ECON441B : Intro Machine Learning Lab

Week 9, Lecture 9 | K-Means Clustering

Sam Borghese

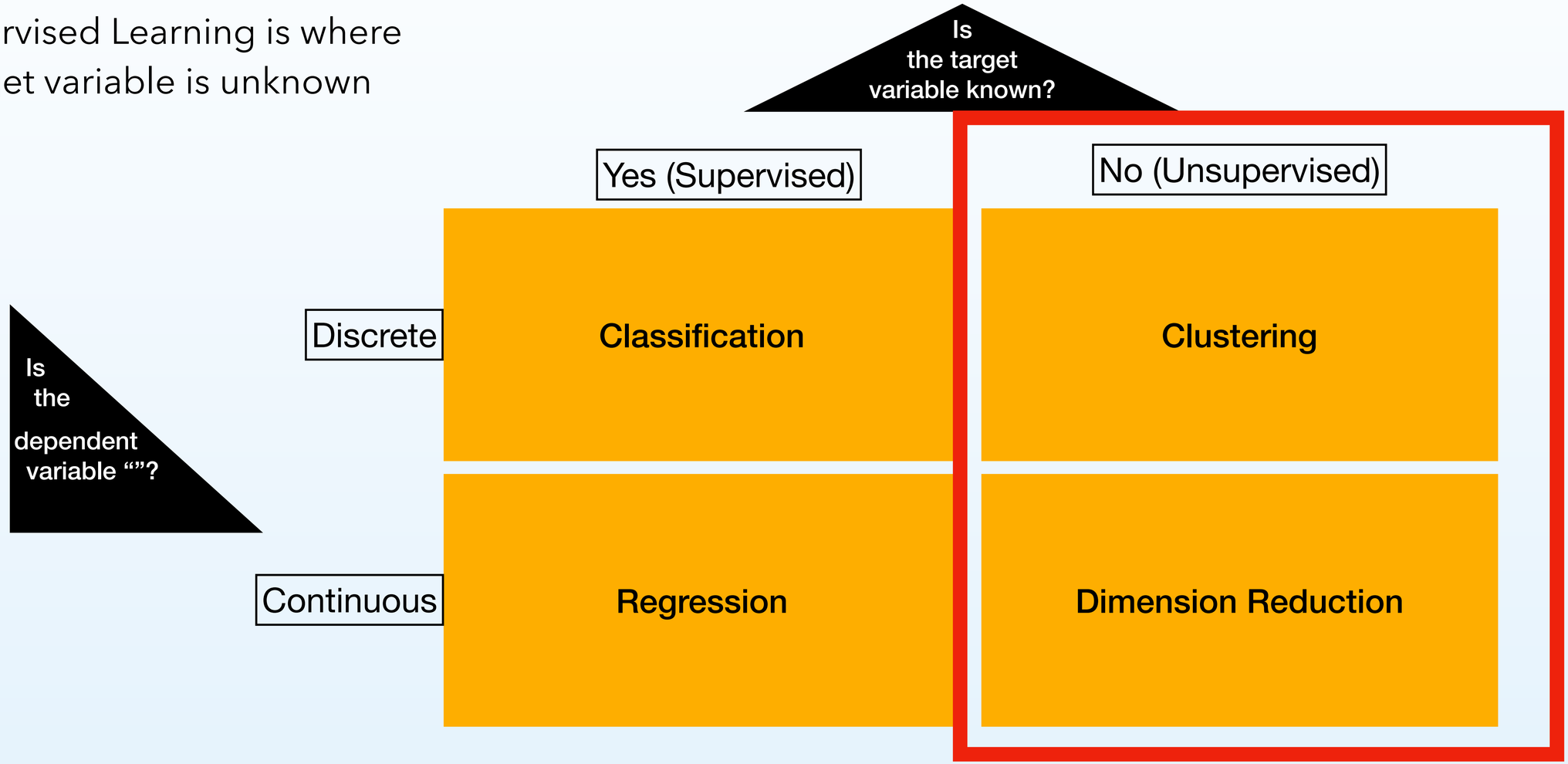
Wednesday, March 6th, 2024

1. Unsupervised Learning
2. K-means Clustering
3. Coding
4. In-Class Assignment

Unsupervised Learning

Unsupervised Learning

Unsupervised Learning is where the target variable is unknown



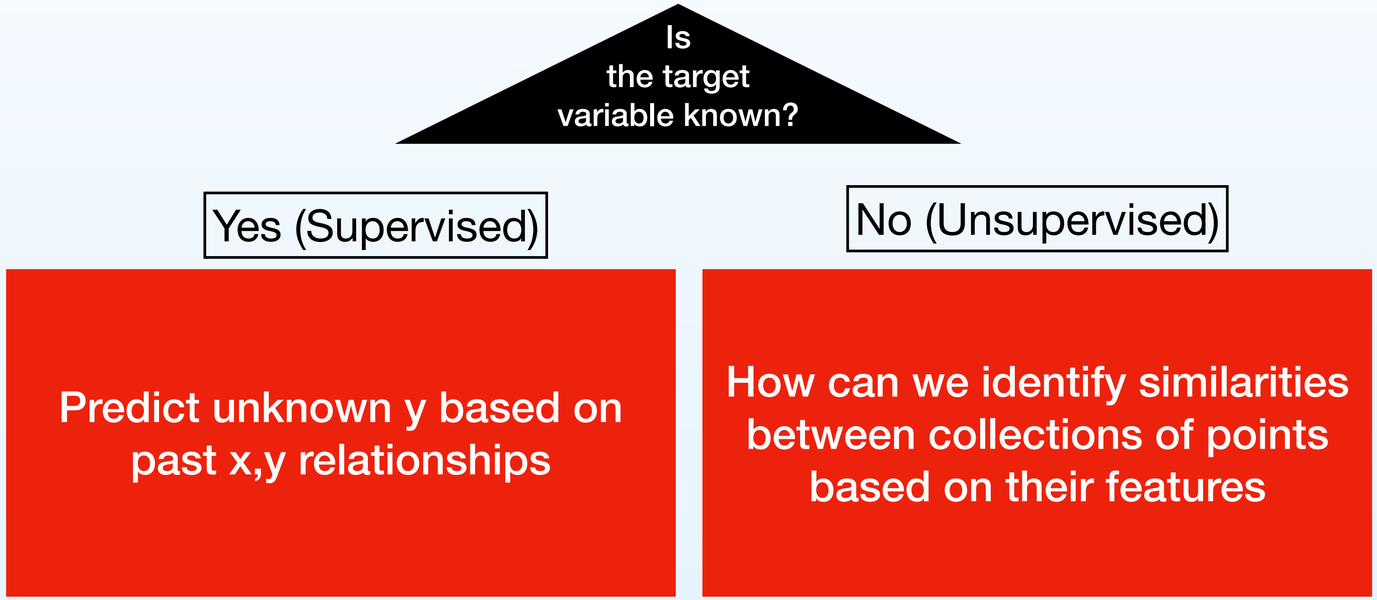
Unsupervised Learning

All we have is independent variables

$$(X_1, X_2, \ldots, X_n)$$

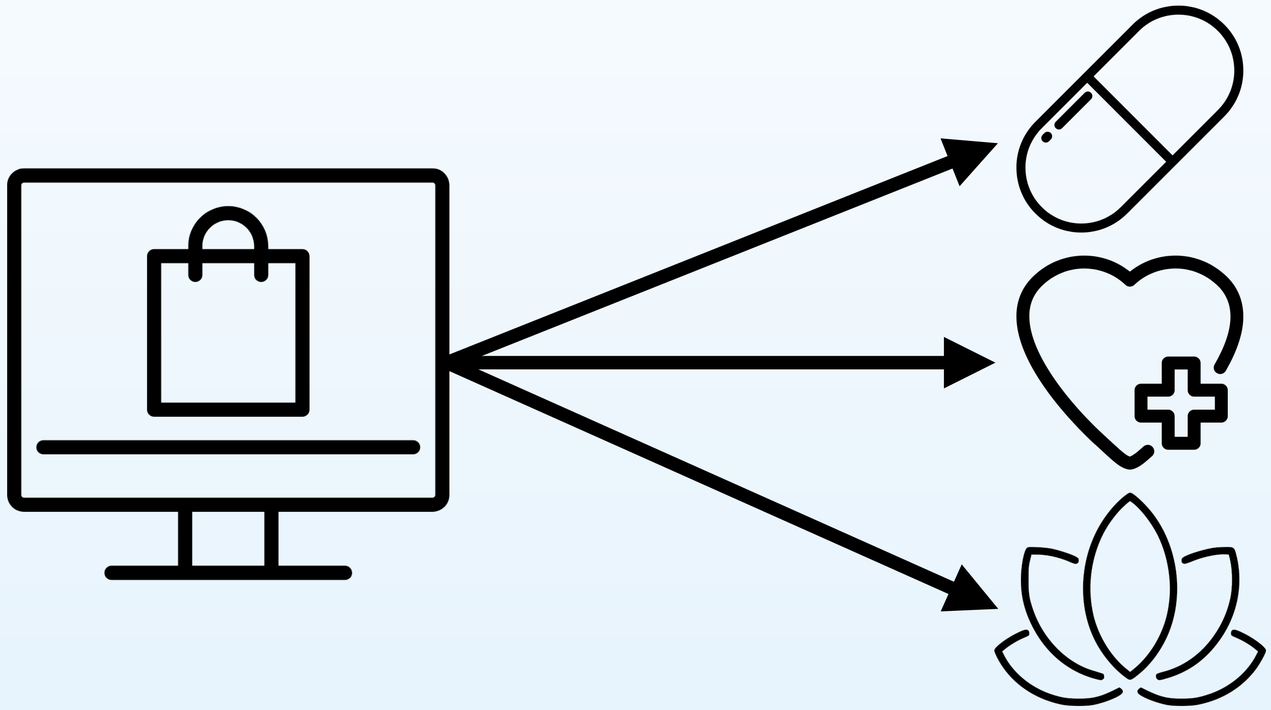
What is the Question?

Coming up with the question in supervised learning is more difficult than supervised



An Example : Health Company

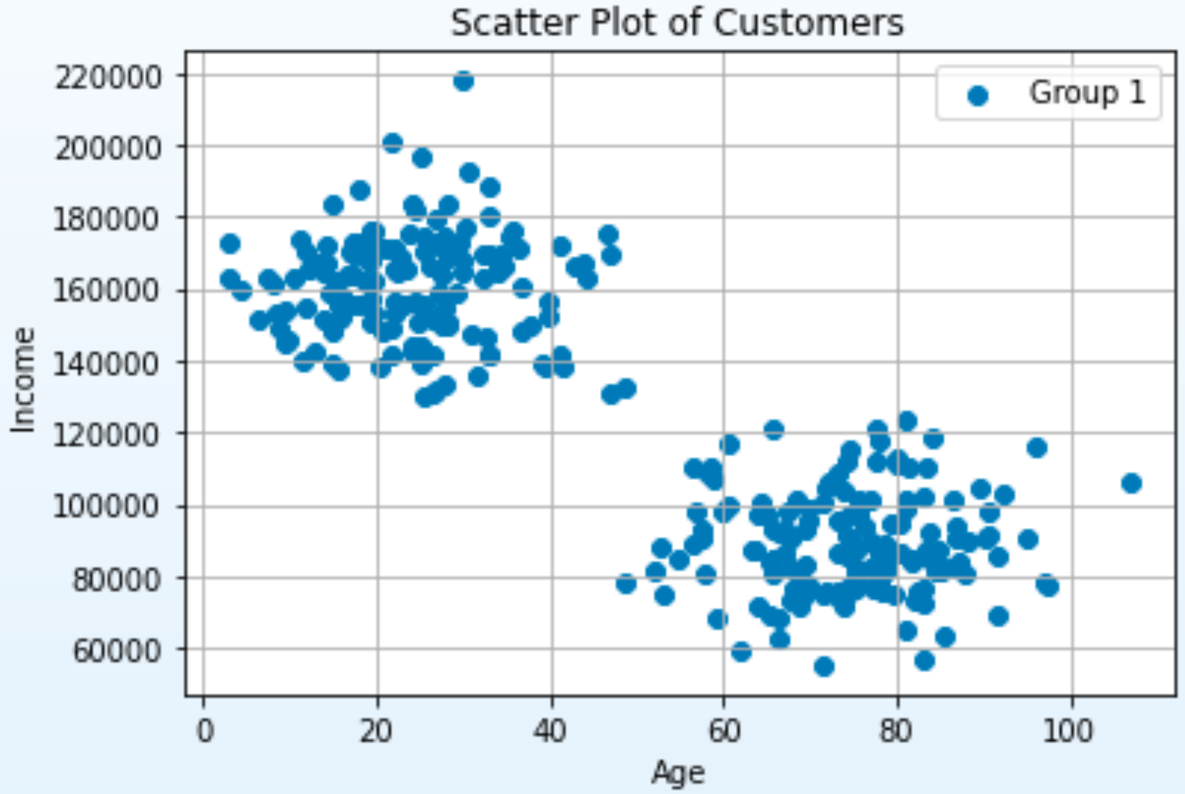
An online store *“Company A”* Sells Health and wellness products to customers



