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

**Homework 6**

Due May 7

**과제목표**

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

**제출**

* **요구한대로 작성한 보고서와 주피터 노트북(주석 필수)**
* **최종 점수는 비율적으로 수정해서 들어갈 예정**

**Linear regression용 데이터**

캘리포니아 주택가격 데이터셋을 사용, y값은 주택중앙값을 의미.  **homework 2 (assignment 1)** 과제에 올려진 주피터 노트북의 데이터 로드 방식을 따라 과제를 진행.

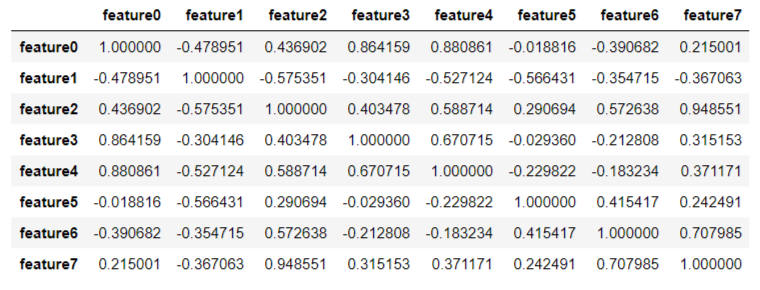
**Logistic regression용 데이터**

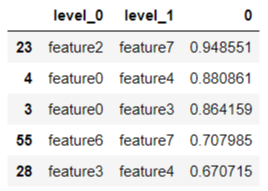
주어진 주피터 노트북에서 make\_classification 메서드로 생성된 데이터를 이용하여 진행, 데이터 생성 부분 코드의 경우 학생이 수정하면 안됨.

**scikit-learn 라이브러리를 바탕으로 과제를 진행한다**. scikit-learn 라이브러리에 대한 매뉴얼 홈페이지는 다음과 같다.

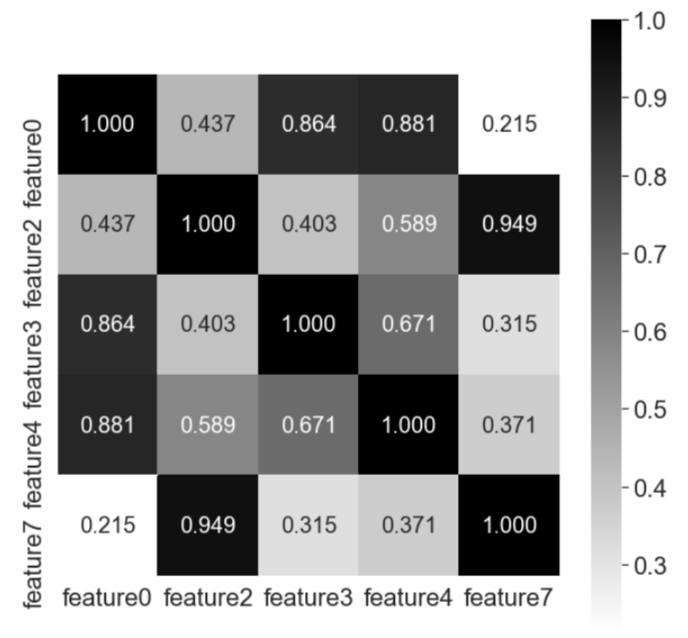
<https://scikit-learn.org/stable/>

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]



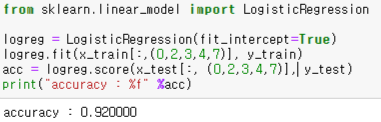


**그림1. feature들간의 correlation coefficient 값과 top 5**



**그림2. 해당 feature들 간의 correlation값 heatmap화**

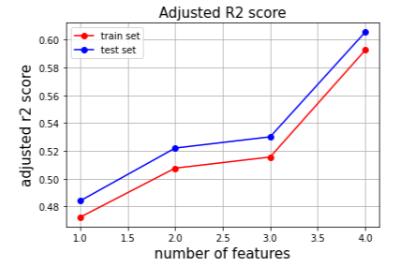
* 1. 위에서 구한 correlation coefficient array를 바탕으로 multiple **logistic** regression 모델에서 사용할 feature를 정하고 90%의 데이터를 학습데이터로 사용하여 regression 모델 학습 후 10% 테스트 데이터에 대한 **test acc를 출력**한다. [5]



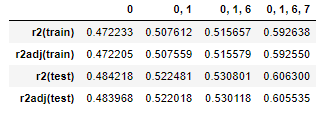
**그림3. Logistic regression 모델의 accuracy**

* 문제 A에서 2번 feature와 7번, 0번과 4번, 0번과 3번간의 correlation coefficient가 높게 측정되었으므로, 0, 2, 3, 4, 7번 feature를 사용했으며, train set의 해당 feature들만 고려하여 학습시킨 뒤, test set으로 측정해본 결과, 정확도는 0.92으로 높게 측정되었다.
* Linear regression은 주어진 x로 y를 예측한다면, logistic regression은 주어진 x로 P(y) 즉 y가 될 확률을 예측하며 예측 값을 통해 이분법적 또는 multi-class로 분류하는 classification 문제를 푸는 방식 중 하나이다. 따라서 데이터가 적합한 회귀선에 얼마나 가까운지에 대한 측정값, 즉 회귀선의 설명력을 나타내는 r-squared score로 logistic model의 성능을 설명하는 것은 올바르지 않다.

