**기계학습 과제물 제출**

**1826074 오현진**

**제출일자 : 22.10.29**



**[1] 442명의 당뇨병 환자를 대상으로 한 검사 결과를 나타내는 데이터셋 diabetes 확인**

**# 소스코드**

from sklearn.datasets import load\_diabetes

import pandas as pd

import seaborn as sns

diabetes = load\_diabetes()

df = pd.DataFrame(diabetes.data, columns=diabetes.feature\_names)

df["target"] = diabetes.target

df

**# 실습결과**

테이블이(가) 표시된 사진

자동 생성된 설명

**# 분석**

diabetes 데이터셋의 구조는 나이 age, 성별 sex, BMI지수 bmi, 평균혈압 bp, 6종류의 혈액검사 수치인 s1~s6이 있습니다.

diabetes.target을 이용하여 1년 후 질병 진행에 대한 수치적인 측정을 불러옵니다.

**[2] 당뇨병 진행도 예측 (선형회귀)**

**# 소스코드**

from sklearn.datasets import load\_diabetes

from sklearn.linear\_model import LinearRegression

from sklearn import model\_selection

from sklearn import metrics

diabetes = load\_diabetes()

X = diabetes.data

y = diabetes.target

X\_train, X\_test, y\_train, y\_test = model\_selection.train\_test\_split(X, y, test\_size=0.3)

lin\_reg = LinearRegression()

lin\_reg.fit(X\_train, y\_train)

y\_predict = lin\_reg.predict(X\_test)

score = metrics.r2\_score(y\_test, y\_predict)

print(f"accuracy = {score}")

**# 실습결과**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

**# 분석**

변수 x에는 diabetes 데이터 행렬인 diabetes.data를,

변수 y에는 1년 후 질병 진행에 대한 실제 정량적 측정인 diabetes.target을 저장합니다.

model\_selection.train\_tesp\_split 함수를 X,y에 대해 진행하여 train 데이터와 test 데이터를 분리합니다.

test\_size의 비율을 다르게 하면 당연히 예측 결과가 다르겠지만, 직접 확인하기 위해서 두 가지의 경우로 실습해보았습니다.

LinearRegression() 을 이용하여 선형회귀를 진행하고자 하였으며, X와 y의 학습용 데이터에 대해 fit 함수를 이용하여 학습을 하였습니다.

이후 X\_test데이터에 대한 y의 예측값을 predict함수로 얻어낸 뒤, 회귀 문제이므로 accuracy\_score이 아닌 r2\_socre을 이용하여 정확도를 얻어 출력하였습니다.

정확도 값은 test\_size가 0.3일 경우에는 약 46%였으며,

test\_size가 0.25일 경우에는 약 48%로 소폭 상승한 모습을 보였습니다.

**[3] 당뇨병 진행도 예측 (Ridge 선형회귀)**

**# 소스코드**

from sklearn.datasets import load\_diabetes

from sklearn.linear\_model import Ridge

from sklearn import model\_selection

from sklearn import metrics

diabetes = load\_diabetes()

X = diabetes.data

y = diabetes.target

X\_train, X\_test, y\_train, y\_test = model\_selection.train\_test\_split(X, y, test\_size=0.3)

lin\_reg = Ridge(alpha=1.0)

lin\_reg.fit(X\_train, y\_train)

y\_predict = lin\_reg.predict(X\_test)

score = metrics.r2\_score(y\_test, y\_predict)

print(f"accuracy = {score}")

**# 실습결과**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

**# 분석**

변수 x에는 diabetes 데이터 행렬인 diabetes.data를,

변수 y에는 1년 후 질병 진행에 대한 실제 정량적 측정인 diabetes.target을 저장합니다.

model\_selection.train\_tesp\_split 함수를 X,y에 대해 진행하여 train 데이터와 test 데이터를 분리합니다.

