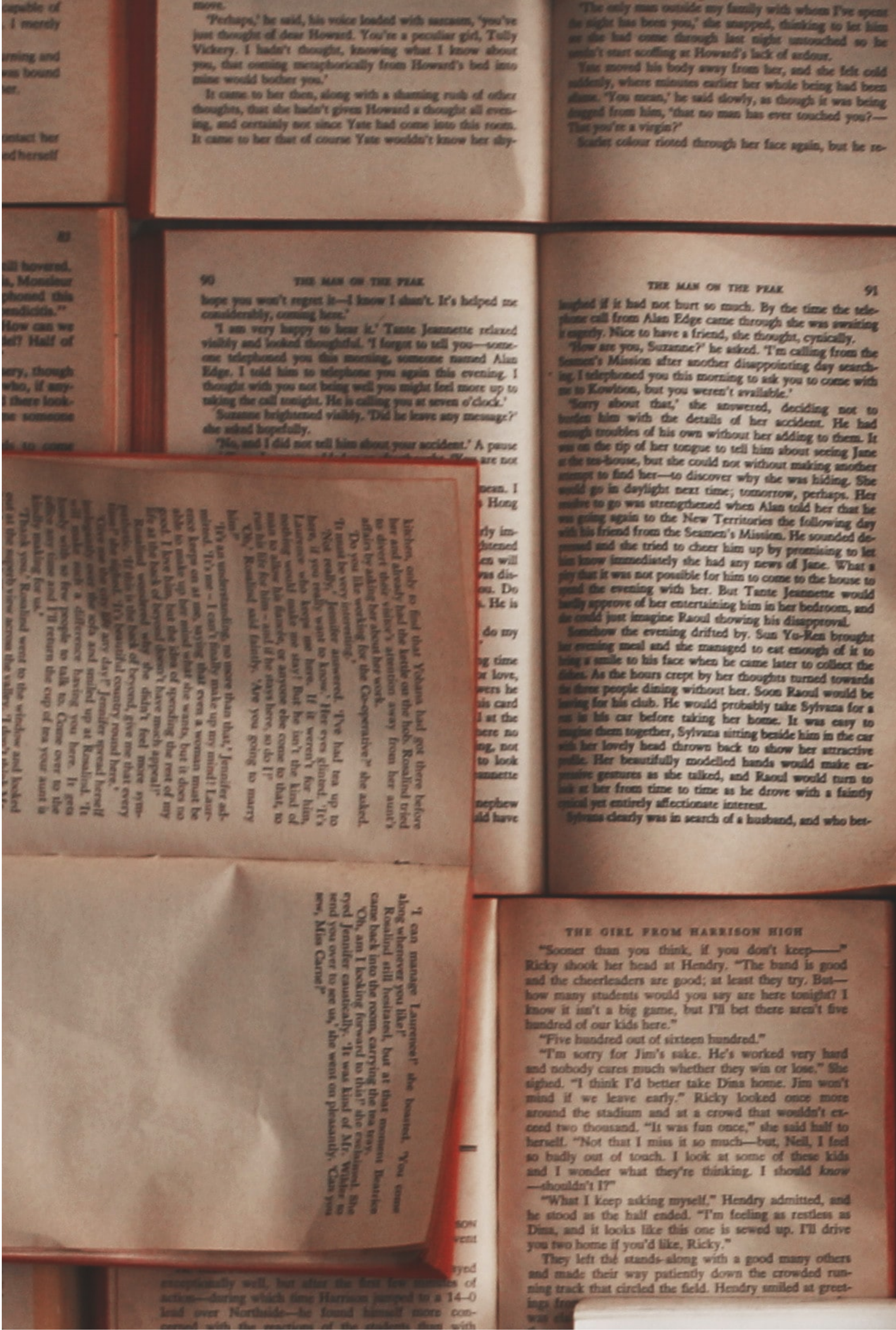
Decision Tree Algorithm -A Complete Guide

