

Name: Chepkorir Beril

Batch Code: LISP01

Submission Date: 19/03/2021

Submitted to: <https://github.com/dangerous2>

DEPLOYING MACHINE LEARNING MODEL

This is a project to elaborate how machine learning are deployed on production using flask

Project structure

Prerequisites

You must have Scikit Learn, Pandas (for Machine Learning Model) and Flask (for API) installed.

Project Structure

This project has four major parts:

1. iris.py - This contains code for our Machine Learning model to predict iris flower based on data in 'Iris.csv' file.
2. app.py - This contains Flask APIs that receives iris details through GUI or API calls, computes the predicted value based on our model and returns it.
3. basics.py - Calls APIs already defined in app.py and displays the returned value.

4. templates - This folder contains the HTML template to allow user to enter iris flower details and displays the predicted iris flower.

Running the project

1. Ensure that you are in the project home directory. Create the machine learning model by running below command -

```
'''
```

```
python iris.py
```

```
'''
```

This would create a serialized version of our model into a file iris.pkl

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

df = pd.read_csv('Iris.csv')

X = df.drop('Species', axis=1)
y = df.Species

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

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

from sklearn.svm import SVC
sv = SVC(kernel='linear').fit(X_train,y_train)

pickle.dump(sv, open('iris.pkl', 'wb'))
```

2. Run app.py using below command to start Flask API

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

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

app = Flask(__name__)

@app.route('/')
def man():
    return render_template('home.html')

@app.route('/predict', methods=['POST'])
def home():
    data1 = request.form['a']
    data2 = request.form['b']
    data3 = request.form['c']
    data4 = request.form['d']
    arr = np.array([[data1, data2, data3, data4]])
    pred = model.predict(arr)
    return render_template('after.html', data=pred)

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

...

python app.py

...

```
base) C:\Users\Beyrylle\Desktop\flask>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
* Restarting with stat
* Debugger is active!
* Debugger PIN: 334-243-063
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Enter valid numerical values in all 3 input boxes and hit Predict

By default, flask will run on port 5000.

3. Navigate to URL <http://localhost:5000>

You should be able to view the homepage as below:

IRIS FLOWER DETECTION

First value :

Second value :

Third value :

Fourth value :

A photograph of a monarch butterfly with orange and black wings, perched on a pink and yellow flower. The background is a soft-focus field of similar flowers.

Enter valid float values in all 4 input boxes and hit Predict.

If everything goes well, you should be able to see the predicted flower on the HTML page!

PREDICTION :

Iris-setosa

