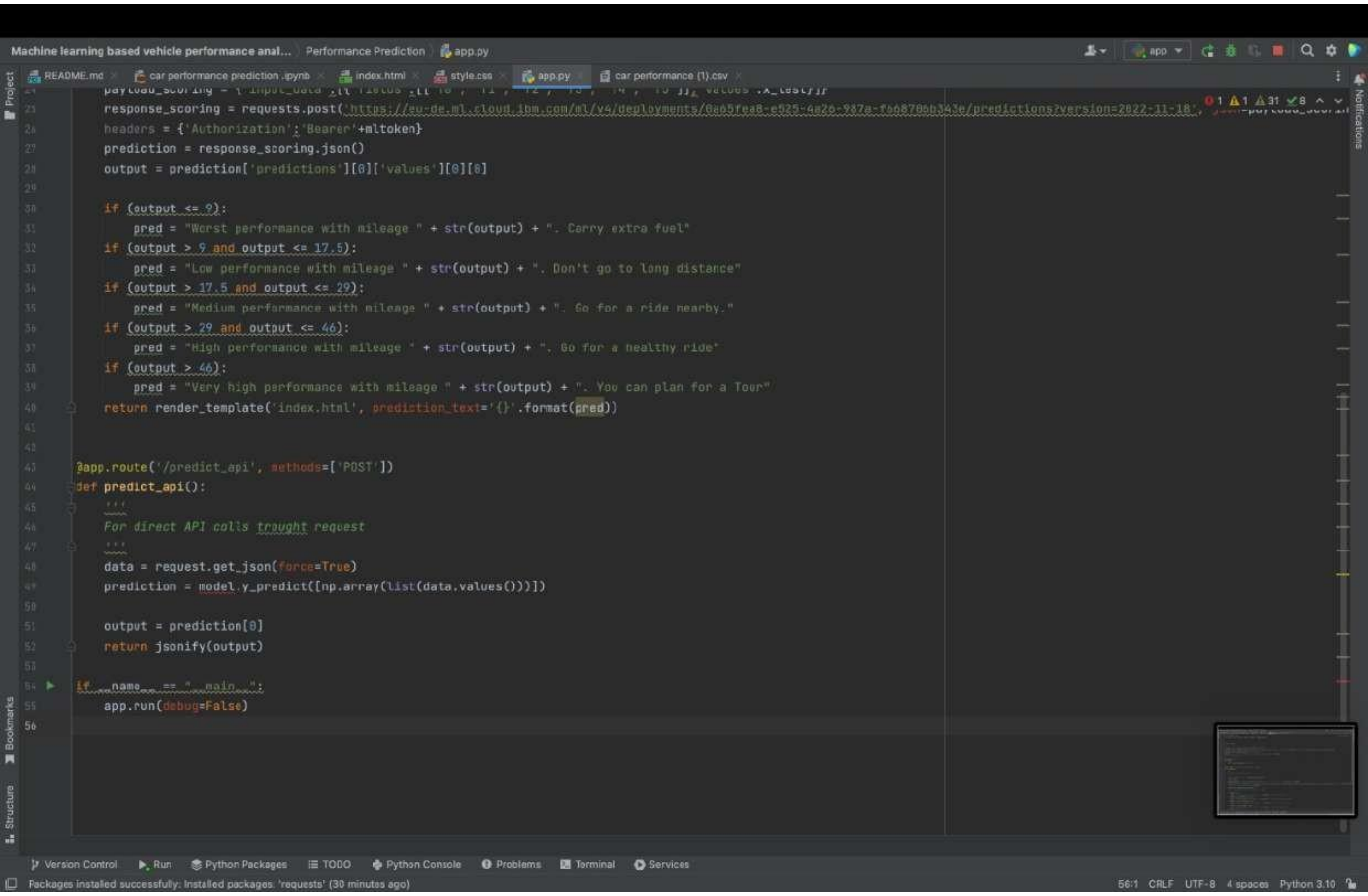
Machine Learning Based Vehicle Performance Analyzer

