🍔 Ensembling methods

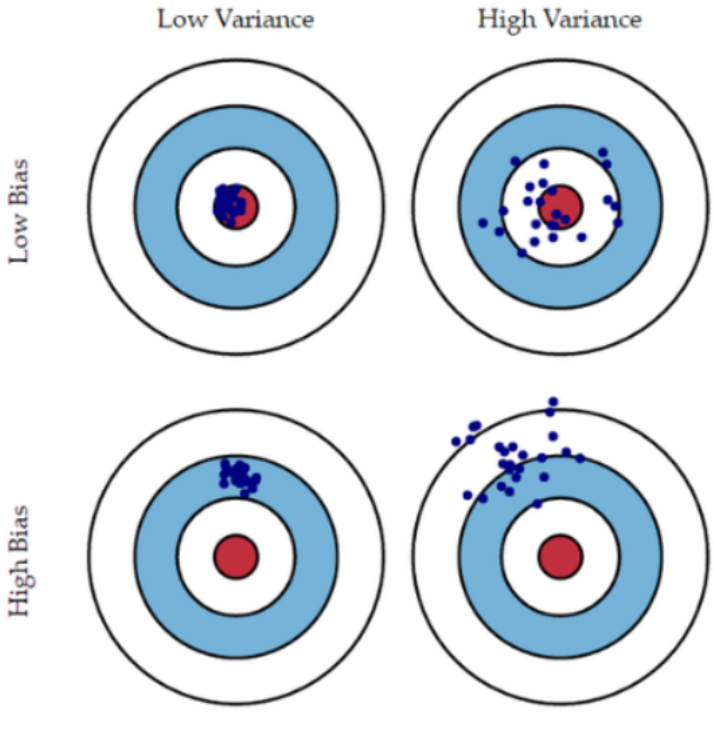
🏝Emsembling

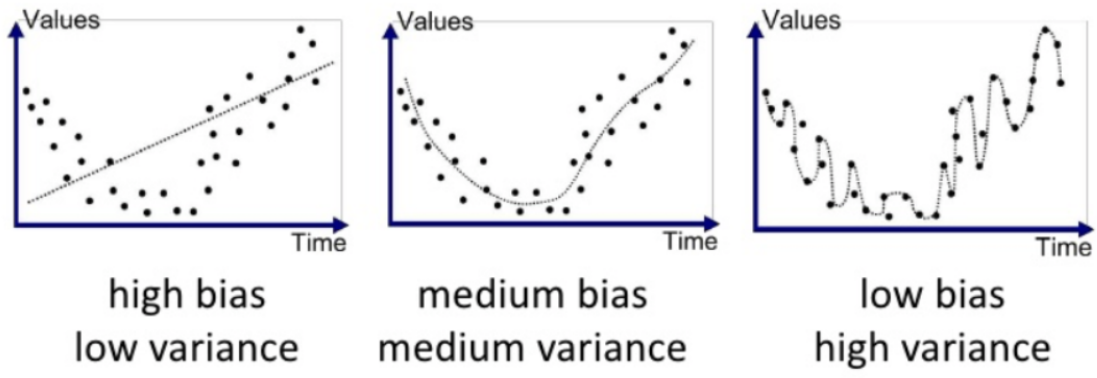
- 여러개의 모델을 결합하여 하나의 모델보다 더 좋은 성능을 내는 머신러닝 기법

🏝Errors : 모델에서 무엇이 오류를 일으키는지.

- 어느 모델에서나 나타나는 오류는 수학적으로 세 부분으로 나눌수 있다.
- $$Err(x) = \left(E[\hat{f}\,(x)] - f(x)\right)^2 + E\left[\hat{f}\,(x) - E[\hat{f}\,(x)]\right]^2 + \sigma_e^2$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

- 

- 

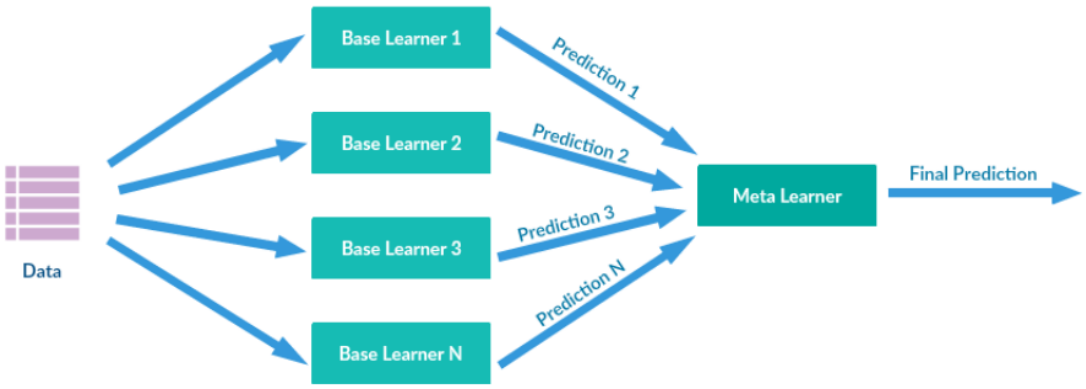
1. **Bias 오류**
    - 높은 바이어스는 중요한 추세를 계속 놓치고 있는 성능이 떨어지는 모델을 의미한다.
    - 데이터 안에 있는 데이터 간의 상관관계를 충분히 풀어내지 못할때 발생.

2. **Variance 오류 (분산)**
    - 높은 분산을 가지고 있으면(예측값들이 떨어져있는 정도가 크다) overfitting .
        - 선이 구불구불하게 복잡해져서 새로운 데이터를 예측하기 힘들다.
    - 트레이닝 데이터에 너무 민감하게 반응하여 발생
3. **Irreducible 오류**
    - X 들로는 완전히 Y에 대해 결정할수 없다는 점에서 기인함.
    - X와는 의존적이지 않지만, Y에 영향을 미치는 요소를 말함

- 예측변수의 그룹을 앙상블이라 하며, 예측 변수 그룹의 예측 집계를 통해 최상의 개별 변수보다 더 나은 예측을 얻는다.
- 정답과 예측값 가의 거리 + 예측값과 예측값 평균간의 거리 + 어떤모델로도 제거 불가능한 오류

🏝Types of ensembling :

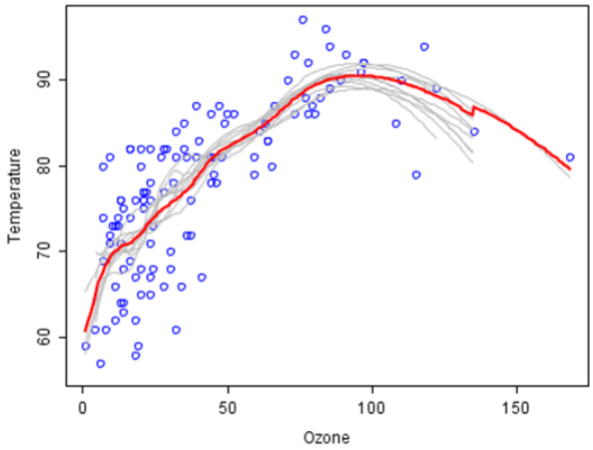
1. **Max Voting**
    - 일반적으로 Classification 문제에서 사용된다.
    - 각 데이터 포인트에 대한 예측을 여러 모델에 대해 진행하고, 투표를 통해 대다수의 모델로 부터 얻는 예측을 최종 예측으로 사용한다.
    - hard voting : 다수결
    - soft voting : 클래스의 확률을 추정할수 있는경우, 클래스의 확률이 가장 높은 클래스를 예측하도록 한다.
2. **Averaging**
    - 회귀에서 사용된다.
    - Max Voting 방법과 비슷하게 각 데이터 포인트에 대해 평균화 한다.
3. **Weighted Average**
    - 각 모델의 중요성의 정의하는 다른 가중치에 따라 가중치를 반영한 평균
4. **Stacking**

- 

- 머신러닝 알고리즘으로 **훈련 데이터 셋을 통해 새로운 데이터 셋을 만들고 이를 데이터셋으로 사용하여 다시 머신러닝 알고리즘을 돌리는것**. (서로 다른 타입의 모델을 결합한다.)
    - 다시 정리해서 말하면 여러 개의 개별 모델들이 생성한 예측 데이터를 기반으로 최종 메타 모델이 학습할 별도의 학습 데이터 세트와 예측할 테스트 데이터 세트를 재 생성하는 기법입니다.

5. **Blending**
    - 스태킹과 매우 유사한 방법.

- 차이점이 있다면 Stacking에서는 cross-fold-validation을 사용하고, Blending은 holdout validation을 사용합니다.

6. **Bagging (Bootstrap Aggregating)**
   - 샘플을 여러 번 뽑아 각 모델을 학습시켜 결과를 집계(Aggregating) 하는 방법이다.
   - 평균을 통해 **분산 값을 줄여** 모델을 더 일반화 시킨다.
   - 분산을 줄이고 과적합을 피하도록 해준다.
   - 

   - 전체 데이터에서 일부 데이터만 가지고(Bootstrapping) 곡선을 추정하고, 얻은 여러개의 곡선의 평균을 취해 빨간 곡선을 얻게됨.
   - 표본 데이터가 작은 경우에는 사용하면 안된다.
   - 데이터에 잡음이 많거나(특이점들이 추정을 왜곡시킬가능성이 존재), 데이터에 의존성이 있는경우
   - 모델들의 알고리즘은 모두 같다.
   - 특정 feature가 정답에 많은 영향을 줄때, 모든 tree들이 비슷한 결과를 도출하게 되는 문제가 생긴다. > random forest 를 통해서 모든 모델을 서로 다른 feature로 학습하게 할수 있다.
7. **Boosting**
   - 여러개의 약한 학습기를 순차적으로 학습-예측 하며 잘못 예측한 데이터에 가중치 부여를 통해 오류를 개선해 나가면서 학습하는 방식
   - 부스팅 : 약한 학습자를 강한 학습자로 전환하는 알고리즘을 말한다.

🏝Algorithms - bagging, boosting

Bagging Algorithms

1. **Bagging meta-estimator**
   - bagging 방법을 사용하는것을 말함.
2. **Random Forest**
   - Bagging meta-estimator과 다르게 결정트리(decision tree)만 사용하고, 특성(feature)을 랜덤으로 선택하여 Bagging을 진행한다는 점이 다릅니다.
   - 앙상블 알고리즘중 비교적 빠른 속도
   - 하이퍼파라미터 튜닝시간이 많이 소모된다.

Boosting Algorithms

1. **AdaBoost**
   - 가장 간단한 부스팅 알고리즘 중에 하나다.
   - 의사결정 트리 모델링을 위해 사용된다.
   - 잘못 분류한 데이터에 가중치를 부여하여, 다음 분류기는 이를 더 잘 분류하게 만드는것이다.

- 잡음이 많은 데이터와 이상점에 취약한 모습을 보인다.
- 한 번 학습 후, error(ε) 계산, 모델 별 가중치(α) 계산, 데이터 가중치(D)를 갱신
- error값으로 가중치들을 갱신합니다 .

2. **GBM**
    - 가중치 업데이트를 Grandient Descent 로 한다. (학습 전단계 모델에서의 잔여오차로 한다)
3. **XGB (eXtra Gradient Boost)**
    - binary classification, regression, multiclass classification 을 지원한다.
4. **Light GBM(Light한 GBM)**
5. **CatBoost**
    - 범주형 변수를 위해 만든 Boosting 알고리즘.