Data Intake Report

Name: Iris Dataset Deployment

Report date: 19/06/2022

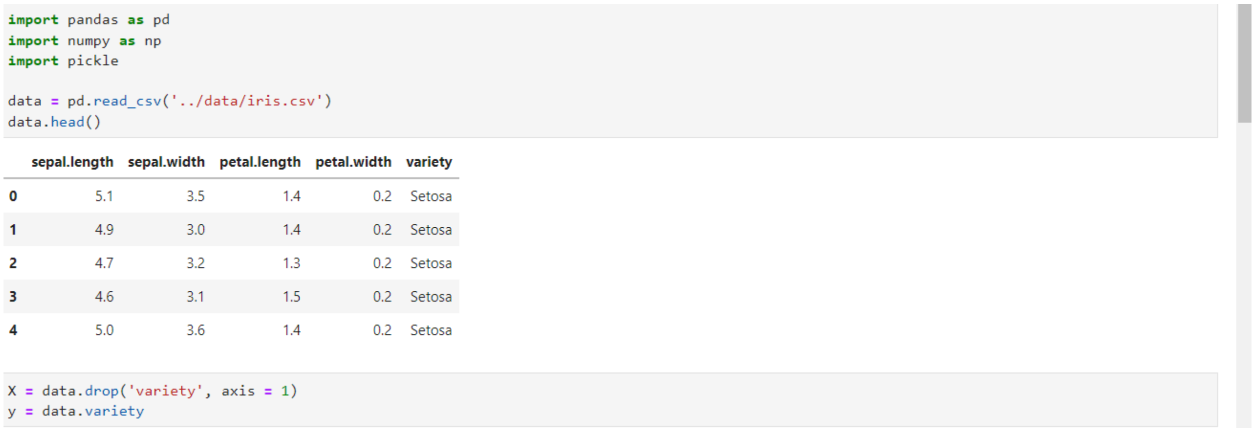
Internship Batch: LISUM10: 30

Data intake by: Amr Elbana

Data intake reviewer: Data-Glacier

Steps

1) Load Data and select the dependent and independent variables.

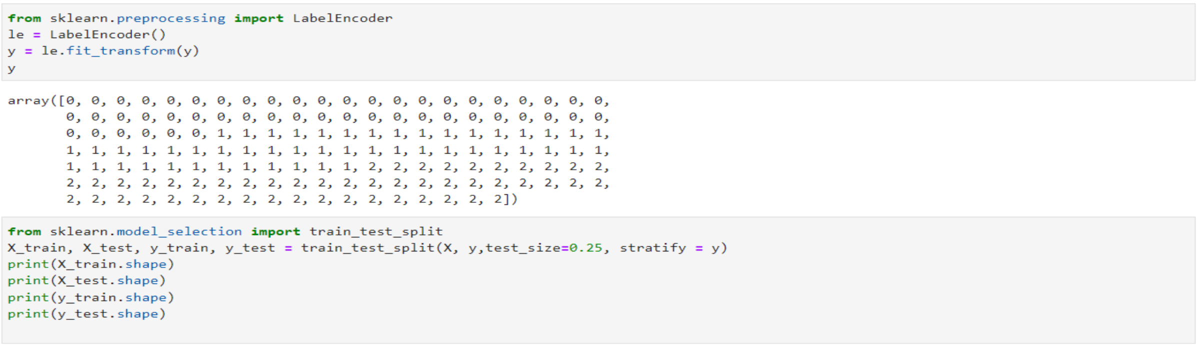


```
import pandas as pd
import numpy as np
import pickle

data = pd.read_csv('../data/iris.csv')
data.head()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa

```
X = data.drop('variety', axis = 1)
y = data.variety
```

2) Label encoding the target variable , then split the data



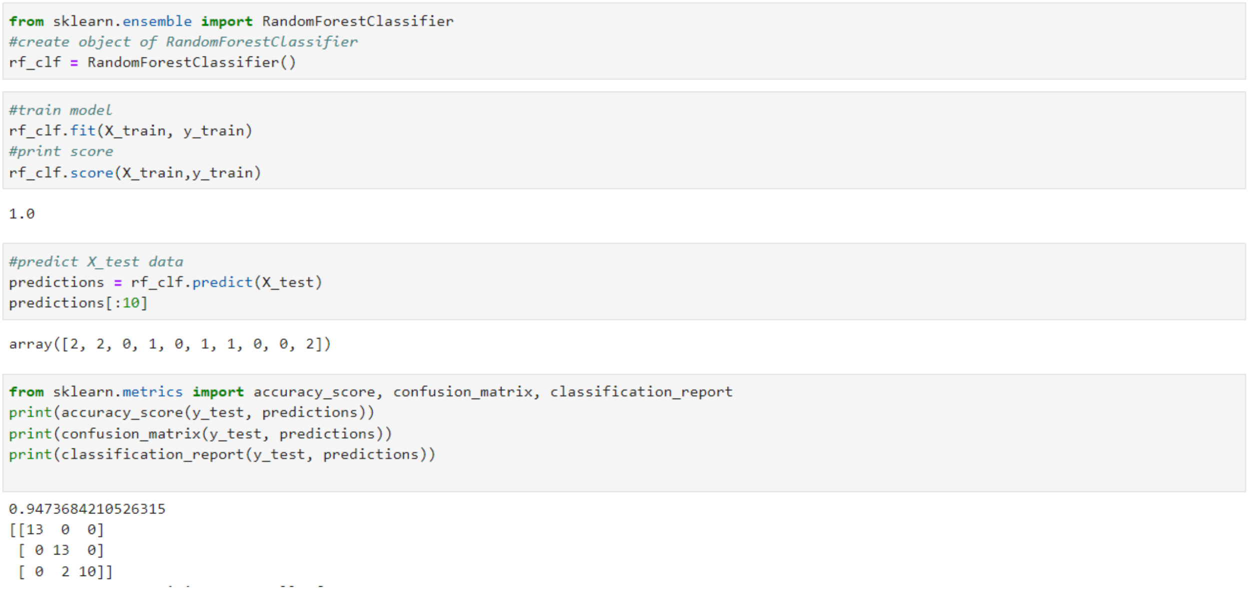
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
y
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.25, stratify = y)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

3) Train Model and check accuracy and make prediction

a) Random Forest classifier



```
from sklearn.ensemble import RandomForestClassifier
#create object of RandomForestClassifier
rf_clf = RandomForestClassifier()
```

```
#train model
rf_clf.fit(X_train, y_train)
#print score
rf_clf.score(X_train,y_train)
```

```
1.0
```

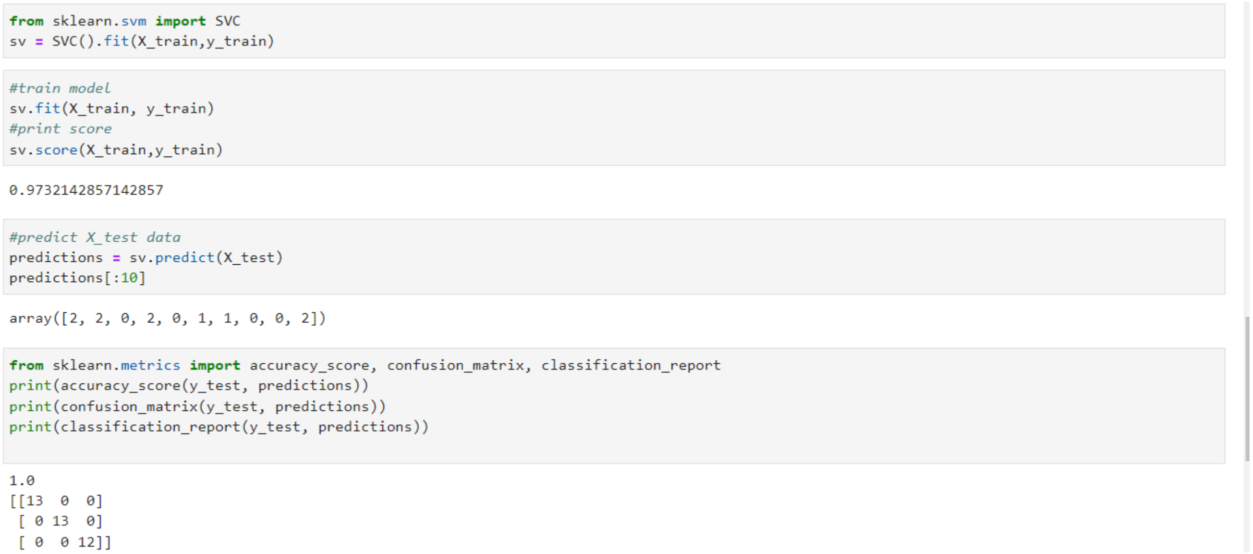
```
#predict X_test data
predictions = rf_clf.predict(X_test)
predictions[:10]
```

