Topic 2 - Classification

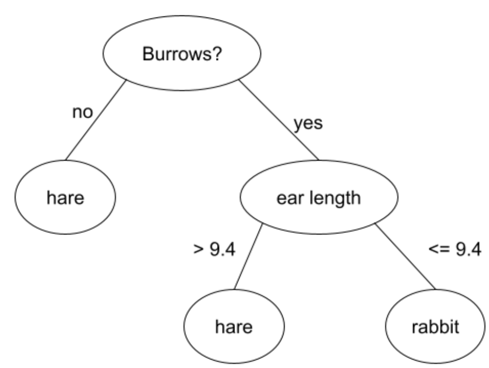
Classification is a type of supervised learning where the labels on a data are discrete or categorical

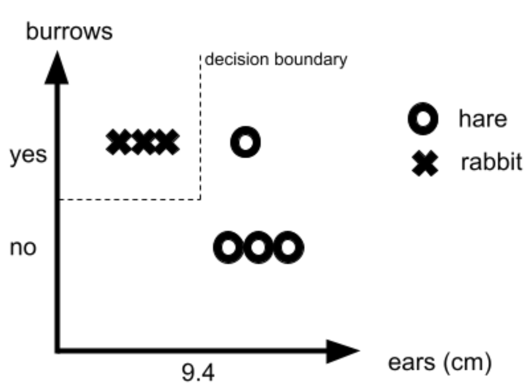
K-Nearest Neighbours Classification

- This works along the basic premise that things are similar if they're closer together
- Steps:
  - Measure **distance** from test image X to every image in training set X
  - Assign the label of the K=1 “nearest neighbor” to test data X
- This method is known as a **lazy learning algorithm**
- **Supervised** Machine Learning Algorithm
- K-NN classifies an object to a class by evaluating the nearest neighbors of a test sample.
- K-NN variants exist that weight the contribution of neighbors depending on their distance from the test sample to provide more accurate results.
- K-NN can be extended to regression by taking the mean value of the neighbors of a sample
- The 'K' in KNN represent the number of nearest neighbors to use when calculating the classification

Decision tree

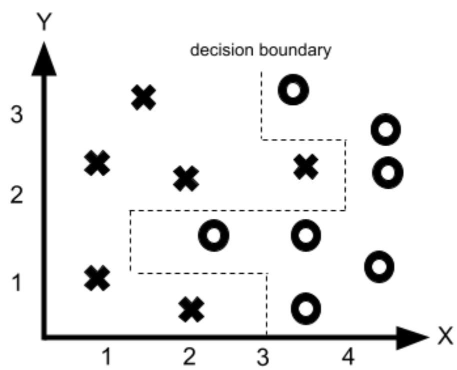
- They are capable of handling both **classification and regression** tasks and they're able to deal with **complex, non-linear datasets**
- Decision trees are known as "**White Box**"
- There are different types of decision tree algorithms.

