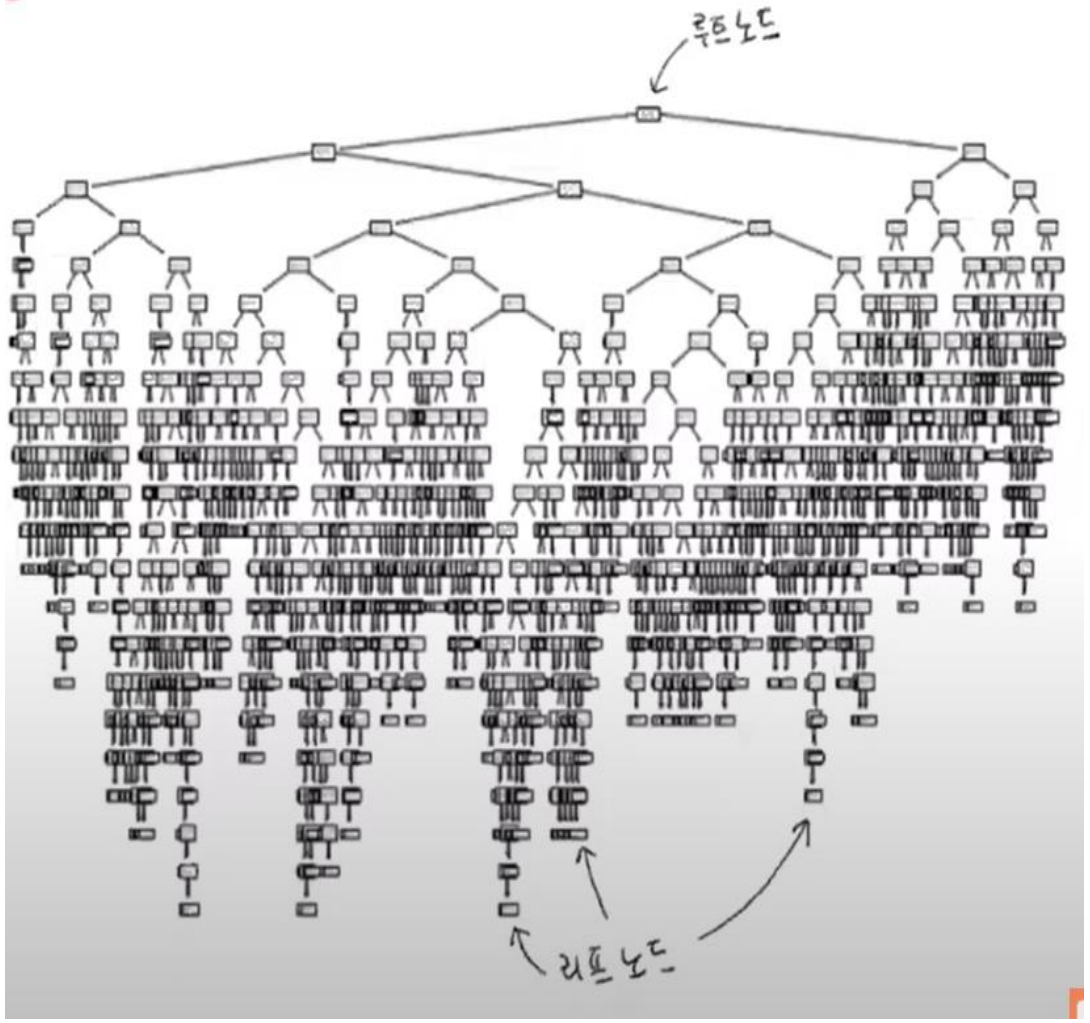
