ARTIFICIAL INTELLIGENCE INTERNSHIP

PROJECT REPORT

PROJECT TITLE:

LOAN STATUS PREDICTION USING EXPLORATORY DATA ANALYSIS

TEAM MEMBERS:

M NAVEEN

M ABHIJITH REDDY

K KARTHIK REDDY

SK IKRAMULLA SHAREEF

M JASWANTH REDDY

TABLE OF CONTENTS

1 INTRODUCTION

1.1 OVERVIEW

- Loan Prediction is extremely helpful for employee of banks as well as for the applicant also.
- The aim of this Project is to provide quick, immediate, and effortless way for bank employees whether the customer can pay back the amount for given interest rate.
- Company or bank wants to automate the loan eligibility process (real time) based on customer details provided while filling application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and other.

1.2 PURPOSE

- This study assesses the classification accuracy of an artificial neural network (ANN) model. It examines the application of loan applied customers whether they can repay the amount for the given interest rate.
- Loan approval is an important process for banking organizations. The system approved or reject the loan applications. So, for a bank employee with this application can easily come to a decision whether the applicant is eligible or not.
- Recovery of loans is a major contributing parameter in the financial statements of a bank.
- It is difficult to predict the possibility of payment of loan by the customer. Using Artificial Neural Network (ANN) we can predict the loan status of applicant.

2 LITERATURE SURVEY:

2.1 PROBLEM STATEMENT:

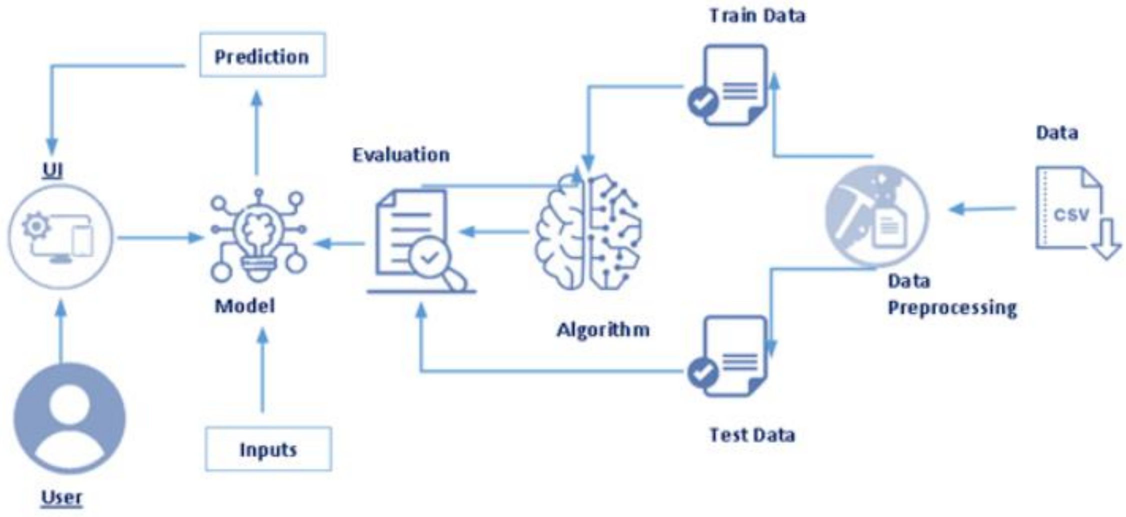
- In India, the number of people applying for loans gets increased for distinct reasons in recent years.
- The bank employees are not able to analyze or predict whether the customer can pay back the amount or not (good customer or bad customer) for the given interest rate.
- The aim is to find the nature of the client applying for a personal loan.

2.2 SOLUTION:

- Using Artificial Neural Network Model (ANN) our project is developed.
- The predictions predicted by the model gives an exceptionally good accuracy so, the bank employers get beneficent.
- The result of the analysis shows that short term loans are preferred by most of the clients and the clients majorly apply loans for debt consolidation.
- The results are shown in graphs that help the bankers to understand the client’s behavior.
- These are the steps for this project:
    - Installing the required packages and libraries.
    - Importing the required libraries for the model to run.
    - Downloading the dataset, feeding it to the model, and understanding the dataset
    - Data Preprocessing – Checking for outliers and null values. If there any null values we use Label Encoding to convert them into binary format.
    - Dividing the model into Train and Test data. Fitting the model and predicting.
    - Building Flask Web Application.

3) THEORITICAL ANALYSIS:

3.1 BLOCK DIAGRAM:



3.2 SPECIFICATIONS:

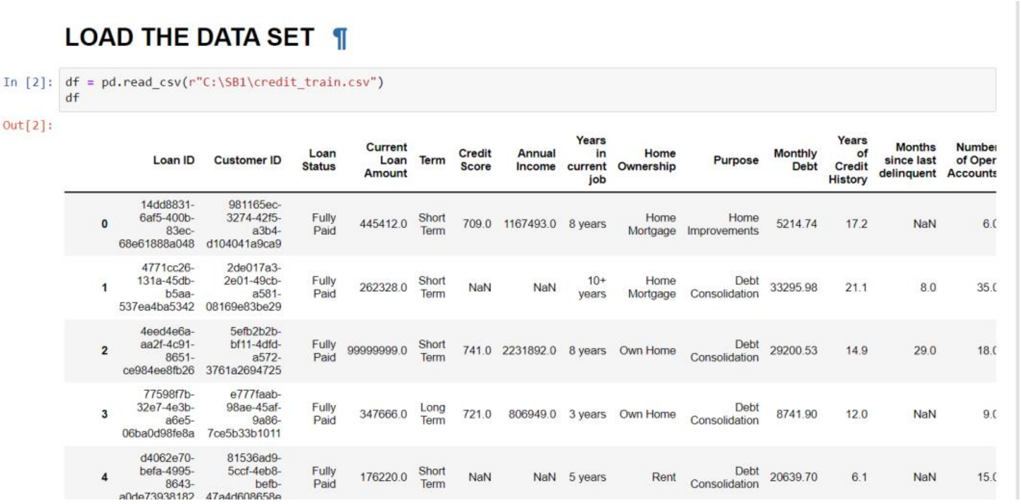
A) HARDWARE: Laptop / Computer

B) SOFTWARE: Python, HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), TensorFlow, Keras, Spyder, Jupyter Notebook etc.

4) EXPERIMENTAL INVESTIGATIONS:

The following shows the pseudo code for the proposed loan prediction method.