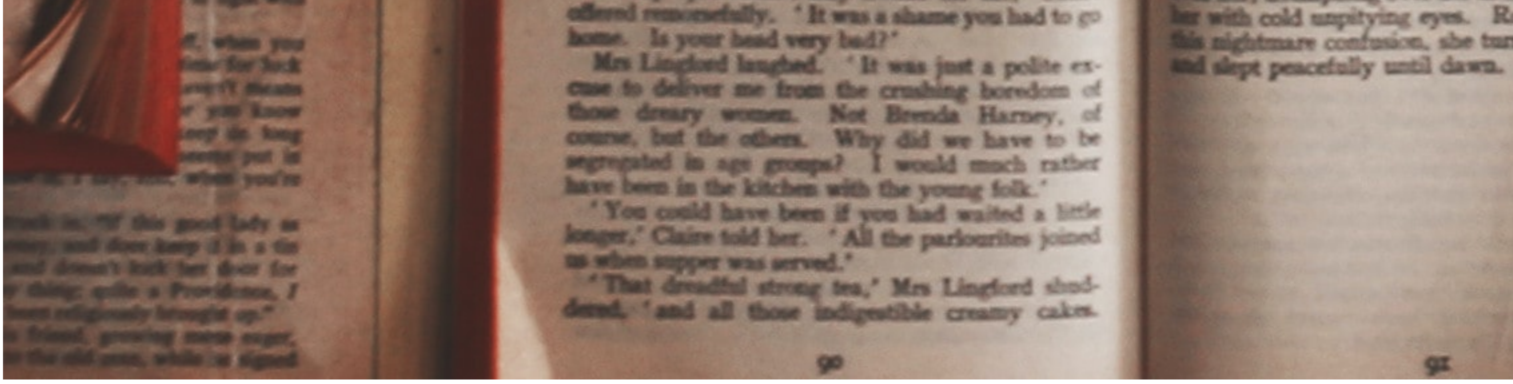
ALGORITHM BEGINNER CLASSIFICATION DATA SCIENCE MACHINE LEARNING SUPERVISED

This article was published as a part of the Data Science Blogathon



'Perhaps,' he said, his voice loaded with sarcasm, 'you've just thought of dear Howard. You're a peculiar girl, Tully Vickery. I hadn't thought, knowing what I know about you, that coming metaphorically from Howard's bed into mine would bother you.'
It came to her then, along with a shaming rush of other thoughts, that she hadn't given Howard a thought all evening, and certainly not since Yate had come into this room. It came to her that of course Yate wouldn't know her shy-
'The only man outside my family with whom I've spent the night has been you,' she snapped, thinking to let him see she had come through last night untouched so he couldn't start scoffing at Howard's lack of ardour.
Yate moved his body away from her, and she felt cold suddenly, where minutes earlier her whole being had been aflame. 'You mean,' he said slowly, as though it was being dragged from him, 'that no man has ever touched you?—That you're a virgin?'
Scarlet colour rioted through her face again, but he re-
90 THE MAN ON THE PEAK
hope you won't regret it—I know I shan't. It's helped me considerably, coming here.'
'I am very happy to hear it.' Tante Jeannette relaxed visibly and looked thoughtful. 'I forgot to tell you—someone telephoned you this morning, someone named Alan Edge. I told him to telephone you again this evening. I thought with you not being well you might feel more up to taking the call tonight. He is calling you at seven o'clock.'
Suzanne brightened visibly. 'Did he leave any message?' she asked hopefully.
'No, and I did not tell him about your accident.' A pause
THE MAN ON THE PEAK 91
laughed if it had not hurt so much. By the time the telephone call from Alan Edge came through she was awaiting it eagerly. Nice to have a friend, she thought, cynically.
'How are you, Suzanne?' he asked. 'I'm calling from the Seamen's Mission after another disappointing day searching. I telephoned you this morning to ask you to come with me to Kowloon, but you weren't available.'
'Sorry about that,' she answered, deciding not to burden him with the details of her accident. He had enough troubles of his own without her adding to them. It was on the tip of her tongue to tell him about seeing Jane at the tea-house, but she could not without making another attempt to find her—to discover why she was hiding. She would go in daylight next time; tomorrow, perhaps. Her resolve to go was strengthened when Alan told her that he was going again to the New Territories the following day with his friend from the Seamen's Mission. He sounded depressed and she tried to cheer him up by promising to let him know immediately she had any news of Jane. What a pity that it was not possible for him to come to the house to spend the evening with her. But Tante Jeannette would hardly approve of her entertaining him in her bedroom, and she could just imagine Raoul showing his disapproval.
Somehow the evening drifted by. Sun Yu-Ren brought her evening meal and she managed to eat enough of it to bring a smile to his face when he came later to collect the dishes. As the hours crept by her thoughts turned towards the three people dining without her. Soon Raoul would be leaving for his club. He would probably take Sylvana for a run in his car before taking her home. It was easy to imagine them together, Sylvana sitting beside him in the car with her lovely head thrown back to show her attractive profile. Her beautifully modelled hands would make expressive gestures as she talked, and Raoul would turn to look at her from time to time as he drove with a faintly cynical yet entirely affectionate interest.
Sylvana clearly was in search of a husband, and who bet-
THE GIRL FROM HARRISON HIGH
"Sooner than you think, if you don't keep——"
Ricky shook her head at Hendry. "The band is good and the cheerleaders are good; at least they try. But—how many students would you say are here tonight? I know it isn't a big game, but I'll bet there aren't five hundred of our kids here."
"Five hundred out of sixteen hundred."
"I'm sorry for Jim's sake. He's worked very hard and nobody cares much whether they win or lose." She sighed. "I think I'd better take Dina home. Jim won't mind if we leave early." Ricky looked once more around the stadium and at a crowd that wouldn't exceed two thousand. "It was fun once," she said half to herself. "Not that I miss it so much—but, Neil, I feel so badly out of touch. I look at some of these kids and I wonder what they're thinking. I should know—shouldn't I?"
"What I keep asking myself," Hendry admitted, and he stood as the half ended. "I'm feeling as restless as Dina, and it looks like this one is sewed up. I'll drive you two home if you'd like, Ricky."
They left the stands along with a good many others and made their way patiently down the crowded running track that circled the field. Hendry smiled at greet-


