




Decision tree with Map Reduce

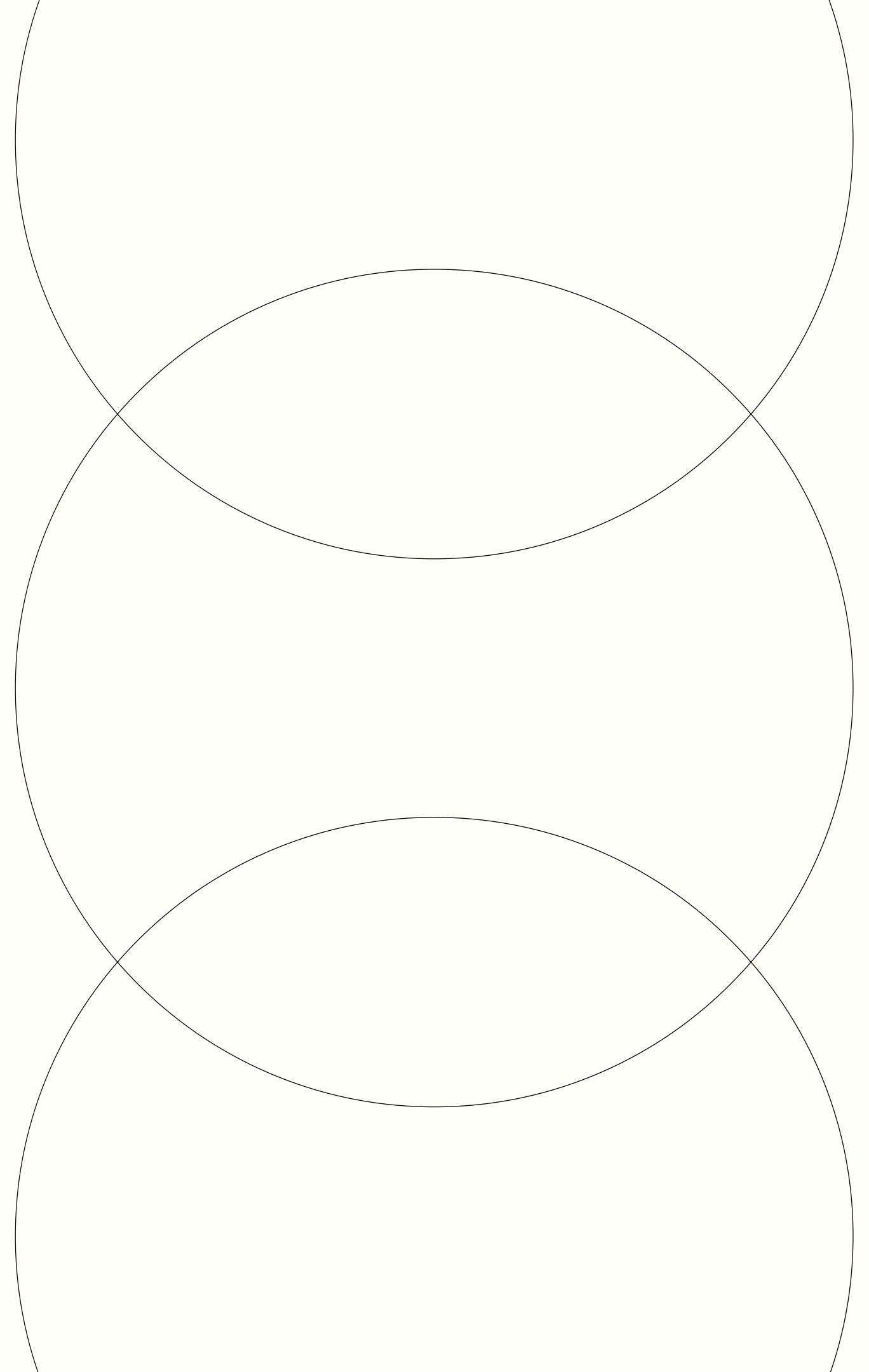
Yuting Mei

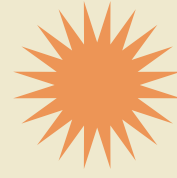




Introduction

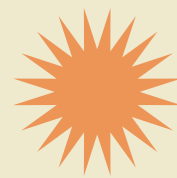
This is a project for implementing and designing decision tree classification model built from scratch and use map reduce strategy for a prediction task on 'Adult' dataset. Possible scenario for model is assumed and model performance under certain setting is measured





