boaz

머신러닝 리뷰

BOAZ 17기 박종은

BOAZ

PART 01 EDA를 위한 PANDAS, NUMPY, 시각화

PART 02 분류 및 회귀, 오버피팅과 언더피팅

PART 03 SVM, 결정나무

PART 04 앙상블, 랜덤포레스트, 부스팅, 스태킹

boaz

머신러닝 파이프라인으로 생각해봅시다

Data Collection

- 기업 : DataBase / SQL
- 학생 : Kaggle, Dacon, 공공데이터(data.go.kr
- 크롤링: Selenium. BeautifulSoup

데이터수집 | **데이터 전처리** | 피쳐 엔지니어링 | 모델링 | 하이퍼 파라미터 튜닝 | 평가

Data Preprocessing

- 시각화 활용
- 이상치, 결측치
- 데이터 정규화
    - 표준화(standardization)
    - 정규화(normalization) with "min-max normalization"*
- 탐색적 데이터분석 (EDA)
    - pandas
    - numpy
    - matplotlib

*Min-Max Scaling은 모든 피처가 정확하게 [0,1] 사이에 위치하도록 데이터를 변경한다

Feature Engineering

Raw data를 분석하기 좋은 피쳐로 만들어서 머신 러닝 알고리즘의 성능을 향상시키는 과정!

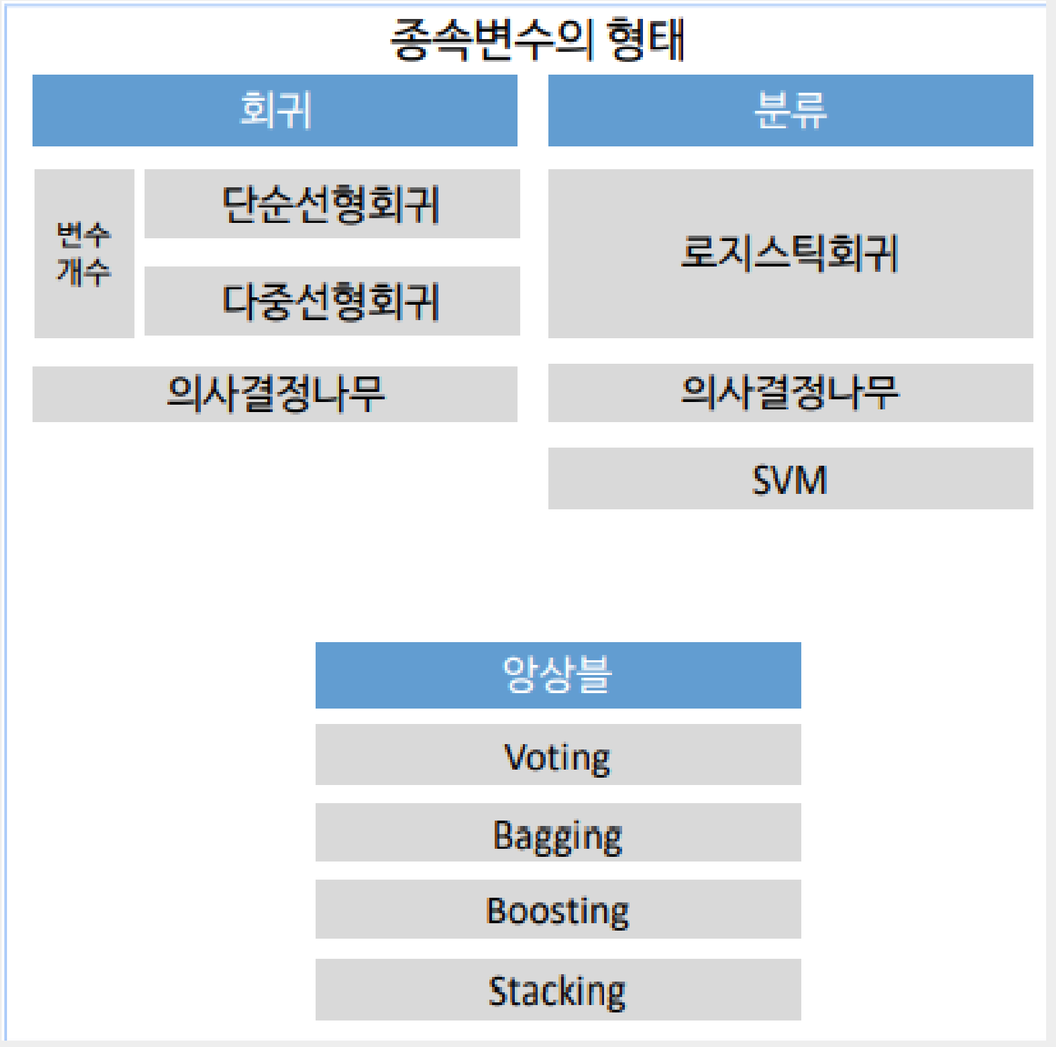
- 공모전의 수상여부가 걸려있음
- 파생변수 생성 (도메인 지식 활용)
- 피쳐 스케일링 (독립 변수 또는 데이터 피처의 범위를 정규화하는 방법)
- 정규표현식을 배워두면 유용함

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼 파라미터 튜닝 → 평가

Modeling

그동안 배웠던 모델들 (그리고 앞으로 배울 DNN, CNN) 우선 간단하게 만들고(Baseline Model)
그 뒤에 복잡한 것을 적용해보기!

