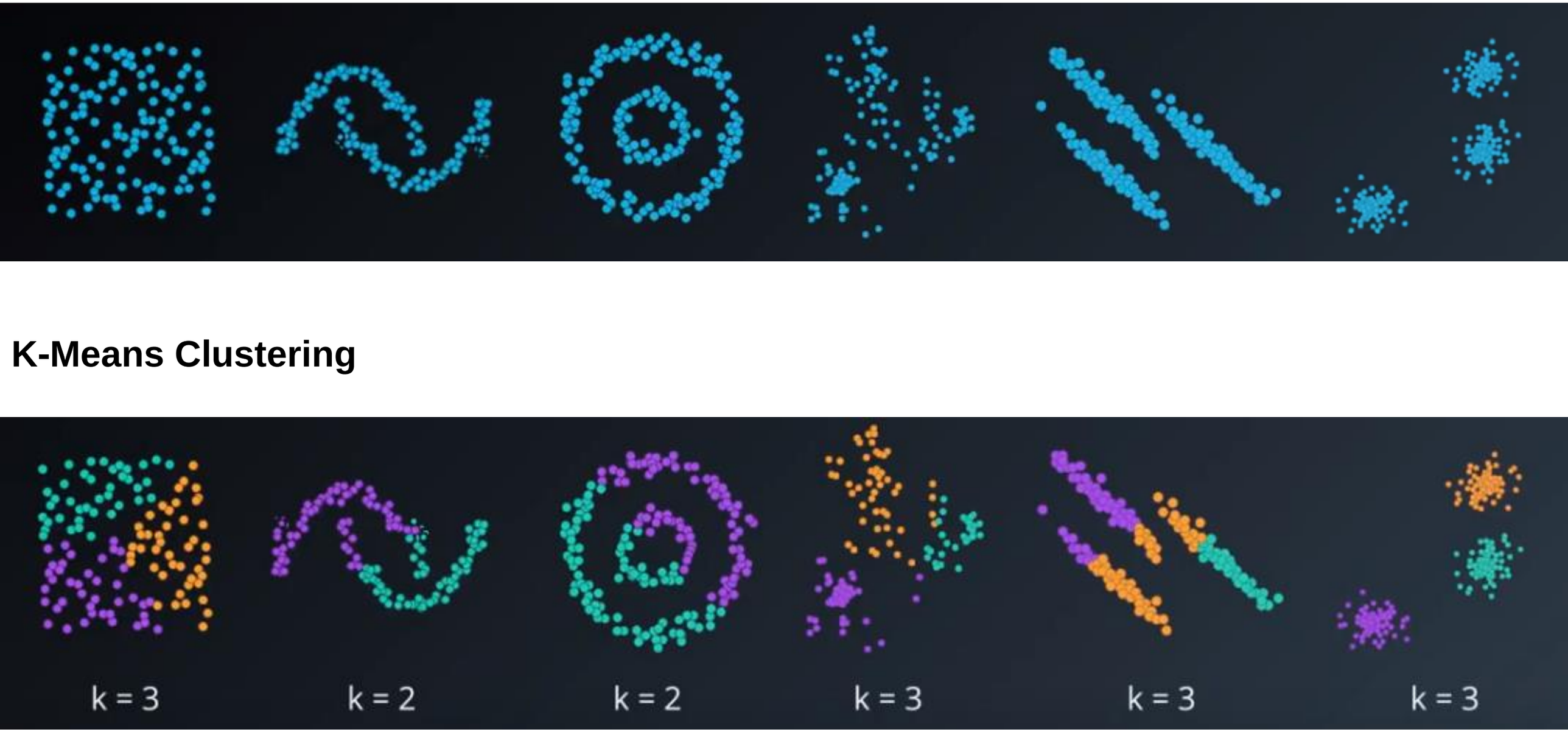
DICE ANALYTICS

DATA SCIENCE & MACHINE LEARNING COURSE

https://www.facebook.com/diceanalytics/
https://pk.linkedin.com/company/diceanalytics

K-Means Consideration

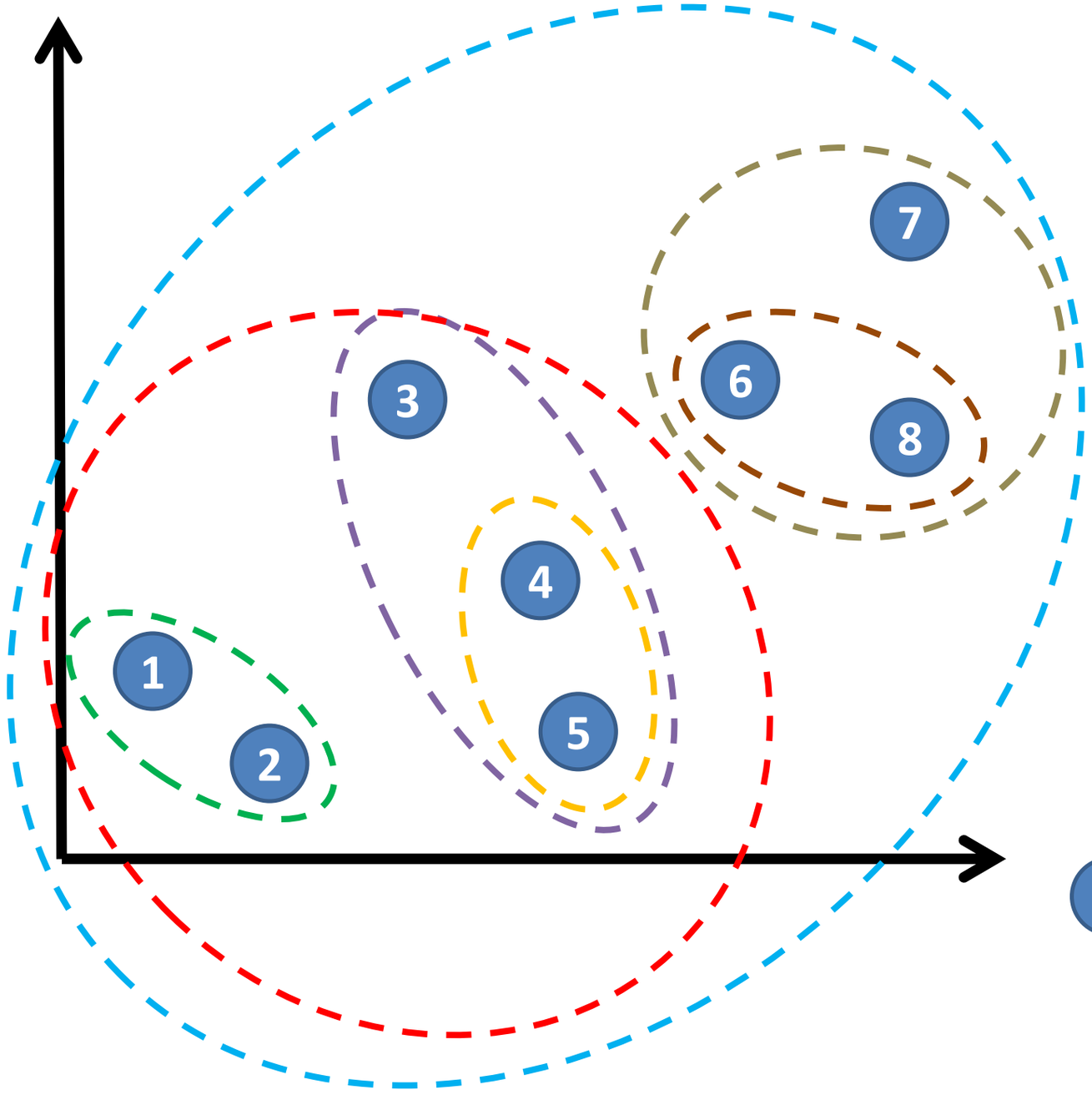
DATASETS

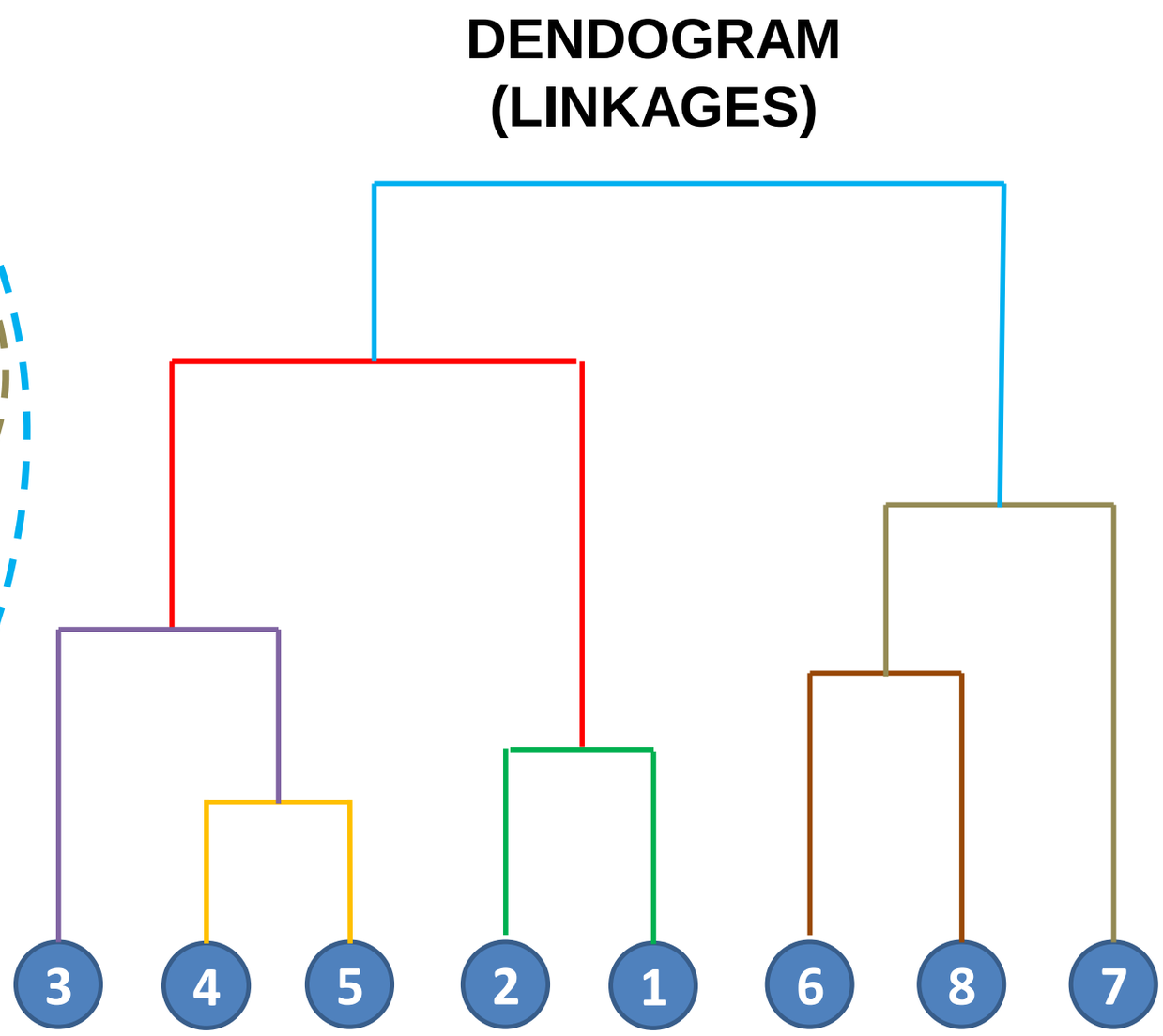


K-Means Clustering

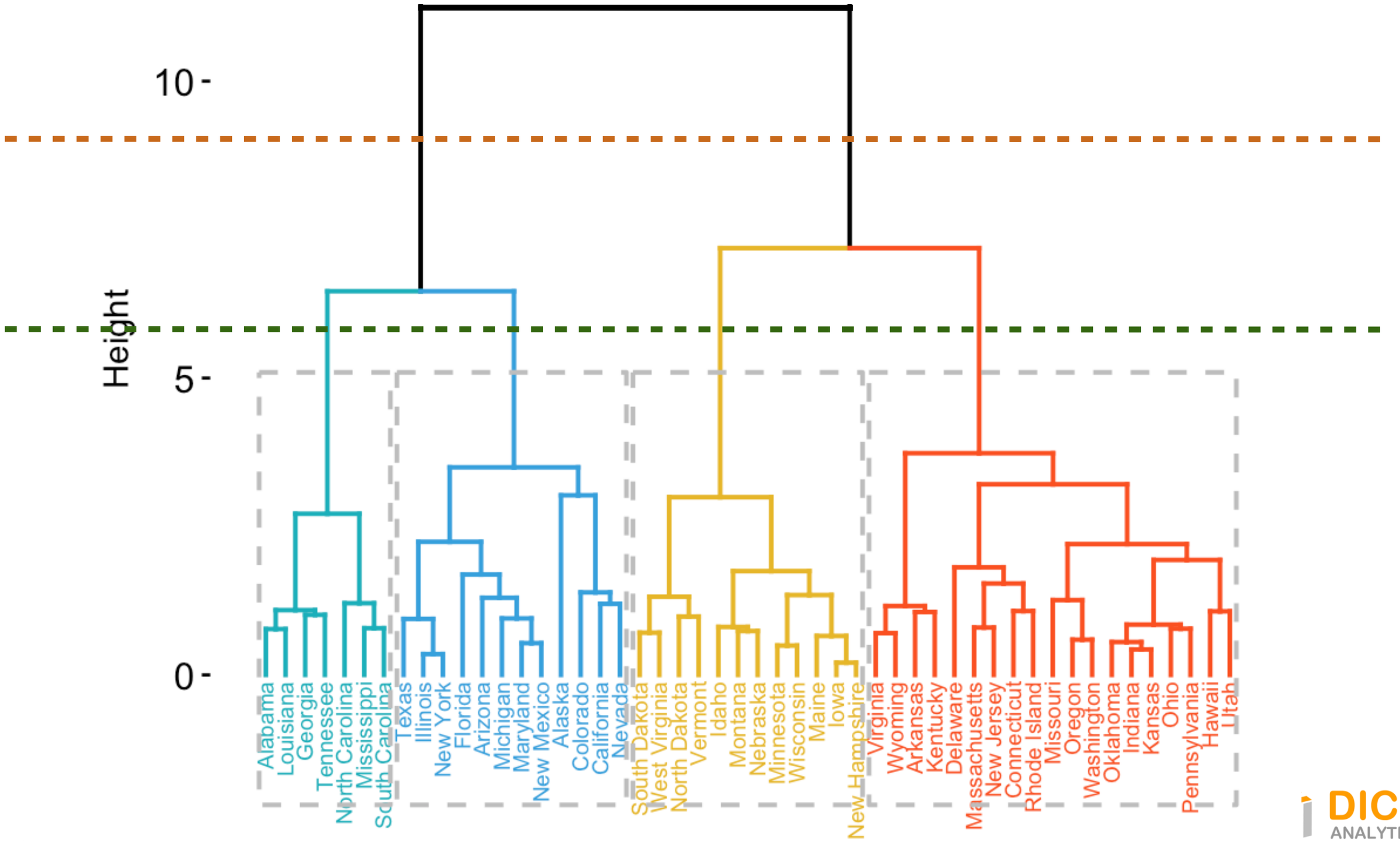
Source: Udacity Machine Learning Nano-Degree

