**Machine Learning & Data Mining, Spring 2020**

**Homework 4**

사이버보안학과

201620641

유 상 정

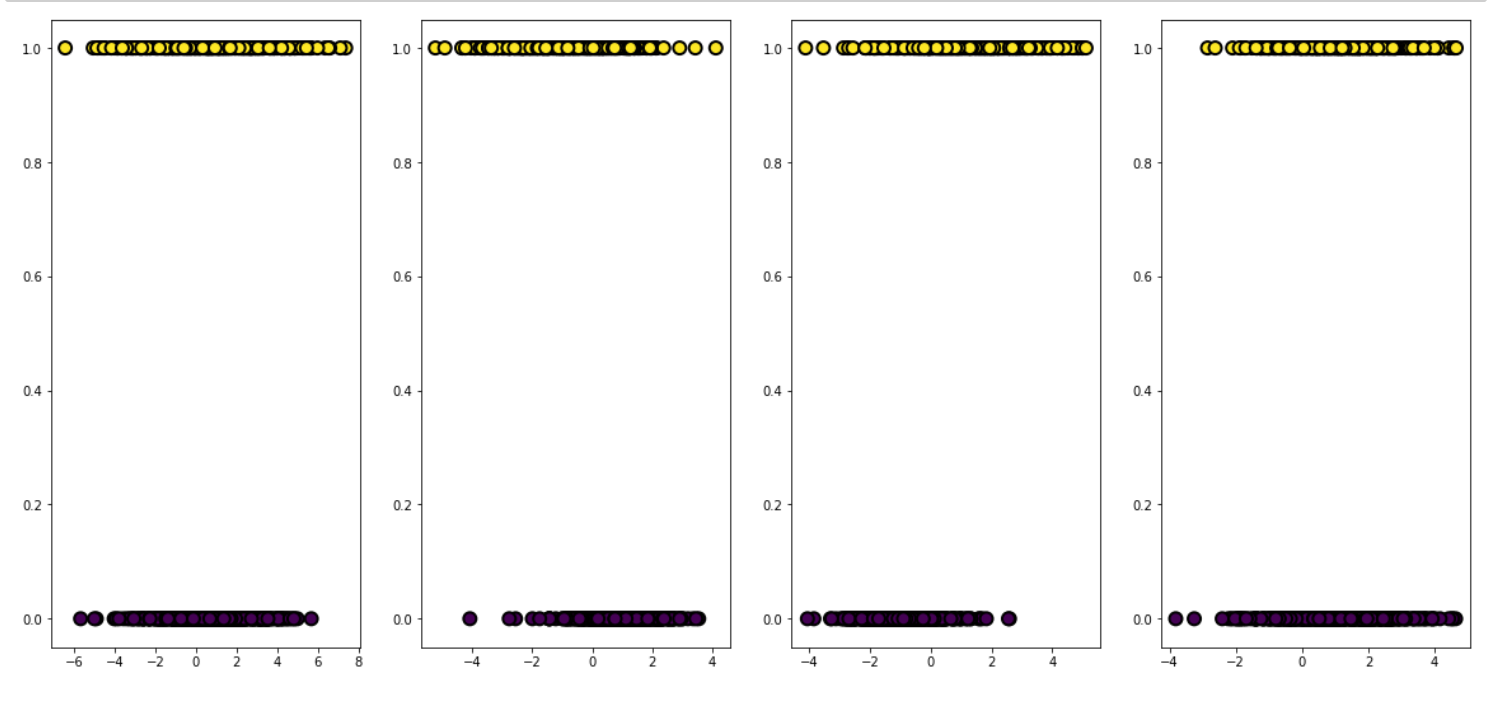
**과제목표**

* **수업시간에 배운 linear regression과 logistic regression 모델을 실제로 적용시켜 본다.**
* **Regression 모델에서 최적화하기 위한 variable selection 모델을 구현하고 적용한다.**

