머신러닝 리뷰

BOAZ 16기 분석 박은지

2021. 08. 19

지금까지 무얼 배웠죠?

1주차 - EDA를 위한 Pandas, numpy, 시각화

2주차

- 머신러닝과 딥러닝? 회귀 vs분류
- 단순선형회귀 : 최소제곱법, OLSE, MSE, RMSE
- 다중선형회귀 : 다중공선성
- 로지스틱회귀 : MLE, 가능도
- Bias vs Variance / 오버 vs 언더피팅 / 표준화와 정규화 (L1, L2)

3주차

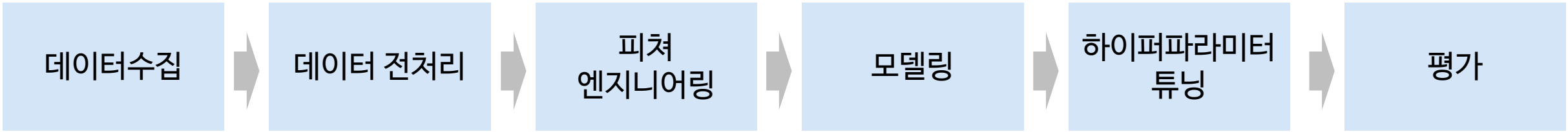
- Decision Tree
- 앙상블

4주차

- 부스팅 (AdaBoost, Gradient Boost, XGBoost, LightGBM)
- 스태킹
- SVM

공동세션 : 웹 크롤링

그동안 배웠던 것을 리뷰해봅시다!



머신러닝 파이프라인으로 생각해보기

머신러닝 파이프라인으로 생각해봅시다

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → 모델링 → 하이퍼파라미터 튜닝 → 평가

- 기업 : DataBase/SQL
- 학생 :

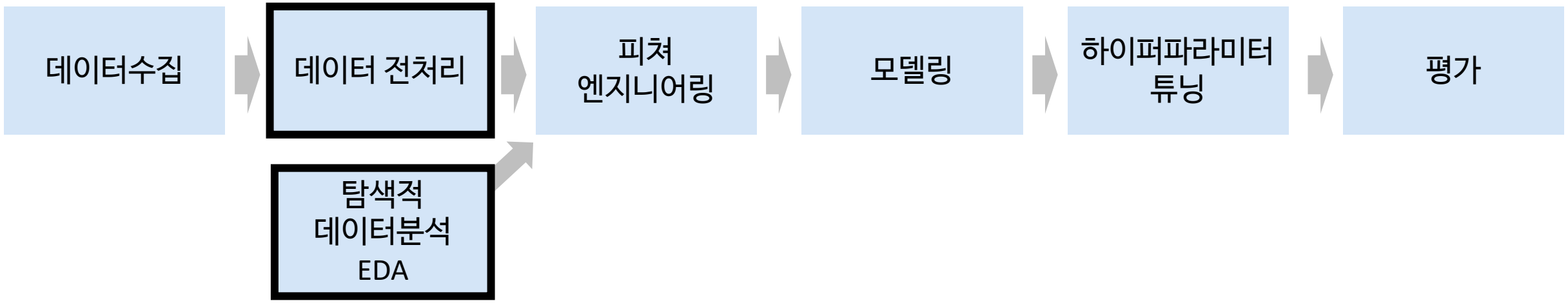
kaggle DACON DATA 공공데이터포털 .GO.KR

- 크롤링

Selenium BeautifulSoup

- 시간이 되면 SQL 공부도 해보세요!

머신러닝 파이프라인으로 생각해봅시다



- 데이터를 씹고 뜯고...맛도?
- 시각화 활용
- 이상치, 결측치
- 데이터 정규화
  - Standardization (평균이0, 표준편차가1)
  - min-max normalization (상대적 크기로 0~1)
- 문제정의, 가설 수립

	표준화(standardization)	정규화(normalization)
공통점	데이터 rescaling	
정의 &목적	데이터가 평균으로부터 얼마나 떨어져있는지 나타내는 값으로, 특정 범위를 벗어난 데이터는 outlier로 간주, 제거	데이터의 상대적 크기에 대한 영향을 줄이기 위해 데이터범위를 0~1로 변환
값의 범위	±1.96(또는 ±2) 데이터만 선택	0~1
공식	$z = \frac{x - \bar{x}}{\sigma}$ (분모가 표준편차)	$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$ (분모가 max값)

