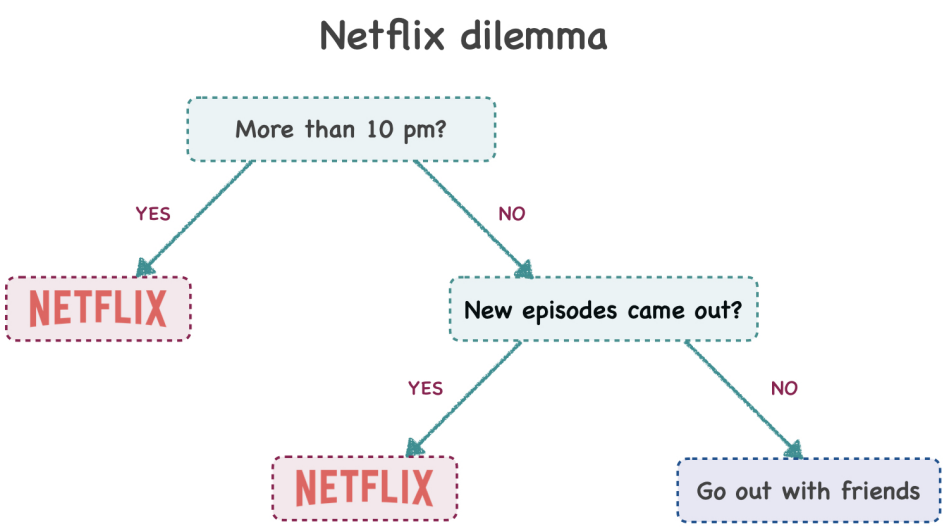
Theory: Decision Trees

22 minutes 0 / 5 problems solved

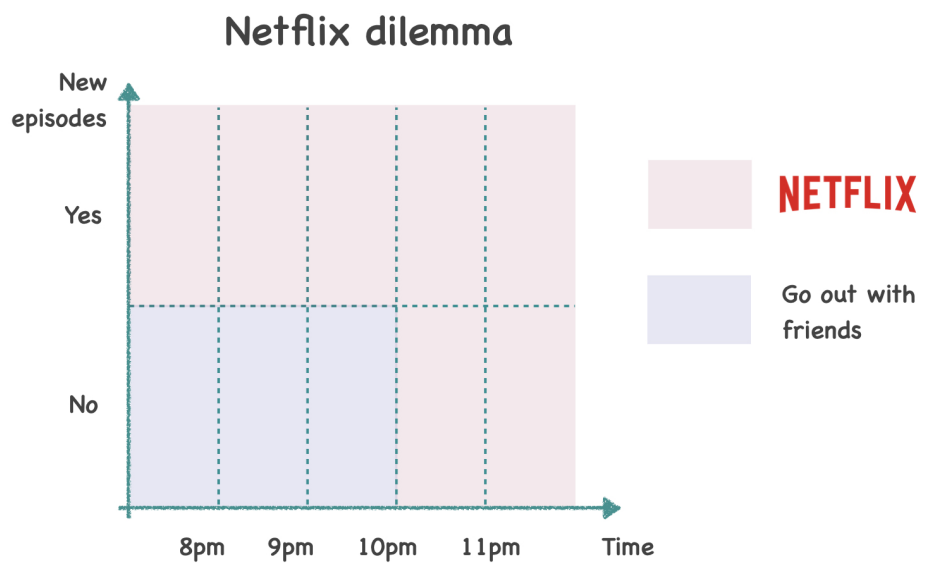
This topic will focus on a popular machine learning algorithm, **decision trees**. Decision trees are a reasonably popular machine learning algorithm that shows the human decision-making process. This characteristic makes this algorithm logically easy to understand and makes it a versatile tool for regression and classification problems.

Imagine a situation where you are deciding between going for a walk with your friends or staying at home in the evening.



In terms of Machine Learning, we can say that it is a problem of binary classification of entertainment with features "time" and "new series on Netflix".

In a different guise, these features can be positioned along the X and Y axes, obtaining a two-dimensional plane. If we make the divisions of the plane using lines, then we will see that each subdomain of the plane represents some class of entertainment.



A little more imagination is required to represent multi-feature spaces, but first things first.

§1. The main idea

Before we get to see how these trees are built, let's start with the terminology.

- **Node.** A tree node is a part of a tree containing a condition. It has *zero or more* child nodes.
- **Root.** The first node of a tree.

