Hyperparameter Tuning

Hyperparameter?

모델을 학습하기 전에 사전에 설정해야 하는 값들
이 값은 머신러닝이 스스로 찾을 수 없기 때문에, 사람이 직접 찾아줘야 한다

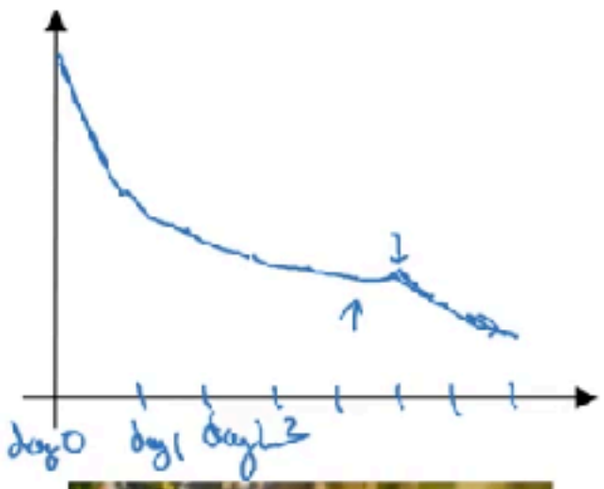
```
from lightgbm import LGBMClassifier

model = LGBMClassifier(boosting_type='gbdt',
                       n_estimators=100,
                       learning_rate=0.1,
                       num_leaves=31,
                       max_bin=255,
                       min_child_samples=20,
                       min_split_gain=0.0,
                       subsample=1.0,
                       subsample_freq=0,
                       colsample_bytree=1.0,
                       n_jobs=-1,
                       random_state=random_state)
```

