

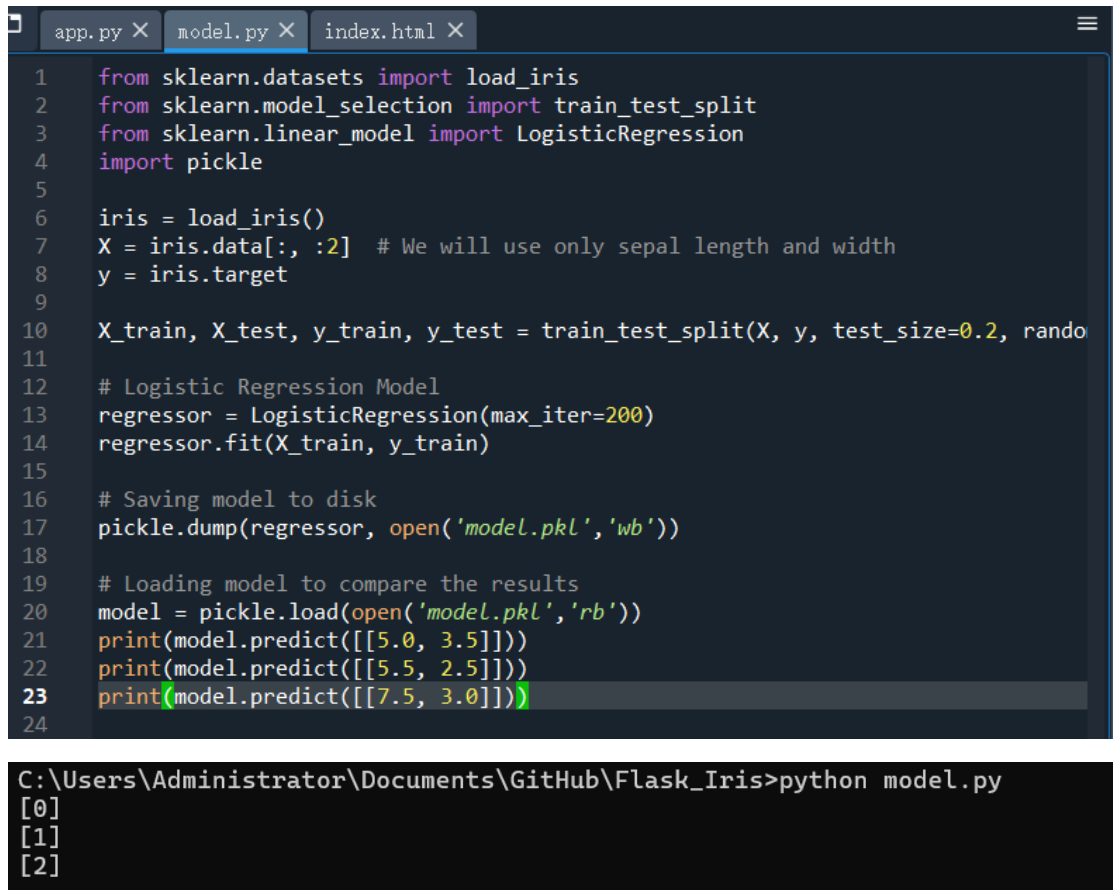
Name: Chenxin Shen

Batch Code: LISUM35

Submission Date: August 5th, 2024

Submitted to: Week 5: Cloud and API deployment

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 testing a Logistic Regression model. The code imports necessary libraries, loads the Iris dataset, splits it into training and testing sets, trains the model, saves it to a file named 'model.pkl', and then loads it back to test on specific input values. Below the code editor, a terminal window shows the command 'python model.py' being executed, resulting in three lines of output: '[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. Create app on Heroku

App name

Choose a region


 United States


Add to pipeline...


Create app Cancel

5. Connect Github repository to the app

Deployment method

 Heroku Git
Use Heroku CLI

 GitHub
Connected

 Container Registry
Use Heroku CLI

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to [AmyQShen/Flask_Iris](#) by [AmyQShen](#) [Disconnect...](#)

Releases in the [activity feed](#) link to GitHub to view commit diffs

6. Deploy the main branch

Manual deploy


Deploy the current state of a branch to this app.


Deploy a GitHub branch


This will deploy the current state of the branch you specify below. [Learn more](#)


Choose a branch to deploy

[Deploy Branch](#)

Receive code from GitHub 

Build main @39da3fb 

Release phase 

Deploy to Heroku 

Your app was successfully deployed.

[View](#)

7. I think I ran into a problem because I do not have a plan. I will try to do this on some other platforms.