MMA/MMAI 869
Machine Learning and AI

Unsupervised Learning: Clustering

Stephen Thomas

Updated: Nov 14, 2022

Smith SCHOOL OF BUSINESS | Queen's University

Outline

- What is clustering?
- What are distance metrics?
- What are the main clustering algorithms?
- How can we interpret/use the clusters?
- How do we know if the clusters are "good" or "bad"?
- Practical Issues

Reminder: Machine Learning Terminology

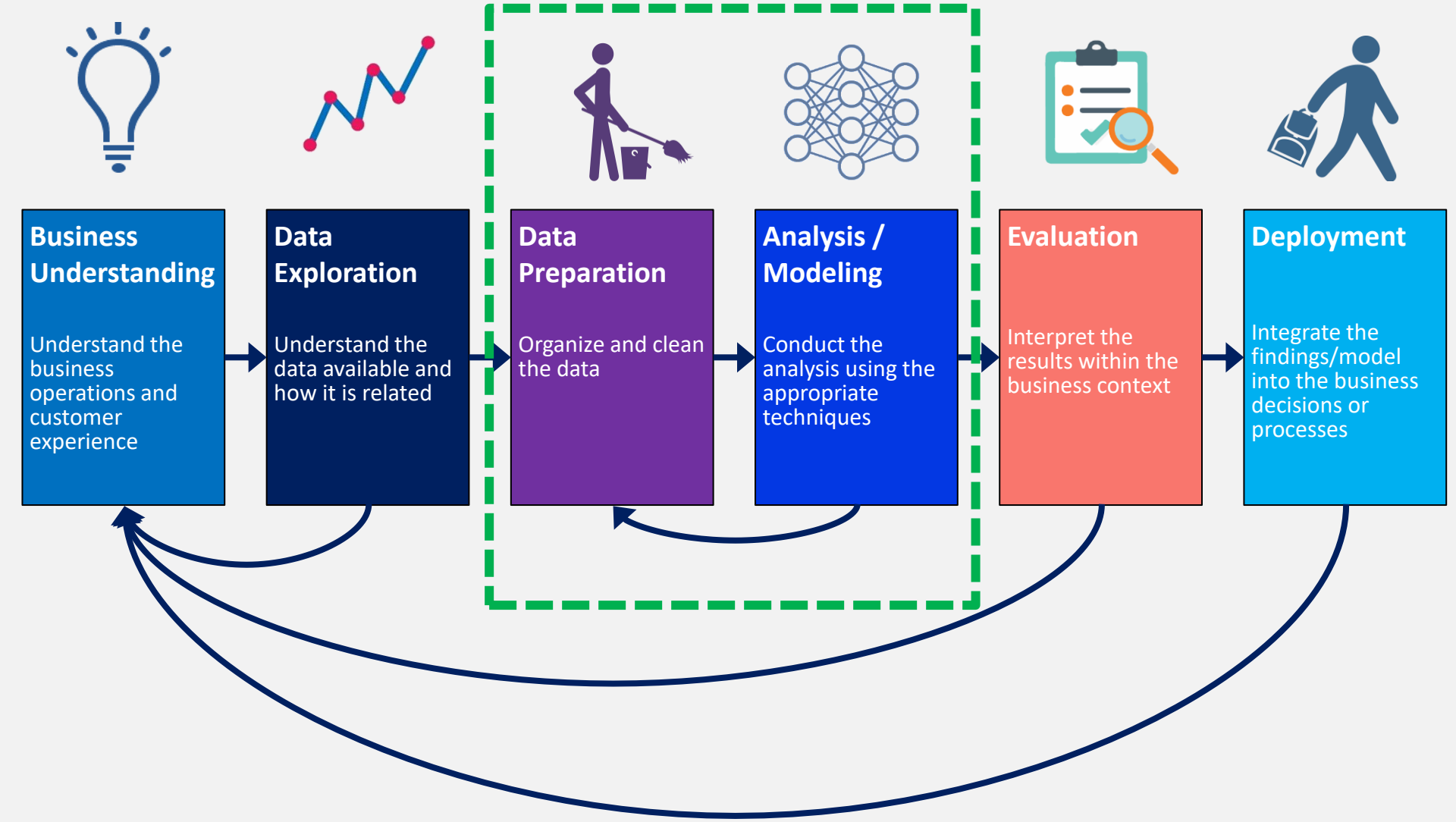
Feature
(inputs, independent variables, X, columns)

Target or **Label**
(response, output, dependent variable, Y)

Instance
(rows, cases, records)

Age	Income	Married	Citizenship	Default
55	36,765	True	Canada	True
66	87,983	True	Canada	True
21	24,354	False	USA	False
24	56,654	True	Canada	False
34	98,324	False	UK	False
36	132,229	False	Germany	True
28	35,000	True	Canada	False
49	50,334	True	Canada	False

The Analytics Process: CRISP-DM



More Detail

Data Preparation

Organize and clean the data

Cleaning

Outliers
Missing data
Data types
Inconsistencies

Feature Engineering

Normalization
Discretization
Coding
Temporal, text, image

Feature Selection

Filter
Wrapper

Analysis / Modeling

Conduct the analysis using the appropriate techniques

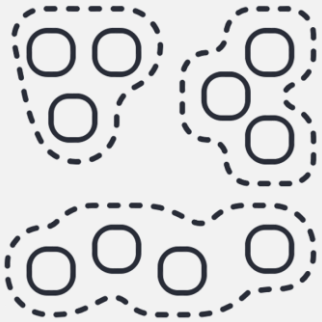
Model Training

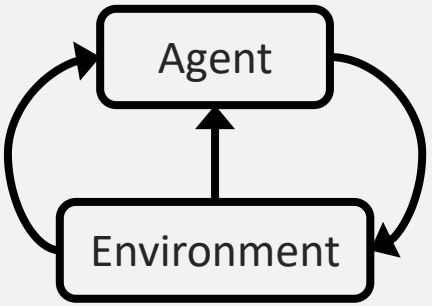
Algorithms
Ensembles
Parameter tuning

Model Selection/ Evaluation

Cross validation
Performance metrics

Three Types of Machine Learning





	Supervised	Unsupervised	Reinforcement
What	Predict something in the future	Find relationships	Learn through trial and error
How	Algorithm builds model from past data	Algorithms finds patterns in data	Algorithm takes actions, gets rewards
Data	Labeled	Unlabeled	None
Tasks/ Algorithms	• Classification – Decision Tree, SVM, Naïve Bayes • Regression – Linear, Polynomial, Lasso • Recommenders – Collaborative filtering, matrix decomposition	• Clustering – K-Means, DBSCAN, Hierarchical • Association rules – Apriori, Eclat, FP-Growth • Dimensionality Reduction – PCA, NMF, LDA, GDA, t-SNE	• Q-learning • SARSA • Deep Q Network

OVERVIEW

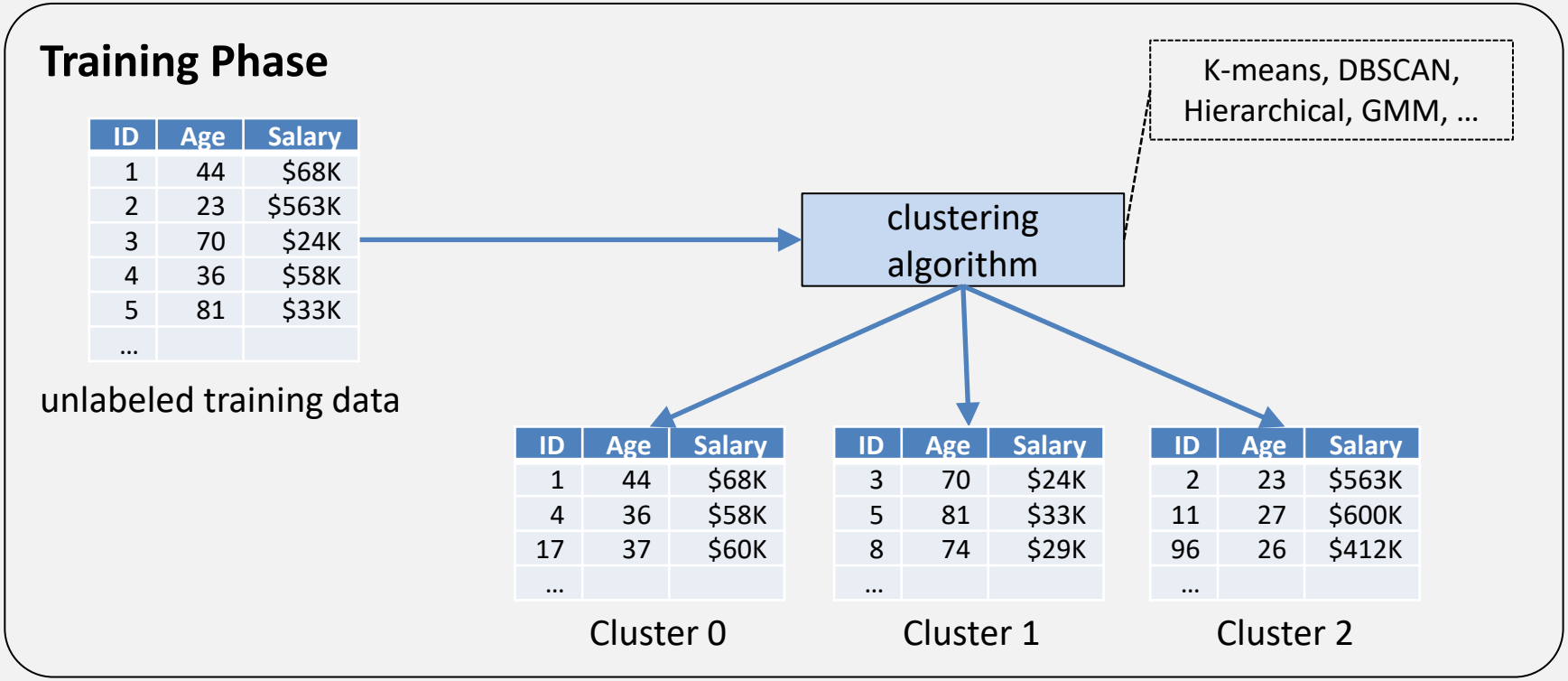

LET 'S TAKE
ALL THE DATA
AND PUT IT
INTO CLUSTERS
imgflip.com

Unsupervised Learning

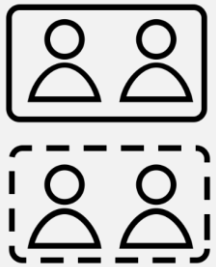
Clustering, Cluster analysis

noun

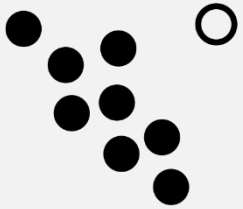
- Putting instances in clusters/groups so that:
  - Instances in the same group are "similar" to each other
  - Instances in different groups are "not similar" to each other