```
array([2, 2, 0, 1, 0, 1, 1, 0, 0, 2])
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print(accuracy_score(y_test, predictions))
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
```

```
0.9473684210526315
[[13  0  0]
 [ 0 13  0]
 [ 0  2 10]]
```

b) Support Vector Machine SVM:



```
from sklearn.svm import SVC
sv = SVC().fit(X_train,y_train)
```

```
#train model
sv.fit(X_train, y_train)
#print score
sv.score(X_train,y_train)
```

```
0.9732142857142857
```

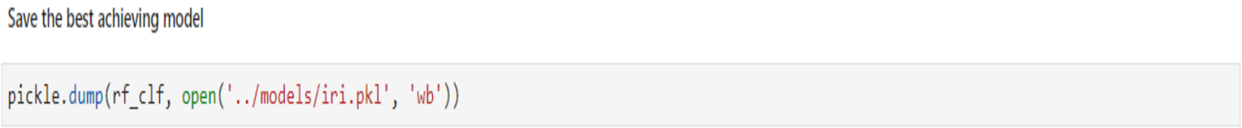
```
#predict X_test data
predictions = sv.predict(X_test)
predictions[:10]
```

```
array([2, 2, 0, 2, 0, 1, 1, 0, 0, 2])
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print(accuracy_score(y_test, predictions))
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
```

