

Week – 4: Deployment on Flask

Name: Ishwarya Rajaiya

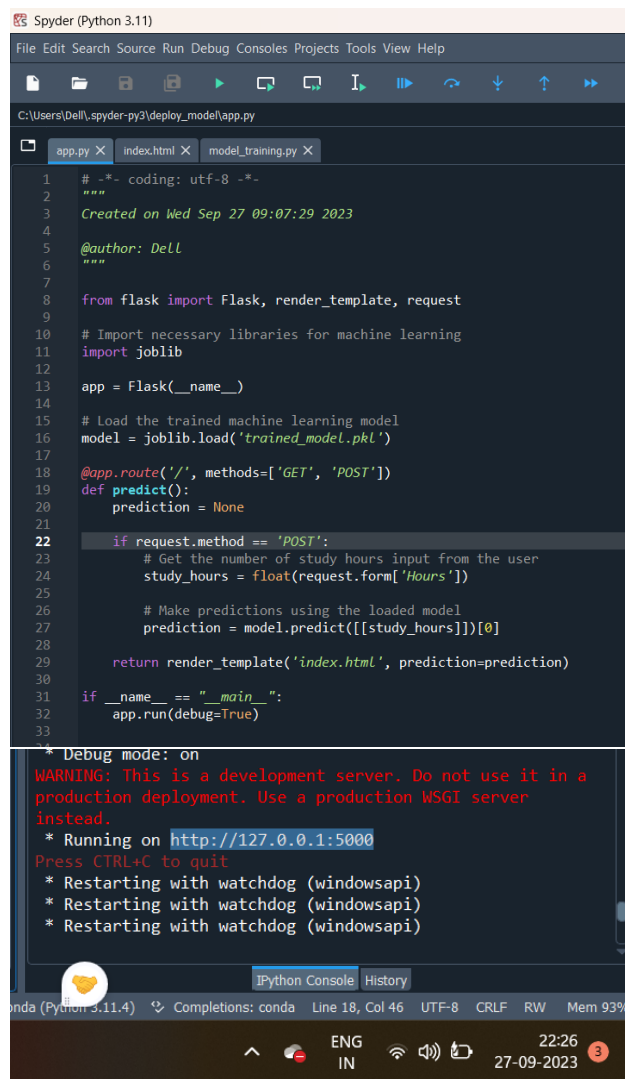
Batch code: LISUM25

Submission date: 27-09-2023

Submitted to: Data Glacier Dashboard

SCREENSHOTS OF MODEL DEPLOYMENT

1. Code of training the model, Flask application and HTML index.



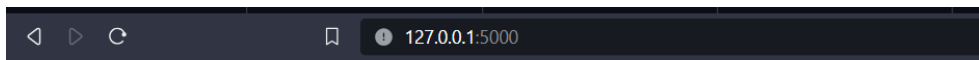
The screenshot displays the Spyder Python IDE interface. The top pane shows the code for `app.py`, which is a Flask application. The code includes imports for `Flask`, `render_template`, and `request` from the `flask` module, and `joblib` for loading the trained model. The application is configured to run on `http://127.0.0.1:5000`. The `predict` function is defined to handle POST requests, where it takes the number of study hours from the user's input and uses the loaded model to make a prediction. The bottom pane shows the console output, which includes a warning about using a development server in production and several messages indicating that the application is running and restarting with watchdog.

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Sep 27 09:07:29 2023
4
5  @author: Dell
6  """
7
8  from flask import Flask, render_template, request
9
10 # Import necessary libraries for machine learning
11 import joblib
12
13 app = Flask(__name__)
14
15 # Load the trained machine learning model
16 model = joblib.load('trained_model.pkl')
17
18 @app.route('/', methods=['GET', 'POST'])
19 def predict():
20     prediction = None
21
22     if request.method == 'POST':
23         # Get the number of study hours input from the user
24         study_hours = float(request.form['Hours'])
25
26         # Make predictions using the loaded model
27         prediction = model.predict([[study_hours]])[0]
28
29         return render_template('index.html', prediction=prediction)
30
31 if __name__ == "__main__":
32     app.run(debug=True)
33
```

* 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 watchdog (windowsapi)
* Restarting with watchdog (windowsapi)
* Restarting with watchdog (windowsapi)

```
app.py X index.html X model_training.py X
1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Sep 27 09:18:04 2023
4
5  @author: Dell
6  """
7
8  <!DOCTYPE html>
9  <html>
10 <head>
11 <title>Study Hours vs. Scores Prediction</title>
12 </head>
13 <body>
14 <h1>Study Hours vs. Scores Prediction</h1>
15
16 <form method="POST" action="/">
17 <label for="Hours">Enter Study Hours:</label>
18 <input type="text" id="Hours" name="Hours" required>
19 <input type="submit" value="Predict Score">
20 </form>
21
22 {% if prediction %}
23 <div>
24 <h2>Predicted Score:</h2>
25 <p>{{ prediction }}</p>
26 </div>
27 {% endif %}
28
29 </body>
30 </html>
31
app.py X index.html X model_training.py X
1  # -*- coding: utf-8 -*-
2  """
3  Created on Wed Sep 27 22:14:08 2023
4
5  @author: Dell
6  """
7
8  import pandas as pd
9  from sklearn.model_selection import train_test_split
10 from sklearn.linear_model import LinearRegression
11 import joblib
12
13 # Load your dataset (replace 'your dataset.csv' with the actual filename)
14 data = pd.read_excel("C:/Users/Dell/.spyder-py3/deploy_model/student_scores.xlsx")
15
16 # Split the data into features (study hours) and target (scores)
17 X = data['Hours'].values.reshape(-1, 1) # Features
18 y = data['Scores'].values # Target
19
20 # Split the data into training and testing sets
21 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
22
23 # Create and train a linear regression model
24 model = LinearRegression()
25 model.fit(X_train, y_train)
26
27 # Evaluate the model (optional)
28 train_score = model.score(X_train, y_train)
29 test_score = model.score(X_test, y_test)
30 print(f"Training R-squared: {train_score:.2f}")
31 print(f"Testing R-squared: {test_score:.2f}")
32
33 # Save the trained model to a file (e.g., 'trained_model.pkl')
34 joblib.dump(model, 'trained_model.pkl')
35
```

2. Model deployment with predictions:



-*- coding: utf-8 -*- """ Created on Wed Sep 27 09:18:04 2023 @author: Dell """

Study Hours vs. Scores Prediction

Enter Study Hours:

-*- coding: utf-8 -*- """ Created on Wed Sep 27 09:18:04 2023 @author: Dell """

Study Hours vs. Scores Prediction

Enter Study Hours:



-*- coding: utf-8 -*- """ Created on Wed Sep 27 09:18:04 2023 @author: Dell """

Study Hours vs. Scores Prediction

Enter Study Hours:

Predicted Score:

99.6476738984567