Machine Learning to build Intelligent Systems

Manas Dasgupta

Decision Tree Classifier

Structure of this Module

Decision Trees

TOPICS
Introduction to Classification Algorithms
Introduction to Decision Trees
Gini Index and Entropy
Decision Tree in Practice
Measuring Performance

Classification Problems

Classification is the process of recognizing, understanding, and grouping ideas and objects into pre-set categories or “sub-populations.” Using pre-categorized training datasets, machine learning programs use a variety of algorithms to classify future datasets into categories.

- Categorising whether a mail is Spam or Ham
- Predicting whether a Credit Card transaction is Fraud or legitimate
- Predicting whether or not a customer browsing on a ecommerce portal is likely to buy anything or not
- Predicting whether a person sentenced for a crime is likely to commit another crime or not
- Finding whether a person has a given medical condition from diagnostic images
- Face Detection Technologies

Classification Problems

- A classification problem requires that examples be classified and labelled into one of two or more ***classes***.
- A classification can have real-valued or discrete input variables.
- A problem with two classes is often called a two-class or ***binary classification problem.***
- A problem with more than two classes is often called a ***multi-class classification problem.***
- A problem where an example is assigned multiple classes is called a ***multi-label classification problem.***

Types of Classification Algorithms

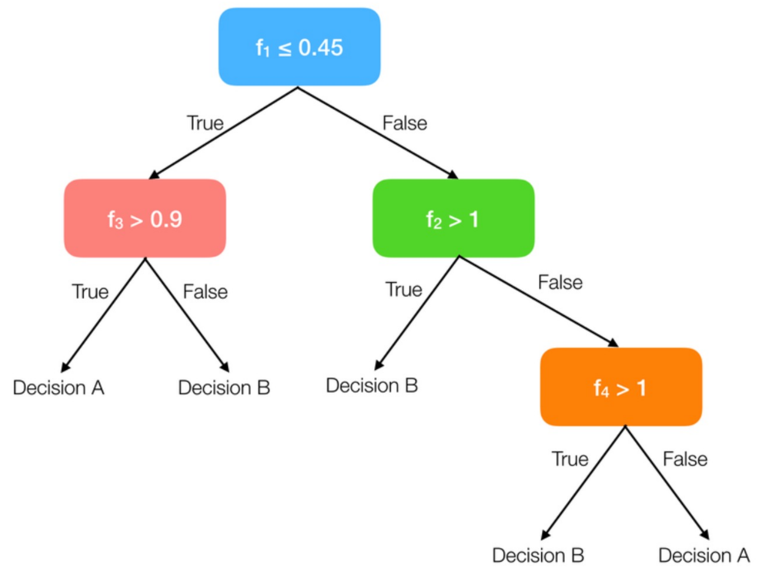
- **Decision Tree**: In a Decision Tree, a tree like structure is created wherein each node is a rule and branches are decisions or further sub-rules. Chains of rules along the nodes and branches upto the final leaves are created to arrive at final decisions or labels/categories.
- **Random Forest**: It is an Ensemble method which uses a technique called Bagging wherein the Algorithm takes in random slices of data and features and creates an ensemble of Decision trees to take a majority vote on labels.
- **Naïve Bayes**: Uses the Bayes Theorem for Classification. Assumes that the presence of a particular feature in a class is unrelated to any other feature. Works well even with small datasets.
- **KNN (K-Nearest Neighbour)**: Classifies observations based on similarity measures. Used for visual pattern recognition. Also used for Credit Card usage patter recognition for Fraud detection.
- **Support Vector Machine**
- **Logistic Regression**

Decision Trees

Decision Trees are versatile Machine Learning algorithms that ***can perform both classification and regression tasks***, and even multioutput tasks. They are very powerful algorithms, capable of fitting complex datasets.

