Name: Tony Nguyen

Batch Code: LISUM30

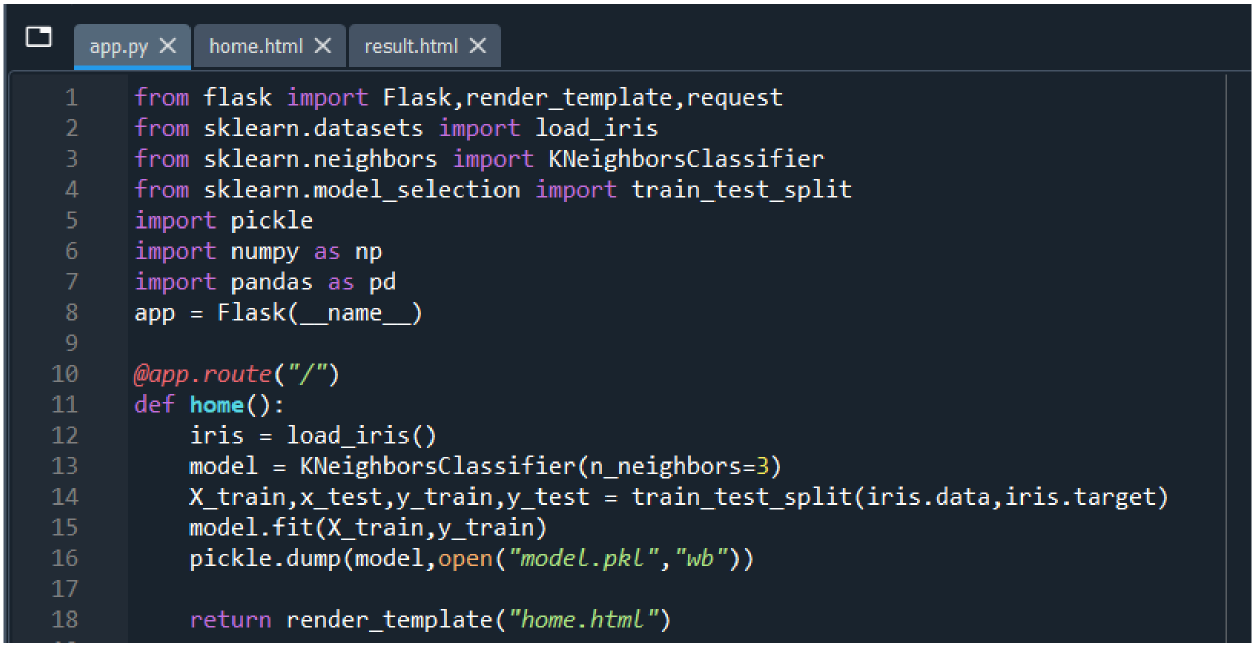
Submission date: 2/27/2024

Submitted to: Data Glacier

Deployment on Flask

Step 1:

