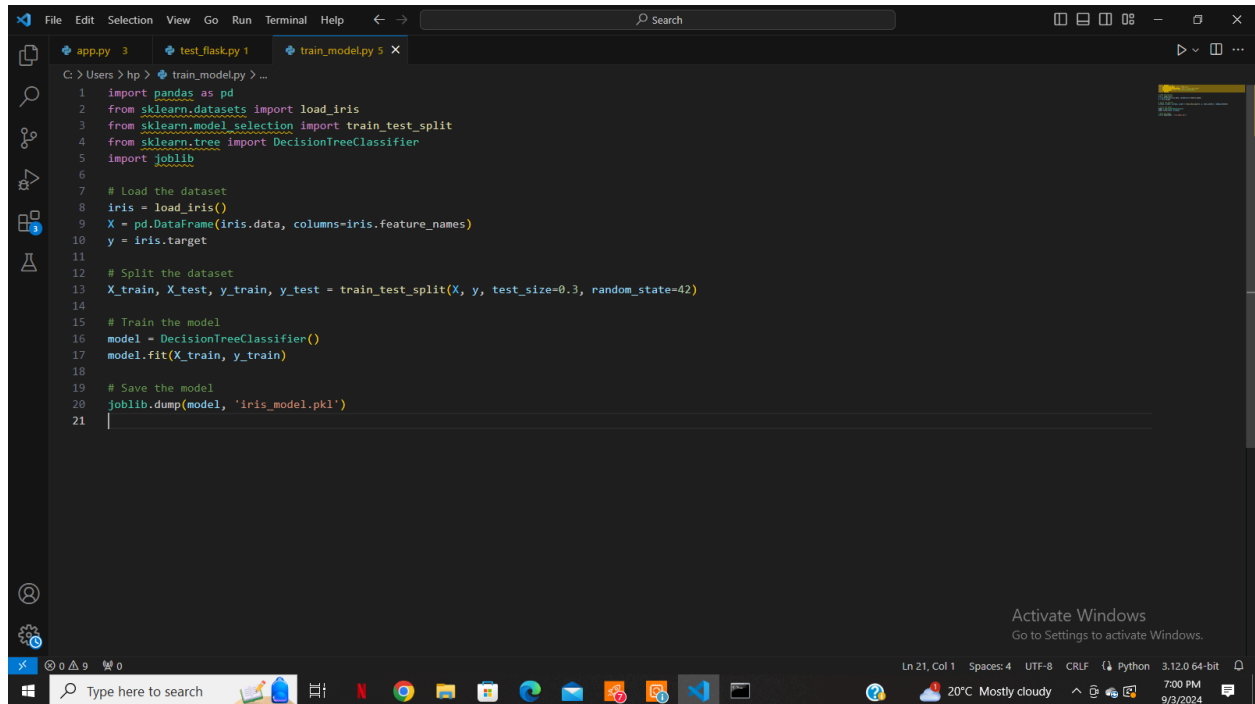


Name: Noella Mutuku

Batch code: LISUM31

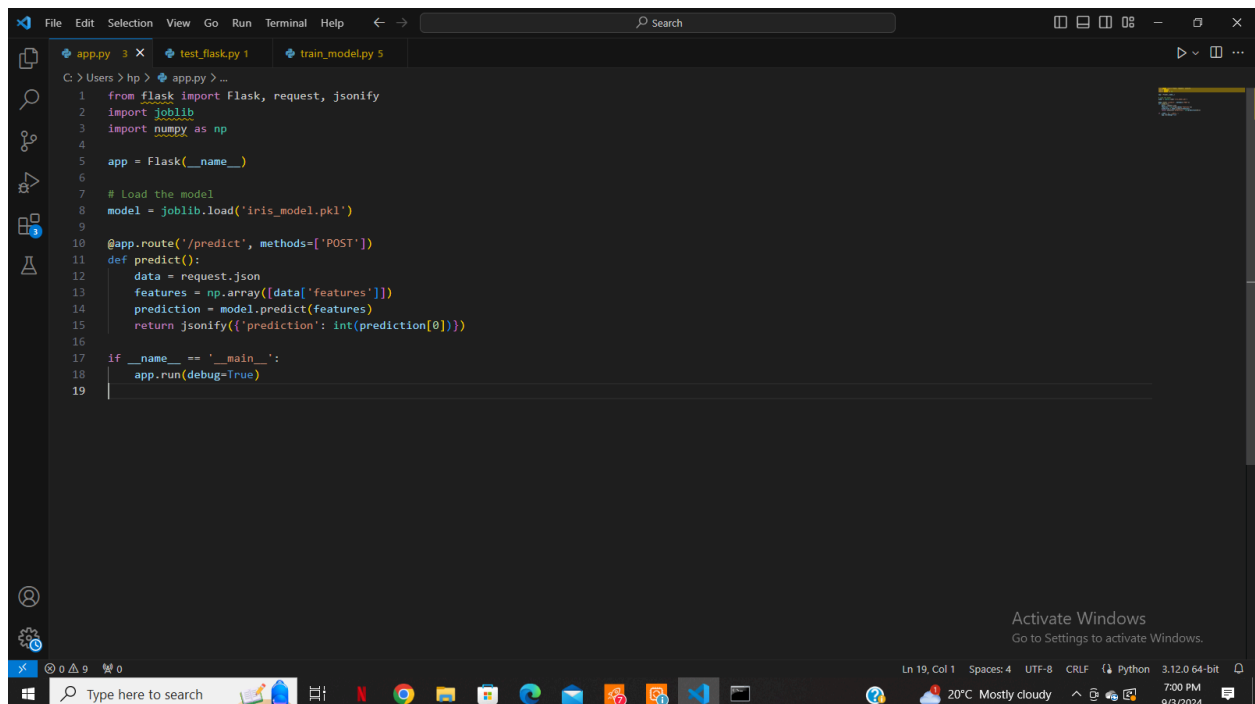
Submission date: 9/3/2024

Submitted to: Data Glacier



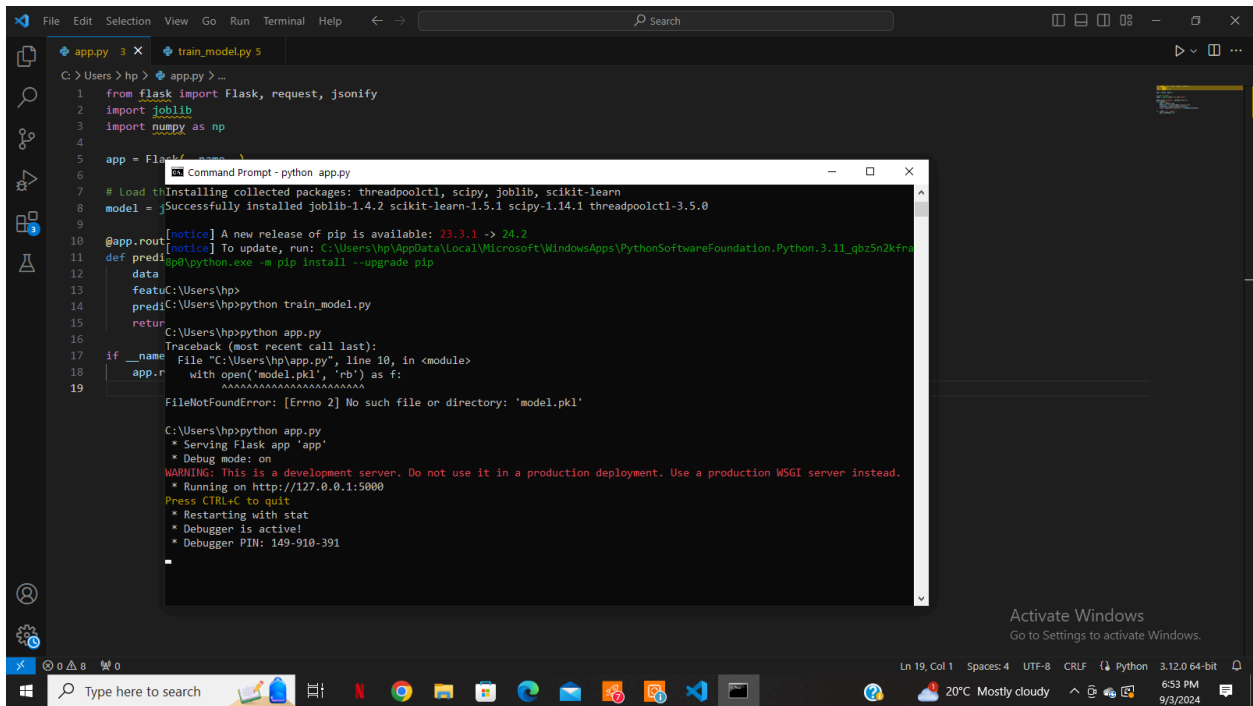
This screenshot shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows three files: `app.py`, `test_flask.py`, and `train_model.py`. The `train_model.py` file is open in the editor, showing Python code for training a Decision Tree Classifier on the Iris dataset. The code includes imports for `pandas`, `sklearn.datasets`, `sklearn.model_selection`, `sklearn.tree`, and `joblib`. It loads the Iris dataset, splits it into training and testing sets, trains a `DecisionTreeClassifier` model, and saves it as `iris_model.pkl`. The status bar at the bottom indicates the file is at line 21, column 1, using UTF-8 encoding and CRLF line endings. The Windows taskbar at the bottom shows the date and time as 7:00 PM on 9/3/2024.

```
File Edit Selection View Go Run Terminal Help Search
C:\Users\hp> train_model.py ...
1 import pandas as pd