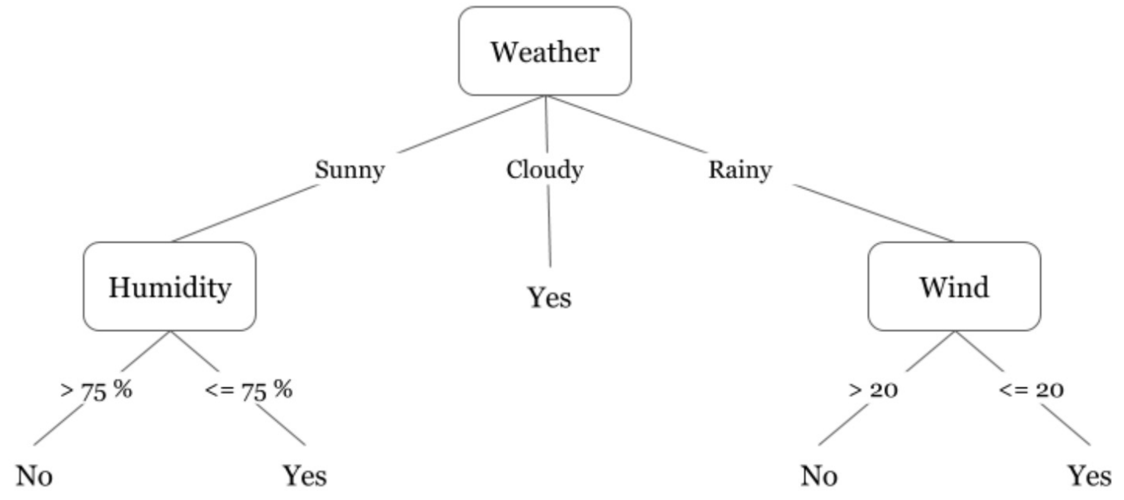
The goal of a Decision Tree is to create a model that predicts the value of a target variable by ***learning simple decision rules in the form of a tree inferred from the data features***.

Decision Trees are also the fundamental components of ***Random Forests,*** *which are among the most powerful Machine Learning algorithms available today.*



Example

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



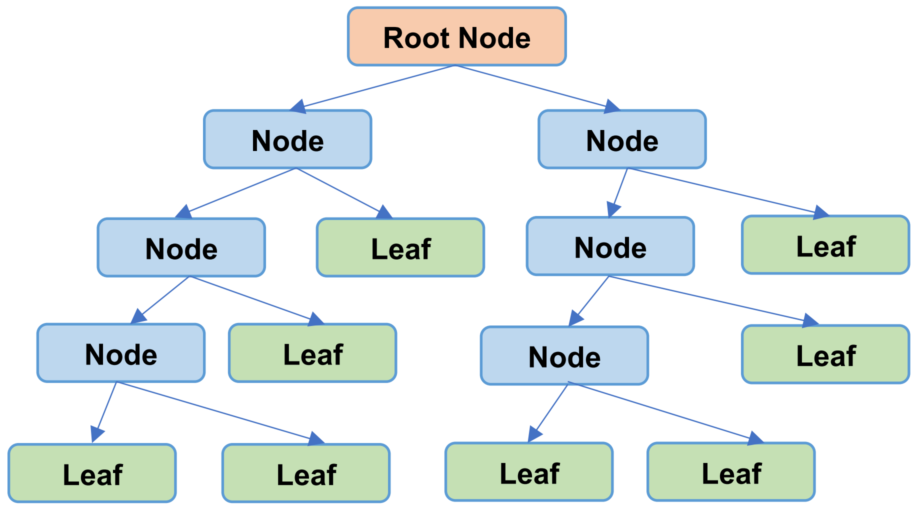
A general algorithm for a decision tree can be described as follows:

1. Pick the best attribute/feature. The best attribute is one which best splits or separates the data.
2. Ask the relevant question.
3. Follow the answer path.
4. Go to step 1 until you arrive to the answer.

