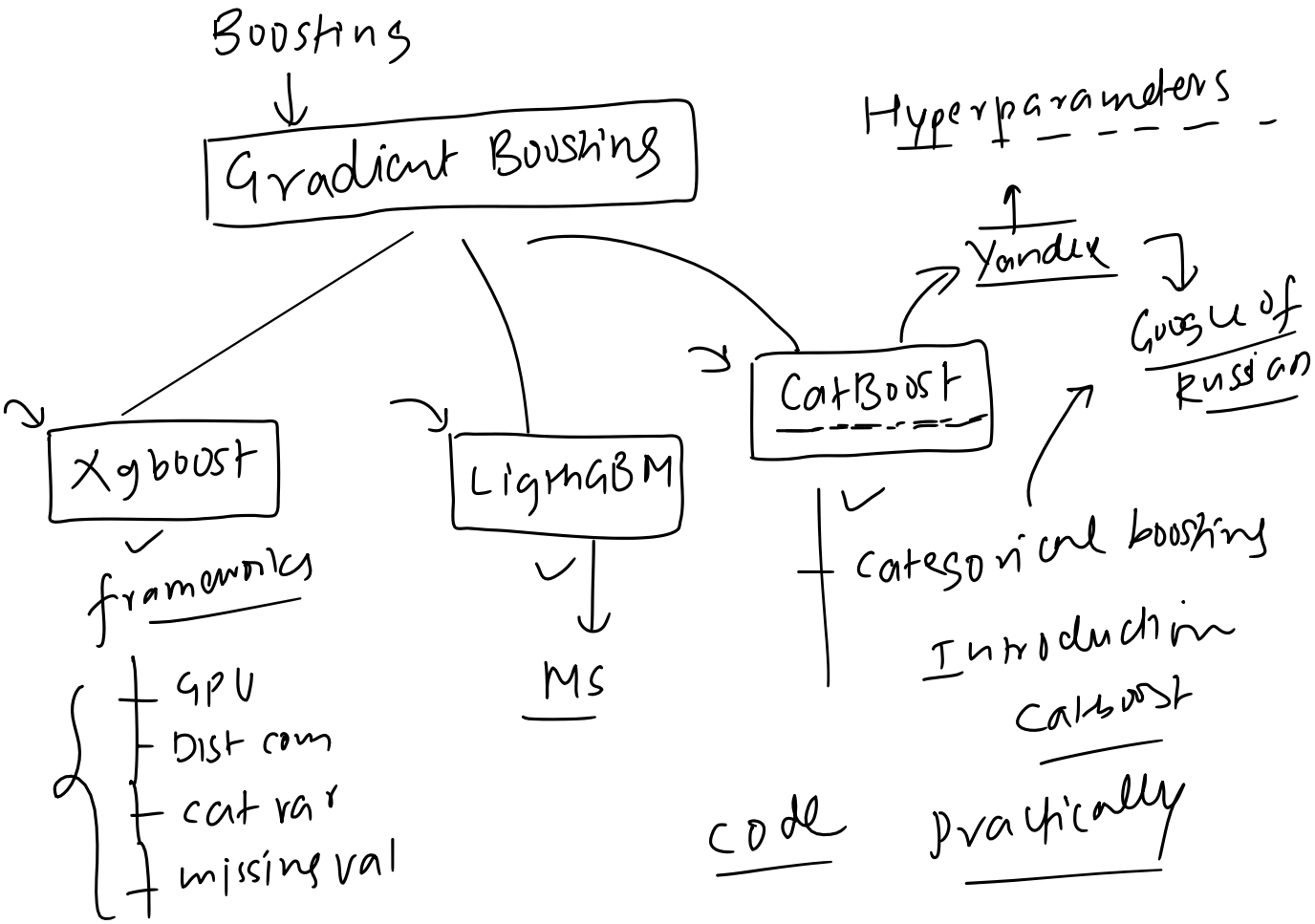
Plan of Attack

20 April 2024 19:59



Introduction

19 April 2024 10:08

CatBoost is an algorithm for gradient boosting on decision trees. It is developed by Yandex researchers and engineers, and is used for search, recommendation systems, personal assistant, self-driving cars, weather prediction and many other tasks at Yandex and in other companies, including CERN, Cloudflare, Careem taxi. It is in open-source and can be used by anyone.