An Example : Health Company

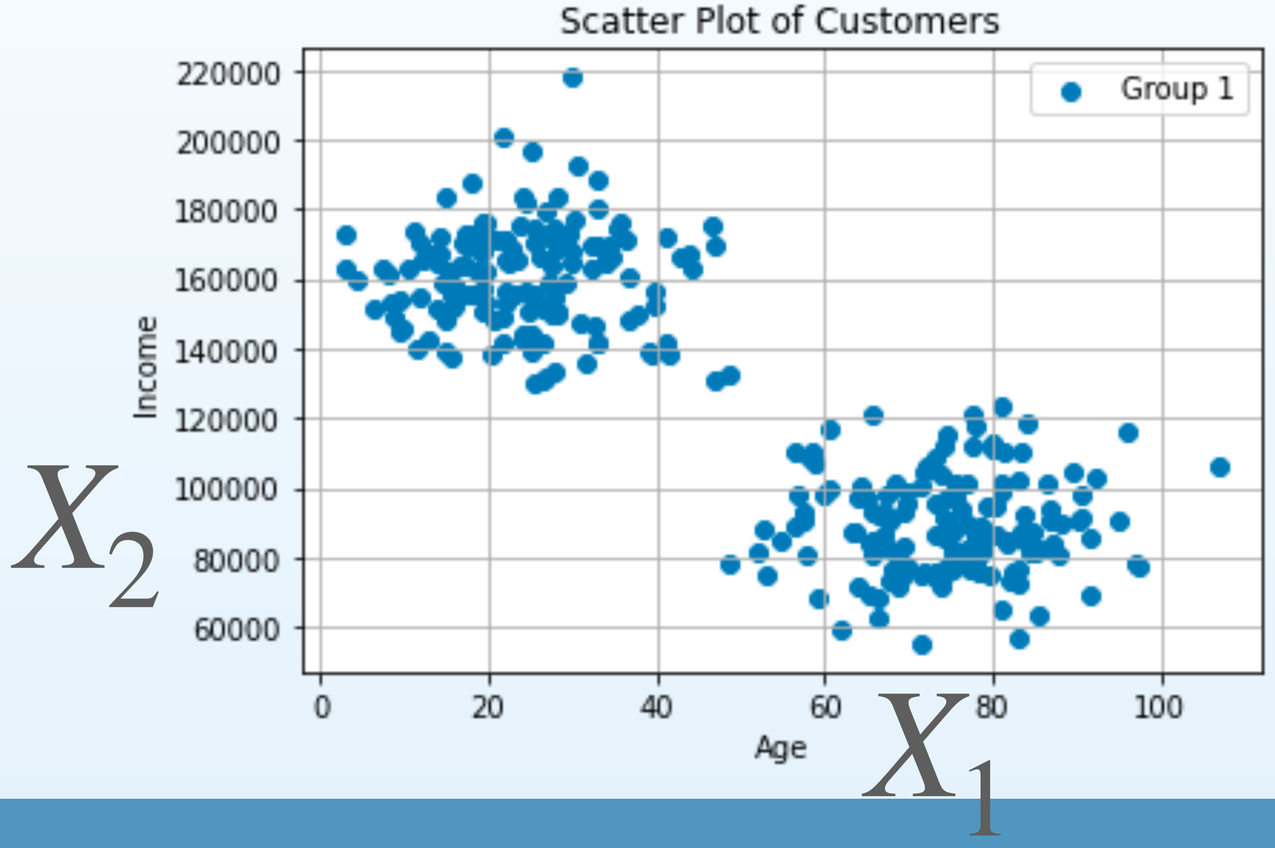
They have data on their clients and want to learn more about who is buying their products

Customer	Age	Income
1	20	100,000
2	22	110,000
3	25	180,000
4	80	42,000



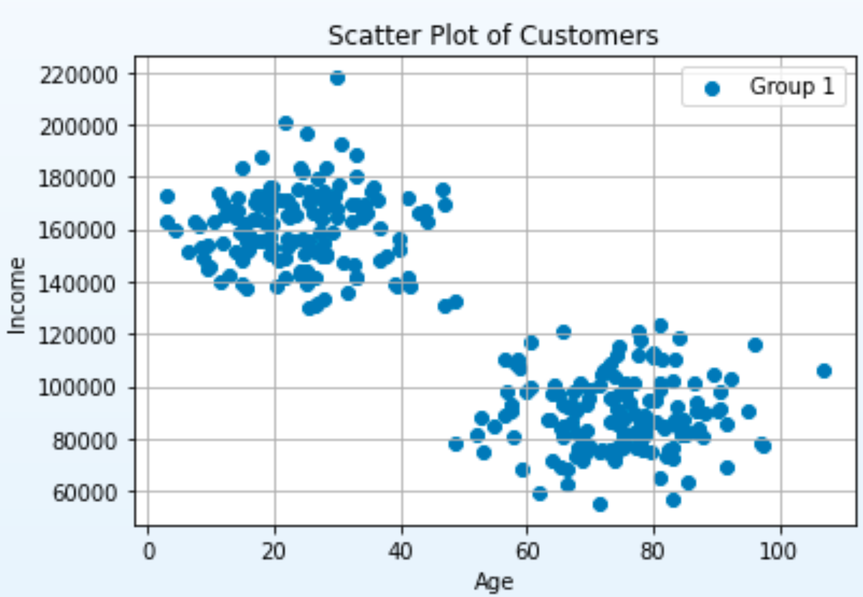
An Example : Health Company

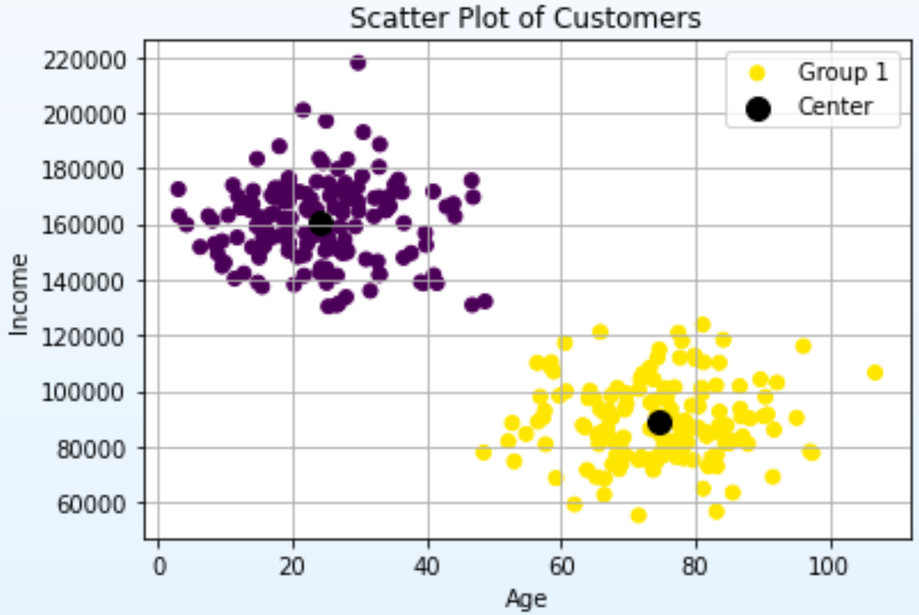
Discrete unsupervised learning can statistically tell us the different groups of data points



An Example : Health Company

Discrete unsupervised learning can statistically tell us the different groups of data points

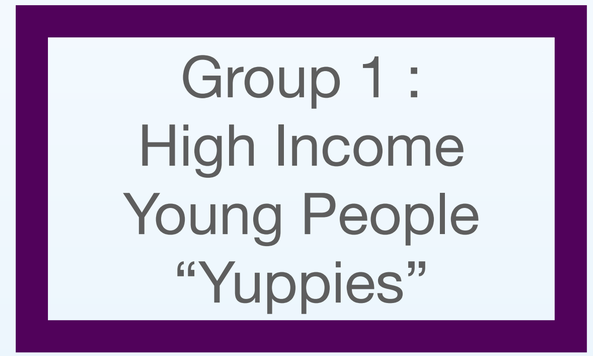


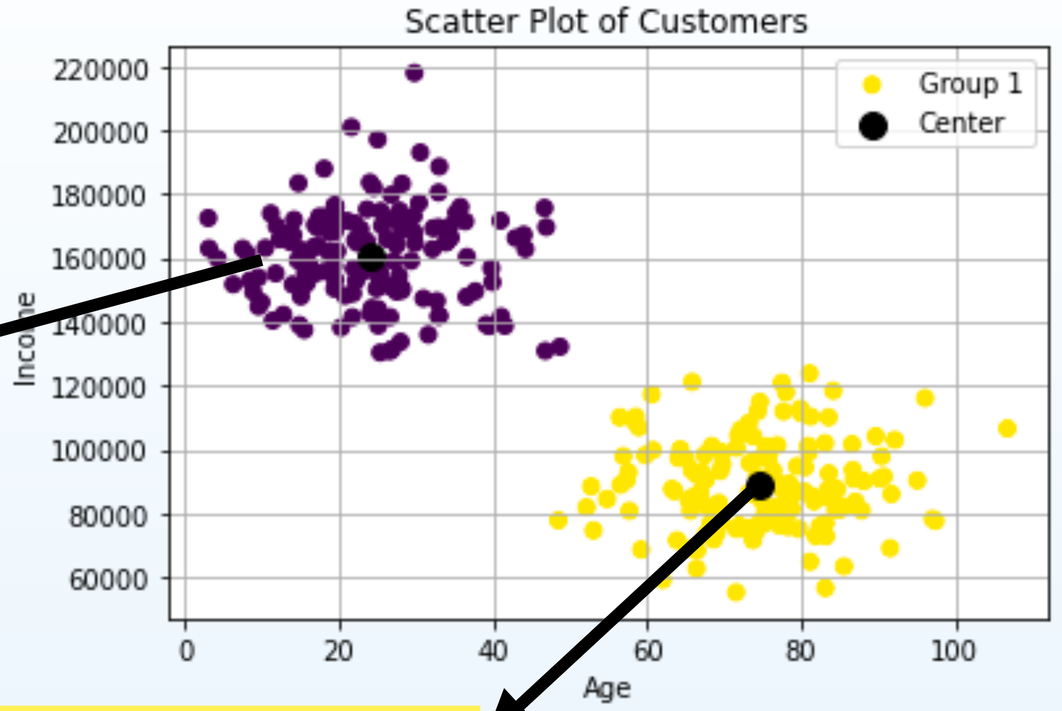


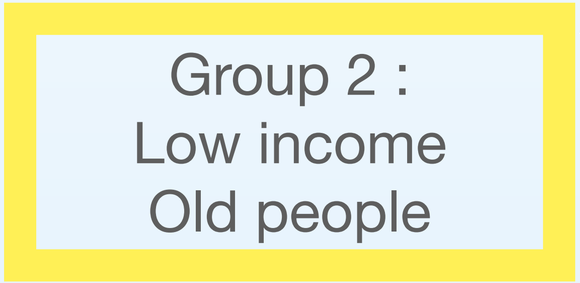
Example Interpretation

With a small number of features, there easy interpretation

Info can be used in marketing







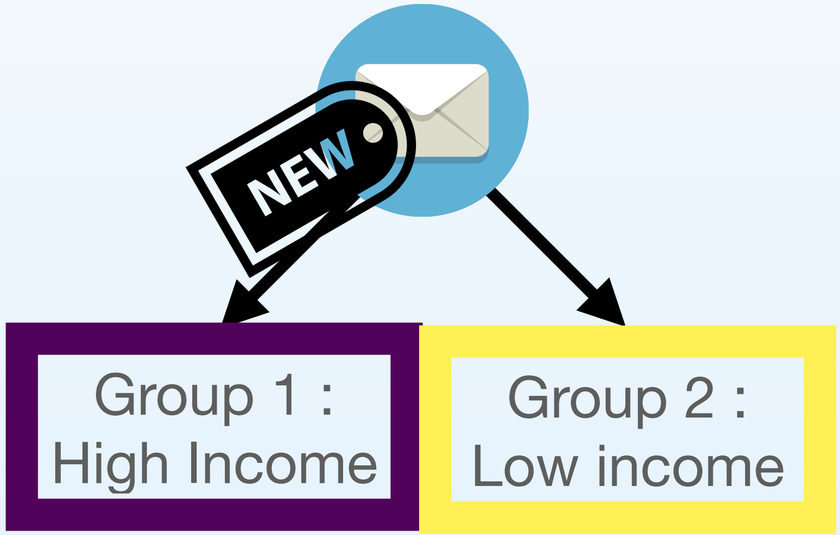
Example Actionable Insight

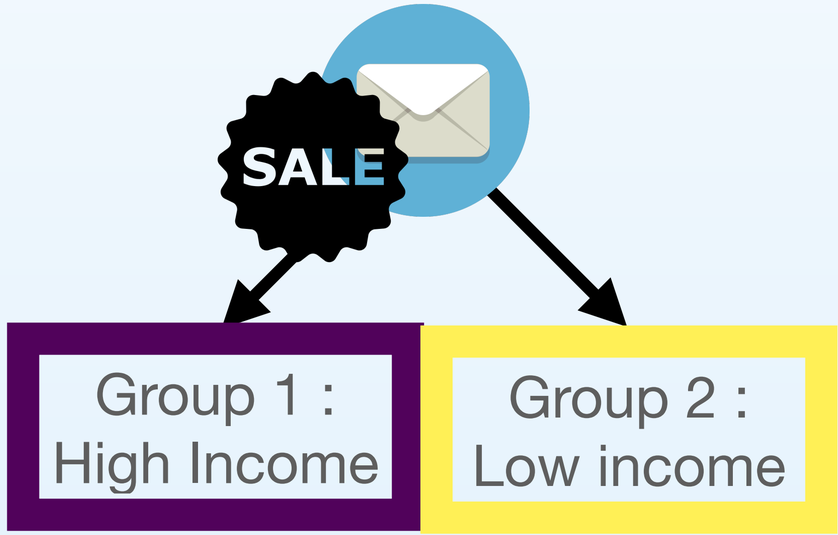
Let’s say there are two email marketing campaigns run to all of the clients

The two campaigns are :

1.) For a New product

2.) For a sale product





Example Actionable Insight

Unsupervised learning helped us identify distinct groups that will allow us to bucket future clients and know their behavior

	Group 1 : High Income	Group 2 : Low income
Open Rate	80%	15%
Unsubscribe Rate	1%	10%

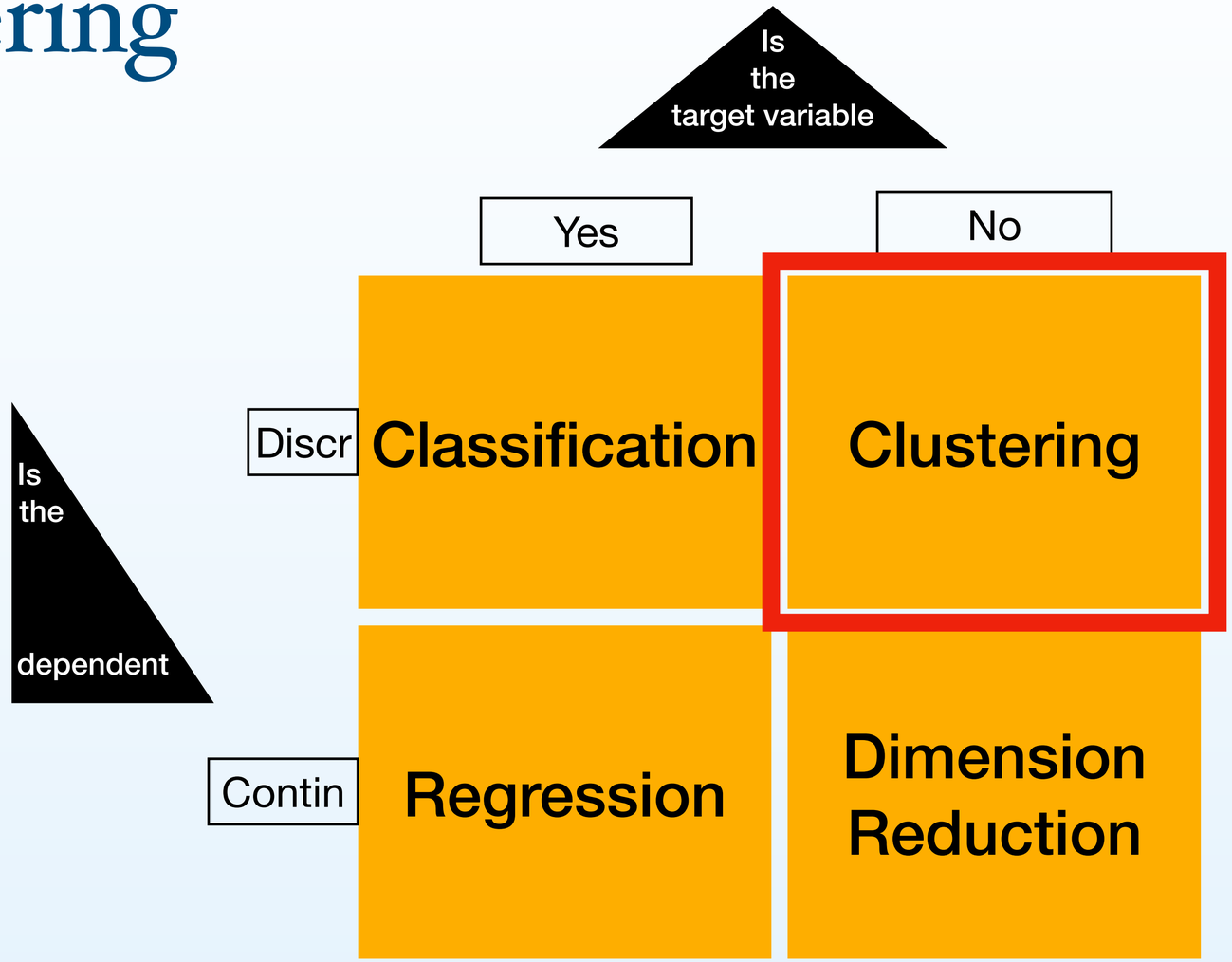
	Group 1 : High Income	Group 2 : Low income
Open Rate	10%	45%
Unsubscribe Rate	25%	2%

Clustering

Introduction to Clustering

Definition : ML technique that involves grouping together similar objects or data points based on attributes or similarity measures

Goal : Discover patterns or structure in data without use of labels



Types of Clustering

1.) K-Means
2.) Hierarchical Clustering
3.) Density Based Clustering
4.) Spectral Clustering
5.) Affinity Propagation
6.) Agglomerative Clustering
7.) Gaussian Mixture Models
8.) Self-Organizing Maps
9.) Fuzzy Clustering
10.) Subspace Clustering

K-means Algorithm

1.) Randomly Assign “Centroids”

Centroids : Center of the cluster of points

2.) Create regions of points that belong to each centroid

3.) Move the centroid to the mean position of those points

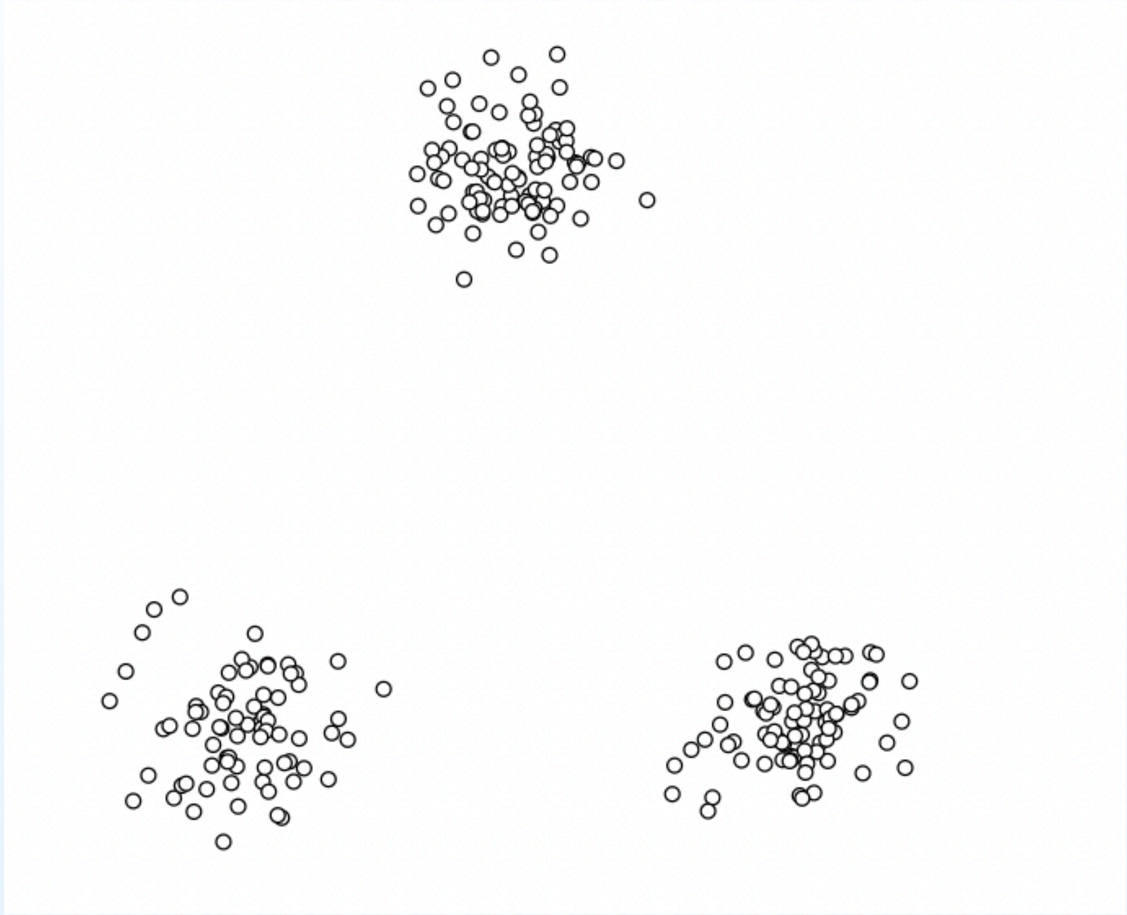
4.) Repeat 2/3 until no points change in the reassign

K-means Algorithm

Randomly assign centroid to the

x_i : is the individual point

n : total number of points



K-means Algorithm

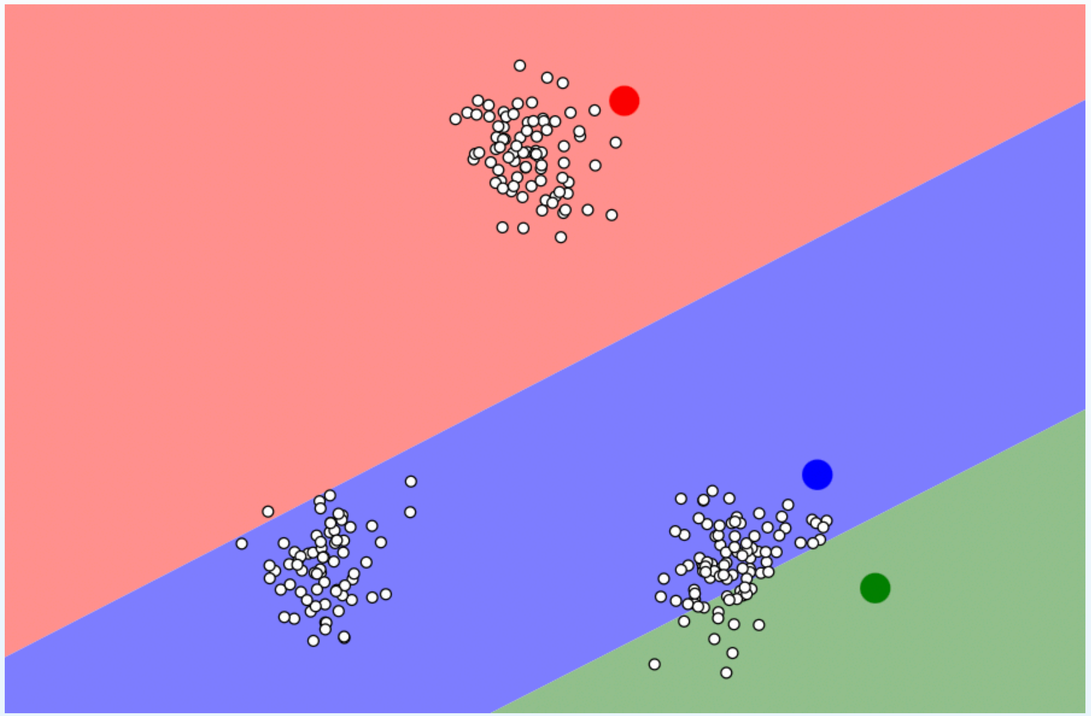
Randomly assign centroid to the points

C : is the set of cluster centers

μ_j : is the center of cluster

Calculate Decision Boundary

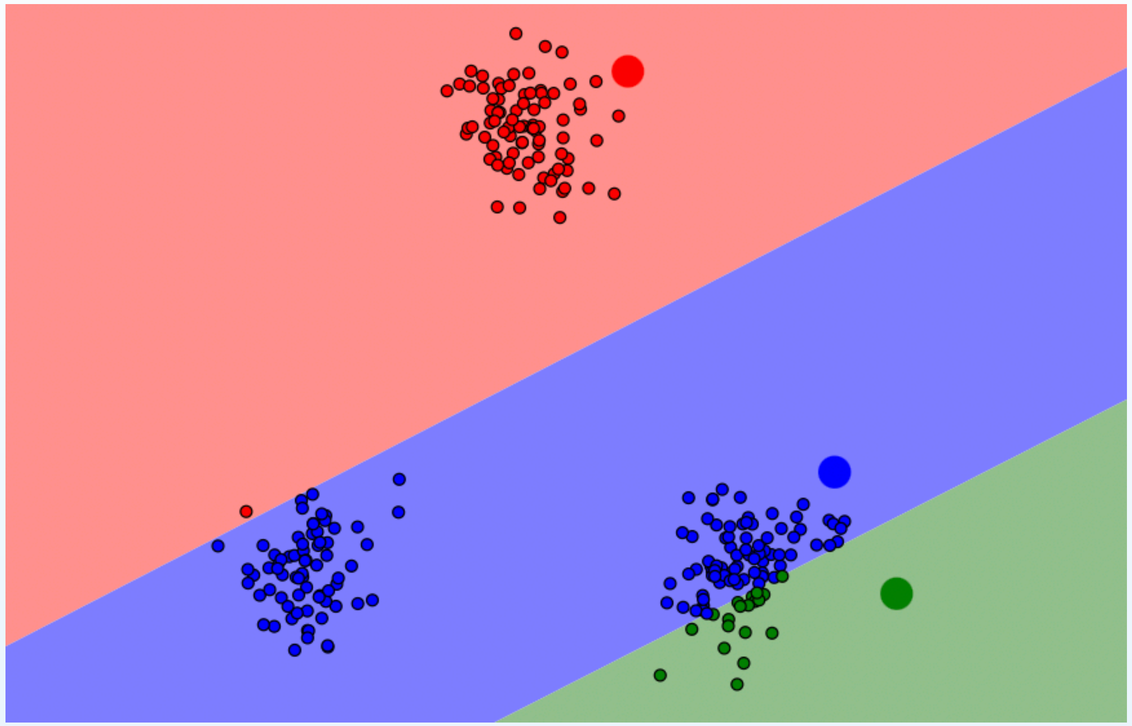
$$\min_{\mathbf{C}} \sum_{i=1}^{n} \min_{\mu_j \in \mathbf{C}} \left| \mathbf{x}_i - \mu_j \right|^2$$



