How to improve model performance?

(1) Model Centric Approach :- We are trying to tweak the hyper-parameters of the model or use different set of models.

1A) Choose the right model :-

We train different models with same training dataset & measure the performance of the models in hold-out / test dataset. We can also perform cross-validation & report average cross-validated model performance.
We will choose the model which is performing best.

1B> Choosing the right metric :-

There are several metric we use for assesing the performance of a model.

→ Accuracy : Good measure if the dataset is balanced.

Let's assume a binary classifier. is trained on a imbalanced dataset. [Imbalanced datasets are those where atleast one of the classes has very small representation]

(0) class - 1 (Majority) : 99,900 observations [99.9 %]

(1) class - 2 (Minority) : 100 observations. [0.1 %]

total samples : 100000

Classifier-1 always gives me output 0.

99.9% output will be correct.
0.1% output will be incorrect.

Accuracy = 99.9%

Ideal classifier

Pred \ Actual	0	1
0	99900	0
1	0	100

classifier-1

Pred \ Actual	0	1
0	99900	100
1	0	0

For imbalanced dataset the right metric to choose is F_1-score or Recall or Precision or AUC-ROC

(1C) Hyper-Parameter Tuning

A model comes with lots of hyperparameters.

Decision tree :- tree - depth (max_depth)
minimum no. of samples in leaf nodes.
tree building criteria (GINI, Entropy etc.)
⋮

Random forest :- No. of estimators.
max tree depth
min no. of samples.
⋮

XGBoost :- No. of Estimators.
L_1 regularization parameter, L_2 reg. parameter
Learning rate, min child weight etc.

Neural Network:-

→ No. of hidden layers.

→ No. of nodes in each hidden layer.

Hyperparameter tuning:-

XGBoost

model-1

n_est = 100

α = 2

λ = 1

colsample_bytree = 0.7

model-2

n_est = 200

α = 3

λ = 2

col_sample by level = 0.8

model-3

n_est = 150

α = 1

λ = 0.5

col_sample by level = 0.6

. — — — . model-n

GRID-SEARCH of Hyperparameter:

n_estimators: [50, 100, 150, 200] (4 values)

col_sample_by_tree: [0.6, 0.7, 0.8, 0.9, 1.0] (5 values)

reg_alpha: [0.1, 0.3, 1, 3, 10] (5 values)

reg_lambda: [0.1, 0.3, 1, 3, 10] (5 values)

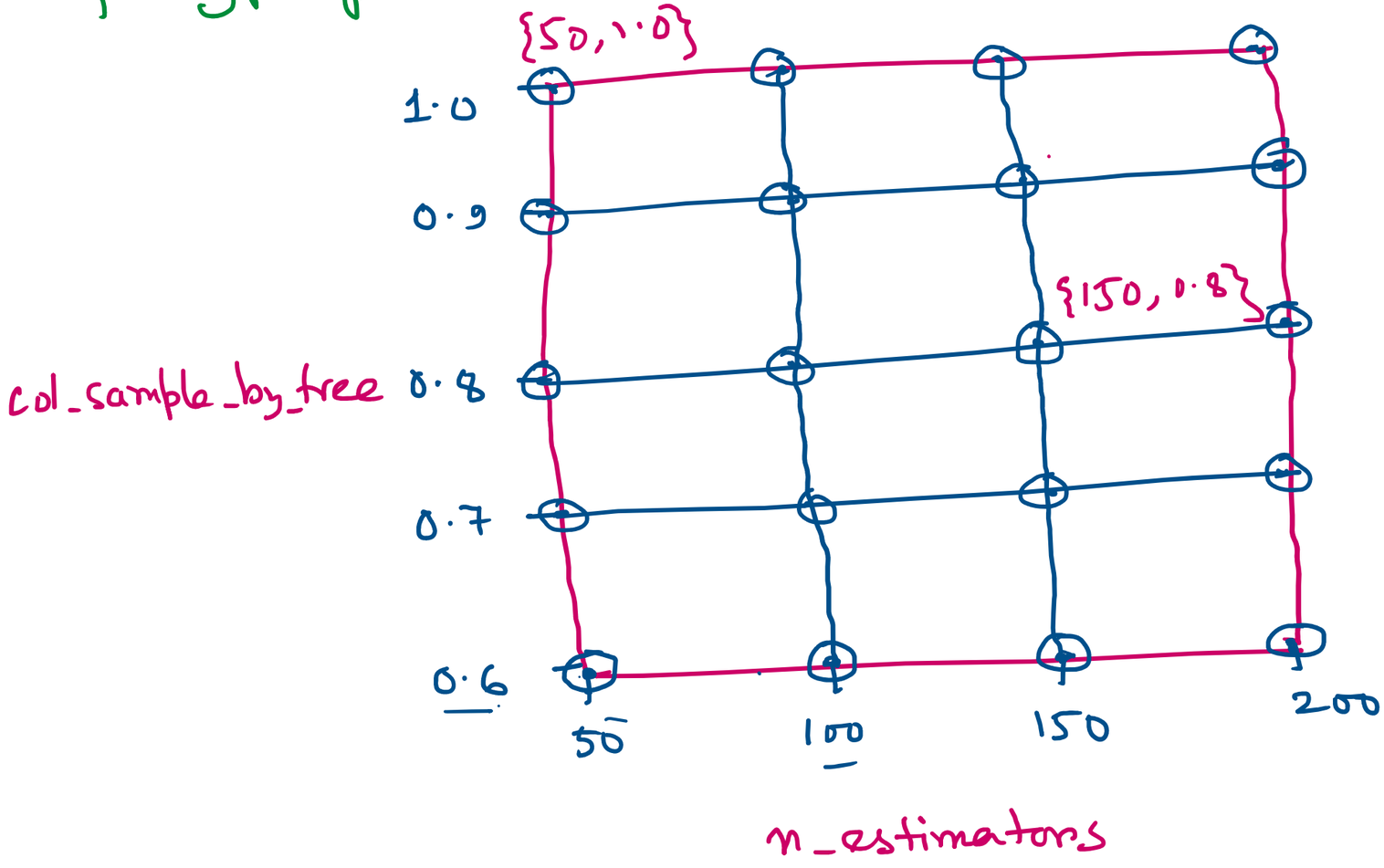
gamma: [5, 10, 15] (3 values)

How many different set of hyperparameters can be formed?

$4 \times 5 \times 5 \times 5 \times 3 = 1500$

Random Search will pick a set of hyperparameters (will not evaluate all the possible combinations) & perform tuning.

Randomized Search is less time consuming compared to Grid search but Randomized search gives less optimum result as it doesn't account for all the possible combinations of hyperparameters.



Each of these nodes corresponds to a set of Hyperparameters.

(2) Data Centric Approach :-

Data that we feed to the models need to be good enough so that model can learn useful information from the data.

2A) Feature Engineering

→ choosing the right features for the model.
(Feature elemination, Feature selection).

→ Combining the features in a meaningful way.
(This requires domain knowledge and experience).

→ Feature transformation
— Feature scaling
— Feature encoding — Log transformation etc.

2B) Using external data :-

2C) Use a sample data to identify good features, models etc.
& then use entire data to train the ML model.

↓

This saves development time.