1. Load the data



LOAD THE DATA SET ¶

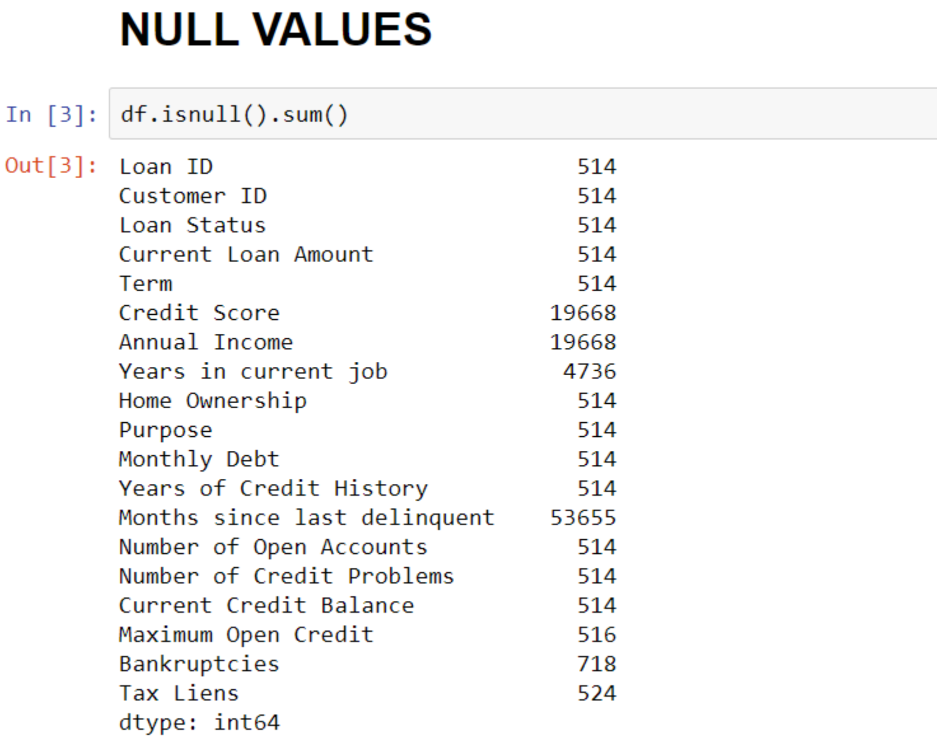
```
In [2]: df = pd.read_csv(r"C:\SB1\credit_train.csv")
        df
```

Out[2]:

	Loan ID	Customer ID	Loan Status	Current Loan Amount	Term	Credit Score	Annual Income	Years in current job	Home Ownership	Purpose	Monthly Debt	Years of Credit History	Months since last delinquent	Numbe of Ope Accounts
0	14dd8831-6af5-400b-83ec-68e61888a048	981165ec-3274-42f5-a3b4-d104041a9ca9	Fully Paid	445412.0	Short Term	709.0	1167493.0	8 years	Home Mortgage	Home Improvements	5214.74	17.2	NaN	6.
1	4771cc26-131a-45db-b5aa-537ea4ba5342	2de017a3-2e01-49cb-a581-08169e83be29	Fully Paid	262328.0	Short Term	NaN	NaN	10+ years	Home Mortgage	Debt Consolidation	33295.98	21.1	8.0	35.
2	4eed4e6a-aa2f-4c91-8651-ce984ee8fb26	5efb2b2b-bf11-4dfd-a572-3761a2694725	Fully Paid	99999999.0	Short Term	741.0	2231892.0	8 years	Own Home	Debt Consolidation	29200.53	14.9	29.0	18.
3	77598f7b-32e7-4e3b-a6e5-06ba0d98fe8a	e777faab-98ae-45af-9a86-7ce5b33b1011	Fully Paid	347666.0	Long Term	721.0	806949.0	3 years	Own Home	Debt Consolidation	8741.90	12.0	NaN	9.
4	d4062e70-befa-4995-8643-a0de73938182	81536ad9-5ccf-4eb8-befb-47a4d608658e	Fully Paid	176220.0	Short Term	NaN	NaN	5 years	Rent	Debt Consolidation	20639.70	6.1	NaN	15.

