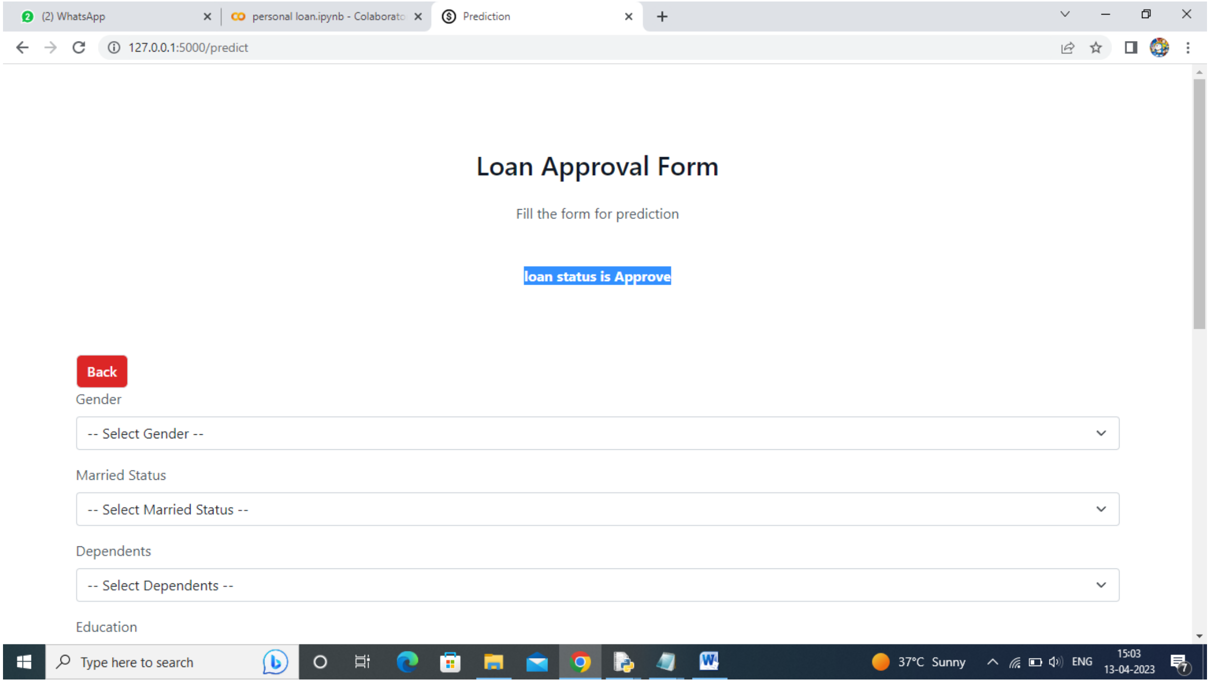
Main Page:



Prediction page :



Result :


(2) WhatsApp
personal loan.ipynb - Colaborato
Prediction
127.0.0.1:5000/predict
Loan Approval Form
Fill the form for prediction
loan status is Approve
Back
Gender
-- Select Gender --
Married Status
-- Select Married Status --
Dependents
-- Select Dependents --
Education
Type here to search
37°C Sunny
ENG
15:03
13-04-2023

ADVANTAGES & DISADVANTAGES

Advantage

Keep Control of the Company

A bank loans money to a business based on the value of the business and its perceived ability to service the loan by making payments on time and in full. Unlike with equity finance where the business issues shares, banks do not take any ownership position in businesses. Bank personnel also do not get involved in any aspect of running a business to which a bank grants a loan. This means you ghet to retain full management and control of your business with no external interference.

Bank Loan is Temporary

Once a business borrower has paid off a loan, there is no more obligation to or involvement with the bank lender unless the borrower wishes to take out a subsequent loan. Compare this with equity finance, where the company may be paying out dividends to shareholders for as along as the business exists.

