Branching

A decision tree is a map of the possible outcomes of a series of related choices. It allows an individual or organization to weigh possible actions against one another based on their costs, probabilities, and benefits. They can can be used either to drive

informal discussion or to map out an algorithm that predicts the best choice mathematically.

A decision tree typically starts with a single node, which branches into possible outcomes. Each of those outcomes leads to additional nodes, which branch off into other possibilities. This gives it a treelike shape.

There are three different types of nodes: chance nodes, decision nodes, and end nodes. A chance node, represented by a circle, shows the probabilities of certain results. A decision node, represented by a square, shows a decision to be made, and an end node shows the final outcome of a decision path.

Decision trees can also be drawn with flowchart symbols, which some people find easier to read and understand.

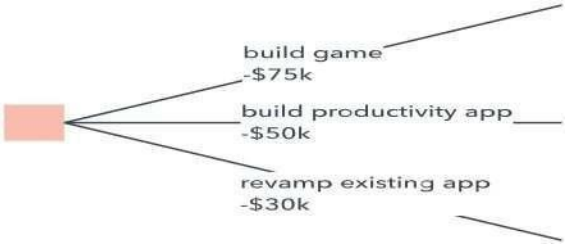
Decision tree symbols

Shape	Name	Meaning
	Decision node	Indicates a decision to be made
	Chance node	Shows multiple uncertain outcomes
	Alternative branches	Each branch indicates a possible outcome or action
	Rejected alternative	Shows a choice that was not selected
	Endpoint node	Indicates a final outcome

How to draw a decision tree

To draw a decision tree, first pick a medium. You can draw it by hand on paper or a whiteboard, or you can use special decision tree software. In either case, here are the steps to follow:

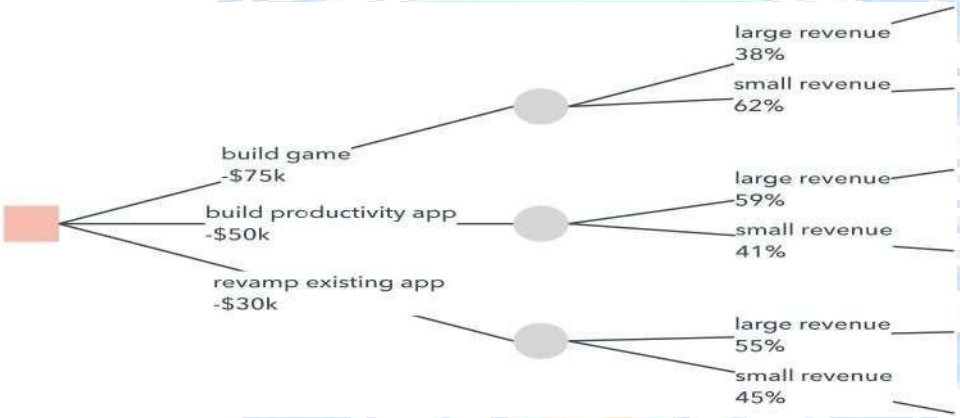
1. **Start with the main decision**. Draw a small box to represent this point, then draw a line from the box to the right for each possible solution or action. Label them accordingly.



2. **Add chance and decision nodes** to expand the tree as follows:

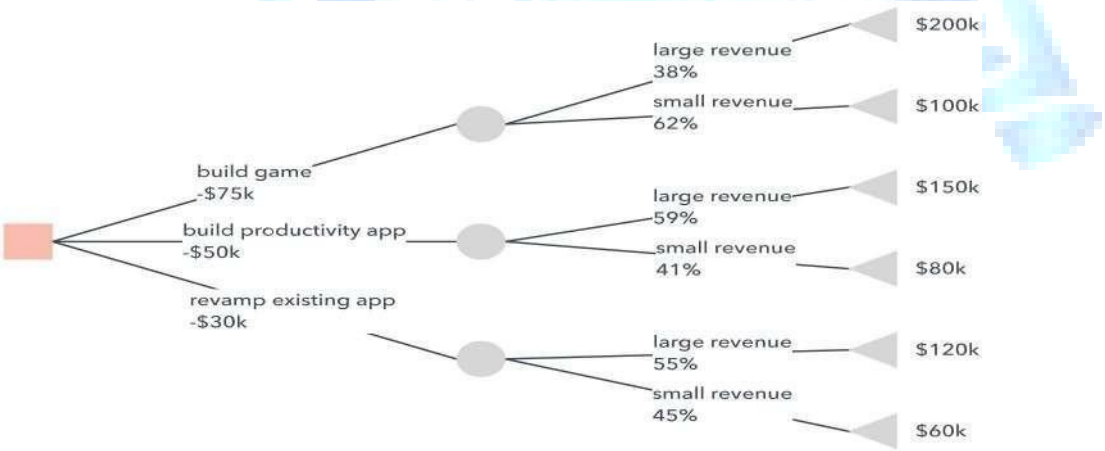
- If another decision is necessary, draw another box.
- If the outcome is uncertain, draw a circle (circles represent chance nodes).

If the problem is solved, leave it blank (for now)



From each decision node, draw possible solutions. From each chance node, draw lines representing possible outcomes. If you intend to analyze your options numerically, include the probability of each outcome and the cost of each action.

3. **Continue to expand until every line reaches an endpoint**, meaning that there are no more choices to be made or chance outcomes to consider. Then, assign a value to each possible



outcome. It could be an abstract score or a financial value. Add triangles to signify endpoints.

With a complete decision tree, you're now ready to begin analyzing the decision you face.

Advantages and disadvantages

Decision trees remain popular for reasons like these:

- How easy they are to understand
- They can be useful with or without hard data, and any data requires minimal preparation
- New options can be added to existing trees
- Their value in picking out the best of several options
- How easily they combine with other decision making tools

However, decision trees can become excessively complex. In such cases, a more compact influence diagram can be a good alternative. Influence diagrams narrow the focus to critical decisions, inputs, and objectives.

Multiple Branches

Multiclass classification is a popular problem in supervised machine learning.

Problem – Given a dataset of **m** training examples, each of which contains information in the form of various features and a label. Each label corresponds to a class, to which the training example belongs. In multiclass classification, we have a finite set of classes. Each training example also has **n** features.

For example, in the case of identification of different types of fruits, "Shape", "Color", "Radius" can be featured, and "Apple", "Orange", "Banana" can be different class labels.

A decision tree classifier is a systematic approach for multiclass classification. It poses a set of questions to the dataset (related to its attributes/features). The decision tree classification algorithm can be visualized on a binary tree. On the root and each

of the internal nodes, a question is posed and the data on that node is further split into separate records that have different characteristics. The leaves of the tree refer to the classes in which the dataset is split.

Greedy Algorithm

What Does Greedy Algorithm Mean?

A greedy algorithm is an algorithmic strategy that makes the best optimal choice at each small stage with the goal of this eventually leading to a globally optimum solution. This means that the algorithm picks the best solution at the moment without regard for consequences. It picks the best immediate output, but does not consider the big picture, hence it is considered greedy.

A greedy algorithm works by choosing the best possible answer in each step and then moving on to the next step until it reaches the end, without regard for the overall solution. It only hopes that the path it

takes is the globally optimum one, but as proven time and again, this method does not often come up with a globally optimum solution. In fact, it is entirely possible that the most optimal short-term solutions lead to the worst possible global outcome.

Think of it as taking a lot of shortcuts in a manufacturing business: in the short term large amounts are saved in manufacturing cost, but this eventually leads to downfall since quality is compromised, resulting in product returns and low sales as customers become acquainted with the “cheap” product. But this is not always the case, there are a lot of applications where the greedy algorithm works best to find or approximate the globally optimum solution such as in constructing a Huffman tree or a decision learning tree.

For example: Take the path with the largest sum overall. A greedy algorithm would take the blue path, as a result of shortsightedness, rather than the orange path, which yields the largest sum.