머신러닝 파이프라인으로 생각해봅시다

데이터수집 → 데이터 전처리 → **피쳐 엔지니어링** → 모델링 → 하이퍼파라미터 튜닝 → 평가

- 제일 중요한데 제일 귀찮고 어려움
- 공모전의 수상여부가 걸려있음
- Garbage in, Garbage out

- 파생변수 생성 (도메인 지식 활용)
- 피쳐 스케일링
- 피쳐 선택 (다중공선성 고려)

- 정규표현식을 배워두면 유용함
  (링크) 점투파 정규표현식

정규 표현식	의미	축약표현
[0-9]	숫자를 찾음	₩d
[^0-9]	숫자가 아닌 것을 찾음	₩D
[₩t₩n₩r₩f₩v]	문자(텍스트, 특수문자, 숫자)인 것을 찾음	₩s
[^ ₩t₩n₩r₩f₩v]	문자가 아닌 것을 찾음	₩S
[a-z]	소문자를 찾음	
[^a-z]	소문자가 아닌 것을 찾음	
[A-Z]	대문자인 것을 찾음	
[^A-Z]	대문자가 아닌 것을 찾음	
[A-Za-z0-9]	영문자,숫자를 찾음	₩w
[^A-Za-z0-9]	영문자,숫자가 아닌 것을 찾음	₩W

머신러닝 파이프라인으로 생각해봅시다

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

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

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)

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

종속변수의 형태

	회귀	분류
변수 개수	단순선형회귀	로지스틱회귀
	다중선형회귀	
	의사결정나무	의사결정나무
		SVM

앙상블
Voting
Bagging
Boosting
Stacking

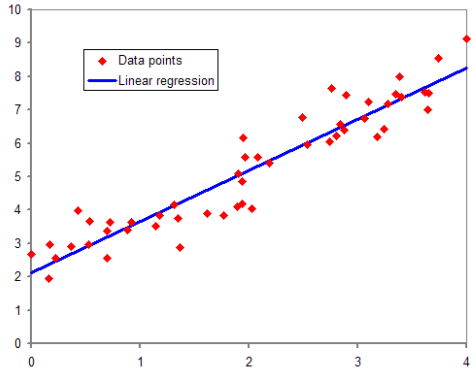
데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

종속변수의 형태

	회귀	분류
변수 개수	단순선형회귀	로지스틱회귀
	다중선형회귀	
	의사결정나무	의사결정나무
		SVM

앙상블
- Voting
- Bagging
- Boosting
- Stacking

Simplie Linear Regression
단순선형회귀



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

target · coefficients · input · random error

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

1. MSE(Mean Squared Error) = $\frac{SSE}{n-2}$
2. RMSE(Mean Squared Error) = $\sqrt{MSE}$
3. R-Squared = $\frac{SSR}{SST}$

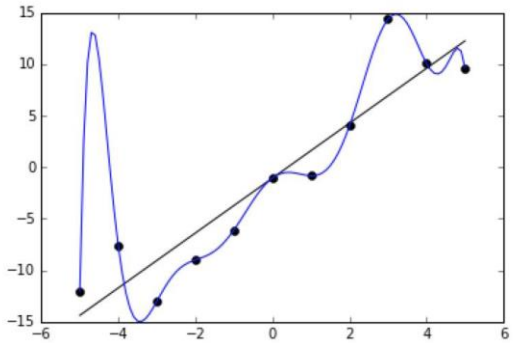
데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

종속변수의 형태

	회귀	분류
변수 개수	단순선형회귀	로지스틱회귀
	다중선형회귀	
	의사결정나무	의사결정나무
		SVM

앙상블
- Voting
- Bagging
- Boosting
- Stacking

Multiple Linear Regression
다중선형회귀, 다항선형회귀



- 독립변수 X가 여러개인 경우
- 가정 : 추정치가 선형관계, 등분산, 자기상관X, 다중공선성X

- 다중공선성에 유의 (ex 월평균음주량,혈중알코올농도→성적)
- VIF를 이용 10이상인 경우 다중공선성 판단

$$VIF_1 = \frac{1}{1-{R_1}^2} \qquad VIF_i > 10 \Leftrightarrow \frac{1}{1-r_i} > 10$$