the game itself. After two quarters he decided rather glumly that if school spirit at Harrison wasn't totally dead, it didn't have far to go.
He turned to Ricky Trent, who was sitting next to him and trying to keep a bored Dina quiet. Ricky looked as sad as he felt.
"Nobody seems to be having much fun, do they?" he said, trying a smile.
Ricky brushed syrup from Dina's popsickle off the sleeve of her dress. "Maybe I'm just getting old, but it was different, wasn't it, when—Dina, be still until I clean up this mess."
"When are we going home, Mo-ther?"
132
'I'll walk across the fields with you,' Dermot offered.
She was surprised at this courtesy. 'I'll just fetch my coat,' she said. Taking it from the stand in the hall, she remembered little Maeve's remark about high hats and was foolishly annoyed again.
'I can see myself home, thanks,' she told Dermot coldly when she rejoined him—a remark he ignored. With a touch on her arm he piloted her through the garden gate, calling over his shoulder to a group milling about the moonlit yard: 'I'll be back in a tick, Sheila. I'm just seeing Miss Lingford across the fields.'
Miss Lingford! And everyone else had been calling her Claire all the evening. What a stuffed shirt the man was! They walked in heavy silence. There was no magic in the moonlight now, nor in the night scent of flowers and dew wet grass. In the buttercup field the three horses lifted heads to whinny as they passed. Bats wheeled in the warm air. At the Castle door their goodnight was brief. 'Don't let me keep you from Sheila,' Claire said.
She found her mother sitting up in bed, reading.



Introduction

Till now we have learned about linear regression, logistic regression, and they were pretty hard to understand. Let's now start with Decision tree's and I assure you this is probably the easiest algorithm in Machine Learning. There's not much mathematics involved here. Since it is very easy to use and interpret it is one of the most widely used and practical methods used in Machine Learning.

Contents

What is a Decision Tree?

It is a tool that has applications spanning several different areas. Decision trees can be used for classification as well as regression problems. The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.

Image Source: https://wiki.pathmind.com/decision-tree

Before learning more about decision trees let’s get familiar with some of the terminologies.

Root Nodes – It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.