- Scikit-learn (사이킷런)
    1. 모델 생성 model = DecisionTree()
    2. 모델 학습 model.fit(X_train,y_train)
    3. 모델 평가 model.score(X_test, y_test)
    4. 모델 예측 model.predict(X_unseen)

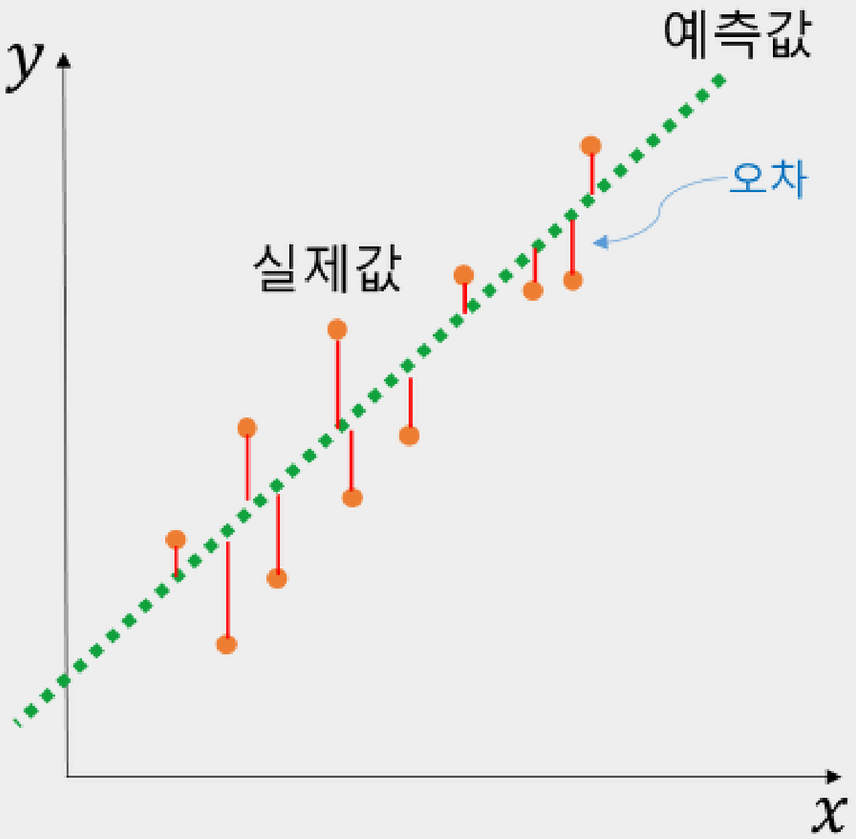


Simple Linear Regression (단순선형회귀)