2. Data cleaning and pre-processing.

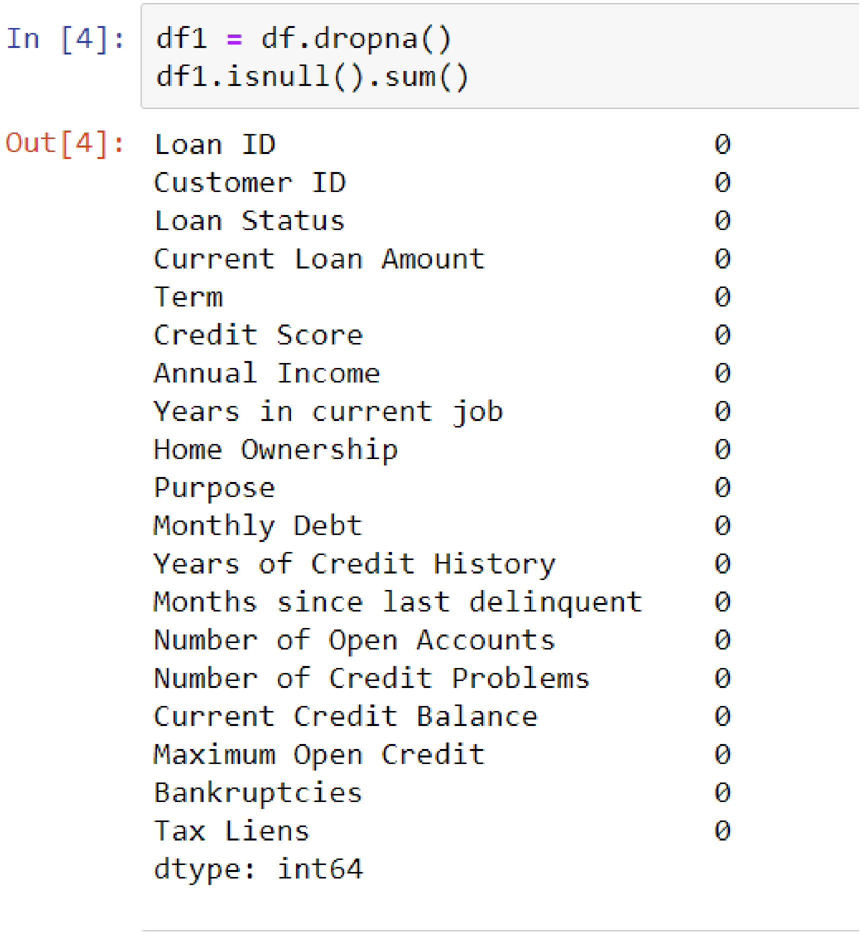
a) Check for null values in the dataset



NULL VALUES

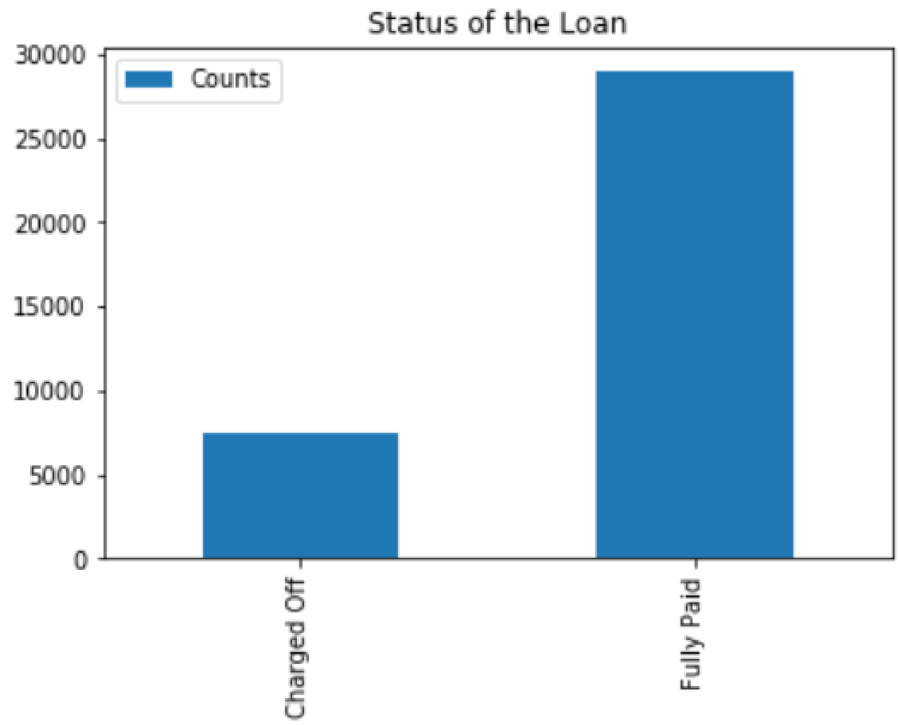
```
In [3]: df.isnull().sum()

Out[3]: Loan ID                           514
        Customer ID                       514
        Loan Status                       514
        Current Loan Amount               514
        Term                              514
        Credit Score                    19668
        Annual Income                   19668
        Years in current job             4736
        Home Ownership                    514
        Purpose                           514
        Monthly Debt                      514
        Years of Credit History           514
        Months since last delinquent    53655
        Number of Open Accounts           514
        Number of Credit Problems         514
        Current Credit Balance            514
        Maximum Open Credit               516
        Bankruptcies                      718
        Tax Liens                         524
        dtype: int64
```

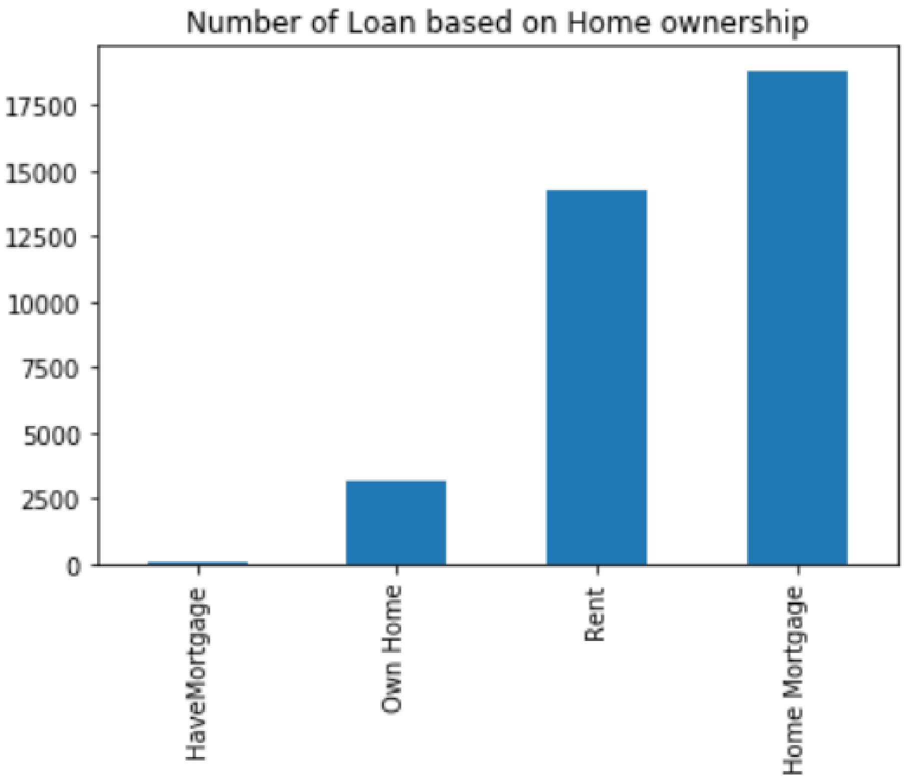
b) Fill the missing values with mean values or drop them from the data set.

