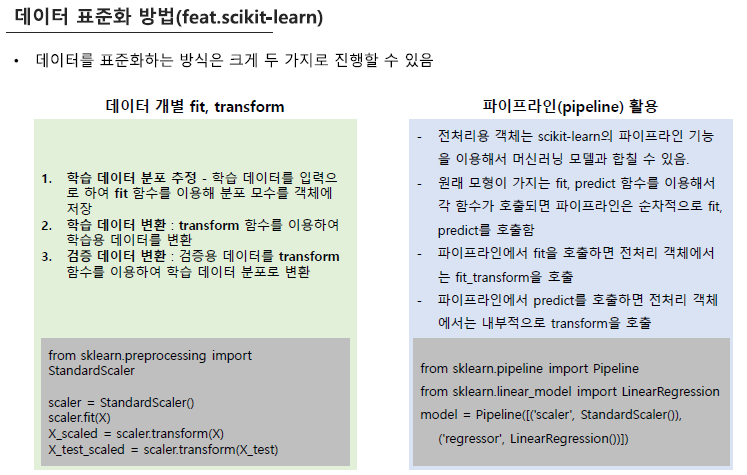


Apply - 어떠한 처리를 공통적으로 적용하겠다 라는 것



분포가 shift되는 현상을 표준화해야 함

Scaling – 제일 많이 보는 표준화



주의) 학습 데이터에서 구한 평균과 표준편차를 테스트 데이터에 적용 해야함

Scaler = StandardScaler()//객체 생성

Scaler.fit(x)//Fit을 하면 평균과 표준편차가 구해짐

X\_scaled=scaler.transform(x)//x데이터를 변환하여 가져옴

X\_test\_scaled=scaler.transform(x\_test)//테스트 데이터에 대해서 스케일링을 진행

model = Pipeline([('scaler', StandardScaler()), ('regressor', LinearRegression())])

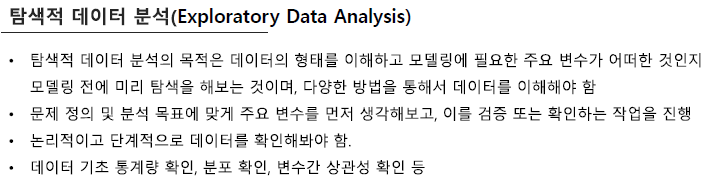
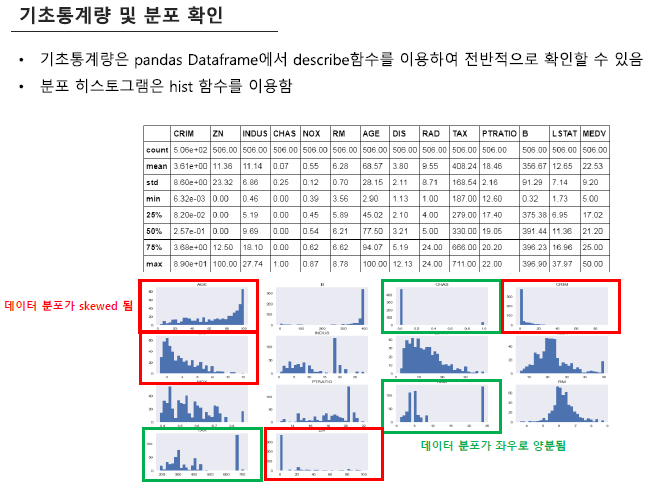
model로 한꺼번에 처리할 수 있다. (pipeline 활용) 🡪 model.fit이나 model.predict 활용

