

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:

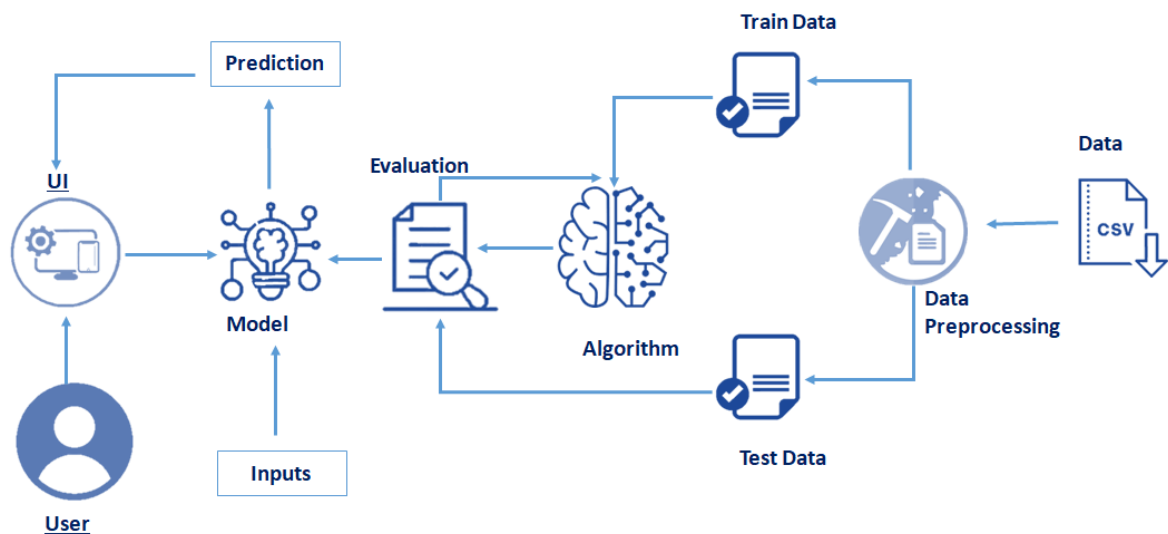
We are proposed the method credit card approval prediction using IBM Watson by machine learning:

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:

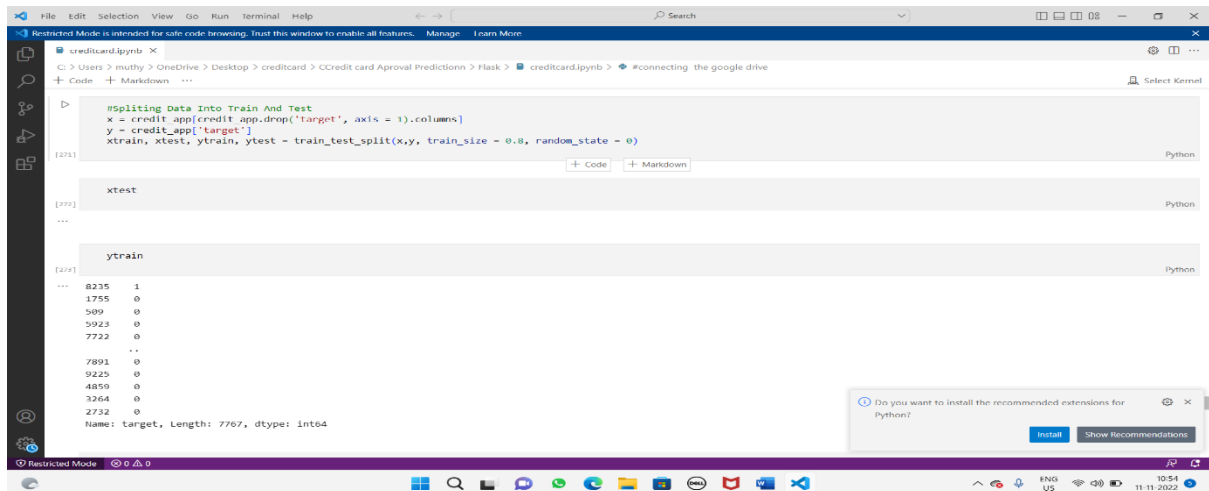
Operating system	Windows, Mac, Linux
CPU (for training)	Multi Core Processors (i3 or above/ equivalent)
GPU (for training)	NVIDIA AI Capable / Google's TPU

Software Requirements:

Python	V3.10.0 or Above
Python Packages	Flask, tensorflow, opencv-python, keras, nump , Pandas, virtualenv, pillow
Web Browser	Mozilla Firefox, Google Chorme 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:



This screenshot shows the first part of a Jupyter Notebook. The code splits the data into training and testing sets using `train_test_split`. It then displays the training set (`ytrain`) and the testing set (`xtest`). A message box prompts the user to install recommended extensions for Python.

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

[275]

```
xtest
```

[277]

...

