HyperionDev

# Lecture - Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
  - Please review Code of Conduct (in Student Undertaking Agreement) if unsure
- No question is daft or silly - **ask them!**
- There are Q&A sessions midway and at the end of the session, should you wish to ask any follow-up questions.
- Should you have any questions after the lecture, please schedule a mentor session.
- For all non-academic questions, please submit a query: www.hyperiondev.com/support

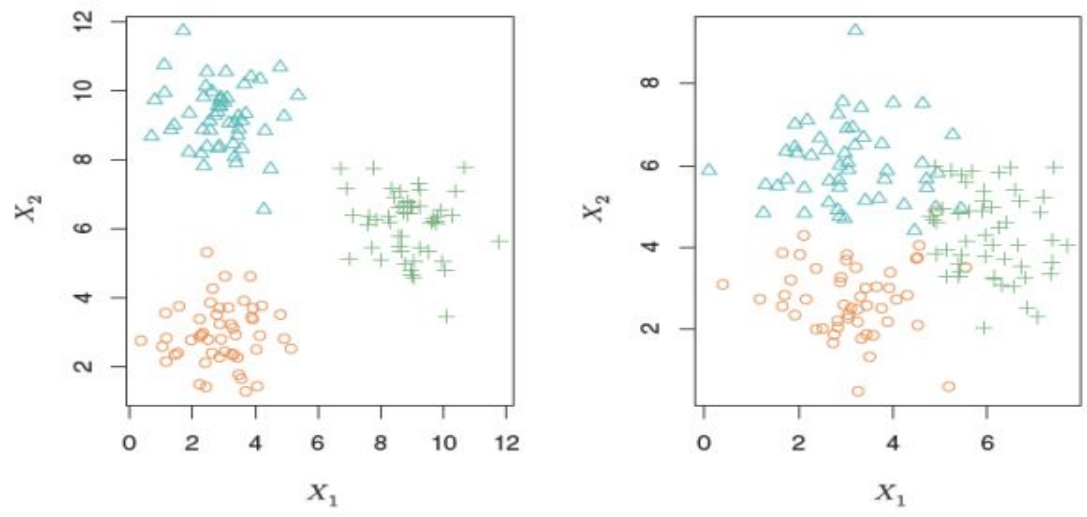
# Lecture Objectives

1. Introduction to Unsupervised Learning
2. Introduction to Clustering
3. Walk through the K-means clustering algorithm

# Introduction to Clustering

- Previously, when supervised learning was covered, it involved datasets with both input and output variables.
- However, we will take a look at another class of problems, unsupervised learning problems, where only the input variables are observed.
- Here we will look at the unsupervised method : clustering.

# Introduction to Clustering




The graphs above show two datasets that are good candidates for applying clustering. The data on the left shows a clear grouping that a clustering algorithm could readily identify for us. The right hand side data groups with more overlap and will be harder to identify, but still suitable for a clustering approach, rather than using linear regression for example.

# Introduction to Clustering

If a clustering approach seems suitable, we can use the cluster analysis to assert that observations within a group are similar to each other, while observations in different groups are different from each other.

# K-Means Clustering

K-Means clustering is the most well-known clustering algorithm. It is a *simple* and elegant approach for partitioning a dataset into K distinct clusters. To perform K-Means clustering, we first specify the desired number of clusters, K, and then assign each observation to exactly one of the K clusters.

# Feature Space

- There are a number of different distance metrics that are used in algorithms to decide how similar observations are.
- The most common one is the Euclidean distance.

$(x_i, y_i)$ and $(x_j, y_j)$ is $\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$.

- To compute the mean of a number of observations, we divide the sum of the observations by the number of observations
- To compute the mean of a certain number of (x,y) points, we compute the mean of all x values and the mean of all y values

# The K-means Algorithm

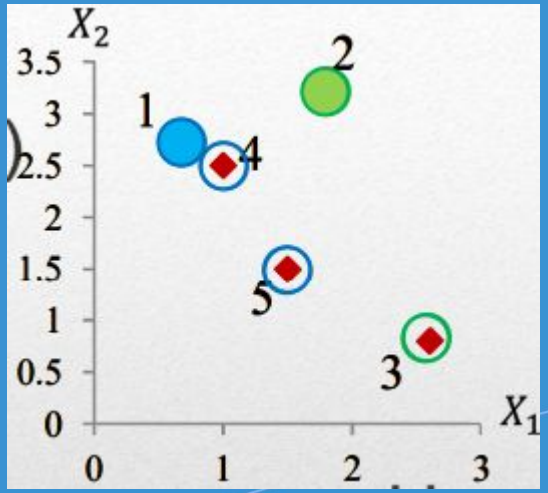
- The K-means algorithm follows the following steps :
  - Select number of clusters , K
  - Select random points from the data as starting values and initialise the mean of each cluster.
  - For *n* number of iterations :
    - Assign each point to the cluster whose mean (or “centroid”) is the nearest.
    - Re-compute the means for each cluster based on its current members.
  - Repeat steps until convergence.

