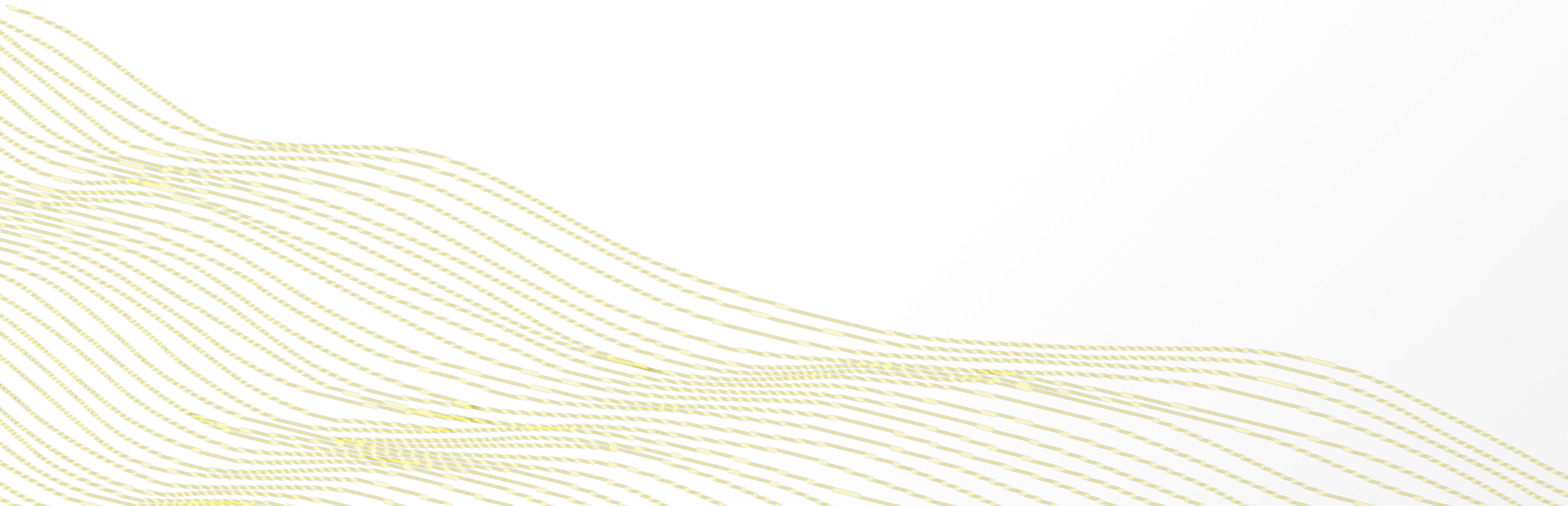
Gradient Boosted Machines

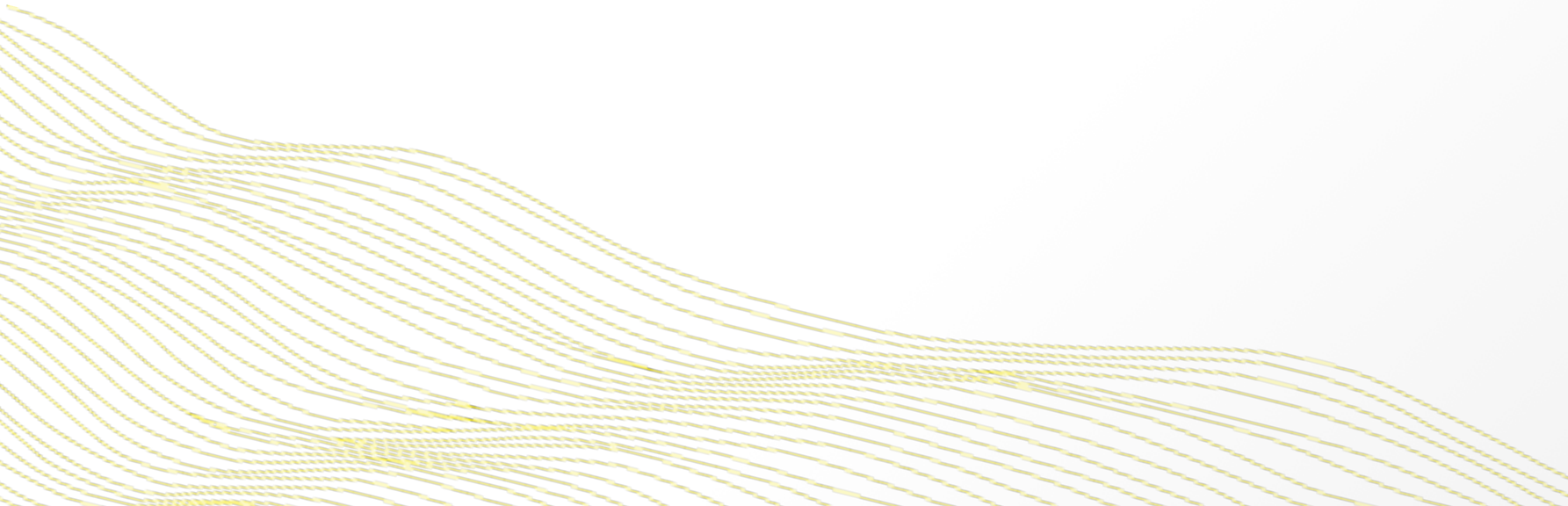


Agenda

Today's Talk

1. GBM Algorithm
2. GBM in H2O
3. Best Practices
2. GBM Demo

GBM Algorithm



What is a Decision Tree?

1. Separate the data into a left and right bucket by splitting on a feature.

2. Choose the point and feature as the one that causes the greatest reduction in the error function.

   Example: Squared error $(\text{predicted} - \text{actual})^2$

Advantage

- Robust
- Intuitive
- Non-linear

Disadvantage

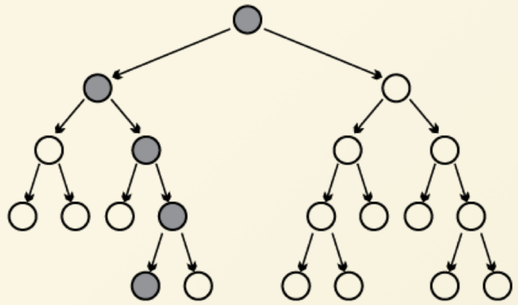
- Poor Accuracy

What is GBM?

1 Make Initial Prediction $f_0 = 0$

2 Calculate Residuals $r = y - f_0$

3 Build Regression Tree on Residuals

$h =$ 

Fits consecutive trees where each solves for the net loss of the prior trees.

4 Update $f_1 = f_0 + learn\_rate * h$

Results of new tree are partially applied to the entire solution.

When to use GBM?

