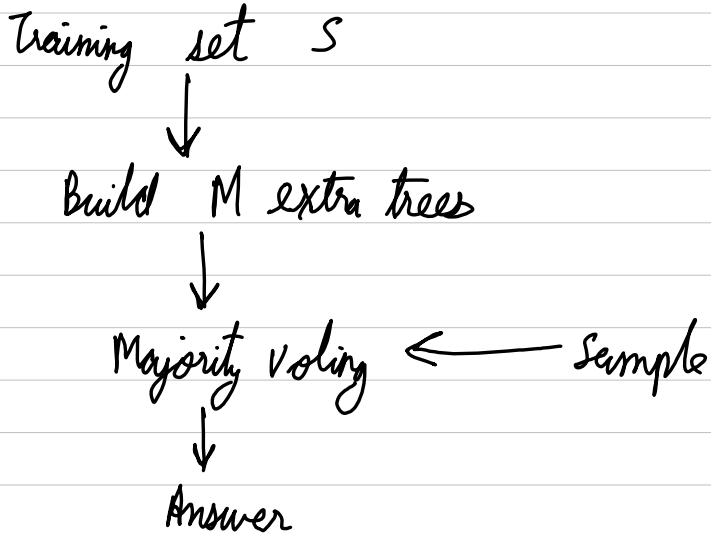
Extra Trees Ensemble

Similar to random forest. It can achieve as-good or better performance

Unlike random forest, it fits the entire training dataset

Like random forest it will randomly sample the features at each split point.

Unlike random forest which chooses the split points greedy, extra trees choses the split points randomly.



Build an extra tree →

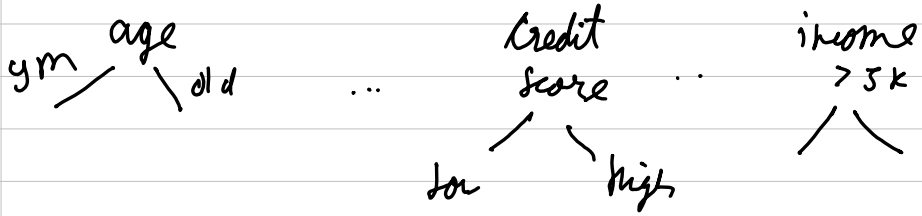
Training set S

↓

Select k random features (eg age, credit score)

↓

Generate k splits (*)



↓

select the best feature for the split

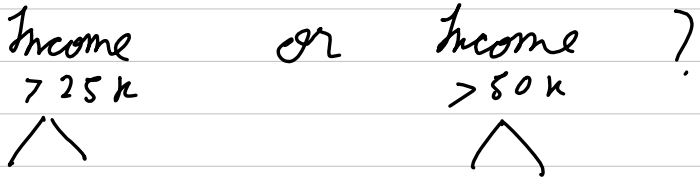
↓

split dataset

↓

Recursively apply the algorithm

How to pick a split?



In random forest, we used to select the split in greedy manner checking each combination.

However in extra trees this is done randomly on basis of certain rules →

If the attribute is catagorical

1. Let the set of possible values be A
2. Compute subset of values occuring in the training set S as A_S

Note that although only set A_S is present in the training set provided for the split during evaluation other elements may come.

Note S is not entire data. It is the subset of data provided to the recursive algorithm

3. Randomly draw a proper non-empty subset A_1 from A_S and subset A_2 from $A \setminus A_S$

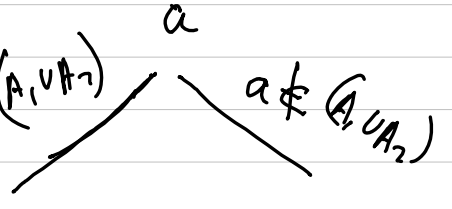
Proper subset → B is proper subset of A if A contains at least one element that is not in B

eg $A = \{1, 2, 3\}$

$B = \{2, 3\}$ } Proper
$B = \{2\}$ }
$B = \{1, 2, 3\}$ → Not proper

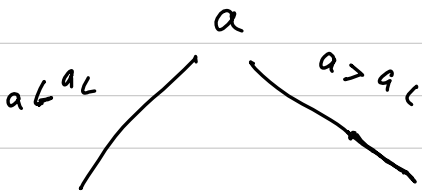
$A \setminus B$ means $x \in A$ & $x \notin B$

4. Return the split $a \in (A_1 \cup A_2)$



If a is numerical

1. Compute the maximal & minimal value of a in S a^S_{min} & a^S_{max}
2. Draw a random cutpoint uniformly in $[a^S_{min}, a^S_{max}]$ as a_c
3. Return split $a < a_c$



The rationale is that such randomized splits reduce the variance more strongly than the weaker randomization schemes used by the other methods.

The usage of the entire learning sample is done in order to reduce the bias.