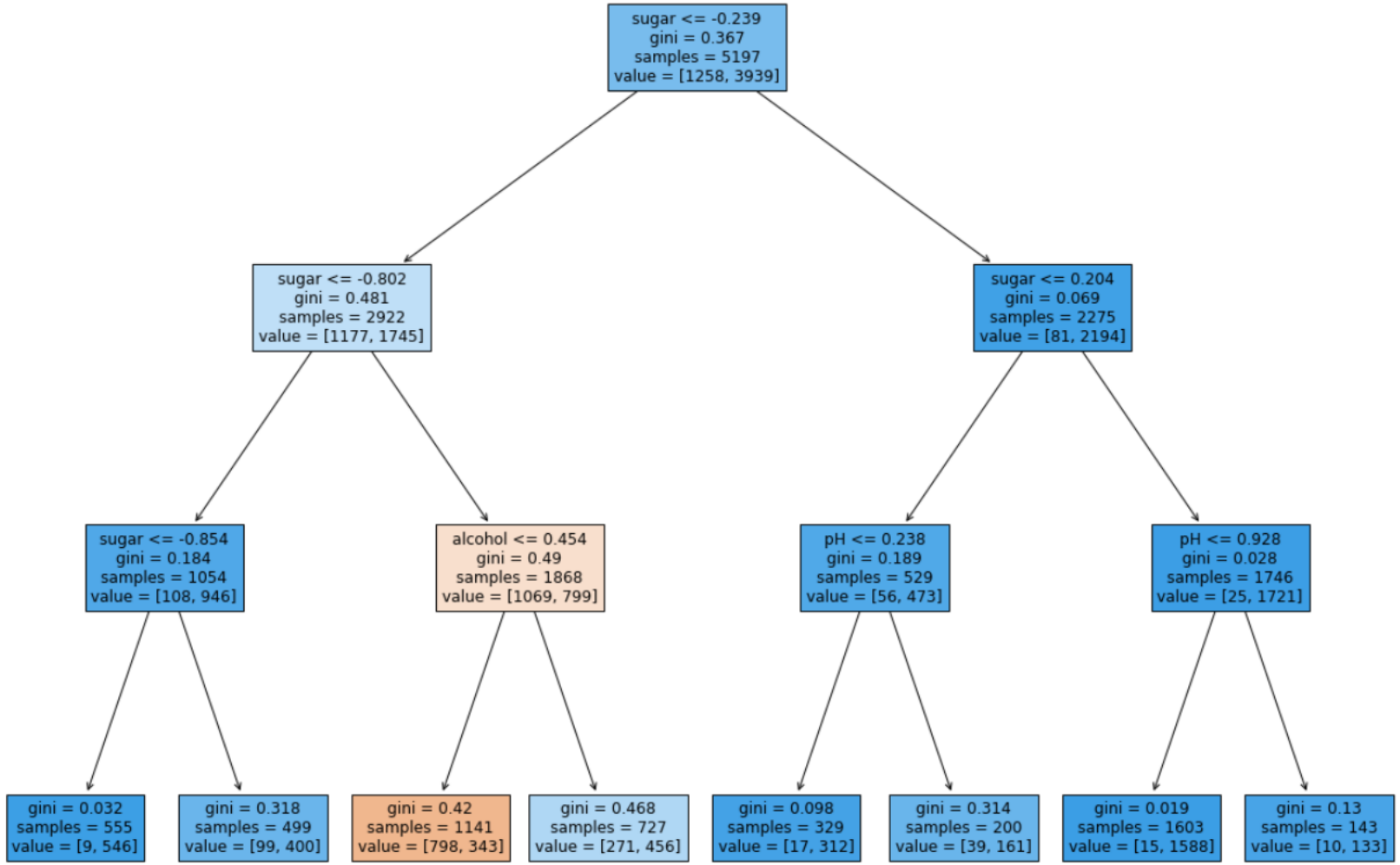
결정 트리

