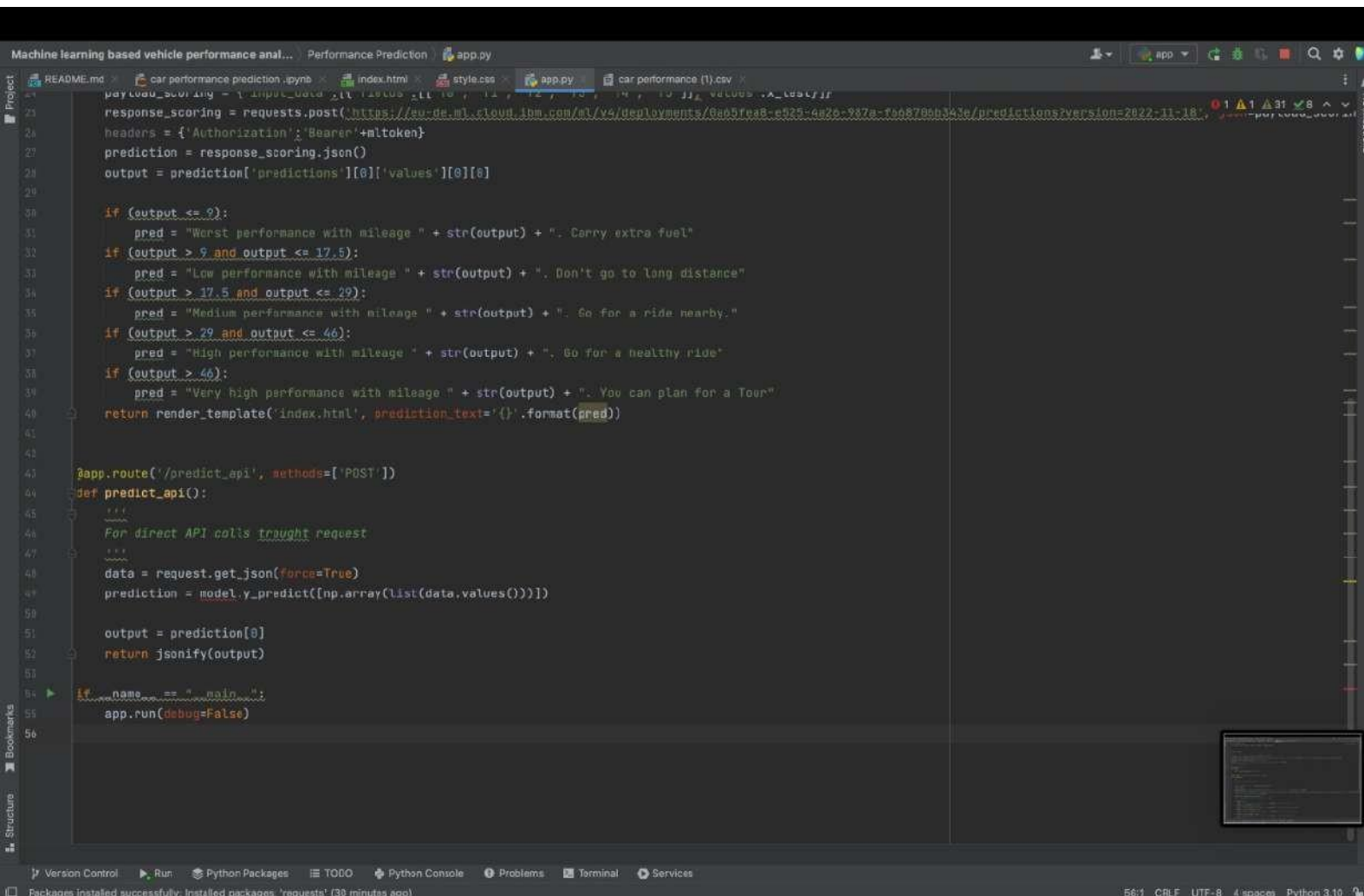


Machine Learning Based Vehicle Performance Analyzer

Team ID : PNT2022TMID31242



The image shows a screenshot of a Visual Studio Code editor window. The title bar reads "Machine learning based vehicle performance anal... | Performance Prediction | app.py". The editor has several tabs open: "README.md", "car performance prediction.py", "index.html", "style.css", "app.py", and "car performance (1).csv". The "app.py" file is the active tab and contains the following Python code:

```
payload_scoring = {'input_data': {'mileage': 12, 'engine': 14, 'displacement': 10, 'values': 1.0}}
response_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/8a65feab-e025-4a26-987a-f6b8786b343e/predictions?version=2022-11-18',
headers = {'Authorization': 'Bearer'+altoken})
prediction = response_scoring.json()
output = prediction['predictions'][0]['values'][0][0]

if (output <= 2):
    pred = "Worst performance with mileage " + str(output) + ". Carry extra fuel"
if (output > 9 and output <= 17.5):
    pred = "Low performance with mileage " + str(output) + ". Don't go to long distance"
if (output > 17.5 and output <= 29):
    pred = "Medium performance with mileage " + str(output) + ". Go for a ride nearby."
if (output > 29 and output <= 46):
    pred = "High performance with mileage " + str(output) + ". Go for a healthy ride"
if (output > 46):
    pred = "Very high performance with mileage " + str(output) + ". You can plan for a Tour"
return render_template('index.html', prediction_text='{}'.format(pred))

@app.route('/predict_api', methods=['POST'])
def predict_api():
    """
    For direct API calls through request
    """
    data = request.get_json(force=True)
    prediction = model.predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

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

The bottom status bar of VS Code shows "56:1 CRLF UTF-8 4 spaces Python 3.10". A small terminal window is visible in the bottom right corner.

```
Machine learning based vehicle performance anal... Performance Prediction app.py
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3
4
5 import requests
6
7 API_KEY = "cPFRZC9gPSGV2pFnx5yDC8we0porg88Lk8ldM8le1n9v"
8 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": "urn:ibm:params:oauth:grant-type:apikey"})
9 mltoken = token_response.json()["access_token"]
10 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer'+mltoken}
11 app = Flask(__name__)
12
13 @app.route('/')
14 def home():
15     return render_template('index.html')
16
17 @app.route('/y_predict', methods=['POST', 'GET'])
18 def y_predict():
19     """
20     For rendering results on HTML GUI
21     """
22     x_test = [[int(x) for x in request.form.values()]]
23     print(x_test)
24     payload_scoring = {"input_data": [{"fields": [{"f0", 'f1', 'f2', 'f3', 'f4', 'f5'}], "values": x_test}]}
25     response_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/0a65feb8-e525-4a26-987a-f668786b343e/predictions?version=2022-11-18', json=payload_scoring, headers=header)
26     headers = {'Authorization': 'Bearer'+mltoken}
27     prediction = response_scoring.json()
28     output = prediction['predictions'][0]['values'][0][0]
29
30     if (output <= 9):
31         pred = "Worst performance with mileage " + str(output) + ". Carry extra fuel"
32     if (output > 9 and output <= 17.5):
33         pred = "Low performance with mileage " + str(output) + ". Don't go to long distance"
34     if (output > 17.5 and output <= 29):
35         pred = "Medium performance with mileage " + str(output) + ". Go for a ride nearby."
36     if (output > 29 and output <= 46):
37         pred = "High performance with mileage " + str(output) + ". Go for a healthy ride"
38     if (output > 46):
39         pred = "Out of range"
40
41     return render_template('index.html', output=output, pred=pred)
```

56:1 CRLF UTF-8 4 spaces Python 3.10