

Name : Bilal Yildiz

Batch code : LISUM02

Submission date : 08/08/2021

Submitted to : GitHub

This is the data I am working on ,its pretty simple ,We are predicting the salary of employees based on their experience, test score and interview score .

Index	experience	test_score	interview	salary
0	nan	8	9	50000
1	nan	8	6	45000
2	five	6	7	60000
3	two	10	10	65000
4	seven	9	6	70000
5	three	7	10	62000
6	ten	nan	7	72000
7	eleven	7	8	80000

I used Linear regression ML model

You can see the comments to explain what I did in the module

```
model.py x
C: > Users > MONSTER > Desktop > Week4 > project > model.py
1  # Importing the libraries
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import pandas as pd
5  import pickle
6
7  dataset = pd.read_csv('hiring.csv')
8
9  dataset['experience'].fillna(0, inplace=True)
10
11 dataset['test_score'].fillna(dataset['test_score'].mean(), inplace=True)
12
13 X = dataset.iloc[:, :3]
14
15 #Converting words to integer values
16 def convert_to_int(word):
17     word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
18                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
19     return word_dict[word]
20
21 X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))
22
23 y = dataset.iloc[:, -1]
24
25 #Splitting Training and Test Set
26 #Since we have a very small dataset, we will train our model with all available data.
27
28 from sklearn.linear_model import LinearRegression
29 regressor = LinearRegression()
30
31 #Fitting model with training data
32 regressor.fit(X, y)
33
34 # Saving model to disk
35 pickle.dump(regressor, open('model.pkl','wb'))
36
37 # Loading model to compare the results
38 model = pickle.load(open('model.pkl','rb'))
39 print(model.predict([[2, 9, 6]]))
```

I made a .pkl file for my model

```
pickle.dump(regressor, open('model.pkl','wb'))
```

```
# Saving model to disk
pickle.dump(regressor, open('model.pkl','wb'))

# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
print(model.predict([[2, 9, 6]]))
```

I have made HTML File to render my interface

```
index.html X
C: > Users > MONSTER > Desktop > Week4 > project > templates > index.html > html
1  <!DOCTYPE html>
2  <html>
3  <!--From https://codepen.io/frytyler/pen/E6dtg-->
4
5  <head>
6    <meta charset="UTF-8">
7    <title>ML API</title>
8    <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
9    <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
10   <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
11   <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
12 </head>
13
14 <body id="home">
15   <div class="login">
16     <h1>Predict Salary Analysis</h1>
17     <!-- Main Input For Receiving Query to our ML -->
18     <form action="{{ url_for('predict')}}" method="post">
19       <input id="test" type="text" name="experience" placeholder="Experience" required="required" />
20       <br><br> <input id="test" type="text" name="test_score" placeholder="Test Score" required="required" />
21       <br><br> <input id="test" type="text" name="interview_score" placeholder="Interview Score" required="required" />
22       <br><br><br> <button id="but" type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
23     </form>
24     <br>
25     <br> {{ prediction_text }}
26   </div>
27 </body>
28
29 </html>
```


I used `pick` `pickle.load` to load my model

```
C: > Users > MONSTER > Desktop > Week4 > project > app.py
1  import numpy as np
2  from flask import Flask, request, jsonify, render_template
3  import pickle
4
5  app = Flask(__name__)
6  model = pickle.load(open('model.pkl', 'rb'))
7  |
8  @app.route('/')
9  def home():
10 |     return render_template('index.html')
11
```