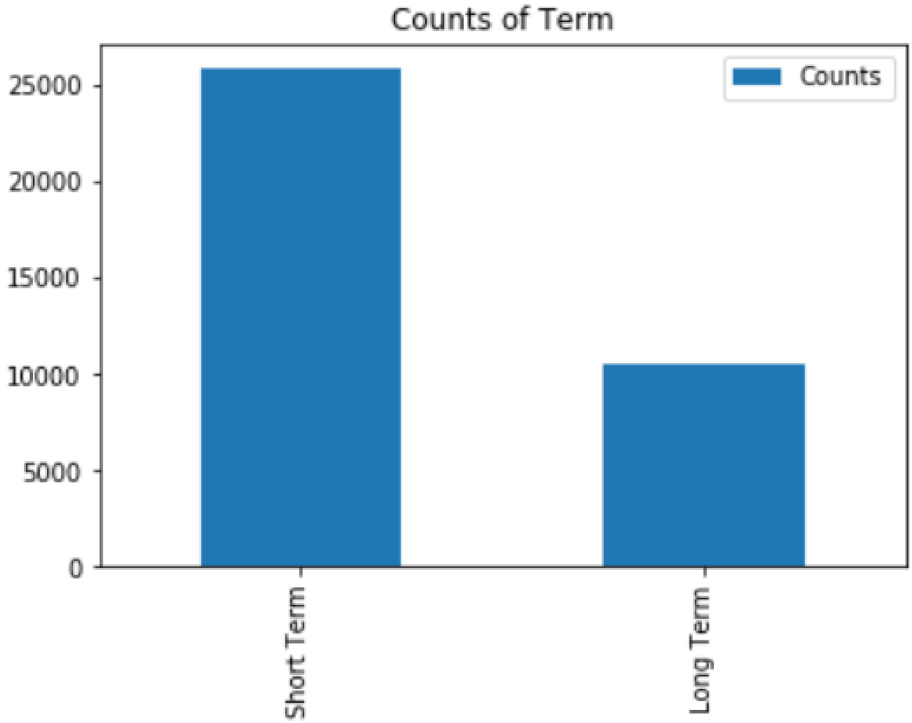


```
In [4]: df1 = df.dropna()
        df1.isnull().sum()

Out[4]: Loan ID                         0
        Customer ID                     0
        Loan Status                     0
        Current Loan Amount             0
        Term                            0
        Credit Score                    0
        Annual Income                   0
        Years in current job            0
        Home Ownership                  0
        Purpose                         0
        Monthly Debt                    0
        Years of Credit History         0
        Months since last delinquent    0
        Number of Open Accounts         0
        Number of Credit Problems       0
        Current Credit Balance          0
        Maximum Open Credit             0
        Bankruptcies                    0
        Tax Liens                       0
        dtype: int64
```

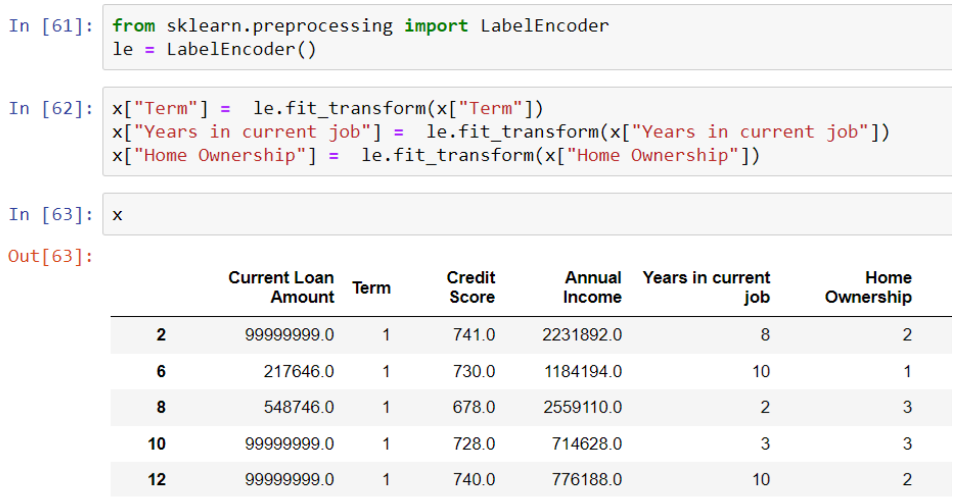
c) The Target column is loan status







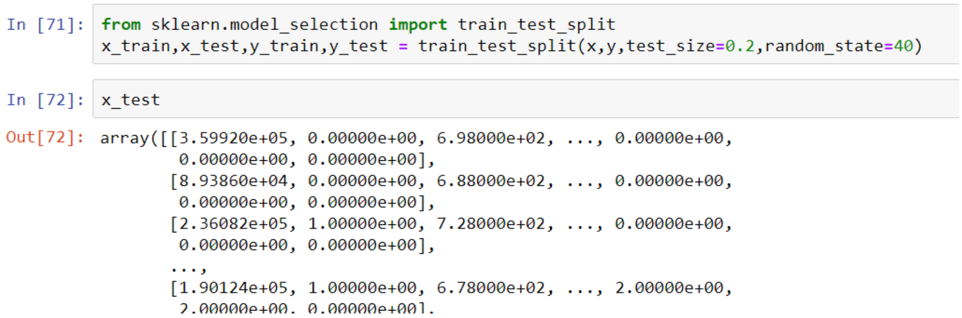
d) Convert the categorical data into numerical values using Label Encoder.



```
In [61]: from sklearn.preprocessing import LabelEncoder
         le = LabelEncoder()

In [62]: x["Term"] =  le.fit_transform(x["Term"])
         x["Years in current job"] =  le.fit_transform(x["Years in current job"])
         x["Home Ownership"] =  le.fit_transform(x["Home Ownership"])

In [63]: x
Out[63]:
```

	Current Loan Amount	Term	Credit Score	Annual Income	Years in current job	Home Ownership
2	99999999.0	1	741.0	2231892.0	8	2
6	217646.0	1	730.0	1184194.0	10	1
8	548746.0	1	678.0	2559110.0	2	3
10	99999999.0	1	728.0	714628.0	3	3
12	99999999.0	1	740.0	776188.0	10	2

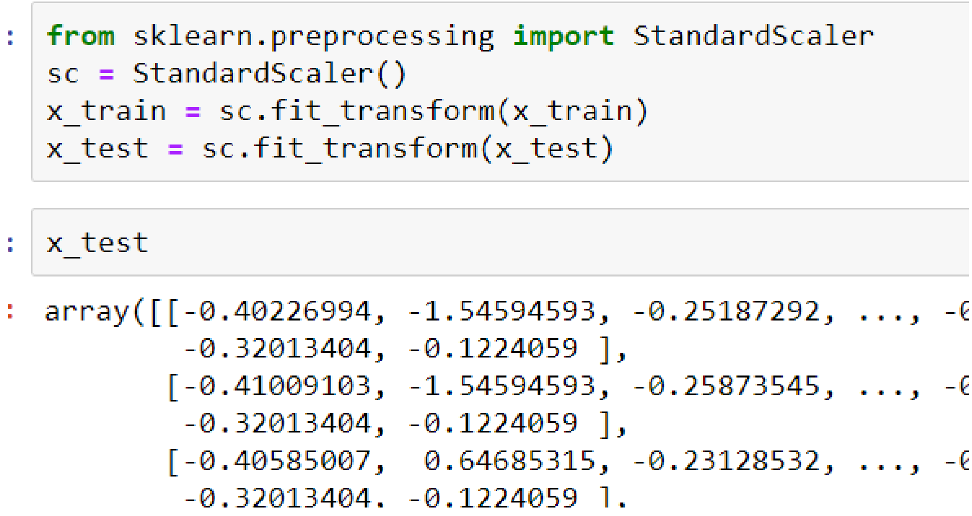
3. Determine the training and testing data



```
In [71]: 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=40)

In [72]: x_test

Out[72]: array([[3.59920e+05, 0.00000e+00, 6.98000e+02, ..., 0.00000e+00,
                 0.00000e+00, 0.00000e+00],
                [8.93860e+04, 0.00000e+00, 6.88000e+02, ..., 0.00000e+00,
                 0.00000e+00, 0.00000e+00],
                [2.36082e+05, 1.00000e+00, 7.28000e+02, ..., 0.00000e+00,
                 0.00000e+00, 0.00000e+00],
                ...,
                [1.90124e+05, 1.00000e+00, 6.78000e+02, ..., 2.00000e+00,
                 2.00000e+00, 0.00000e+00],
```