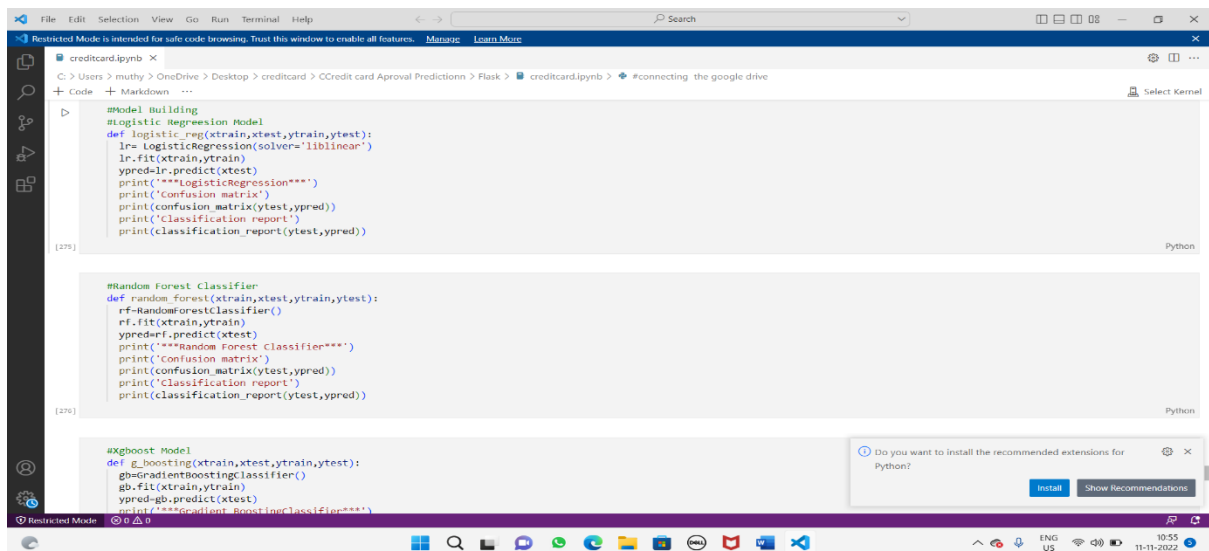
```
ytrain
```

[285]

```
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
```

Do you want to install the recommended extensions for Python?

Install Show Recommendations



This screenshot shows the second part of the Jupyter Notebook. It defines three functions: `logistic_reg` for building a Logistic Regression model, `random_forest` for building a Random Forest Classifier, and `g_boost` for building an XGBoost model. Each function prints the confusion matrix and classification report.

```
#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))
```

[276]

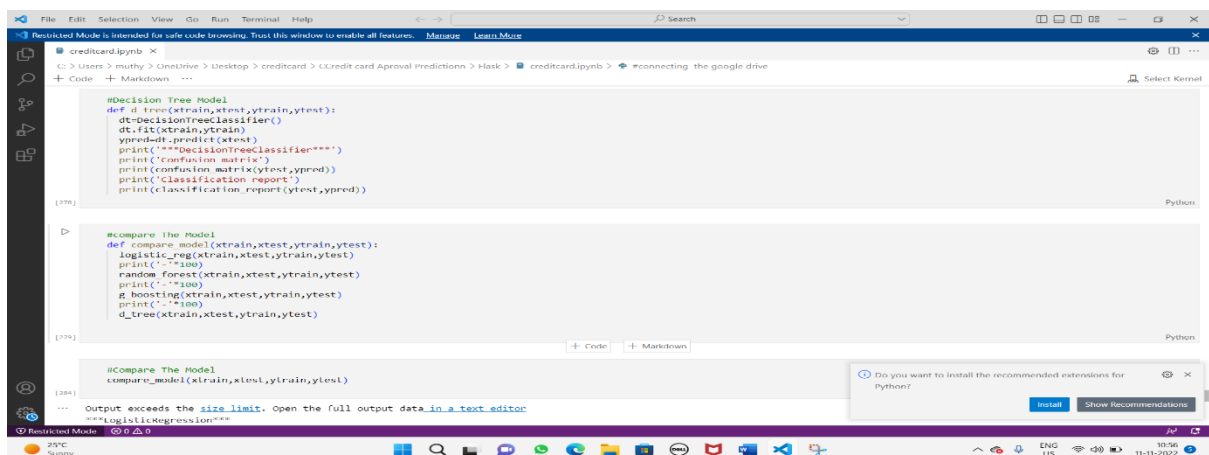
```
#Random Forest Classifier
def random_forest(xtrain,xtest,ytrain,ytest):
    rf=RandomForestClassifier()
    rf.fit(xtrain,ytrain)
    ypred=rf.predict(xtest)
    print('***Random Forest Classifier***')
    print('Confusion matrix')
    print(confusion_matrix(ytest,ypred))
    print('Classification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

[276]

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

Do you want to install the recommended extensions for Python?

Install Show Recommendations



This screenshot shows the third part of the Jupyter Notebook. It defines a function `dt` for building a Decision Tree model and a function `compare_model` to compare the performance of the Logistic Regression, Random Forest, XGBoost, and Decision Tree models. The `compare_model` function prints the confusion matrix and classification report for each model.

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

[278]

