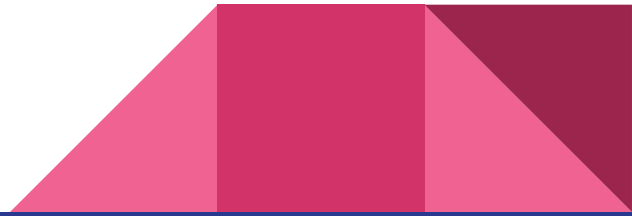
I.E. Mechatronics Manipal

Introduction to ML Algorithms- Decision Tree,Random Forest,XGBoost

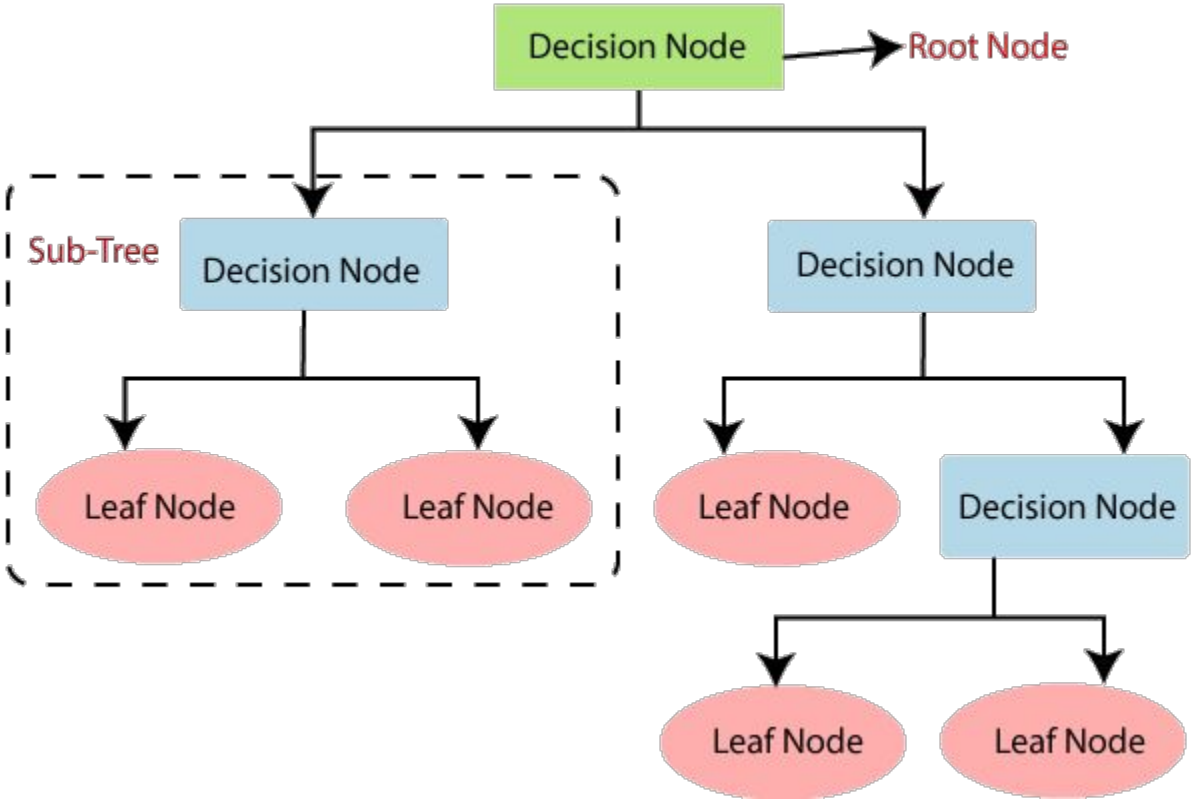
By Pratinav Seth and Siddharth Singh

What is a Decision Tree?