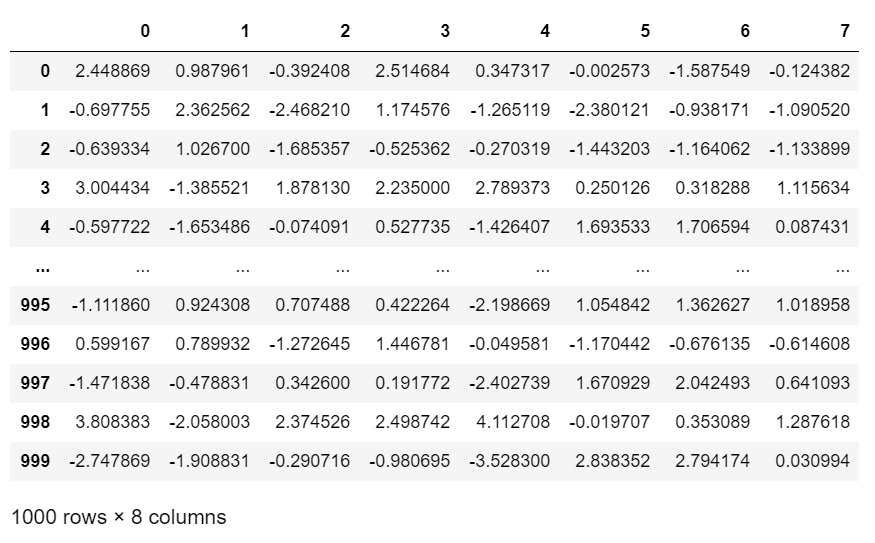
1. **Coefficient interpretation (logistic regression)**
   1. 주어진 데이터의 모든 feature에 대해서 correlation coefficient array를 구하고 이 중 correlation coefficient 가 높은 5개의 feature를 이용해 feature간의 correlation을 **heatmap으로 그린다.** [5]

A-1. 데이터 확인

주어진 데이터는 Make\_classification 으로 만들어졌다.

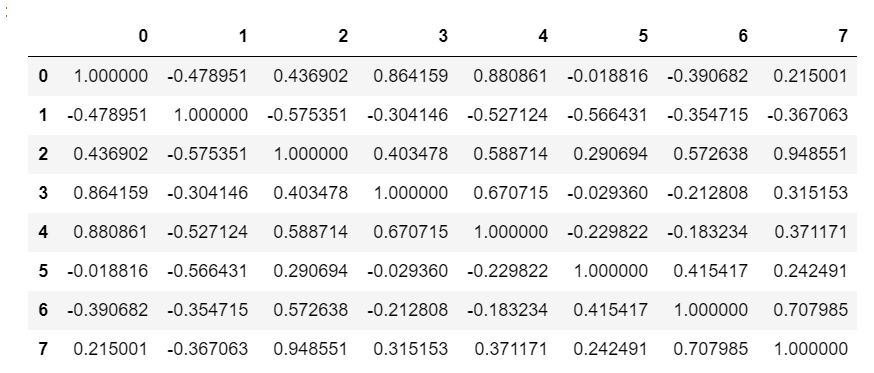


주어진 데이터의 분포를 보면 class가 0,1로 구분되는데, categorical y 값이 나오는 경우에는 선형관계가 없기 때문에 linear를 쓰지 않고 logistic reggresion을 써야한다.

