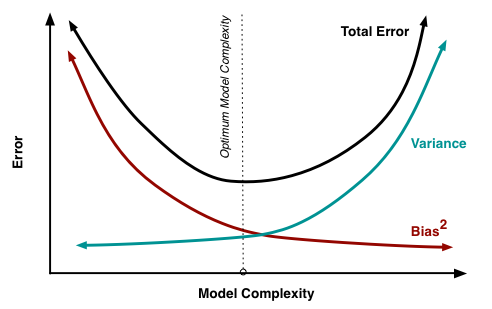
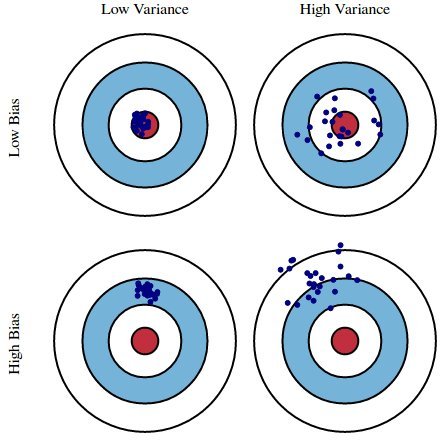
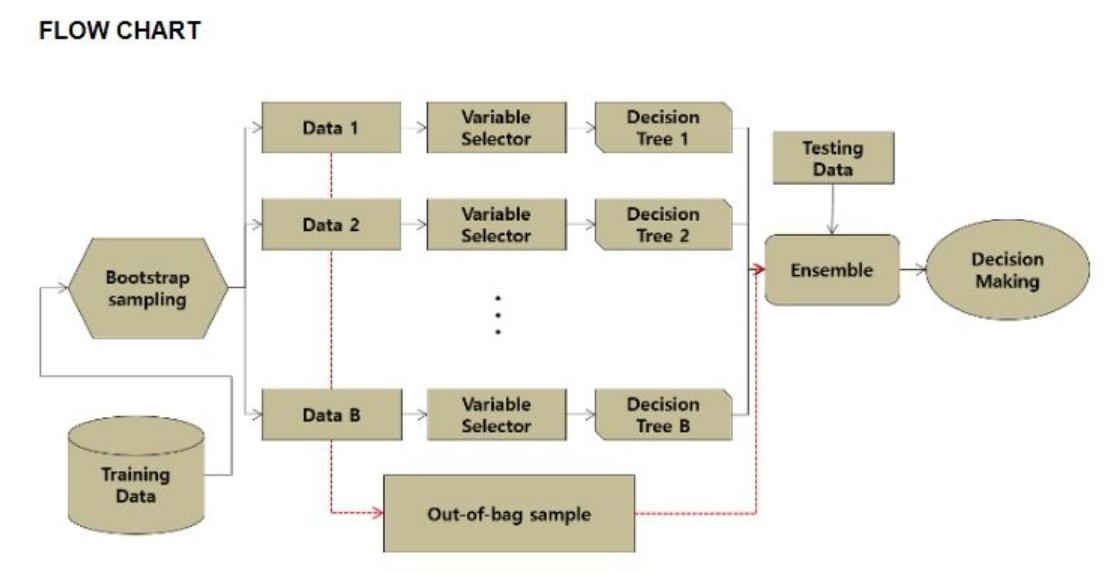
Random forest

Random forest는 bagging approach 방식을 사용하는 대표적인 머신 러닝 알고리즘이다. Bagging(=Bootstrap Aggregation)이란 주어진 데이터에서 랜덤하게 부분 집합을 샘플링해 N개의 예측모형을 만들어 개별 예측모형에 voting하는 방식이다. Bagging의 장점은 bias-variance trade off를 극복할 수 있다는 점이다. 머신 러닝에서 bias와 variance는 learning error의 주범이지만 서로 zero sum 양상을 띄기 때문에 이 두 값을 낮게 유지해주는 것이 중요하다. (bias가 높으면 예측결과가 실제 결과와 비교해 부정확해지고 variance가 높으면 예측 결과가 어떤 dataset에서는 잘 맞고 어떤 dataset애서는 크게 맞지 안는 등 예측결과의 안전성이 떨어진다.)



Random forest는 불안전성과 같은 decision tree의 단점을 개선하기 위한 알고리즘으로 다수의 decision tree를 결합하여 하나의 모델을 생성하는 방식이다. 간단하게 알고리즘을 설명하자면 먼저 N개의 bootstrap sample를 생성하고 임의의 bootstrap sample로 N개의 트리를 생성한다. 각각의 앙상블로부터 train classifier를 생성하고 예측 결과를 투표방식으로 선택한다. (Bootstrap은 주어진 train data에서 중복을 허용하여 원해 데이터와 같은 크기의 데이터를 만드는 과정이다. / data를 분류하는 알고리즘 자체를 classifier라고 하며 위의 경우인 decision tree에서는 특정 질문에 대한 응답을 따라가는 방식으로 data를 분류한다.)



Random forest는 bagging approach 방식으로써 bias-variance trade off를 극복한다고 했는데 이를 살펴 보자면 개별 decision tree의 평균값을 이용해 낮은 bias를 유지하고 표본의 크기가 충분하면 표본평균의 분포가 정규 분포를 이룬다는 중심 극한 정리를 이용해 개별 decision tree의 평균값을 사용하는 것으로 variance를 낮게 유지한다. 즉 예측력은 높지만 안전성이 떨어지는 decision tree 여러 개를 평균값으로 사용해 예측력과 안전성을 보장한다.

지금까지 random forest에 대해 설명했지만 사실 random forest는 black box와 같이 결과가 나오는 과정을 사림이 하나하나 이해할 수는 없다. 하지만 프로그램을 하는데 있어서는 3가지의 hyperparameter만 정의해주면 되기 때문에 사용방법 자체는 용이한 편이다. 먼저 사용할 decision tree의 수와 각각의 decision tree에 사용할 feature의 수를 정해 주어야한다. 마지막으로 전체 data set크기의 몇%로 샘플링을 할지도 정해야 하는데 보통 80%를 사용한다고 한다.

지금까지 random forest를 사용할 때 얻을 수 있는 장점인 예측력, 안전성, 용이성 등을 살펴 봤지만 random forest 방식이 black box와 같다고 한 것처럼 기존 decision tree의 장점인 설명력을 잃는 등의 단점 또한 가지고 있다. 특히 random forest는 data set이 큰 경우 변수가 많은 경우에 권장되지 않는 방법인데 우리가 참여한 WSDM에서는 기본적으로 20여개의 feature 가 제공되었으며 앞에서 살펴본 모델들에 있어서 feature의 수가 많을수록 높은 예측성공률을 보여준 것과 달리 비교적 낮은 결과값을 보여줬다.