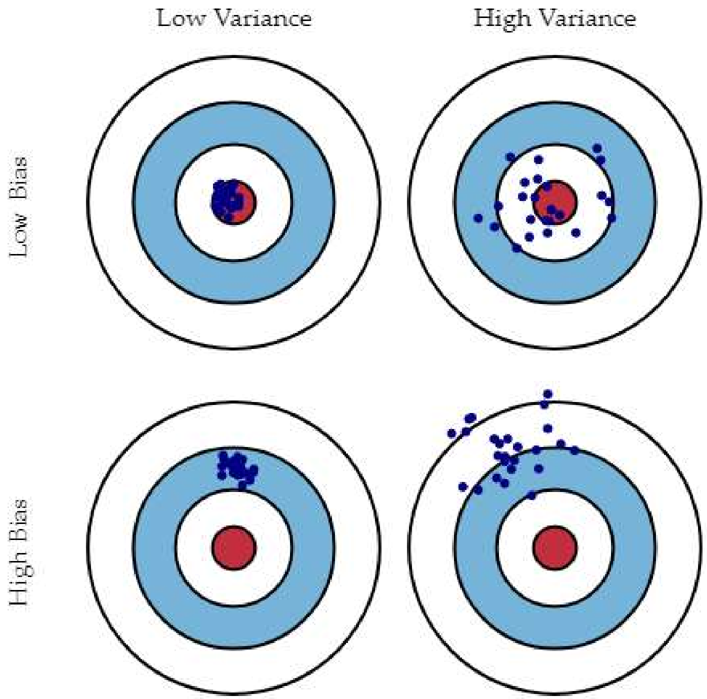
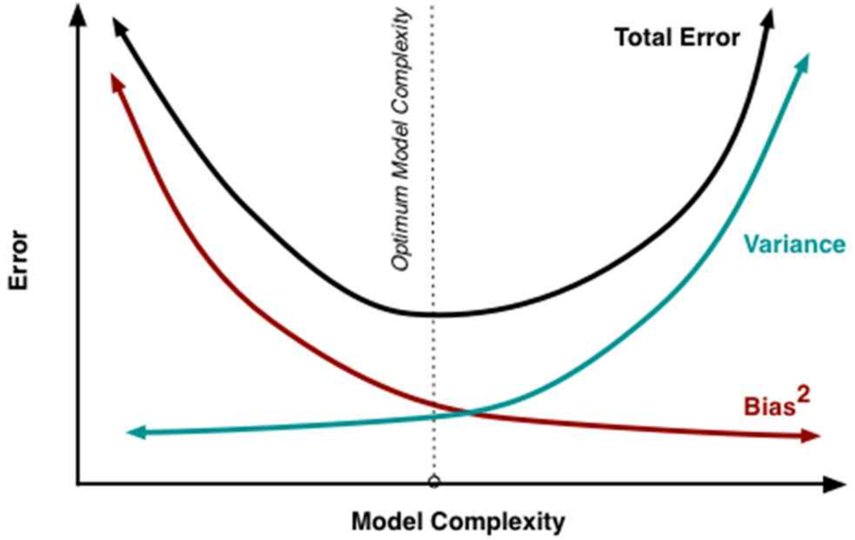
Bagging & Boosting