We used this function for
rendering results on HTML GUI

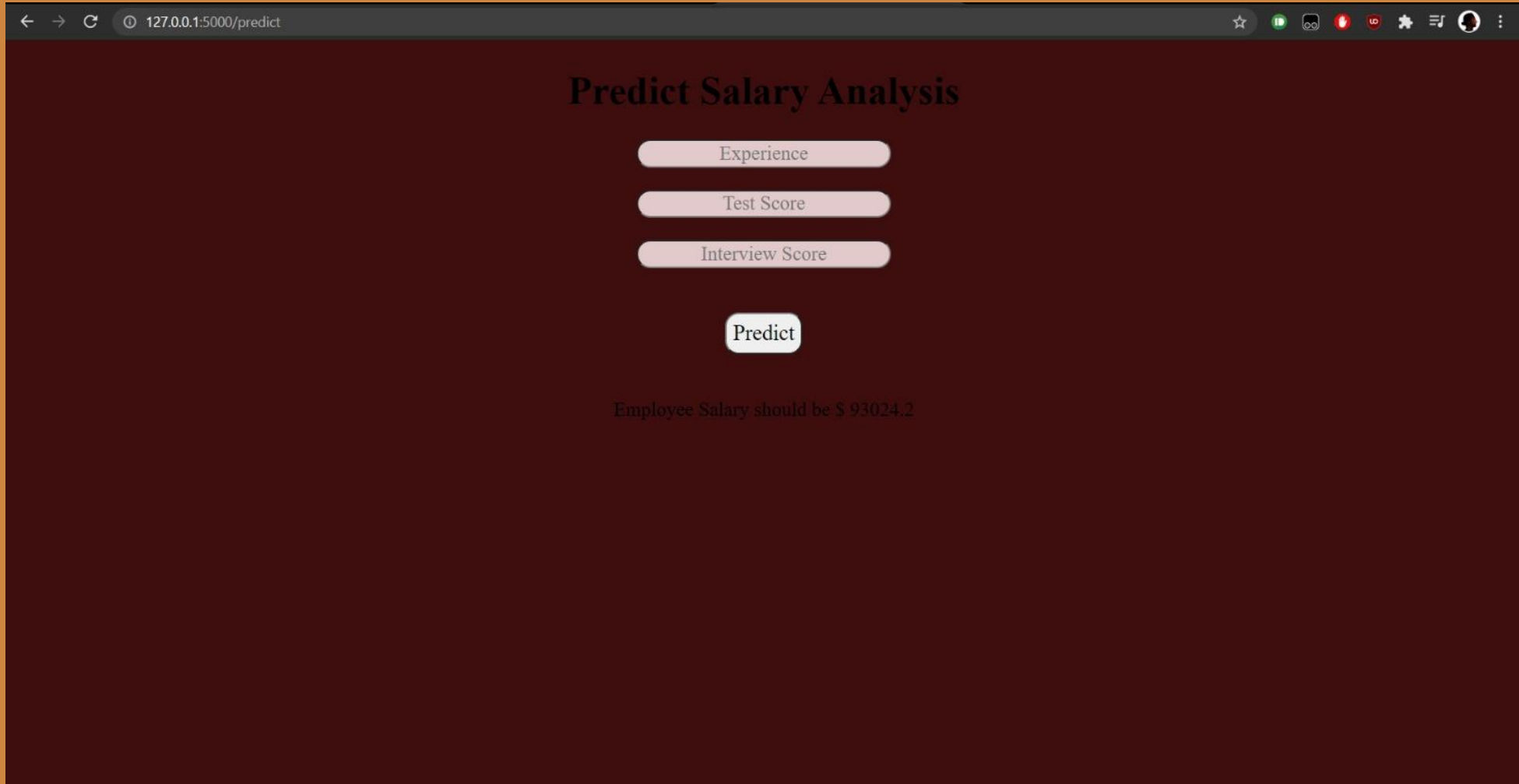
```
12 @app.route('/predict',methods=['POST'])
13 def predict():
14
15     int_features = [int(x) for x in request.form.values()]
16     final_features = [np.array(int_features)]
17     prediction = model.predict(final_features)
18
19     output = round(prediction[0], 2)
20
21     return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))
22
```

Now our program is ready run

C: > Users > MONSTER > Desktop > Week4 > project >  app.py

```
1  import numpy as np
2  from flask import Flask, request, jsonify, render_template
3  import pickle
4
5  app = Flask(__name__)
6  model = pickle.load(open('model.pkl', 'rb'))
7
8  @app.route('/')
9  def home():
10     return render_template('index.html')
11
12  @app.route('/predict', methods=['POST'])
13  def predict():
14
15     int_features = [int(x) for x in request.form.values()]
16     final_features = [np.array(int_features)]
17     prediction = model.predict(final_features)
18
19     output = round(prediction[0], 2)
20
21     return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))
22
23
24  if __name__ == "__main__":
25     app.run(debug=True)
```


After running our program we can see the result in browser



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/predict". The page has a dark blue background and features the title "Predict Salary Analysis" in a large, bold, white font. Below the title, there are three light blue input fields with rounded corners, labeled "Experience", "Test Score", and "Interview Score" in a light gray font. A white "Predict" button with rounded corners is positioned below these fields. At the bottom of the form, the text "Employee Salary should be \$ 93024.2" is displayed in a light gray font.