$$1 > 10 - 10r_i$$

$$r_i > 0.9$$

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

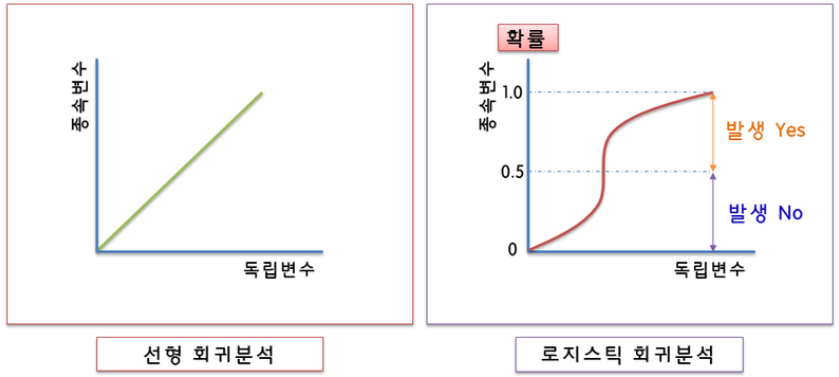
종속변수의 형태

회귀	분류
변수 개수	
단순선형회귀	**로지스틱회귀**
다중선형회귀	
의사결정나무	의사결정나무
	SVM

앙상블

- Voting
- Bagging
- Boosting
- Stacking

Losistic Regression 로지스틱회귀



선형 회귀분석 | 로지스틱 회귀분석

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

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

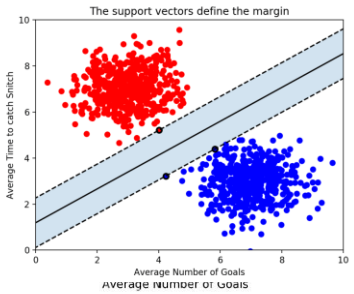
종속변수의 형태

회귀		분류
변수 개수	단순선형회귀	로지스틱회귀
	다중선형회귀	
의사결정나무		의사결정나무
		SVM

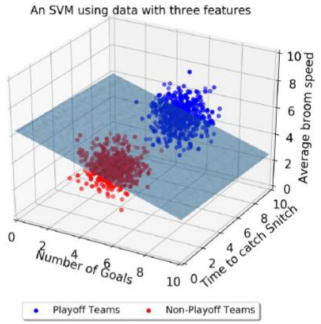
앙상블
Voting
Bagging
Boosting
Stacking

Support Vector Machine

속성이 2개일 때



속성이 3개일 때



- 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

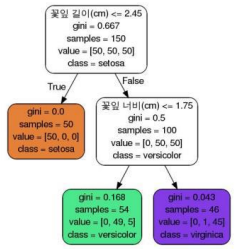
데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