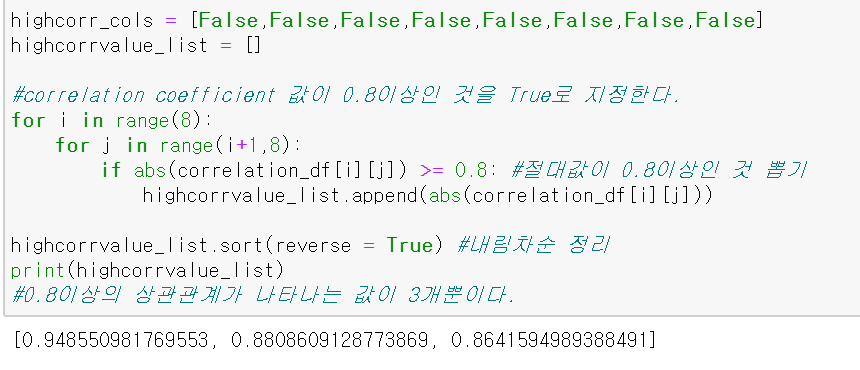


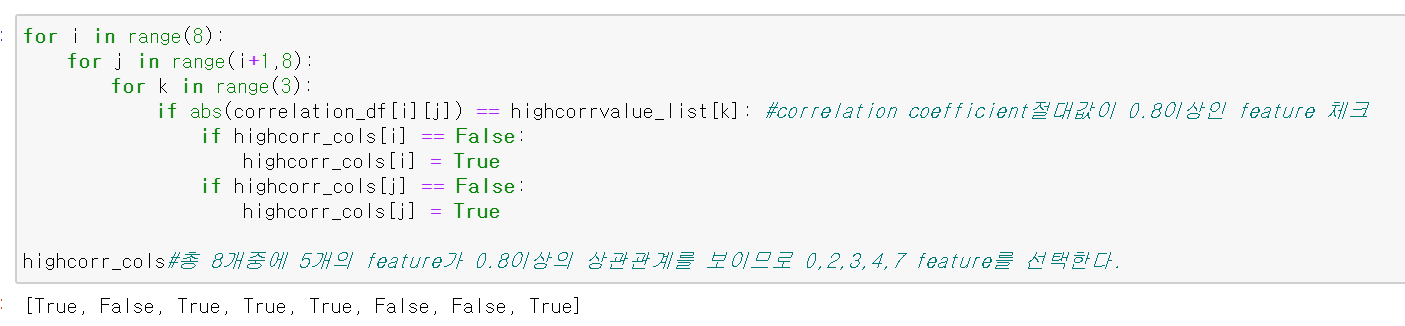
주어진 데이터를 pandas의 DataFrame으로 바꾸고, 확인하였다. Feature의 개수는 총 8개이다.

A-2. correlation coefficient 가 높은 5개의 feature 찾기

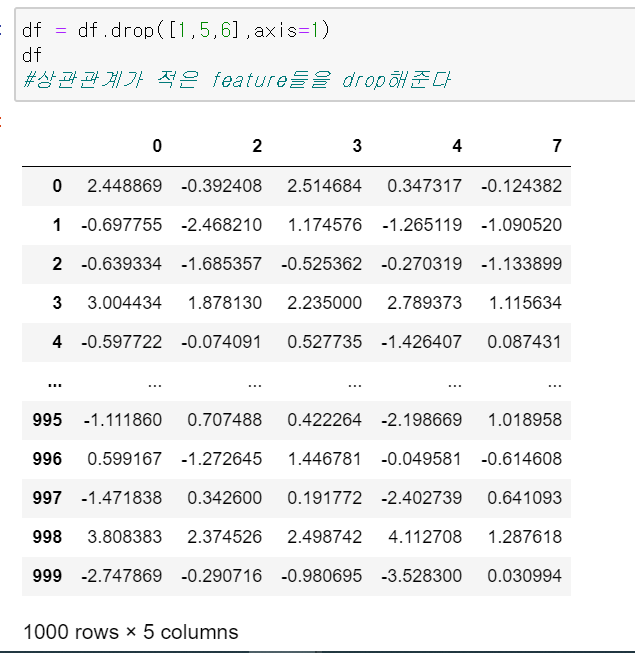


Python의 pandas는 Pearson Correlation Coefficient를 구할 수 있는 corr()함수를 제공한다. 이 함수를 이용하여 각 feature사이의 correlation coefficient를 구해서 list로 정리한 후, Np.array를 이용하여 array변환하였다. 그래서 correlation coefficient array를 구하였다. 그리고 보기 편하기 위하여, DataFrame으로 바꾸었다. 하지만, 위를 보고 바로 판단하기는 어려웠다. 그래서 correlation coefficient값이 0.8이상인 것만 뽑아서 내림차순으로 정리한 후, 그 값을 가지는 feature들만 뽑았다.

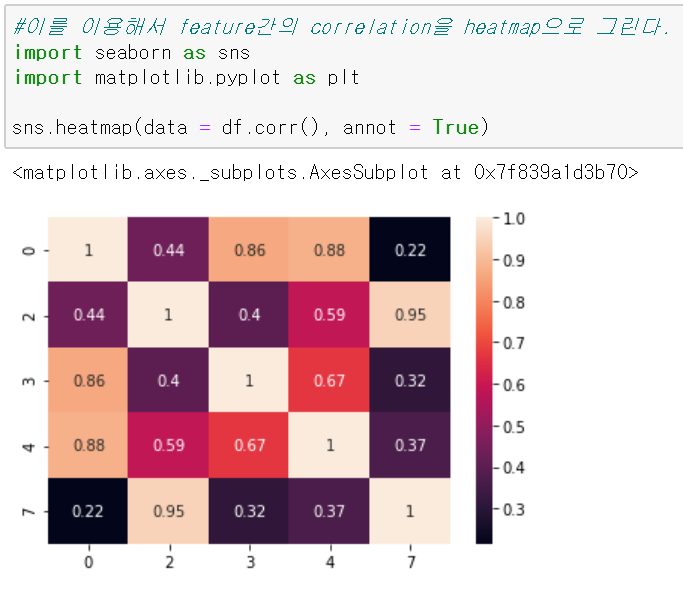


 coefficient array으로부터 위의 과정을 진행하면서 가장 correlation coefficient값이 높은 feature 5개를 골랐다. (위의 true값인 index들이 관련이 가장 높은 feature 5개이다.)