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]
      * forward selection 을 이용하여 variable을 4개 까지 선택한다. (for 문 이용)
      * sklearn의 r2\_score 함수에 기반하여 Adjusted 를 만든다.



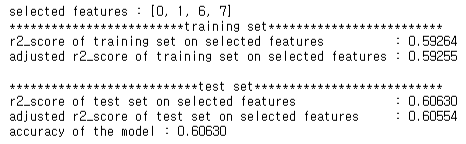
**그림1. Test, train set의 adjusted r2 score**



**그림2. Feature가 4개까지 선택될 때 까지 feature 개수에 따른 r2, adjusted r2 score**

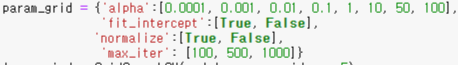
위 그림2의 columns는 feature의 index이다(0은 MedInc, 1은 HouseAge…..). 각 단계마다 r2 score가 가장 높은 feature들이 선택되었으며, 최종적으로 뽑힌 4개의 feature는 MedInc, HouseAge, Latitude, Longitude이며, r2 점수와 adjusted r2 점수가 점점 향상되는 것을 확인할 수 있었다. Sample의 수가 적지 않기에 adjusted R2와 R2의 차이가 크게 나지 않았다. Feature의 수를 늘려 5개 이상 선택했을 때부터는 4개를 선택했을 때와 크게 차이가 나지 않았다(R2, adjusted R2 모두 0.605 근처였다). 따라서 해당 데이터의 target(Median Value of house)은 다른 추가적인 변수들에 의해 크게 영향을 받지 않으며 예측 모델의 복잡성만 커지게 된다. 따라서 해당 데이터에서 위의 4개 변수를 사용했을 때, 모델의 복잡성도 크지 않으며 성능도 높아진다.

(r squared score는 선형 회귀 모델에 대한 적합성에 대한 측정도인데, 이는 선형 회귀 모델의 성능을 의미하므로, accuracy와 r2 score가 동일한 것을 알 수 있다.)



**그림3. 4개의 feature만 고려했을 때의 accuracy와 r2 score**

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]
   * Scikit learn의 라이브러리를 기반하여 수행한다.

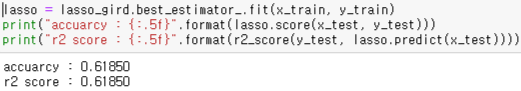


**그림1. 설정한 parameter 중 best parameters.**

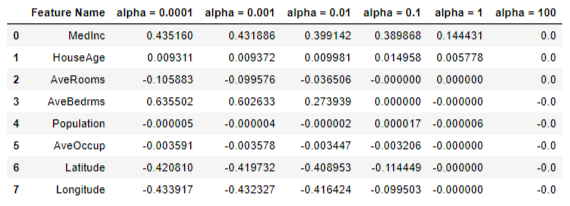


**그림2. Best parameters**

* 위 그림1과 같이 사용해볼 파라미터들을 설정하였으며, train set에 대해 최적인 모델은 그림2와 같이 설정되었을 때였다. 위 모델로 lasso 모델을 설정하여 test set으로 성능 측정 결과 0.6185의 정확도를 보였으며, 이는 r2 score결과와 일치하였다.



**그림3. 최적 lasso 모델에서의 정확도와 r2 score.**



**그림4. alpha값에 따른 각 feature의 coefficient값 변화**

위 그림4는 lasso regression에 의해 실제로 coefficient값들이 0이 되는지 확인하기 위해 나타냈으며, best parameter로 선정된 alpha값이 0.001일 때, MedInc, AveBedrms, Latitude, Longitude값의 영향이 컸으며, 0.1일 때에는 AveBedrms보다는 HouseAge의 영향이 더 크게 미쳤다. (또한 위의 forward selection 방식으로 선택된 feature들의 coefficient값이 다른 feature들에 비해 높게 나왔음을 확인할 수 있다)

즉, 해당 데이터의 타겟인 California 지역의 평균 주택 가치는 평균 수입, 주택의 연식(나이), 주택의 위치(위도, 경도)에 영향을 많이 받는 것을 알 수 있다.

**참고자료**

- correlation: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html>

- heatmap: <https://seaborn.pydata.org/generated/seaborn.heatmap.html>

- forward selection: <https://www.kdnuggets.com/2018/06/step-forward-feature-selection-python.html>

- make classification: <https://datascienceschool.net/view-notebook/ec26c797cec646e295d737c522733b15/>

\* 주어진 모든 문제에 대해서 전체의 10% 데이터를 test set으로 두어 이 데이터에서의 성능을 test accuracy/로 출력한

\* lasso in sklearn : linear\_model.Lasso