K-means Algorithm

Points are assigned to a centroid

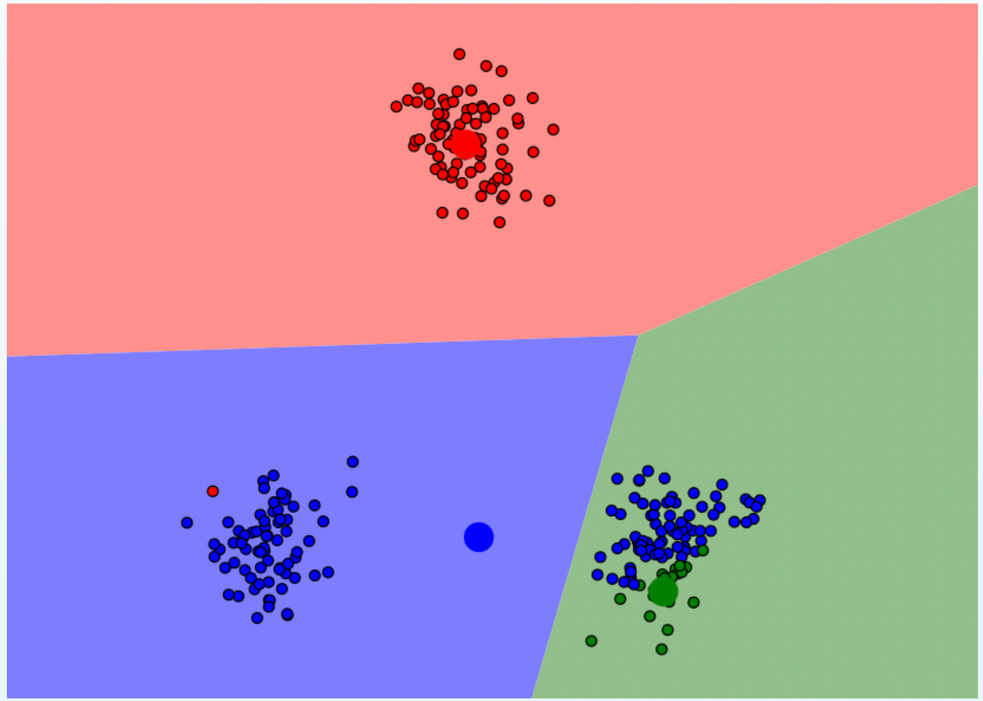
x_i^j : each point s assigned to cluster j



K-means Algorithm

Centroids are moved to the mean of their assigned points

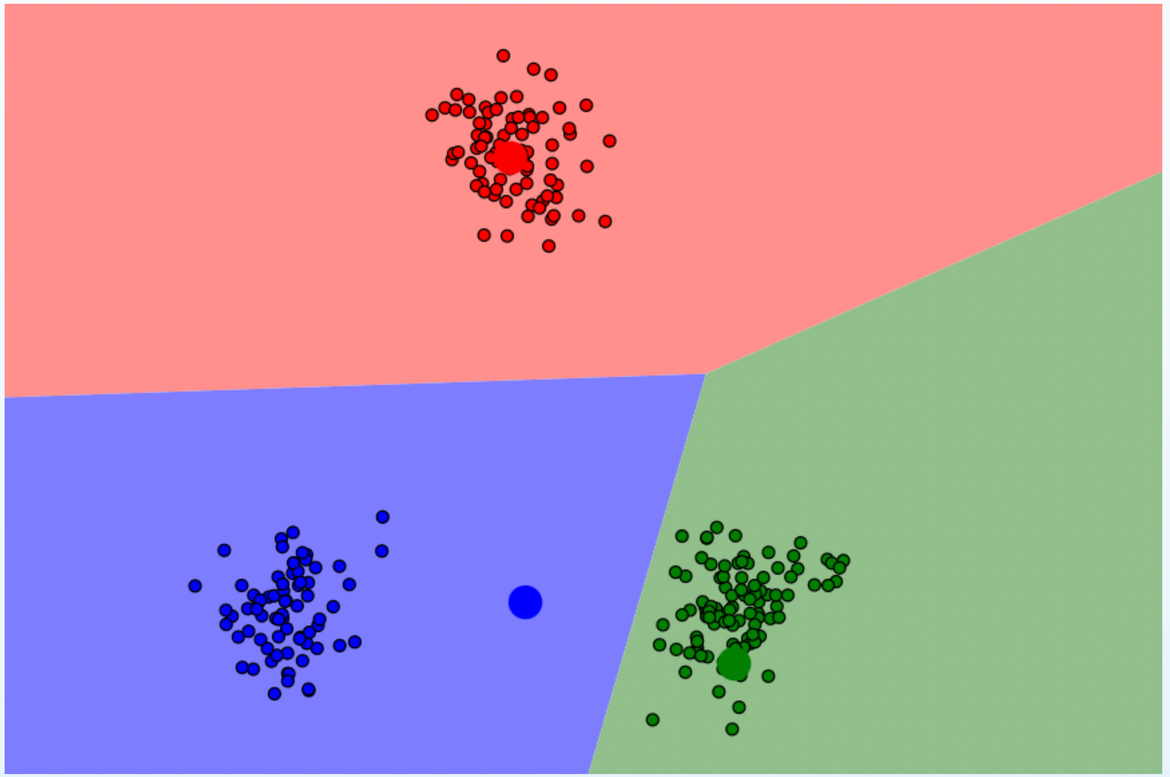
$$\mu_j := \frac{\sum_{i=1}^{n} \mathbf{x}_i^j}{|X^j|}$$



K-means Algorithm

Recreate Decision Boundary

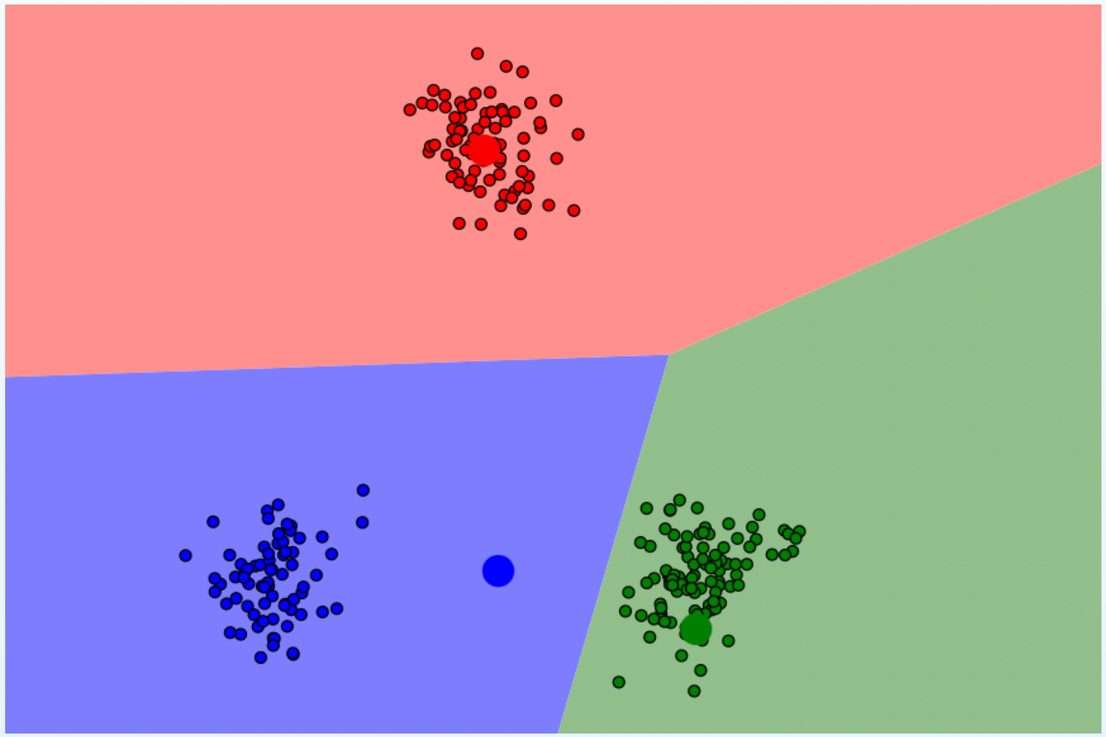
$$\min_{\mathbf{C}} \sum_{i=1}^{n} \min_{\mu_j \in \mathbf{C}} \left| \mathbf{x}_i - \mu_j \right|^2$$

