

Yandex



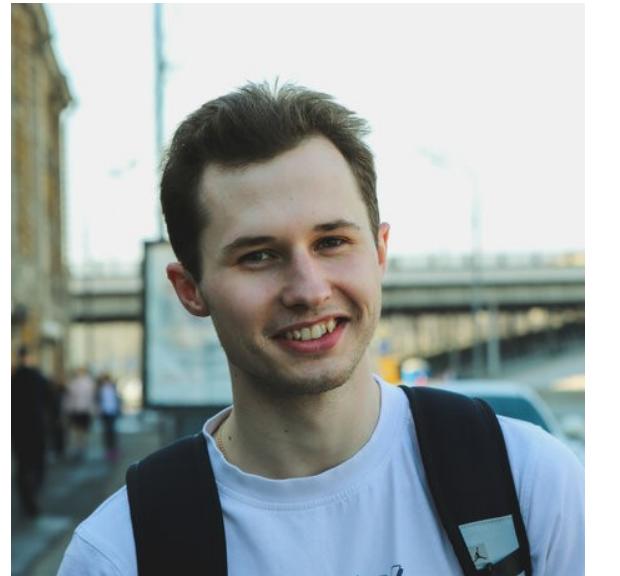
# CatBoost: Introduction lecture

Kirillov Stanislav,  
Head of ML Systems group @ Yandex

# Plan

- > Introduction
- > Main features of CatBoost
- > GPU support: single host and distributed training
- > CatBoost for Apache Spark
- > Uncertainty estimation
- > Further plans

# CatBoost core developers



Nikita  
Dmitriev



Ekaterina  
Ermishkina



Andrew  
Khropov



Stanislav  
Kirillov



Ivan  
Lyzhin

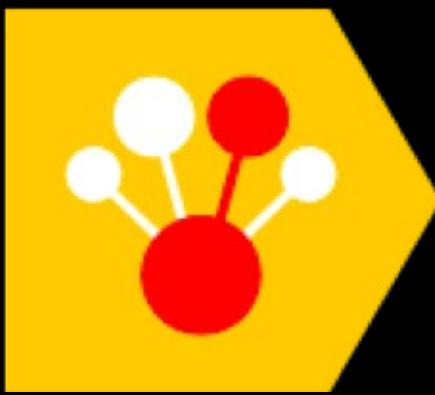


Dmitry  
Oganesyan



Eugeny  
Petrov

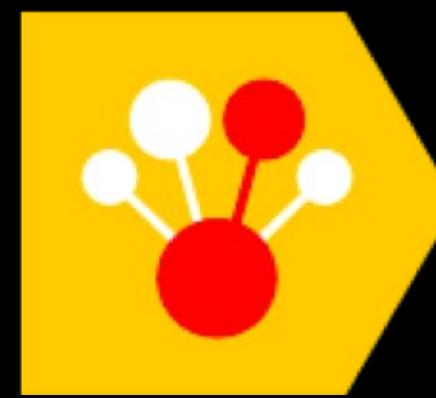
# What is CatBoost?



Yandex  
CatBoost

- › CatBoost is a machine learning algorithm that uses gradient boosting on decision trees (GBDT).
- › It is available as an open source library.

# Why CatBoost?

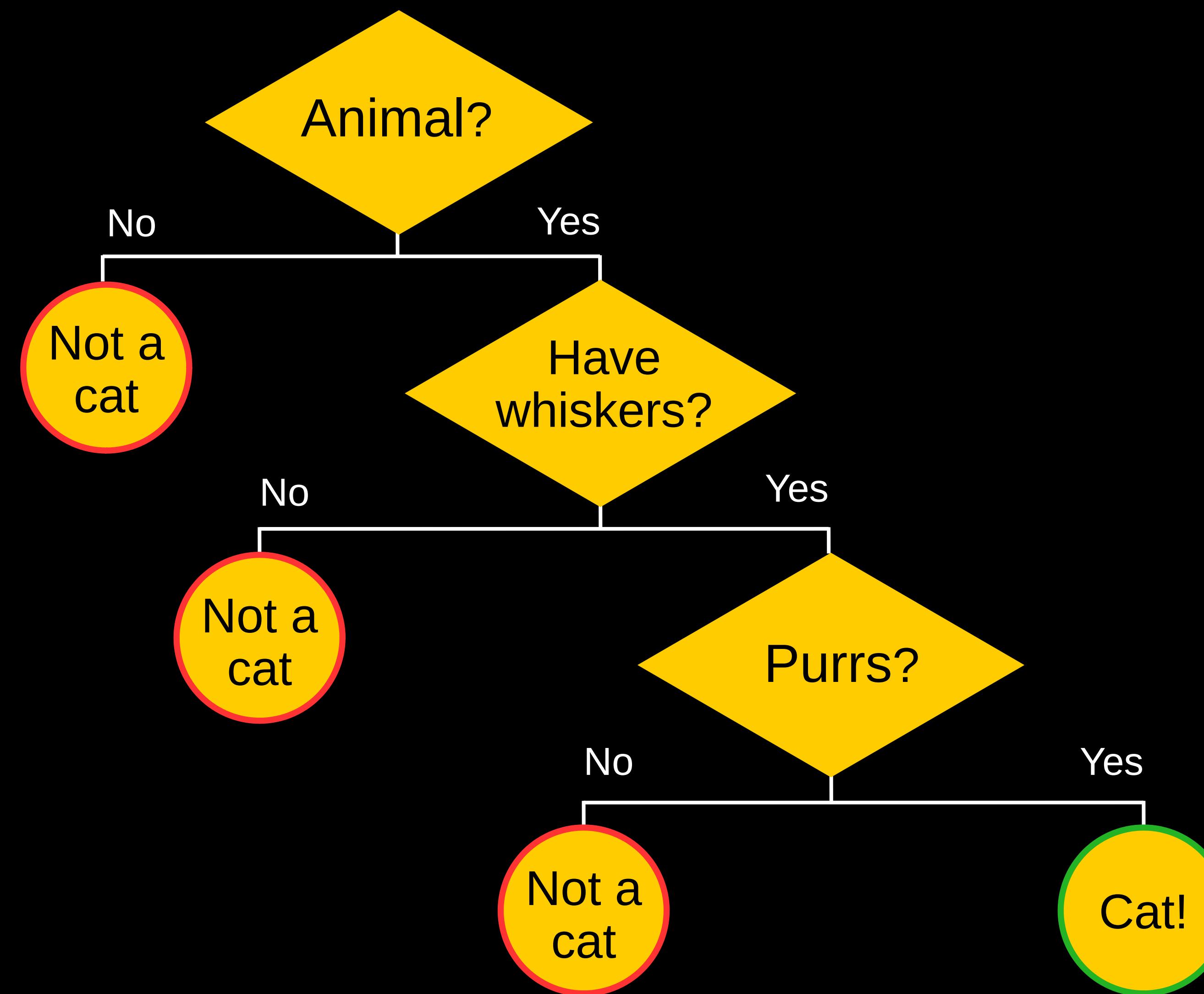


Yandex  
CatBoost

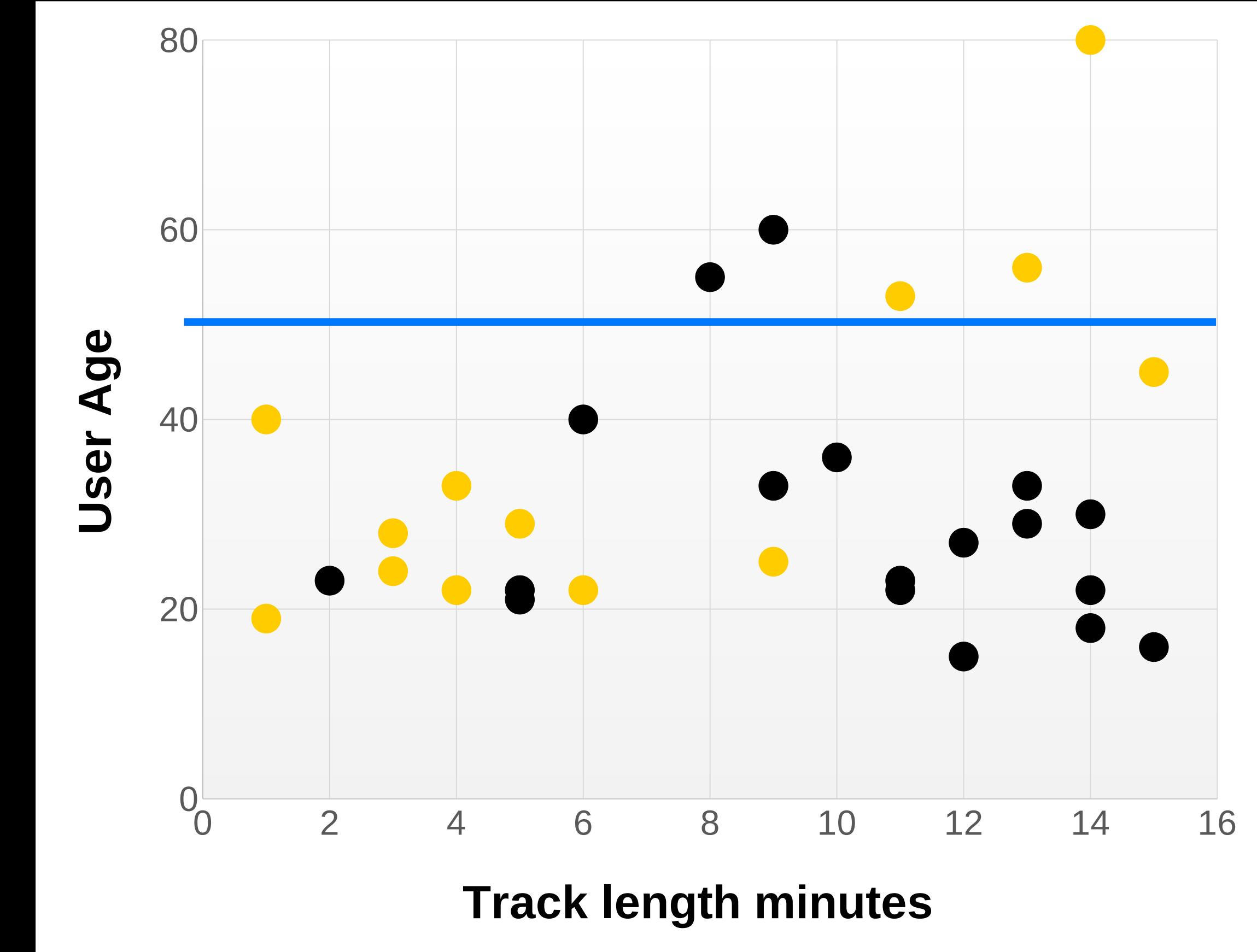
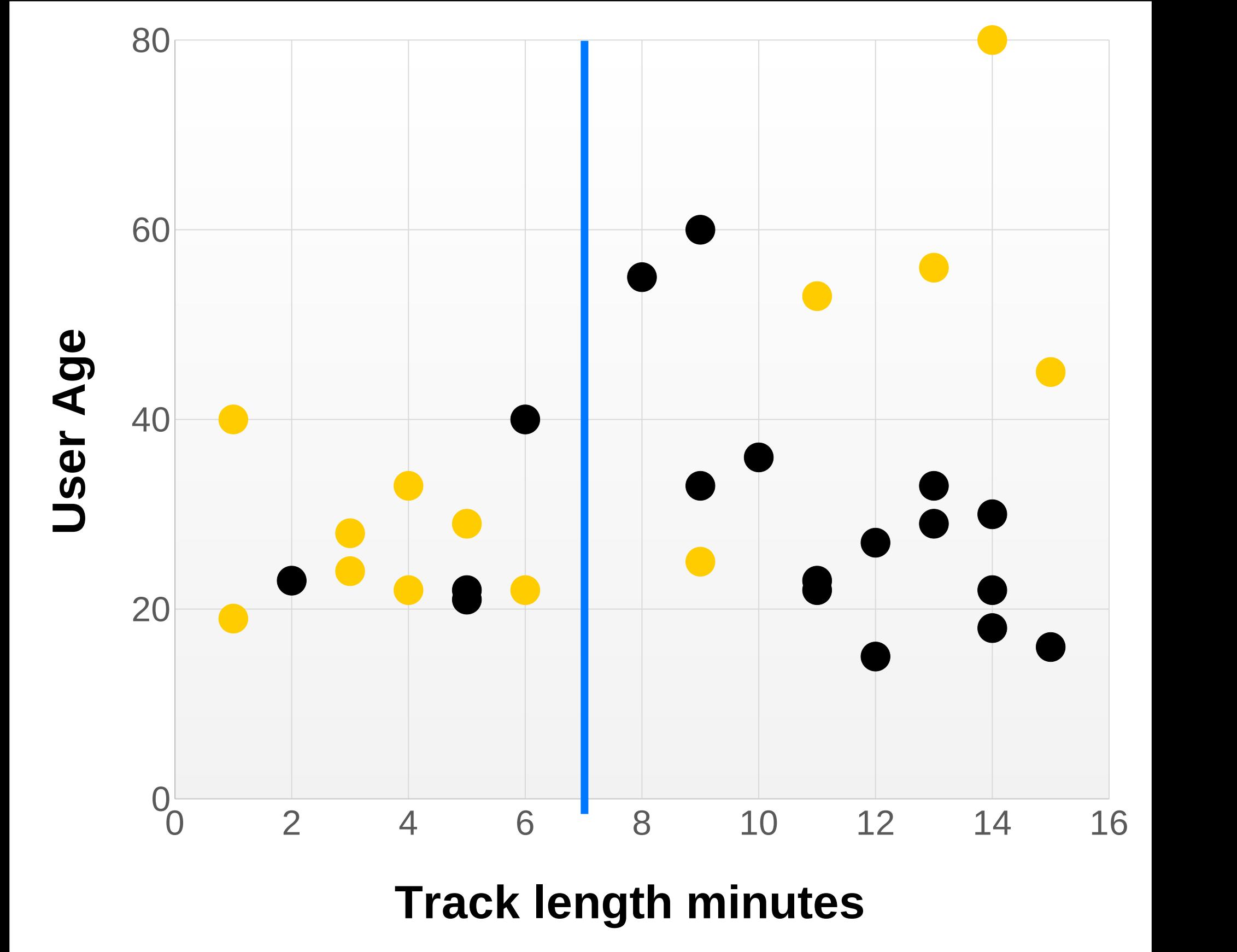
- › Sophisticated feature types support: categorical, text and embedding features
- › Good quality with default parameters
- › Multiple tree grow policies: symmetric, loss guide and depthwise
- › Fast applier
- › Model analysis tools
- › Novel uncertainty estimation support

