Done by:Deepti.G

College:Gitam University

Project Name:Loan Status Prediction Using IBM Watson Machine Learning

Introduction:

Loan Prediction is very helpful for employee of banks as well as for the applicant also.In India, the number of people applying for loans gets increased for various 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.
The result of the analysis shows that short term loans are preferred by the majority 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.

a)Overview:

Loan approval is a very important process for banking organizations. The system approved or reject the loan applications. Recovery of loans is a major contributing parameter in the financial statements of a bank. It is very difficult to predict the possibility of payment of loan by the customer. Using Machine learning we predict the loan approval.

b)Purpose:

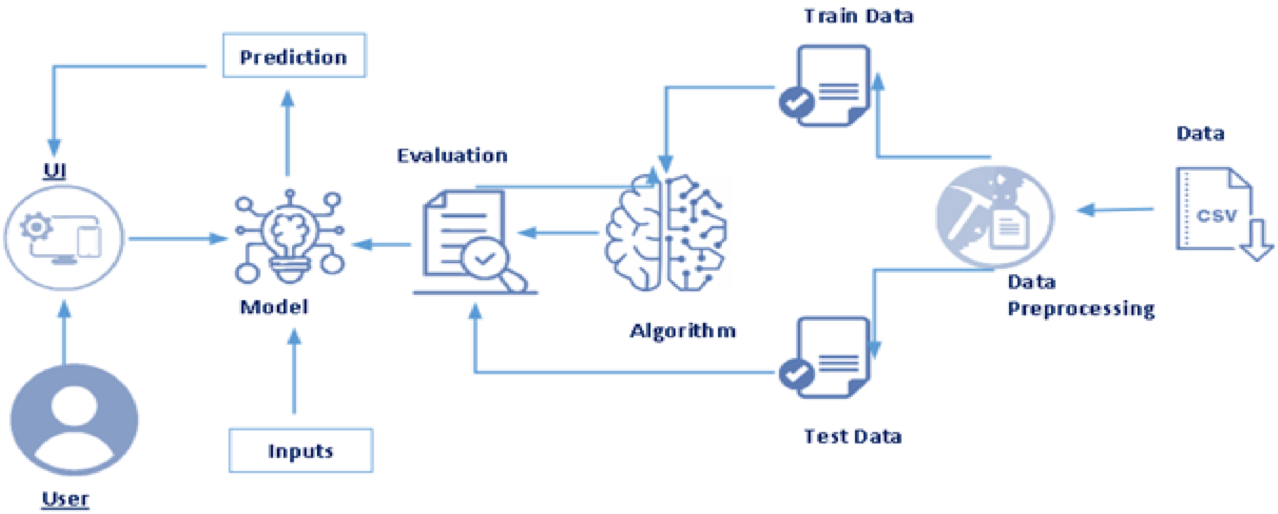
The project's goal is to extract the libraries for machine learning for loan prediction using Python's pandas, matplotlib, and seaborn libraries. Second, for the Logistic Regression machine learning algorithm, learn how to hyper tune the parameters using grid search cross validation. Finally, utilising voting ensemble techniques of pooling predictions from many machine learning algorithms and withdrawing conclusions, predict whether the loan applicant can repeat the loan or not.

Literature Survey:

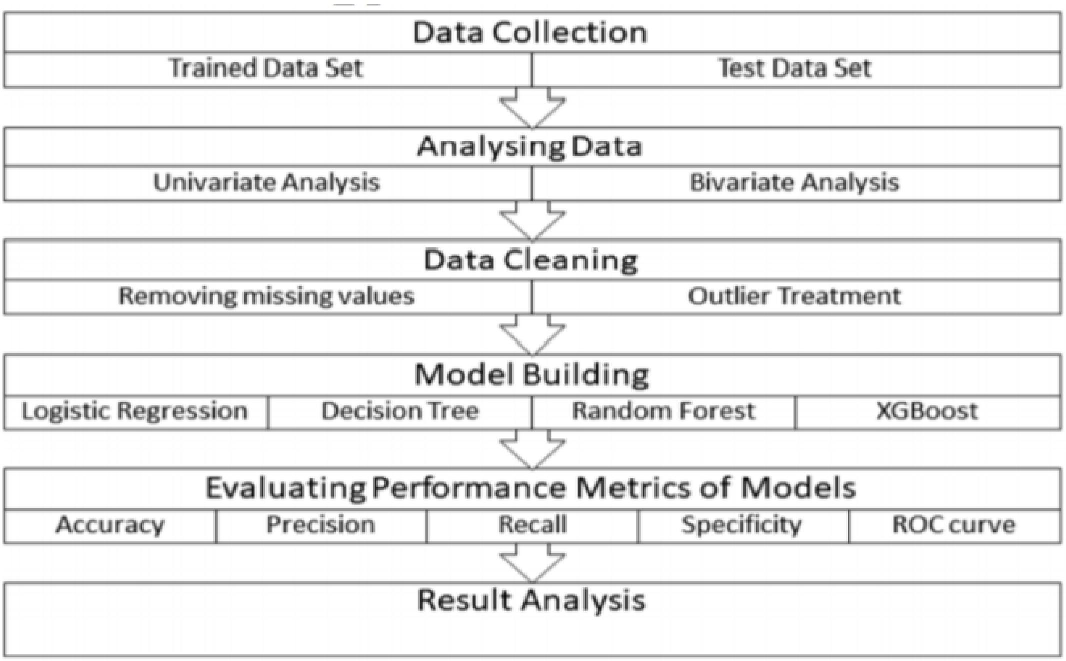
Loan prediction is a much-talked-about subject in the sectors of banking and finance. Credit scoring has become a key tool for the same in this competitive financial world. Furthermore, following the recent improvements in data science and several notable developments in the field of artificial intelligence, this topic has gained more attention and research interest. In recent years, it has attracted more focus towards research on loan prediction and credit risk assessment. Due to the high demands of loan now, demand for further improvements in the models for credit scoring and loan prediction is increasing significantly. A multitude of techniques have been used to assign individuals a credit score and much research has been done over the years on the topic. Unlike previously, where experts were hired and the models depended on professional opinions were used for assessing the individual's creditworthiness, the focus has shifted to an automated way of doing the same job. In recent years, the researchers and banking authorities have been focused on applying machine learning algorithms and neural networks for credit scoring and risk assessment. Many noteworthy conclusions have been drawn in this regard which serve as stepping-stones for researches and studies.

Banks, Housing Finance Companies and some NBFC deal in various types of loans like housing loan, personal loan, business loan etc in all over the part of countries. These companies have existence in Rural, Semi Urban and Urban areas. After applying loan by customer these companies validates the eligibility of customers to get the loan or not. This project provides a solution to automate this process by employing machine learning algorithm

Architecture:



Methodology:



Software Designing:

- Jupyter Notebook Environment
- Machine Learning Algorithms
- Python (pandas, numpy, matplotlib, seaborn, sklearn)
- HTML
- Flask

The models are implemented using Python 3.7 with listed libraries:

Pandas

Pandas is a Python package to work with structured and time series data. The data from various file formats such as csv, json, SQL etc can be imported using Pandas. It is a powerful open-source tool used for data analysis and data manipulation operations such as data cleaning, merging, selecting as well wrangling.

Seaborn

Seaborn is a python library for building graphs to visualize data. It provides integration with pandas. This open-source tool helps in defining the data by mapping the data on the informative and interactive plots. Each element of the plots gives meaningful information about the data.

Sklearn

This python library is helpful for building machine learning and statistical models such as clustering, classification, regression etc. Though it can be used for reading, manipulating and summarizing the data as well, better libraries are there to perform these functions.