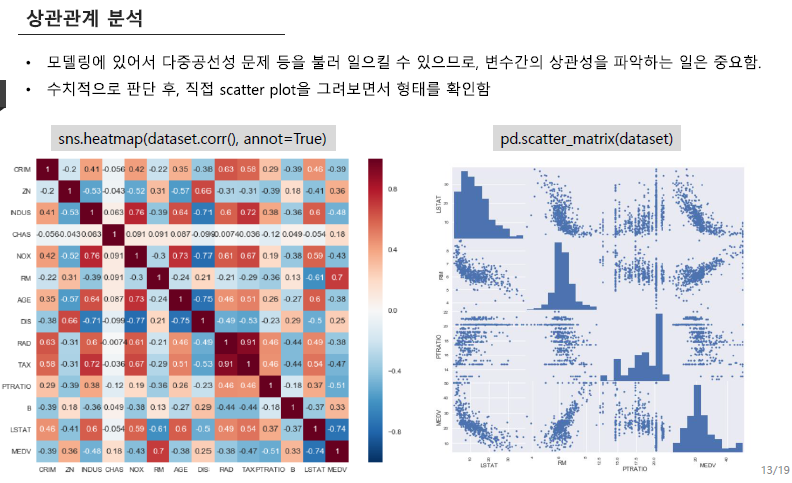
**스케일링**

스케일링은 자료 집합에 적용되는 전처리 과정으로 모든 자료에 선형 변환을 적용하여 전체 자료의 분포를 평균 0, 분산 1이 되도록 만드는 과정이다.스케일링은 자료의 오버플로우(overflow)나 언더플로우(underflow)를 방지하고 독립 변수의 공분산 행렬의 조건수(condition number)를 감소시켜 최적화 과정에서의 안정성 및 수렴 속도를 향상시킨다.

scikit-learn에서는 다음과 같은 스케일링 클래스를 제공한다.

* StandardScaler(X): 평균이 0과 표준편차가 1이 되도록 변환.
* RobustScaler(X): 중앙값(median)이 0, IQR(interquartile range)이 1이 되도록 변환.
* MinMaxScaler(X): 최대값이 각각 1, 최소값이 0이 되도록 변환
* MaxAbsScaler(X): 0을 기준으로 절대값이 가장 큰 수가 1또는 -1이 되도록 변환

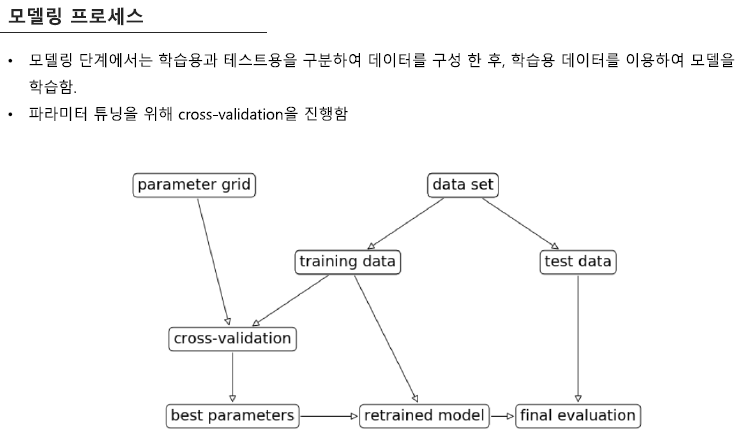


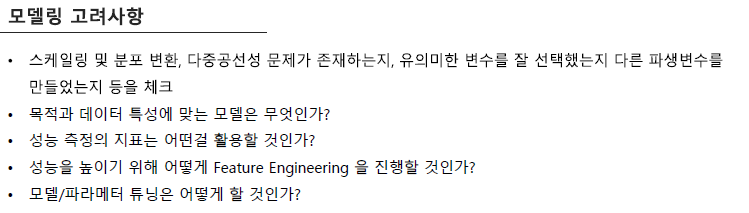


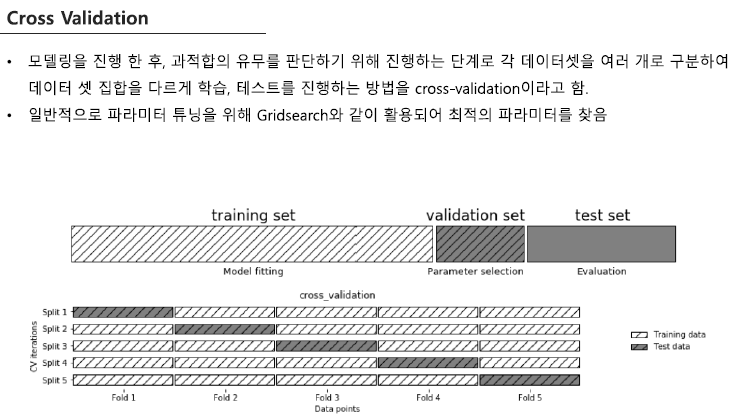
변수 간의 상관성을 수치적으로 나타냄 🡪 히트 맵을 통해 색깔별로 시각적으로 확인할 수 있다.

