Placement Prediction App using Flask

Last Updated : 08 Dec, 2023

Machine learning is a widely employed method for making predictions. Numerous algorithms are accessible in different libraries for predictive tasks. In this article, we'll construct a placement prediction model using Random Forest Classifier with historical data and later we will store that model to .pkl file to integrate it with our **Flask app using** Python.

Placement Prediction in ML using Flask

The motive behind this project is to predict the chance or probability of students getting placed in campus placement drives. Our motive would be to create a full working application that would make predictions. For all this, we require a basic knowledge of Flask, HTML, and Machine Learning. We've taken historical campus placement data and constructed a robust model that can forecast your chances of getting placed.

Topics Covered

- Virtual Environment Setup
- ML model for predictions
- Integration with Flask
- Deployment on the local host

Steps for Creating a Placement Prediction App

Step 1: Create a virtual environment

Pycharm. To create a virtual Environment write the following code in the terminal.

- *python -m venv <enviroment name>*
- *<enviroment name>\Scripts\activate*

```
\PycharmProjects\plac> python -m venv myenv
\PycharmProjects\plac> myenv\Scripts\activate
```

ing1: to create an environment

Step 2: Create a Predictive model

placement.py

In this code, a dataset ('Placement_Data_Full_Class.csv') is loaded using pandas, and Label Encoding is applied to convert categorical variables (gender, ssc_b, hsc_b, hsc_s, degree_t, workex, specialisation, status) into numerical format. Missing values in the 'salary' column are filled with the median. The dataset is split into features (X) and target variable (y). A RandomForestClassifier with 26 estimators is trained on the training set, and the resulting model is saved as 'placement.pkl' using the pickle module.

Dataset: Dataset link

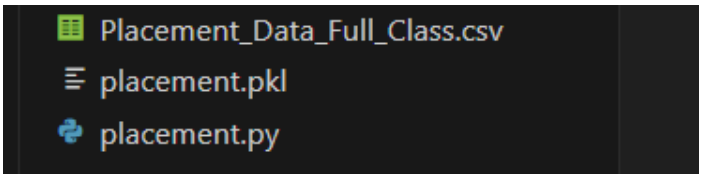
Python3

```
import pandas as pd
import numpy as np
import pickle

df=pd.read_csv('Placement_Data_Full_Class.csv')

from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
df['gender']=encoder.fit_transform(df['gender'])
df['ssc_b']=encoder.fit_transform(df['ssc_b'])
df['hsc_b']=encoder.fit_transform(df['hsc_b'])
df['hsc_s']=encoder.fit_transform(df['hsc_s'])
df['degree_t']=encoder.fit_transform(df['degree_t'])
df['workex']=encoder.fit_transform(df['workex'])
df['specialisation']=encoder.fit_transform(df['specialisat
ion'])
df['status']=encoder.fit_transform(df['status'])
```

```
df['salary'] = df['salary'].fillna(df['salary'].median())

X=df.drop('status',axis=1)
y=df['status']

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.3,random_state=101)

from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=26)
rf.fit(X_train,y_train)

pickle.dump(rf,open('placement.pkl','wb'))
```

Now, paste the above code into your .py file and run it using python <Name of File>.py in the terminal. After running the code, a .pkl file would be generated.

Output



img2: .pkl file generated

Step 3: Setting up GUI

front.html: This code creates a Flask web application that lets users input data. When users submit the form, the data is used to make predictions with a pre-trained machine learning model. The results are then displayed on a webpage. The 'placement.pkl' model is loaded using the 'pickle' library.

