머신러닝

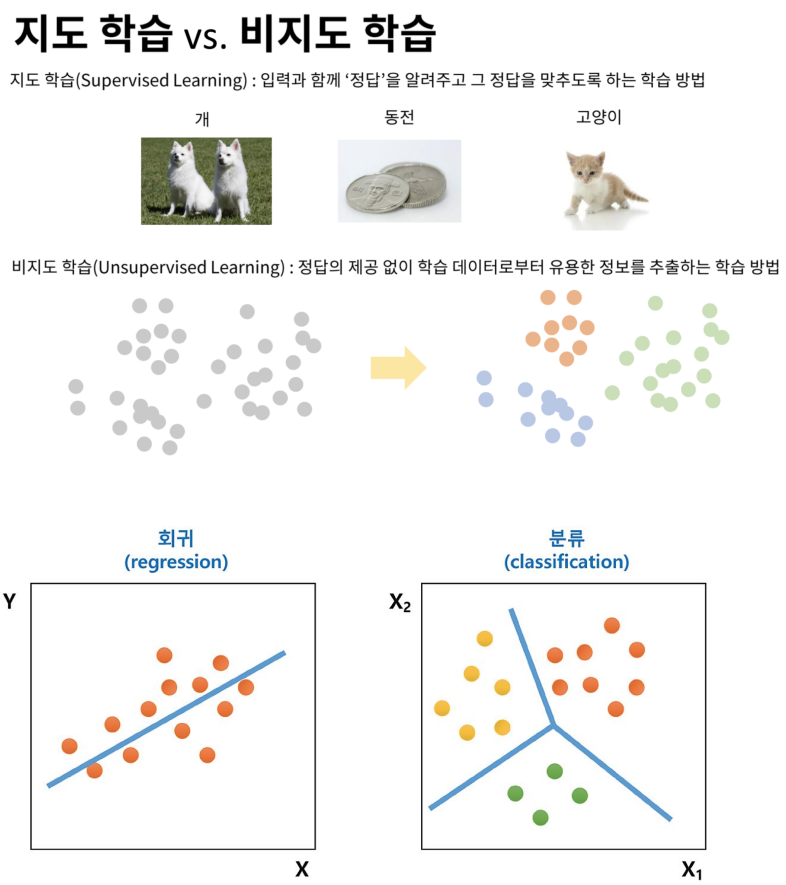
머신러닝

- 알고리즘이 데이터를 입력받은 후 스스로 학습..

- 지도학습 (supervised learning) - 예측하는(분류하는) 알고리즘
  회귀(regression) - 예측 대상이 연속값 (광고매출, 내년 성장률, 주가, 부동산가격...)
  분류(classification) - 예측 대상이 불연속값 (암, 성별, 합격여부, 대출, 등급..)

  - 이진분류(binary classification) : 두 분류(예측) - 남녀, 암, 합격불합격,...
  - 다지분류(multiclassification) : 꽃종류, 등급,

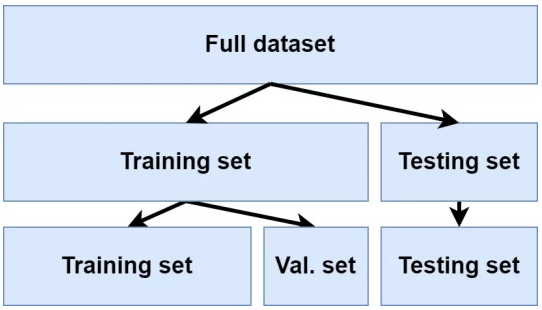
  (대부분의 알고리즘은 회귀, 분류(이진분류, 다지분류) 를 모두 수행한다)
  (LinearRegression - 회귀만, LogisticRegression - 분류)

