

Flask - POST

1. POST 예제 연습

```
from flask import Flask, request, jsonify
# 백엔드로 통신해야 하기 때문에 request를 import
# 통신은 json으로 하기 때문에 jsonify를 import
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score

app = Flask(__name__)

learn_data = [[0, 0], [1, 0], [0, 1], [1, 1]]
learn_label = [0, 0, 0, 1] # 실질적인 정답

# 객체 생성
svc = LinearSVC()

# 학습
svc.fit(learn_data, learn_label)

# Get 방식이 아닌 POST 방식
@app.route('/predict', methods=['POST'])
def predict():
    # post 방식에는 postman 사용
    data = request.json['data']
    pred = svc.predict(data)
    acc = accuracy_score(learn_label, pred)
    return jsonify(acc) # json 형식으로 보낸다.

if __name__ == '__main__':
    app.run(debug=True) # Flask를 띄운다.
```

Figure 1. POST 부분 코드

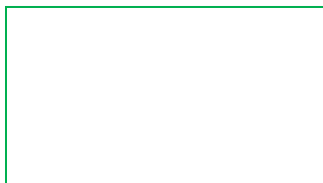


-> Flask 클래스의 인스턴스를 생성한다.

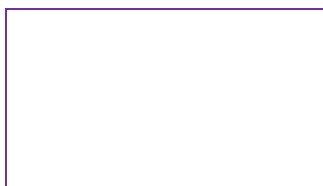
-> Python에서 Flask 웹 애플리케이션을 생성하는 코드



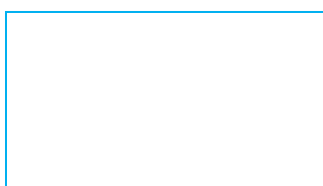
-> 학습시킬 데이터와 label



-> LinearSVC 객체 생성 및 data와 label 학습



-> py 파일 자체에서 실행을 할 경우 app.py 파일을 실행한다.



-> nodeJS에서는 router에 해당하는 부분

-> 이전과는 달리 methods를 'POST'로 설정하였다.

-> POST를 사용할 때는 POSTMAN으로 제대로 설정이 되었는지 확인을 해준다.

-> data = request.json['data']에서 Figure2를 보면 request.json['data']는 Figure 2를 보면 json 형태로 key의 이름이 data인 value 값들을 data라는 변수에 넣는다는 의미이다.

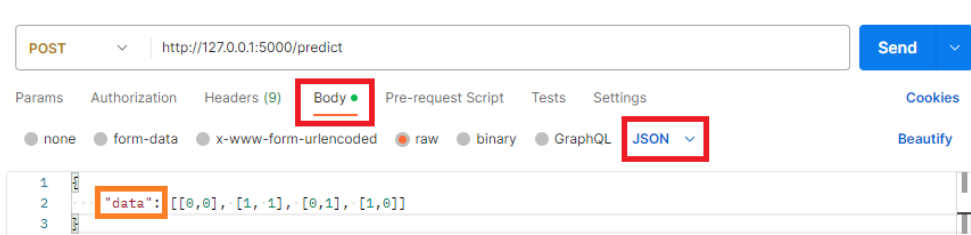


Figure 2. POSTMAN에서 POST 부분

-> jsonify 부분을 보면 flask에서는 결과가 json 형태 말고는 나오지 못하기 때문에 jsonify를 이용해 결과를 json 형태로 바꿔준다.

2. POST 실전 연습

Q: 13. pytorch로 구현한 선형회귀에서 나온 다중회귀 예시를 pth 모델로 저장한 다음 Flask에서 실행하기

```
from flask import Flask, request, jsonify
import torch
import torch.nn as nn

app = Flask(__name__)

# 모델 로드
model = nn.Linear(3, 1)
model.load_state_dict(torch.load('saved_model.pth'))
model.eval()

@app.route('/predict', methods=['POST'])
def predict():
    data = torch.FloatTensor(request.json['data']) #

    # 모델에 입력 데이터 전달하여 예측 수행
    with torch.no_grad():
        output = model(data)

    # 예측 결과 반환 (예를 들어 JSON 형식으로 반환)
    return jsonify({'prediction': output.tolist()})

if __name__ == '__main__':
    app.run()
```

Figure 1. POST 실전 버전 코드



-> Flask 클래스의 인스턴스를 생성한다.

-> Python에서 Flask 웹 애플리케이션을 생성하는 코드



-> pth 모델을 불러온다.

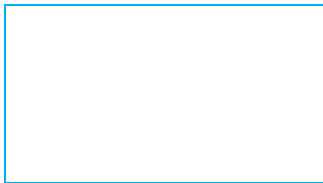
-> 모델을 불러오기(load) 전에 같은 형태의 임시 모델을 미리 정의 해주어야 한다. (model = nn.Linear(3, 1))



-> nodeJS에서는 router에 해당하는 부분

-> 이전과는 달리 methods를 'POST'로 설정하였다.

-> POST를 사용할 때는 POSTMAN으로 제대로 설정이 되었는지 확인을 해준다.



-> py 파일 자체에서 실행을 할 경우 app.py 파일을 실행한다.