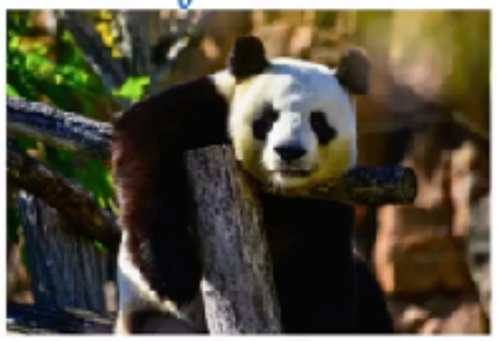
이 값들이 전부 hyperparameter다

Panda vs Caviar

크게 하나의 모델을 붙잡고 튜닝하는 Panda 전략,
여러개의 모델을 동시에 돌려서 가장 좋은 모델을 선택하는 Caviar 전략이 있다

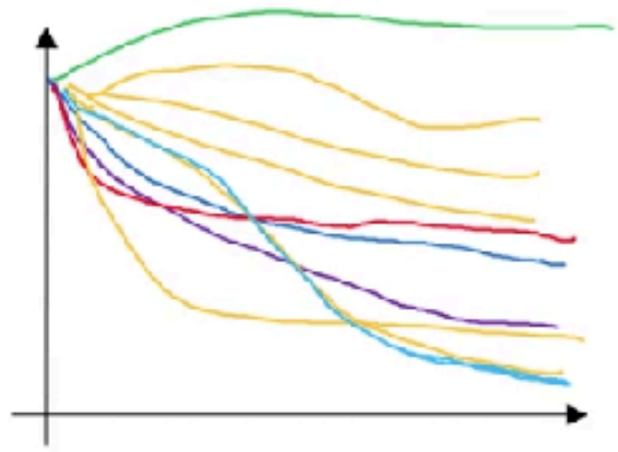
Babysitting one model





Panda

Training many models in parallel

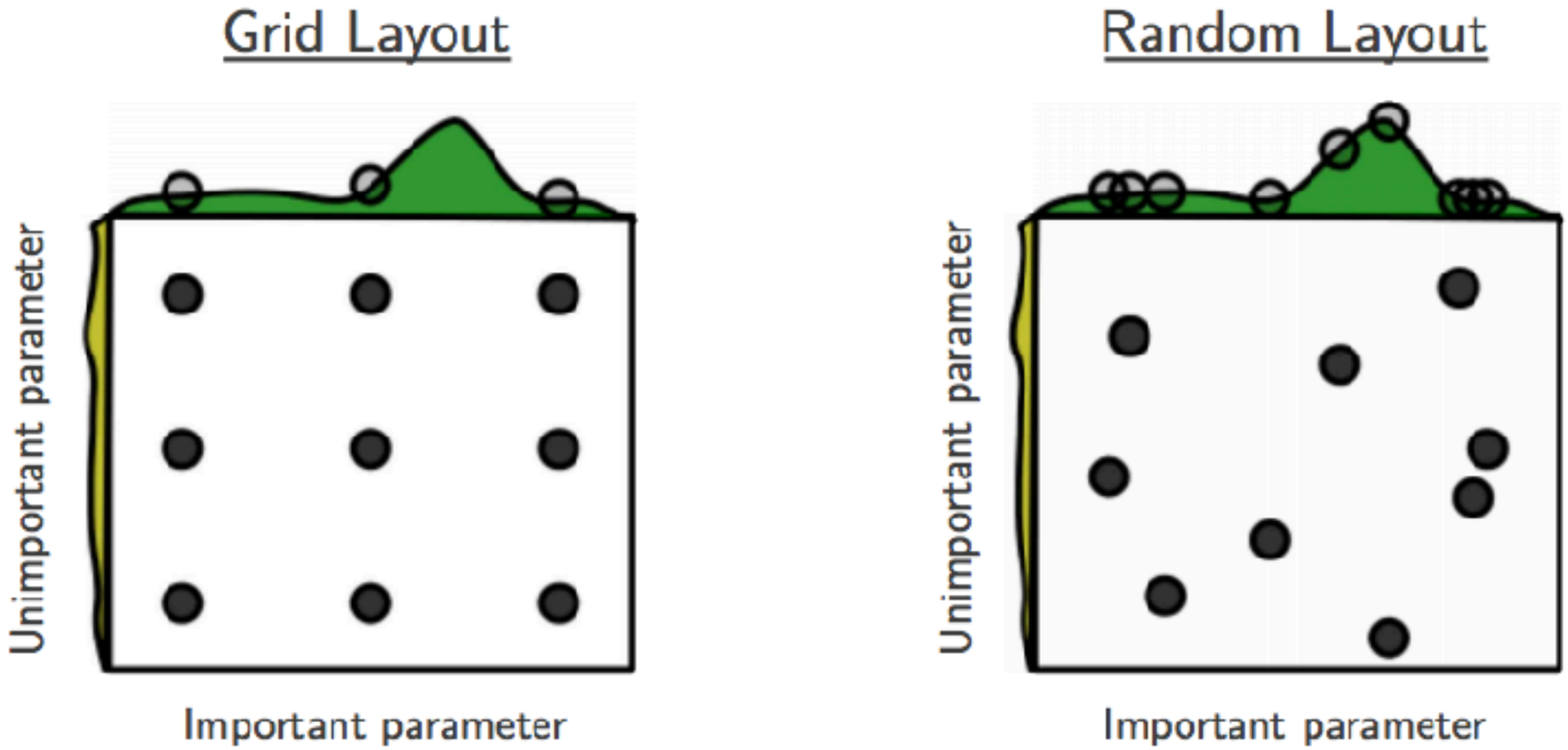


Caviar

Andrew N

Caviar Approach - Random Search

하이퍼패러미터의 범위를 정한 뒤, 그 범위 안에서 random search를 한다
Caviar approach 중 가장 유명하고 성능이 잘 나오는 전략 중 하나



Bergstra, et al. 2012. Random Search for Hyper-Parameter Optimization
https://goo.gl/fuYhbE

Panda Approach 1

중요하지 않은 hyperparameter는 수정할 필요가 없다
중요한 몇몇 hyperparameter를 집중적으로 수정한다

모델의 정확도를 늘리고 싶다면

- n_estimators를 높게, learning_rate를 낮게 준다.
- max_bin을 크게 준다. (느려지고, overrfitting될 여지가 있음)
- num_leaves을 크게 준다. (overfitting될 여지가 있음)

Overfitting을 방지하고 싶다면

- max_bin을 작게 준다.
- num_leaves을 작게 준다.
- min_child_samples, max_depth를 줘서 가지를 덜 치게 한다.
- min_gain_to_split을 줘서 가지를 덜 치게 한다.
- subsample, subsample_freq, colsample_bytree 를 줘서 몇몇 outlier 데이터에 영향을 덜 미치게 한다.

LightGBM Parameters Tuning Guide
https://goo.gl/qFpy3C

Panda Approach 2

n_estimators와 learning_rate를 고정한 뒤 나머지 hyperparameter를 튜닝한다
마지막으로 n_estimators와 learning_rate를 조정한다

먼저 n_estimators와 learning_rate를 고정한다.

- n_estimators는 시간이 오래 걸리니 100 정도로 고정하고,
- 이 n_estimators에서 가장 성능이 잘 나오는 learning_rate를 찾는다

이후 나머지 hyperparameter를 튜닝한다.

- Tree hyperparameter - max_depth, min_child_samples, max_bin, etc
- Bagging hyperparameter - subsample, subsample_freq, colsample_bytree

마지막으로 n_estimators를 늘리고, learning_rate를 줄인다.

- n_estimators를 절반으로 늘였으면, learning_rate는 반으로 줄인다.
- n_estimators를 10배로 늘였으면, learning_rate는 1/10으로 줄인다.

Complete Guide to Parameter Tuning in XGBoost (with codes in Python)
https://goo.gl/efXktr