

# CREDIT CARD APPROVAL PREDICTION USING IBM WATSON MACHINE LEARNING

## 1. INTRODUCTION:

### 1.1 Overview:

With the increasing number of credit card applications, banks are opting towards the use of prediction-based algorithms as opposed to manual approval methods. Data analysis has exhibited a strong correlation between several financial and personal factors of a client and the likelihood of said client complying with their respective bank's credit policies. In this paper, we propose the use of the Machine Learning algorithm to predict and grant credit cards to applicants based on the customers' activity history. We used some financial and personal factors. We predicted the resulting factors through the use of Machine Learning algorithm with an emphasis on error minimization. Using this Machine Learning model, the machine-learned which of these applicants are most likely to accumulate bad debts and granted or rejected the applications based on the prediction.

### 1.2 Purpose:

Now a days every person needs a credit card but banks not provide credit card to everyone. Before giving the credit card the bank employee's needs to identify whether the person is fraud or good. But we are not able to analyse any one by seeing their face or look. For that we need to check his personal details like - Income, Education, Family, etc., But it is not possible to check manually now a days because population is very high and suppose if we are able to check then no one has that much of time. For time saving and growing the business we are using machine learning model. We are training the model based on historic data to check these details and after checking the details this will gives results at that time. So Machine Learning models save our time, money, energy etc.



## 2 LITERATURE SURVEY:

### 2.1 Existing problem:

Some of existing solution for solving this problem are:

Credit risk as the board in banks basically centers around deciding the probability of a customer's default or credit decay and how expensive it will end up being assuming it happens. It is important to consider major factors and predict beforehand the probability of consumers defaulting given their conditions. Which is where a machine learning model comes in handy and allows the banks and major financial institutions to predict whether the customer, they are giving the loan to, will default or not. This project builds a machine learning model with the best accuracy possible using python. First we load and view the dataset. The dataset has a combination of both mathematical and non-mathematical elements, that it contains values from various reaches, in addition to that it contains a few missing passages. We preprocess the dataset to guarantee the AI model we pick can make great expectations. After the information is looking great, some exploratory information examination is done to assemble our instincts. Finally, we will build a machine learning model that can predict if an individual's application for a credit card will be accepted.

-

### 2.2 Proposed solution:

We are proposed the method credit card approval prediction using IBM Watson by machine learning:

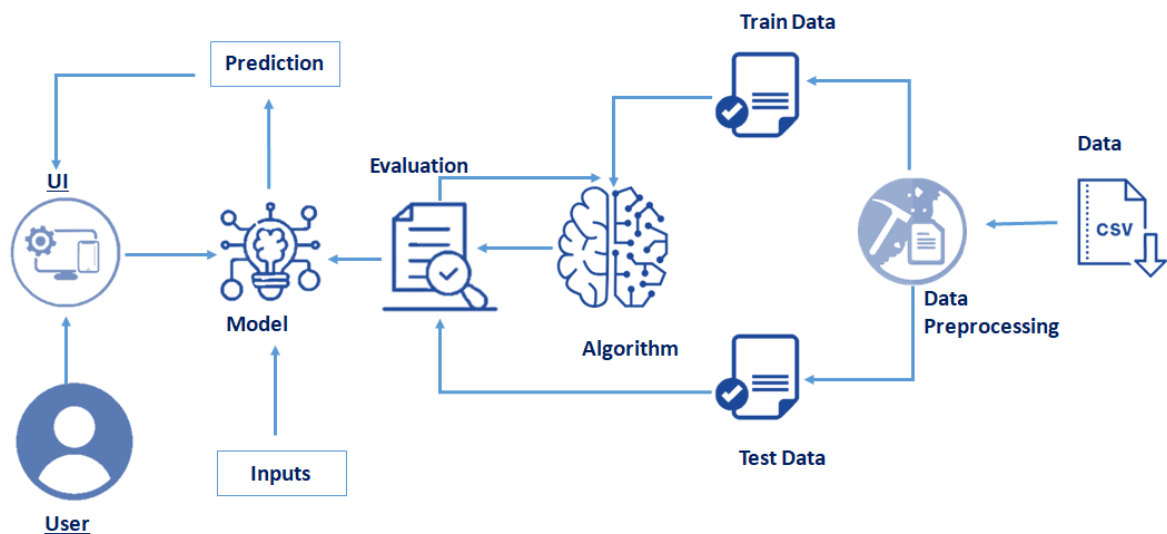
Here we can predict whether the person will eligible for the credit card or not. For this we can use the machine learning algorithm to train, test and implementation with the help of datasets. We can use the user interface for the user interactions.

The IBM Watson is an cloud service by using this we can service the new developer to access our data. How are done this application.