# Main features & concepts

# Tabular data? Decision trees!

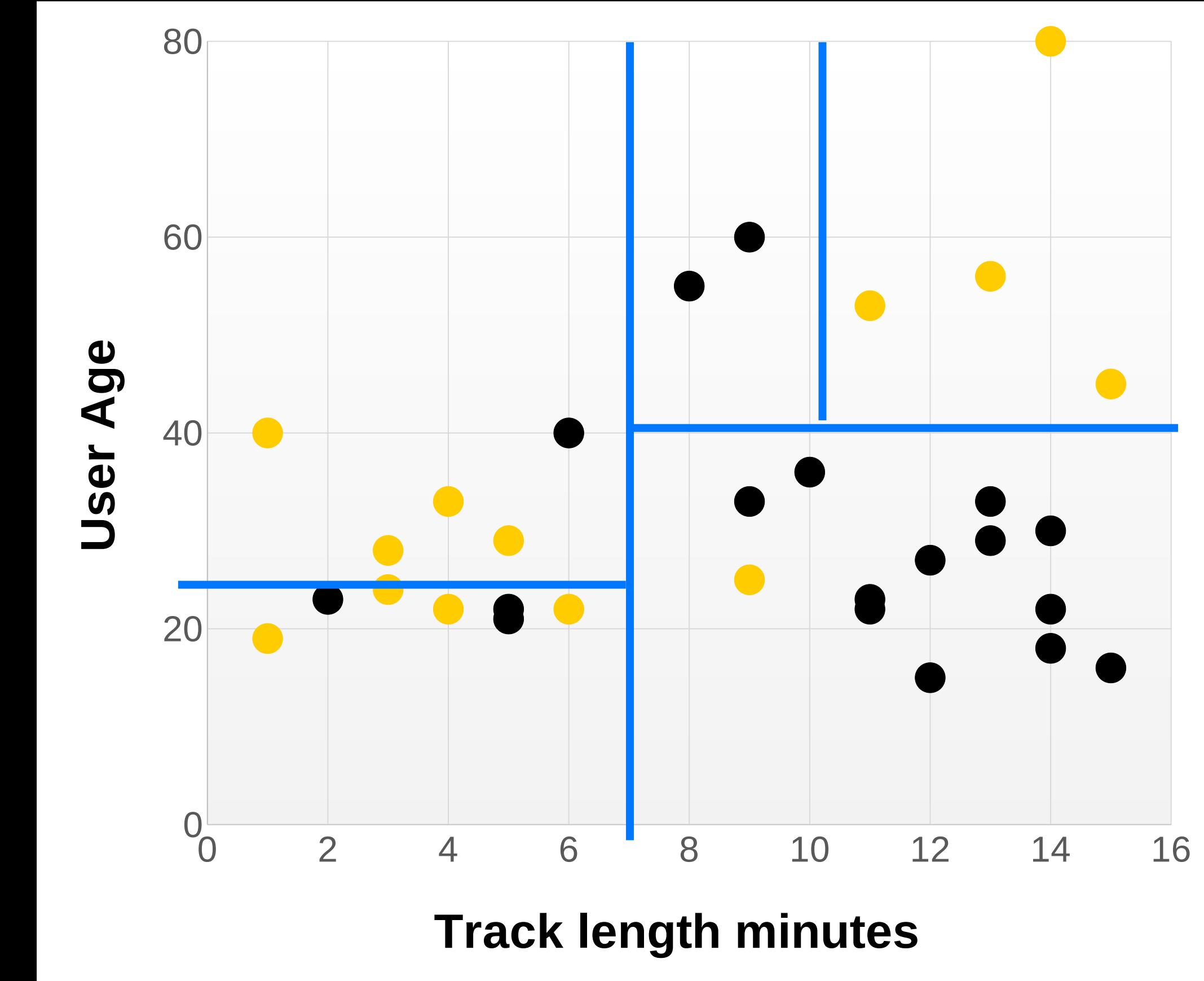
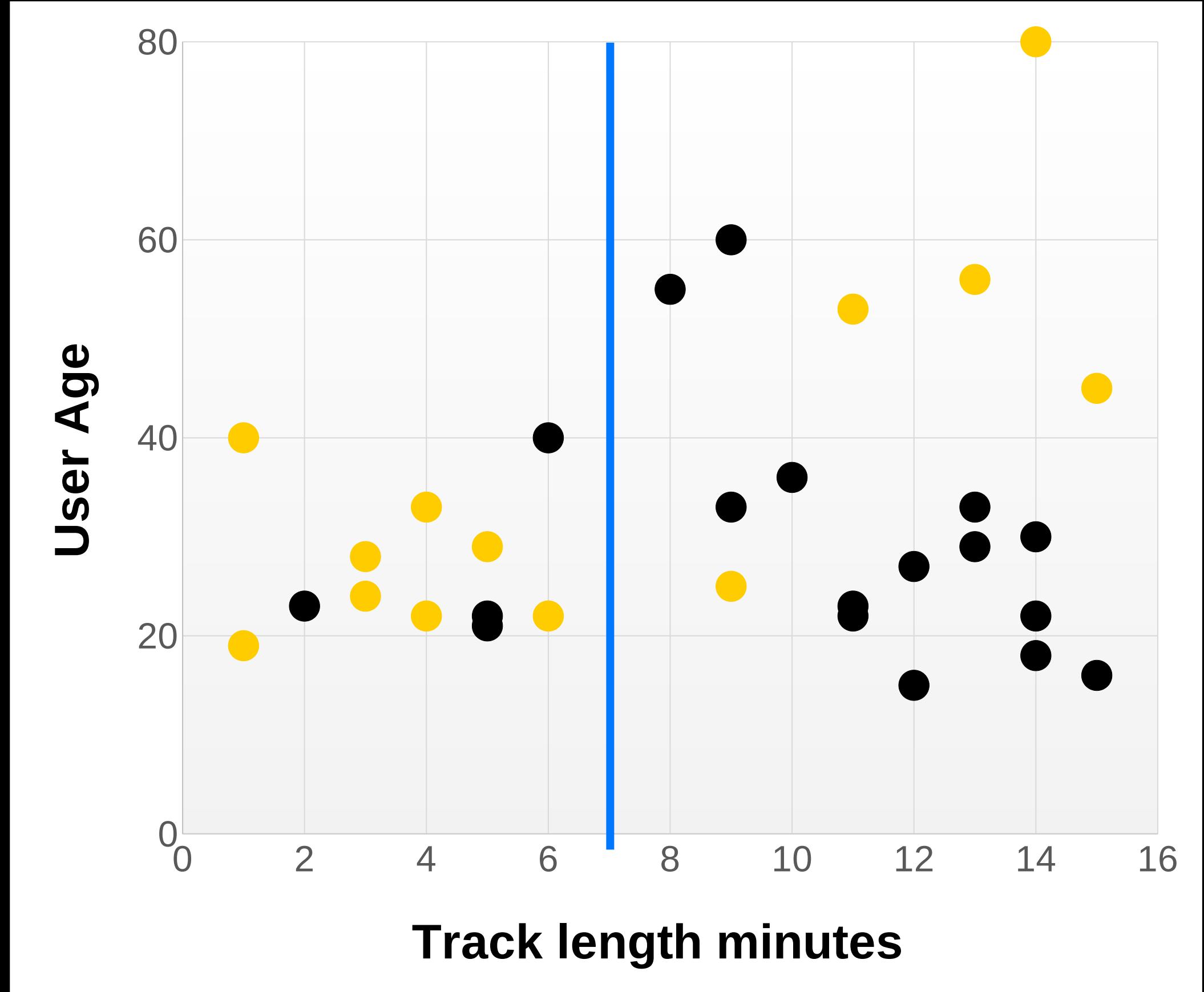


# Decision tree: classification



Best split: Track length minutes > 7

# Decision tree: classification



# CatBoost: score functions for best split search

- $L2 = - \sum_i w_i \cdot (a_i - g_i)^2$
- $Cosine = \frac{\sum w_i \cdot a_i \cdot g_i}{\sqrt{\sum w_i a_i^2} \cdot \sqrt{\sum w_i g_i^2}}$

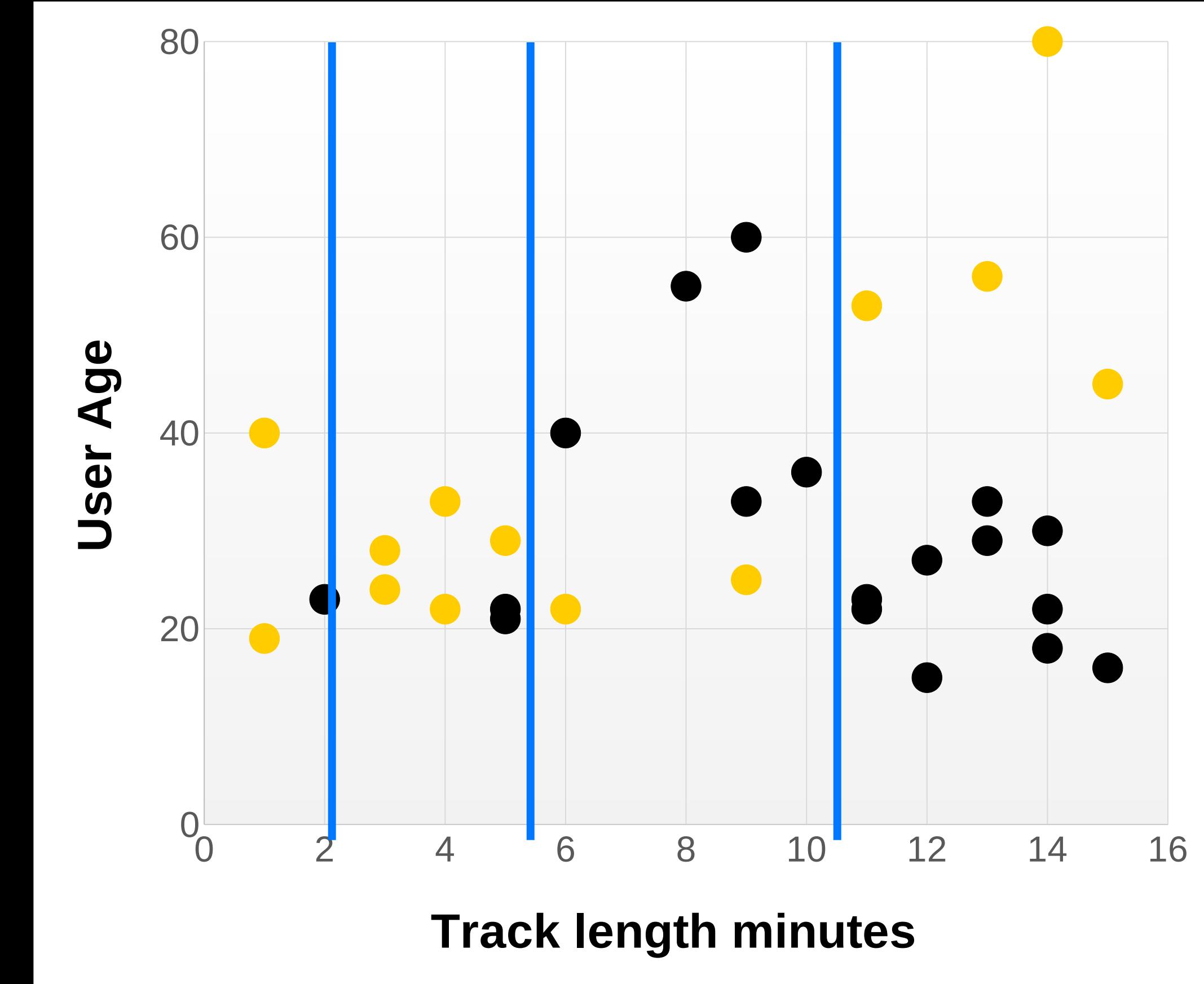
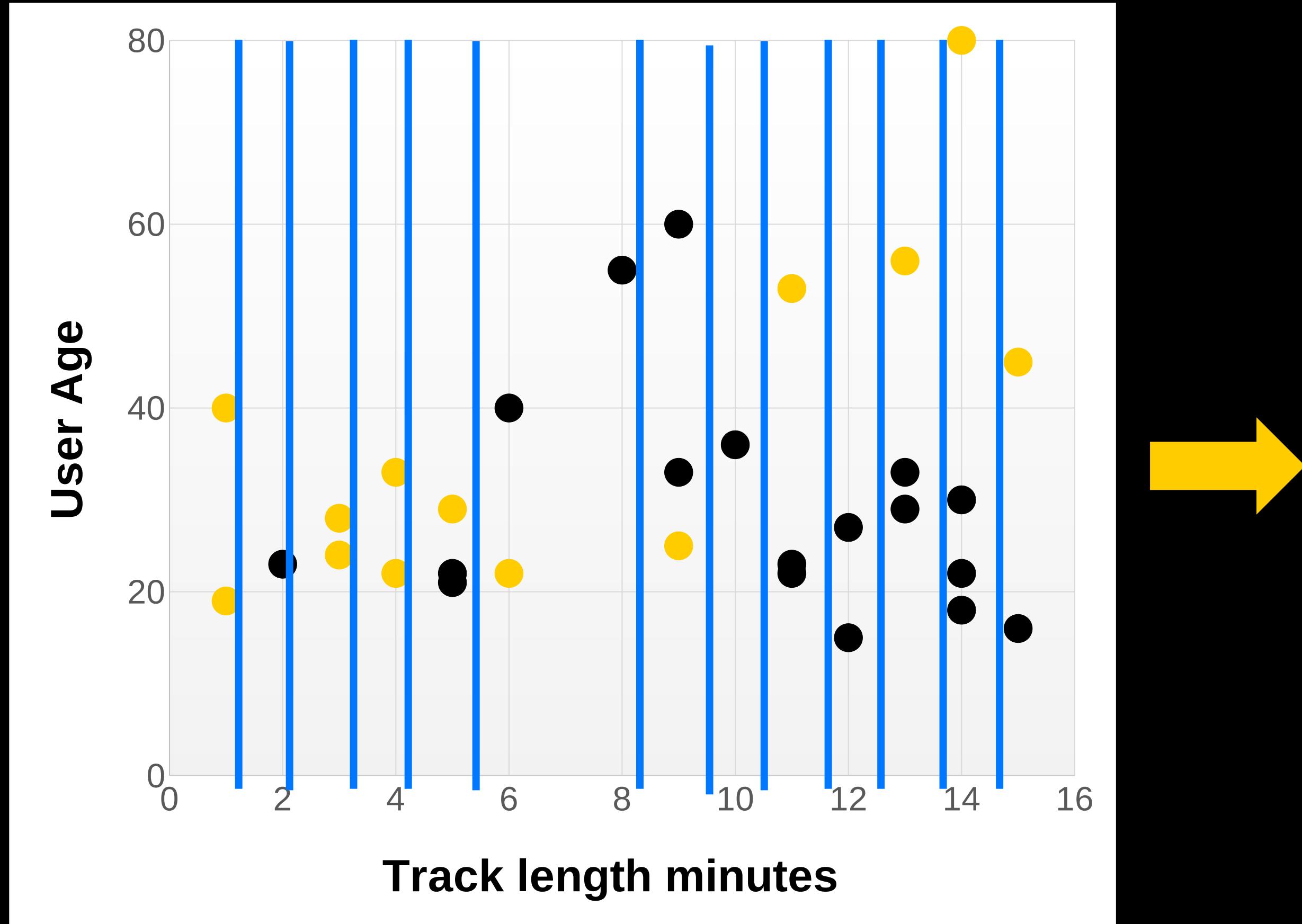
First order leaf estimation