선형 모델의 성능을 향상시키기 위한 정규화 방법 중, 기존의 최소제곱합 식에 페널티 항을 추가한 Ridge 회귀법을 이용하여 선형회귀를 진행하고자 하였으며, 규제는 1.0으로 두었습니다. X와 y의 학습용 데이터에 대해 fit 함수를 이용하여 학습을 하였습니다.

규제 값인 alpha의 크기를 다르게 하면 당연히 예측 결과가 다르겠지만, 직접 확인하기 위해서 두 가지의 경우로 실습해보았습니다.

이후 X\_test데이터에 대한 y의 예측값을 predict함수로 얻어낸 뒤, 회귀 문제이므로 accuracy\_score이 아닌 r2\_socre을 이용하여 정확도를 얻어 출력하였습니다.

정확도 값은 규제가 1.0일 경우에는 약 45%였으며,

규제가 1.5일 경우에는 약 43%로 소폭 감소한 모습을 보였습니다.

이는 규제와 정확도가 반비례 관계에 있기 때문으로 이해하였습니다.

**[4] 당뇨병 진행도 예측 (Lasso 선형회귀)**

**# 소스코드**

from sklearn.datasets import load\_diabetes

from sklearn.linear\_model import Lasso

from sklearn import model\_selection

from sklearn import metrics

diabetes = load\_diabetes()

X = diabetes.data

y = diabetes.target

X\_train, X\_test, y\_train, y\_test = model\_selection.train\_test\_split(X, y, test\_size=0.3)

lin\_reg = Lasso(alpha=1.0)

lin\_reg.fit(X\_train, y\_train)

y\_predict = lin\_reg.predict(X\_test)

score = metrics.r2\_score(y\_test, y\_predict)

print(f"accuracy = {score}")

**# 실습결과**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

**# 분석**

변수 x에는 diabetes 데이터 행렬인 diabetes.data를,

변수 y에는 1년 후 질병 진행에 대한 실제 정량적 측정인 diabetes.target을 저장합니다.

model\_selection.train\_tesp\_split 함수를 X,y에 대해 진행하여 train 데이터와 test 데이터를 분리합니다.

선형 모델의 성능을 향상시키기 위한 정규화 방법 중, Ridge 회귀법의 페널티 항에 변화를 준 Lasso 회귀법을 이용하여 선형회귀를 진행하고자 하였으며, 규제는 1.0으로 두었습니다. X와 y의 학습용 데이터에 대해 fit 함수를 이용하여 학습을 하였습니다.

규제 값인 alpha의 크기를 다르게 하면 당연히 예측 결과가 다르겠지만, 직접 확인하기 위해서 두 가지의 경우로 실습해보았습니다.

이후 X\_test데이터에 대한 y의 예측값을 predict함수로 얻어낸 뒤, 회귀 문제이므로 accuracy\_score이 아닌 r2\_socre을 이용하여 정확도를 얻어 출력하였습니다.

정확도 값은 규제가 1.0일 경우에는 약 33%였으며,

규제가 1.5일 경우에는 약 22%로 소폭 감소한 모습을 보였습니다.

이는 규제와 정확도가 반비례 관계에 있기 때문으로 이해하였습니다.

**[5] 느낀점 및 배운점**

위의 분석 항목에 제가 적었듯이 선형 회귀의 경우 test 데이터의 비율이 작아질수록 모델의 정확도가 높아지는 모습을 보였으며, 이는 곧 test 데이터의 비율이 작으면 이에 대한 검증이 충분하지 못해 정확도가 높더라도 신빙성이 낮을 수 있다는 것으로로 해석했습니다.

또한 선형 회귀 모델의 성능을 향상시키기 위해 Ridge나 Lasso 회귀법을 이용하였을 경우, 규제의 정도가 크면 클 수록 정확도가 낮아지는 경향을 파악하였습니다.

계속되는 실습을 통해 각 경우에 어떤 모델을 선택하는 것이 좋으며, test 데이터의 비율과, 규제의 정도를 어느 정도로 두어야 신빙성 있는 결과 값을 얻을 수 있을지 감을 잡는 것이 중요할 것이라고 생각하였습니다.