4. Apply the modelling for prediction using Artificial Neural Networks (ANN)

a) Apply standard scalar

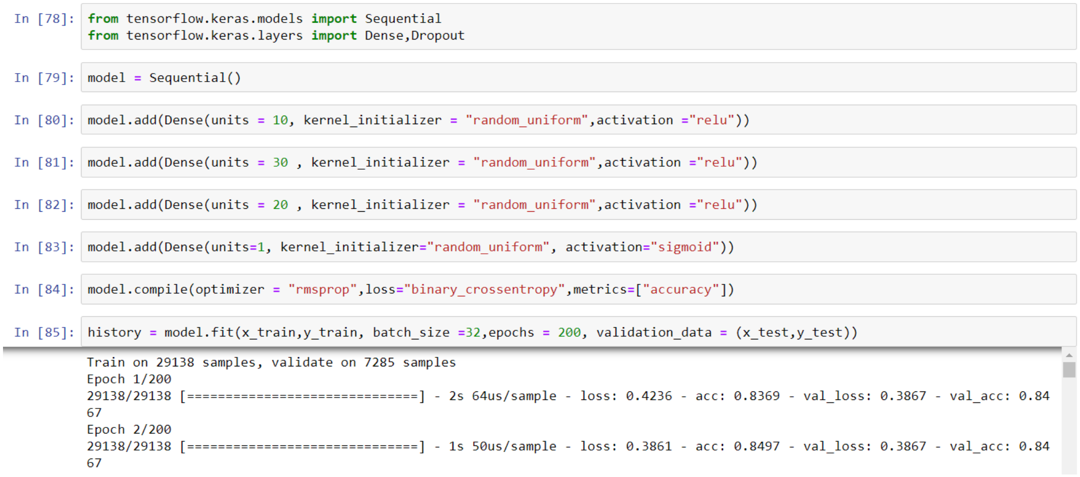


```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
```

```
x_test
```

```
array([[-0.40226994, -1.54594593, -0.25187292, ..., -0
        -0.32013404, -0.1224059 ],
       [-0.41009103, -1.54594593, -0.25873545, ..., -0
        -0.32013404, -0.1224059 ],
       [-0.40585007,  0.64685315, -0.23128532, ..., -0
        -0.32013404, -0.1224059 ],
```

b) Build the ANN model

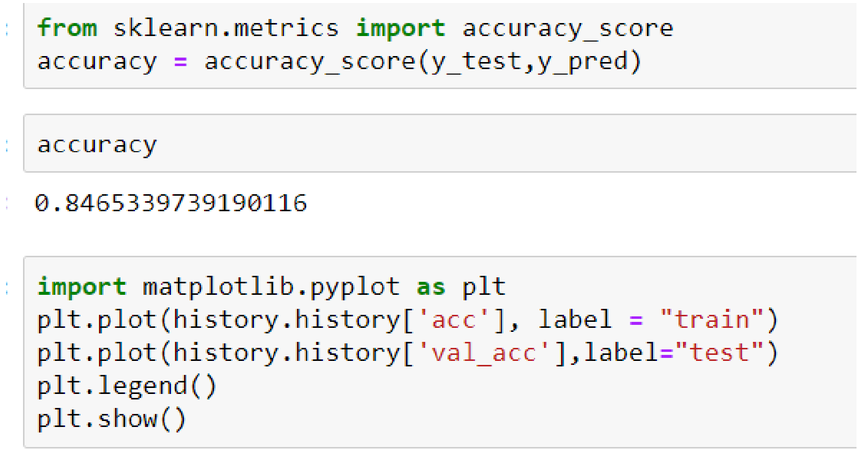


```
In [78]: from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Dense,Dropout

In [79]: model = Sequential()

In [80]: model.add(Dense(units = 10, kernel_initializer = "random_uniform",activation ="relu"))

In [81]: model.add(Dense(units = 30 , kernel_initializer = "random_uniform",activation ="relu"))

In [82]: model.add(Dense(units = 20 , kernel_initializer = "random_uniform",activation ="relu"))

In [83]: model.add(Dense(units=1, kernel_initializer="random_uniform", activation="sigmoid"))

In [84]: model.compile(optimizer = "rmsprop",loss="binary_crossentropy",metrics=["accuracy"])

In [85]: history = model.fit(x_train,y_train, batch_size =32,epochs = 200, validation_data = (x_test,y_test))
```

```
Train on 29138 samples, validate on 7285 samples
Epoch 1/200
29138/29138 [==============================] - 2s 64us/sample - loss: 0.4236 - acc: 0.8369 - val_loss: 0.3867 - val_acc: 0.84
67
Epoch 2/200
29138/29138 [==============================] - 1s 50us/sample - loss: 0.3861 - acc: 0.8497 - val_loss: 0.3867 - val_acc: 0.84
67
```

d) Save the model

5. Determine the accuracy

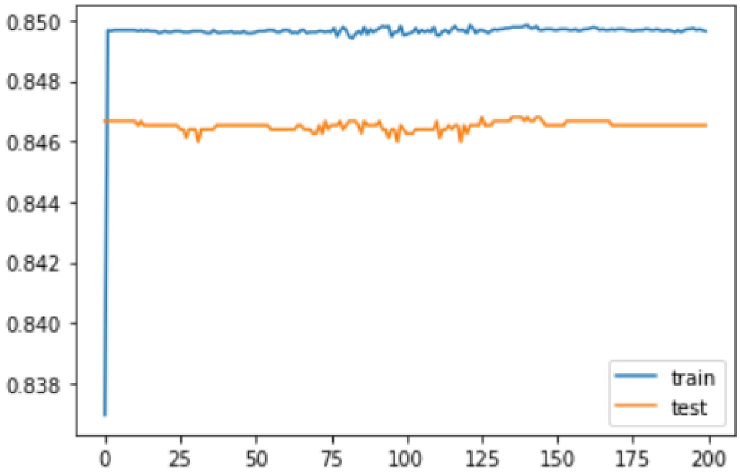


```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test,y_pred)
```

```
accuracy
```