Current topic:

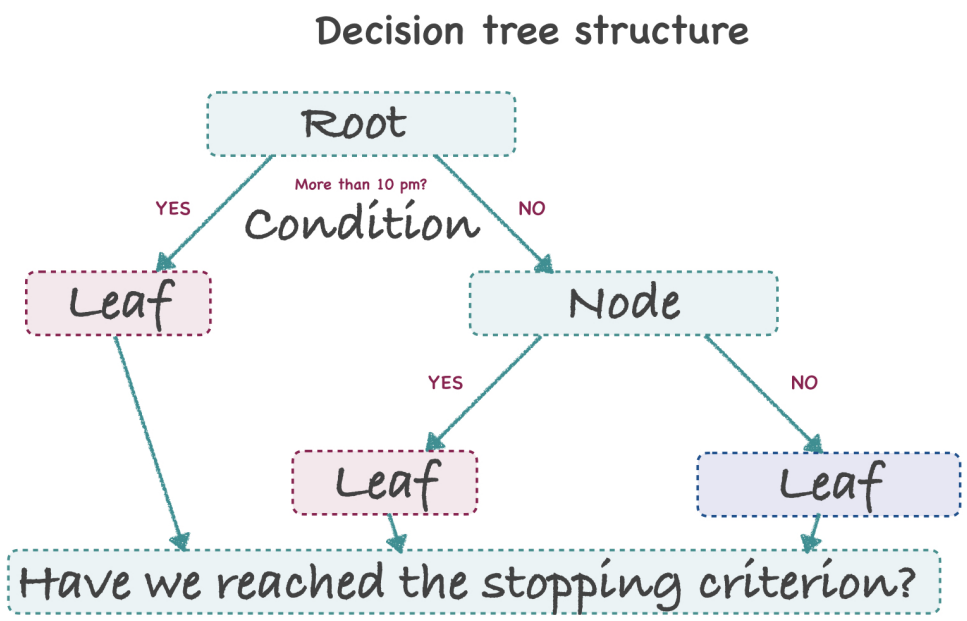
Decision Trees

Topic depends on:

- ✕ Classification performance metrics
- ✕ Introduction to regression

Table of contents:

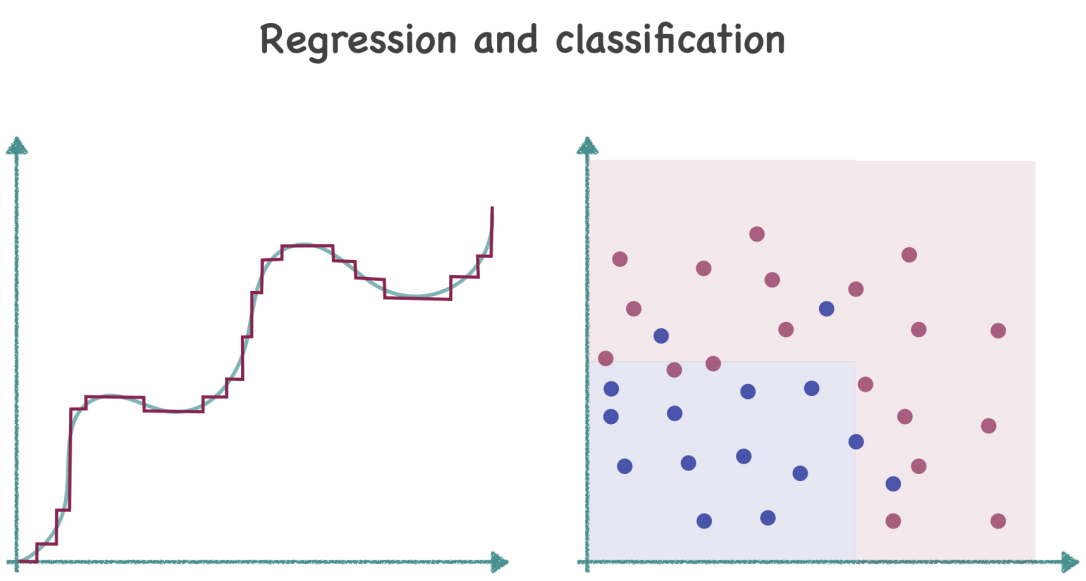
- Leaf or terminal node. A child node or a node without child nodes (an outcome).
- Condition or an internal node. A condition in a node indicates a possible result or the next action with the sample.
- Stopping criterion. Some particular condition where a decision was made and further splitting of the tree was stopped.



In such terms, we can write the main algorithm as follows. At each node, we want to find the best split into two parts, two leaves, with a predetermined quality loss function. We will clarify what the loss function is below. Still, it is intuitively clear that, at each step, we want to get a question, the answer to which will make our solution more accurate. Depending on a condition, a sample goes to the right node or the left node.

It is important to note that splitting at a node does not always create two leaves. In the case of categorical features, the node can be split into N leaves.

Further, each leaf becomes the next node. We check this node for a predetermined stopping criterion. If it is fulfilled, the node becomes a leaf — the final point. Otherwise, the action is repeated recursively. The final leaf (terminal node) determines the solution for each sample that falls into it for prediction. In other words, all samples from one leaf are of one class or value. In the task of regression, the leaf produces an average real number; in classification, the leaf produces a class or the probability of belonging to that class.