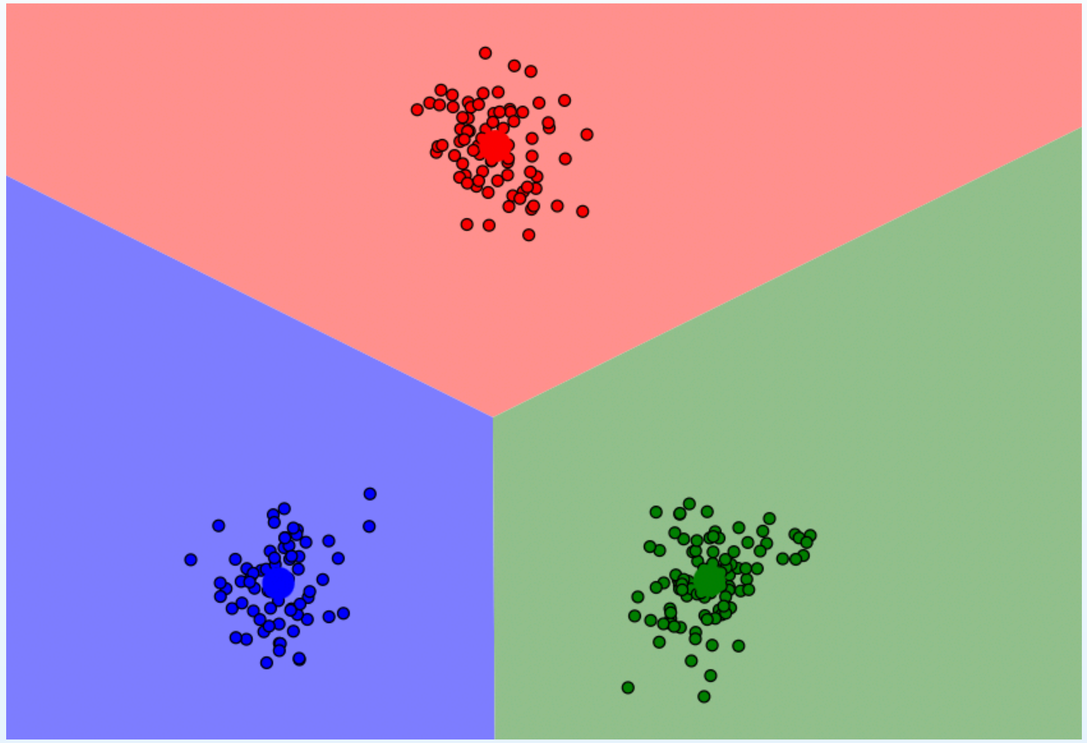


K-means Algorithm

Repeat

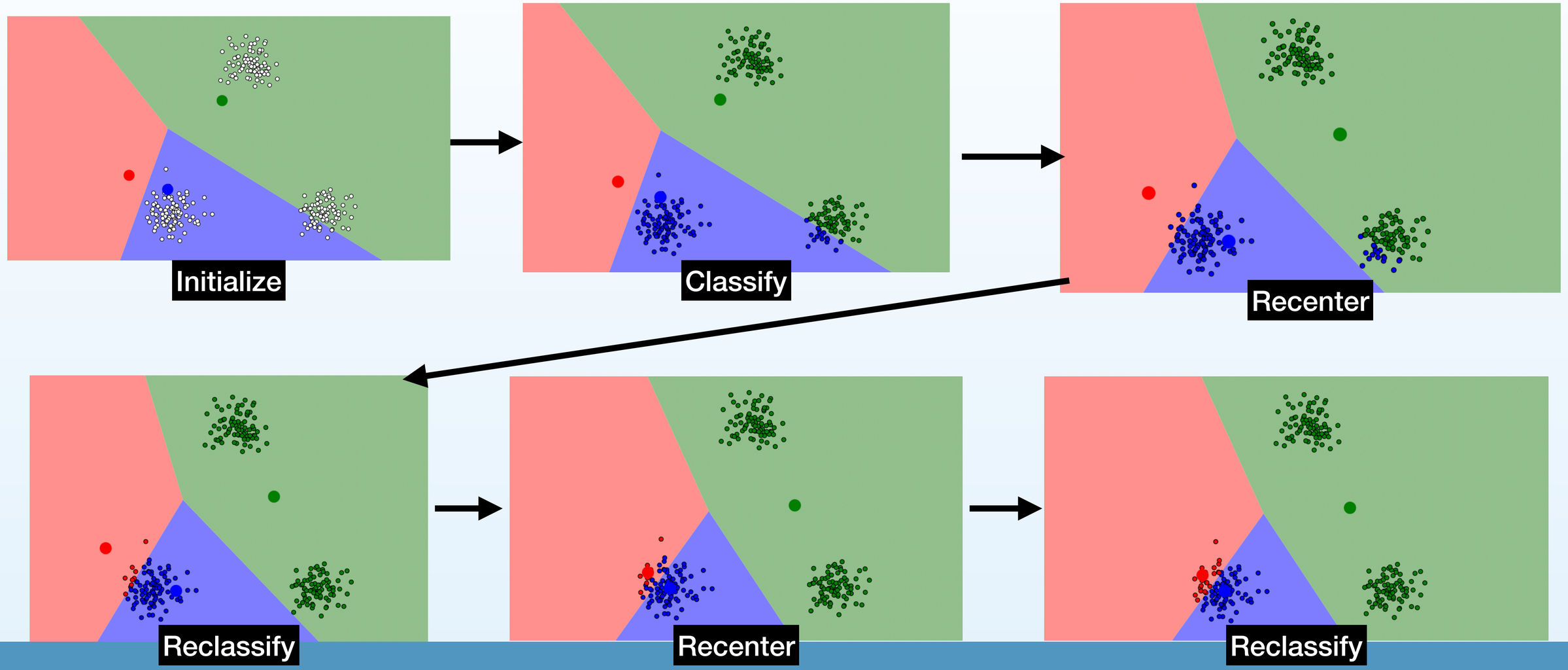
$$\mu_j := \frac{\sum_{i=1}^{n} \mathbf{x}_i^j}{|X^j|}$$





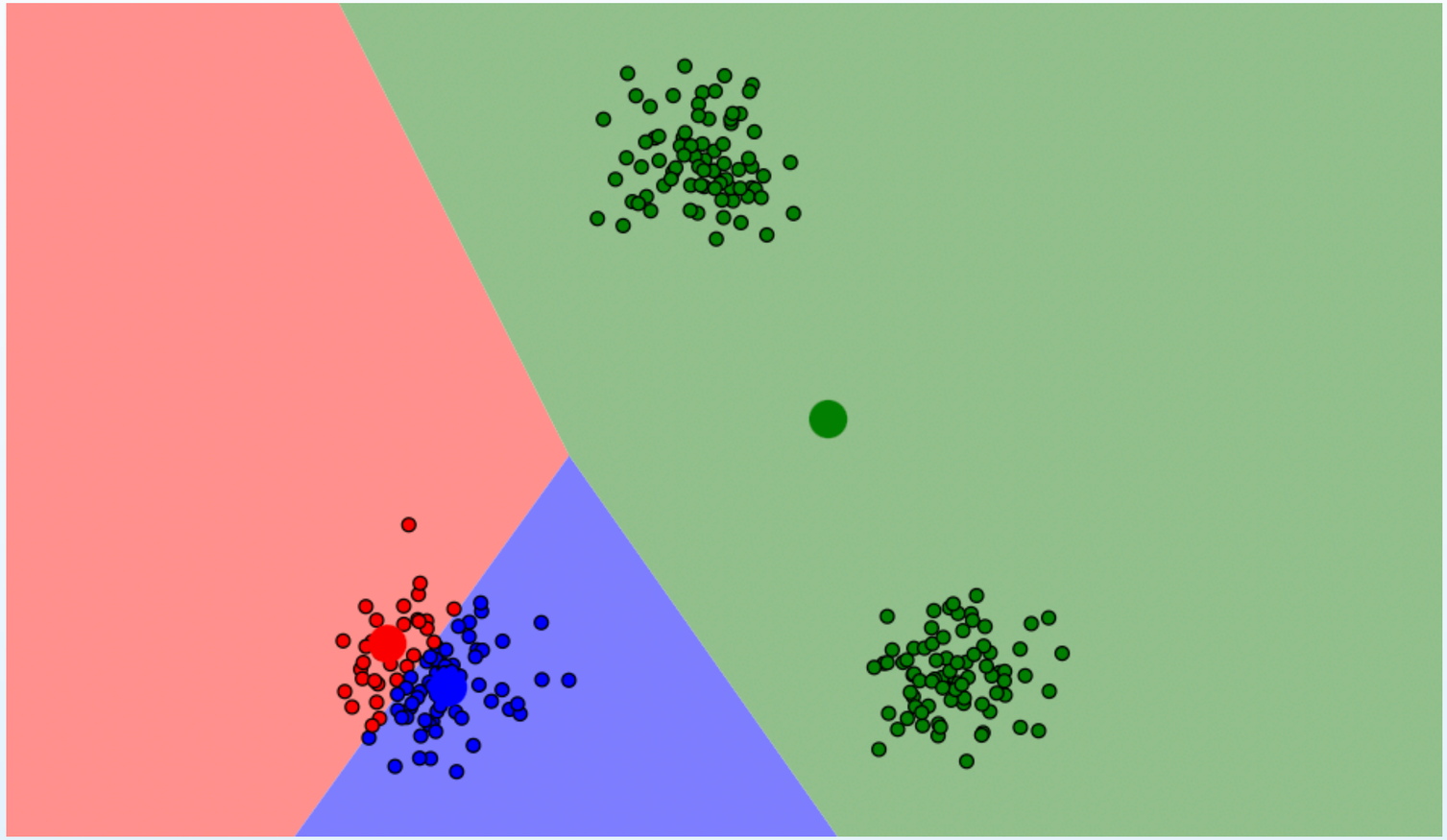
K-means Algorithm

Where to initialize the centroids?
The outcome can be sensitive to initial conidtions



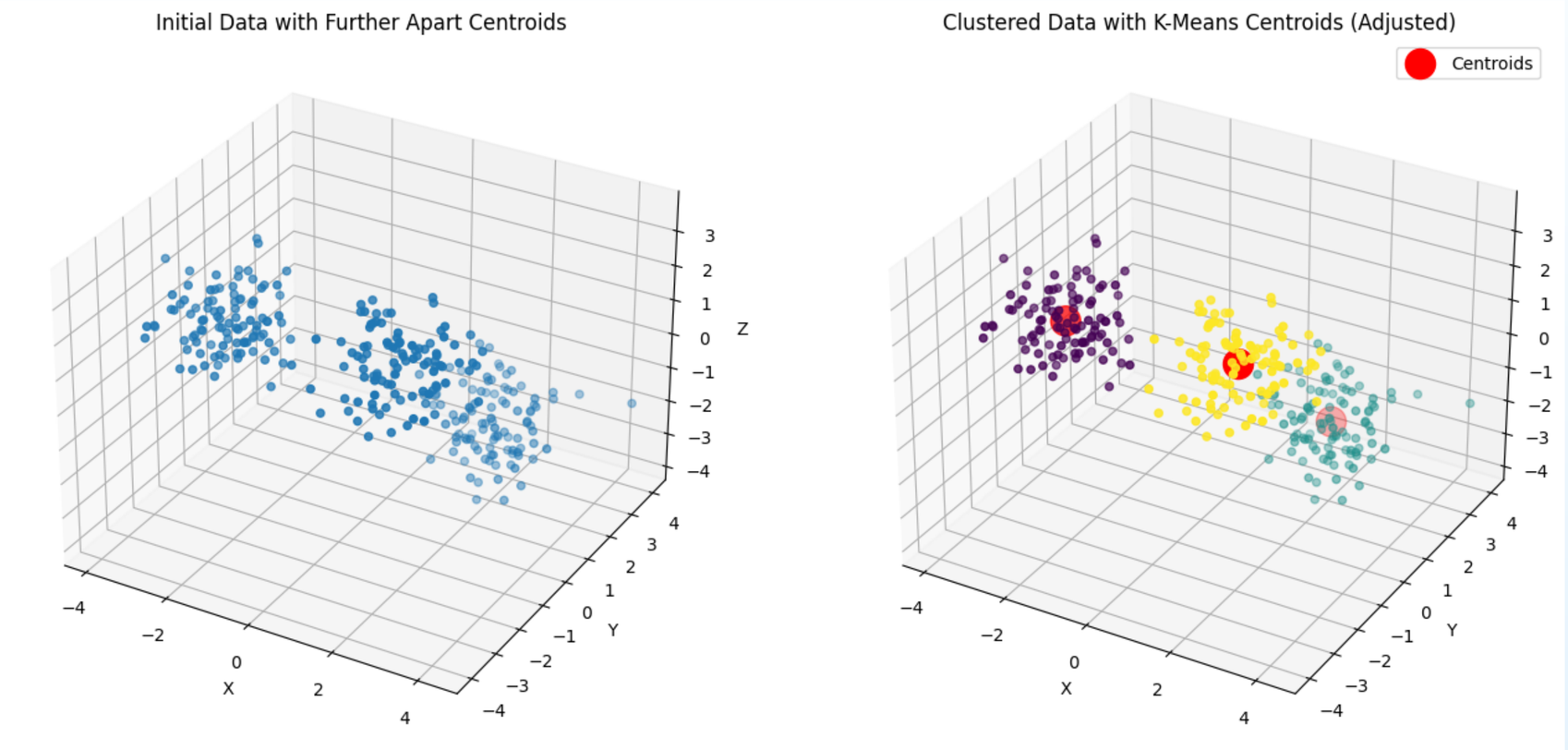
K-means Algorithm

Final Solution of K-Means



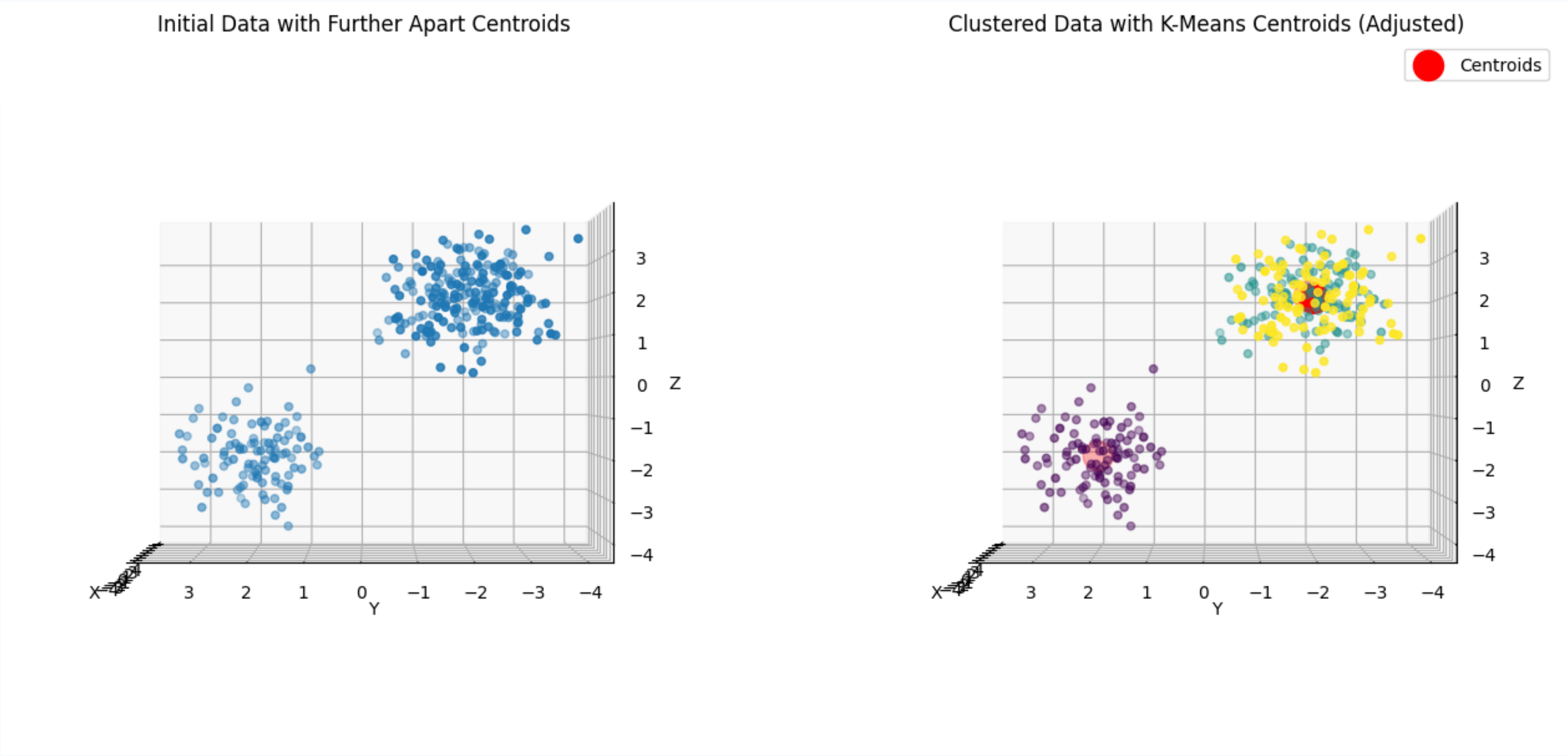
In higher dimensions the same algorithm is repeated