$$\frac{\sum w_i g_i}{W}$$

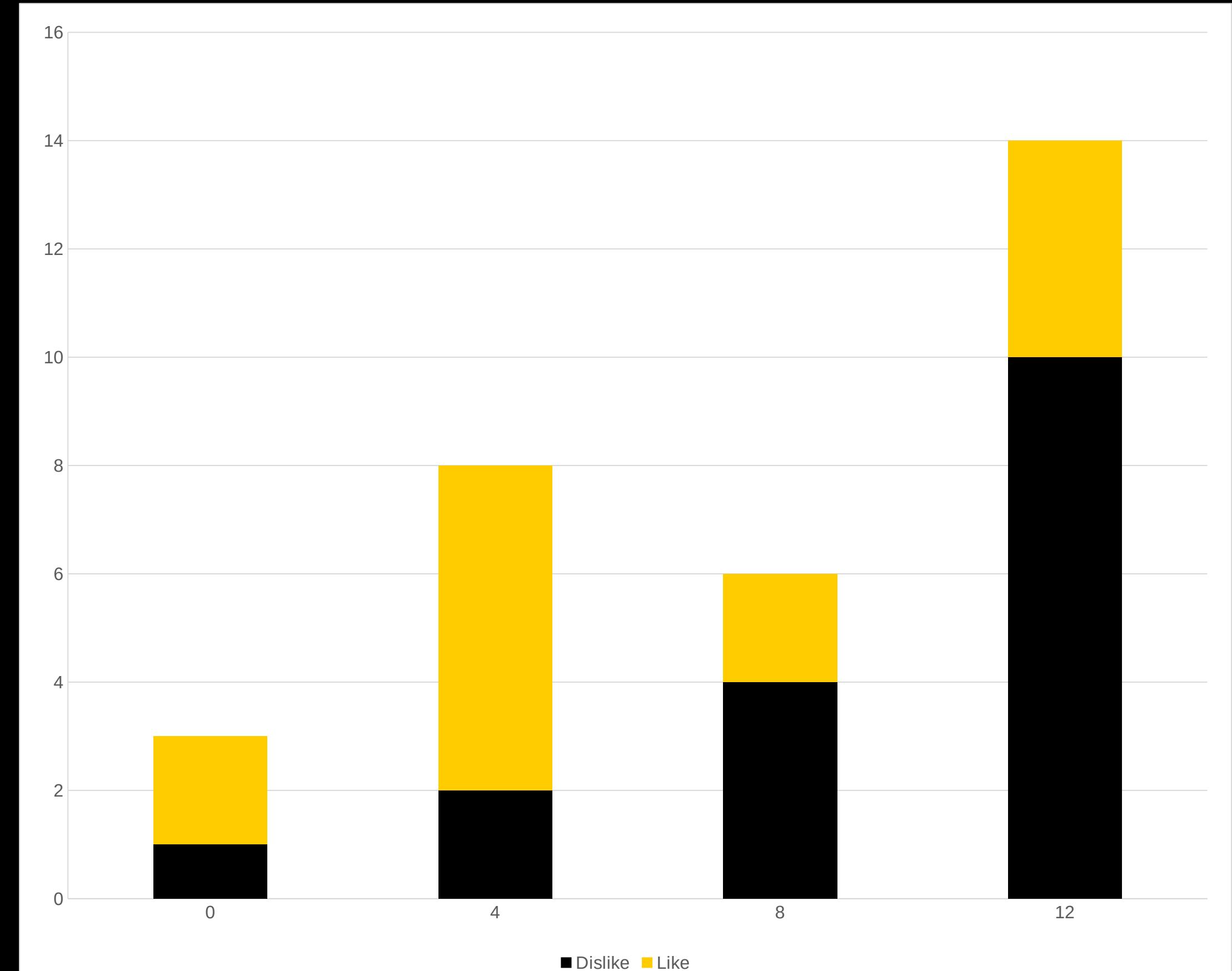
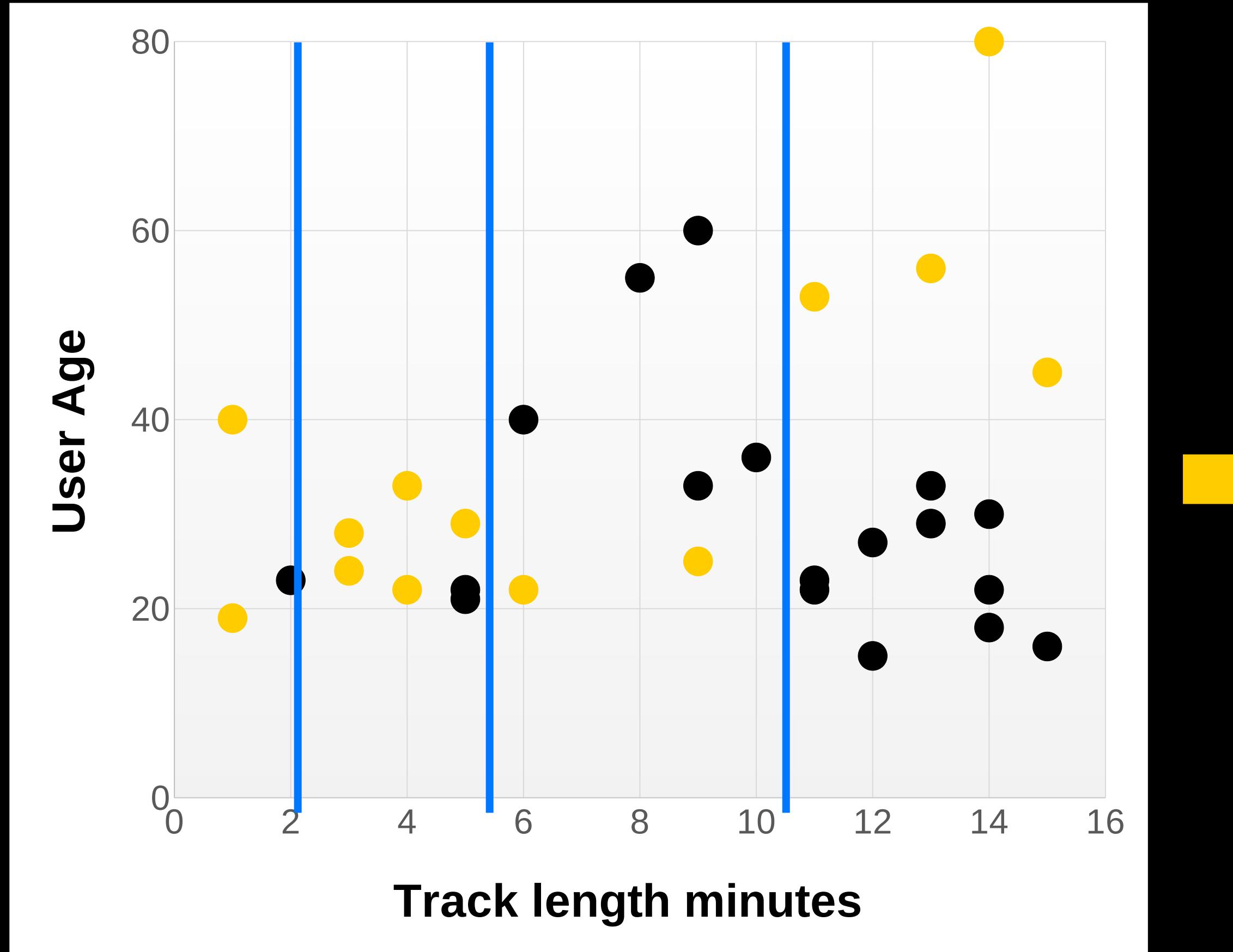
Second order leaf estimation

$$-\frac{\sum_i w_i l_i'}{\sum_i w_i l_i'' + \lambda}$$

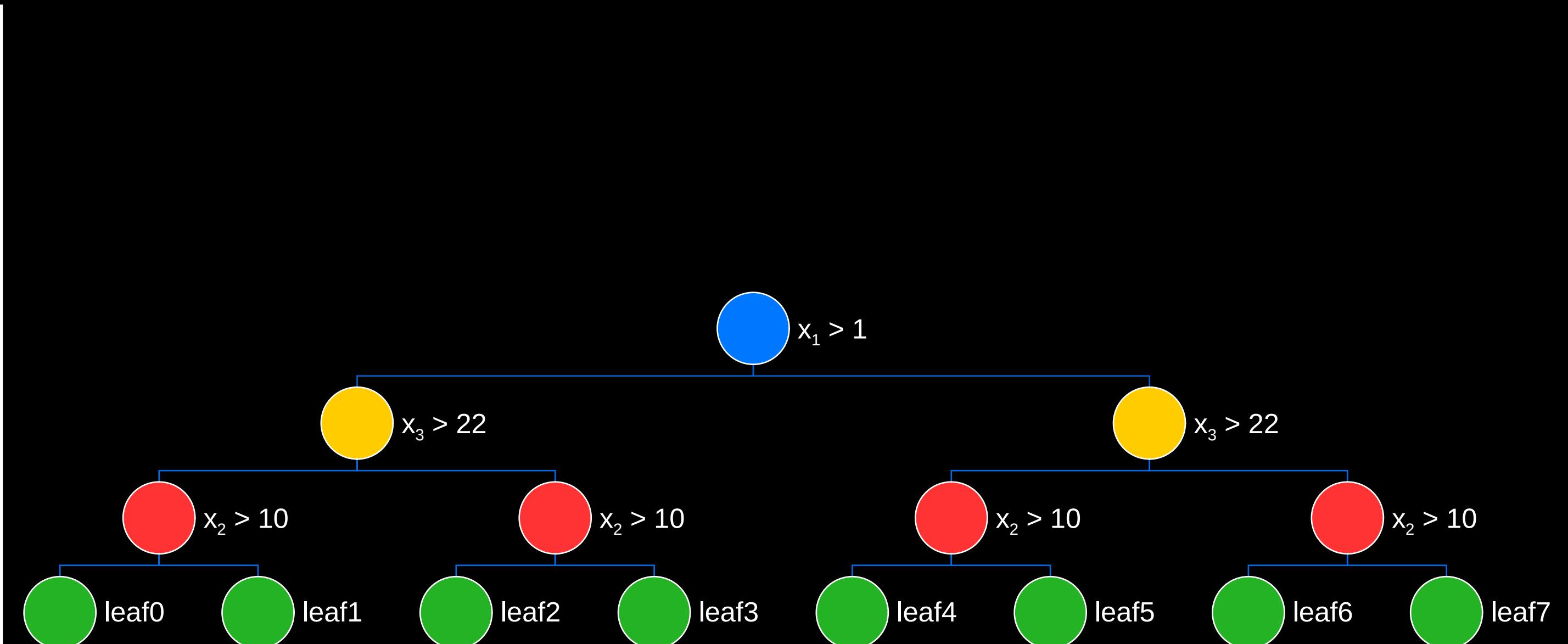
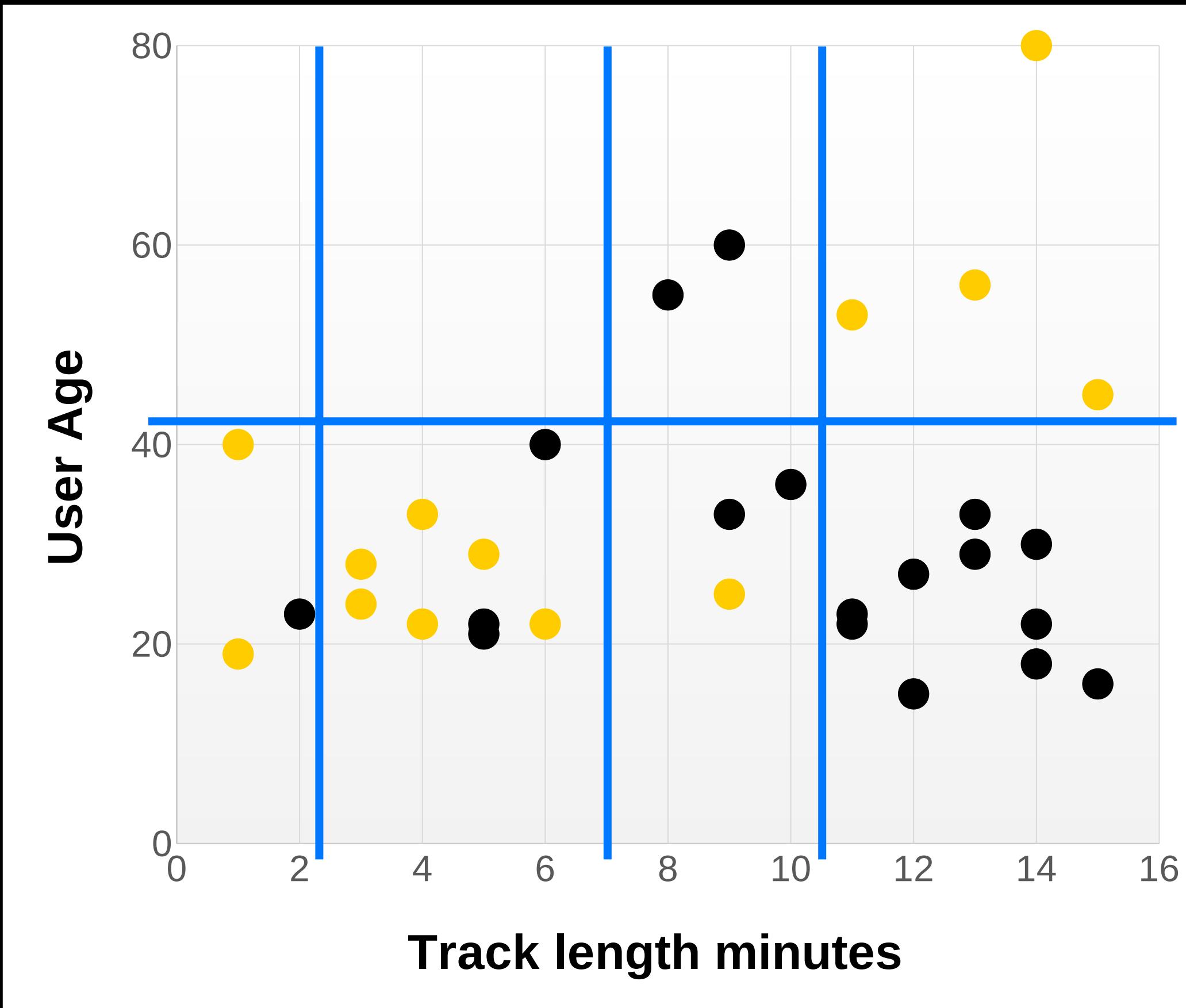
# Decision tree: quantization