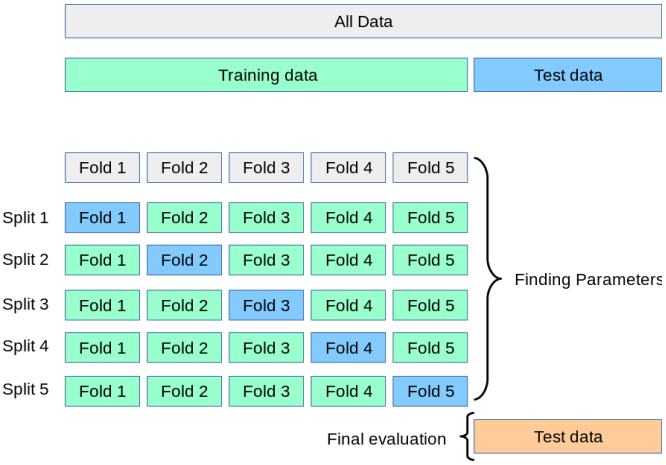


모델학습

- 입력데이터 : X (예측을 위해서 입력되는 데이터)
- 출력데이터 : Y (예측대상 - target, label, class)

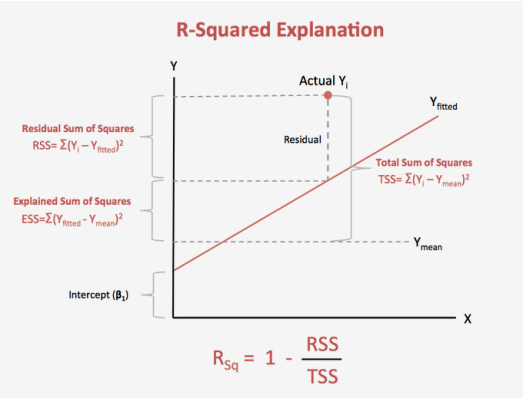
- train/ test data 분리 =>
- train data - 모델을 학습 시 사용
- test data - 학습된 모델을 평가 시 사용.. 모델의 평가가 현실에서는 매우 중요하다..
- train_test_split(): 데이터를 shuffle해서 train, test를 나눈다..





모델평가

- 회귀모델
  - Mean Squared Error(MSE) : 예측오차를 제곱하여 더한 후 평균한 것...np.mean(sigma(y_hat - y)**2), RMSE, MAE
  - R-square : 모델이 예측하는데 기여한 정도…(y_hat - y_mean)/(y - y_mean)



- 분류모델
  - accuracy - 맞춘데이터수/ 전체데이터수

- precision(정밀도) - True로 맞춘데이터수/ 모델이 True라고 예측한 데이터
- recall(재현율) - True로 맞춘데이터수/ 실제 True인 데이터
- F1_score - precision과 recall의 조화평균..
- ROC_AUC_Score - roc곡선의 아래 면적
-
- 예) 로보어드바이저 개발자 - 정밀도
- 예) 암예측 모델 - recall
-
- 임계값 - 모델은 True False의 확률값으로 예측을 하는데, 0.5를 기준으로 True, False로 분류한다..

		예측 결과	
		TRUE	FALSE
실제 정답	TRUE	TP (True Positive)	FN (False Negative)
	FALSE	FP (False Positive)	TN (True Negative)

- 정밀도, 재현율, 특이도
  - 분류 모형의 목적에 따라 다양한 지표를 볼 수 있음

클래스 ={정상, 불량}		예측한 클래스	
		정상	불량
실제 클래스	정상	TN	FP
	불량	FN	TP

$$\text{정밀도}(Precision) = \frac{\text{옳게 분류된 불량 데이터의 수}}{\text{불량으로 예측한 데이터}} = \frac{TP}{FP+TP}$$

$$\text{재현율}(Recall) = \frac{\text{옳게 분류된 불량 데이터의 수}}{\text{실제 불량 데이터의 수}} = \frac{TP}{FN+TP}$$

$$\text{특이도}(Specificity) = \frac{\text{옳게 분류된 정상 데이터의 수}}{\text{실제 정상 데이터의 수}} = \frac{TN}{TN+FP}$$

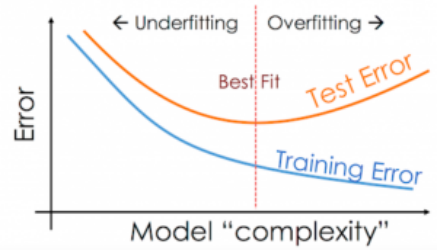
정밀도(precision)는 분류 모형이 불량을 진단하기 위해 얼마나 잘 작동했는지 보여주는 지표