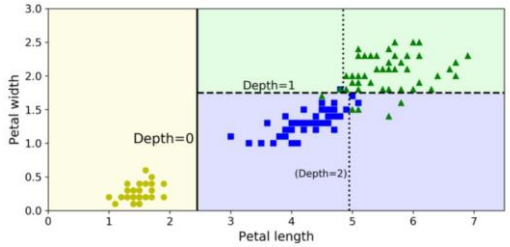
종속변수 Y의 형태

	수치형 - 회귀	명목형 - 분류
변수 개수	단순선형회귀	로지스틱회귀
	다중선형회귀	
	의사결정나무	의사결정나무
		SVM

- 앙상블
  - Voting
  - Bagging
  - Boosting
  - Stacking

Decision Tree 의사결정나무

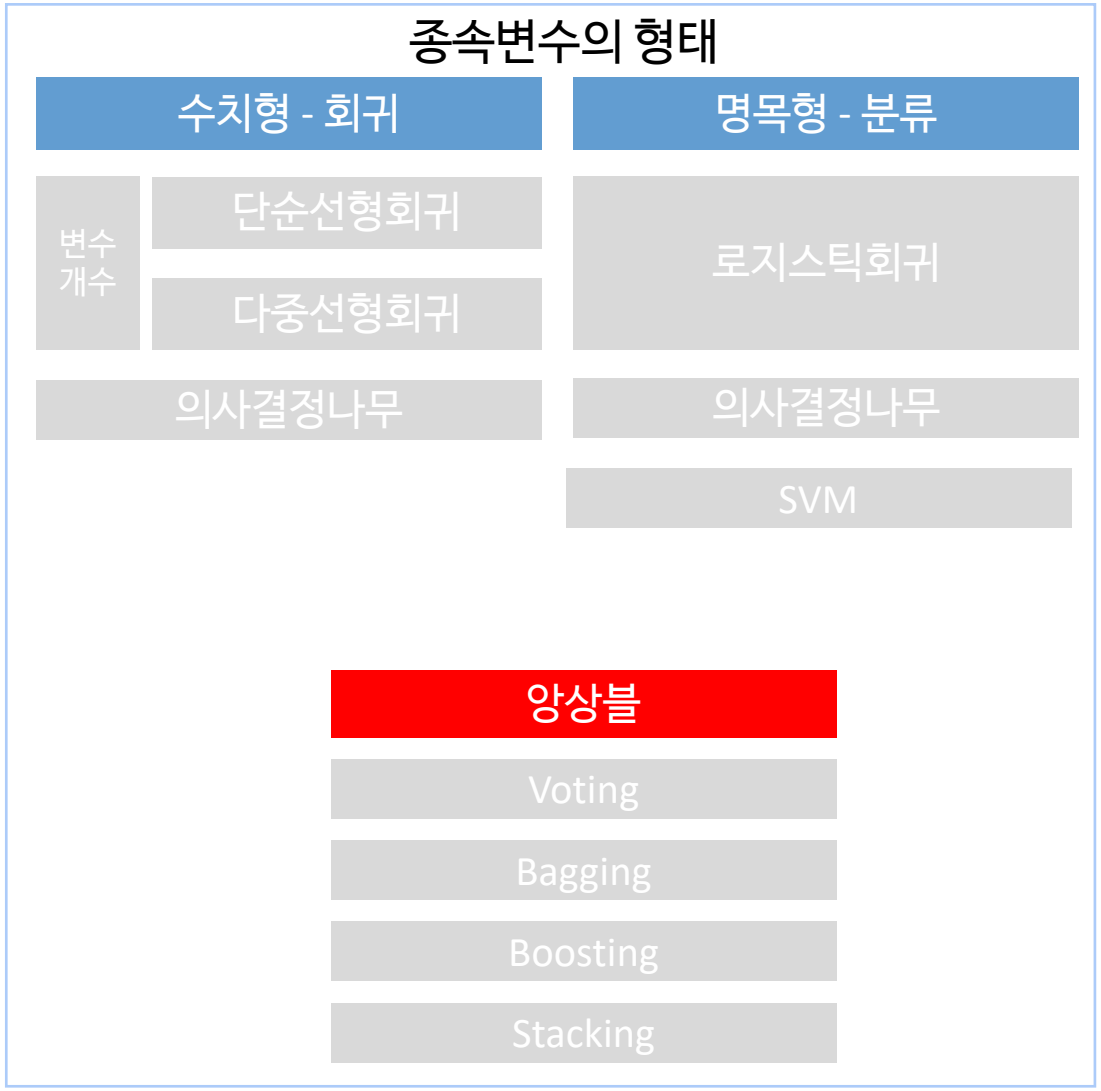




- 분류와 회귀 모두 사용 가능
- 적절한 '분리규칙'과 '정지규칙'으로 예측값을 할당하며 학습
- 장점 : 직관적, 이상치와 노이즈 영향 적음, 모델 해석력 등
- 단점 : 일반화 어려움, **<u>오버피팅 가능성 높음</u>**

- 분리규칙
  1) 지니계수 $G(S) = 1 - \sum_{i}^{c} p_i^2$
  2) 엔트로피 $Entropy(S) = \sum_{i=1}^{c} p_i * I(x_i)$ $IG(S, A) = E(S) - E(S|A)$
- 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)

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가



Ensemble

- 여러 모델을 바탕으로 새로운 모델을 만드는 방식 *(집단지성)*

- **보팅** : 1개의 데이터셋에 여러 모델의 예측결과로 **투표**하는 방식
- **배깅** : 여러 Subset에 같은 모델의 **예측결과를 결합**하는 방식
- **부스팅** : 앞선 모델의 틀린 예측에 가중치를 더하며 여러 모델 학습
- **스태킹** : 여러모델의 학습결과를 메타모델의 학습데이터로 재학습

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

종속변수의 형태

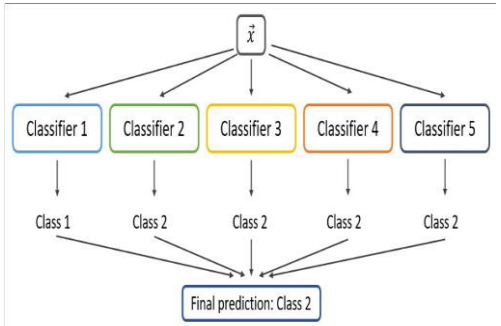
수치형 - 회귀	명목형 - 분류
변수 개수	
단순선형회귀	로지스틱회귀
다중선형회귀	
의사결정나무	의사결정나무
	SVM

