기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 03. 지도학습 알아보기

정 정 민

programmers

Chapter 07. 패키지 소개

1. Scikit-learn
2. 기타 패키지
    - Numpy
    - pandas
    - matplotlib

Scikit-learn

scikit-learn 이란?

- “사이킷런”은 다양한 머신러닝 알고리즘이 구현되어 있는 오픈 소스 패키지
  - 그래서 내부 작동 과정을 하나하나 확인할 수 있음
- 데이터 처리, 파이프라인, 여러 학습 알고리즘, 전/후처리 등 다양한 기능을 제공
  - 실제 산업 현장이나 학계에서도 널리 사용됨
- 또한, 타 파이썬 패키지와 과학 분석 목적 패키지와 연동이 좋음

- 여러 알고리즘을 설명한 문서 페이지 : 링크
- 사용자 가이드 : 링크
- API 문서 : 링크

https://en.wikipedia.org/wiki/Scikit-learn

scikit-learn의 주요 객체

- scikit-learn에서는 아래의 주요 기능을 갖는 객체를 제공
- 제공하는 머신러닝 모델 및 알고리즘은 아래 객체의 메서드를 전부 혹은 일부를 사용
- 이는 통일된 API 호출 시스템을 구성해 사용자가 손쉽게 사용할 수 있는 인터페이스를 제공하기 위함

- Estimator (추정기)
  - fit() 메서드를 활용하여 학습을 진행
  - 데이터로부터 패턴을 학습하고 결과로 모델 내부 파라미터를 조정
- Predictor (예측기)
  - predict() 메서드를 활용
  - 학습된 모델을 사용해 새로운 데이터에 대한 예측을 수행
- Transformer (변환기)
  - 데이터를 새로운 형태로 변환하기 위해 transform() 메서드 활용
  - 군집화, PCA 차원 축소 등의 과정을 위해 fit_transform() 메서드를 활용
- Model (모델)
  - 모델의 학습 적합도를 확인할 수 있도록 score() 메서드를 제공
  - 학습된 모델의 성능을 평가할 수 있음

https://scikit-learn.org/stable/developers/develop.html#apis-of-scikit-learn-objects

scikit-learn의 주요 객체를 활용한 예시 코드

```python
from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error

# 샘플 데이터 생성
X, y = make_regression(n_samples=100, n_features=1,
                       noise=0.4, random_state=0)

# StandardScaler로 데이터 표준화 (Transformer)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 선형 회귀 모델 생성 및 학습 (Estimator)
model = LinearRegression()
model.fit(X_scaled, y)

# 새로운 데이터에 대한 예측 수행 (Predictor)
y_pred = model.predict(X_scaled)

# 모델 성능 평가 (Model)
score = model.score(X_scaled, y)
mse = mean_squared_error(y, y_pred)
```

파이프라인, Pipeline

- 머신러닝 워크플로우의 여러 단계를 하나의 수준으로 연결하는 작업
- 데이터 전처리부터 모델 학습과 예측까지 원하는 범위의 작업을 하나로 묶을 수 있음
- 효율적으로 코드를 작성하고 관리할 수 있음

- Pipeline은 이름과 추정기 객체가 (key, value) 쌍으로 구성되어야 함
  - Pipeline 구성을 손쉽게 하려면 make_pipeline 함수 활용 (링크) : key 값 자동 할당

```python
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

# 파이프라인 생성
pipeline = Pipeline([
    ('scaler', StandardScaler()), # 첫 번째 단계: 데이터 스케일링
    ('clf', LogisticRegression()) # 두 번째 단계: 로지스틱 회귀 모델
])

# 파이프라인을 사용한 학습
pipeline.fit(X_train, y_train)

# 파이프라인을 사용한 예측
predictions = pipeline.predict(X_test)
```

```python
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

# make_pipeline을 사용하여 파이프라인 생성
pipeline = make_pipeline(
    StandardScaler(),    # 데이터 스케일링
    LogisticRegression() # 로지스틱 회귀 모델
)

# 파이프라인을 사용한 학습
pipeline.fit(X_train, y_train)

# 파이프라인을 사용한 예측
predictions = pipeline.predict(X_test)
```

https://scikit-learn.org/stable/modules/compose.html#pipeline-chaining-estimators