- 독립변수 X가 1개인 경우
- 예측값 – 실제값의 차이를 최소화하는 직선 찾기
- 회귀식의 정확도 평가방법

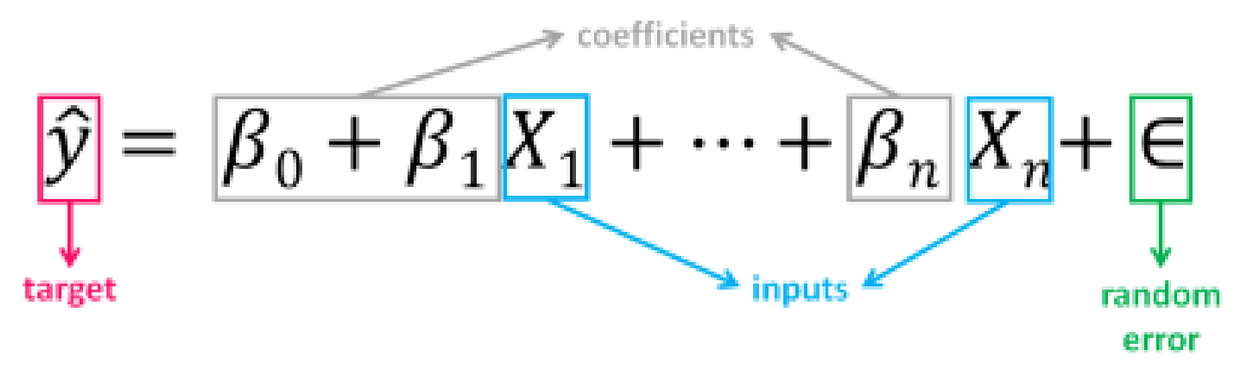
$$\hat{y} = \beta_0 + \beta_1 X + \in$$

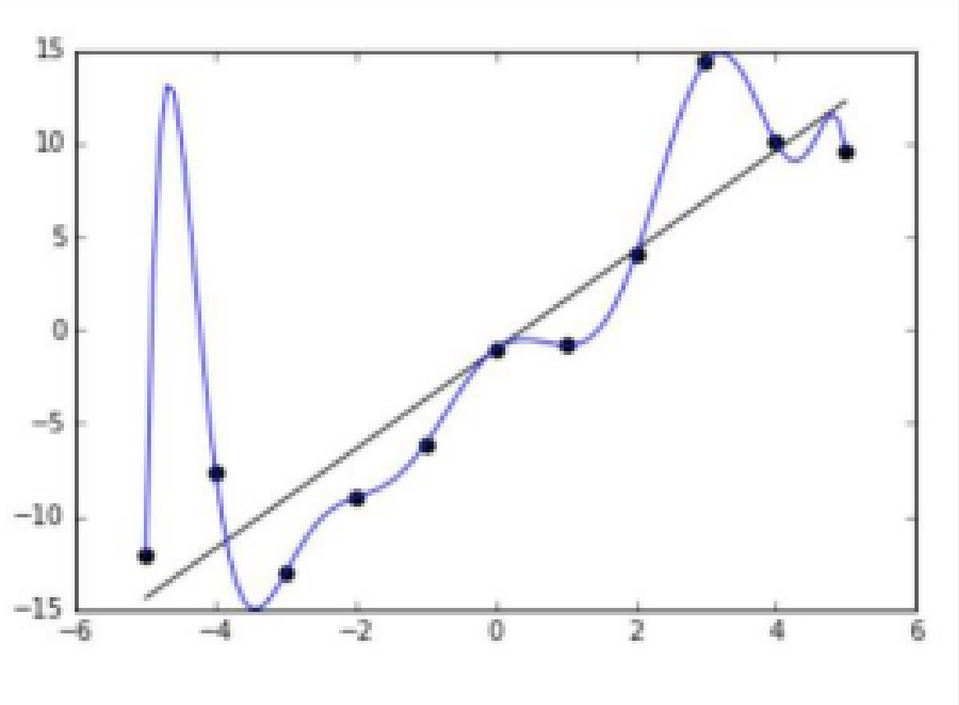
target　coefficients　input　random error



Multiple Linear Regression (다중선형회귀, 다항선형회귀)

- 독립변수 X가 여러개인 경우
- 가정 : 추정치가 선형관계, 등분산, 자기상관X, 다중공선성X
- 다중공선성에 유의 (ex 월평균음주량,혈중알코올농도→성적)





Logistic Regression (로지스틱회귀)

- 종속변수 Y가 범주형일 경우, 확률값을 계산하여 분류에 적용
- 범주에 속하면1, 속하지 않으면 0으로 (이진분류) 예측

Odds

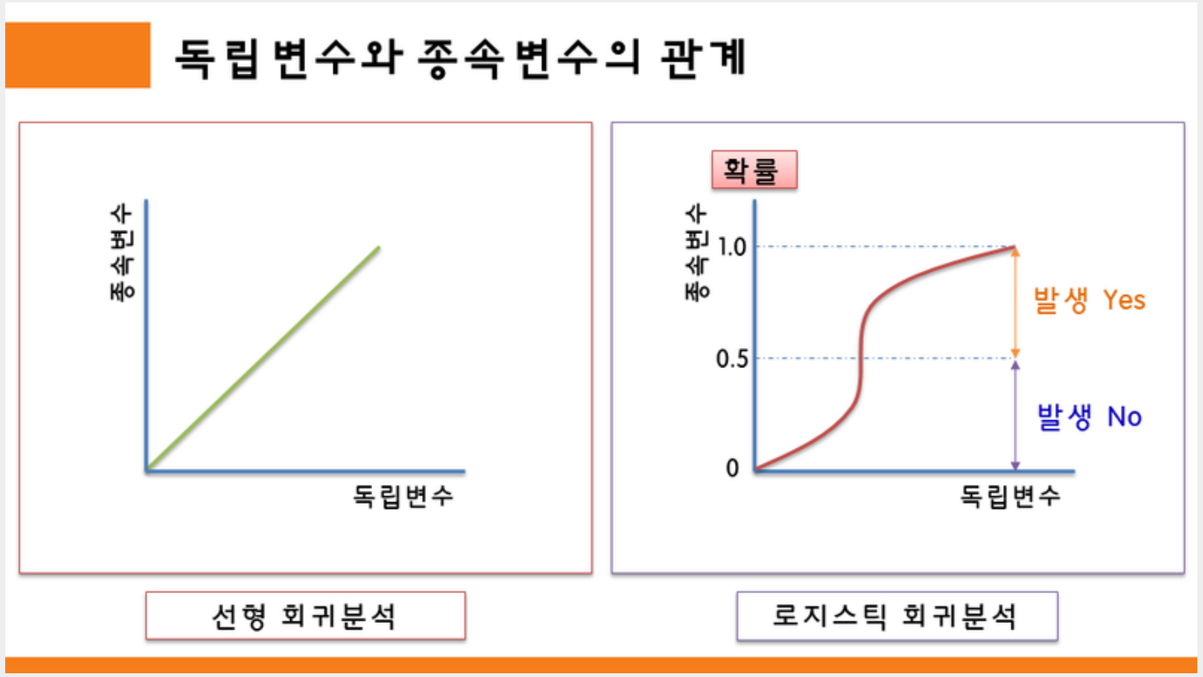
$$\text{odds} = \frac{p(y=1|x)}{1-p(y=1|x)}$$

Losit 변환 (오즈에 로그취함)

$$\text{logit}(p) = \log\frac{p}{1-p}$$

로지스틱함수

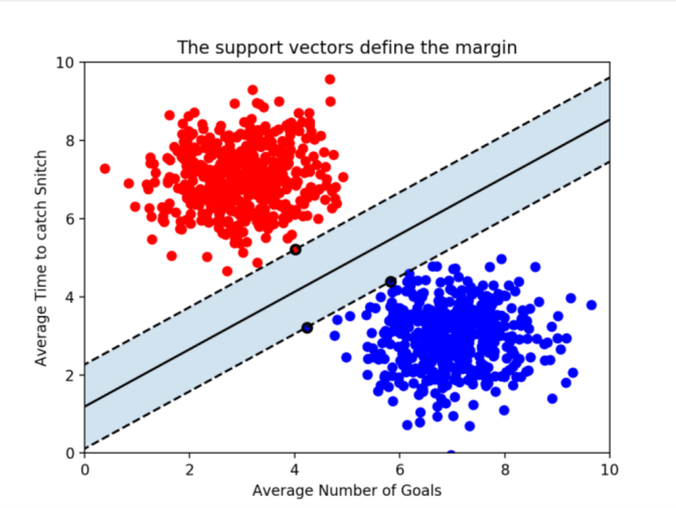
$$\text{logistic function} = \frac{e^{\beta \cdot X_i}}{1+e^{\beta \cdot X_i}}$$



