* [Use a decision tree](#gjdgxs)
* [Imbalanced datasets](#30j0zll)

# CSE 450 - Machine Learning & Data Mining

## Module 03 — More Hints

Don't Sell Yourself Short

Coding and Data Science is hard. Most of what I'm going to put here, you could look up on your own from official docs, tutorials, and Stack Overflow questions.

As we proceed throughout the semester, you'll get fewer hints and be expected to look more stuff up on your own.

If you haven't tried independent research yet, you might want to try that for a few hours, then come back here.

## Use a decision tree

If you're really struggling with how to make a decision tree, you should try reading through the documentation a couple more times.

If that still doesn't help, this notebook can give you some more guidance:



## Imbalanced datasets

Often in classification problems, you'll build a model that seems to do fantastically well when testing. Sometimes it'll seem like no matter what you do with the model, you get high levels of accuracy.

When this happens, it is often because of an imbalanced dataset — one where you have many more instances of one target value than another.

For example, if 80% of my samples report true for the target, I could achieve close to an 80% accuracy rate on my test data just by using this code:

def true\_or\_false\_for\_this\_sample(sample):  
 return True

There are [several ways to address this problem](https://stats.stackexchange.com/questions/28029/training-a-decision-tree-against-unbalanced-data), but two of the most common are oversampling and undersampling.

This Colab notebook shows an example:

