

Week 4 Assignment - Week4: Deployment on Flask

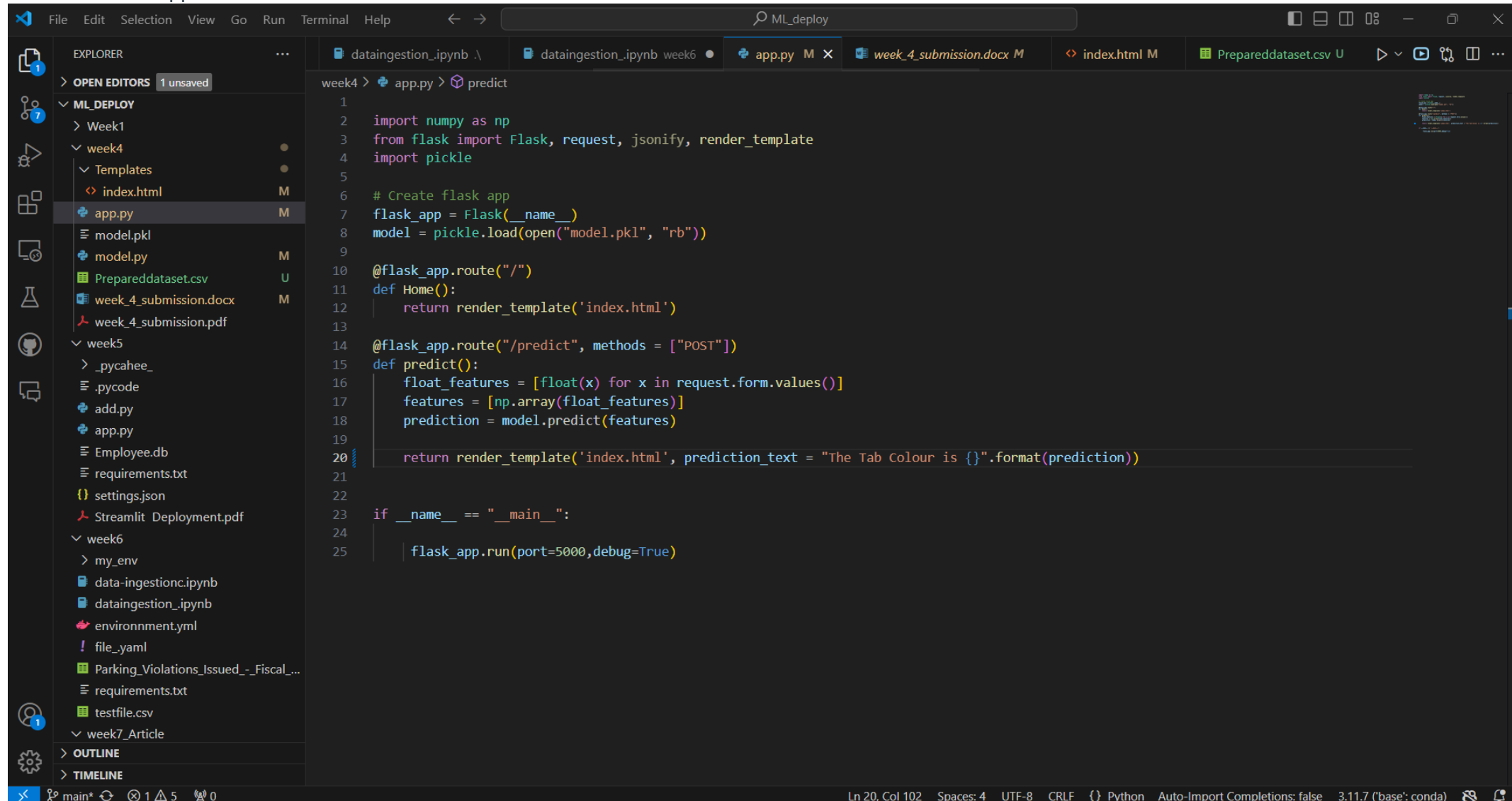
Name: Vijayarajan Vijaya Jothi

Batch code: LISUM37

Submission date: Sep 23, 2024

Submitted to: Data Glacier

1. Code of Flask app



```
1
2 import numpy as np
3 from flask import Flask, request, jsonify, render_template
4 import pickle
5
6 # Create flask app
7 flask_app = Flask(__name__)
8 model = pickle.load(open("model.pkl", "rb"))
9
10 @flask_app.route("/")
11 def Home():
12     return render_template('index.html')
13
14 @flask_app.route("/predict", methods = ["POST"])
15 def predict():
16     float_features = [float(x) for x in request.form.values()]
17     features = np.array(float_features)
18     prediction = model.predict(features)
19
20     return render_template('index.html', prediction_text = "The Tab Colour is {}".format(prediction))
21
22
23 if __name__ == "__main__":
24
25     flask_app.run(port=5000, debug=True)
```

