K-Means Clustering

K-Means Clustering

- Section Overview
  - Understanding Clustering
  - Intuition of K-Means
  - Mathematical Theory of K-Means
  - Example of K-Means
  - K-Means Project Exercise
  - K-Means Project Exercise Solution

K-Means Clustering

- ***Important Note:***
    - *Do not confuse K-Means with KNN!*
    - *The "K" is completely different in both algorithms, they solve completely different problems and are not related in any way!*

Clustering

General Concepts

Clustering

- Clustering uses **unlabeled data** and looks for similarities between groups (clusters) in order to attempt to segment the data into separate clusters.
- Keep in mind that we don’t actually know the true correct label for this data!

Clustering

- Imagine an example data set:

X1	X2
2	4
6	3
…	…
1	2

Clustering

- Notice again we only have features!

X1	X2
2	4
6	3
...	...
1	2

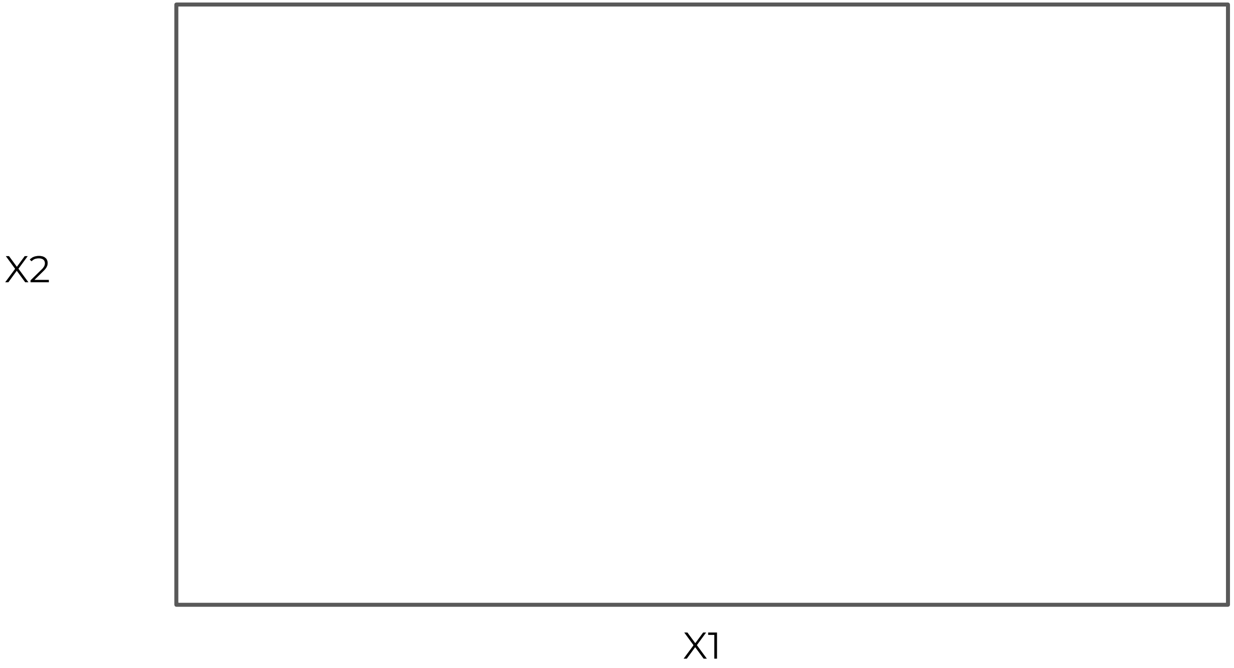
Clustering

- How could we cluster this data together?

X1	X2
2	4
6	3
...	...
1	2

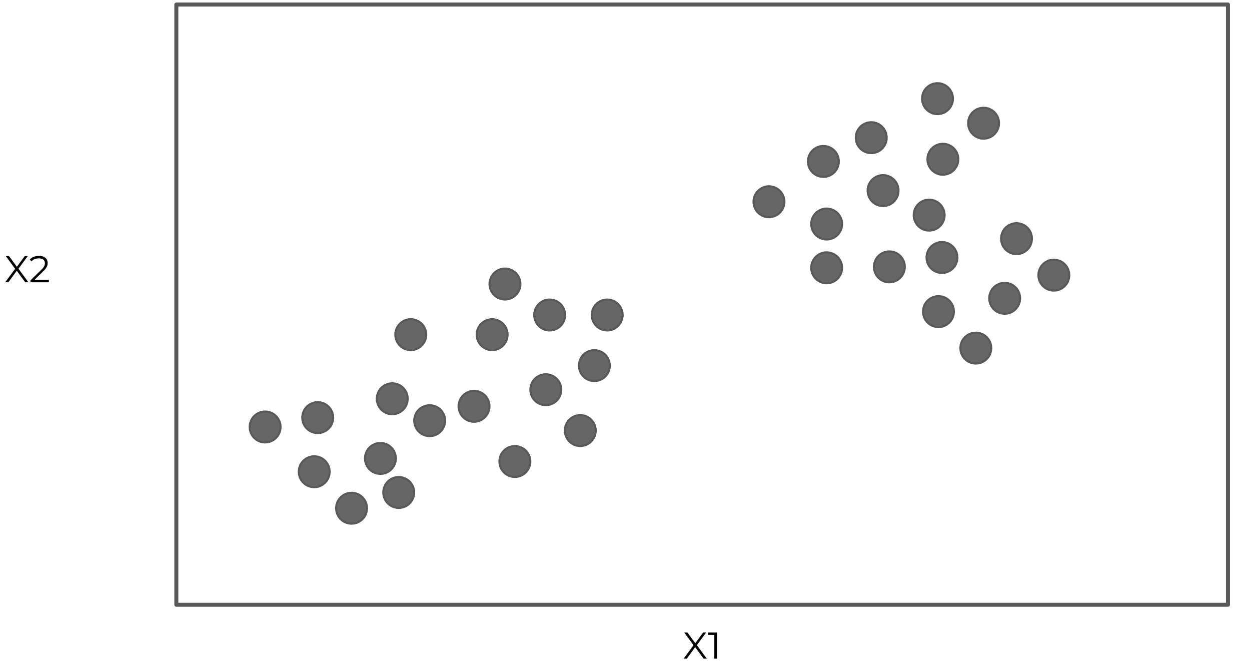
Clustering

- Could simply plot and discover patterns:



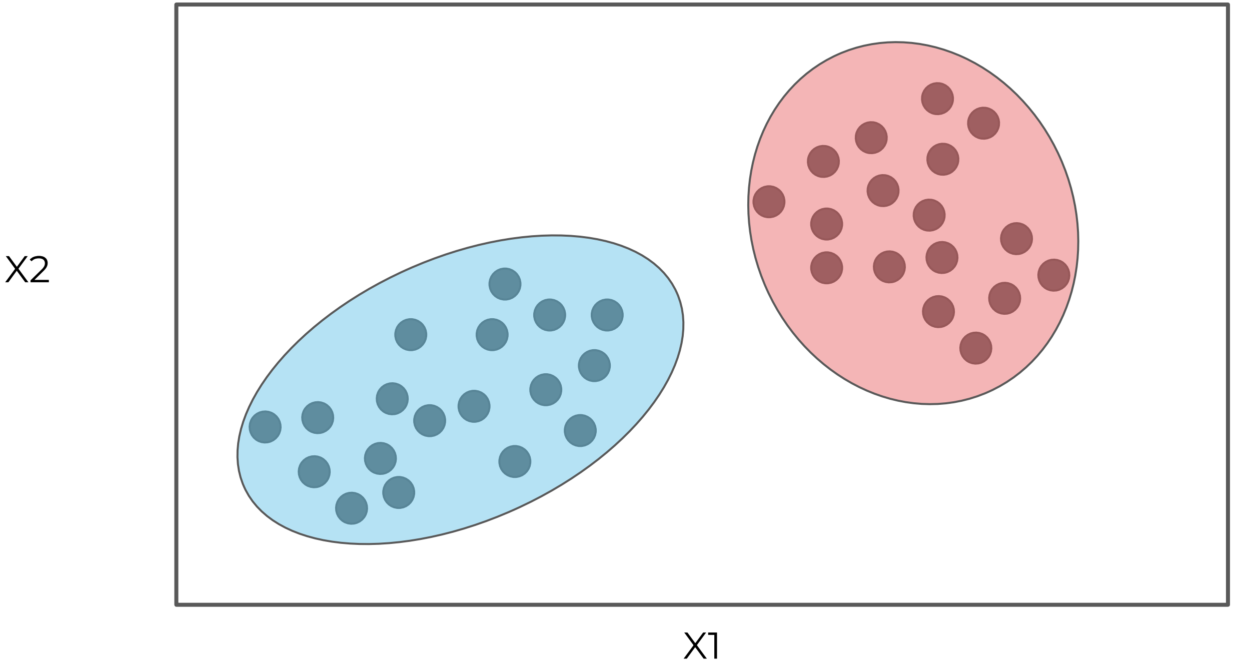
Clustering

- Here we intuitively see 2 groupings:



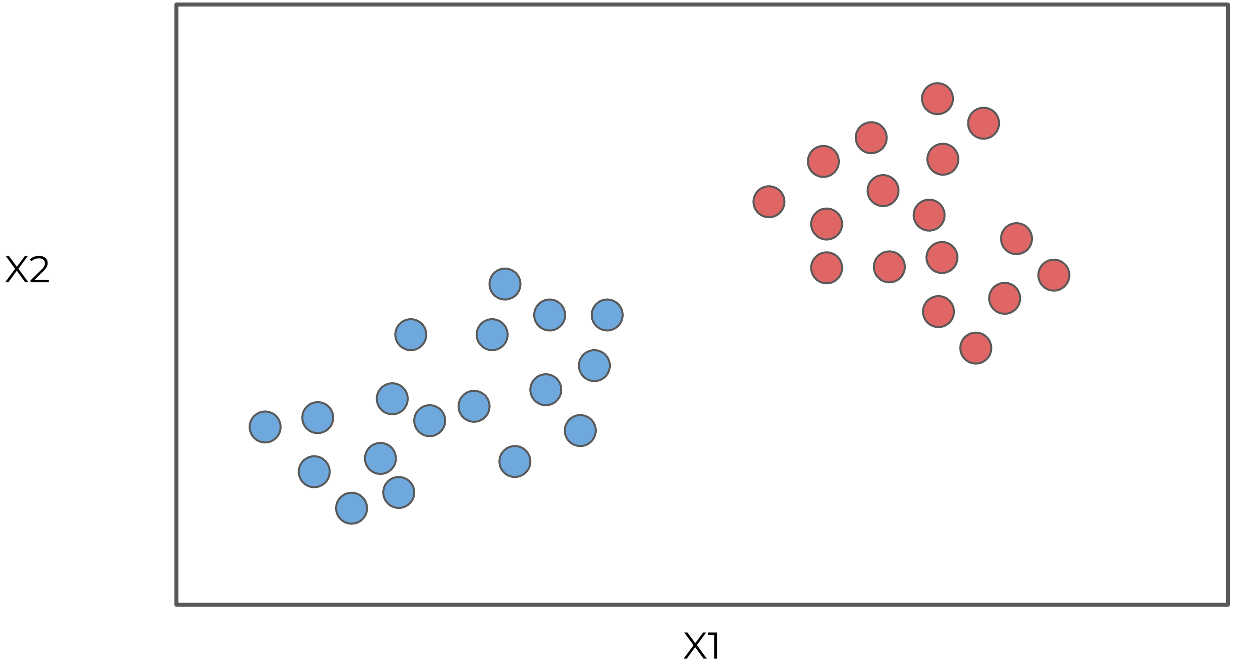
Clustering

- Note how distance is the intuitive metric:



Clustering

- We could then assign clusters:

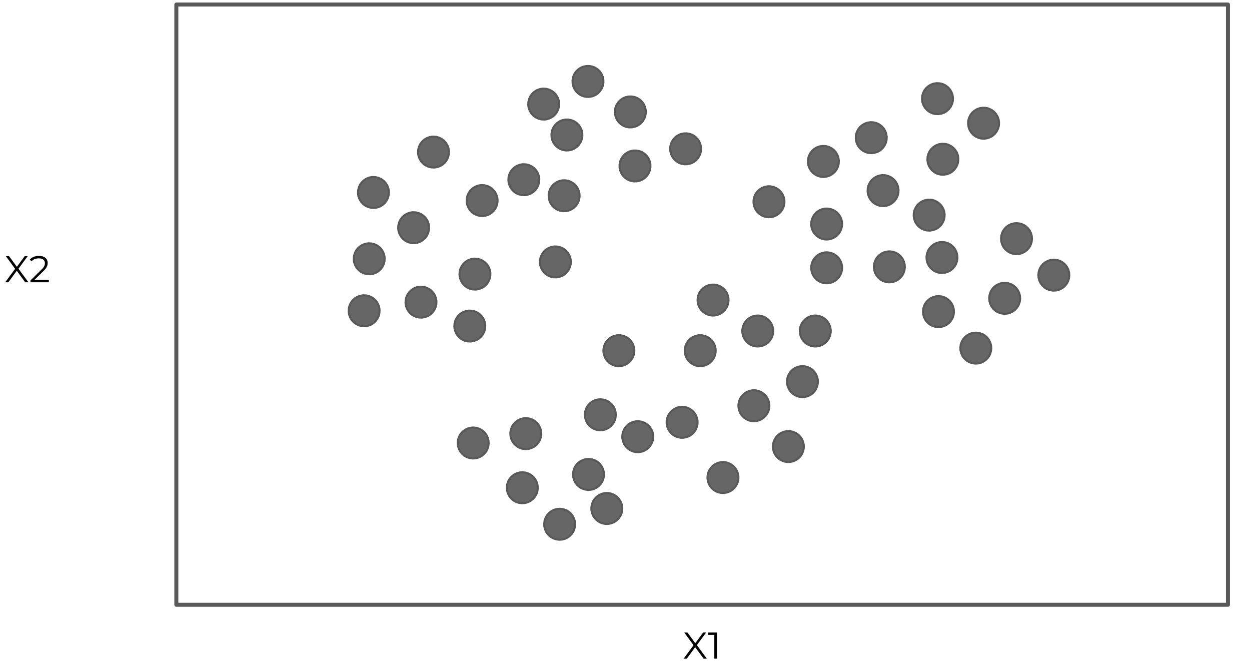


Clustering

- Notice how we don’t actually know for sure if this is a correct way of grouping together these data points, there was no correct label to begin with!
- And what about situations that are not so obvious or multi-dimensional?