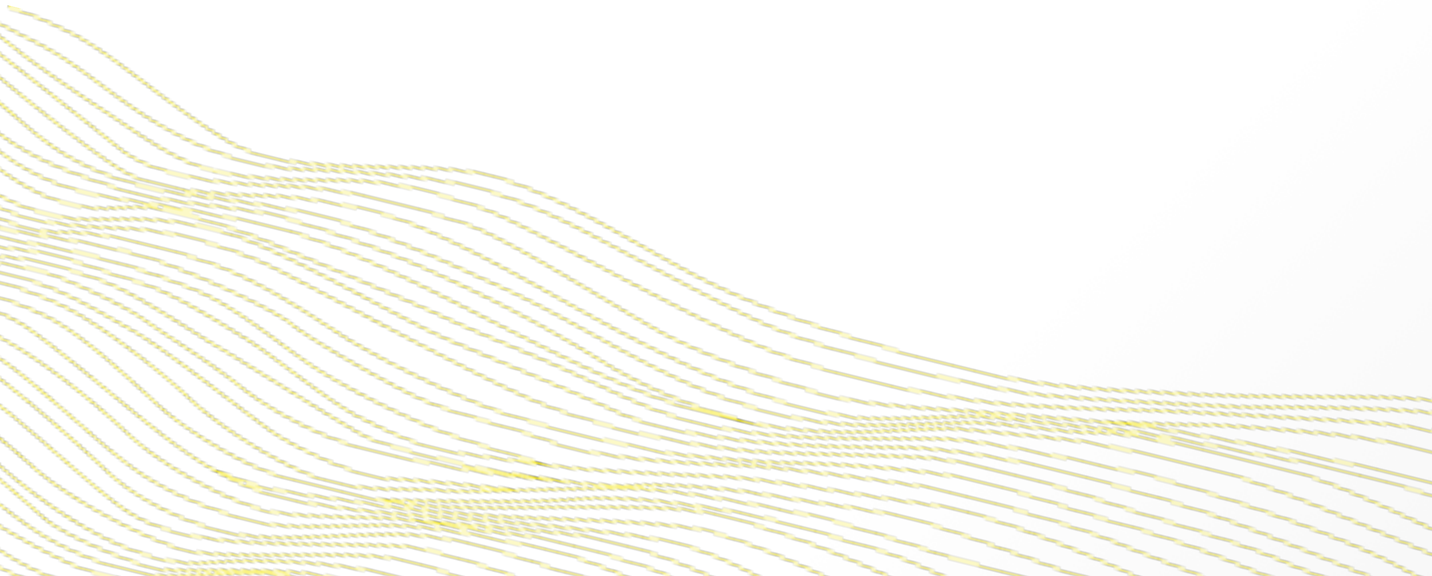
Advantages

- Often the best model
- Nonlinear models
- Robust to correlated features
- Robust to feature distribution
- Robust to missing values

Disadvantages

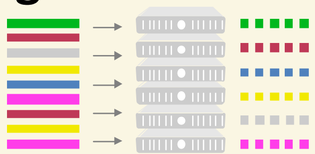
- Tends to overfit on the training set
- Sensitive to noise and outliers
- Several hyper-parameters

H2O Implementation of GBM



Scalable Implementation in H2O

1 Parallel Data Ingest



Data is stored in-memory on all cluster compute nodes

- **Rows are evenly distributed across the cluster**
- **Columns are stored separately and compressed**

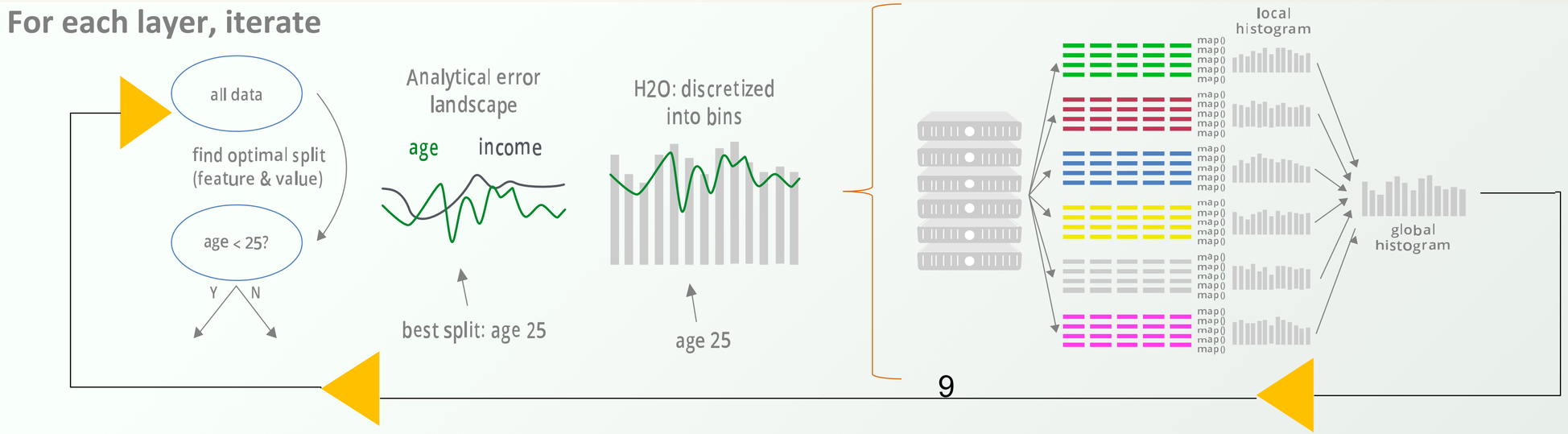
Basis for fine-grain Map/Reduce for histogram calculation