종속변수는 MEDV인데 0.7(일반적)을 기준으로 판단할 수 있다.

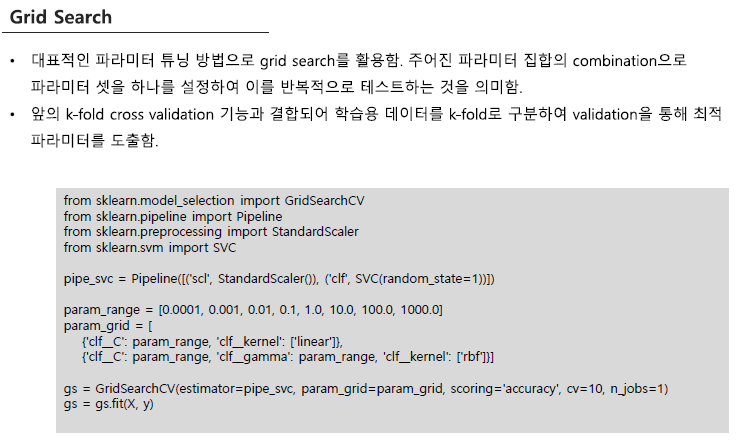
Scatterplot은 x축과 y축은 변수 별 값을 의미하므로 2차원 상에서 산 점도를 파악하여 상관 관계를 파악할 수 있다.





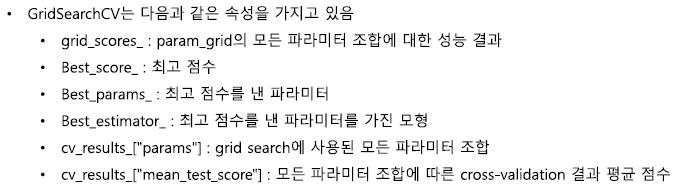


2번째 그림의 Test data는 첫 번째 그림의 validation set과 같은 부분



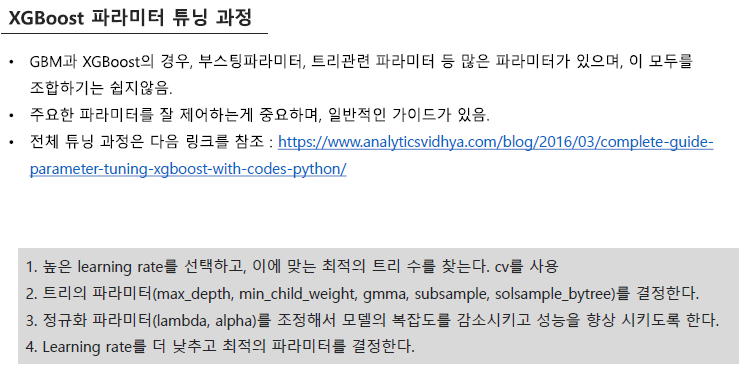
clf\_\_’무엇’와 같이 설정을 해주어야함(파이프라인에 clf라고 선언했기때문) 🡪 그래야 찾아갈 수 있음

cv는 cross validation



참고

<https://www.kdnuggets.com/2018/01/managing-machine-learning-workflows-scikit-learn-pipelines-part-2.html>



강의자료\_4\_데이터분석실습 p.20에 링크 있음

-solsample\_bytree 오타 🡪 colsample\_bytree

Gridsearch.best\_score\_

Np.sqrt(-gridsearch.best\_score)) 🡪 RMSE : cross validation을 돌렸을 때 나온 값