Interest is Tax Deductible

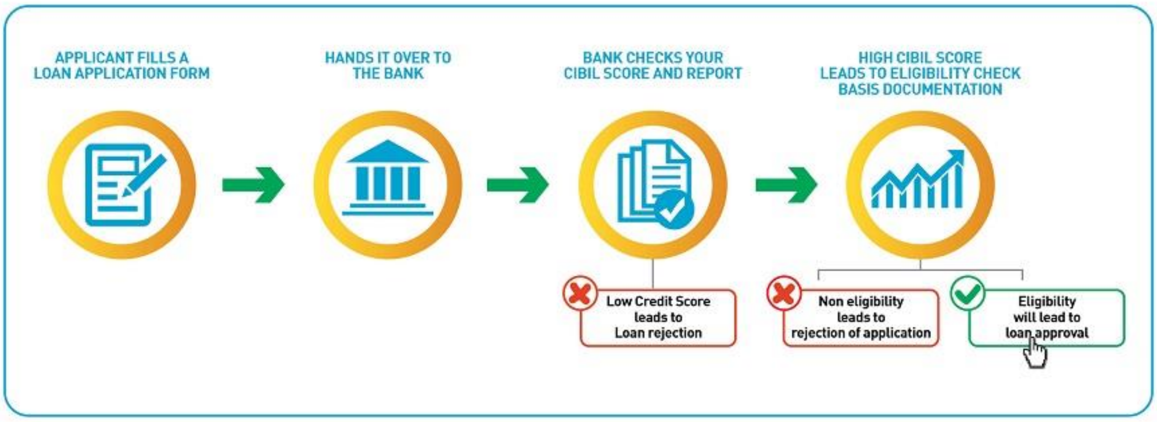
The interest on business bank loans is tax-deductible. In addition, especially with fixed-rate loans, in which the interest rate does not change during the course of a loan, loan servicing payments remain the same throughout the life of the loan. This makes it easy for businesses to budget and plan for monthly loan payments. Even if the loan is an adjustable-rate loan, business owners can use a simple spreadsheet to compute future payments in the event of a change in rates.

Disadvantage

Tough to Qualify

One of the greatest disadvantages to bank loans is that they are very difficult to obtain unless a small business has a substantial track record or valuable collateral such as real estate. Banks are careful to lend only to businesses that can clearly repay their loans, and they also make sure that they are able to cover losses in the event of default.

APPLICATIONS



WHAT DO BANKS BROADLY CHECK?

1. **CIBIL Score and Report:** It is one of the most important factor that affects your loan approval. A good credit score and report is a positive indicator of your credit health.

2. **Employment Status:** Apart from a good credit history, lenders also check for your steady income and employment status.

3. **Account Details:** Credit Facility statuses and suit filed cases are carefully examined by lenders.

4. **Payment History:** Lenders check for any default on payments or amount overdue cases, which might project a negative overview of your overall report.

5. **EMI to Income Ratio:** Banks also consider the proportion of your existing loans when compared to your salary at the time of loan application. Your chances of loan approval gets reduced if your total EMI's exceed your monthly salary by 50%.

CONCLUSION

This system would be able to determine the status of the loan whether it would get approved or denied swiftly in real-time. Displays accuracy with various algorithms. We have compared the Logistic regression algorithm to two other algorithms, random forest, and decision tree Table III. However, of all the algorithms, Logistic regression has the highest accuracy. Also, it can fill the missing values of the datasets, treat categorical values, scalability problems, overfitting problems, and provide a good visualization of the data using a confusion matrix. Applicants who have a poor credit history are likely to be rejected, especially to the risk of not repaying the loan. Applicants with high income who request low-interest loans have a stronger chance of being accepted, which is logical because they have a strong chance to repay their debts. Few essential characteristics, such as marital status and gender, appear to be overlooked by the organization, but the number of dependents is taken into consideration. The libraries are used professionally and are sufficient for now because we chose the Python programming language, but many aspects require additional exploration. Many areas of our project are left unexplored and might be studied and explored further. For further research, applicants' Age, past health records, as well as the type of occupation they have will be utilized to evaluate the ambiguity factor of paying debts, and possible defaults of corporate loans for businesses and startups can be forecasted. Another method could be developed to forecast defaulters on different types of loans as well. We used a medium-sized data set to train our model, which may have influenced the outcome; therefore, a big and well-defined data set is required for more accurate results. This paperwork could be expanded to a higher level in the future, allowing the software to be improved to make it more dependable, secure, and accurate. The system has been trained using current data sets that may become older in the future, allowing it to participate in fresh testing to pass new test cases.