2 Distributed Tree Building via Fine-Grain Map/Reduce to find optimal split points of data layer by layer

Start with root node and build layers of tree nodes **[ILLUSTRATION BELOW]**

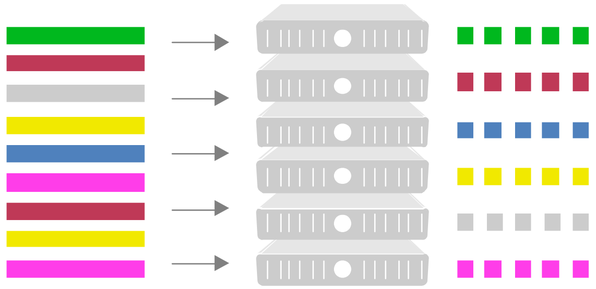
For each layer, repeat the following:

- **For a set of features, split the data at every possible split point**
- **Find the split that leads to best model improvement**
- **Use discretization to limit the number of potential splits**
  - **To find the split, local histograms are calculated on each node and then aggregated into a global histogram**
  - **From the global histogram, the best split column is chosen**

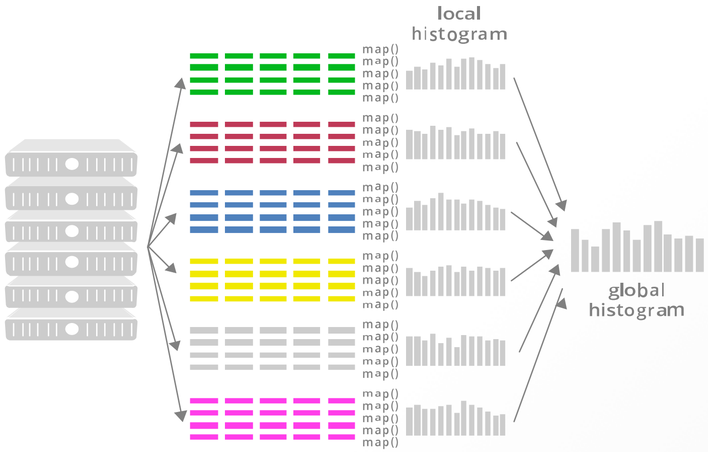


H2O.ai

Scalable Distributed Histogram Calculation



Parallel Parse into **Distributed Rows**

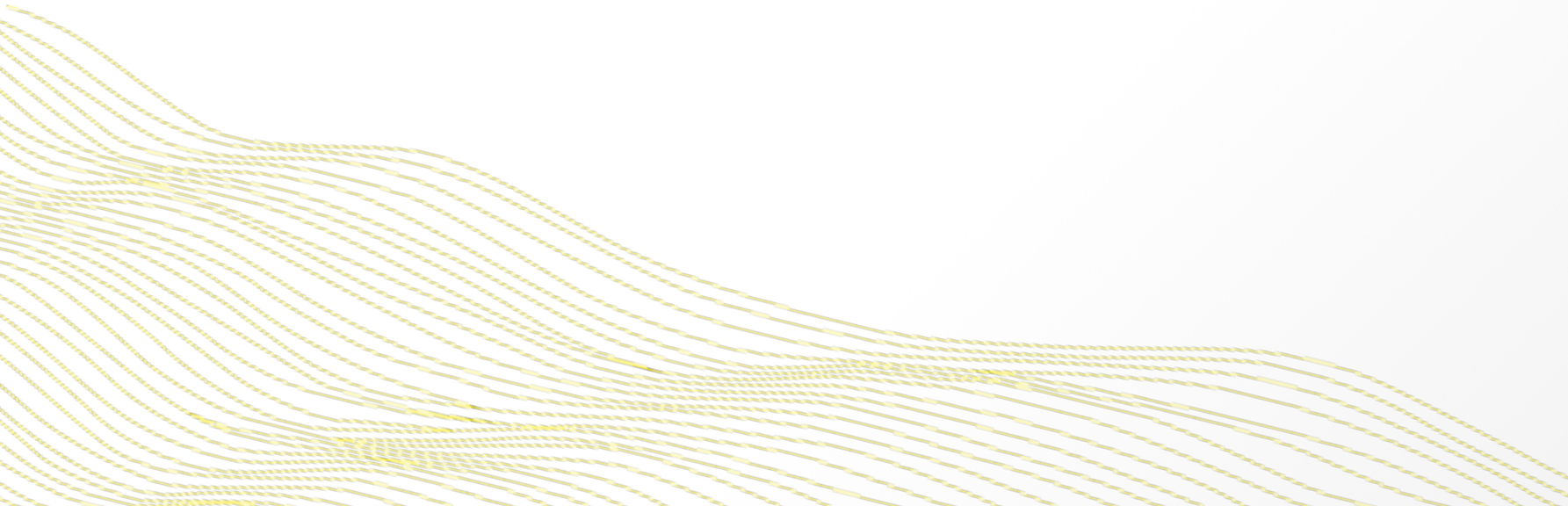


Fine Grain Map Reduce Illustration: Scalable Distributed Histogram Calculation for GBM

GBM Functionalities in H2O

- **Regression and Classification Models** Exponential distributions including Poisson, Gamma, and Tweedie are available in addition to Bernoulli, multinomial, and gaussian distributions.
- **Fast and CPU Efficient** Parallel and distributed computation across multiple nodes and many cores.
- **Grid Search** Hyperparameter optimization allows the user to run through many parameters before selecting the best models.
- **Early Stopping** The user can specify the metric and the incremental change in the metric as convergence. If additional trees no longer provide an increase in AUC prune the tree early.
- **Stochastic** User can specify the sample rate that the algorithm will sample the column and row by for better generalization.
- **Model Output** The model is exportable as Java code and if you find the model overfitted after a certain number of trees, it is easy to reduce the number of trees in a POJO before putting it in production without rerunning model build.

GBM: Best Practices



Establish a Validation Schema

Training Data	Model Tuning
Validation Data	Parameter Tuning
Testing Data	Final estimate of generalization error

Establish a Baseline

Train multiple models using different supervised learning algorithms with the default parameters.

```
CS  buildModel
```