Decision Nodes – the nodes we get after splitting the root nodes are called Decision Node

Leaf Nodes – the nodes where further splitting is not possible are called leaf nodes or terminal nodes

Sub-tree – just like a small portion of a graph is called sub-graph similarly a sub-section of this decision tree is called sub-tree.

Pruning – is nothing but cutting down some nodes to stop overfitting.

Image source: https://wiki.pathmind.com/decision-tree

Example of a decision tree.

Let’s understand decision trees with the help of an example.

Image Source: www.hackerearth.com

Decision trees are upside down which means the root is at the top and then this root is split into various several nodes. Decision trees are nothing but a bunch of if-else statements in layman terms. It checks if the condition is true and if it is then it goes to the next node attached to that decision.

In the below diagram the tree will first ask what is the weather? Is it sunny, cloudy, or rainy? If yes then it will go to the next feature which is humidity and wind. It will again check if there is a strong wind or weak, if it’s a weak wind and it’s rainy then the person may go and play.

Image Source: www.hackerearth.com

Did you notice anything in the above flowchart? We see that if the *weather is cloudy* then we must go to play. Why didn't it split more? Why did it stop there?

To answer this question, we need to know about few more concepts like entropy, information gain, and Gini index. But in simple terms, I can say here that the output for the training dataset is always yes for cloudy weather, since there is no disorderliness here we don't need to split the node further.

The goal of machine learning is to decrease uncertainty or disorders from the dataset and for this, we use decision trees.

Now you must be thinking how do I know what should be the root node? what should be the decision node? when should I stop splitting? To decide this, there is a metric called "Entropy" which is the amount of uncertainty in the dataset.

Entropy

Entropy is nothing but the uncertainty in our dataset or measure of disorder. Let me try to explain this with the help of an example.

Suppose you have a group of friends who decides which movie they can watch together on Sunday. There are 2 choices for movies, one is ***"Lucy"*** and the second is ***"Titanic"*** and now everyone has to tell their choice. After everyone gives their answer we see that *"Lucy" gets 4 votes* and *"Titanic" gets 5 votes*. Which movie do we watch now? Isn't it hard to choose 1 movie now because the votes for both the movies are somewhat equal.

This is exactly what we call disorderness, there is an equal number of votes for both the movies, and we can't really decide which movie we should watch. It would have been much easier if the votes for "Lucy" were 8 and for "Titanic" it was 2. Here we could easily say that the majority of votes are for "Lucy" hence everyone will be watching this movie.

In a decision tree, the output is mostly "yes" or "no"

The formula for Entropy is shown below:

Here p_+ is the probability of positive class

p_- is the probability of negative class

S is the subset of the training example

How do Decision Trees use Entropy?

Now we know what entropy is and what is its formula, Next, we need to know that how exactly does it work in this algorithm.

Entropy basically measures the impurity of a node. Impurity is the degree of randomness; it tells how random our data is. A **pure sub-split** means that either you should be getting "yes", or you should be getting "no".

Suppose *feature 1* had 8 yes and 4 no, after the split *feature 2 get 5 yes and 2 no* whereas f*eature 3 gets 3 yes and 2 no*.

We see here the split is not pure, why? Because we can still see some negative classes in both the feature. In order to make a decision tree, we need to calculate the impurity of each split, and when the purity is 100% we make it as a leaf node.

To check the impurity of feature 2 and feature 3 we will take the help for Entropy formula.

For feature 2 the entropy is as follows:

Image Source: Author

For feature 3,

We can clearly see from the tree itself that feature 2 has low entropy or more purity than feature 3 since feature 2 has more "yes" and it is easy to make a decision here.

Always remember that the higher the Entropy, the lower will be the purity and the higher will be the impurity.

As mentioned earlier the goal of machine learning is to decrease the uncertainty or impurity in the dataset, here by using the entropy we are getting the impurity of a feature or a particular node, we don't know if the parent entropy or the entropy of a particular node has decreased or not.

For this, we bring a new metric called "Information gain" which tells us how much the parent entropy has decreased after splitting it with some feature.

To read more about Entropy you can read this article.

Information Gain

Information gain measures the reduction of uncertainty given some feature and it is also a deciding factor for which attribute should be selected as a decision node or root node.

It is just entropy of the full dataset – entropy of the dataset given some feature.