Clustering Applications



Customer Segmentation
(Marketing, Behavior analysis)

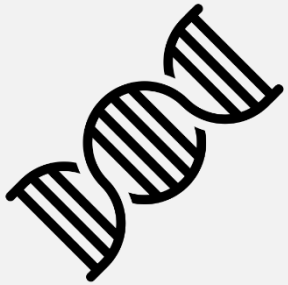


Anomaly Detection
(Insurance Fraud, etc.)

Document Clustering
(Survey analysis, EDA)

Feature Engineering

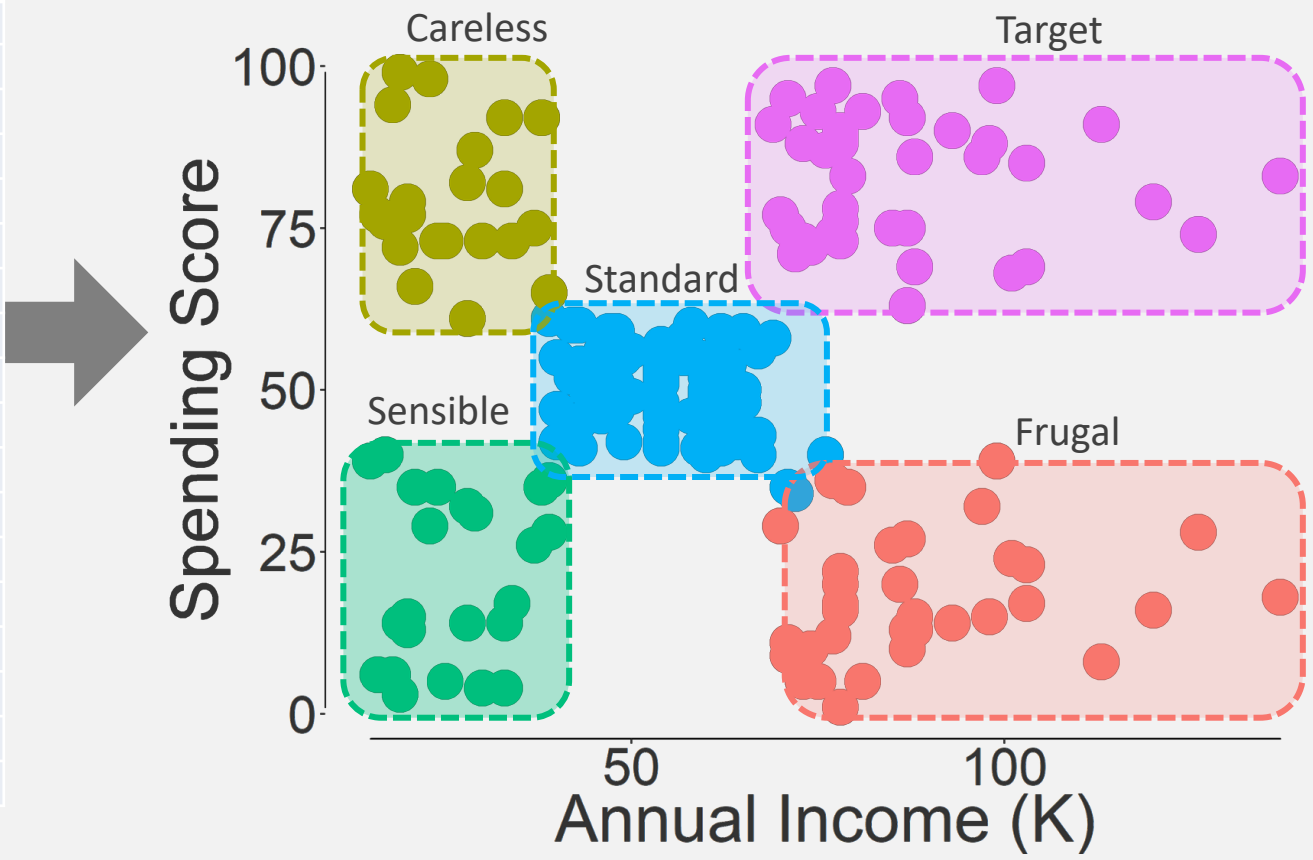
Educational Data Mining



Science/Biology

Example: Clustering Mall Customers

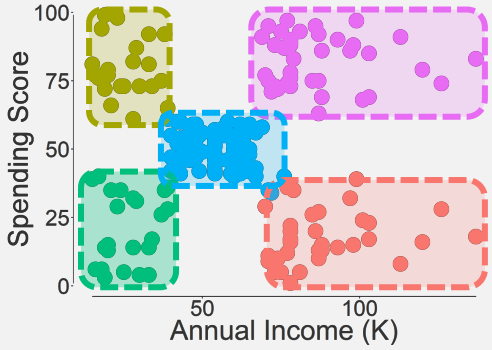
ID	Annual Income	Spend Score	Cluster ID
1	15	39	1
2	16	6	1
3	16	77	2
4	17	40	1
5	17	76	2
6	18	6	1
7	18	94	2
8	19	3	1
9	19	72	2
10	19	14	1
11	19	99	2
12	20	15	1
13	20	77	2
14	20	13	2
15	20	79	1
16	21	35	1
17	21	66	2
18	23	29	1
	...		



Clustering Beyond 2-D Data

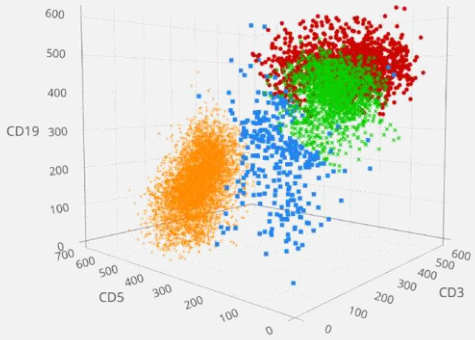
- Normally have more than two features in your data
- No problem! Just harder to visualize

2 Features



2-D plot

3 Features



3-D plot

4+ Features

Hard to visualize

No plot

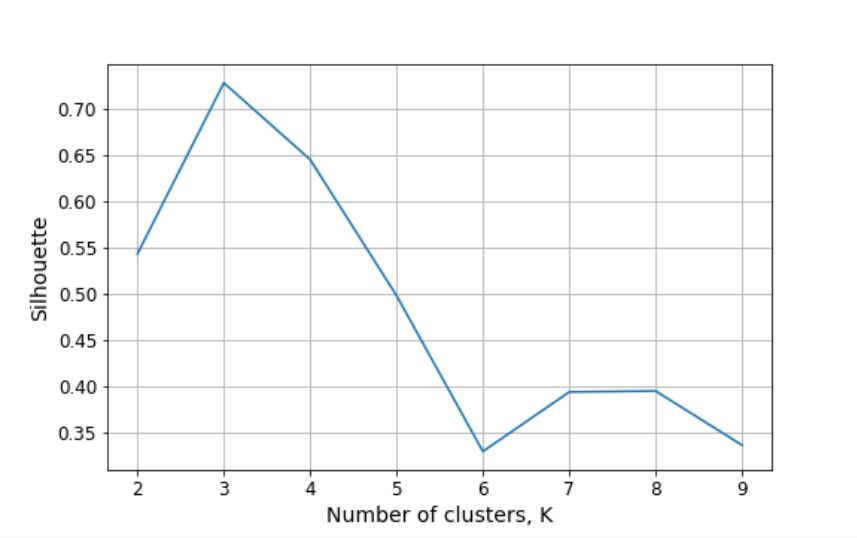
Might try PCA, t-SNE

But not worth it

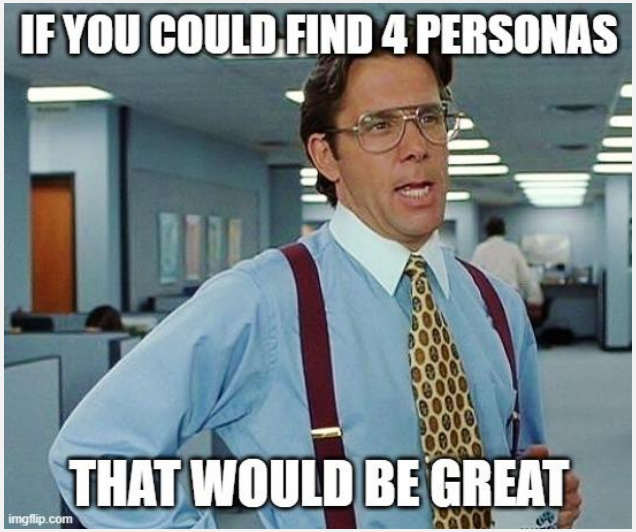
How Many Clusters?