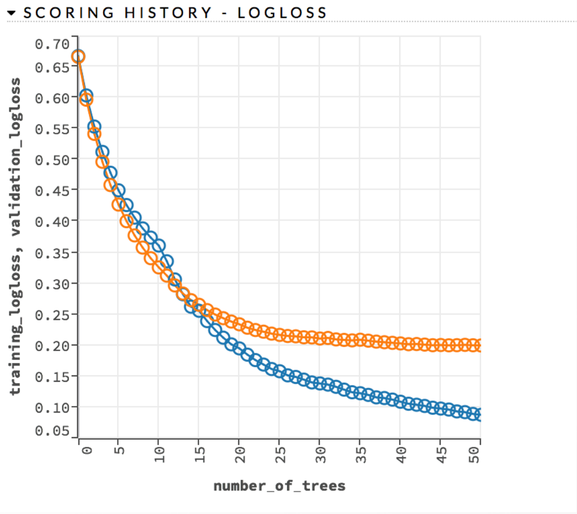
Build a Model

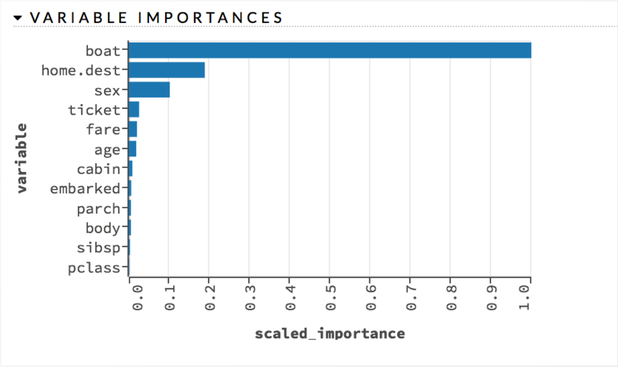
Select an algorithm

- ✓ (Algorithm)
- Aggregator
- Deep Learning
- Distributed Random Forest
- Gradient Boosting Machine
- Generalized Linear Modeling
- Generalized Low Rank Modeling
- K-means
- Naive Bayes
- Principal Components Analysis
- Word2Vec