HTML

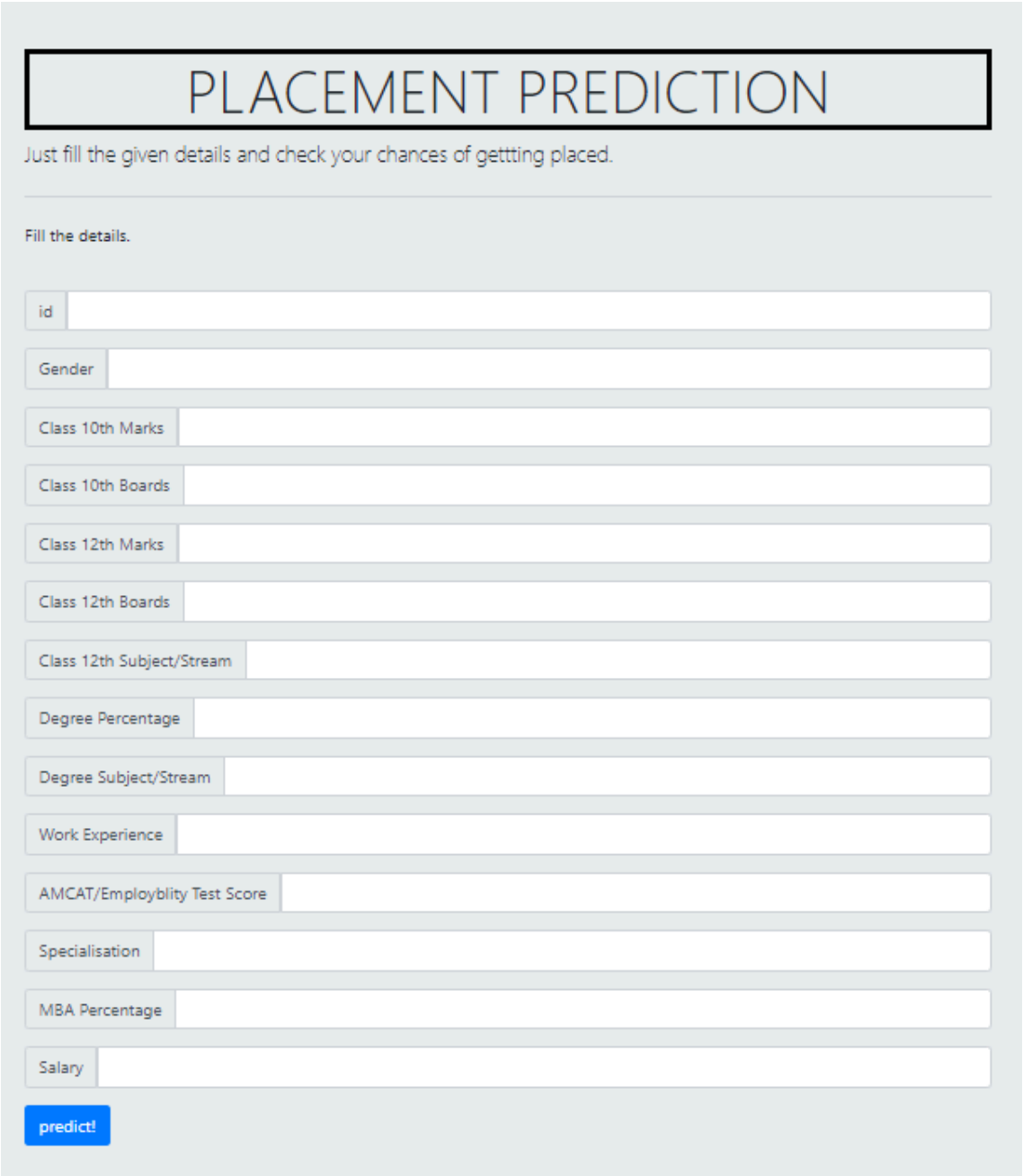
```
<!DOCTYPE html>
<html>
<head>
    <title>Placement Prediction</title>
    <style type="text/css">
         h1{ font-size: 60px;
             text-align: center;
             color: #e32037;
             border: 5px solid black;
             font-family: 'Impact';
             }

         body{
         background-size:cover;
   }

        legend{font-size: 20px;
```

```
                text-align: center;}


        .note{color: #FF0000;}


    </style>

 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0
 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrit
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/pop
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/boots
</head>
<body>

    <div class="jumbotron">
        <h1 class="display-4">PLACEMENT PREDICTION</h1>
        <p class="lead">Just fill the given details and check your chanc
        <hr class="my-4">
        <p>Fill the details.</p>
        <p class="lead">
            <form method="POST", action="{{url_for('home')}}">
                <fieldset>
                <br>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number" name="id" class="form-control'
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input  type ="number"  name="gender" min="0" max="1
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number"  name="marks1" min="0" max="10
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number" name="boards1" min="0" max="1'
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
```

```
                    <input type ="number"  name="marks2" min="0" max="10
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number" name="boards2" min="0" max="1'
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number"  name="strm" min="0" max="2" c
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number"  name="deg_p" min="0" max="100
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number"  name="deg_s" min="0" max="2"
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number"  name="wrx" min="0" max="1" cl
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number"  name="amcat" min="0" max="100
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number" name="sp" min="0" max="1" clas
                </div>

                <div class="input-group mb-3">
                    <div class="input-group-prepend">
                      <span class="input-group-text" id="inputGroup-sizi
                    </div>
                    <input type ="number" name="mba_p" min="0" max="100'
```

```
                    </div>
133
134                 <div class="input-group mb-3">
135                     <div class="input-group-prepend">
136                       <span class="input-group-text" id="inputGroup-sizi
137                     </div>
138                     <input type ="number" name="sal"class="form-control'
139                 </div>
140
141
142                 <input type="submit" value ="predict!" class="btn btn-pr
143                 <a href="{{ url_for('home')}}"></a>
144                 </fieldset>
145                 </form>
146
147                 <fieldset><legend>Note</legend>
148
149                   1. For Gender: press 0 for female and 1 for male<br>
150                   2. For Class 10th and 12th Boards: 0 for central and 1
151                   3. For Class 12th Subject/Stream: 0 for Commerce 1 for
152                   4. For degree stream: 0 for Commerce, 1 for other and
153                   5. For work Experience: 0 for No and 1 for Yes<br>
154                   6. MBA Specialisation: 0 for (Finance and Marketing) a
155
156                 </fieldset>
157
158           </p>
159         </div>
160
161 </body>
162 </html>
```

