

FLASK DEPLOYMENT

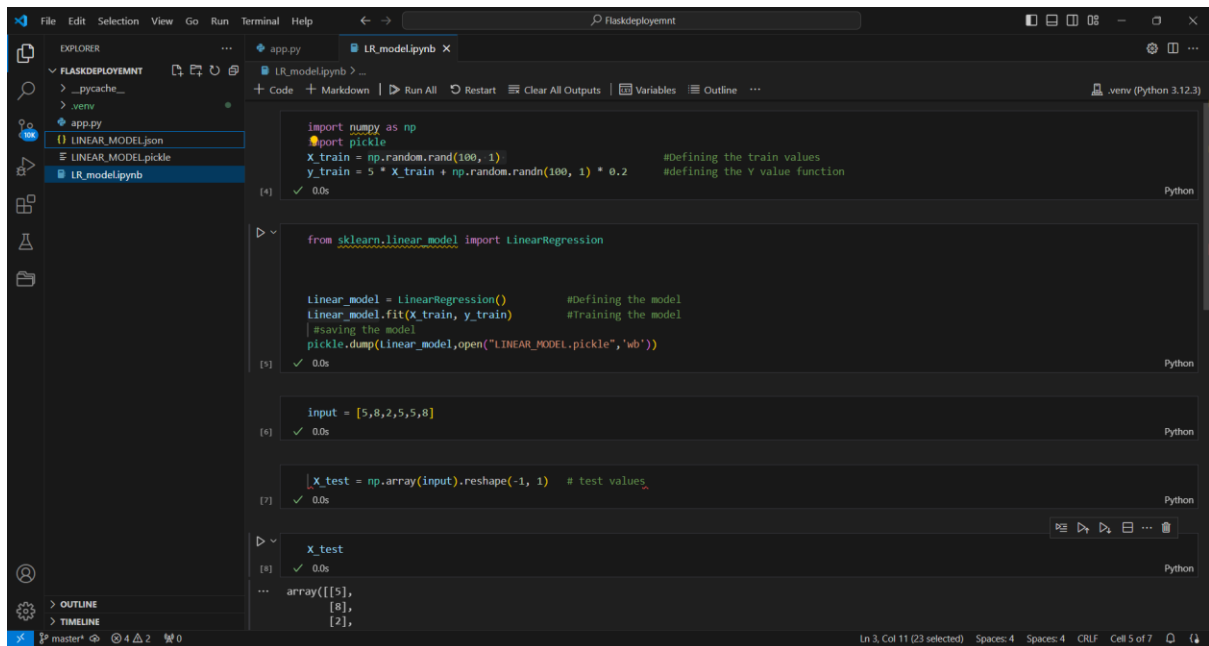
Name: Nazrin Thanikattil Rafeeqe

Batch code: LISUM33

Submission date: 28-05-2024

Submitted To: Tutor

Step 1: Create a Linear Regression model and save it using pickle..



The screenshot shows a Jupyter Notebook interface with the following code and outputs:

```
import numpy as np
import pickle
X_train = np.random.rand(100, 1) #Defining the train values
y_train = 5 * X_train + np.random.randn(100, 1) * 0.2 #defining the y value function
```

Output: 0.0s

```
from sklearn.linear_model import LinearRegression

linear_model = LinearRegression() #Defining the model
linear_model.fit(X_train, y_train) #Training the model
#saving the model
pickle.dump(linear_model, open("LINEAR_MODEL.pickle", 'wb'))
```

Output: 0.0s

```
input = [5,8,2,5,5,8]
```

Output: 0.0s

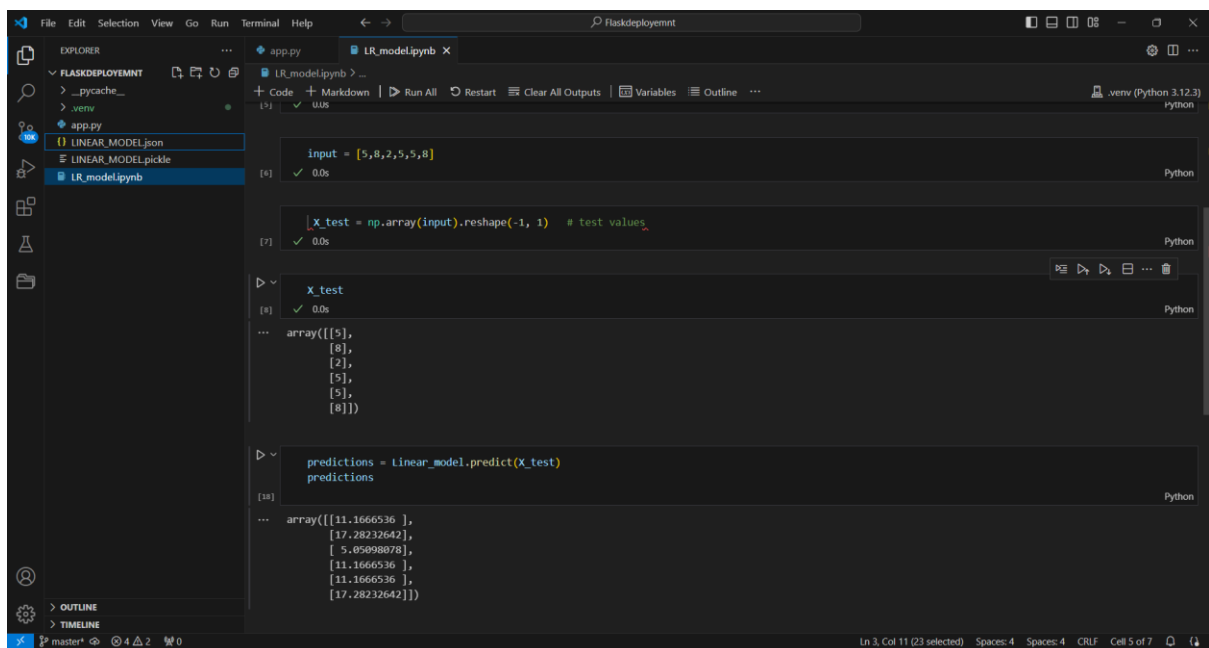
```
X_test = np.array(input).reshape(-1, 1) # test values
```