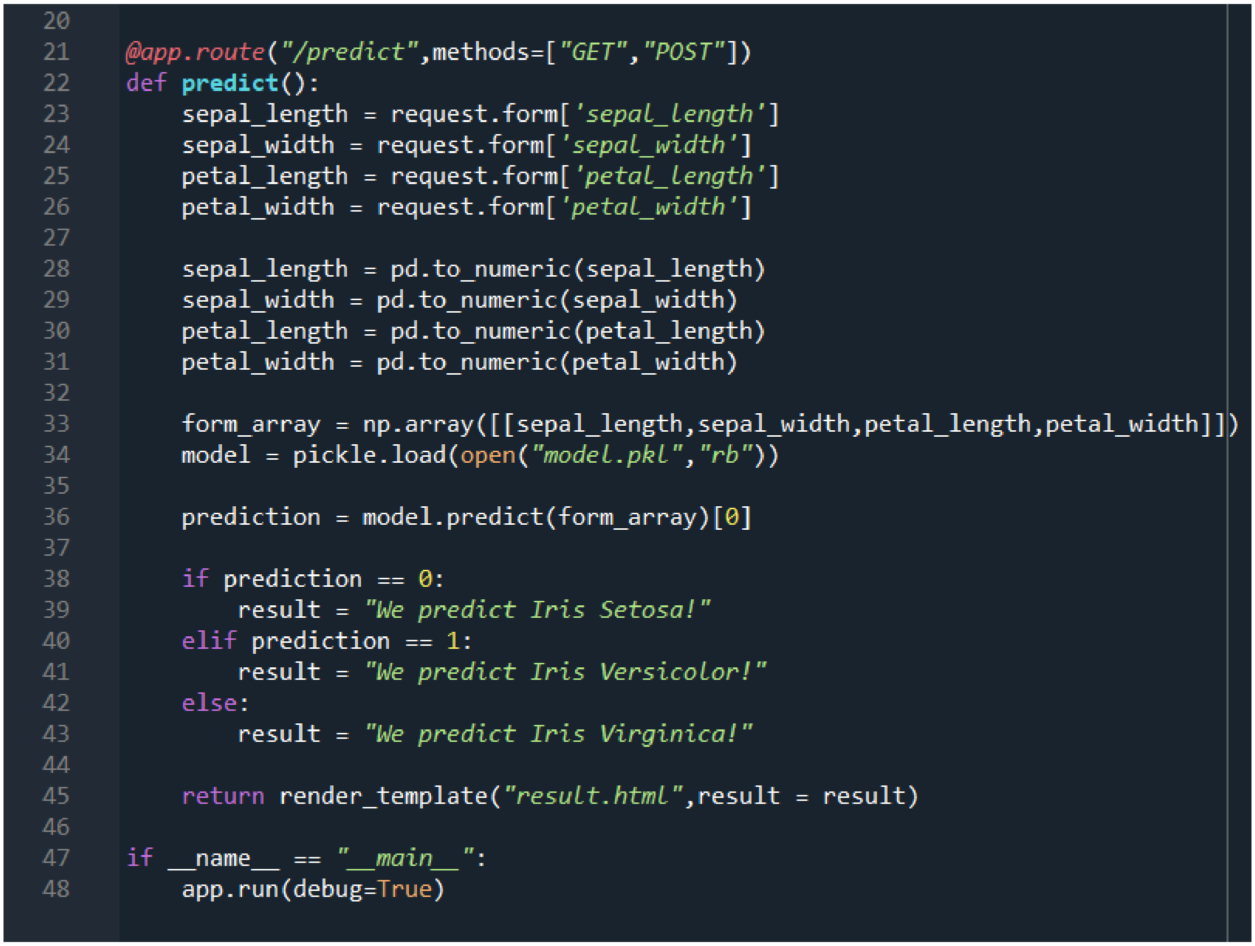
Develop Model – Predict the type of iris flower based on the length and width of the sepal and petal.



```
from flask import Flask,render_template,request
from sklearn.datasets import load_iris
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
import pickle
import numpy as np
import pandas as pd
app = Flask(__name__)

@app.route("/")
def home():
    iris = load_iris()
    model = KNeighborsClassifier(n_neighbors=3)
    X_train,x_test,y_train,y_test = train_test_split(iris.data,iris.target)
    model.fit(X_train,y_train)
    pickle.dump(model,open("model.pkl","wb"))

    return render_template("home.html")
```

Step 2:

Saving model and Deployment – Uses *pickle* library to save trained model. Takes given input from users and predicts iris flower based on the inputs given.

