

Flask(플라스크) 활용한_머신러닝_웹서비스구축

Flask(플라스크) 활용한 머신러닝 웹 서비스 구축

학습 목표

가. Flask를 활용한 머신러닝 웹 서비스를 구현할 수 있다.

학습 내용

- 가. 가상환경 만들기
- 나. 필수 라이브러리 설치
- 다. 전체 폴더 구조
- 라. iris.py 소스 코드
- 마. flask_app 소스 코드
- 사. home.html 소스 코드
- 아. after.html 소스 코드
- 자. 실행

목차

가. 가상환경 만들기

```
conda create -n flask_ml python=3.8
conda activate flask_ml
```

나. 필수 라이브러리 설치

[방법1] requirements.txt 가 있을 경우,

```
pip install -r requirements.txt
```

requirements.txt 내용

```
Flask
Jinja2
scikit-learn
scipy
virtualenv
pillow
imageio
opencv-python
numpy
```

[방법2]

```
pip install flask==2.1.3 Jinja2
pip install scikit-learn
pip install numpy==1.23.1
pip install virtualenv
pip install pillow
pip install seaborn
pip install imageio
pip install opencv-python
pip install pandas
```

[리스트 확인]

```
conda list
```

```
(flask_ml) C:\WUsers\Wtotofriend>conda list
# packages in environment at C:\WUsers\Wtotofriend\Wanaconda3\Wenvs\Wflask_ml:
#
# Name                   Version           Build Channel
ca-certificates         2022.07.19        haa95532_0
certifi                 2022.9.24         py38haa95532_0
click                   8.1.3             pypi_0 pypi
colorama                0.4.5             pypi_0 pypi
contourpy               1.0.5             pypi_0 pypi
cyclor                  0.11.0            pypi_0 pypi
distlib                 0.3.6             pypi_0 pypi
filelock                3.8.0             pypi_0 pypi
flask                   2.1.3             pypi_0 pypi
fonttools               4.37.4            pypi_0 pypi
imageio                 2.22.1            pypi_0 pypi
importlib-metadata      5.0.0             pypi_0 pypi
itsdangerous            2.1.2             pypi_0 pypi
jinja2                  3.1.2             pypi_0 pypi
```

joblib	1.2.0	pypi_0	pypi
kiwisolver	1.4.4	pypi_0	pypi
markupsafe	2.1.1	pypi_0	pypi
matplotlib	3.6.1	pypi_0	pypi
numpy	1.23.1	pypi_0	pypi
opencv-python	4.6.0.66	pypi_0	pypi
openssl	1.1.1q	h2bbff1b_0	
packaging	21.3	pypi_0	pypi
pandas	1.5.0	pypi_0	pypi
pillow	9.2.0	pypi_0	pypi
pip	22.2.2	py38haa95532_0	
platformdirs	2.5.2	pypi_0	pypi
pyparsing	3.0.9	pypi_0	pypi
python	3.8.13	h6244533_0	
python-dateutil	2.8.2	pypi_0	pypi
pytz	2022.4	pypi_0	pypi
scikit-learn	1.1.2	pypi_0	pypi
scipy	1.9.2	pypi_0	pypi
seaborn	0.12.0	pypi_0	pypi
setuptools	63.4.1	py38haa95532_0	
six	1.16.0	pypi_0	pypi
sqlite	3.39.3	h2bbff1b_0	
threadpoolctl	3.1.0	pypi_0	pypi
vc	14.2	h21ff451_1	
virtualenv	20.16.5	pypi_0	pypi
vs2015_runtime	14.27.29016	h5e58377_2	
werkzeug	2.2.2	pypi_0	pypi
wheel	0.37.1	pyhd3eb1b0_0	
wincertstore	0.2	py38haa95532_2	
zipp	3.9.0	pypi_0	pyp

다. 전체 폴더 구조

[리스트 확인]

```

----[폴더명]
-----[폴더] model
-----[폴더] static
-----[폴더] templates
-----iris.py
-----flask_app.py

```

iris.py : 파일을 불러오고, 실제 모델 구축 후, 이를 pickle 파일로 저장한다.
 flask_app.py : iris.py에서 생성한 pickle파일을 불러온다. Flask App을 띄운다.