Hierarchical Clustering

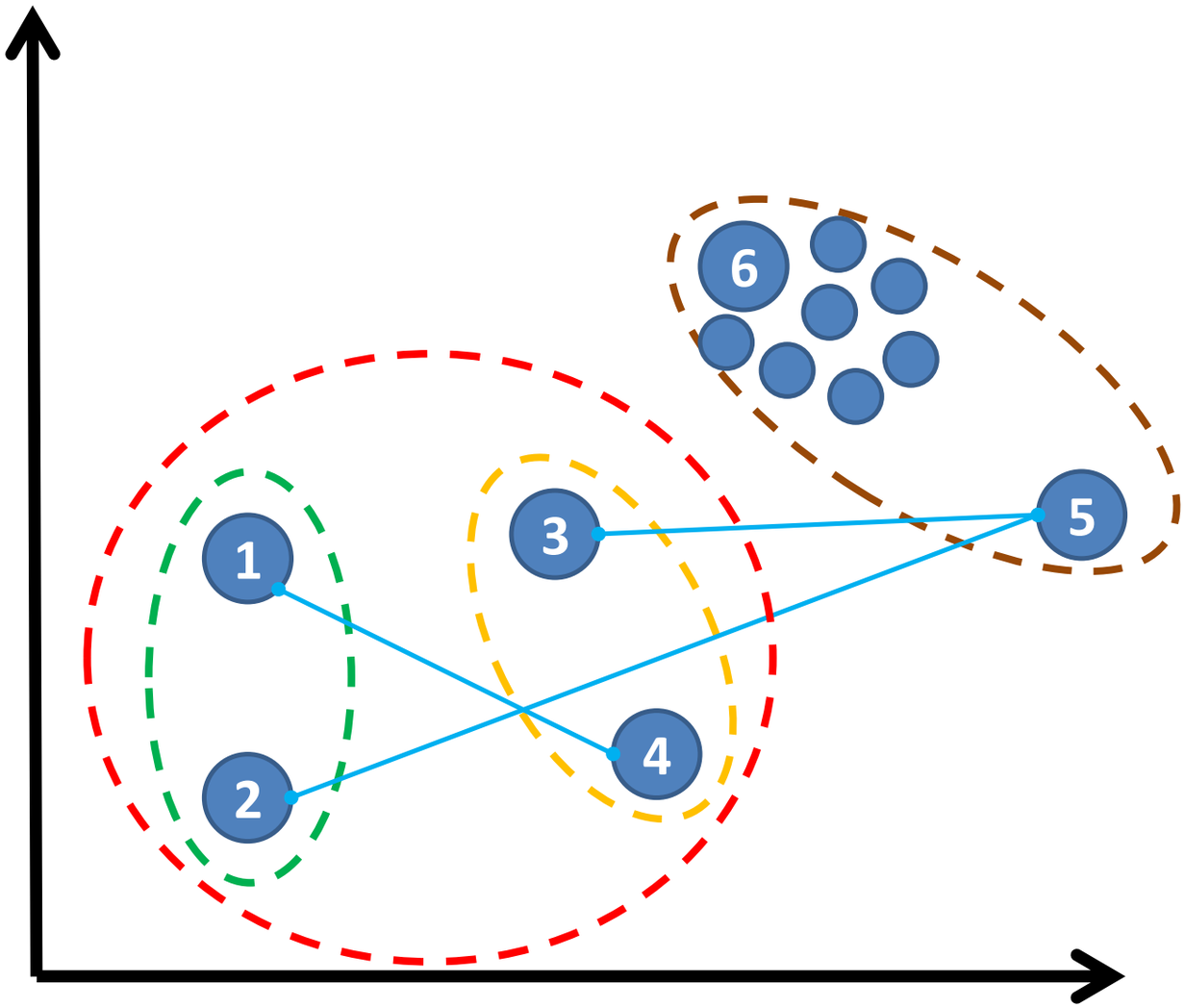




Choosing K for Hierarchical Clustering

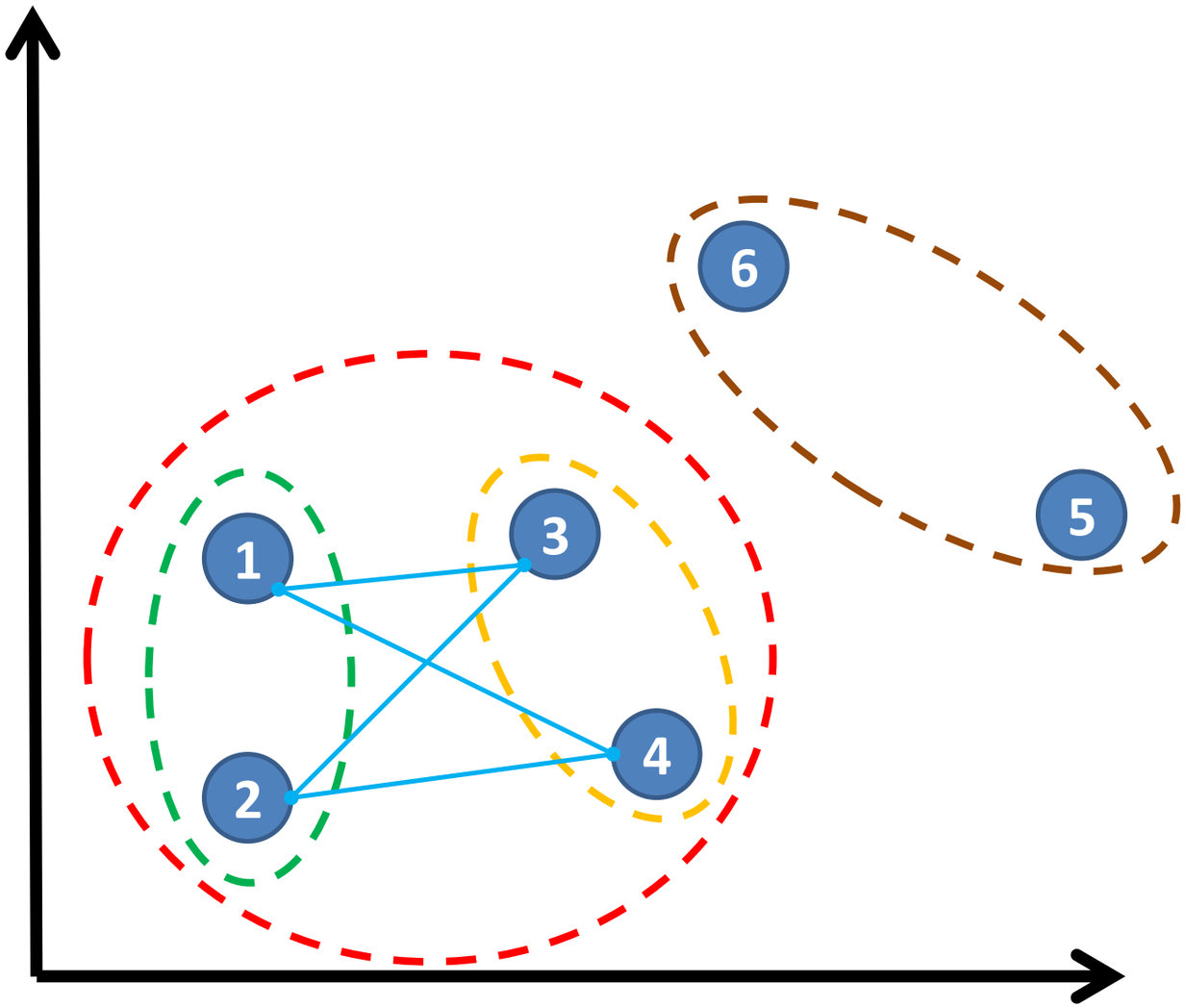


Distance Measure – Complete Link



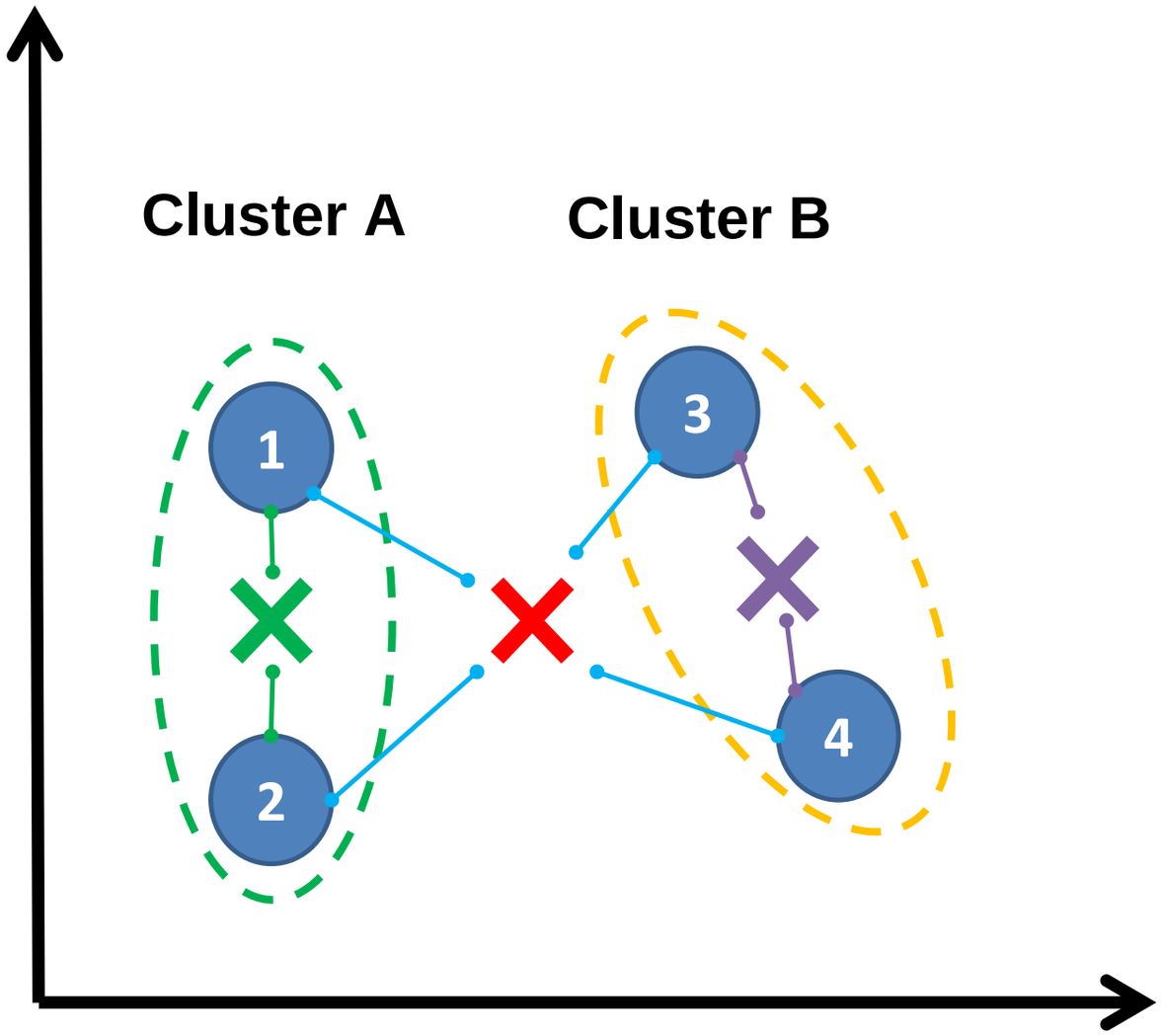
Source: Udacity Machine Learning Nano-Degree