Features of a Decision Tree

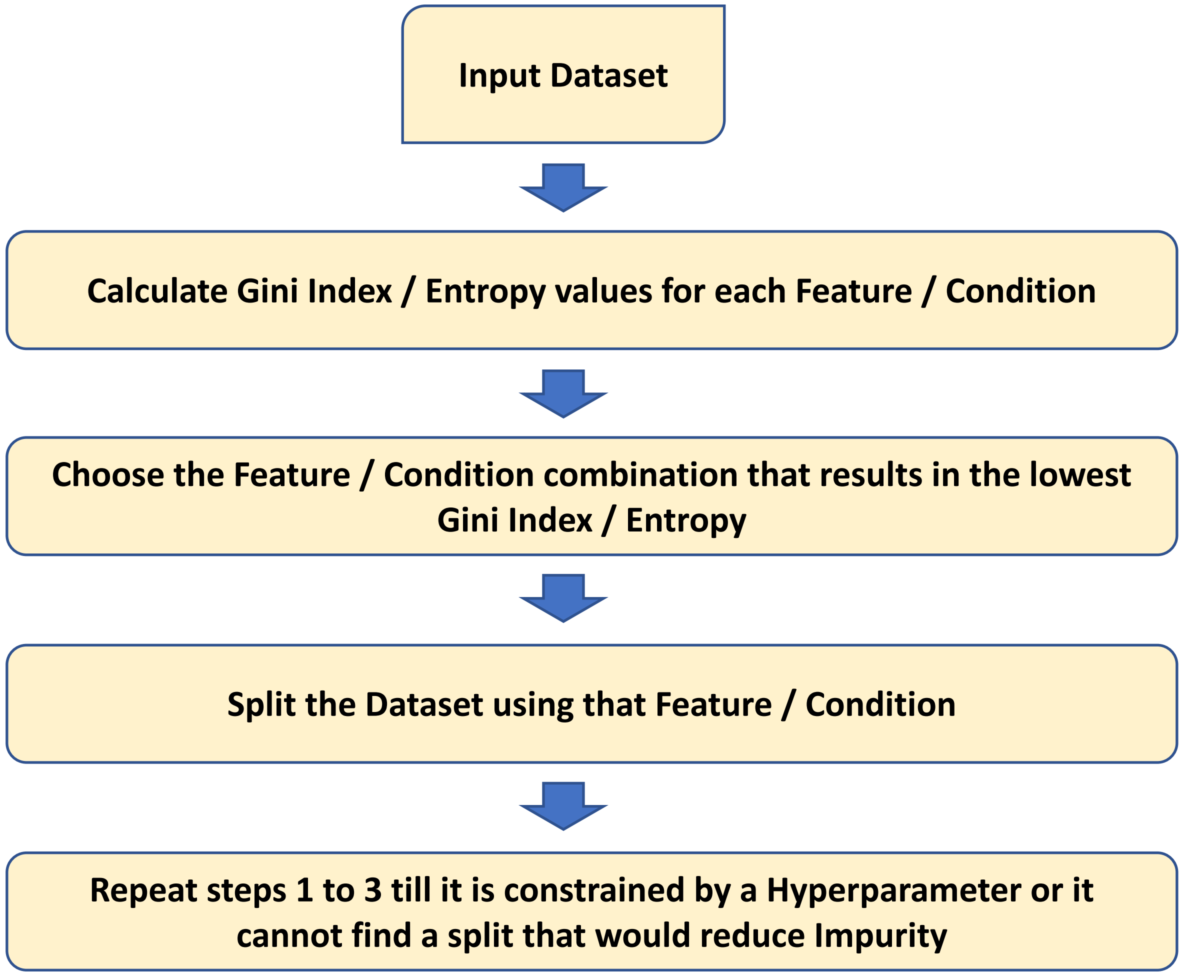
- Simple to understand and to interpret. Trees can be visualised.
- Requires little data preparation. Other techniques often require data Normalisation or Standardisation (Scaling), Dummy variables need to be created for Categorical values and blank values to be removed.
- Able to handle multi-output problems.
- Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by Boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Predictions are fast.

