FLASK DEPLOYMENT

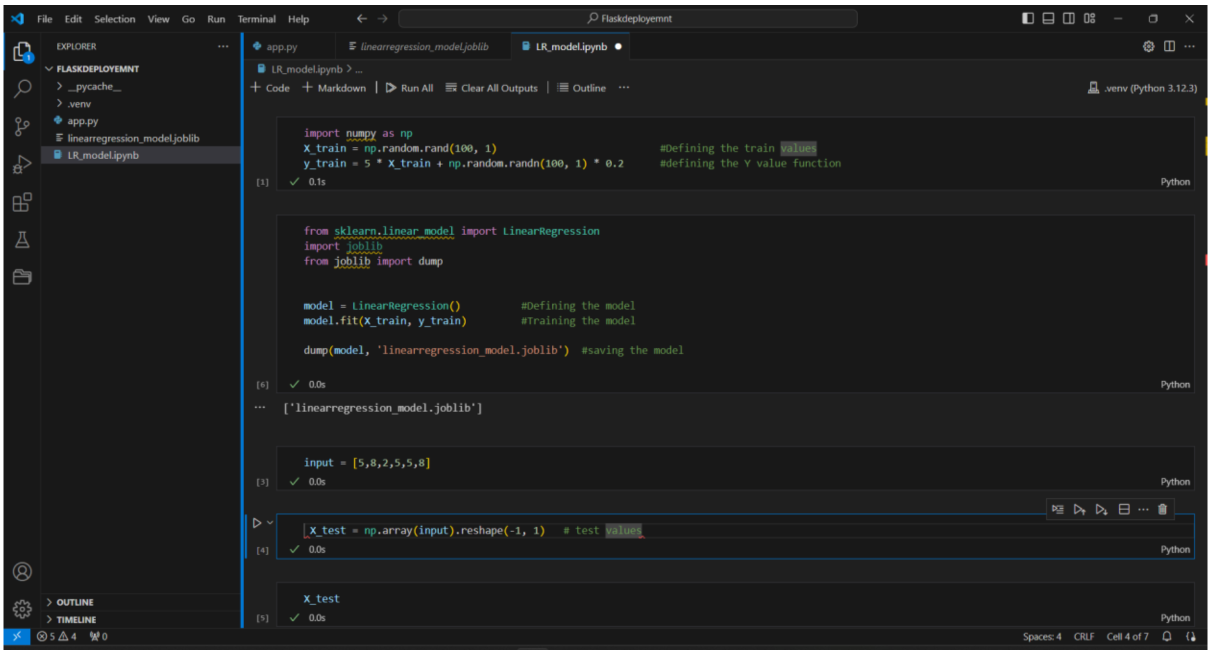
Name : Nazrin Thanikattil Rafeeque

Batch code : LISUM33

Submission date: 28-05-2024

Submitted To: Tutor

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



```
import numpy as np
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
```

```
from sklearn.linear_model import LinearRegression
import joblib
from joblib import dump


model = LinearRegression()          #Defining the model
model.fit(X_train, y_train)         #Training the model

dump(model, 'linearregression_model.joblib')  #saving the model
```