2. index.html

The image shows a Visual Studio Code editor window with the following components:

- Explorer Panel (Left):** Displays the project structure. The 'ML_DEPLOY' folder is expanded, showing subfolders 'Week1', 'week4', and 'week5'. The 'week4' folder is further expanded, showing 'Templates' and 'index.html' (selected). Other files in the project include 'app.py', 'model.pkl', 'model.py', 'Prepareddataset.csv', 'week_4_submission.docx', 'week_4_submission.pdf', 'week5', 'week6', and 'week7_Article'.
- Editor Panel (Center):** Displays the content of 'index.html'. The code is as follows:

```
1 <html ></html>
2 <!--From https://codepen.io/frytyler/pen/EGdtg-->
3 <head>
4   <meta charset="UTF-8">
5   <title>ML API</title>
6   <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
7   <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
8   <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
9   <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
10
11 </head>
12
13 <body>
14   <div class="login">
15     <h1>Tab Color Prediction</h1>
16
17     <!-- Main Input For Receiving Query to our ML -->
18     <form action="{{ url_for('predict')}}" method="post">
19       <input type="text" name="Length1Cm" placeholder="Length1Cm" required="required" />
20       <input type="text" name="Width1Cm" placeholder="Width1Cm" required="required" />
21       <input type="text" name="Length2Cm" placeholder="Length2Cm" required="required" />
22       <input type="text" name="Width2Cm" placeholder="Width2Cm" required="required" />
23
24       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
25     </form>
26
27     <br>
28     <br>
29     {{ prediction_text }}
30
31   </div>
32
33 </body>
34 </html>
```
- Terminal Panel (Bottom):** Shows the status bar with 'Ln 12, Col 1', 'Spaces: 2', 'UTF-8', 'CRLF', 'HTML', and a search icon.

3. Running Flask in debug mode in VisualStudioCode

The screenshot displays the Visual Studio Code interface with a Python Flask application open in the editor. The Explorer panel on the left shows the project structure, including files like `index.html`, `app.py`, `model.pkl`, `model.py`, `Prepareddataset.csv`, `week_4_submission.docx`, `week_4_submission.pdf`, `week5`, `week6`, and `week7_Article`. The main editor window shows the `app.py` file, which defines a Flask application with a `predict` endpoint. The terminal at the bottom shows the command to run the application in debug mode, and the output indicates that the server is running on `http://127.0.0.1:5000` and that the debugger is active.

```
week4 > app.py > predict
1
2 import numpy as np
3 from flask import Flask, request, jsonify, render_template
4 import pickle
5
6 # Create flask app
7 flask_app = Flask(__name__)
8 model = pickle.load(open("model.pkl", "rb"))
9
10 @flask_app.route("/")
11 def Home():
12     return render_template('index.html')
13
14 @flask_app.route("/predict", methods = ["POST"])
15 def predict():
16     float_features = [float(x) for x in request.form.values()]
17     features = [np.array(float_features)]
18     prediction = model.predict(features)
19
20     return render_template('index.html', prediction_text = "The Tab Colour is {}".format(prediction))
21
22
23 if __name__ == "__main__":
24
```

Terminal Output:

```
cours@Shiv MINGW64 C:/Users/cours/AppData/Local/Programs/Microsoft VS Code (main)
$ C:/Users/cours/anaconda3/python.exe c:/Users/cours/OneDrive/Desktop/ML_deploy/week4/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 watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 226-013-682
```

4. Running app homepage

New tab

×

ML API

×

+

←

↻

i

127.0.0.1:5000

★

⚙️

|

📄

☆

🔍

📌

📶

⋮

Tab Color Prediction

Length1Cm

Width1Cm

Length2Cm

Width2Cm

Predict

5. Filling test data

New tab

ML API

ML API

127.0.0.1:5000/predict

Tab Color Prediction

The Tab Colour is ['green']

