

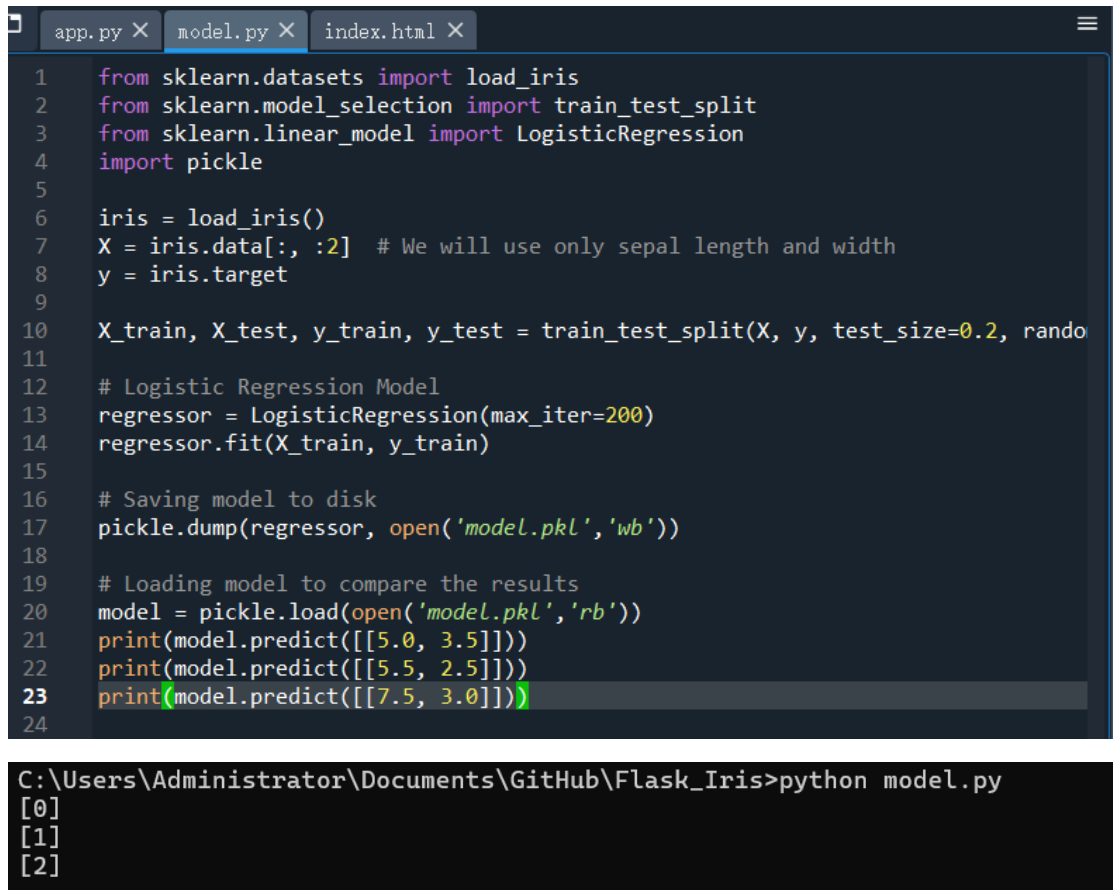
Name: Chenxin Shen

Batch Code: LISUM35

Submission Date: July 28, 2024

Submitted to: Week4: Deployment on Flask

1. Create model and run to obtain pickle file (and also check saved model works)



The image shows a code editor with three tabs: 'app.py', 'model.py', and 'index.html'. The 'model.py' tab is active, displaying Python code for training and saving a Logistic Regression model. The code imports necessary libraries, loads the Iris dataset, splits it into training and testing sets, trains the model, and saves it as 'model.pkl'. It also includes a test section where the model is loaded and predictions are made for three specific input points. Below the code editor, a terminal window shows the command to run the model.py file, followed by the output of the predictions: [0], [1], and [2].

```
1 from sklearn.datasets import load_iris
2 from sklearn.model_selection import train_test_split
3 from sklearn.linear_model import LogisticRegression
4 import pickle
5
6 iris = load_iris()
7 X = iris.data[:, :2] # We will use only sepal length and width
8 y = iris.target
9
10 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
11
12 # Logistic Regression Model
13 regressor = LogisticRegression(max_iter=200)
14 regressor.fit(X_train, y_train)
15
16 # Saving model to disk
17 pickle.dump(regressor, open('model.pkl', 'wb'))
18
19 # Loading model to compare the results
20 model = pickle.load(open('model.pkl', 'rb'))
21 print(model.predict([[5.0, 3.5]]))
22 print(model.predict([[5.5, 2.5]]))
23 print(model.predict([[7.5, 3.0]]))
24
```

```
C:\Users\Administrator\Documents\GitHub\Flask_Iris>python model.py
[0]
[1]
[2]
```

2. Create HTML template

```
app.py × model.py × index.html ×
1 <!DOCTYPE html>
2 <html >
3 <head>
4   <meta charset="UTF-8">
5   <title>ML API</title>
6 </head>
7
8 <body>
9   <div class="login">
10    <h1>Predict Iris Class</h1>
11
12    <!-- Main Input For Receiving Query to our ML -->
13    <form action="{{ url_for('predict')}}"method="post">
14      <input type="text" name="sepal_length" placeholder="Sepal Length" required="required" />
15      <input type="text" name="sepal_width" placeholder="Sepal Width" required="required" />
16
17      <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
18    </form>
19
20    <br>
21    <br>
22    {{ prediction_text }}
23
24   </div>
25
26 </body>
27 </html>
```

3. Create Flask website and run it

```
app.py × model.py × index.html ×
1 import numpy as np
2 from flask import Flask, request, render_template
3 import pickle
4
5 app = Flask(__name__)
6 model = pickle.load(open('model.pkl', 'rb'))
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12
13 @app.route('/predict',methods=['POST'])
14 def predict():
15     int_features = [float(x) for x in request.form.values()]
16     final_features = [np.array(int_features)]
17     prediction = model.predict(final_features)
18
19     classes = ["Setosa", "Versicolor", "Virginica"]
20     output = classes[prediction[0]]
21
22     return render_template('index.html', prediction_text='Sepal should be from the class {}'.format(output))
23
24 if __name__ == "__main__":
25     app.run(debug=True)
```

```
C:\Users\Administrator\Documents\GitHub\Flask_Iris>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 production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 173-653-045
```

4. Open the website

## Predict Iris Class

<input type="text" value="Sepal Length"/>	<input type="text" value="Sepal Width"/>	<input type="button" value="Predict"/>
---	--	--

5. Enter some test value and hit predict for result

## Predict Iris Class

<input type="text" value="5.5"/>	<input type="text" value="7"/>	<input type="button" value="Predict"/>
----------------------------------	--------------------------------	--

## Predict Iris Class

<input type="text" value="Sepal Length"/>	<input type="text" value="Sepal Width"/>	<input type="button" value="Predict"/>
---	--	--

Sepal should be from the class Setosa.