Output



home.html: The page displays a result message, which can be either "Congratulations!!!" in green if the prediction is positive or "Sorry!" in red if the prediction is negative. This code also converts the numeric values(input values) into human readable text(e.g., '0' for gender becomes 'Female').

HTML

```
<html>
<head>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.
 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrit
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/pop
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/boots
</head>
<body bgcolor=#1ff0d0>

    <center>

        <div class="jumbotron">
```

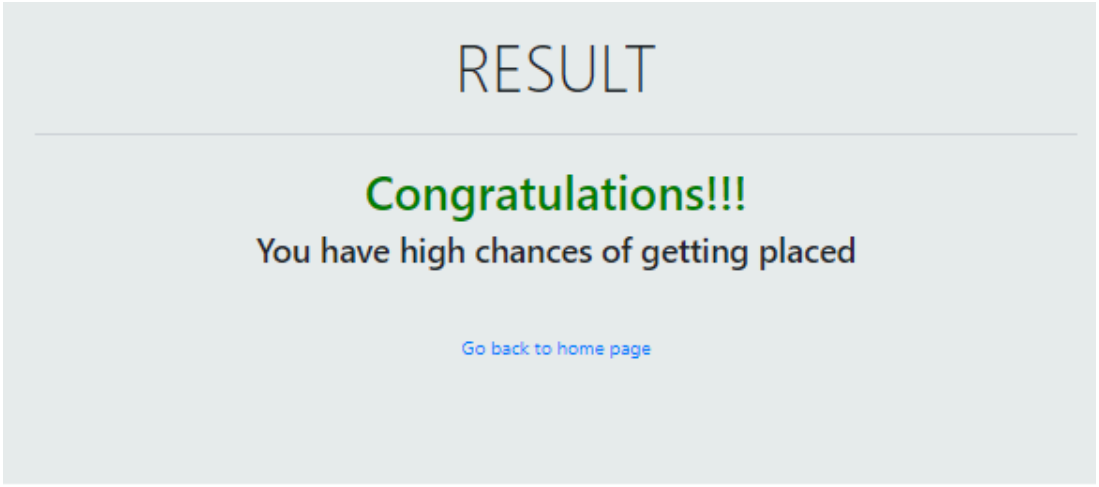
```
            <h1 class="display-4">RESULT</h1>
            <hr class="my-4">
            {%if data == 0%}

           <h1 style = "Color : red">Sorry!</h1>
           <h3>You have less chances of getting placed</h3>


           {%else%}
           <h1 style ="Color : GREEN">Congratulations!!!</h1>
           <h3>You have high chances of getting placed</h3>


           {%endif%}

        <br><br>
        <a href='/'>Go back to home page</a>
        <br><br>

            <p class="lead">

            </p>
          </div>

    </center>


    <table class="table">
        <thead class="thead-dark">
          <tr>
            <th scope="col">Index</th>
            <th scope="col">Information</th>
            <th scope="col">Value</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <th scope="row">1</th>
            <td>ID</td>
            <td>{{ data1 }}</td>
        </tr>
        <tr>
            <th scope="row">2</th>
            <td>Gender</td>
            <td>
              {% if data2 =='0' %}
                {% set data2_new = 'Female' %}
              {% elif data2 == '1' %}
                {% set data2_new = 'Male' %}
              {% else %}
                {% set data2_new = 'Not a valid response' %}
              {% endif %}

              {{ data2_new }}
              </td>
        </tr>

```

```
                <tr>
                  <th scope="row">3</th>
                  <td>Class 10th Marks</td>
                  <td>{{data3}}</td>

                </tr>

                <tr>
                  <th scope="row">4</th>
                  <td>Class 10th Boards</td>
                  <td>
                    {% if data4 =='0' %}
                      {% set data4_new = 'Central' %}
                    {% elif data4 == '1' %}
                      {% set data4_new = 'Other' %}
                    {% else %}
                      {% set data4_new = 'Not a valid response' %}
                    {% endif %}

                    {{ data4_new }}

                  </td>

                </tr>
                <tr>
                  <th scope="row">5</th>
                  <td>Class 12th Marks</td>
                  <td>{{data5}}</td>

                </tr>
                <tr>
                  <th scope="row">6</th>
                  <td>Class 12th Boards</td>
                  <td>
                    {% if data6 =='0' %}
                      {% set data6_new = 'Central' %}
                    {% elif data6 == '1' %}
