

05

변수 선택법

[변수선택법]

최근 머신러닝에서는 다중공선성을 해결해주는 알고리즘들이

많이 있습니다. 하지만, 시험을 대비하여 알아두어야 합니다.

00:00

5.6 변수선택법 0 APP.

5.6.1 최적회귀방정식의 선택

- 모형 내 설명변수의 수가 증가할수록 데이터 관리에는 많은 노력이 요구된다. 따라서 상황에 따라 종속변수에 영향을 미치는 유의미한 독립변수들을 선택하여 최적의 회귀방정식을 도출하는 과정이 필요하다.
- 변수를 선택할 때는 F-통계량이나 AIC와 같은 특정 기준을 근거로 변수를 제거하거나 선택한다.
- t-통계량의 유의확률이 유의수준보다 큰 변수는 통계적으로 유의하지 않으므로 제거해야하고, AIC와 같은 별점화 기준을 가장 낮게 만드는 변수 조합을 선택해야 한다.

$$AIC = -2 \ln(L) + 2k$$

여기서 $-2\ln(L)$ 은 모형의 적합도를 의미하며, k 는 모형의 추정된 파라미터의 개수이다. $-2\ln(L)$ 에서 L 은 Likelihood function 을 의미하며, AIC 값이 낮다는 것은 즉 모형의 적합도가 높은 것을 의미한다.

(모형의 적합도란 실제 자료와 연구자의 연구 모형이 얼마나 부합하는지 평가하는 것)

여기서 $2k$ 는 모형의 추정된 파라미터의 개수를 의미하며, 해당 모형에 패널티를 주기 위해 사용한다.

실제로 어떤 모형이 $2\ln(L)$ 즉 적합도를 높이기 위해 여러 불필요한 파라미터를 사용할 수도 있다. 실제 모형 비교 시 독립변수가 많은 모형이 적합도 면에서 유리하게 되는데, 이는 즉 독립변수에 따라서 모형의 적합도에 차이가 난다는 의미이다. 따라서 이를 상쇄시키기 위하여 불필요한 파라미터, 즉 독립변수의 수가 증가할수록 $2k$ 를 증가시켜 패널티를 부여하여 모델의 품질을 평가한다.

[Cars93 데이터를 통한 변수선택법 실습]

04:35

```

In [1]: 1 import pandas as pd
        2 from pandas import DataFrame
        3
        4 # 데이터 불러오기
        5 Cars = pd.read_csv('../data/Cars93.csv')
        6

In [2]: 1 import numpy as np
        2 import statsmodels.api as sm
        3 import statsmodels.formula.api as smf
        4
        5 model = smf.ols(formula = "Price ~ EngineSize + RPM + Weight + Length", data = Cars)
        6 result = model.fit()
        7 result.summary()

Out[2]: OLS Regression Results

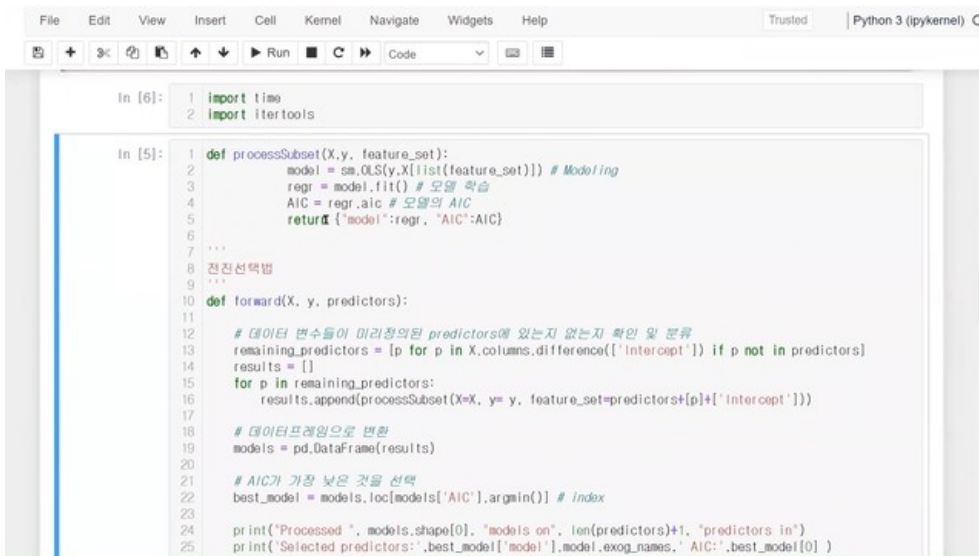
Dep. Variable: Price      R-squared: 0.563
Model: OLS      Adj. R-squared: 0.543
Method: Least Squares      F-statistic: 28.34
Date: Mon, 22 Nov 2021      Prob (F-statistic): 3.93e-15

```

[단계적 선택법 사용]

함수를 직접 만들고 사용하는 방법은 파이썬 기초문법에 대한

공부가 필요합니다. 그러므로 이 강의에서는 함수를 만들고 사용하는 방법까지는 설명하지 않겠습니다.



```

File Edit View Insert Cell Kernel Navigate Widgets Help Trusted Python 3 (ipykernel) O
In [6]: 1 import time
        2 import itertools

In [5]: 1 def processSubset(X,y, feature_set):
        2     model = sm.OLS(y,X[list(feature_set)]) # Modeling
        3     regr = model.fit() # 모델 학습
        4     AIC = regr.aic # 모델의 AIC
        5     return {'model':regr, "AIC":AIC}
        6
        7
        8 # 전진선택법
        9
        10 def forward(X, y, predictors):
        11
        12     # 데이터 변수들이 미리정의된 predictors에 있는지 없는지 확인 및 분류
        13     remaining_predictors = [p for p in X.columns,difference(['intercept']) if p not in predictors]
        14     results = []
        15     for p in remaining_predictors:
        16         results.append(processSubset(X=X, y= y, feature_set=predictors+[p]+'intercept'))
        17
        18     # 데이터프레임으로 변환
        19     models = pd.DataFrame(results)
        20
        21     # AIC가 가장 낮은 것을 선택
        22     best_model = models.loc[models['AIC'].argmin()] # index
        23
        24     print("Processed ", models.shape[0], "models on", len(predictors)+1, "predictors in")
        25     print("Selected predictors:",best_model['model'].model.exog_names,' AIC:',best_model[0] )
  
```