



TREE BASED MODELS



- Lucy Xu



QUIZ

- x What is Gini?
- x What is entropy?
- x What's the difference and similarity between Random Forest and XGBoost?
- x What's key parameters to train a tree based model?



TOPICS

- X Decision Tree
- X Bagging and Random Forest
- X Boosting Tree



DECISION TREE

TYPES OF DECISION TREE



“Decision tree is a type of supervised learning algorithm, which is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) ”

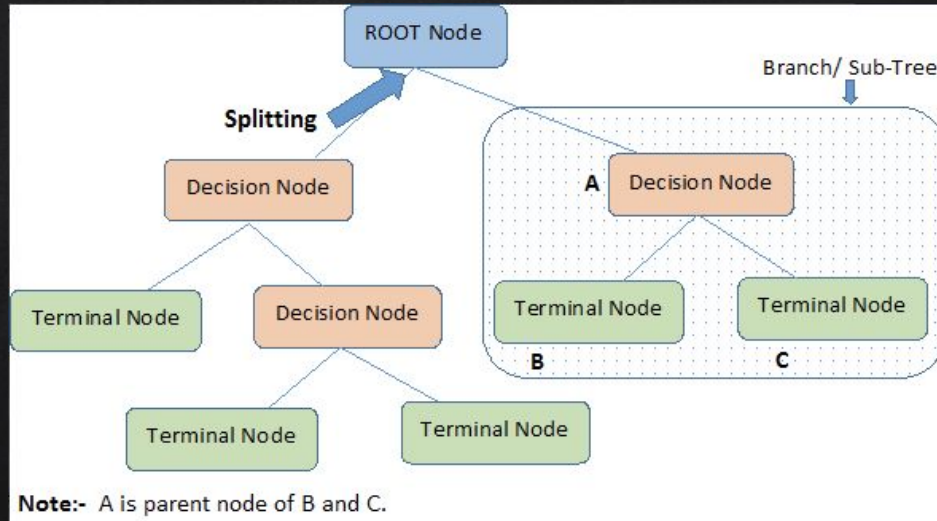
- ✗ Classification Trees: Decision Tree which has categorical target variable then it called as categorical variable decision tree.
- ✗ Regression Trees: Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

Therefore, Decision Tree also called Classification tree And Regression Tree(CART)

IMPORTANT DEFINITIONS RELATED TO DECISION TREE



- ✗ **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- ✗ **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- ✗ **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
- ✗ **Leaf/ Terminal Node:** Nodes do not split is called Leaf or Terminal node.
- ✗ **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
- ✗ **Branch / Sub-Tree:** A sub -section of entire tree is called branch or sub-tree.
- ✗ **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.



REGRESSION TREE VS DECISION TREE



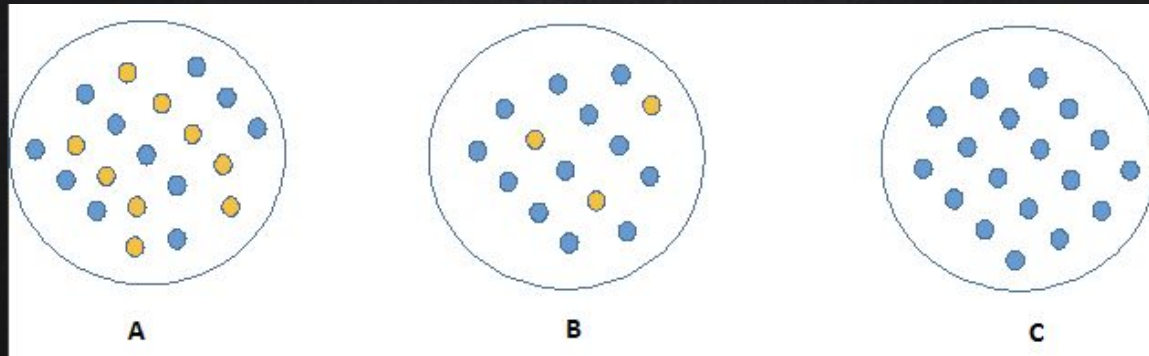
1. Target
2. Prediction
3. Both the trees divide the predictor space (independent variables) into distinct and non-overlapping regions.
4. Algorithm – Both the trees follow a top-down greedy approach known as recursive binary splitting.
5. This splitting process is continued until a user defined stopping criteria is reached.
6. Pruning is used to prevent overfitting