Distance Measure – Average Link



Source: Udacity Machine Learning Nano-Degree

Distance Measure – Ward Method

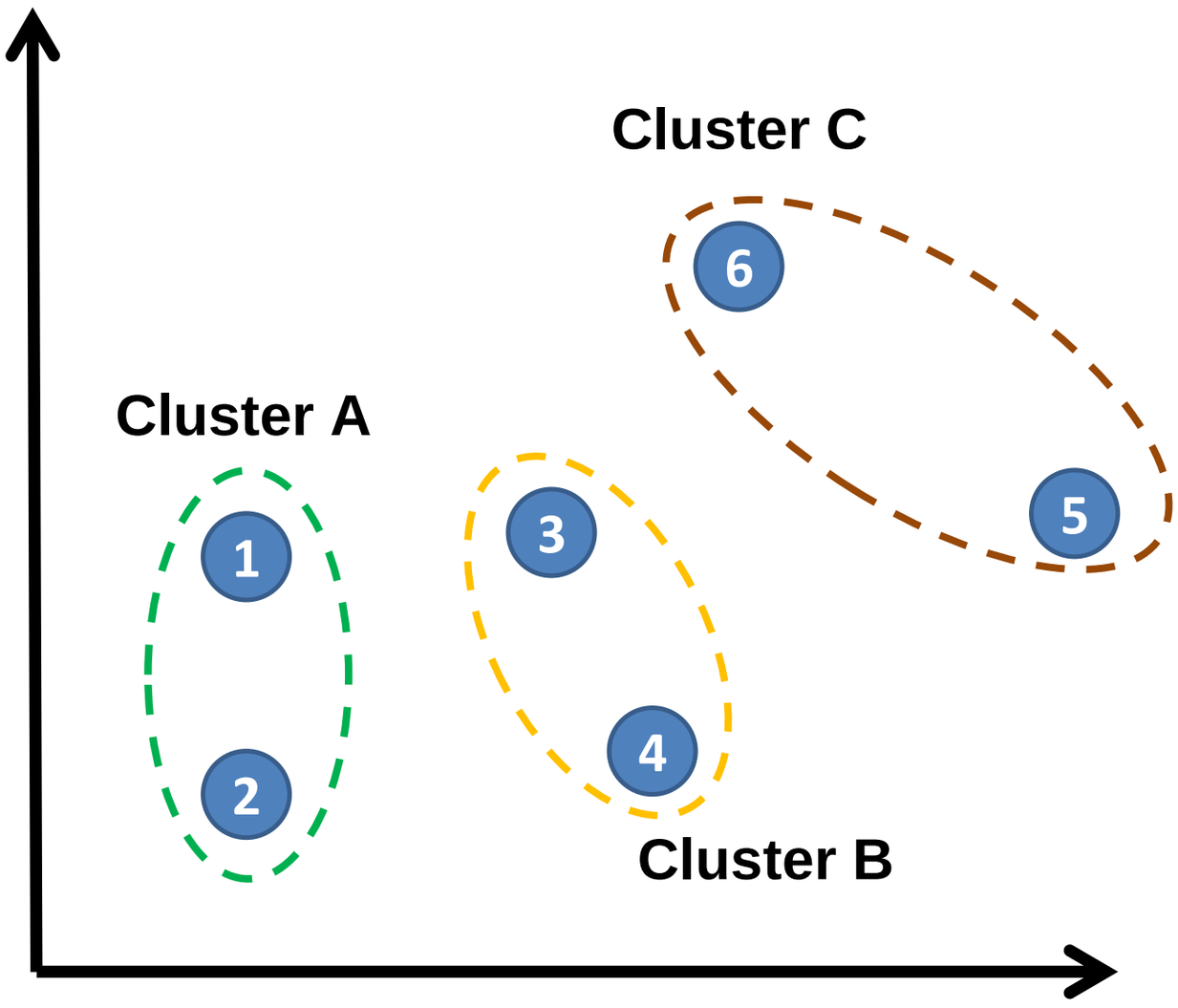


Distance between Clusters A an B

$$\Delta(A,B) = C_1^2 + C_2^2 + C_3^2 + C_4^2 - A_1^2 - A_2^2 - B_1^2 - B_2^2$$

Source: Udacity Machine Learning Nano-Degree

Distance Measure – Ward Method



MIN(Δ(A,B), Δ(A,C), Δ(B,C))

Source: Udacity Machine Learning Nano-Degree

Hierarchical Clustering Implementation

class sklearn.cluster. **AgglomerativeClustering** (*n_clusters=2, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', pooling_func='deprecated', distance_threshold=None*)

[source]

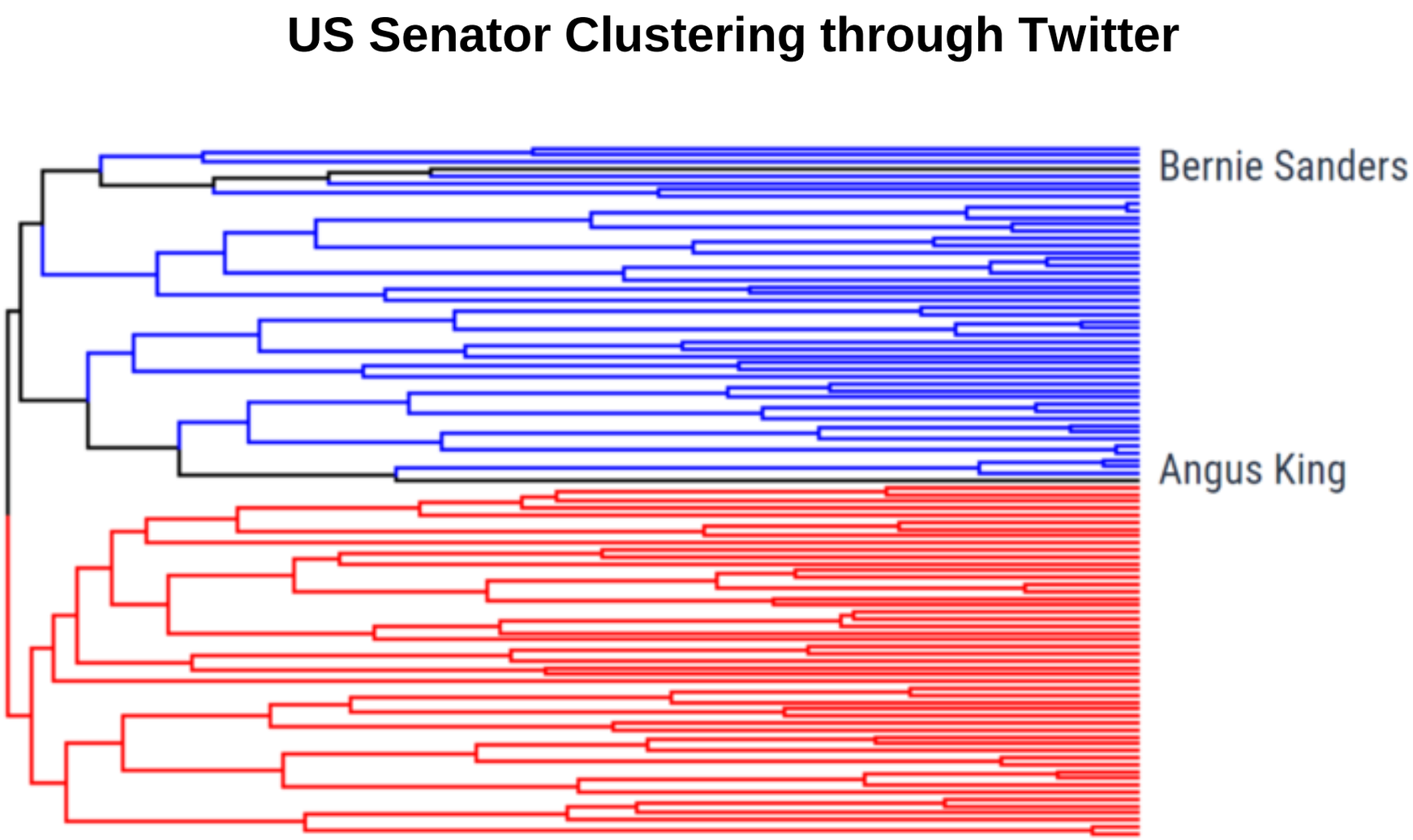
```
>>> from sklearn.cluster import AgglomerativeClustering
>>> import numpy as np
>>> X = np.array([[1, 2], [1, 4], [1, 0],
...               [4, 2], [4, 4], [4, 0]])
>>> clustering = AgglomerativeClustering().fit(X)
>>> clustering
AgglomerativeClustering(affinity='euclidean', compute_full_tree='auto',
                        connectivity=None, distance_threshold=None,
                        linkage='ward', memory=None, n_clusters=2,
                        pooling_func='deprecated')
>>> clustering.labels_
array([1, 1, 1, 0, 0, 0])
```

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

https://docs.scipy.org/doc/scipy-0.14.0/reference/cluster.hierarchy.html

https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.cluster.hierarchy.dendrogram.html#scipy.cluster.hierarchy.dendrogram

Hierarchical Clustering Application

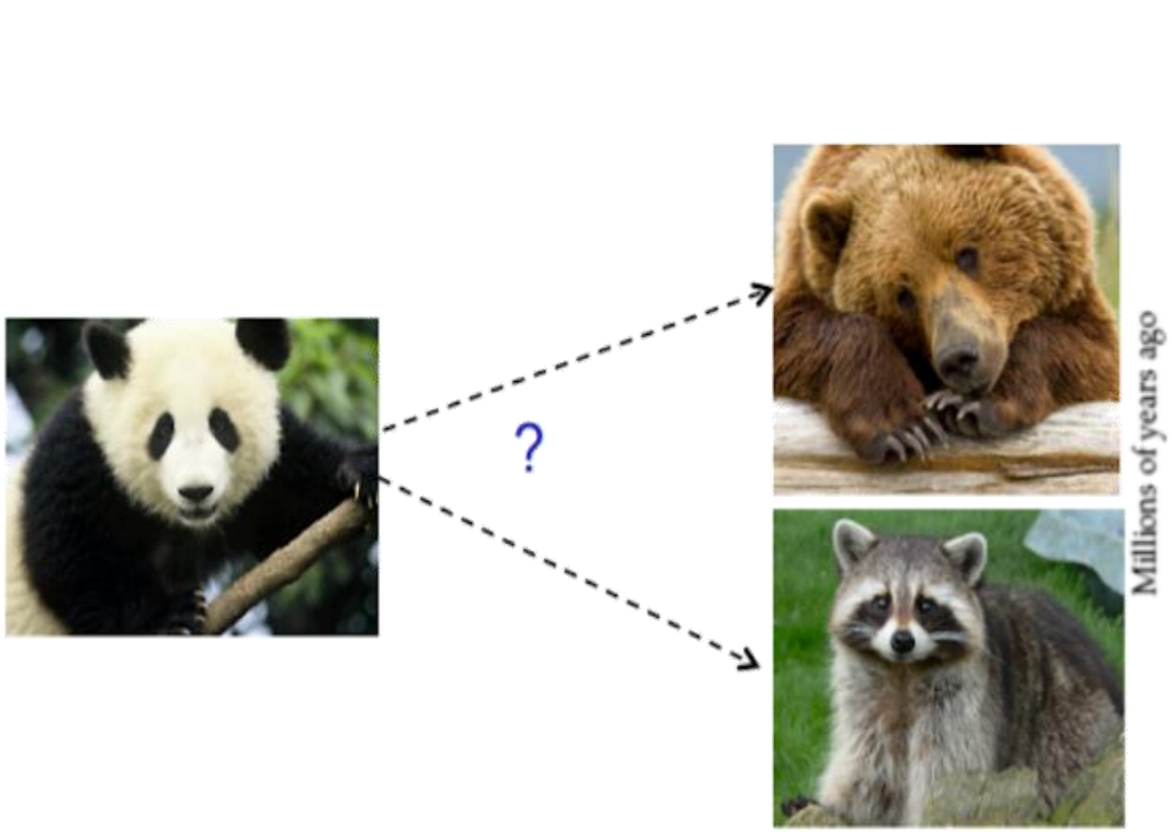


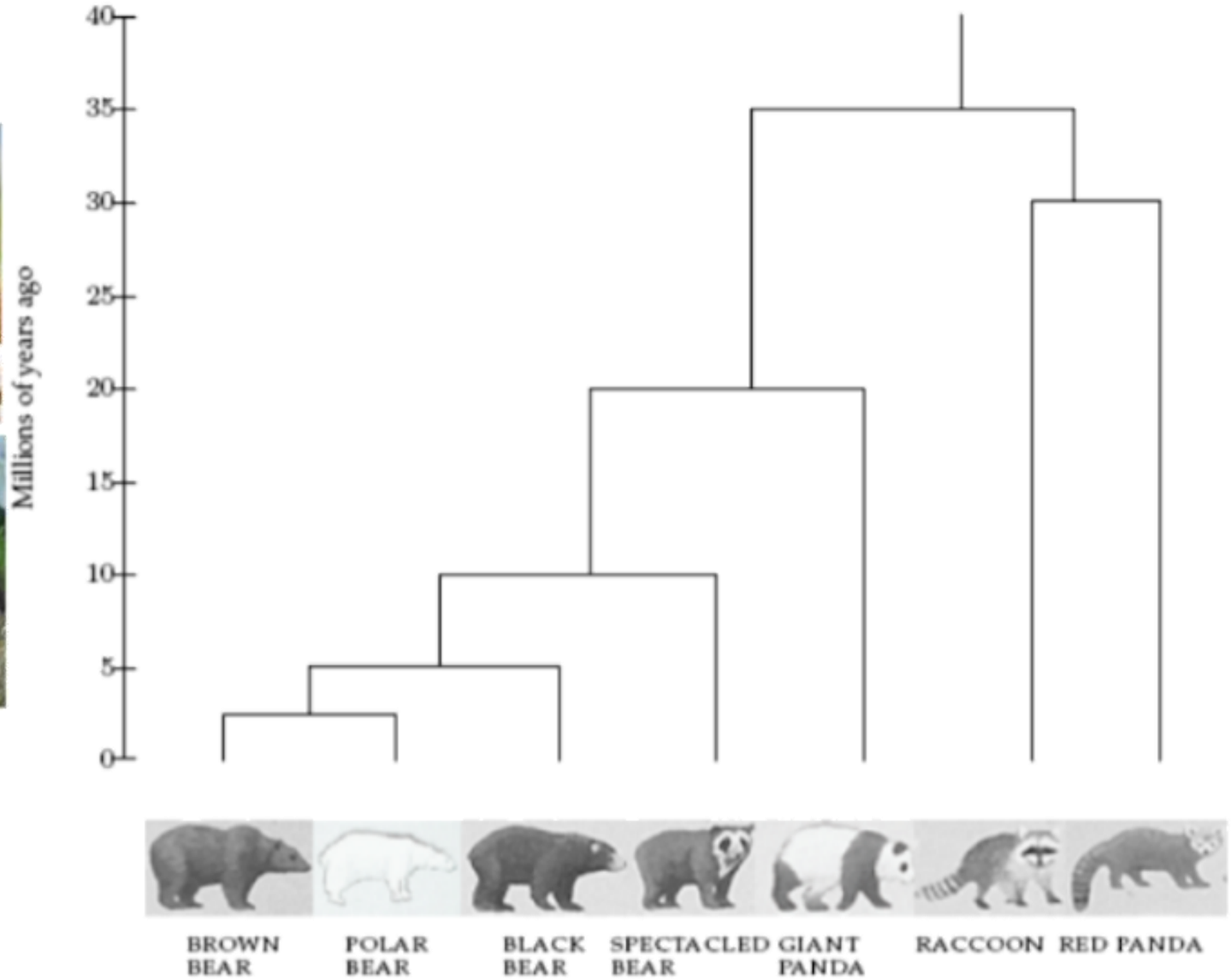
Reds are Republicans, Blues are Democrats, Blacks are independent