Clustering

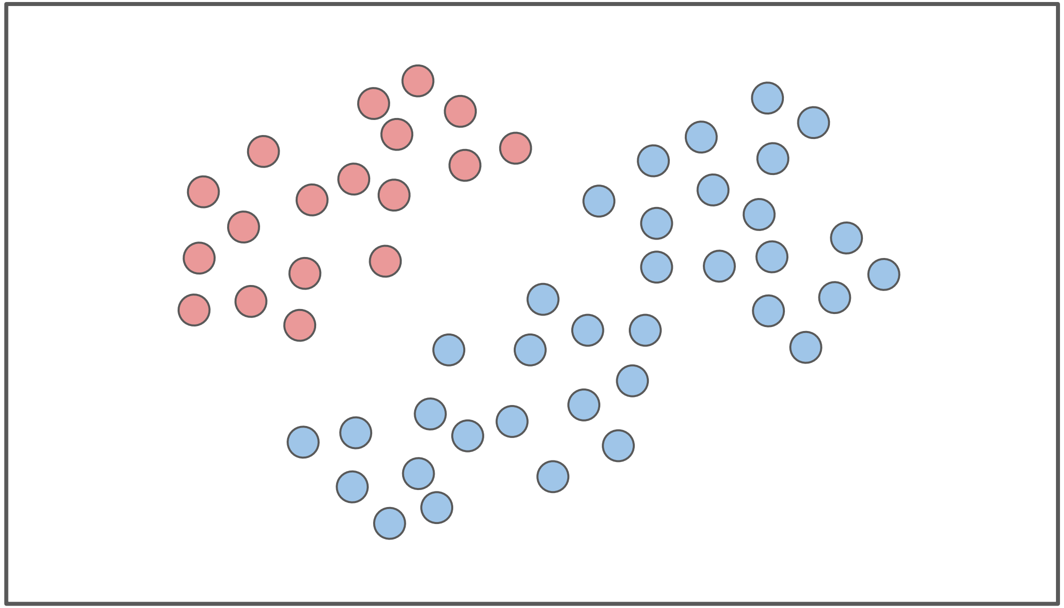
- 2 or 3 clusters could both be reasonable:



Clustering

- 2 or 3 clusters could both be reasonable:

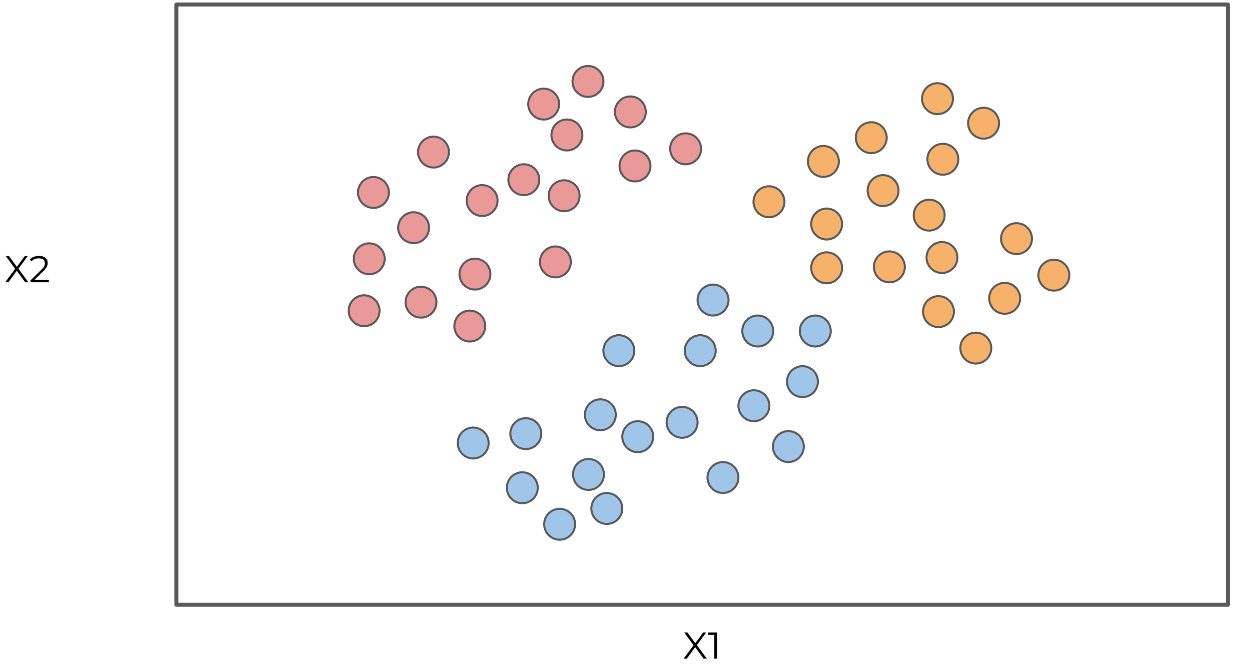
X2



X1

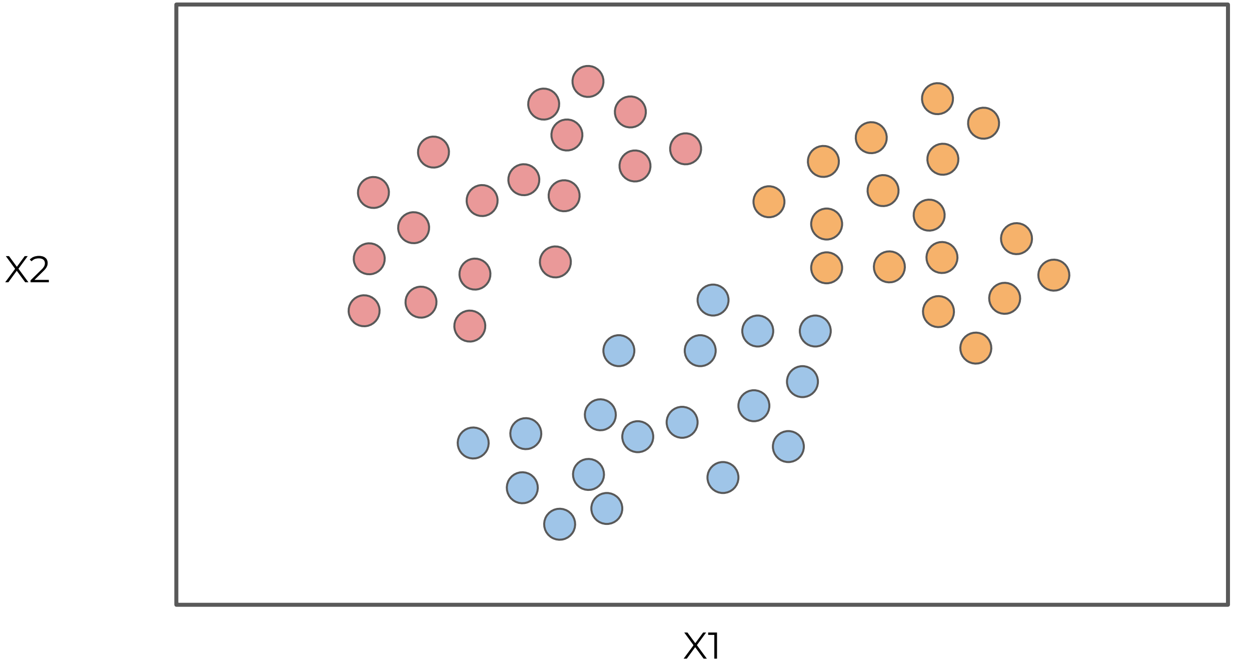
Clustering

- 2 or 3 clusters could both be reasonable:



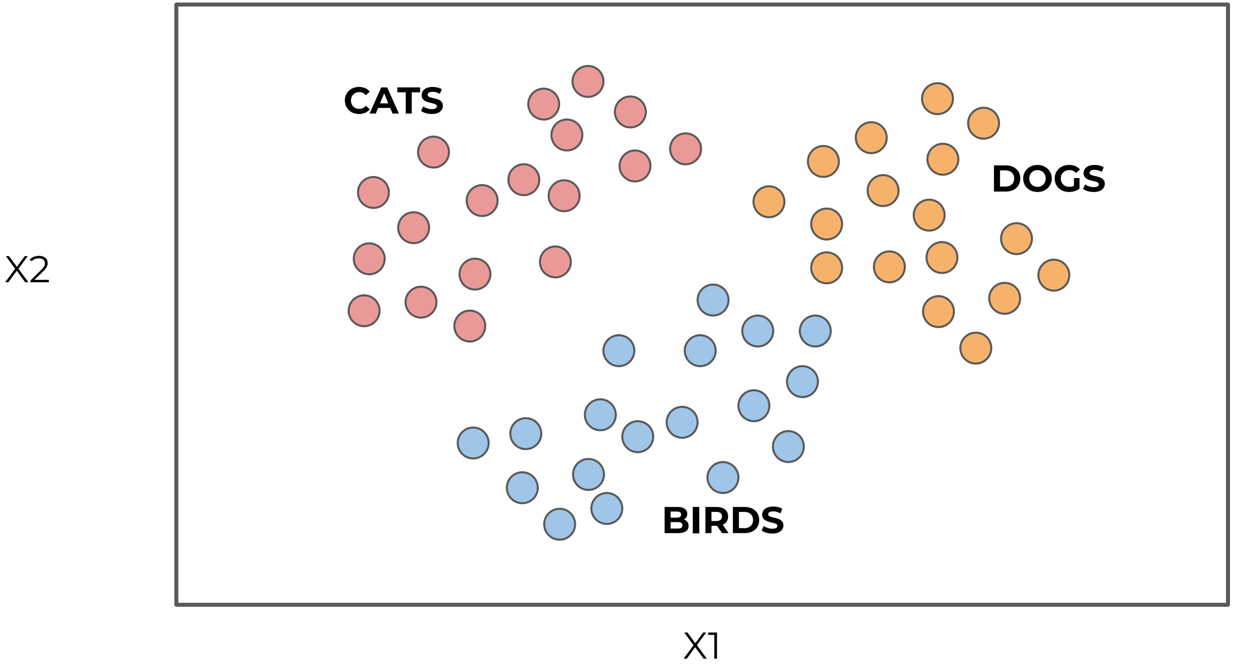
Clustering

- Different methods can be used to decide!



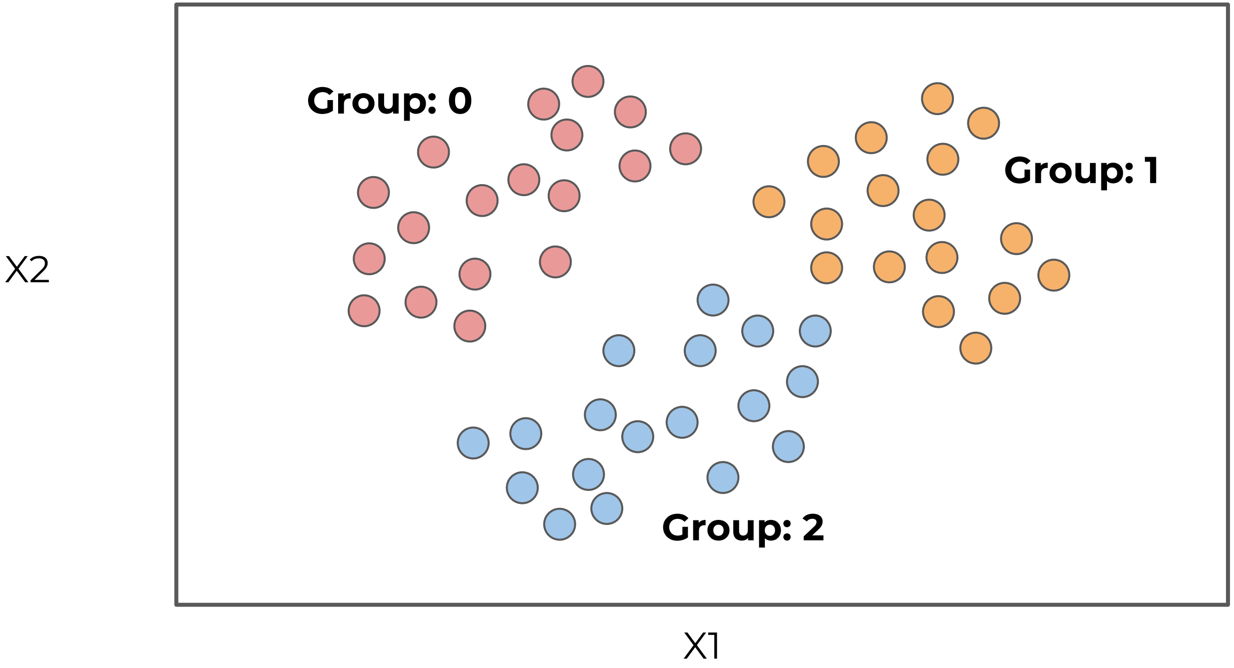
Clustering

- Clustering doesn’t “label” these for you!



Clustering

- Clustering maps data points to groups



Clustering

- Main Clustering Ideas:
  - Use features to decide which points are most similar to other points.
  - Realize that there is no final correct **y** label to compare cluster results to.
  - We can think of clustering as an unsupervised learning process that “discovers” potential labels.

Clustering

- Unsupervised Learning Paradigm Shift:
    - *What about a new unlabeled data point?*
    - *How do we assign it to a cluster?*
    - *Was it the correct cluster for assignment?*

Clustering

- Unsupervised Learning Paradigm Shift:
  - *How do we assign a new data point to a cluster?*
    - Different approaches depending on the unsupervised learning algorithm used.
    - Use features to assign most appropriate cluster.

Clustering

- Unsupervised Learning Paradigm Shift:
  - *How do we assign a new data point to a cluster?*
    - Just as before, no way to measure if this was the “correct” assignment.

Clustering

- Unsupervised Learning Paradigm Shift:
    - *If we've discovered these new cluster labels, could we use that as a **y** for supervised training?*
        - Yes! We can use unsupervised learning to discover possible labels, then apply supervised learning on new data points.

Clustering

- Unsupervised Learning Paradigm Shift:
  - *If we've discovered these new cluster labels, could we use that as a **y** for supervised training?*
    - What's the trade-off?

Clustering

- Unsupervised Learning Paradigm Shift:
    - *If we've discovered these new cluster labels, could we use that as a* ***y*** *for supervised training?*
        - Clustering doesn't tell you what these new cluster labels represent, no real way of knowing if these clusters are truly significant.

Clustering

- Clustering ideas still to come:
  - How do we decide which number of clusters is best?
  - Do we decide or let the algorithm decide?
  - How can we measure "goodness of fit" for clustering without a **y** label for comparison?

Clustering

- Machine Learning as an art:
  - *What is ground truth?*
  - *What trade-offs are we making by using unsupervised learning as a substitute for ground truth of the y label that was not given?*

Clustering

- Keep these ideas in mind as we discover the different algorithms behind clustering!
- Also keep in mind that it’s much harder to compare unsupervised algorithms against each other due to the lack of ground truth based performance metrics (e.g. can’t use accuracy or RMSE).