# K-means Algorithm

K = 2

$$X = \begin{bmatrix} 0.7 & 2.7 \\ 1.8 & 3.2 \\ 2.6 & 0.8 \\ 1.0 & 2.5 \\ 1.5 & 1.5 \end{bmatrix}$$

| | Obj1 | Obj2 |
|---|---|---|
| Obj1 | | |
| Obj2 | 1.21 | |
| Obj3 | 2.69 | 2.53 |
| Obj4 | 0.36 | 1.06 |
| Obj5 | 1.44 | 1.73 |

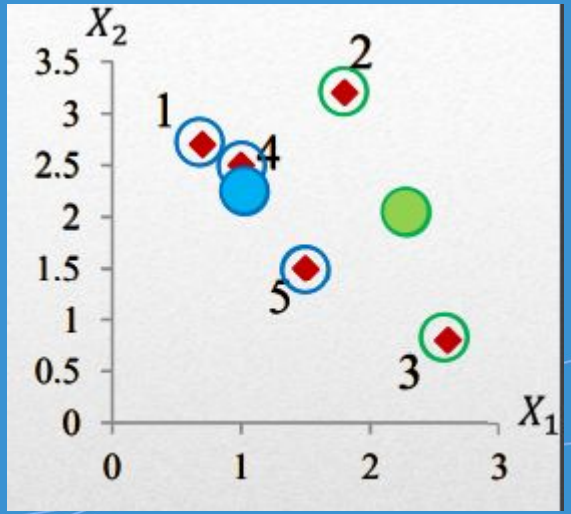


# K-means Algorithm

Calculate the new centroids

$$X = \begin{bmatrix} 0.7 & 2.7 \\ 1.0 & 2.5 \\ 1.5 & 1.5 \end{bmatrix} \text{ and } X = \begin{bmatrix} 1.8 & 3.2 \\ 2.6 & 0.8 \end{bmatrix}$$

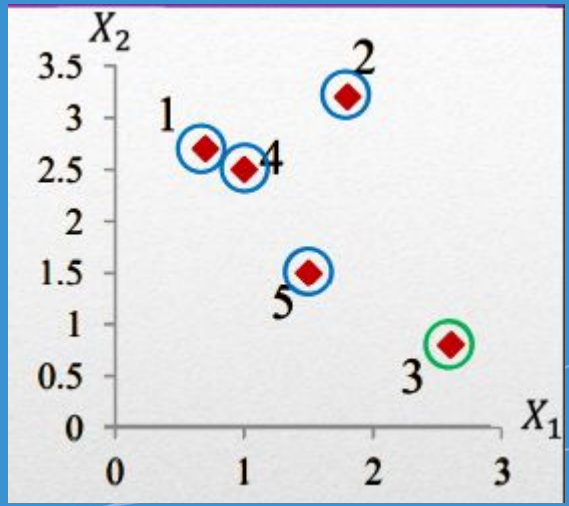
[1.07 2.23] and [2.20 2.00]



# K-means Algorithm

Find nearest centroid for each observation

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Blue | 0.60 | 1.21 | 2.09 | 0.28 | 0.85 |
| Green | 1.66 | 1.26 | 1.26 | 1.30 | 0.86 |

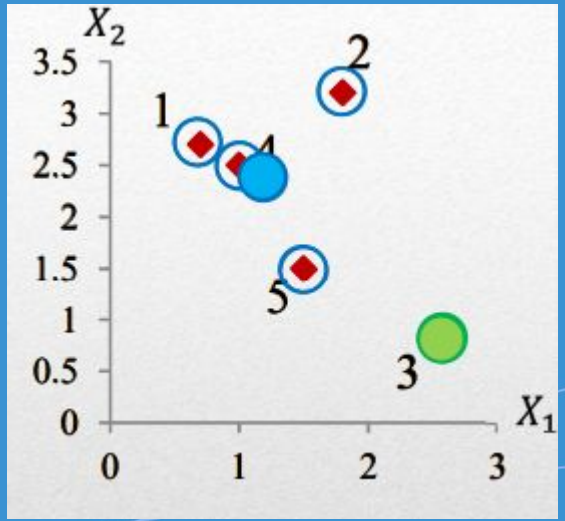


# K-means Algorithm

Calculate the new centroids

$$X = \begin{bmatrix} 0.7 & 2.7 \\ 1.8 & 3.2 \\ 1.0 & 2.5 \\ 1.5 & 1.5 \end{bmatrix} \text{ and } X = [2.6 \quad 0.8]$$