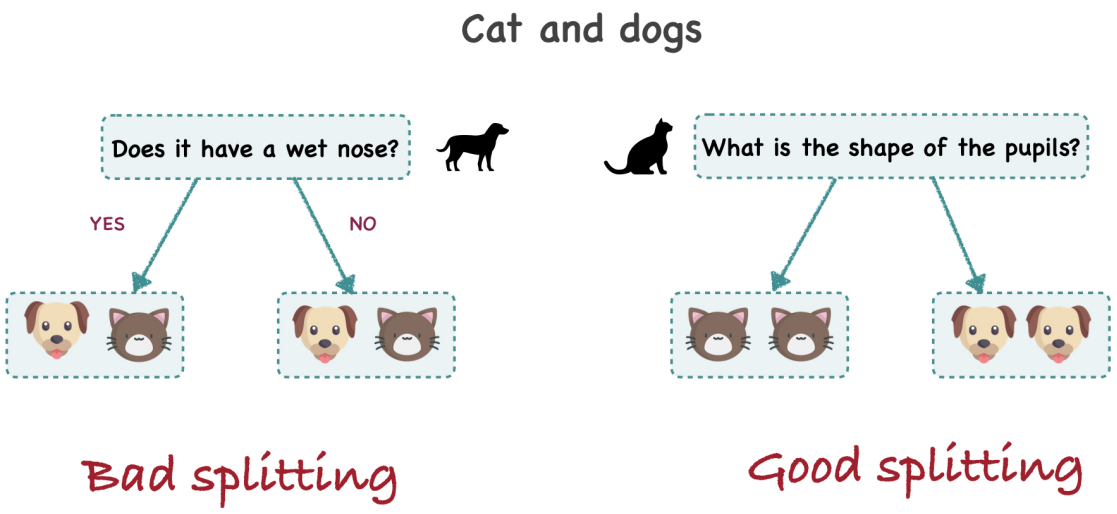


This architecture of the algorithm allows you to build a very fast, interpretable tool that can identify nonlinear dependencies within the data. Another essential property of trees is the ability to work with missing values as well as with non-normalized data. These advantages make trees a potent tool compared to linear models. Quite an interesting fact that trees are nonlinear models, but still build a linear model on the feature space. You can see this in the picture of splitting the plane into sectors.

On the other hand, it would be wrong not to mention the disadvantages of decision trees. First of all, the algorithm is discrete, which means that it is difficult to optimize. A too complex design makes trees more prone to overfitting.

§2. Quality loss function

Despite the seeming simplicity, the main problem of trees is how to choose a new splitting. Intuitively, it is obvious that at each iteration we split the node, we want different objects to go to different leaves and similar objects to stay together.



The example above illustrates the mathematical description of the Shannon Entropy for the state of a system — also known as Information Gain. In simple terms, entropy is a measure of the chaos of a system: more chaos means large entropy. If all the cats are on some leaf, then the entropy of this leaf equals zero. This is our goal.

Another criterion may be familiar for those who came to ML from economics. It is the Gini Impurity and it can be used for calculating the informativeness of the system in classification problems. It is perhaps more complex from a mathematical point of view but can be interpreted as follows. An error of the algorithm is built from the prediction of the class through its probability of occurrence at this node. This approach allows this metric to be used on unbalanced data.

For the regression problem, the criterion is more trivial, reduction in variance. It is necessary to average the values (for example, mean or median) that fall into one leaf: the smaller variance of the variable in the leaf means the lower the informativeness criteria.

§3. Stopping criteria and pruning

Sooner or later, the splitting of the tree must be stopped. Otherwise, only one sample will fall into each leaf, and the tree will be overfitted. It means that a tree will be adapted to the training set and poorly generalized to new data.

There are two ways to find a balance between uninformative trees and overfitting trees: stopping criterion and pruning.

Pruning is the removal of leaves from a tree to reduce complexity and increase generalizability. To use this, we build a complete tree and remove some of the leaves, the splitting into which did not give much information there.

The stopping criterion is some criterion that we check at each node to understand whether we have reached the required accuracy. It can be very diverse: the maximum depth of the tree, the number of samples in a leaf, improvement by a certain percentage of accuracy, and many others.

§4. Conclusion

In this topic, we learned about such a robust machine learning tool as decision trees. Its main parts are the nodes, which in the final leaf give us the prediction.

The primary technique for splitting a node into leaves is selecting the actual loss function for the problem. To optimize and generalize the work of the tree we can use various stopping criteria (anywhere on the tree from the root to the leaves) or pruning, to remove leaves towards the root.

Report a typo

6 users liked this theory. **0** didn't like it. **What about you?**

Start practicing

Comments (0) Hints (0) Useful links (0) Show discussion