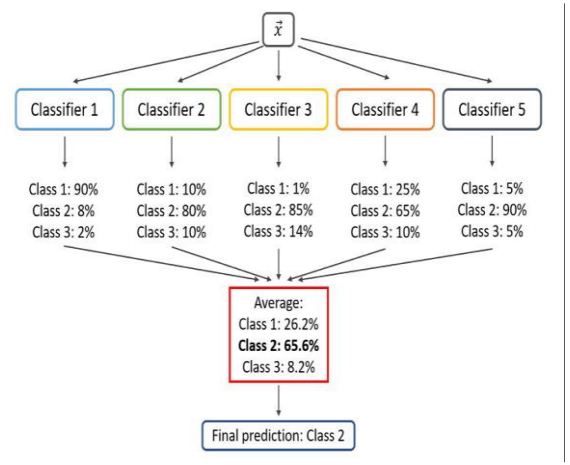
- 앙상블
  - **Voting**
  - Bagging
  - Boosting
  - Stacking

Voting

- **보팅** : 1개의 데이터셋에 여러 모델의 예측결과로 투표하는 방식
- 편향-분산 Trand-off의 효과를 극대화함

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





종속변수의 형태

	수치형 - 회귀	명목형 - 분류
변수 개수	단순선형회귀	로지스틱회귀
	다중선형회귀	
	의사결정나무	의사결정나무
		SVM

- 앙상블
  - Voting
  - **Bagging**
  - Boosting
  - Stacking

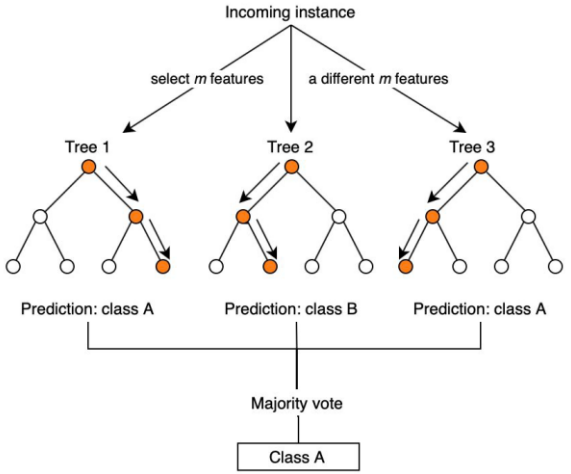
Bagging

- **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

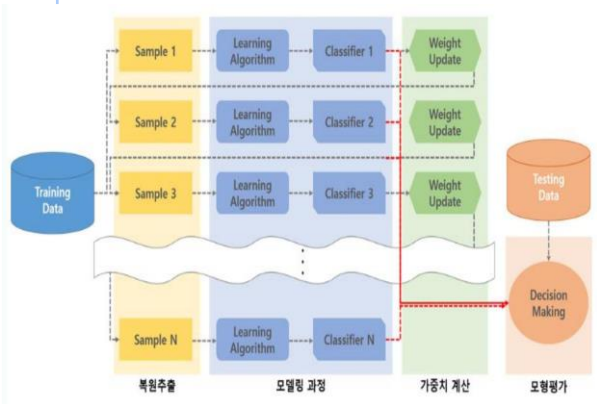
  학습시간이 빠르고, 과적합 방지가 가능하며, 정확도가 좋은편

  → 병렬처리를 지원해서, 일단 Baseline으로 짜기 적절함



종속변수의 형태

수치형 - 회귀	명목형 - 분류
변수 개수	로지스틱회귀
