

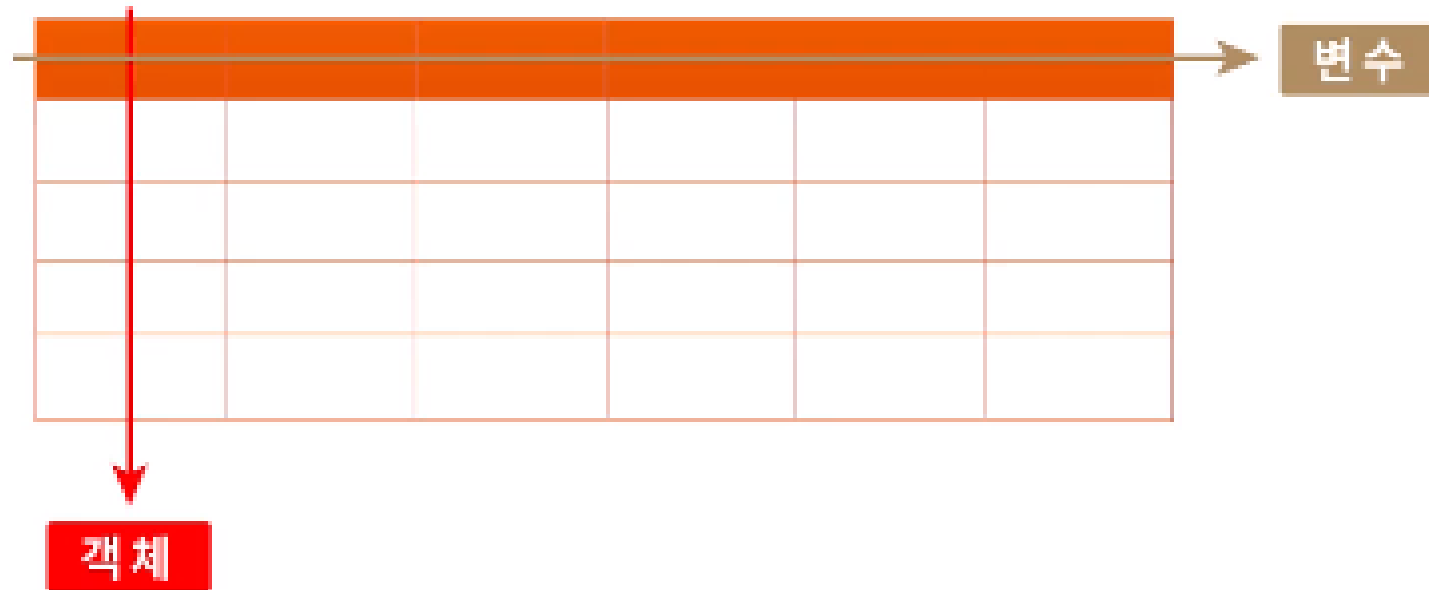
공통기초스터디 8주차

DONGDUK AI LEARNING CREW TEAM SCALAR

TIME: PM 4:00~6:00



Scikit-Learn 알고리즘



Scikit-Learn Api

- Scikit-Learn Api 사용 단계

1

Scikit Learn API에서 적절한 추정기 클래스를 임포트 해서 사용하고자 하는 모델을 선택한다.

2

클래스로부터 인스턴스를 생성하고
초모수(Hyper-parameter)를 설정한다.

3

데이터를 특징 배열과 대상 배열로 준비한다.

4

모델 인스턴스의 fit() 메소드를 호출해서 데이터를 학습한다.

5

정확도를 확인하고 새로운 데이터에 모델을 적용한다.



Scikit-Learn Api

- fit메소드

`model.fit(X,y)`

모델확인

- 계수값

`model.coef_`

- 절편값

`model.intercept_`

- 모델의 정확도

`model.score(X,y)`



Scikitlearn 다항식회귀분석

- 모델의 성과 개선 방법

1

더 복잡하거나 더 유연한 모델 사용

2

덜 복잡하거나 덜 유연한 모델 사용

3

더 많은 훈련 표본 수집

4

각 표본에 특징을 추가하기 위해 더 많은 데이터 수집



Scikitlearn 다항식회귀분석

- “파이프 라인” :전처리 프로그램과 선형 회귀 모델을 묶어주는 프로그램

```
from sklearn.preprocessing import PolynomialFeatures
```

```
from sklearn.linear_model import LinearRegression
```

```
from sklearn.pipeline import make_pipeline
```

```
def PolynomialRegression(degree=2, **kwargs):
```

- return make_pipeline(PolynomialFeatures(degree), LinearRegression (**kwargs))

“데이터 시각화”

```
%matplotlib inline
```

```
import matplotlib.pyplot as plt
```

```
import seaborn; seaborn.set()
```

```
X_test = np.linspace(-0.1, 1.1, 500) plt.scatter(x, y, color='black') axis = plt.axis()
```

- “데이터 적합”
- “데이터 검증 곡선”



Scikitlearn 다항식회귀분석더보기

- “파이프 라인” :전처리 프로그램과 선형 회귀 모델을 묶어주는 프로그램

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import make_pipeline
def PolynomialRegression(degree=2, **kwargs):
    return make_pipeline(PolynomialFeatures(degree), LinearRegression(**kwargs))
```

“데이터 시각화”

%matplotlib inline

import matplotlib.pyplot as plt

import seaborn; seaborn.set()

X_test = np.linspace(-0.1, 1.1, 500) plt.scatter(x, y, color='black') axis = plt.axis()

“데이터 적합 ”

plt.scatter(x, y, color='black')

for degree in [1,2,3]:

model = PolynomialRegression(degree) model.fit(X[:, np.newaxis],y)

y_test=model.predict(X_test[:,np.newaxis])

plt.plot(x_test, y_test, label=f'Degree={degree}')

plt.axis([-0.1, 1.0, -2, 12])

plt.legend(loc='best');

“데이터 검증 곡선”

from sklearn.model_selection import validation_curve

degree=np.arange(1, 10)

train_score, val_score=validation_curve(PolynomialRegression(), X[:, np.newaxis], y, 'polynomialfeatures_degree', degree, cv=7)

plt.plot(degree, np.median(train_score, 1), color='blue', label="training score")

plt.plot(degree, np.median(val_score, 1), color='red', label='validation score')

plt.legend(loc='best')

plt.ylim(0,1)





THANK YOU