한성근

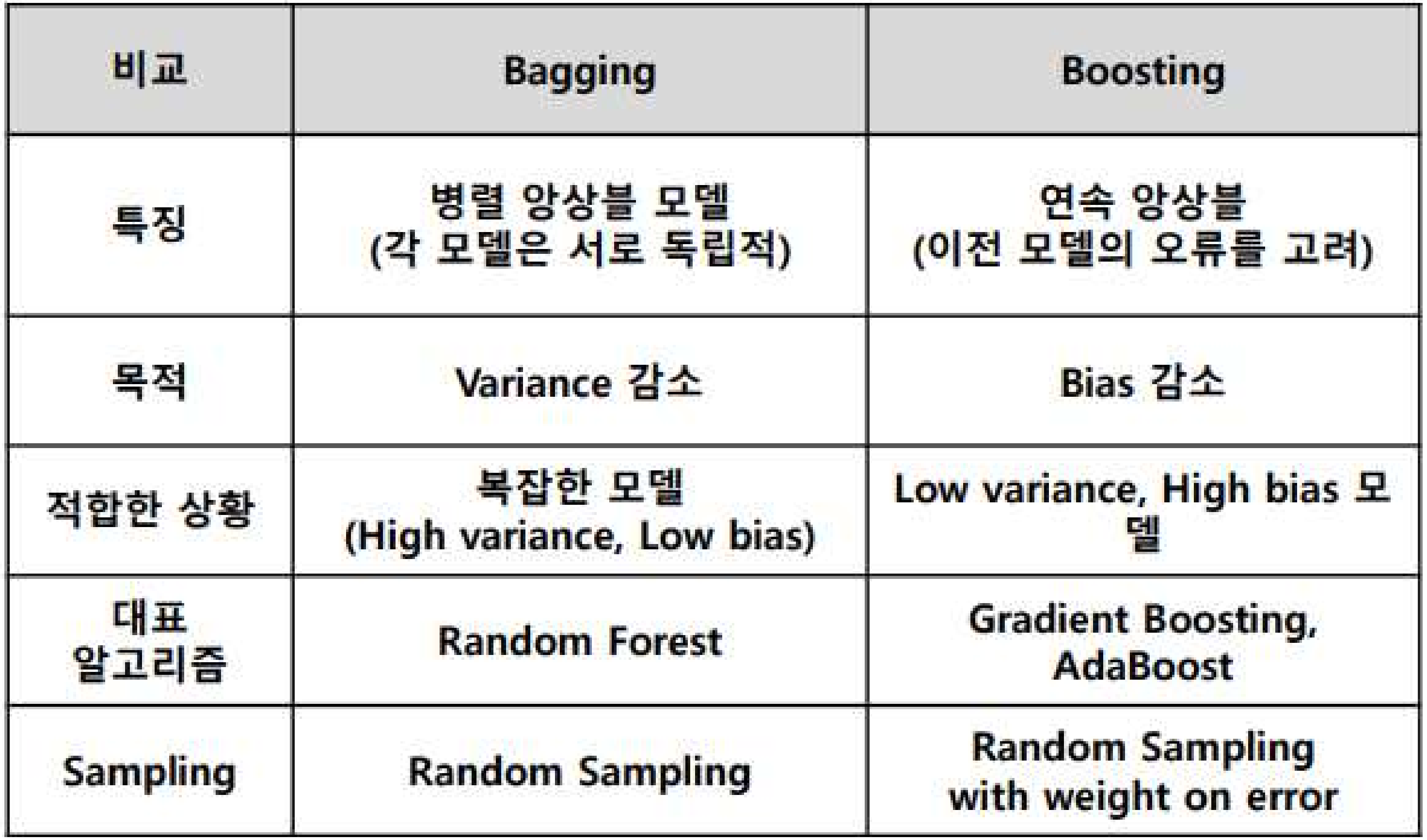
Bias & Variance





http://scott.fortmann-roe.com/docs/BiasVariance.html

Bagging & Boosting 비교



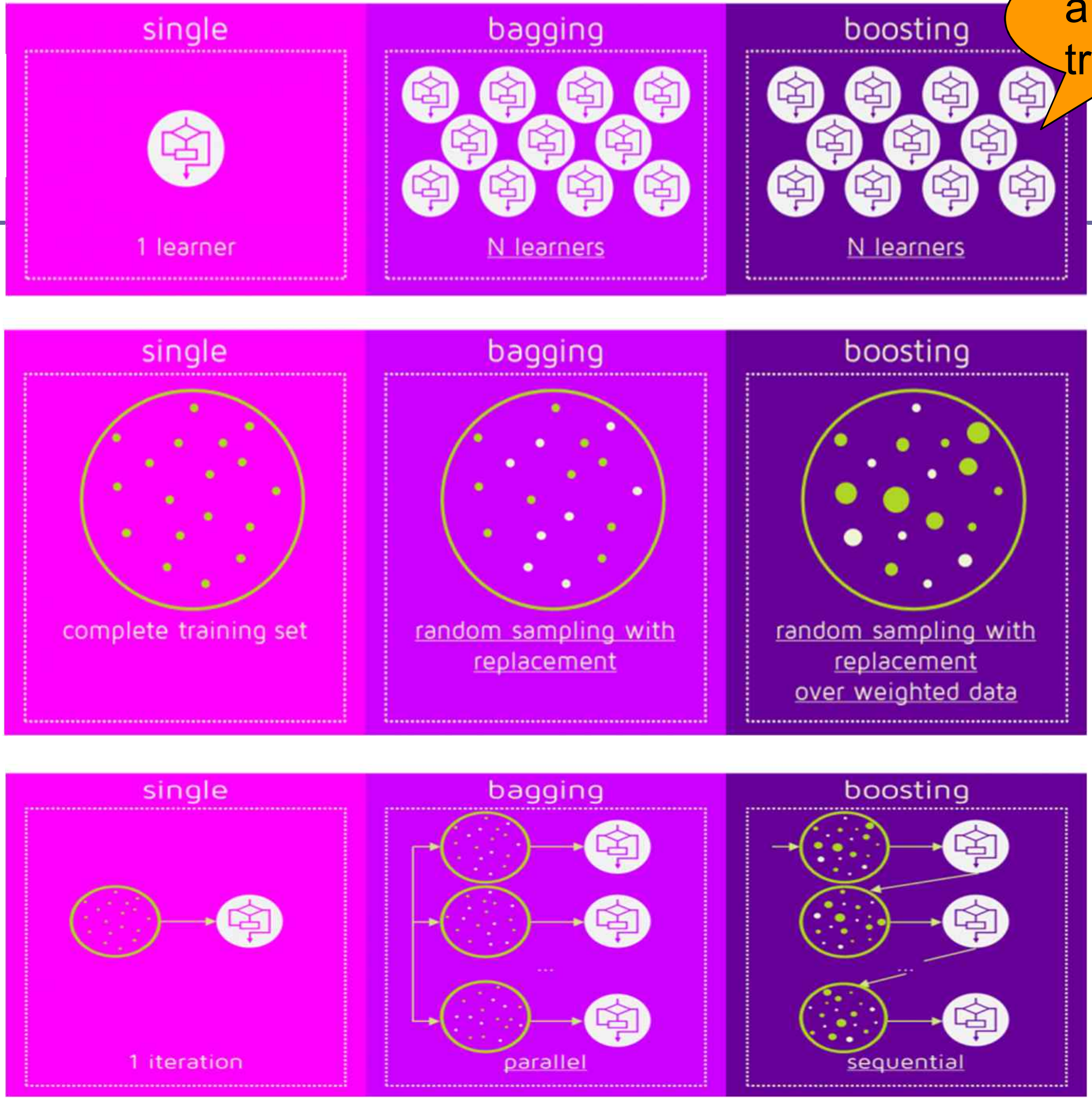
비교	Bagging	Boosting
특징	병렬 앙상블 모델 (각 모델은 서로 독립적)	연속 앙상블 (이전 모델의 오류를 고려)
목적	Variance 감소	Bias 감소
적합한 상황	복잡한 모델 (High variance, Low bias)	Low variance, High bias 모델
대표 알고리즘	Random Forest	Gradient Boosting, AdaBoost
Sampling	Random Sampling	Random Sampling with weight on error