FUTURE SCOPE

Factors like digitalization, demonetization, and then Covid-19 have pushed the use of apps for making monetary transactions. New-age digital consumers and initiatives by the government have paved the way for a robust digital economy. These radical changes have led to a massive transformation in the way we access different banking services today.

The process of lending has evolved immensely to facilitate easier discovery as well as hassle-free processing of loans.

Also, the Indian government launched India Stack, which is a set of open APIs like eKYC. It allows finance companies to perform the KYC verification digitally using mobile OTP or biometrics. Sign also enables applicants to sign and submit documents digitally. Such technological advancements have given a considerable edge to the alternative lending segment in India.

Because of digitization and such initiatives by the government, digital marketplaces connecting lenders to credit-seekers are continuously enhancing the efficiency of the lending process. As a result, online loan providers are making availing of a loan seamless, quick, and paperless for consumers.

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, RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier

from sklearn.model_selection import RandomizedSearchCV

from sklearn.preprocessing import StandardScaler

import imblearn
import pandas as pd

from datetime import datetime
```

```
[140] loan_dataset=pd.read_csv('/content/train_u6lujuX_CVtuZ9i.csv')
```

```
loan_dataset.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
```

```
[142] loan_dataset.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

```
loan_dataset.isnull().sum()
```

```
Loan_ID               0
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```

```
[144] loan_dataset = loan_dataset.dropna()
```

```python
loan_dataset['Gender'] = loan_dataset['Gender'].fillna(loan_dataset['Gender'].mode()[0])
loan_dataset['Married'] = loan_dataset['Married'].fillna(loan_dataset['Married'].mode()[0])
#replacing + with space for filling the non values
loan_dataset['Dependents'] = loan_dataset['Dependents'].fillna(loan_dataset['Dependents'].mode()[0])
loan_dataset['Self_Employed'] = loan_dataset['Self_Employed'].fillna(loan_dataset['Self_Employed'].mode()[0])
loan_dataset['LoanAmount'] = loan_dataset['LoanAmount'].fillna(loan_dataset['LoanAmount'].mode()[0])
loan_dataset['Loan_Amount_Term']= loan_dataset['Loan_Amount_Term'].fillna(loan_dataset['Loan_Amount_Term'].mode()[0])
loan_dataset['Credit_History'] = loan_dataset['Credit_History'].fillna(loan_dataset['Credit_History'].mode()[0])
```

```python
loan_dataset['Gender']=loan_dataset['Gender']
loan_dataset['Married']=loan_dataset['Married']
loan_dataset['Dependents']=loan_dataset['Dependents']
loan_dataset['Self_Employed' ]=loan_dataset['Self_Employed']
loan_dataset['CoapplicantIncome']=loan_dataset['CoapplicantIncome']
loan_dataset['LoanAmount']=loan_dataset['LoanAmount']
loan_dataset['Loan_Amount_Term']=loan_dataset['Loan_Amount_Term']
loan_dataset['Credit_History']=loan_dataset['Credit_History']
```

```
scaler=StandardScaler()
scaler.fit(X)
standardized_data=scaler.transform(X)
print(standardized_data)
x=standardized_data
y=loan_dataset['Loan_Status']
print(x)
print(y)
```

```
[[ 0.46719815  0.73716237  0.11235219 ...  0.27554157  0.41319694
  -1.31886834]
 [ 0.46719815  0.73716237 -0.70475462 ...  0.27554157  0.41319694
   1.25977445]
 [ 0.46719815  0.73716237 -0.70475462 ...  0.27554157  0.41319694
   1.25977445]
 ...
 [ 0.46719815  0.73716237  0.11235219 ...  0.27554157  0.41319694
   1.25977445]
 [ 0.46719815  0.73716237  0.92945899 ...  0.27554157  0.41319694
   1.25977445]
 [-2.14041943 -1.35655324 -0.70475462 ...  0.27554157 -2.42015348
  -0.02954695]]
[[ 0.46719815  0.73716237  0.11235219 ...  0.27554157  0.41319694
  -1.31886834]
```

✓ 0s completed at 11:23 AM

