Data Glacier

Your Deep Learning Partner

DEPLOYMENT ON FLASK

NAME - JOHN ESHO
BATCH CODE - LISUM12
DATE - 10/09/2022
SUBMITTED TO - DATA GLACIER

DATA INFORMATION

The Iris dataset was used in R.A. Fisher's classic 1936 paper, The Use of Multiple Measurements in Taxonomic Problems, and can also be found on the UCI Machine Learning Repository.

It includes three iris species with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other.

Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa

INDEX.HTML

```html
<!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>
```

BUILD MODEL

- import libraries
- load dataset
- select independent and dependent variables
- split the dataset
- feature scaling
- instantiate model
- fit model
- create pickle file

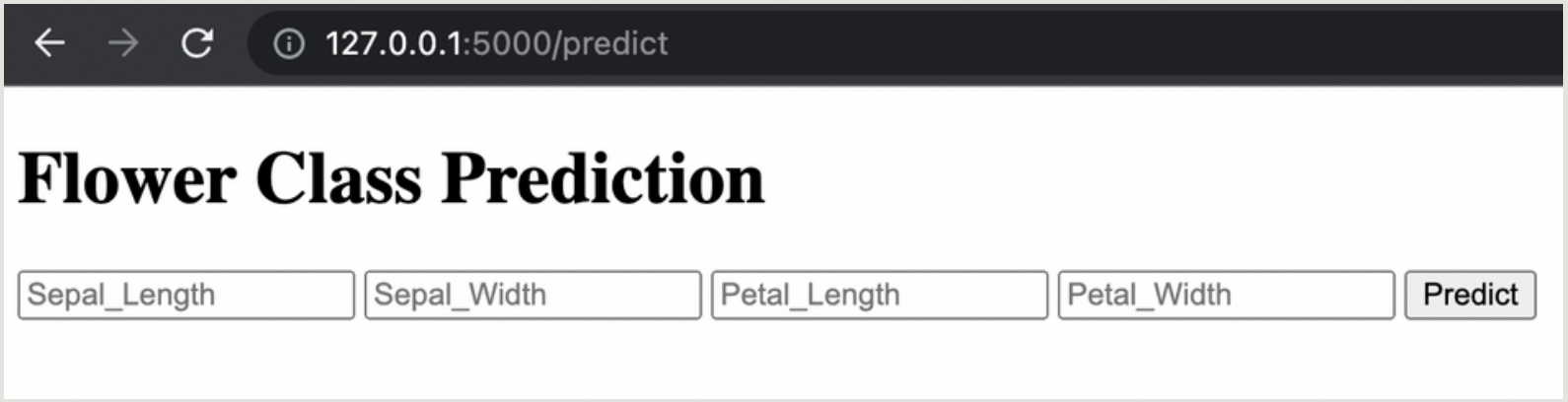
```python
# import libraries
import pickle
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

# load the dataset
df = pd.read_csv("Iris.csv")
# print(df)

# select independent and dependent variables
X = df[["SepalLengthCm", "SepalWidthCm", "PetalLengthCm", "PetalWidthCm"]]
y = df["Species"]


# split the dataset
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42)

# feature scaling
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)

# instantiate model
classifier = RandomForestClassifier()

# fit model
classifier.fit(X_train, y_train)


# create pickle file
pickle.dump(classifier, open("model.pkl", "wb"))
```

CREATE APP

- create flask app
- load pickle model

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

# create flask app
app = Flask(__name__)

# load the pickle model
model = pickle.load(open("model.pkl", "rb"))

@app.route("/")
def Home():
    return render_template("index.html")

@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__":
    app.run(debug=True)
```

WEB APP


127.0.0.1:5000/predict
Flower Class Prediction
Sepal_Length
Sepal_Width
Petal_Length
Petal_Width
Predict