Random Forests(TM) in XGBoost

- XGB is normally used for gradient-boosted decision trees or models
- Random forest does the same, but with a different training algorithm
- You can use a standalone random forest or a base training model

Standalone Random Forest With XGBoost API

- Booster set to gbtree
- Subsample set to less than 1
- Colsamply_by set to less than 1
- Num_parrallel_tree should be set to the size of the forest
- Num_boost_round set to 1, to prevent boosting multiple random forests
- Eta (learning_rate) set to 1
- Random_state to seed the random number generator
- Objective typically reg:squarederror for regression and binary:logistic for classification
- Sample params

```
params = {
  "colsample_bynode": 0.8,
  "learning_rate": 1,
  "max_depth": 5,
  "num_parallel_tree": 100,
  "objective": "binary:logistic",
  "subsample": 0.8,
  "tree_method": "hist",
  "device": "cuda",
```

Standalone Random Forest With Scikit-Learn-Like API

- XGBRFClassifier and XGBRFRegressor
- Caveats
  - XGB uses 2nd order approximation, this differs from RF implementation even if the objective function is the same value
  - XGBoost does not perform replacement when subsampling training cases. Each training case can occur in a subsampled set either 0 or 1 time.