Build Model

Inspect models in Flow



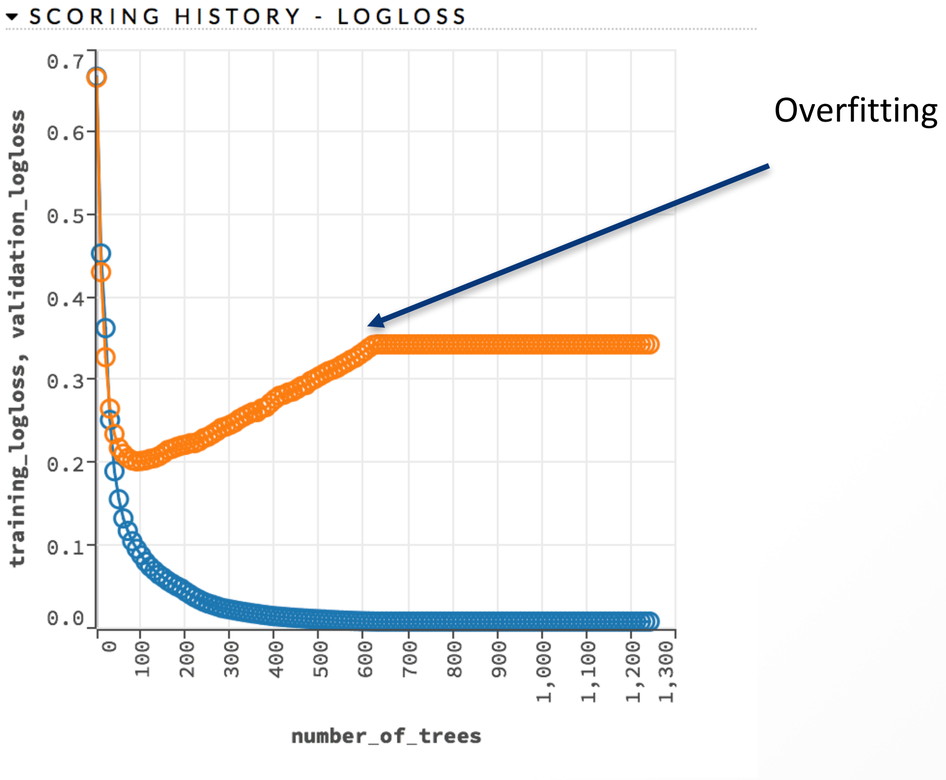


H2O.ai

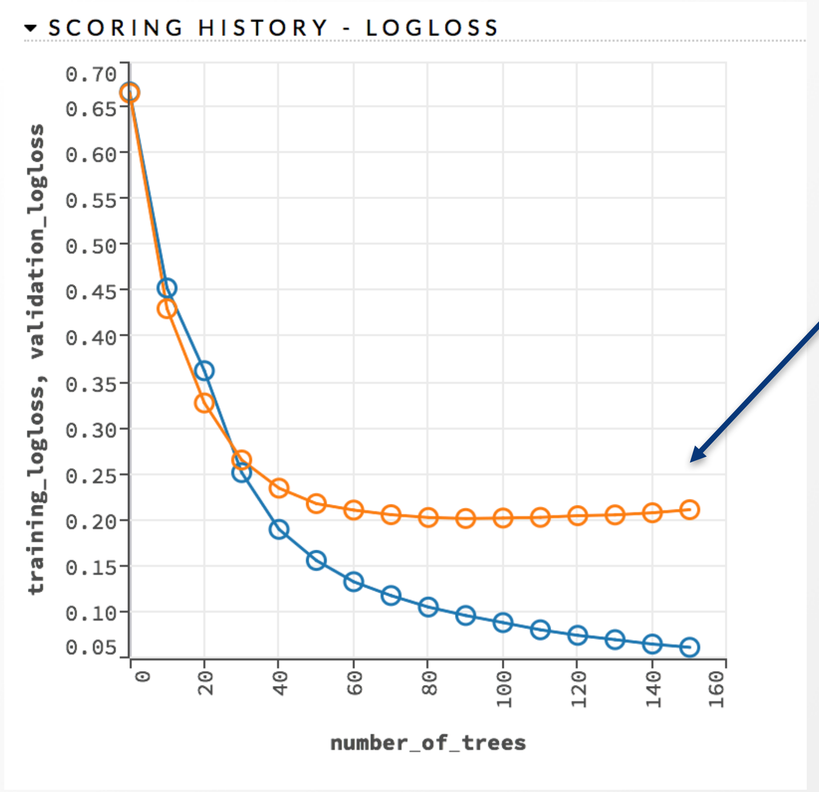
Use Early Stopping

Early stopping once the moving average (window length = 5) of the validation AUC doesn't improve by at least 0.1% for 5 consecutive scoring events

- `stopping_rounds = 5`
- `stopping_tolerance = 0.001`
- `stopping_metric = "AUC"`



Overfitting



Early Stopping

H2O.ai