6. Prediction result shown on web page

New tabML APIML API

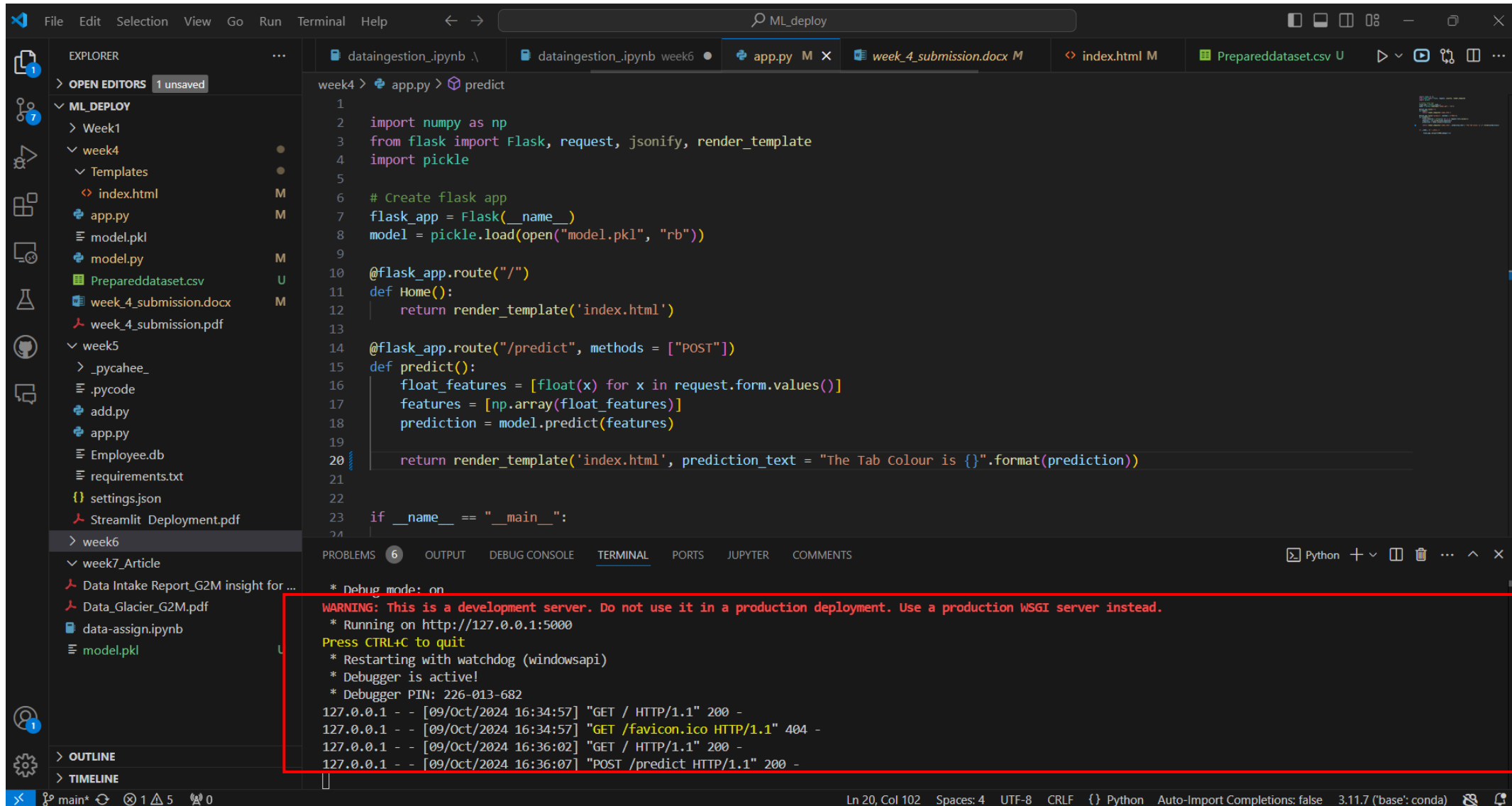
127.0.0.1:5000/predict

☆⚙️📄🔖📌📧⋮

Tab Color Prediction

The Tab Colour is ['green']

7. Flask output in terminal window



The screenshot shows a VS Code editor with a project named 'ML_deploy'. The Explorer panel on the left shows a file structure with folders 'week4' and 'week5', and files like 'index.html', 'app.py', 'model.pkl', 'Prepareddataset.csv', 'week_4_submission.docx', and 'week_4_submission.pdf'. The main editor window displays the code for 'app.py', which is a Flask application. The code includes imports for numpy, flask, jsonify, render_template, and pickle. It defines a Flask app, loads a model from 'model.pkl', and has two routes: a home route and a '/predict' route that takes a POST request and returns a prediction. The terminal window at the bottom shows the output of running the application. It includes a warning about using a development server, the URL 'http://127.0.0.1:5000', and several log entries showing GET and POST requests.

```
1
2 import numpy as np
3 from flask import Flask, request, jsonify, render_template
4 import pickle
5
6 # Create flask app
7 flask_app = Flask(__name__)
8 model = pickle.load(open("model.pkl", "rb"))
9
10 @flask_app.route("/")
11 def Home():
12     return render_template('index.html')
13
14 @flask_app.route("/predict", methods = ["POST"])
15 def predict():
16     float_features = [float(x) for x in request.form.values()]
17     features = [np.array(float_features)]
18     prediction = model.predict(features)
19
20     return render_template('index.html', prediction_text = "The Tab Colour is {}".format(prediction))
21
22
23 if __name__ == "__main__":
24
```

```
* 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)
* Debugger is active!
* Debugger PIN: 226-013-682
127.0.0.1 - - [09/Oct/2024 16:34:57] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Oct/2024 16:34:57] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [09/Oct/2024 16:36:02] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Oct/2024 16:36:07] "POST /predict HTTP/1.1" 200 -
```