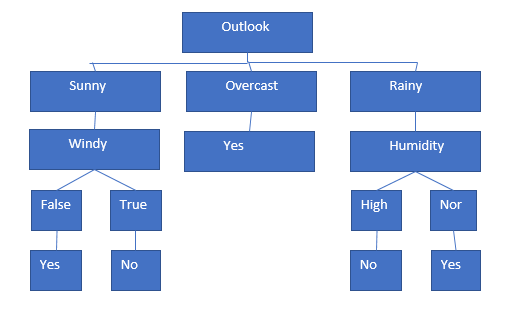
A decision tree is used for classification or regression problems. A Decision tree essentially breaks down a dataset into smaller subsets, simultaneously developing a tree-like structure. The final result happens to be a tree with decision nodes and leaf nodes. A decision node is a node that has branches while a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees are especially useful because they can handle categorical and numerical data.

Consider the following Table of Data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outlook | Temprature | Humidity | Windy | Play Outside |
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Overcast | Hot | High | FALSE | Yes |
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Overcast | Cool | Normal | TRUE | Yes |
| Rainy | Mild | High | FALSE | No |
| Rainy | Cool | Normal | FALSE | Yes |
| Sunny | Mild | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |
| Overcast | Mild | High | TRUE | Yes |
| Overcast | Hot | Normal | FALSE | Yes |
| Sunny | Mild | High | TRUE | No |

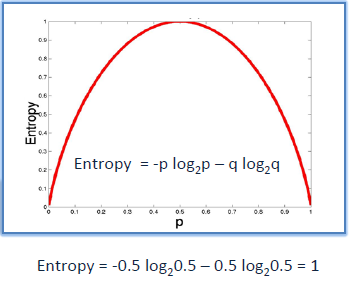
Through an algorithm that generates a Decision Tree, this would generate the following:



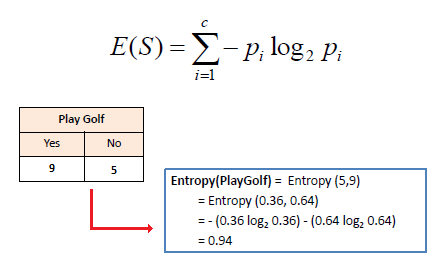
The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses *Entropy* and *Information Gain* to construct a decision tree.

**Entropy:**

ID3 algorithm uses entropy to calculate the homogeneity of a sample

****

* Entropy using the frequency table of one attribute



**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).

|  |
| --- |
| *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.  *Step 4a*: A branch with entropy of 0 is a leaf node.  *Step 4b*: A branch with entropy more than 0 needs further splitting.  *Step 5*: The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified. |
|  |

The script (decision\_tree.py) contains the code that generates a decision tree based on the data that is assumed to be fetched from a number of hits to the website. At first, the entropy of the entire data set is calculated. Then the dataset is split into columnar values, such as source, country, # views etc. and the entropy for each one of them is calculated and added proportionately to get total entropy for the split. This is then used to calculate the information gain.

This is then ranked, and the attribute with the largest information gain is considered as the decision node. If a branch has an entropy of 0, it means it is the leaf node (in our cases, that’s the ‘Type’ of subscription). We calculate the entropy of numerical values in the form of splitting into buckets and calculating the entropy, then taking the best entropy there is.

After the splits have been ranked, we recursively call a ‘build\_tree’ function, which makes our decision nodes from the decisionnode class. This is then plotted using PIL’s image plotter (some custom width and height handling functions were written)