K-Means Clustering

Intuition and Theory

Clustering

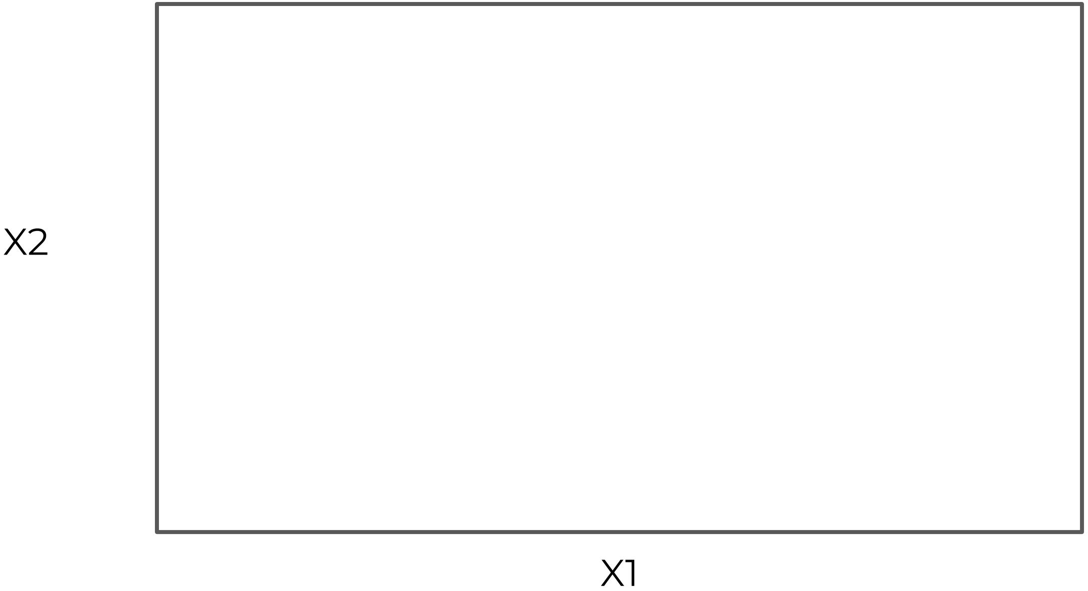
- So how does K-Means clustering actually work?
    - The main concept is actually very simple!
    - Let's walk through an example of clustering unlabeled data.

Clustering

- First a set of properties each point must satisfy:
    - Each point must belong to a cluster.
    - Each point can only belong to one cluster (no single point can belong to multiple clusters).

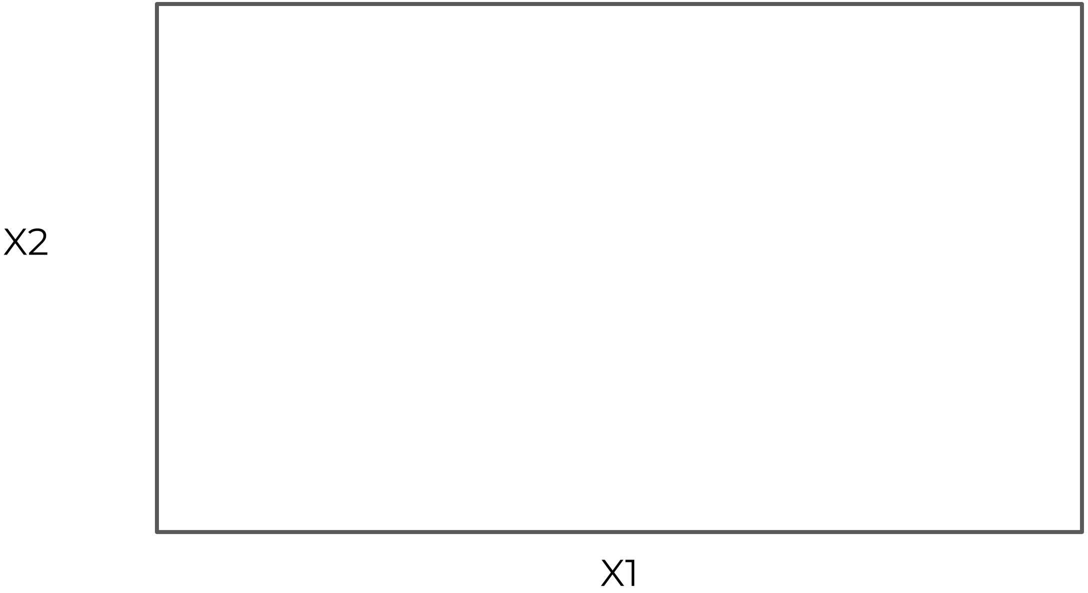
Clustering

- So how does K-Means clustering actually work?



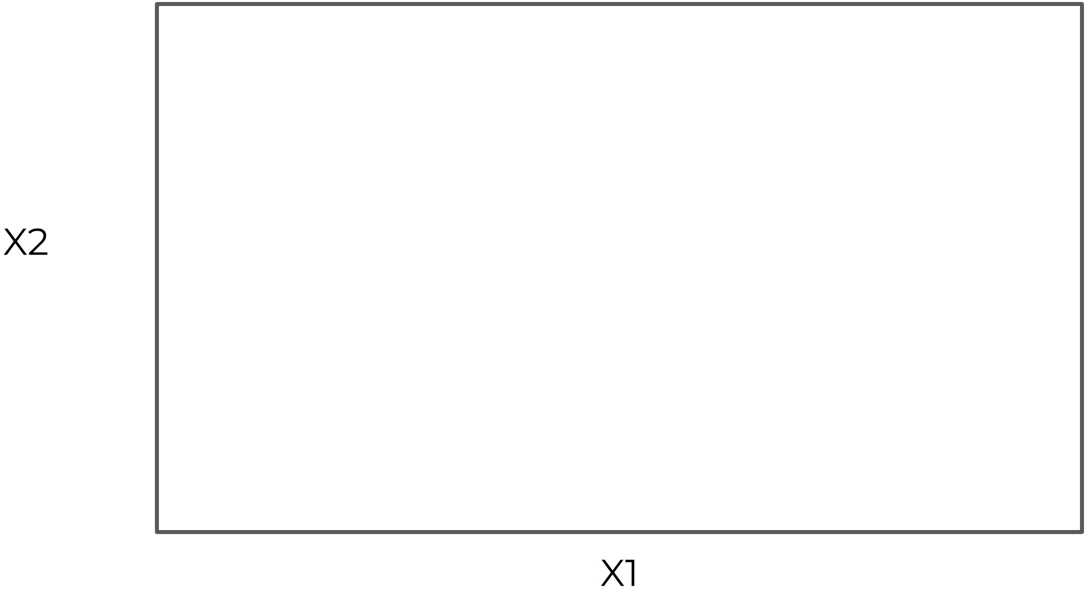
Clustering

- Note: For visualization purposes, we'll work with a simple dataset with only 2 features.



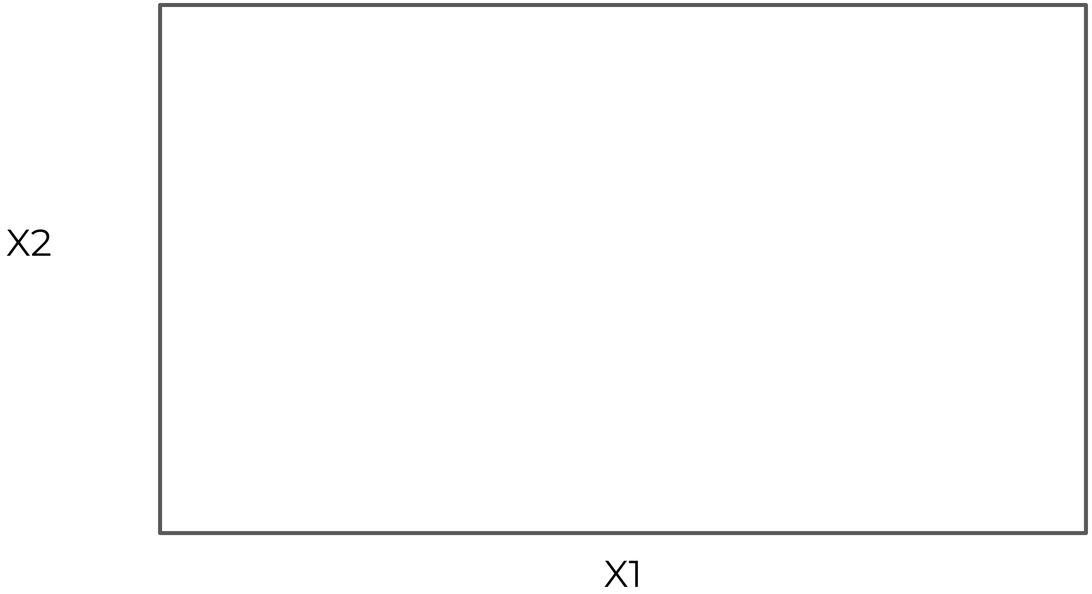
Clustering

- The process shown here easily extends to **N** feature dimensions.



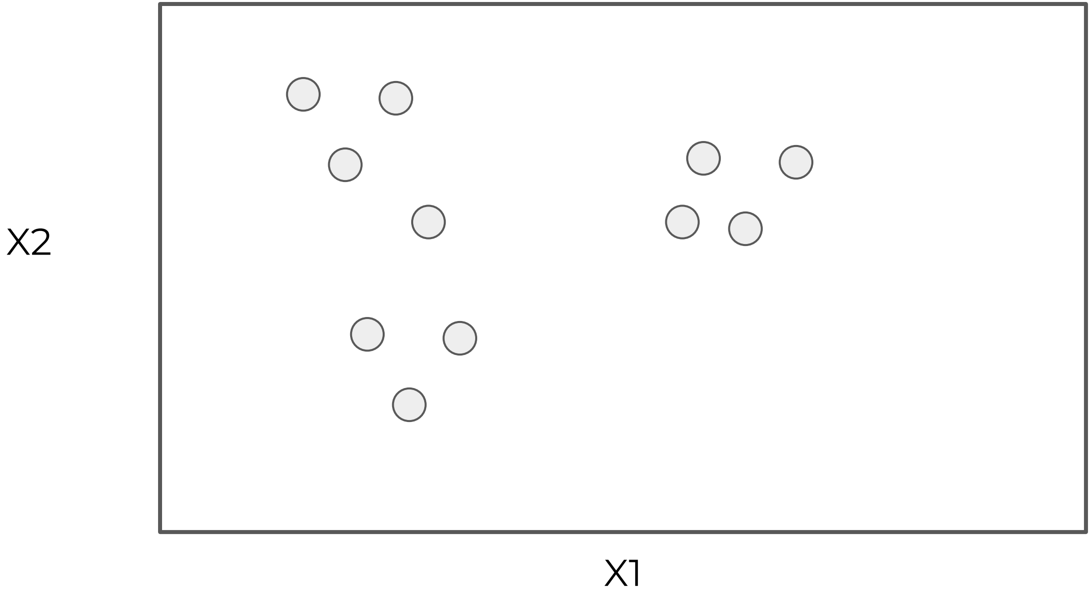
Clustering

- Step 1: Start with unlabeled data (only features).



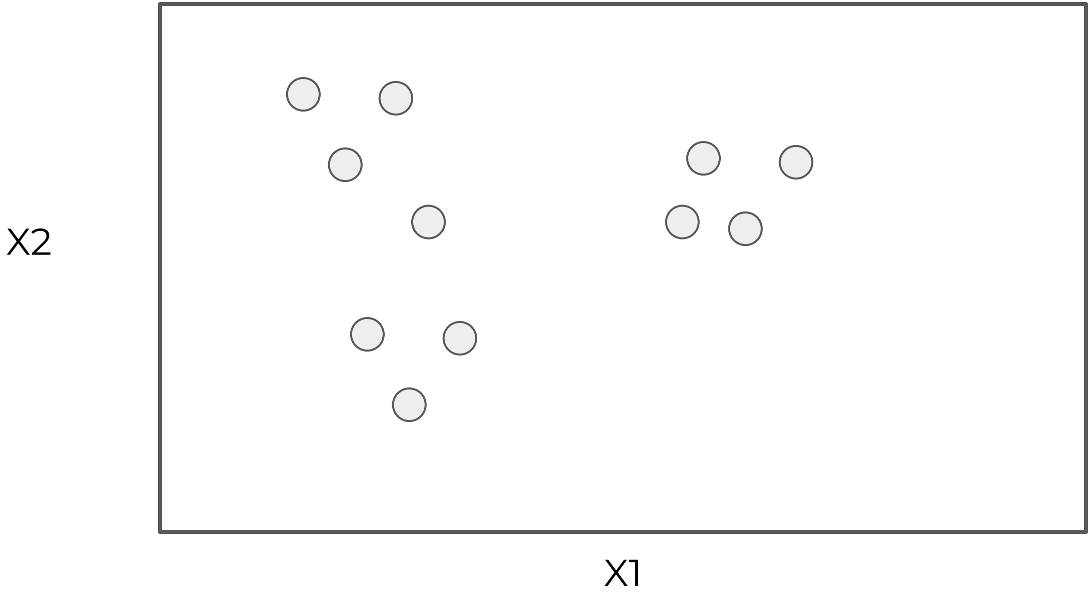
Clustering

- Step 0: Start with unlabeled data (only features).



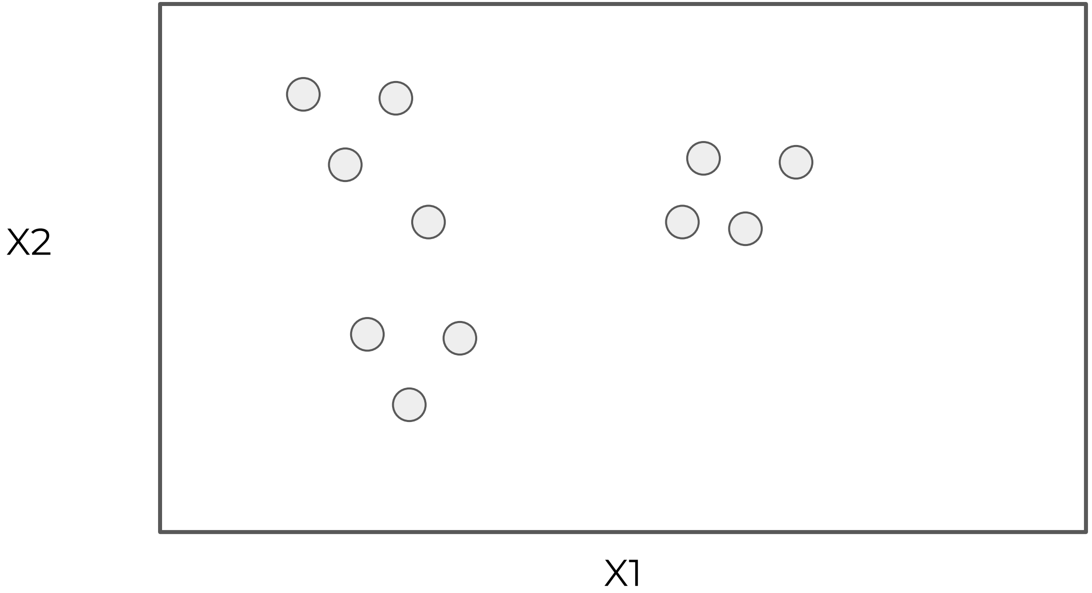
Clustering

- Note: *If we had the group labels, it wouldn't make sense to cluster!*



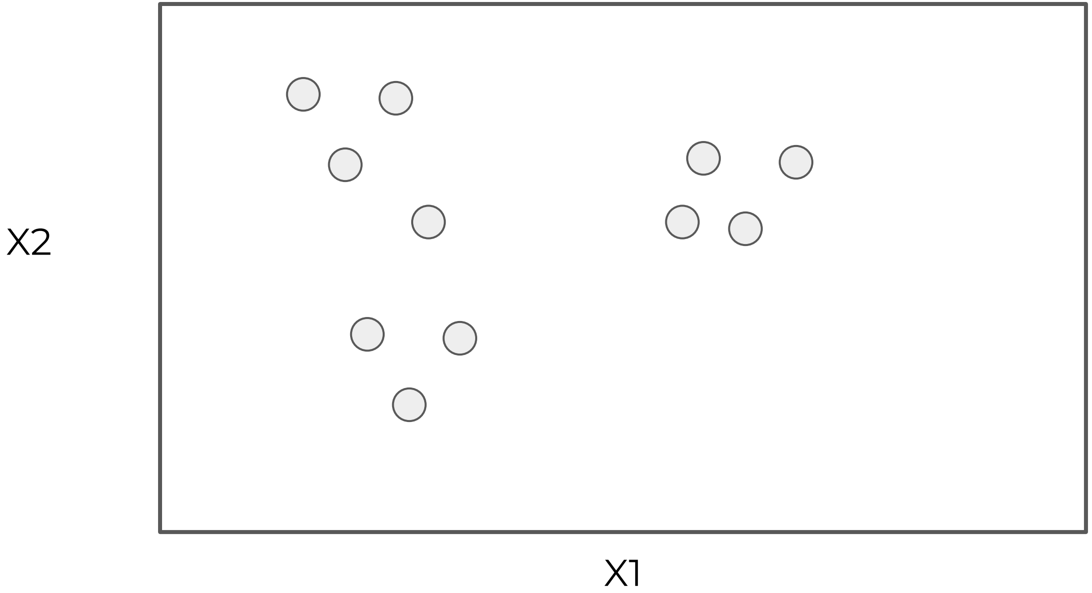
Clustering

- Step 1: Choose the number of clusters to create (this is the K value).