A Decision Tree is a tree-like graph with “nodes” representing the place where we pick an attribute and ask a question, “edges” represent the answers to the questions asked, and the “leaves” represent the actual output or class label. They are used in non-linear decision making with simple linear decision surface



Representation of a Decision Tree



Example Dataset

x or input values

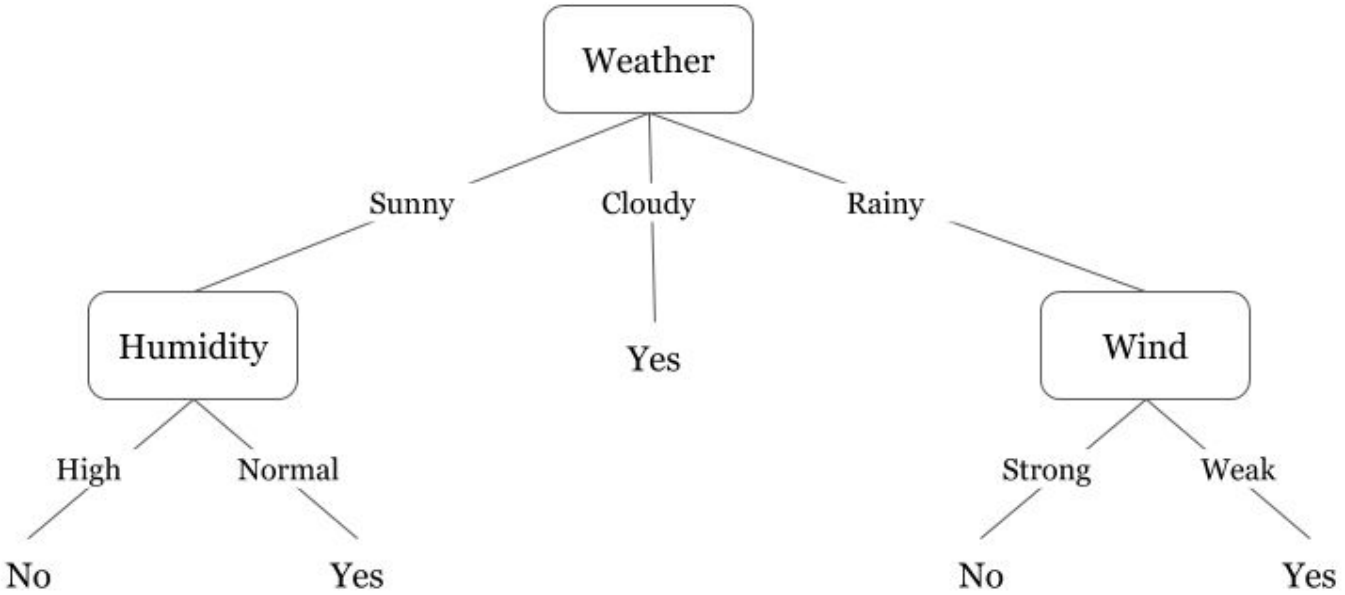
Attributes

y or output values

Training Examples

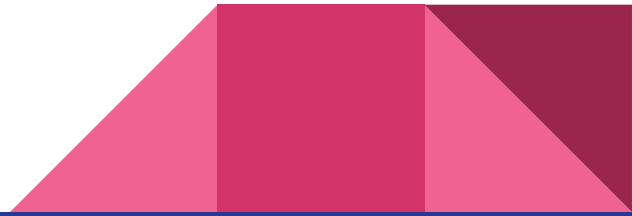
Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

Decision Tree based on the Example Dataset



A General Algorithm for a Decision Tree

1. Pick the best attribute/feature.
2. Ask the relevant questions.
3. Follow the answer path.
4. Go to Step - 1 until you arrive to the answer.



How do you define the best attribute?

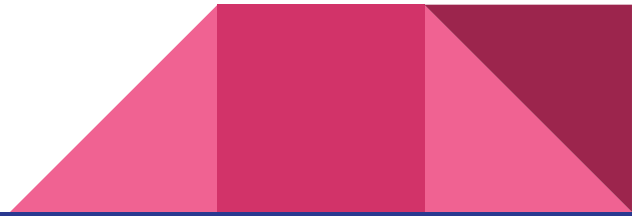
The best attribute is an attribute which can easily classify the available training examples.