To understand this better let's consider an example:
Suppose our entire population has a total of 30 instances. The dataset is to predict whether the person will go to the gym or not. Let's say 16 people go to the gym and 14 people don't

Now we have two features to predict whether he/she will go to the gym or not.

Feature 1 is **"Energy"** which takes two values *"high" and "low"*

Feature 2 is **"Motivation"** which takes 3 values *"No motivation", "Neutral" and "Highly motivated"*.

Let's see how our decision tree will be made using these 2 features. We'll use information gain to decide which feature should be the root node and which feature should be placed after the split.

Image Source: Author

Let’s calculate the entropy:

To see the weighted average of entropy of each node we will do as follows:

Now we have the value of E(Parent) and E(Parent|Energy), information gain will be:

Our parent entropy was near 0.99 and after looking at this value of information gain, we can say that the entropy of the dataset will decrease by 0.37 if we make “Energy” as our root node.

Similarly, we will do this with the other feature “Motivation” and calculate its information gain.

Image Source: Author

Let’s calculate the entropy here:

To see the weighted average of entropy of each node we will do as follows:

Now we have the value of E(Parent) and E(Parent|Motivation), information gain will be:

We now see that the “Energy” feature gives more reduction which is 0.37 than the “Motivation” feature. Hence we will select the feature which has the highest information gain and then split the node based on that feature.

Image Source: Author

In this example “Energy” will be our root node and we’ll do the same for sub-nodes. Here we can see that when the energy is “high” the entropy is low and hence we can say a person will definitely go to the gym if he has high energy, but what if the energy is low? We will again split the node based on the new feature which is “Motivation”.

When to stop splitting?

You must be asking this question to yourself that when do we stop growing our tree? Usually, real-world datasets have a large number of features, which will result in a large number of splits, which in turn gives a huge tree. Such trees take time to build and can lead to overfitting. That means the tree will give very good accuracy on the training dataset but will give bad accuracy in test data.

There are many ways to tackle this problem through hyperparameter tuning. We can set the maximum depth of our decision tree using the ***max_depth*** parameter. The more the value of ***max_depth***, the more complex your tree will be. The training error will off-course decrease if we increase the ***max_depth*** value but when our test data comes into the picture, we will get a very bad accuracy. Hence you need a value that will not overfit as well as underfit our data and for this, you can use GridSearchCV.

Another way is to set the minimum number of samples for each spilt. It is denoted by ***min_samples_split***. Here we specify the minimum number of samples required to do a spilt. For example, we can use a minimum of 10 samples to reach a decision. That means if a node has less than 10 samples then using this parameter, we can stop the further splitting of this node and make it a leaf node.

There are more hyperparameters such as :

min_samples_leaf – represents the minimum number of samples required to be in the leaf node. The more you increase the number, the more is the possibility of overfitting.

max_features – it helps us decide what number of features to consider when looking for the best split.

To read more about these hyperparameters you can read it here.

Pruning

It is another method that can help us avoid overfitting. It helps in improving the performance of the tree by cutting the nodes or sub-nodes which are not significant. It removes the branches which have very low importance.

There are mainly 2 ways for pruning:

(i) **Pre-pruning** – we can stop growing the tree earlier, which means we can prune/remove/cut a node if it has low importance **while growing** the tree.

(ii) **Post-pruning** – once our **tree is built to its depth**, we can start pruning the nodes based on their significance.

Endnotes

To summarize, in this article we learned about decision trees. On what basis the tree splits the nodes and how to can stop overfitting. why linear regression doesn't work in the case of classification problems.

In the next article, I will explain Random forests, which is again a new technique to avoid overfitting.
To check out the full implementation of decision trees please refer to my Github repository.

Let me know if you have any queries in the comments below.

About the Author

I am an undergraduate student currently in my last year majoring in Statistics (Bachelors of Statistics) and have a strong interest in the field of data science, machine learning, and artificial intelligence. I enjoy diving into data to discover trends and other valuable insights about the data. I am constantly learning and motivated to try new things.
I am open to collaboration and work.

For any **doubt and queries,** feel free to contact me on Email

Connect with me on LinkedIn and Twitter

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

Article Url - https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

anshul508