### 3. THEORITICAL ANALYSIS:

#### 3.1 Block Diagram:



#### 3.2 Hardware/software designing:

##### Hardware Requirements:

Operating system	Windows, Mac, Linux
CPU (for training)	Multi Core Processors (i3 or above/ equivalent)
GPU (for training)	NVIDIA AI Capable / Google's TPU

##### Software Requirements:

Python	V3.10.0 or Above
Python Packages	Flask, tensorflow, opencv-python, keras, numpy, Pandas, virtualenv, pillow

Web Browser	Mozilla Firefox, Google Chrome or any modern web browser
IBM Cloud (for training)	Watson Studio-Model Training & Deployment as Machine Learning Instance

## 4. EXPERIMENTAL INVESTIGATIONS:

### 4.1 Training the train dataset:

The screenshot shows a Jupyter Notebook window with the file 'creditcard.ipynb'. The code in the first cell splits the data into training and testing sets using `train_test_split`. The output displays the `xtest` and `ytrain` arrays. The `ytrain` array is a 1D integer array of length 7767, with values 0 and 1. A notification box at the bottom right asks if the user wants to install recommended extensions for Python.

```
#Splitting Data into Train And Test
x = credit_app[credit_app.dropped['target'], axis = 1].columns
y = credit_app['target']
xtrain, xtest, ytrain, ytest = train_test_split(x,y, train_size = 0.8, random_state = 0)
```

```
xtest
```

```
ytrain
```

```
8235 1
1755 0
509 0
5923 0
7722 0
..
7891 0
9225 0
4859 0
3264 0
2732 0
Name: target, Length: 7767, dtype: int64
```

The screenshot shows the same Jupyter Notebook window with the file 'creditcard.ipynb'. The code in the second cell defines three functions for building models: `logistic_reg` for Logistic Regression, `random_forest` for Random Forest Classifier, and `xgboost_model` for XGBoost Model. Each function takes `xtrain`, `xtest`, `ytrain`, and `ytest` as inputs and returns the predicted values and a classification report. A notification box at the bottom right asks if the user wants to install recommended extensions for Python.

```
#Model Building
#Logistic Regression Model
def logistic_reg(xtrain,xtest,ytrain,ytest):
    lr= LogisticRegression(solver='liblinear')
    lr.fit(xtrain,ytrain)
    ypred=lr.predict(xtest)
    print('***LogisticRegression***')
    print('Confusion matrix')
    print(confusion_matrix(ytest,ypred))
    print('Classification report')
    print(classification_report(ytest,ypred))
```

```
#Random Forest Classifier
def random_forest(xtrain,xtest,ytrain,ytest):
    rf=RandomForestClassifier()
    rf.fit(xtrain,ytrain)
    ypred=rf.predict(xtest)
    print('***Random Forest Classifier***')
    print('Confusion matrix')
    print(confusion_matrix(ytest,ypred))
    print('Classification report')
    print(classification_report(ytest,ypred))
```

```
#Xgboost Model
def xgboosting(xtrain,xtest,ytrain,ytest):
    gb=GradientBoostingClassifier()
    gb.fit(xtrain,ytrain)
    ypred=gb.predict(xtest)
    print('***Gradient BoostingClassifier***')
```



The screenshot shows a Jupyter Notebook with the following code:

```
#Decision Tree Model
def dt(xtrain,xtest,ytrain,ytest):
    dt=DecisionTreeClassifier()
    dt.fit(xtrain,ytrain)
    ypred=dt.predict(xtest)
    print('***DecisionTreeClassifier***')
    print(confusion_matrix(ytest,ypred))
    print(classification_report(ytest,ypred))

#compare the Model
def compare_model(xtrain,xtest,ytrain,ytest):
    logistic_rlg(xtrain,xtest,ytrain,ytest)
    print('***100')
    random_forest(xtrain,xtest,ytrain,ytest)
    print('***100')
    g_boosting(xtrain,xtest,ytrain,ytest)
    print('***100')
    d_tree(xtrain,xtest,ytrain,ytest)
```

The output shows a message: "Output exceeds the size limit. Open the full output data in a text editor".

The screenshot shows a Jupyter Notebook with the following code:

```
#save Model
dt=DecisionTreeClassifier()
dt.fit(xtrain,ytrain)
ypred=dt.predict(xtest)

import pickle
pickle.dump(dt,open("creditcard.pickle","wb"))
```

## Testing the test dataset:

The screenshot shows a Jupyter Notebook with the following code:

```
#importing the Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#from imblearn.combine import SMOTETomek

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import roc_auc_score, classification_report, confusion_matrix, accuracy_score, f1_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier

#read the datasets
app = pd.read_csv('/content/drive/myDrive/application_record.csv')
credit = pd.read_csv('/content/drive/myDrive/credit_record.csv')

app.head()

app.columns
```



Edit with WPS Office

File Edit Selection View Go Run Terminal Help Search

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

creditcard.ipynb X

C:\Users> muthy > OneDrive > Desktop > creditcard > CCredit card Aproval Predictionn > Flask > creditcard.ipynb > #connecting the google drive

+ Code + Markdown ...

```
#univariate Analysis
print("Number of people working status :")
print(app['OCCUPATION_TYPE'].value_counts())
sns.set(rc = {'figure.figsize':(18,6)})
sns.countplot(x='OCCUPATION_TYPE', data=app, palette = 'Set2')
```

[240]

Number of people working status :

Laborers	78240
Core staff	43007
Sales staff	41098
Managers	35487
Drivers	26090
High skill tech staff	17289
Accountants	15985
Medicine staff	13520
Cooking staff	8076
Security staff	7993
Cleaning staff	5845
Private service staff	3456
Low-skill Laborers	2140
Secretaries	2044
Walters/barman staff	1665
Health agents	1041
HR staff	774
IT staff	604

Name: OCCUPATION\_TYPE, dtype: int64

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f3db0e3dd50>

Python

Do you want to install the recommended extensions for Python?