Advantages

1. Easy to use
2. Great result without hyperparameter tuning
3. Improved accuracy -> benchmark

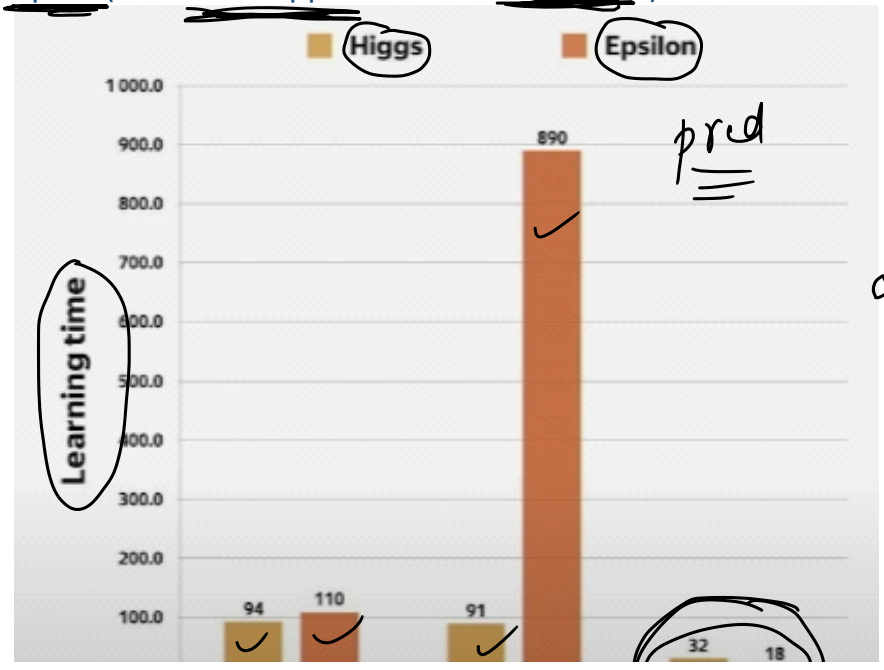
1.

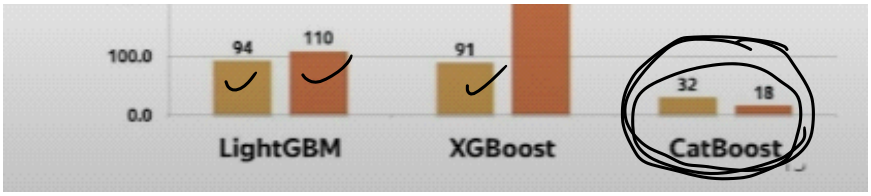
	CatBoost	LightGBM		XGBoost		H2O	
Adult	0.269741	0.276018	+2.33 %	0.275423	+2.11%	0.275104	+1.99%
Amazon	0.137720	0.163600	+18.79 %	0.163271	+18.55%	0.162641	+18.09%
Appet	0.071511	0.071795	+0.40 %	0.071760	+0.35%	0.072457	+1.32%
Click	0.390902	0.396328	+1.39 %	0.396242	+1.37%	0.397595	+1.71%
Internet	0.208748	0.223154	+6.90 %	0.225323	+7.94%	0.222091	+6.39%
Kdd98	0.194668	0.195759	+0.56 %	0.195677	+0.52%	0.195395	+0.37%
Kddchurn	0.231289	0.232049	+0.33 %	0.233123	+0.79%	0.232752	+0.63%
Kick	0.284793	0.295660	+3.82 %	0.294647	+3.46%	0.294814	+3.52%