Methods

1. Tree structure design
2. Map Reduce Parallel



Methods – General Tree flow chart

Stopping Criteria (either one is meet) :

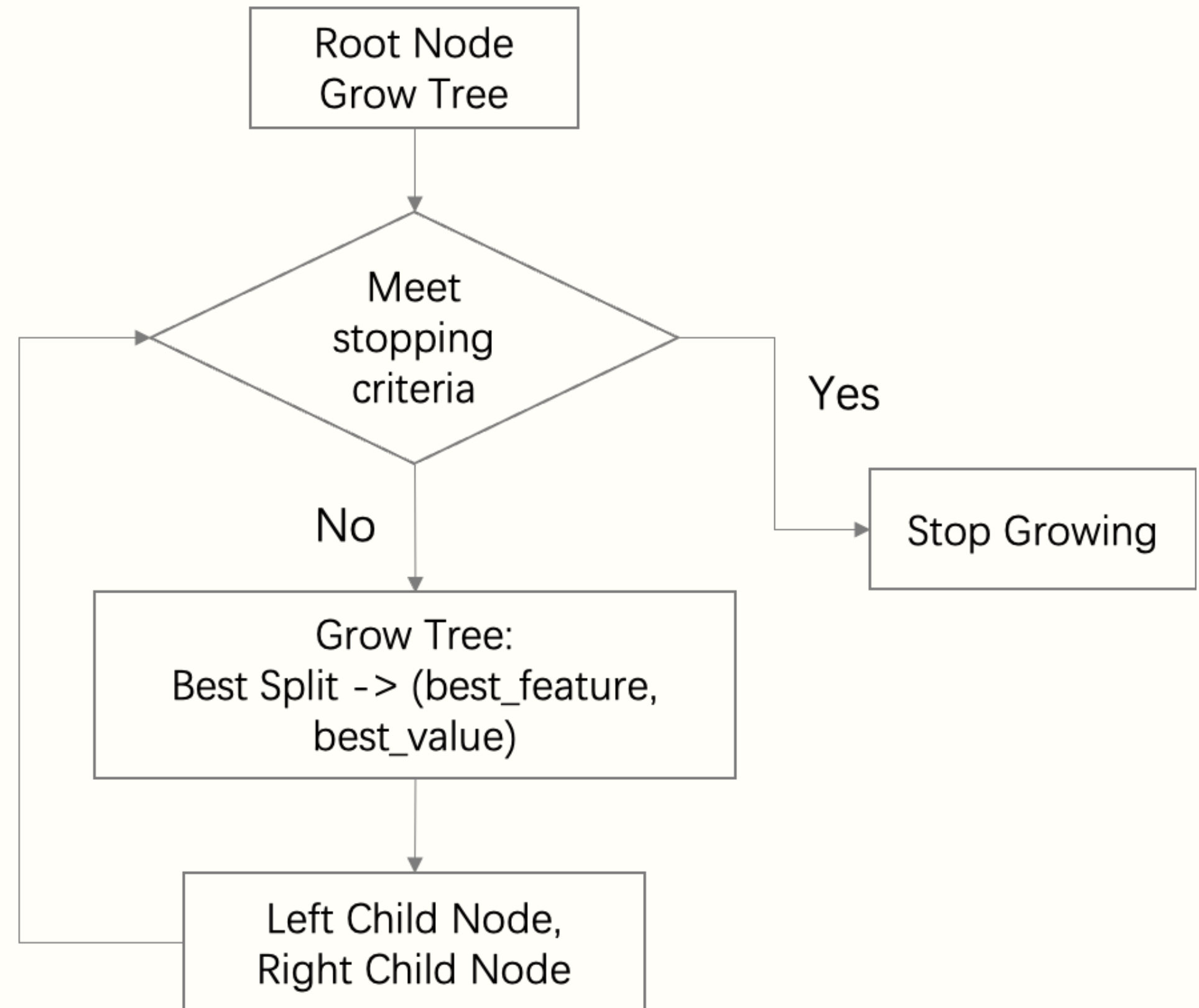
- Tree Depth < maximum depth
- Samples in Node \geq minimum number of samples

Goal:

Find best splits at each node which get informative and effective tree in unseen dataset

- maximize info gain(overall reduction in uncertainty)
- minimize impurity of each node with accurate label(homogeneous subsets)