https://www.sciencedirect.com/science/article/pii/S187775031100007X

Hierarchical Clustering Application

Charting Evolution through Phylogenetic Trees

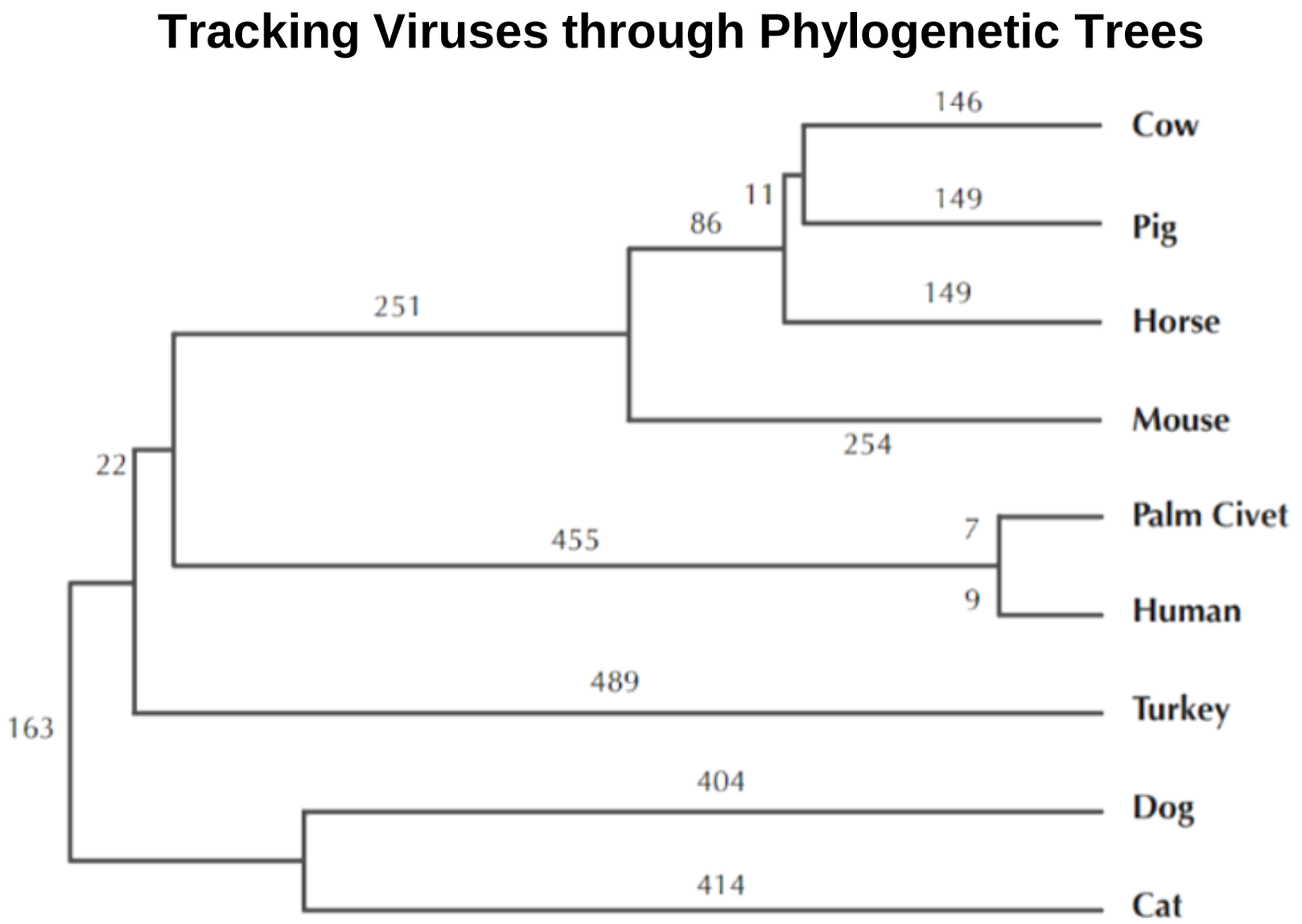




https://en.wikipedia.org/wiki/Phylogenetic_tree

DICE ANALYTICS

Hierarchical Clustering Application



Study was also done for finding the animal that gave the humans the SARS virus

https://www.sciencedaily.com/releases/2008/02/080219150146.htm

Cluster Validation – External Indices

Matching clustering structure to information we know beforehand

Metric	Range	Available in sklearn
Adjusted Rand Score	[-1,1]	Yes
Fawlks and Mallows	[0,1]	Yes
NMI Measure	[0,1]	Yes
Jaccard	[0,1]	Yes
F-measure	[0,1]	Yes
Purity	[0,1]	No

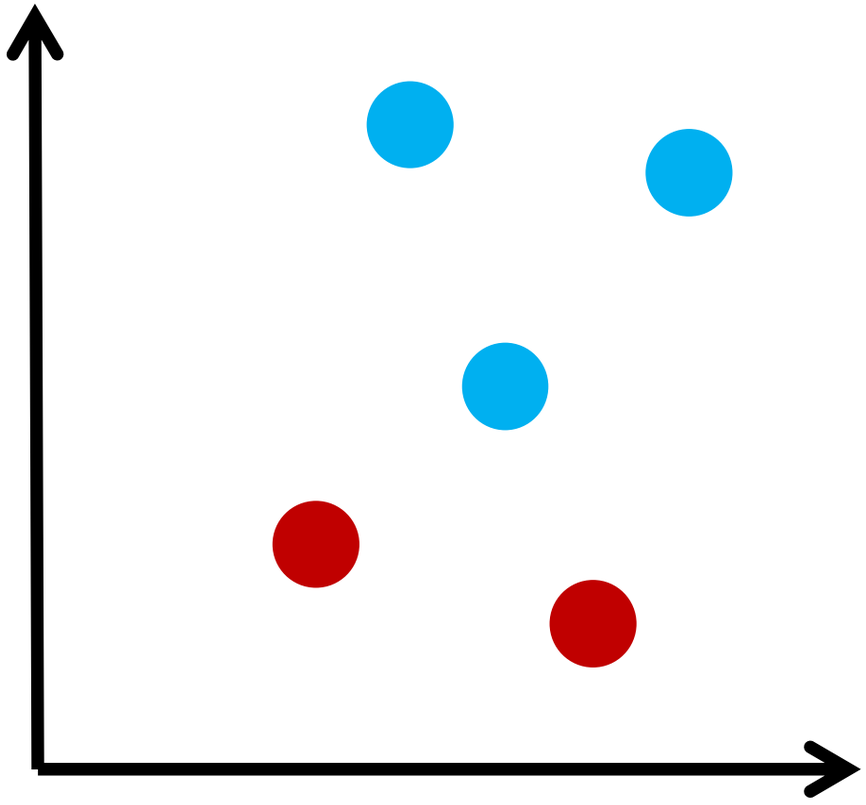
Source: Udacity Machine Learning Nano-Degree

Cluster Validation – External Indices

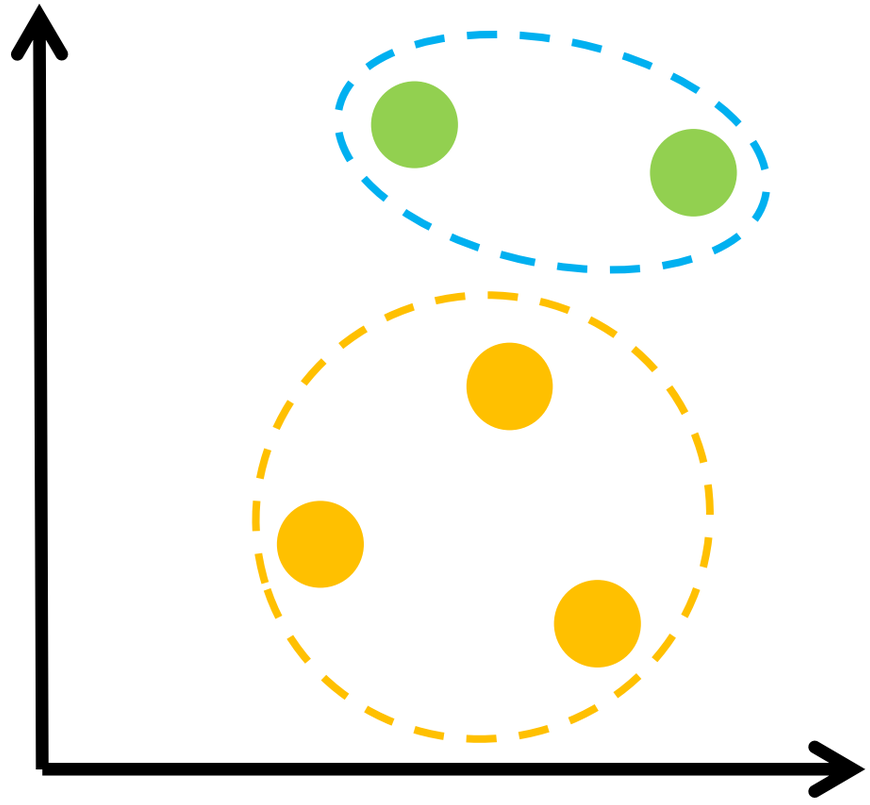
Adjusted Rand Index
Between -1 and 1

Gives a score using comparison of actual vs cluster label groups and pairs of labels

Actual Labels



Clustering Result

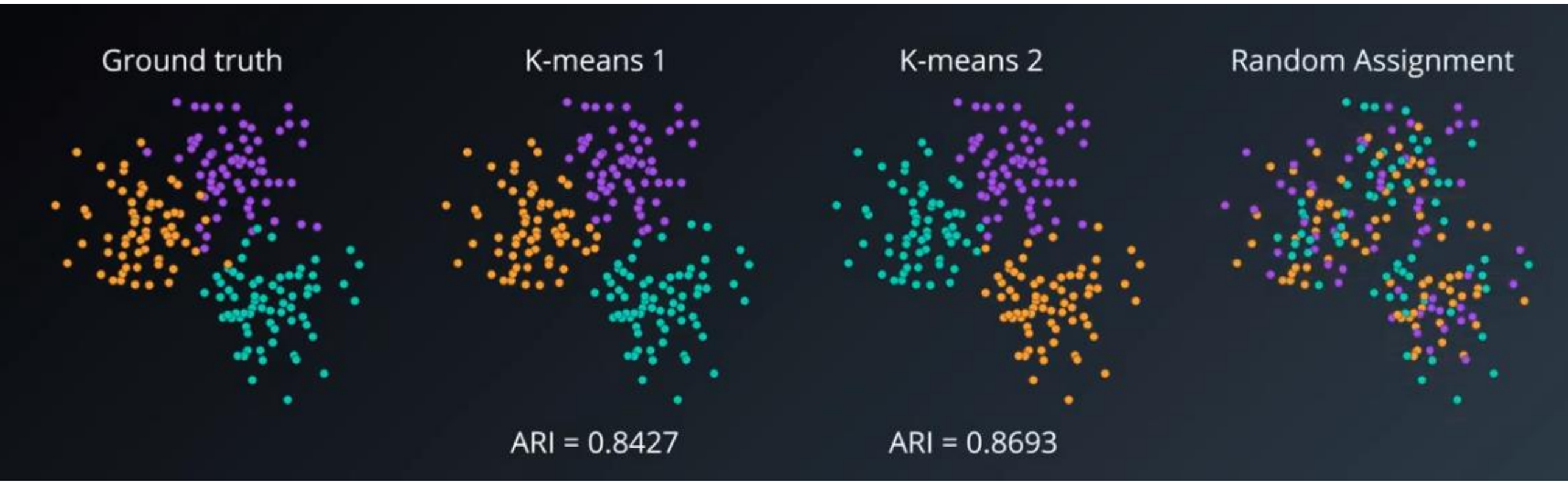


http://faculty.washington.edu/kayee/pca/supp.pdf

Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

Cluster Validation – External Indices



Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

DATA SCIENCE & MACHINE LEARNING COURSE

https://www.facebook.com/diceanalytics/
https://pk.linkedin.com/company/diceanalytics