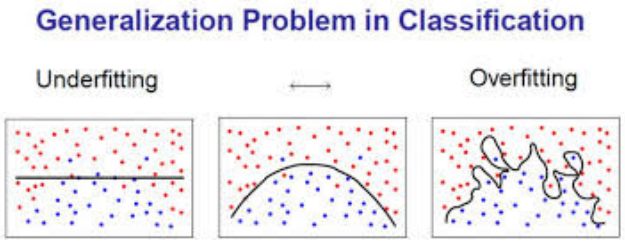
재현율(recall)은 불량 데이터중 실제로 불량이라고 진단한 제품의 비율 (진단 확률)

특이도(specificity)는 분류 모형이 정상을 진단하기 위해 잘 작동하는지를 보여주는 지표

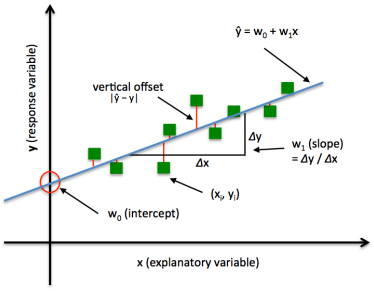
- 모델학습 - 과적합
  모델이 Train data에 과도하게 최적화 되어서 다른 데이터에 대한 예측성능이 낮아지는 것 - 일반화 성능이 낮다
  - 데이터를 더 많이 확보
  - 모델이 과도하게 학습되지 않도록 모델을 단순화 시키는 방법
  - 모델에 규제를 추가하는 방법

- 모델학습 - 과소적합
  모델이 훈련 적게되어서 train data, test data 모두에 대해서 예측성능이 낮은 경우.





- LinearRegression (회귀분석) - 실제값과 모델이 예측한 값과의 오차의 합을 최소화하는 모델. gradient descent를 사용
  hyper parameter : Ridge(alpha), Lasso(alpha), Elasticnet(alpha, l1_ratio)



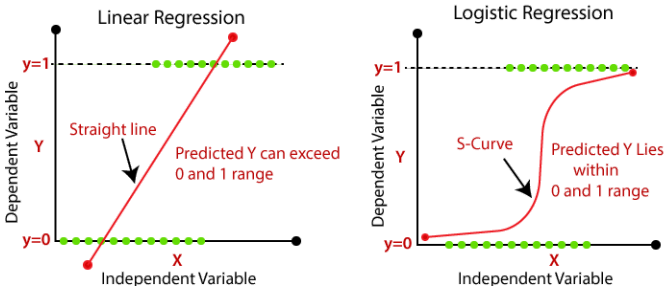
Ridge Regression

- *Ridge regression* penalize the size of the regression coefficients based on their l^2 norm:

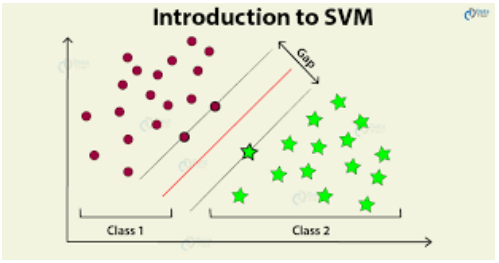
$$argmin_\beta \sum_i (y_i - \boldsymbol{\beta}' \boldsymbol{x}_i)^2 + \lambda \sum_{k=1}^{K} \beta_k^2$$

- The tuning parameter serves λ to control the relative impact of these two terms on the regression coefficient estimates.
  - Selecting a good value for λ is critical; cross-validation is used for this.

