

# K-means Clustering



@cambridgespark



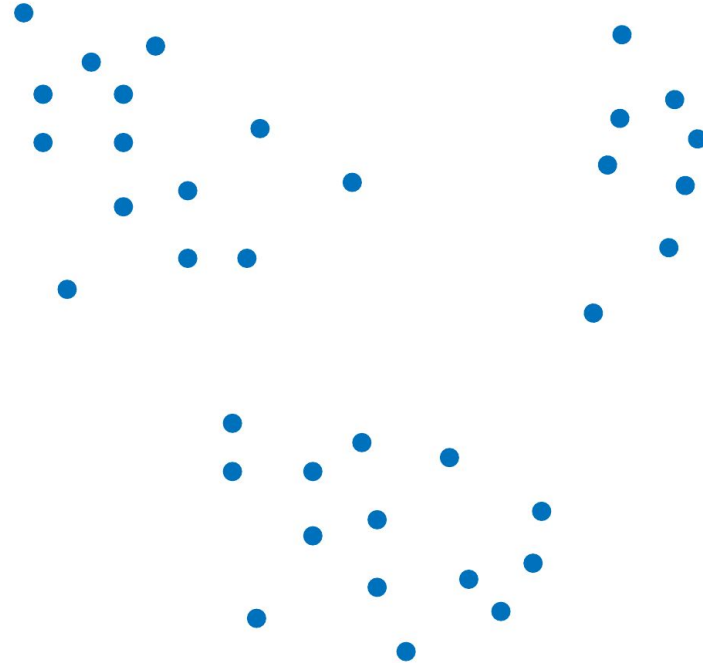
CAMBRIDGE SPARK

# Brief Overview

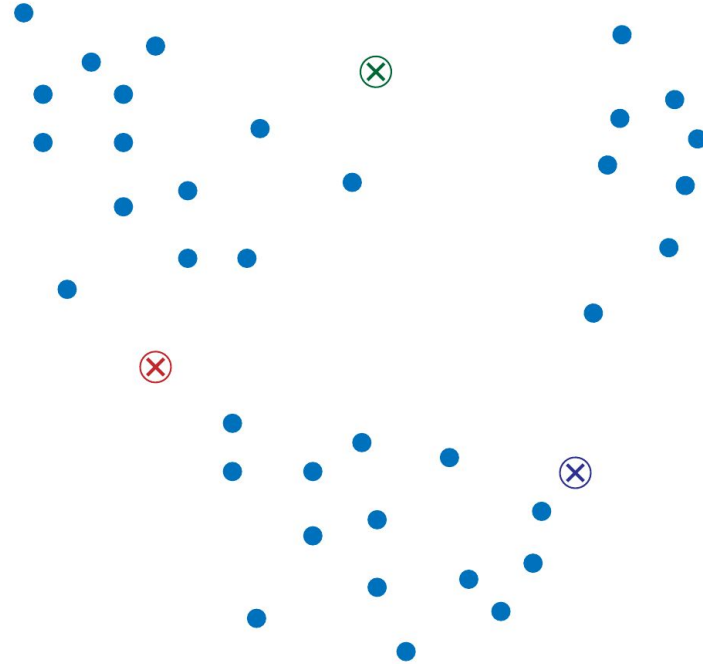
---

1. K-Means partition  $n$  data points into  $k$  clusters
2. Each data point belongs to the cluster with the nearest mean
3. The algorithm produces exactly  $k$  different clusters of greatest possible distinction
4. But the process is sensitive to our choice of  $k$  and the initialisation

