Deployment Report

Name: Patryk Potocki
Batch Code: 1991
Submission Date: 2023-04-24
Submitted To: Data Glacier

Step 1: Toy Data Generation

The toy data was generated using the following code:

<>

The resulting dataset was saved to a CSV file called `toy_data.csv`.

Step 2: Model Training

The linear regression model was trained using scikit-learn:

<>

The trained model was saved to a file called `linear_regression.joblib`.

Step 3: Model Deployment

The Flask app was created using the following code:

<>

The app listens on port 5000 and has an endpoint `/predict`.

Snapshot 1:

```
# Generate toy data
np.random.seed(42)
num_samples = 100
hours_studied = np.random.uniform(low=0.0, high=10.0, size=num_samples)
exam_scores = 2.0 * hours_studied + np.random.normal(loc=0.0, scale=1.0, size=num_samples)
```

Snapshot 2:

```
# Save data to file
np.savetxt('toy_data.csv', np.column_stack((hours_studied, exam_scores)), delimiter=',', header='Hours Studied, Exam Score', comm
```

Snapshot 3:

```
from flask import Flask
import joblib
from flask import jsonify
import pandas as pd

# Create Flask app
app = Flask(__name__)

model = joblib.load("C:\\Users\\arjot\\linear_regression.joblib")


@app.route('/predict', methods=['GET'])
def index():
    data = pd.read_csv("C:\\Users\\arjot\\tax_data.csv")
    # Extract the feature data as a 2D array
    feature_data = data.values[:, :-1]
    predicted_data = model.predict(feature_data)
    return jsonify({'predicted_data': predicted_data.tolist()})


if __name__ == '__main__':
    app.run(debug=True)
```