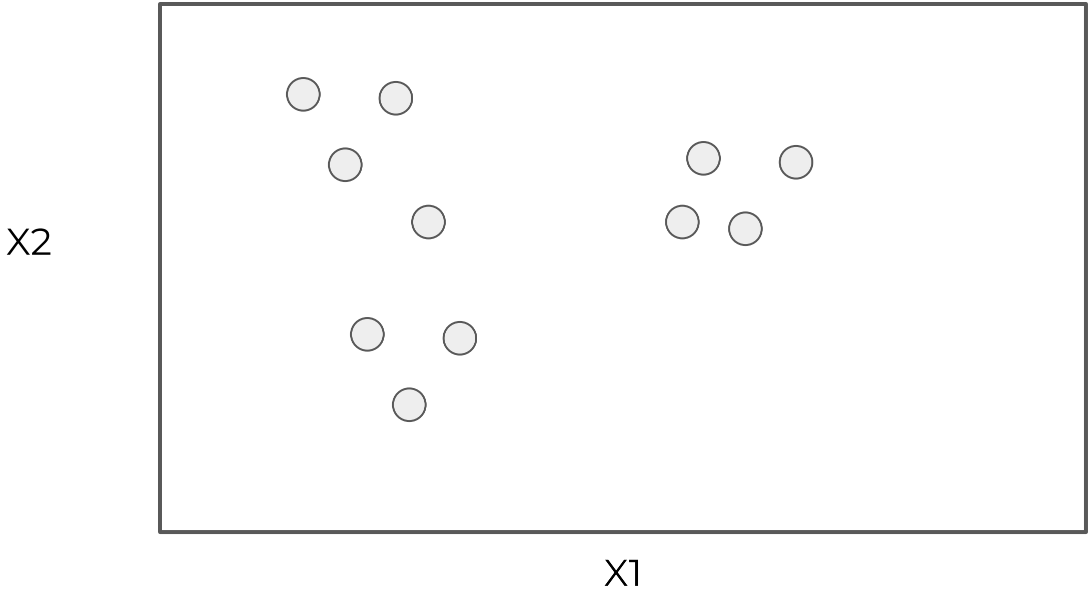
Clustering

- Step 1: We'll choose K=3. Note in most situations you won't visualize the data!



Clustering

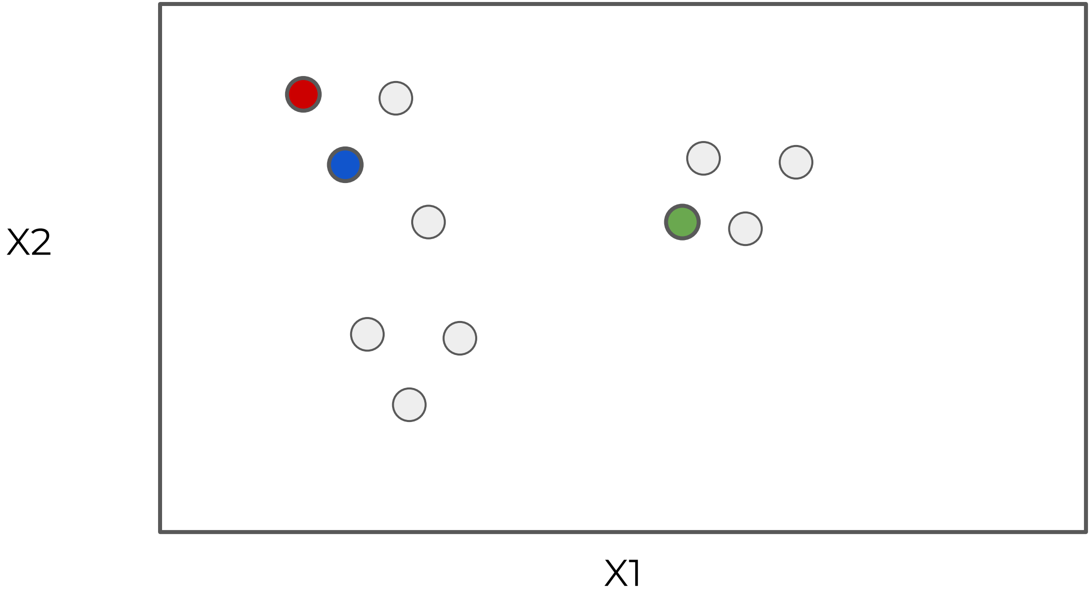
- Step 2: Randomly select K distinct data points.



DATAVALLEY
IT SOLUTIONS | TRAININGS | PLACEMENTS

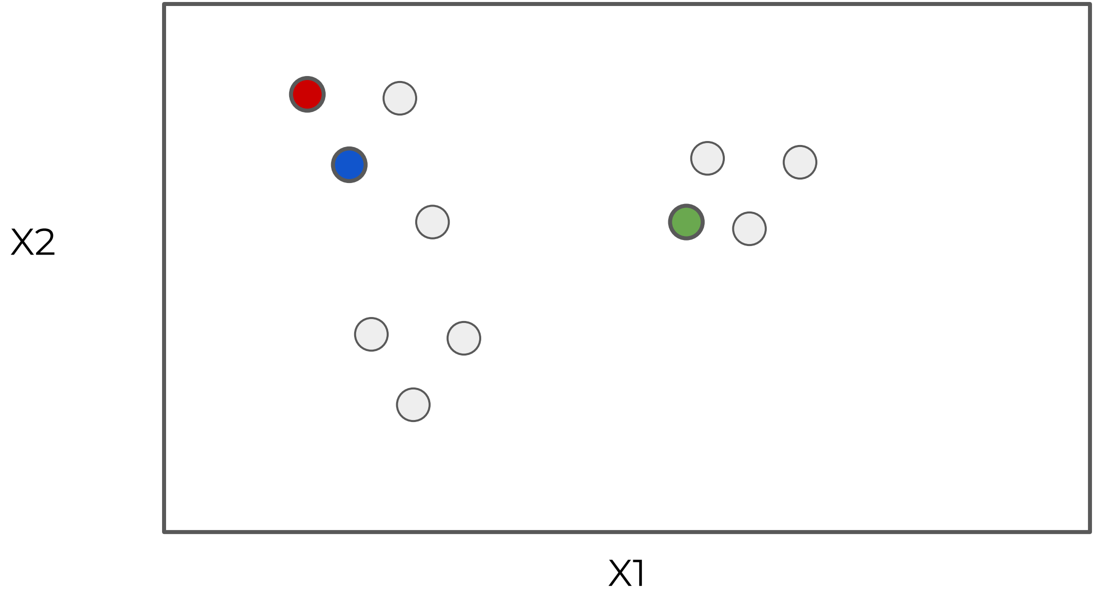
Clustering

- Step 2: Randomly select K distinct data points. Our K=3:



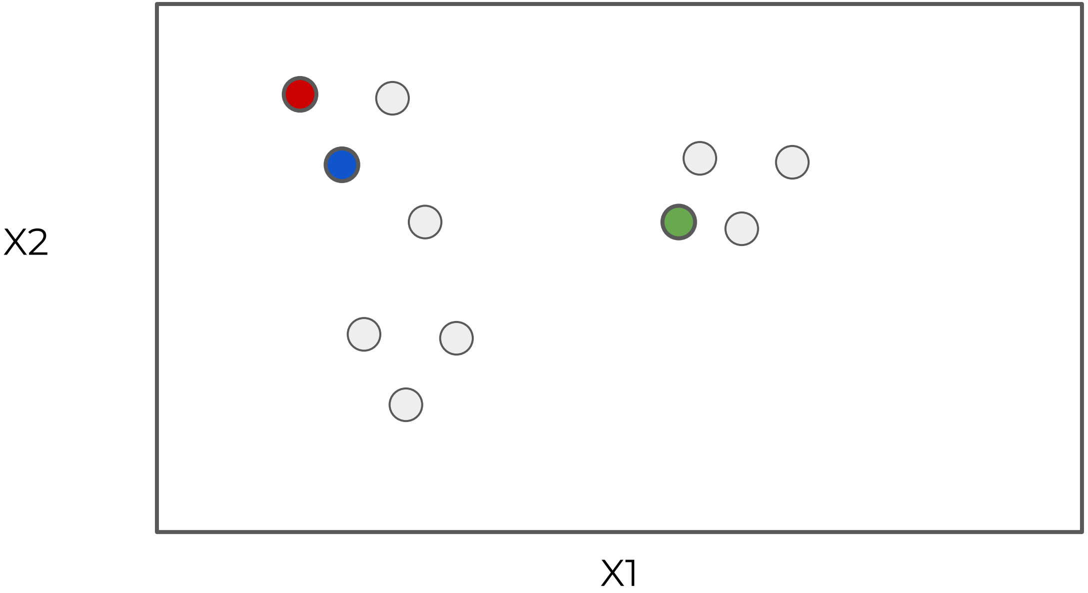
Clustering

- Step 2: We'll treat these new K points as our "cluster" points.



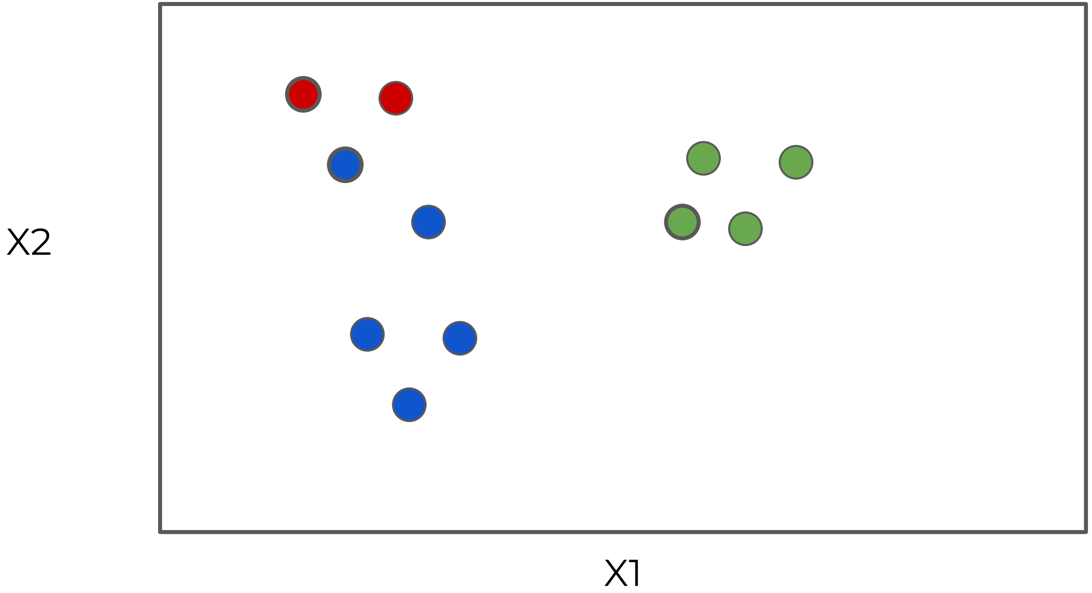
Clustering

- Step 3: Assign each remaining point to the nearest “cluster” point.



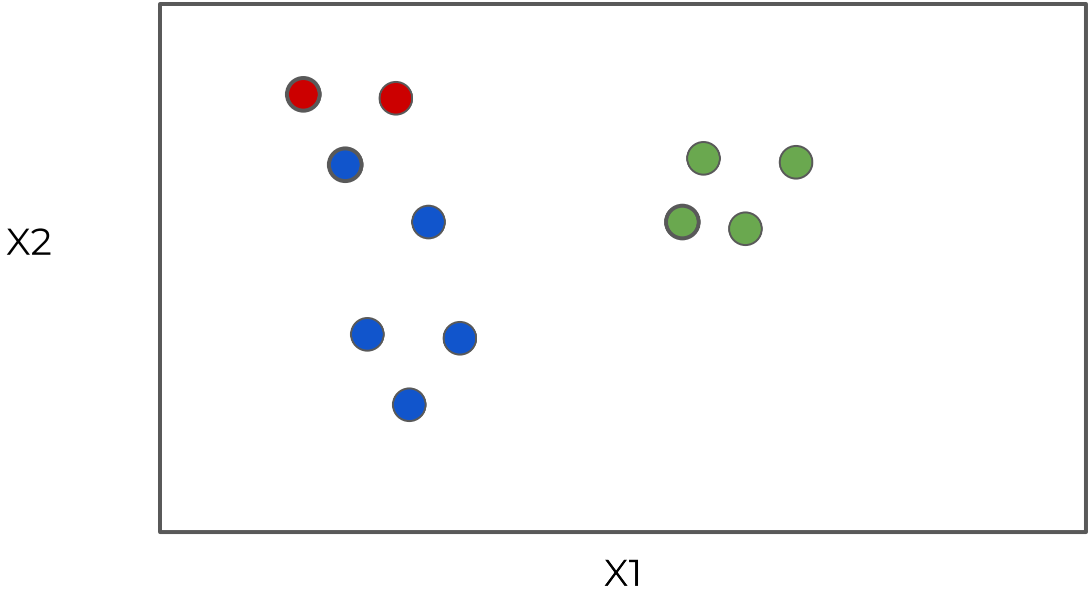
Clustering

- Step 3: Assign each remaining point to the nearest “cluster” point.