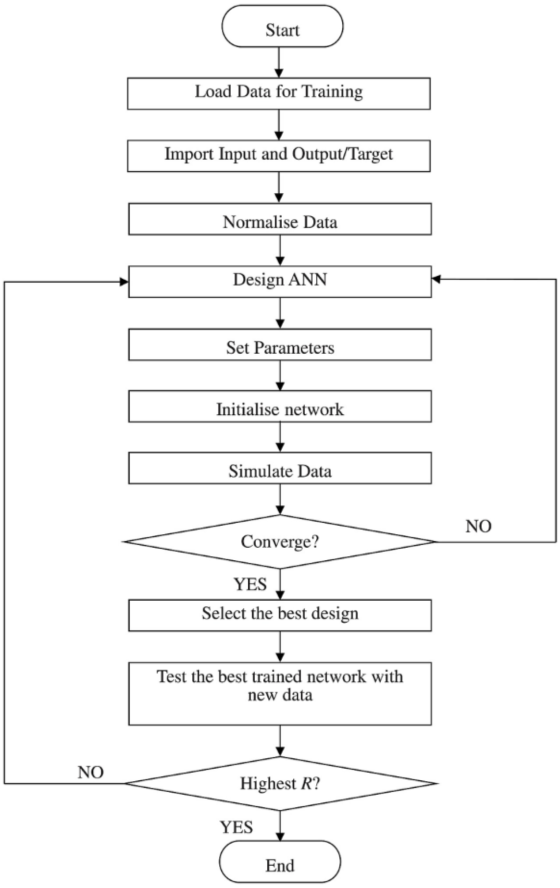
```
0.8465339739190116
```

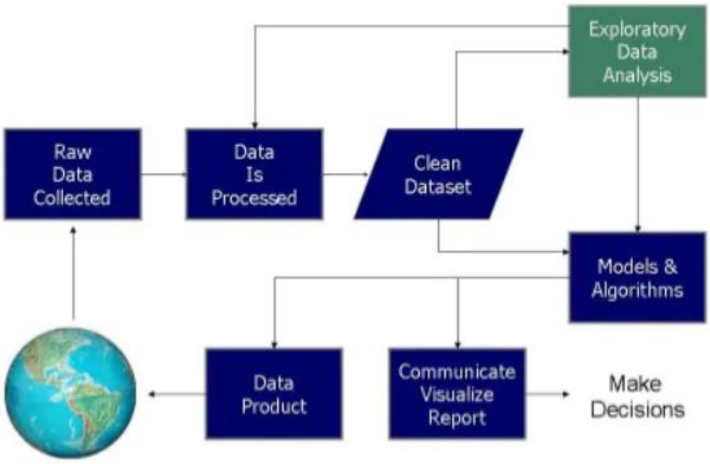
```
import matplotlib.pyplot as plt
plt.plot(history.history['acc'], label = "train")
plt.plot(history.history['val_acc'],label="test")
plt.legend()
plt.show()
```



6) Import the model to Flask and link with the Web Applications.

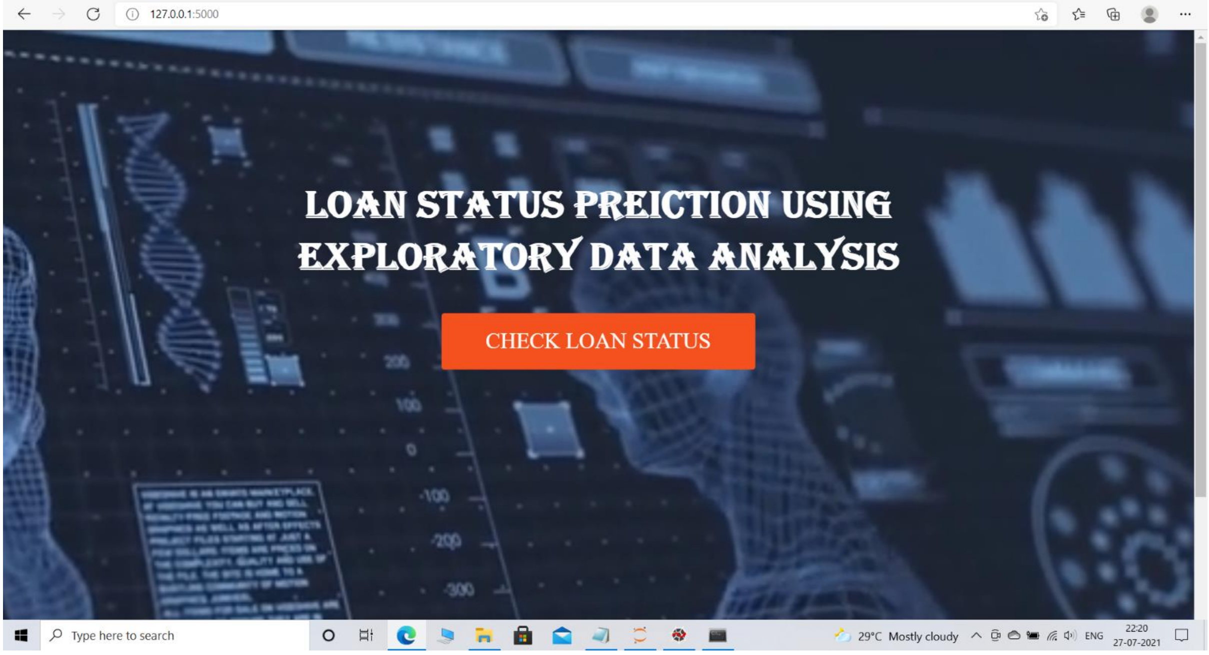
5) FLOW CHART:





6) RESULT:

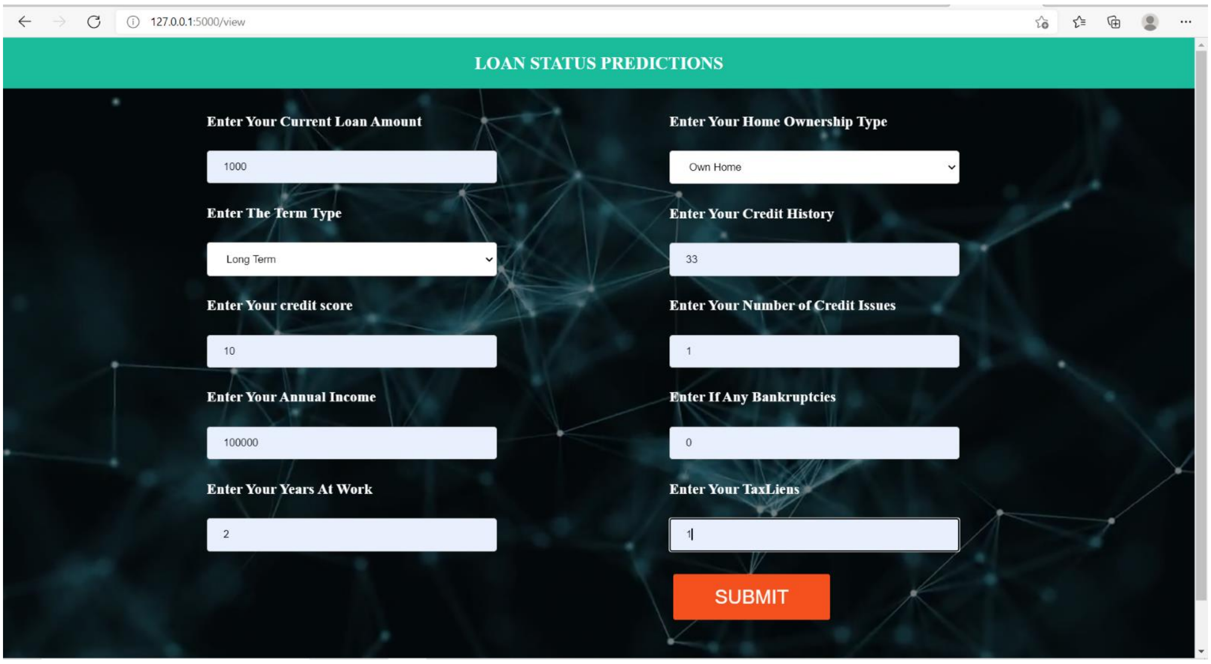
The following images shows the screenshot of our applications of Loan Status Prediction.



LOAN STATUS PREICTION USING EXPLORATORY DATA ANALYSIS

CHECK LOAN STATUS

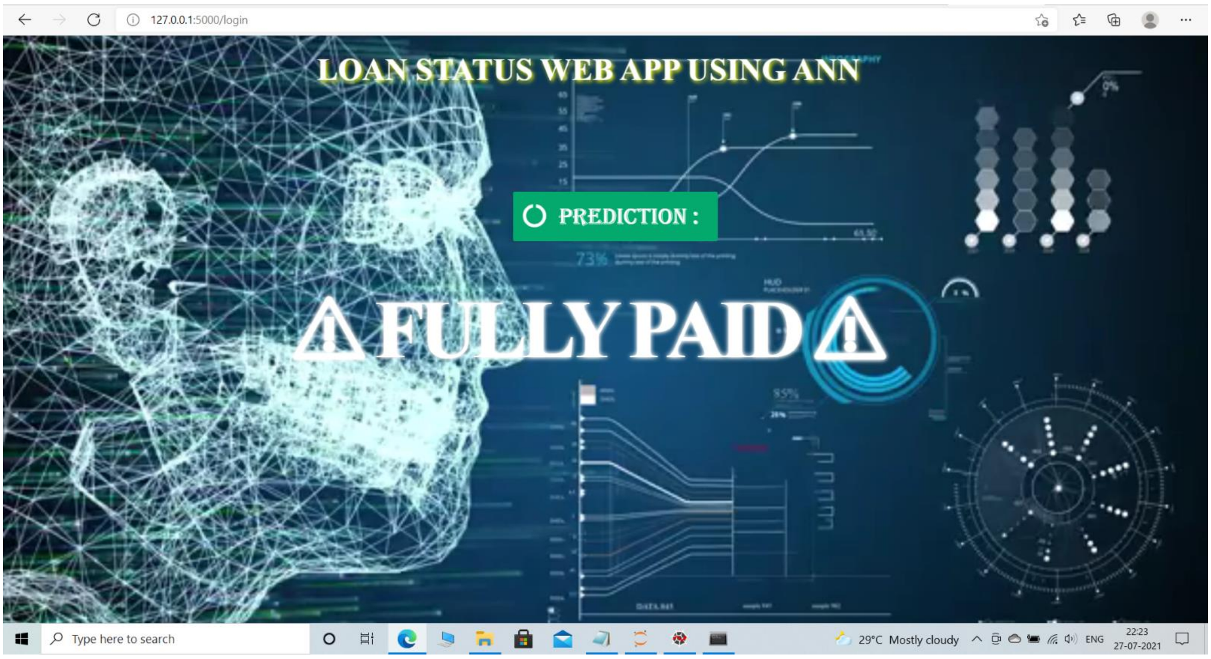
FULLY PAID



LOAN STATUS PREDICTIONS

Enter Your Current Loan Amount: 1000	Enter Your Home Ownership Type: Own Home
Enter The Term Type: Long Term	Enter Your Credit History: 33
Enter Your credit score: 10	Enter Your Number of Credit Issues: 1
Enter Your Annual Income: 100000	Enter If Any Bankruptcies: 0
Enter Your Years At Work: 2	Enter Your TaxLiens: 1