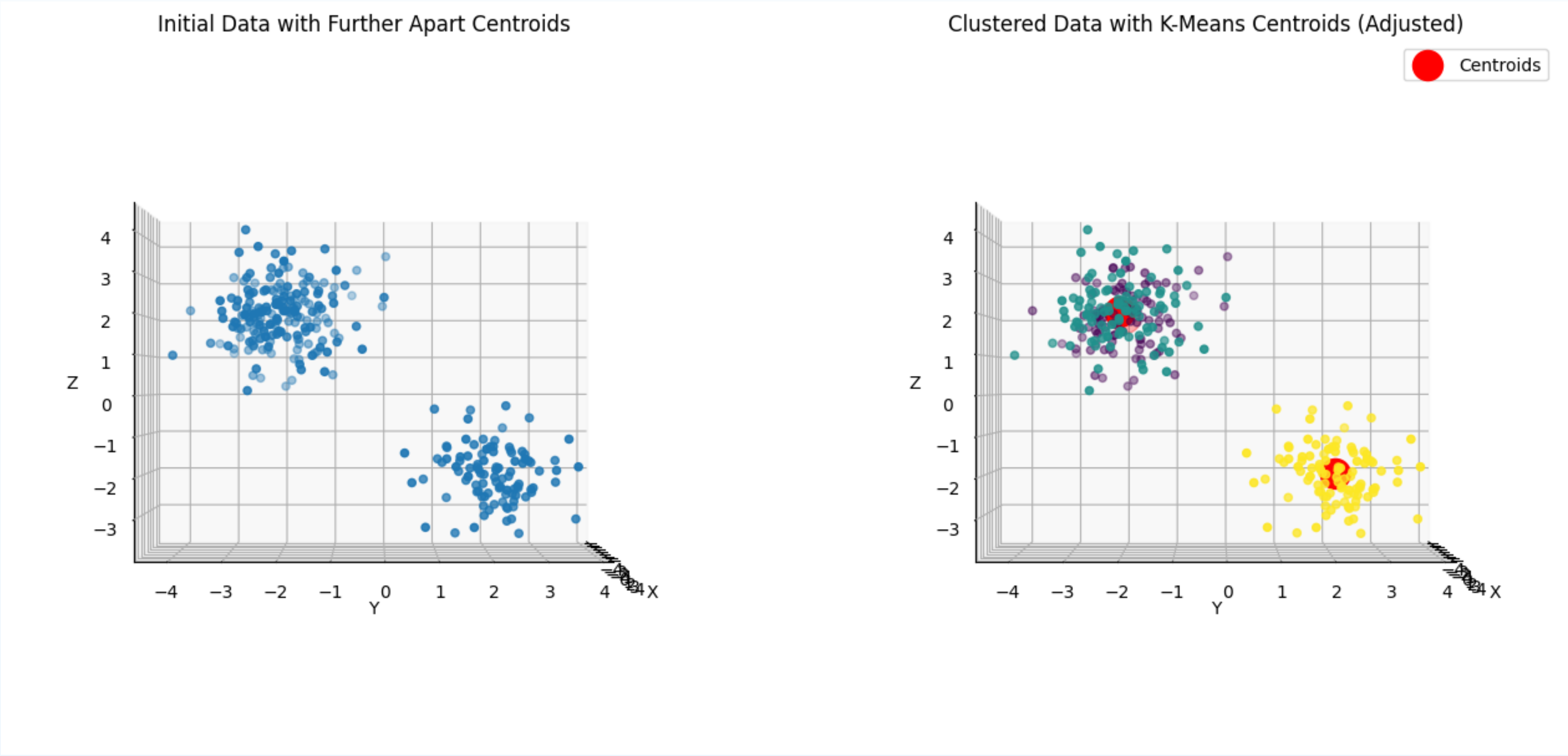
3 Dimensions

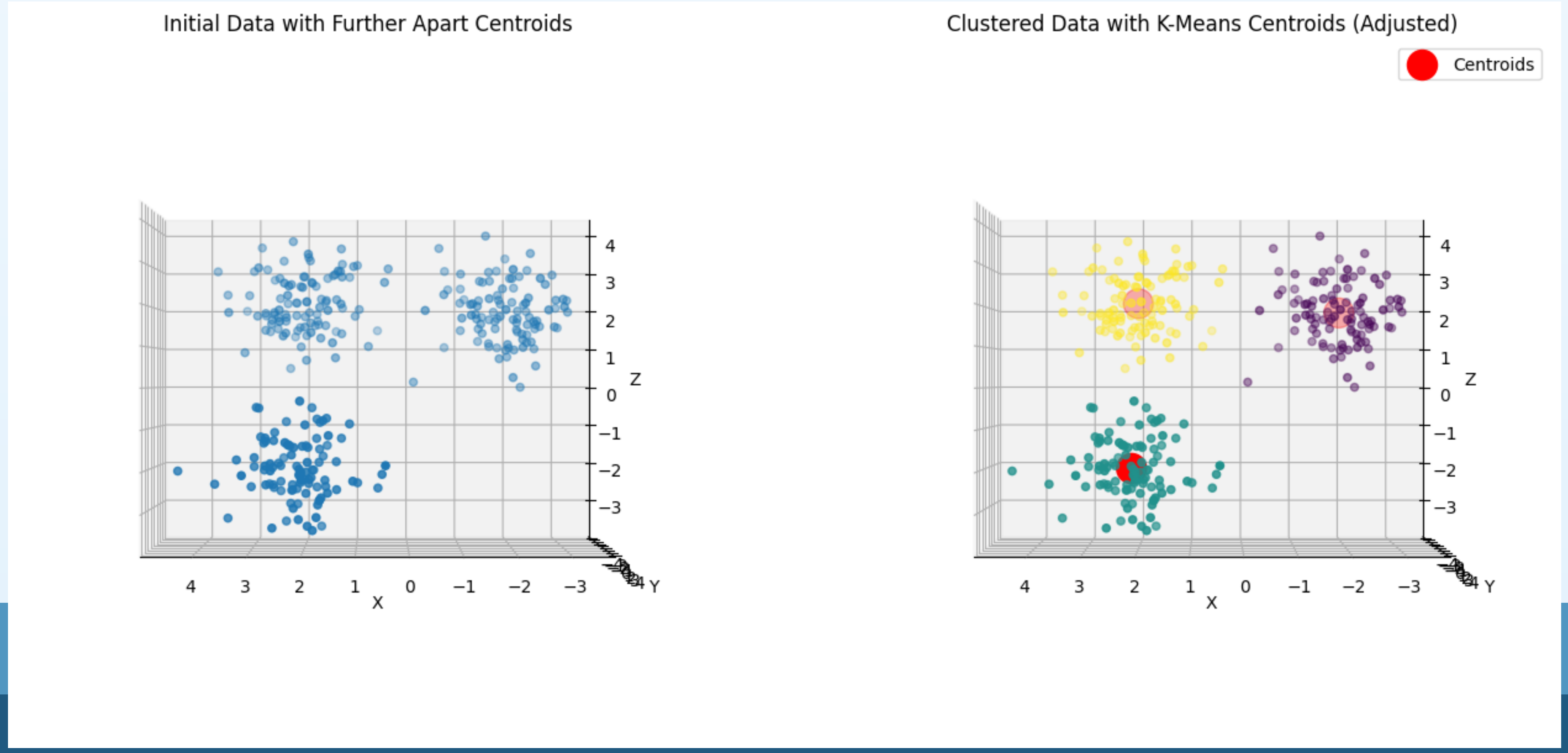


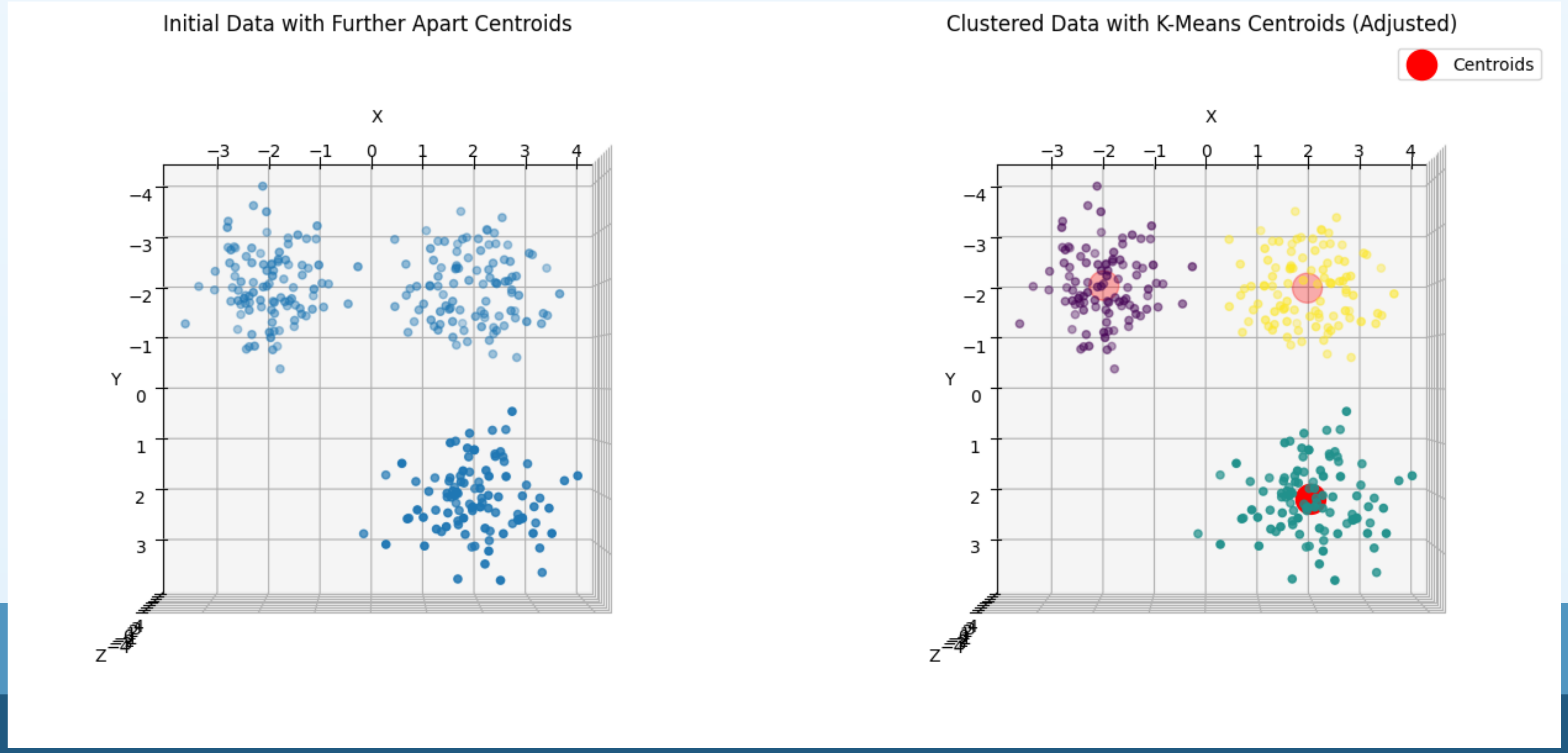
Higher dimensions is where clustering becomes practical

Humans are not good at conceptualizing higher dimensions. Clustering helps









K-means Algorithm

Where to initialize the centroids?

Randomly initialize centroids, run the algorithm and pick the final model with the lowest SSE

$$\text{SSE} = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i,j)} \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(j)}\|_2^2$$

```
KMeans(n_init = )
```

n_init is the parameter for the number of random centroid starts

K-means Algorithm

How many clusters to use?

