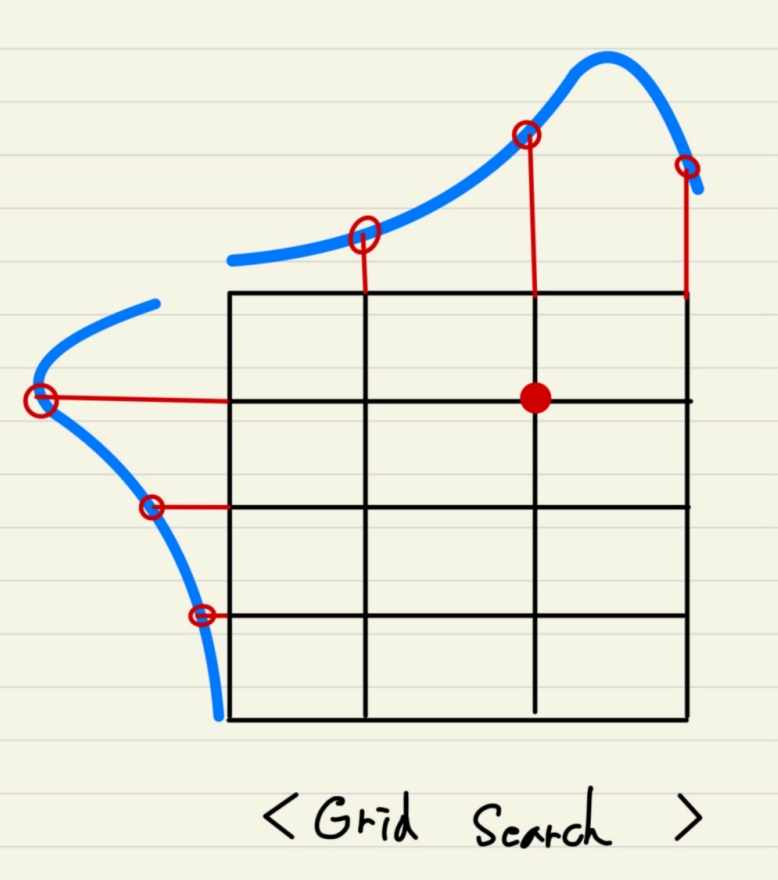
1. HyperParameter Tuning

Function	XGBoost	Light GBM
Important parameters which control overfitting	1. **learning_rate or eta** – optimal values lie between 0.01-0.2 2. **max_depth** 3. **min_child_weight:** similar to min_child leaf; default is 1	1. **learning_rate** 2. **max_depth**: default is 20. Important to note that tree still grows leaf-wise. Hence it is important to tune **num_leaves** (number of leaves in a tree) which should be smaller than 2^(max_depth). It is a very important parameter for LGBM 3. **min_data_in_leaf**: default=20, alias= min_data, min_child_samples
Parameters for categorical values	Not Available	1. **categorical_feature:** specify the categorical features we want to use for training our model
Parameters for controlling speed	1. **colsample_bytree**: subsample ratio of columns 2. **subsample**: subsample ratio of the training instance 3. **n_estimators**: maximum number of decision trees; high value can lead to overfitting	1. **feature_fraction:** fraction of features to be taken for each iteration 2. **bagging_fraction:** data to be used for each iteration and is generally used to speed up the training and avoid overfitting 3. **num_iterations:** number of boosting iterations to be performed; default=100

Learning_rate	Max_depth	N_estimators
0.0001	3	100
0.001	5	300
0.01	10	500
0.1	20	1,000

1. Grid SearchCV



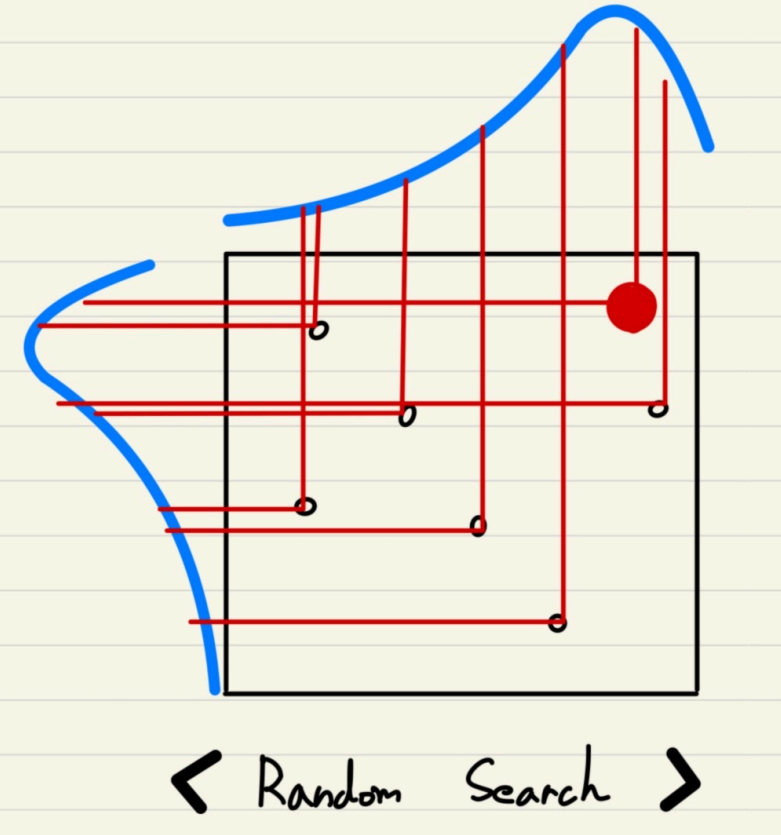
기법 : Grid Search는 사전에 탐색할 값들을 미리 지정해주고,
그 값들의 모든 조합을 바탕으로 성능의 최고점을 찾아냅니다.