그래서 DataFrame을 상관관계가 높은 5개의 feature만을 이용하여 구성하였다.



A-3. correlation coefficient 가 높은 5개의 feature를 이용해 feature간의 correlation을 **heatmap**



5개의 feature를 이용하여 feature간의 correlation을 heatmap을 이용하여 나타냈다.

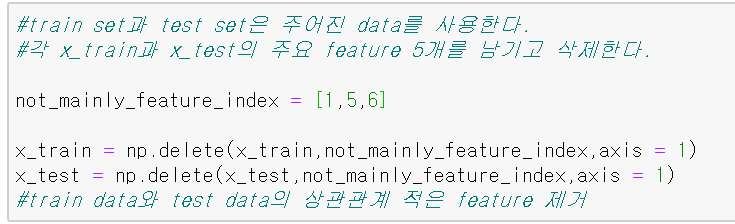
각 feature간의 correlation heatmap을 보면, 0번 feature는 3번, 4번 feature와 강한 양적 상관관계가 보인다. 2번 feature는 7번 feature와 강한 양적 상관관계 보인다. 또한, 3번 feature는 0번 feature와 강한 양적 상관관계를 보이고 4번feature와 뚜렷한 양적 상관관계를 보인다. 그리고 4번 feature는 0번 feature와 강한 양적 상관관계, 3번 feature와 뚜렷한 양적 상관관계를 보인다. 마지막으로 7번 feature는 2번 feature와 강한 양적 상관관계를 보인다. 이와 같이 Heatmap을 통해서 각 feature간의 상관관계를 알아보았다.

위의 판단은 아래 이미지를 참고하였다.

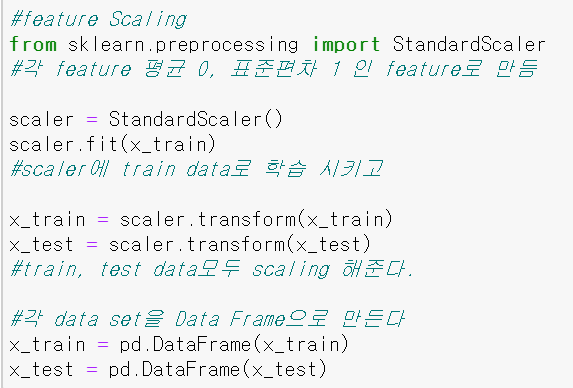


(출처: <https://gomguard.tistory.com/173>)

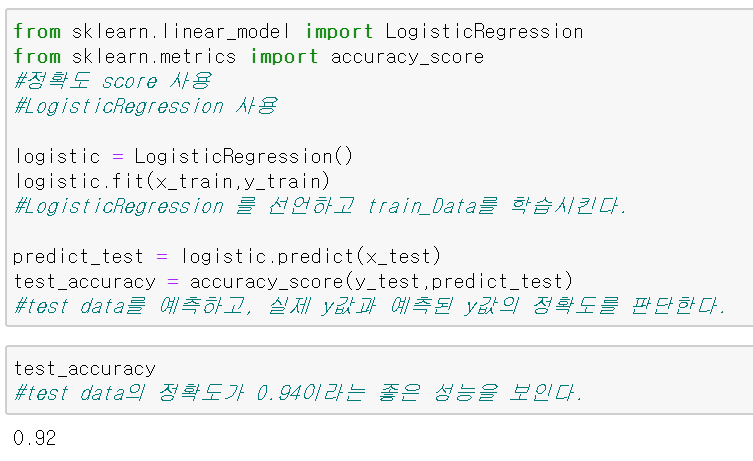
* 1. 위에서 구한 correlation coefficient array를 바탕으로 multiple **logistic** regression 모델에서 사용할 feature를 정하고 90%의 데이터를 학습데이터로 사용하여 regression 모델 학습 후 10% 테스트 데이터에 대한 **test acc를 출력**한다. [5]



위의 correlation coefficient array를 바탕으로 주요 5개의 feature를 제외하고 나머지 3개의 feature를 train data와 test data에서 제거하였다.