Impurity measure:
gini, variance

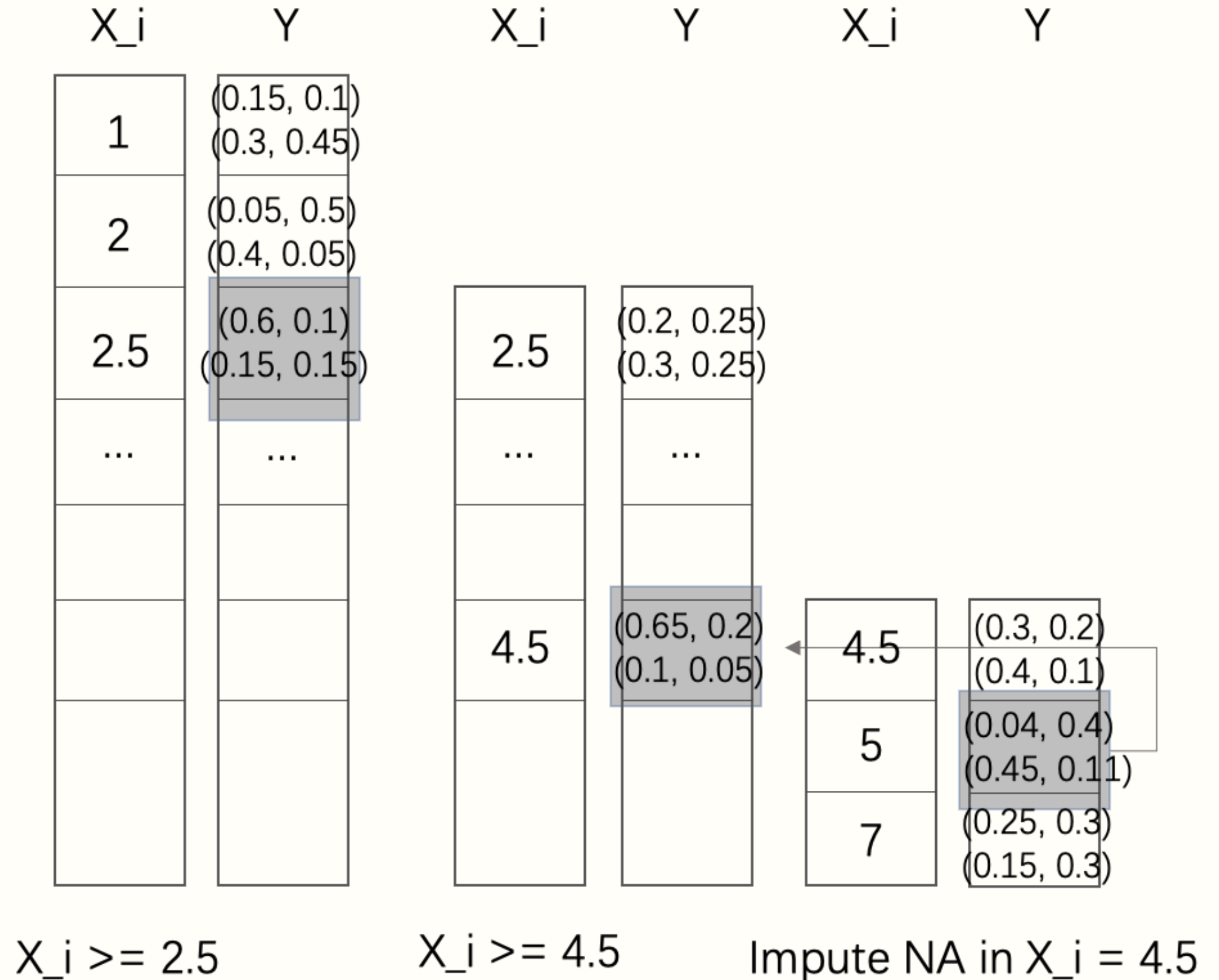


Methods – Missing value

Removing NA first ->

Missing value imputation strategy:
Inspired from C4.5

- Recursive imputation(bracketing)
- Mean imputation(not for categorical)
- Median imputation(not for categorical)