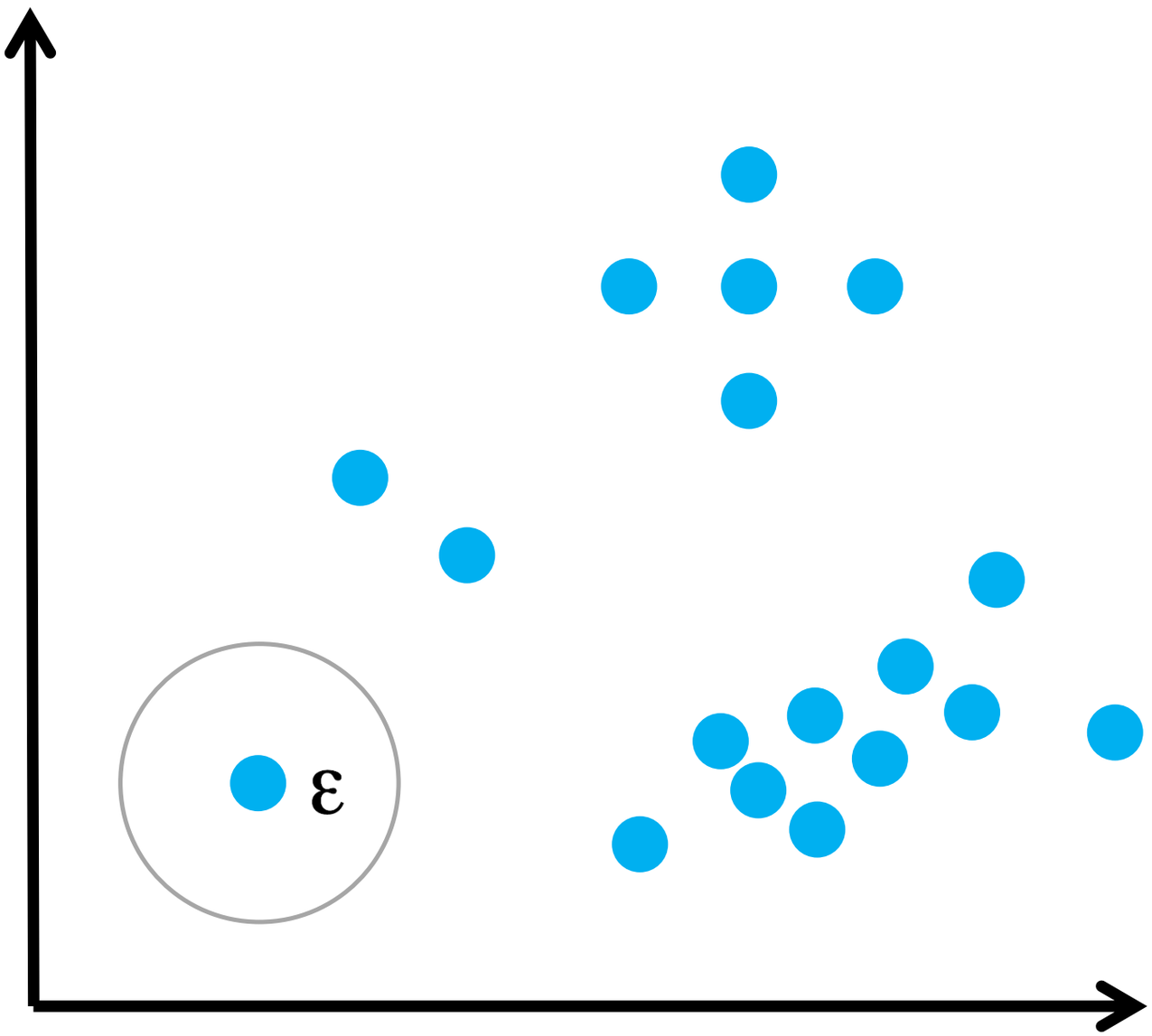
DBSCAN – Density Based Clustering

Density-Based Spatial Clustering of Applications with Noise

epsilon: 1.0
MinPoints: 4

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



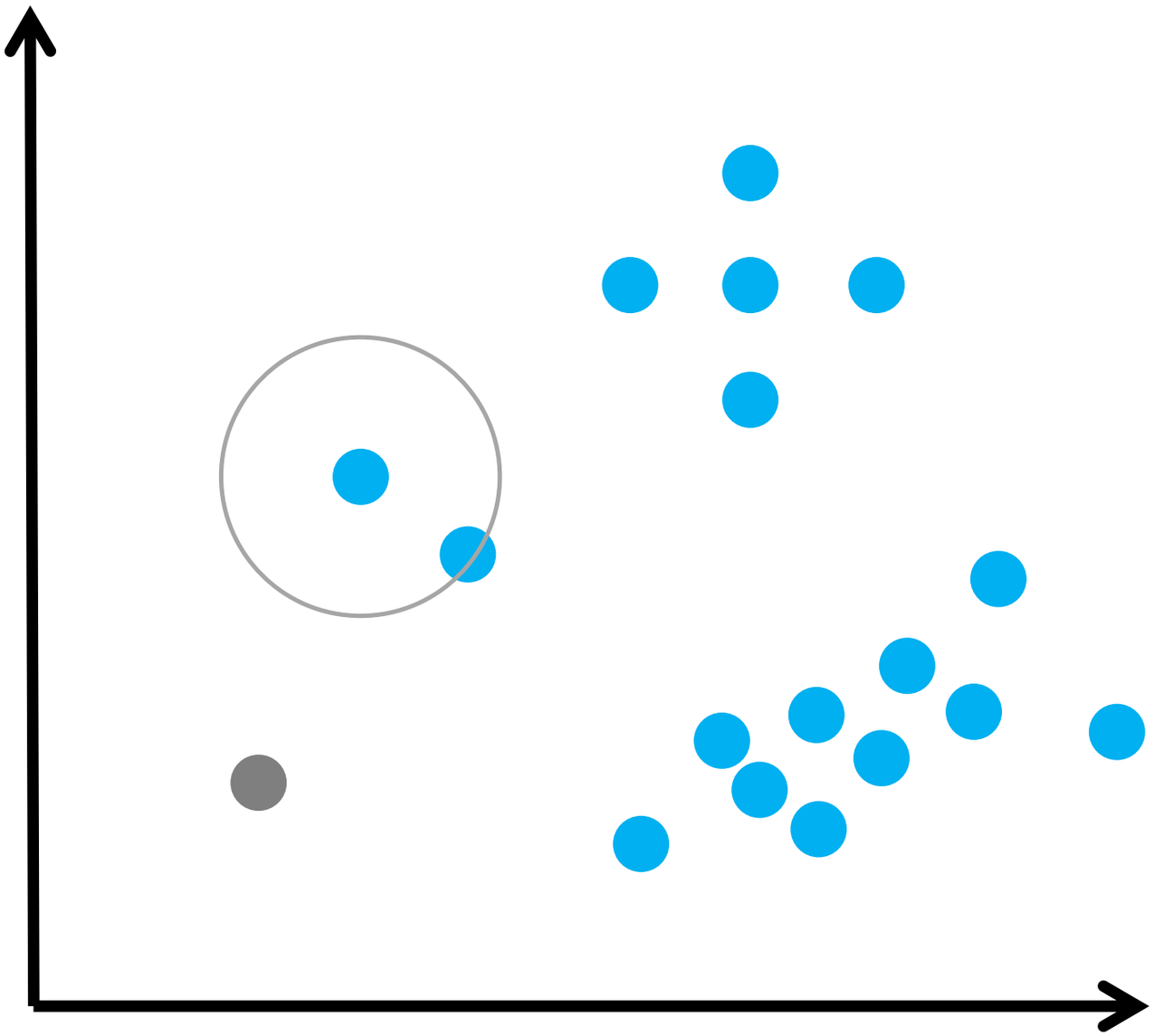
Inputs

Epsilon = 1.0

ε

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

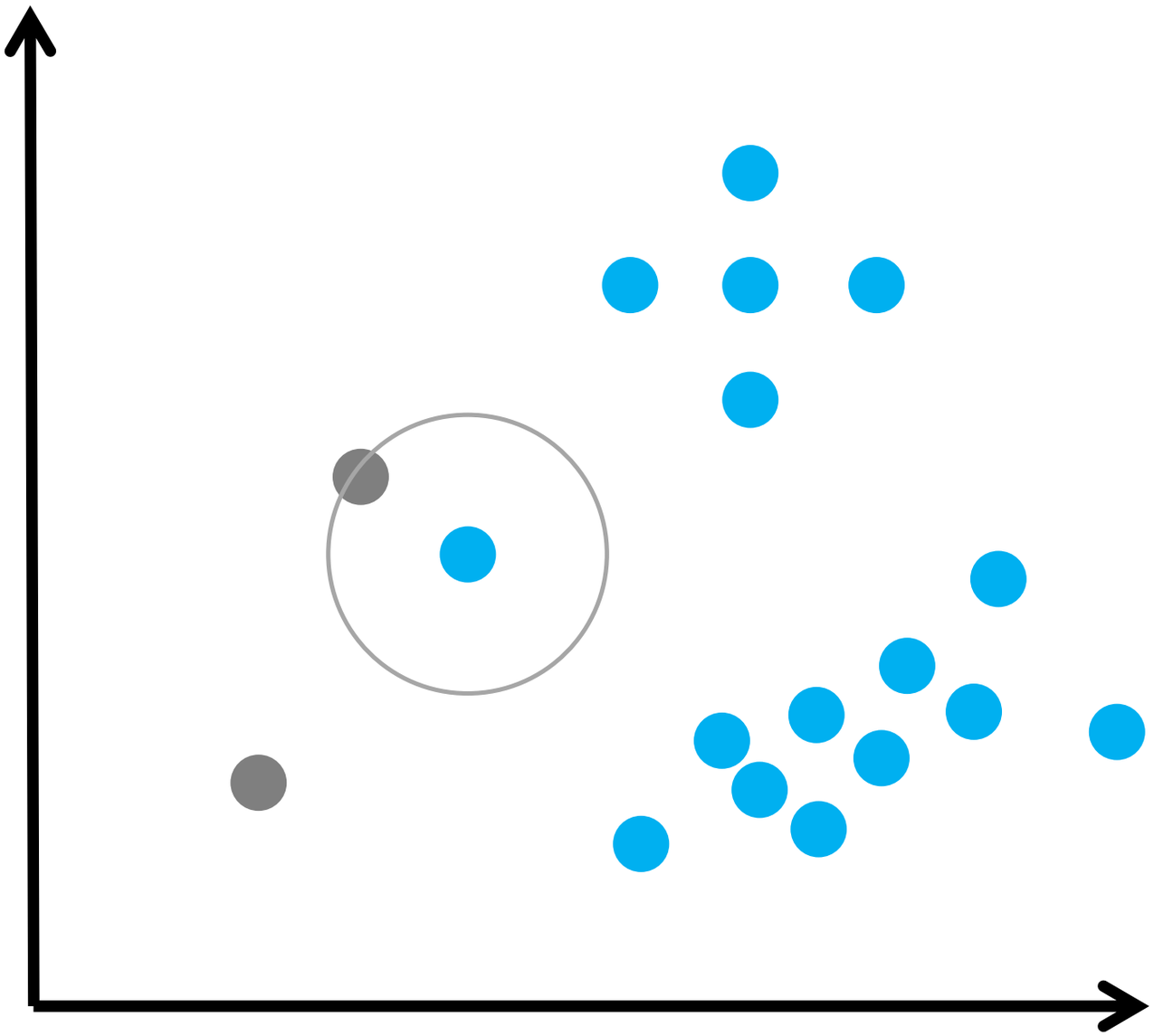
Epsilon = 1.0
Search distance around points

Min Points = 5
Minimum points required to form a density cluster

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

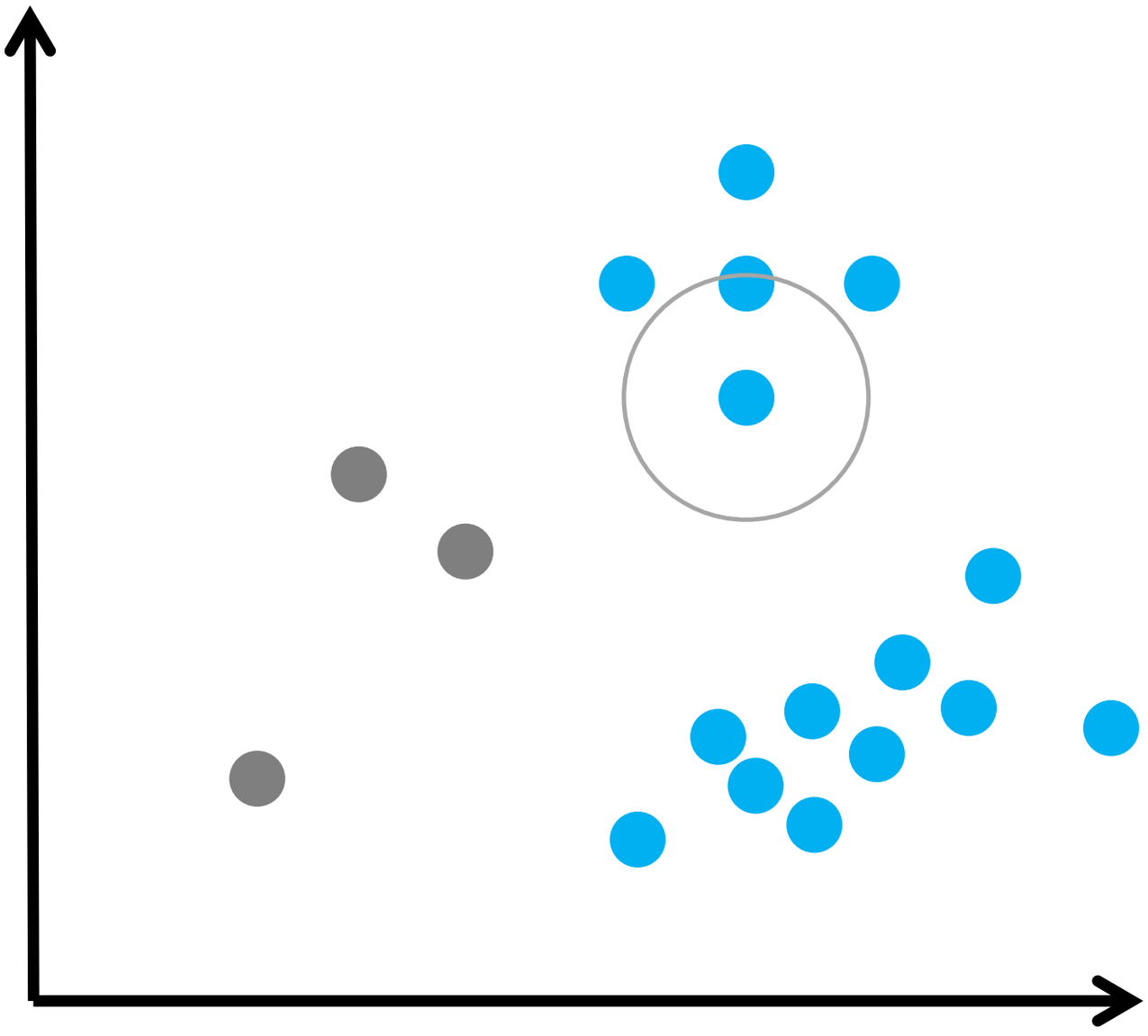
Epsilon = 1.0
Search distance around points