Install Show Recommendations

25°C Sunny

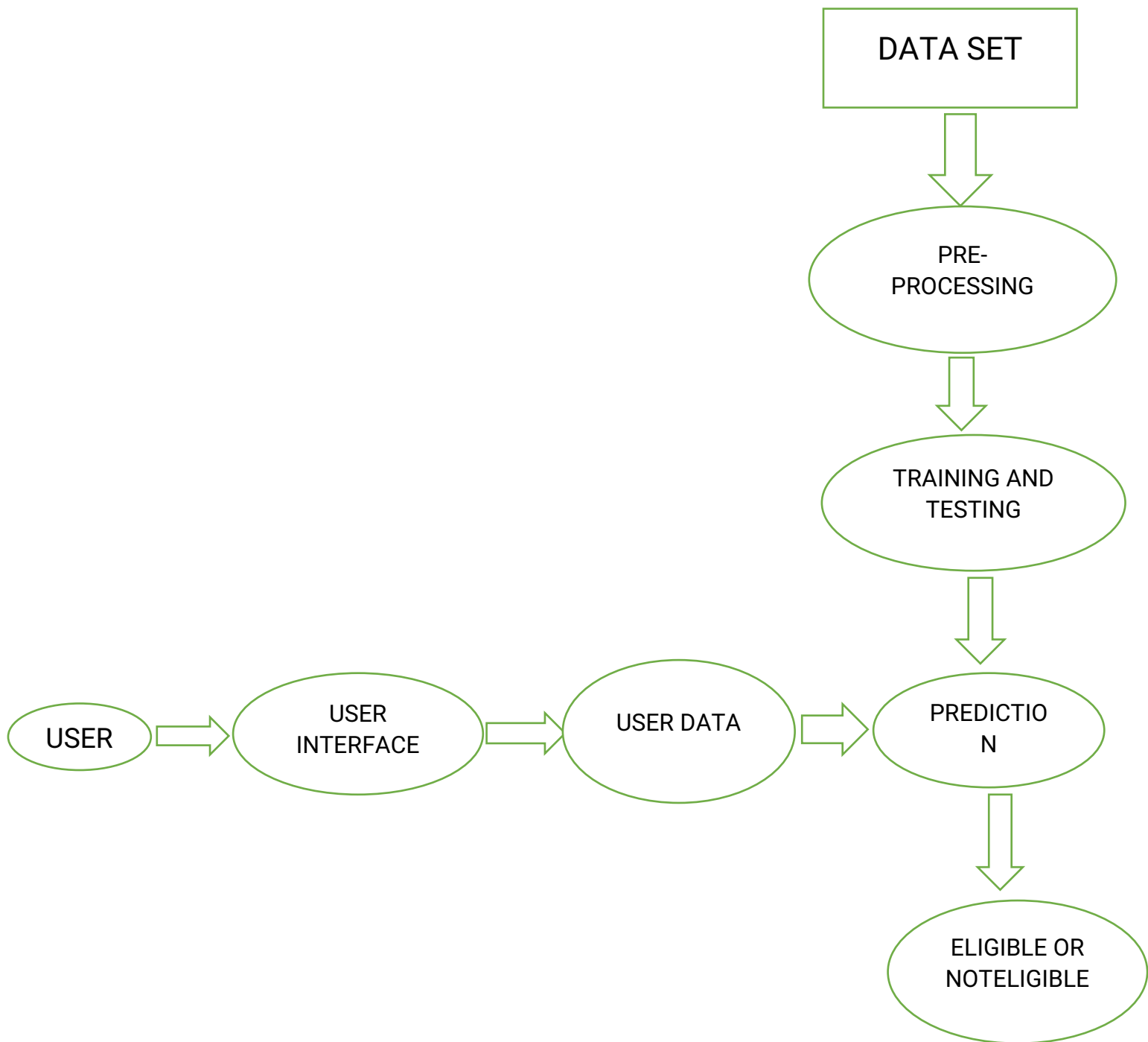
ENG US

11:06 11-11-2022