                      {% set data6_new = 'Other' %}
                    {% else %}
                      {% set data6_new = 'Not a valid response' %}
                    {% endif %}

                    {{ data6_new }}

                  </td>

                </tr>

                <tr>
                  <th scope="row">7</th>
                  <td>Class 12th Subject/Stream</td>
                  <td>
                  {% if data7 =='0' %}
                    {% set data7_new = 'Commerce' %}
                  {% elif data7 == '1' %}
                    {% set data7_new = 'Science' %}
                  {% elif data7 == '2' %}
```

```
                {% set data7_new = 'Arts' %}
              {% else %}
                {% set data7_new = 'Not a valid response' %}
              {% endif %}

              {{ data7_new }}

              </td>

            </tr>
            <tr>
              <th scope="row">8</th>
              <td>Degree Percentage</td>
              <td>{{data8}}</td>

            </tr>
            <tr>
              <th scope="row">9</th>
              <td>Degree Subject/Stream</td>
              <td>

              {% if data9 =='0' %}
                {% set data9_new = 'Commerce and Management' %}
              {% elif data9 == '1' %}
                {% set data9_new = 'Other' %}
              {% elif data9 == '2' %}
                {% set data9_new = 'Science and Technology' %}
              {% else %}
                {% set data9_new = 'Not a valid response' %}
              {% endif %}

              {{ data9_new }}


              </td>

            </tr>

            <tr>
              <th scope="row">10</th>
              <td>Work Experience</td>
              <td>
              {% if data10 =='0' %}
                {% set data10_new = 'No' %}
              {% elif data10 == '1' %}
                {% set data10_new = 'Yes' %}
              {% else %}
                {% set data10_new = 'Not a valid response' %}
              {% endif %}

              {{ data10_new }}

              </td>

            </tr>
            <tr>
              <th scope="row">11</th>
```

```
                <td>AMCAT/Employblity Test Score</td>
185             <td>{{data11}}</td>
186
187           </tr>
188           <tr>
189             <th scope="row">12</th>
190             <td>MBA Specialisation</td>
191             <td>
192             {% if data12 =='0' %}
193               {% set data12_new = 'Finance and Marketing' %}
194             {% elif data12 == '1' %}
195               {% set data12_new = 'HR and Marketing' %}
196             {% else %}
197               {% set data12_new = 'Not a valid response' %}
198             {% endif %}
199
200             {{data12_new}}
201             </td>
202
203           </tr>
204           <tr>
205             <th scope="row">13</th>
206             <td>MBA Percentage</td>
207             <td>{{data13}}</td>
208
209           </tr>
210
211           <tr>
212             <th scope="row">14</th>
213             <td>Salary</td>
214             <td>{{data14}}</td>
215
216           </tr>
217
218
219         </tbody>
220       </table>
221
222
223
224
225 </body>
226
227 </html>
```