```
#compare the Model
def compare_model(xtrain,xtest,ytrain,ytest):
    logistic_reg(xtrain,xtest,ytrain,ytest)
    print('***100')
    random_forest(xtrain,xtest,ytrain,ytest)
    print('***100')
    g_boost(xtrain,xtest,ytrain,ytest)
    print('***100')
    dt(xtrain,xtest,ytrain,ytest)
```

[279]

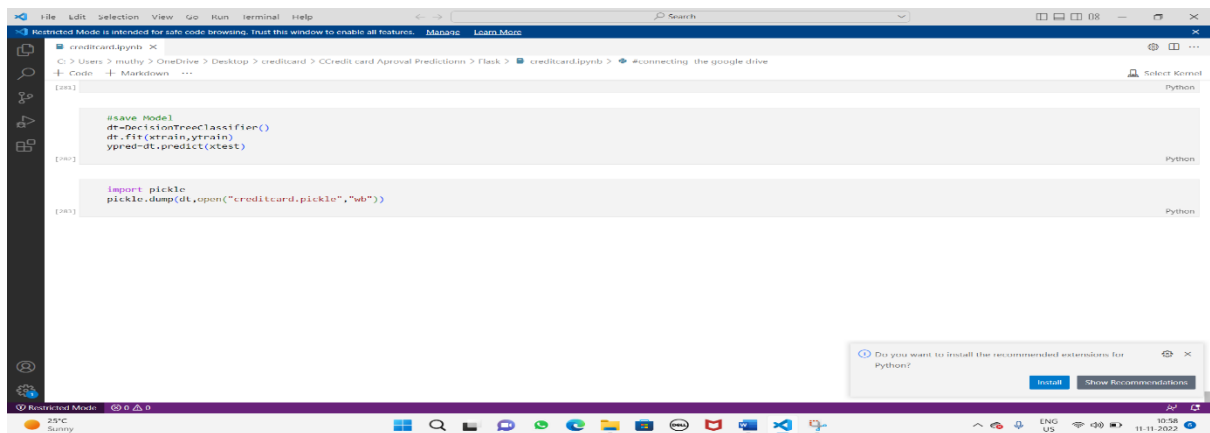
```
#Compare The Model
compare_model(xtrain,xtest,ytrain,ytest)
```

[284]

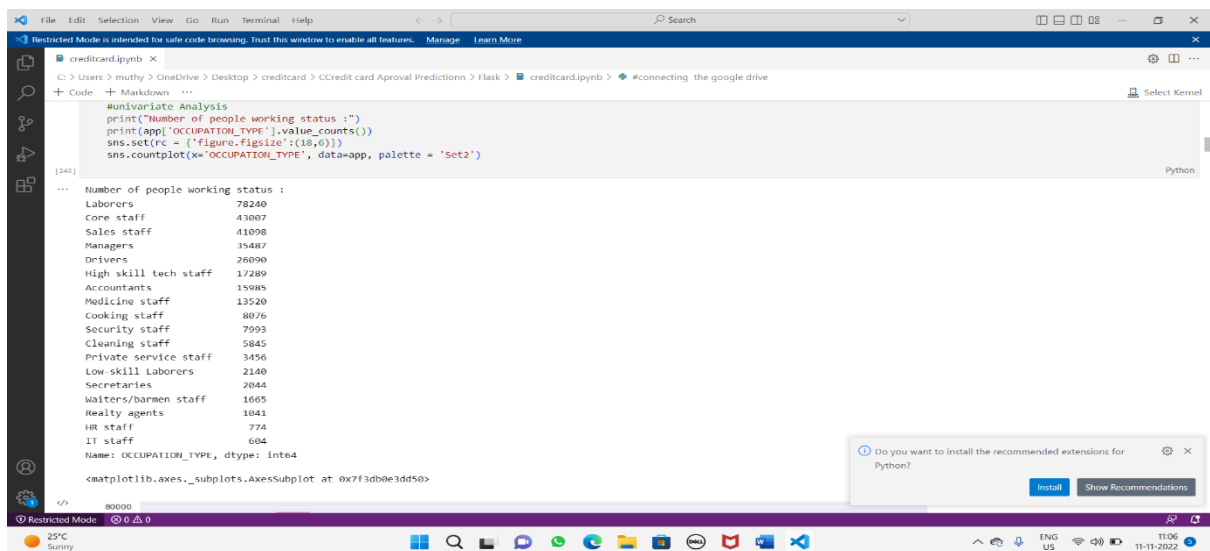
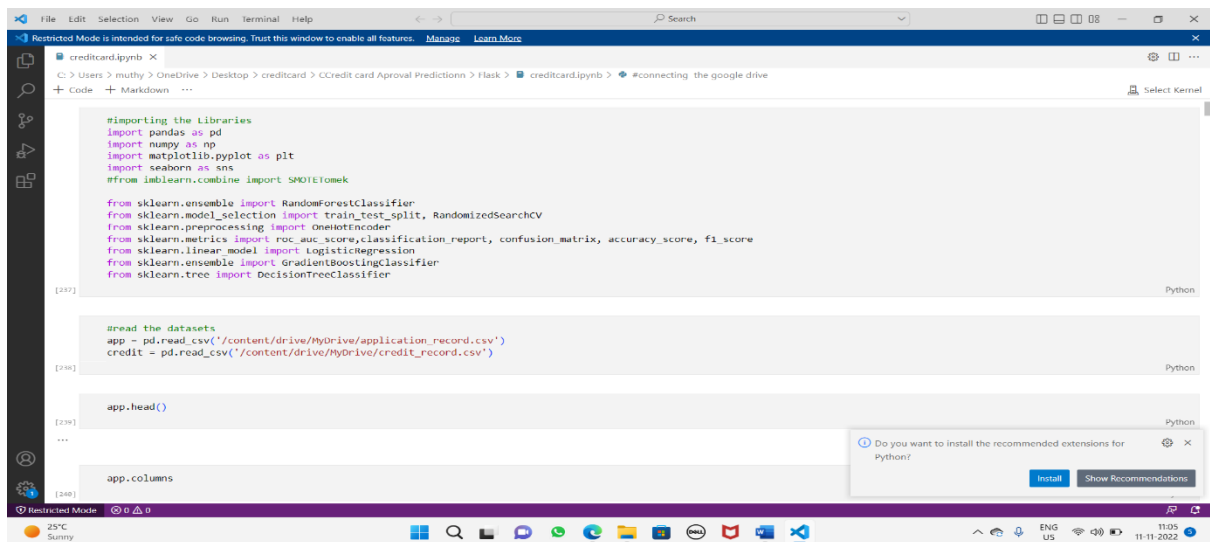
Output exceeds the size limit. open the full output data in a text editor

Do you want to install the recommended extensions for Python?

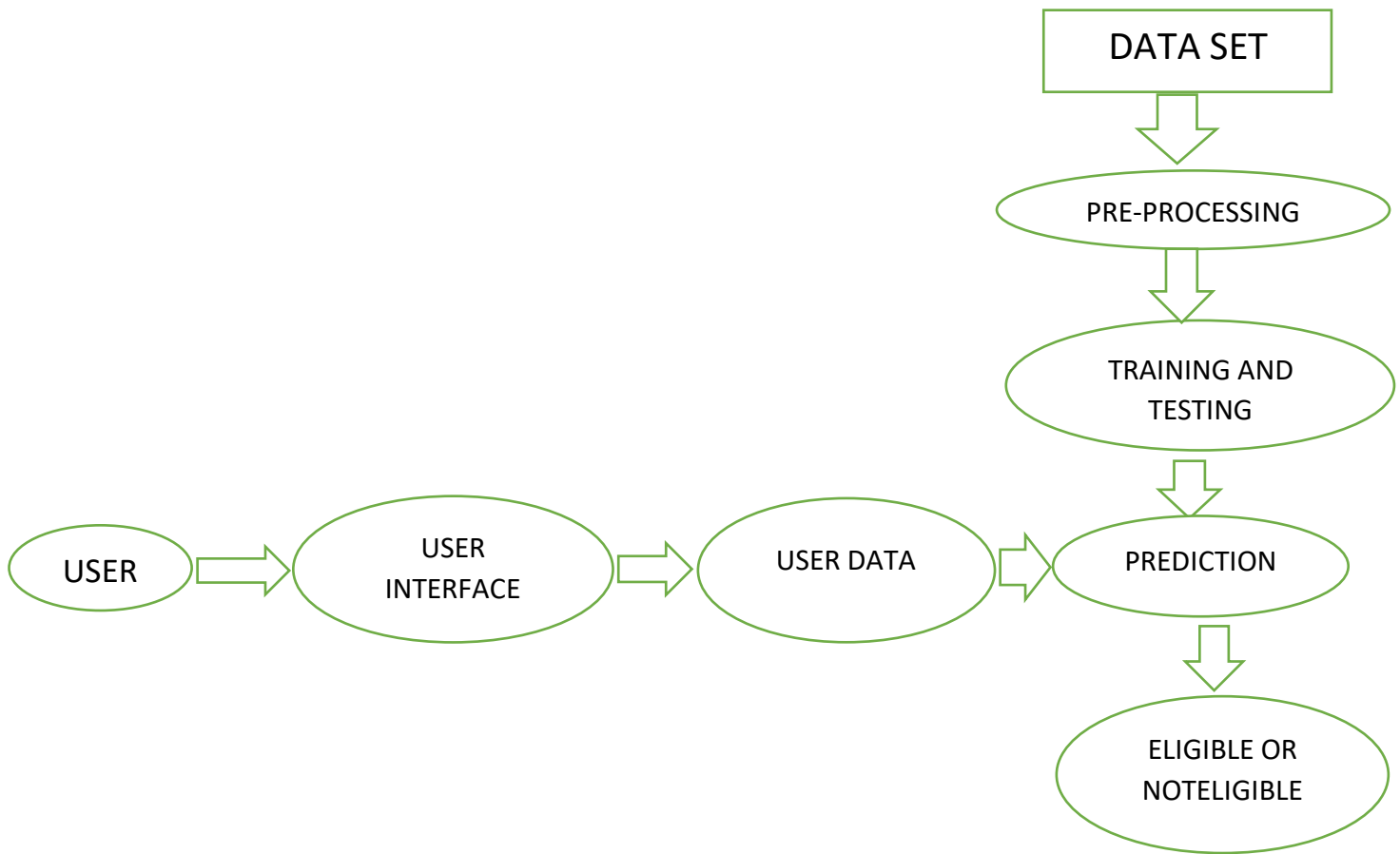
Install Show Recommendations



Testing the test dataset:



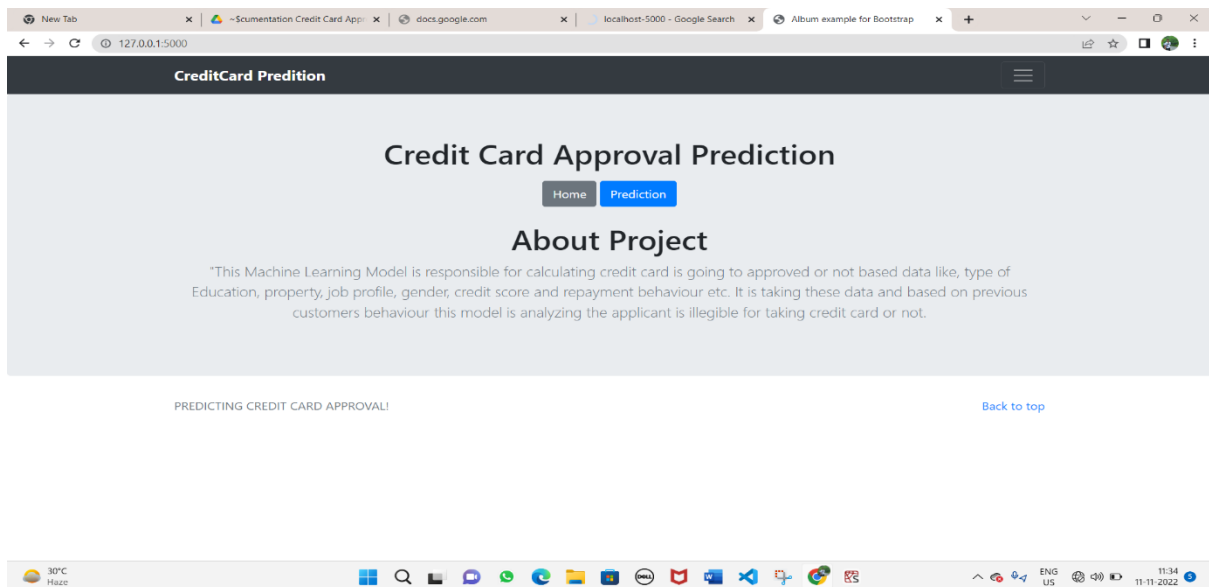
5 FLOWCHART:



6 RESULT:

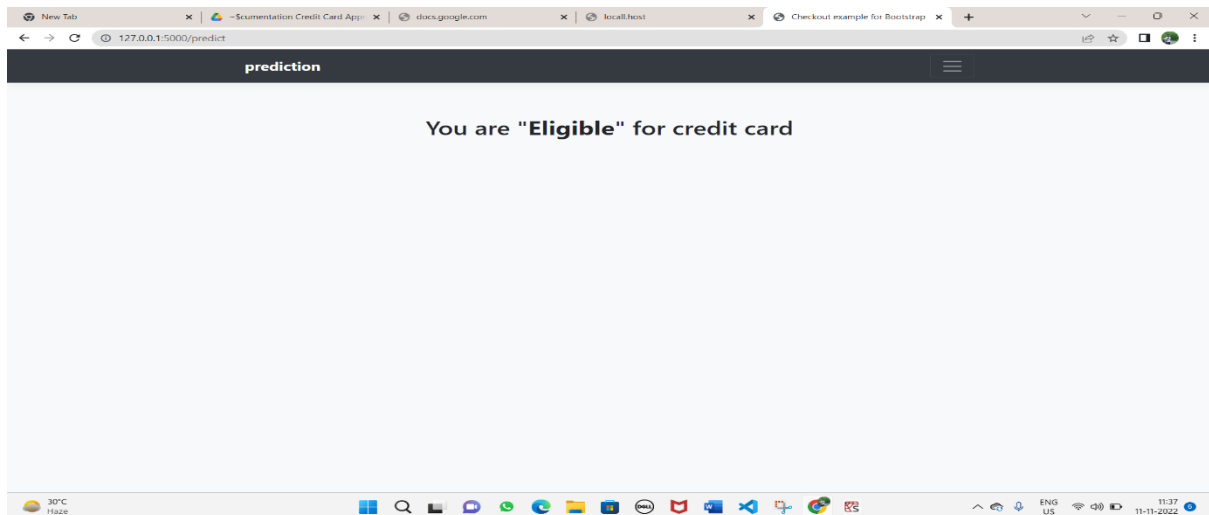
The proposed procedure was implemented and tested with set of Databases. The set of data is Read, pre-process, model building, training, testing and executed. The result is present in below

The output of Credit Card Prediction is provided below:



The screenshot shows the same web browser window, but now displaying the "Prediction" form. The form is titled "Credit Card Approval Prediction" and contains several input fields for user data. The fields are organized into a grid-like structure. At the bottom of the form is a large blue "Predict" button. The browser's address bar shows the URL 127.0.0.1:5000/Prediction?.