```
1.0
[[13  0  0]
 [ 0 13  0]
 [ 0  0 12]]
```

4) Save the best performing model

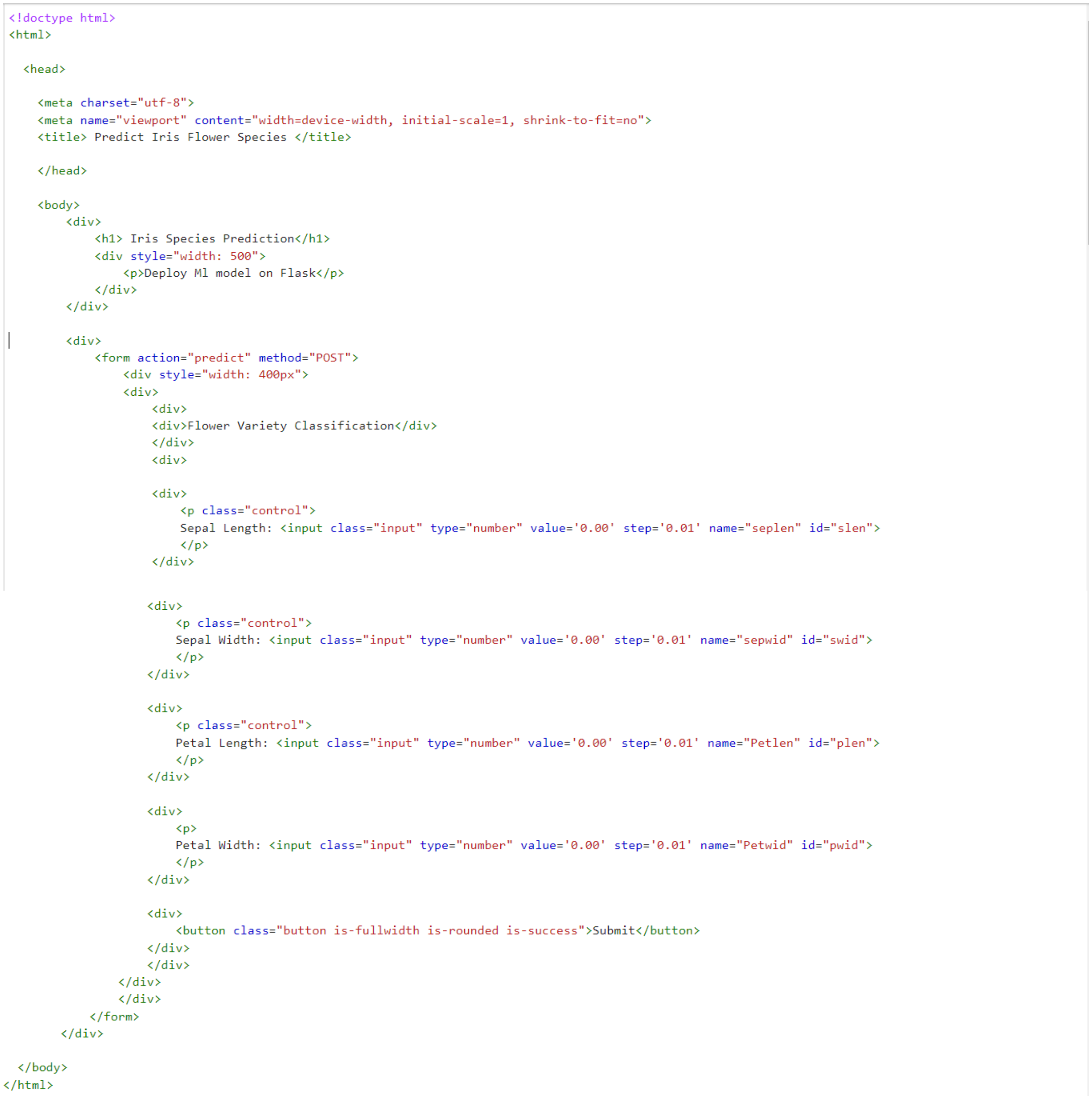


Save the best achieving model