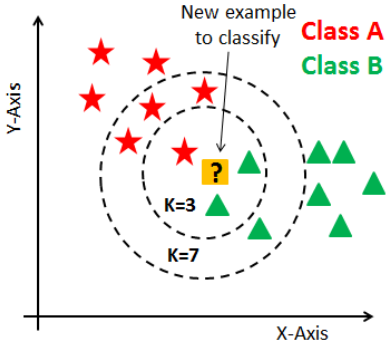
- LogisticRegression(분류분석) - 회귀모델을 sigmoid함수를 통해 'S'자 모양으로 변형, 임계값 0.5를 기준으로 0과 1로 분류
  hyper parameter : **penalty*{'l1', 'l2', 'elasticnet', 'none'}, default='l2', C***



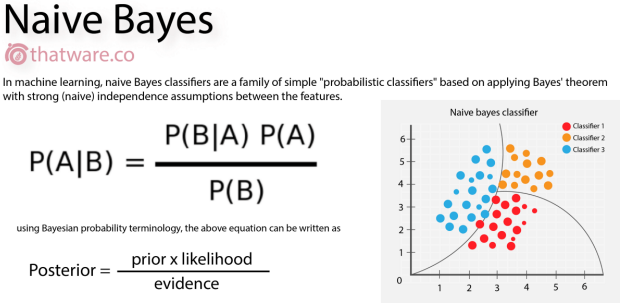
- SVC(회귀, 분류)
  hyper parameter : C, gamma



- KNN(회귀, 분류) - 가장 simple , 가장 가까운 데이터 n개를 추출해서 다수결에 의해 분류
  hyper parameter : n_neighbors

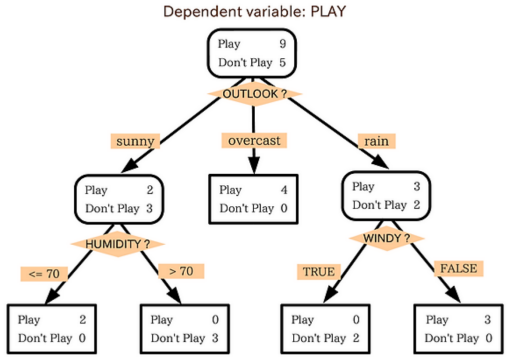


- Naivebaysian(분류) - 조건부 확률을 이용, 희소데이터에도 잘 작동 -> 텍스트분석에 많이 사용



- DT(회귀, 분류) - 현재 가장 많이 사용하는 모델.. 스무고개 찾기게임..

  hyper parameter : **max_depth, min_samples_split, min_samples_leaf**



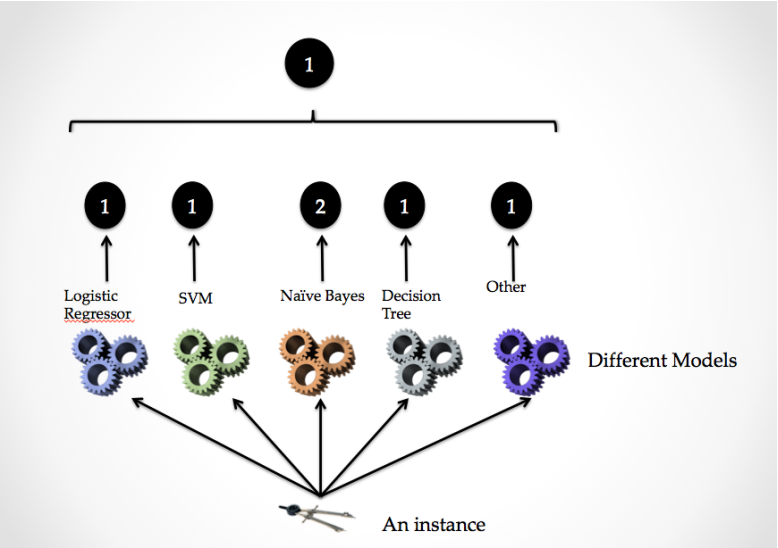
노드, 엣지, root node, leaf node, parent node, child node..
노드의 혼잡도를 계산해서 혼잡도을 낮추는 방향으로 노드의 데이터들을 쪼게어 나간다..
노드의 혼잡도 계산