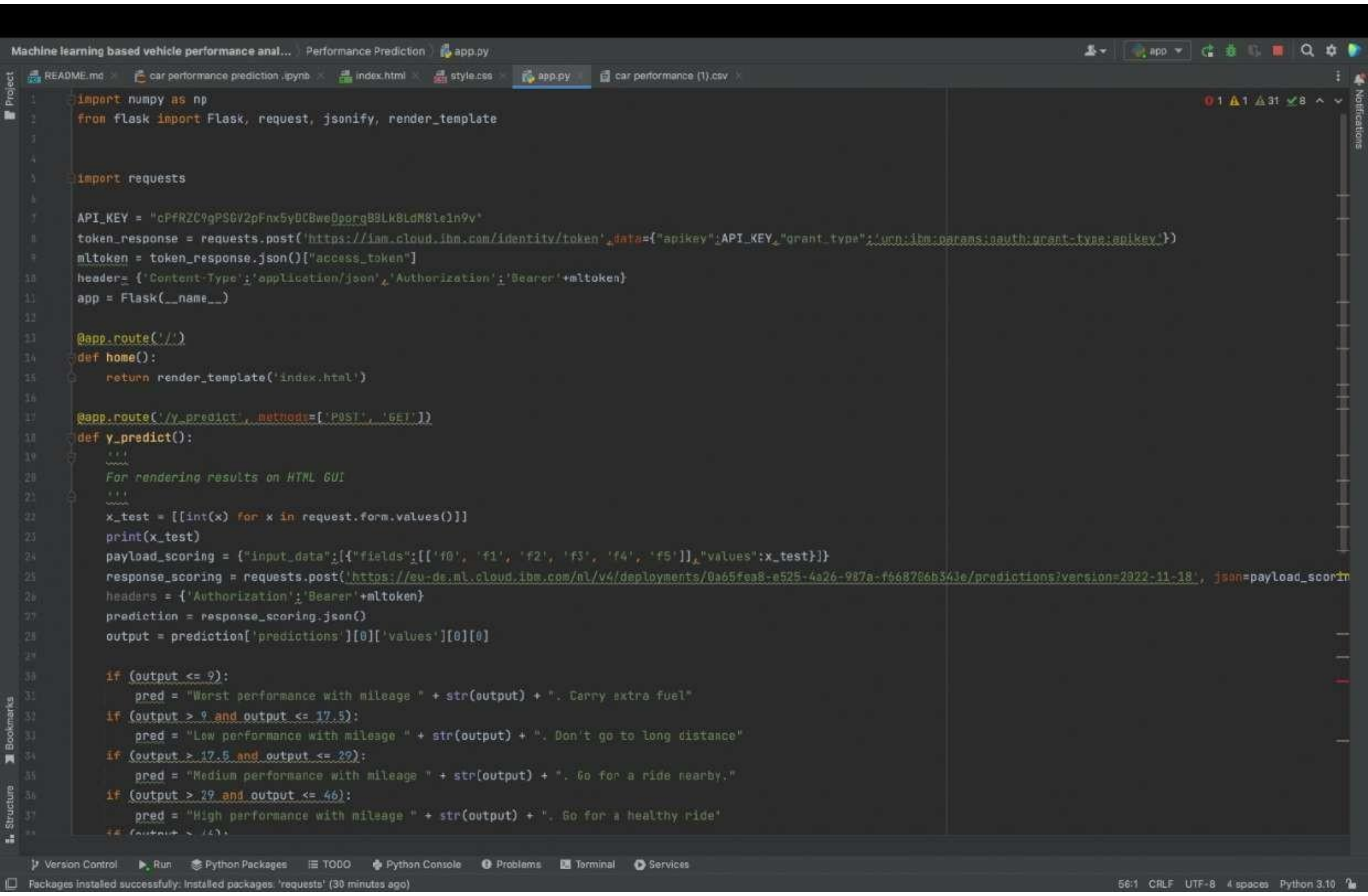
Team ID : PNT2022TMID31242



```
    payload_scoring = {"input_data": [{"fields": [['f0', 'f1', 'f2', 'f3', 'f4', 'f5']], "values": x_test}]}
    response_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/0ab5fea8-e525-4a26-987a-fb68705b343e/predictions?version=2022-11-18', json=payload_scori
    headers = {'Authorization': 'Bearer '+mltoken}
    prediction = response_scoring.json()
    output = prediction['predictions'][0]['values'][0][0]

    if (output <= 9):
        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 trought request
    '''
    data = request.get_json(force=True)
    prediction = model.y_predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)

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

```

Packages installed successfully: Installed packages: 'requests' (30 minutes ago)

56:1 CRLF UTF-8 4 spaces Python 3.10



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


import requests

API_KEY = "cPfRZC9gPSGV2pFnx5yDCBweDporgB8LKBLdM8le1n9v"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',data={"apikey":API_KEY,"grant_type":'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header= {'Content-Type':'application/json','Authorization':'Bearer'+mltoken}
app = Flask(__name__)

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

@app.route('/y_predict', methods=['POST', 'GET'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    x_test = [[int(x) for x in request.form.values()]]
    print(x_test)
    payload_scoring = {"input_data":[{"fields":[['f0', 'f1', 'f2', 'f3', 'f4', 'f5']],"values":x_test}]}
    response_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/0a65fea8-e525-4a26-987a-f668786b343e/predictions?version=2022-11-18', json=payload_scorin
    headers = {'Authorization':'Bearer'+mltoken}
    prediction = response_scoring.json()
    output = prediction['predictions'][0]['values'][0][0]

    if (output <= 9):
        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):
```

Packages installed successfully: Installed packages: 'requests' (30 minutes ago)