How do you choose the best attribute among all the attributes?

Metrics for Decision Tree Classifiers

Generally we use two Techniques:

1. Information Gain:
    i. Basically, entropy is the measurement of the impurity or randomness in the data points
    ii. Lower the entropy, higher the information gain
2. Gini Index:
    i. It is calculated by subtracting the sum of squared probabilities of each class from one. It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values.
    ii. A feature with lower Gini index is chosen for a split.

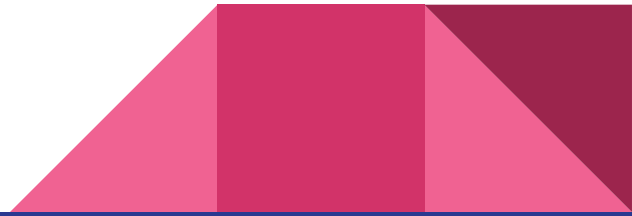


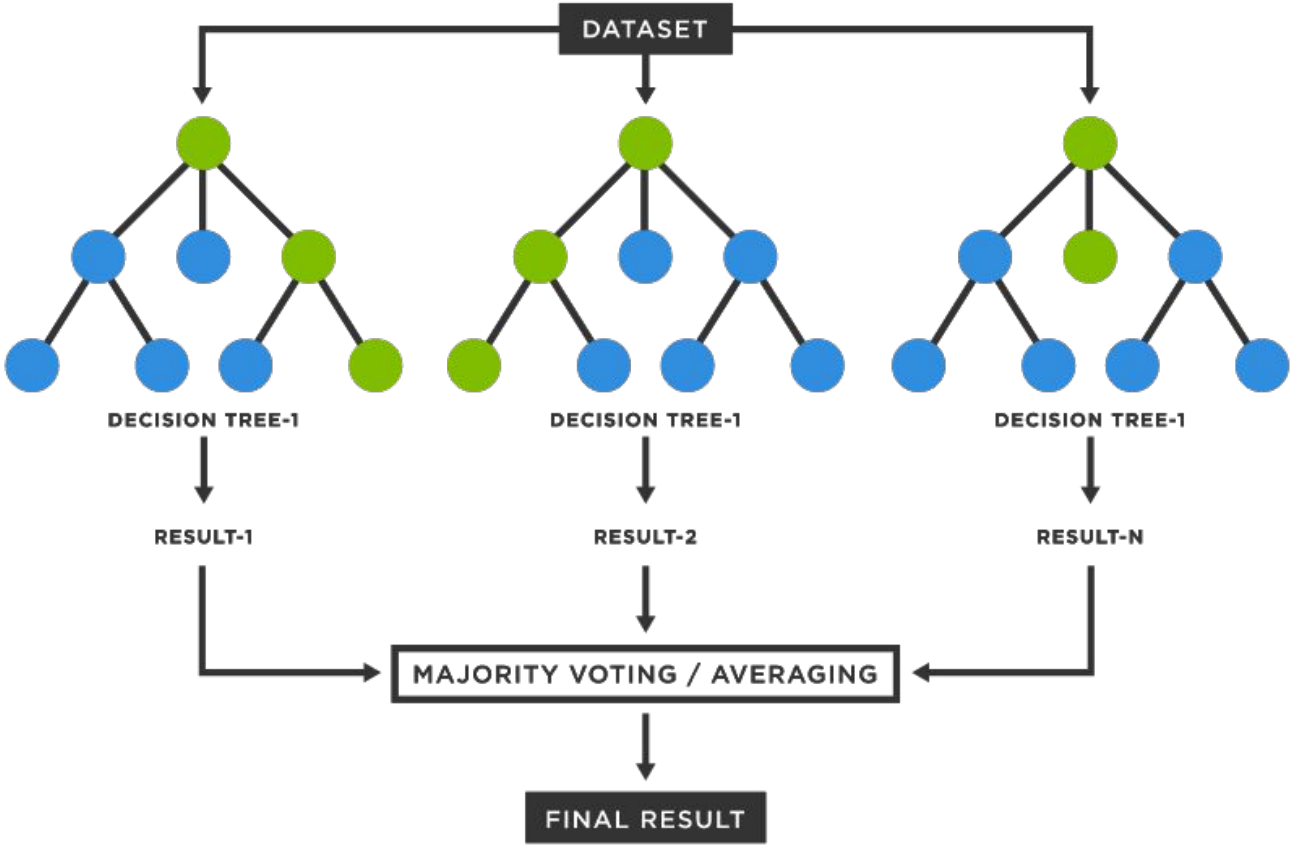
Metrics for Decision Tree Regressor

Random Forest

What is a Random Forest

Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

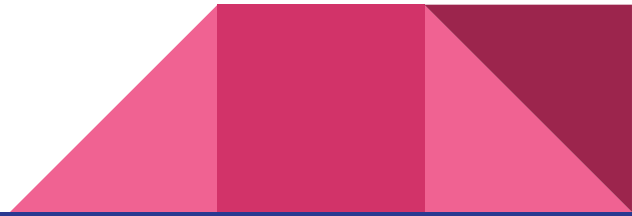


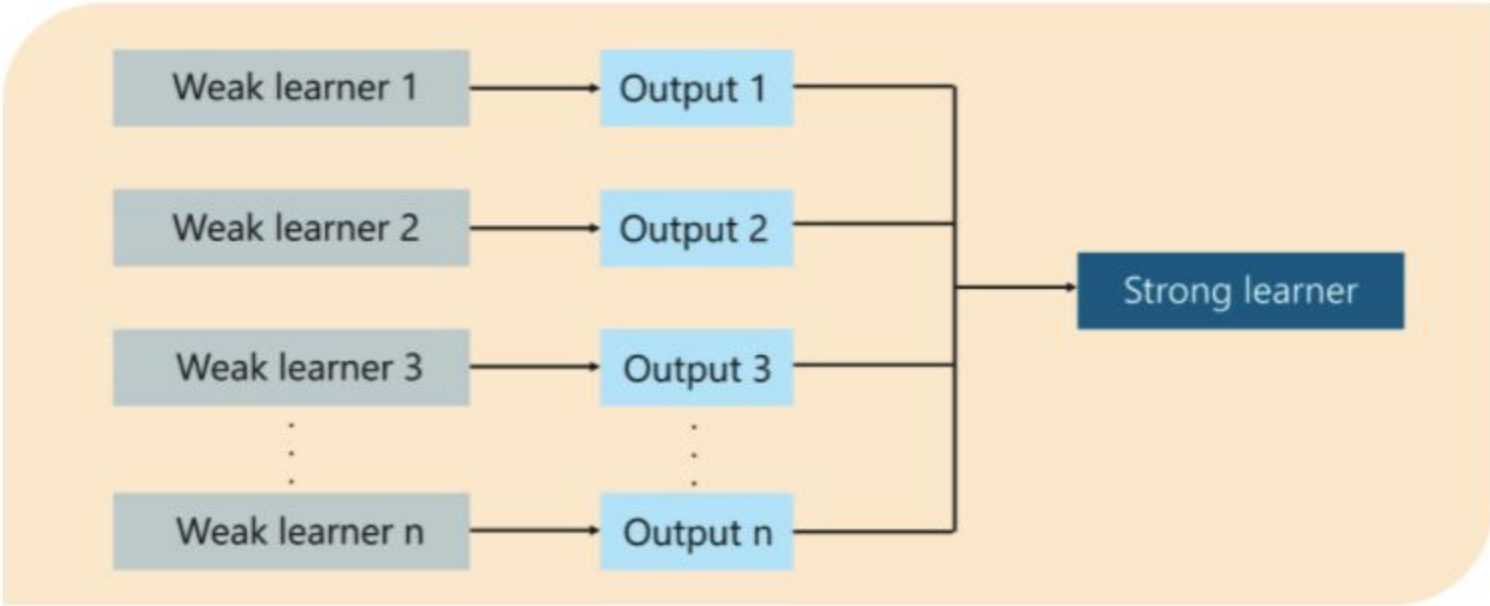

DATASET
DECISION TREE-1
DECISION TREE-1
DECISION TREE-1
RESULT-1
RESULT-2
RESULT-N
MAJORITY VOTING / AVERAGING
FINAL RESULT

Boosting

What is Boosting?

Boosting is an ensemble learning technique that uses a set of Machine Learning algorithms to convert weak learner to strong learners in order to increase the accuracy of the model.




Weak learner 1
Weak learner 2
Weak learner 3
Weak learner n
Output 1