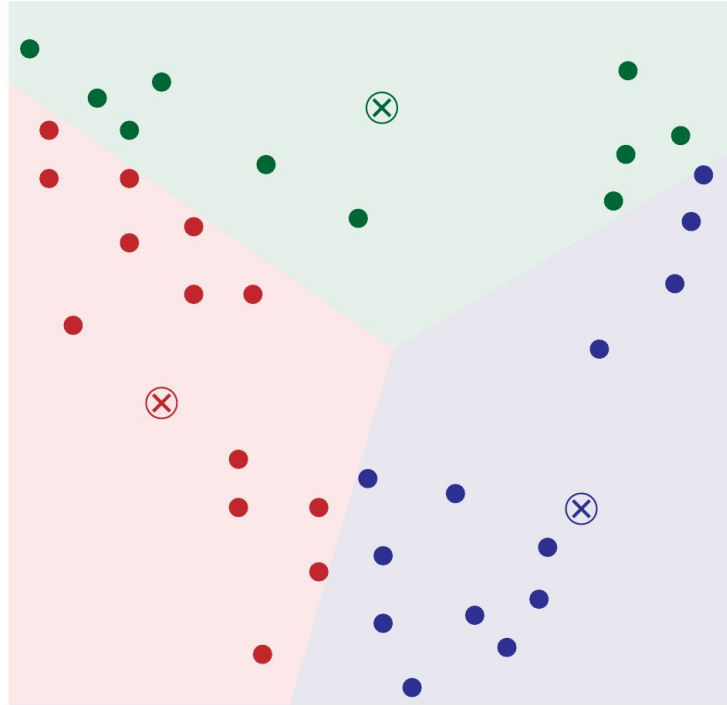
# K-means



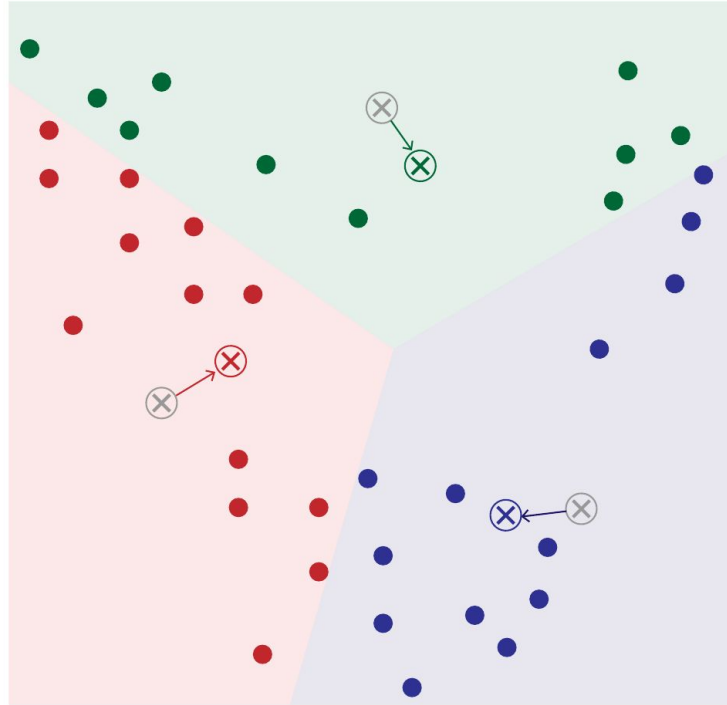
# K-means



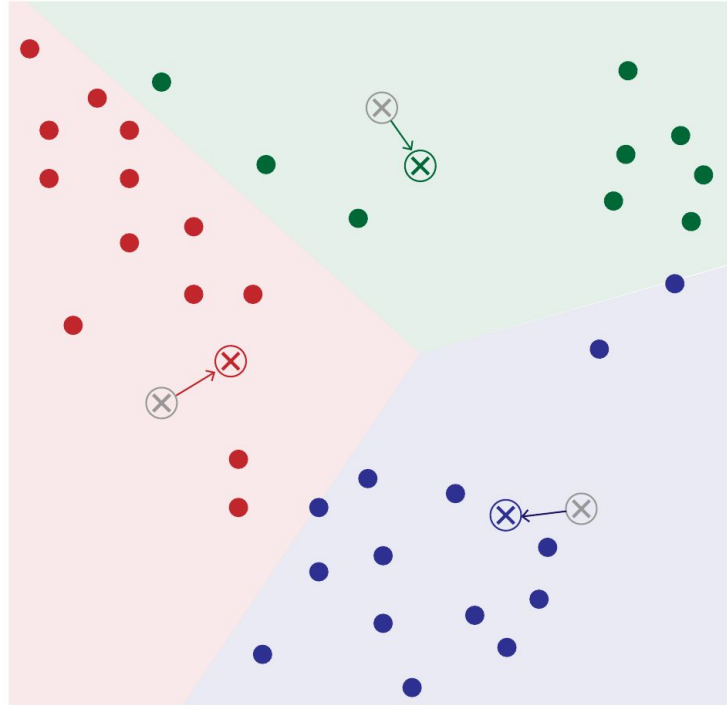
# K-means



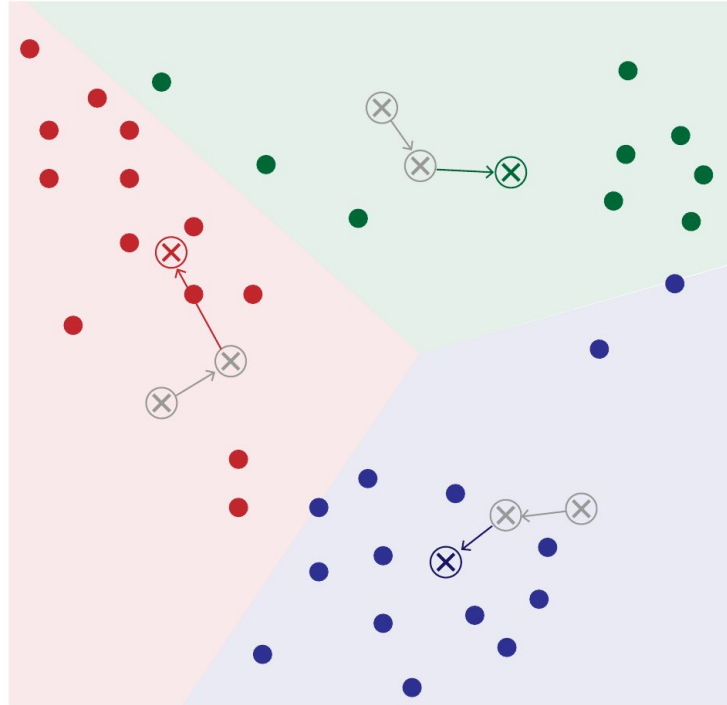
# K-means



# K-means

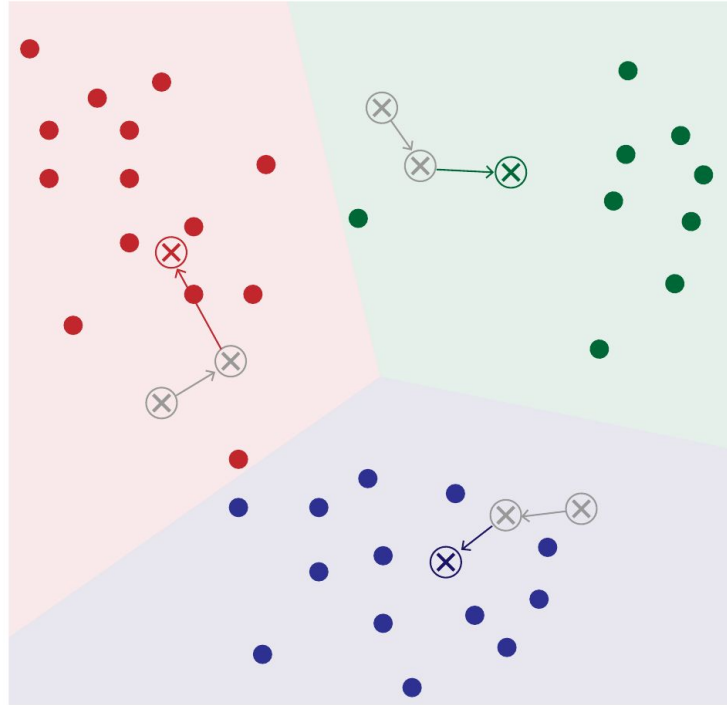


# K-means

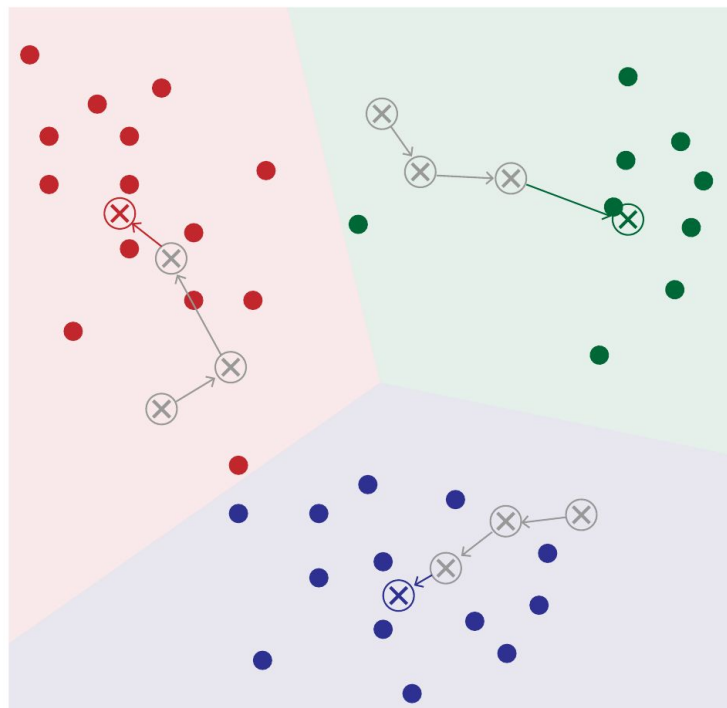




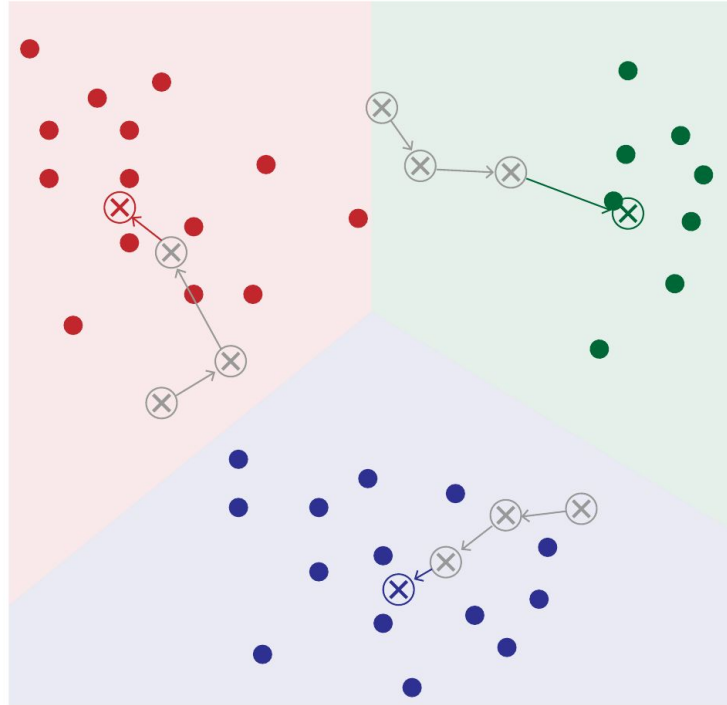
# K-means



# K-means



# K-means



# K-means - Summary

---

- Start with  $k$  “means” drawn at random
- Assign data points to the nearest  $k$
- Update the position of each  $k$  to correspond to the mean of its assigned points
- Rinse and repeat...

# K-means - Pros and cons

---

## Pros

- Cheap to compute
- Easy to interpret
- Efficient implementations available
- Assigning a new point is straightforward

## Cons

- Need to guess  $k$
- Clusters are globular
- Sensitive to initialisation
- Sensitive to noise



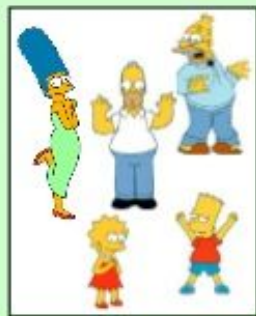
Hands-on session

*kmeans.ipynb*

What is a natural grouping among these objects?



Clustering is subjective



Simpson's Family



School Employees



Females



Males



CAMBRIDGE SPARK