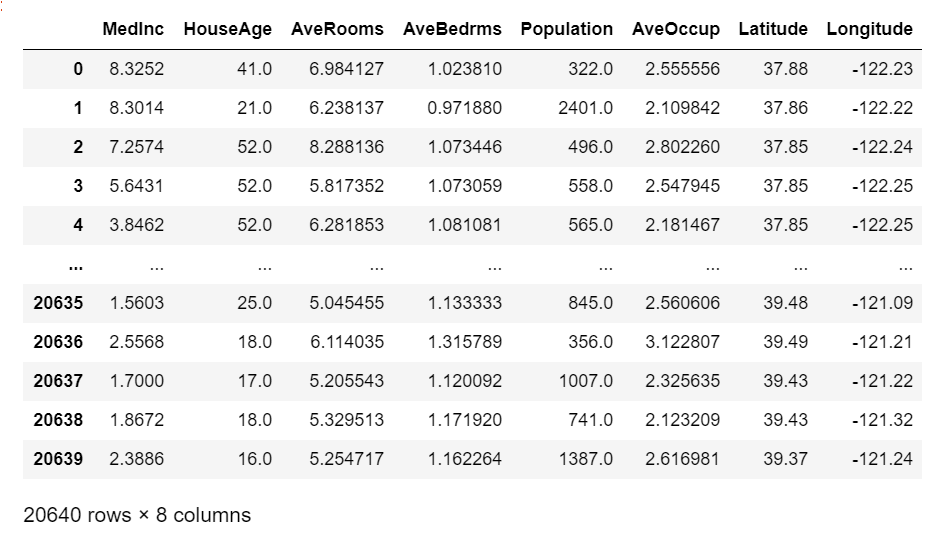
다음으로 feature scaling을 진행했다.



다음으로, logistic Regression를 선언하고, 전체 data의 90%에 해당하는 주어진 train data를 학습시킨 후, 전체 data의 10%에 해당하는 test data의 정확도를 판단했다. 그리고 과제의 요구사항인 test accuracy를 출력하였다. 0.92라는 준수한 성능을 보인다.

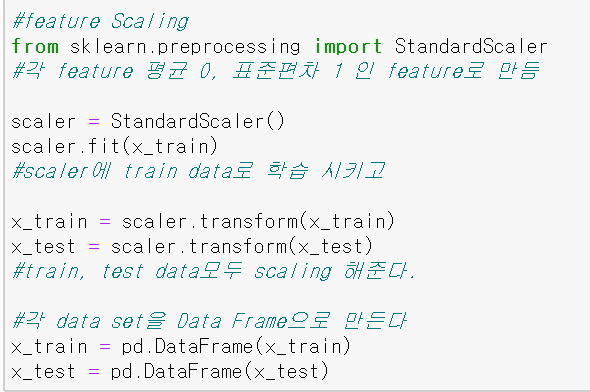
1. **Variable selection (linear regression)**
   1. Linear regression 모델의 variable 선택을 통한 모델을 비교하기 위해서 수업시간에 배운 Adjusted 을 기반으로 forward selection을 수행한 후 **test 를 출력하고** x축은 선택한 variable의 개수, y축은 테스트 데이터의 Adjusted 과 학습데이터의 Adjusted 를 각각 다른 색으로 **그래프를 이용해 출력한다**.[10]

A-1. California data set 확인



California data를 pandas를 변환한 후, 데이터의 feature개수가 8개임을 확인한다.

A-2. feature scaling



Linear regression의 최적화 과정의 안정성 및 수렴 속도 향상을 위해서 feature scaling을 진행한다.

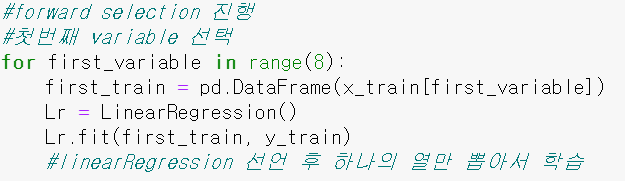
A-3. forward selection을 이용하여 variable을 4개까지 선택



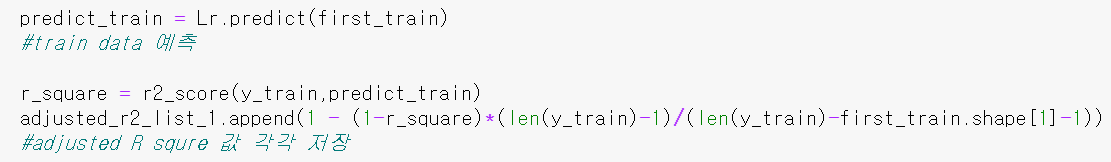
Linear regression과 r2\_score를 import 한다.



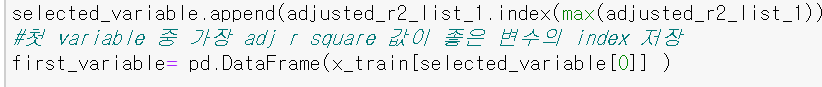
Forward selection을 진행하기 전에, 1,2,3,4번째에 선택된 variable을 저장하기 위한 list를 선언한다.



