Decision Trees

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter.
Decision Tree Classifier, repetitively divides the working area(plot) into sub part by identifying lines. (repetitively because there may be two distant regions of same class divided by other).

There are two main types of Decision Trees:

1. **Classification trees** (Yes/No types)

What we've seen above is an example of classification tree, where the outcome was binary. Here the decision variable is **Categorical**.

2. **Regression trees** (Continuous data types)

   Here the decision or the outcome variable is **Continuous**, e.g. that is, any number.

The following example uses the **ID3 Algorithm**. ID3 Stands for **Iterative Dichotomiser 3**.

Entropy

Entropy, also called as Shannon Entropy is denoted by H(S) for a finite set S, is the measure of the amount of uncertainty or randomness in data.

$$\boxed{H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}}$$

OR

$$H = -\sum p(x) \log p(x)$$

Intuitively, it tells us about the predictability of a certain event. Example, consider a coin toss whose probability of heads is 0.5 and probability of tails is 0.5. Here the entropy is the highest possible, since there's no way of determining what the outcome might be. Alternatively, consider a coin which has heads on both the sides, the entropy of such an event can be predicted perfectly since we know beforehand that it'll always be heads. In other words, this event has **no randomness** hence it's entropy is zero.

Information Gain

Information gain is also called as Kullback-Leibler divergence denoted by IG(S,A) for a set S is the effective change in entropy after deciding on a particular attribute A. It measures the relative change in entropy with respect to the independent variables.

$$\boxed{IG(S,A) = H(S) - H(S,A)}$$

Alternatively,

$$\boxed{IG(S,A) = H(S) - \sum_{i=0}^{n} P(x) * H(x)}$$

where IG(S, A) is the information gain by applying feature A. H(S) is the Entropy of the entire set, while the second term calculates the Entropy after applying the feature A, where P(x) is the probability of event x.

That is pretty much all the math we need for forming decision trees.

EXAMPLE:

Suppose we have a following data for playing a golf on various conditions:

Predictors				Decision
Outlook	**Temperature**	**Humidity**	**Windy**	**Play Golf?**
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

Now if the weather condition is given as:

Outlook: Rainy, **Temperature**: Cool, **Humidity**: High, **Windy**: False

Should we play golf?

We have outcomes at beginning as NNYYYNYN (Y = Yes and N = No) taken in given order. Entropy at this root node is 0.3
Now try to divide on various predictors outlook, temperature, humidity and Windy. Calculate the information gain in each case. Which one has highest information gain?

For example, if we divide based on Outlook, we have divisions as
Rainy : NNN (entropy = 0)
Sunny : YYN (entropy = 0.041)
Overcast : YY (entropy = 0)

So information gain = 0.3 - [0 + (3/8)*0.041 + 0]

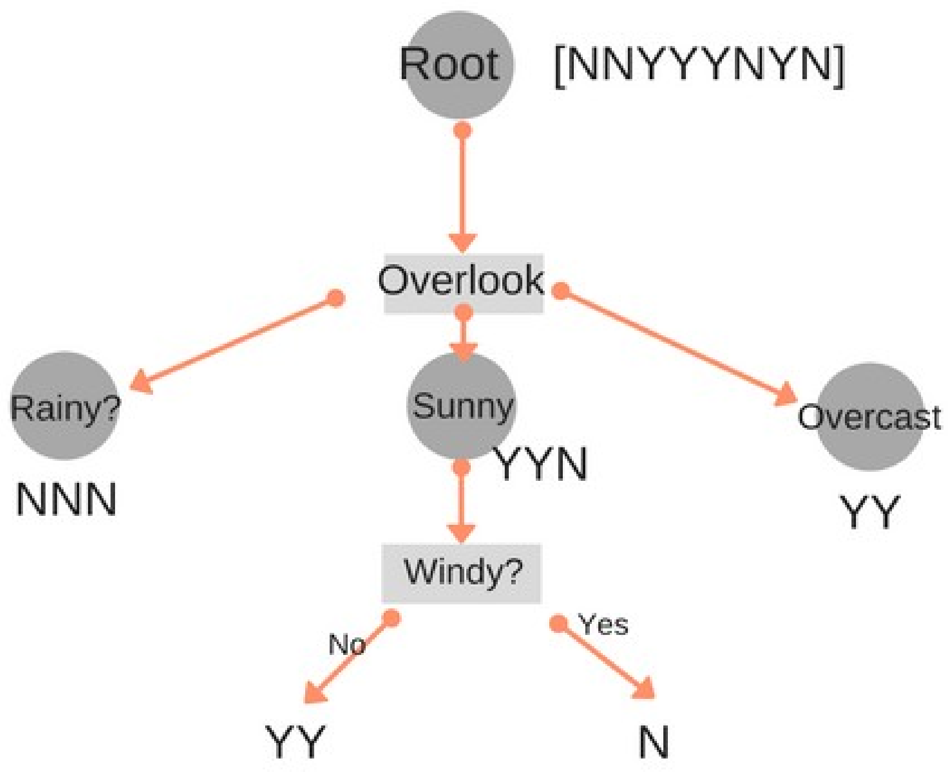
= 0.28

Try out for other cases.

The information gain is max when divided based on Outlook.
Now the impurity for Rainy and Overcast is 0. We stop for them here.

Next we need to separate Sunny,
If we divide by Windy, we get max information gain.
Sunny
YYN
Windy? Yes : N
No : YY

So decision tree look like something as shown in image below.



Now the prediction data is
Outlook : Rainy, **Temperature**: Cool, **Humidity**: High, **Windy**: FalseFlowing down from tree according to result, we first check Rainy? So answer is No, we don't play golf.

Advantages of using Decision tree classifiers:

1. Decision trees can be used to predict both continuous and discrete values i.e. they work well for both regression and classification tasks.
2. They require relatively less effort for training the algorithm.
3. They can be used to classify non-linearly separable data.
4. They're very fast and efficient compared to KNN and other classification algorithms.