SUBMIT

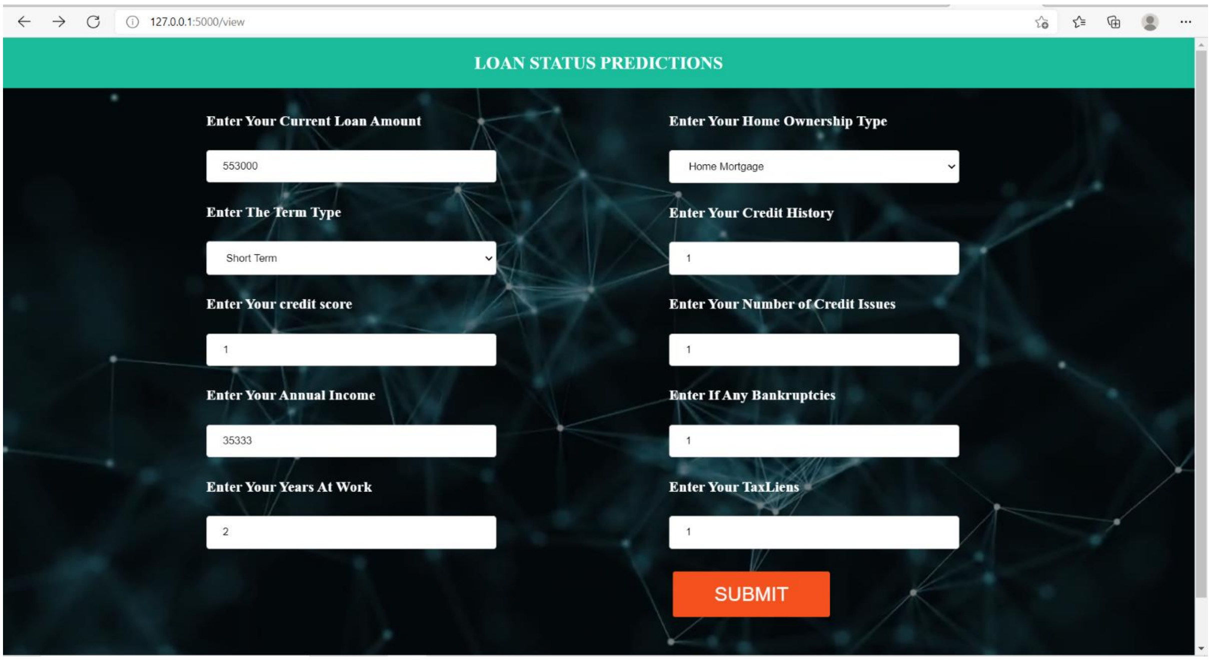


LOAN STATUS WEB APP USING ANN

PREDICTION :

⚠ FULLY PAID ⚠

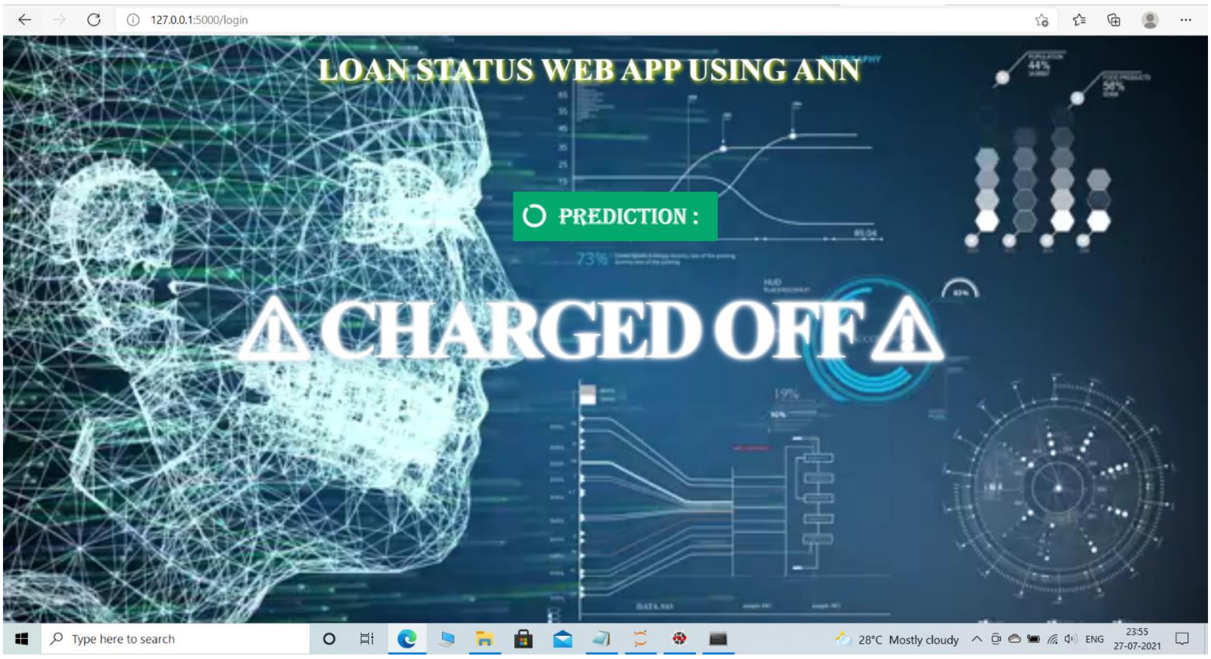
CHARGED OFF



LOAN STATUS PREDICTIONS

Enter Your Current Loan Amount	Enter Your Home Ownership Type
553000	Home Mortgage
Enter The Term Type	Enter Your Credit History
Short Term	1
Enter Your credit score	Enter Your Number of Credit Issues
1	1
Enter Your Annual Income	Enter If Any Bankruptcies
35333	1
Enter Your Years At Work	Enter Your TaxLiens
2	1

SUBMIT



LOAN STATUS WEB APP USING ANN

PREDICTION :

⚠CHARGED OFF⚠

7.1) ADVANTAGES:

☺ To analyze the data the python libraries, help a lot.

☺ The accuracy of the existing model is exceptionally good.

☺ Statistical and prediction is quite easy comparing to existing technologies.

☺ Bank employees can easily come to a solution with existing model.

7.2) DISADVANTAGES:

☹ Complexity in analyzing the data.

☹ Prediction is challenging task working in the model

☹ More work should be done web UI.

☹ Coding is complex maintaining multiple methods.

8) APPLICATIONS:

★ This project web application can be used at various banking systems.

★ It can be used for bank employees to predict the behavior of the customer with data visualization.

- ★ It can be used at net banking system, so it can automatically predict it self whether the customer can repay or not.

9) CONCLUSION:

- ✓ So here, it can be concluded with confidence that the Artificial Neural Network (ANN) model is extremely efficient and gives a better result when compared to other models.
- ✓ It works correctly and fulfills all requirements of bank employees. This system properly works and accurately calculates the result.
- ✓ It predicts the nature of the client applying for a personal loan.

10) FUTURE WORK:

- ❖ This application can be inserted into various applications regarding bank statements.
- ❖ The UI of the web Application can be developed in variety of ways to look it more attractive.
- ❖ There have been numbers cases of computer glitches and most important weight of features is fixed in automated prediction system.
- ❖ In near future this module of prediction can be integrated with the module of automated processing system.

11) BIBILOGRAPHY:

- https://towardsdatascience.com/an-introduction-to-exploratory-data-analysis-in-python- 9a76f04628b8
- https://www.youtube.com/watch?v=T9kgWBmUIRk
- https://www.youtube.com/watch?v=XckM1pFgZmg

12) APPENDIX:

Source Code: Git Hub