장점 : 내가 원하는 범위를 정확하게 비교 분석이 가능합니다.

단점 : 시간이 오래걸린다.
(4개의 파라미터에 대해서, 4가지 값들을 지정해두고,
한 번 탐색하는데 1분이 걸린다면 -> 4*4*1분 = 16분 소요)
성능의 최고점이 아닐 가능성이 높다.

"최적화 검색" (여러개들을 비교 분석해서 최고를 찾아내는 기법)이지,
"최적화 탐색"(성능이 가장 높은 점으로 점차 찾아가는 기법)이 아니다.

2. RandomSearchCV

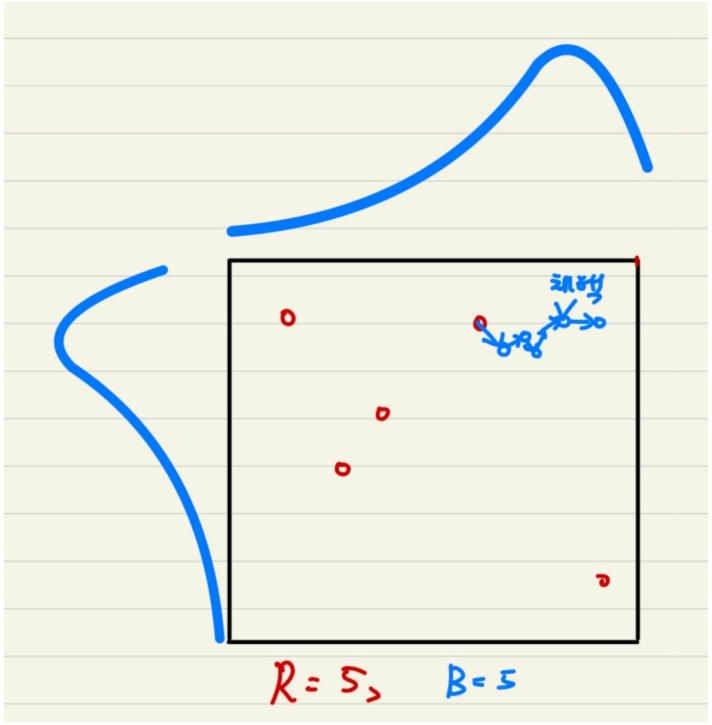


기법 : 사전에 탐색할 값들의 범위를 지정해주고,
그 범위 속에서 가능한 조합을 바탕으로 최고점을 찾아냅니다.

장점 : Grid Search에 비해 시간이 짧게 걸린다.
Grid Search보다, 랜덤하게 점들을 찍으니,
성능이 더 좋은 점으로 갈 가능성이 높다.

단점 : 반대로 성능이 Grid Search보다 낮을 수 있다.
하이퍼파라미터의 범위가 너무 넓으면,
일반화된 결과가 나오지 않는다. (할 때 마다 달라진다)
Seed를 고정하지 않으면, 할 때 마다 결과가 달라진다.
마찬가지로, "최적값 검색"의 느낌이지, "최적화 탐색"의 개념이 아니다.

3. Bayesian Optimization



기법 : 하이퍼파라미터의 범위를 지정한 후,
Random하게 R 번 탐색한 후, B번 만큼 최적의 값을 찾아간다.

장점 : 정말 "최적의 값"을 찾아갈 수 있다.
상대적으로 시간이 덜 걸린다.
엔지니어가 그 결과값을 신뢰할 수 있다.

단점 : Random하게 찍은 값이 달라질 경우, 최적화 하는데 오래 걸릴 수 있다.
값이 부족하면, 최적의 값을 탐색하는게 불가능 할 수 있다.
값이 너무 많으면, 최적화 이전에 이미 최적값을 가지고 있을 수도 있다.