NumPy and Pandas:Open-source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib and Seaborn: Used for visualisation with python. The finalised model is now to be saved. We will be saving the model as a pickle or pkl file. HTML pages "pca.html" for our home page and "result1.html" which comes to use when we print out the final predictions made, both of these are stored in the templates folder. Let us build app.py flask file which is a web framework written in python for server-side scripting. Let's see step by step procedure for building the backend application Import required libraries. Configure app.py to fetch the user inputs from the UI, process the values, and return the prediction.

Importing The Libraries:

```
import pandas as pd
import numpy as np
from collections import Counter as c
import matplotlib.pyplot as plt
from sklearn import preprocessing
import seaborn as sns
from sklearn.model_selection import train_test_split
```

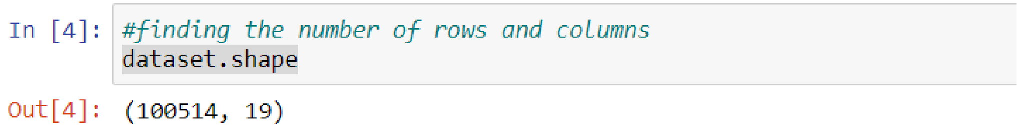
Loading the dataset:

The machine learning model is trained using the training data set. Every new applicant details filled at the time of application form acts as a test data set. On the basis of the training data sets, the model will predict whether a loan would be approved or not.
Train file will be used for training the model, i.e. our model will learn from this file. It contains all the independent variables and the target variable.

For this problem, we have three CSV files: Credit_train.csv file.

dataset = pd.read_csv('credit_train.csv')
dataset.head()

Datainformation:



```
In [4]: #finding the number of rows and columns
        dataset.shape

Out[4]: (100514, 19)
```

Handling Null Values:

```
data.isnull().any()


Loan ID                         True
Customer ID                     True
Loan Status                     True
Current Loan Amount             True
Term                            True
Credit Score                    True
Annual Income                   True
Years in current job            True
Home Ownership                  True
Purpose                         True
Monthly Debt                    True
Years  of  Credit  History     True
Months  since  last  delinquent True
Number  of  Open  Accounts     True
Number   of  Credit  Problems  True
Current        Credit        Balance
```

```
True
Maximum Open Credit            True
Bankruptcies                   True
Tax Liens                      True
dtype: bool
```

After removing the null values:

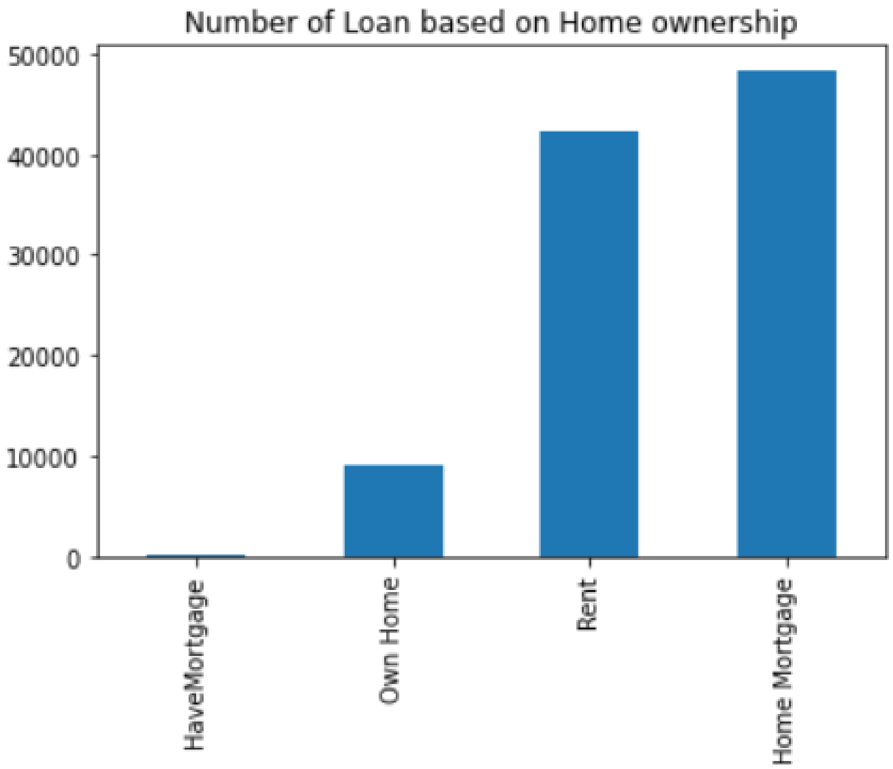
```
data.isnull().any()
```

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

Data Visualization:

Home Ownership Column   we are sorting the elements with values in ascending order.

```
dataset['Home Ownership'].value_counts().sort_values(ascending =
True).plot(kind='bar', title="Number of Loan based on Home ownership")
```



Categorical Independent Variable vs Target Variable:

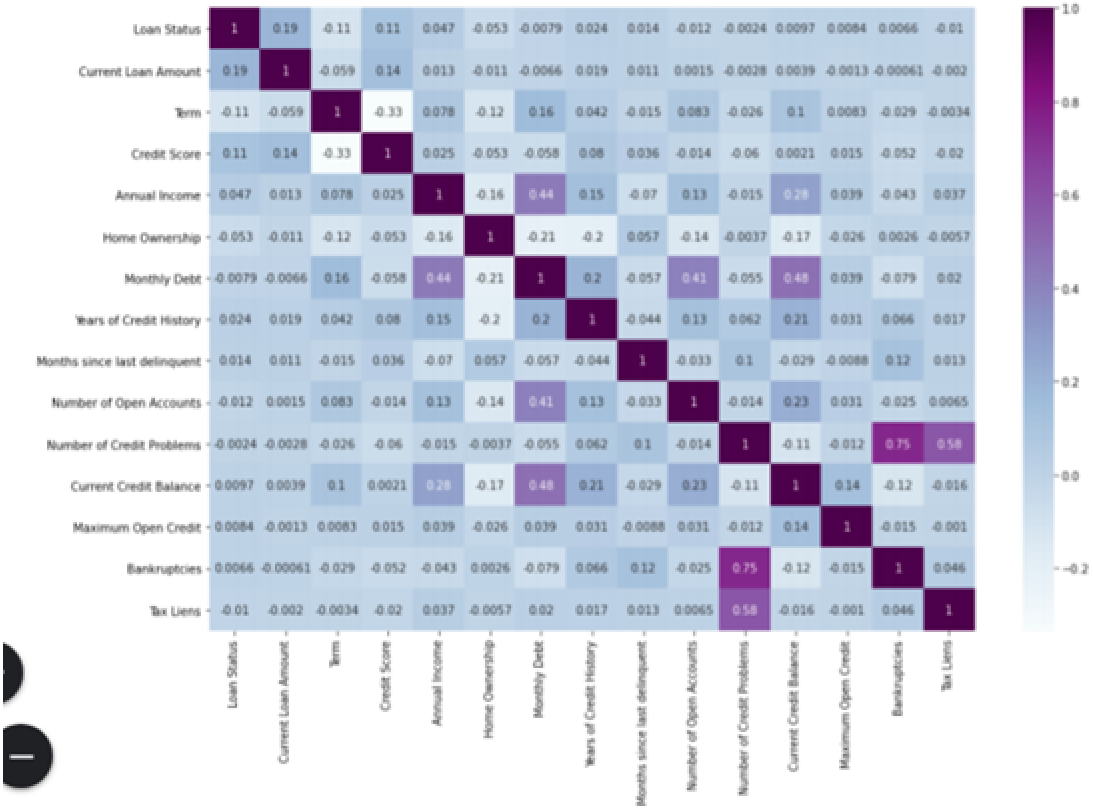
First, we'll figure out how the target variable and categorical independent variables are related. Now let's take a look at the stacked bar plot, which shows the percentage of granted and unapproved loans.

It can be assumed that the proportion of Fully Paid and Charged Off applicants for approved and unapproved loans is around the same.