BASICS ALGORITHMS TO SPLIT(GROW) TREE



x Purity Algorithms

- Gini: $P^2 + (1-p)^2$; Gini Impurity = $1 - \text{Gini}$
- Entropy: $-p \cdot \log_2(p) - (1-p) \cdot \log_2(1-p)$



x Chi-square

x Reduction in Variance (MSE/RMSE)

RECURSIVE BINARY PARTITION



1. Initialize A with the entire data set.
2. Apply the partitioning algorithm to split A into two subpartitions, A 1 and A 2 . (Basic algorithms embedded here for best splitting)
3. Repeat step 2 on subpartitions A 1 and A 2 .
4. The algorithm terminates when no further partition can be made that sufficiently improves the homogeneity of the partitions.



PARAMETERS CONTROL TREE GROW

- ★ Minimum samples for a node split
- ★ Minimum samples for a terminal node (leaf)
- ★ Maximum depth of tree (vertical depth)
- ★ Maximum number of terminal nodes
- ★ Maximum features to consider for split



2.

BAGGING AND RANDOM FOREST

KEY DEFINITIONS



Ensemble “Forming a prediction by using a collection of models.”

Bagging “A general technique to form a collection of models by bootstrapping the data.” Also called bootstrap aggregation

Random forest “A type of bagged estimate based on decision tree models.”

Variable importance “A measure of the importance of a predictor variable in the performance of the model.”

BAGGING ALGORITHM



Bagging is like the basic algorithm for ensembles, except that, instead of fitting the various models to the same data, each new model is fitted to a bootstrap resample.

1. Initialize M , the number of models to be fit, and n , the number of records to choose ($n < N$). Set the iteration.
2. Take a bootstrap resample (i.e., with replacement) of n records from the training data to form a subsample and (the bag).
3. Train a model using and to create a set of decision rules.
4. Increment the model counter. If $m \leq M$, go to step 2.
5. Sum average all $f(x)$ (M trees)

RANDOM FOREST



This is an advanced algorithm add the bootstrap predictors for each tree split.

Major Tuning Parameters:

- ☐ Number of trees build
- ☐ Sample size of Observation
- ☐ Sample size of inputs
- ☐ Minimum samples for a node split
- ☐ Minimum samples for a terminal node (leaf)
- ☐ Maximum depth of tree (vertical depth)
- ☐ Maximum number of terminal nodes
- ☐ Maximum features to consider for split



3.

BOOSTING TREE

BOOSTING TREES



Boosting trees, are a series of models, in which each successive model seeks to minimize the error of the previous model.

Several isoforms of boosting trees:

- Adaboost
- Gradient boosting
- Stochastic gradient boosting (XGB)

Normally boosting tree need more intensive hyperparameter tuning



ADVANTAGE OF XGB

- Regularization
- Parallel Processing
- High Flexibility
- Handling Missing Values
- Tree Pruning
- Built-in Cross-Validation
- Continue on Existing Model



XGB HYPER-PARAMETERS

- eta – Analogous to learning rate in GBM
- min_child_weight
- max_depth [default=6]
- Max_leaf_nodes
- gamma [default=0]
- max_delta_step [default=0]
- subsample [default=1]
- colsample_bytree [default=1]
- colsample_bylevel [default=1]
- lambda [default=1]
- alpha [default=0]
- scale_pos_weight [default=1] – A value greater than 0 should be used in case of high class imbalance as it helps in faster convergence.



RECOMMENDED READINGS

<https://online.stat.psu.edu/stat555/node/100/>

<https://www.slideshare.net/ShangxuanZhang/xgboost>

<https://towardsdatascience.com/machine-learning-part-18-boosting-algorithms-gradient-boosting-in-python-ef5ae6965be4>

<https://towardsdatascience.com/decision-tree-in-python-b433ae57fb93>

Package of Trees detail parameters

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier>

<https://xgboost.readthedocs.io/en/latest/parameter.html>



REFERENCES

[Bruce, Peter, Bruce, Andrew, Gedeck, Peter. Practical Statistics for Data Scientists \(Kindle Location 5267\). O'Reilly Media. Kindle Edition.](#)

<https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/>