```
pickle.dump(rf_clf, open('../models/iri.pkl', 'wb'))
```

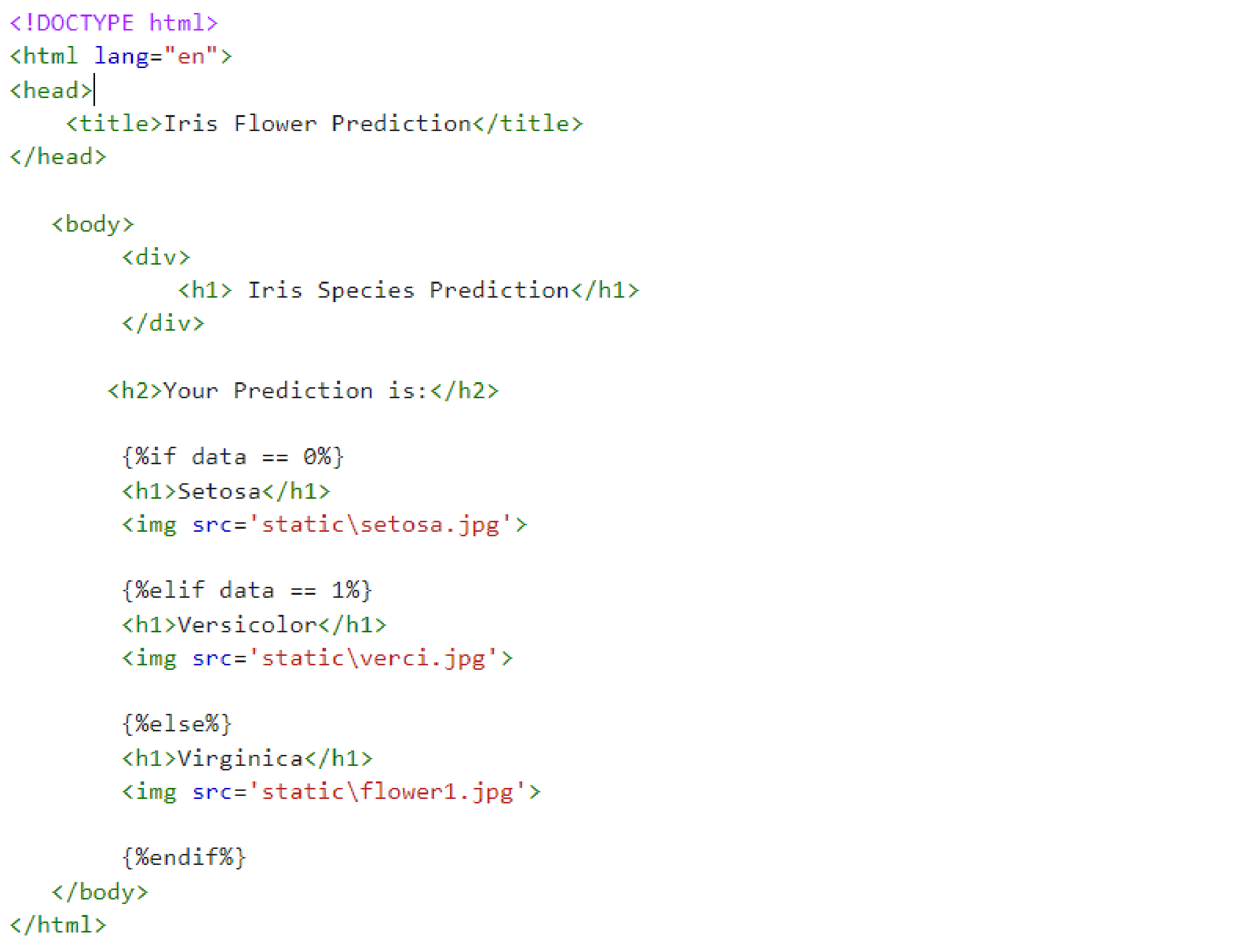
5) Create template HTML files.

a) Home.html



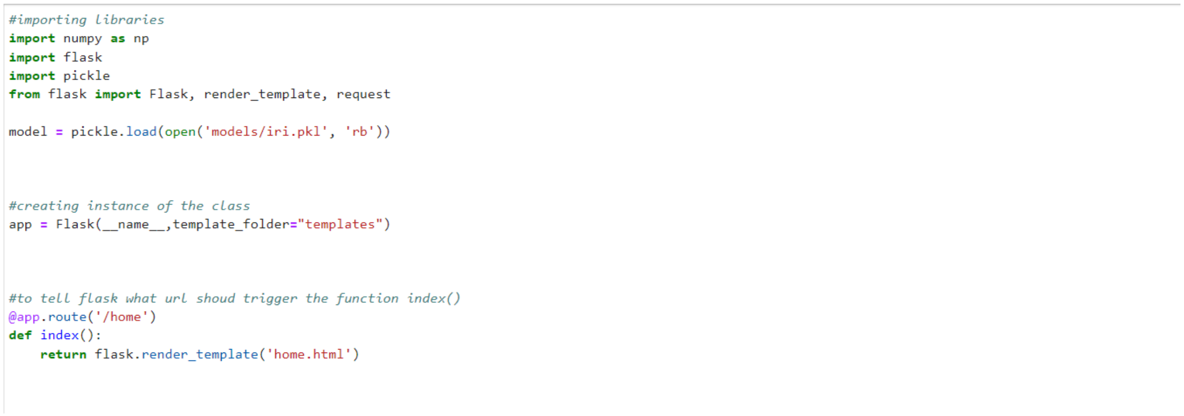
```
<!doctype html>
<html>

  <head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title> Predict Iris Flower Species </title>