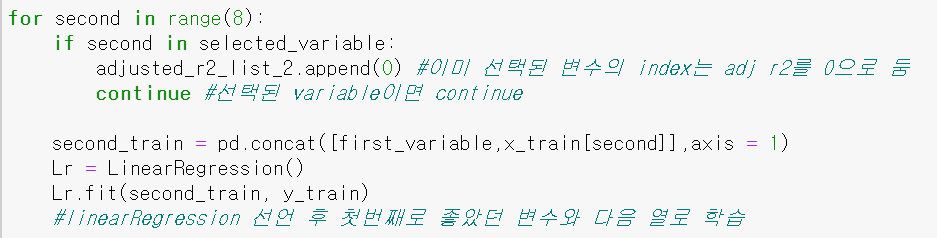
for문을 사용하여, x\_train의 feature를 0~7을 선택한 후, 학습한다.



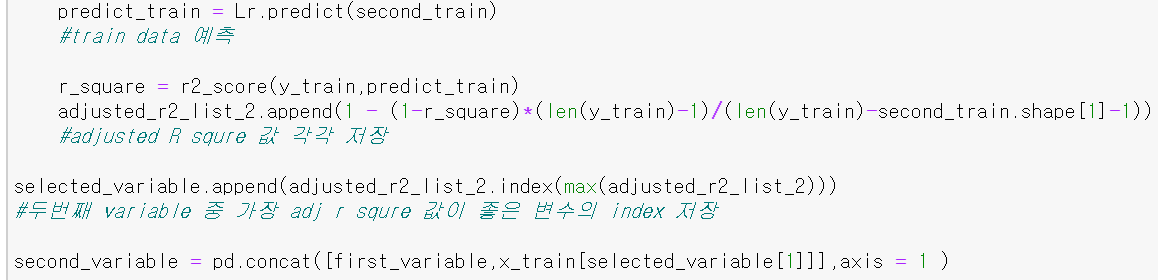
그리고 각 경우의 train data의 R Square값을 sklearn의 r2\_score 함수에 기반하여 구한 뒤, Adjusted R Square 값을 구한다. 그리고, list로 저장한다. 과제 요구사항이 Adjusted 을 기반으로 forward selection을 수행하는 것이므로, 가장 높은 Adjusted 의 값을 가지는 variable을 첫번째 변수로 선택한다.



이렇게 selected\_variable[0]에 첫번째, variable이 들어가고, 이 feature만을 이용하여 x\_train의 DataFrame을 만든다.



다음으로 두번째 variable를 선택한다. 이 때, 첫 variable 선택하는 경우와 루틴은 거의 비슷하다. 똑같이 for문을 0~7까지 도는데, 이때, 이미 선택된 변수라면 for문을 진행하지 않고 통과한다. 그리고 이미 선택된 변수의 index의 내부 값을 0으로 두어서 최종 두번째 variable 판단할 때, 선택되지 않도록 한다. 그리고 각 경우마다, 첫번째 변수 선택 후, 생성한 dataframe과 concat함수를 사용하여 열이 2개인 x\_train을 생성하고 학습시킨다.

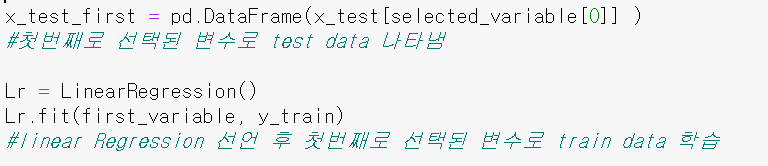


그리고 각 adjusted R Square값을 list에 저장해 놓은 뒤, 가장 높은 adjusted R square값을 가지는 index를 selected\_variable에 저장하여서 2번째 variable을 선택한다. 그리고 선택된 2번째 variable을 첫번째 선택된 variable과 결합하여 열이 두개인 x\_train data를 생성한다. 3번째variable 선택과 4번째 variable 선택은 2번째 variable 선택과 같으므로 설명하지 않겠다.

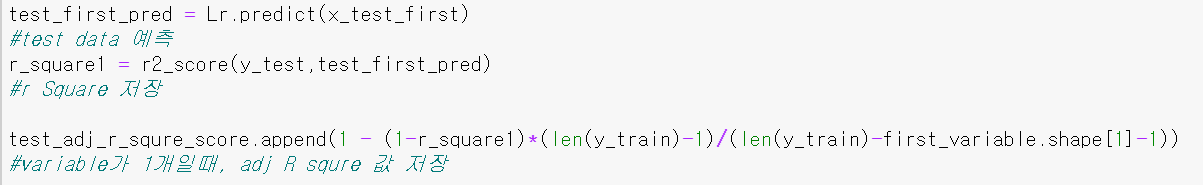


최종적으로 forward selection을 통해서 선택된 4개의 feature로 구성된 x\_train data를 구하였다.