- 스무고개 같은 질문을 통해서 훈련 데이터의 예측값을 만들어나가는 모델
- 노드 : 훈련 데이터의 특성에 대한 테스트를 표현한 것
  - 루트노드 : 가장 위 노드
  - 리프노드 : 가장 아래 노드
  - Filled=True 노드의 색 부여
    (양성 클래스 : 파랑 음성 :빨강)
- 가지 : 테스트의 결과 (True / False)를 나타내며 하나의 노드는 2개의 가지를 갖는다.

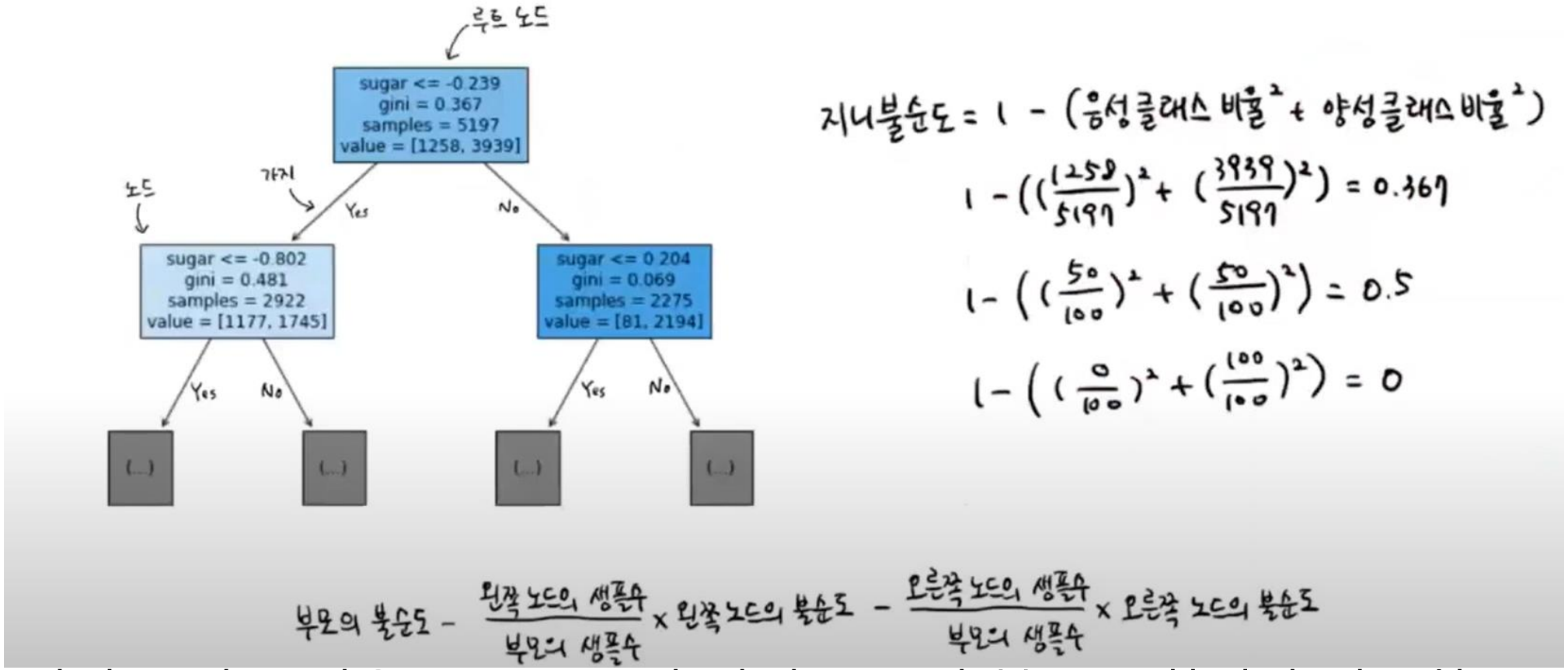


결정 트리를 왜 사용할까?

- **로지스틱 회귀를 사용시에는 가중치나 계수의 의미를 표현하기 힘들다**
- **특성공학의 경우 가중치를 설명하는 것이 더욱 어려워짐**
- **선형함수를 학습하는 것이 아니므로 전처리하지 않은 데이터를 사용할 수 있다.(회귀의 경우 규제를 위해 스케일 조정이 필요)**
- **그래프로 표현하기 좋고 분류 과정을 쉽게 확인할 수 있다.(특성의 중요도를 확인하기 쉽다)**


sugar <= -0.239
gini = 0.367
samples = 5197
value = [1258, 3939]
sugar <= -0.802
gini = 0.481
samples = 2922
value = [1177, 1745]
sugar <= 0.204
gini = 0.069
samples = 2275
value = [81, 2194]
sugar <= -0.854
gini = 0.184
samples = 1054
value = [108, 946]
alcohol <= 0.454
gini = 0.49
samples = 1868
value = [1069, 799]
pH <= 0.238
gini = 0.189
samples = 529
value = [56, 473]
pH <= 0.928
gini = 0.028
samples = 1746
value = [25, 1721]
gini = 0.032
samples = 555
value = [9, 546]
gini = 0.318
samples = 499
value = [99, 400]
gini = 0.42
samples = 1141
value = [798, 343]
gini = 0.468
samples = 727
value = [271, 456]
gini = 0.098
samples = 329
value = [17, 312]
gini = 0.314
samples = 200
value = [39, 161]
gini = 0.019
samples = 1603
value = [15, 1588]
gini = 0.13
samples = 143
value = [10, 133]

어떤 기준으로 노드를 분할하는가?