Decision Tree Concepts



- **Root Node:** The representation of the Root Node is the first condition of split that the algorithm places with the aim of ultimately labelling all the observations correctly.
- **Decision Node:** Any further node down the tree other than the Root Node, where a decision of data split is being made based on a condition on one or more attributes is a Decision Node or simply a Node.
- **Splitting:** Splitting is the process of separating the input dataset at a node into multiple output sets based on the condition at that Node.
- **Leaf:** Observations that have received the target label at the end of branching through the tree / conditions are called Leaf. There is no further splitting from Leaf nodes.
- **Pruning:** Decision Trees unless controlled through Hyperparameters, have the tendency to run through the entire dataset and place / create conditions / nodes to split data to the minutest level to achieve a 100% accuracy thereby causing overfitting. Pruning is a method to limit the number of levels of a tree to avoid overfitting. There are multiple Hyperparameters that can be used to achieve this.
- **Branch / Sub-Tree:** Tree beneath a Node that’s created as a result of conditions in that Node.
- **Parent and Child Node:** Nodes beneath a particular Node are child nodes and the Node in question is termed as the Parent Node.

Decision Tree Steps


Input Dataset
Calculate Gini Index / Entropy values for each Feature / Condition
Choose the Feature / Condition combination that results in the lowest Gini Index / Entropy
Split the Dataset using that Feature / Condition
Repeat steps 1 to 3 till it is constrained by a Hyperparameter or it cannot find a split that would reduce Impurity

To begin with, the Decision Tree Algorithm aims to find a single Feature ‘***k***’ and a Threshold ‘t_k’ to split the Data.

The approach taken is to find out the Feature ’k’, that yields the lowest value of ***Gini Impurity Index*** or **‘Entropy’**.

Gini Index

Gini Index or Gini Impurity: It a measure of impurity in the resulting node when the current node is split on a particular attribute.

The values of the Gini index vary between the values 0 and 1, where 0 expresses the **absolute purity of classification**, i.e., all the elements belong to a specific class. The Gini value of 1 indicate presence of data belonging to multiple target classes. The value of 0.5 indicate an equal distribution of elements over the target classes.

While designing the decision tree, the feature resulting in the minimum value of the Gini Index would get be chosen as the attribute to split the current node on. Gini Indexes are calculated for all the attributes at a node to determine the minimum value and corresponding attribute, which is then taken to split the node.
The Gini Index is calculated as the sum of squared probabilities of each class.

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^{2}$$

P_i = **Probability of an element** at the resultant node being classified for a specific class.

Entropy

Entropy is a measure of Chaos, or measure or impurity against a feature, to be calculated at the resultant node when the current node is split on that said feature.

Like the Gini index, the value of Entropy also range between 0 and 1. However, that is only for Binary classification.
The maximum value for Entropy depends on the number of classes.

- Two classes: Max entropy is 1.
- Four Classes: Max entropy is 2.
- Eight Classes: Max entropy is 3.
- 16 classes: Max entropy is 4.

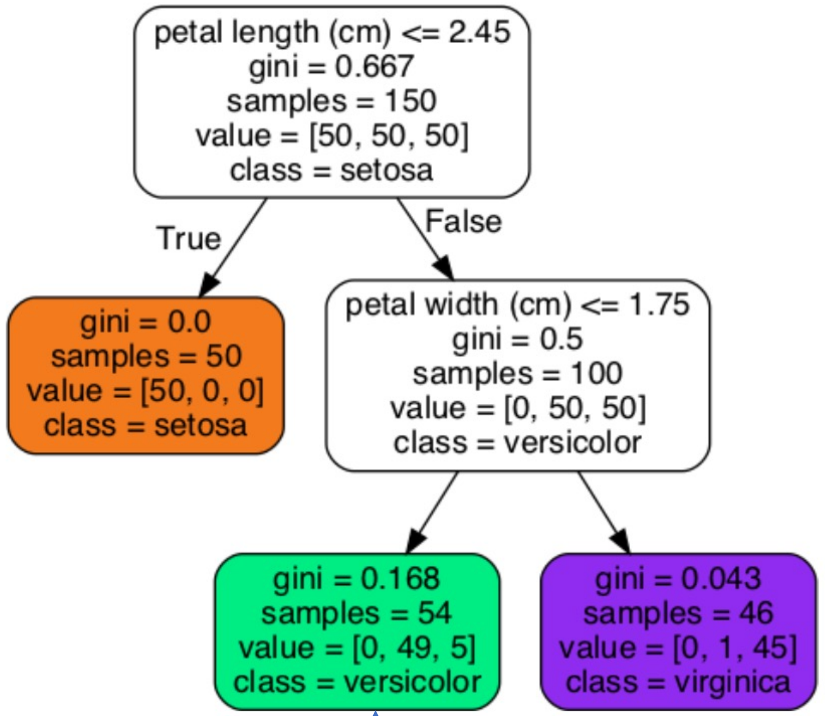
$$H_i = -\sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^{n} p_{i,k} \log_2 \left(p_{i,k}\right)$$

