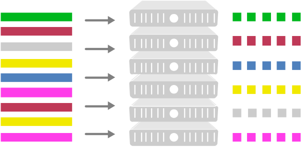
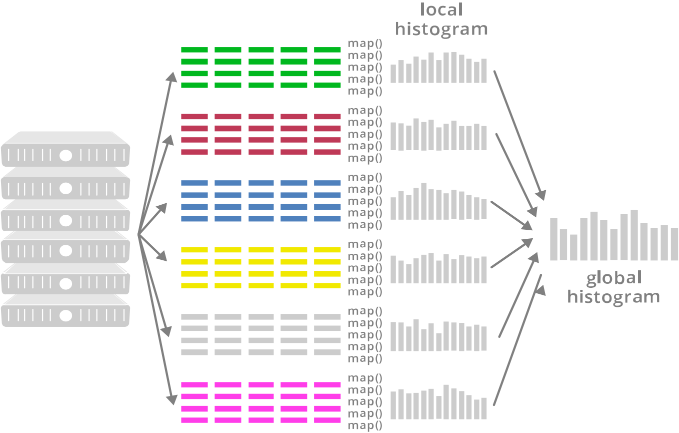
H_2O.ai

Parallel Data Ingest

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

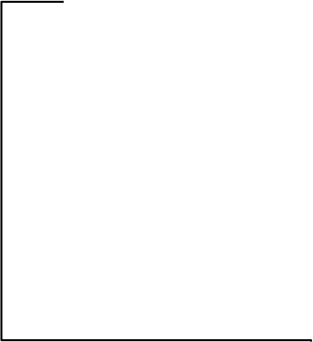


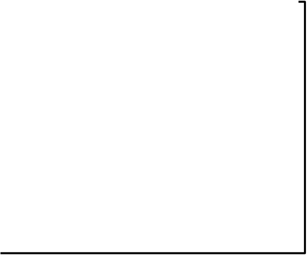

local histogram
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
map()
global histogram

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

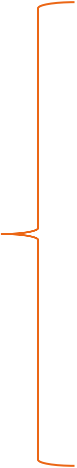
For each layer, repeat the following:

- **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**





For each layer, iterate



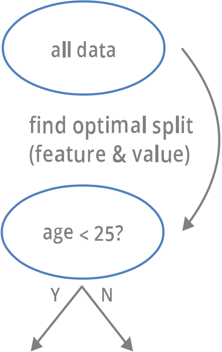
1

2

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



all data

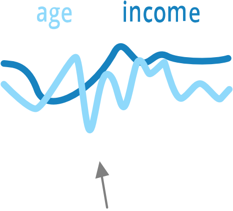
find optimal split
(feature & value)

age < 25?