# Decision tree: histograms

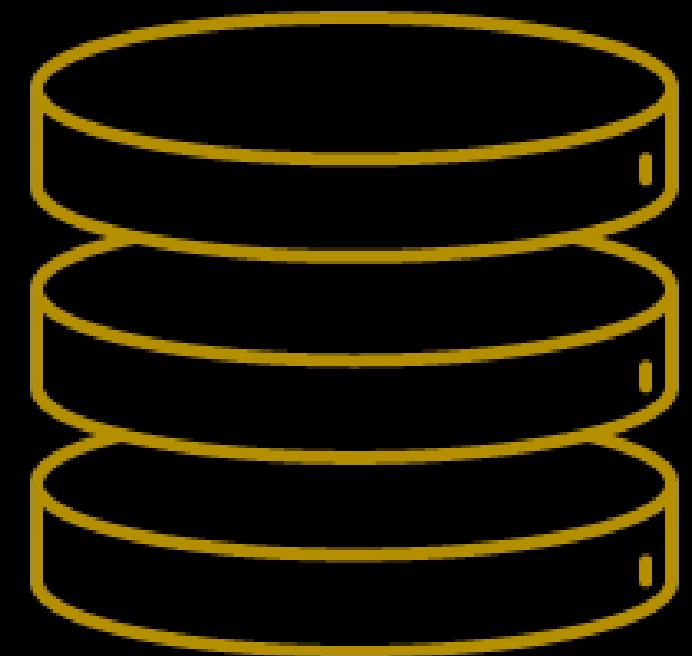


# Decision tree: symmetric(oblivious) trees



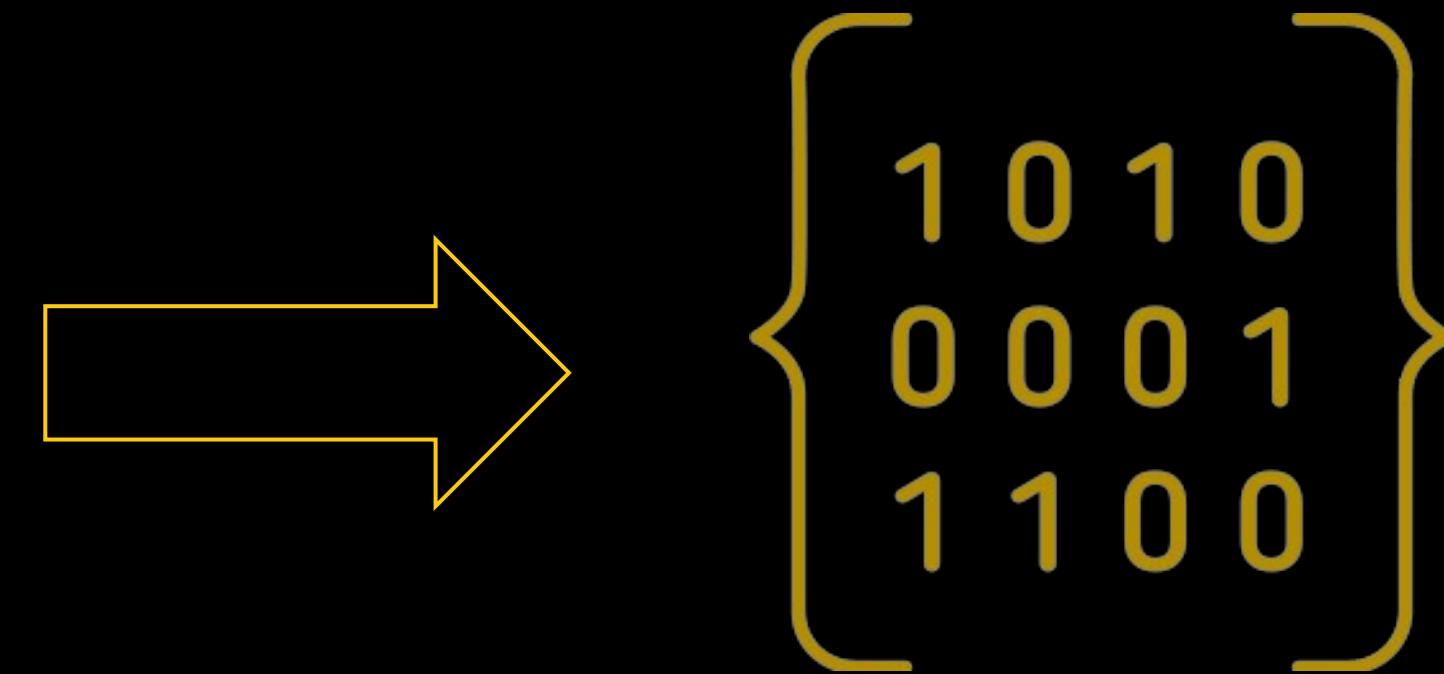
Catboost supports multiple tree grow policies:  
Symmetric (default option)  
Lossguide - like in LightGBM  
Depthwise - like in XGBoost

# Training on Large Data



Load raw

1TB  
(tsv data)



Quantize

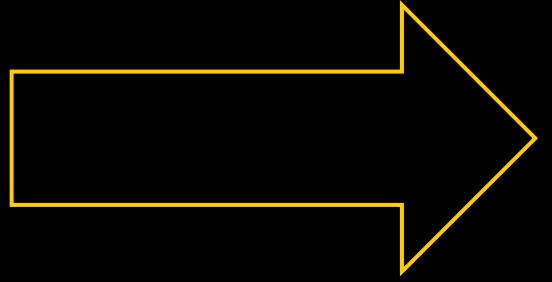
~160GB  
quantized  
representation



Train

~160GB  
in RAM or even less on GPU

# Quantization before training



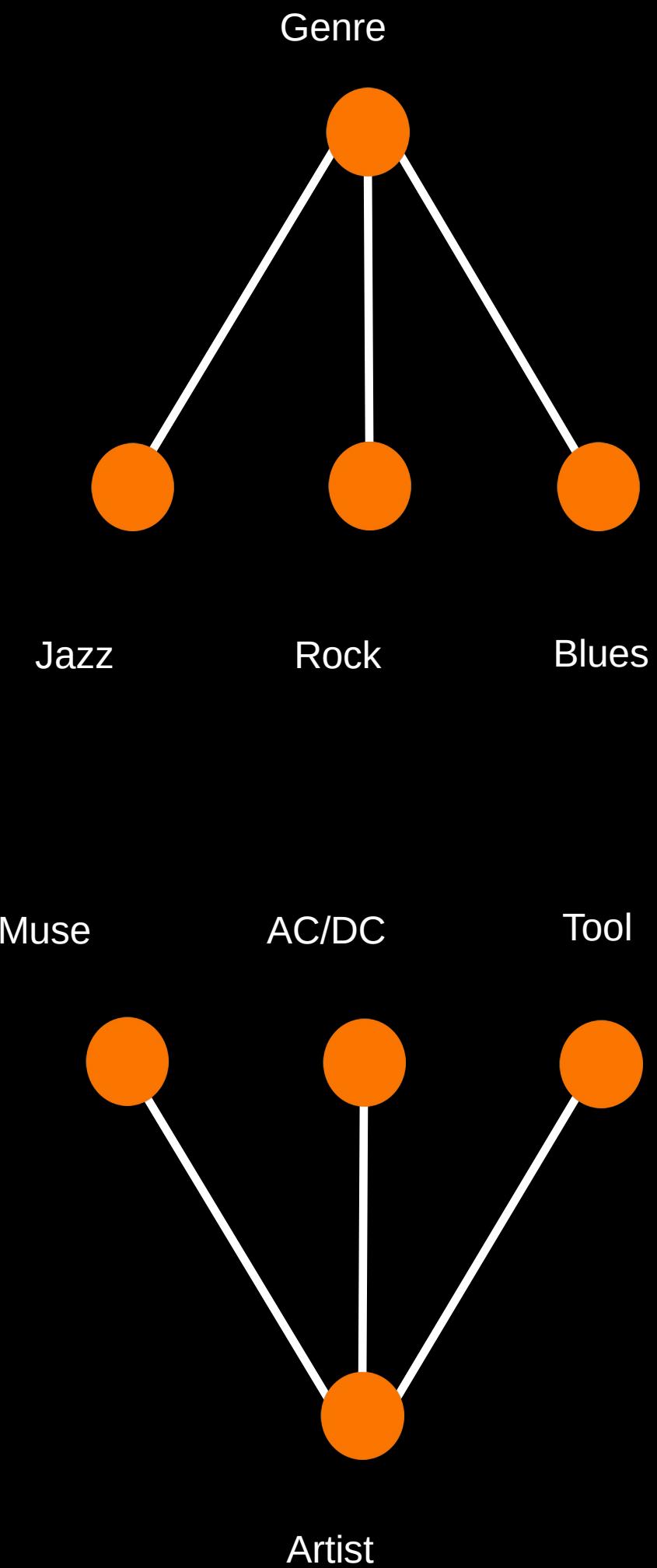
Load quantized

~160GB  
quantized  
representation

Train

~160GB  
in RAM or even less on GPU

# Categorical features handling



One-hot encoding

Statistics based on category

Category-based

Label based:  
calculated “online”  
(aka CatBoost-encoding)

Greedy search for feature combinations

# Online features for categories

$$ctr_i = \frac{countInClass + prior}{totalCount + 1}$$

For every sample “online” feature is calculated using objects with the same category before this one

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

$$i \longrightarrow \frac{1 + 1 + 0}{3}$$

# Text features

at first i was afraid i  
was petrified kept  
thinkin i could never  
live without you by  
my side

[0, 1, 2, 3, 4, 2,  
3, 5, 6, 7, 2, 8,  
9, 10, 11, 12,  
13, 14, 15]



BM25

Naïve Bayes

Bag-of-Words

...

# Bag of Words

- > Default: word unigrams + word bigrams
- > For a given dictionary (set of tokens) – is this token present in text?

Dictionary 1:  
yesterday, petrified  
Dictionary 2:  
at+first, i+has,  
could+never, ...

at first i was afraid i  
was petrified kept  
thinkin i could never  
live without you by  
my side

Feature values:  
0,1,1,0,1

# Naïve Bayes

- › For every class a new feature  $P(\text{Class}|\text{Text})$
- ›  $P(\text{Class}|\text{Text})$  is replaced with  $P(\text{Class}) * \prod_i P(\text{word}_i | \text{Class})$
- › **Most importantly:** calculate it “online”

# BM-25 for MultiClass

- › A new numerical feature for every class of multi-classification
- › TF – frequency of a word in text
- › IDF – inverted frequency of a word in a “document”, where “document” is a **concatenation of all texts in this class**
- › Most importantly: calculate it “online”