Support Vector Machine (SVM)

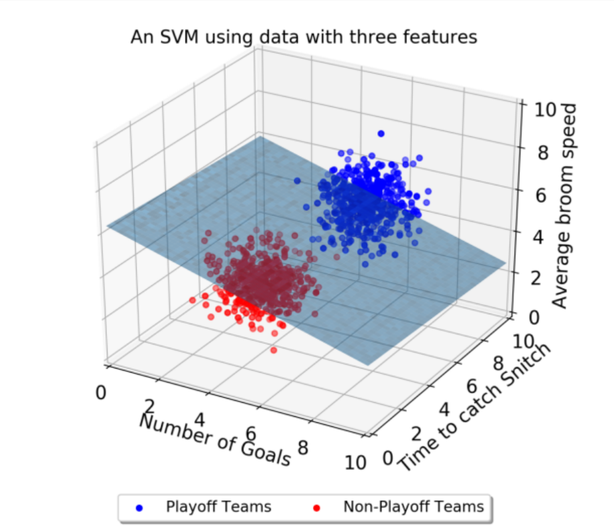
- 결정경계(분류를 위한 선)를 정의하는 방식의 분류모델
- 마진(결정경계와 서포트백터의 사이거리) 최대화
- 서포트벡터(결정경계 가까운 데이터들)만 정의해도 됨 → 속도가 매우 빠름

하드마진	소프트마진
서포트벡터-결정경계 사이 좁음	서포트벡터-결정경계 사이 멀음
마진이 작아짐	마진이 커짐
오버피팅 문제 발생 (오류 허용X)	언더피팅 문제 발생 (오류 허용O)
파라미터 C값을 크게	파라미터 C값을 작게



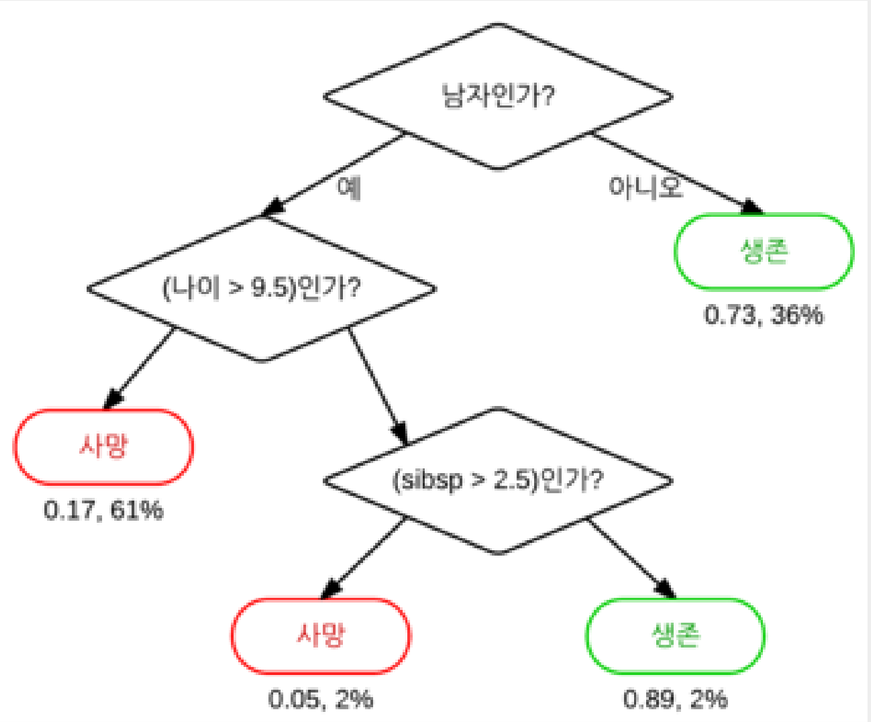
< 속성 2개일 때

속성 3개일 때 >



Decision Tree (의사결정나무)

- 분류와 회귀 모두 사용 가능
- 적절한 '분리규칙'과 '정지규칙'으로 예측값을 할당하며 학습
- 장점 : 직관적, 이상치와 노이즈 영향 적음, 모델 해석력 등
- 단점 : 일반화 어려움, 오버피팅 가능성 높음
- 분리규칙 (지니계수, 엔트로피)
- 정지규칙 : 불순도가 줄지 않음 / Sample 수 부족 / 규제매개변수 도달
- 규제매개변수 : max_depth, min_samples_split 등 → 오버피팅 막기
- 가치치기 : 마디를 잘라내어 단순화하고, 오버피팅 막기 (merge)