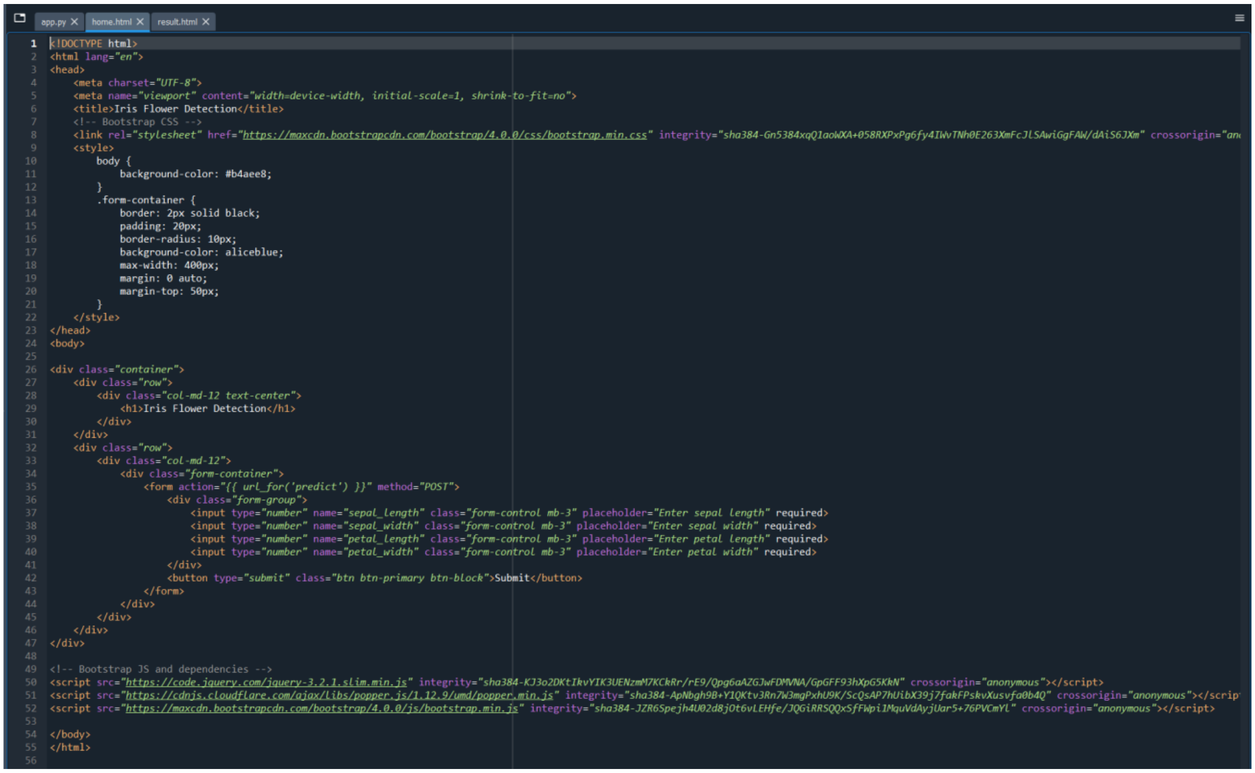


```
@app.route("/predict",methods=["GET","POST"])
def predict():
    sepal_length = request.form['sepal_length']
    sepal_width = request.form['sepal_width']
    petal_length = request.form['petal_length']
    petal_width = request.form['petal_width']

    sepal_length = pd.to_numeric(sepal_length)
    sepal_width = pd.to_numeric(sepal_width)
    petal_length = pd.to_numeric(petal_length)
    petal_width = pd.to_numeric(petal_width)

    form_array = np.array([[sepal_length,sepal_width,petal_length,petal_width]])
    model = pickle.load(open("model.pkl","rb"))

    prediction = model.predict(form_array)[0]

    if prediction == 0:
        result = "We predict Iris Setosa!"
    elif prediction == 1:
        result = "We predict Iris Versicolor!"
    else:
        result = "We predict Iris Virginica!"

    return render_template("result.html",result = result)

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

Step 3:

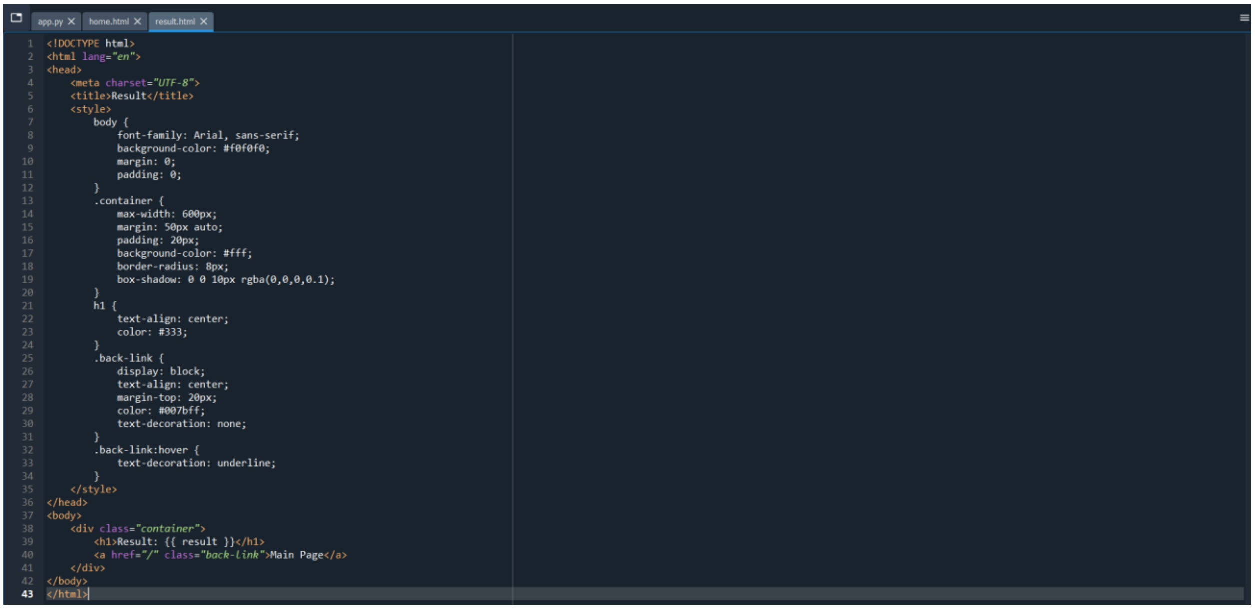
Creating home page and result page formats

Main Page:



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Iris Flower Detection</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="an
    <style>
        body {
            background-color: #b4aee8;
        }
        .form-container {
            border: 2px solid black;
            padding: 20px;
            border-radius: 10px;
            background-color: aliceblue;
            max-width: 400px;
            margin: 0 auto;
            margin-top: 50px;
        }
    </style>
</head>
<body>

<div class="container">
    <div class="row">
        <div class="col-md-12 text-center">
            <h1>Iris Flower Detection</h1>
        </div>
    </div>
    <div class="row">
        <div class="col-md-12">
            <div class="form-container">
                <form action="{{ url_for('predict') }}" method="POST">
                    <div class="form-group">
                        <input type="number" name="sepal_length" class="form-control mb-3" placeholder="Enter sepal length" required>
                        <input type="number" name="sepal_width" class="form-control mb-3" placeholder="Enter sepal width" required>
                        <input type="number" name="petal_length" class="form-control mb-3" placeholder="Enter petal length" required>
                        <input type="number" name="petal_width" class="form-control mb-3" placeholder="Enter petal width" required>
                    </div>
                    <button type="submit" class="btn btn-primary btn-block">Submit</button>
                </form>
            </div>
        </div>
    </div>
</div>

<!-- Bootstrap JS and dependencies -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></scrip
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

</body>
</html>