Edit with WPS Office

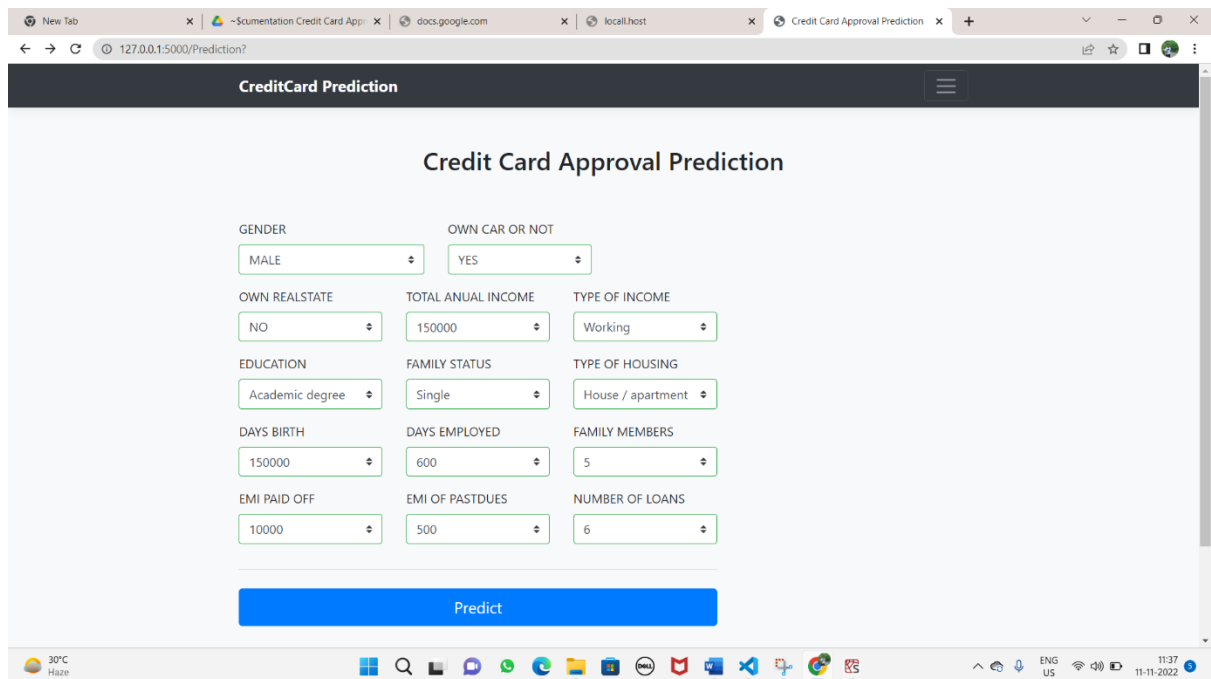
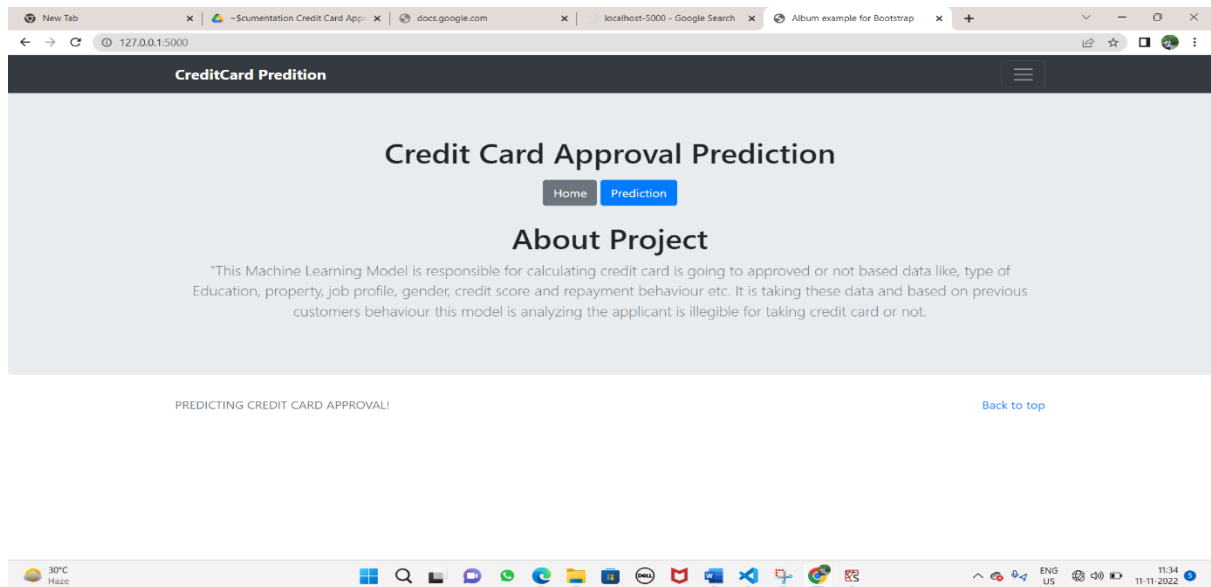
## 5 FLOWCHART:



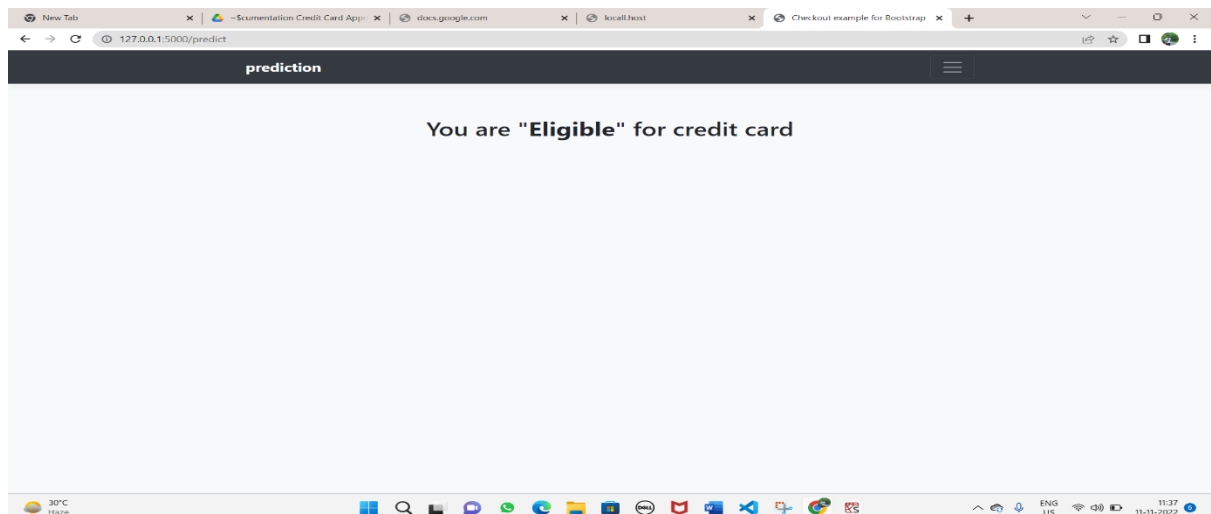
## 6 RESULT:

The proposed procedure was implemented and tested with set of Databases. The set of data is Read, pre-process, model building, training, testing and executed. The result is present in below

The output of Credit Card Prediction is provided below:







## 7 ADVANTAGES AND DISADVANTAGES:

### Advantages:

1. We can save the applicant time to whether the credit card approval is ACCEPTED OR REJECTED by using our UI.
2. Here the prediction is done Automatically without human errors.
3. The applicant will learn how much salary, job, loans get the credit card by using this application.
4. The applicant time is consumed.

### Disadvantage:

- 1 We can user the user interface with the internet connection.
- 2 The single error in data set can change the entire data.
- 3 Minimum due trap
- 4 Easy to overuse
- 5 High interest rate

## 8 APPLICATIONS:

- Authorized user for the Credit card.
- Remove stress button to bank employees.

## 9 CONCLUSION:

In this project, we will be using regression algorithms such as Decision tree, Random forest, KNN, and XGBoost. We will train and test the data with these algorithms. From this the best model is selected and saved in pickle format. We will be doing flask integration and IBM deployment.

This feature can predict whether the applicant will eligible for the credit card or not. By the data which is given by the applicant.

## 10 FUTURE SCOPE:

- In the future more and more customers will join with the banking industry, so immense amount for the will be generated handling which can be quiet an impossible task.
- So, in order to tackle with this situation this task can be automated with the power of machine learning which pretty much every bank does so nowadays.
- In the future this application can build as an app which can be present in the play store and every person will used it with the free of cost.
- We can extend this application with how much the limit of the card.

## 11 BIBILOGRAPHY:

- 1 Kaggle data set: <https://www.kaggle.com/namphuengauawatcharo/credit-card-approval-prediction/data>
- 2 Split data into train and test: <https://www.geeksforgeeks.org/how-to-split-a-dataset-into-train-and-test-sets-using-python/>
- 3 Register for IBM Cloud: <https://www.ibm.com/academic/home>
- 4 Login to IBM Cloud: <https://cloud.ibm.com/login>
- 5 The Train Model on IBM Cloud: <https://youtu.be/TysuP3KgSzc>