1) entropy -
2) gini -

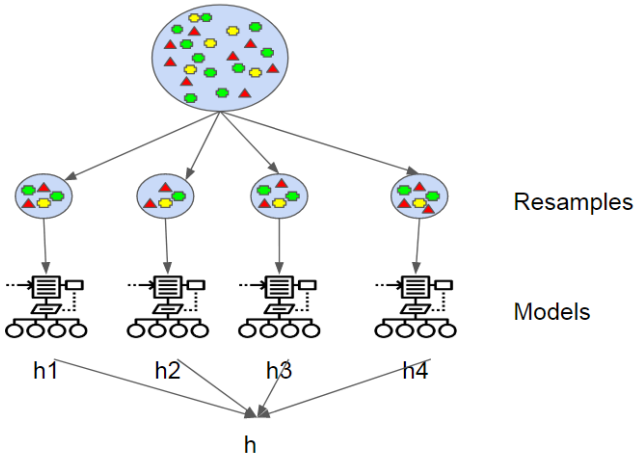
노드의 혼잡도를 가장 낮춰주는 feature를 기준으로 해당 node를 분리한다..
혼잡도가 얼마나 낮아졌는지 평가하는 지표 - infomation gain

- Ensemble Model
  여러가지 모델을 복합적으로 사용하는 방식
  decision tree를 여러개 사용해서 분류하는 방식

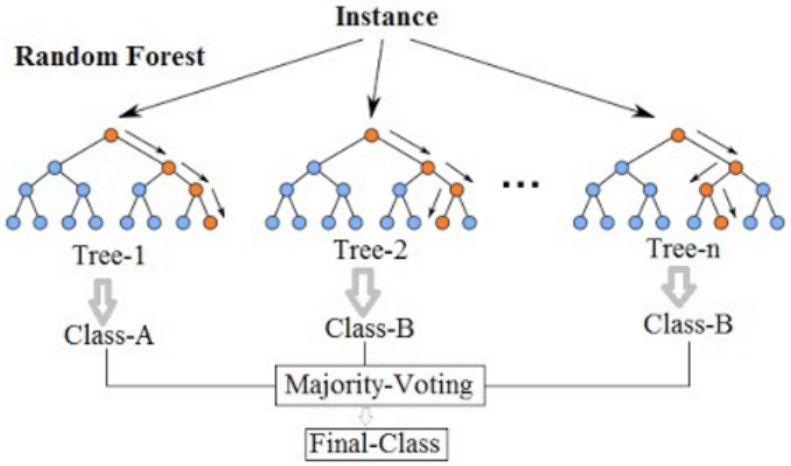
- 1. voting
  tree 포함한 이외의 다양한 종류의 모델을 사용해서 모델들이 예측한 값을 통합해서 최종 예측
  hard - 모델이 예측한 결과값(라벨)의 다수결..
  soft - 모델이 예측한 확률값을 평균해서 최종예측



- 2. bagging
  base model은 tree를 사용한다..
  boostrap방식(복원추출 샘플링)을 통해 원본데이터에서 복원추출 random sampling한 데이터로 여러모델을 학습시킨다..
  여러나무가 예측한 결과값을 통합해서 최종 예측
  hyper parameter : **n_estimators**

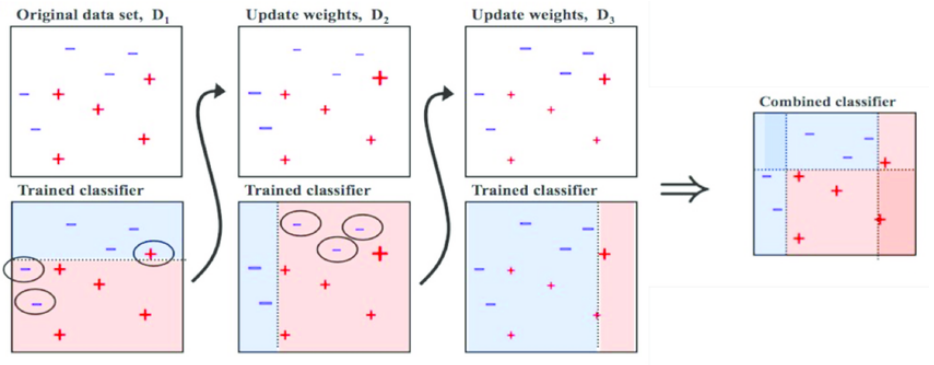


- 3. randomforest
  bagging방식을 그대로 적용한다. 그리고 아래의 방식이 추가된다..
  노드의 혼잡도를 가장 낮춰주는 feature를 선택할 때, 전체 feature를 고려하지 않고, random하게 sampling된 feature들 중에서 최적의 feature를 찾는다.
  데이터 sampling시 random방식 추가 - bagging과 같다
  feature 선택도 random방식이 추가 - randomforest 의 차이점
  hyper parameter : **n_estimators, max_depth, min_samples_split, min_samples_leaf**