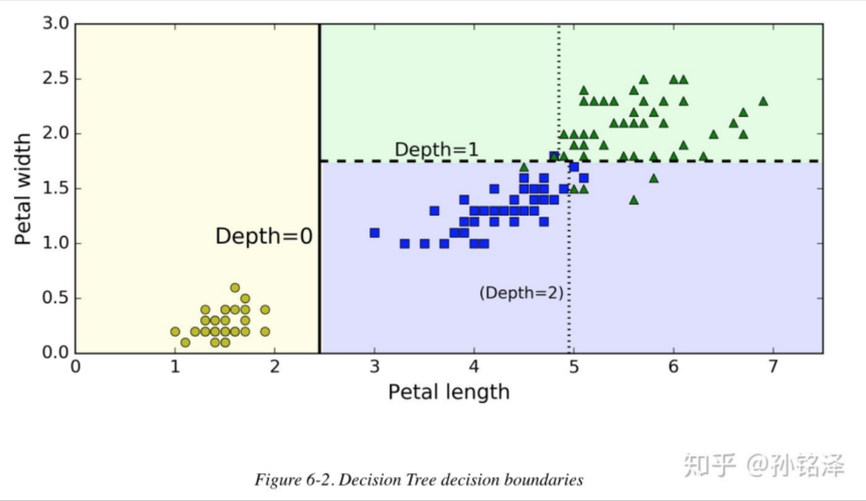


Figure 6-2. Decision Tree decision boundaries

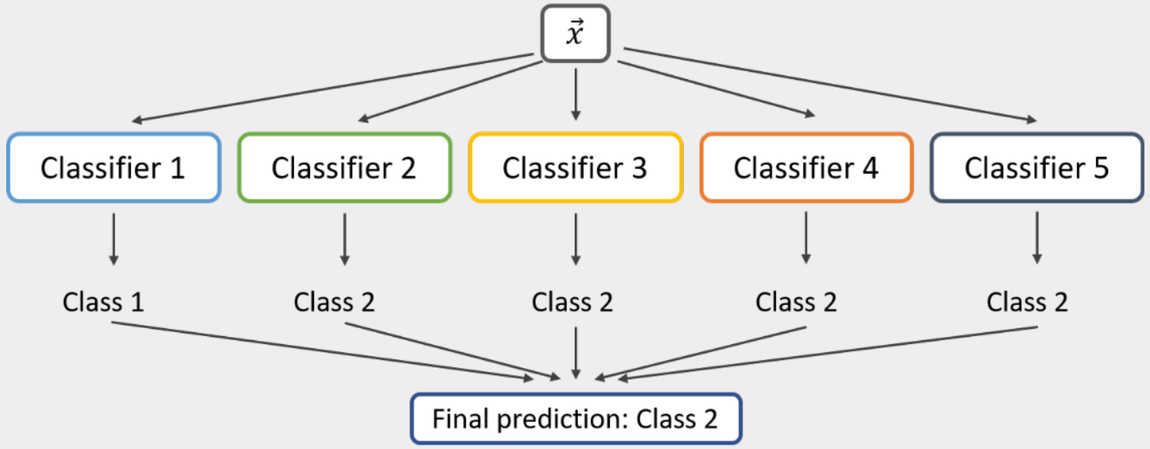
Ensemble (앙상블)

- 여러 모델을 바탕으로 새로운 모델을 만드는 방식 (집단지성)
- 보팅 : 1개의 데이터셋에 여러 모델의 예측결과로 투표하는 방식
- 배깅 : 여러 Subset에 같은 모델의 예측결과를 결합하는 방식
- 부스팅 : 앞선 모델의 틀린 예측에 가중치를 더하며 여러 모델 학습
- 스태킹 : 여러모델의 학습결과를 메타모델의 학습데이터로 재학습

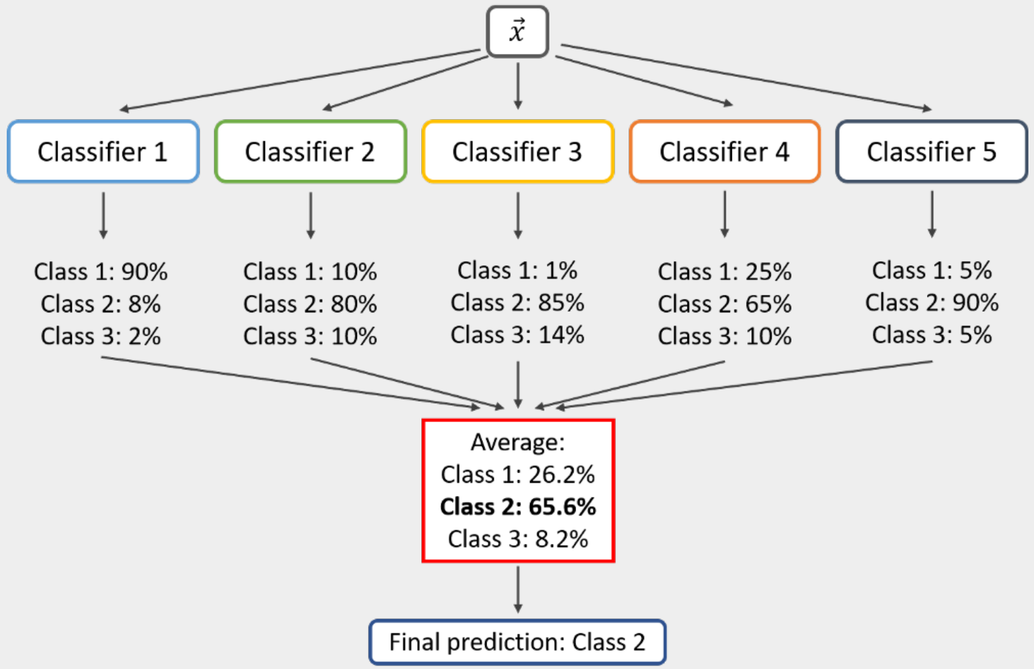
Voting (보팅)