- Algorithms need help figuring how many clusters are in the dataset
  - K-means: *K*
  - Hierarchical: *break point*
  - DBSCAN: *eps*, *minpts*
  - GMM: *K*
- How do you know?

Option 1: hyperparameter tuning



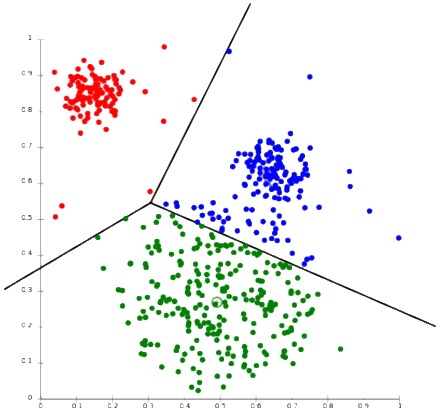
Option 2: Business Requirements



ALGORITHMS

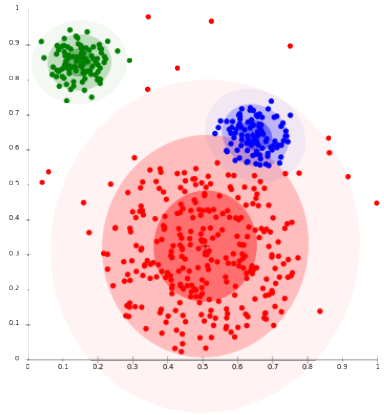
Popular Clustering Algorithms

Centroid Models



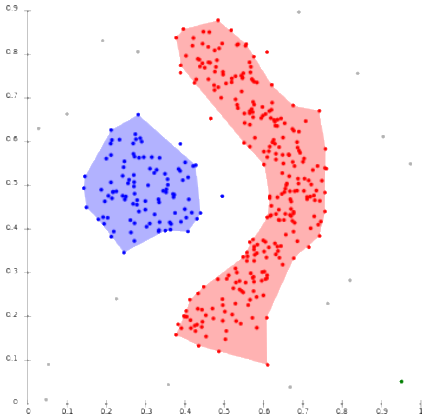
k-means, k-mediods, k-medians, k-means++

Distribution Models



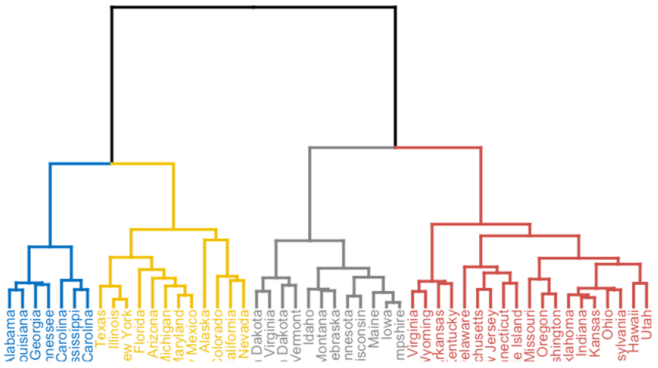
Gaussian Mixture Models

Density Models



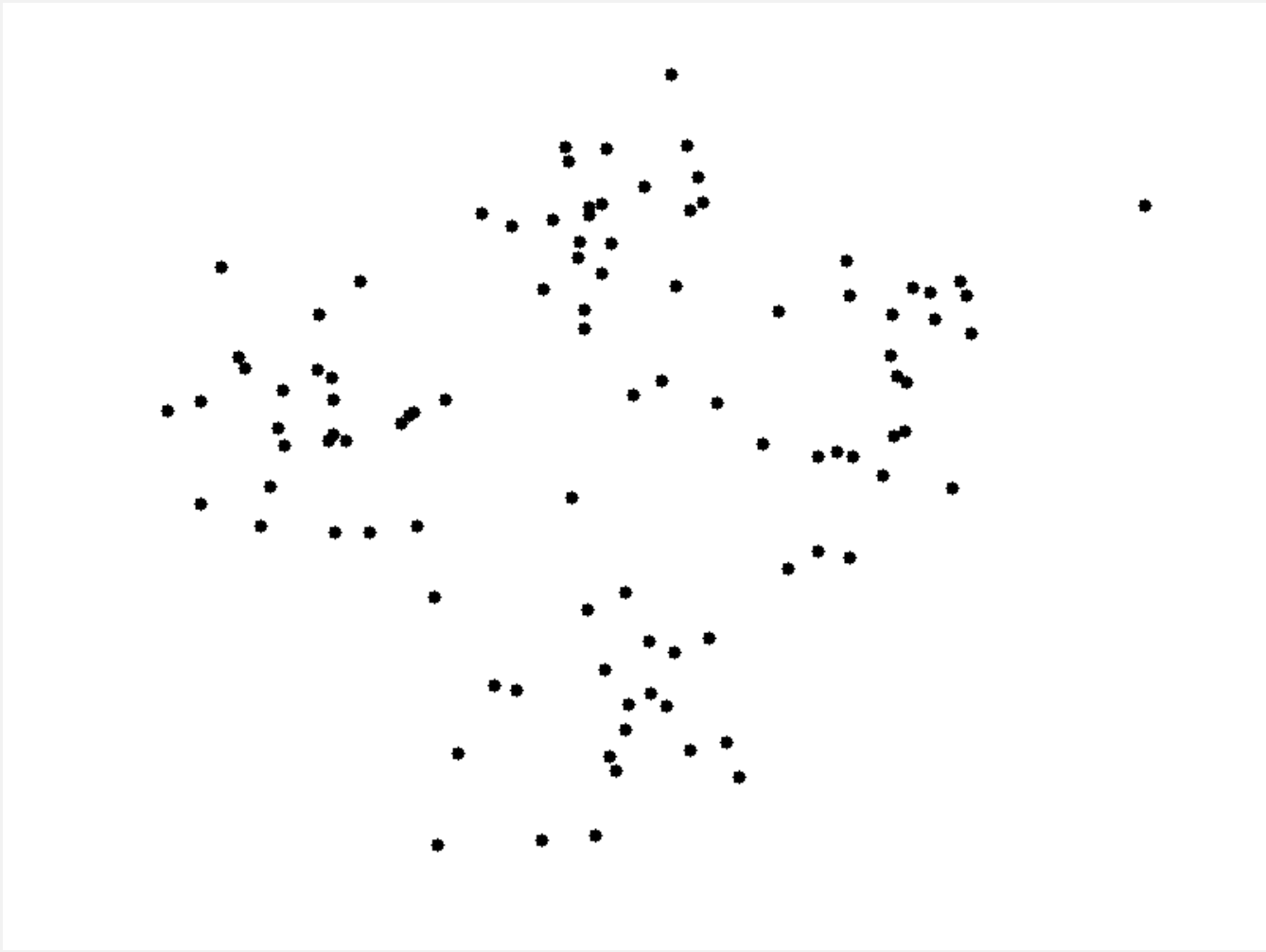
DBSCAN, OPTICS

Connectivity Models



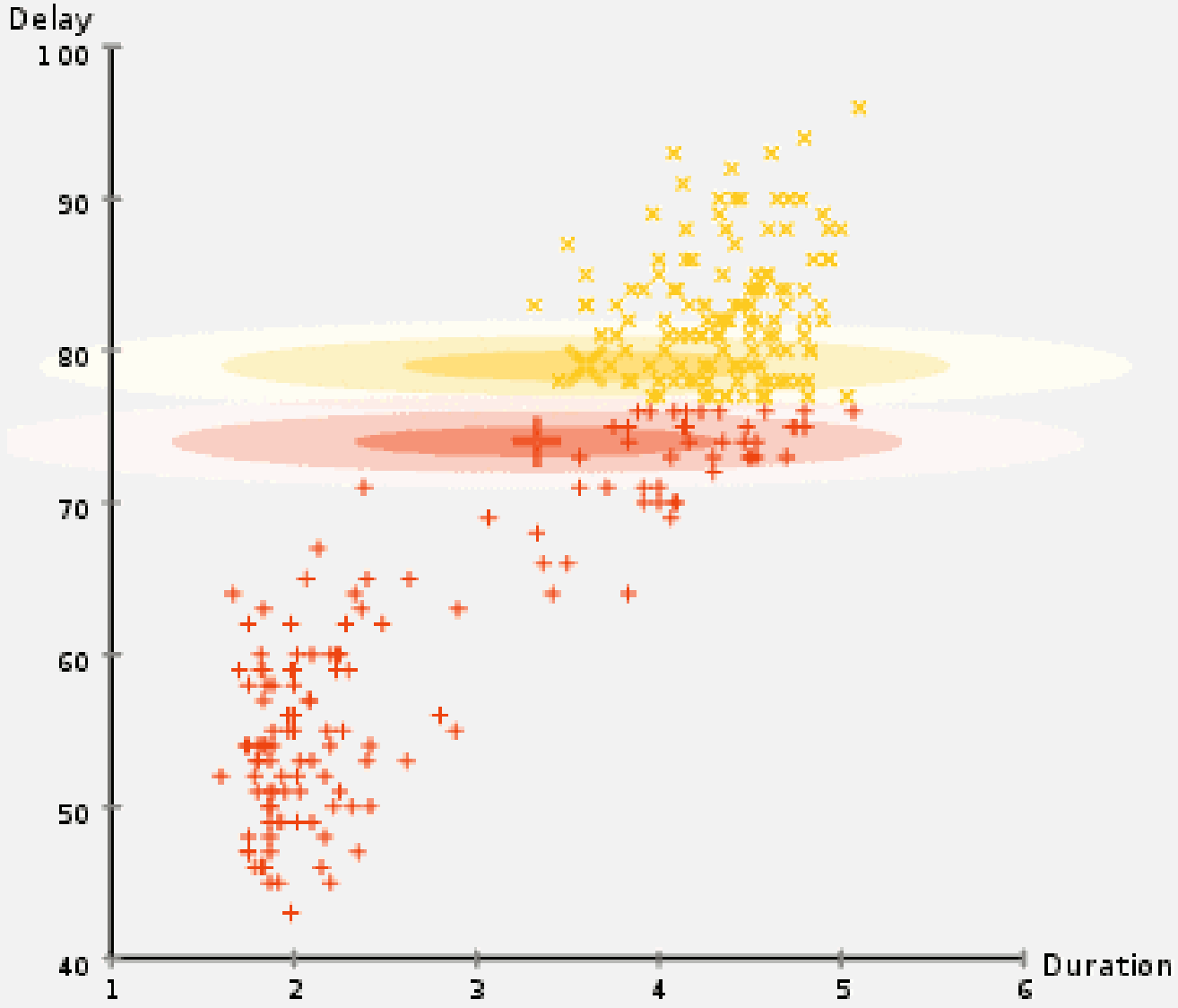
Agglomerative Hierarchical, DIANA

K-Means (Centroid)