Output 2
Output 3
Output n
Strong learner

What is Ensemble Learning?

Ensemble Learning

- Ensemble learning is a method that is used to enhance the performance of Machine Learning model by combining several learners.

- When compared to a single model, this type of learning builds models with improved efficiency and accuracy.

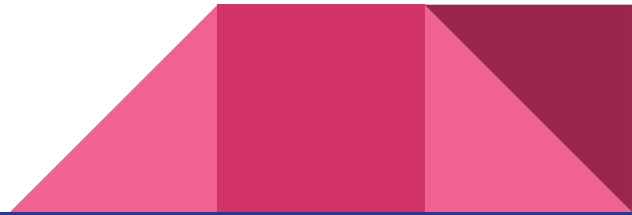
- The idea of boosting is to train weak learners sequentially, each trying to correct its predecessor.


single
1 iteration
bagging
parallel
boosting
sequential

Difference Between Boosting and Bagging

1. **Sequential ensemble**, popularly known as **boosting**, here the weak learners are sequentially produced during the training phase. The performance of the model is improved by assigning a higher weightage to the previous, incorrectly classified samples. An example of boosting is the AdaBoost algorithm.

2. **Parallel ensemble**, popularly known as **bagging**, here the weak learners are produced parallelly during the training phase. The performance of the model can be increased by parallelly training a number of weak learners on bootstrapped data sets. An example of bagging is the Random Forest algorithm.



XGBOOST

XGBoost

- XGBoost stands for e**X**treme **G**radient **B**oosting.