```
[132] X_train, X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,stratify=Y,random_state=0)

      print(X.shape, X_train.shape, X_test.shape)
```

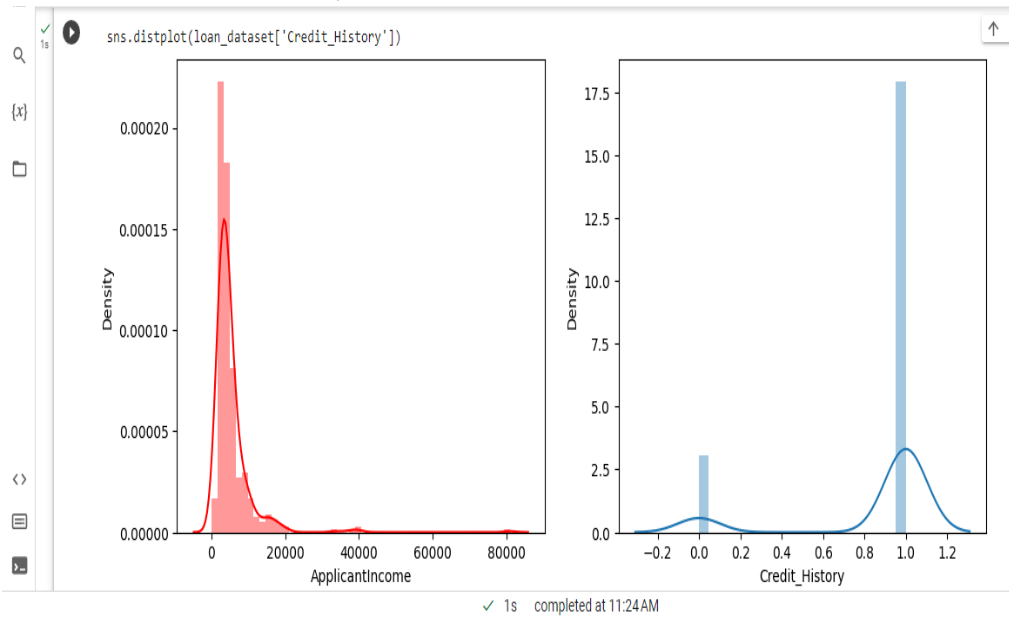
```
(480, 11) (384, 11) (96, 11)
```

A. \

```
#plotting the using distplot

plt.figure(figsize=(12,5))

plt.subplot(121)

sns.distplot(loan_dataset['ApplicantIncome'], color='r')

plt.subplot(122)

sns.distplot(loan_dataset['Credit_History'])

plt.show()
```

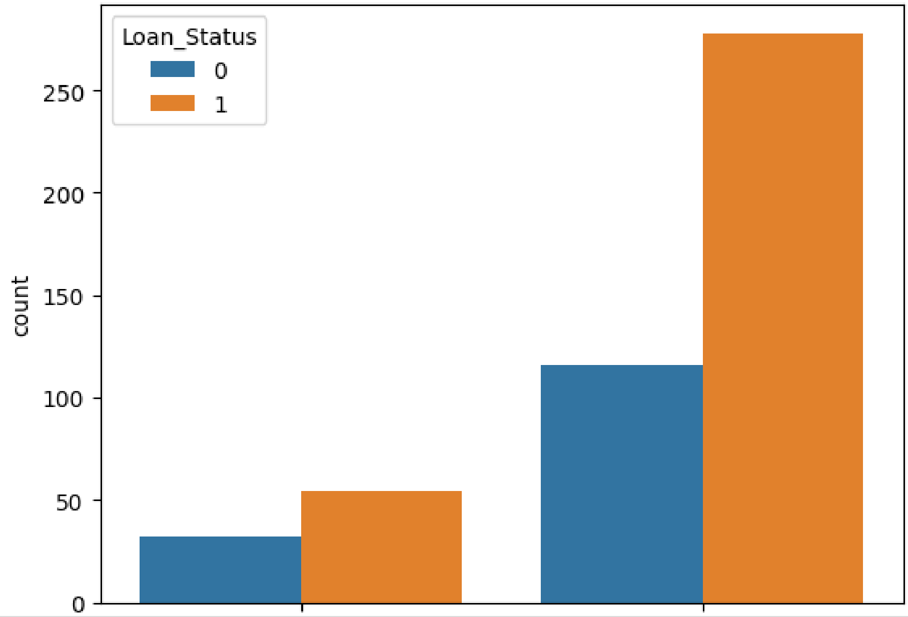
```
<ipython-input-151-9f72075372ce>:7: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(loan_dataset['ApplicantIncome'], color='r')
```

✓ 1s completed at 11:24 AM