Let's see how the other categorical variables compare to the target variable. People with a credit score of 1 appear to have a higher chance of getting their loans authorized.

In comparison to rural and urban areas, the proportion of loans authorized in semi-urban areas is higher.
Let's look at numerical independent variables in relation to the target variable now

numerical variable. We will also convert the target variable's categories into 0 and 1 so that we can find its correlation with numerical variables. One more reason to do so is few models like logistic regression takes only numeric values as input. We will replace N with 0 and Ywith 1



.

Building Model

Once the pre-processing of data is done next, we apply the train data to the algorithm. There are several Machine learning algorithms to be used depending on the data you are going to process such as images,sound, text, and numerical values.The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms.

Example: 1. Linear Regression.

a. Logistic Regression.

b. Random Forest Regression / Classification.

c. Decision Tree Regression / Classification.

You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate.

Now we apply the Decision Tree algorithm on our dataset.

Machine Learning and Concepts:

Four machine learning models have been used for the prediction of loan approvals. Below are the description of the models used:

Decision Tree Regression Algorithm

This is a supervised machine learning algorithm that is primarily used for classification tasks. In this model, all features should be discretized such that the population may be divided into two or more homogenous groups or subsets. This model divides a node into two or more sub-nodes using a separate algorithm. The homogeneity and purity of the nodes grows in relation to the dependent variable as additional sub-nodes are created

By using DecisionTree we are fitting the model

```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)

y_pred_dt =dt.predict(X_test)  #prediction
c(y_pred_dt)
```