    </head>

    <body>
        <div>
            <h1> Iris Species Prediction</h1>
            <div style="width: 500">
                <p>Deploy Ml model on Flask</p>
            </div>
        </div>

        <div>
            <form action="predict" method="POST">
                <div style="width: 400px">
                <div>
                    <div>
                    <div>Flower Variety Classification</div>
                    </div>
                    <div>

                    <div>
                        <p class="control">
                        Sepal Length: <input class="input" type="number" value='0.00' step='0.01' name="seplen" id="slen">
                        </p>
                    </div>

                    <div>
                        <p class="control">
                        Sepal Width: <input class="input" type="number" value='0.00' step='0.01' name="sepwid" id="swid">
                        </p>
                    </div>

                    <div>
                        <p class="control">
                        Petal Length: <input class="input" type="number" value='0.00' step='0.01' name="Petlen" id="plen">
                        </p>
                    </div>

                    <div>
                        <p>
                        Petal Width: <input class="input" type="number" value='0.00' step='0.01' name="Petwid" id="pwid">
                        </p>
                    </div>

                    <div>
                        <button class="button is-fullwidth is-rounded is-success">Submit</button>
                    </div>
                    </div>
                </div>
                </div>
            </form>
        </div>

  </body>
</html>
```

b) Predict.html, the prediction page

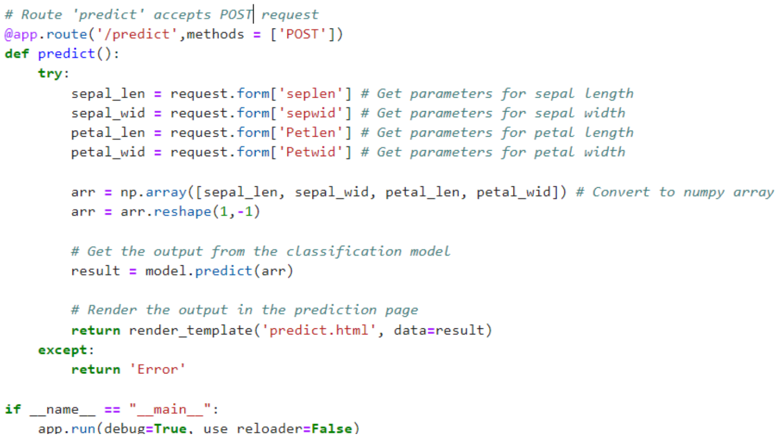


```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Iris Flower Prediction</title>
</head>

   <body>
        <div>
            <h1> Iris Species Prediction</h1>
        </div>

       <h2>Your Prediction is:</h2>

        {%if data == 0%}
        <h1>Setosa</h1>
        <img src='static\setosa.jpg'>

        {%elif data == 1%}
        <h1>Versicolor</h1>
        <img src='static\verci.jpg'>

        {%else%}
        <h1>Virginica</h1>
        <img src='static\flower1.jpg'>

        {%endif%}
   </body>
</html>
```

6) App.py

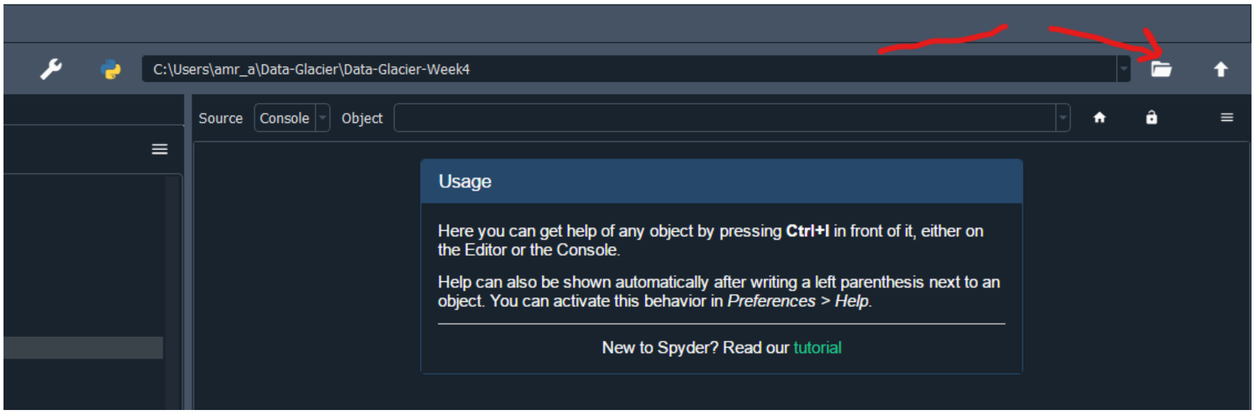