```

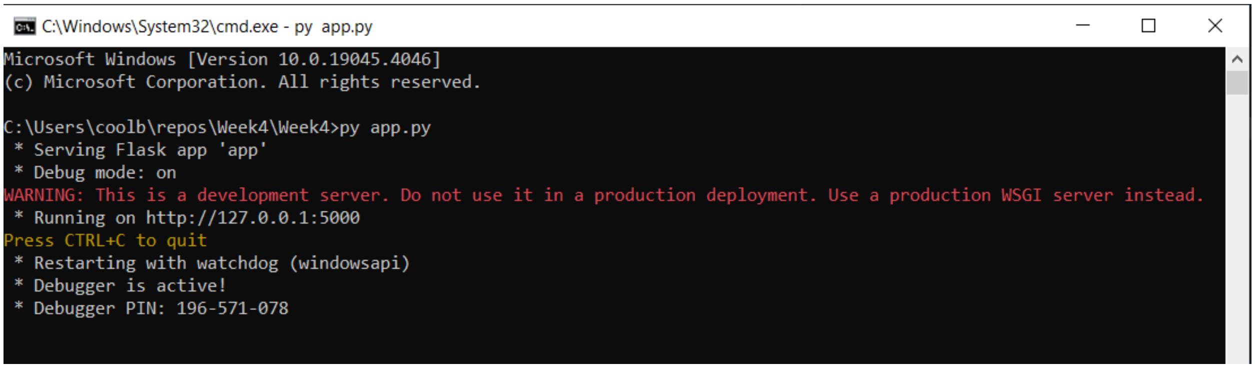
Results Page:



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Result</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            margin: 0;
            padding: 0;
        }
        .container {
            max-width: 600px;
            margin: 50px auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }
        h1 {
            text-align: center;
            color: #333;
        }
        .back-link {
            display: block;
            text-align: center;
            margin-top: 20px;
            color: #007bff;
            text-decoration: none;
        }
        .back-link:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Result: {{ result }}</h1>
        <a href="/" class="back-link">Main Page</a>
    </div>
</body>
</html>
```

Step 4:

Python app.py file in CMD.

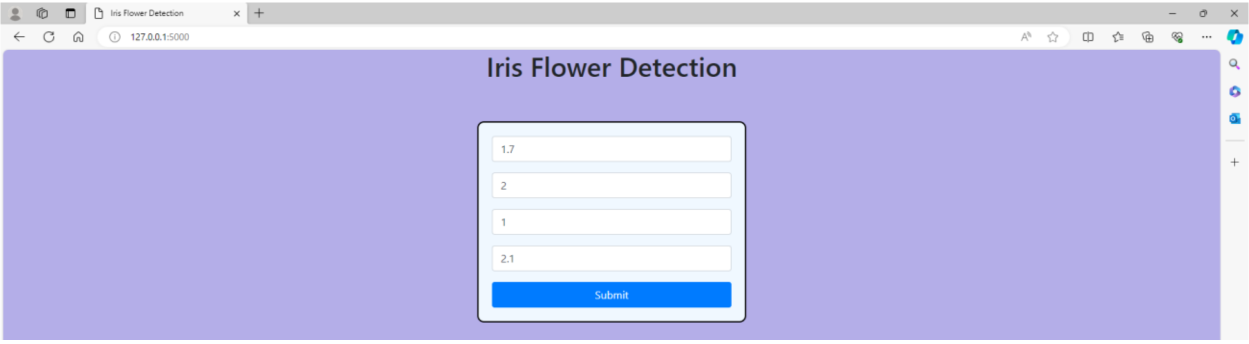


```
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\coolb\repos\Week4\Week4>py 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 watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 196-571-078
```

Step 5:

Web App – Using the URL and testing the application.

Main Page:



Iris Flower Detection

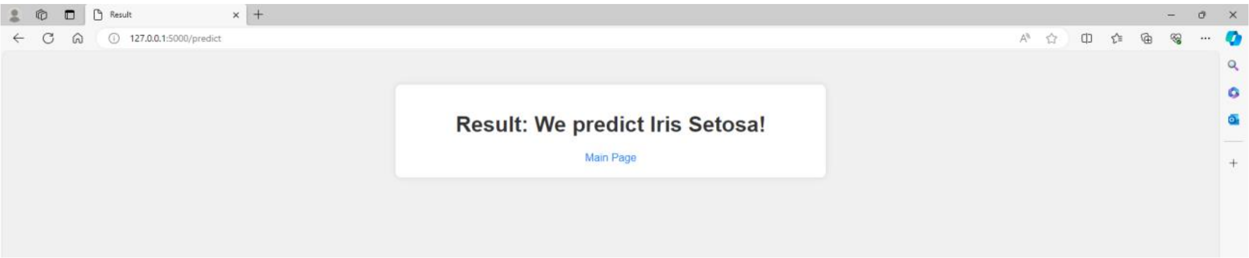
1.7

2

1

2.1

Results Page:



Result: We predict Iris Setosa!