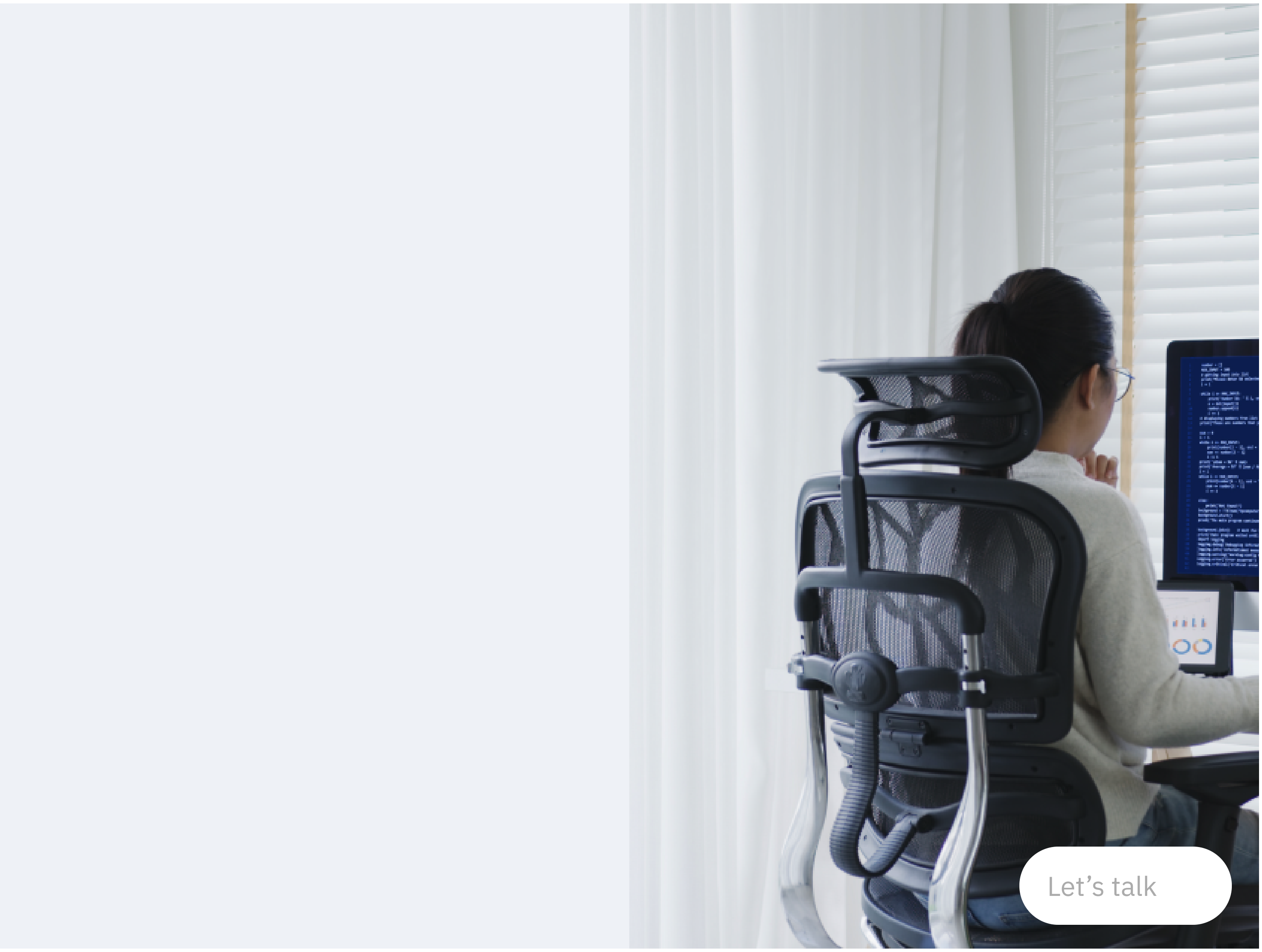
What is a Decision Tree?

Learn the pros and cons of using decision trees for data mining and knowledge discovery tasks.