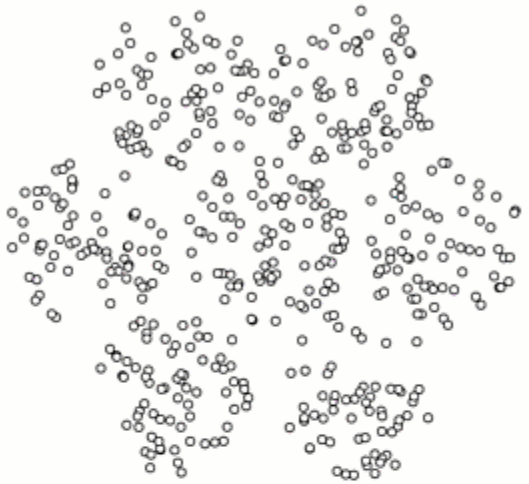
Source: http://shabal.in/visuals/kmeans/6.html

GMM (Distribution)



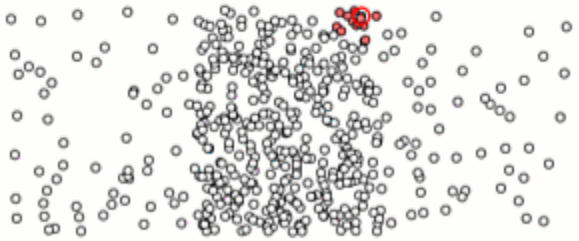
DBSCAN (Density)

Globs



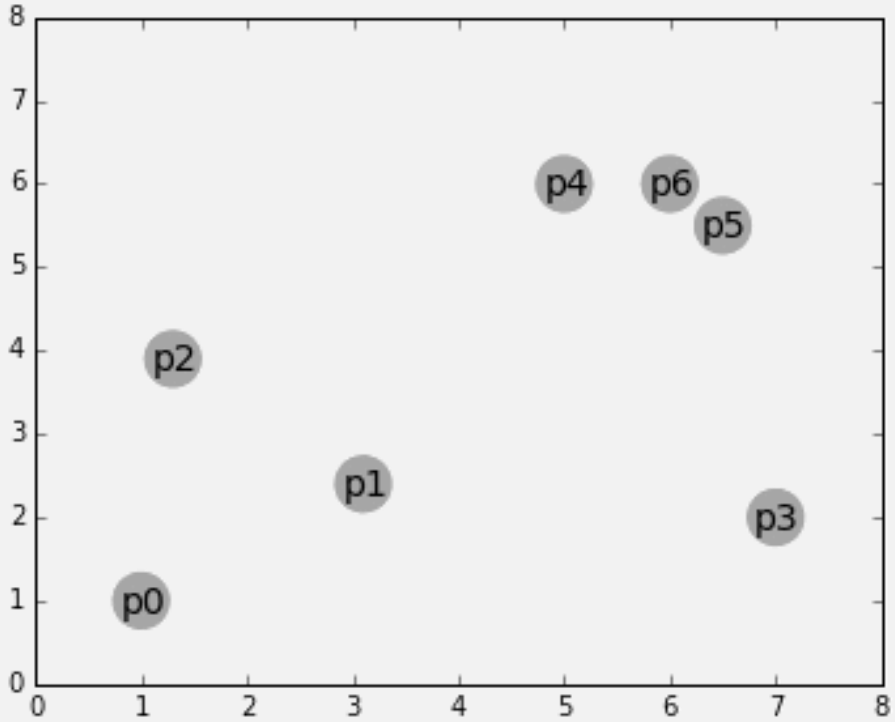
Packed

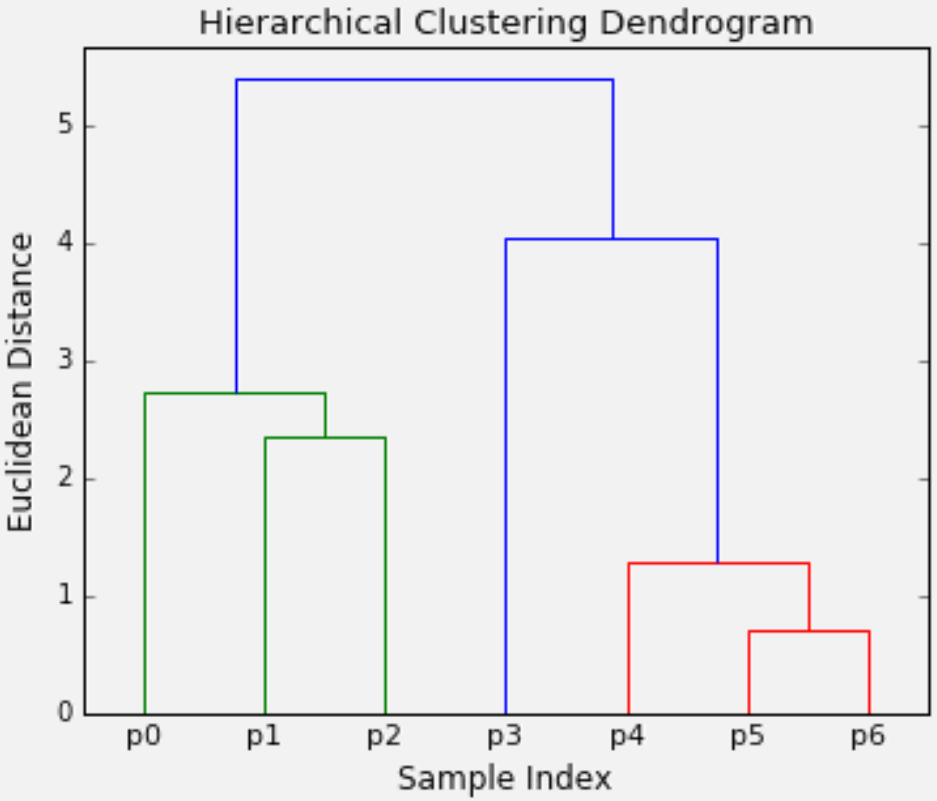
Smiley Face



Uniform

Agglomerative Hierarchical (Connectivity)





Further Details

https://stream.queensu.ca/Watch/e6HKg75W

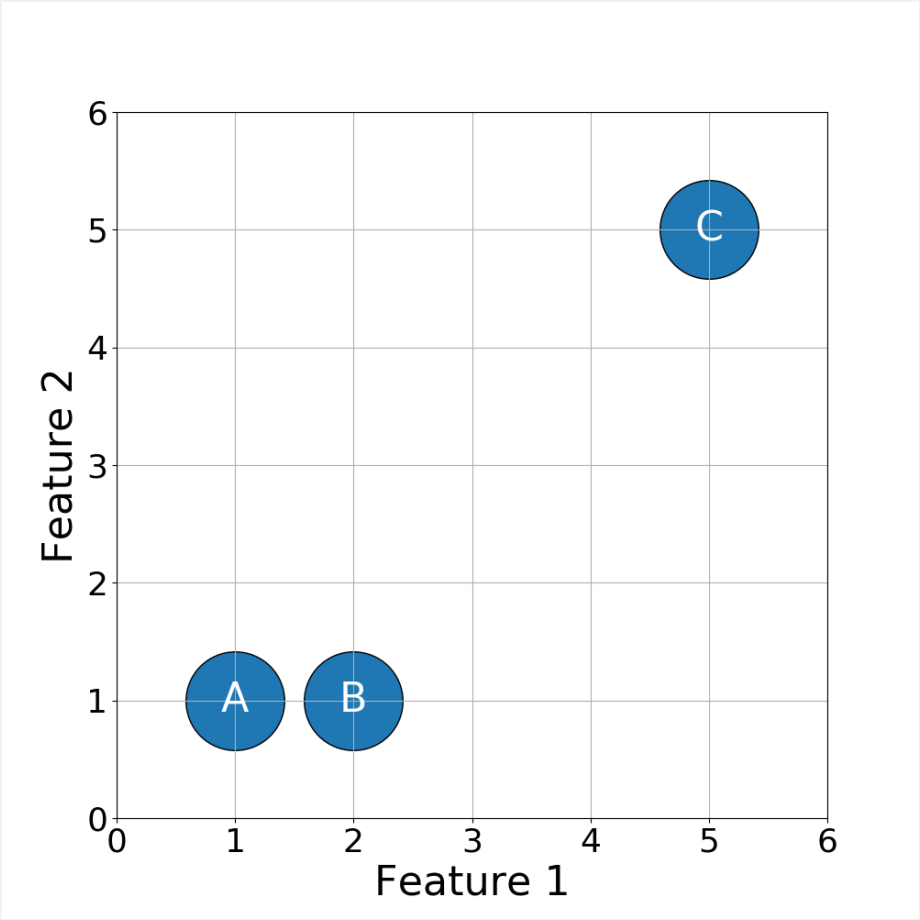
https://stream.queensu.ca/Watch/Lk28BmZb

https://stream.queensu.ca/Watch/j6P4ZiTw

DISTANCE METRICS

Distance Metrics

- Some clustering algorithms require you to specify a distance metric
- ***Distance metrics*** measure how "far apart" two instances are from each other
  - Equivalently: how "close" they are (*similarity metrics*)