2 from sklearn.datasets import load_iris
3 from sklearn.model_selection import train_test_split
4 from sklearn.tree import DecisionTreeClassifier
5 import joblib
6
7 # Load the dataset
8 iris = load_iris()
9 X = pd.DataFrame(iris.data, columns=iris.feature_names)
10 y = iris.target
11
12 # Split the dataset
13 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
14
15 # Train the model
16 model = DecisionTreeClassifier()
17 model.fit(X_train, y_train)
18
19 # Save the model
20 joblib.dump(model, 'iris_model.pkl')
21
```



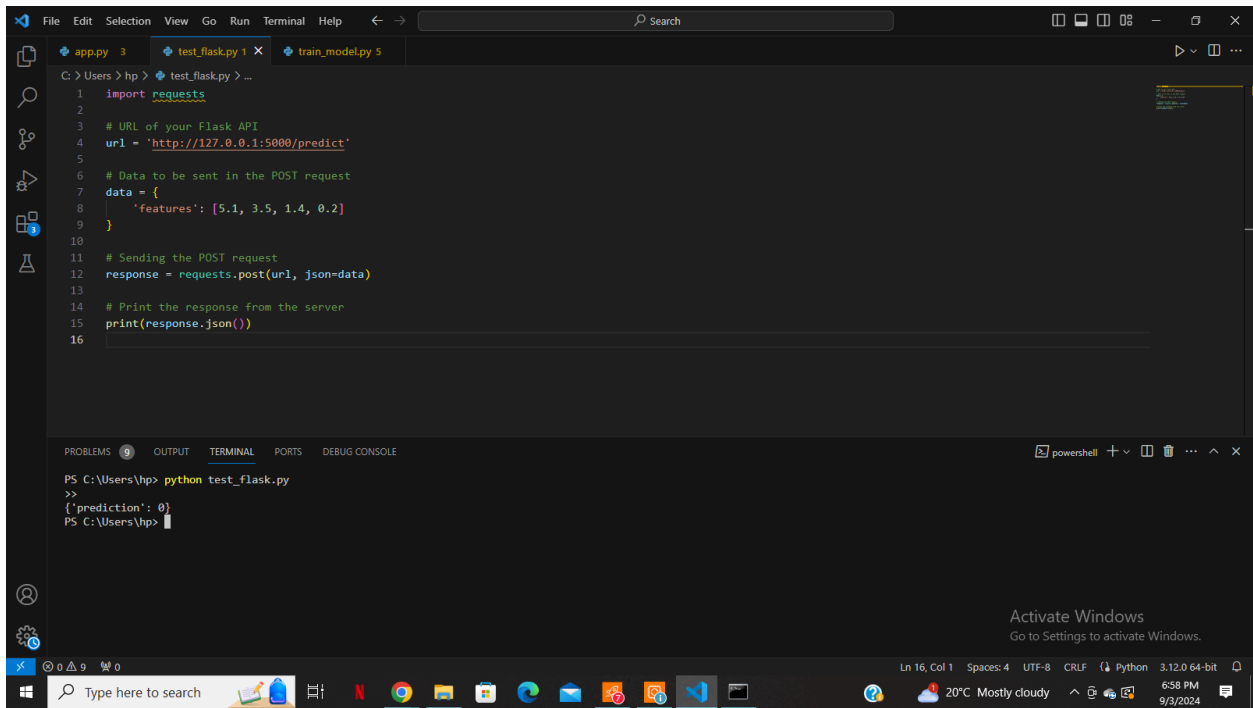
This screenshot shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows three files: `app.py`, `test_flask.py`, and `train_model.py`. The `app.py` file is open in the editor, showing Python code for a Flask web application. The code imports `Flask`, `request`, `jsonify`, `joblib`, and `numpy`. It loads the `iris_model.pkl` file and defines a `predict` endpoint that takes a JSON request, extracts the features, and returns the prediction. The status bar at the bottom indicates the file is at line 19, column 1, using UTF-8 encoding and CRLF line endings. The Windows taskbar at the bottom shows the date and time as 7:00 PM on 9/3/2024.