1.) Elbow Method - Algorithm
2.) Silhouette Method - Algorithm
3.) Gap Statistic - Algorithm
4.) Domain Knowledge - Expert Intuition

K-means Algorithm

Elbow Method

Identify a pivot point where the model complexity does not help accuracy as much

K-means Algorithm

Elbow Method

WCSS (With-in Cluster Sum of Squares

$$\text{WCSS} = \sum_{C_k}^{C_n} \left(\sum_{d_i \text{ in } C_i}^{d_m} distance(d_i, C_k)^2 \right)$$

Where,

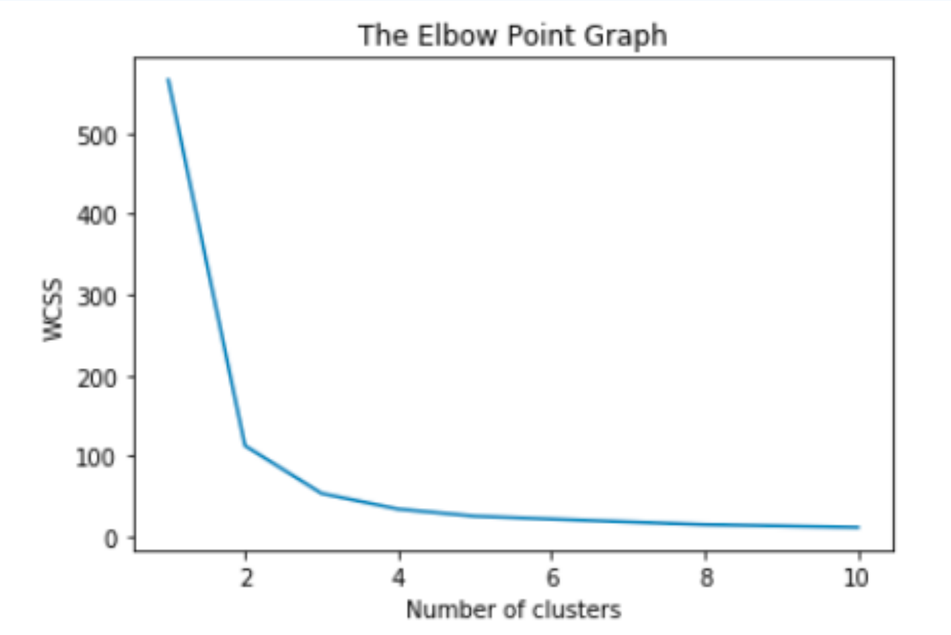
C is the cluster centroids and d is the data point in each Cluster.

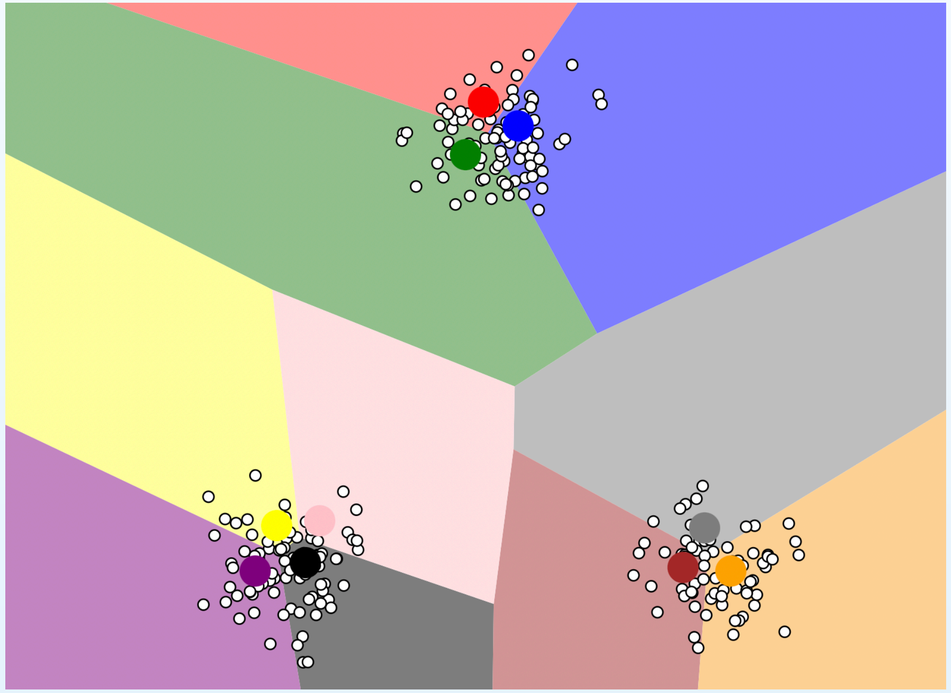
```
kmeans.inertia_
```

K-means Algorithm

Issue with WCSS as a metric :

It is always decreasing as clusters increase

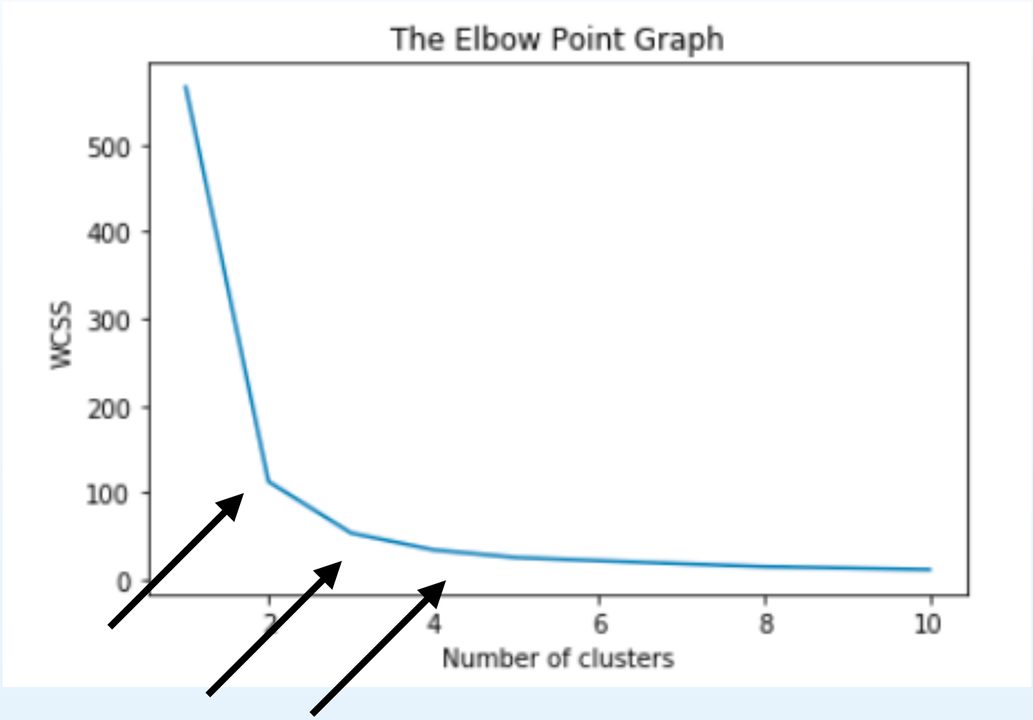




K-means Algorithm

Elbows occur when :
The drop in WCSS drops by not as much

There is no definite or objective way to determine where the elbow point is



Silhouette Score

$$s = \frac{b - a}{max(a, b)}$$

a (Seperation)

b (Cohesion)

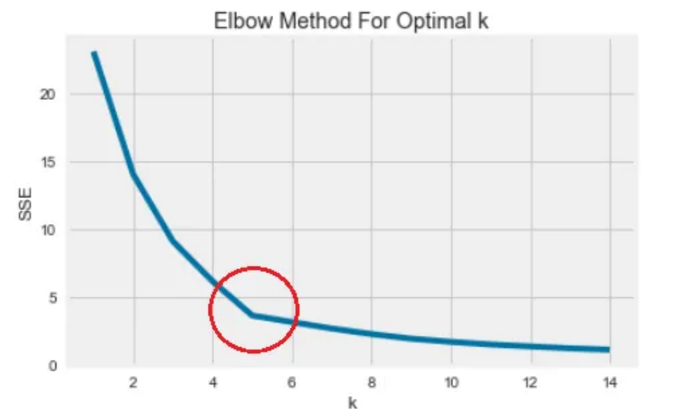
a : is the mean distance between a sample and all other points in the same class.

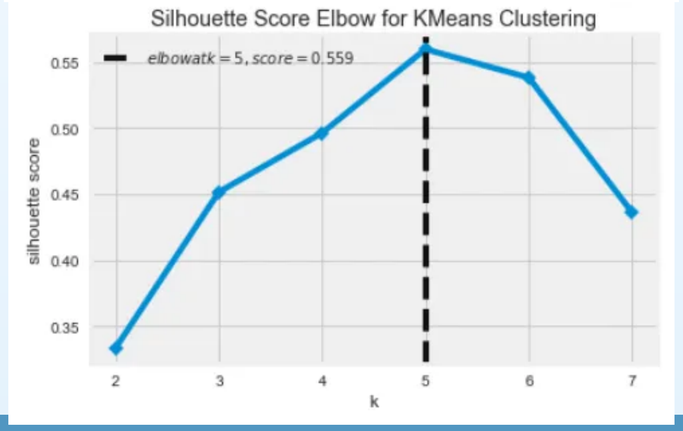
b : is the mean distance between a sample and all other points in the next nearest cluster.

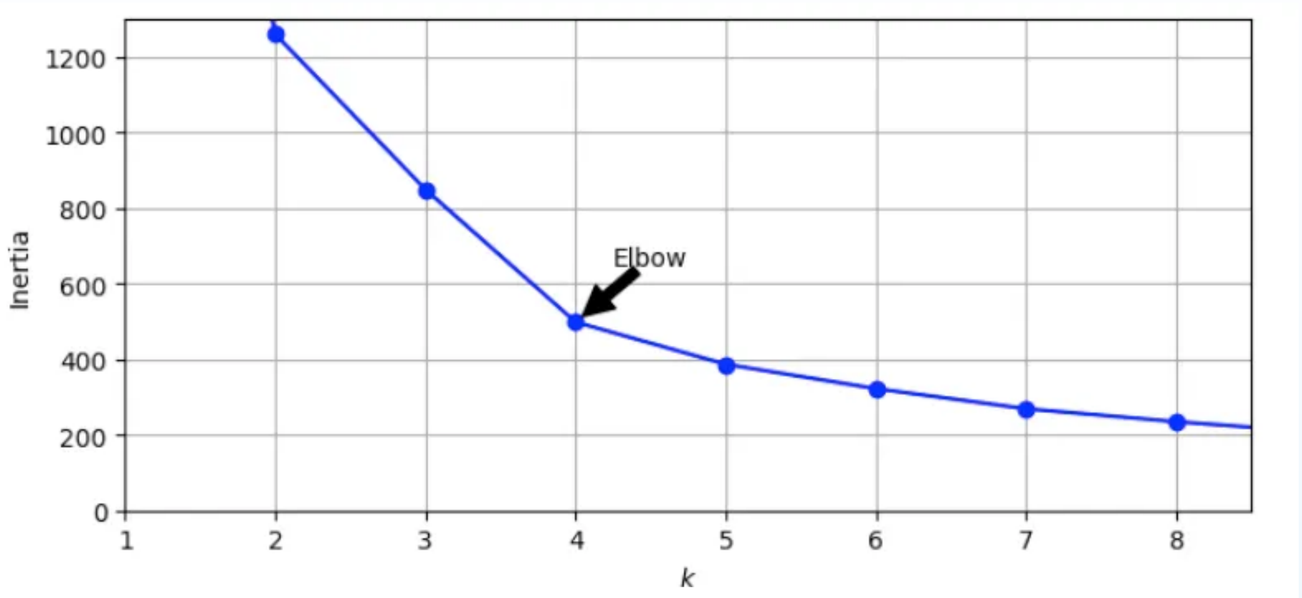
Silhouette Score

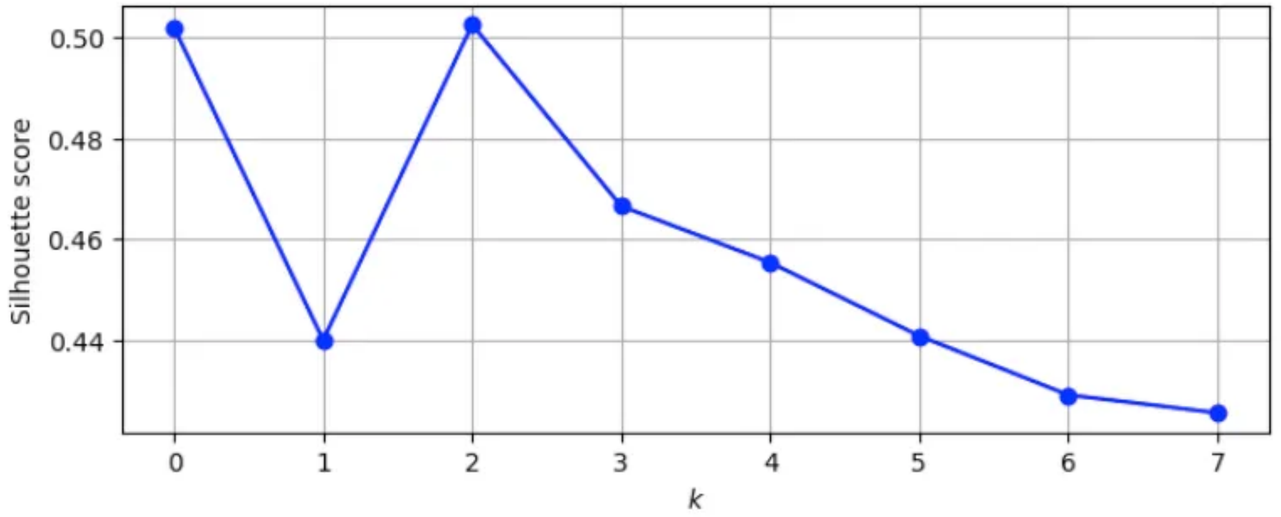
Typically align with Elbow analysis

Not always though









Clustering Preprocessing

Scale your Data!