----- [폴더] model : 학습된 모델 저장할 폴더
 (예) iris_knn.pkl

----- [폴더] static : 참조할 이미지 폴더. 이곳에서 이미지를 불러와 웹에서 보여주게 된다.
 (예) flower_iris.jpg, iris_setosa.jpg, iris_versicolor.jpg, iris_virginica.jpg

----- [폴더] templates : 첫 웹 페이지와 데이터 입력 후, 실행될 웹 html 파일 저장 폴더
 home.html
 after.html

라. iris.py 소스 코드

```

import time
import pickle
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

start_time = time.time()

# 01 데이터 불러오기
df = sns.load_dataset("iris")
print( df.shape )
print( df.columns )
print( df.head() )

sel = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']

# 02 라벨 인코딩(Label Encoding)
le = LabelEncoder()
y_lbl = le.fit_transform(df['species'])

```

```
# 03 데이터 나누기 - 입력, 출력
X = df[sel]
y = y_lbl

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2, random_state=77)

# 04 모델 구축 및 학습, 모델 저장
knn = KNeighborsClassifier().fit(X_train, y_train)
print(knn.score(X_test, y_test))
pickle.dump(knn, open('./model/iris_knn.pkl', 'wb'))

print("시간 (초): ", time.time() - start_time )
```

마. flask_app 소스 코드

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

# 01 학습된 모델 가져오기
model = pickle.load(open('./model/iris_knn.pkl', 'rb'))

# 02 플라스크 사용을 위한 준비
app = Flask(__name__)

# 03 플라스크 앱의 루트 디렉터리를 초기화
@app.route('/')
def main():
    return render_template('start.html')

# 04 초기 웹 페이지에서 submit 했을 때 실행
# request.form[""]을 사용하여 HTML 페이지에서 입력한 데이터를 가져온다.
# model.predict()를 통해 클래스를 예측한다.
# 예측값에 따라 어떤 텍스트와 이미지를 보낼지, after.html에 설정.
@app.route('/predict', methods=['POST'])
def home():
    val1 = request.form['a']
    val2 = request.form['b']
    val3 = request.form['c']
    val4 = request.form['d']
    val1 = float(val1); val2 = float(val2); val3 = float(val3); val4 = float(val4);
    arr = np.array([[val1, val2, val3, val4]])
    pred = model.predict(arr)

    # 렌더링할 html 파일명, 전달할 변수
    return render_template('after.html', data=pred)

# 05 직접 실행된 경우, 앱을 디버그 on 모드로 실행
if __name__ == "__main__":
    app.run(debug=True)
```

사. start.html 소스 코드

```
<html>
<body bgcolor=#d4a3ae>
  <center>
    <h1> IRIS Species Predict </h1>

    <form method="POST", action="{{url_for('home')}}">
      <b> sepal_length(cm) : <input type="text", name='a', placeholder="enter 1">

        sepal_width(cm) : <input type="text", name='b', placeholder="enter 2">

        petal_length(cm) : <input type="text", name='c', placeholder="enter 3">

        petal_width(cm) : <input type="text", name='d', placeholder="enter 4">

    </b>
    <input type="submit" , value='predict!' >
  </form>
  <img src='staticWiris_flower2.png' alt="flower" width="500" height="400">
</center>
```

```
</body>
</html>
```

아. after.html 소스 코드

```
<html>
<body bgcolor=#a3cfb4>
  <center>
    <h1> PREDICTION : </h1>
    {%if data == 0%}
    <h1>Iris-setosa</h1>
    <img src='static\iris_setosa.jpg'>
    {%endif%}

    {%if data == 1%}
    <h1>Iris-versicolor</h1>
    <img src='static\iris_versicolor.jpg'>
    {%endif%}

    {%if data == 2%}
    <h1>Iris-virginica</h1>
    <img src='static\iris_virginica.jpg'>
    {%endif%}

    <a href='/'>go back to home page</a>
  </center>
</body>
</html>
```

flask에서 if 문을 사용

```
{%if (조건식) %}
--
{%endif%}

{%if (조건식) %}
--
{%endif%}

{%if (조건식) %}
--
{%endif%}
```

자. 실행

01. iris.py를 실행
02. flask_app.py를 실행

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 140-126-807
```

03. 웹페이지 URL을 웹 사이트에 입력
<http://127.0.0.1:5000>

IRIS Species Predict

sepal_length(cm) :

sepal_width(cm) :

petal_length(cm) :

petal_width(cm) :

iris setosa



petal

sepal

iris versicolor



petal

sepal

iris virginica



petal

sepal