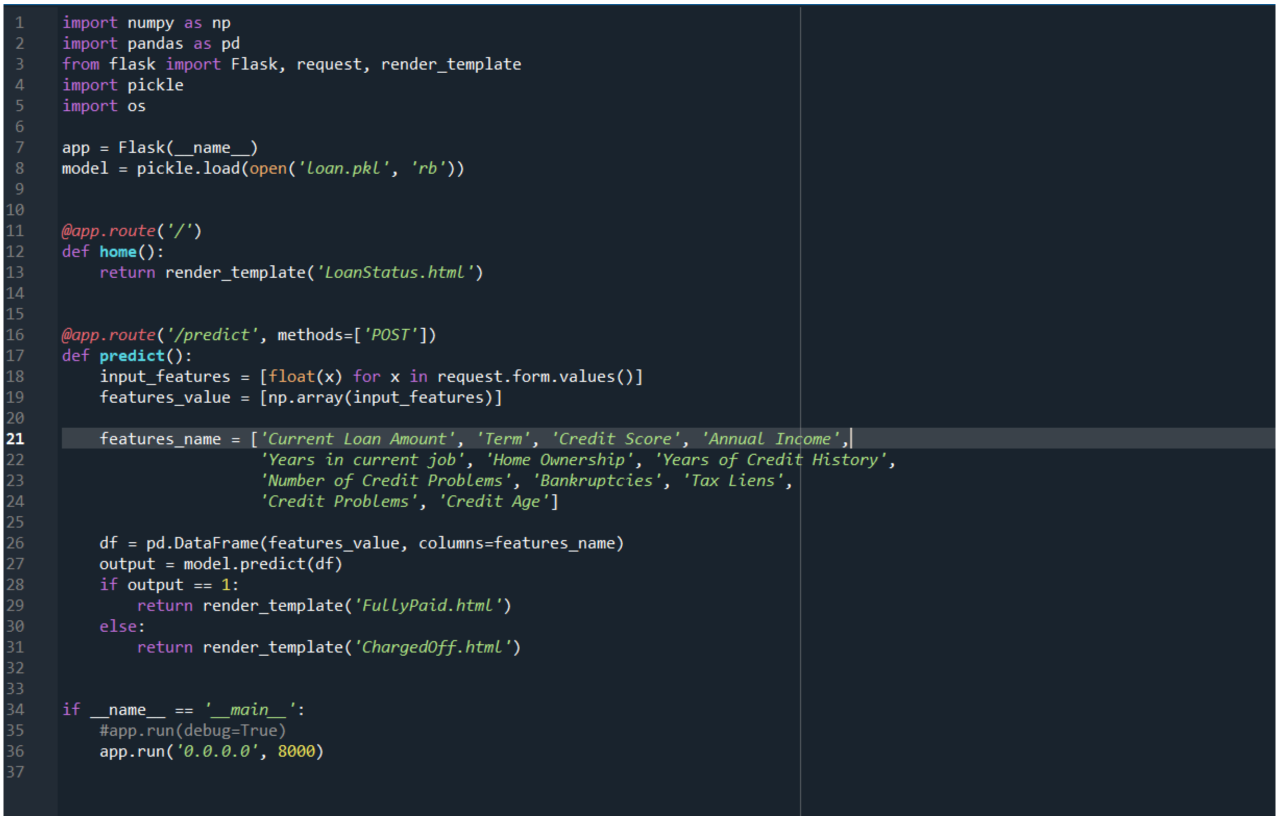
```
Counter({0: 6725, 1: 26275})
```

Creating a pickle file dumping the model in it

```
#importing the pickle file
import pickle
#Dumping the model into the pickle file
pickle.dump(dt,open('loan.pkl','wb'))
```