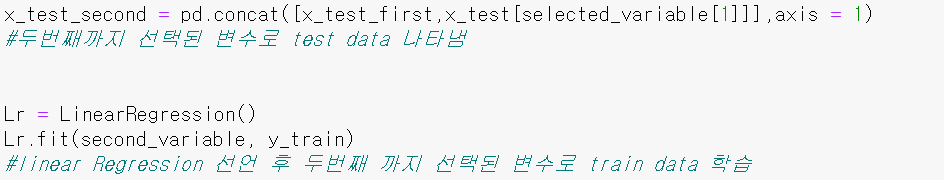
A-4. variable 개수에 따른 test data의 Adjusted 구하기



우선 x\_test data를 train 단계에서 선택된 첫번쨰 feature로 DataFrame을 만들고, linear regression을 variable 하나인 train data로 학습시킨다.

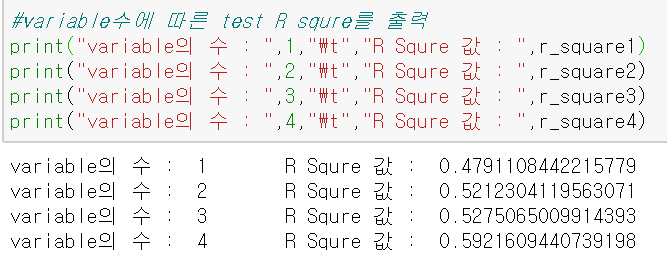


그리고 test data의 Adjusted R Square 값을 저장한다.



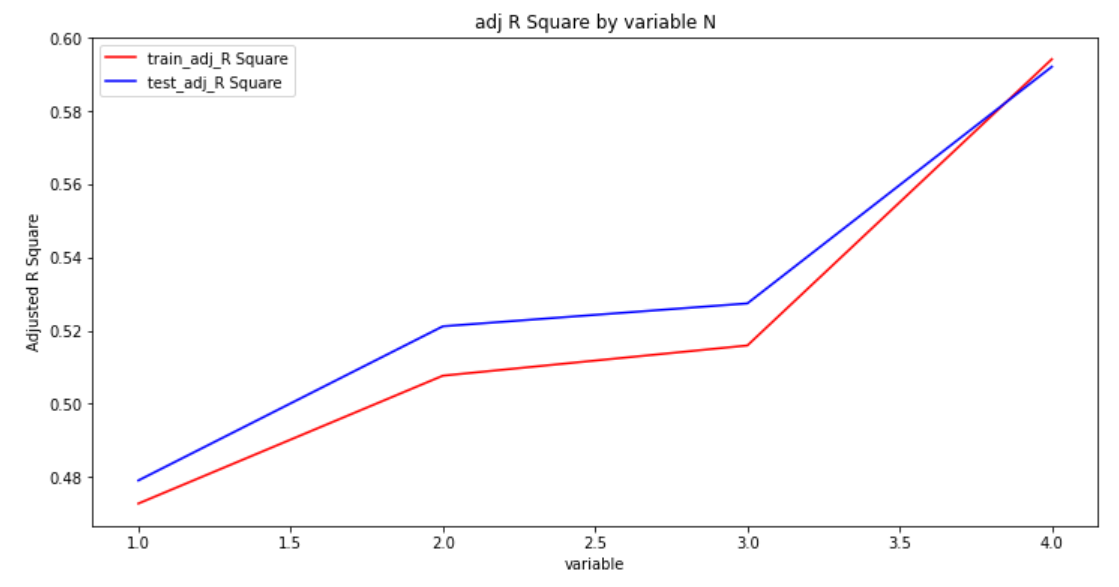
다음으로 train 단계에서 첫번째, 두번째에 선택된 variable 로 이루어진 test dataFrame을 만들고 variable 2개인 train data로 학습시킨 후, Adjusted R Square 값을 저장한다. 3번째, 4번째 경우도 마찬가지이므로 설명하지 않는다.

A-5. variable 수에 따른 test 출력



위와 같이 variable의 수에 따른 R Square 값을 나타냈다.