```
#importing libraries
import numpy as np
import flask
import pickle
from flask import Flask, render_template, request

model = pickle.load(open('models/iri.pkl', 'rb'))



#creating instance of the class
app = Flask(__name__,template_folder="templates")



#to tell flask what url shoud trigger the function index()
@app.route('/home')
def index():
    return flask.render_template('home.html')
```



```
# Route 'predict' accepts POST request
@app.route('/predict',methods = ['POST'])
def predict():
    try:
        sepal_len = request.form['seplen'] # Get parameters for sepal length
        sepal_wid = request.form['sepwid'] # Get parameters for sepal width
        petal_len = request.form['Petlen'] # Get parameters for petal length
        petal_wid = request.form['Petwid'] # Get parameters for petal width

        arr = np.array([sepal_len, sepal_wid, petal_len, petal_wid]) # Convert to numpy array
        arr = arr.reshape(1,-1)

        # Get the output from the classification model
        result = model.predict(arr)

        # Render the output in the prediction page
        return render_template('predict.html', data=result)
    except:
        return 'Error'

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

7) Open Spyder.
   a) Open the app.py
   b) Set the environment in Spyder to be the location of your folder as followed: follow the red line.



C:\Users\amr_a\Data-Glacier\Data-Glacier-Week4

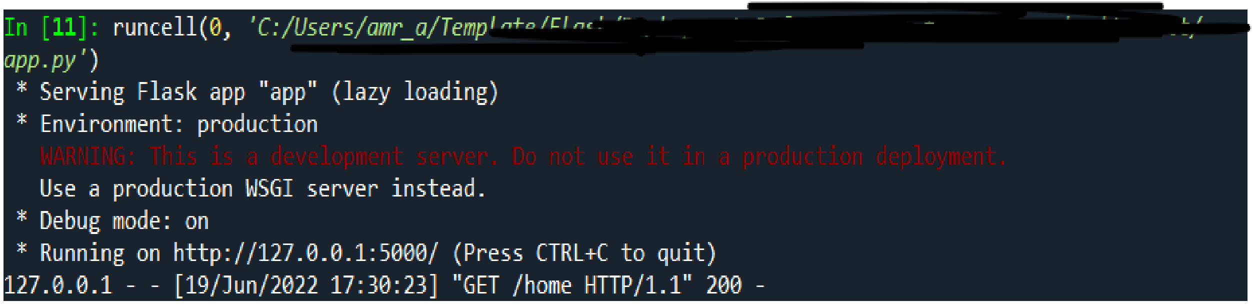
Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