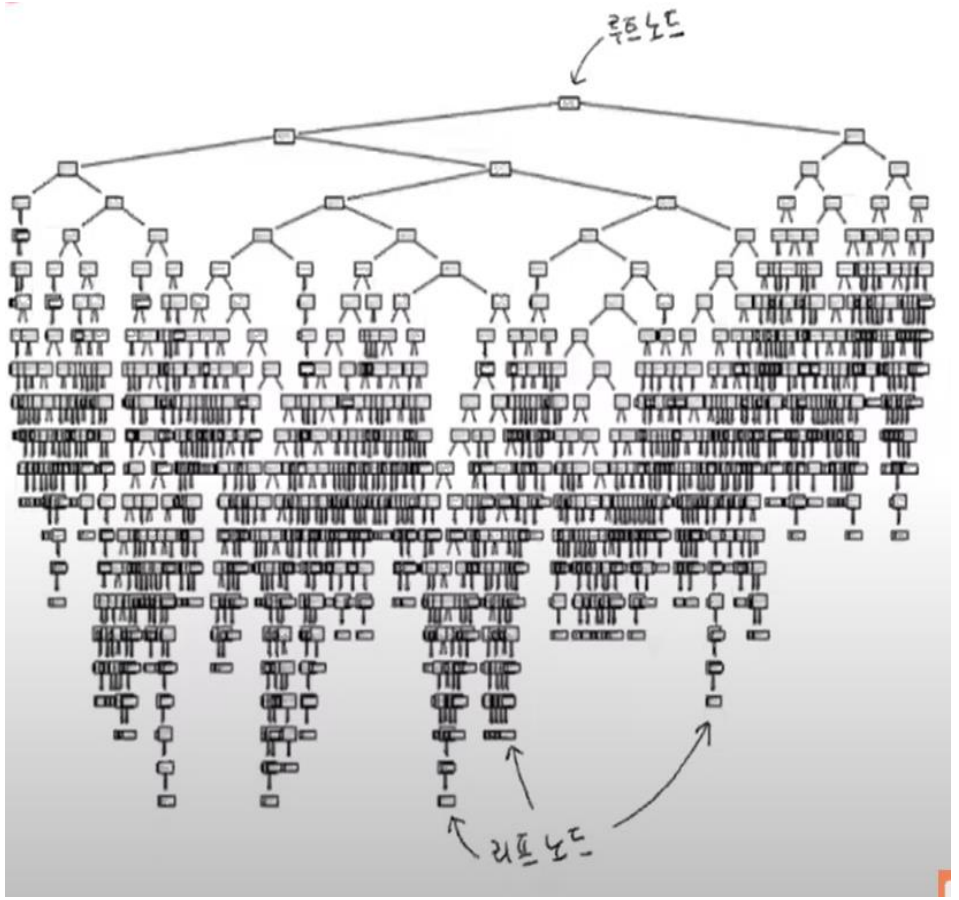


결정 트리 모델은 부모 노드와 자식 노드의 불순도 차이가 가능한 크도록 트리를 성장시킨다.

부모의 지니불순도와 자식의 지니불순도의 차이를 뺐을 때 그 차이가 가장 크게 나오는 방향으로 노드를 진행한다의 의미?

- 리프 노드가 순수 노드가 될 때 까지 분할한 것

- 정확도를 확인해보면 과대적합되어있음을 알 수 있다.




```
print(dt.score(train_scaled, train_target))
0.996921300750433
print(dt.score(test_scaled, test_target))
0.8592307692307692
```

- max_depth를 통해서 노드의 개수를 제한할 수 있다.
- 너무 적게하면 과소적합이 일어난다.

```
dt = DecisionTreeClassifier(max_depth=1, random_state=42)
dt.fit(train_input, train_target)
print(dt.score(train_scaled, train_target))
print(dt.score(test_scaled, test_target))


plt.figure(figsize=(20,15))
plot_tree(dt, filled=True, feature_names=['alcohol', 'sugar', 'pH'])
plt.show()

print(dt.feature_importances_)
```

