

Sensitivity focuses on the positive:

* How many spams are the system notified correctly?

►Accuracy: Overall, how often is the prediction correct?

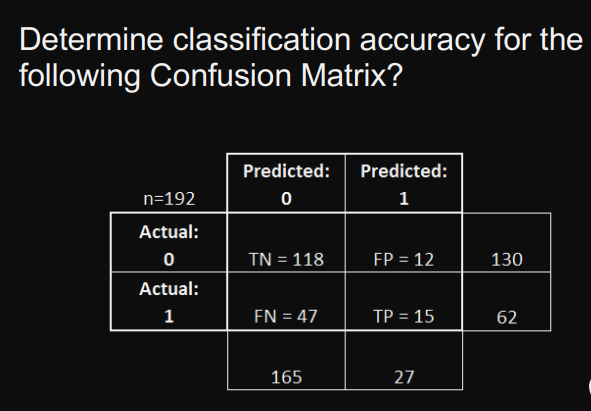
►Sensitivity (Recall): When the actual value is positive, how often is the prediction correct?

►Specificity: When the actual value is negative, how often is the prediction correct?

►False Positive Rate: When the actual value is negative, how often is the prediction incorrect?

►Precision: When a positive value is predicted, how often is the prediction correct

FLUX:

0.69

## Which Metrics Should be Used?

Spam filter: Optimise precision or specificity

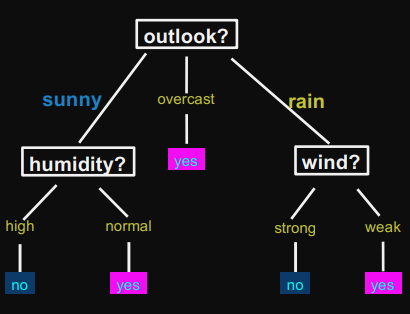
* False negatives (spam goes to the inbox) are more acceptable than false positives (non-spam is caught by the spam filter)

Fraudulent transaction detector: Optimise sensitivity

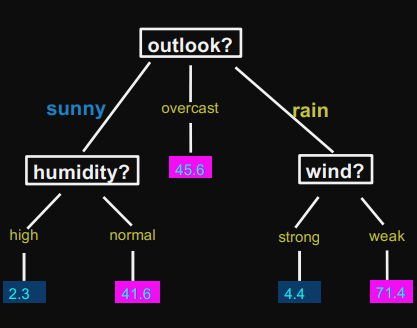
* False positives (normal transactions that are flagged as possible fraud) are more acceptable than false negatives (fraudulent transactions that are not detected)

## Decision Trees and Regression Trees

Decision Trees

* Predict binary (or categorical) outcomes
* Prediction is the most common values
* 

Regression Trees

* Predict continuous (i.e. real) values
* Prediction is usually the average value
* 

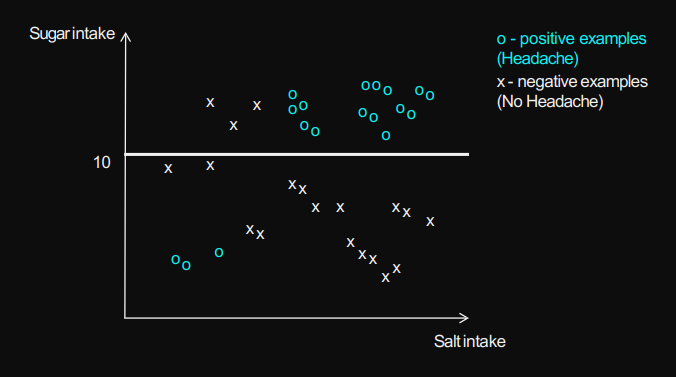
## Build Regression and Decision Trees

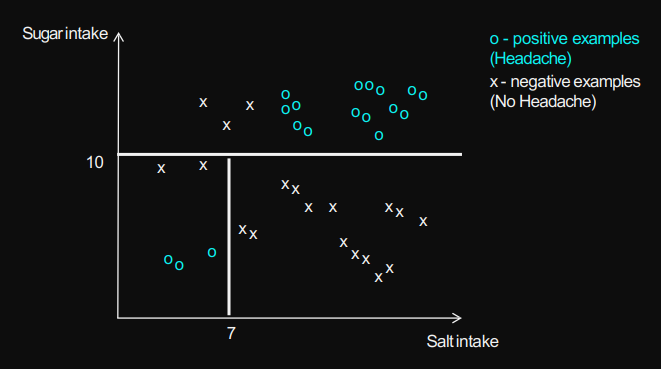
► Recursively partition (divide up) the feature space into regions