Methods – Best split

Partition strategy for different variable:
Categorical variable: Bool type

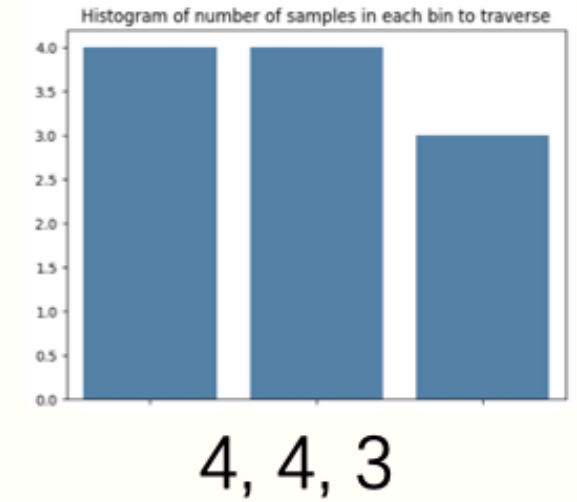
Discrete variable: moving average with two time window if under threshold else threshold

Continuous variable:

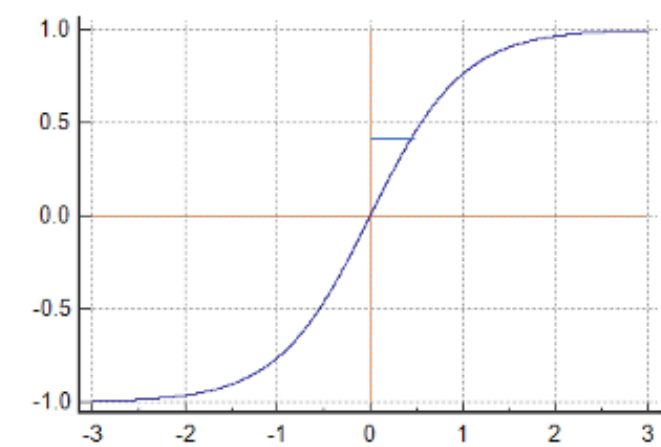
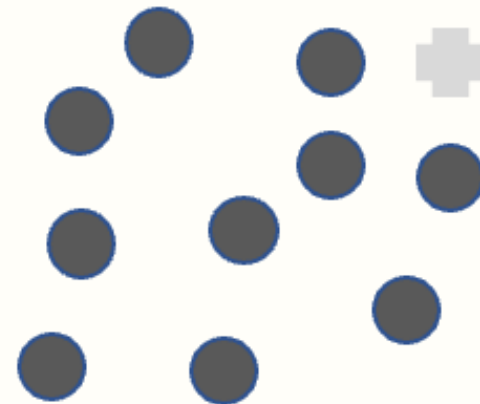
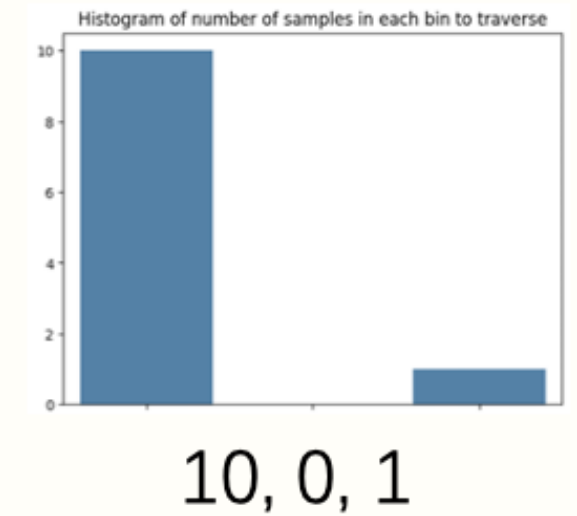
- Bin method for partition(bin numbers, outlier handing):
- Number of bins specify method:
 - User customized bin number
 - (X info based)Sturges' Rule ($\log_2 n + 1$)
 - (X info based)Scott's Rule (related to the standard deviation(σ) of the data)
 - (Y info based) activation functions for shrinking (tanh, logistic)

Bin method outlier(based on if outlier can give useful info for classification):