Min Points = 5
Minimum points required to form a density cluster

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

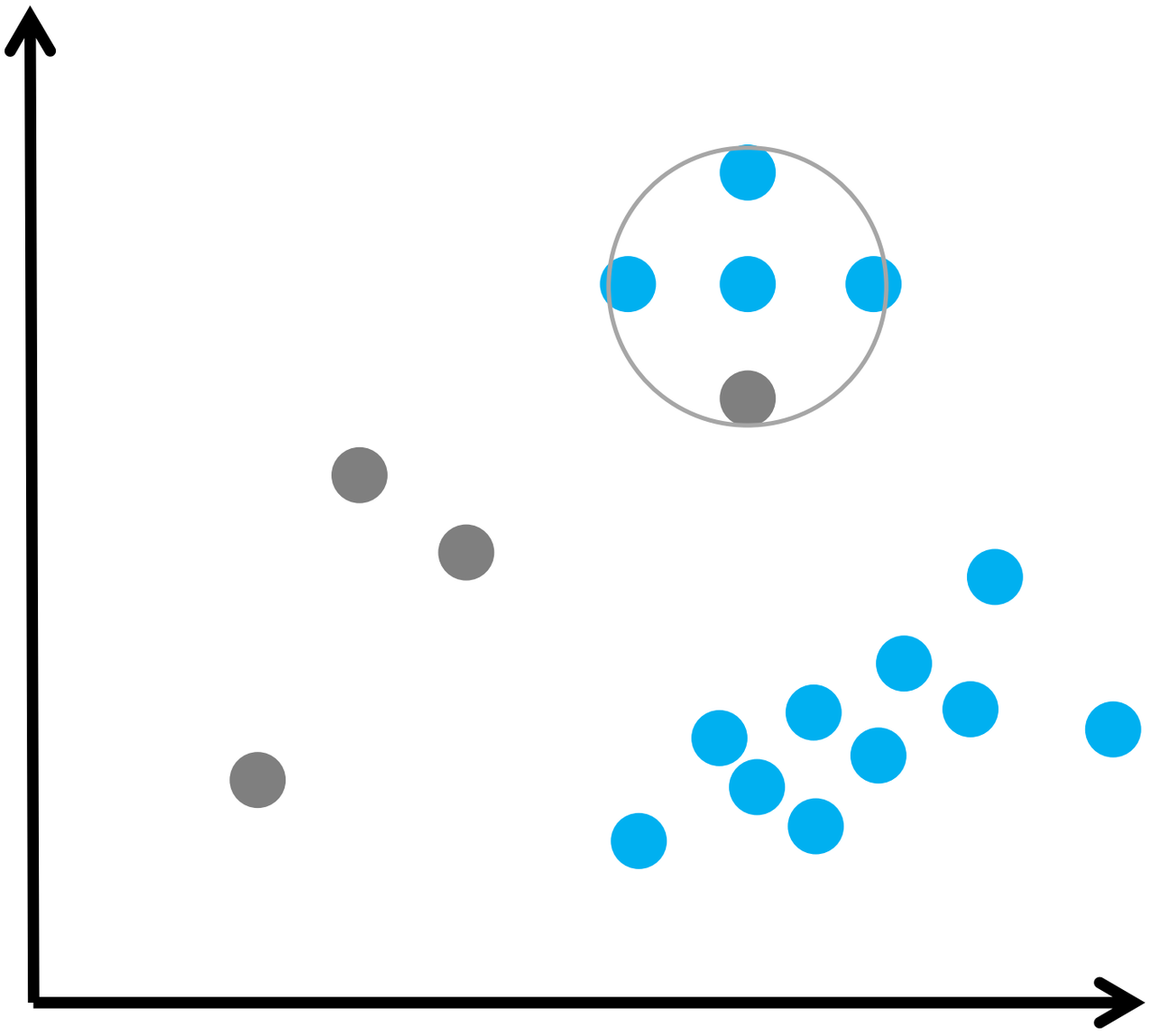
Epsilon = 1.0
Search distance around points

Min Points = 5
Minimum points required to form a density cluster

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

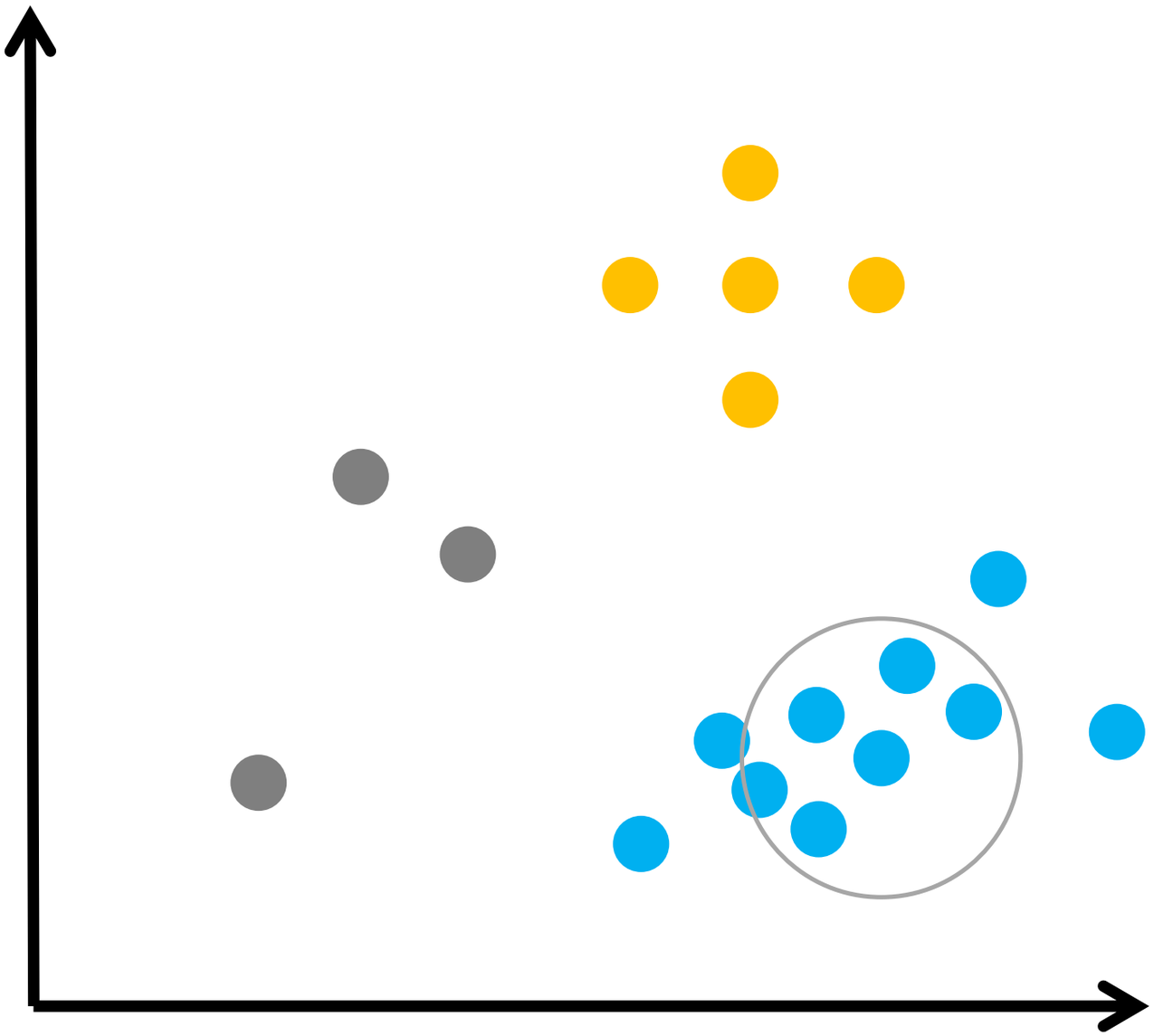
Epsilon = 1.0
Search distance around points

Min Points = 5
Minimum points required to form a density cluster

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

Epsilon = 1.0
Search distance around points

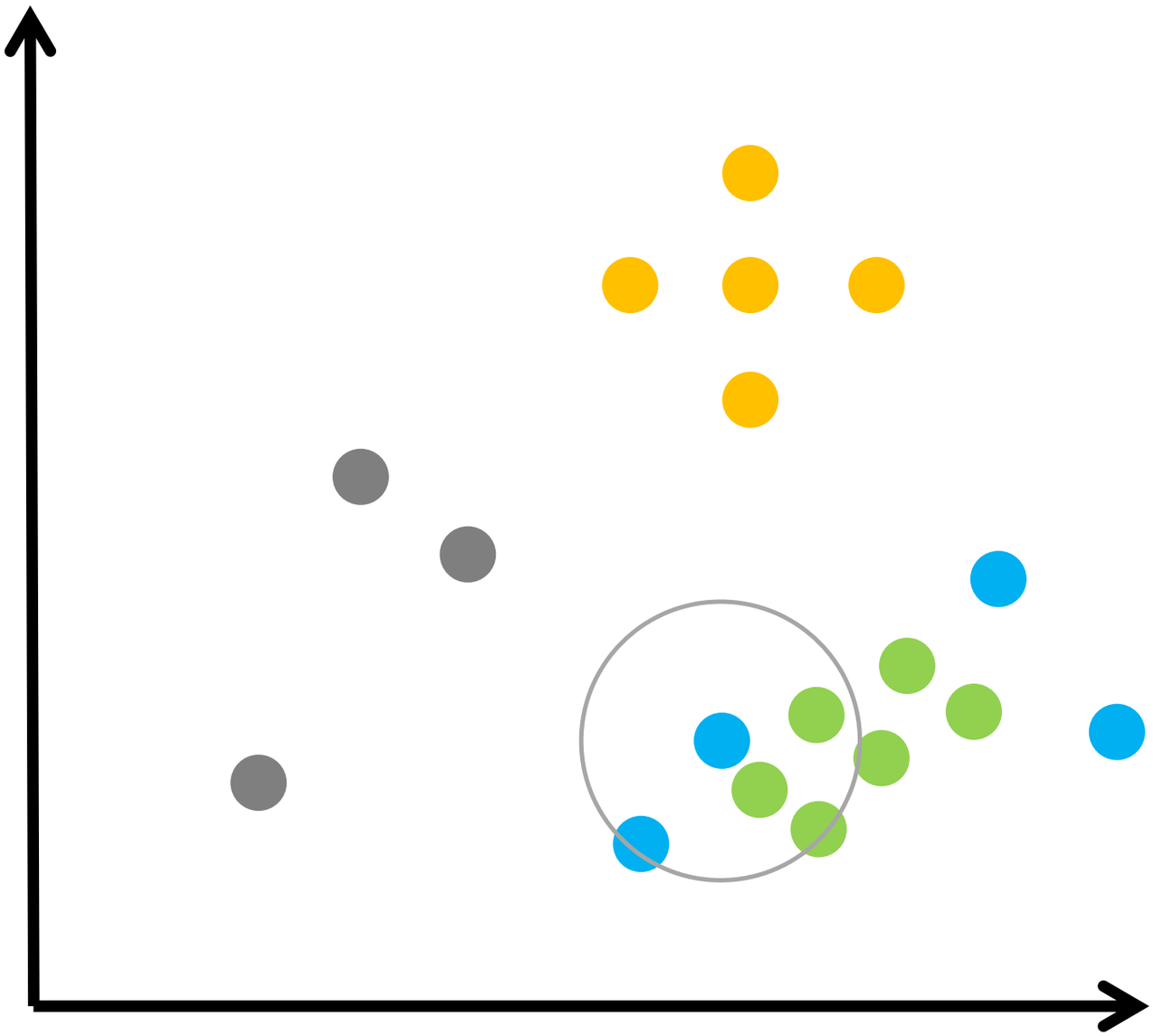
Min Points = 5
Minimum points required to form a density cluster