https://www.slideshare.net/freepsw/boosting-bagging-vs-boosting

Select a base learner algorithm

N-random sampling datasets

Training stage

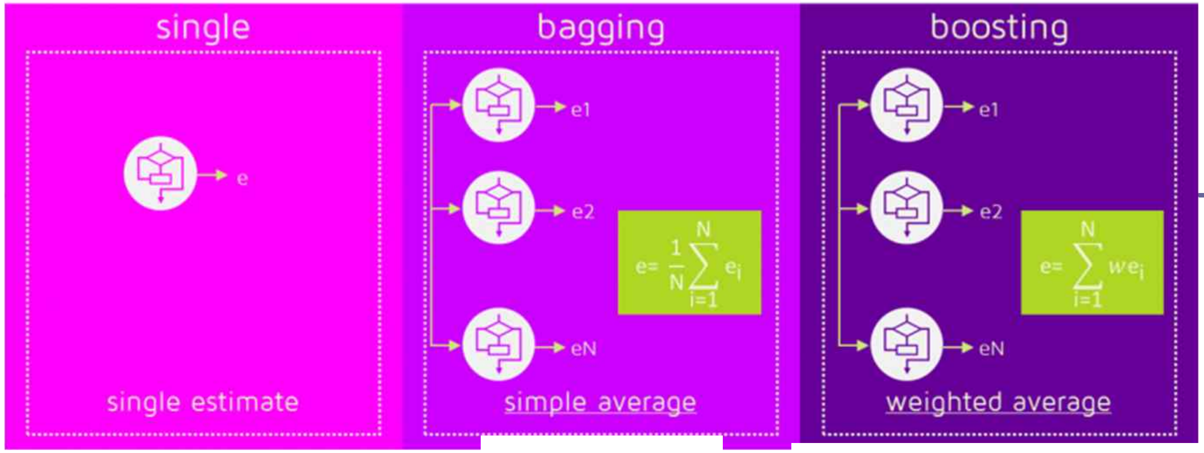


a pool of trees

Parallel

Sequential

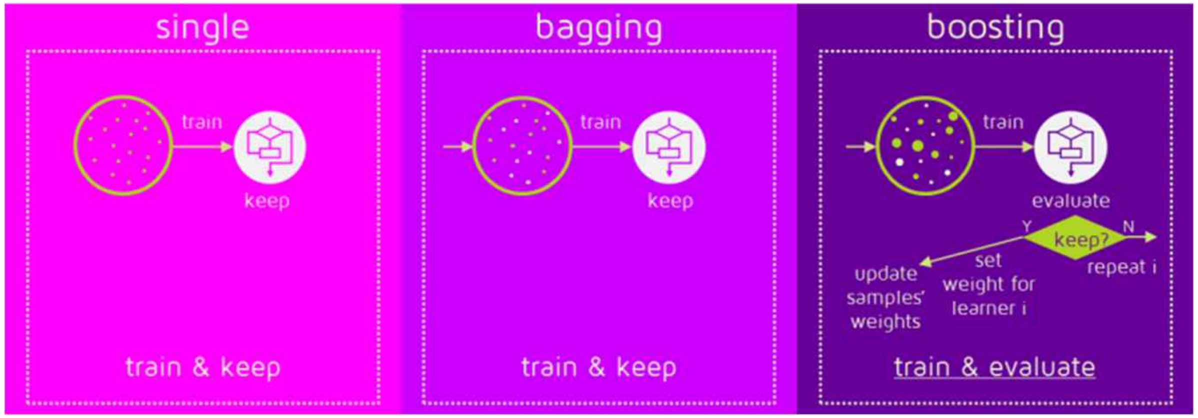
