

Name: Alankrita Mitra

Batch Code: LISUM21

Submission Date: 20/05/2023

Submitted to:

Screenshots:

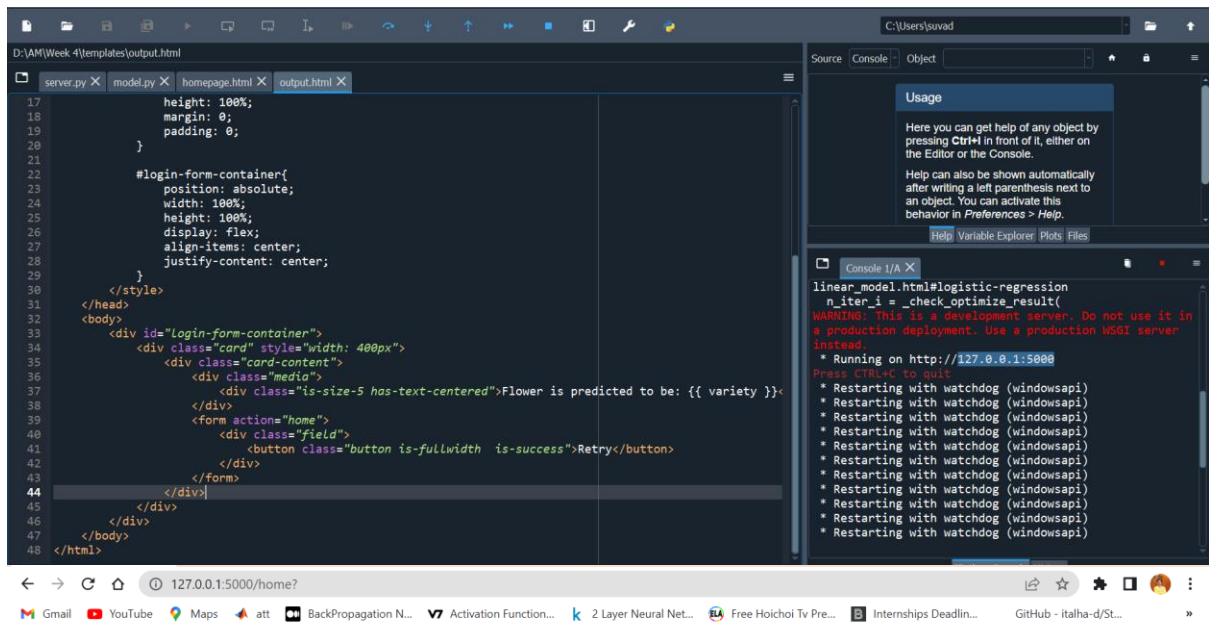
The image displays two screenshots of the Spyder Python IDE interface. The top screenshot shows a Python script named 'model.py' with the following code:

```
1 # Importing necessary libraries
2 import pandas as pd
3 import numpy as np
4 from sklearn.linear_model import LogisticRegression
5
6 # Importing the dataset
7 df = pd.read_csv('iris.csv')
8
9 # mapping
10 variety_mappings = {0: 'Setosa', 1: 'Versicolour', 2: 'Virginica'}
11
12 # Encoding the target variables
13 df = df.replace(['Setosa', 'Versicolour', 'Virginica'], [0, 1, 2])
14
15 X = df.iloc[:, 0:-1]
16 y = df.iloc[:, -1]
17 # Initializing the Logistic Regression model
18 logreg = LogisticRegression()
19 # Fitting the model
20 logreg.fit(X, y)
21
22
23 def classify(a, b, c, d):
24     arr = np.array([a, b, c, d])
25     arr = arr.astype(np.float64)
26     query = arr.reshape(1, -1)
27     prediction = variety_mappings[logreg.predict(query)[0]]
28     return prediction # Return the prediction
29
```

The bottom screenshot shows a Flask web application script named 'server.py' with the following code:

```
1 import model
2 from flask import Flask, request, render_template
3
4 # Initialize the flask class and specify the templates directory
5 app = Flask(__name__, template_folder="templates")
6
7 # Default route set as 'home'
8 @app.route('/')
9 def home():
10     return render_template('homepage.html')
11
12
13 @app.route('/classify', methods=['POST', 'GET'])
14 def classify_type():
15     try:
16         sepal_len = request.args.get('slen')
17         sepal_wid = request.args.get('swid')
18         petal_len = request.args.get('plen')
19         petal_wid = request.args.get('pwid')
20
21         # Get the output from the classification model
22         variety = model.classify(sepal_len, sepal_wid, petal_len, petal_wid)
23
24         # Render the output in new HTML page
25         return render_template('output.html', variety=variety)
26     except:
27         return 'Error'
28
29 # Run the Flask server
30 if __name__ == '__main__':
31     app.run(debug=True)
```

Both screenshots show the Spyder IDE with a file explorer on the left, a code editor in the center, and a console/output pane on the right. The console in the top screenshot shows the output of the model training and a warning message about the development server. The console in the bottom screenshot shows the output of the Flask application running on port 5000.



Flower Classification

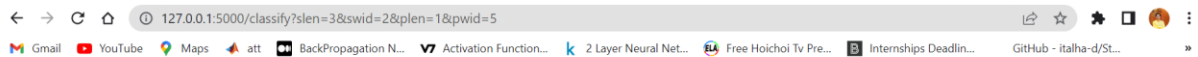
Sepal Length:

Sepal Width:

Petal Length:

Petal Width:

Submit



Flower is predicted to be: Setosa

Retry