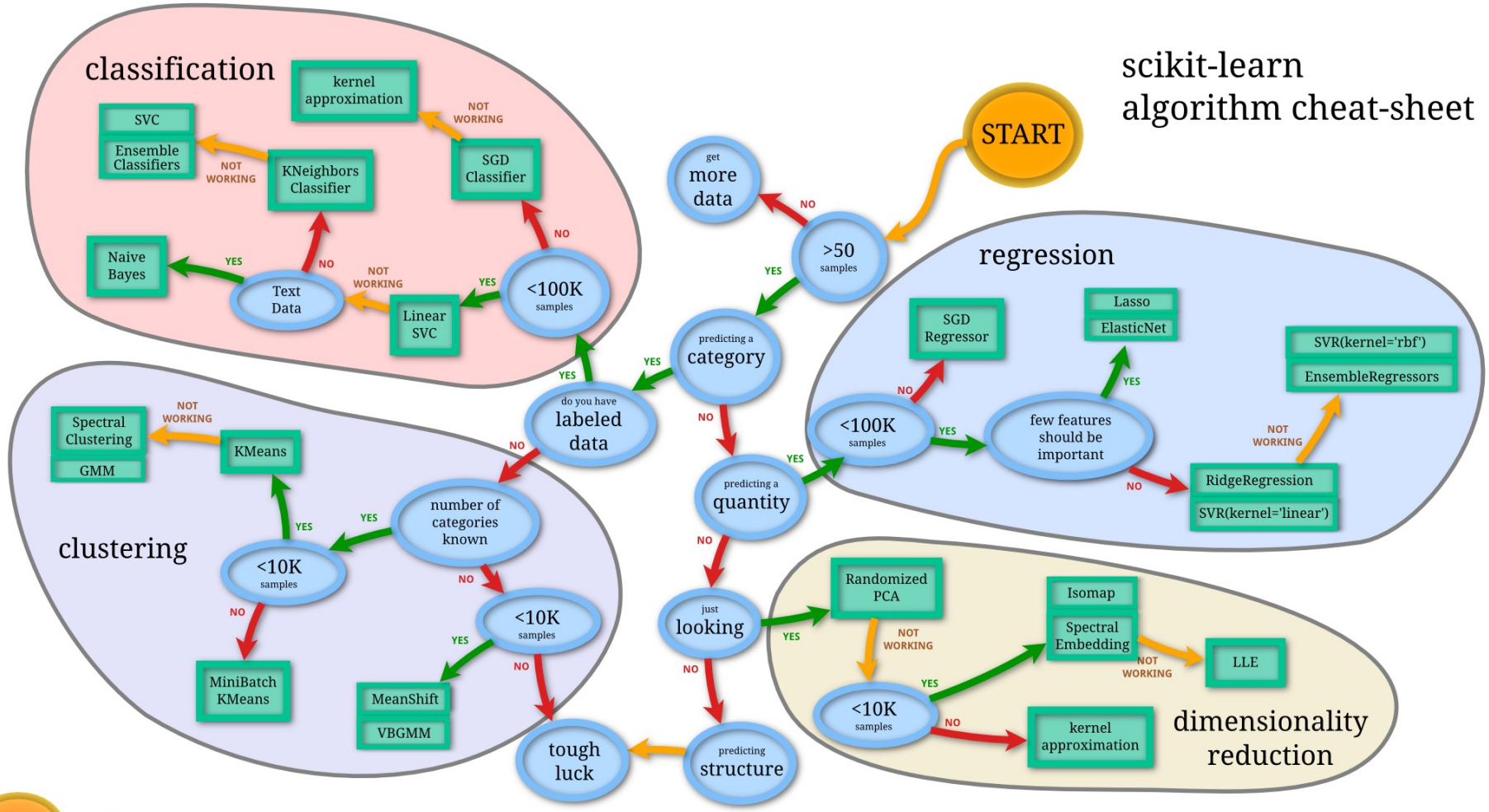
파이프라인, Pipeline

- Pipeline의 가장 마지막을 제외하고는 모두 변환기(transformer)여야 함
  - 다음 과정으로 데이터를 넘겨주어야 하므로
- 마지막은 추정기(estimator), 예측기(predictor), 변환기(transformer)가 올 수 있음
  - 추정기나 예측기가 온다면 fit()과 predict()를 사용해 모델 학습과 추론을 진행할 수 있고
  - 번환기가 온다면 단순 전처리기 정도로 사용할 수 있음

- 마지막 단계의 컴포넌트는 파이프라인의 최종 메서드에 영향을 미침
  - 만약, 마지막 컴포넌트가 변환기라면 fit()과 predict() 메서드를 사용할 수 없음

https://scikit-learn.org/stable/modules/compose.html#pipeline-chaining-estimators

Scikit-learn의 Learning Map



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

기타 패키지

Numpy

- Python에서 다차원 배열 연산, 행렬 연산, 고수준의 수학 함수, 난수 생성과 같은 과학적인 계산을 위한 패키지
- 데이터 분석, 머신러닝의 기반이 되는 필수 라이브러리

- 특히, scikit-learn에서 사용하는 기본 데이터 구조가 Numpy의 배열(array)

- 공식 문서 : 링크
- 튜토리얼 : 링크
- API 문서 : 링크

https://ko.wikipedia.org/wiki/NumPy

pandas

- Python에서 사용하는 패키지로 데이터 분석 기능을 제공하는 패키지
- 엑셀, CSV, 데이터베이스와 같은 다양한 파일에서 데이터를 읽어 들일 수 있는 기능을 제공
- DataFrame이라는 데이터 구조를 기반으로 SQL처럼 테이블에 Query 명령을 수행할 수 있음

- 공식 문서 : 링크
- 튜토리얼 : 링크
- API 문서 : 링크

```python
import pandas as pd

# DataFrame으로 만들 데이터 생성
data = {
    "이름": ["그랩", "프로그래머스", "머신러닝"],
    "나이": [28, 34, 22],
    "도시": ["서울", "부산", "대전"]
}

# 딕셔너리를 사용하여 DataFrame 생성
df = pd.DataFrame(data)

# 생성된 DataFrame 출력
print(df)
```

https://pandas.pydata.org/

Matplotlib

- Python에서 사용하는 과학 계산용 그래프 시각화 오픈소스 라이브러리
- 데이터를 바탕으로 선 그래프, 히스토그램, 산점도 등의 다양한 그래프를 시각화
- Colab과 같이 웹 상황에서 Matplotlib을 사용하려면 '%matplotlib inline' 매직 명령어를 사용해야 함

- 공식 문서 : 링크
- 튜토리얼 : 링크
- API 문서 : 링크

https://github.com/matplotlib/matplotlib

E.O.D