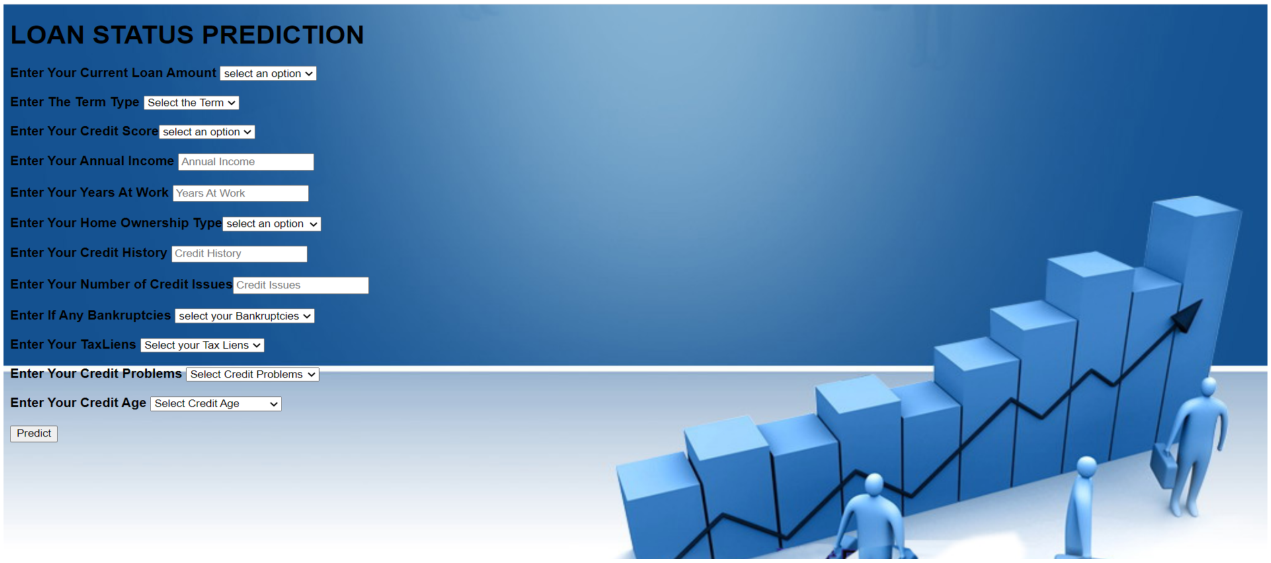
Build Flask Application

Flask Frame Work with Machine Learning Model In this section, we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.



```python
import numpy as np
import pandas as pd
from flask import Flask, request, render_template
import pickle
import os

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


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


@app.route('/predict', methods=['POST'])
def predict():
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]

    features_name = ['Current Loan Amount', 'Term', 'Credit Score', 'Annual Income',
                     'Years in current job', 'Home Ownership', 'Years of Credit History',
                     'Number of Credit Problems', 'Bankruptcies', 'Tax Liens',
                     'Credit Problems', 'Credit Age']

    df = pd.DataFrame(features_value, columns=features_name)
    output = model.predict(df)
    if output == 1:
        return render_template('FullyPaid.html')
    else:
        return render_template('ChargedOff.html')


if __name__ == '__main__':
    #app.run(debug=True)
    app.run('0.0.0.0', 8000)
```

Enter the values, press the predict button, and the result/prediction will be displayed on the web page



LOAN STATUS PREDICTION

Enter Your Current Loan Amount: select an option

Enter The Term Type: Select the Term

Enter Your Credit Score: select an option

Enter Your Annual Income: Annual Income

Enter Your Years At Work: Years At Work

Enter Your Home Ownership Type: select an option

Enter Your Credit History: Credit History

Enter Your Number of Credit Issues: Credit Issues

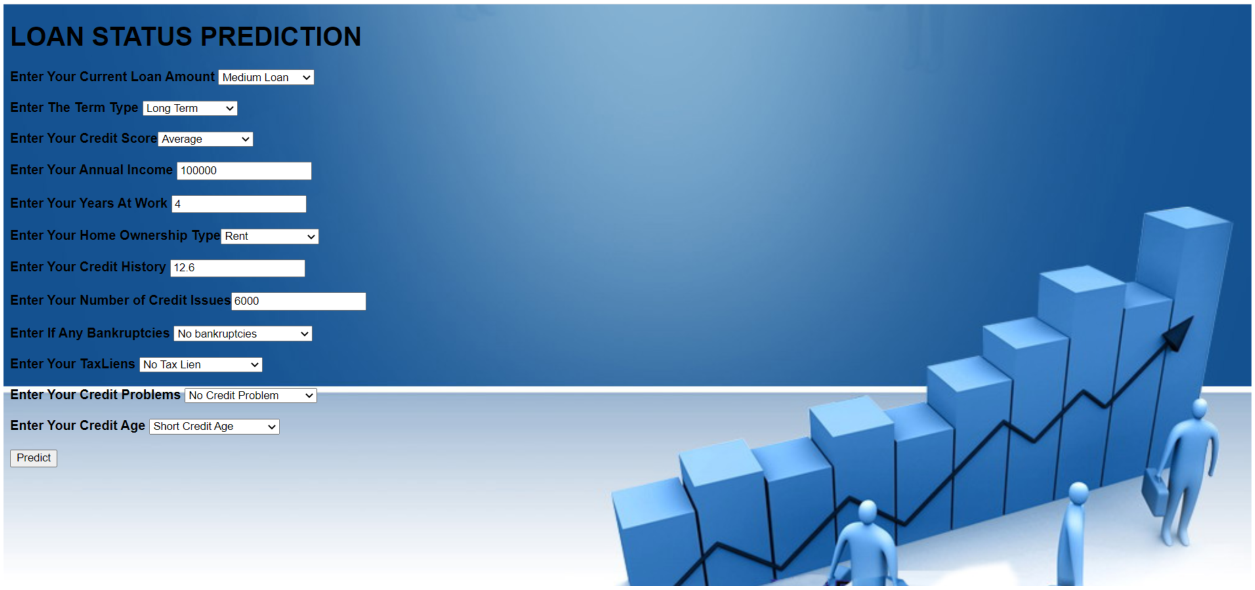
Enter If Any Bankruptcies: select your Bankruptcies

Enter Your TaxLiens: Select your Tax Liens

Enter Your Credit Problems: Select Credit Problems

Enter Your Credit Age: Select Credit Age

Input:



