8.4 – knn 모델은 학습한 데이터를 모두 저장하는가?

=> KNN 모델을 저장해보아서 학습데이터가 적을때와 많을 때의 파일 크기 달라지고 KNN 모델안에 학습데이터를 저장하고 있는지를 확인이 불가능하다.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

L = 400          # 작은 수를 사용하면 프린트하기 수월하다.
data = np.random.randint(0, 100, (L, 2)).astype(np.float32)
labels = np.random.randint(0, 2, (L, 1)).astype(np.float32)
print('data for training:', type(data), data.shape)
print('Labels for training:', type(labels), labels.shape)


knn = cv2.ml.KNearest_create()

import time

s_time = time.time()
knn.train(data, cv2.ml.ROW_SAMPLE, labels)

import os
import sys
from os.path import getsize
import os

#파일 저장
file_path = 'knn_data.xml'
knn.save(file_path)
f_size = getsize("knn_data.xml")
print(f'knn file size before training={f_size}')
print(f'model size before training= {sys.getsizeof(knn)}')
file_size = os.path.getsize(file_path)
```

```
print(f"Model file size: {file_size} bytes")

# 파일 존재 여부 확인
if os.path.exists(file_path):
        # 파일 읽기 권한 확인
    if os.access(file_path, os.R_OK):
        print("File exists and is readable")
        knn_loaded = cv2.ml.KNearest_create()
        knn_loaded.load(file_path)
    else:
        print("File is not readable")
else:
    print("File does not exist")

# 학습 데이터의 양 출력 (원본 데이터 크기를 기반으로)
print("Number of training samples:", len(data))
print(f'model size after training= {sys.getsizeof(knn_loaded)}')
file_size = os.path.getsize(file_path)
print(f"Model file size: {file_size} bytes")
```

학습 데이터를 400개를 설정했을 때

```
data for training: <class 'numpy.ndarray'> (400, 2)
Labels for training: <class 'numpy.ndarray'> (400, 1)
knn file size before training=5134
model size before training= 32
Model file size: 5134 bytes
File exists and is readable
Number of training samples: 400
model size after training= 32
Model file size: 5134 bytes

Process finished with exit code 0
```

=> model file size가 5134 byte로 출력값이 나옵니다.

학습데이터를 40000개로 설정했을 때

```
data for training: <class 'numpy.ndarray'> (40000, 2)
Labels for training: <class 'numpy.ndarray'> (40000, 1)
knn file size before training=471840
model size before training= 32
Model file size: 471840 bytes
File exists and is readable
Number of training samples: 40000
model size after training= 32
Model file size: 471840 bytes

Process finished with exit code 0
```

=> model file size가 471840 bytes로 늘어난 것을 확인할 수 있습니다.

Python의 pickle 모듈은 일부 C 기반 라이브러리 객체, 특히 OpenCV 객체 같은 경우 직렬화하는데 한계가 있습니다. 이런 객체들은 내부적으로 복잡한 C++ 구조를 가지고 있기 때문에 pickle로 knn 모델을 직렬화할 수 없습니다.

OpenCV의 KNN 모델과 같은 경우 다른 방법으로 모델 데이터를 저장하고 로드할 수 있습니다. OpenCV는 이를 위해 cv2.FileStorage를 제공합니다. 이를 사용하여 모델의 파라미터를 XML 또는 YAML 파일로 저장하고 불러올 수 있습니다.

Knn 모델은 학습 단계에서 데이터를 저장하는 방식으로 작동하기 때문에, "학습 전"과 "학습 후"의 데이터 양은 변경되지 않습니다. KNN 은 학습 과정에서 데이터를 변형하거나 압축하지 않고, 전체 데이터 세트를 메모리에 그대로 저장합니다.

sys.getsizeof()를 호출하는 것은 모델 객체 자체의 기본 크기만을 제공합니다. 크기는 객체 자체의 저장 공간과 함께 객체가 내부적으로 참조하는 다른 객체들의 크기를 포함하지 않습니다. 리스트 객체의 크기를 반환할 때 리스트가 담고 있는 항목들의 크기는 포함되지 않고, 리스트 구조 자체의 메모리 크기만을 계산합니다.

반면에 os.path.getsize()를 호출하는 것은 파일 시스템에 있는 파일의 크기를 바이트 단위로 반환합니다. 이는 실제 파일의 디스크 상의 크기를 의미하며, 파일 내용에 따라 달라진다는 것을 확인했습니다.

결론적으로, KNN 모델은 xml로 저장하여 학습 데이터가 적을 때와 많을 때의 파일 크기가 달라집니다. 학습데이터가 양이 많아질수록 bytes가 늘어나는 것을 확인할 수 있습니다. 하지만 학습데이터가 모델안에 학습 데이터를 저장하고 있는 지는 확인할 수 가 없습니다. 왜냐하면 KNN 모델은 학습단계에서만 저장하는 기능을 하고 있어 학습 후에 모델 객체가 지정하는 학습 데이터가 포함되는지는 판단이 불가능합니다.