ID	Feature 1	Feature 2
A	1.0	1.0
B	2.0	1.0
C	5.0	5.0

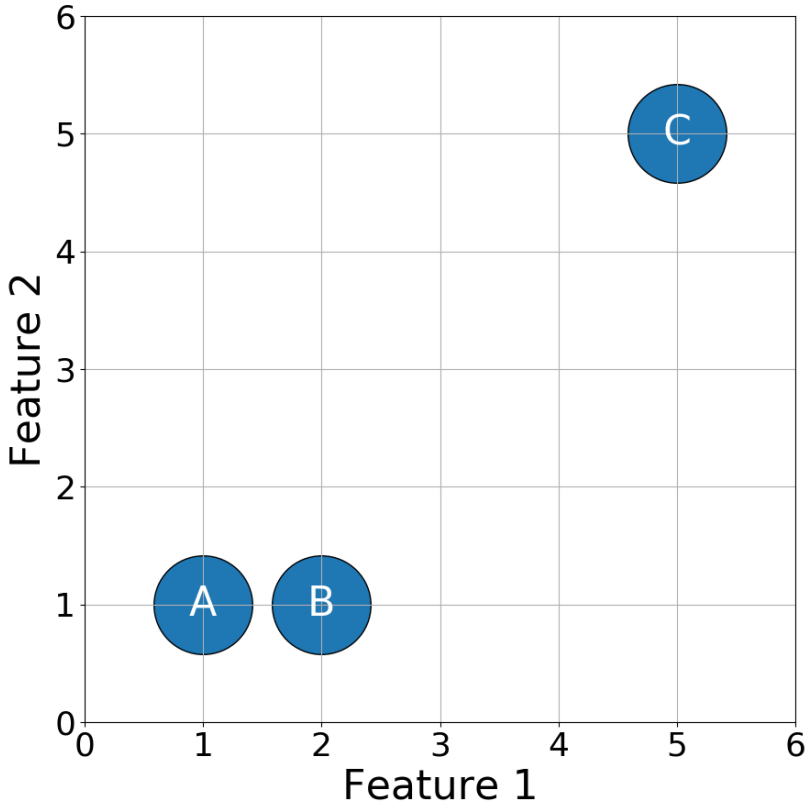
Common distance metrics:

- Euclidean
- Cosine
- Manhattan
- Chebyshev
- Canberra
- Pearson Correlation
- Hamming
- Jaccard

Euclidean Distance

Length of the "straight line" between two instances

$$dist_{euc}(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$



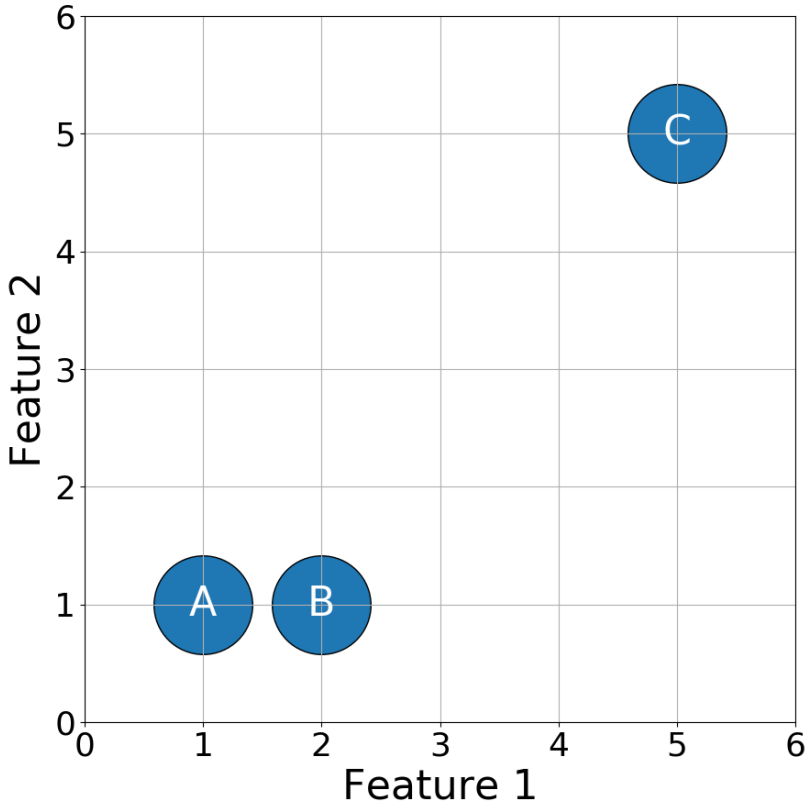
ID	Feature 1	Feature 2
A	1.0	1.0
B	2.0	1.0
C	5.0	5.0

	A, B	A, C	B, C
Euclidean	1.00	5.66	5.00

Cosine Distance

Measures the cosine of the angle between two vectors

$$dist_{cos}(A, B) = 1 - \frac{\sum_{i=0}^{n} A_i B_i}{\sqrt{\sum_{i=0}^{n} A_i^2} \sqrt{\sum_{i=0}^{n} B_i^2}}$$



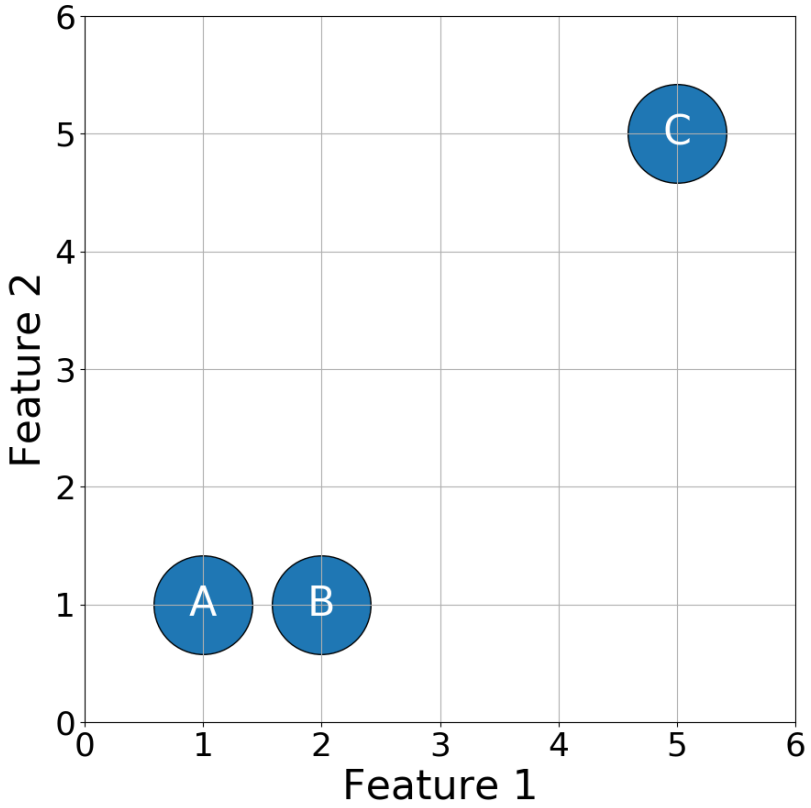
ID	Feature 1	Feature 2
A	1.0	1.0
B	2.0	1.0
C	5.0	5.0