# Profit from text features

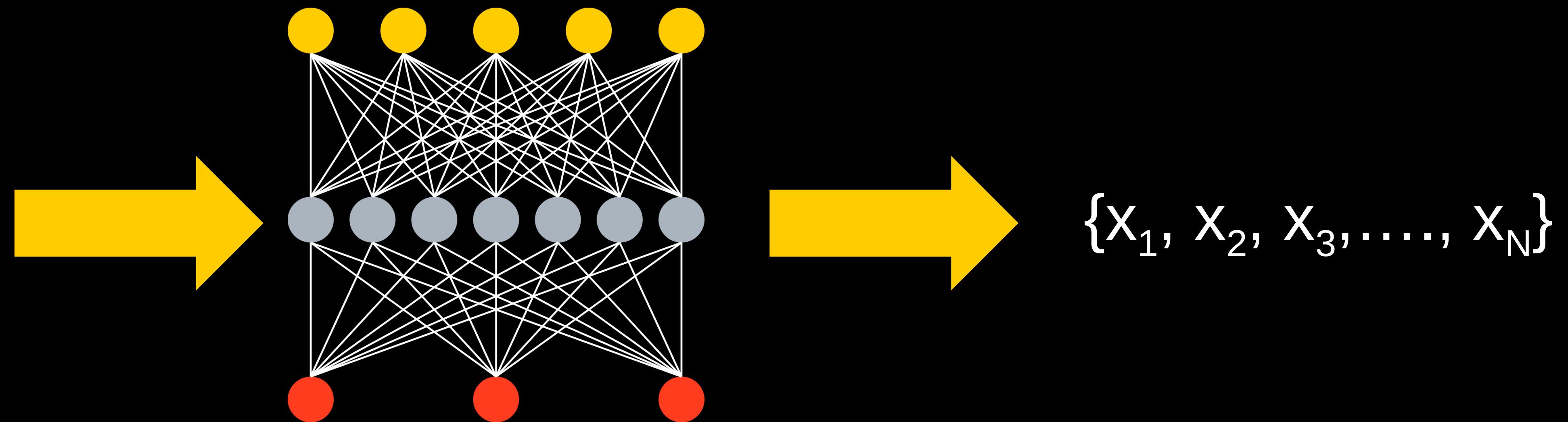
Accuracy on Rotten Tomatoes

Numerical + Caterogical  
0.4592

+ BOW  
0.4616

+ Online Text Features  
**0.4714**

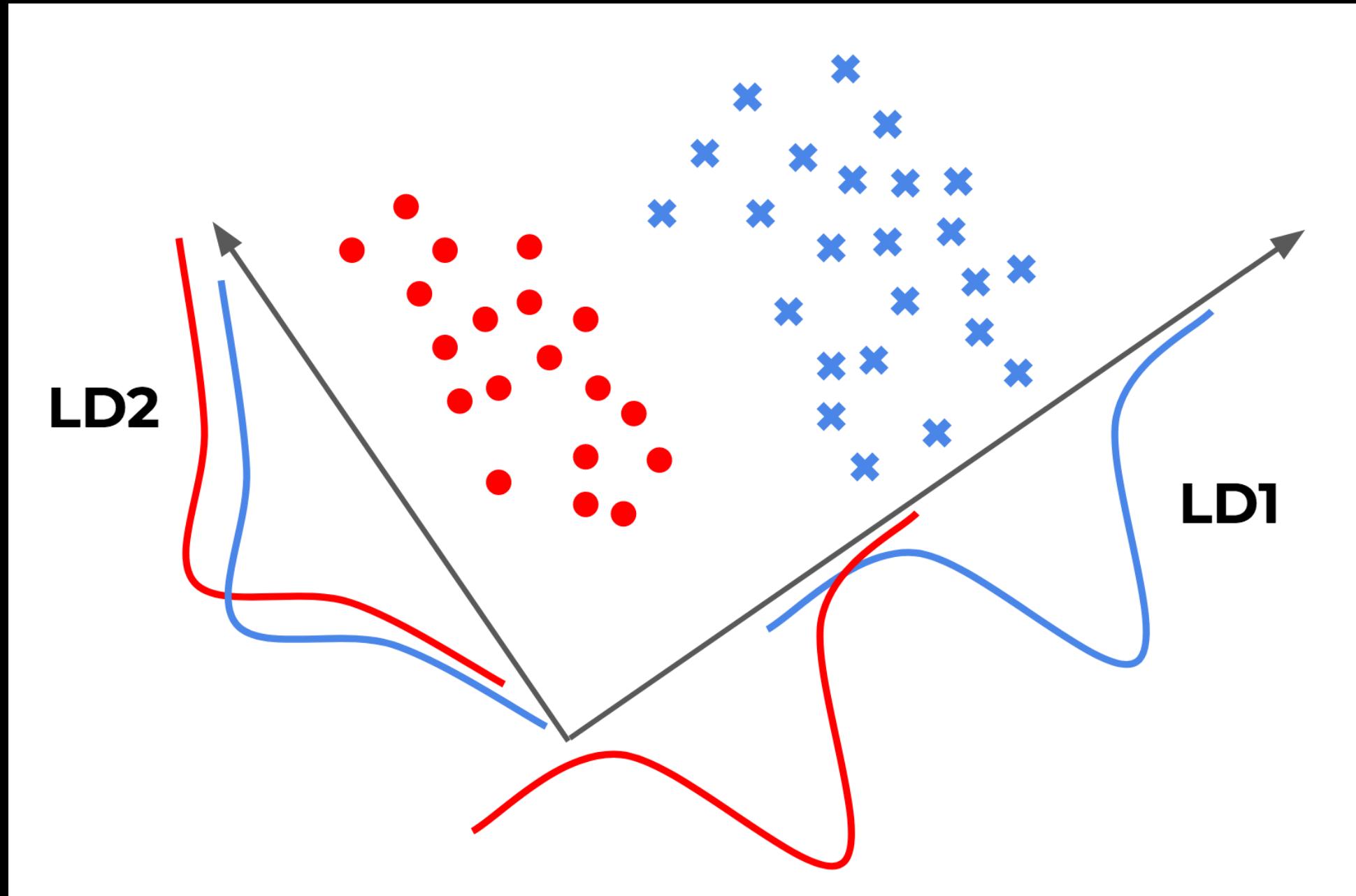
# Object embeddings



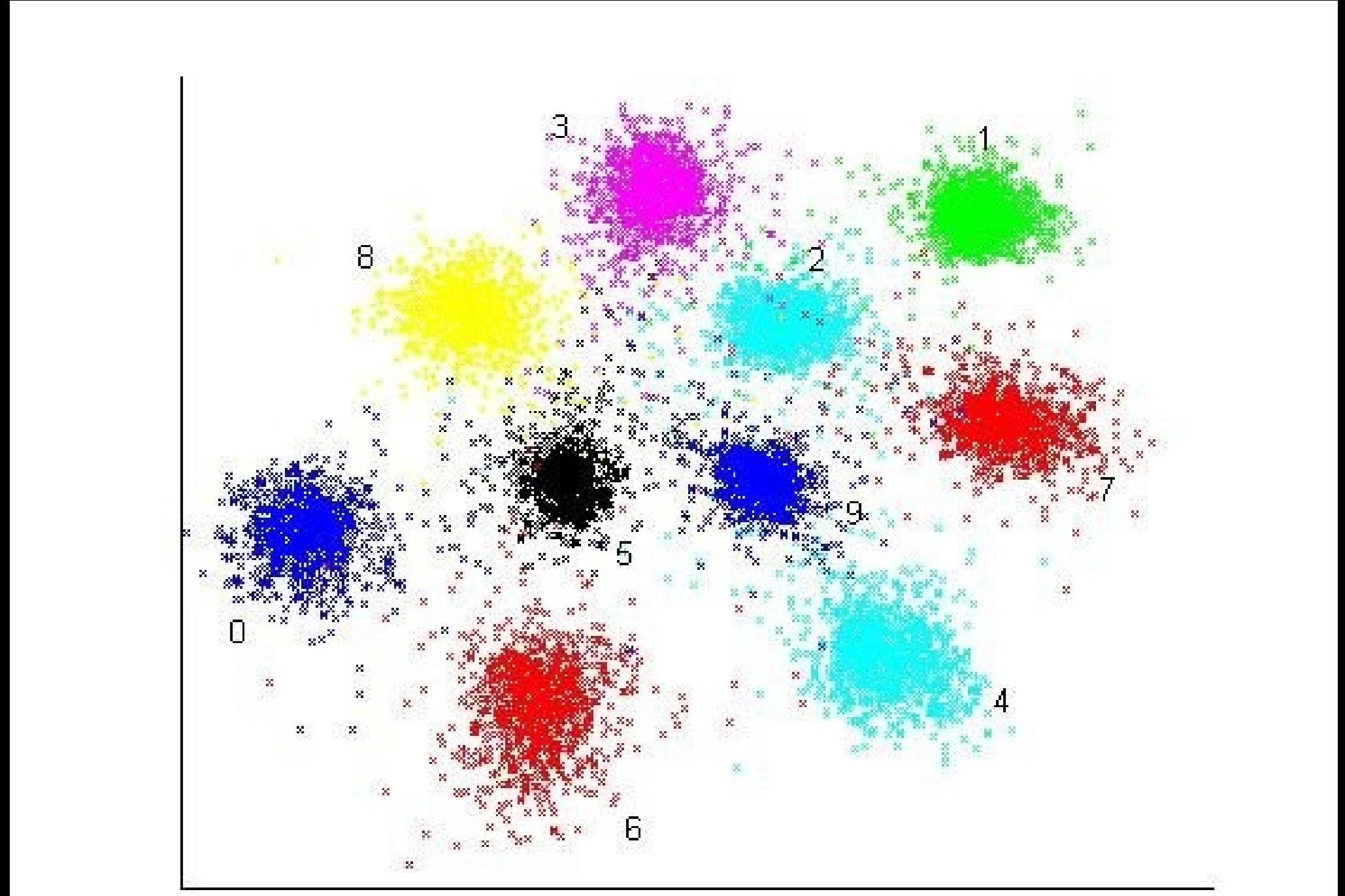
For example well known text embeddings: Word2Vec, DSSM, BERT, GPT and another language models.

# Experimental embeddings support

## Linear discriminant analysis



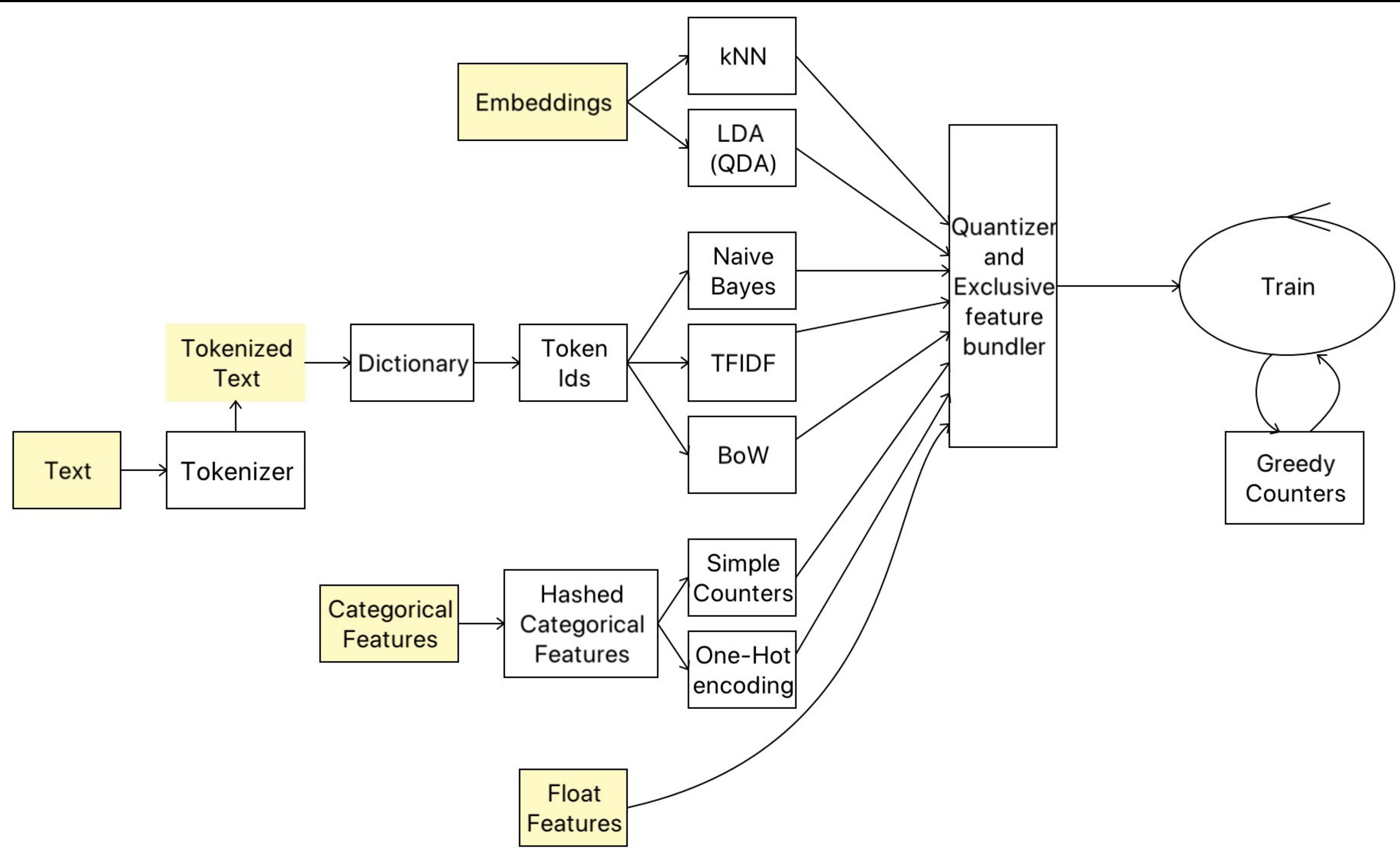
$$\log P(y = k|x) = \omega_k^t x + \omega_{k0} + Cst.$$



<https://raw.githubusercontent.com/eigenfoo/eigenfoo.xyz/master/assets/images/lda-pic.png>

A Deep Non-Linear Feature Mapping for Large-Margin kNN Classification - Scientific Figure on ResearchGate.  
Available from: [https://www.researchgate.net/figure/Two-dimensional-embedding-of-10-000-MNIST-test-data-using-the-Deep-Neural-Network-kNN\\_fig9\\_22076648](https://www.researchgate.net/figure/Two-dimensional-embedding-of-10-000-MNIST-test-data-using-the-Deep-Neural-Network-kNN_fig9_22076648)

# CatBoost – bird eye view

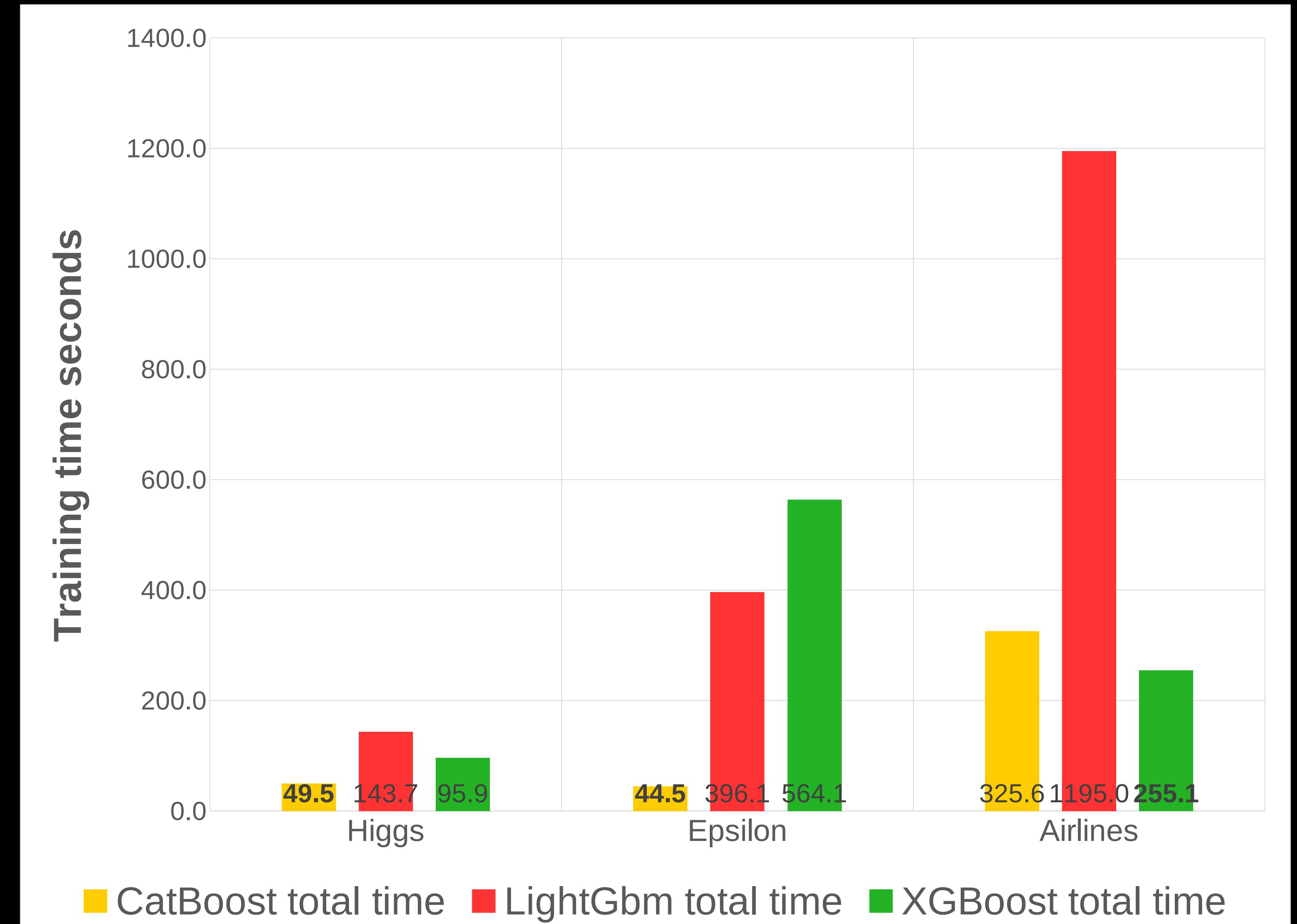


# V100 Training performance: 1000 trees

GPU training time on single  
NVIDIA Tesla V100

- Higgs  
10.5mln objects  
28 features
- Epsilon  
400k objects  
2k features
- Airlines  
23mln objects  
13 features

CatBoost v0.24.1, Lightgbm 2.3.1, XGBoost 1.0.2



# API is fully compatible with Spark ML library

```
from pyspark.sql import SparkSession

sparkSession = (SparkSession.builder
    .master("local[*]")
    .config(
        "spark.jars.packages",
        "ai.catboost:catboost-spark_2.4_2.12:0.25-rc4"
    ).getOrCreate()
)

import catboost_spark

trainDf = sparkSession.read().load("/my_datasets/train")

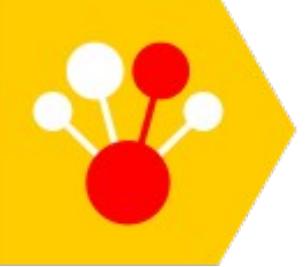
classifier = catboost_spark.CatBoostClassifier(iterations=20)
classifier.write().save("/my_classifier")

model = classifier.fit(trainDf)
model.write().save("/my_model")

applyDf = sparkSession.read().load("/my_datasets/for_application")

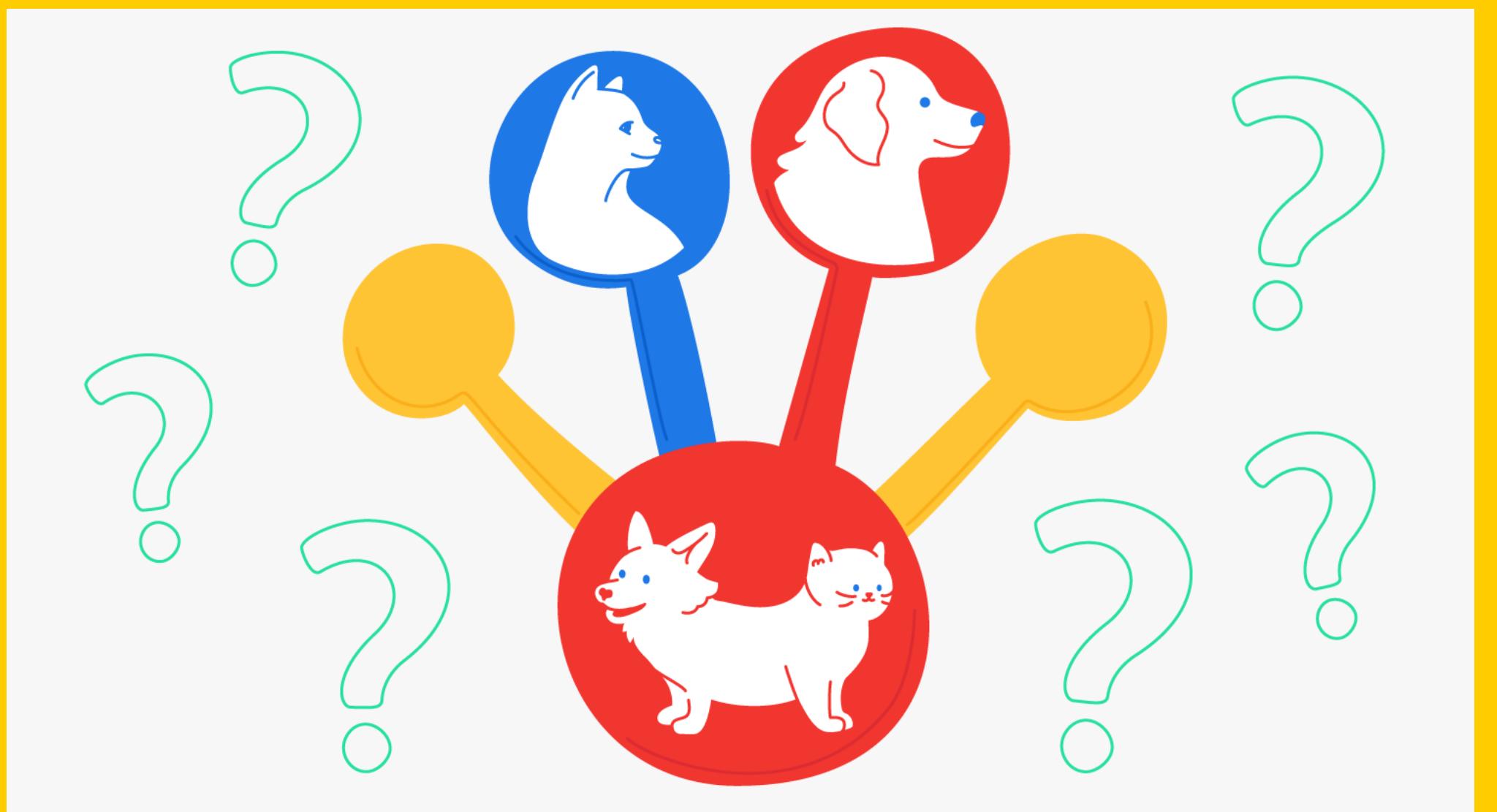
dfWithPredictions = model.predict(applyDf)
```