- Use min/max scaler, normalization or log transform

Feature Creation, Extraction and Selection.

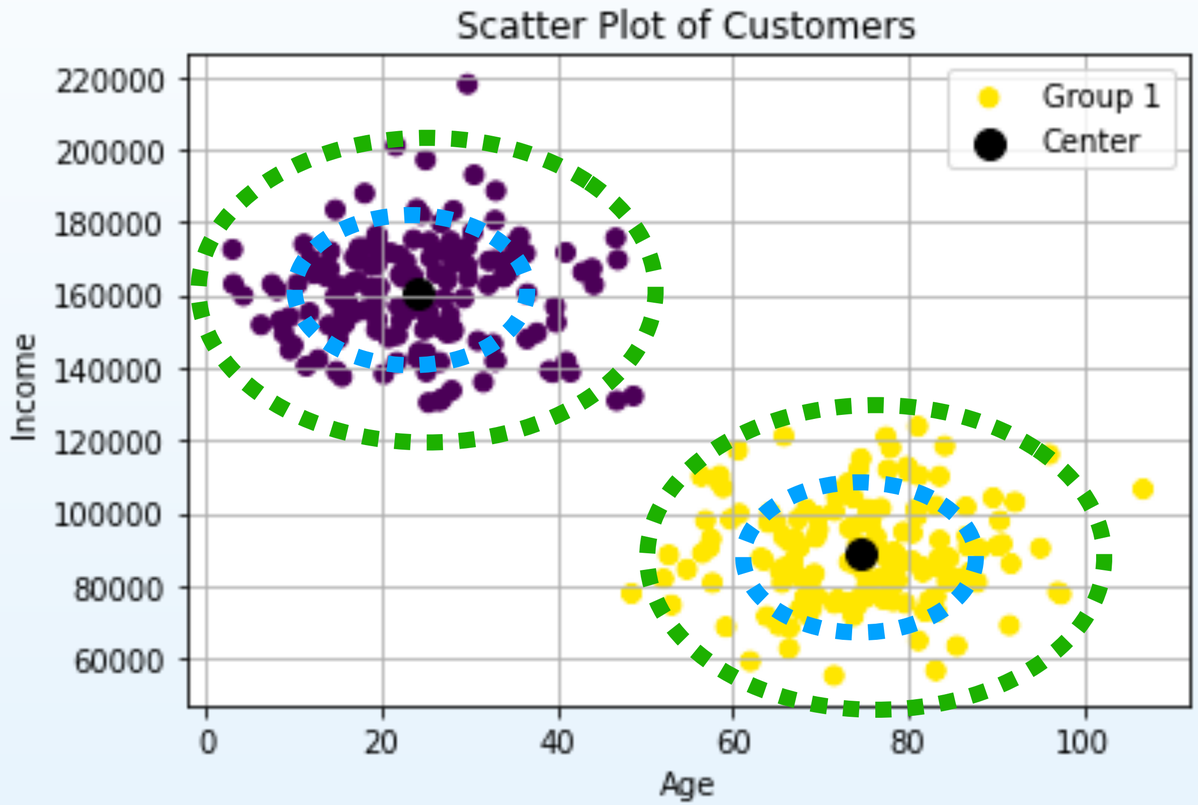
- At this stage you have to use your intuition to create the relevant features

Use Case : “Distance-Based Anomaly Detection”

Use a threshold “Z” that is some multiple of a standard deviation around a centroid.

It is an anomaly if :

$$\sqrt{(x_i^j - \mu^j)^2} > Z \cdot \sigma(x_i^j - \mu^j)$$



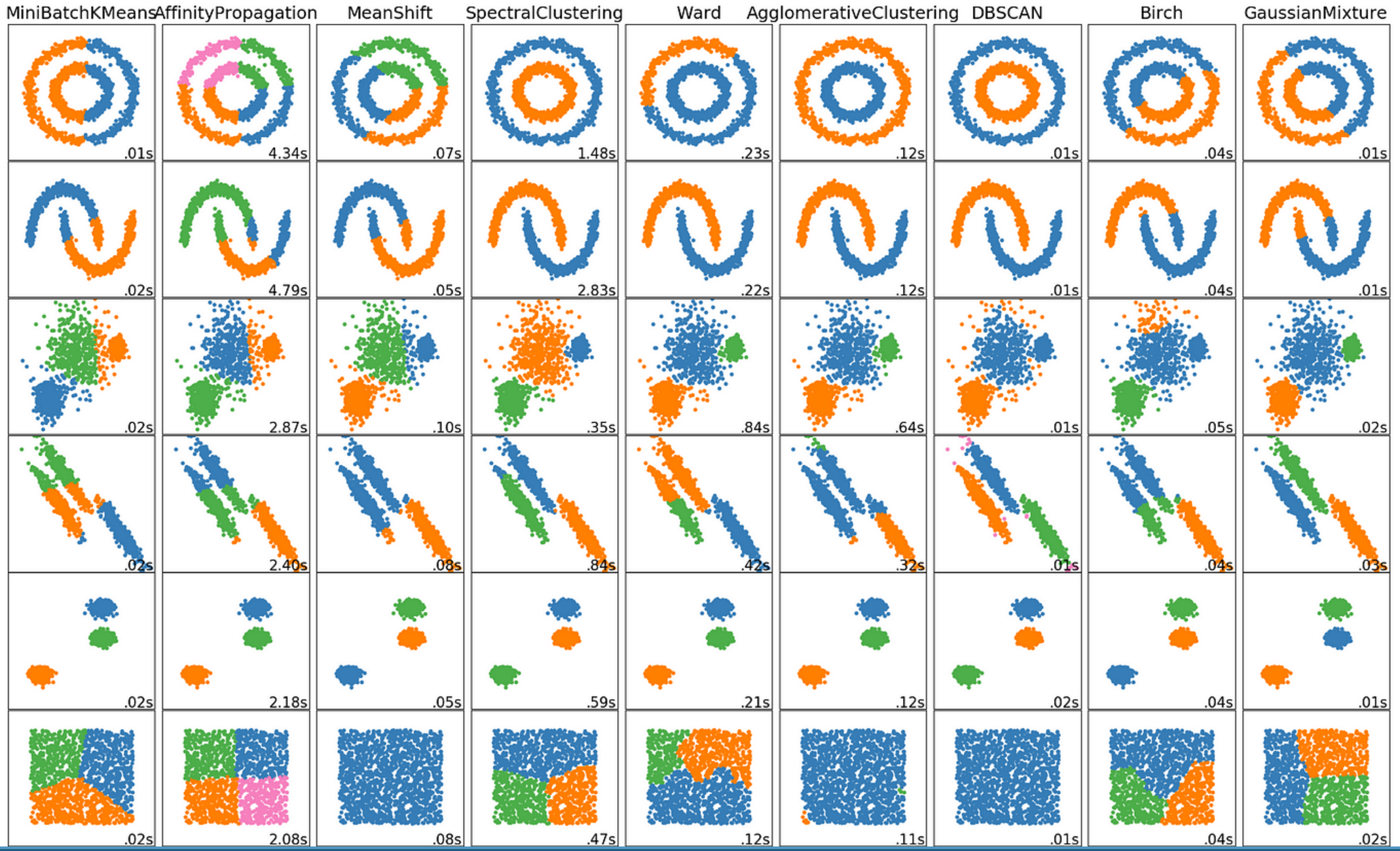
Z=1

Z=2

K-means is really just the starting point of Clustering

It is a very large field with many models and nuances

Suggested Reading
https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68


MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
Birch
GaussianMixture
.01s
4.34s
.07s
1.48s
.23s
.12s
.01s
.04s
.01s
.02s
4.79s
.05s
2.83s
.22s
.12s
.01s
.04s
.01s
.02s
2.87s
.10s
.35s
.84s
.64s
.01s
.05s
.02s
.02s
2.40s
.08s
.84s
.42s
.32s
.01s
.04s
.03s
.02s
2.18s
.05s
.59s
.21s
.12s
.02s
.04s
.01s
.02s
2.08s
.08s
.47s
.12s
.11s
.01s
.04s
.02s

Coding

Coding

To Begin, you trying a Machine Learning Model the same as we've done

Step 1: Packages

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

Step 2: Preprocess data,
Feature selection
Feature Creation
Scaling

```
scaler = StandardScaler().fit(X)
X_scaled = scaler.transform(X)
```

Coding

To Begin, you trying a Machine Learning Model the same as we've done

Step 3: Pick a number of clusters and fit

```
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)
```

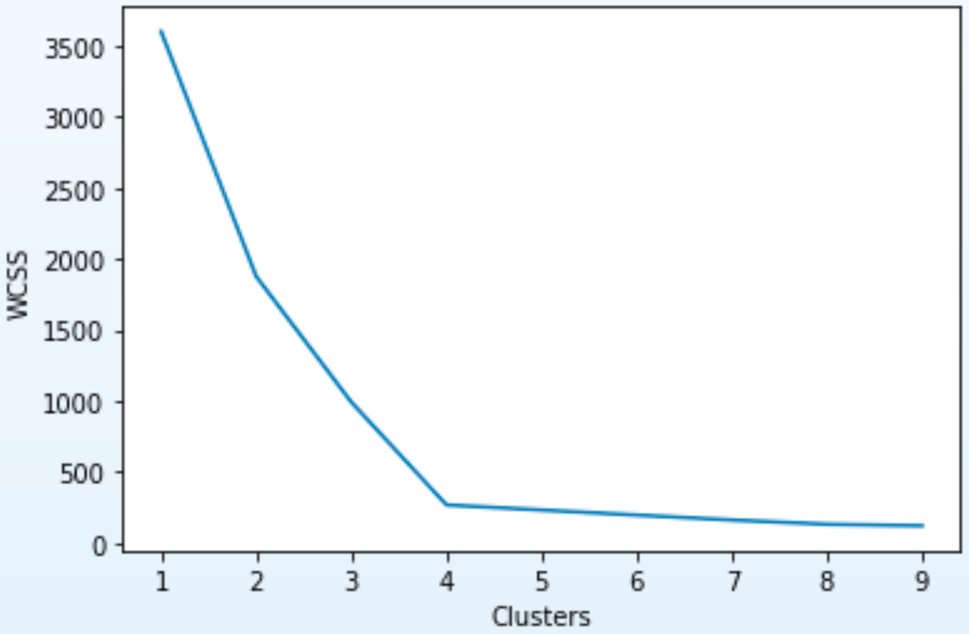
Step 4: Test a range of Ks

```
WCSSs = []
Ks = range(1,10)
for k in Ks:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(X_scaled)
    WCSSs.append(kmeans.inertia_)
```

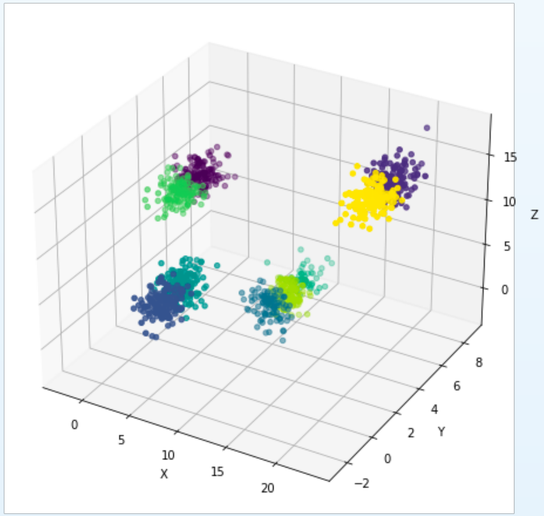
Coding

To Begin, you trying a Machine Learning Model the same as we've done

Step 5: Visualize and pick



Step 6: Refit and Visualize a few features



In - Class Assignment

Data

Country-data.csv

country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
Afghanistan	90.2	10	7.58	44.9	1610	9.44	56.2	5.82	553
Albania	16.6	28	6.55	48.6	9930	4.49	76.3	1.65	4090
Algeria	27.3	38.4	4.17	31.4	12900	16.1	76.5	2.89	4460
Angola	119	62.3	2.85	42.9	5900	22.4	60.1	6.16	3530
Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200
Argentina	14.5	18.9	8.1	16	18700	20.9	75.8	2.37	10300
Armenia	18.1	20.8	4.4	45.3	6700	7.77	73.3	1.69	3220
Australia	4.8	19.8	8.73	20.9	41400	1.16	82	1.93	51900
Austria	4.3	51.3	11	47.8	43200	0.873	80.5	1.44	46900
Azerbaijan	39.2	54.3	5.88	20.7	16000	13.8	69.1	1.92	5840
Bahamas	13.8	35	7.89	43.7	22900	-0.393	73.8	1.86	28000
Bahrain	8.6	69.5	4.97	50.9	41100	7.44	76	2.16	20700
Bangladesh	49.4	16	3.52	21.8	2440	7.14	70.4	2.33	758
Barbados	14.2	39.5	7.97	48.7	15300	0.321	76.7	1.78	16000
Belarus	5.5	51.4	5.61	64.5	16200	15.1	70.4	1.49	6030

Follow the instructions in the Jupyter Notebook

Reference

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/