- Write the differences between ID3, CART, C4.5

Ans:

Name	Spliting Type	Attribute Type	Missing Values	Pruning Strategy
ID3	Information Gain	only Categorical Value	Don't handle missing values	No Pruning
CART	Delta Gini	Categorical & Numeric Value.	Handle missing values	Cost Complexity Pruning is used
C4.5	Gain Ratio	Categorical & Numeric Value	Handle missing values	Error based Pruning.

- Gini ; Binary value নিয়ে কাজ করে,
- Gini শুধু দুই ভাগে ভাগ করে,

- What is tree Pruning.

Ans: Pruning means removing specific branches to benefit the whole tree. Pruning tree methods address the ~~Pro~~ Problem of overfitting the data.

- Training data তে noise থাকলে decision tree তে ~~ano~~ anomalies দেখা যায়,
- Tree Pruning করলে, Performance বাড়ে,
- Tree Pruning is being done to increase the efficiency.
- দুই ধরনের Pruning হয়, a. Pre Pruning b. Post Pruning

⊞ Define Pre-Pruning

Ans: A tree is Pruned by halting its Construction early. Upon halting the node becomes a leaf.

⊞ Define Post-Pruning

Ans: Removing sub-trees from a full grown tree.

=> Pre-Pruning মনে tree বানাচ্ছি। কোন একটা node এর নিচে আর নামবো না। Leaf হিসেবে রাখবো।

=> Pre-Pruning এর বেলায় Construction থামিয়ে দেওয়া হয়।

=> Pre-Pruning এ leaf ; most frequent class কে ধরবে।

=> Pre-Pruning এ Probability distribution হিসেবে ধরা হয়।

=> Pre-Pruning এর বেলায় threshold maintain করা হয়ে থাকে।

⊞ Threshold যদি high হয়, তাহলে tree তে সব leaf হয়ে যাবে, তখন tree টা over simplified হয়ে যাবে,

⊞ Threshold যদি low হয়, তাহলে tree তে খুব সামান্য পরিমান পরিবর্তন আসবে।

▣ Post Pruning -এর বেলায় Pre Pruning থেকেও বেশি Computation করতে হয়, কিন্তু Post Pruning most reliable.

▣ Define Repetition

Ans: It occurs when an attribute is repeatedly tested along a given branch of the tree.

▣ Define Replication

Ans: Duplicates subtrees exist within the tree.

▣ Pruning -এর কিছু টেকনিক আছে, মূলত ৩টা টেকনিক,

a. Pruning by Cost Complexity

b. Pruning set

c. Pessimistic Pruning

▣ Pruning by Cost Complexity → এটা Post-Pruning approach

▣ Pruning set →

Pruning By Cost Complexity

- CART [Classification & Regression Trees] / Detta Gini Cost Complexity approach ব্যবহার করে।
- Cost Complexity হলো মূলত একটা function.
- ~~$f(x)$ = Number of decision tree in the leaf node~~.
- $f(x)$ = Number of leaf node in decision tree + Error rate.
- Error rate বলতে বুঝায় miss classify করা।
- Error rate should be equal or less.
- Cost Complexity এর সময় দুইটা জিনিস Compare করা হয়, সেগুলো হলো

  a. Present Error Rate

  b. Leaf node এর সংখ্যা,
- ছোট tree Pruning By Cost Complexity এর জন্য better
- Define Error Rate

Ans. The Percentage of the instances misclassified by the tree.

Pruning Set:

- Labled instance এর পরিবর্তে Cost Complexity estimate করা হয়।
- এই algorithm Set of Pruned trees generate করে।
- $\boxed{D_L} = \{DT_1, DT_2, DT_3, \ldots, DT_N\}$ ← Set of Pruned trees.

এই set এ minimum error rate, minimum number of leaf এবং Maximum accuracy থাকে।

Pessimistic Pruning:

- এটা মূলত C4.5 algorithm এ use করা হয়।
- এর ক্ষেত্রে

$f(x)$ = Number of leaf node in the decision tree + Error rate হিসেবে Recall ব্যবহার হয়।

- Pessimistic Pruning; Pruned set ব্যবহার করে না, এটা training set ব্যবহার করে, কারণ training set unique.

- Define Recall

Ans.

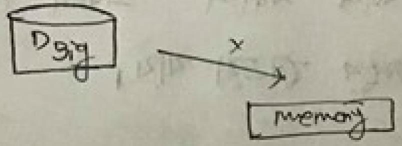
Recall means true Positive rate

প্র Write the imporotance of Scalability & Decision tree Induction

Ans.

Most often, training data would not fit in the memory. ID3, C4.5, CART does well for small data sets.

For this reason tree construction becomes inefficient. As swapping the the training instances in & out of main & cache memory.



Big data একবারে memory তে আটানো যায় না। ফলে decision tree induction কঠিন হয়, এজন্য Scalable approach হিসেবে BOAT এবং Rainforest ব্যবহার করা হয়,

Rain Forest:

প্র Divide the big data into small chunks.

প্র এর AVC - Set [Attribute Value Class label] Maintain করে,

$D_{Big} = \{x_1, x_2, x_3, x_4, --- x_n\}$

↓ এখানে x গুলো হল chunk / subdata / Portion of the data.