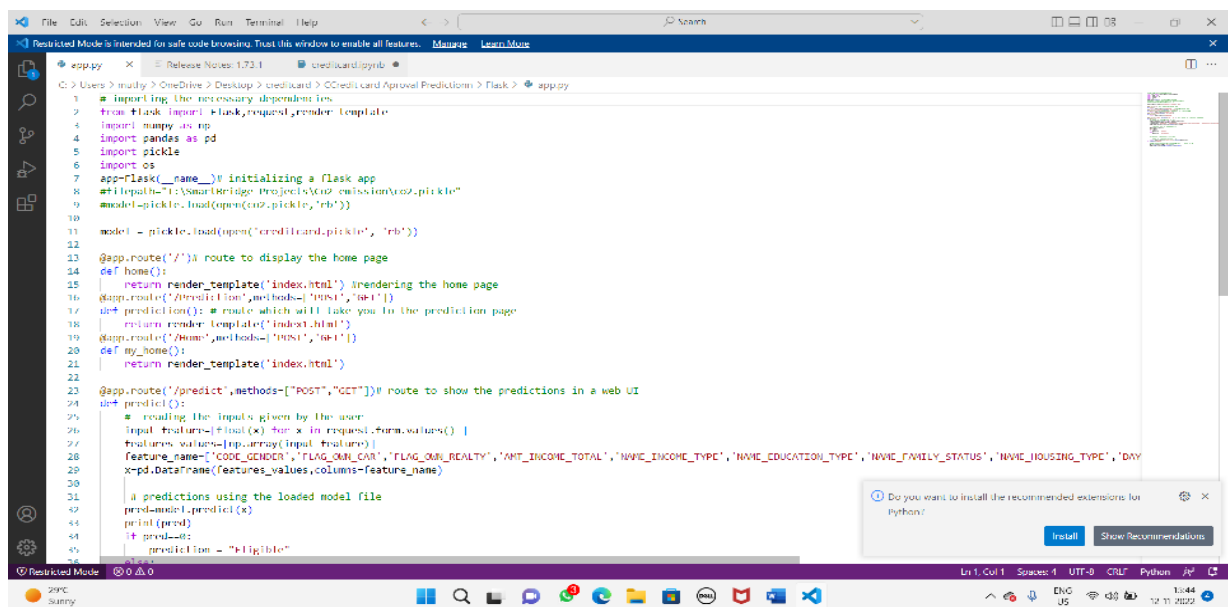


## APPENDIX :

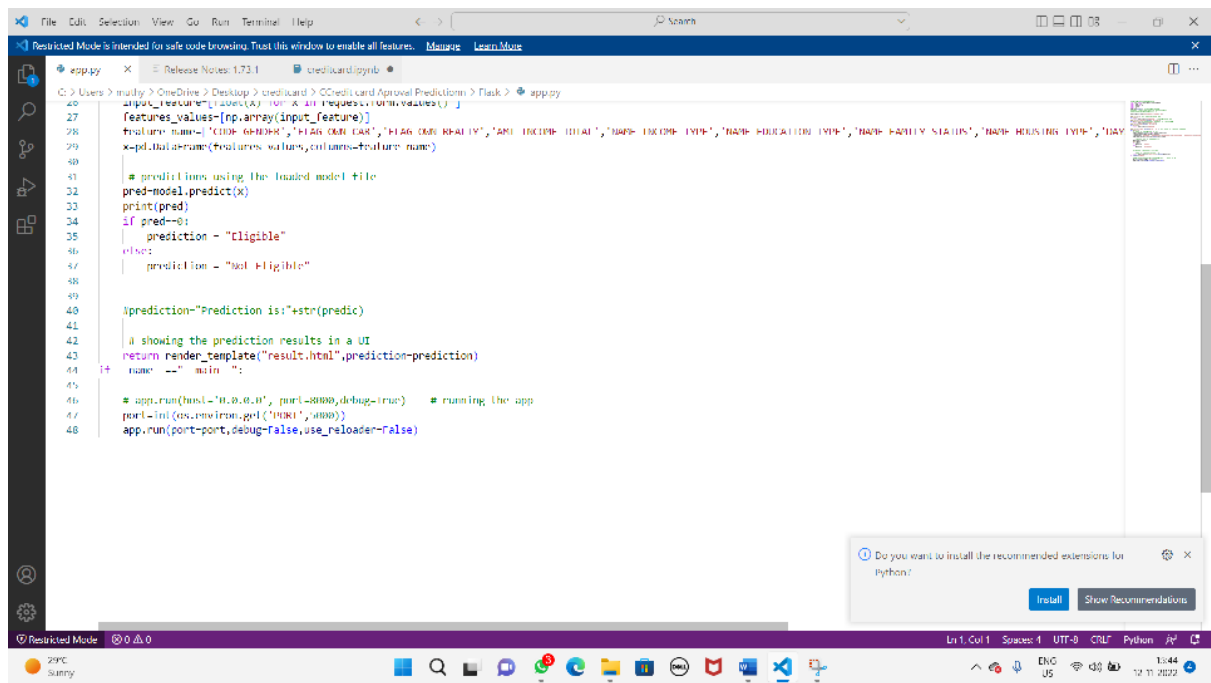
### A. COLAB NOTEBOOK

<https://colab.research.google.com/drive/1o8IGkUeOulan6ZWHd4cTdNtkSiQoLRZZ?usp=sharing>

### B. FLASK CODE



```
1 # Importing the necessary dependencies
2 from flask import Flask, request, render_template
3 import numpy as np
4 import pandas as pd
5 import pickle
6 import os
7 app = Flask(__name__) # initializing a flask app
8 # filepath = "C:\SmartBridge\project\src\emission\croz.pickle"
9 model = pickle.load(open('croz.pickle', 'rb'))
10
11 model = pickle.load(open('creditcard.pickle', 'rb'))
12
13 @app.route('/') # route to display the home page
14 def home():
15     return render_template('index.html') # rendering the home page
16 @app.route('/prediction', methods=['POST', 'GET'])
17 def prediction(): # route which will take you to the prediction page
18     return render_template('index.html')
19 @app.route('/home', methods=['POST', 'GET'])
20 def my_home():
21     return render_template('index.html')
22
23 @app.route('/predict', methods=['POST', 'GET']) # route to show the predictions in a web UI
24 def predict():
25     # reading the inputs given by the user
26     input_features = [float(x) for x in request.form.values()]
27     features_values = np.array(input_features)
28     feature_name = ["CODE_GENDER", "FLAG_OUR_CAR", "FLAG_OUR_REALTY", "AMT_INCOME_TOTAL", "NAME_INCOME_TYPE", "NAME_EDUCATION_TYPE", "NAME_FAMILY_STATUS", "NAME_HOUSING_TYPE", "DAY"]
29     x = pd.DataFrame(features_values, columns=feature_name)
30
31     # predictions using the loaded model file
32     preds = model.predict(x)
33     print(preds)
34     if preds == 0:
35         prediction = "Eligible"
36
37 Do you want to install the recommended extensions for Python?
```



