Name: Yue Hu
Batch Code: LISUM33
Submission Date: 2024/05/29
Submission To: Glacier

Process:

1. Select a data set:
   I chose a data set named 'iris', which is contained in Python. Then I separate the data set in two parts, 80% of which is for training the model and 20% of which is for testing.

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
import pandas as pd

#load the data set
iris = load_iris()
X = iris.data
y = iris.target

# 20% for testing and 80%for training
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

2. Build a model:
   I chose to build a Random Forest model to predict the species of iris.

```
from sklearn.ensemble import RandomForestClassifier
# random forest
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

import pickle
pickle.dump(model, open('model.pickle','wb'))
```

3. Use Flask application:

```
from flask import Flask,jsonify,request
app=Flask(__name__)
model=pickle.load(open('model.pickle','rb'))

@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict/',methods=['POST'])
def class_predict():
    int_features=[int(x) for x in request.form.values()]
    final_features=[np.array(int_features)]
    prediction=model.predict(final_features)
    return render_template('index.html',prediction_text='The species of iris is ${}'.format(prediction))

if __name__=='__main__':
    app.run(debug=True)
```

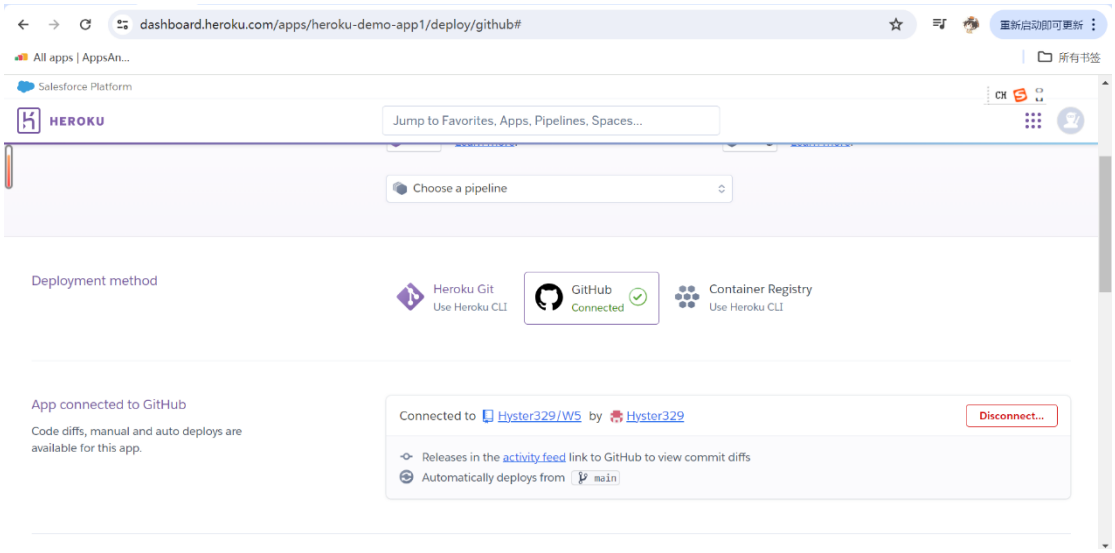
4. Write index.html:

```
<body>
<div class="login">
   <h1>Iris species prediction</h1>

    <!-- Main Input For Receiving Query to our ML -->
   <form action="{{ url_for('predict')}}"method="post">
       <input type="text" name="sepal_length" placeholder="sepal_length" required="required" />
       <input type="text" name="sepal_width" placeholder="sepal_width" required="required" />
       <input type="text" name="petal_length" placeholder="petal_length" required="required" />
       <input type="text" name="petal_width" placeholder="petal_width" required="required" />

       <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
   </form>

   <br>
   <br>
   {{ prediction_text }}

</div>
<img src="/static/images/Original.svg" style="width: 400px;position: absolute;bottom: 10px;left: 10px;" alt="Company Logo"/>

</body>
```

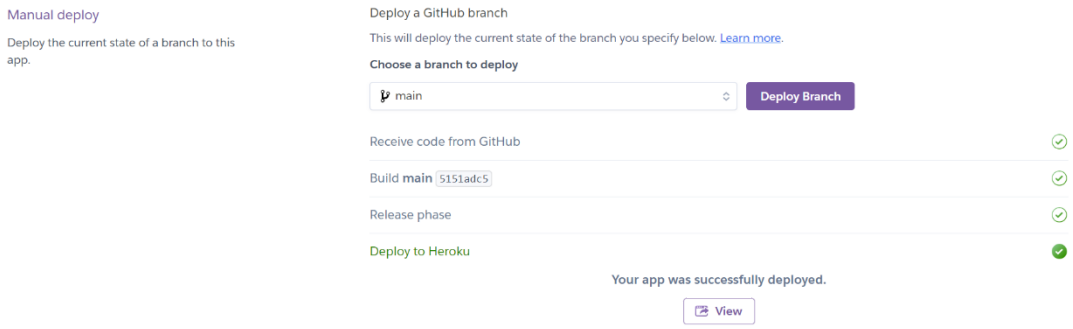
5. Run the python file:

```
D:\programming\Python\Anaconda\python.exe C:\Users\Hyste\Downloads\App1.py
 * Serving Flask app 'App1'
 * 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: 523-706-781
 * Detected change in 'C:\\Users\\Hyste\\Downloads\\App1.py', reloading
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 523-706-781
```

6. Link the Heroku to Github:



7. Deploy


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
main
Deploy Branch
Receive code from GitHub
Build main 5151adc5
Release phase
Deploy to Heroku
Your app was successfully deployed.
View