**Decision Trees**

Decision Trees are a simple yet powerful algorithm for classification. They mirror the logical decision-making process that humans naturally use—often without even realizing it. One way to visualize and analyze a Decision Tree in action is through a familiar game: ***Guess Who***.

Break into small teams of 2-4 students and grab a game of *Guess Who*. Take turns playing a few rounds. As you are playing (or watching over someone’s shoulder), evaluate the algorithm that you are using.

1. What is your basic strategy?
2. What types of questions do you ask first?
3. How do you decide which question to ask next?
4. Do you follow a pattern, or do you adjust your approach as the game progresses?
5. Does anyone on your team have a different strategy? If so, describe it here.

**Connecting *Guess Who* to Decision Trees**

In the game *Guess Who*, each character represents a class label (y) that we are trying to predict, while their physical characteristics serve as the features for the input data (X). Interestingly, the game itself works in the opposite direction of a traditional machine-learning prediction.

* In a decision tree model, a user provides input features (e.g., hair color, glasses, hat), and the model predicts which class (character) the features belong to.
* In *Guess Who*, a player chooses a class (character) and must devise questions about features to eliminate possibilities, working backward to narrow down the correct answer.

This highlights how decision trees function: they systematically split data based on features that best separate different classes.

A standard *Guess Who* game contains 24 different characters, which would be similar to a training set with 24 samples and 24 classes—one per sample. While technically possible, this kind of distribution is unrealistic in most machine-learning datasets. A more typical training set might group samples into 4–6 different classes, where each class contains similar samples.

**Activity: Creating Your Own Class Groups**

Work together as a team to divide the *Guess Who* characters into 4–6 groups based on patterns or rules. It is important that you do not assign characters arbitrarily. Your groups should be based on clear, shared characteristics. Once your team has defined the groups, each person should work individually to describe the groups.

1. Write out each of your groups below.
   * Assign a label or name to each group.
   * Describe the defining features that determine group membership. Your *Guess Who* game probably has names for each character. Do not use these names to identify the individual members. Instead, you must describe them. There’s a table on the next page to help you.
   * Hint: The order that you describe the features matters. For example, you might notice that every person in a group is a girls wearing a green shirt and sunglasses. But there are also boys wearing green shirts and other girls wearing sunglasses. None of the features, on their own, is enough to distinguish one group from the other. Instead, it is the combination of characteristics. There is probably a hierarchical description with some features that must be distinguished first and others later.

|  |  |  |
| --- | --- | --- |
| Group Name or Class Label | Count | Description of Members |
| Little Rascals  (or an ordinal) | 9 | Elementary school age Grass stains on the knees Kool-Aid stains on the corners of mouth Cute but sneaky expression Wearing a ball cap |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Once everyone has created their groups, share them with your teammates. As you listen to each person’s explanation, analyze whether there is a way to simplify or refine the descriptions. For example: A teammate describes a group as “superheroes with an animal theme wearing bow ties.” But upon closer inspection, those are the only characters wearing a mask—a more efficient description.

1. Did your teammates improve any of your descriptions? Write down any improved (shortened) descriptions suggested by your teammates.

**Training a Decision Tree**

The training process for a decision tree model mirrors the thought process you just used to classify the *Guess Who* characters.

* Given a set of input samples (X) and categories (y), you identified rules that distinguished each group from the others.
* A decision tree algorithm does something similar—it finds the most informative features and structures them into a sequence of "questions" that guide classification.

