Name : Bilal Yıldız

Batch code : LISUM02

Submission date : 06/08/2021

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 | experienc | 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

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

```python
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle

dataset = pd.read_csv('hiring.csv')

dataset['experience'].fillna(0, inplace=True)

dataset['test_score'].fillna(dataset['test_score'].mean(), inplace=True)

X = dataset.iloc[:, :3]

#Converting words to integer values
def convert_to_int(word):
    word_dict = {'one':1, 'two':2, 'three':3, 'four':4, 'five':5, 'six':6, 'seven':7, 'eight':8,
                 'nine':9, 'ten':10, 'eleven':11, 'twelve':12, 'zero':0, 0: 0}
    return word_dict[word]

X['experience'] = X['experience'].apply(lambda x : convert_to_int(x))

y = dataset.iloc[:, -1]

#Splitting Training and Test Set
#Since we have a very small dataset, we will train our model with all availabe data.

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()

#Fitting model with trainig data
regressor.fit(X, y)

# 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 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

C: > Users > MONSTER > Desktop > Week4 > project > templates > index.html > html

```
<!DOCTYPE html>
<html>
<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
</head>

<body id="home">
  <div class="login">
    <h1>Predict Salary Analysis</h1>
    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}" method="post">
      <input id="test" type="text" name="experience" placeholder="Experience" required="required" />
      <br><br> <input id="test" type="text" name="test_score" placeholder="Test Score" required="required" />
      <br><br> <input id="test" type="text" name="interview_score" placeholder="Interview Score" required="required" />
      <br><br><br> <button id="but" type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>
    <br>
    <br> {{ prediction_text }}
  </div>
</body>

</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

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

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

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

@app.route('/predict',methods=['POST'])
def predict():

    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Employee Salary should be $ {}'.format(output))


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

After running our program we can see the result in browser