***P**$_i$* = **Probability of an element** at the resultant node being classified for a specific class.

Calculations from the IRIS Dataset

	sepallength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

Graphviz Representation



Gini Index

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2 \Rightarrow 1 - (0/54)^2 - (49/54)^2 - (5/54)^2 \approx 0.168$$

Entropy

$$H_i = - \sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^{n} p_{i,k} \log_2 \left(p_{i,k}\right) \Rightarrow -\frac{49}{54} \log_2 \left(\frac{49}{54}\right) - \frac{5}{54} \log_2 \left(\frac{5}{54}\right) \approx 0.445$$

Gini vs Entropy

Most of the time it does not make a big difference whether you use Gini or Entropy, they would lead to similar trees. Gini impurity is slightly faster to compute, so it is a good default. However, when they differ, Gini impurity tends to isolate the most frequent class in its own branch of the tree, while entropy tends to produce slightly more balanced trees.

Decision Tree Advantages and Disadvantages

Advantages	Disadvantages
• Easily interpretable through visual representations. • Decision Trees split starting from the most Important Attributes and go down gradually along the decreasing measure of importance. This increases understandability. • Decision Trees can handle Categorical Data unlike most other algorithms. In other algorithms, Categorical data need to be converted into Numeric ones. • Decision Trees do not need Input Data to be Normalised. • Decision Trees do not make assumptions on the input attributes. It can ingest all kinds of data — numeric, categorical, strings, Boolean, etc.	• Decision Trees tend to Overfit the data if left uncontrolled. Without hyperparameter driven generalisation, a tree will keep splitting till it has correctly classified all the data points in the training set. • Decision Trees tend to be very unstable (high variance), which is a direct fallout of overfitting. Minor changes in the input data can change a tree considerably.

Tree Pruning / Constraining Hyperparameters (Regularization)

To avoid the problem of ***overfitting in a Decision Tree classifier***, we use a technique called Pruning which is essentially Regularization of the Decision Tree. The Pruning technique helps in cutting down the branches of the Decision Tree using multiple Hyperparameters to limit the number of Branches and levels thereby avoiding overfitting.

Such limiting of branches with the right use of Hyperparameter will increase generalisability and reduce variance of a tree model and improve Test accuracy.

- ***min_samples_split***: This parameter ensures that a split is done only when the node will has at least the number of samples specified in ‘min_samples_split’.
- ***max_leaf_nodes***: This parameter will ensure the number of leaf nodes do not grow beyond the value specified against this.
- ***max_depth***: Limits the number of levels the Tree can grow to.
- ***min_samples_leaf***: The minimum number of samples that will go to the resultant leaf from a split.
- ***max_features***: The maximum number of Features that are evaluated for splitting at each node.

Practice using Python Code

In this section, we will look at a practical example of how to create a Decision Tree using Python. The dataset we will be using is the Iris, a very well-known publicly available dataset. Iris contains 3 classes of 50 instances each (total of 150 observations in the dataset), where each class refers to a type of the Iris plant. The task for us to use the independent attributes of the Iris dataset to predict the Class of Iris plant.

List of Attributes:

- Sepal length in cm
- Sepal width in cm
- Petal length in cm
- Petal width in cm
- Classes:
  Iris-setosa
  Iris-versicolour
  Iris-virginica

Decision Trees

Python Demo

- **Iris Flower Classification**

Hope you have liked this Video.
Please help us by providing your Ratings and Comments for this Course!

Thank You!!
Manas Dasgupta

Happy Learning!!