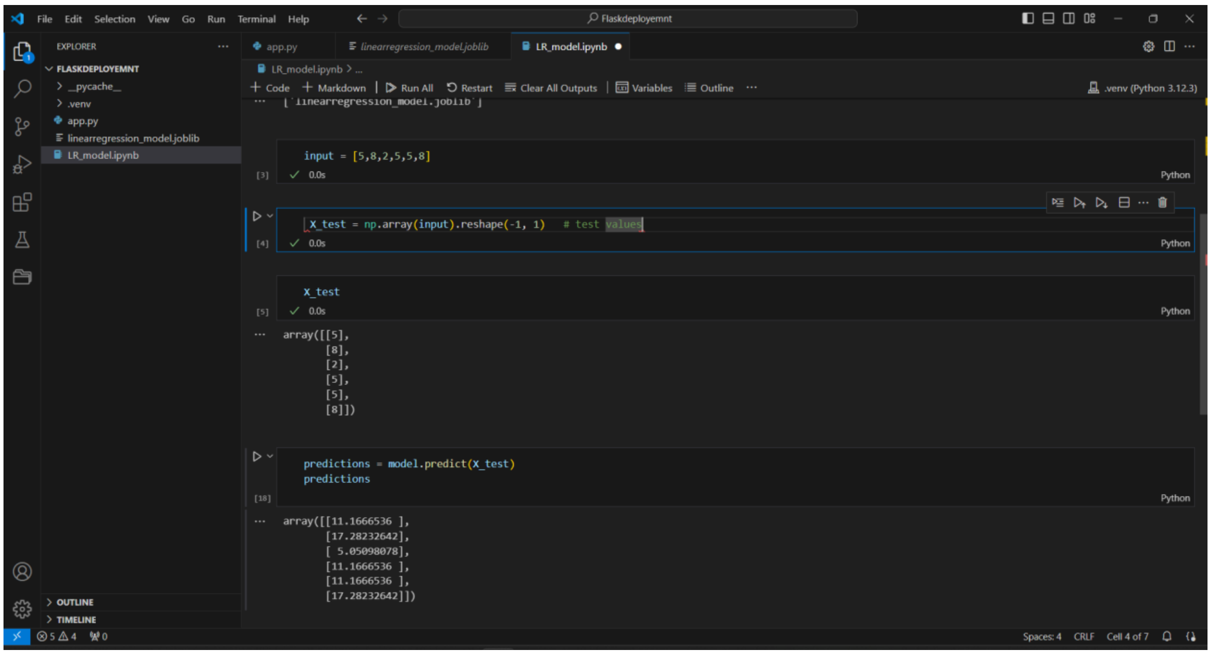
```
['linearregression_model.joblib']
```

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

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

```
X_test
```

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



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

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

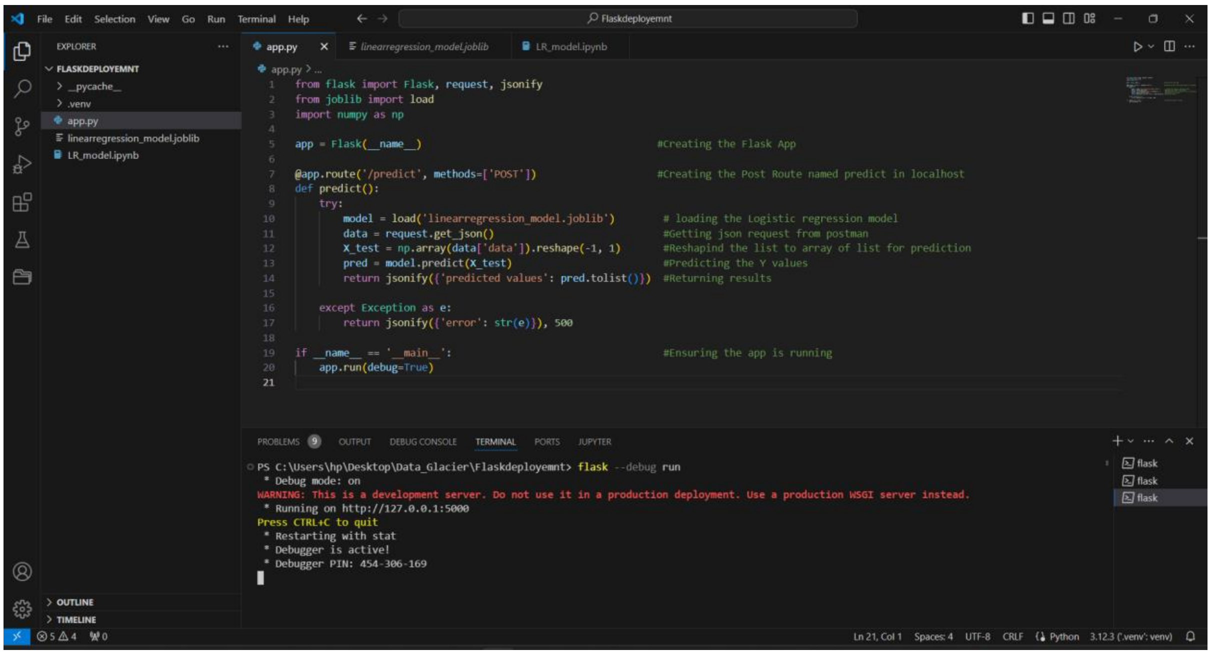
```
X_test
```

```
array([[5],
       [8],
       [2],
       [5],
       [5],
       [8]])
```