```
File Edit Selection View Go Run Terminal Help Search
C:\Users\hp> app.py ...
1 from flask import Flask, request, jsonify
2 import joblib
3 import numpy as np
4
5 app = Flask(__name__)
6
7 # Load the model
8 model = joblib.load('iris_model.pkl')
9
10 @app.route('/predict', methods=['POST'])
11 def predict():
12     data = request.json
13     features = np.array([data['features']])
14     prediction = model.predict(features)
15     return jsonify({'prediction': int(prediction[0])})
16
17 if __name__ == '__main__':
18     app.run(debug=True)
19
```



```
C:\Users\hp> python app.py
1 from flask import Flask, request, jsonify
2 import joblib
3 import numpy as np
4
5 app = Flask(__name__)
6
7 # Load the trained model
8 model = joblib.load('model.pkl')
9
10 @app.route('/predict', methods=['POST'])
11 def predict():
12     data = request.get_json()
13     features = data['features']
14     prediction = model.predict(features)
15     return jsonify({'prediction': prediction})
16
17 if __name__ == '__main__':
18     app.run(debug=True)
```

```
C:\Users\hp> python train_model.py
Installing collected packages: threadpoolctl, scipy, joblib, scikit-learn
Successfully installed joblib-1.4.2 scikit-learn-1.5.1 scipy-1.14.1 threadpoolctl-3.5.0
[notice] A new release of pip is available: 23.3.1 -> 24.2
[notice] To update, run: C:\Users\hp\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra
python.exe -m pip install --upgrade pip
C:\Users\hp> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 149-910-391
```



```
C:\Users\hp> python test_flask.py
1 import requests
2
3 # URL of your Flask API
4 url = 'http://127.0.0.1:5000/predict'
5
6 # Data to be sent in the POST request
7 data = {
8     'features': [5.1, 3.5, 1.4, 0.2]
9 }
10
11 # Sending the POST request
12 response = requests.post(url, json=data)
13
14 # Print the response from the server
15 print(response.json())
```

```
PS C:\Users\hp> python test_flask.py
>>
{'prediction': 0}
PS C:\Users\hp>
```