

Week4: Deployment on Flask

Name: *Sevde Büşra Bayrak*

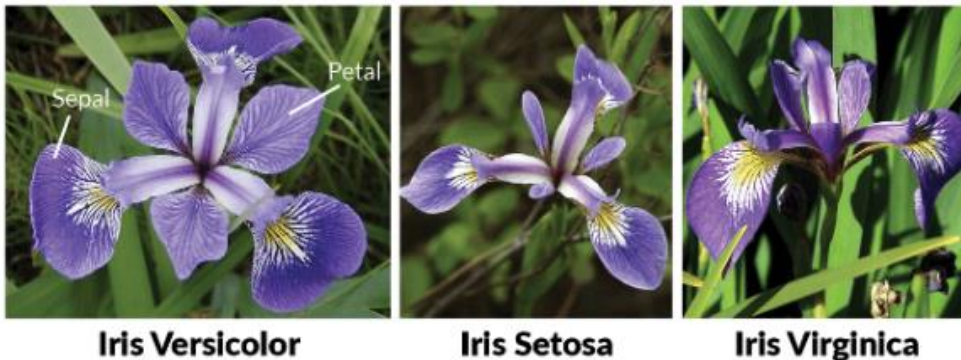
Report date: 23 May 2022

Internship Batch: *LISUM04*

Data storage location: <https://github.com/sevdebayrak94/FlaskDeployment>

1.Dataset Desription

In this project the well-known dataset, which is called iris dataset, was used. The iris dataset contains three different species of Iris (Iris setosa, Iris virginica, and Iris versicolor) with four different features (length and width of petals and sepals). There are 50 samples for each species. This dataset were used to create Flask app by using SVM classification algorithm. Here pictures of species.



2.Required libraries must download

First install Flask: pip install Flask, pickle to create a Flask 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"))
```

Create root and check localhost:5000

```
* 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)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 108-712-055
127.0.0.1 - - [23/May/2022 17:44:23] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/May/2022 17:44:23] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [23/May/2022 17:44:31] "POST /predict HTTP/1.1" 200 -
```

Create an app.py

```
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 index():
    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", result_of_prediction = "The flower species is {}".format(prediction))

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

3. Create SVM Model

With additional required libraries do not forget to create a model.py to fit SVM model to make prediction. Additional files like index.html was supported in my github page.

```

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import pickle
from sklearn.svm import SVC

# Load the csv file
data_file = pd.read_csv(r"C:\Users\casper\Desktop\iris.csv")
my_data= data_file[["sepal.length", "sepal.width", "petal.length", "petal.width"]]
my_label = data_file["variety"]

# Split the dataset into train and test
my_data_train, my_data_test, my_label_train, my_label_test = train_test_split(my_data, my_label, test_size=0.3, random_state=42)

# The model
SVM_model = SVC()
# Fit the model
SVM_model.fit(my_data_train, my_label_train)

#Let's Make pickle file of our model
pickle.dump(SVM_model, open("model.pkl", "wb"))
print(SVM_model.score(my_data_test, my_label_test))

```

4. Go

Flower Class Prediction

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Predict
--------------	-------------	--------------	-------------	---------