**교육일지**

**교육 제목 : 머신러닝**

**교육 장소 : YGL C6 강의실**

**교육 일시 : 2021/10/18**

**특성공학과 규제**

**데이터 준비**

**df = pd.read\_csv('https://bit.ly/perch\_csv\_data')**

**perch\_full = df.to\_numpy()**

**print(perch\_full) # 훈련 점수와 테스트 점수를 저장**

**import numpy as np**

**perch\_weight = np.array(**

**[5.9, 32.0, 40.0, 51.5, 70.0, 100.0, 78.0, 80.0, 85.0, 85.0,**

**110.0, 115.0, 125.0, 130.0, 120.0, 120.0, 130.0, 135.0, 110.0,**

**130.0, 150.0, 145.0, 150.0, 170.0, 225.0, 145.0, 188.0, 180.0,**

**197.0, 218.0, 300.0, 260.0, 265.0, 250.0, 250.0, 300.0, 320.0,**

**514.0, 556.0, 840.0, 685.0, 700.0, 700.0, 690.0, 900.0, 650.0,**

**820.0, 850.0, 900.0, 1015.0, 820.0, 1100.0, 1000.0, 1100.0,**

**1000.0, 1000.0]**

**)**

**from sklearn.model\_selection import train\_test\_split**

**train\_input, test\_input, train\_target, test\_target = train\_test\_split(perch\_full, perch\_weight, random\_state=42)**

**사이킷런 변환기**

**from sklearn.preprocessing import PolynomialFeatures**

**poly = PolynomialFeatures()**

**poly.fit([[2, 3]])**

**print(poly.transform([[2, 3]]))**

**poly = PolynomialFeatures(include\_bias=False)**

**poly.fit([[2, 3]])**

**print(poly.transform([[2, 3]]))**

**poly = PolynomialFeatures(include\_bias=False)**

**poly.fit(train\_input)**

**train\_poly = poly.transform(train\_input)**

**print(train\_poly.shape)**

**poly.get\_feature\_names()**

**test\_poly = poly.transform(test\_input)**

**다중회귀 모델 훈련하기**

**from sklearn.linear\_model import LinearRegression**

**lr = LinearRegression()**

**lr.fit(train\_poly, train\_target)**

**print(lr.score(train\_poly, train\_target))**

**print(lr.score(test\_poly, test\_target))**

**poly = PolynomialFeatures(degree=5, include\_bias=False)**

**poly.fit(train\_input)**

**train\_poly = poly.transform(train\_input)**

**test\_poly = poly.transform(test\_input)**

**print(train\_poly.shape)**

**lr.fit(train\_poly, train\_target)**

**print(lr.score(train\_poly, train\_target))**

**print(lr.score(test\_poly, test\_target))**

**규제**

**from sklearn.preprocessing import StandardScaler**

**ss = StandardScaler()**

**ss.fit(train\_poly)**

**train\_scaled = ss.transform(train\_poly)**

**test\_scaled = ss.transform(test\_poly)**

**릿지**

**from sklearn.linear\_model import Ridge**

**ridge = Ridge()**

**ridge.fit(train\_scaled, train\_target)**

**print(ridge.score(train\_scaled, train\_target))**

**print(ridge.score(test\_scaled, test\_target))**

**import matplotlib.pyplot as plt**

**train\_score = []**

**test\_score = []**

**alpha\_list = [0.001, 0.01, 0.1, 1, 10, 100]**

**for alpha in alpha\_list:**

**# 릿지 모델을 만듬**

**ridge = Ridge(alpha=alpha)**

**# 릿지 모델을 훈련**

**ridge.fit(train\_scaled, train\_target)**

**# 훈련 점수와 테스트 점수를 저장**

**train\_score.append(ridge.score(train\_scaled, train\_target))**

**test\_score.append(ridge.score(test\_scaled, test\_target))**

**plt.plot(np.log10(alpha\_list), train\_score)**

**plt.plot(np.log10(alpha\_list), test\_score)**

**plt.xlabel('alpha')**

**plt.ylabel('R^2')**

**plt.show()**

**라쏘**

**from sklearn.linear\_model import Lasso**

**lasso = Lasso()**

**lasso.fit(train\_scaled, train\_target)**

**print(lasso.score(train\_scaled, train\_target))**

**print(lasso.score(test\_scaled, test\_target))**

**train\_score = []**

**test\_score = []**

**alpha\_list = [0.001, 0.01, 0.1, 1, 10, 100]**

**for alpha in alpha\_list:**

**# 라쏘 모델을 만듬**

**lasso = Lasso(alpha=alpha, max\_iter=10000)**

**# 라쏘 모델을 훈련**

**lasso.fit(train\_scaled, train\_target)**

**# 훈련 점수와 테스트 점수를 저장**

**train\_score.append(lasso.score(train\_scaled, train\_target))**

**test\_score.append(lasso.score(test\_scaled, test\_target))**