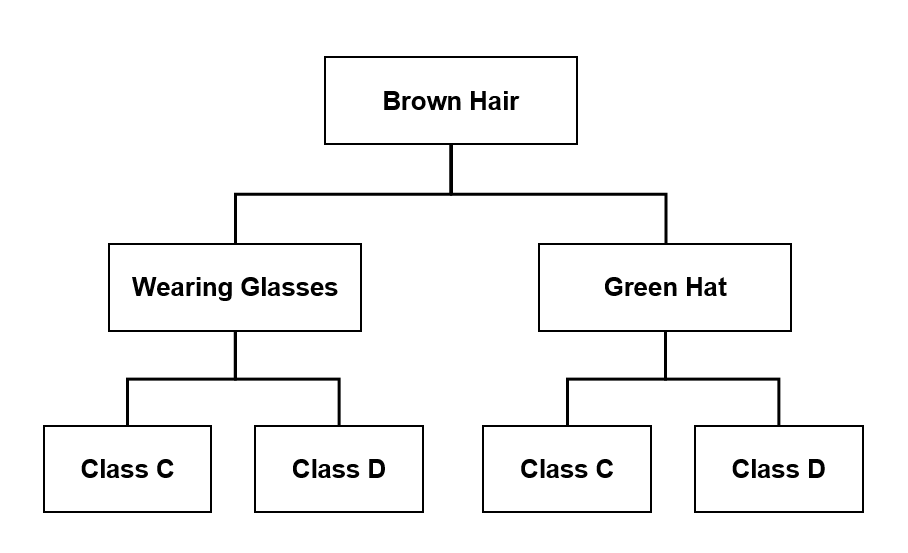
Just like in your grouping activity, the order of the questions matters—some features are more useful for quickly narrowing down the possibilities. The tree organizes these questions into a flowchart-like structure, which the model follows when making future predictions.

**Activity: Building a *Guess Who* Decision Tree**

Now, you will draw a decision tree as a flowchart that classifies *Guess Who* characters based on the groups you created earlier.

1. On the next page, draw a flowchart that models your decision tree with the description/ characteristics that you individually chose and your teammates helped you refine.

* Start with a root question by choose a feature that splits the characters into two large groups. This question should be based on the top-level characteristics that you previously identified.
* Continue splitting at each node. Select the most distinguishing question for the remaining characters in that branch. Each question should be expressed as the logic in a computer branching statement with True/False answers.
* Keep the tree as balanced as possible. Try to make each split divide the remaining characters into roughly equal-sized groups whenever possible. Avoid questions that eliminate only one or two characters at a time, unless necessary.
* End with class labels at the leaf nodes. The final nodes of your tree should correspond to your previously created groups (not individual characters).



**Your Decision Tree "Flowchart"**

**The Power of Multiple Decision Trees**

Compare your decision tree to the flowcharts created by your teammates. Even though everyone on your team grouped the characters into the same classes, each person likely structured their flowchart a little differently. While some questions and patterns may be similar, the specific features chosen and the order of questions will vary.

Now, imagine if the game included thousands of characters (samples) instead of just 24. The variation between individual decision trees would increase even more. Some characters might fit into multiple groups depending on subtle differences in clothing style, facial expressions, hair color, or even the order of distinguishing questions.

While any single decision tree would likely classify most characters correctly, edge cases—characters that don’t fit neatly into one category—might be misclassified. This same problem occurs in machine learning when relying on just one decision tree.

One way to reduce errors and improve accuracy is to follow multiple decision trees and predict the majority answer. Imagine using each of your team’s flowcharts to classify a character. No single diagram is perfect, but by combining them, you get a more reliable classification.

This is the core idea behind Random Forests—a machine learning model that builds multiple decision trees and combines their predictions to make more accurate and stable classifications.

**Application to Ethics in AI**

Decision Trees are one of the most satisfying machine learning models because their internal logic is easy to understand and visualize. Most libraries include a feature to draw out the trained logic as a flow chart. This high level of “explainability” is rare in the field of artificial intelligence and makes them an ideal choice for applications where transparency is of utmost importance.

1. What fields of Artificial Intelligence would benefit most from such transparent models?

**Worksheet Summary by Generative AI**

Decision Trees provide an intuitive way to classify data by breaking it down into a series of logical questions. However, as you've seen, different decision trees can lead to slightly different results. By combining multiple trees, we can improve accuracy and reduce errors—a concept that forms the basis of Random Forests. This exercise demonstrates not only how decision trees work but also why they remain one of the most interpretable machine-learning models. Their ability to visually represent decision-making makes them a powerful tool in both artificial intelligence and real-world problem-solving.