Choose the Correct Distribution Function

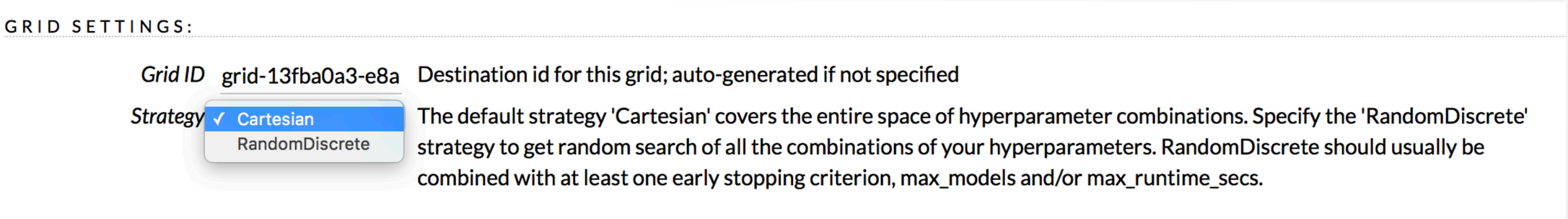
Distribution	Details
Gaussian	Squared error loss - sensitive to outliers
Laplace	Absolute loss - very robust to outliers
Huber	Combination of squared error and absolute loss - Robust to outliers
Poisson	Used for estimating counts
Gamma	Used for estimating total values
Tweedie	Used for estimating densities
Quantile	Used for estimating a specified percentile