Output



Index	Information	Value
1	ID	1234
2	Gender	Male
3	Class 10th Marks	92
4	Class 10th Boards	Other
5	Class 12th Marks	85
6	Class 12th Boards	Central
7	Class 12th Subject/Stream	Science
8	Degree Percentage	75
9	Degree Subject/Stream	Science and Technology
10	Work Experience	Yes
11	AMCAT/Employblity Test Score	86
12	MBA Specialisation	HR and Marketing
13	MBA Percentage	0
14	Salary	25000

Step 4: App Code

app.py: This app.py creates a Flask web application to serve a machine learning model. Users input data through a form, and the code passes this data to the model to make predictions. The results are then displayed on a web page. The pickle library is used to load the pre-trained model, and Flask handles the web interface and routing.

Python3

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


model = pickle.load(open('placement.pkl', 'rb'))

```

```
app = Flask(__name__)


@app.route('/', methods=['GET', 'POST'])
def man():
    return render_template('front.html')


@app.route('/predict', methods=['GET', 'POST'])
def home():
    data1 = request.form['id']
    data2 = request.form['gender']
    data3 = request.form['marks1']
    data4 = request.form['boards1']
    data5 = request.form['marks2']
    data6 = request.form['boards2']
    data7 = request.form['strm']
    data8 = request.form['deg_p']
    data9 = request.form['deg_s']
    data10 = request.form['wrx']
    data11 = request.form['amcat']
    data12 = request.form['sp']
    data13 = request.form['mba_p']
    data14 = request.form['sal']

    arr = np.array([[data1, data2, data3, data4, data5, data6, data7, data8,
                     data9, data10, data11, data12, data13, data14]])
    pred = model.predict(arr)
    return render_template('home.html', data=pred, data1=data1, data2=data2, data3=data
                           data4=data4, data5=data5, data6=data6, data7=data7, data8=da
                           data9=data9, data10=data10, data11=data11, data12=data12,
                           data13=data13, data14=data14)


