TEXT ANALYSIS ASSESSMENT
Knowing which attributes are the most predictive

There are two ways to assess which attributes are most predictive: look at the classification model itself; or using an attribute weighting algorithm

1. **Looking at the classification model:**

Some classification models give feedback on which attributes were most predictive.

- **Decision Tree**: the most predictive terms will be towards the top of the tree

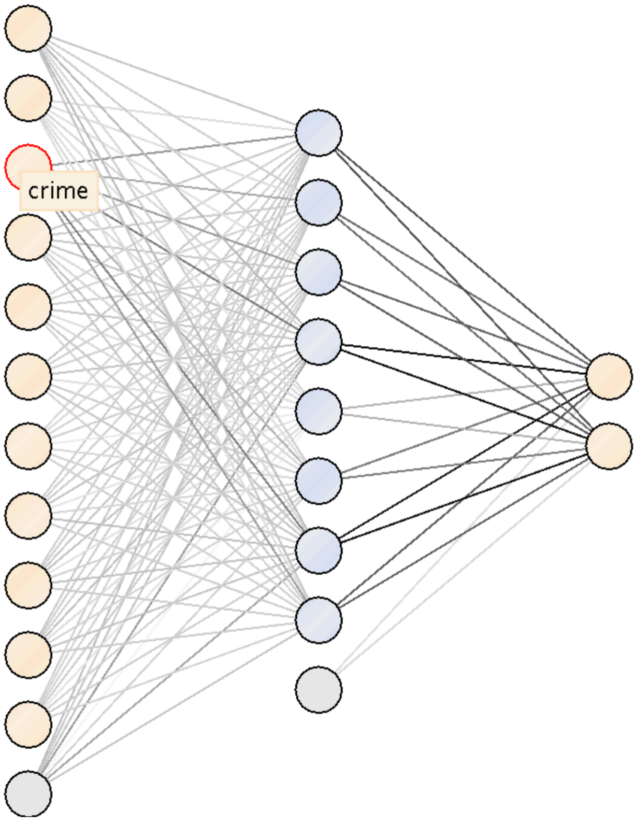
- **SVM**: if you are not using a kernel function (i.e. put kernel = dot), then the model returns attribute weights. Predictive attributes are those with weights furthest from 0 (i.e. Looking for high positive or high negative values). SVM has an output port for weights, or they can be read from the model itself.

Example of an SVM model, the higher weights are highlighted:

```
w[annan] = 0.022
w[clash] = 0.027
w[court] = -0.023
w[crime] = -0.028
w[eight] = 0.006
w[kenya] = 0.027
w[latest] = 0.001
w[kenya] = 0.027
w[kept] = -0.012
w[kibaki] = 0.018
w[latest] = 0.001
w[murder] = -0.025
w[nearbi] = 0.018
w[number] = -0.027
w[odinga] = 0.018
w[offend] = -0.012
w[partner] = -0.019
w[peopl] = 0.029
w[rift] = 0.027
```

```
w[right] = -0.006
w[vallei] = 0.027
w[visit] = 0.011
w[western] = 0.029
w[violent] = -0.021
```

- **Neural Network**: , take a look at the network itself, and click input neruons with the darkest connection lines. This is hard to see if you have too many attributes. The text description of the model also lists the final weights for each connection. You could also search this for high positive or high negative values.



The other classification algorithms do not give you feedback on which attributes were most predictive. However, the information can be got from their corresponding weighting algorithms.

2. Attribute weighting:

Different classification algorithms find different attributes useful. There is an attribute-weighting algorithm to correspond to most classification algorithms, which ranks attributes based on how predictive they are. Attributes are ranked by giving them a weight in the range [0,1]. A higher weight means a more predictive attribute.

Decision tree: Use the Weight by information gain operator to rank attributes based on which would be best at the root of the tree

k-NN: use weight by relief, and set neighborhood size to the same value as the 'k' you are using in k-NN

Naïve Bayes: Use Weight by uncertainty

SVM: Use weight by SVM. This gives the same information at the weights outputted from the algorithm itself.

There isn't a weighting algorithm for Neural Networks