- It is an implementation of gradient boosted Decision Trees designed for speed and performance.

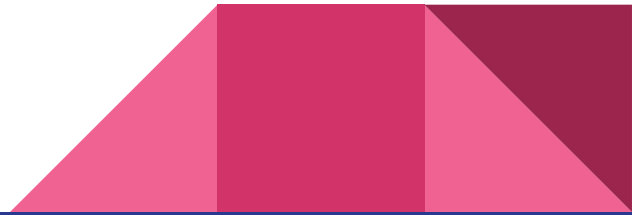
- Gradient boosting machines are generally very slow in implementation because of sequential model training. Hence, they are not very scalable. Thus, XGBoost is focused on computational speed and model performance.

A Little about Gradient Boosting

Gradient Boosting Method uses Gradient Descent to find the shortcomings in the previous learner's predictions. GBM algorithm can be given by following steps.

1. Fit a model to the data, $F1(x) = y$
2. Fit a model to the residuals, $h1(x) = y - F1(x)$
3. Create a new model, $F2(x) = F1(x) + h1(x)$

By combining weak learner after weak learner, our final model is able to account for a lot of the error from the original model and reduces this error over time.



XGBoost Provides:

- **Parallelization** of tree construction using all of your CPU cores during training.
- **Distributed Computing** for training very large models using a cluster of machines.
- **Out-of-Core Computing** for very large datasets that don’t fit into memory.
- **Cache Optimization** of data structures and algorithm to make the best use of hardware.