-편향-분산 Trand-off의 효과를 극대화함

- 하드보팅 : 다수결의 방식
- 소프트보팅 : 결정확률을 더하는 예측값 방식 (일반적으로 더 나음)



하드보팅



소프트보팅

Bagging(배깅)

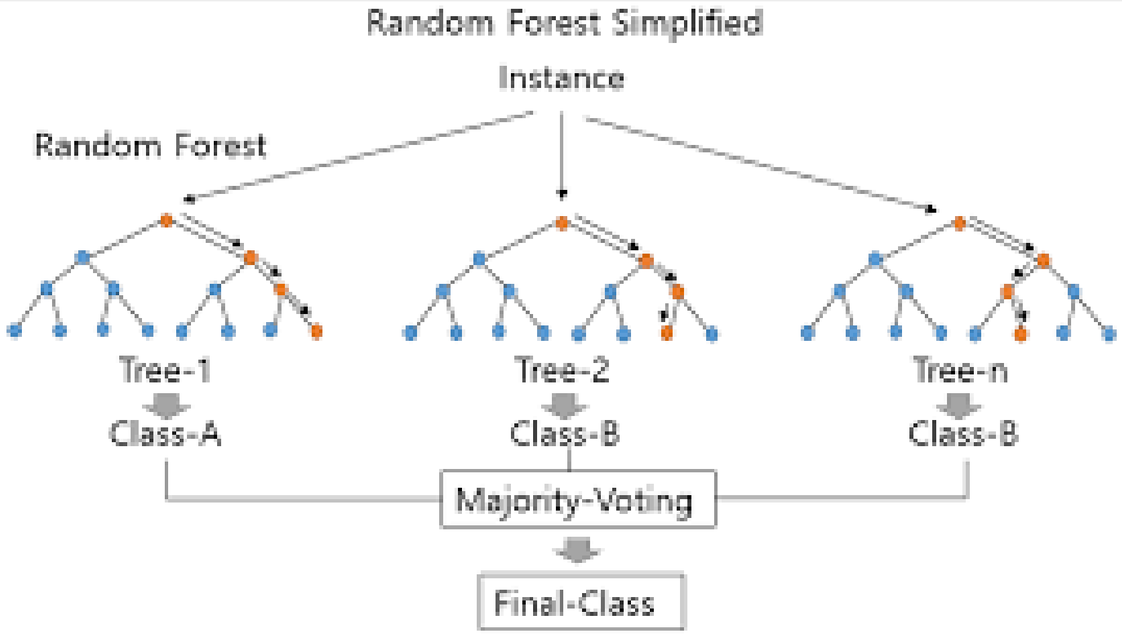
배깅 : 복원추출한 여러 Subset (= bootstrap 방식) 에 같은 모델의 예측결과 결합하는 방식

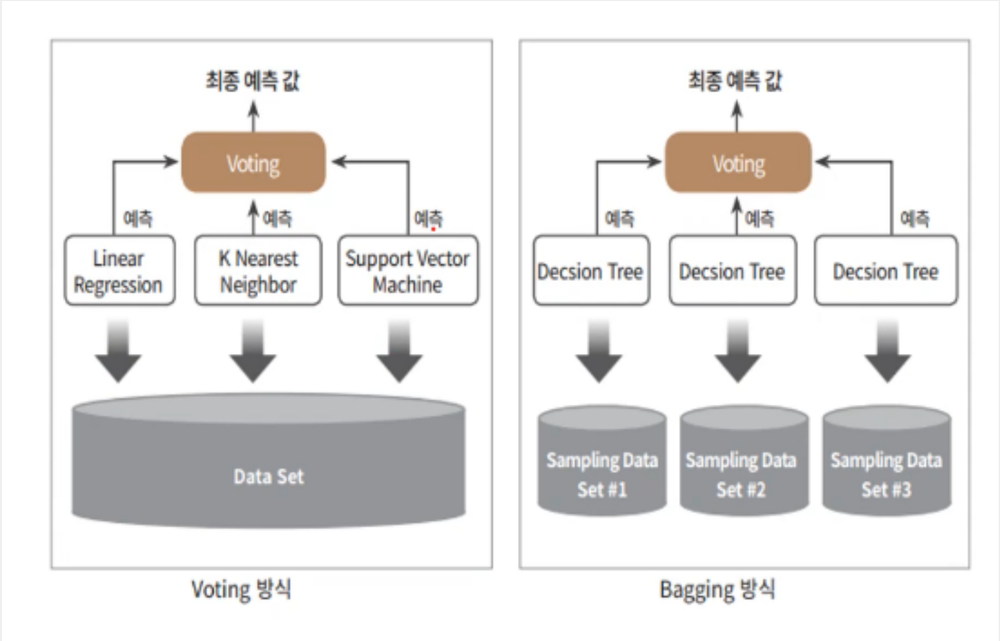
- RandomForest 모델 : 결정트리기반 알고리즘

  과적합 확률이 큰 여러 트리 결과를 투표(혹은 평균)하는 방식

  학습시간이 빠르고, 과적합 방지가 가능하며, 정확도가 좋은편

  → 병렬처리를 지원해서, 일단 Baseline으로 짜기 적절함





Boosting (부스팅)

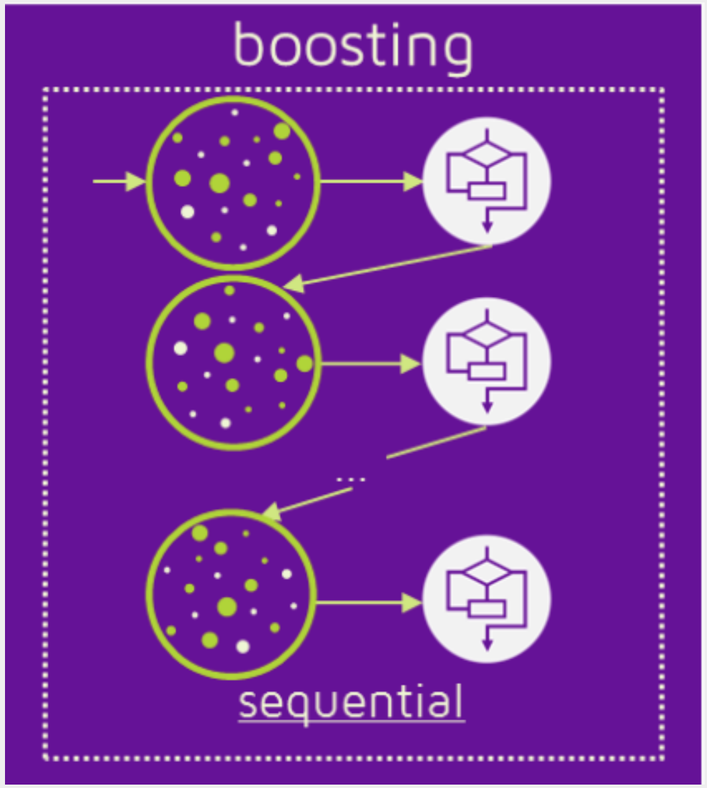
- 부스팅 : 여러 모델을 순차적으로 학습-예측 해가면서 앞선 모델의 틀린 예측에 가중치를 부여해 오류를 개선하며 학습

장점 : 오류를 개선해가기에 정확도가 높음

단점 : 순차적 진행으로 속도 느림, 오버피팅 가능성 높음

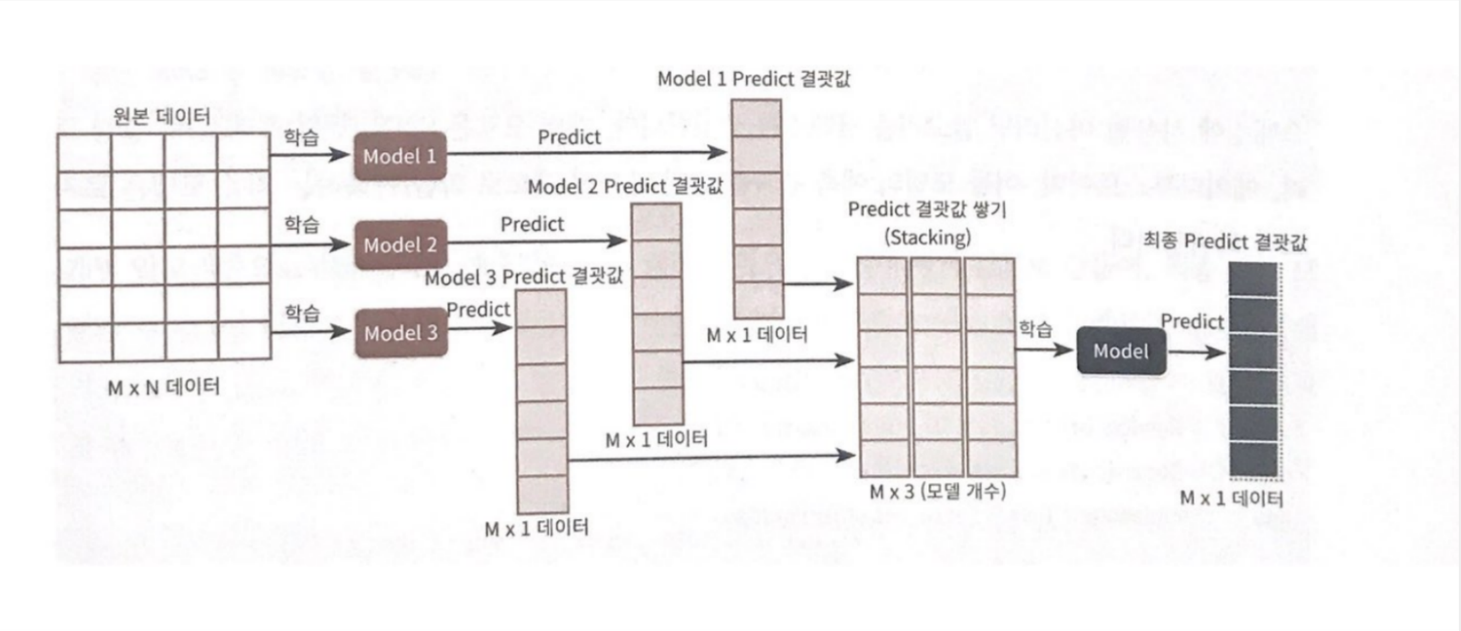
[알고리즘]

- Adaboost : 간단하고 약한 학습기간에 상호보완 순차적학습
- GBM : 잔차(Residual)를 줄여가는 방식으로 순차적학습
- XGBoost : GBM보다 성능/시간이 뛰어남, 조기중단가능,
- LGBM : 리프중심분할, XGBoot과 유사한성능, 빠른속도, 적은메모리



Stacking (스태킹)

- 스태킹 : 여러모델의 학습결과를 메타모델의 학습데이터로 재학습
- 현실에서 적용하기보단, 캐글 성능향상 목적으로 유용함!
- K-Fold CV를 이용하여 데이터셋을 나누면서 과적합 방지가능



지도학습 (supervised learning)

수치형 - 회귀

변수 개수

- 단순선형회귀
- 다중선형회귀

- 의사결정나무

명목형 - 분류

- 로지스틱회귀
- 의사결정나무
- SVM

앙상블

- Voting
- Bagging
- Boosting
- Stacking

비지도학습 (supervised learning)

군집

- K-Means
- DBSCAN
- GMM(Gaussian Mixture Model)

차원축소

- PCA
- LDA
- SVD
- NFM

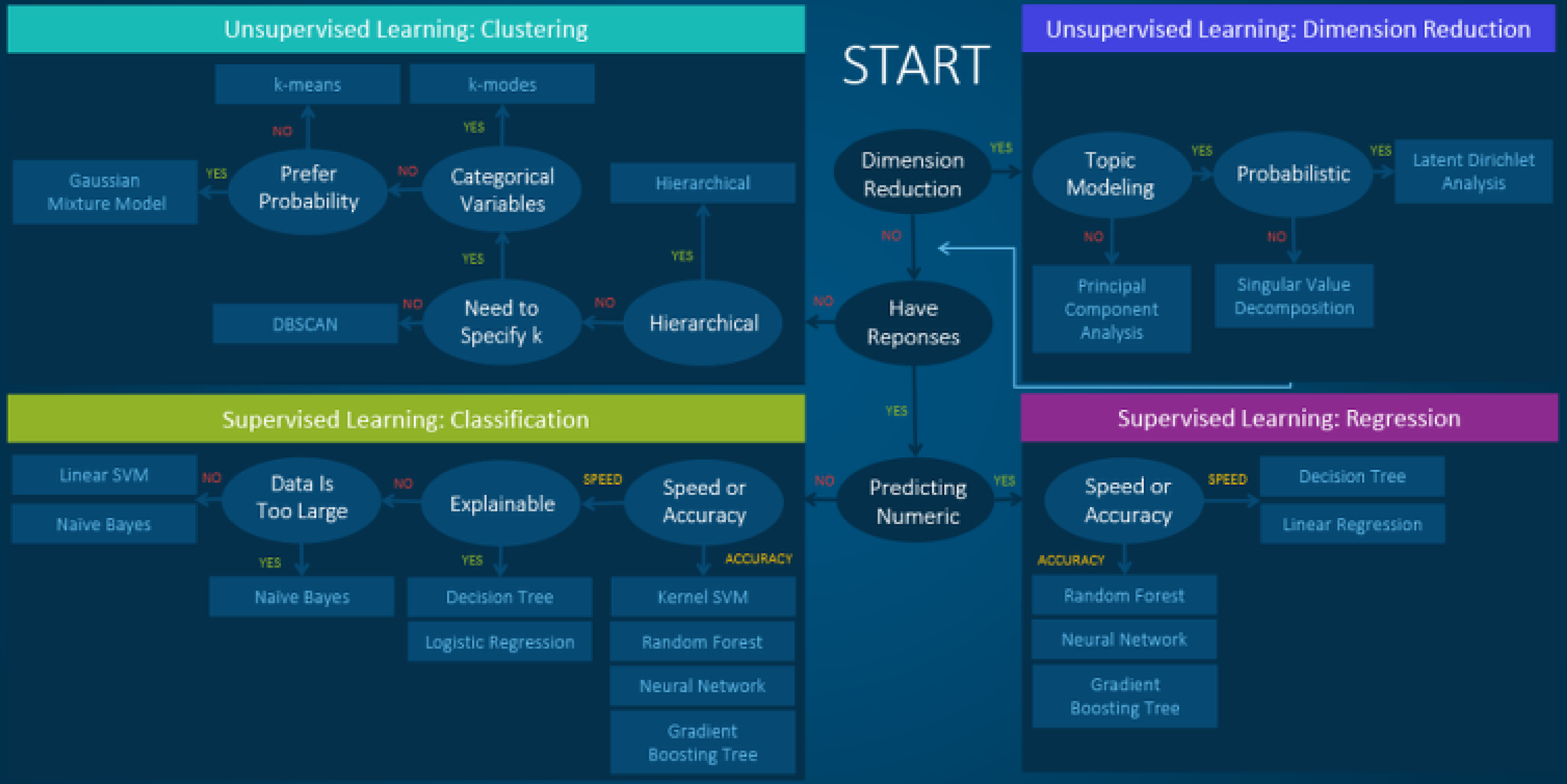
추천시스템

NLP

강화학습

.....

Machine Learning Algorithms Cheat Sheet



Hyperparameter optimization (하이퍼파라미터 튜닝)

- 그리드 서치 : 모든 경우를 테이블로 만든뒤 격자로 탐색 연산비용 큼.
처음에는 넓은 간격으로 탐색하는 것이 좋음
- 랜덤 서치 : 하이퍼 파라미터 값을 랜덤하게 탐색

Evaluation (평가)

- 학습, 검증, 평가 데이터
- MSE, RMSE, MAE …
- 오버피팅 / 언더피팅 해결
  - 언더피팅: 피쳐 수 늘리기, 학습을 더 반복하기
  - 오버피팅: 피쳐 수 줄이기, 데이터의 양 늘리기,
  - 교차검증 사용하기, L1, L2 규제 (정규화)
- 교차검증

감사합니다!