A-6. 테스트 데이터의 Adjusted 과 학습데이터의 Adjusted 를 **그래프를 이용해 출력**



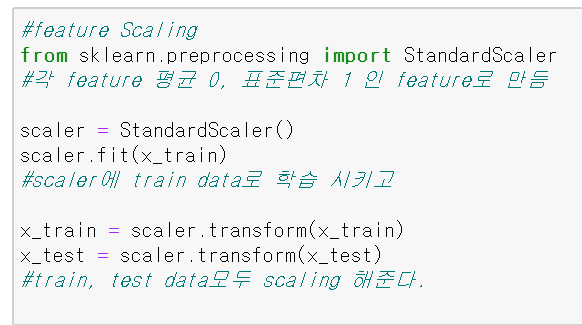
위의 그래프를 보면, variable의 수에 따른 test data와 train data의 Adjusted 값을 그래프로 나타냈다. Train data와 test data의 그래프 추이가 비슷하다. 또한 variable의 수가 많아질수록 모두 상향 그래프를 그린다. 즉, feature가 너무 적으면 잘못된 판단을 할 확률이 올라간다고 할 수 있다. 또한 forward selection을 통한 variable 선택이 잘 이루어 졌음을 나타낸다.

1. **Regularized linear model (linear regression)**
   1. Regularization 방법 중 하나인 lasso regression을 이용하여 5-fold cross validation을 통해 최적 하이퍼파라미터 탐색 후, 최종 모델의 test 를 출력한다. [10]

A-1. x\_train, x\_test data 구성

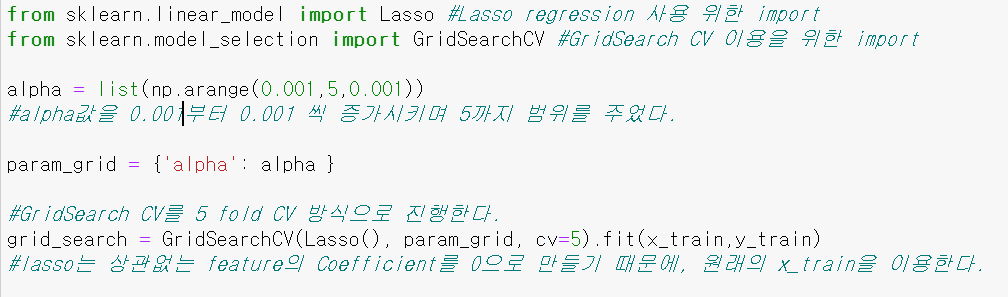


Lasso regression은 전체 feature중 관련이 없는 feature의 Coefficient를 정확히 0으로 만들기 때문에, feature 선택을 진행하지 않아도 된다.

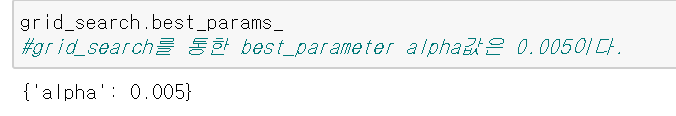


Lasso regression의 5-fold 방식인 Grid search CV 진행하기 전에, feature scaling을 진행한다.

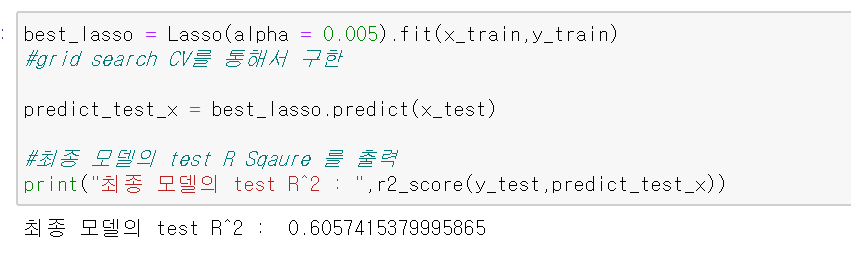
A-2. lasso regression을 이용하여 5-fold cross validation을 통해 최적 hyperparameter 탐색



Hyperparameter인 alpha 값의 최적의 hyperparamter을 찾기 위해서, Lasso regression을 5 fold CV방식으로 Grid Search CV를 진행한다. 이 때, alpha 값을 0.001부터 5까지, 0.001씩 증가시키면서 테스트한다. 여기서 0을 제외한 이유는, alpha가 0이라면 penalty 값의 영향이 완전히 없어지기 때문에, Lasso를 쓰지 않은 linear regression과 같아진다.



gridSearch CV를 통한 최적의 hyperparameter는 0.005이다.

위의 최적의 hyperparameter를 바탕으로 최종 모델을 만들고, test R Square값을 출력한다.

최종 모델의 test R Square 값은 0.6057… 이다.