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

```
array([[11.1666536 ],
       [17.28232642],
       [ 5.05098078],
       [11.1666536 ],
       [11.1666536 ],
       [17.28232642]])
```

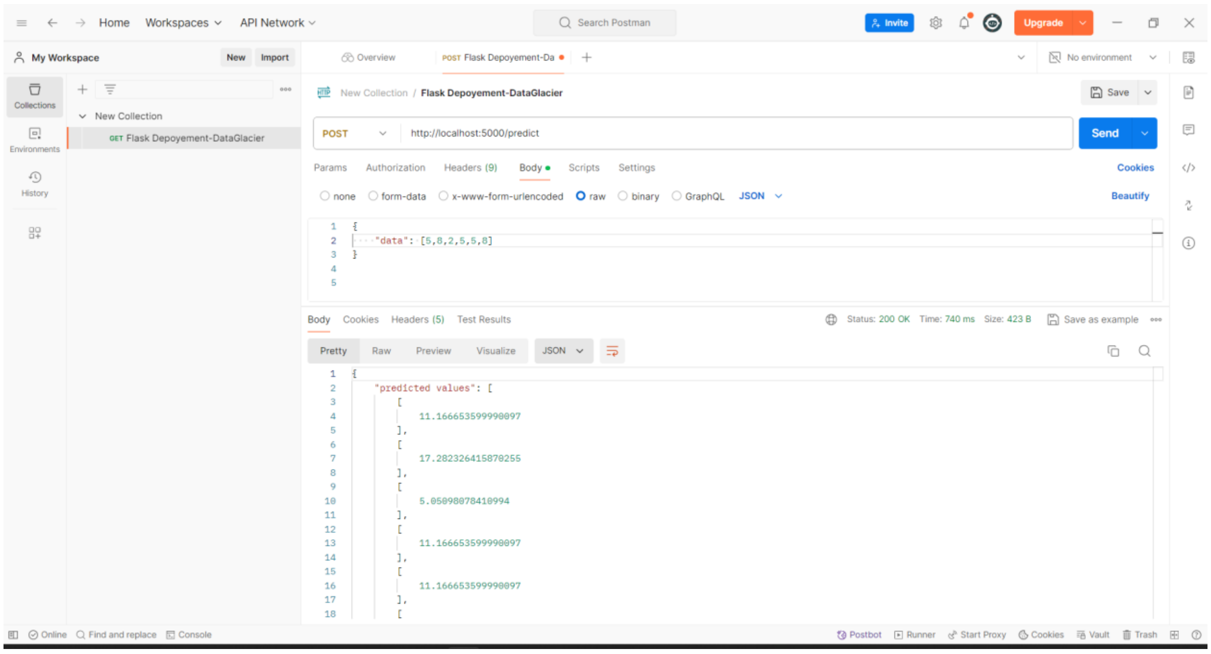
Step 3: Create the Flask app by loading the logistic regression model and deploying it at the local host



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

app = Flask(__name__)                                          #Creating the Flask App

@app.route('/predict', methods=['POST'])                       #Creating the Post Route named predict in localhost
def predict():
    try:
        model = load('linearregression_model.joblib')           # loading the Logistic regression model
        data = request.get_json()                               #Getting json request from postman
        X_test = np.array(data['data']).reshape(-1, 1)          #Reshapind the list to array of list for prediction
        pred = model.predict(X_test)                            #Predicting the Y values
        return jsonify({'predicted values': pred.tolist()})     #Returning results

    except Exception as e:
        return jsonify({'error': str(e)}), 500

if __name__ == '__main__':                                      #Ensuring the app is running
    app.run(debug=True)
```

```
PS C:\Users\hp\Desktop\Data_Glacier\Flaskdeployemnt> flask --debug run
 * 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: 454-306-169
```

Step4: Testing the endpoint in postman



POST http://localhost:5000/predict

```
{
    "data": [5,8,2,5,5,8]
}
```

Status: 200 OK   Time: 740 ms   Size: 423 B

```
{
    "predicted values": [
        [
            11.166653599990097
        ],
        [
            17.282326415870255
        ],
        [
            5.05098078410994
        ],
        [
            11.166653599990097
        ],
        [
            11.166653599990097
        ],
        [
```

******