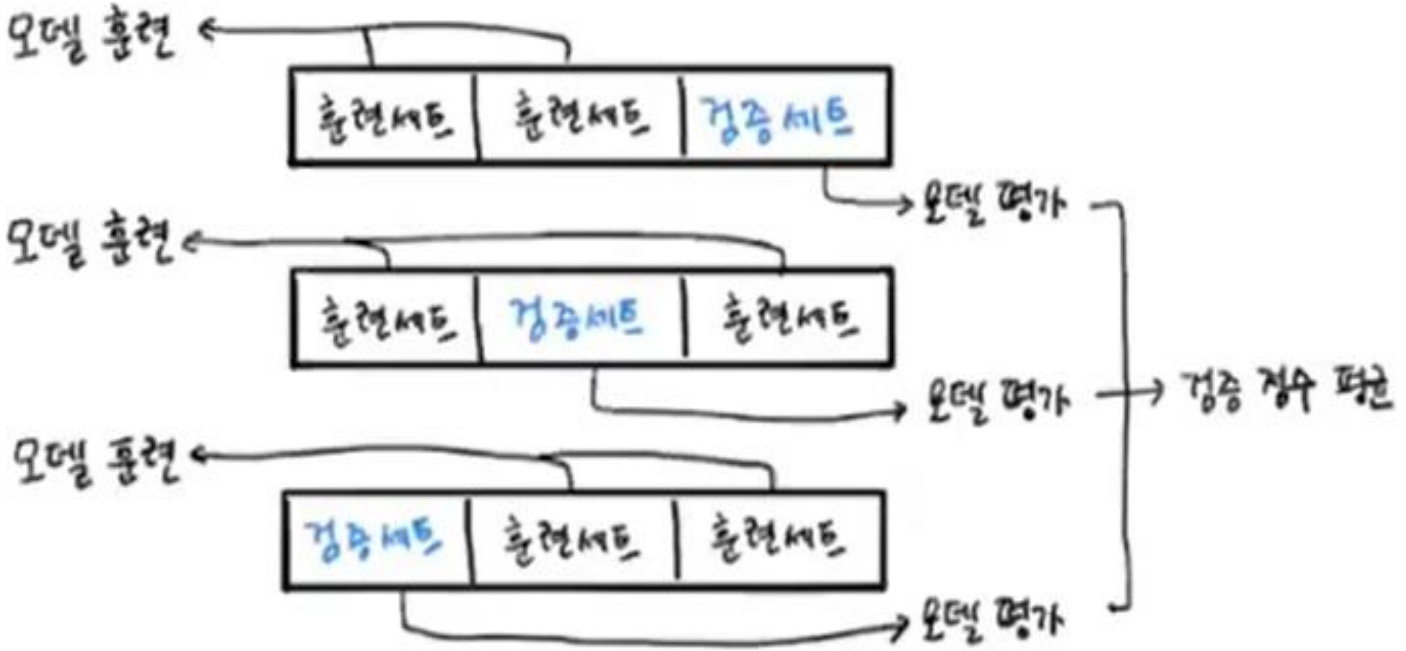
```
0.7579372715027901
0.7376923076923076
```

```
print(dt.score(train_scaled, train_target))
0.996921300750433
print(dt.score(test_scaled, test_target))
0.8592307692307692
```

검증 세트

- 훈련 세트와 테스트 세트에 더불어서 검증 세트가 필요하다
- 과대적합 과소적합을 판단하기 위새 검증 세트를 사용한다.(validation set)
- 테스트 세트는 처음 본 데이터(실전에 투입했을 때 받는 새로운 샘플) 모델을 만들고 마지막에 사용하는 것

교차 검증



훈련 세트의 양이 많지 않을 때에 사용하는 방식
StratifiedKFold 메소드를 통해서 교차검증의 횟수 지정

GridSearchCV

- 각각의 매개변수들이 서로에게 영향을 줄 수 있기 때문에 순차적으로 매개변수 찾기는 힘들다
  - Ex) max_depth / min_sample_split

- GridSearchCV 를 사용하여 매개변수를 일일이 바꿔가며 교차검증을 수행하지 않고 원하는 매개변수 값을 나열하면 자동으로 교차 검증을 수행한다.

```
params = {'min_impurity_decrease': np.arange(0.0001, 0.001, 0.0001),
          'max_depth' : range(5,20,1),
          'min_sample_split' : range(2, 100, 10)}

gs = GridSearchCV(DecisionTreeClassifier(random_state=42), params, n_jobs=-1)
gs.fit(train_input, train_target)
```

랜덤 서치

- 그리드 서치에서는 간격을 지정해서 그 간격을 모두 계산해야 하여서 수행시간이 길어진다.
- 간격을 균등 분포로 샘플링하여 수행 시간을 줄이는 방법

```
params = {'min_impurity_decrease': uniform(0.0001, 0.001),
          'max_depth' : randint(20,50),
          'min_sample_split' : randint(20,50),
          'min_sample_split' : randint(1,25),
          }

from sklearn.model_selection import RandomizedSearchCV

gs = RandomizedSearchCV(DecisionTreeClassifier(random_state=42),params,
                        n_iter=100, n_jobs= -1, random_state=42)
gs.fit(train_input, train_target)
```

- https://www.youtube.com/watch?v=ZaIKUvHquEQ&list=PLVsNizTWUw7HpqmdphX9hgyWl15nobgQX&index=12
- https://www.youtube.com/watch?v=tOzxDGp8rsg&list=PLVsNizTWUw7HpqmdphX9hgyWl15nobgQX&index=11