	A, B	A, C	B, C
Euclidean	1.00	5.66	5.00
Cosine	0.05	0.00	0.05

Manhattan Distance

Length of "right angled" lines between two points

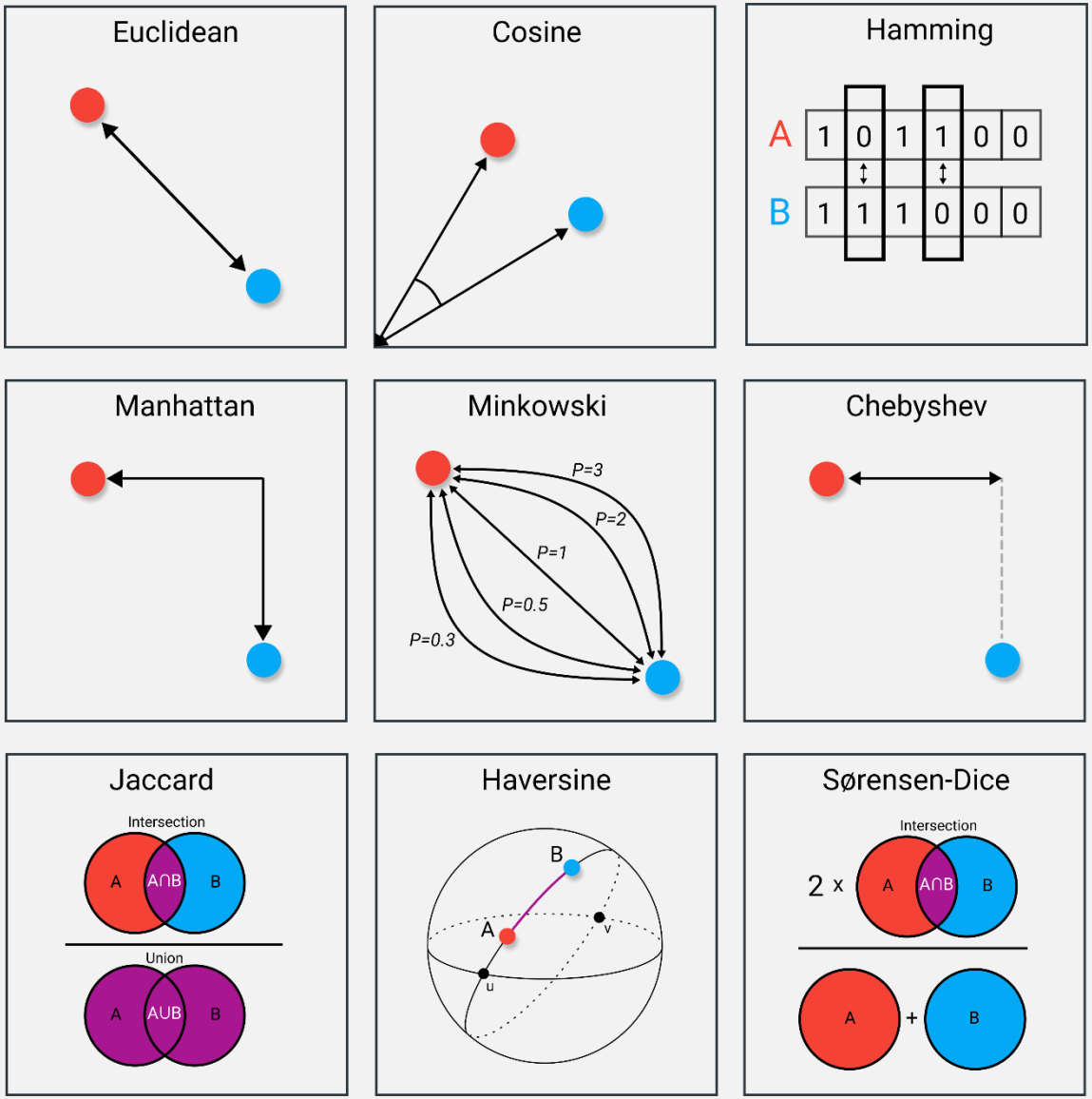
$$dist_{man}(A, B) = \sum_{i=1}^{n} |A_i - B_i|$$



ID	Feature 1	Feature 2
A	1.0	1.0
B	2.0	1.0
C	5.0	5.0

	A, B	A, C	B, C
Euclidean	1.00	5.66	5.00
Cosine	0.05	0.00	0.05
Manhattan	1.00	8.00	7.00

Which Distance Metric is Best?



Source: [Medium]

Which Distance Metric is Best?

- Depends on data shape/characteristics
- Recent paper suggests:
  - Euclidean is good all-purpose (default in scikit-learn)
    - But not good with lots of features
  - Cosine is good when absolute value is not important
    - E.g., customer shopping patterns
    - E.g., in sciences: gene expressions
  - Others have niche use cases
- Trial-and-error is often used

2015 PLOS Paper Guide

Distance Measure	Equation	Time complexity	Advantages	Disadvantages	Applications
Euclidean Distance	$d_{euc} = \left[\sum_{i=1}^{n}(x_i - y_i)^2\right]^{\frac{1}{2}}$	O(n)	Very common, easy to compute and works well with datasets with compact or isolated clusters [27,31].	Sensitive to outliers [27,31].	*K*-means algorithm, Fuzzy *c*-means algorithm [38].
Average Distance	$d_{ave} = \left(\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2\right)^{\frac{1}{2}}$	O(n)	Better than Euclidean distance [35] at handling outliers.	Variables contribute independently to the measure of distance. Redundant values could dominate the similarity between data points [37].	*K*-means algorithm
Weighted Euclidean	$d_{we} = \left(\sum_{i=1}^{n}w_i(x_i - y_i)^2\right)^{\frac{1}{2}}$	O(n)	The weight matrix allows to increase the effect of more important data points than less important one [37].	Same as Average Distance.	Fuzzy *c*-means algorithm [38]
Chord	$d_{chord} = \left(2 - 2\frac{\sum_{i=1}^{n}x_i y_i}{\|x\|_2\|y\|_2}\right)^{\frac{1}{2}}$	O(3n)	Can work with un-normalized data [27].	It is not invariant to linear transformation [33].	Ecological resemblance detection [35].
Mahalanobis	$d_{mah} = \sqrt{(x-y)S^{-1}(x-y)^T}$	O(3n)	Mahalanobis is a data-driven measure that can ease the distance distortion caused by a linear combination of attributes [35].	It can be expensive in terms of computation [33]	Hyperellipsoidal clustering algorithm [30].
Cosine Measure	$\mathrm{Cosine}(x,y) = \frac{\sum_{i=1}^{n}x_i y_i}{\|x\|_2\|y\|_2}$	O(3n)	Independent of vector length and invariant to rotation [33].	It is not invariant to linear transformation [33].	Mostly used in document similarity applications [28,33].
Manhattan	$d_{man} = \sum_{i=1}^{n}(x_i - y_i)$	O(n)	Is common and like other Minkowski-driven distances it works well with datasets with compact or isolated clusters [27].	Sensitive to the outliers. [27,31]	*K*-means algorithm
Mean Character Difference	$d_{MCD} = \frac{1}{n}\sum_{i=1}^{n}\|x_i - y_i\|$	O(n)	*Results in accurate outcomes using the K-medoids algorithm.	*Low accuracy for high-dimensional datasets using K-means.	Partitioning and hierarchical clustering algorithms.
Index of Association	$d_{IOA} = \frac{1}{n}\sum_{i=1}^{n}\left\|\frac{x_i}{\sum_{i=1}^{n}x_i} - \frac{y_i}{\sum_{i=1}^{n}y_i}\right\|$	O(3n)	-	*Low accuracy using K-means and K-medoids algorithms.	Partitioning and hierarchical clustering algorithms.
Canberra Metric	$d_{canb} = \sum_{i=1}^{n}\frac{\|x_i - y_i\|}{(x_i + y_i)}$	O(n)	*Results in accurate outcomes for high-dimensional datasets using the K-medoids algorithm.	-	Partitioning and hierarchical clustering algorithms.
Czekanowski Coefficient	$d_{czekan} = 1 - \frac{2\sum_{i=1}^{n}min(x_i, y_i)}{\sum_{i=1}^{n}(x_i + y_i)}$	O(2n)	*Results in accurate outcomes for medium-dimensional datasets using the K-means algorithm.	-	Partitioning and hierarchical clustering algorithms.
Coefficient of Divergence	$d_{canb} = \left(\frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - y_i}{x_i + y_i}\right)^2\right)^{\frac{1}{2}}$	O(n)	*Results in accurate outcomes using the K-means algorithm.	-	Partitioning and hierarchical clustering algorithms.
Pearson coefficient	$Pearson(x,y) = \frac{\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}}$	O(2n)	*Results in accurate outcomes using the hierarchical single-link algorithm for high dimensional datasets.	-	Partitioning and hierarchical clustering algorithms.

*Points marked by asterisk are compiled based on this article's experimental results.

doi:10.1371/journal.pone.0144059.t001

Shirkhorshidi AS, Aghabozorgi S, Wah TY (2015) A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data.
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0144059

2015 PLOS Paper Results

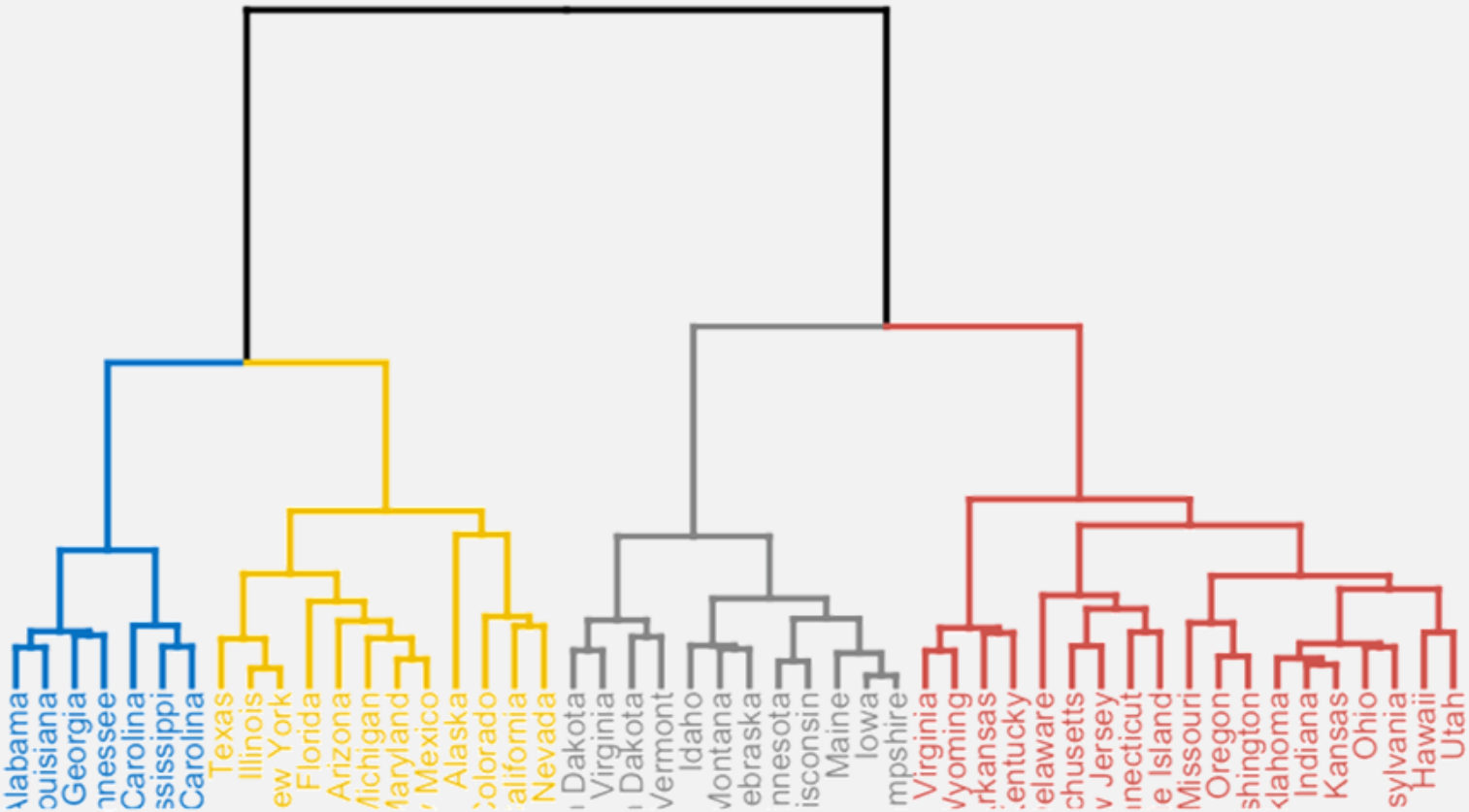
	Low Dimensional								Higher Dimentional							Overall Average
Dimensions	2	2	2	2	2	2	2	2	4	4	5	7	9	24	90	
	sensor_2	Aggregation	Compound	Flame	Pathbased	R15	Spiral	D31	Iris	sensor_4	_User_Mod	Seeds	Glass	sensor_24	Libras movement	
Euclidean	0.881	0.934	0.982	1.000	0.932	1.000	0.880	0.999	0.546	0.000	1.000	0.972	0.999	0.488	0.505	0.808
Average	0.912	0.934	0.972	1.000	0.932	1.000	0.881	0.995	0.546	0.091	1.000	0.968	1.000	0.655	0.941	0.855
Cosine	0.708	0.518	0.580	0.273	0.027	0.887	1.000	0.918	1.000	0.186	0.158	0.987	0.988	0.596	0.364	0.613
Chord	0.708	0.520	0.580	0.273	0.027	0.886	1.000	0.918	1.000	0.186	0.161	0.987	0.996	0.707	0.941	0.659
Mahalanobis	0.891	0.929	1.000	0.977	1.000	1.000	0.883	1.000	0.546	0.050	0.909	0.972	0.976	0.000	0.677	0.787
Canberra	0.942	0.910	0.915	0.844	0.815	0.999	0.817	0.994	0.874	1.000	0.793	0.939	0.654	0.754	0.166	0.828
CoeffDiv	0.934	0.855	0.958	0.000	0.797	0.998	0.839	0.994	0.915	0.730	0.711	0.795	0.649	0.853	0.587	0.774
Czekan	1.000	1.000	0.906	0.823	0.899	0.998	0.851	1.000	0.794	0.941	0.826	1.000	0.981	0.572	0.000	0.839
IndOfAssoc	0.715	0.519	0.576	0.227	0.014	0.883	0.986	0.918	0.957	0.274	0.159	0.997	0.977	0.482	0.417	0.607
Manhattan	0.901	0.929	0.953	0.977	0.914	0.998	0.901	0.998	0.515	0.860	0.801	0.963	0.980	0.279	0.495	0.831
MCharDiff	0.901	0.929	0.948	0.977	0.914	0.998	0.898	0.999	0.515	0.870	0.789	0.963	0.974	0.509	0.254	0.829
Pearson	0.000	0.000	0.000	0.147	0.000	0.000	0.000	0.000	0.000	0.438	0.000	0.000	0.000	1.000	1.000	0.172

Fig 3. K-means color scale table for normalized Rand index values (green represents the highest and it changes to red, which is the lowest Rand index value).

HIERARCHICAL

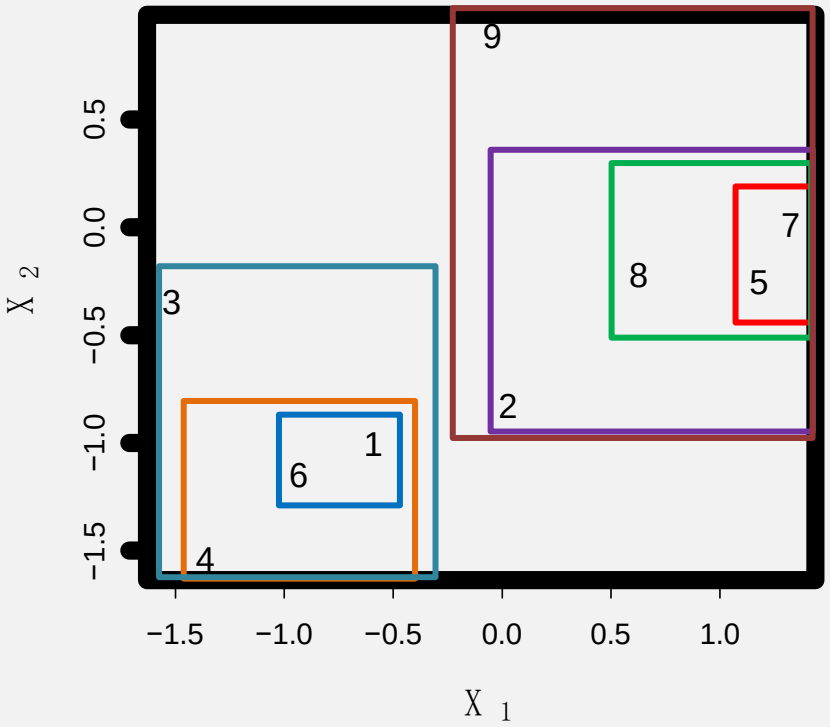
Hierarchical Clustering

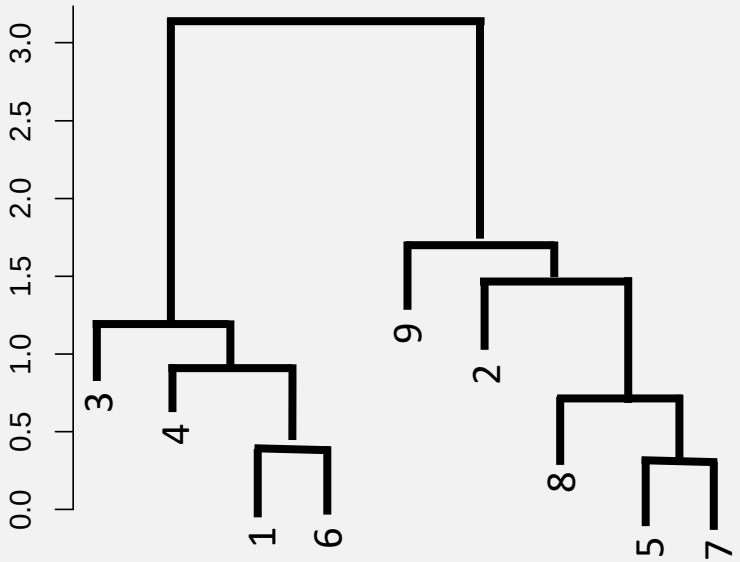
- aka, Agglomerative Clustering
- Iteratively joins instances to build a hierarchy of clusters
- Resulting hierarchy is shown as a dendrogram
- Hyperparameters: distance metric, linkage function



How Does it Work?

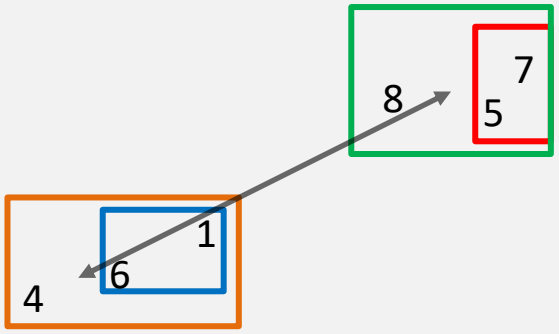
- Fuse together closest two instances/clusters
- Dendrogram "keeps track" of all the fuses
- Repeat





Wait…

- How do we define the distance (*linkage*) between clusters?

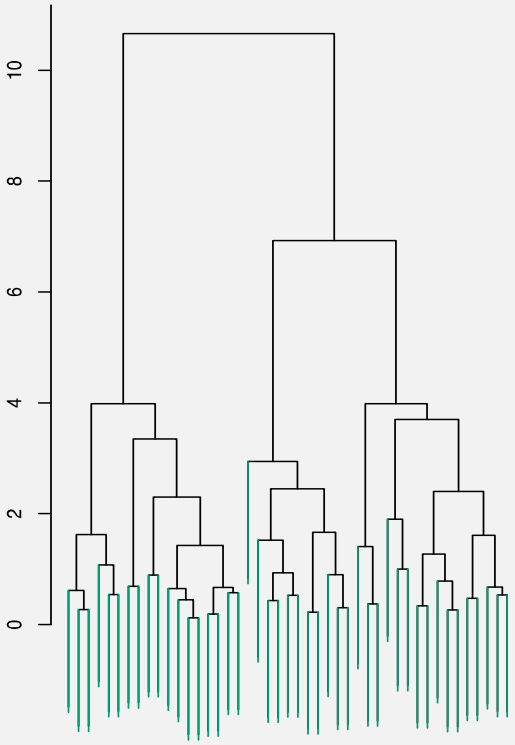


Ward Linkage

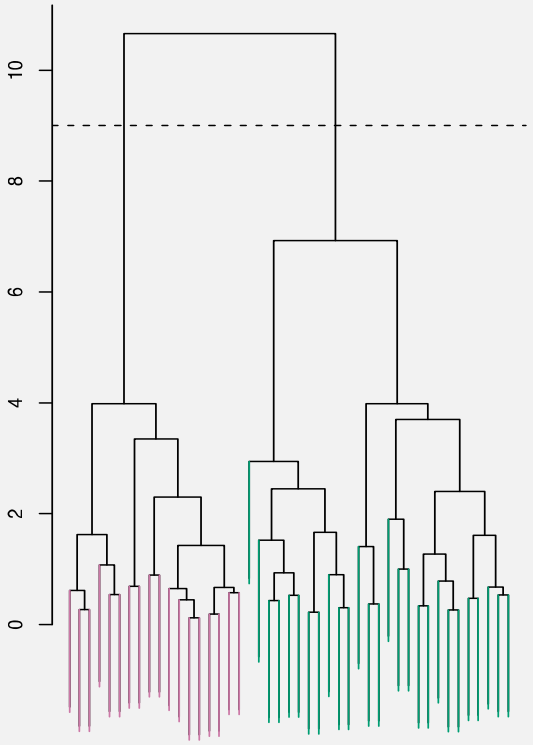
Distance between cluster centroids

How to Read a Dendrogram?

