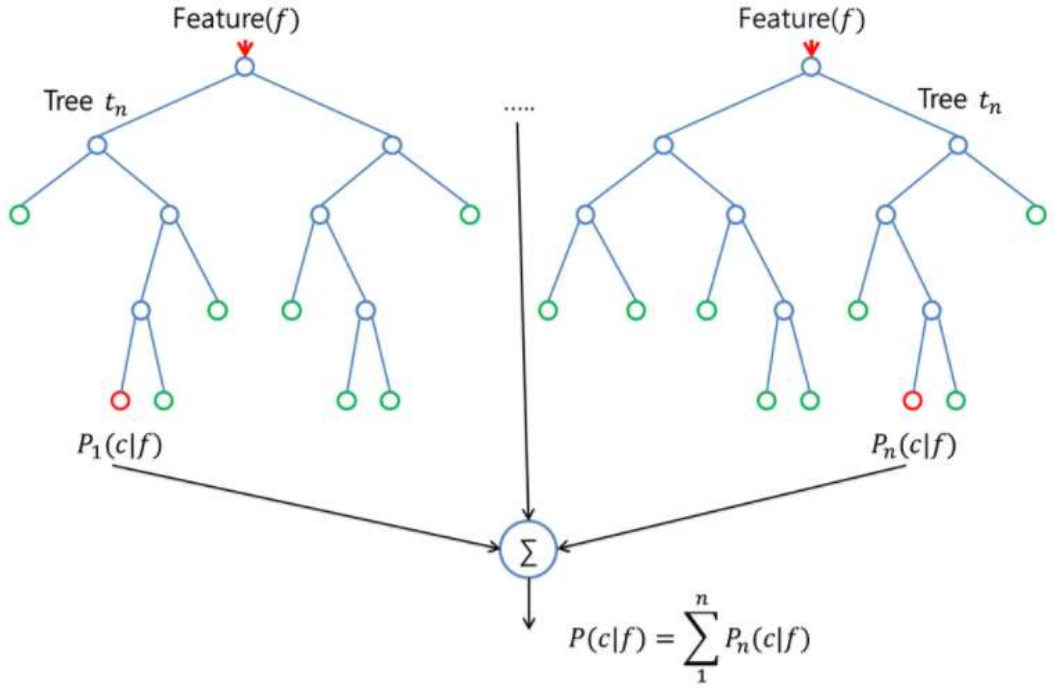
랜덤 포레스트(Random Forest)

데이크루 1기 jihyeheo

1. 랜덤 포레스트(Random Forest) 개념

- 랜덤 포레스트(Random Forest)는 의사결정나무 모델 여러 개를 훈련시켜서 그 결과를 종합해 예측하는 앙상블 알고리즘입니다.
- 각 의사결정나무 모델을 훈련시킬 때 배깅(Bagging) 방식을 사용합니다.
- 배깅(Bagging)은 전체 train dataset에서 중복을 허용하여 샘플링한 dataset으로 개별 의사결정나무 모델을 훈련하는 방식입니다. 그 후 예측한 값은 평균을 취해 최종적인 예측값을 산출합니다. 이 방식은 예측 모델의 일반화 성능을 향상하는데 도움이 됩니다.
- 랜덤포레스트는 분류와 회귀 모두에 사용되는 알고리즘이다.



2. 랜덤 포레스트(Random Forest) 장단점

장점	단점
일반화 성능 우수합니다.	개별 트리 분석이 어렵고 트리 분리가 복잡해 지는 경향이 존재합니다.
파라미터 조정 용이합니다.	차원이 크고 희소한 데이터는 성능이 미흡합니다.
데이터 scale 변환 불필요합니다.	훈련시 메모리 소모가 큽니다.
Overfitting이 잘 되지 않습니다.	Train data를 추가해도 모델 성능 개선이 어렵습니다.

3. scikit-learn 랜덤 포레스트(Random Forest) 파라미터

class sklearn.ensemble.RandomForestClassifier(*n_estimators=100*, ***, *criterion='gini'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features='auto'*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *bootstrap=True*, *oob_score=False*, *n_jobs=None*, *random_state=None*, *verbose=0*, *warm_start=False*, *class_weight=None*, *ccp_alpha=0.0*, *max_samples=None*)

- **n_estimators** : 랜덤 포레스트 안의 결정 트리 개수

1) n_estimators는 클수록 좋은 성능을 기대할 수 있지만 계속 증가시킨다고 무조건 성능이 향상되지는 않습니다.
2) 개수에 비례하여 학습 수행 시간이 증가합니다.

- **max_features** : 무작위로 선택할 feature의 개수

1) max_features가 30이면 30개의 feature를 모두 선택해서 결정트리를 만든다는 의미입니다.
2) max_features값이 크다면 랜덤 포레스트의 트리들이 매우 비슷해지고 가장 두드러진 특성에 맞게 예측을 할 것이고, 작다면 오버피팅이 줄어들 것입니다.

- **max_depth** : 트리의 깊이를 뜻합니다.
- **min_samples_leaf** : 리프 노드(마지막 노드)가 되기 위한 최소한의 샘플 데이터 수입니다.
- **min_samples_split** : 노드를 분할하기 위한 최소한의 데이터 수입니다.
- **max_leaf_nodes** : 리프 노드의 최개 개수

Reference

https://heytech.tistory.com/149

https://hleecaster.com/ml-random-forest-concept/

https://wooono.tistory.com/115

https://kimdingko-world.tistory.com/180