Toy candidates eg without outlier
[1,2,3,4,5,7,9,12, 10, 6, 7] ->



Toy candidates eg with outlier:
[100,2,3,4,5,7,9,5000, 1, 0, 3] ->



Methods – Map Reduce

subsampling Dataset

Map Reduce get traverse candidates for all features

Root Tree
Grow for subsamples
1 to s

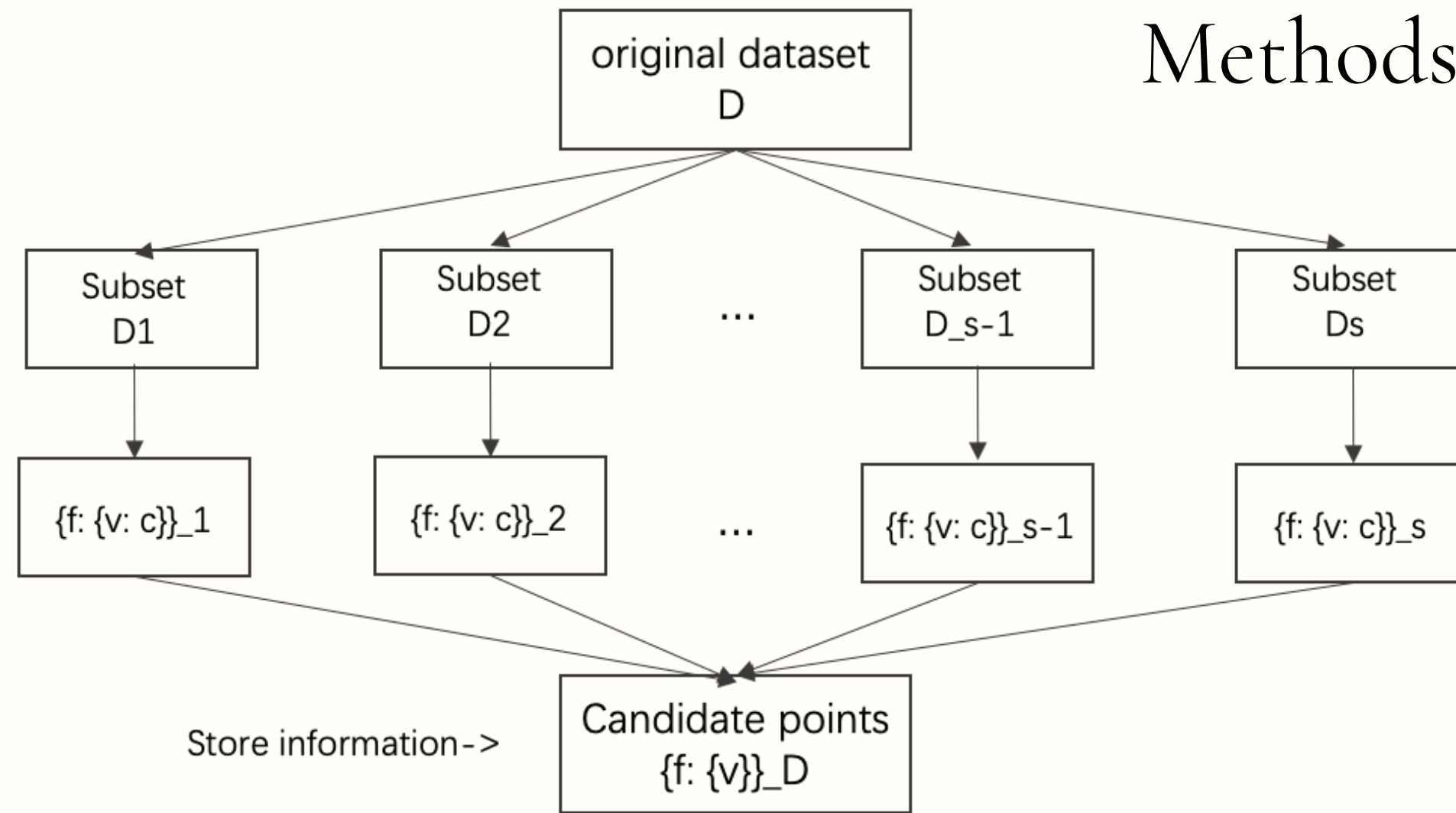
Traverse at
feature f, value v

Subsamples
Map reduce stat