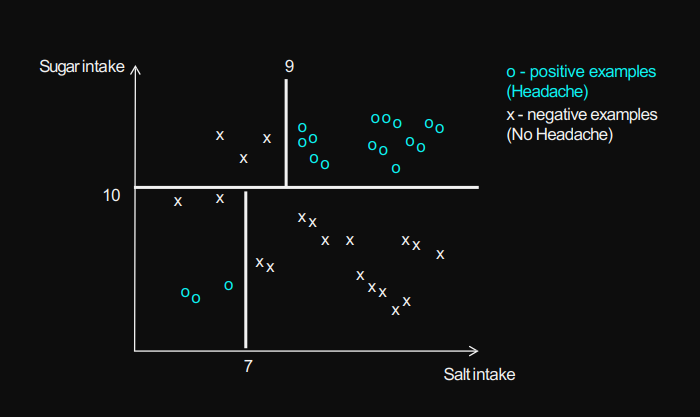
► While grouping similar instances together

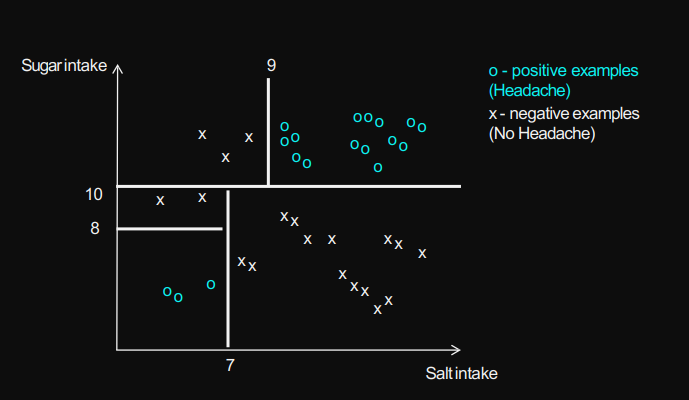
## Recursive Partitioning

At each iteration, we divide the data to group similar instances together

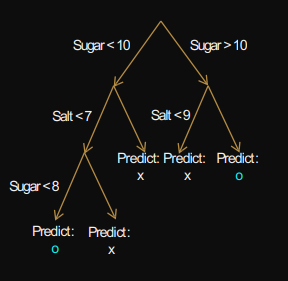








## Prediction Model is a Tree

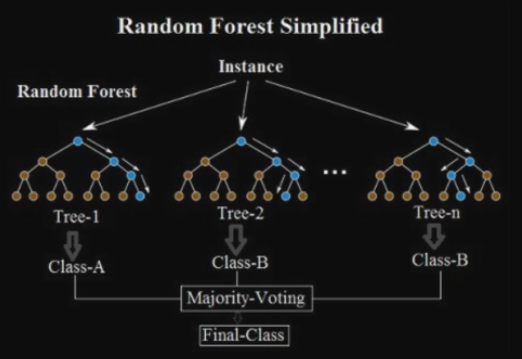


Algorithms (e.g., Entropy) for building trees differ in the criteria used to:

* Decide on which feature to split on in each iteration
* Decide when to stop

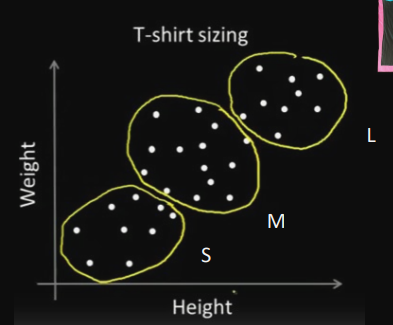
## Random Forest

constructing a number of decision trees

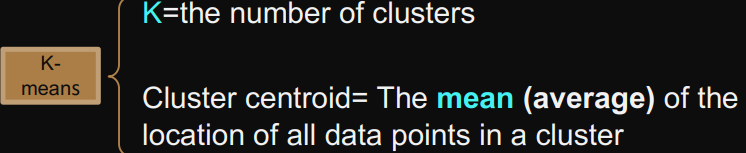


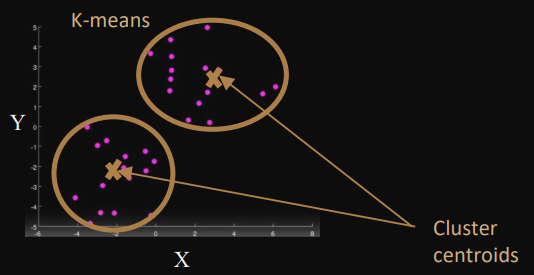
## Clustering

Grouping a set of data points into different subgroups based on their similarity



### K-means Clustering





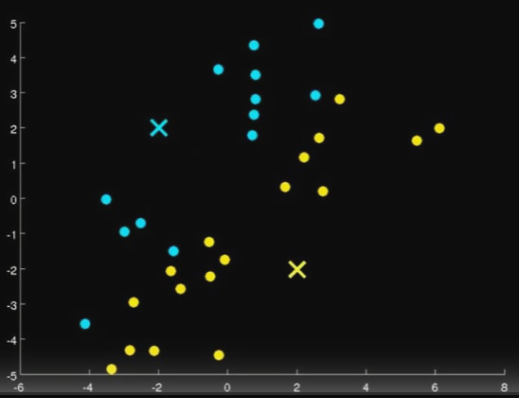
#### Initial Step

Randomly initialize two points

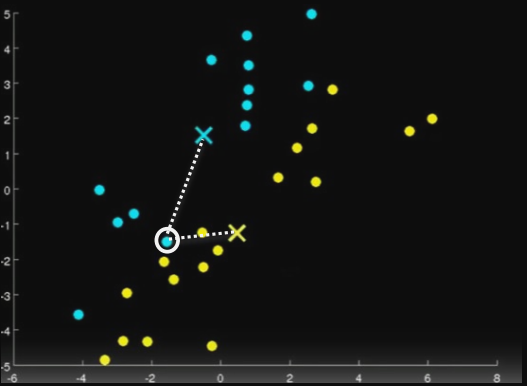


#### Two Main Steps- Iterate until there is no changes

1. Cluster assignment



2. Move centroid



#### K-means Algorithm

Input: - A set of data points - The number of clusters (K)

Method:

- Select K initial random points

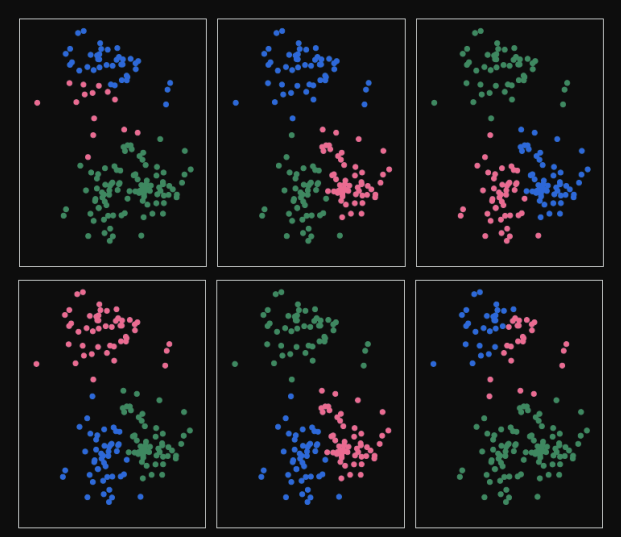
- Repeat

– Cluster assignment

– Move the cluster centroids to the mean value of data points in the cluster

- Until no change

#### Impact of Random Initial Points



## Decision Tree Implementation

