**. Information Gain**

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

*Steps Involved*

**Step 1:**

Calculate entropy of the target.

**Step 2:**

The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain, or decrease in entropy.

**Step 3:**

Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

**Pros:**

Computionally cheap to use, easy for humans to understand results and it can deal with irrelevant feaures also

**Cons:**

Prone to Overfitting.(It refers to the process when models is trained on training data too well that any noise in testing data can bring negative impacts to performance of model.)

**In Nutshell**

A decision tree classifier is just like a flowchart diagram with the terminal nodes representing classification outputs/decisions. Starting with a dataset, you can measure the entropy to find a way to split the set until all the data belonngs to the same class. There are several approaches to decision trees like ID3, C4.5, CART and many more. For splitting nominal valued datasets you can use the ID3 algorithm. You can use matplotlib library to visualize the tree data. Decision Trees are prone to overfitting, thus to avoid overfitting you can prune the decision tree by combining the adjacent nodes that have low information gain.

Scikit learn

scracth