# CatBoost Spark Performance vs Competitors

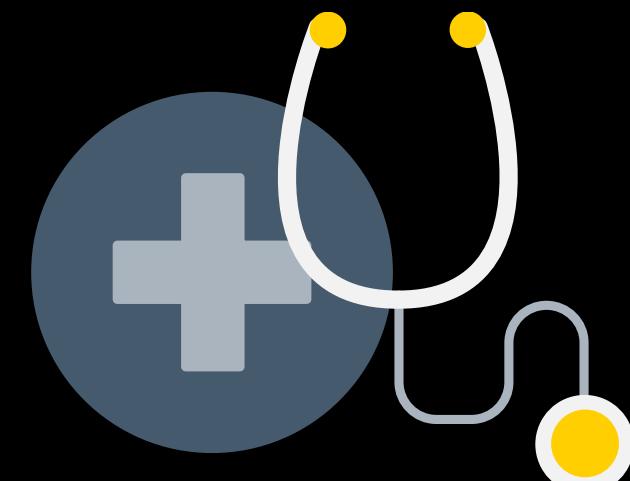
	Criteo derived 170 m samples 65 features		Epsilon 400 k samples 2000 features		Higgs 10.5 m samples 28 features	
	total	per iter	total	per iter	total	per iter
 CatBoost	53 m 52 s	2.8 s	1 h 5 m	3.7 s	7 m 40 s	0.31 s
 LightGBM	59 m	3.5 s	2 h 36 m	9.4 s	2 h 25 m	8.7 s
 XGBoost	est. 8 h	28.9 s	17 m 25 s	1 s	10 m	0.6 s

- › Cluster configuration – 16 nodes x 16 cores
- › Time on 1000 iterations, total time includes preprocessing

# Uncertainty estimation



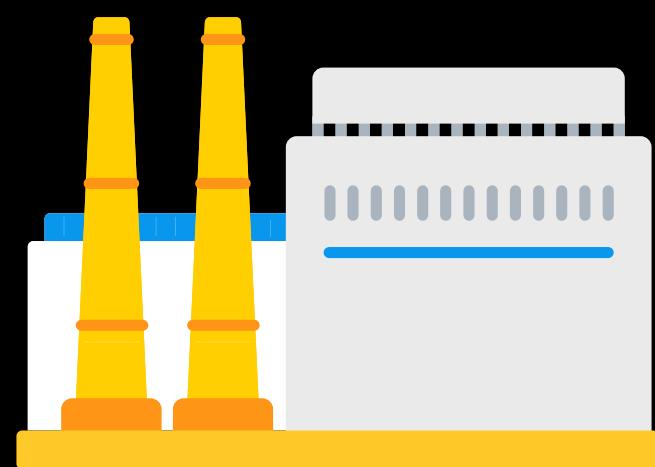
# Whom it may concern?



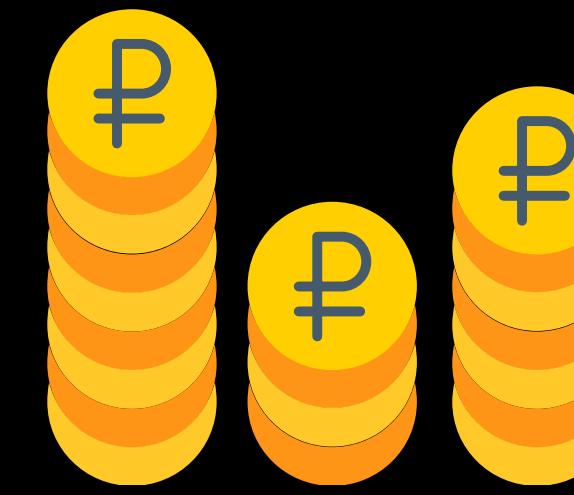
Medcine



Sales prediction



Production  
processes



Finance &  
Credit

# Data uncertainty

For classification – data uncertainty is entropy of  $H(\bar{p}) = -\bar{p} \ln \bar{p} - (1 - \bar{p}) \ln(1 - \bar{p})$   
prediction

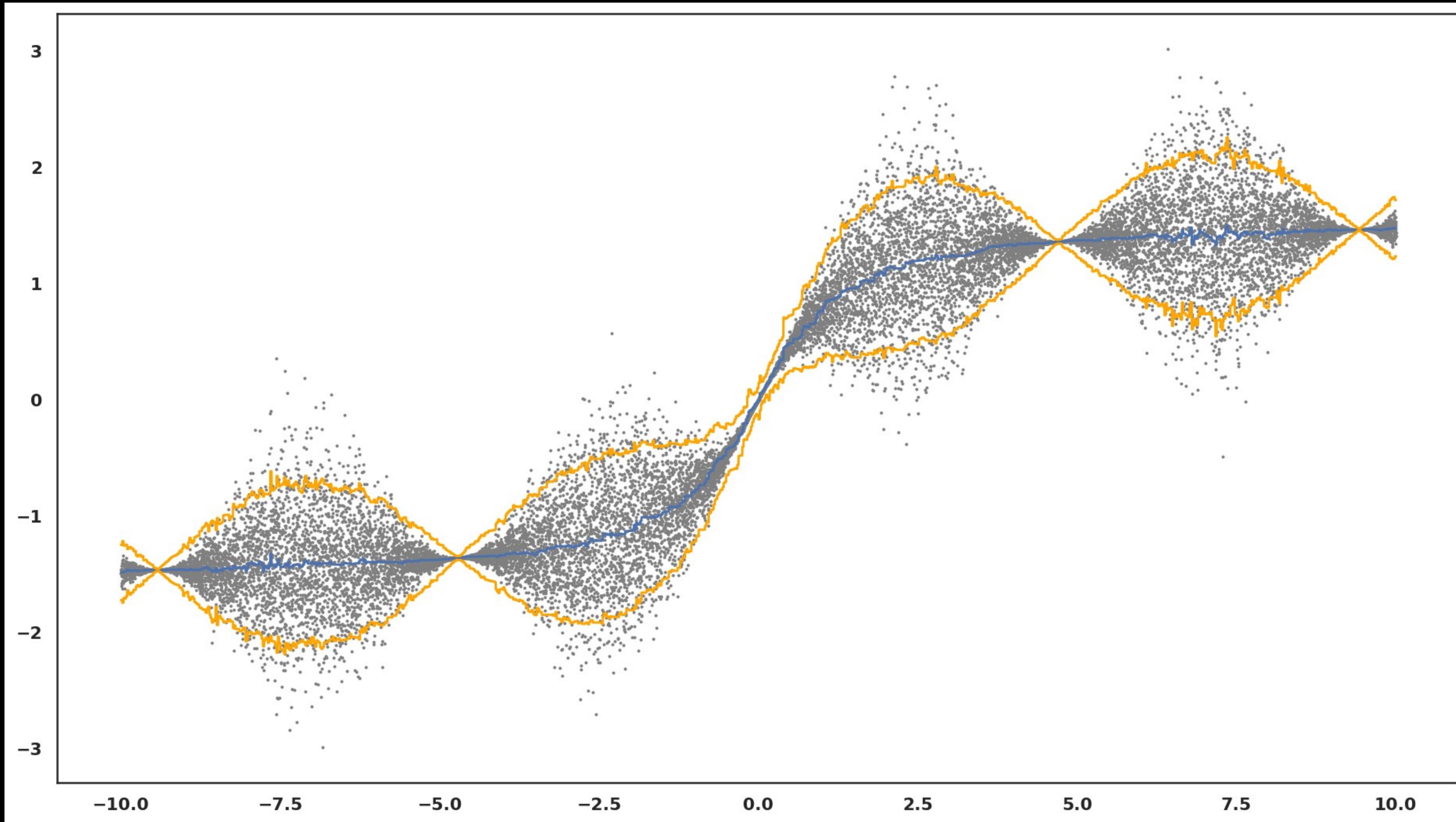
For regression data uncertainty could be estimated with Natural Gradient Boosting (NGBoost)  
This loss function is named RMSEWithUncertainty in CatBoost library

$$-\frac{1}{N} \sum_{i=1}^N \log p(t_i | a_i) = -\frac{1}{N} \sum_{i=1}^N \log\left(\frac{1}{2\pi\sigma^2} \exp\left(-\frac{(y - \mu)^2}{2\sigma^2}\right)\right) = C + \frac{1}{N} \sum_{i=1}^N \left(a_{i,1} + \frac{1}{2} \exp(-2a_{i,1}(t_i - a_{i,0})^2)\right),$$

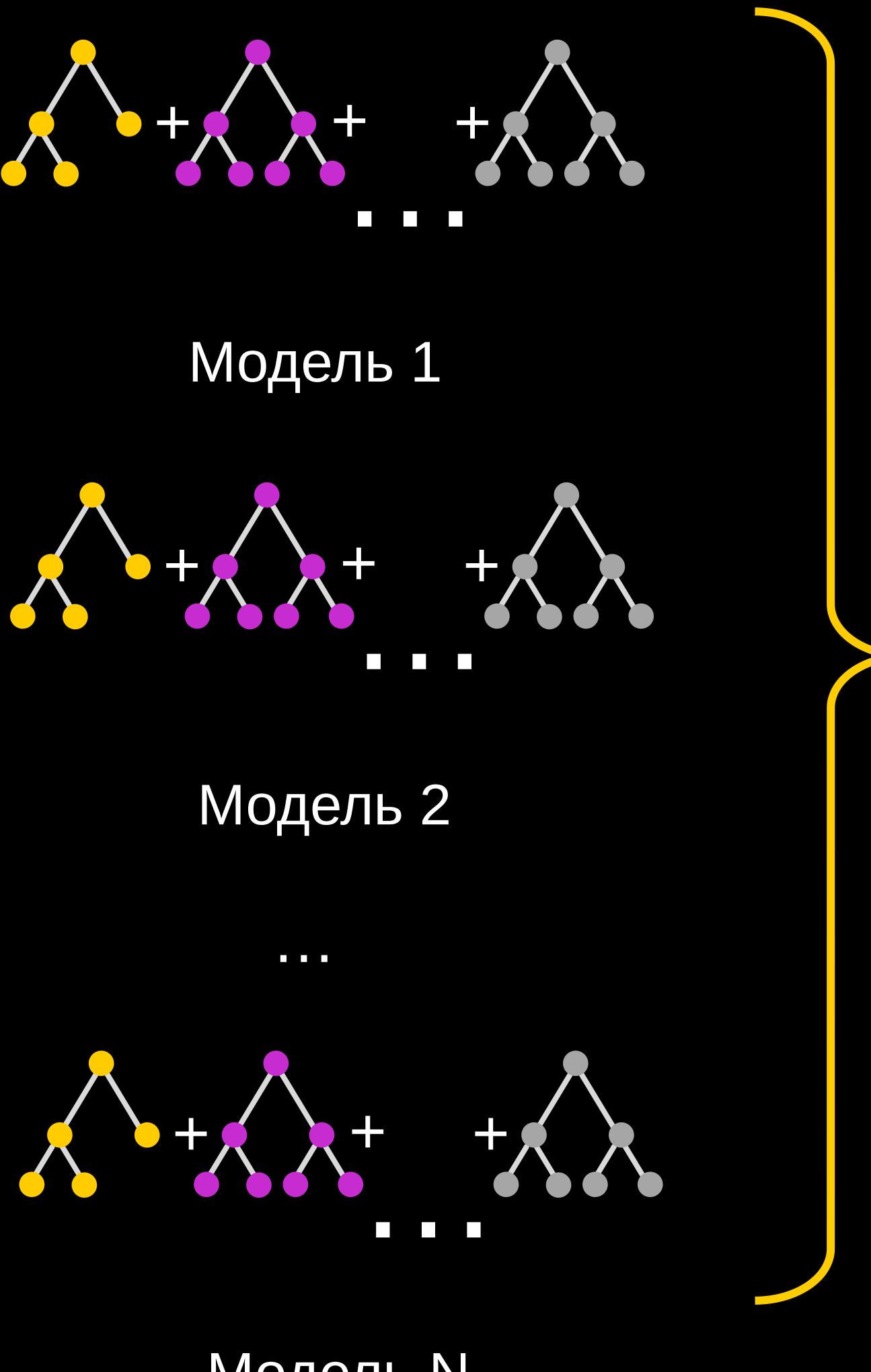
where t is target, a -- 2-dimensional approx  $a_0$  is target predict,  $a_1$  is log  $\sigma$  predict, and p has normal distribution

$$p(t|a) = \mathcal{N}(y|a_0, e^{2a_1}) = \mathcal{N}(y|\mu, \sigma^2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(y - \mu)^2}{2\sigma^2}\right):$$