Classification stage



Average

Weighted Average

New data

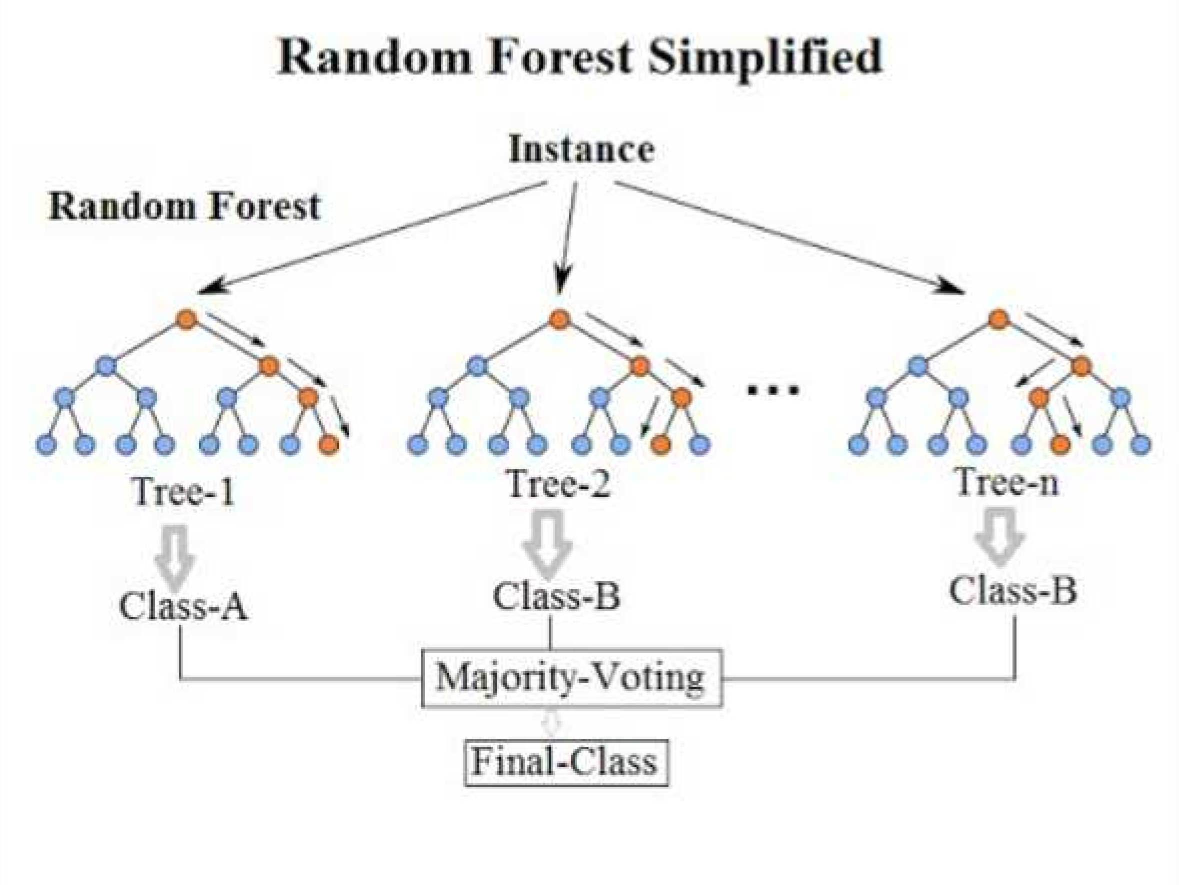


keep

evaluate & update

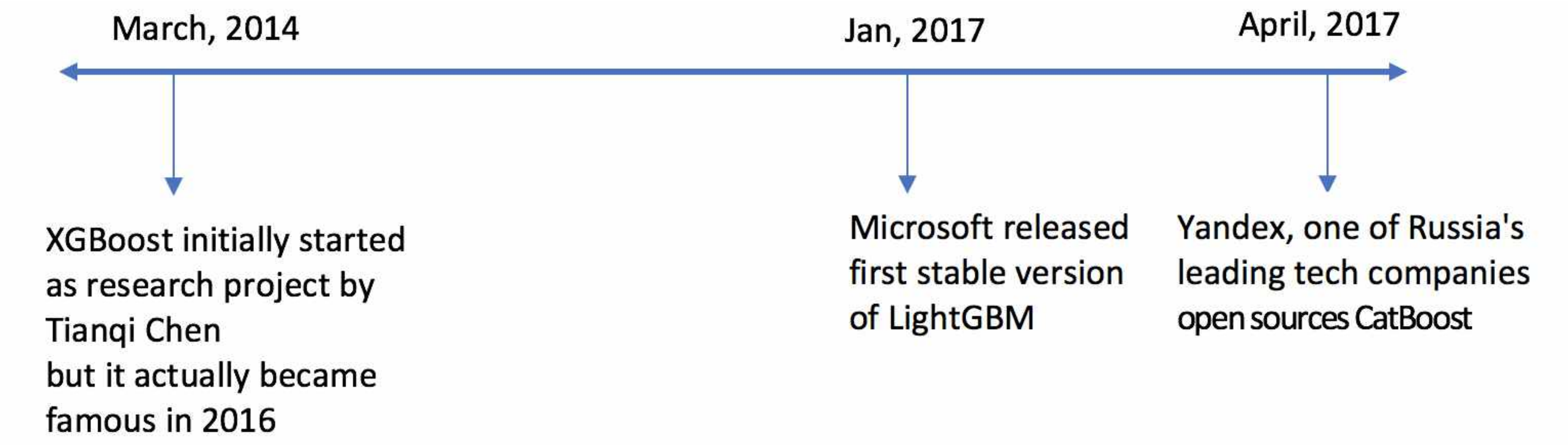
https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