Cluster-1

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

Epsilon = 1.0
Search distance around points

Min Points = 5
Minimum points required to form a density cluster

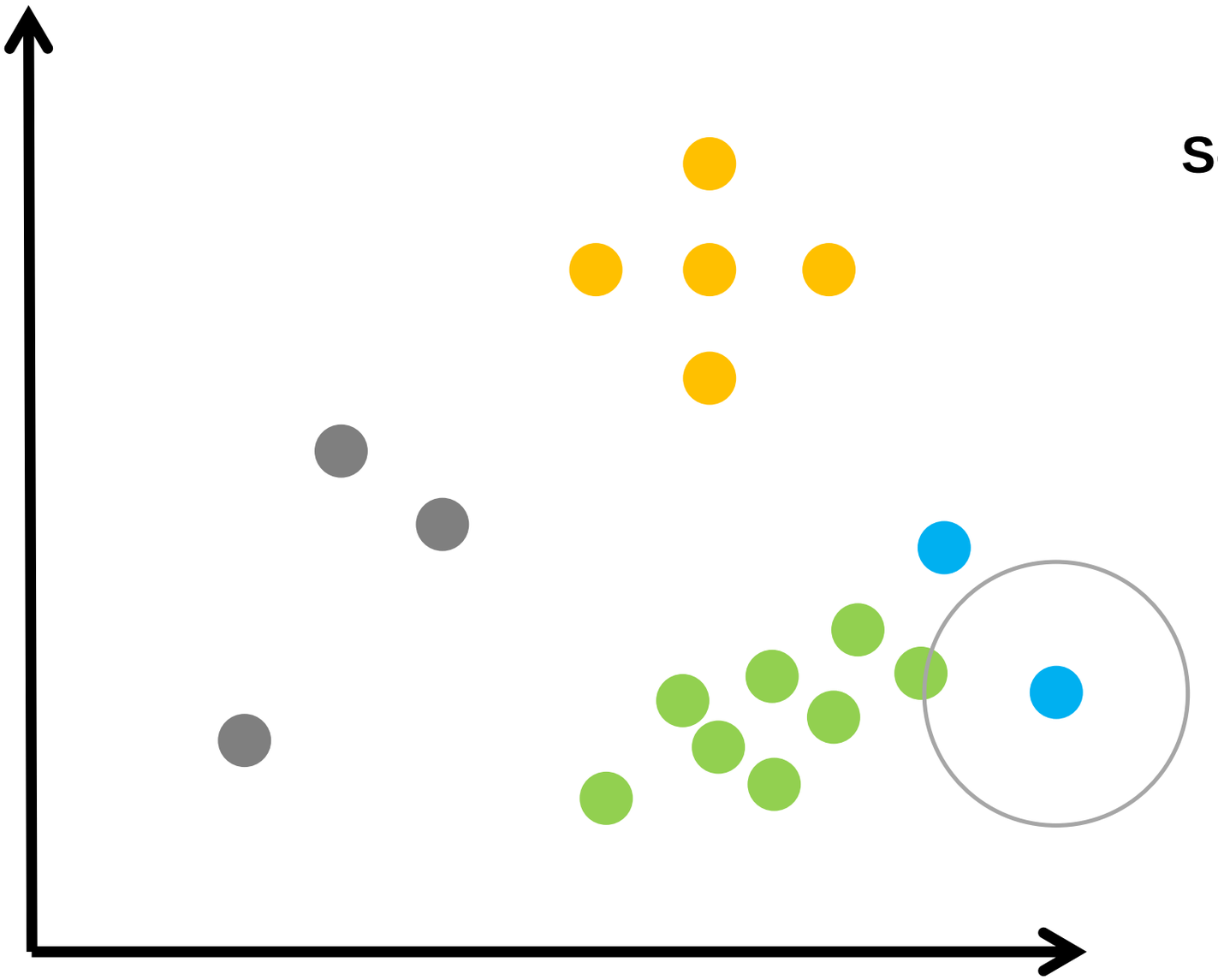
Cluster-2

Cluster-1

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

Epsilon = 1.0
Search distance around points

Min Points = 5
Minimum points required to form a density cluster

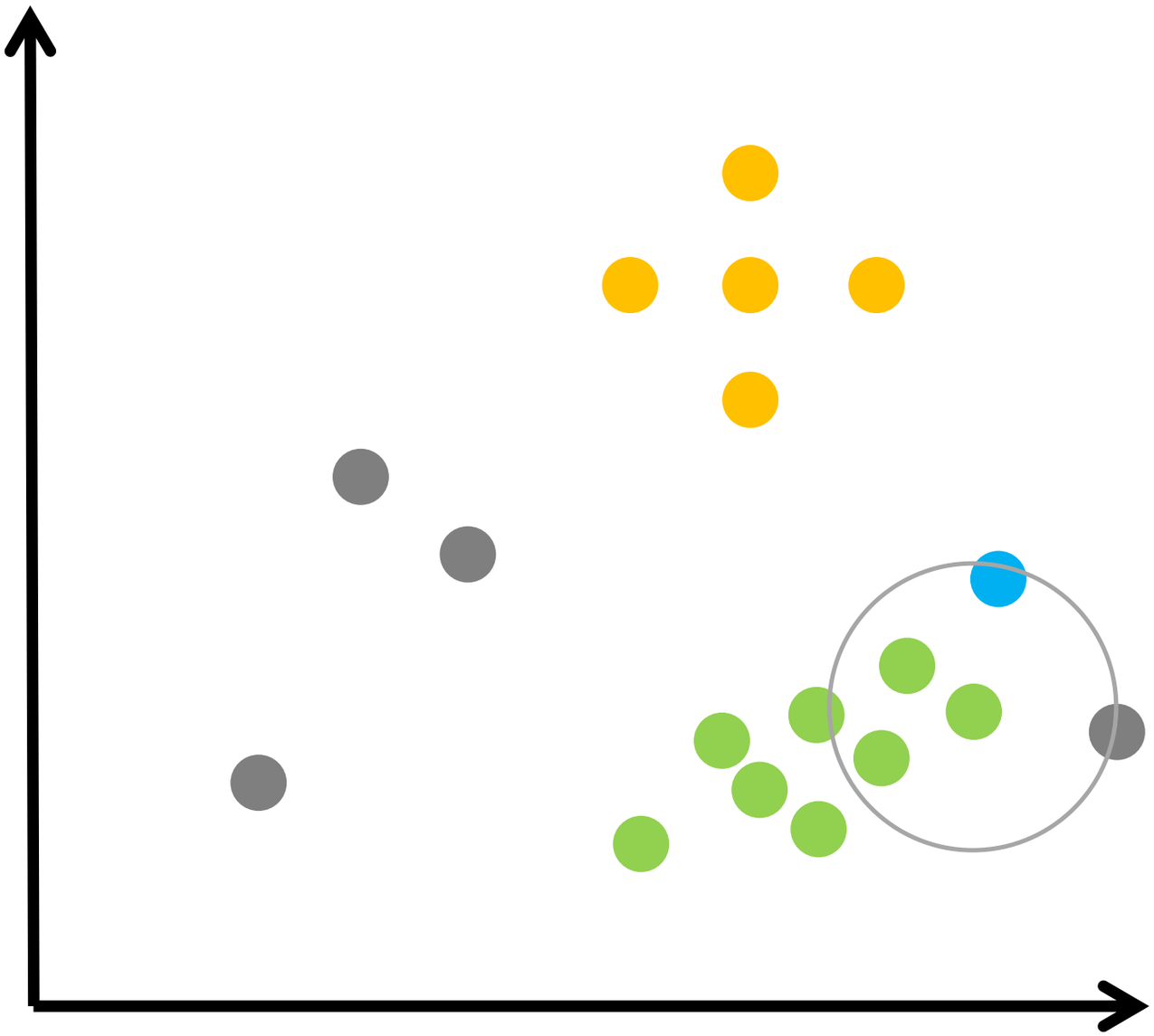
Cluster-2

Cluster-1

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

Epsilon = 1.0
Search distance around points

Min Points = 5
Minimum points required to form a density cluster

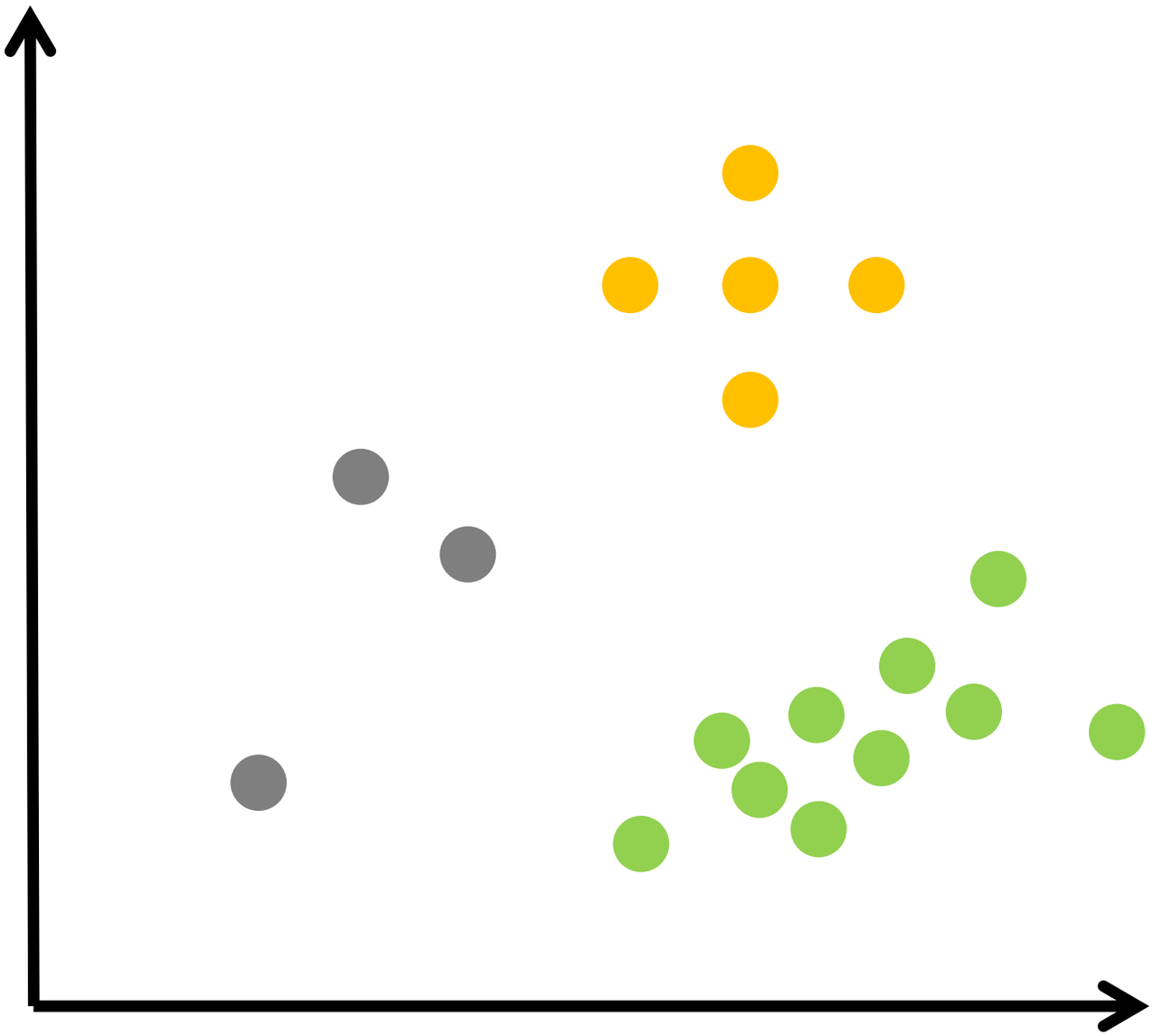
Cluster-2

Cluster-1

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN – Density Based Clustering



Inputs

ε

Epsilon = 1.0
Search distance around points

Min Points = 5
Minimum points required to form a density cluster

Cluster-2

Cluster-1

Noise Point

Source: Udacity Machine Learning Nano-Degree

DBSCAN Implementation

class `sklearn.cluster.` **DBSCAN** (*eps=0.5, min_samples=5, metric='euclidean', metric_params=None, algorithm='auto', leaf_size=30, p=None, n_jobs=None*) [source]

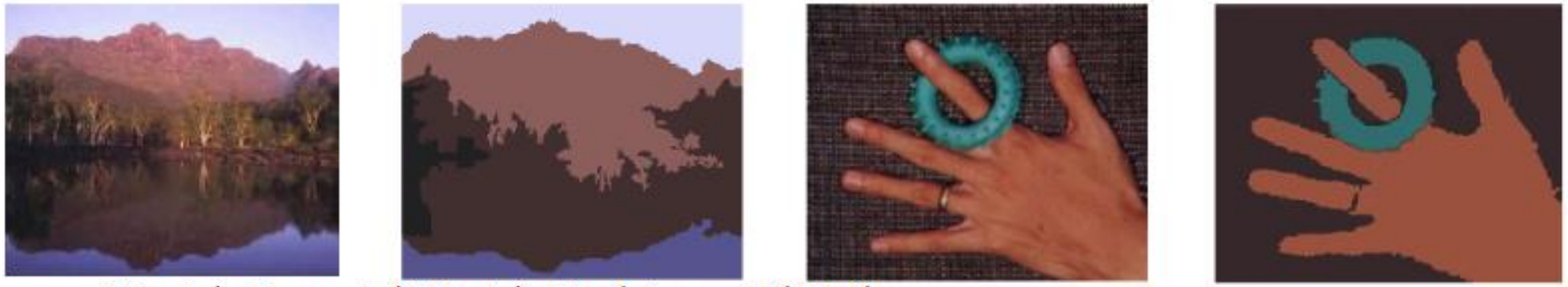
```
>>> from sklearn.cluster import DBSCAN
>>> import numpy as np
>>> X = np.array([[1, 2], [2, 2], [2, 3],
...               [8, 7], [8, 8], [25, 80]])
>>> clustering = DBSCAN(eps=3, min_samples=2).fit(X)
>>> clustering.labels_
array([ 0,  0,  0,  1,  1, -1])
>>> clustering
DBSCAN(algorithm='auto', eps=3, leaf_size=30, metric='euclidean',
    metric_params=None, min_samples=2, n_jobs=None, p=None)
```

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

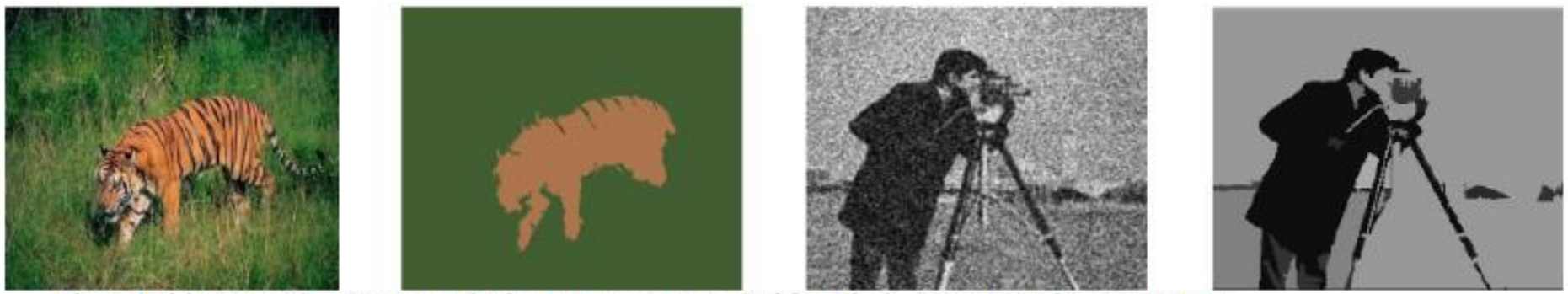
DBSCAN Applications

Color image segmentation using density-based clustering

Pepper Segmented Pepper Plane Segmented Plane



Mountain Segmented Mountain Hand Segmented Hand

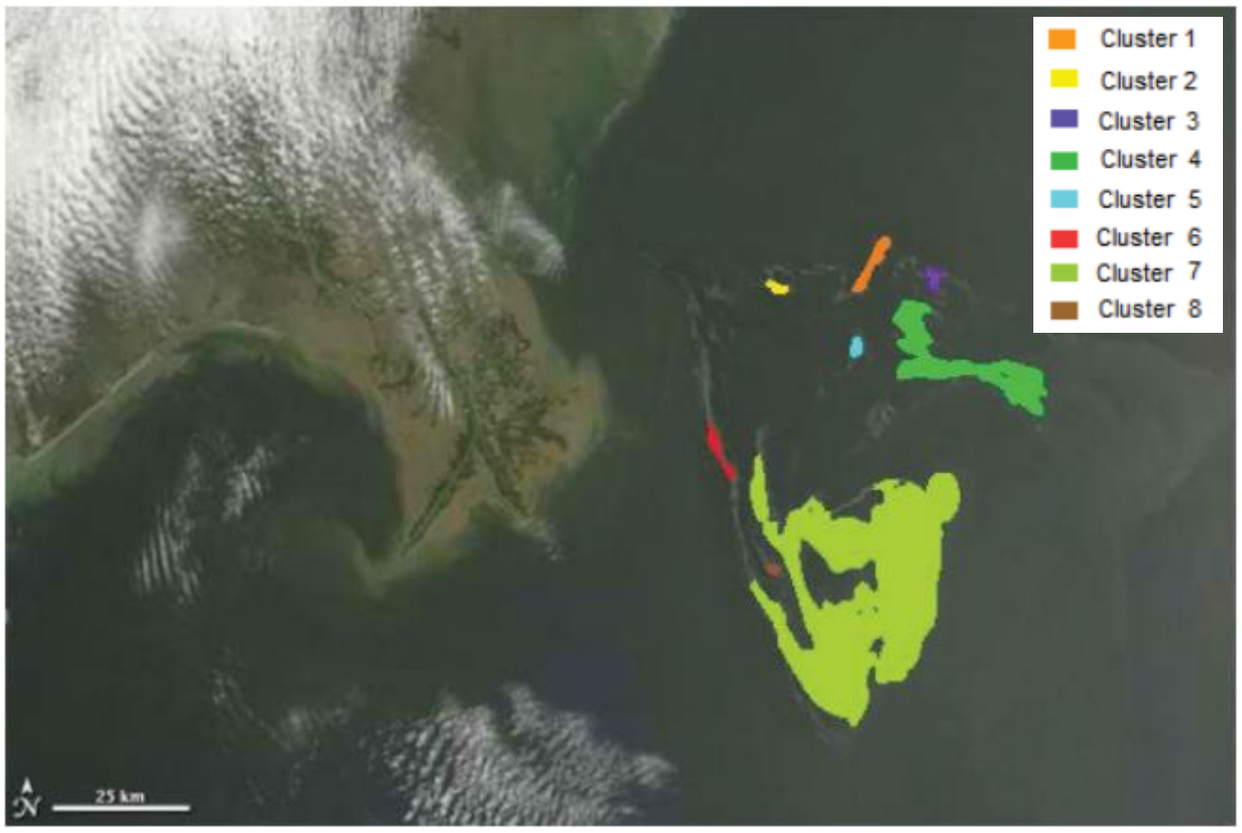


Tiger (with texture) Segmented Tiger Cameraman (with noise) Segmented Cameraman

https://www.researchgate.net/publication/4028066_Color_image_segmentation_using_density-based_clustering