단순선형회귀	
다중선형회귀	
의사결정나무	의사결정나무
	SVM



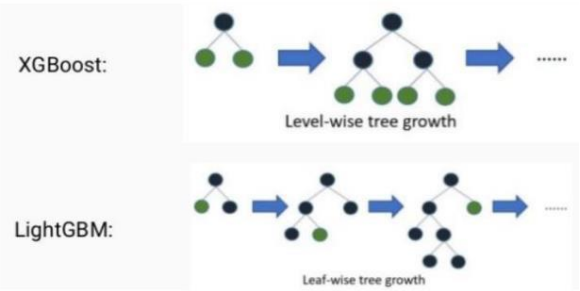
앙상블
Voting
Bagging
Boosting
Stacking

Boosing

- 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

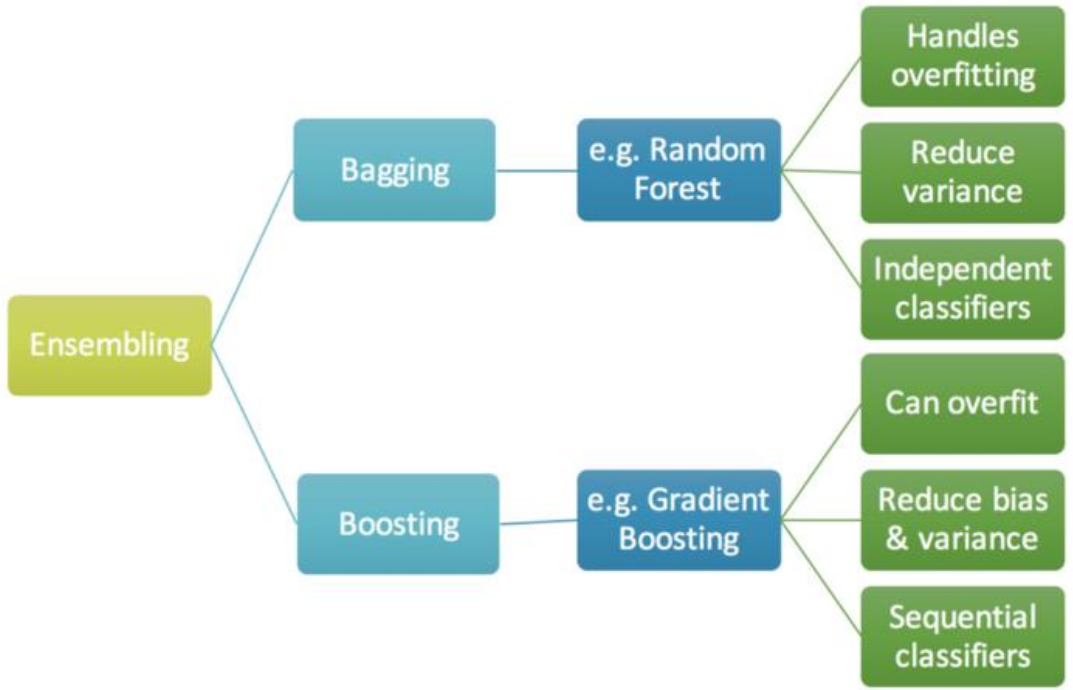


데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

종속변수의 형태

수치형 - 회귀		명목형 - 분류
변수 개수	단순선형회귀	로지스틱회귀
	다중선형회귀	
의사결정나무		의사결정나무
		SVM

앙상블
Voting
Bagging
Boosting
Stacking



종속변수의 형태

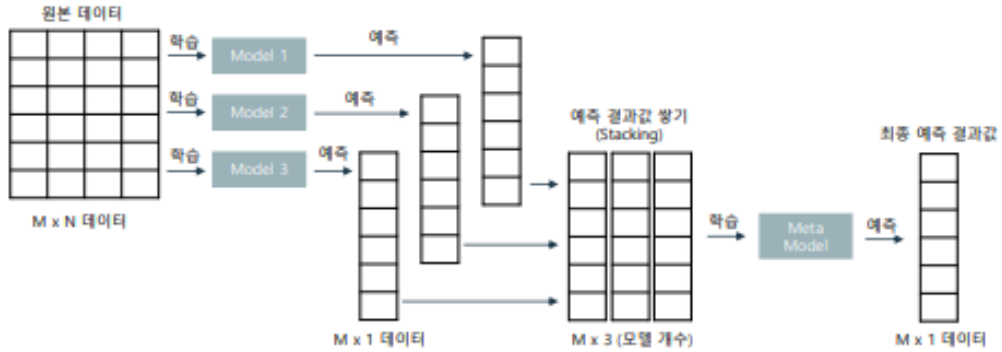
수치형 - 회귀	명목형 - 분류
변수 개수	단순선형회귀
	다중선형회귀
	로지스틱회귀
의사결정나무	의사결정나무
	SVM

앙상블

- Voting
- Bagging
- Boosting
- **Stacking**

Stacking

- 스태킹 : 여러모델의 학습결과를 메타모델의 학습데이터로 재학습
- 현실에서 적용하기보단, 캐글 성능향상 목적으로 유용함 !



