NAME: TEHESUMA MENSAH IMORO

BATCH CODE: LISUM30 (30TH JANUARY 2024 – 30TH APRIL 2024)

SUBMISSION DATE: 4TH MARCH 2024

SUBMITTED TO: DATA GLACIER VIRTUAL INTERNSHIP

In [27]:
```
# Putting models in a dictionary
models = {"Logistic Regression": LogisticRegression(),
          "KNN": KNeighborsClassifier(),
          "Random Forest": RandomForestClassifier()}

# Creating a function to fit and score models
def fit_and_score(models, X_train, X_test, y_train, y_test):
    """
    Fits and evaluates given machine learning models.
    models : a dict of differetn Scikit-Learn machine learning models
    X_train : training data (no labels)
    X_test : testing data (no labels)
    y_train : training labels
    y_test : test labels
    """
    # Setting random seed
    np.random.seed(42)
    # Making a dictionary to keep model scores
    model_scores = {}
    # Looping through models
    for name, model in models.items():
        # Fitting the model to the data
        model.fit(X_train, y_train)
        # Evaluating the model and append its score to model_scores
        model_scores[name] = model.score(X_test, y_test)
    return model_scores
```

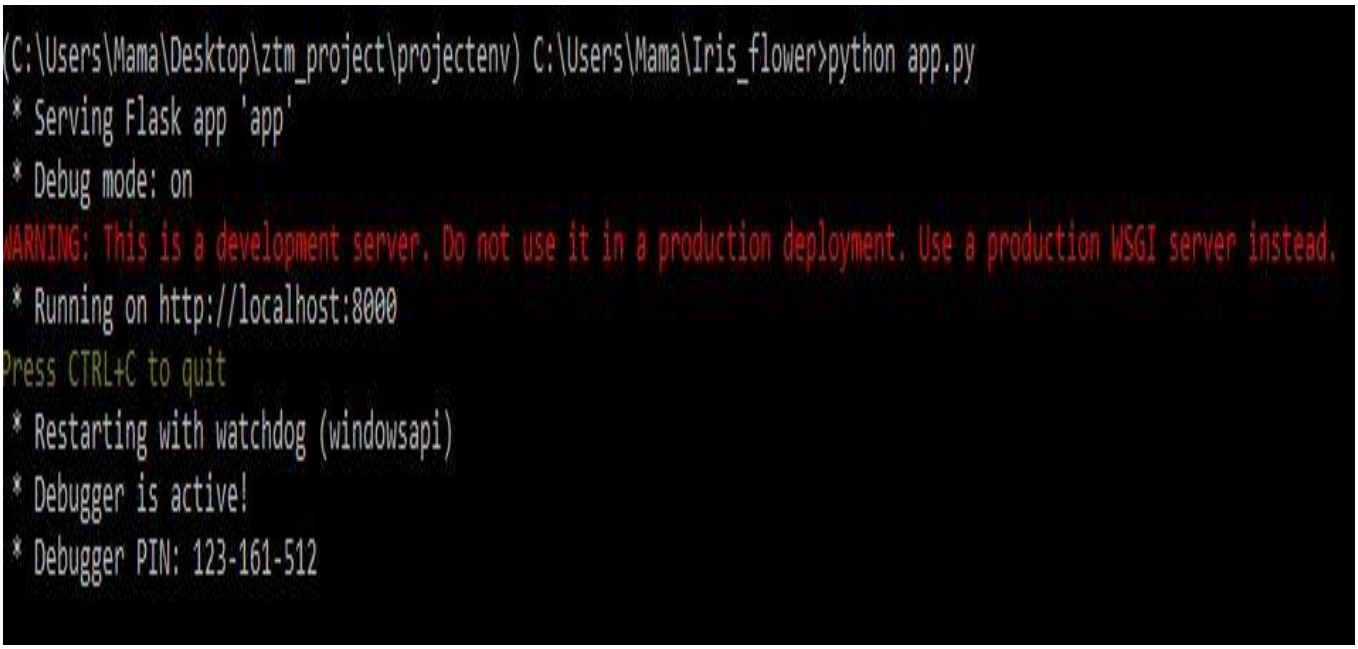
In [28]:
```
model_scores = fit_and_score(models=models,
                             X_train=X_train,
                             X_test=X_test,
                             y_train=y_train,
                             y_test=y_test)

model_scores
```

Out[28]:
```
{'Logistic Regression': 0.9736842105263158,
 'KNN': 0.9736842105263158,
 'Random Forest': 0.9736842105263158}
```

In [31]:
```
# Saving our preferred model as a pickle file
pickle.dump(model,open('iris.pkl','wb'))

# Loading our model
ideal_model = pickle.load(open('iris.pkl','rb'))
```

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


app = Flask(__name__)
model = pickle.load(open('iris.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('iris.html', **locals())

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

    result = model.predict([[sepal_length, sepal_width, petal_length, petal_width]])[0]
    return render_template('iris.html', **locals())

if __name__ == "__main__":
    app.run(host='localhost', port=8000, debug=True)
```

```html
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
  <meta charset="UTF-8">
  <title>Iris Flower Predictor</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>Iris Flower Species Prediction</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="/predict" method="POST">
      Sepal length: <input type='text' name="sepal_length"><br>
        Sepal width: <input type='text' name="sepal_width"><br>
        Petal length: <input type='text' name="petal_length"><br>
        Petal width: <input type='text' name="petal_width"><br>
        Species: {{result}} <br>
        <input type="submit" value="Predict">
    </form>
```


(C:\Users\Mama\Desktop\ztm_project\projectenv) C:\Users\Mama\Iris_flower>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://localhost:8000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 123-161-512

Iris Flower Species Prediction

Sepal length:
Sepal width:
Petal length:
Petal width:
Species: iris-virginica
Predict