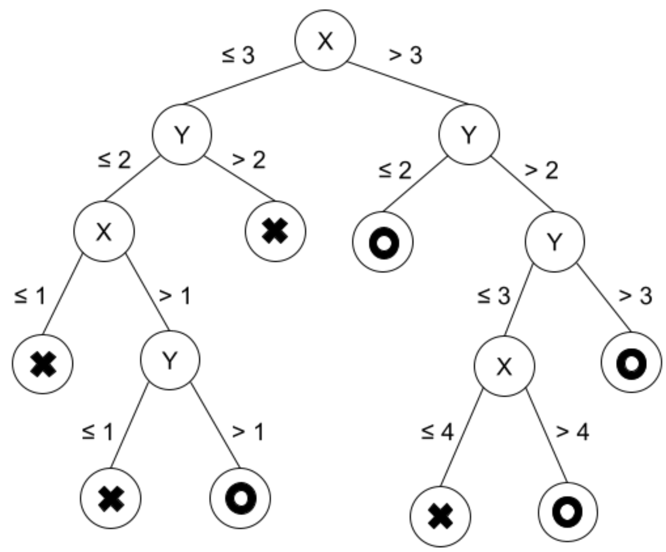




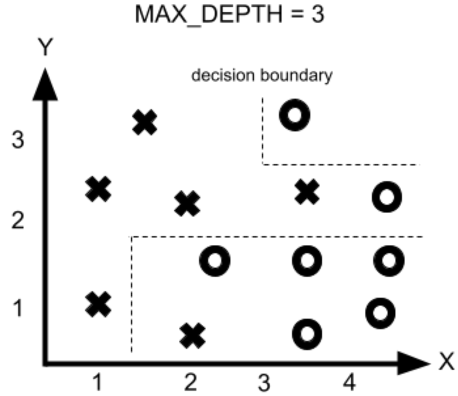
Classification and Regression Decision tree

- Also known as CART. This is a binary tree; that is each node, that's not a leaf, **spawns into two nodes**.
- Growing a tree and creating an ever more complex decision boundary is a major advantage of decision trees, in that it can be used to **capture complex relationships in the data**
- The danger of a decision tree is that they're **prone to overfitting**. Might not be able to generalize to new data that hasn't been seen before
- A simple solution to solve this is to force the algorithm to **restrict the depth of a tree**, forcing leaf nodes to be formed despite the fact that there may be misclassifications.
- The CART algorithm described here uses **greedy training**. That is, it computes only the optimal split at each node and sequence, but doesn't consider the optimization of the tree as a whole. It often provides a **good overall solution** despite **not being optimal**.
- The algorithm can **work on raw feature data** without the need for data pre-processing.

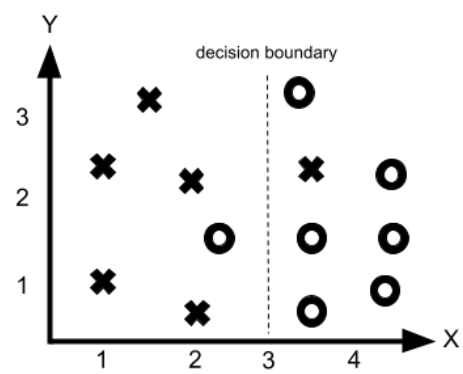




MAX_DEPTH = 3



MAX_DEPTH = 1



Classifier evaluation

- One way of evaluating the performance of an algorithm is to **measure the accuracy**. That is, the **percentage of correctly classified** examples from the total number of samples
- It's **not always a good measure** of the quality of a classifier.

Confusion Matrix

- A **confusion matrix** is a matrix where each cell contains a number that represents a population of labels that belong to a specific combination of predicted and actual classes.
- A confusion matrix presents a table layout of the different outcomes of **prediction** and **results** of a classification problem and helps visualize its outcomes
- The confusion matrix helps us **identify the correct prediction of a model** for different individual classes as well as **the errors**
- Predictions are **columns,** Real truths are **rows**

		Prediction		
		Class 0	Class 1	
Ground truth	Class 0	1	3	4 samples
	Class 1	0	6	6 samples

```
Ground Truth [1 0 0 1 0 0 1 1 1 1]
Predicted    [1 1 1 1 1 0 1 1 1 1]
```

Recall: Given a positive example, how likely will the classifier accept it?
Precision: Given a positive prediction, how likely is it to be correct?

		Prediction		
		Non-dog	dog	
Ground truth	Non-dog	TN	FP	(Negative class)
	dog	FN	TP	(Positive class)

$$Accuracy = TN + TP / (TN + TP + FN + FP)$$
$$Recall = TP / (TP + FN)$$
$$Precision = TP/(TP + FP)$$

- In **security applications** that use face recognition, it **may be safer** to fail to identify someone who should have access. That is, **get a false negative**, then get a false positive and wrongly allow access to a prohibited person
- In **health applications** is **more costly** to miss a diagnosis to **get a false negative** than to make an incorrect diagnosis with low probability of false positive

Machine Learning Tools for Classification:

● Naïve Bayes	[Small data]
● Logistic Regression	[Small data]
● Decision Tree	[Medium data]
● Random Forest	[Large data]