- K-Fold CV를 이용하여 데이터셋을 나누면서 과적합 방지가능

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → **모델링** → 하이퍼파라미터 튜닝 → 평가

지도학습 (supervised learning)

수치형 - 회귀		명목형 - 분류
변수 개수	단순선형회귀	로지스틱회귀
	다중선형회귀	
의사결정나무		의사결정나무
		SVM

앙상블

- Voting
- Bagging
- Boosting
- Stacking

비지도학습 (supervised learning)

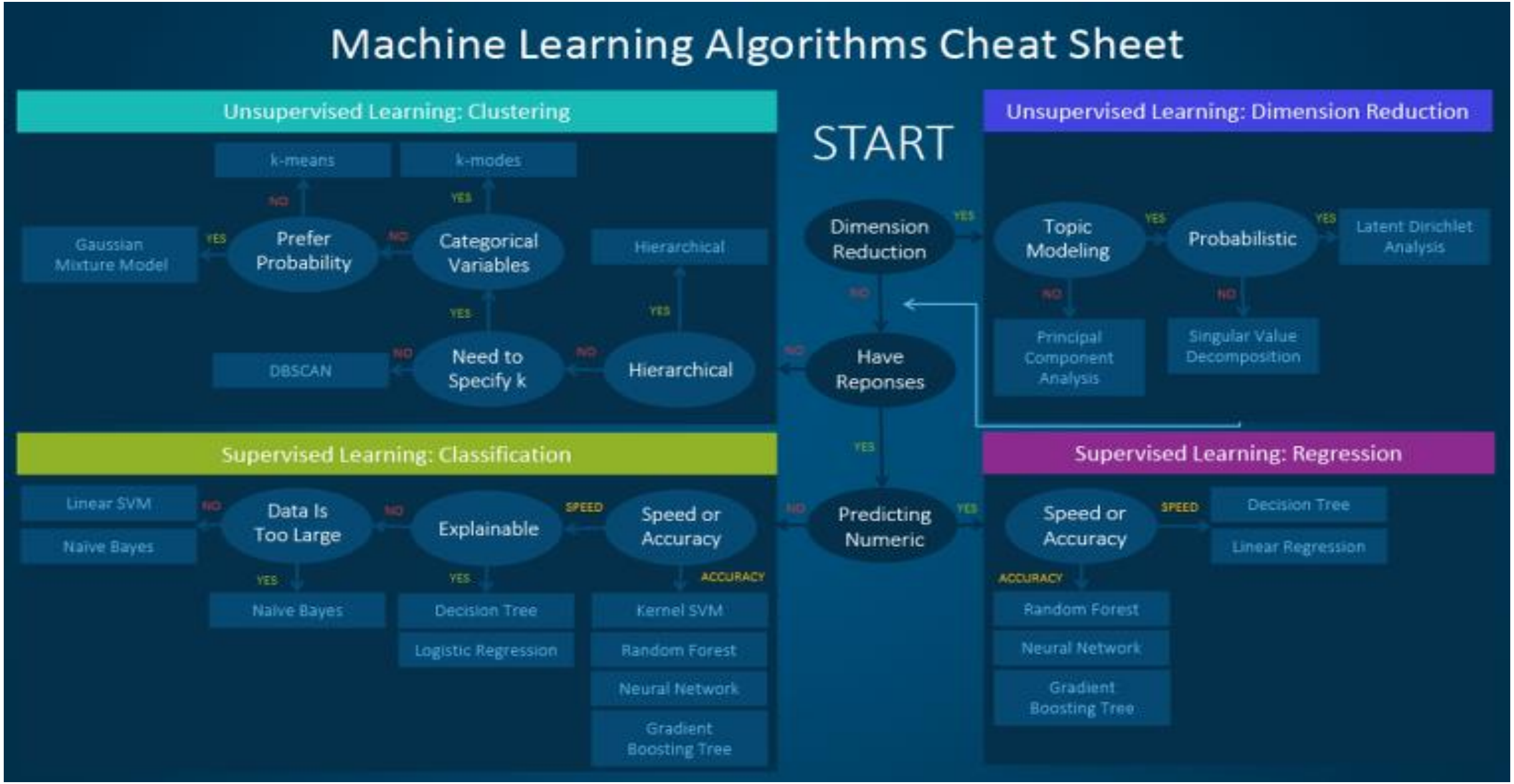
군집	차원축소
K-Means	PCA
DBSCAN	LDA
GMM(Gaussian Mixture Model)	SVD
	NFM

추천시스템

NLP

강화학습

……

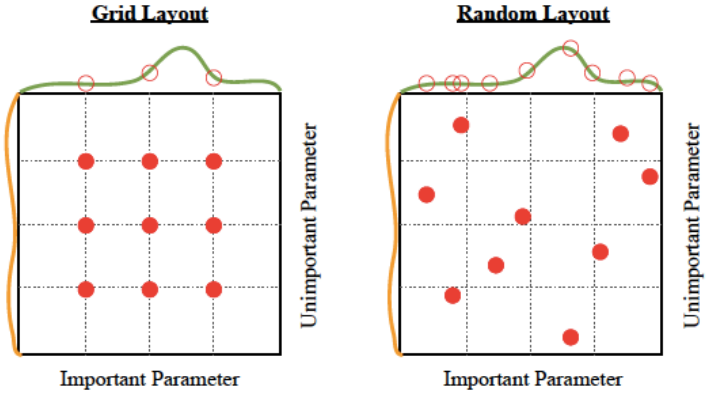


출처 : SAS 공식블로그 (링크)