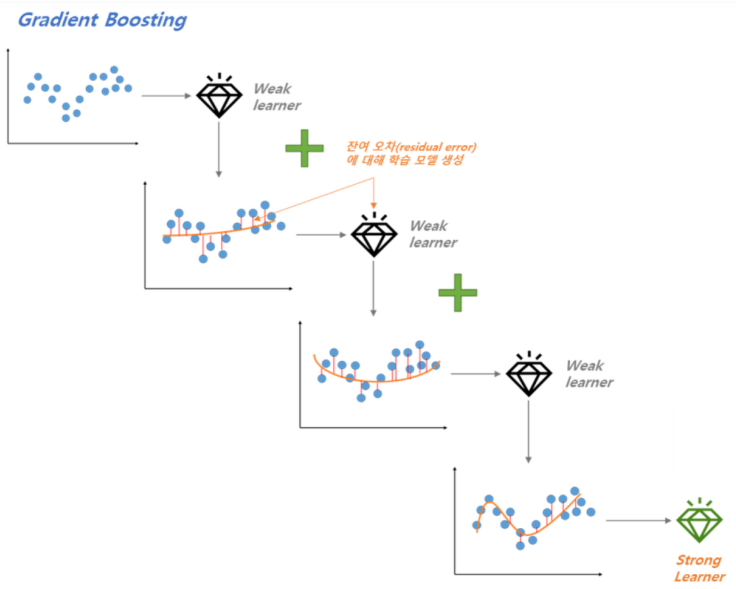


- boosting방식의 ensemble - 여러 모델을 순차적으로 학습시킨다..

- 4. adboost
  bagging이 적용된다
  데이터를 sampling할 때, 앞 모델에서 예측에 실패한 데이터를 다음 모델이 우선적으로 학습하도록 데이터를 sampling한다.
  앞 모델이 예측에 실패한 데이터를 다음 모델이 학습하도록 모델을 순차적으로 생성한다..
  hyper parameter : **n_estimators**



- 5. gradientboosting
  bagging이 적용된다
  앞 모델의 예측 오차 크기를 다음 모델이 예측하도록 순차적으로 모델을 생성한다..
  단점 - 학습에 시간이 너무 오래 걸린다..
  hyper parameter : **n_estimators, learning_rate, max_depth, min_samples_split, min_samples_leaf**



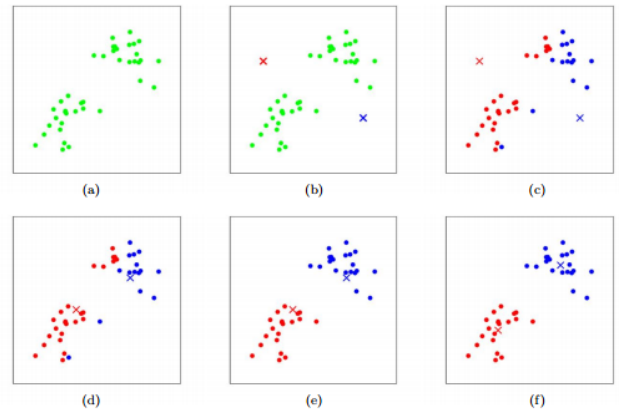
- XGBoost
  gradientboosting의 속도를 개선한 알고리즘..
  몇 기능도 개선, 사후에 가지치기 추가
  속도가 개선되었음에도 다른 알고리즘에 비해서는 속도가 느리다

- LGBM(LightGBM)
  XGBoost의 속도를 개선함
  hyper parameter : **max_depth, min_samples_split, min_samples_leaf**