$[1.25 \quad 2.48]$ and $[2.6 \quad 0.8]$

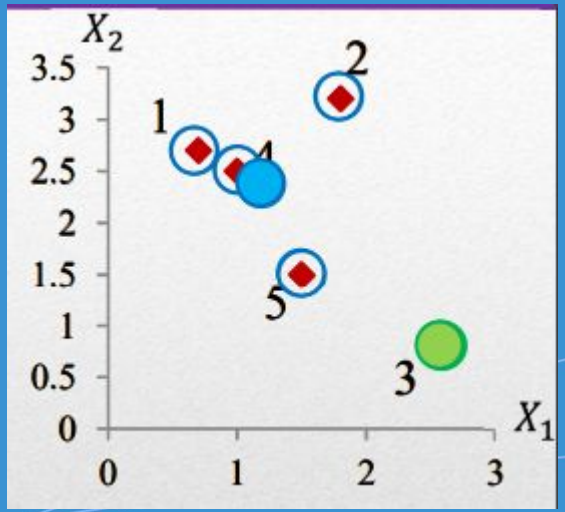


# K-means Algorithm

Find distances to nearest centroid

Unchanged => converged

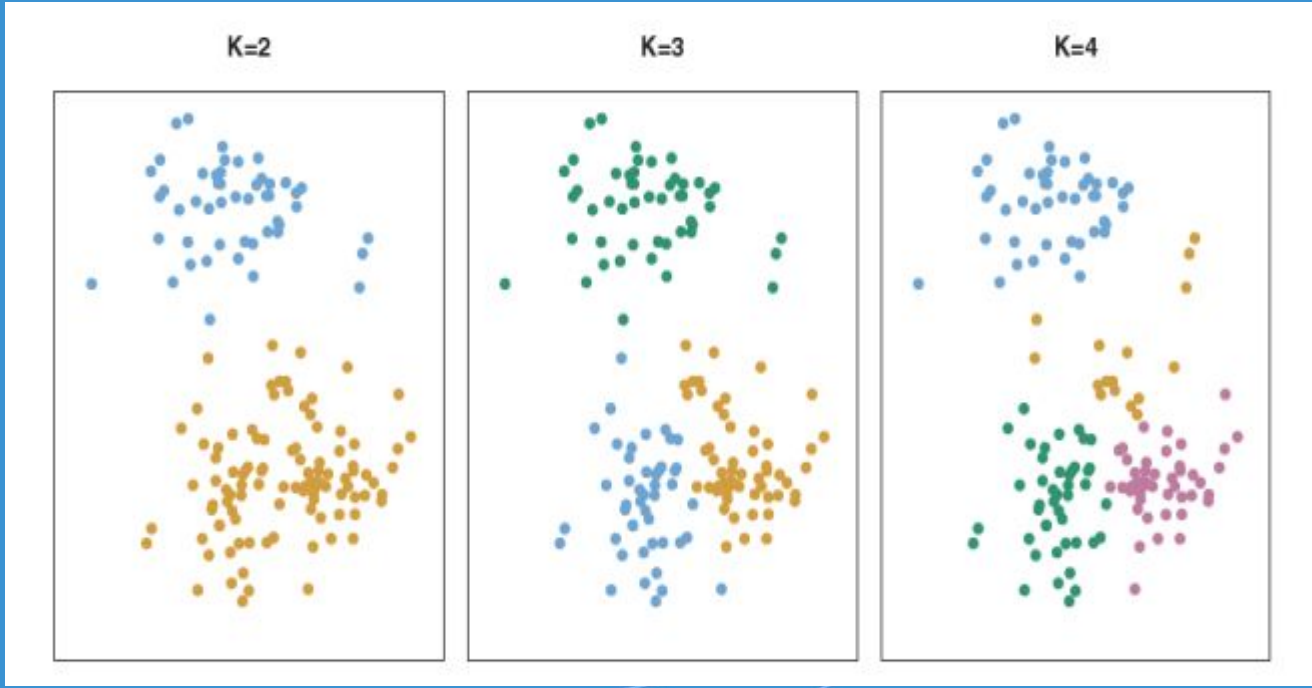
| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Blue | 0.59 | 0.91 | 2.15 | 0.25 | 1.01 |
| Green | 2.69 | 2.53 | 0.00 | 2.33 | 1.30 |



# Choosing K

- An important thing to consider when applying K-means is that we need to choose a value for K before running the analysis.
- Choosing K will greatly affect the outcome and accuracy of the clusters.
- The following plots displays different outcomes of the algorithm depending on the value chosen for K.

# Choosing K



# Validating the clusters

- It’s possible to find clusters in any data, but it is important to determine if these clusters actually represent underlying subgroups in the data or are merely grouping with similar noise.

- This is a very hard question to answer. There exist a number of techniques for assigning a significance value to a cluster in order to assess whether there is more evidence for the cluster than one would expect due to chance. However, there has been no consensus on a single best approach. The Silhouette Coefficient (sklearn.metrics.silhouette_score) is an example of an evaluation metric which indicates how similar samples within a cluster are, compared to other clusters. A higher Silhouette Coefficient score relates to a model with better-defined clusters

# Questions and Answers

## Questions around K-means Clustering