6. scikit-learn Model 서비스하기

- scikit-learn 공식 문서 : https://scikit-learn.org/stable/

▶️ 사용할 Minio Bucket 만들기 (Python API 사용)

```
$ python3
>>> from minio import Minio
>>> mc = Minio("127.0.0.1:9000", access_key="admin", secret_key="V4j7SPWAyQNAdmDyqSVGEbZSS
tX0osLf", secure=False)
>>> mc.make_bucket("random-forest")
>>> buckets = mc.list_buckets()
>>> bucket_list = [bucket.name for bucket in buckets]
>>> bucket_list
```

▶️ 모델 학습해 Minio에 저장하기

✅ 디렉터리 만들기

```
$ mkdir random-forest-train && cd random-forest-train
$ faas-cli new --lang python3-debian random-forest-train --prefix="127.0.0.1:5000"
```

✅ Dependency 추가

```
# random-forest-train/requirements.txt

minio
numpy
scikit-learn
```

✅ Input & Output

[Input : JSON]

```
{
  "n_estimator": [NUMBER OF ESTIMATOR],
  "cretirion": ["gini" or "entropy"],
  "train_test_split": [true or false],
  "test_ratio": [RATIO]
}
```

[Output: JSON]

```
{
  "minio_save_dir": {
    "model": [MODEL FILE PATH],
    "info": [INFO FILE PATH]
  },
  "performance": {
    "train": [TRAIN SCORE],
    "test" : [TEST SCORE]
  }
}
```

✅ 함수 작성하기

```
# random-forest-train/handler.py

import os
import json
from minio import Minio
from datetime import datetime

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import pickle


def handle(req):
    req = json.loads(req)
    check(req)

    mc = Minio(
        os.environ['minio_hostname'],
        access_key=os.environ['minio_access_key'],
        secret_key=os.environ['minio_secret_key'],
        secure=False
    )
    iris = load_iris()
    if req["train_test_split"]:
        X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_s
ize=req["test_ratio"], random_state=42)
    else:
        X_train, X_test, y_train, y_test = iris.data, None, iris.target, None

    model, score = train(req["n_estimator"], req["cretirion"], X_train, y_train, X_test, y
_test)
    model_path, score_path = save(mc, model, score)

    return json.dumps({
        "minio_save_dir": {
            "model": model_path,
            "info": score_path
        },
        "performance": score
    })


def check(req):
    assert "n_estimator" in req.keys()
    assert type(req["n_estimator"]) is int

    assert "cretirion" in req.keys()
    assert req["cretirion"] in ["gini", "entropy"]

    if "train_test_split" in req.keys() and req["train_test_split"]:
        assert "test_ratio" in  req.keys()
        assert type(req["test_ratio"]) is float


def train(n_estimator, cretirion, X_train, y_train, X_test=None, y_test=None):
    model = RandomForestClassifier(n_estimators=n_estimator, criterion=cretirion, verbose=
True)
    model.fit(X_train, y_train)

    score = {
        "train": accuracy_score(model.predict(X_train), y_train)
    }
    if X_train is not None:
        score["test"] = accuracy_score(model.predict(X_test), y_test)

    return model, score


def save(mc, model, score):
    now = datetime.now().strftime("%y%m%d-%H%M")

    model_filename = now + "-" + "RandomForest.pkl"
    model_path = os.path.join("/tmp", model_filename)

    score_filename = now + "-" + "Info.json"
    score_path = os.path.join("/tmp", score_filename)
    with open(model_path, "wb") as f:
        pickle.dump(model, f)
    with open(score_path, "w") as f:
        json.dump(score, f)

    mc.fput_object("random-forest", model_filename, model_path)
    mc.fput_object("random-forest", score_filename, score_path)

    return os.path.join("/data", "random-forest", model_filename), \
            os.path.join("/data", "random-forest", score_filename)
```

✅ YAML 수정하기

```
# random-forest-train.yml

...
functions:
  random-forest-train:
    ...
    environment:
      minio_hostname: [Your IP]:9000
      minio_access_key: admin
      minio_secret_key: V4j7SPWAyQNAdmDyqSVGEbZSStX0osLf
      write_debug: true
```

✅ 빌드, 푸시, 배포

```
$ faas-cli up -f ./random-forest-train.yml --gateway http://127.0.0.1:9888
$ kubectl get pods -n openfaas-fn
```

✅ 테스트

```
$ echo '{
  "n_estimator": 100,
  "cretirion": "gini",
  "train_test_split": true,
  "test_ratio": 0.2
}' | faas invoke random-forest-train --gateway http://127.0.0.1:9888
```

# {"minio_save_dir": {"model": "/data/random-forest/220126-0524-RandomForest.pkl", "info": "/data/random-forest/220126-0524-Info.json"}, "performance": {"train": 1.0, "test": 1.0}}

▶️ Minio에서 모델 읽어 서비스하기

✔️ 디렉터리 만들기

```
$ mkdir random-forest-pred && cd random-forest-pred
$ faas-cli new --lang python3-debian random-forest-pred --prefix="127.0.0.1:5000"
```

✔️ Dependency 추가

```
# random-forest-pred/requirements.txt

minio
numpy
scikit-learn
```

✔️ Input & Output

[Input : JSON]

```
{
  "model": [MODEL NAME],
  "data": [DATA]
}
```

[Output: JSON]

```
{
  "result": [CLASS NUMBER]
}
```

✔️ 함수 작성하기

```
# random-forest-pred/handler.py

import os
import json
import pickle
import numpy as np
```

```
from minio import Minio


def handle(req):
    req = json.loads(req)

    mc = Minio(
        os.environ['minio_hostname'],
        access_key=os.environ['minio_access_key'],
        secret_key=os.environ['minio_secret_key'],
        secure=False
    )

    mc.fget_object("random-forest", req["model"], os.path.join("/tmp", req["model"]))
    with open (os.path.join("/tmp", req["model"]), "rb") as f:
        model = pickle.load(f)

    data = np.array(req["data"]).reshape(1, -1)

    return json.dumps({
        "result": model.predict(data).tolist()[0]
    })
```

✅ YAML 수정하기

```
# random-forest-pred.yml

...
functions:
  random-forest-pred:
    ...
    environment:
      minio_hostname: [Your IP]:9000
      minio_access_key: admin
      minio_secret_key: V4j7SPWAyQNAdmDyqSVGEbZSStX0osLf
      write_debug: true
```

✅ 빌드, 푸시, 배포

```
$ faas-cli up -f ./random-forest-pred.yml --gateway http://127.0.0.1:9888
$ kubectl get pods -n openfaas-fn
```

✅ 테스트

```
$ echo '{
  "model": "220126-0524-RandomForest.pkl",
  "data": [4.6, 3.6, 1.0, 0.2]
}' | faas invoke random-forest-pred --gateway http://127.0.0.1:9888
```

# {"result": 0}