P, L, R

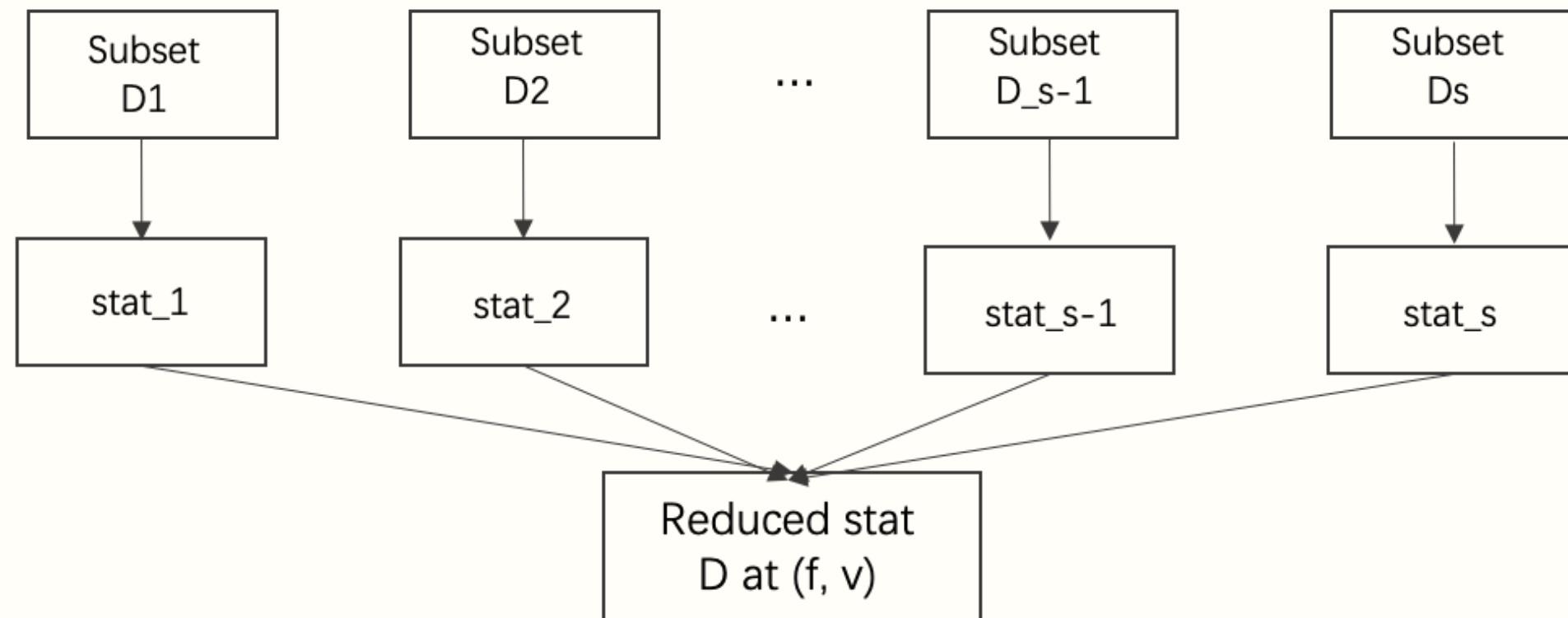
Information gain
f, v at D

Best split

Counter
frequency
of value ->



Possible scenario:
Large size
dataset(eg:25 GB)
Limited memory(eg:
5GB)
Multiprocessor(eg. 5),
 $S = 5$



<- $D_i: [(NP, SP, QP), (NL, SL, QL), (NR, SR, QR)]$ if
variance
else $[(TP, OP, 1P), (TL, OL, 1L), (TR, OR, 1R)]$
 $i = 1, 2, \dots, s$