# Data uncertainty example



# Knowledge uncertainty: ensemble approach



Regression

Prediction variance = knowledge uncertainty

Classification

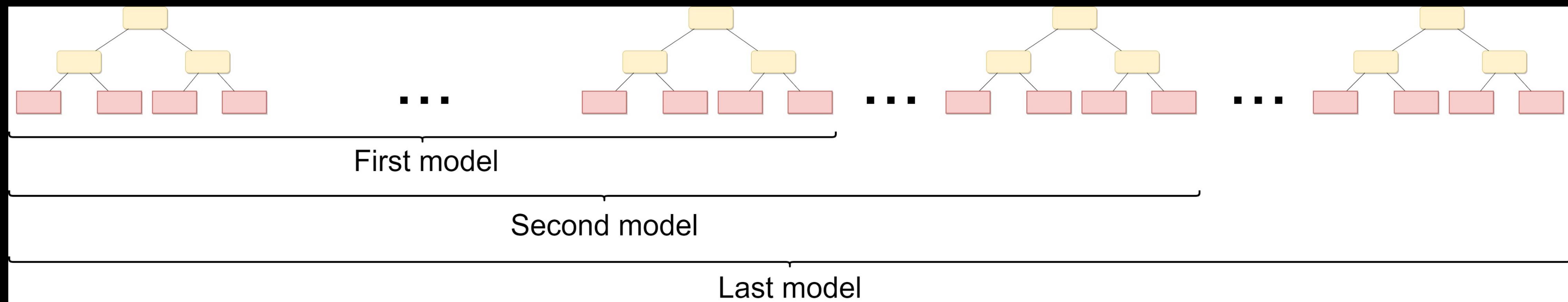
Mean entropy of predictions = full uncertainty

$$H(\bar{p}) = -\bar{p} \ln \bar{p} - (1 - \bar{p}) \ln(1 - \bar{p})$$

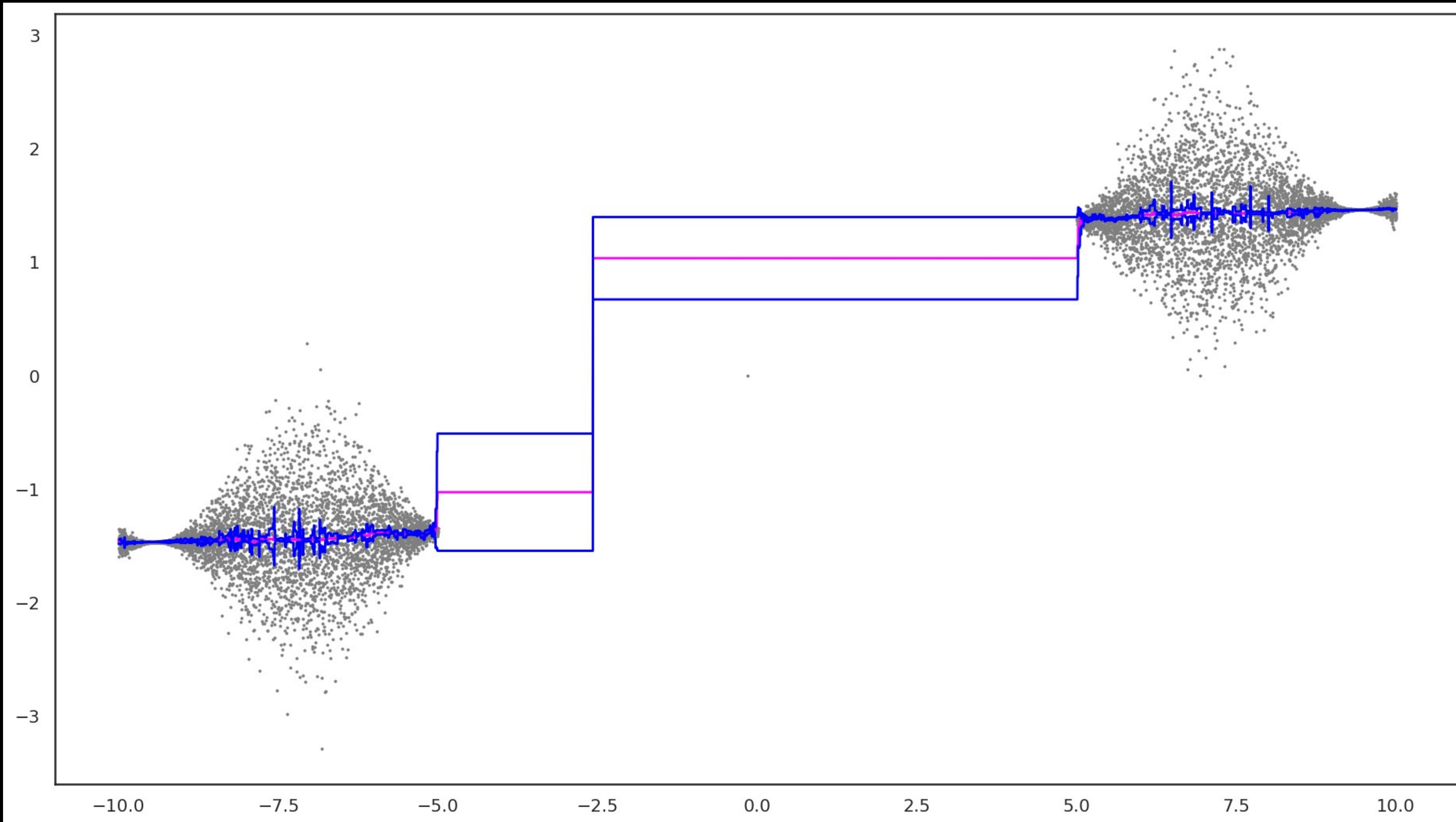
The entropy of mean prediction – data uncertainty

$$H(\bar{p}) = -\bar{p} \ln \bar{p} - (1 - \bar{p}) \ln(1 - \bar{p}) \text{ where } \bar{p} = \frac{\sum_i p_i}{N}$$

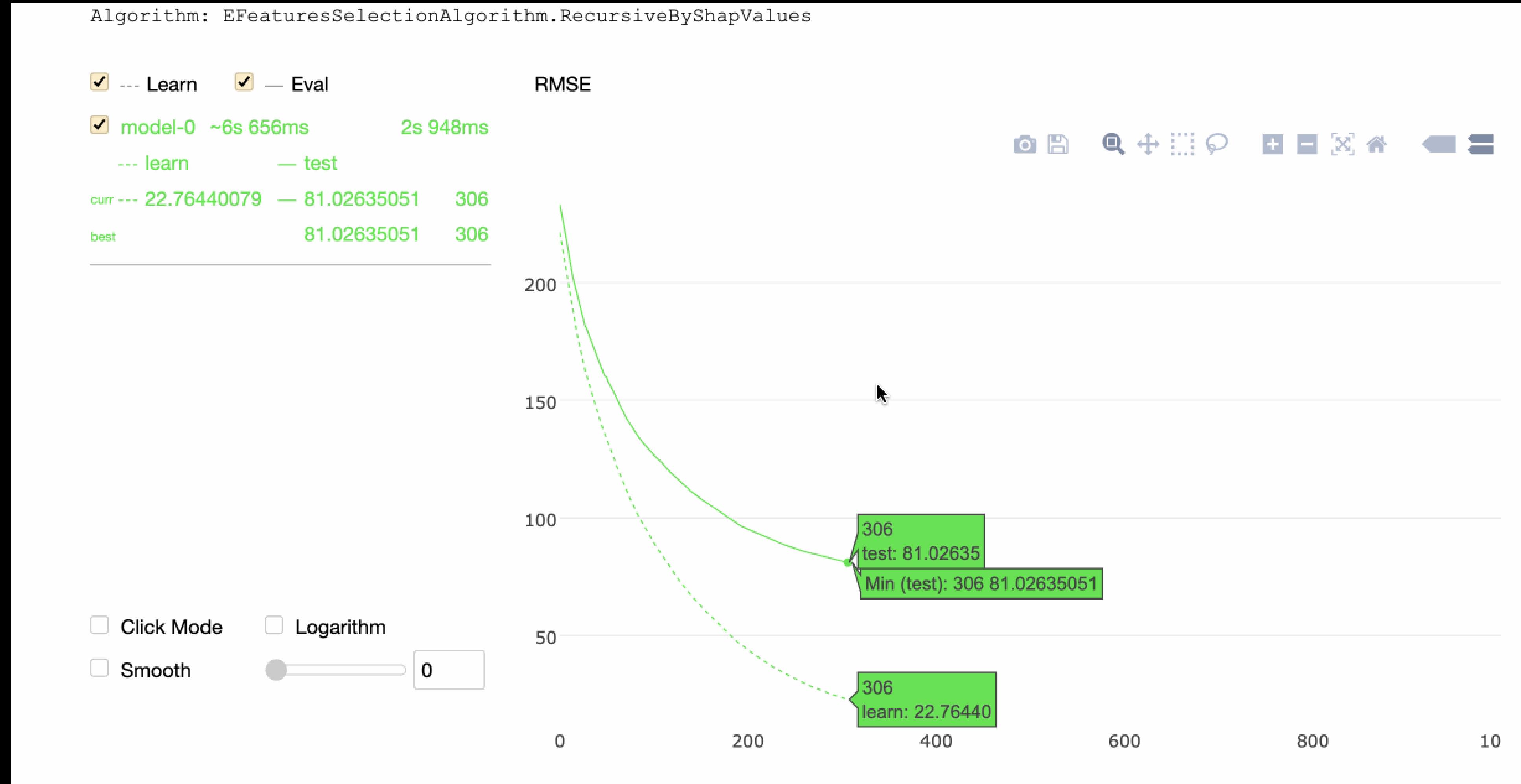
# Knowledge uncertainty estimation with virtual ensembles