```
sns.countplot(x='Gender',hue='Loan_Status',data=loan_dataset)
```

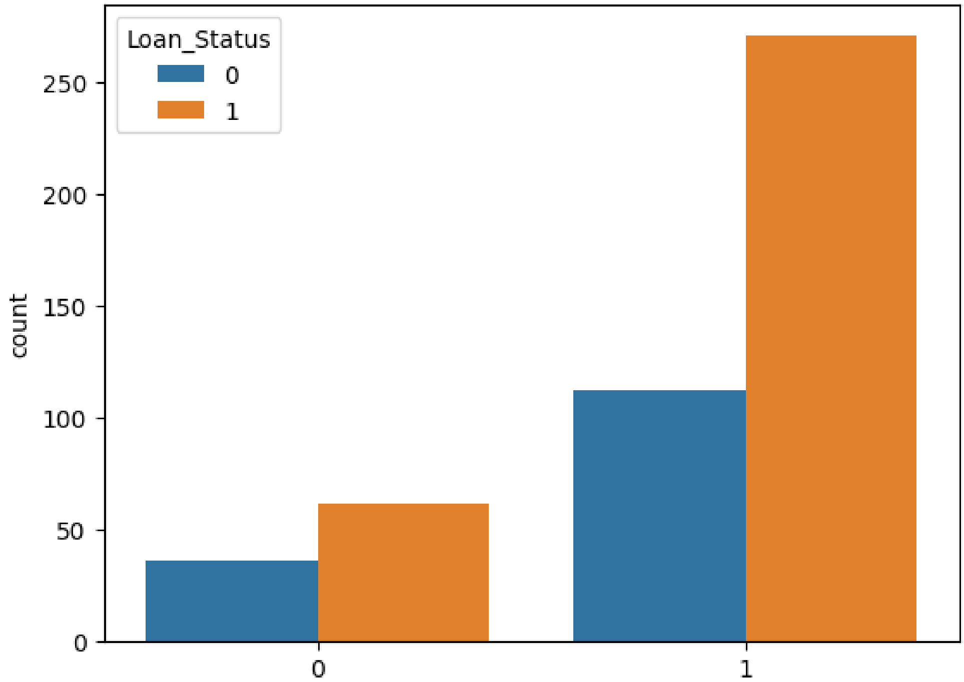
```
<Axes: xlabel='Gender', ylabel='count'>
```



✓ 1s completed at 11:24 AM

```
sns.countplot(x='Education',hue='Loan_Status',data=loan_dataset)
```

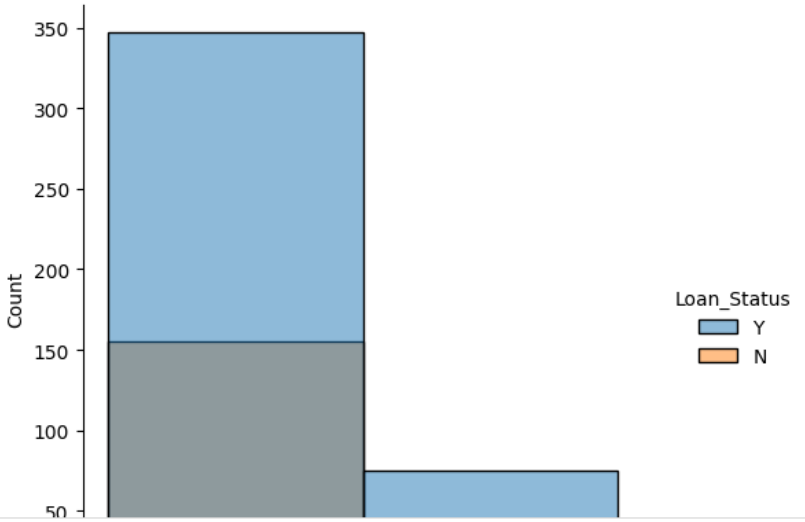
```
<Axes: xlabel='Education', ylabel='count'>
```



✓ 1s completed at 11:24 AM