LOAN STATUS PREDICTION

Enter Your Current Loan Amount: Medium Loan

Enter The Term Type: Long Term

Enter Your Credit Score: Average

Enter Your Annual Income: 100000

Enter Your Years At Work: 4

Enter Your Home Ownership Type: Rent

Enter Your Credit History: 12.6

Enter Your Number of Credit Issues: 6000

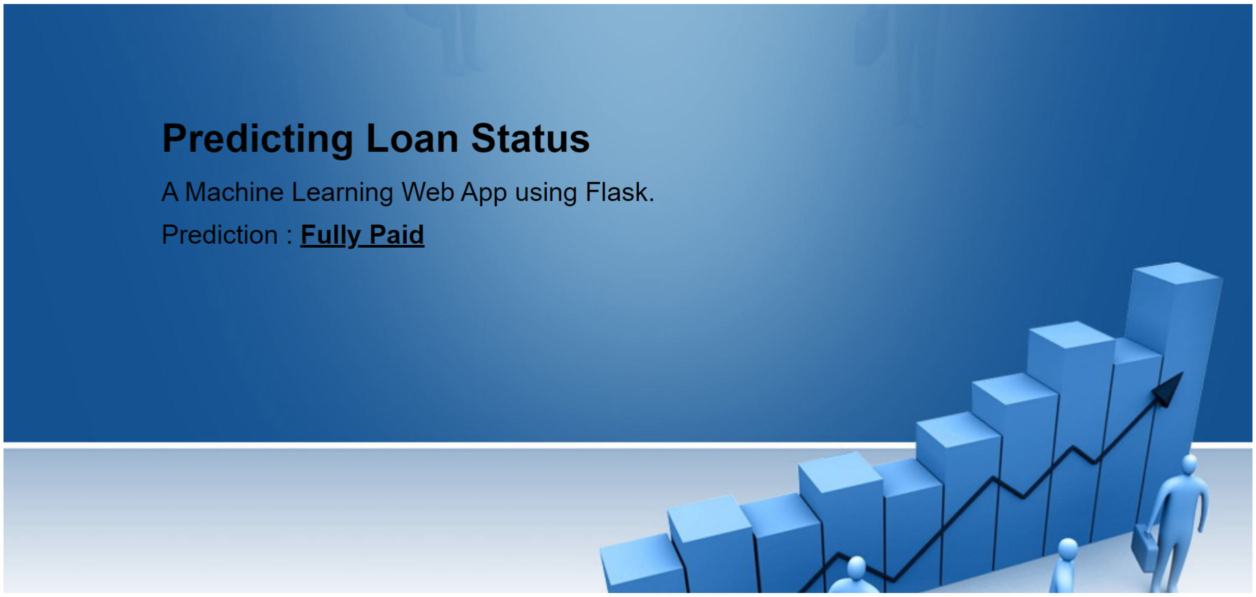
Enter If Any Bankruptcies: No bankruptcies

Enter Your TaxLiens: No Tax Lien

Enter Your Credit Problems: No Credit Problem

Enter Your Credit Age: Short Credit Age

Output:



Predicting Loan Status

A Machine Learning Web App using Flask.

Prediction : **Fully Paid**

Future scope:

In future, this model can be used to compare various machine learning algorithm generated prediction models and the model which will give higher accuracy will be chosen as the prediction model.

Conclusion:

We performed exploratory data analysis on the dataset's attributes to see how they are distributed.
We used charts to perform bivariate and multivariate analysis to understand how their features impacted one another.
We looked at each variable to see if the data was clean and evenly distributed. The data was cleansed and NA values were deleted.
We also devised hypotheses to demonstrate a link between the Independent and Target variables. And we inferred whether or not there is a link based on the findings.
We computed correlations between independent variables and discovered a substantial relationship between applicant income and loan amount.
To build the model, we added fake variables.
We built models that took a variety of variables into consideration and discovered that

credit credit history has the greatest impact on loan approval. Finally, we generated the most accurate model using coapplicant income and credit history as independent variables.

We tested the data and found it to be 69.17 percent accurate.