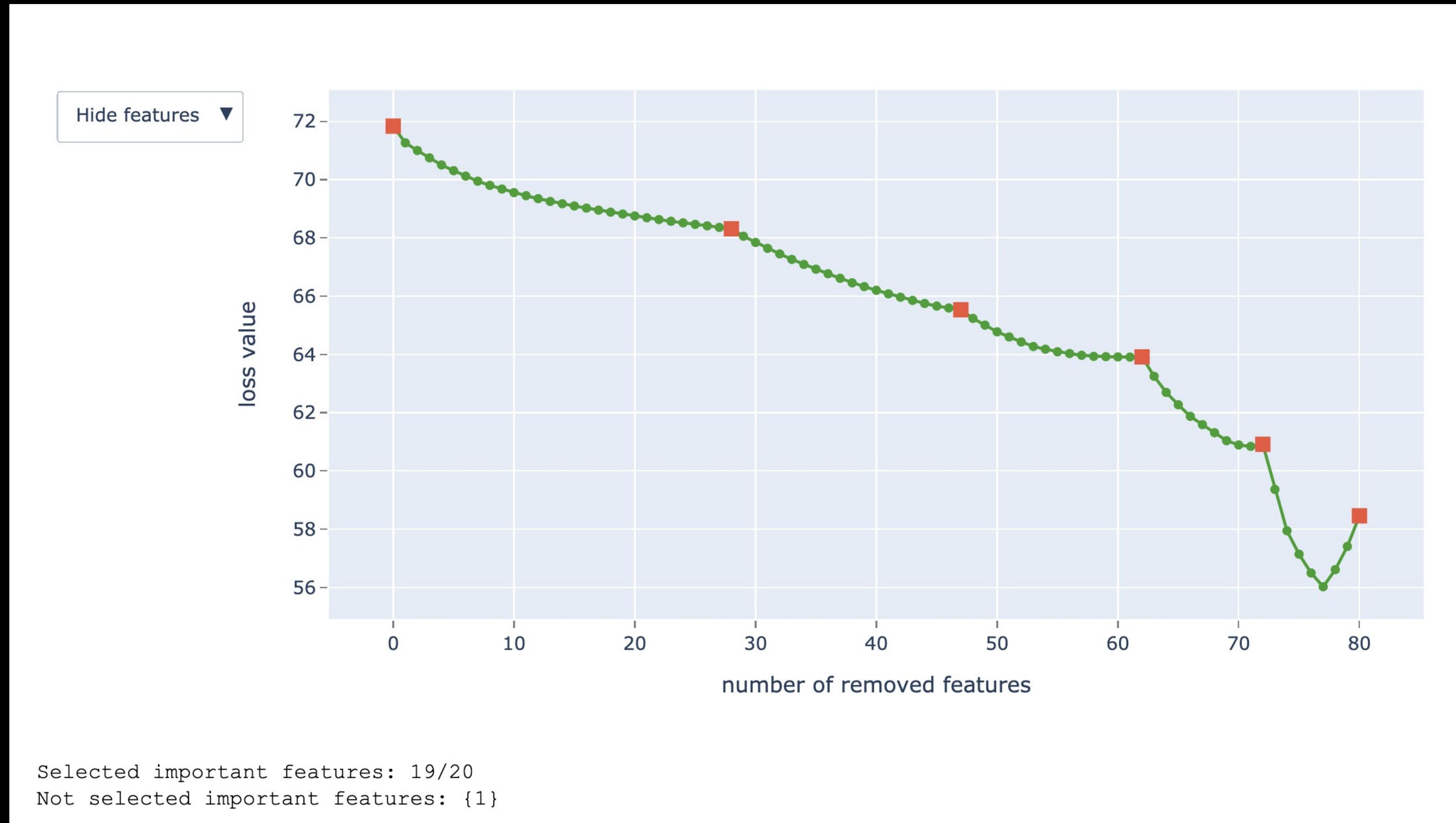
# Knowledge uncertainty example



# Recursive feature elimination: real time visualization

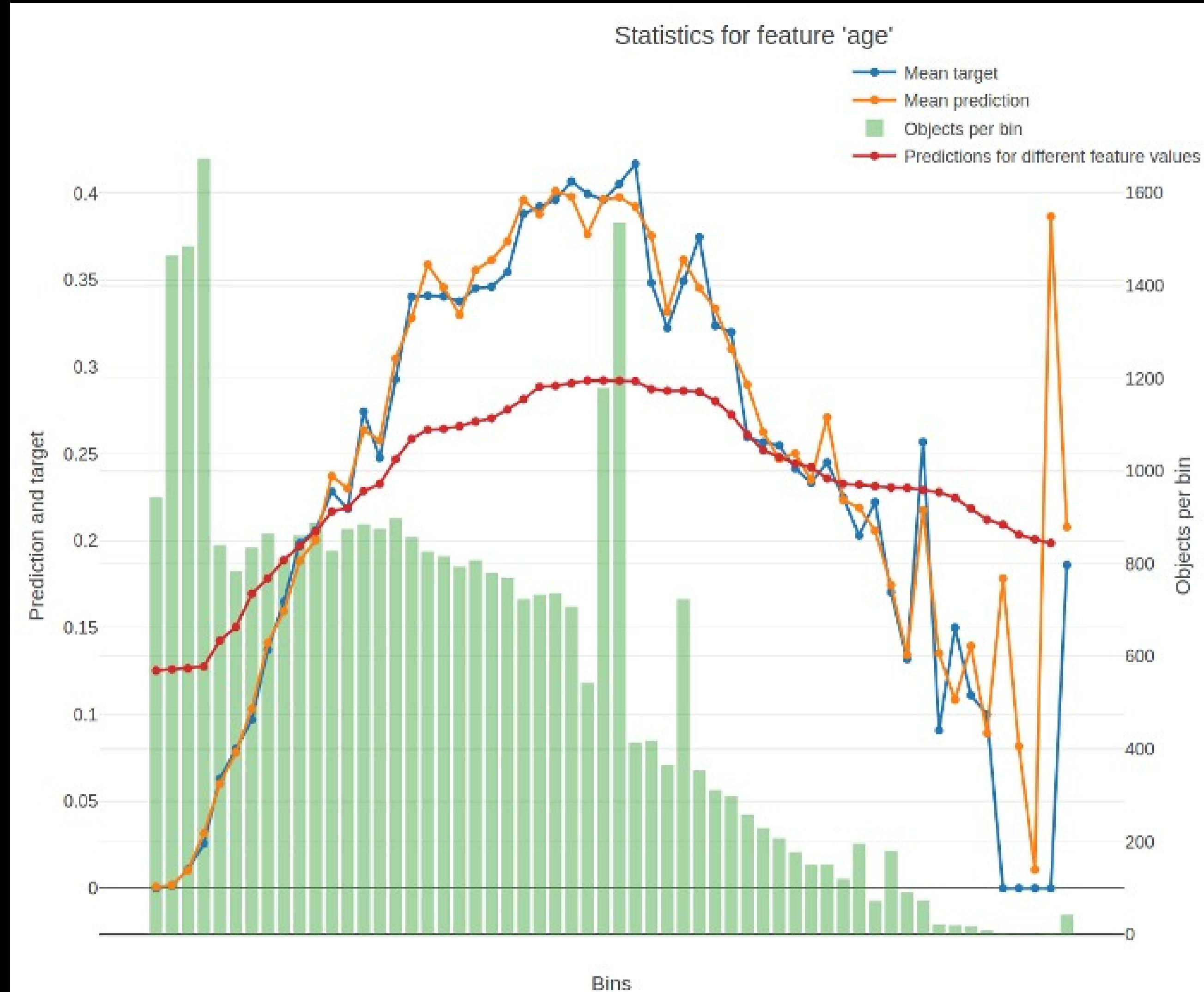
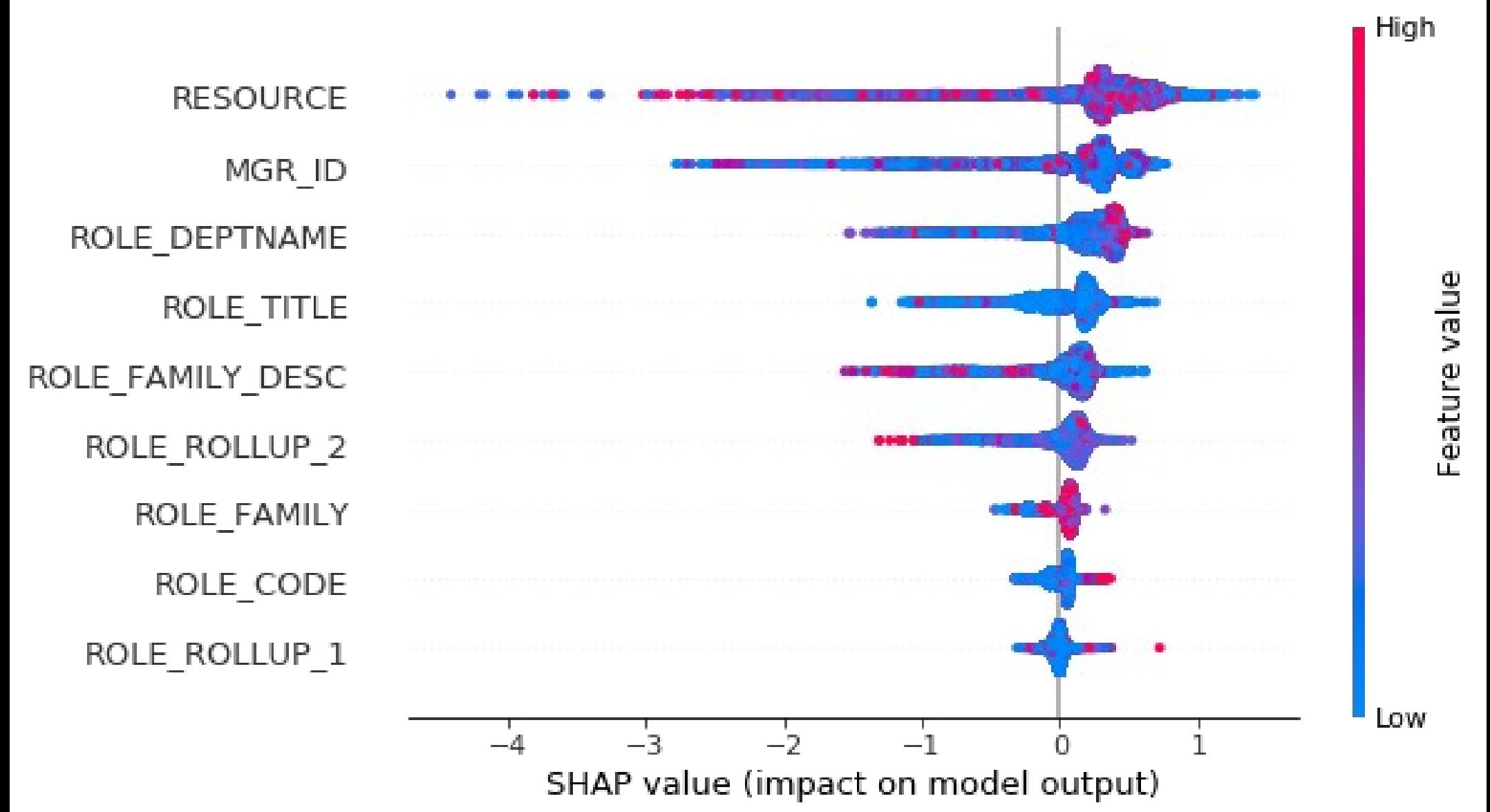


# Recursive feature elimination: result visualization

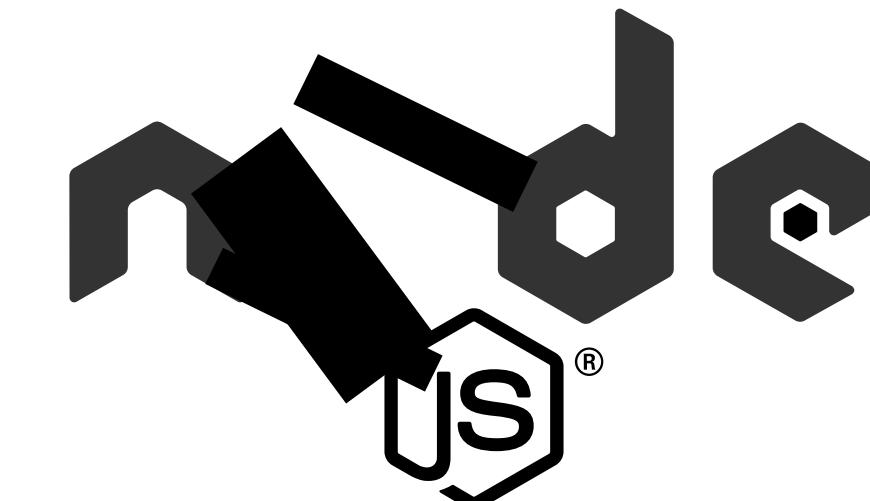
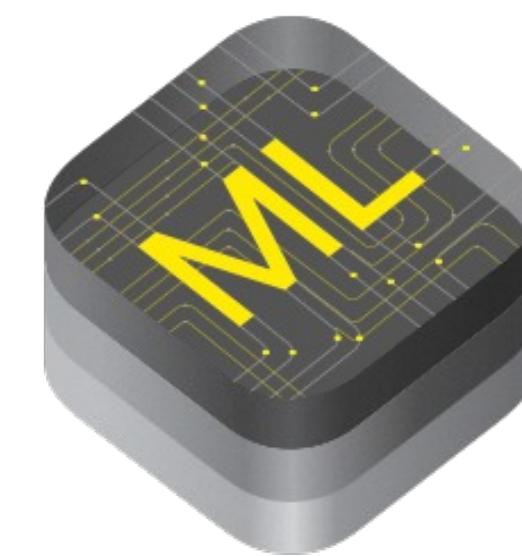
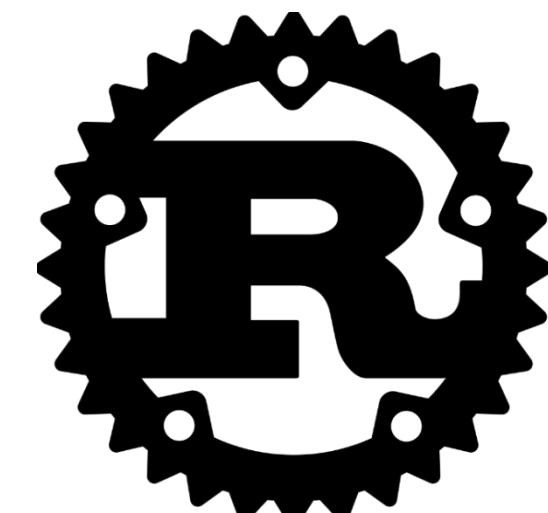
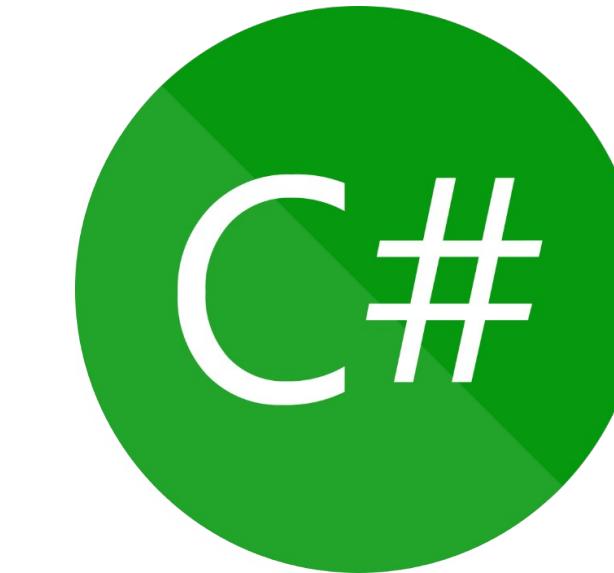


# Model Analysis tools

- > Per feature model analysis charts
- > New types of feature importance
- > Tree visualization
- > Ranking analysis



# Integration in production



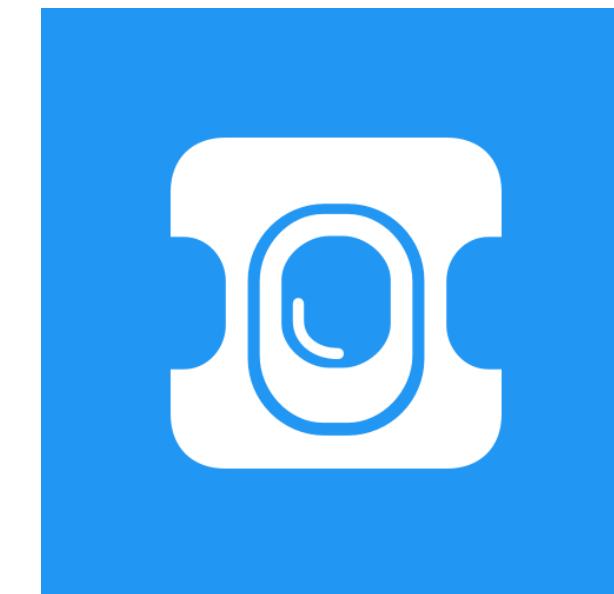
# Plans

- › CatBoost categorical feature encoder available to use outside of trained CatBoost model
- › Solve discrepancy in quality between CPU & GPU version
- › Support GPU for Spark training
- › Better R library support – solve lot's of persisting problems,
- › Make our HNSW implementation as part of CatBoost library
- › Faster and more handy embeddings support
- › Model & counters finetuning on new data for trained model

# CatBoost: usage examples

# Known CatBoost usages

- › Recommendations at Netflix
- › Hotel ranking in Aviasales
- › Protection against bots in CloudFlare
- › Particle classification in CERN
- › Medical research at University of NSW Sydney
- › Destination prediction in Careem taxi service
- › ML competitions on Kaggle



# Yandex.Weather

## Task?

- › Cloudiness type and temperature prediction

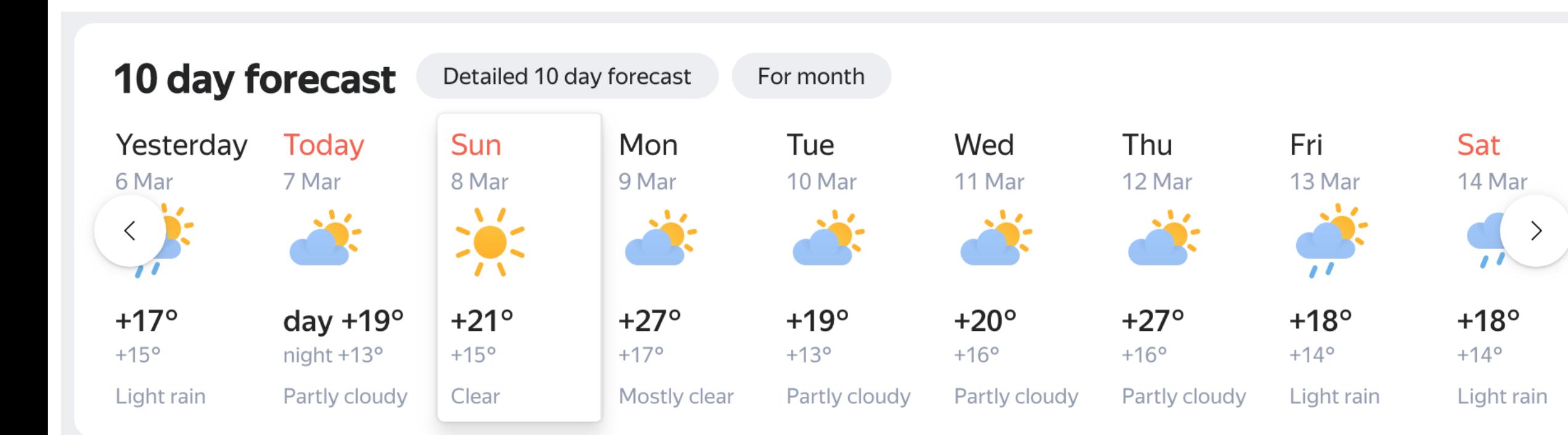
## Task type: multiclassification and regression

## Dataset features

- › Physical weather model output
- › Neural network output
- › Online-data from weather stations
- › Weather historical data

## CatBoost features used:

- › Multiclassification target and RMSE (for temperature)
- › GPU training
- › Feature importance analysis
- › Training process visualization



# Yandex.Search

## Task?

- › Search document order prediction

## Task type: ranking

## Dataset features:

- › Classic features (PageRank, BM25 and others)
- › Neural Networks output

## CatBoost features used:

- › YetiRankPairwise target
- › Distributed GPU training
- › Model blending
- › Feature importance analysis
- › Ranking analysis

The screenshot shows the Yandex search interface with the query "catboost" entered in the search bar. The results are filtered under the "Web" tab. There are 20 thousand results found. The top result is a link to the official GitHub repository of CatBoost, described as an open-source gradient boosting library. Other results include links to the Yandex Technologies website, Stack Overflow questions about CatBoost, and a news article from Habrahabr.ru.

- ➊ **CatBoost - open-source gradient boosting library**  
catboost.yandex.com  
CatBoost is an algorithm for gradient boosting on decision trees. ... New version of CatBoost has industry fastest inference implementation.
- ➋ **CatBoost · GitHub**  
github.com > CatBoost  
CatBoost is an open-source gradient boosting on decision trees library with categorical features support out of the box for Python, R.
- ➌ **CatBoost — Yandex Technologies**  
tech.yandex.com > CatBoost  
CatBoost is a state-of-the-art open-source gradient boosting on decision trees library. Developed by Yandex researchers and engineers...
- ➍ **CatBoost — Overview of CatBoost — Yandex Technologies**  
tech.yandex.com > CatBoost > Documentation  
CatBoost is a machine learning algorithm that uses gradient boosting on decision trees. It is available as an open source library.
- ➎ **Newest 'catboost' Questions - Stack Overflow**  
stackoverflow.com > Catboost  
CatBoost is an open-source gradient boosting on decision trees library with categorical features support out of the box for Python, R.
- ➏ **CatBoost — Технологии Яндекса**  
tech.yandex.ru > CatBoost  
CatBoost использует более универсальный алгоритм, поэтому она подходит для решения и других задач. Преимущества CatBoost.
- ➐ **Яндекс открывает технологию машинного... / Хабрахабр**  
habrahabr.ru > Яндекс > Блог компании Яндекс > 333522  
CatBoost – это новый метод машинного обучения, основанный на градиентном

# Questions?

**Kirillov Stanislav**

Head of ML systems group @ Yandex

- › [catboost.ai](http://catboost.ai)
- › [github.com/catboost](https://github.com/catboost)
- › [twitter.com/CatBoostML](https://twitter.com/CatBoostML)
- › [t.me/catboost\\_en](https://t.me/catboost_en), [t.me/catboost\\_ru](https://t.me/catboost_ru)
- › [ods.ai](https://ods.ai) => slack => tool\_catboost channel