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

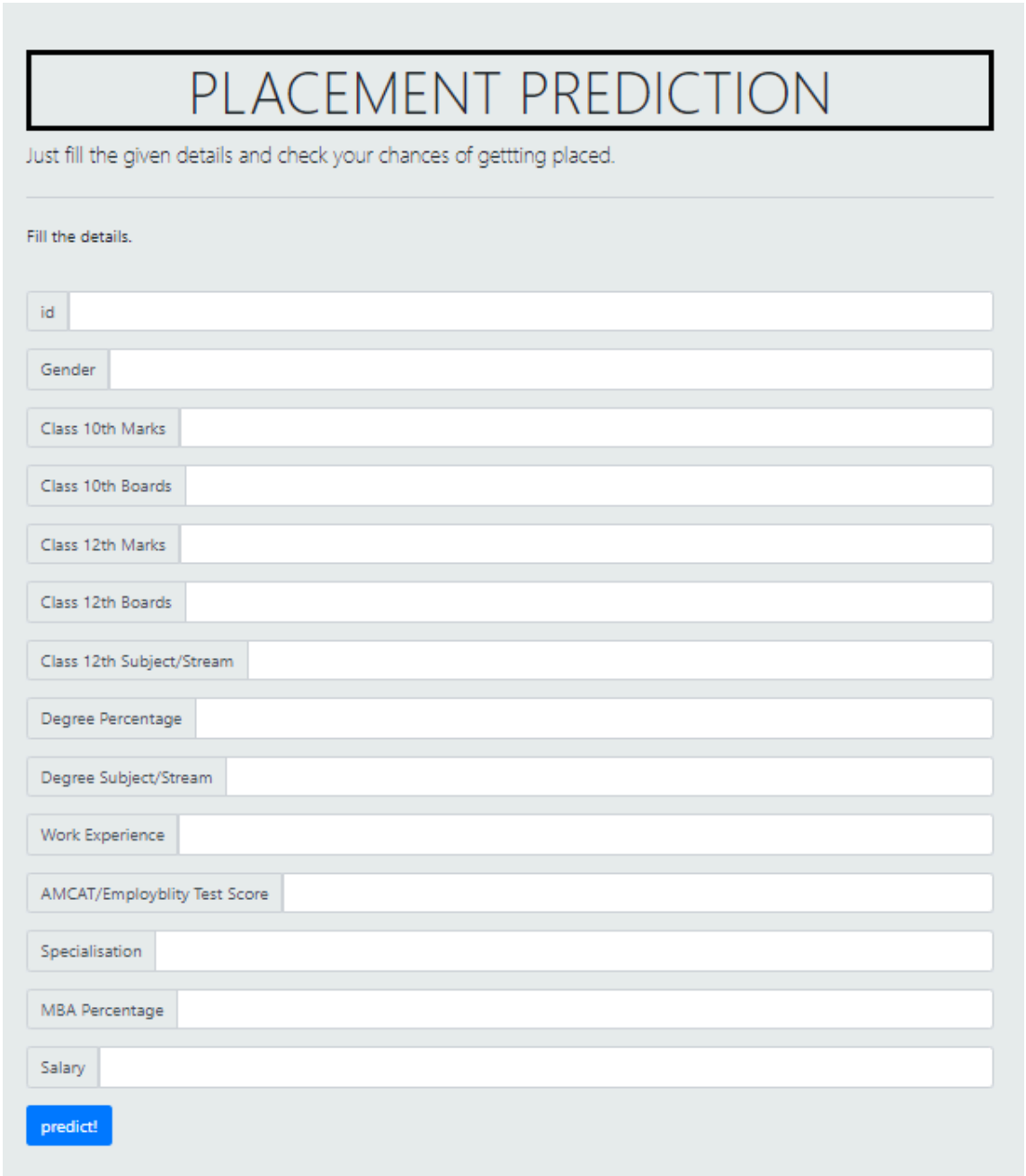
Step 5: Running the app on local host.

```
harmProjects\plac> python app.py
```

Just write "python app.py" on the terminal and this would be generated.

```
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 662-780-665
```