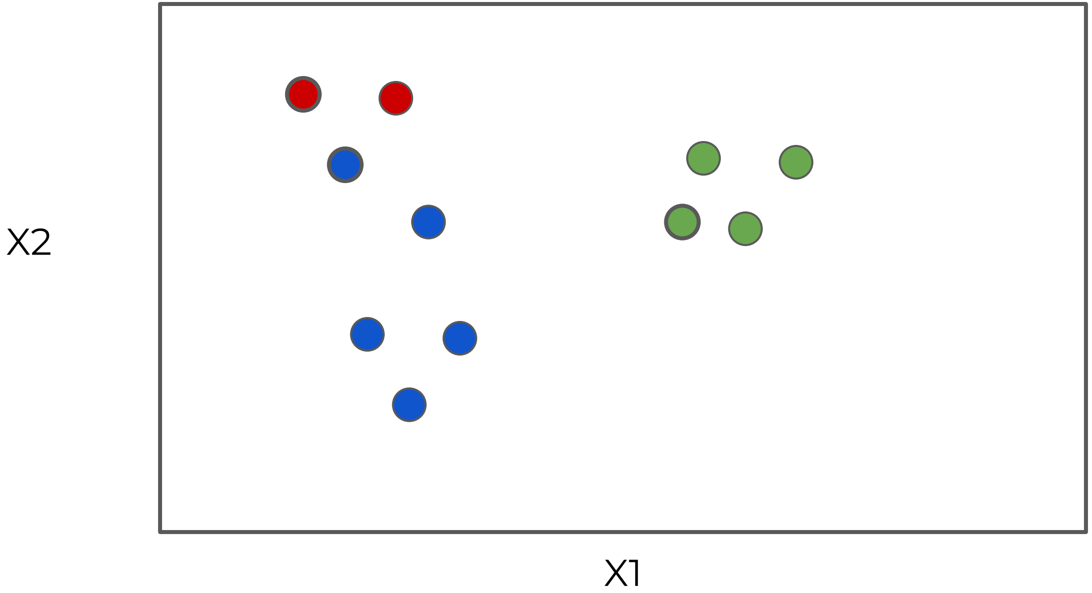
Clustering

- Step 3: Note how this is using a distance metric to judge the nearest point.



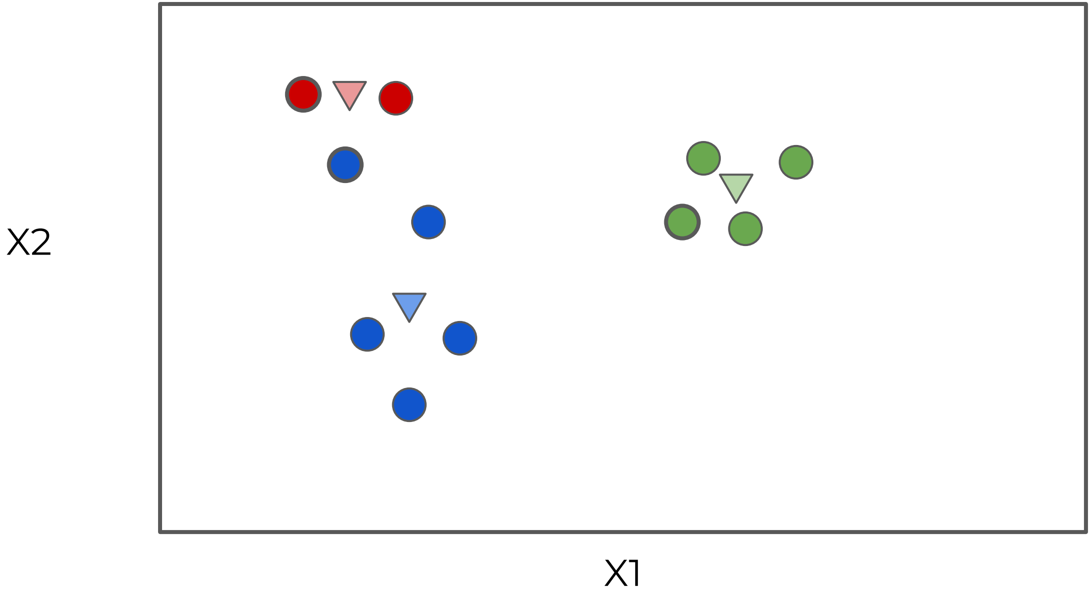
Clustering

- Step 4: Calculate the center of the cluster points (mean value of each point vector).



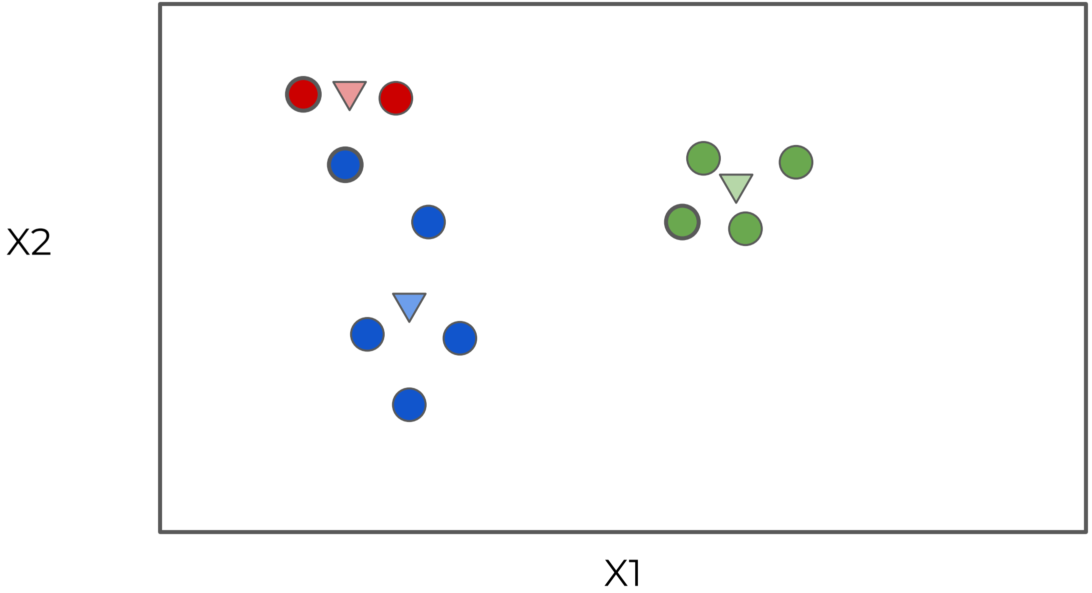
Clustering

- Step 4: Calculate the center of the cluster points (mean value of point vectors).



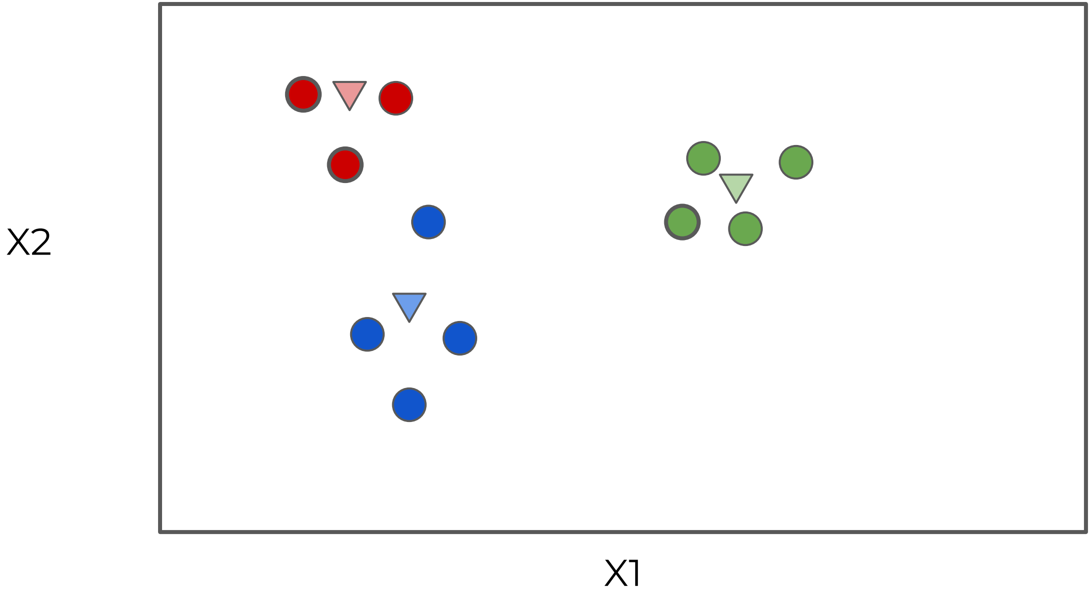
Clustering

- Step 5: Now assign each point to the nearest cluster center.



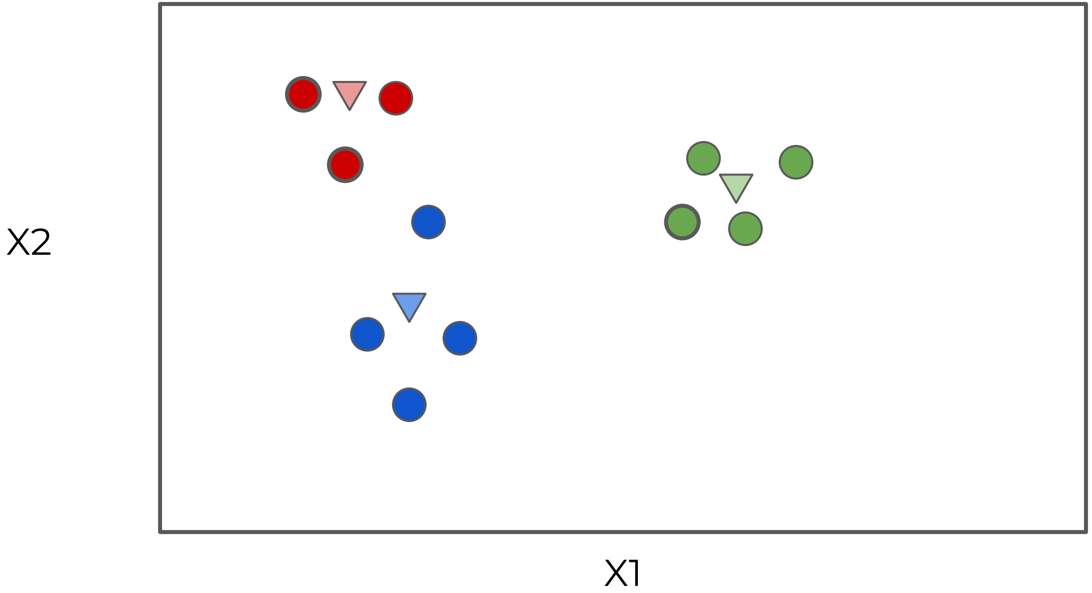
Clustering

- Step 5: Now assign each point to the nearest cluster center.



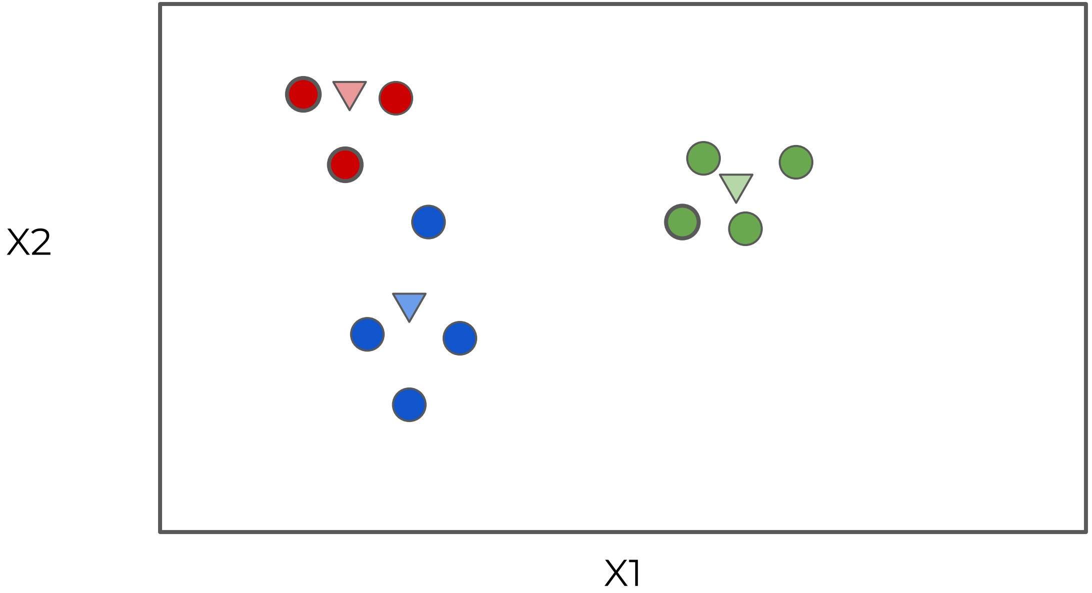
Clustering

- We repeat steps 4 and 5 until there are no more cluster reassignments.



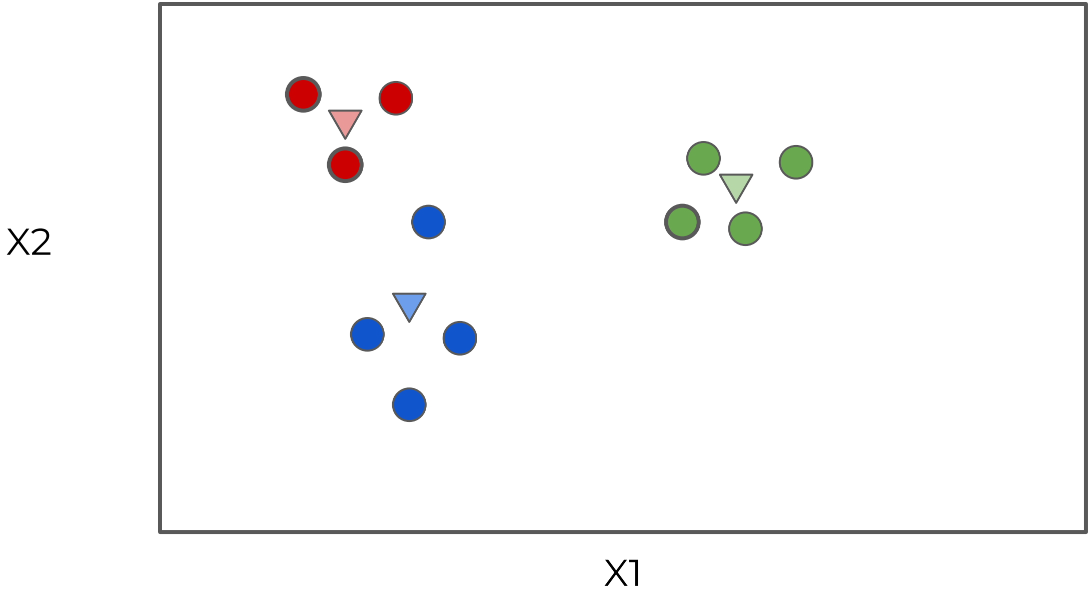
Clustering

- Step 4b: Recalculate new cluster centers:



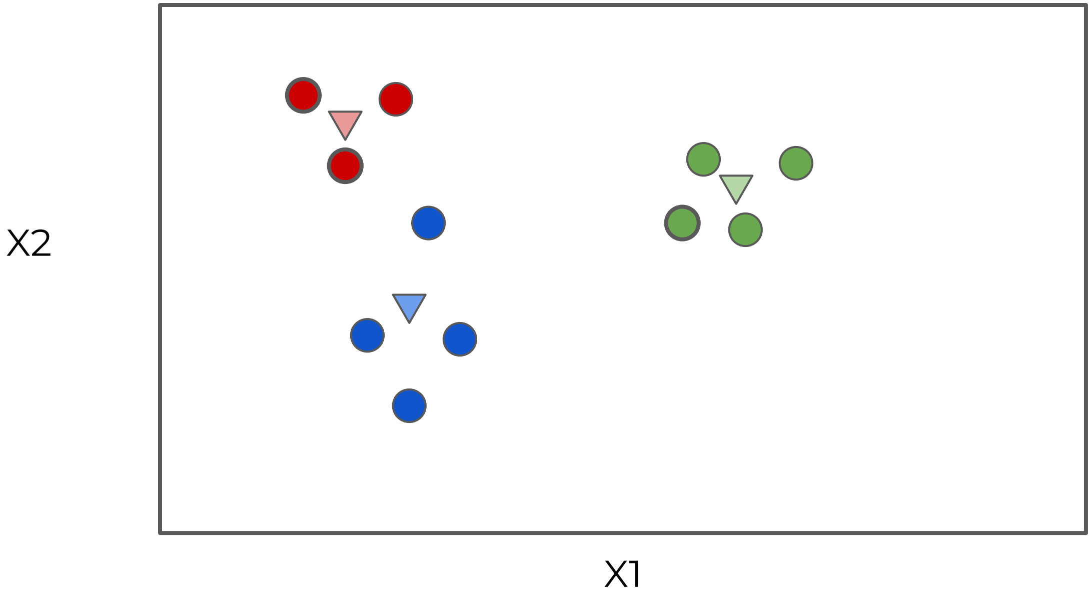
Clustering

- Step 4b: Recalculate new cluster centers:



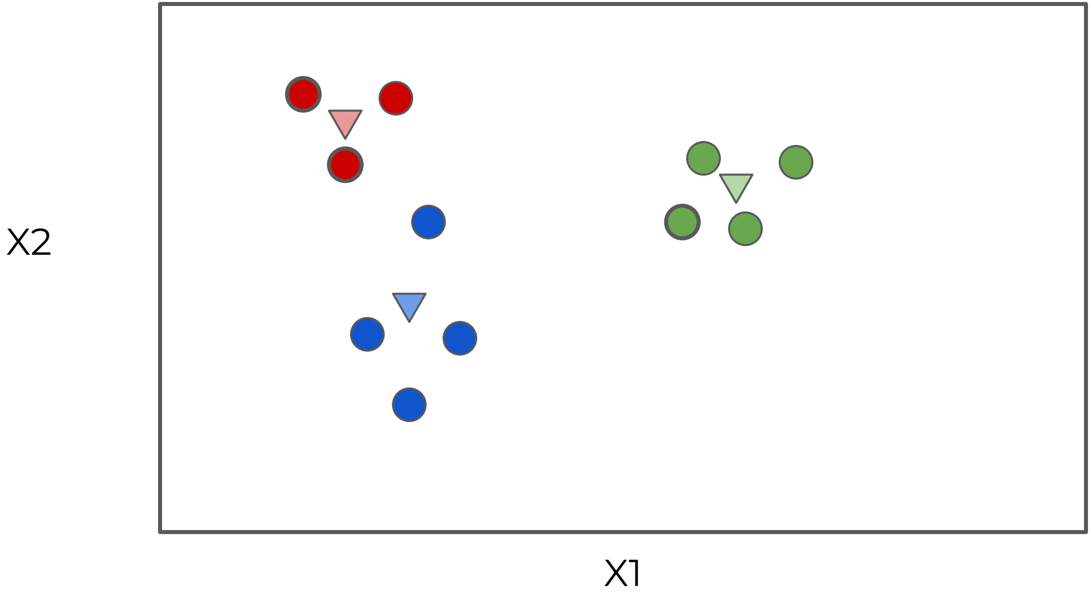
Clustering

- Step 5b: Assign points to nearest cluster center.



Clustering

- If there are no more reassignments, we're done! The clusters have been found.



Clustering

- **Upcoming considerations:**
  - How do we choose a reasonable value for K number of clusters?
  - Is there any way we can evaluate how good our current K value is at determining clusters?

K-Means Clustering

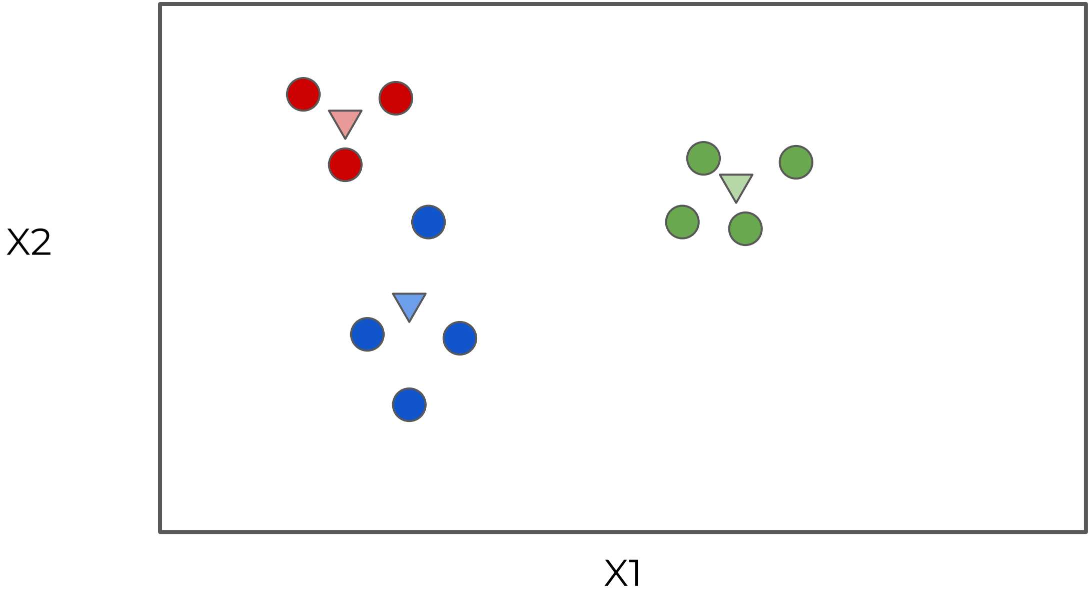
Choosing a K Value

Clustering

- **Recall our previous considerations:**
  - How do we choose a reasonable value for K number of clusters?
  - Is there any way we can evaluate how good our current K value is at determining clusters?

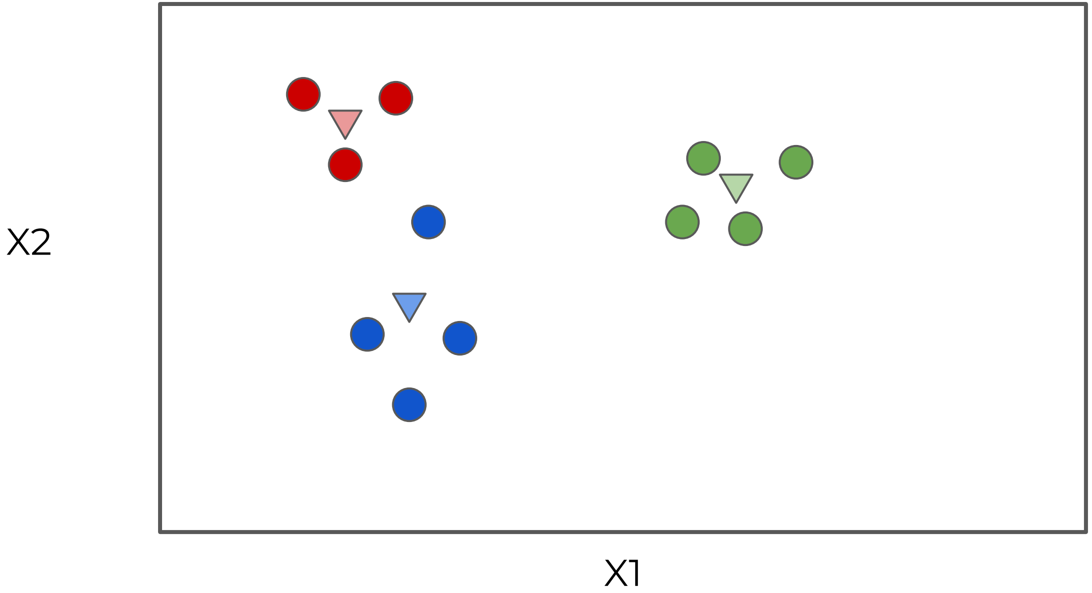
Clustering

- Here we have 3 clusters, how can we measure “goodness of fit”?



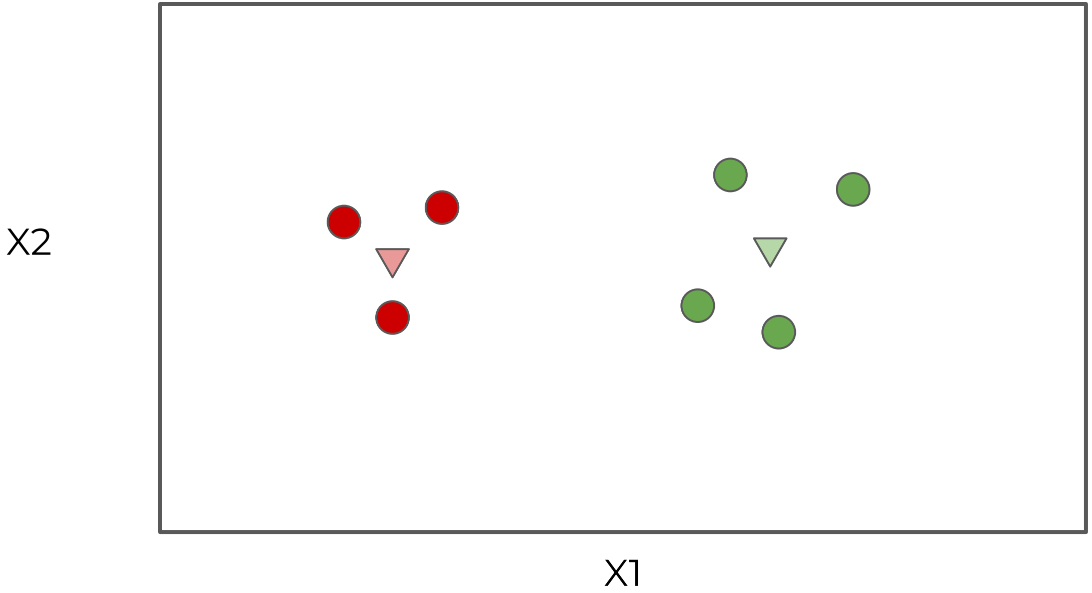
Clustering

- We could measure the sum of the distances from points to cluster centers.



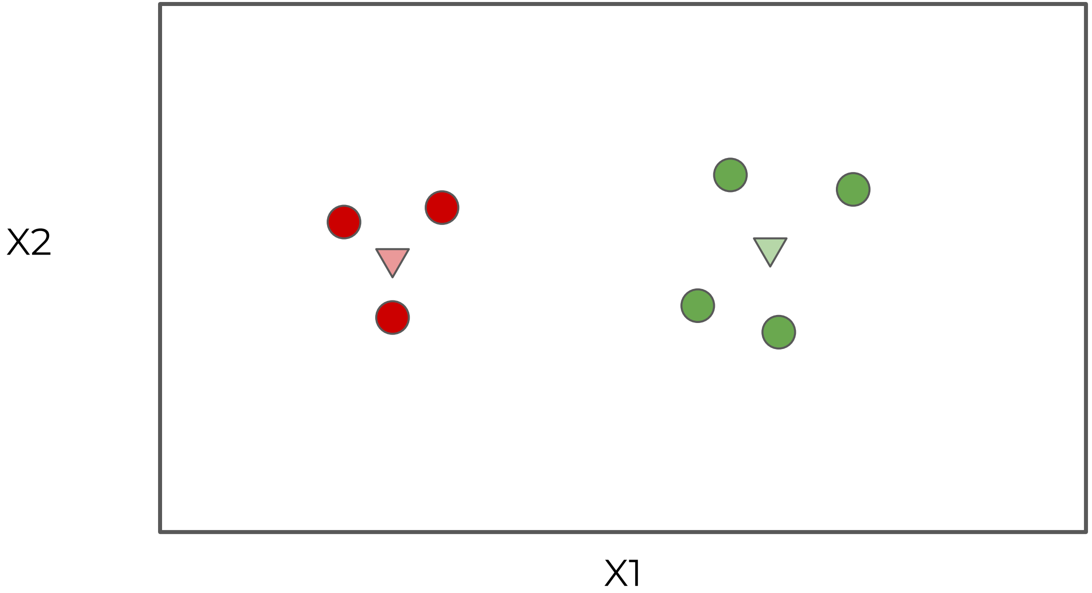
Clustering

- Imagine a simple example starting with K=2.