Edit with WPS Office

## C. HTML FILES

### INDEX.html

```
Album example for Doubtless | Checkout example for Doubtless | Album example for Doubtless | index.html | New Tab
view-source:file:///C:/Users/mulhy/OneDrive/Desktop/creditcard/CreditPG/andSGO/ApprovedPG/Prediction/level/templates/index.html
1
2
3 <!doctype html>
4 <html lang="en">
5
6 <head>
7   <meta charset="utf-8">
8   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
9   <meta name="description" content="">
10  <meta name="author" content="">
11  <link rel="icon" href="static/img/logo.png">
12
13  <title>Album example for Doubtless</title>
14
15  <!-- Bootstrap core CSS -->
16  <link href="static/bootstrap.min.css" rel="stylesheet">
17
18  <!-- Custom styles for this template -->
19  <link href="album.css" rel="stylesheet">
20 </head>
21
22 <body>
23   <form action="/prediction" method="POST">
24     <div>
25
26       <div class="navber navbar-dark bg-dark box-shadow">
27         <div class="container d-flex justify-content-between">
28           <a href="#" class="navbar-brand d-flex align-items-center">
29
30             <strong>Credit Card Prediction</strong>
31
32           <div>
33             <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav"
34               aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
35               <span class="navbar-toggler-icon"></span>
36             </div>
37           </div>
38         </div>
39
40         <div class="main">
41
42           <section class="jumbotron text-center">
43             <div class="container">
44               <h1 class="jumbotron-heading">Credit Card Approval Prediction </h1>
45               <p>
46                 <a href="#" class="btn btn-secondary my-2">Home</a>
47
48                 <input type="submit" class="btn btn-primary my-2" value="Prediction">
49               </p>
50
51             </div>
52           </section>
53
54           <div class="text-center">
55             <h2 class="jumbotron-heading">About Predict </h2>
56             <p>
57               This Machine Learning Model is responsible for calculating credit card is going to approved
58               or not based on data like, type of Education, property, job profile, gender, credit score and repayment behaviour etc. It
59               is taking these data and based on previous customers behaviour this model is analyzing the applicant is eligible for
60               taking credit card or not.
61             </p>
62           </div>
63         </div>
64       </div>
65     </div>
66
67     <div class="text-center">
68       <div class="float-right">
69         <a href="#">Back to top</a>
70       </div>
71       <p>PREDICTING CREDIT CARD APPROVAL 1</p>
72     </div>
73   </form>
74
75   <!-- Bootstrap core JavaScript
76   <!-- Placed at the end of the document so the pages load faster -->
77   <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
78     integrity="sha384-J369zihV1bg564J160DA3yI19jIq412jJzH475R+wsA68D24gj5Rn36B97+<script>
79   <script src="static/js/vendor/popper.min.js"></script>
80   <script src="static/js/vendor/bootstrap.min.js"></script>
81   <script src="static/js/vendor/jquery.min.js"></script>
82   </form>
83 </body>
84 </html>
```

```
Album example for Doubtless | Checkout example for Doubtless | Album example for Doubtless | index.html | New Tab
view-source:file:///C:/Users/mulhy/OneDrive/Desktop/creditcard/CreditPG/andSGO/ApprovedPG/Prediction/level/templates/index.html
51
52 </div>
53
54 <div class="main">
55   <div class="jumbotron text-center">
56     <div class="container">
57       <h1 class="jumbotron-heading">Credit Card Approval Prediction </h1>
58       <p>
59         <a href="#" class="btn btn-secondary my-2">Home</a>
60
61         <input type="submit" class="btn btn-primary my-2" value="Prediction">
62       </p>
63
64       <h1 class="jumbotron-heading">About Predict </h1>
65       <p>
66         This Machine Learning Model is responsible for calculating credit card is going to approved
67         or not based on data like, type of Education, property, job profile, gender, credit score and repayment behaviour etc. It
68         is taking these data and based on previous customers behaviour this model is analyzing the applicant is eligible for
69         taking credit card or not.
70       </p>
71     </div>
72   </div>
73
74   <div class="text-center">
75     <div class="float-right">
76       <a href="#">Back to top</a>
77     </div>
78     <p>PREDICTING CREDIT CARD APPROVAL 1</p>
79   </div>
80 </div>
81
82 <div class="text-center">
83   <div class="float-right">
84     <a href="#">Back to top</a>
85   </div>
86   <p>PREDICTING CREDIT CARD APPROVAL 1</p>
87 </div>
88
89 <!-- Bootstrap core JavaScript
90 <!-- Placed at the end of the document so the pages load faster -->
91 <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
92   integrity="sha384-J369zihV1bg564J160DA3yI19jIq412jJzH475R+wsA68D24gj5Rn36B97+<script>
93 <script src="static/js/vendor/popper.min.js"></script>
94 <script src="static/js/vendor/bootstrap.min.js"></script>
95 <script src="static/js/vendor/jquery.min.js"></script>
96 </form>
97 </body>
98 </html>
```