Learn about SPSS Modeler →

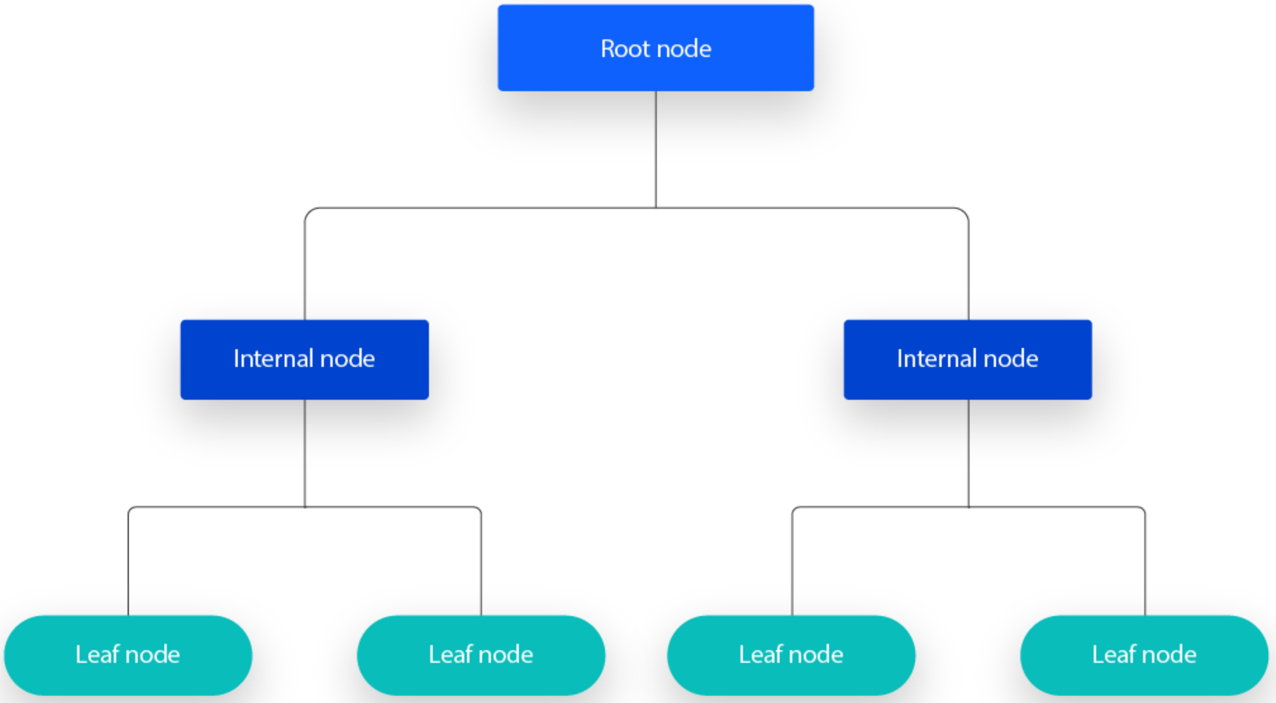


Let’s talk

Decision Trees

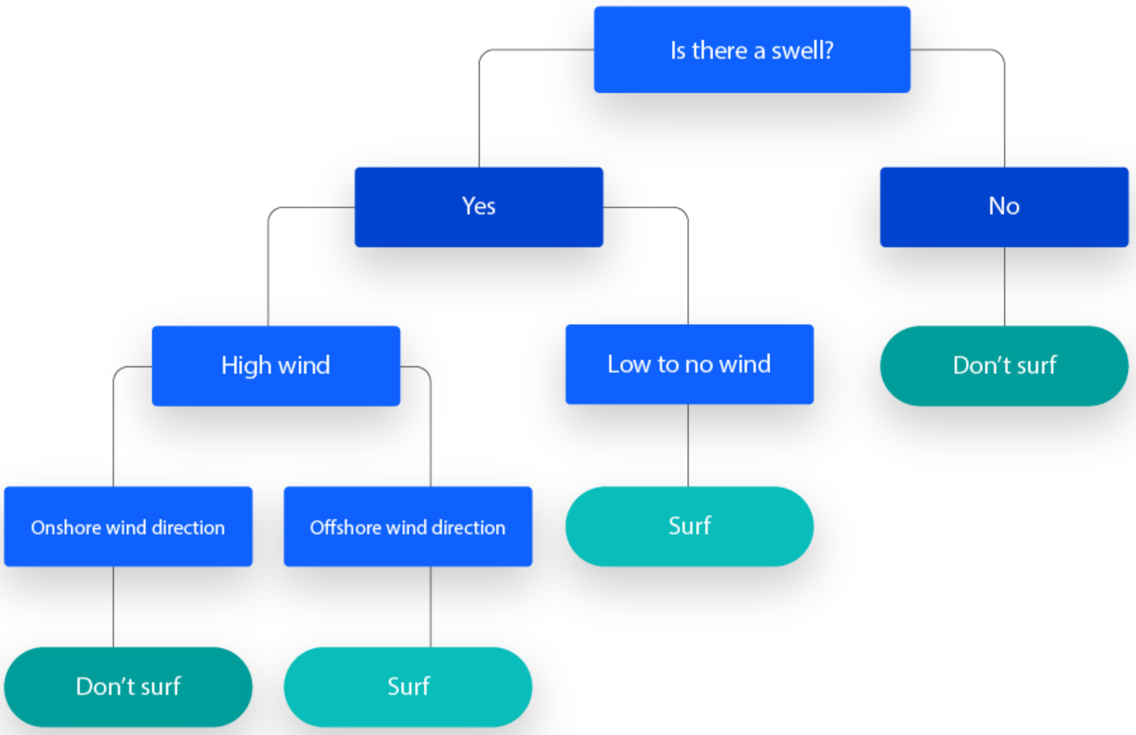
A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