Y

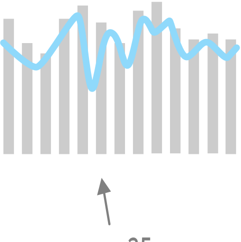
N

Analytical error landscape



best split: age 25

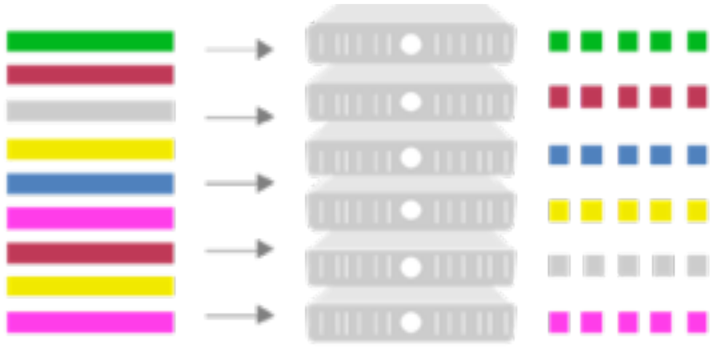
H2O: discretized into bins



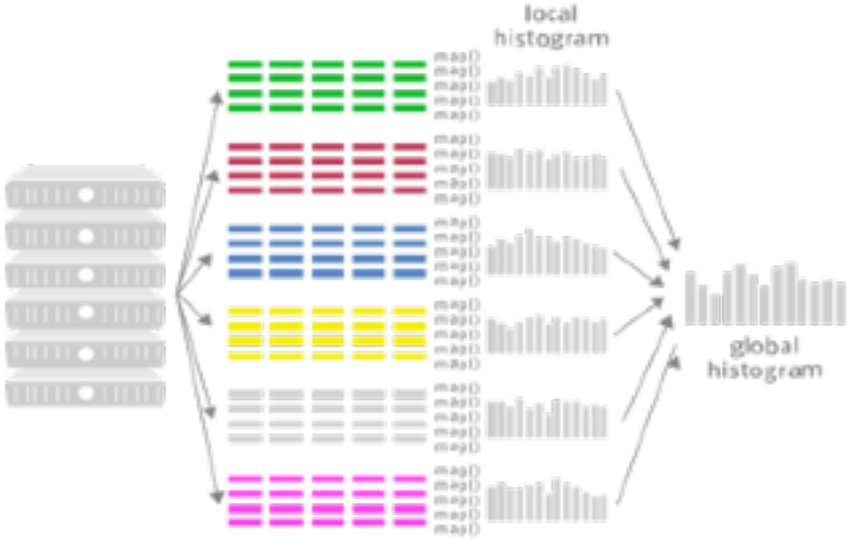
age 25

Scalable Implementation in H2O

Scalable Implementation in H2O



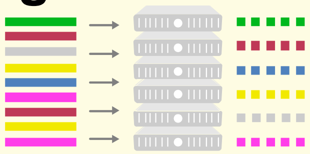
Parallel Parse into **Distributed Rows**



Fine Grain Map Reduce Illustration: Scalable Distributed Histogram Calculation for 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

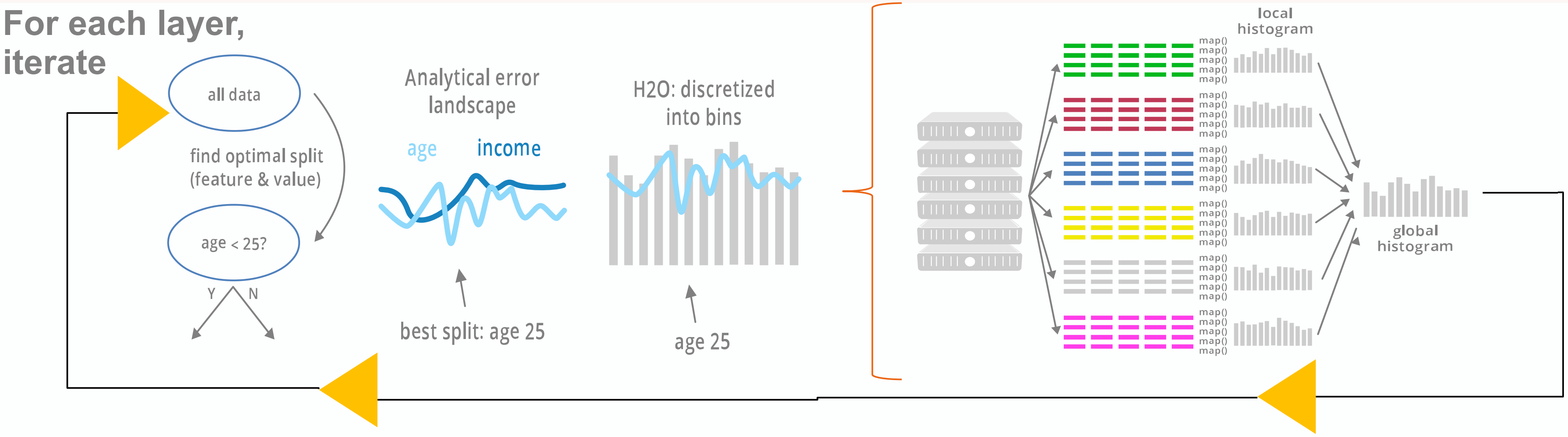
Basis for fine-grain Map/Reduce for histogram calculation

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



For each layer, iterate

all data
find optimal split (feature & value)
age < 25?
Y
N
Analytical error landscape
age
income
best split: age 25
H2O: discretized into bins
age 25
local histogram
map()
global histogram