GENDER	OWN CAR OR NOT	OWN REALSTATE	TOTAL ANNUAL INCOME	TYPE OF INCOME	EDUCATION	FAMILY STATUS	TYPE OF HOUSING	DAYS BIRTH	DAYS EMPLOYED	FAMILY MEMBERS	EMI PAID OFF	EMI OF PASTDUES	NUMBER OF LOANS
MALE	YES	NO	150000	Working	Academic degree	Single	House / apartment	150000	600	5	10000	500	6



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 use 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:

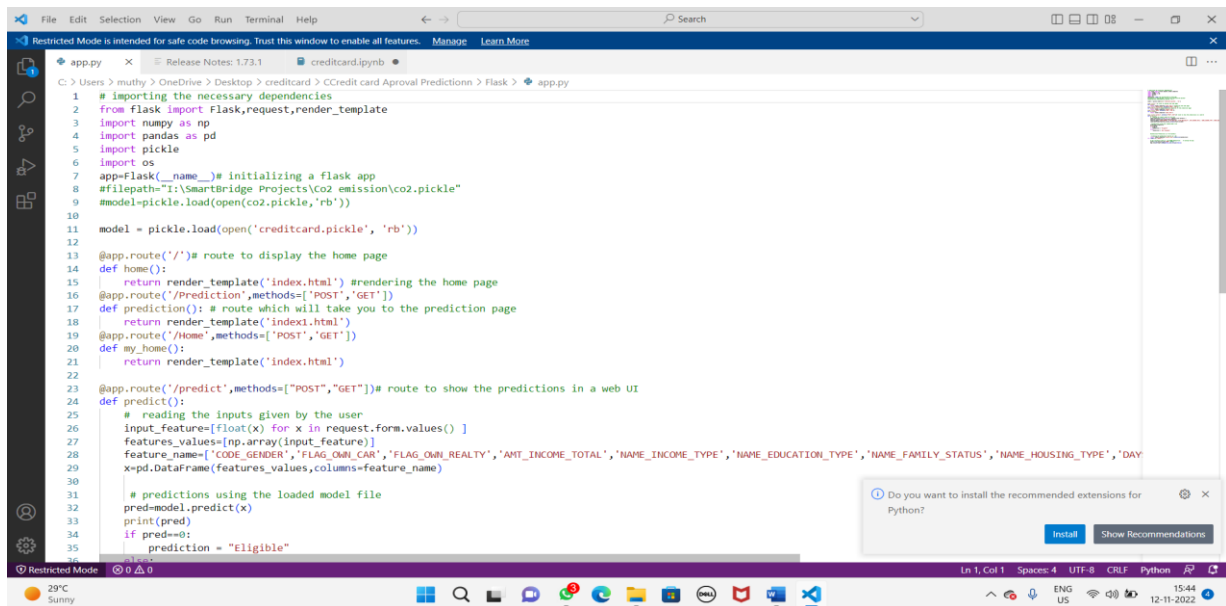
- 1 Kaggle data set: <https://www.kaggle.com/namphuengauawatcharo/credit-card-approval-prediction/data>
- 2 Split data into train and test: <https://www.geeksforgeeks.org/how-to-split-a-dataset-into-train-and-test-sets-using-python/>
- 3 Register for IBM Cloud: <https://www.ibm.com/academic/home>