RandomForest



https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

XGBoost / LightGBM / CatBoost



March, 2014

XGBoost initially started as research project by Tianqi Chen
but it actually became famous in 2016

Jan, 2017

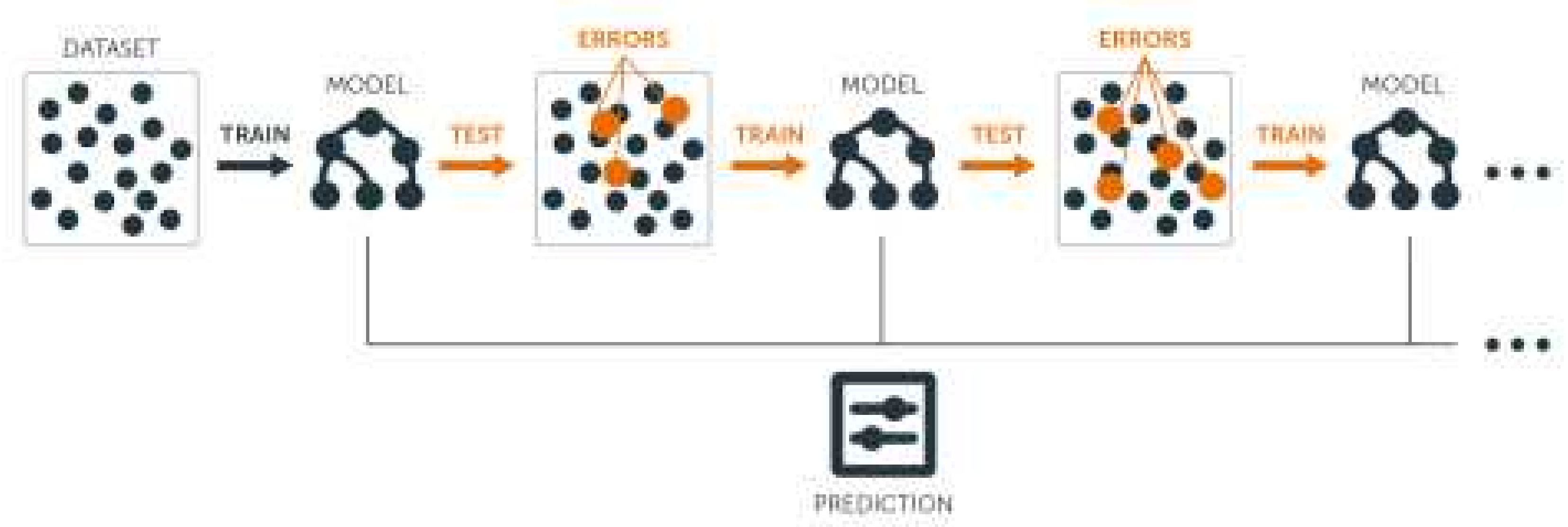
Microsoft released first stable version of LightGBM