```
[137] sns.displot(x='Gender',hue='Loan_Status',data=data)
```

<seaborn.axisgrid.FacetGrid at 0x7ffb7c862370>



✓ 1s completed at 11:24 AM

```
[138] models.append(('DT', DecisionTreeClassifier()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('RC', RandomForestClassifier()))
```

zzz

```
from distutils.log import debug
from flask import Flask, escape, request, render_template
import pickle
import numpy as np

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

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


@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method ==  'POST':
        gender = request.form['gender']
        married = request.form['married']
        dependents = request.form['dependents']
        education = request.form['education']
        employed = request.form['employed']
        credit = float(request.form['credit'])
        area = request.form['area']
        ApplicantIncome = float(request.form['ApplicantIncome'])
        CoapplicantIncome = float(request.form['CoapplicantIncome'])
        LoanAmount = float(request.form['LoanAmount'])
        Loan_Amount_Term = float(request.form['Loan_Amount_Term'])

        # gender
```

```
    # gender
    if (gender == "Male"):
        male=1
    else:
        male=0

    # married
    if(married=="Yes"):
        married_yes = 1
    else:
        married_yes=0

    # dependents
    if(dependents=='1'):
        dependents_1 = 1
        dependents_2 = 0
        dependents_3 = 0
    elif(dependents == '2'):
        dependents_1 = 0
        dependents_2 = 1
        dependents_3 = 0
    elif(dependents=="3+"):
        dependents_1 = 0
        dependents_2 = 0
        dependents_3 = 1
    else:
        dependents_1 = 0
        dependents_2 = 0
        dependents_3 = 0

    # education
```

```
    # education
    if (education=="Not Graduate"):
        not_graduate=1
    else:
        not_graduate=0

    # employed
    if (employed == "Yes"):
        employed_yes=1
    else:
        employed_yes=0

    # property area

    if(area=="Semiurban"):
        semiurban=1
        urban=0
    elif(area=="Urban"):
        semiurban=0
        urban=1
    else:
        semiurban=0
        urban=0


    ApplicantIncomelog = np.log(ApplicantIncome)
    totalincomelog = np.log(ApplicantIncome+CoapplicantIncome)
    LoanAmountlog = np.log(LoanAmount)
    Loan_Amount_Termlog = np.log(Loan_Amount_Term)

    prediction = model.predict([[credit, ApplicantIncomelog,LoanAmountlog, Loan_Amount_Termlog, totalincomelog, male, married_yes, depend
```

```
        semiurban=0
        urban=0


    ApplicantIncomelog = np.log(ApplicantIncome)
    totalincomelog = np.log(ApplicantIncome+CoapplicantIncome)
    LoanAmountlog = np.log(LoanAmount)
    Loan_Amount_Termlog = np.log(Loan_Amount_Term)

    prediction = model.predict([[credit, ApplicantIncomelog,LoanAmountlog, Loan_Amount_Termlog, totalincomelog, male, married_yes, depende

    # print(prediction)

    if(prediction=="N"):
        prediction="Loan will be not approve"
    else:
        prediction="Loan will be approved"


    return render_template("index.html", prediction_text="loan status is {}".format(prediction))



  else:
    return render_template("index.html")


if __name__ == "__main__":
  app.run(debug=True)
```

```
 * Serving Flask app '__main__'
 * Debug mode: on
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:318: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.22
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
/usr/local/lib/python3.9/dist-packages/sklearn/base.py:318: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.22
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug: * Restarting with stat
```