Let’s talk



As you can see from the diagram above, a decision tree starts with a root node, which does not have any incoming branches. The outgoing branches from the root node then feed into the internal nodes, also known as decision nodes. Based on the available features, both node types conduct evaluations to form homogenous subsets, which are denoted by leaf nodes, or terminal nodes. The leaf nodes represent all the possible outcomes within the dataset. As an example, let’s imagine that you were trying to assess whether or not you should go surf, you may use the following decision rules to make a choice:

Let’s talk



This type of flowchart structure also creates an easy to digest representation of decision-making, allowing different groups across an organization to better understand why a decision was made.

Decision tree learning employs a divide and conquer strategy by conducting a greedy search to identify the optimal split points within a tree. This process of splitting is then repeated in a top-down, recursive manner until all, or the majority of records have been classified under specific class labels. Whether or not all data points are classified as homogenous sets is largely dependent on the complexity of the decision tree. Smaller trees are more easily able to attain pure leaf nodes—i.e. data points in a single class. However, as a tree grows in size, it becomes increasingly difficult to maintain this purity, and it usually results in too little data falling within a given subtree. When this occurs, it is known as data fragmentation, and it can often lead to overfitting. As a result, decision trees have preference for small trees, which is consistent with the principle of parsimony in Occam's Razor; that is, "entities should not be multiplied beyond necessity." Said differently, decision trees should add complexity only if necessary, as the simplest explanation is often the best. To reduce complexity and prevent overfitting, pruning is usually employed; this is a process, which removes branches that split on features with low importance. The model's fit can then be evaluated through the process of cros Let's talk validation. Another way that decision trees can maintain their accuracy is by forming an ensemble via a random forest algorithm; this classifier predicts more accurate results, particularly when the individual trees are uncorrelated with each other.

Types of Decision Trees

Hunt's algorithm, which was developed in the 1960s to model human learning in Psychology, forms the foundation of many popular decision tree algorithms, such as the following:

- ID3: Ross Quinlan is credited within the development of ID3, which is shorthand for "Iterative Dichotomiser 3." This algorithm leverages entropy and information gain as metrics to evaluate candidate splits. Some of Quinlan's research on this algorithm from 1986 can be found here (link resides outside of ibm.com).

- C4.5: This algorithm is considered a later iteration of ID3, which was also developed by Quinlan. It can use information gain or gain ratios to evaluate split points within the decision trees.

- CART: The term, CART, is an abbreviation for "classification and regression trees" and was introduced by Leo Breiman. This algorithm typically utilizes Gini impurity to identify the ideal attribute to split on. Gini impurity measures how often a randomly chosen attribute is misclassified. When evaluating using Gini impurity, a lower value is more ideal.

How to choose the best attribute at each node

Let's talk

While there are multiple ways to select the best attribute at each node, two methods, information gain and Gini impurity, act as popular splitting criterion for decision tree models. They help to evaluate the quality of each test condition and how well it will be able to classify samples into a class.

Entropy and Information Gain

It’s difficult to explain information gain without first discussing entropy. Entropy is a concept that stems from information theory, which measures the impurity of the sample values. It is defined with by the following formula, where:

$$\text{Entropy}(S) = -\sum_{c \in C} p(c)\log_2 p(c)$$

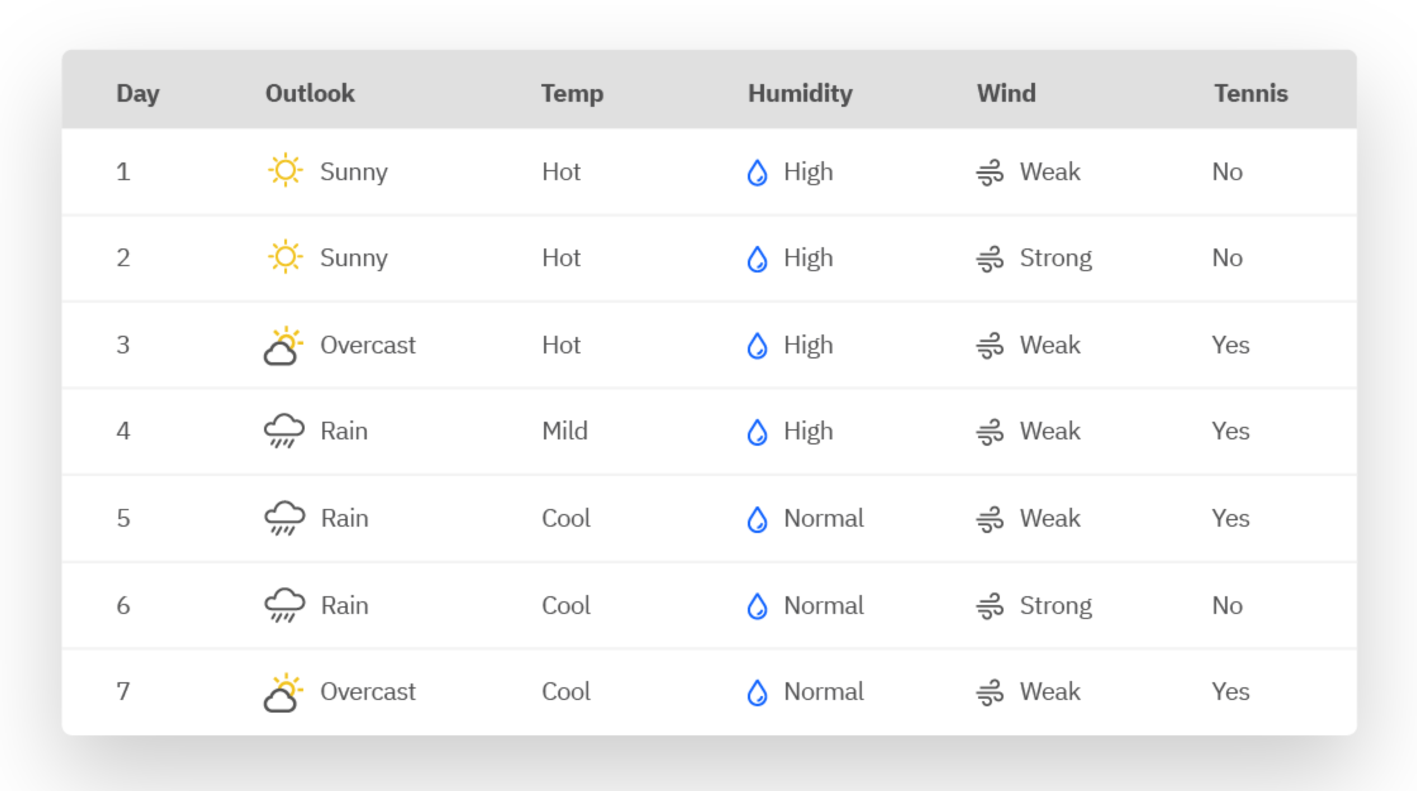
– S represents the data set that entropy is calculated
– c represents the classes in set, S
– p(c) represents the proportion of data points that belong to class c to the number of total data points in set, S

Entropy values can fall between 0 and 1. If all samples in data set, S, belong to on
then entropy will equal zero. If half of the samples are classified as one class and t

other half are in another class, entropy will be at its highest at 1. In order to select the best feature to split on and find the optimal decision tree, the attribute with the smallest amount of entropy should be used. Information gain represents the difference in entropy before and after a split on a given attribute. The attribute with the highest information gain will produce the best split as it's doing the best job at classifying the training data according to its target classification. Information gain is usually represented with the following formula, where:

- *a* represents a specific attribute or class label
- *Entropy(S)* is the entropy of dataset, S
- |Sv|/ |S| represents the proportion of the values in S_v to the number of values in dataset, S
- *Entropy*(S_v) is the entropy of dataset, S_v

Let's walk through an example to solidify these concepts. Imagine that we have the following arbitrary dataset:



Day	Outlook	Temp	Humidity	Wind	Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Weak	Yes

For this dataset, the entropy is 0.94. This can be calculated by finding the proportion of days where "Play Tennis" is "Yes", which is 9/14, and the proportion of days where "Play Tennis" is "No", which is 5/14. Then, these values can be plugged into the entropy formula above.