DBSCAN Applications

Density Based Clustering to Oil Spill Detection on Satellite Images



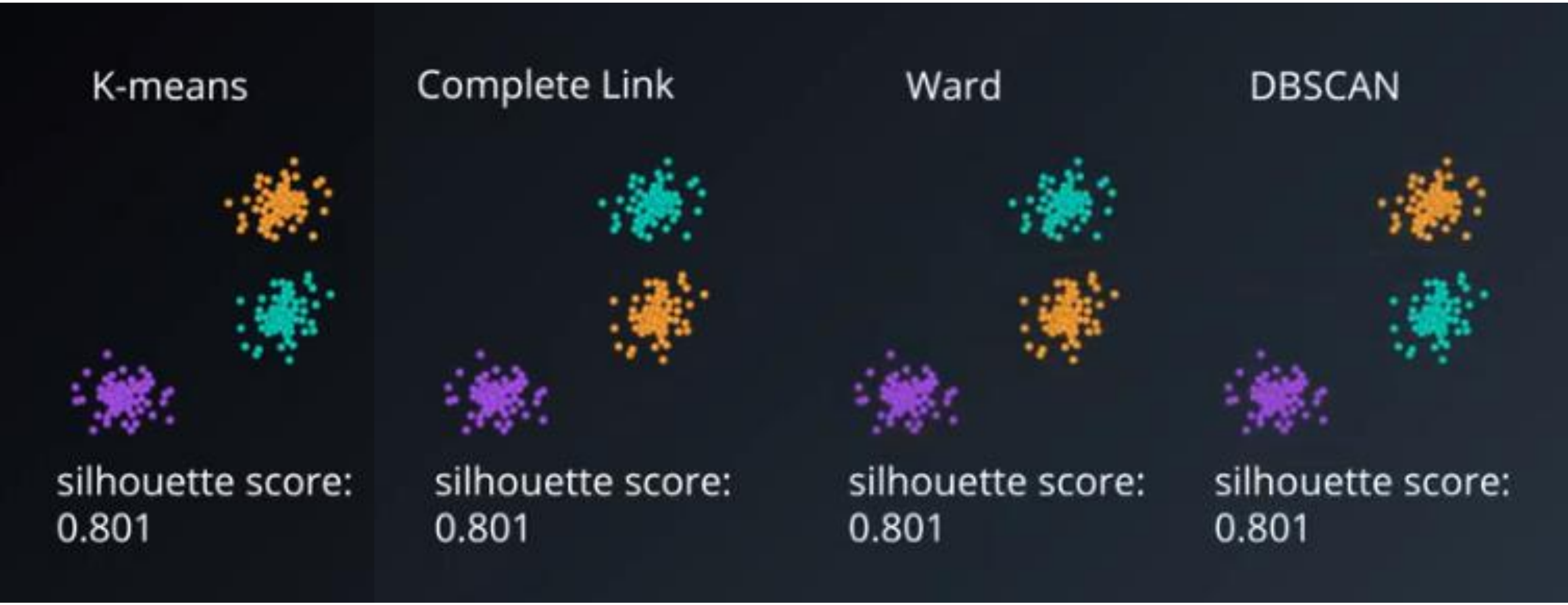
https://shodhganga.inflibnet.ac.in/bitstream/10603/25515/11/11_chapter%205.pdf

DBSCAN Applications

Evolution of Star Formation of Dwarf Galaxies within Extragalactic Cluster Substructures

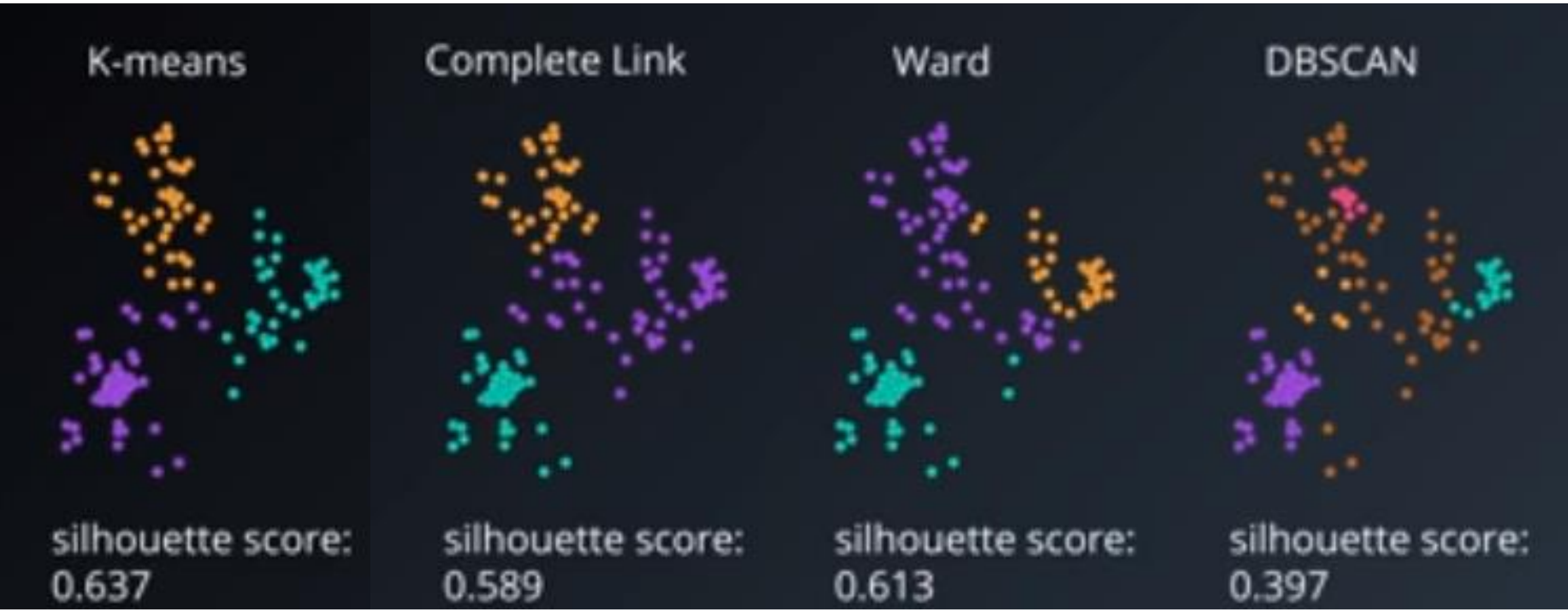
https://www.haystack.mit.edu/edu/reu/2016/files/2016_Archer_Presentation.pdf

Comparing Clustering Algos



Source: Udacity Machine Learning Nano-Degree

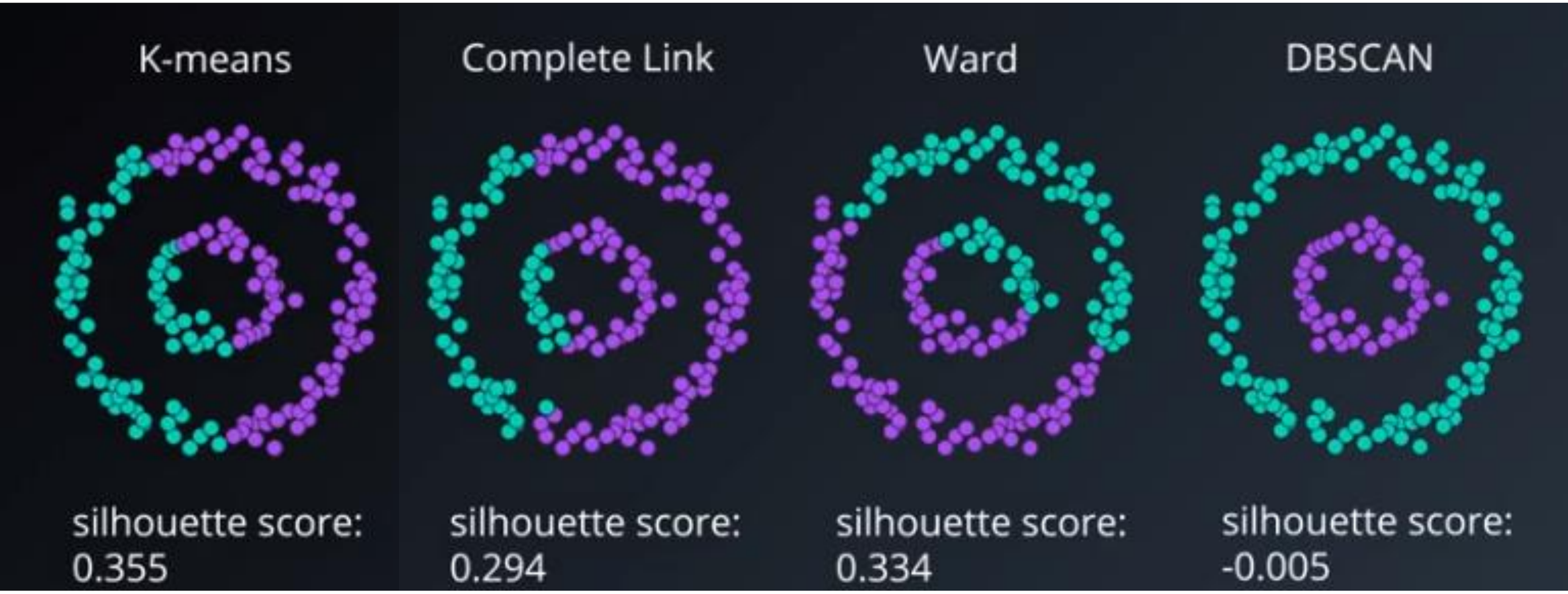
Comparing Clustering Algos



Source: Udacity Machine Learning Nano-Degree

Comparing Clustering Algos

DBCV



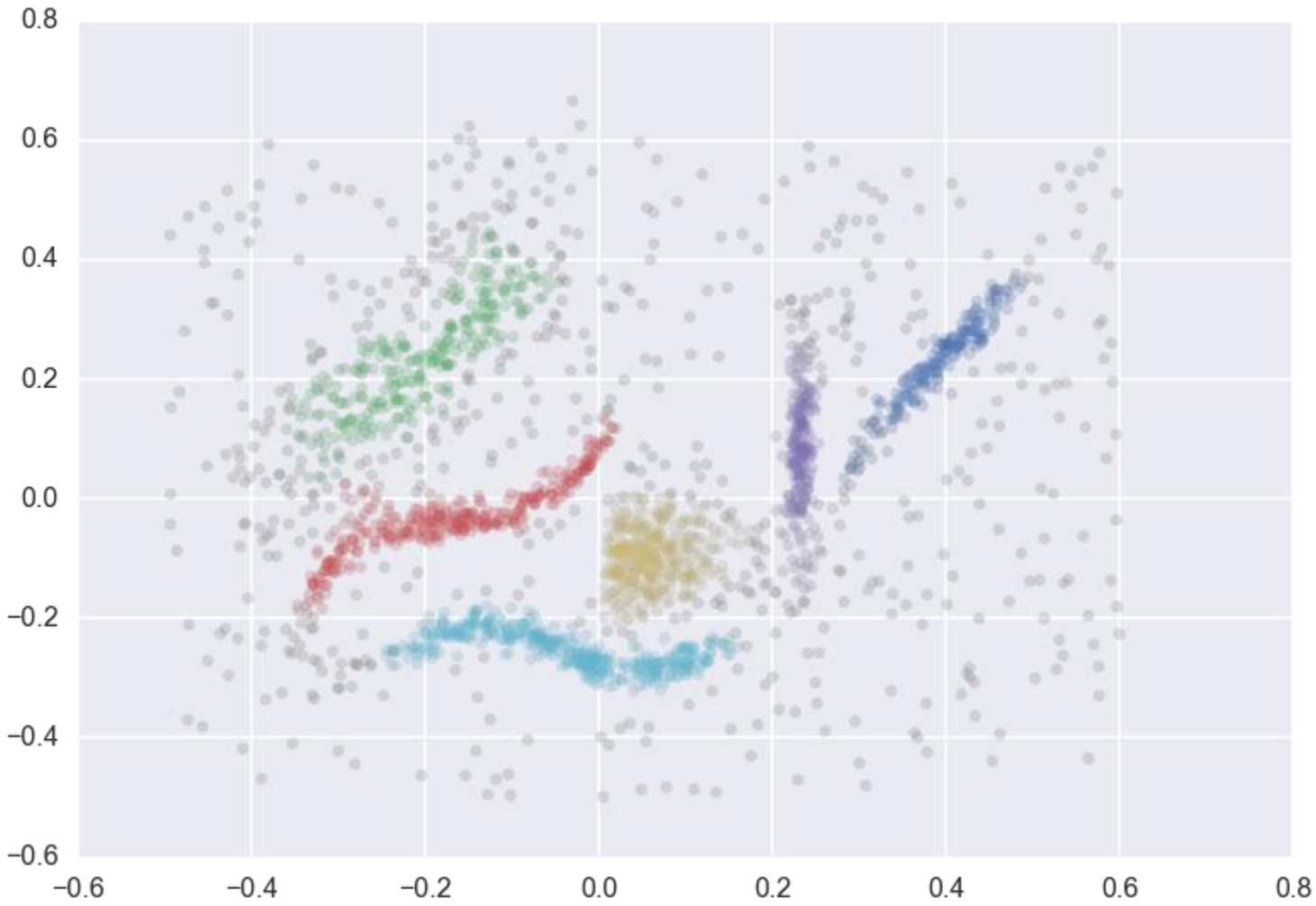
Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

DBCV – Density Based Cluster Validation

DBCV can validate clustering assignments on non-globular, arbitrarily shaped clusters. In essence, DBCV computes two values:

- The density **within** a cluster
- The density **between** clusters



https://epubs.siam.org/doi/pdf/10.1137/1.9781611973440.96

DICE ANALYTICS