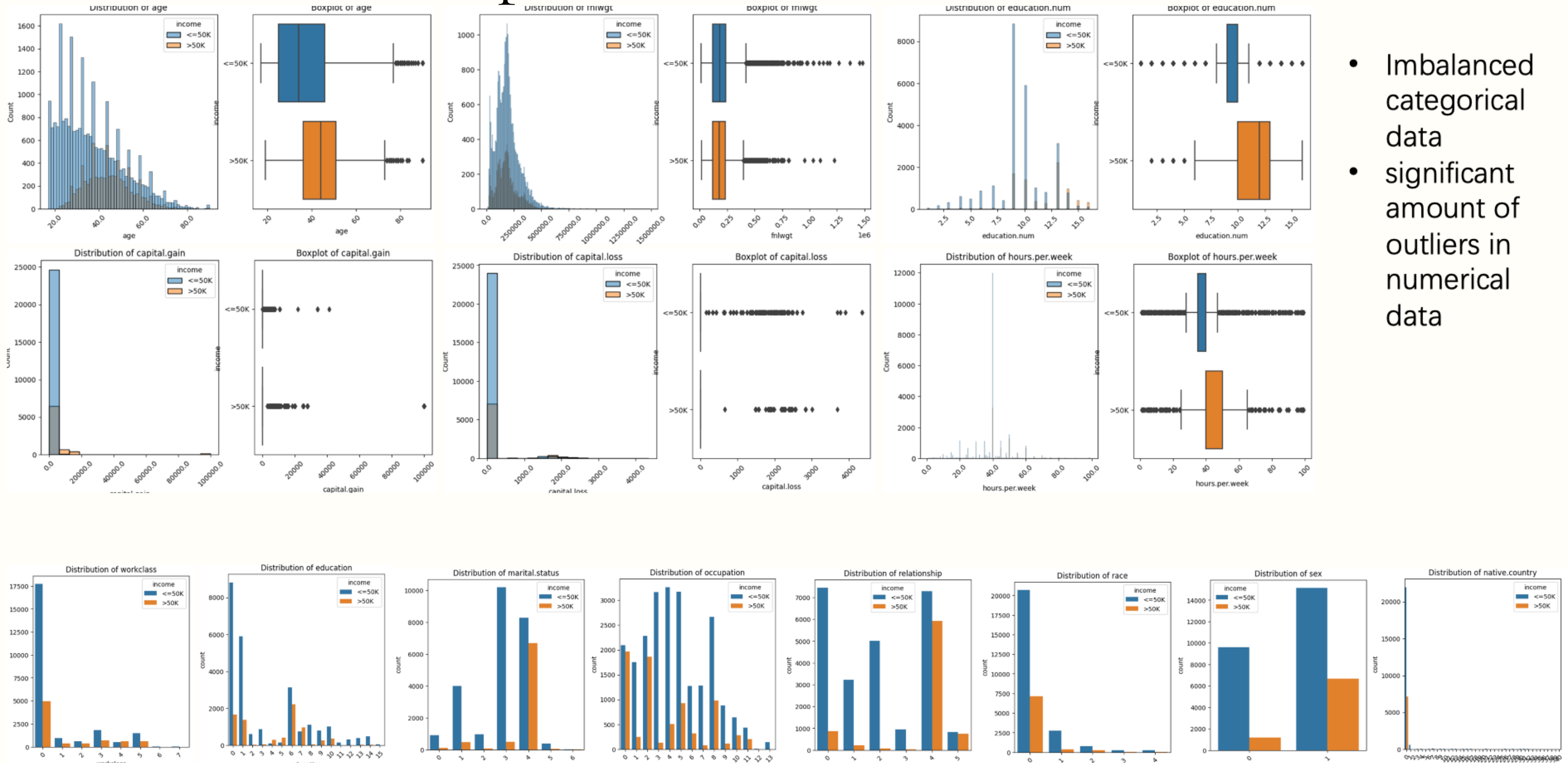
<- $D: [(NP, SP, QP), (NL, SL, QL), (NR, SR, QR)]$
or
 $[(TP, OP, 1P), (TL, OL, 1L), (TR, OR, 1R)]$