Logloss

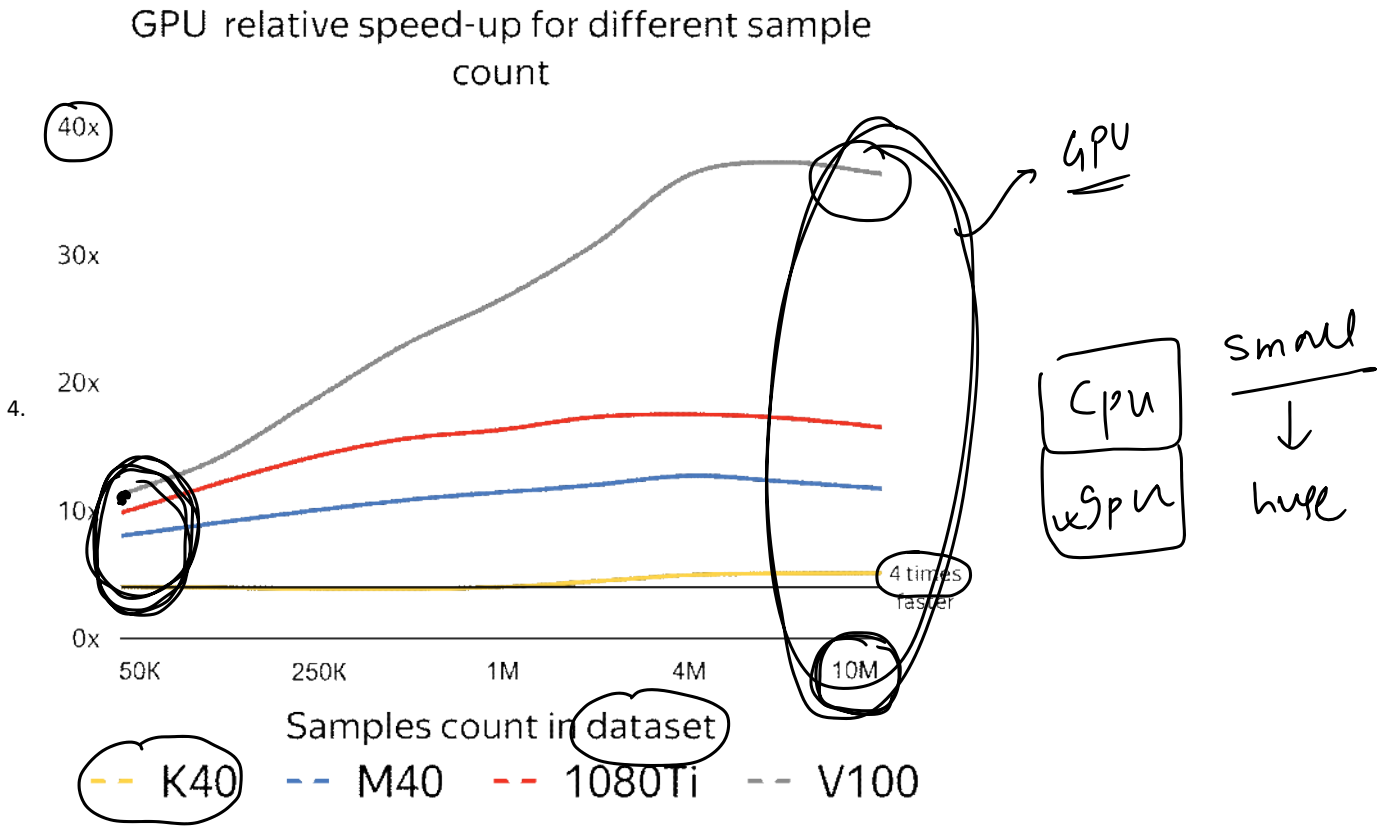
1. Sophisticated Categorical Feature support + textual features
2. Speed(Fast GPU support + CPU + Prediction) -> benchmarks

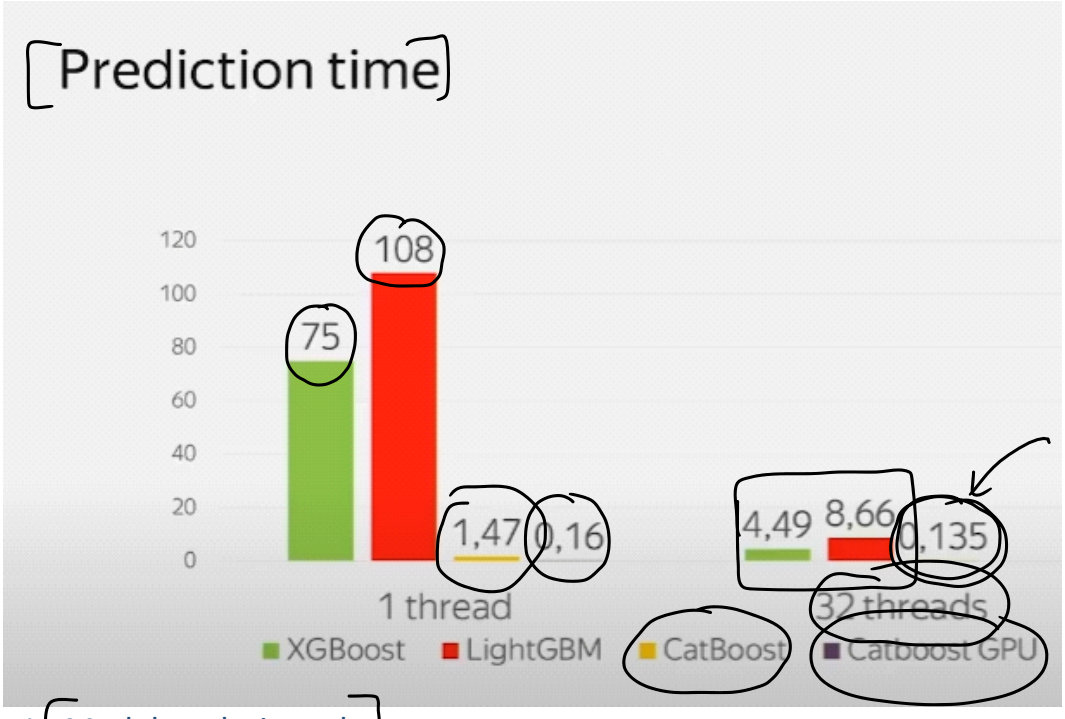
3.



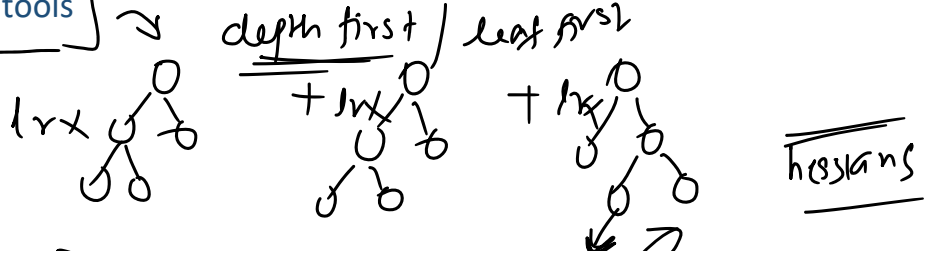


4.



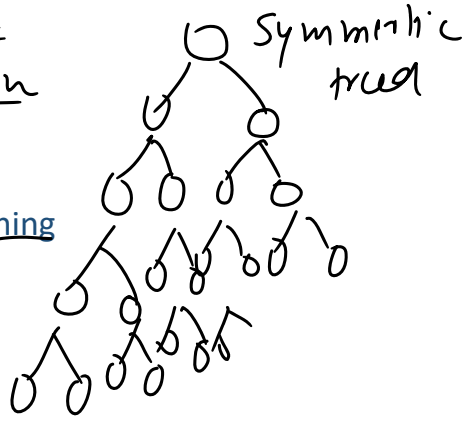


1. Model analysis tools

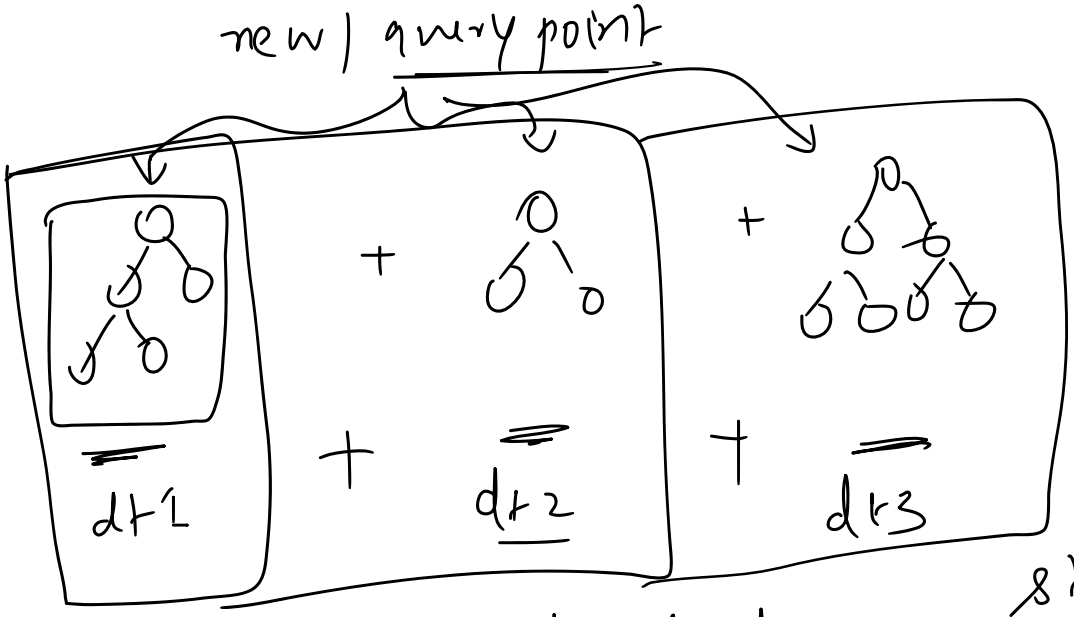


Technical Aspects →

1. Can handle categorical variables(text data also) using Ordered Target Encoding
2. Uses Symmetric Trees
3. Uses technique like Newton Raphson to calculate the output value of the leaves
4. Can dynamically figure out learning rate → dynamic
5. Snapshotting capability
6. Native integration with libraries such as SHAP, Plotly etc
7. Usage of smart data structures like Pool →
8. Built-in support for cross validation and hyperparameter tuning
9. Can handle missing values out of the box
10. Built in overfitting detector
11. Supports custom loss functions and metrics
12. Multi-threading and GPU support
13. Regularization



early stopping



staged output

start 5

end 10

Stage

Stage

Stage 2