- 하이퍼파라미터 최적화
  hyper parameter : 사람이 설정해주는 파라미터
  하이퍼파라미터를 조절해서 과적합이나 과소적합이 아닌 최적합 수준의 모델이 되도록 모델을 조절하는 것.
  GridSearchCV()

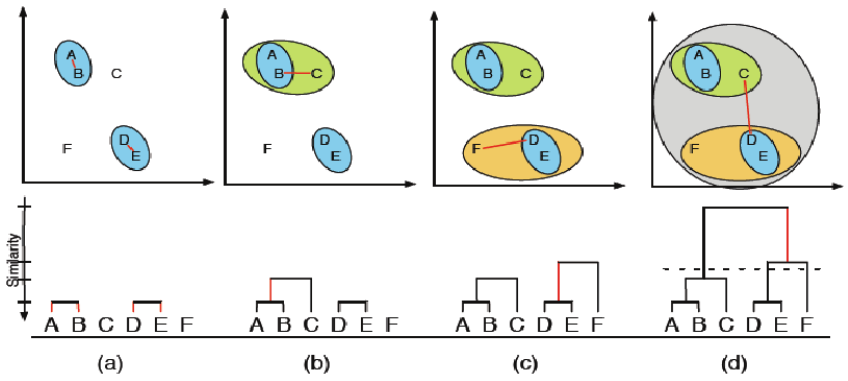
- 비지도학습
  예측대상이 존재하지 않는 알고리즘
  clustering (군집분석) : 데이터들을 동질적인 데이터끼리 묶어주는 알고리즘.
  Grouping
  데이터 전처리 과정에서 사용하는 경우.
  예) 고객세분화, 가격의 변화가 유사한 종목끼리 묶음, 기성복의 종류를 설정할 때, 유통회사에서 지역을 분할...

- kmeans
  거리기반으로 데이터들를 center points 중심으로 clustering
  새로운 cluster중심으로 center points를 이동
  위의 과정을 반복하다가 더 이상 데이터의 변동이 없을 때, clustering을 멈춘다.

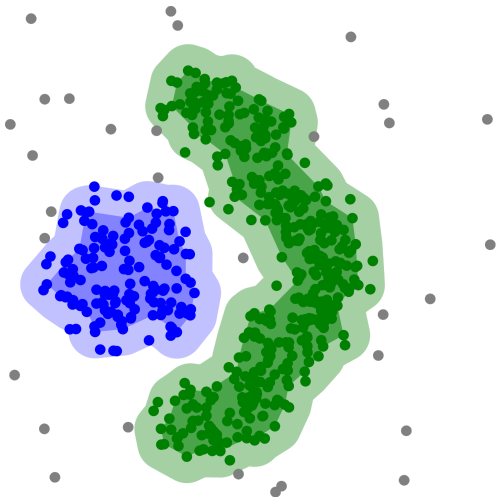


- Hierachical Clustering
  가장 가까운 거래에 있는 데이터를 하나의 cluster로 합쳐나아간다
  마지막으로 모든 데이터가 하나의 cluster가 될 때까지 진행한다.

Example: Hierarchical Agglomerative Clustering



- DBSCAN
  중심점, 경계점, 잡음점
  중심점 - 특정한 반경(e) 내에 데이터 수(num_pts)가 이상 있으면 중심점
  밀도가 높은 지역의 데이터
  경계점 - 본인은 중심점이 아니지만 다른 데이터의 밀도 범위 내에 포함되는 데이터
  잡음점 - 중심점도 경계점도 아닌 데이터
  밀도가 낮은 지역
  중심점의 경계가 겹치는 (밀도가 높은 지역)을 하나의 클러스터로 묶는다.



- clustering 평가
  clustering은 절대적인 평가지표가 존재하지 않는다.
  보조 평가 지표 : inertia_, silhouette
  inertia_ : 그룹 내의 표준편차를 계산해서 표준편차가 작을수록 좋은 clustering으로 평가
  silhouette : 그룹 내의 데이터간의 거리 평균과 그룹 간의 거리평균을 계산해서 그룹 내의 거리는 가깝고, 그룹간의 거리는 멀 수록 좋은 clustering으로 평가