# INDEX1.html

```
view sourcefile:///C:/Users/mutyh/OneDrive/Desktop/creditcard/Credit%20Card%20Approval%20Predictionn/Flask/templates/index1.html
<!-- Bootstrap core CSS -->
<!-- Custom styles for this template -->
</head>
<body class="bg-light">
  <div class="navber navbar-dark bg-dark box-shadow">
    <div class="container d-flex justify-content-between">
      <a href="#" class="navbar-brand d-flex align-items-center">
        <strong>CreditCard Prediction</strong>
      </a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarheader"
        aria-controls="navbarheader" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"/>
      </button>
    </div>
  </div>
  <div class="py-5 text-center">
    <h2>Credit Card Approval Prediction</h2>
  </div>
  <div class="row">
    <div class="col-md-8 order-md-1">
      <form action="/predict1/" method="POST" class="needs-validation" novalidate>
        <div class="form">
          <div class="col-md-6 mb-3">
            <label for="FIRST_NAME">FIRST NAME</label>
            <select class="custom-select d-block w-100" id="CUSTOMER_ID" name="CUSTOMER_ID" required>
              <option>Select the CUSTOMER</option>
              <option value="0">FRANK</option>
              <option value="1">PAUL</option>
            </select>
          </div>
          <div class="col-md-6 mb-3">
            <label for="TINAC_CARD">OWN CAR OR NOT</label>
            <select class="custom-select d-block w-100" id="CountryCode" name="CountryCode" required>
              <option value="">Select the Y/N</option>
              <option value="0">NO</option>
              <option value="1">YES</option>
            </select>
          </div>
          <div class="col-md-6 mb-3">
            <label for="LAYS_OUR_RLALTY">OUR RLALTY</label>
            <select class="custom-select d-block w-100" id="TINAC_CARD_RPT" name="TINAC_CARD_RPT" required>
              <option value="">SELECT Y/N</option>
              <option value="0">NO</option>
              <option value="1">YES</option>
            </select>
          </div>
          <div class="col-md-6 mb-3">
            <label for="NAME_TINCOM_TOTAL">TOTAL AMOUNT</label>
            <input type="number" class="custom-select d-block w-100" id="AMOUNT TOTAL" name="AMOUNT TOTAL" required>
          </div>
          <div class="col-md-6 mb-3">
            <label for="NAME_TINCOM_TYPE">TYPE OF TINCOM</label>
            <select class="custom-select d-block w-100" id="NAME TINCOM TYPE" name="NAME TINCOM TYPE" required>
              <option value="">SELECT Y/N</option>
              <option value="0">NO</option>
              <option value="1">YES</option>
            </select>
          </div>
        </div>
      </form>
    </div>
  </div>
```

```
view sourcefile:///C:/Users/mutyh/OneDrive/Desktop/creditcard/Credit%20Card%20Approval%20Predictionn/Flask/templates/index1.html
<div class="form">
  <div class="col-md-6 mb-3">
    <label for="FIRST_NAME">FIRST NAME</label>
    <select class="custom-select d-block w-100" id="CUSTOMER_ID" name="CUSTOMER_ID" required>
      <option>Select the CUSTOMER</option>
      <option value="0">FRANK</option>
      <option value="1">PAUL</option>
    </select>
  </div>
  <div class="col-md-6 mb-3">
    <label for="TINAC_CARD">OWN CAR OR NOT</label>
    <select class="custom-select d-block w-100" id="CountryCode" name="CountryCode" required>
      <option value="">Select the Y/N</option>
      <option value="0">NO</option>
      <option value="1">YES</option>
    </select>
  </div>
  <div class="col-md-6 mb-3">
    <label for="LAYS_OUR_RLALTY">OUR RLALTY</label>
    <select class="custom-select d-block w-100" id="TINAC_CARD_RPT" name="TINAC_CARD_RPT" required>
      <option value="">SELECT Y/N</option>
      <option value="0">NO</option>
      <option value="1">YES</option>
    </select>
  </div>
  <div class="col-md-6 mb-3">
    <label for="NAME_TINCOM_TOTAL">TOTAL AMOUNT</label>
    <input type="number" class="custom-select d-block w-100" id="AMOUNT TOTAL" name="AMOUNT TOTAL" required>
  </div>
  <div class="col-md-6 mb-3">
    <label for="NAME_TINCOM_TYPE">TYPE OF TINCOM</label>
    <select class="custom-select d-block w-100" id="NAME TINCOM TYPE" name="NAME TINCOM TYPE" required>
      <option value="">SELECT Y/N</option>
      <option value="0">NO</option>
      <option value="1">YES</option>
    </select>
  </div>
</div>
```



[illegible]