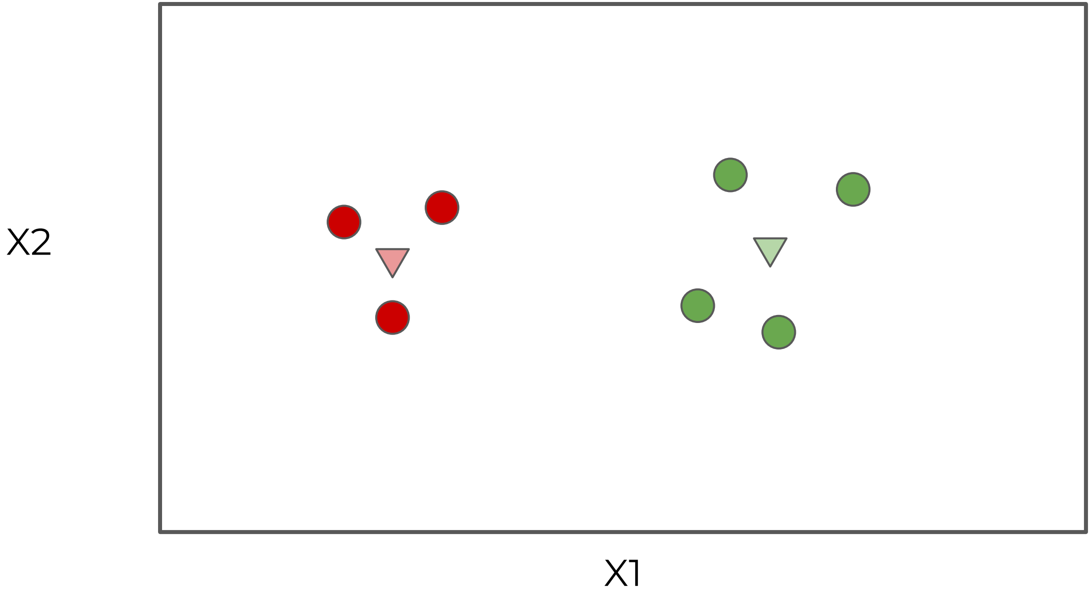
Clustering

- We measure the sum of the squared distances from points to the cluster center:



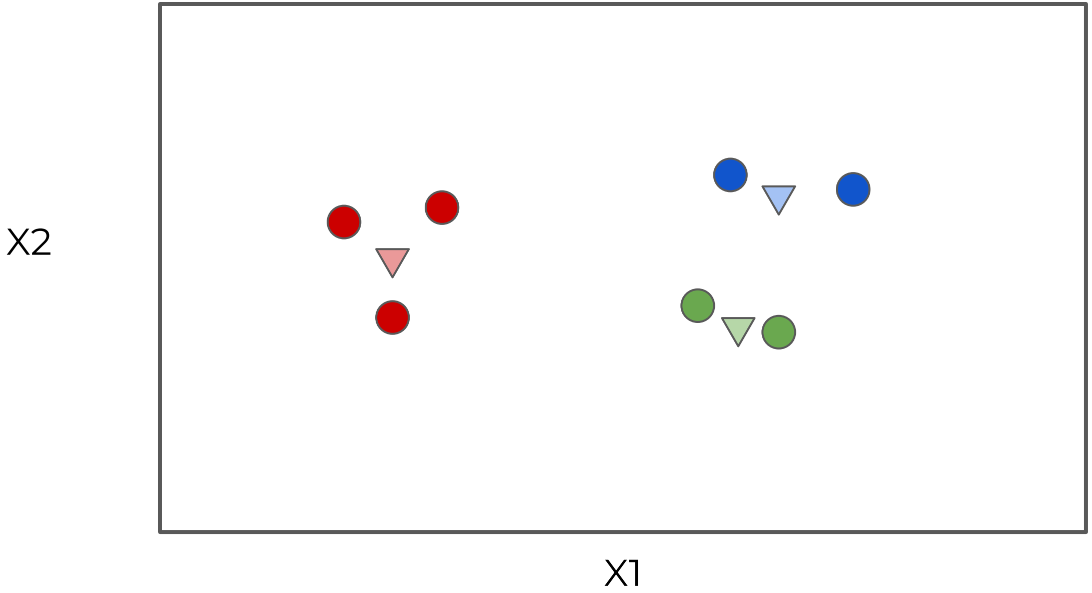
Clustering

- Then we fit an entirely new KMeans model with K+1:



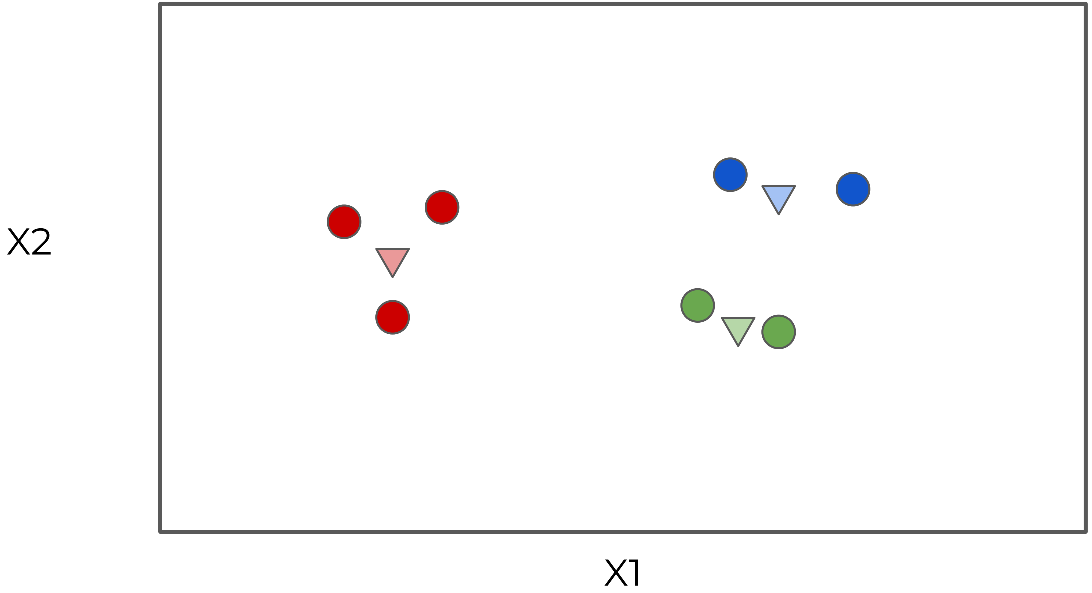
Clustering

- Then we fit an entirely new KMeans model with K+1:



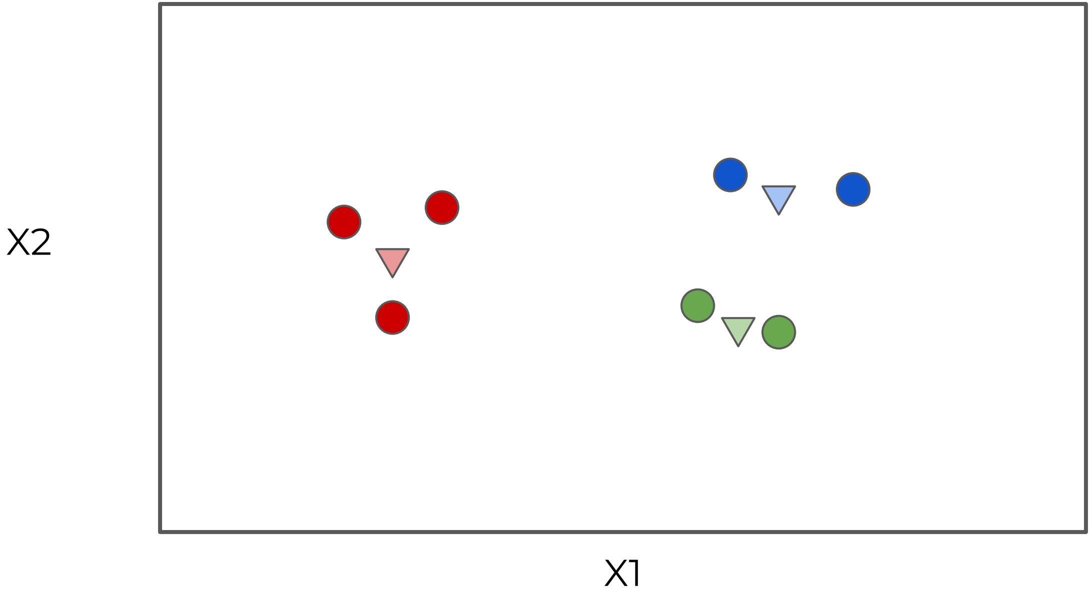
Clustering

- Then measure again the sum of the squared distance (SSD) to center.



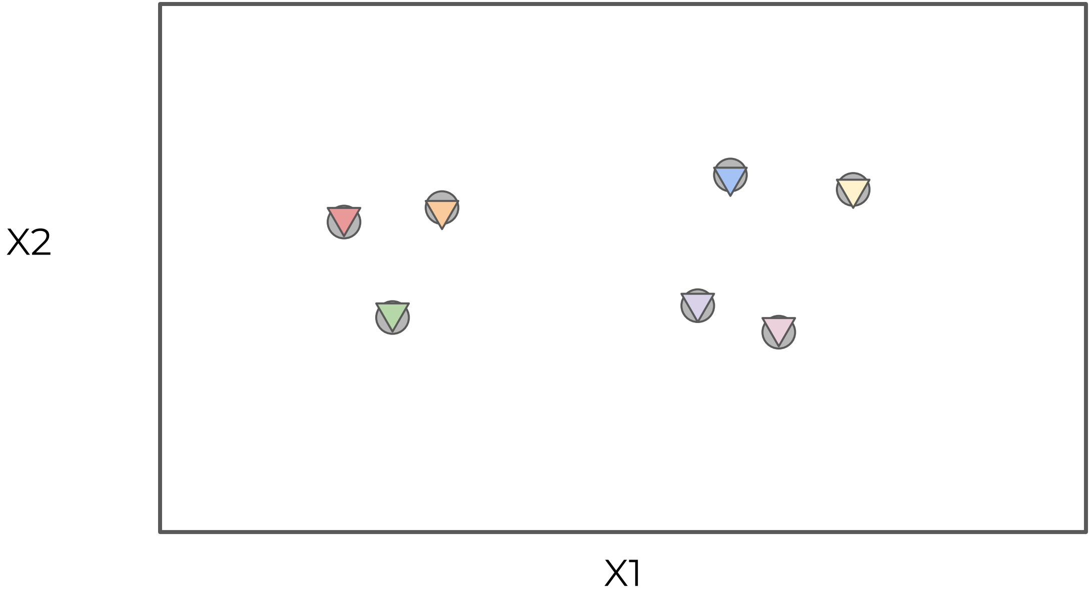
Clustering

- In theory this SSD would go to zero once K is equal to the number of points.



Clustering

- You would have a cluster for each point! SSD would be perfect at 0!



Clustering

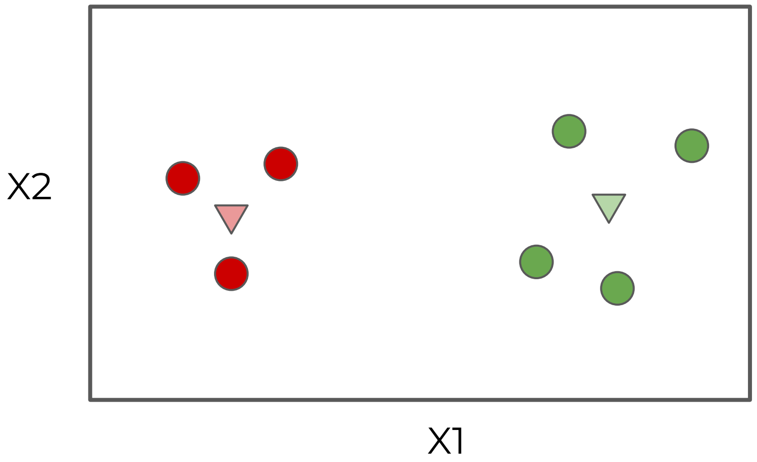
- We keep track of this SSD value for a range of different K values.
- We then look for a K value where **rate of reduction in SSD** begins to decline.
- This signifies that adding an extra cluster is **not** obtaining enough clarity of cluster separation to justify increasing K.

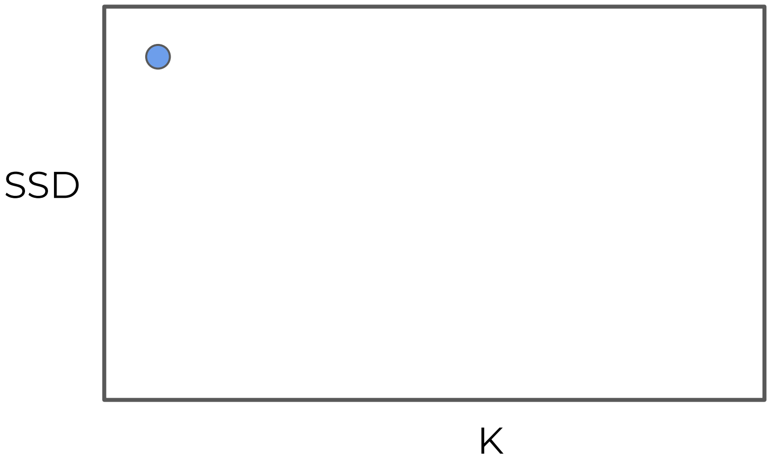
Clustering

- This is known as the “elbow” method since we will track where decrease in SSD begins to flatten out compared to increasing K values.
- Let’s walk through what this chart would look like…

Clustering

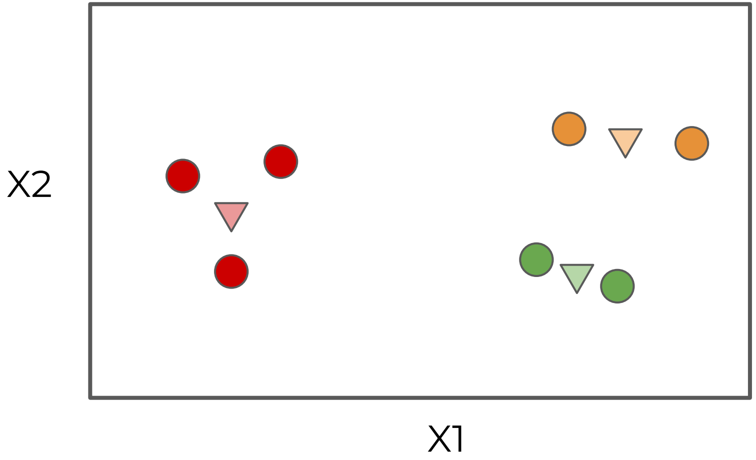
- Start with K=2:

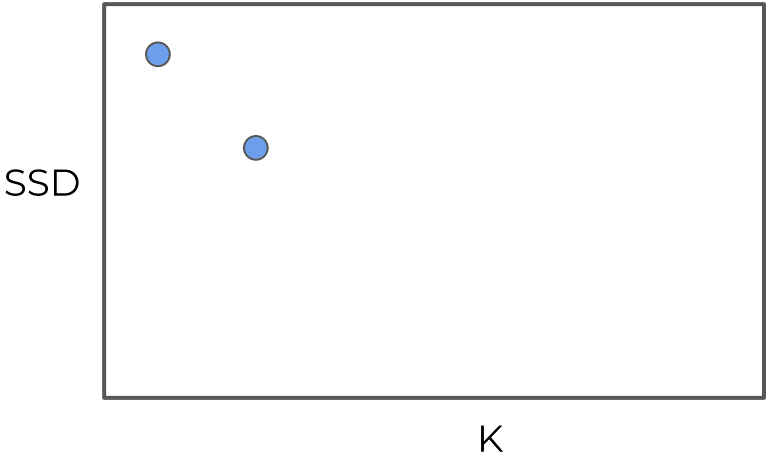




Clustering

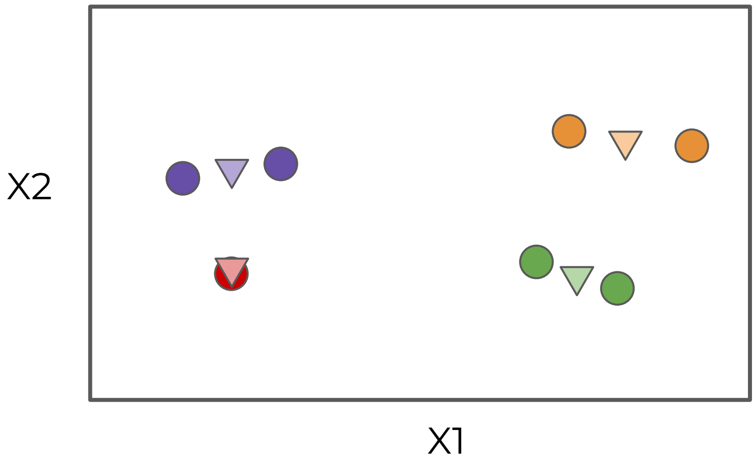
- Increase K and measure SSD:

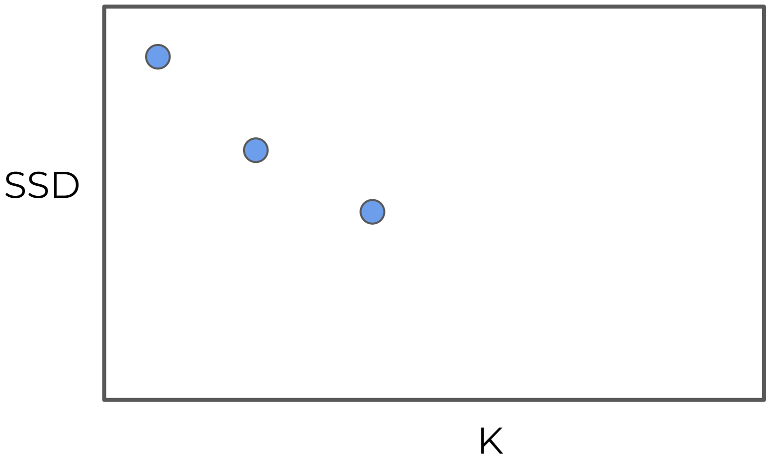




Clustering

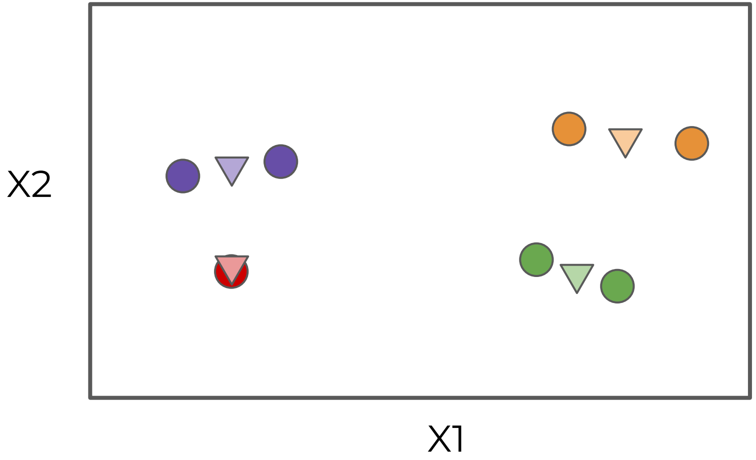
- Increase K and measure SSD:

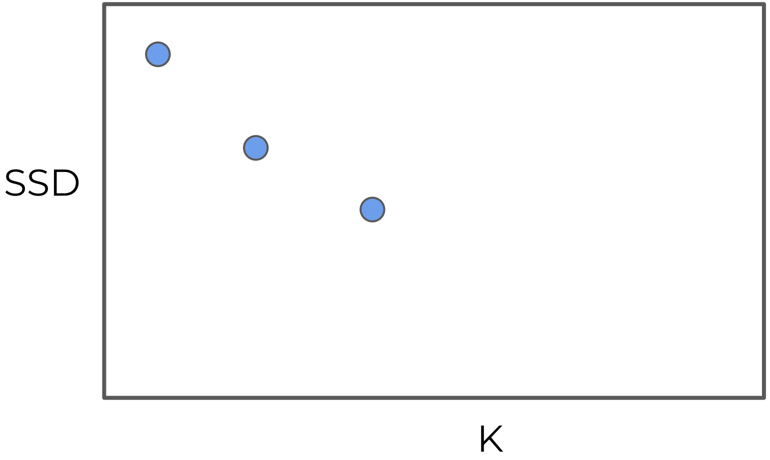




Clustering

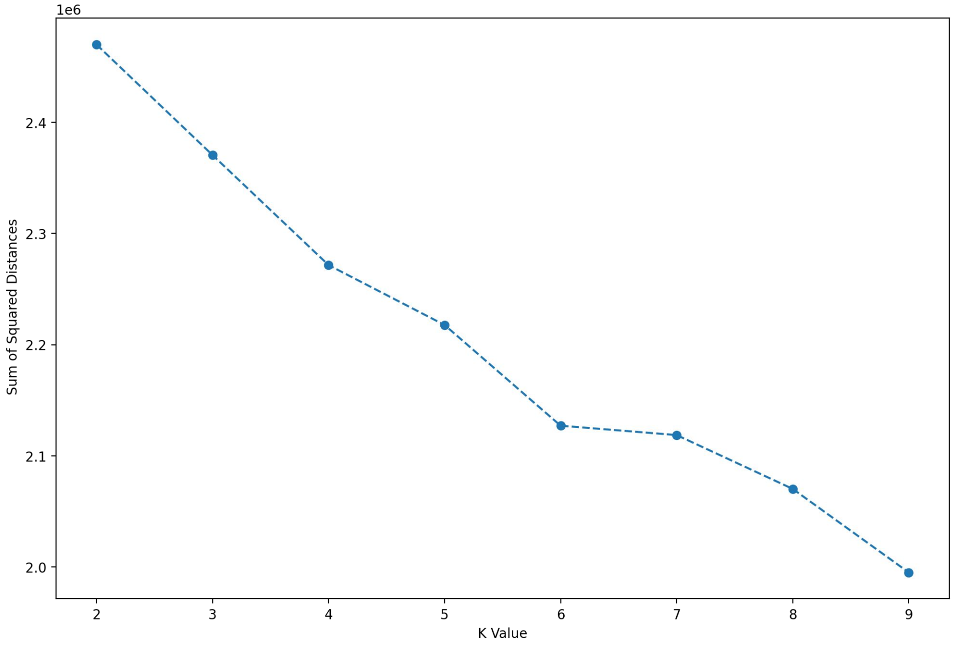
- Repeat this process for some set number of K values:





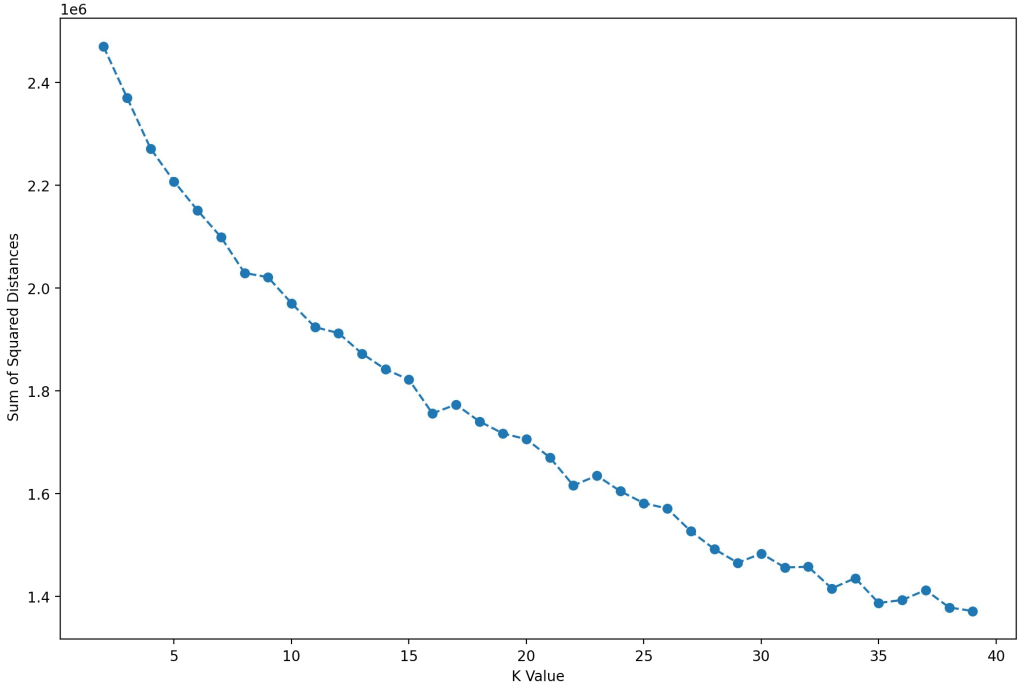
K Means Clustering

- You will see a continuous decline.



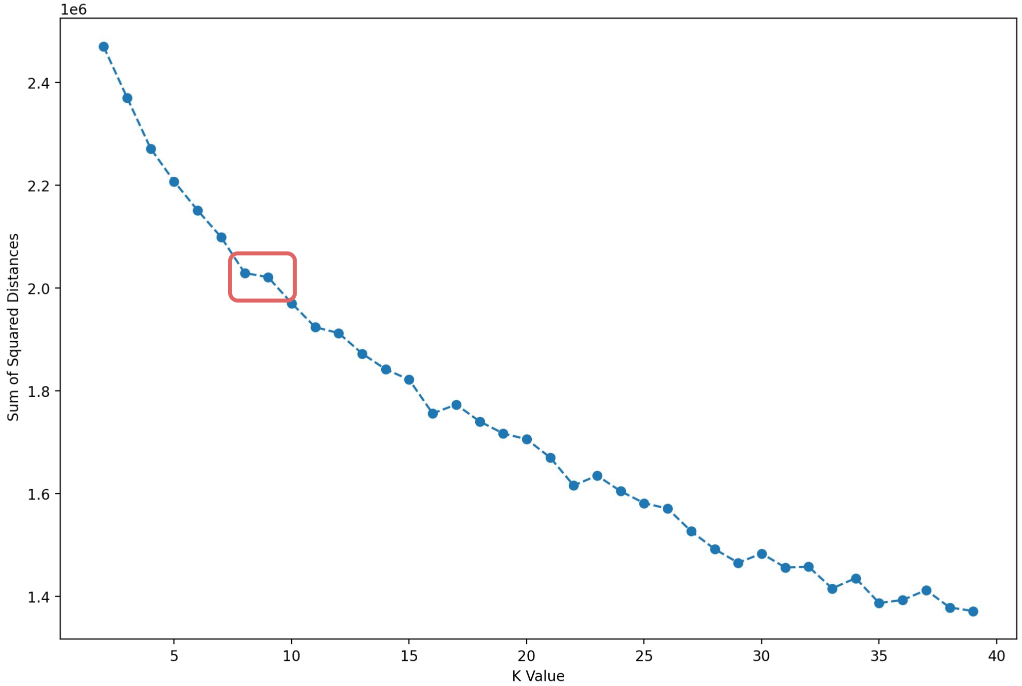
K Means Clustering

- Eventually you will see “elbow” points:



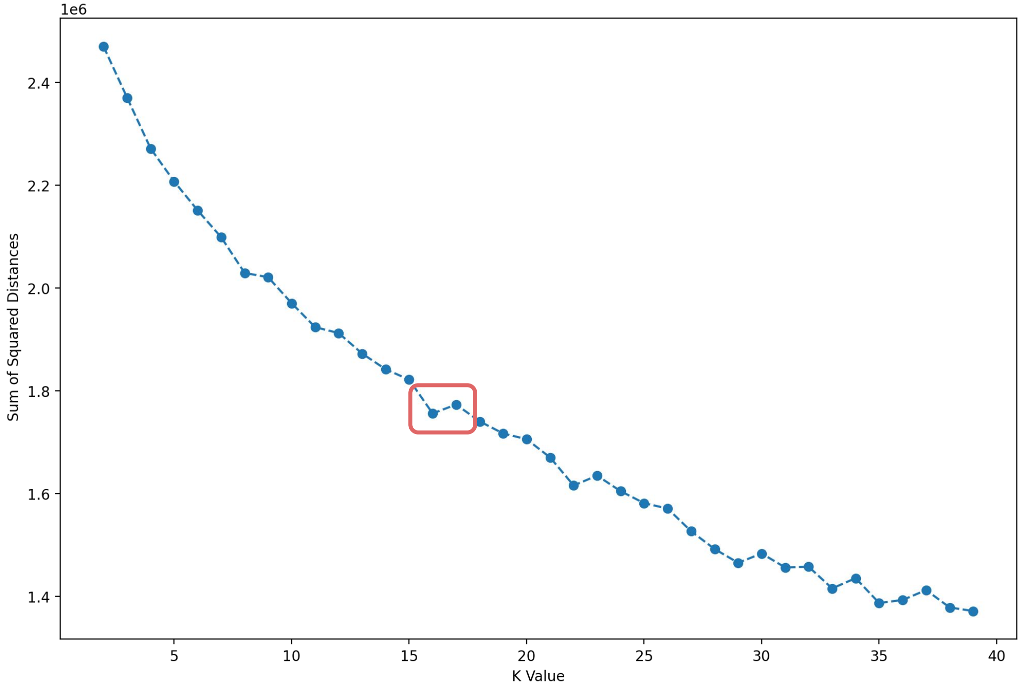
K Means Clustering

- Eventually you will see "elbow" points:



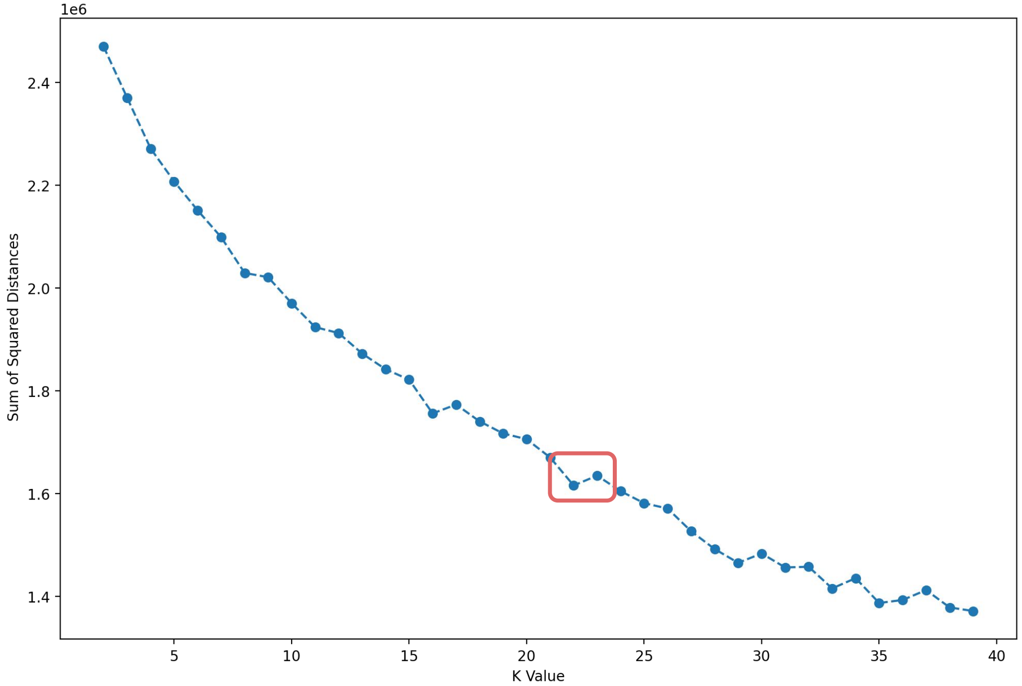
K Means Clustering

- Eventually you will see “elbow” points:



K Means Clustering

- Eventually you will see "elbow" points:



Clustering

- These points are strong indicators that increasing K further is no longer justified as it is not revealing more “signal”.
- You can also measure out this SSD in a barplot.
- Let’s explore this further with code!