- Data explanation
- Model result



Result

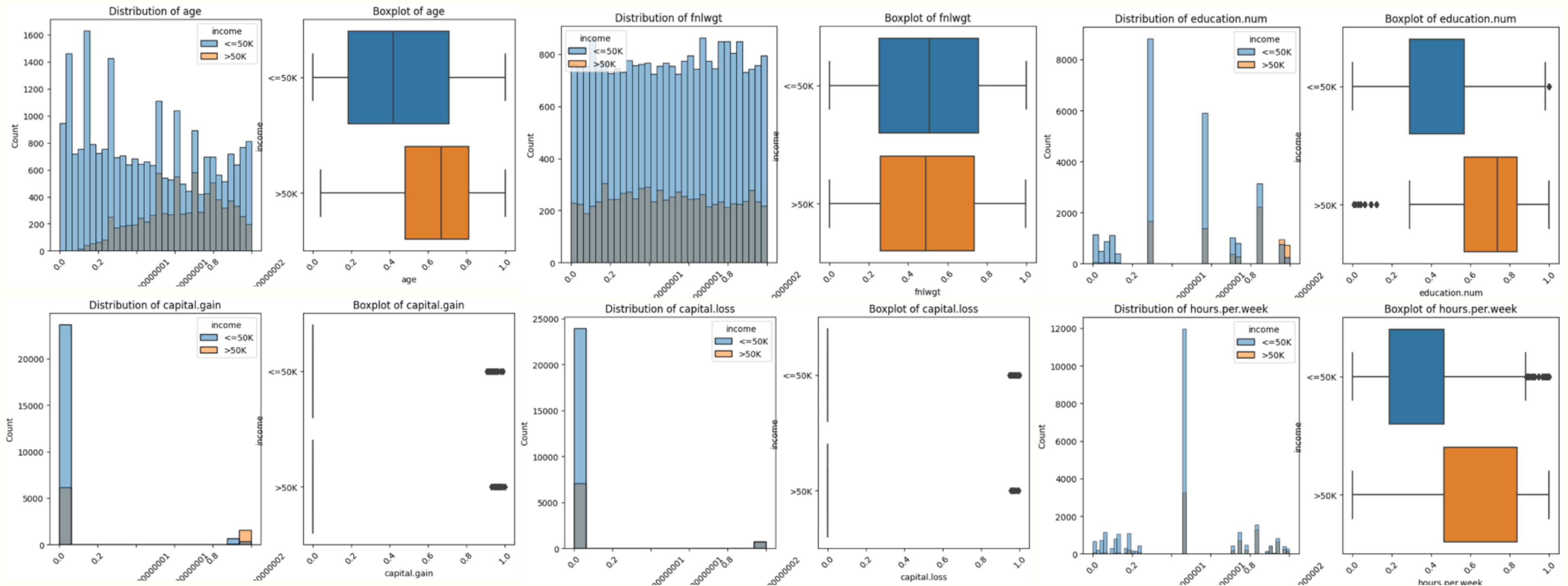
Result – Data explanation



- Imbalanced categorical data
- significant amount of outliers in numerical data

Result – Data explanation

- Quantile transformation for helping reduce impact on outliers



Result – comparison under certain settings of tree model

- Repeated stratified K fold
- Take categorical data as bool type
- Comparison of outlier handling

n_splits = 10, n_repeats = 2, traverse_threshold = 25, min_samples_split = 1850, max_depth = 11, info_method = 'variance', na_method = 'recursive', bins = 'tanh'

Parameter setting	Maximum precision score	Mean precision score	Time(second) per fold
Moving outliers	0.746	0.712 (std=0.026)	65.64
Without moving outliers	0.810	0.769(std = 0.022)	78.35

Result – comparison under certain settings of tree model

- Comparison of different bin number generation criteria

n_splits = 10, n_repeats = 1, traverse_threshold = 25, min_samples_split = 1850, \ max_depth = 11, info_method = 'variance', na_method = 'recursive' , outlier_ = False

Parameter setting	Maximum precision score	Mean precision score	Time(second) per fold
sturges	0.776	0.727 (std=0.032)	104.33
scott	0.789	0.769 (std = 0.022)	61.45
tanh	0.810	0.769 (std = 0.022)	78.35
User specified = 30	0.783	0.740 (std = 0.024)	64.90

Result – comparison under certain settings of tree model

- Comparison of different `traverse_threshold`(basically controls the least number of bins for continuous variable)

```
n_splits = 10, n_repeats = 1, min_samples_split = 1850, \
max_depth = 11, info_method = 'variance', na_method = 'recursive', bins = 'tanh', outlier_ = False'
```

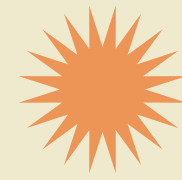
Parameter setting (Traverse_threshold)	Maximum precision score	Mean precision score	Time(second) per fold
15	0.737	0.699 (std=0.022)	63.24
25	0.810	0.769 (std = 0.022)	78.35
35	0.802	0.761 (std = 0.016)	83.06

Result – comparison of map reduce

- Comparison of model performance in map reduce using 1000 data points, due to limited sample size, the precision is not so solid

min_samples_split=600, max_depth=11, info_method=method, method = 'bin', num_bins = 15, threshold = 10

Subset samples number	Group size S	Mean computation time(seconds)	Mean precision	Impurity function
1000	50	110.418	0.853	variance
1000	100	118.424	0.857	variance
1000	500	128.934	0.842	variance
1000	50	154.352	0.848	gini
1000	100	165.996	0.854	gini
1000	500	176.675	0.851	gini




Conclusions & Limitations

- Slow computing speed in map reduce than in raw tree:

Time cost step by step in map reduce:

1. initialize Pool takes ~0.5 seconds in each iteration at feature f value v (the most time consuming step!);
2. map: ~ 0.05 seconds for getting statistics from subgroups;
3. reduce: $\sim e^{-5}$

- For tree model partitioning continuous variable, increasing partition of candidate points to traverse lead more accurate precision, but there's trade off between computational time



Conclusions & Limitations



Thanks for listening!