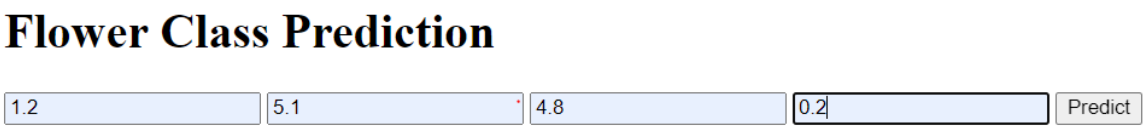
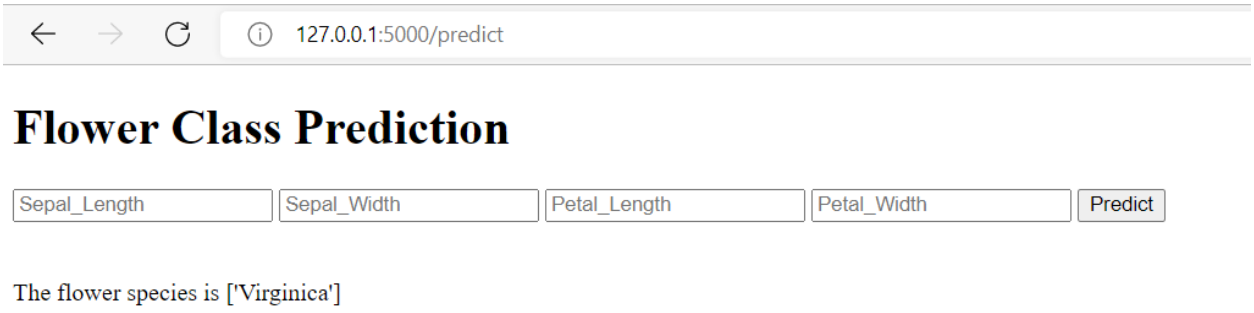
Name: Nahiyan Bin Noor
Internship Batch: LISUM09
Submission date: 2022-05-28
Submitted to: https://github.com/Nahiyan140212/Data-Glacier-Data-Science Internship/tree/main/Week%204

# Deployment on Flask

1. The user is provided with an interface where he/she can insert the features value.




2. After inserting the value, they can click on predict. Then it will show the predicted flower species.




A snapshot of the main code is provided below. The complete code is available on GitHub.

3. This is RandomForest Classifier model.

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import pickle

# Load the csv file
df = pd.read_csv("iris.csv")
```

```
print(df.head())

# Select independent and dependent variable
X = df[["Sepal_Length", "Sepal_Width", "Petal_Length", "Petal_Width"]]
y = df["Class"]

# Split the dataset into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=50)

# Feature scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test= sc.transform(X_test)

# Instantiate the model
classifier = RandomForestClassifier()

# Fit the model
classifier.fit(X_train, y_train)

# Make pickle file of our model
pickle.dump(classifier, open("model.pkl", "wb"))
```

This is the code for app

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

# Create flask app
flask_app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))

@flask_app.route("/")
def Home():
    return render_template("index.html")
@flask_app.route("/predict", methods = ["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)
    return render_template("index.html", prediction_text = "The flower
species is {}".format(prediction))

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

```
#The HTML code
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>
</head>

<body>
 <div class="login">
  <h1>Flower Class 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>


</body>
</html>
```