Perform Hyperparameter Search

In R:

```
grid <- h2o.grid(hyper_params = list(max_depth = c(1:20)),
                 search_criteria = list(strategy = "RandomDiscrete", max_runtime_secs = 3600),
                 algorithm = "gbm“, ..)
```

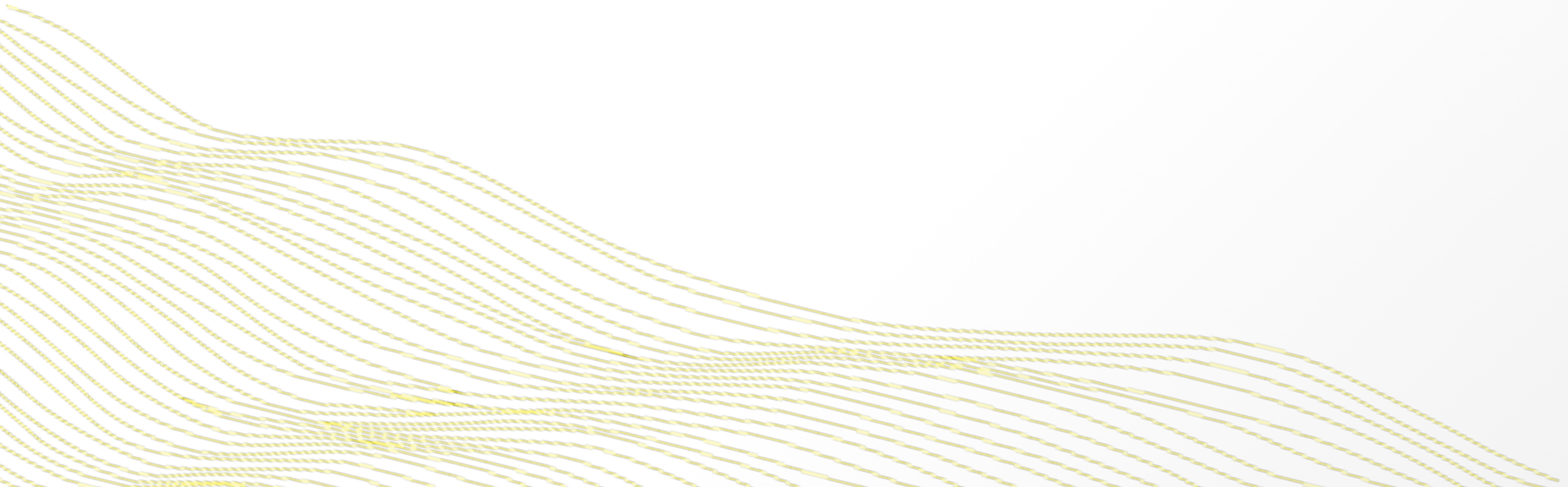
In Flow:



Perform Hyperparameter Search

Parameter	Definition	Typical Values
max_depth	Maximum tree depth	1 to 25
learn_rate	The rate at which the GBM learns while building the model Typically lower levels are better but require more trees	0.01 to 0.1
sample_rate	Row sampling rate per tree	0.2 to 1
col_sample_rate	Column sampling rate per split	0.2 to 1
min_rows	The minimum number of observations for a leaf in order to split	1 to 20
nbin_cats	Number of bins to be included in the histogram Ex: {A, B, C, D, E, F, G} splits to {A, B}, {C, D}, {E, F}, {G} when `nbin_cats = 4`	10, 100, 1000, 10000
histogram_type	What type of histogram to use for finding the optimal split point.	“UniformAdaptive”, “QuantilesGlobal”

GBM Demo



Resources

- Data: data/titanic.csv
- R Script: gbmTuning.R
- Python Script: gbmTuning.ipynb

Questions?