Entropy (Tennis) = -(9/14) log2(9/14) – (5/14) log2 (5/14) = 0.94

We can then compute the information gain for each of the attributes individually. For example, the information gain for the attribute, "Humidity" would be the following:

Gain (Tennis, Humidity) = (0.94)-(7/14)*(0.985) – (7/14)*(0.592) = 0.151

As a recap,

- 7/14 represents the proportion of values where humidity equals "high" to the total number of humidity values. In this case, the number of values where humidity equals "high" is the same as the number of values where humidity equals "normal".

- 0.985 is the entropy when Humidity = “high”

- 0.59 is the entropy when Humidity = “normal”

Then, repeat the calculation for information gain for each attribute in the table above, and select the attribute with the highest information gain to be the first split point in the decision tree. In this case, outlook produces the highest information gain. From there, the process is repeated for each subtree.

Gini Impurity

Gini impurity is the probability of incorrectly classifying random data point in the dataset if it were labeled based on the class distribution of the dataset. Similar to entropy, if set, S, is pure—i.e. belonging to one class) then, its impurity is zero. This is denoted by the following formula:

$$\text{Gini Impurity} = 1 - \sum_i (p_i)^2$$

Advantages and disadvantages of Decision Trees

While decision trees can be used in a variety of use cases, other algorithms typically outperform decision tree algorithms. That said, decision trees are particularly useful for data mining and knowledge discovery tasks. Let's explore the key benefits and challenges of utilizing decision trees more below:

Advantages

- Easy to interpret: The Boolean logic and visual representations of decision trees make them easier to understand and consume. The hierarchical nature of a decision tree also makes it easy to see which attributes are most important, which isn't always clear with other algorithms, like neural networks.

- Little to no data preparation required: Decision trees have a number of characteristics, which make it more flexible than other classifiers. It can handle various data types—i.e. discrete or continuous values, and continuous values can be converted into categorical values through the use of thresholds. Additionally, it can also handle values with missing values, which can be problematic for other classifiers, like Naïve Bayes.

- More flexible: Decision trees can be leveraged for both classification and regression tasks, making it more flexible than some other algorithms. It's also insensitive to underlying relationships between attributes; this means that if two variables are highly correlated, the algorithm will only choose one of the features to split on.

Disadvantages

Let's talk

- Prone to overfitting: Complex decision trees tend to overfit and do not generalize well to new data. This scenario can be avoided through the processes of pre-pruning or post-pruning. Pre-pruning halts tree growth when there is insufficient data while post-pruning removes subtrees with inadequate data after tree construction.

- High variance estimators: Small variations within data can produce a very different decision tree. Bagging, or the averaging of estimates, can be a method of reducing variance of decision trees. However, this approach is limited as it can lead to highly correlated predictors.

- More costly: Given that decision trees take a greedy search approach during construction, they can be more expensive to train compared to other algorithms.

Decision Trees and IBM

IBM SPSS Modeler is a data mining tool that allows you to develop predictive models to deploy them into business operations. Designed around the industry-standard CRISP-DM model, IBM SPSS Modeler supports the entire data mining process, from data processing to better business outcomes.

IBM SPSS Decision Trees features visual classification and decision trees to help you present categorical results and more clearly explain analysis to non-technical audiences. Create classification models for segmentation, stratification, prediction, data reduction and variable screening.

For more information on IBM's data mining tools and solutions, sign up for an IBMid and create an IBM Cloud account today.

Related solutions

Let’s talk

IBM SPSS Modeler

IBM SPSS Modeler is a data mining tool that allows you to develop predictive models to deploy them into business operations. Designed around the industry-standard CRISP-DM model, IBM SPSS Modeler supports the entire data mining process, from data processing to better business outcomes.

Explore IBM SPSS Modeler →

IBM SPSS Decision Trees

IBM SPSS Decision Trees features visual classification and decision trees to help you present categorical results and more clearly explain analysis to non-technical audiences. Create classification models for segmentation, stratification, prediction, data reduction and variable screening.

Explore IBM SPSS Decision Trees →

watsonx.ai

With watsonx.ai, you can train, validate, tune and deploy generative AI, foundation models and machine learning capabilities with ease and build AI applications in a fraction of the time with a fraction of the data.

Learn about watsonx.ai →

Resources

Let's talk

IBM SPSS Software

Find opportunities, improve efficiency and minimize risk using the advanced statistical analysis capabilities of IBM SPSS software.

IBM SPSS Statistics Use Cases

Discover how experts across various industries are adopting IBM SPSS Statistics. Draw from their insights and drive better outcomes in your field.

Take the next step

Train, validate, tune and deploy generative AI, foundation models and machine learning capabilities with IBM watsonx.ai™, a next generation enterprise studio for AI builders. Build AI applications in a fraction of the time with a fraction of the data.

Explore watsonx.ai →

Request a demo →

Let’s talk