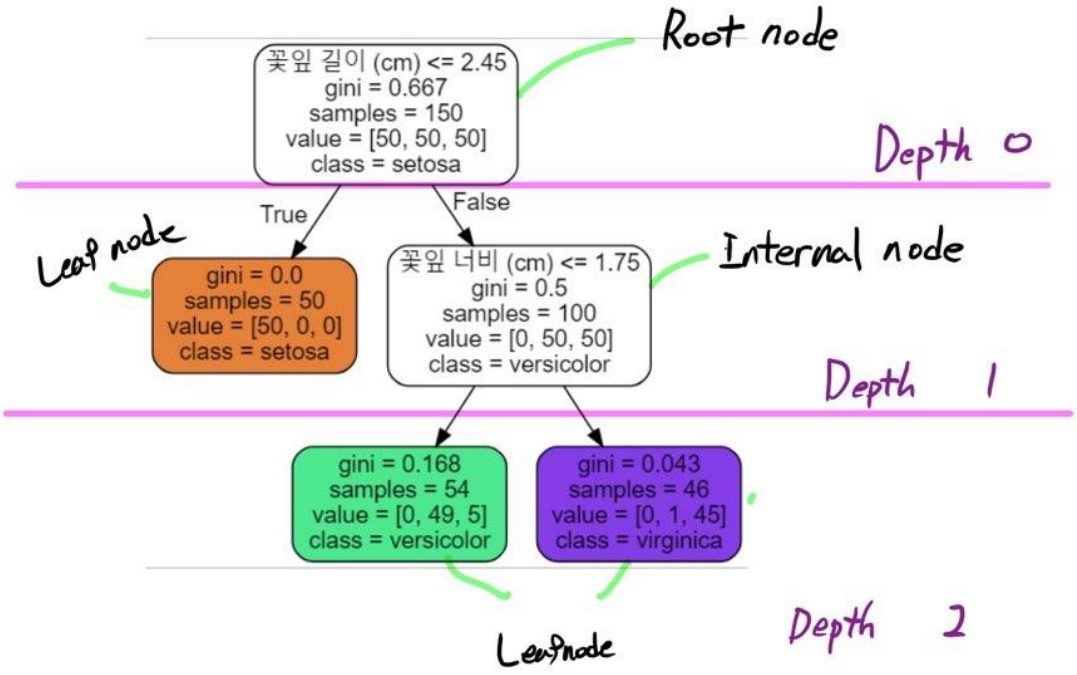
6 장 결정트리

맨 앞에 설명 부분이 약한거 같아서 대충 적었어요.

결정트리는 매우 직관적인 구조를 가지고 있어 시각화와 설명이 간단할 뿐만 아니라, 후에 앙상블 기법들과 함께 학습의 결과가 좋은 분류 모델이다. Decision tree 는 시각화 모습이 나무와 비슷하며, 이에 대한 이해를 하는 것이 모델의 하이퍼파라미터를 튜닝하는데 유용하다.

6.1 그림밑에



밑에 시각화 트리를 보면, 기준에 따라 구분하는 것을 파악할 수 있다. 맨 위의 기준을 root node 라 하며, 맨위에서 몇번이나 내려가냐에 따라 층 depth 라고 할 수 있으며, 중간에 있는 노드들을 internal node 즉 내부노드라고 한다. 분류된 기준에 따라, 나온 맨 밑의 네모들을 leaf node 라고 한다. 즉 가장 크게 분류하는 root node 부터 나무가 밑으로 뻗쳐나가듯 기준들을 바탕으로 분류하는 것으로 볼 수 있다.

6.7

무수히 많은 depth 와 leaf node 를 만든다면 실질적으로 모든 데이터를 맞추는 학습이 가능하지만, 이는 overfitting 일 가능성이 다분하다. 이에 따라, 모델이 너무 정교해지지 않으면서 데이터의 특징들을 충분히 학습할 수 있는 기준이 필요하며, 이러한 하이퍼파라미터를 튜닝하는 능력이 필요하다. 추가로 이는 나중에 randomforest, adaboost 와 같은 개선된 모델에서도 동일하므로 파악하는 것이 중요하다.

Decision tree parameter 는 매우 다양하지만, 앞서 말한 내용을 이해하면 이해할 수 있다.

criterion 같은 경우 비용함수로 기본적으로 gini 불순도를 적용하고 있다.

Min_samples_split 의 경우 default = 2 로 각 노드들은 최소 2 개 이상의 데이터를 가지고 있어야 split 을 고려한다.

Min_samples_leaf 의 경우 default = 1 로 leaf node 가 1 개 이상의 데이터를 가져야 한다.

Max_depth 는 default = None 값으로 pure node 나 min_samples_split 에 다다를 때까지, 즉 모든 학습이 완전히 될때까지 무수히 늘어나므로, 오버피팅이 발생 시 적절하게 줄여줄 필요가 있다.

Max_features 의 경우 default = None 으로 몇 개의 특징을 사용할지 결정하는 것으로 특징의 수가 너무 많다면 중요한 특징만 파악하도록 조절할 필요가 있다.

이 밖에도 splitter class_weight 등 많은 하이퍼파라미터가 존재한다.