- 4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

APPENDIX :

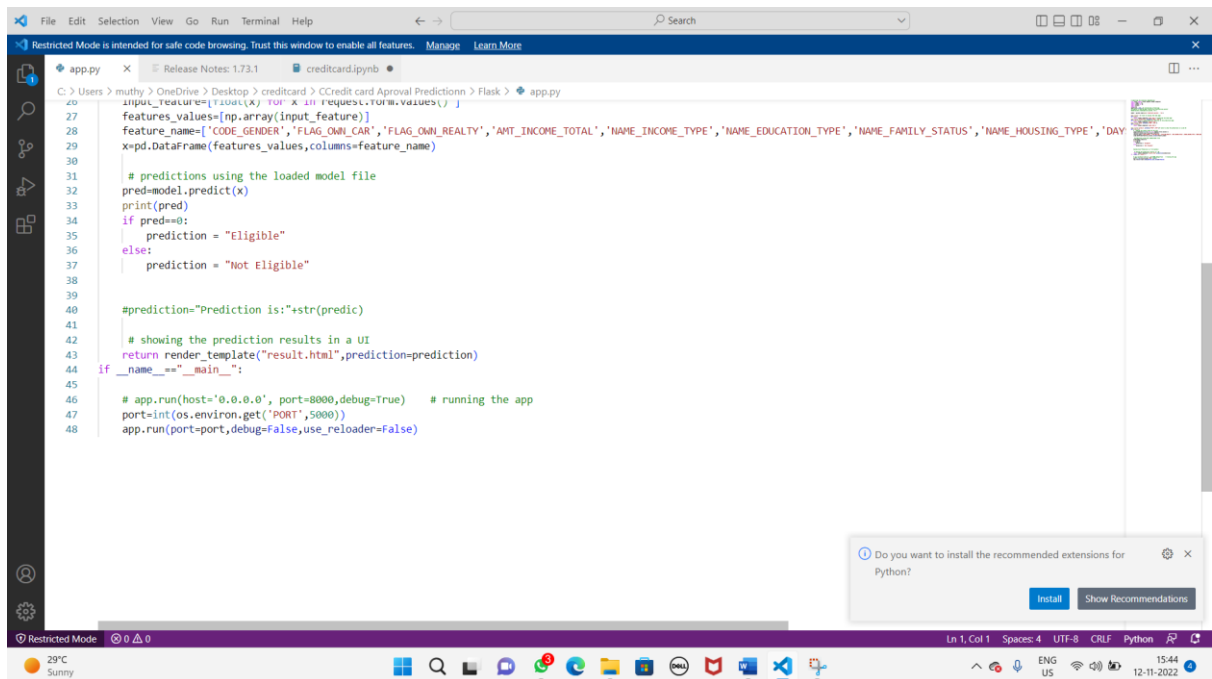
A. COLAB NOTEBOOK

<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="I:\SmartBridge Projects\Co2 emission\co2.pickle"
9 #model=pickle.load(open(co2.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_feature = float(x) for x in request.form.values()
27     features_values = np.array(input_feature)
28     feature_name = ['CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_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     pred = model.predict(x)
33     print(pred)
34     if pred == 0:
35         prediction = "Eligible"
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```