April, 2017

Yandex, one of Russia's leading tech companies open sources CatBoost

https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db

Gradient Boosting

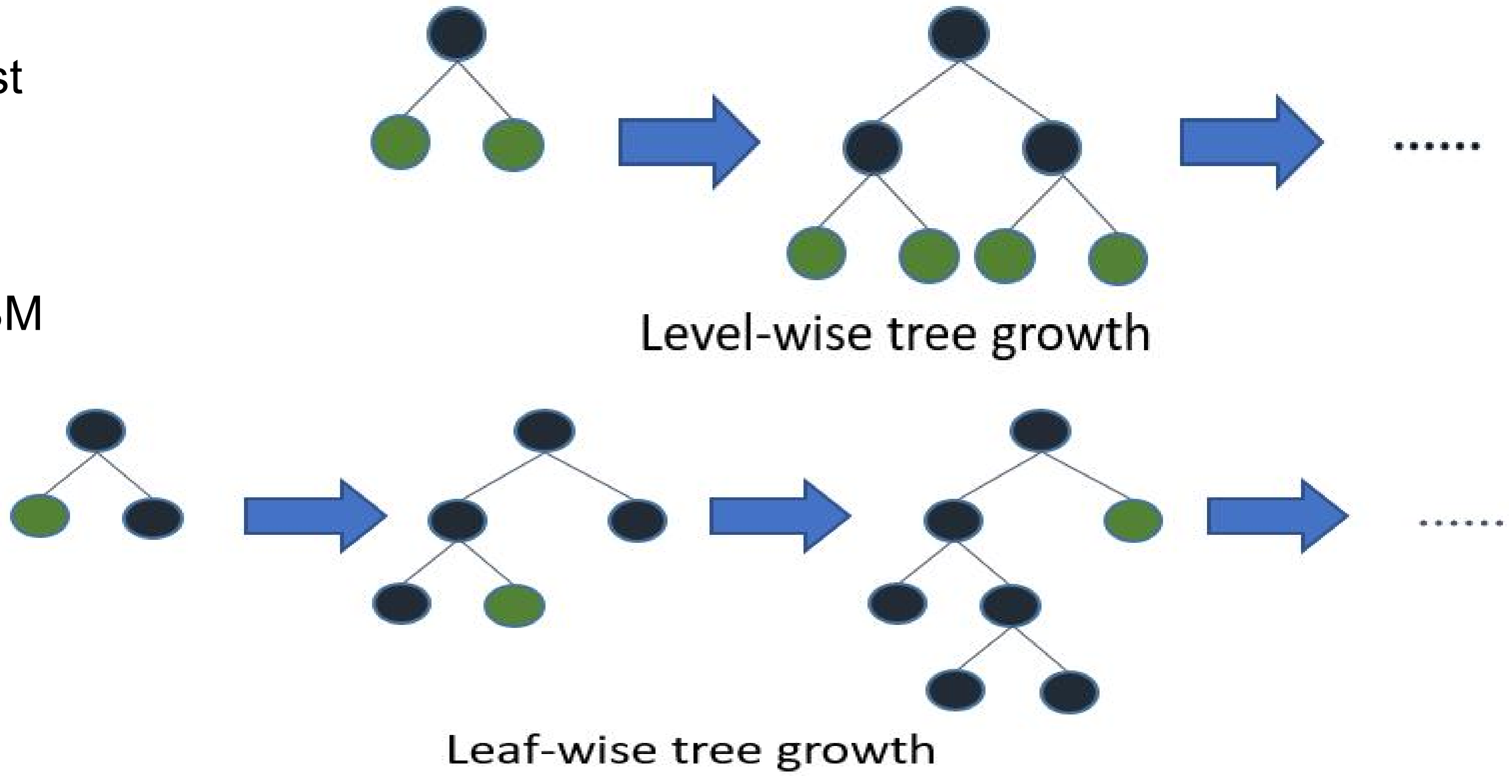


http://uc-r.github.io/gbm_regression

XGBoost & LightGBM

XGBoost

LightGBM



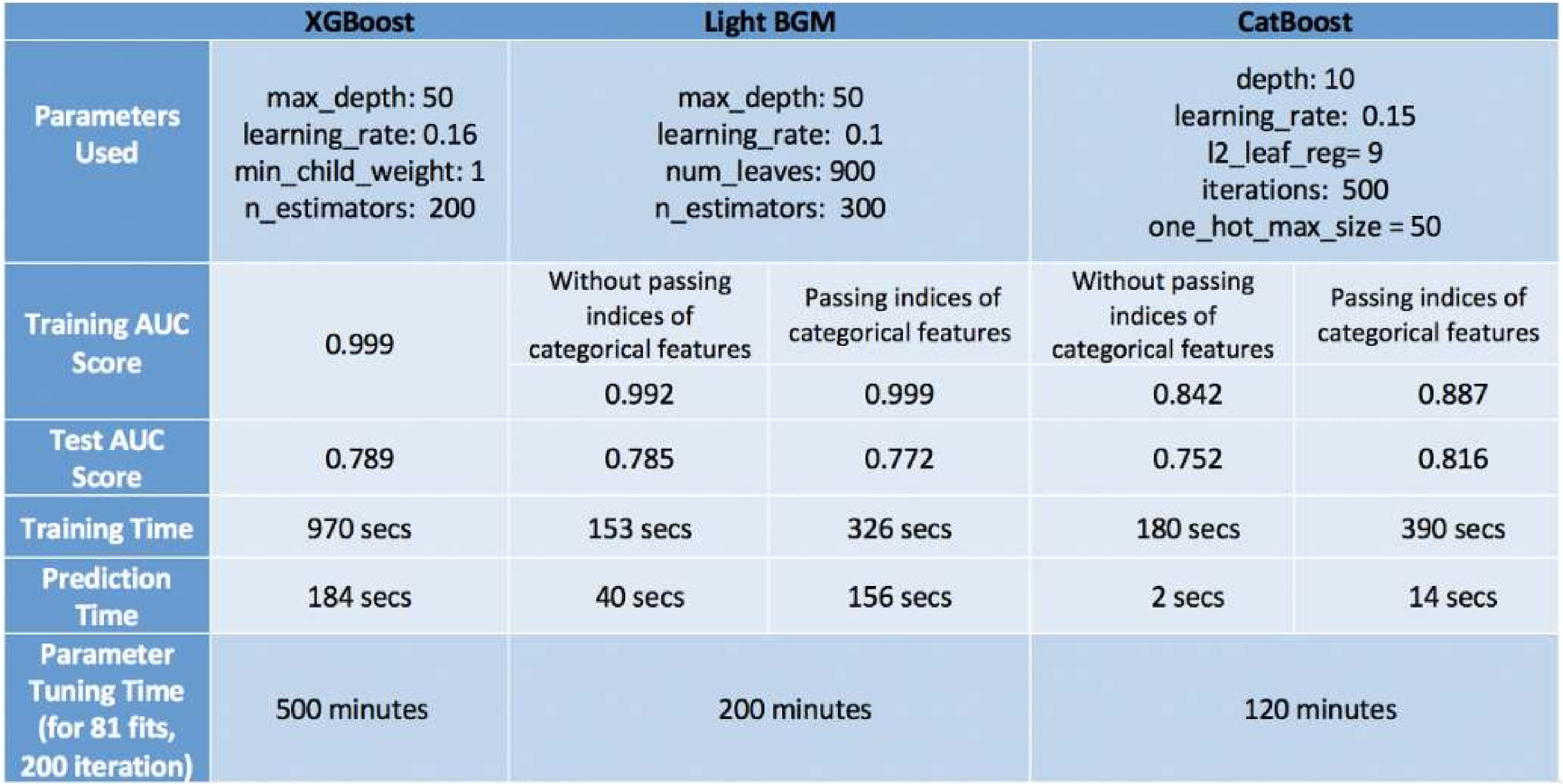
Level-wise tree growth

Leaf-wise tree growth

https://www.slideshare.net/GabrielCyprianoSaca/xgboost-lightgbm

Kaggle dataset of flight delays

https://www.kaggle.com/usdot/flight-delays/data



	XGBoost	Light BGM		CatBoost	
Parameters Used	max_depth: 50 learning_rate: 0.16 min_child_weight: 1 n_estimators: 200	max_depth: 50 learning_rate: 0.1 num_leaves: 900 n_estimators: 300		depth: 10 learning_rate: 0.15 l2_leaf_reg= 9 iterations: 500 one_hot_max_size = 50	
Training AUC Score	0.999	Without passing indices of categorical features	Passing indices of categorical features	Without passing indices of categorical features	Passing indices of categorical features
		0.992	0.999	0.842	0.887
Test AUC Score	0.789	0.785	0.772	0.752	0.816
Training Time	970 secs	153 secs	326 secs	180 secs	390 secs
Prediction Time	184 secs	40 secs	156 secs	2 secs	14 secs
Parameter Tuning Time (for 81 fits, 200 iteration)	500 minutes	200 minutes		120 minutes	

train(categorical_feature =cate_features_name)

https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db