

Name: Dharshana Uvaraj

LISUM 46

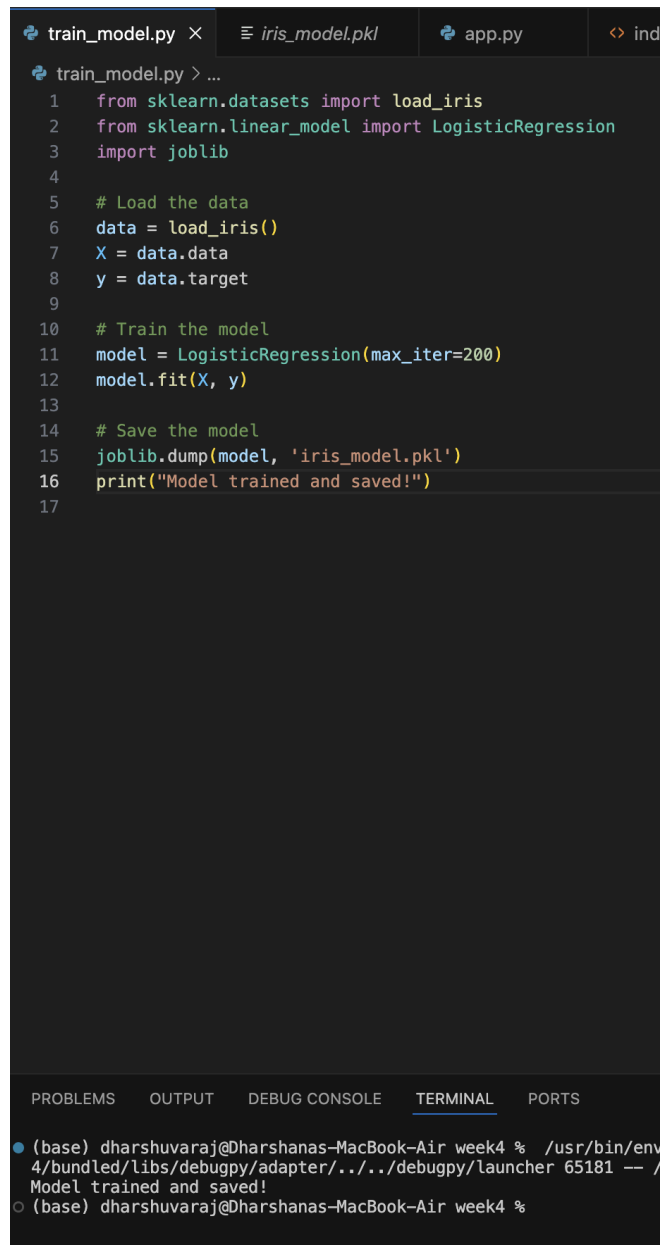
06/24/25

Submitted to: Github

Deploying a ML Model with Flask

Iris Flower Classifier - Model Deployment Report

Loading and Training the Model:



```
train_model.py × iris_model.pkl app.py ind
train_model.py > ...
1  from sklearn.datasets import load_iris
2  from sklearn.linear_model import LogisticRegression
3  import joblib
4
5  # Load the data
6  data = load_iris()
7  X = data.data
8  y = data.target
9
10 # Train the model
11 model = LogisticRegression(max_iter=200)
12 model.fit(X, y)
13
14 # Save the model
15 joblib.dump(model, 'iris_model.pkl')
16 print("Model trained and saved!")
17

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
• (base) dharshuvaraj@Dharshanas-MacBook-Air week4 % /usr/bin/env
4/bundled/libs/debugpy/adapters/.../debugpy/launcher 65181 -- /
Model trained and saved!
○ (base) dharshuvaraj@Dharshanas-MacBook-Air week4 %
```

Create Flask Web App:

```
train_model.py  iris_model.pkl  app.py  index.html  Settings

app.py > ...
1  from flask import Flask, request, jsonify, render_template
2  import joblib
3
4  app = Flask(__name__)
5  model = joblib.load('iris_model.pkl')
6
7  @app.route('/')
8  def home():
9      return render_template('index.html')
10
11 @app.route('/predict', methods=['POST'])
12 def predict():
13     f1 = float(request.form['f1'])
14     f2 = float(request.form['f2'])
15     f3 = float(request.form['f3'])
16     f4 = float(request.form['f4'])
17     prediction = model.predict([[f1, f2, f3, f4]])
18     return jsonify({'prediction': int(prediction[0])})
19
20 if __name__ == '__main__':
21     app.run(debug=True)
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(base) dharshuvaraj@unknown6a4ab0411193 week4 % python 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 pr
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 142-358-036
127.0.0.1 - - [23/Jun/2025 21:40:49] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/Jun/2025 21:40:57] "POST /predict HTTP/1.1" 200 -
```

Create input form (HTML):

```
train_model.py  iris_model.pkl  app.py  index.html ×
templates > <> index.html > ...
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Iris Prediction</title>
5    </head>
6    <body>
7      <h2>Enter Flower Features</h2>
8      <form action="/predict" method="post">
9        <input type="text" name="f1" placeholder="Sepal Length"><br>
10       <input type="text" name="f2" placeholder="Sepal Width"><br>
11       <input type="text" name="f3" placeholder="Petal Length"><br>
12       <input type="text" name="f4" placeholder="Petal Width"><br>
13       <button type="submit">Predict</button>
14     </form>
15   </body>
16 </html>
17
```

Use the Web App to predict:

Enter Flower Features

Sepal Length
Sepal Width
Petal Length
Petal Width
Predict

Enter Flower Features

5.1
2.4
3.6
4.7
Predict

"prediction": 2

What the program does:

The app uses an AI model to predict the type of iris flower based on four measurements. It supports three classes: Setosa (0), Versicolor (1), and Virginica (2).