```
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="I:\SmartBridge Projects\Co2 emission\co2.pickle"
9 #model=pickle.load(open(co2.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_feature = float(x) for x in request.form.values()
27     features_values = np.array(input_feature)
28     feature_name = ['CODE_GENDER', 'FLAG_OWN_CAR', 'FLAG_OWN_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     pred = model.predict(x)
33     print(pred)
34     if pred == 0:
35         prediction = "Eligible"
36     else:
37         prediction = "Not Eligible"
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

C. HTML FILES

INDEX.html

```
Album example for Bootstrap Checkout example for Bootstrap Album example for Bootstrap index.html New Tab
view-source:file:///C:/Users/muthy/OneDrive/Desktop/creditcard/CCredit%20card%20Approval%20Predictionn/Flask/templates/index.html

Line wrap
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/logo.png">
12
13  <title>Album example for Bootstrap</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,GET]">
24     <header>
25
26       <div class="navbar 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>CreditCard Prediction</strong>
31           </a>
32           <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarHeader"
33             aria-controls="navbarHeader" aria-expanded="false" aria-label="Toggle navigation">
34             <span class="navbar-toggler-icon"></span>
35           </button>
36         </div>
37       </div>
38     </header>
39
40     <main role="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         </div>
51       </section>
52
53       <div class="jumbotron text-center">
54         <h1>About Project</h1>
55         <p>This Machine Learning Model is responsible for calculating credit card is going to approved
56           or not based data like, type of Education, property, job profile, gender, credit score and repayment behaviour etc. It
57           is taking these data and based on previous customers behaviour this model is analyzing the applicant is illegible for
58           taking credit card or not.
59         </p>
60       </div>
61
62     </main>
63
64     <footer class="text-muted">
65       <div class="container">
66         <div class="float-right">
67           <a href="#">Back to top</a>
68         </div>
69       </div>
70     </footer>
71
72   </body>
73 </html>
```

```
Album example for Bootstrap Checkout example for Bootstrap Album example for Bootstrap index.html New Tab
view-source:file:///C:/Users/muthy/OneDrive/Desktop/creditcard/CCredit%20card%20Approval%20Predictionn/Flask/templates/index.html

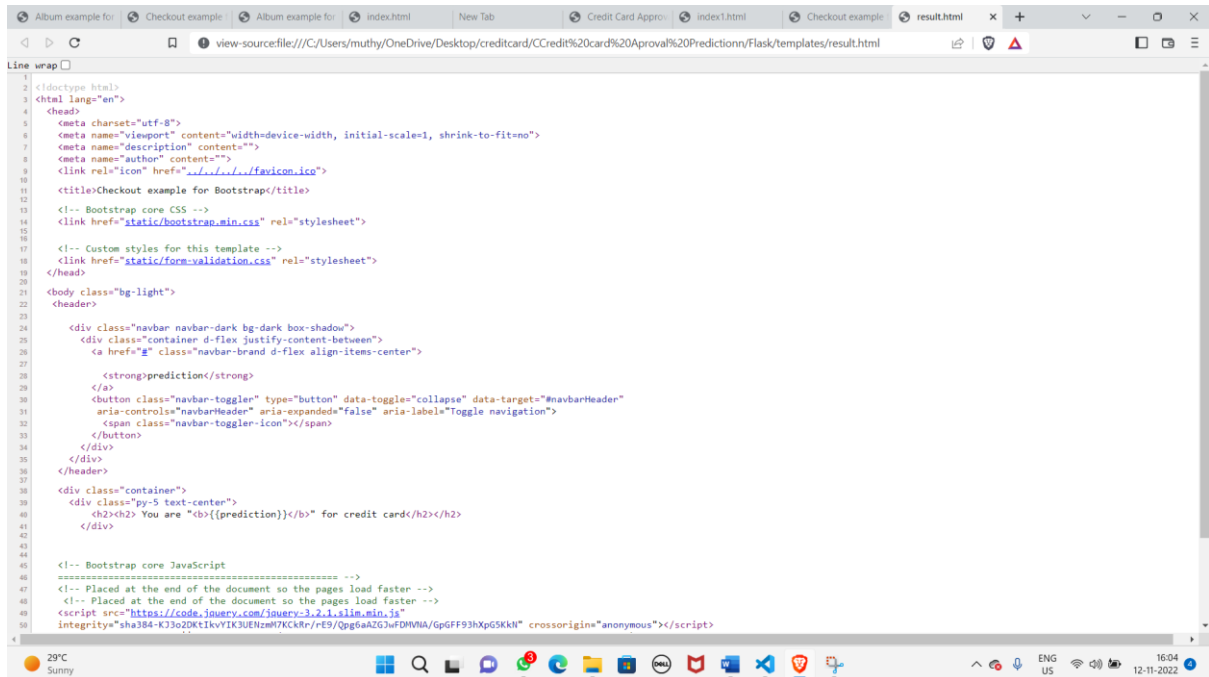
51
52 </header>
53
54 <main role="main">
55
56   <section class="jumbotron text-center">
57     <div class="container">
58       <h1 class="jumbotron-heading">Credit Card Approval Prediction </h1>
59       <p>
60         <a href="#" class="btn btn-secondary my-2">Home</a>
61
62         <input type="submit" class="btn btn-primary my-2" value="Prediction">
63       </p>
64     </div>
65   </section>
66
67   <div class="jumbotron text-center">
68     <h1>About Project</h1>
69     <p>This Machine Learning Model is responsible for calculating credit card is going to approved
70       or not based data like, type of Education, property, job profile, gender, credit score and repayment behaviour etc. It
71       is taking these data and based on previous customers behaviour this model is analyzing the applicant is illegible for
72       taking credit card or not.
73     </p>
74   </div>
75
76 </main>
77
78 <footer class="text-muted">
79   <div class="container">
80     <div class="float-right">
81       <a href="#">Back to top</a>
82     </div>
83   </div>
84 </footer>
85
86 </body>
87 </html>
88
89 <!-- Bootstrap core JavaScript
90 =====>
91 <!-- Placed at the end of the document so the pages load faster -->
92 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
93   integrity="sha384-KJ3o2DKtrIYkZUEnZm7KcRr/rE9/Qp6aAZGJwFDPVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
94 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
95   integrity="sha384-KJ3o2DKtrIYkZUEnZm7KcRr/rE9/Qp6aAZGJwFDPVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
96 <script src="static/js/vendor/popper.min.js"></script>
97 <script src="static/js/vendor/bootstrap.min.js"></script>
98 <script src="static/js/vendor/holder.min.js"></script>
99 </script>
100 </body>
101 </html>
```