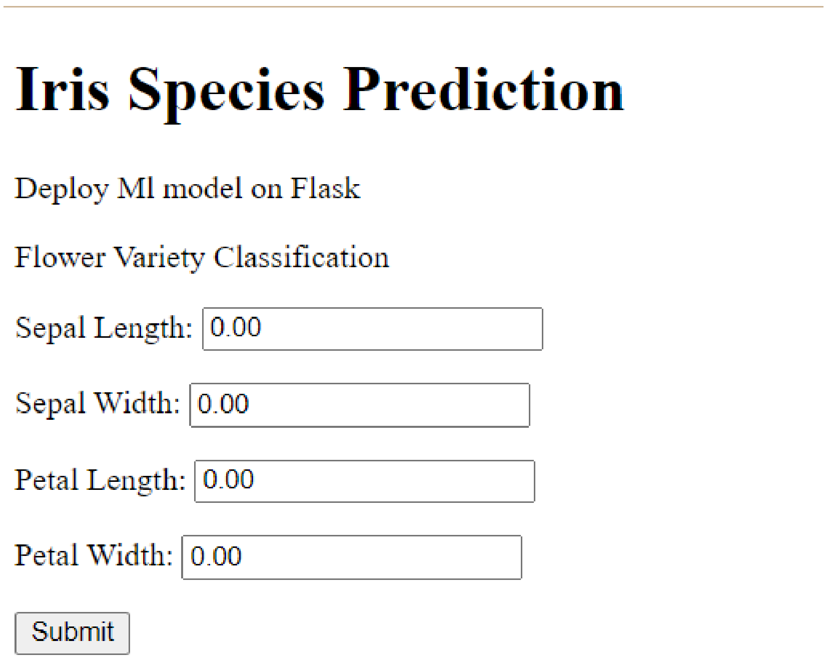
New to Spyder? Read our tutorial

8) Run app.py file, it will give you an IP address



```
In [11]: runcell(0, 'C:/Users/amr_a/Temp[redacted]
app.py')
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [19/Jun/2022 17:30:23] "GET /home HTTP/1.1" 200 -
```

Get the Ip next to “Running on”, put it next to the URL and add the location of the home page. (e.g. http://127.0.0.1:5000/home)



Iris Species Prediction

Deploy Ml model on Flask

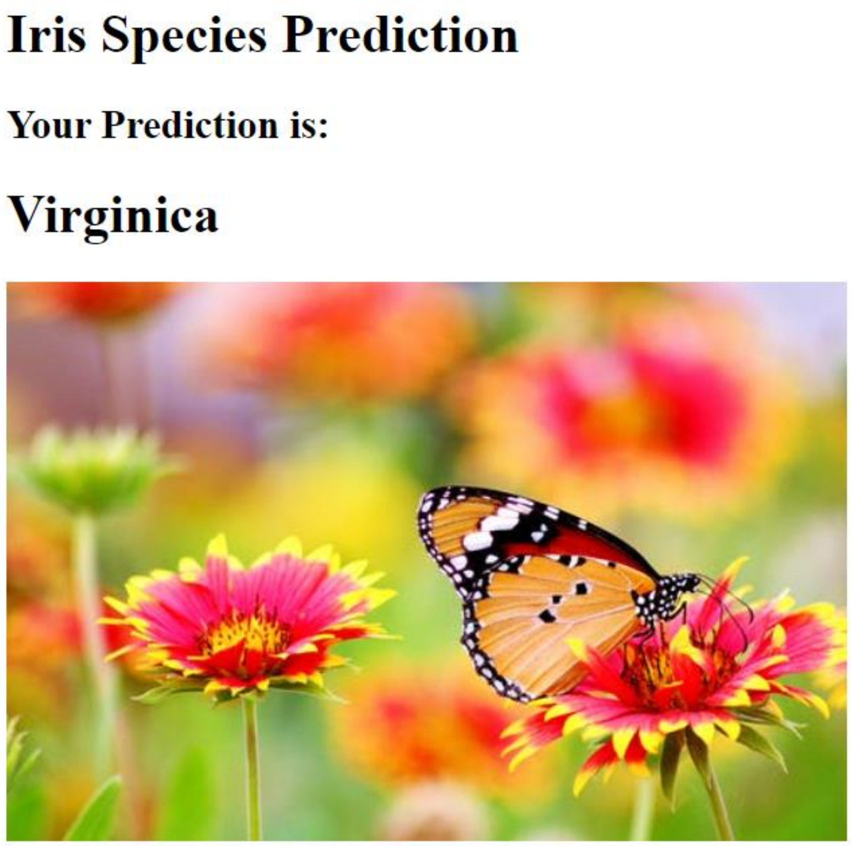
Flower Variety Classification

Sepal Length: 0.00

Sepal Width: 0.00

Petal Length: 0.00

Petal Width: 0.00



Iris Species Prediction

Your Prediction is:

Virginica