머신러닝 파이프라인으로 생각해봅시다

데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → 모델링 → **하이퍼파라미터 튜닝** ↻ → 평가

■ 그리드 서치 vs 랜덤서치



- **그리드 서치** : 모든 경우를 테이블로 만든뒤 격자로 탐색
  연산비용 큼. 처음에는 넓은 간격으로 탐색하는 것이 좋음

- **랜덤 서치** : 하이퍼 파라미터 값을 **랜덤**하게 탐색

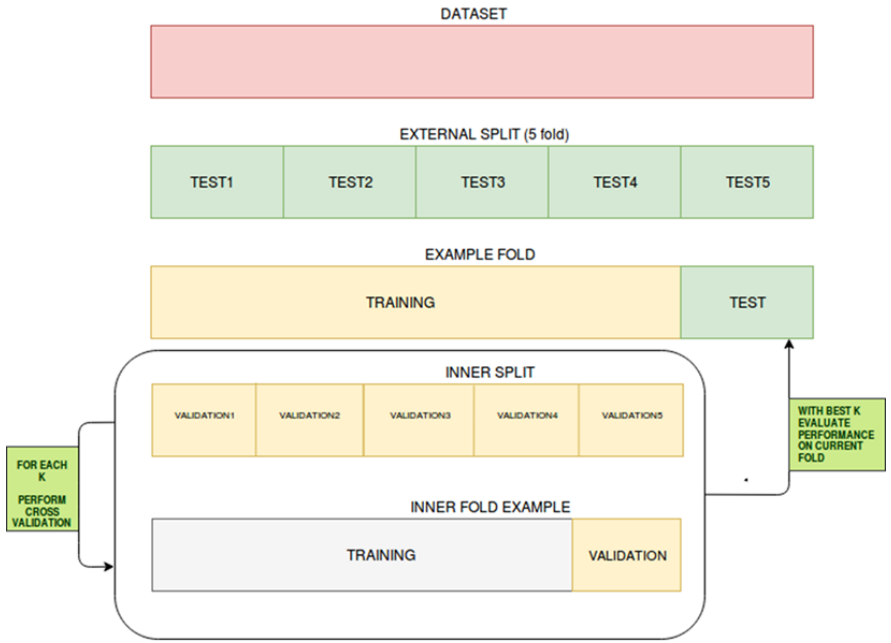
- 랜덤 서치 vs 그리드 서치
- 공모전에선 성능향상을 위한 시간 소요
- Colab GPU를 적절히 활용

머신러닝 파이프라인으로 생각해봅시다

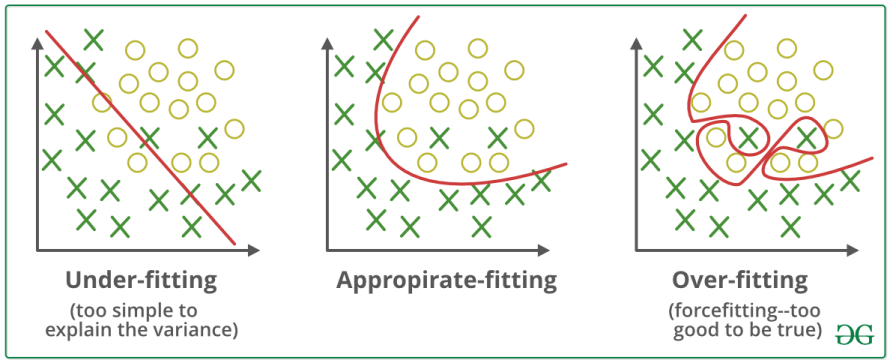
데이터수집 → 데이터 전처리 → 피쳐 엔지니어링 → 모델링 ↻ 하이퍼파라미터 튜닝 ↻ **평가**

- 학습, 검증, 평가 데이터
- 교차검증
- MSE, RMSE, MAE …
- 오버피팅 / 언더피팅 해결

■ 교차검증



■ 오버피팅 vs 언더피팅



[언더피팅 해결]
피쳐 수 늘리기
학습을 더 반복하기

[오버피팅 해결]
피쳐 수 줄이기
데이터의 양 늘리기
교차검증 사용하기
L1, L2 규제 (정규화)

감사합니다 !