Attribute	Class Attribute		
	Yes	No	
Sunny	2	3	= 5
Overcast	4	0	=4
Rain	3	2	=5
Total	9	5	

9 + 5 = 14

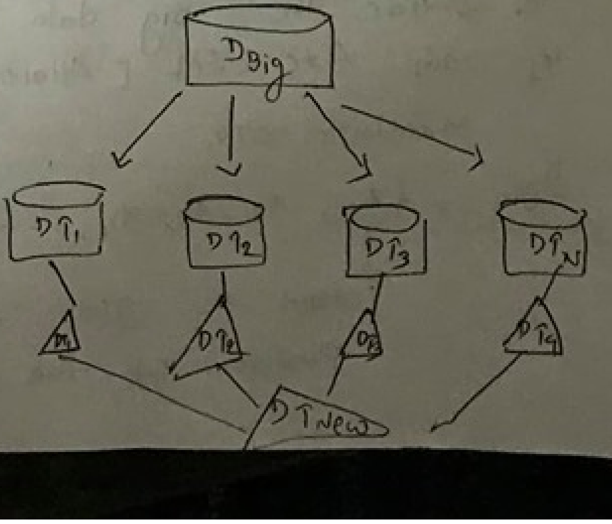
=> Ave-set এ মূলত Attribute এর ভেল্যুগুলোর class number থাকে, ফলে সহজে প্রত্যেক chunk এর Prior Probability বের করে নেওয়া যায়।

=> এই Process এ

$$D_{Big} = \{x_1, x_2, x_3, \dots, x_n\}$$

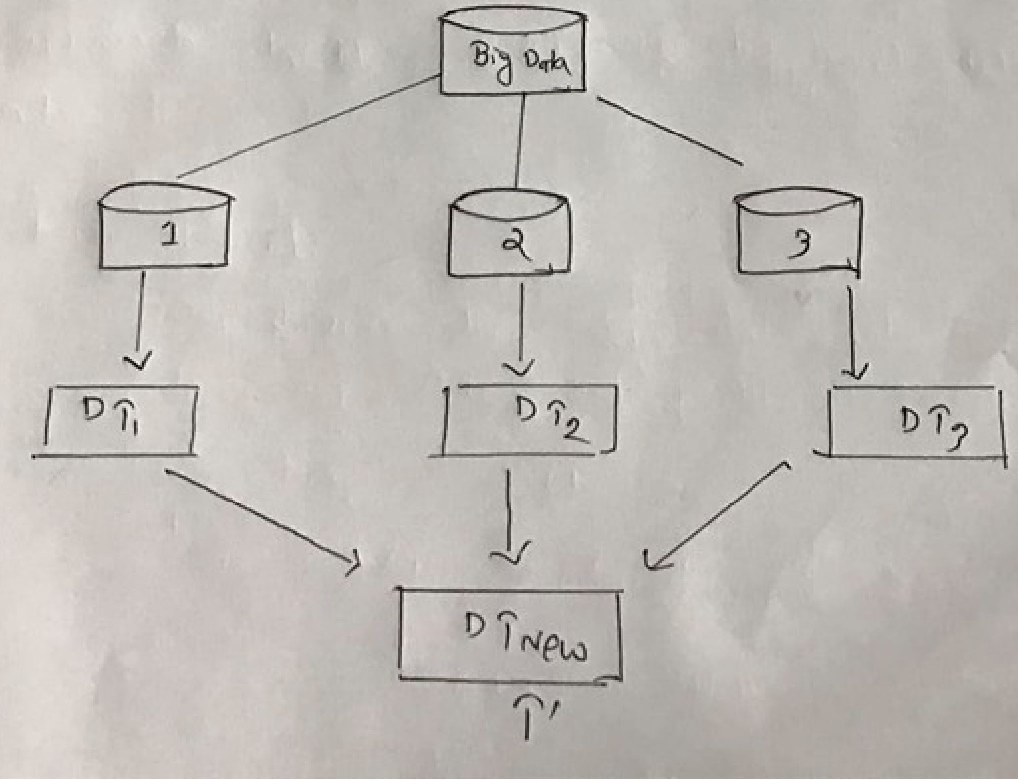
বারবার Update করতে হয়, তাই Rainforest অনেক slow Process.

BOAT: [Bootstrapped Optimistic Algorithm For Tree Construction]



- BOAT এ মূলত Big data ছোট chunk এ বিভক্ত হয়। যা DT_1, DT_2, DT_3, ..., DT_n. এগুলো মূলত AVC-set-
- এই AVC-set থেকে merge করে decision tree বানানো হয়।
- Tree বানানো হয়ে গেলে AVC-set flash করা হয়,
- BOAT হল মূলত Incremental Updates এর জন্য ভাল,
- BOAT Training data তে insertion এবং update করে, এজন্য ইহা tree reconstruct করে না,
- Gini index ; Boat use করে.

Boat এর figure →



□ Write the advantage of BOAT over Rain Forest

An.

a. New data input.

b. Update the tree

c. No need to start from scratch.

d. Incremental update.

e. It guarantees accuracy.