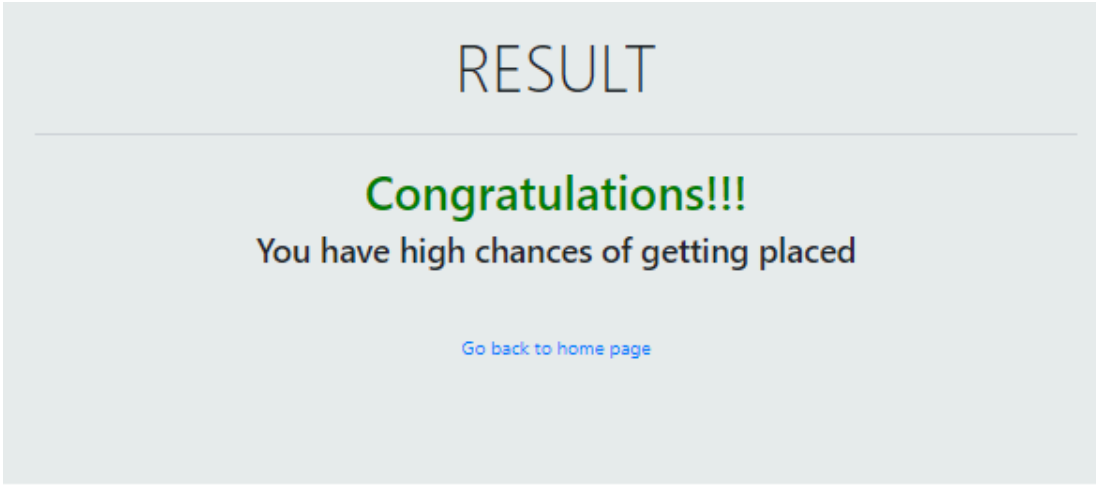
After that just click on the "http://127.0.0.1:5000" and you would be redirected to a webpage, which would the homepage of the application.



app running on local host

After filling all the information click on predict.

Output



Index	Information	Value
1	ID	1234
2	Gender	Male
3	Class 10th Marks	92
4	Class 10th Boards	Other
5	Class 12th Marks	85
6	Class 12th Boards	Central
7	Class 12th Subject/Stream	Science
8	Degree Percentage	75
9	Degree Subject/Stream	Science and Technology
10	Work Experience	Yes
11	AMCAT/Employblity Test Score	86
12	MBA Specialisation	HR and Marketing
13	MBA Percentage	0
14	Salary	25000

Final Prediction

And your model is ready to predict.

Video Demonstration

Previous Article

Wikipedia search app using Flask Framework - Python

Next Article

Similar Reads

Documenting Flask Endpoint using Flask-Autodoc

Documentation of endpoints is an essential task in web development and being able to apply it in different frameworks is always a utility. This article discusses how endpoints ...

4 min read

Minify HTML in Flask using Flask-Minify

Flask offers HTML rendering as output, it is usually desired that the output HTML should be concise and it serves the purpose as well. In this article, we would display minificatio...

12 min read

How to use Flask-Session in Python Flask ?

Flask Session - Flask-Session is an extension for Flask that supports Server-side Session to your application.The Session is the time between the client logs in to the server and...

4 min read

How to Integrate Flask-Admin and Flask-Login

In order to merge the admin and login pages, we can utilize a short form or any other login method that only requires the username and password. This is known as...

8 min read

Flask URL Helper Function - Flask url_for()

In this article, we are going to learn about the flask url_for() function of the flask URL helper in Python. Flask is a straightforward, speedy, scalable library, used for building,...

11 min read

Django - Machine Learning Placement Prediction Project

In this article, we will create Placement Prediction Project using the Django framework using Python. In this project, we are using logistic regression algorithm of Python...

7 min read

Create a Weather app using Flask | Python

Prerequisite : Flask installation Flask is a lightweight framework written in Python. It is lightweight because it does not require particular tools or libraries and allow rapid web...

2 min read

Todo list app using Flask | Python

There are many frameworks that allow building your webpage using Python, like Django, flask, etc. Flask is a web application framework written in Python. Flask is based on...

3 min read

How to Run Python Flask App Online using Ngrok?

Python Flask is a popular web framework for developing web applications, APIs, etc. Running flask apps on the local machine is very simple, but when it comes to sharing th...

2 min read

Create a Real Time Currency Converter app using Flask | Python

Prerequisite: installation of PythonWhat is Flask in Python ? Flask is a popular and lightweight Python web framework, meaning it is a third-party Python library used for...

8 min read

Article Tags :
Python
Python Flask
Machine Learning Projects
Flask Projects
Practice Tags :
python

Interview Questions

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects