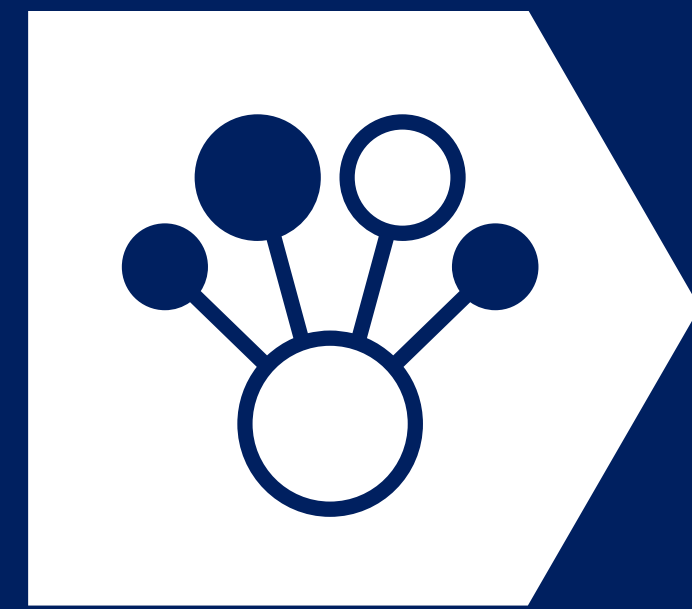



Yandex



CatBoost

The new generation of Gradient Boosting

CatBoost

 **catboost** / **catboost**

Unwatch 167

Unstar 3,033

Fork 399

Code

Issues 77

Pull requests 2

Insights

Settings

CatBoost is an open-source gradient boosting on decision trees library with categorical features support out of the box for Python, R <https://catboost.yandex> Edit

machine-learning

decision-trees

gradient-boosting

gbm

gbdt

python

r

kaggle

gpu-computing

catboost

tutorial

categorical-features

distributed

gpu

coreml

opensource

data-science

big-data

Manage topics

2,268 commits

8 branches

23 releases

70 contributors

Apache-2.0

Branch: master


New pull request

Create new file

Upload files

Find file

Clone or download

 **andrey-khropov** Add missing PEERDIR. ...

Latest commit 370e11f 3 hours ago

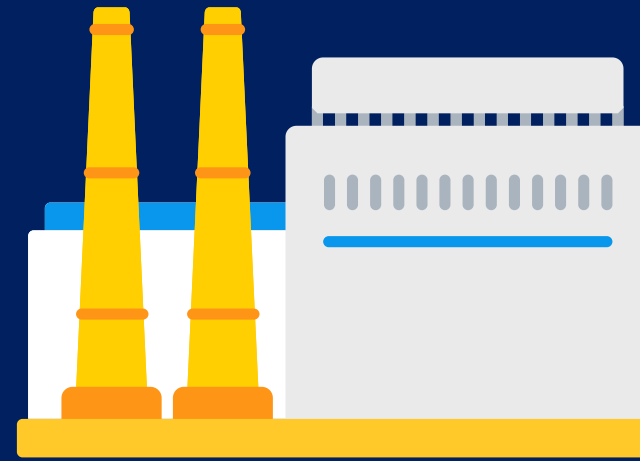
Gradient Boosting

- › Best solution for heterogeneous data
- › Easy to use
- › Works well for small data

Applications



Medicine



Industry



Finance

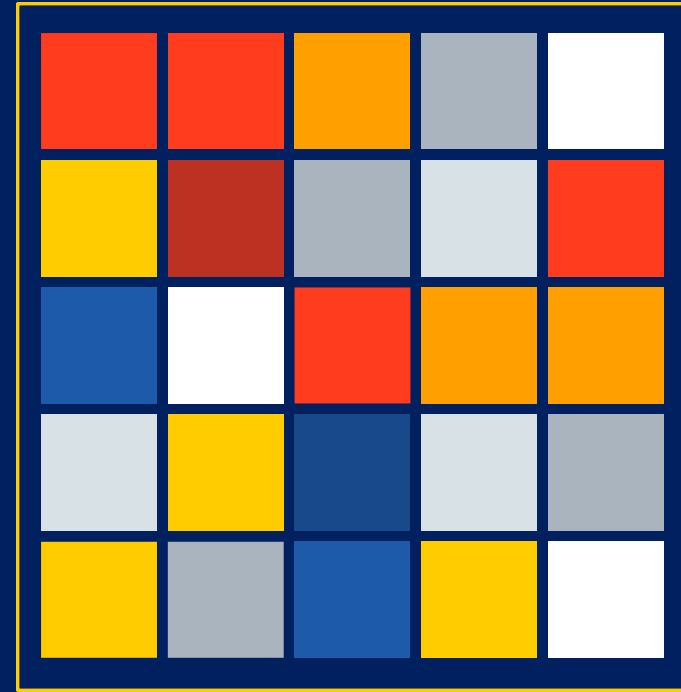


Music and video
recommendations



Sales prediction

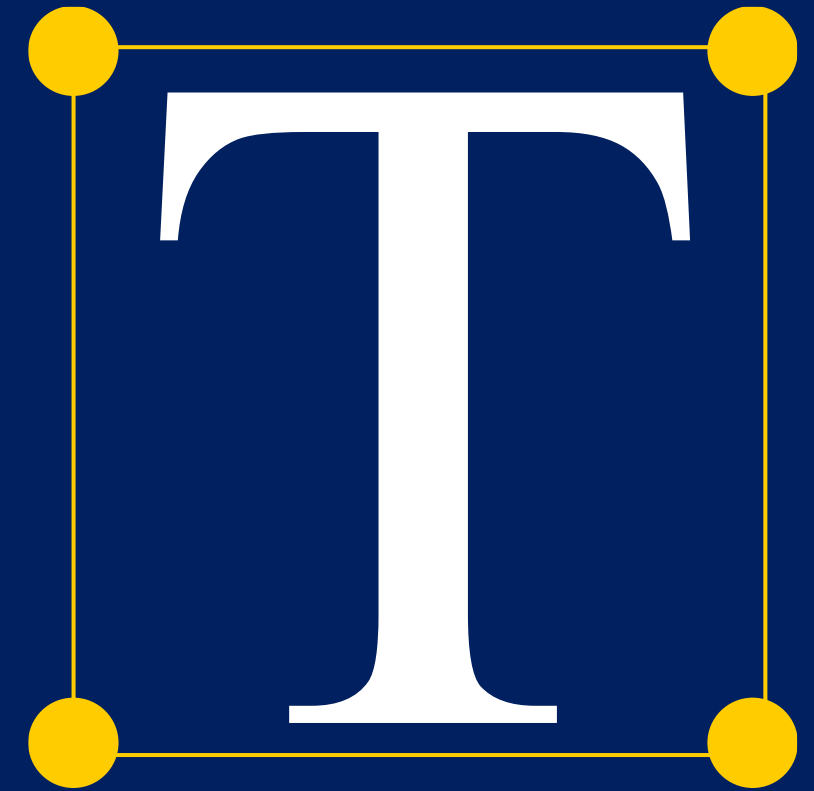
Neural networks



Images

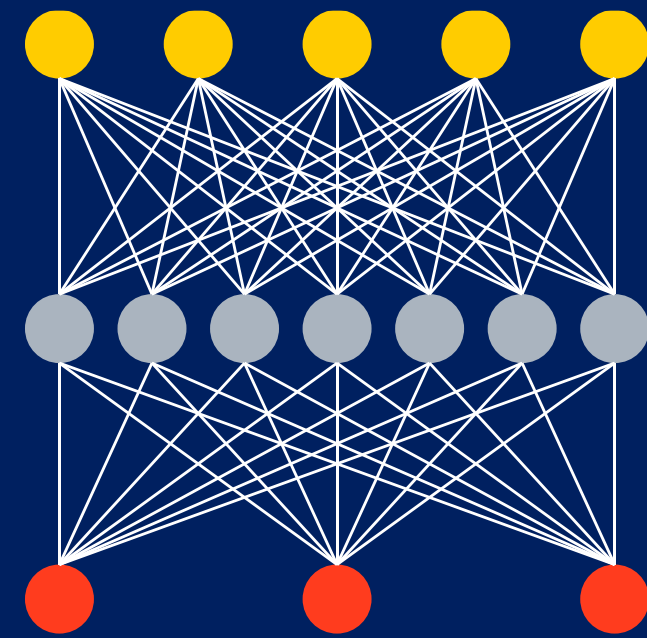


Sound

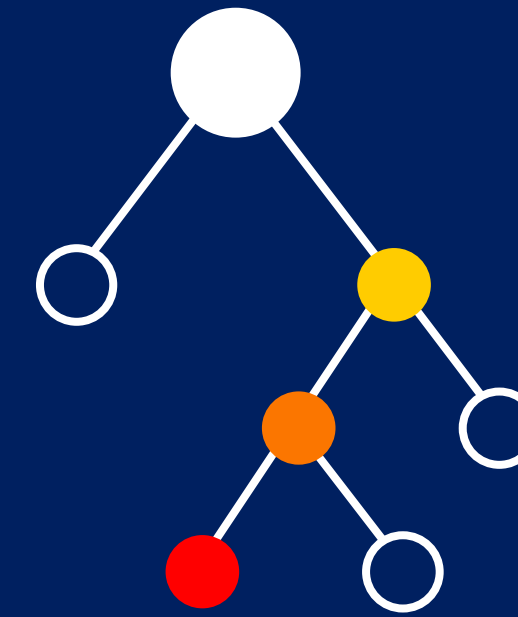


Text

NN + GB

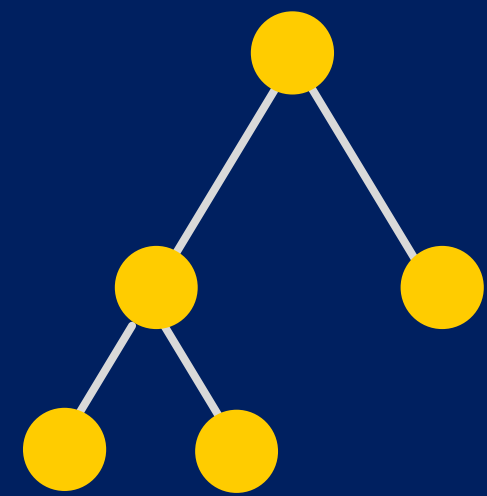


Neural networks

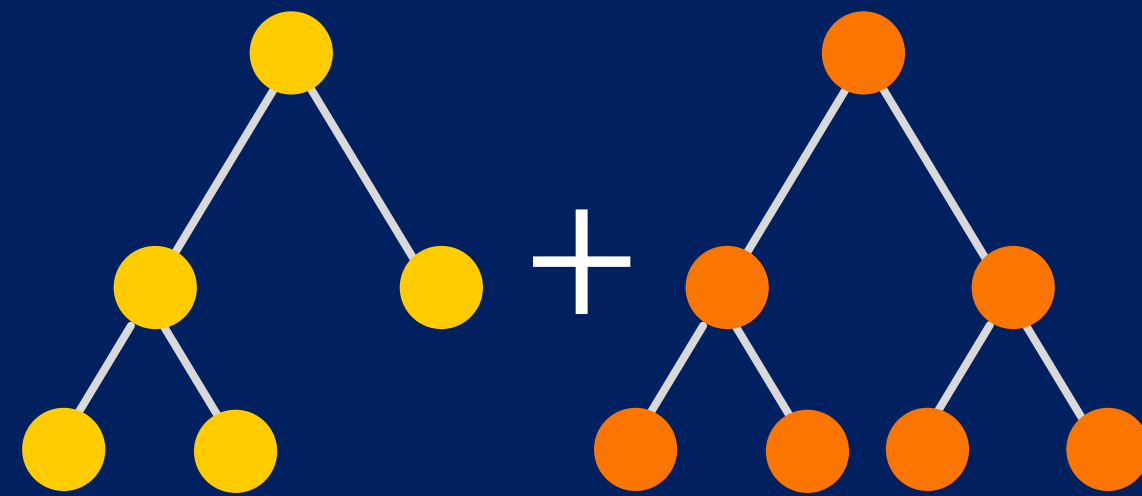


Gradient boosting

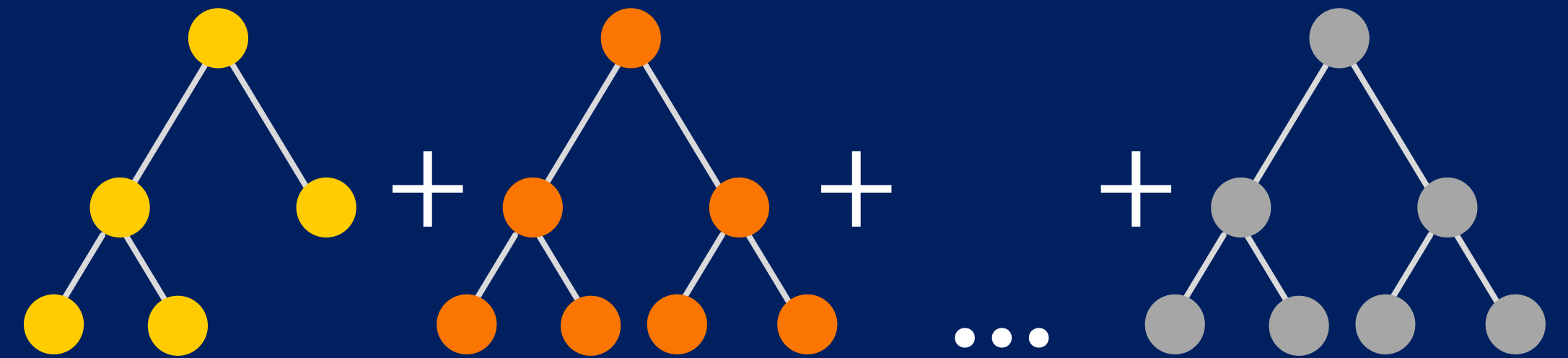
Gradient boosting



Loss



Loss



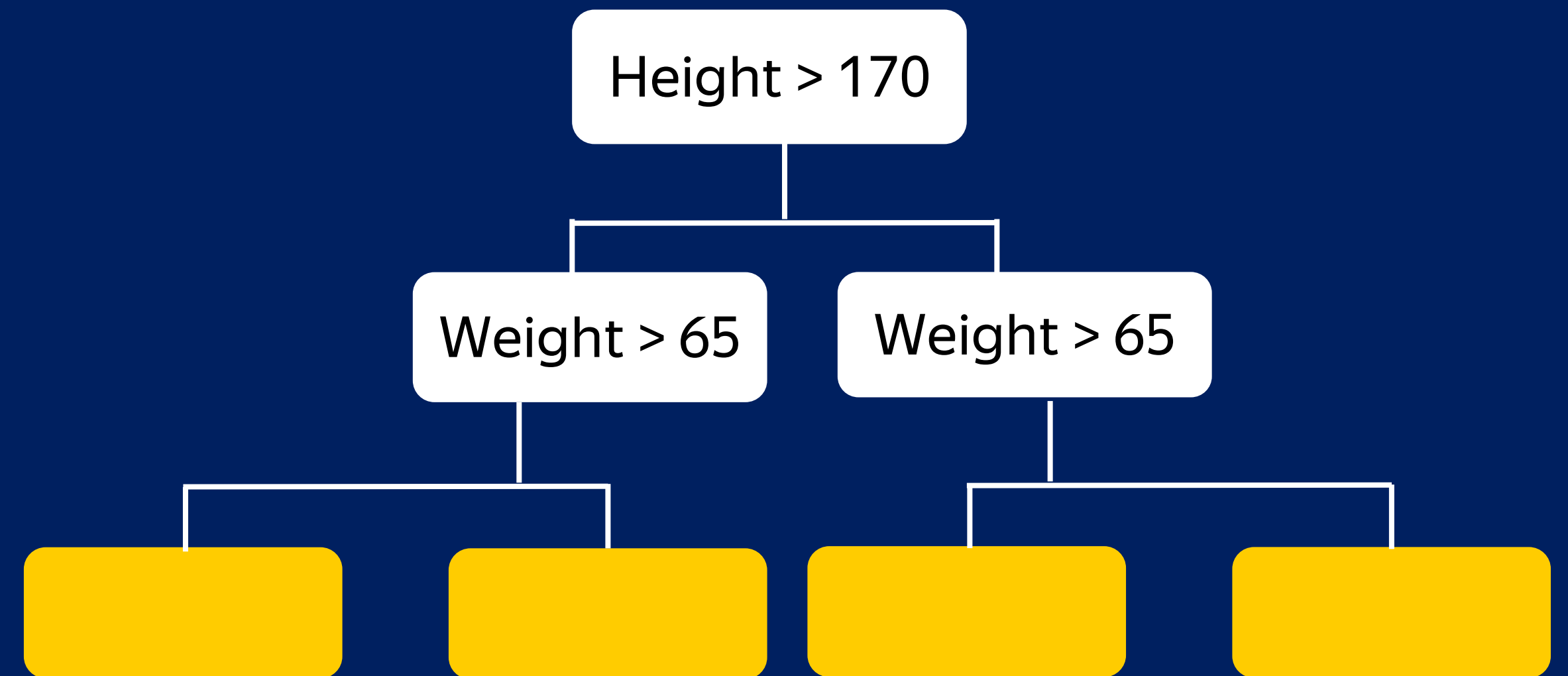
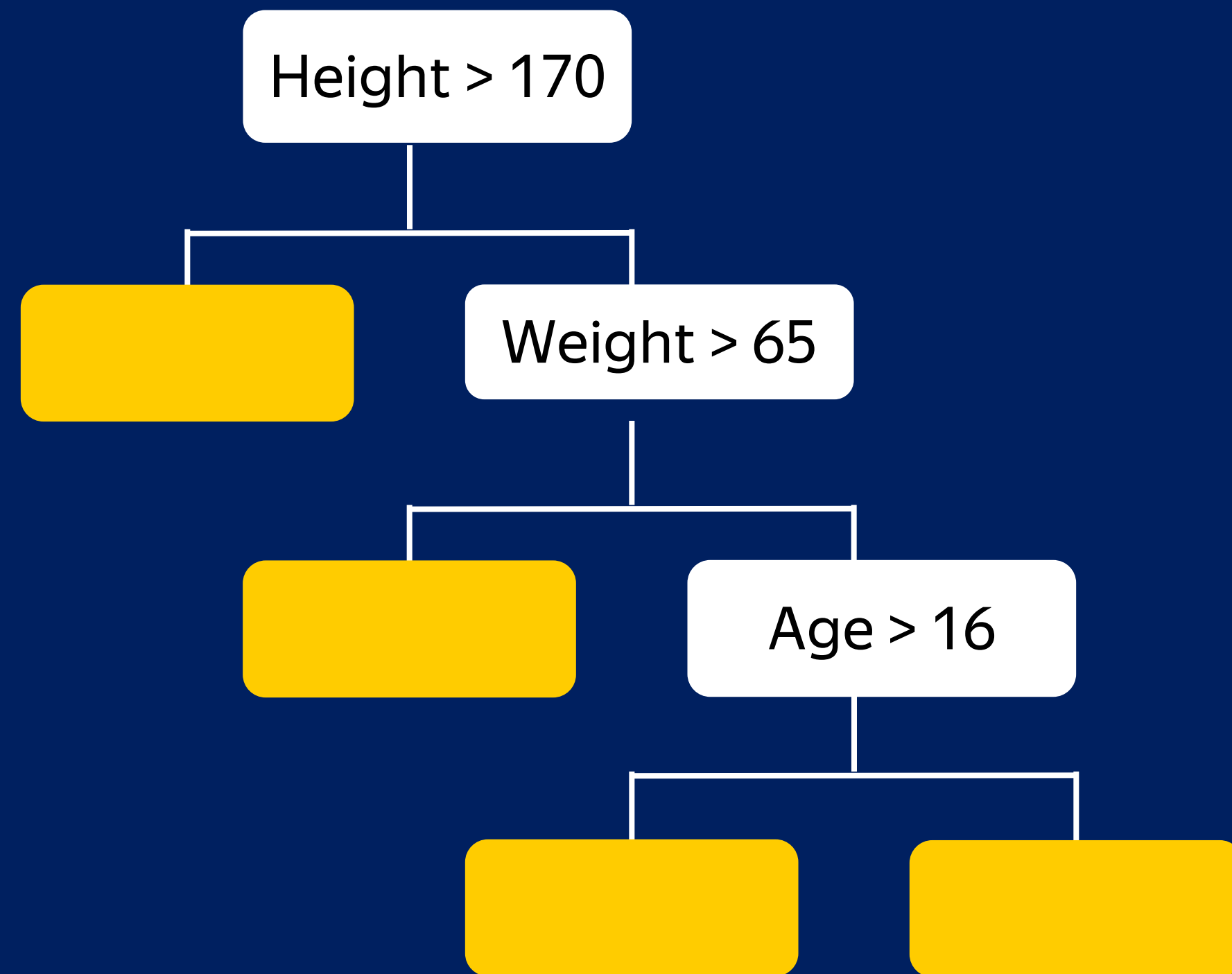
Loss

Algorithm comparison

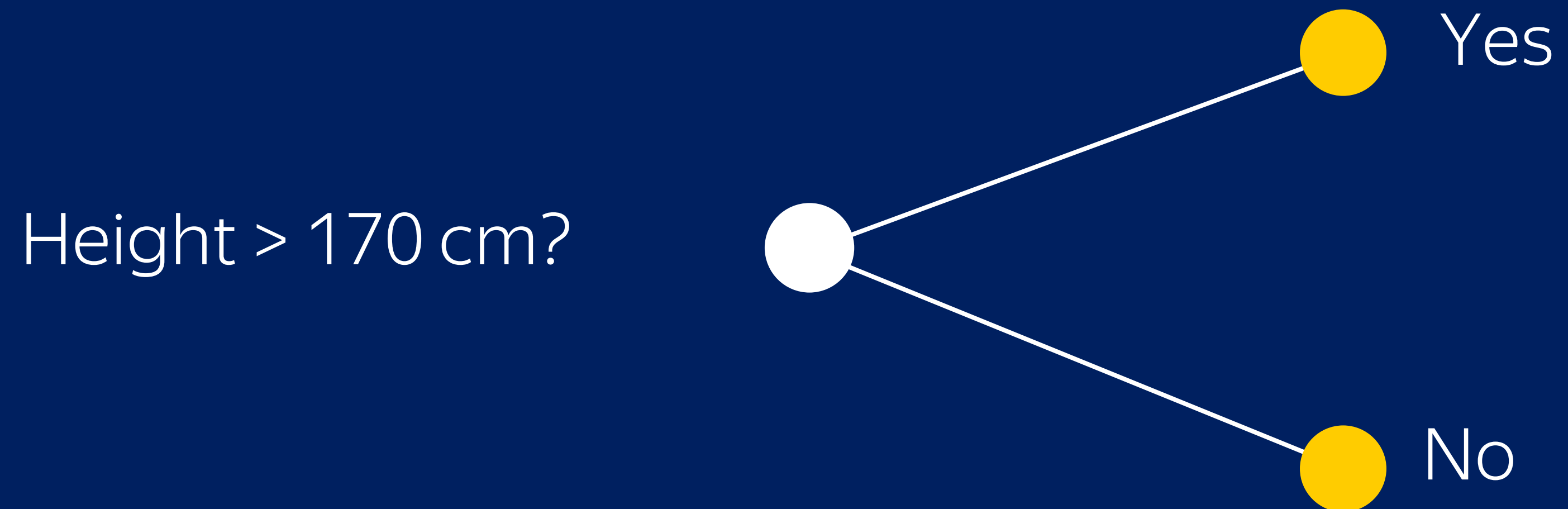
	CatBoost	LightGBM		XGBoost		H2O	
Adult	0.269741	0.276018	+ 2.33 %	0.275423	+ 2.11%	0.275104	+ 1.99%
Amazon	0.137720	0.163600	+ 18.79 %	0.163271	+ 18.55%	0.162641	+ 18.09%
Appet	0.071511	0.071795	+ 0.40 %	0.071760	+ 0.35%	0.072457	+ 1.32%
Click	0.390902	0.396328	+ 1.39 %	0.396242	+ 1.37%	0.397595	+ 1.71%
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

Symmetric trees



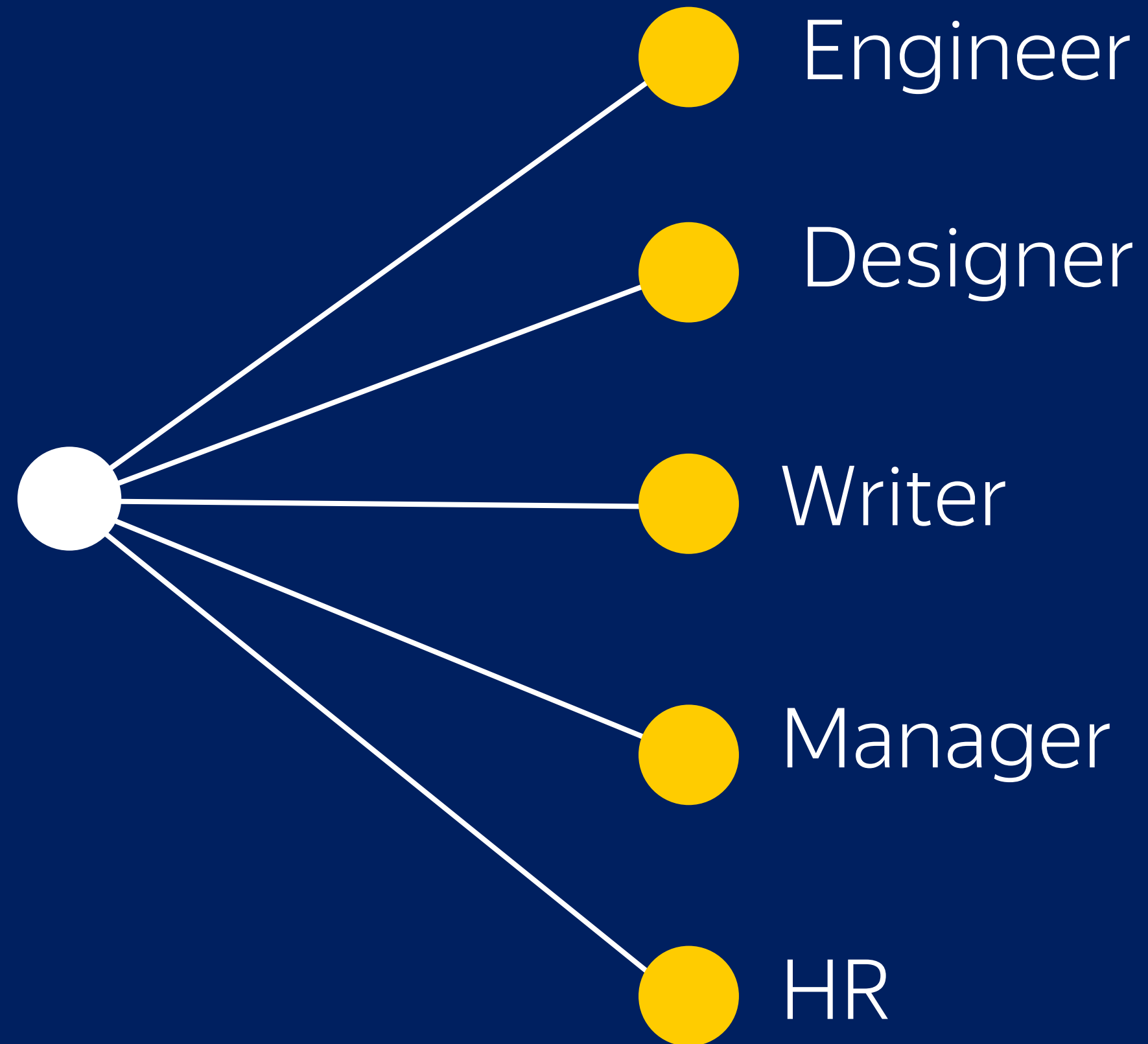
Numerical features



Categorical features

Categorical data

Occupation



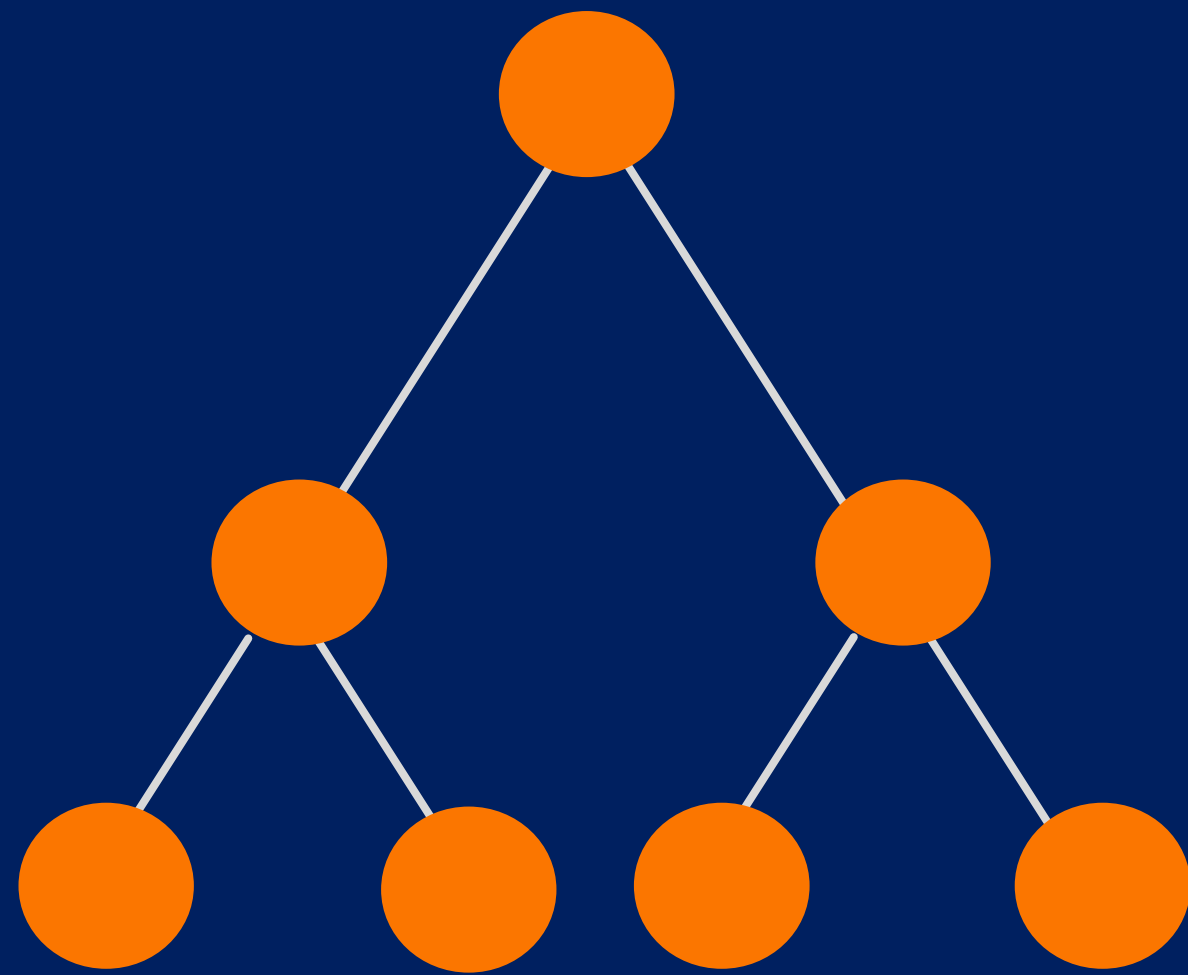
Categorical features support

- › One-hot encoding
- › Statistics based on category and category plus label value
- › Usage of several permutations
- › Greedy constructed feature combinations

i	[SDE		1
			SDE		1
			SDE		0
			PR		
			SDE		1
			PR		

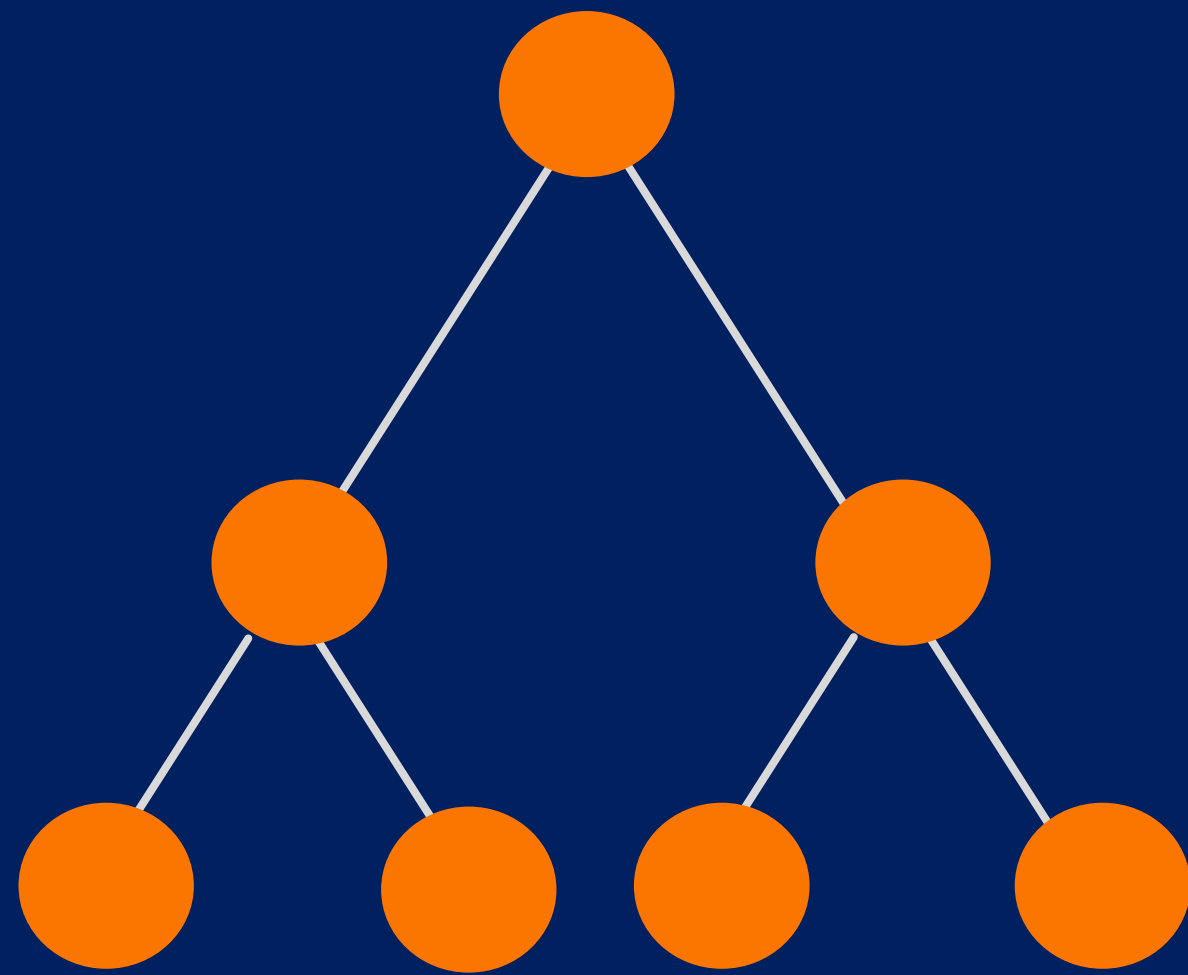
$$i \longrightarrow \frac{1 + 1 + 0 + a * \text{Prior}}{3 + a}$$

Classical boosting



$$\text{leafValue} = \sum_{i=1}^n \frac{g(\text{approx}(i), \text{target}(i))}{n}$$

Ordered boosting



$$\text{leafValue}(\text{doc}) = \sum_{i=1}^{\text{doc}} \frac{g(\text{approx}(i), \text{target}(i))}{\text{docs in the past}}$$

Modes

- › Classification
- › Regression
- › Ranking

Classification

- › Predict if the person will pay the credit
- › Which type of clouds will be present tomorrow

Regression

- › Predict the taxi drive duration
- › Predict dollar exchange rate

Ranking

- › What are top N hotels in Trento?
- › Input data: ratings

Predicting rating is not necessary!

- › Ranking within a group
- › Ranking modes:
 - Ranking (YetiRank, YetiRankPairwise)
 - Pairwise (PairLogit, PairLogitPairwise)
 - Ranking + Classification (QueryCrossEntropy)
 - Ranking + Regression (QueryRMSE)
 - Select top 1 candidate (QuerySoftMax)

Speed

- › CPU training
- › GPU training
- › Prediction speed

CPU: Comparison with other libraries

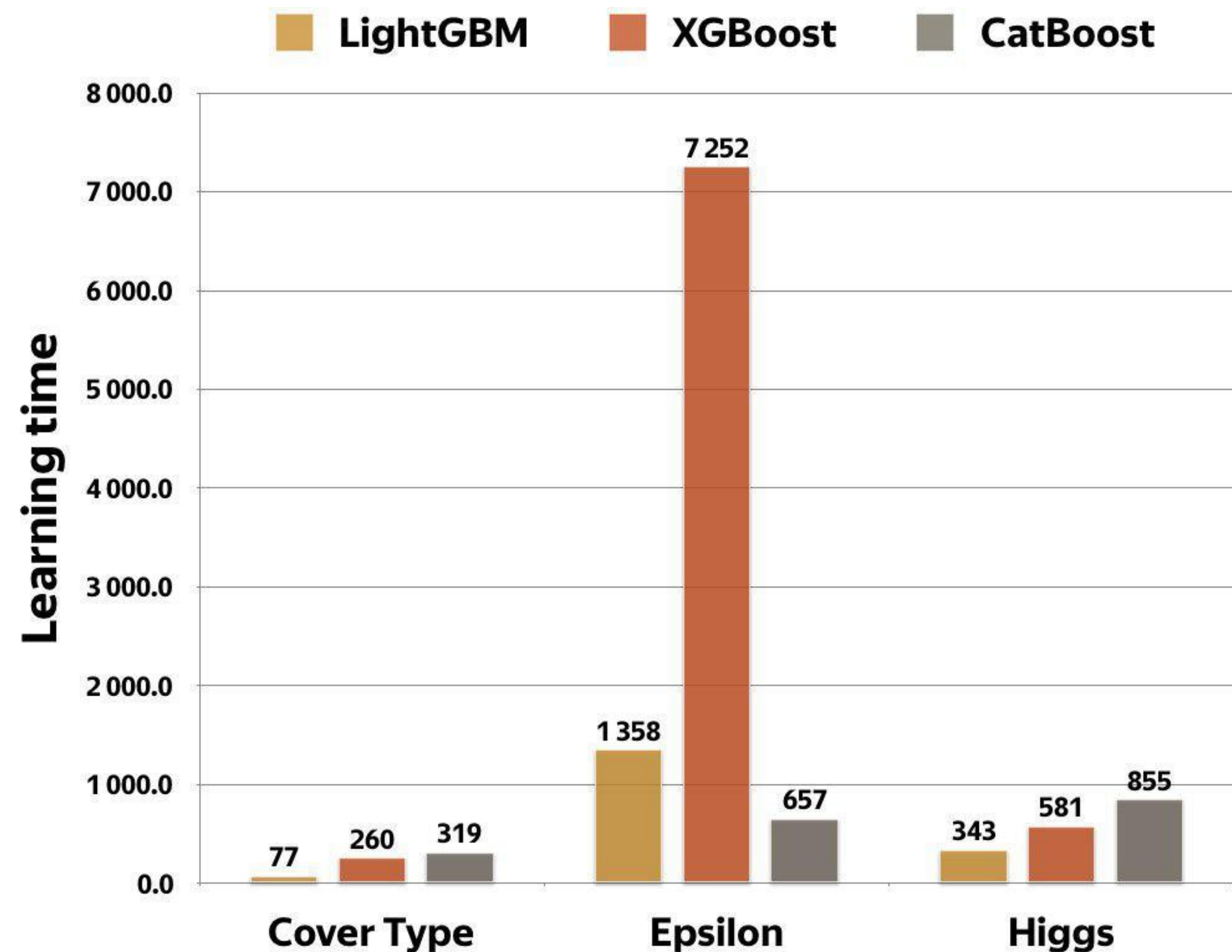
- Parameters:
128 bins, 64 leafs, 1000 iterations

- Cover type:
54 features, 522910 samples

- Epsilon:
2000 features, 400k samples

- Higgs:
28 features, 11M samples

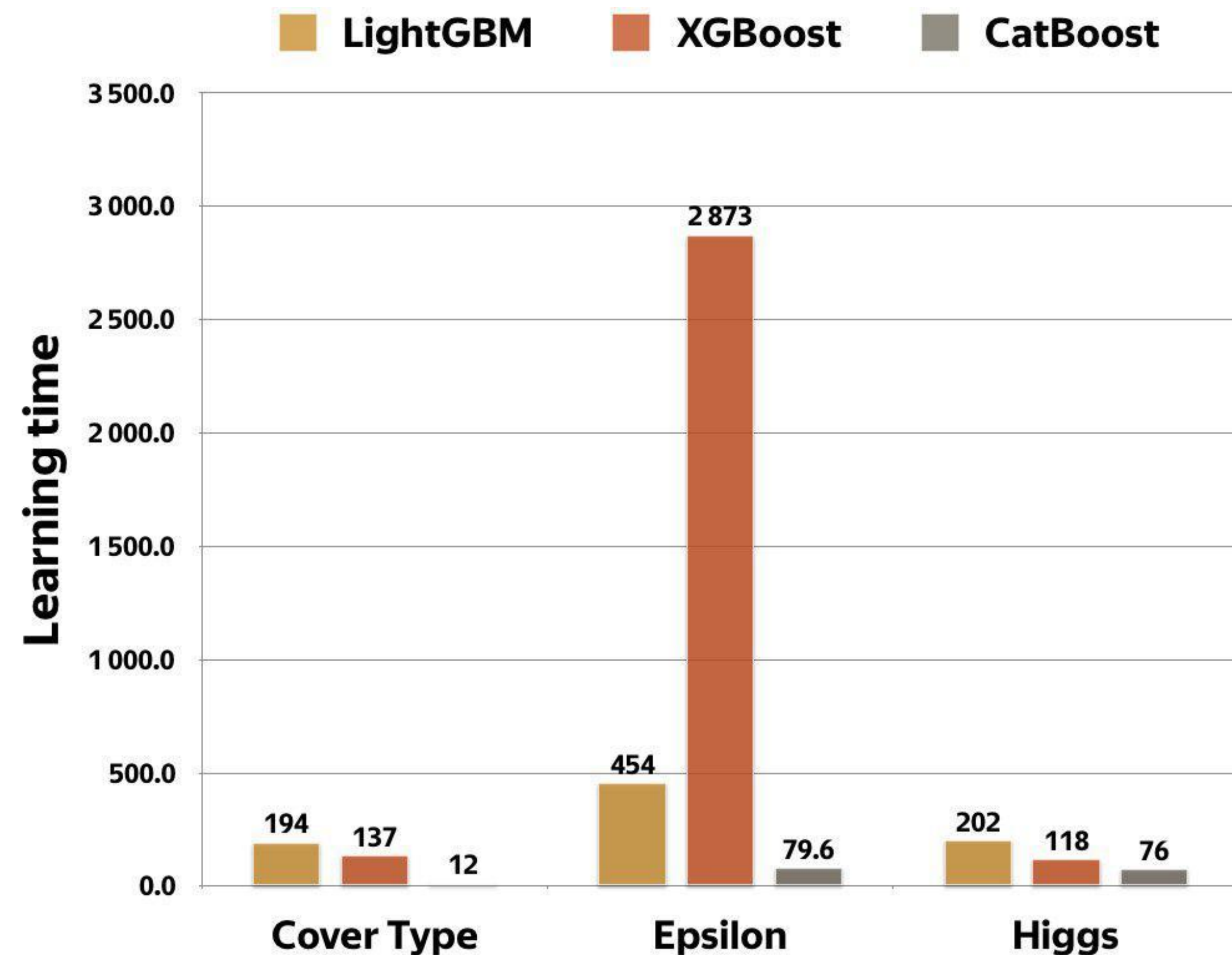
Intel(R) Core(TM) i7-6800K CPU @
3.40GHz



GPU: Comparison with other libraries

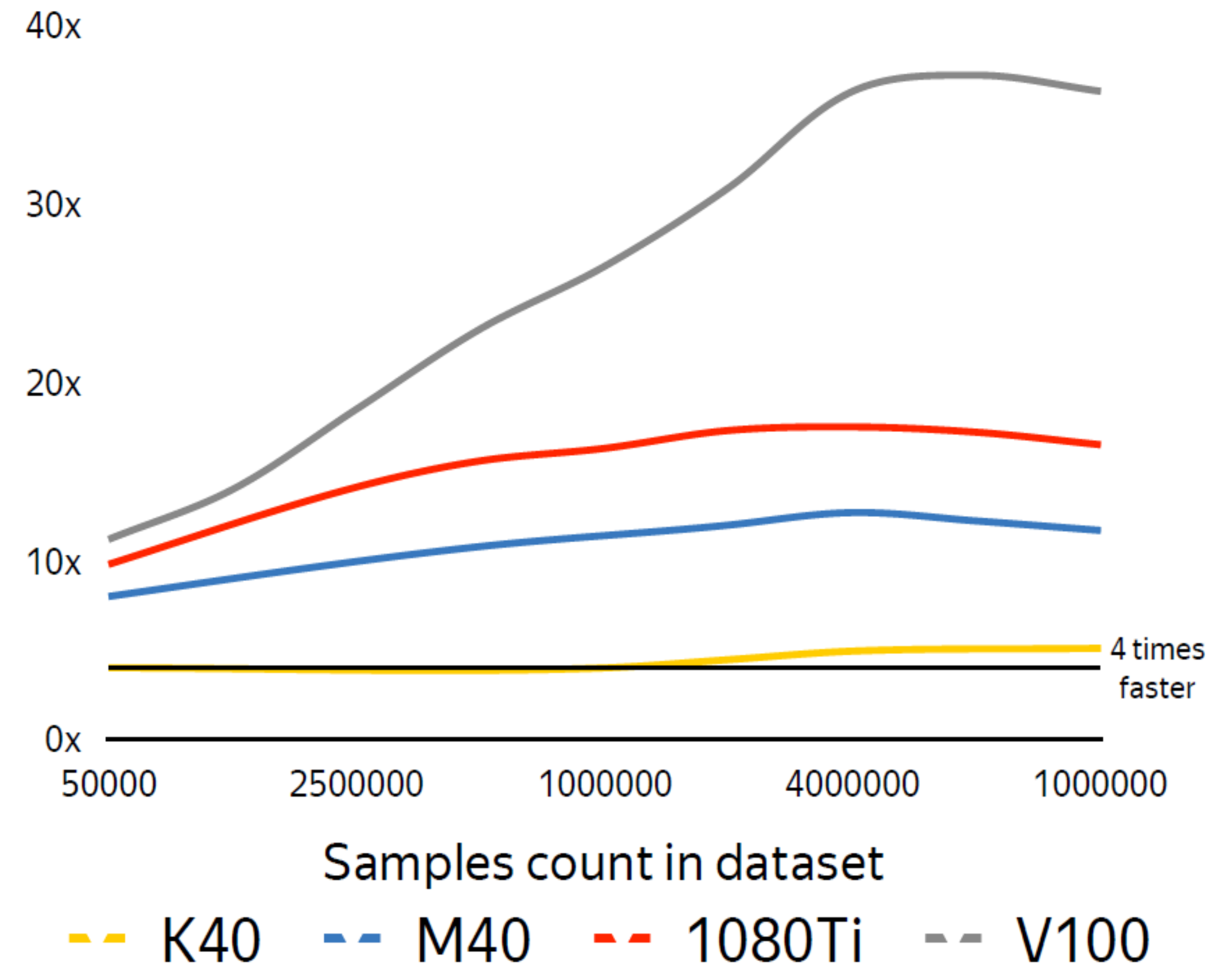
- Parameters:
128 bins, 64 leafs, 1000 iterations
- Cover type:
54 features, 522910 samples
- Epsilon:
2000 features, 400k samples
- Higgs:
28 features, 11M samples

GTX1080Ti

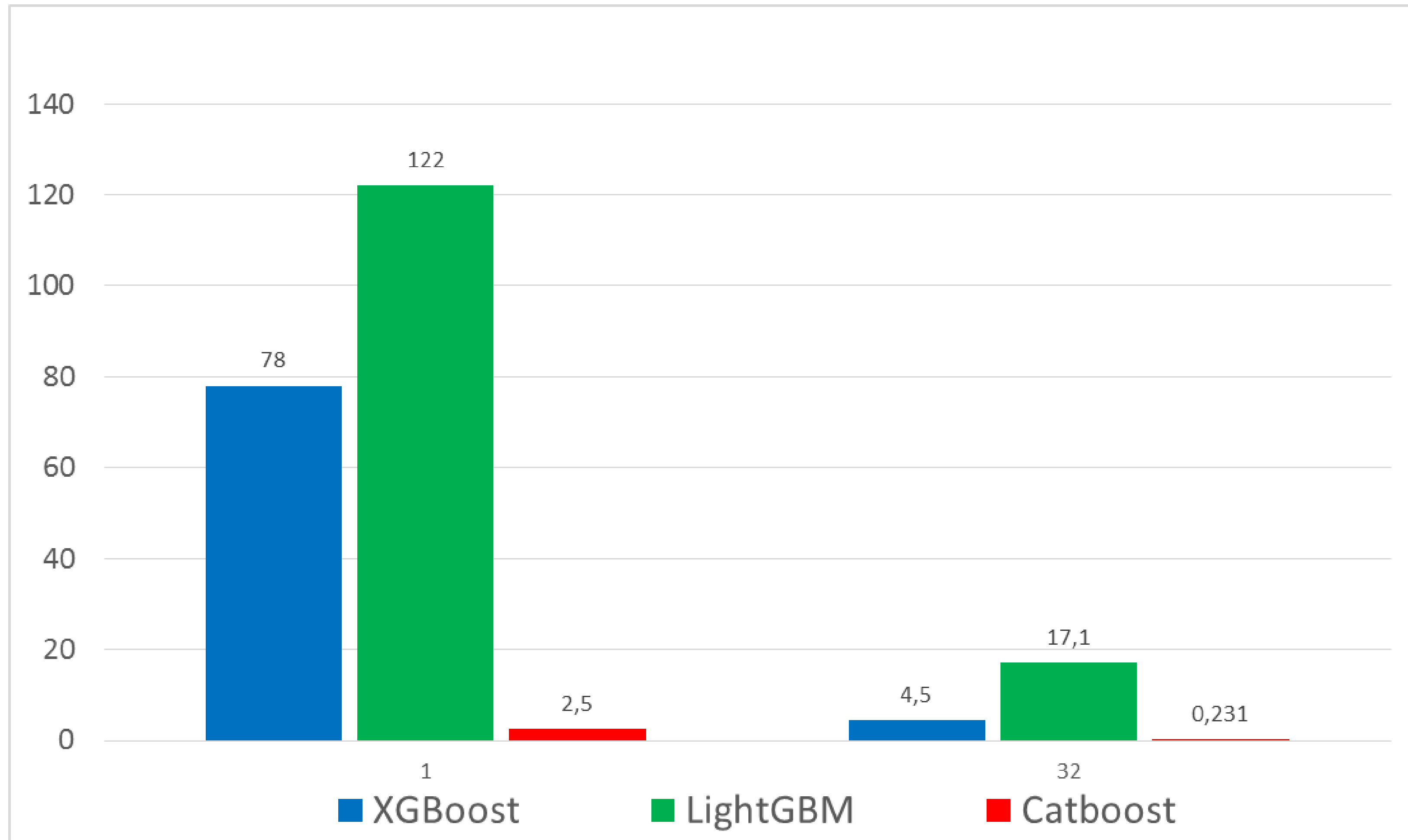


CPU vs GPU

- Dual-Socket Intel Xeon E5-2660v4 as baseline
- Several modern GPU as competitors
- Dataset: 800 features



Prediction time



Ways to explore your data

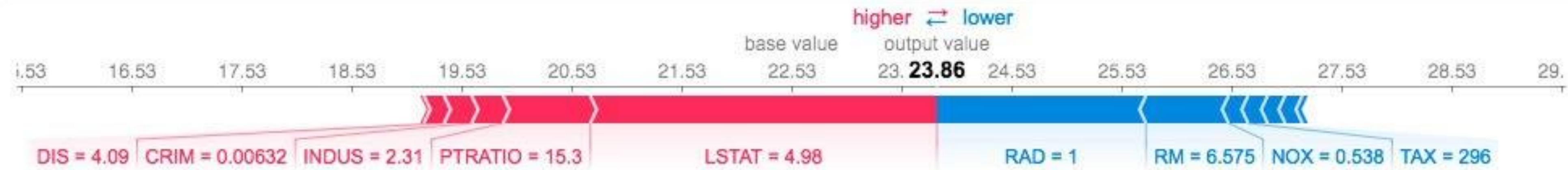
- › Feature importance
- › Feature interaction
- › Per object feature importance (SHAP)

SHAP values

```
In [46]: shap_values = model.get_feature_importance(data=Pool(X,y), fstr_type='ShapValues')
```

```
# visualize the first prediction's explanation  
shap.force_plot(shap_values[0,:], X.iloc[0,:])
```

Out[46]:



Ways to explore your data

- › Feature importance
- › Feature interaction
- › Per object feature importance (SHAP)
- › Influential documents
- › New features evaluation

Useful features

- › Metric evaluation during training

CatBoost Viewer

```
In [11]: model.fit(  
    X_train, y_train,  
    cat_features=categorical_features_indices,  
    eval_set=(X_validation, y_validation),  
    # verbose=True, # you can uncomment this for text output  
    plot=True  
)
```

× ☒ --- Learn ☒ — Test

Logloss Accuracy

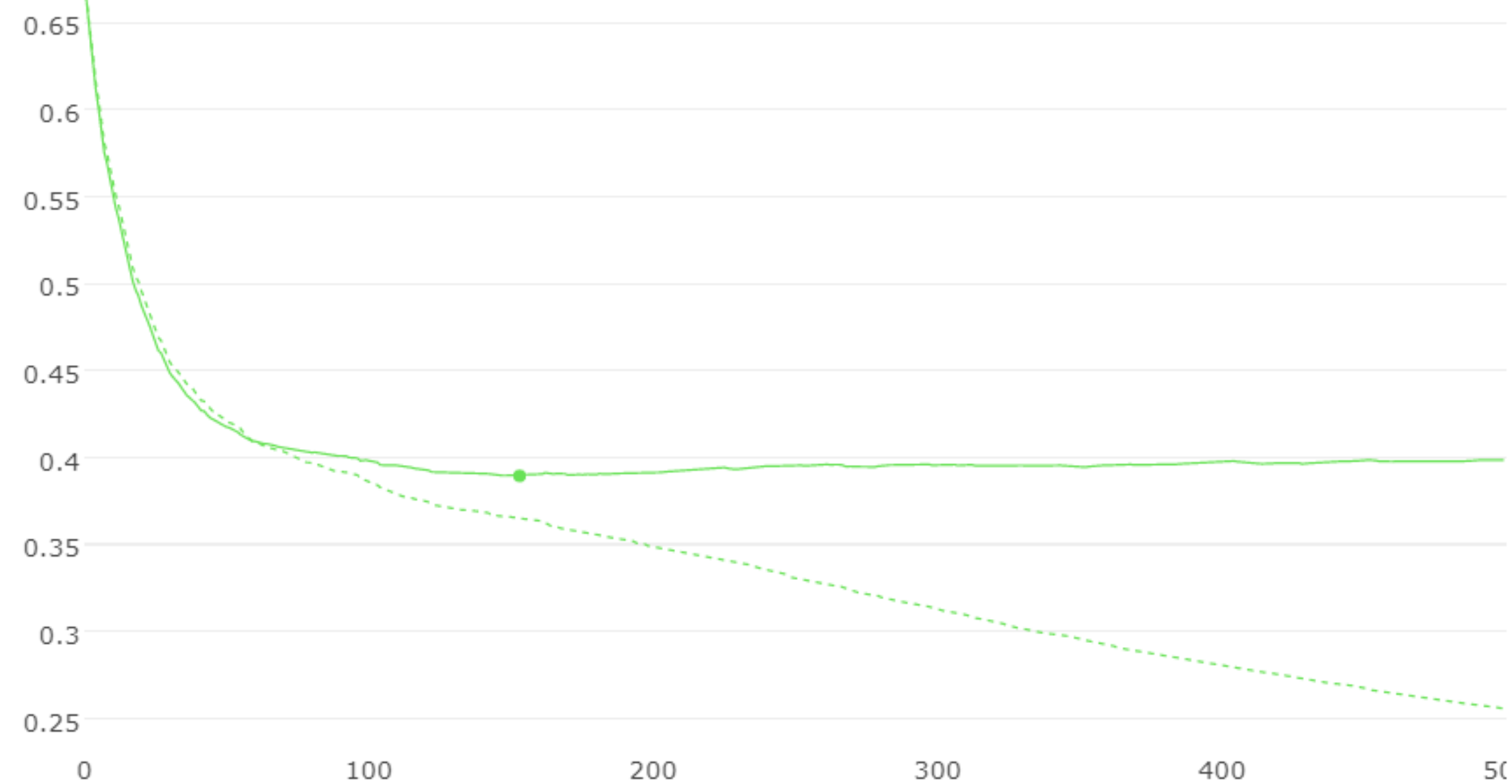
☒ current 19s
curr --- 0.2555652... — 0.3988282... 498
best — 0.3894004... 153

☐ Click Mode

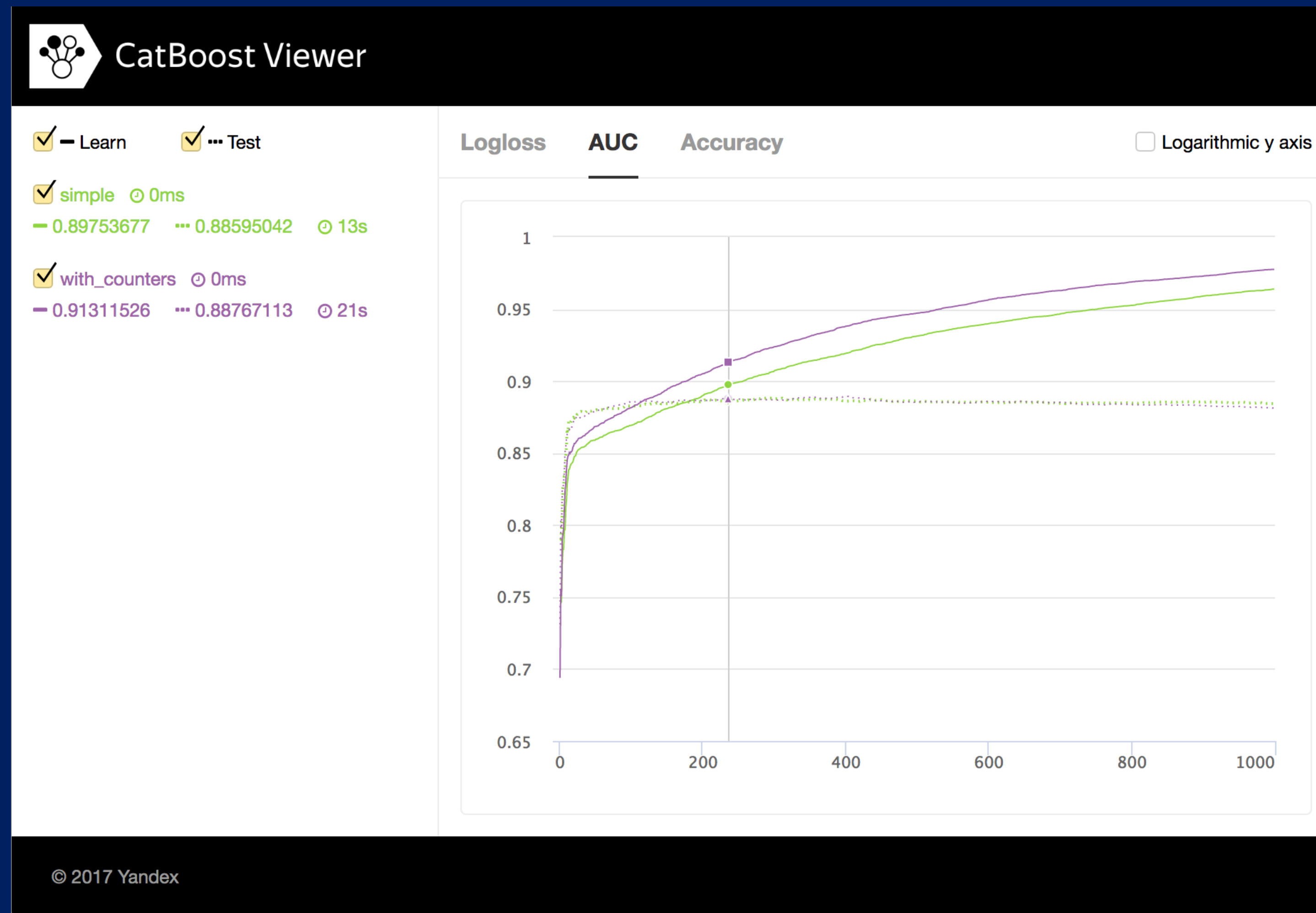
☐ Logarithm

☐ Smooth

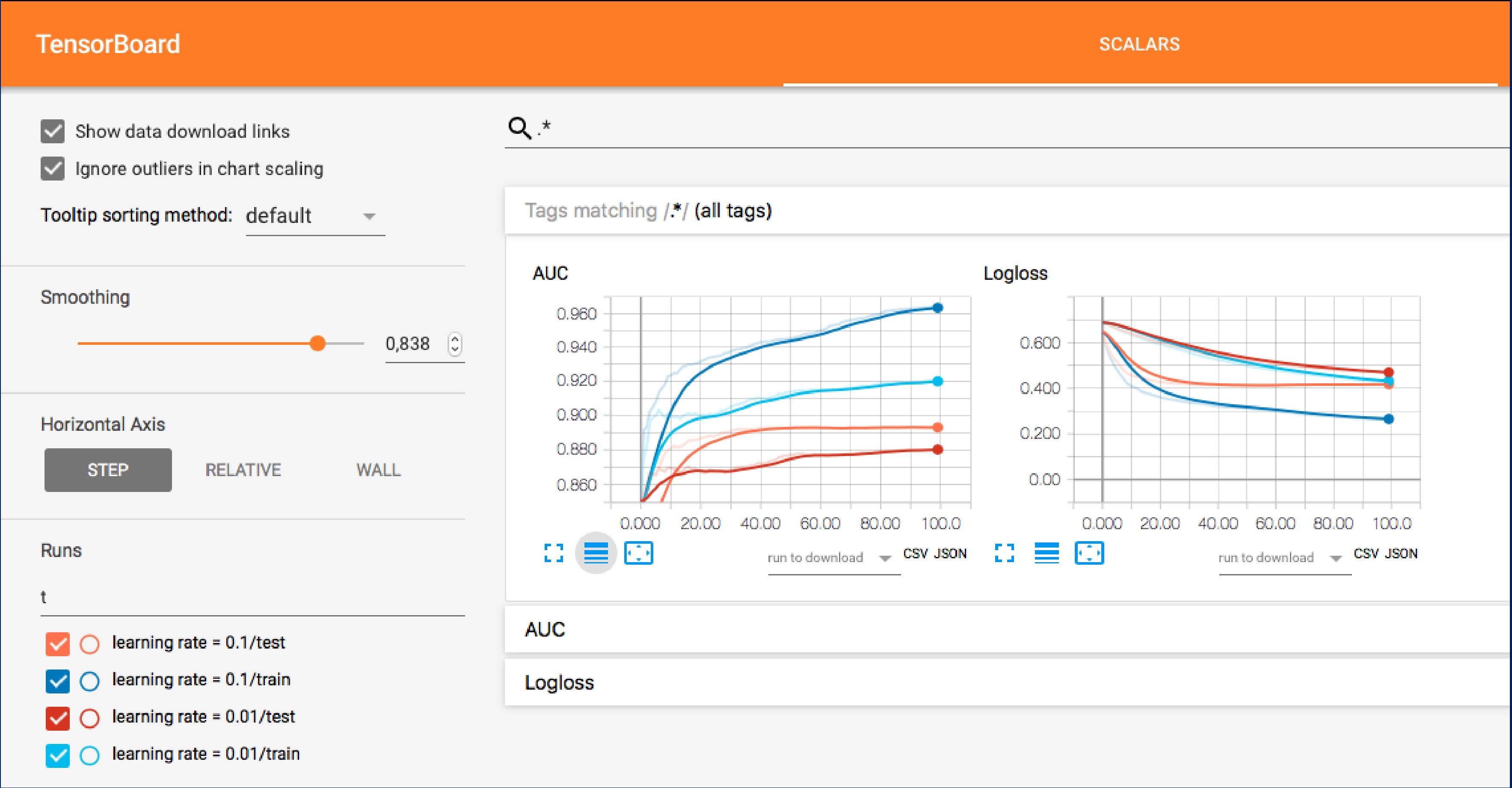
0.5



CatBoost Viewer



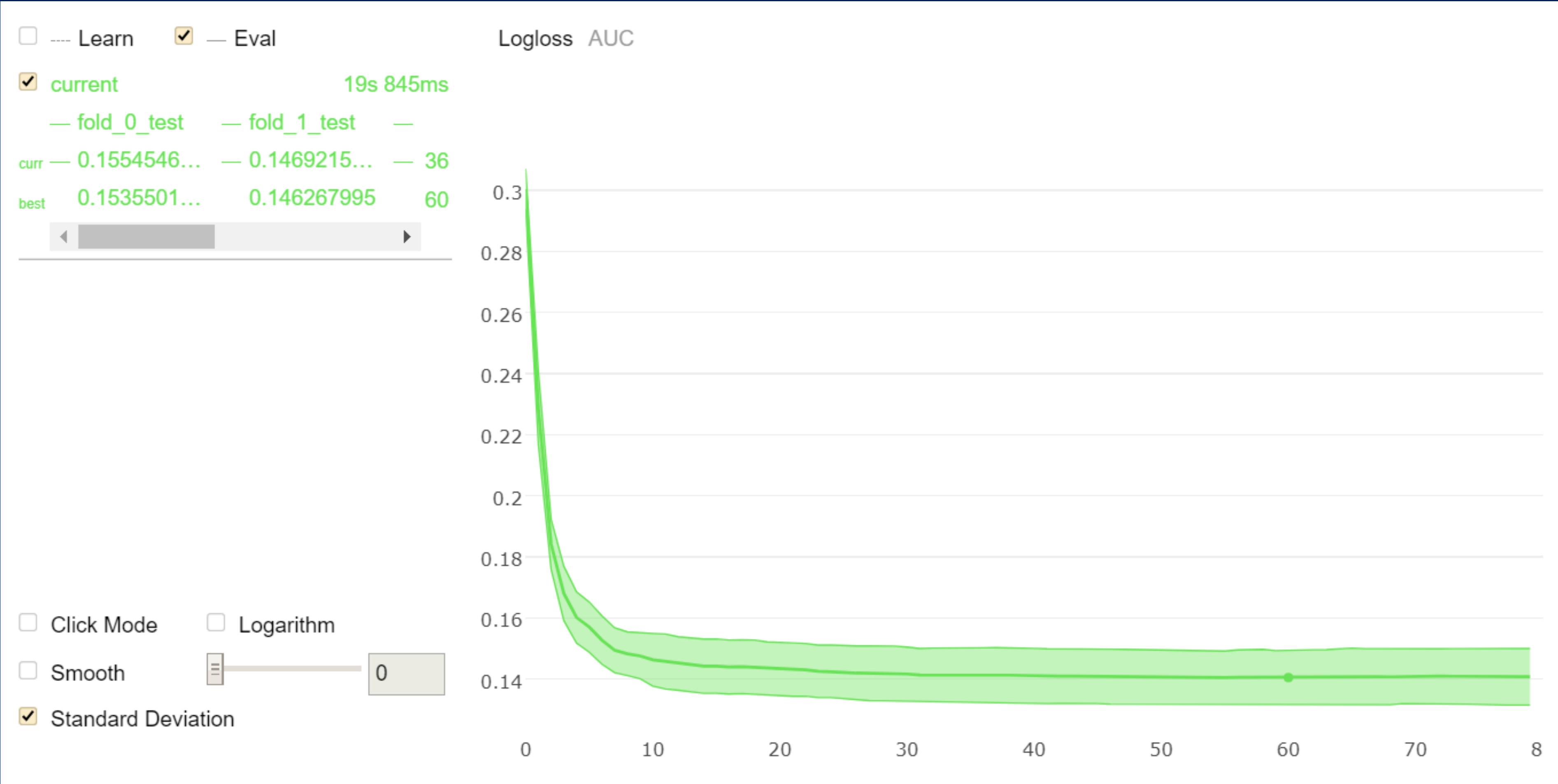
TensorBoard



Useful features

- › Metric evaluation during training
- › Missing values support
- › Cross-validation

Cross-validation



Useful features

- › Metric evaluation during training
- › Missing values support
- › Cross-validation
- › `staged_predict` + metric evaluation on dataset

Reading

- › http://learningsys.org/nips17/assets/papers/paper_11.pdf
- › <https://arxiv.org/abs/1706.09516>
- › <https://github.com/catboost/tutorials>



<http://catboost.ai>

Questions?

Anna Veronika Dorogush

Lead of CatBoost team