Output: 0.0s

```
X_test
```

Output: array([[5],
[8],
[2],
[5],
[5],
[8]])

Step 2: Check the model prediction for test input in ipynb.



The screenshot shows a Jupyter Notebook interface with the following code and outputs:

```
input = [5,8,2,5,5,8]
```

Output: 0.0s

```
X_test = np.array(input).reshape(-1, 1) # test values
```

Output: 0.0s

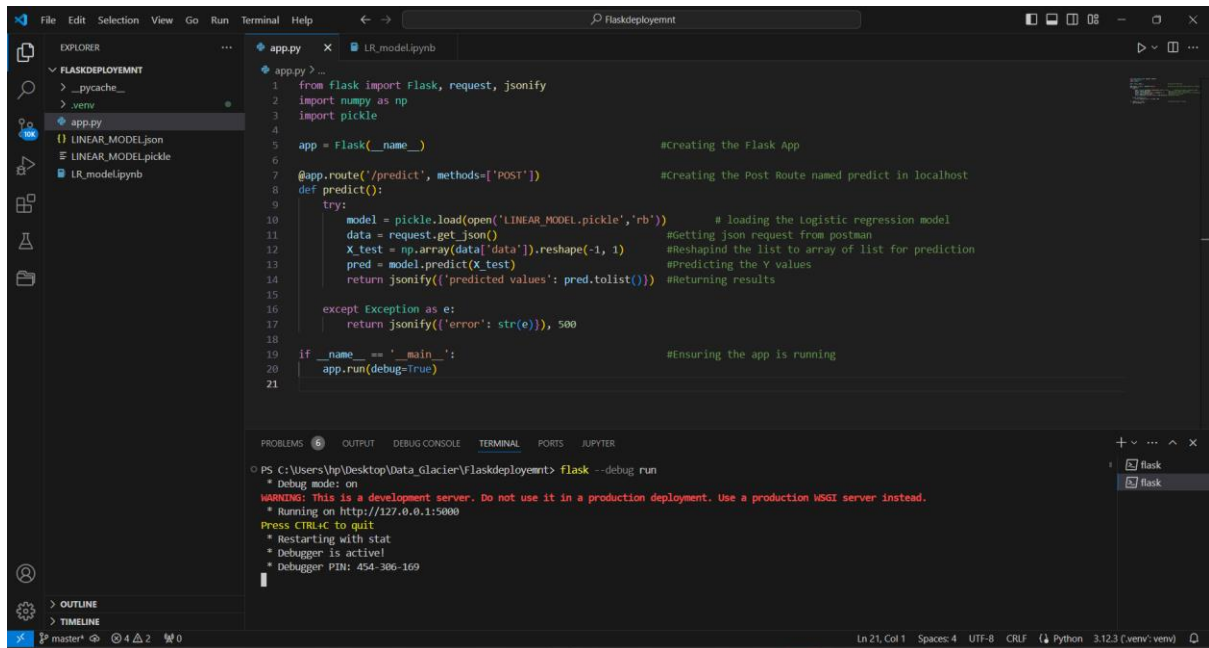
```
X_test
```

Output: array([[5],
[8],
[2],
[5],
[5],
[8]])

```
predictions = linear_model.predict(X_test)
predictions
```

Output: array([[11.1666536],
[17.28232642],
[5.05098078],
[11.1666536],
[11.1666536],
[17.28232642]])

Step 3: Create the Flask app by loading the logistic regression model and deploying it at the local host.



The screenshot shows a VS Code editor with a file explorer on the left containing files like `FLASKDEPLOYMENT`, `__pycache__`, `.venv`, `app.py`, `LINEAR_MODEL.json`, `LINEAR_MODEL.pickle`, and `LR_model.py`. The main editor displays the `app.py` file with the following code:

```
1 from flask import Flask, request, jsonify
2 import numpy as np
3 import pickle
4
5 app = Flask(__name__)
6
7 @app.route('/predict', methods=['POST'])
8 def predict():
9     try:
10         model = pickle.load(open('LINEAR_MODEL.pickle', 'rb'))
11         data = request.get_json()
12         X_test = np.array(data['data']).reshape(-1, 1)
13         pred = model.predict(X_test)
14         return jsonify({'predicted values': pred.tolist()})
15     except Exception as e:
16         return jsonify({'error': str(e)}), 500
17
18 if __name__ == '__main__':
19     app.run(debug=True)
```

The terminal at the bottom shows the command `flask --debug run` being executed, with output indicating the server is running on `http://127.0.0.1:5000`.

Step4: Testing the endpoint in postman