INDEX1.html

```
Line wrap
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6   <meta name="description" content="">
7   <meta name="author" content="">
8   <link rel="icon" href="static/favicon.ico">
9   <title>Credit Card Approval Prediction</title>
10
11   <!-- Bootstrap core CSS -->
12   <link href="static/bootstrap.min.css" rel="stylesheet">
13
14   <!-- Custom styles for this template -->
15   <link href="static/form-validation.css" rel="stylesheet">
16 </head>
17 <body class="bg-light">
18 <header>
19
20   <div class="navbar navbar-dark bg-dark box-shadow">
21     <div class="container d-flex justify-content-between">
22       <a href="#" class="navbar-brand d-flex align-items-center">
23
24         <strong>CreditCard Prediction</strong>
25       </a>
26       <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarHeader"
27         aria-controls="navbarHeader" aria-expanded="false" aria-label="Toggle navigation">
28         <span class="navbar-toggler-icon"></span>
29       </button>
30     </div>
31   </div>
32 </header>
33
34 <div class="container">
35   <div class="py-5 text-center">
36     <h2>Credit Card Approval Prediction</h2>
37   </div>
38   <div class="row">
39
40     <div class="col-md-8 order-md-1">
41
42       <form action="/predict" method="POST" class="needs-validation" novalidate>
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
Album example for Boots: Checkout example for Boots: Album example for Boots: index.html New Tab Credit Card Approval Prediction index1.html
view-source:file:///C:/Users/muthy/OneDrive/Desktop/creditcard/CCredit%20card%20Approval%20Predictionn/Flask/templates/index1.html
49
50
51 <div class="row">
52   <div class="col-md-5 mb-3">
53     <label for="CODE_GENDER">GENDER</label>
54     <select class="custom-select d-block w-100" id="CODE_GENDER" name="CODE_GENDER" required>
55
56       <option>Select the GENDER</option>
57       <option value="0">FEMALE</option>
58       <option value="1">MALE</option>
59
60     </select>
61   </div>
62   <div class="col-md-4 mb-3">
63     <label for="FLAG_OWN_CAR">OWN CAR OR NOT</label>
64     <select class="custom-select d-block w-100" id="CountryCode" name="CountryCode" required>
65
66       <option value="">Select the YES/NO</option>
67       <option value="0">NO</option>
68       <option value="1">YES</option>
69
70     </select>
71   </div>
72   <div class="col-md-4 mb-3">
73     <label for="FLAG_OWN_REALTY">OWN REALSTATE</label>
74     <select class="custom-select d-block w-100" id="FLAG_OWN_REALTY" name="FLAG_OWN_REALTY" required>
75
76       <option value="">SELECT YES/NO</option>
77       <option value="0">NO</option>
78       <option value="1">YES</option>
79
80     </select>
81   </div>
82   <div class="col-md-4 mb-3">
83     <label for="AMT_INCOME_TOTAL">TOTAL ANNUAL INCOME</label>
84     <input type="number" class="custom-select d-block w-100" id="AMT_INCOME_TOTAL" name="AMT_INCOME_TOTAL" required>
85
86   </select>
87   </div>
88   <div class="col-md-4 mb-3">
89     <label for="NAME_INCOME_TYPE">TYPE OF INCOME</label>
90     <select class="custom-select d-block w-100" id="NAME_INCOME_TYPE" name="NAME_INCOME_TYPE" required>
91
92
93
94
95
96
97
98
99
100
```


RESULT.html



```
1
2 <!doctype html>
3 <html lang="en">
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7   <meta name="description" content="">
8   <meta name="author" content="">
9   <link rel="icon" href="...../favicon.ico">
10
11   <title>Checkout example for Bootstrap</title>
12
13   <!-- Bootstrap core CSS -->
14   <link href="static/bootstrap.min.css" rel="stylesheet">
15
16   <!-- Custom styles for this template -->
17   <link href="static/form-validation.css" rel="stylesheet">
18 </head>
19
20 <body class="bg-light">
21 <header>
22
23   <div class="navbar navbar-dark bg-dark box-shadow">
24     <div class="container d-flex justify-content-between">
25       <a href="#" class="navbar-brand d-flex align-items-center">
26
27         <strong>prediction</strong>
28       </a>
29       <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarHeader"
30         aria-controls="navbarHeader" aria-expanded="false" aria-label="Toggle navigation">
31         <span class="navbar-toggler-icon"></span>
32       </button>
33     </div>
34   </div>
35 </header>
36
37 <div class="container">
38   <div class="py-5 text-center">
39     <h2><h2> You are "cb{<prediction>}</b>" for credit card</h2></h2>
40   </div>
41
42
43
44
45 <!-- Bootstrap core JavaScript
46 =====>
47 <!-- Placed at the end of the document so the pages load faster -->
48 <!-- Placed at the end of the document so the pages load faster -->
49 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
50 integrity="sha384-K3o2DKtIkvYIK3UEINZm77KCKR/rE9/Qp6AAZGhFDHWA/Gp6FF93hXpG5KkN" crossorigin="anonymous"></script>
```