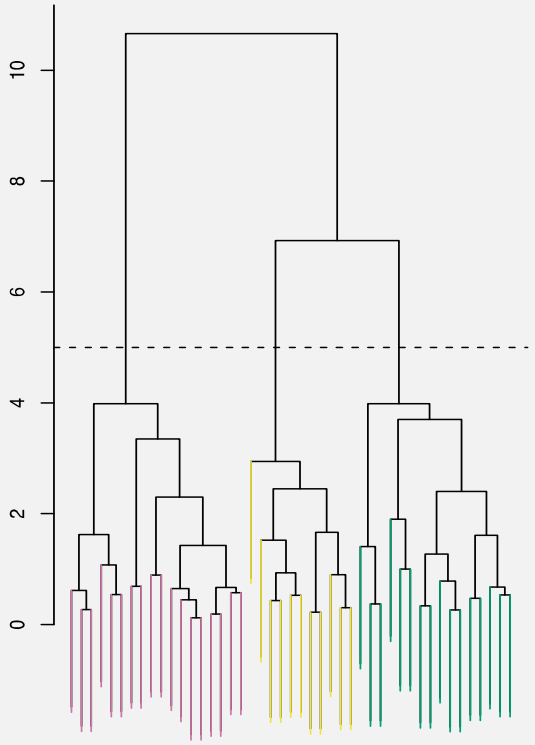
- To create clusters, we cut the dendrogram at a given ***break point***
- We can form any number of clusters depending on where we draw the break point



One Cluster

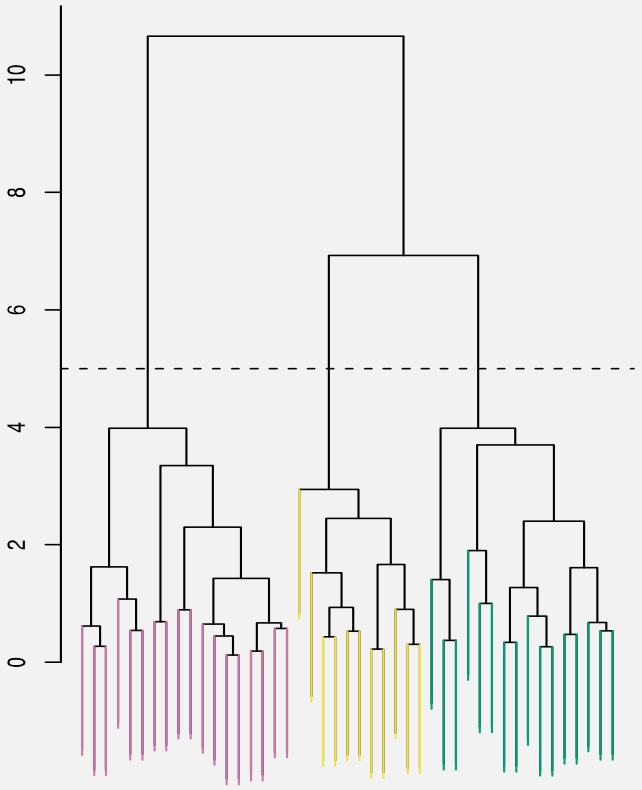


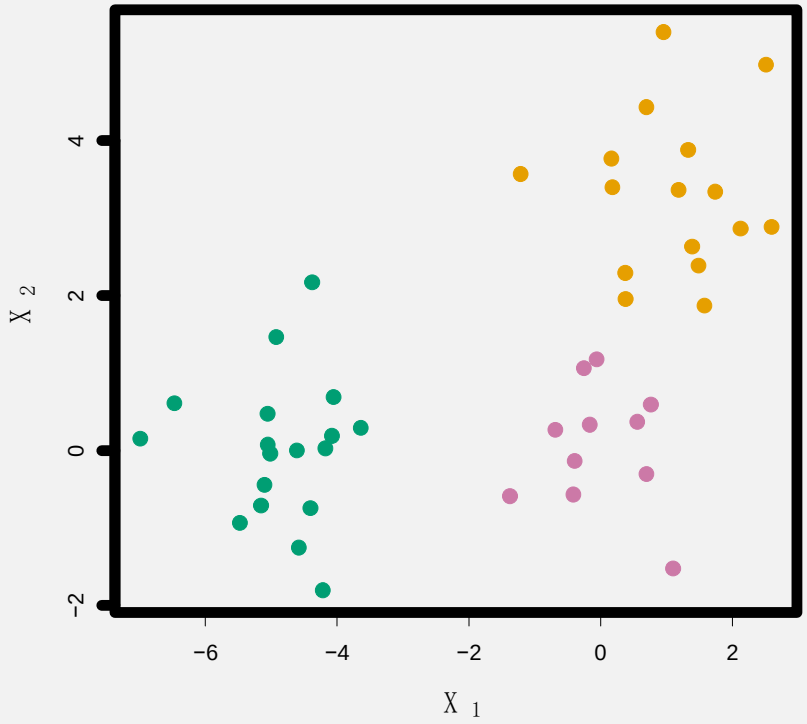
Two Clusters



Three Clusters

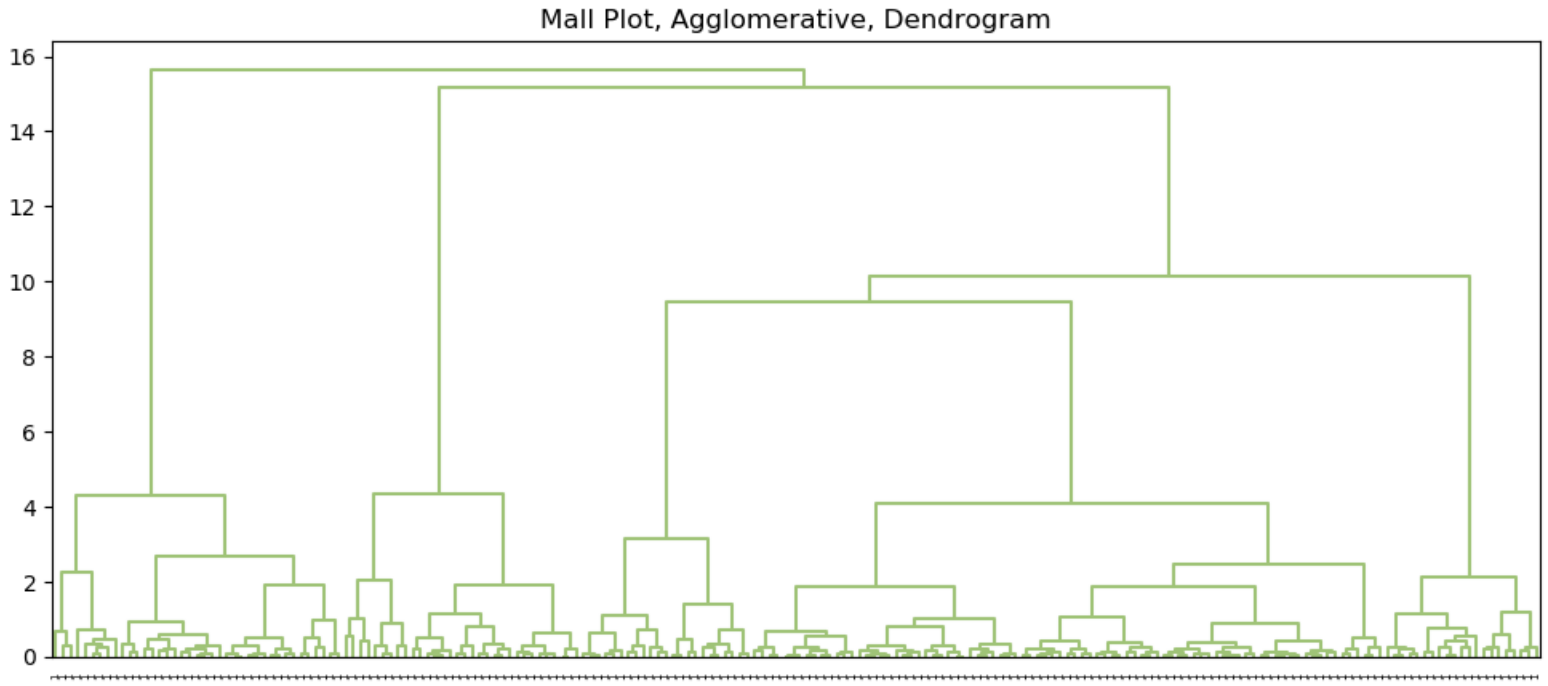
How to Read a Dendrogram?



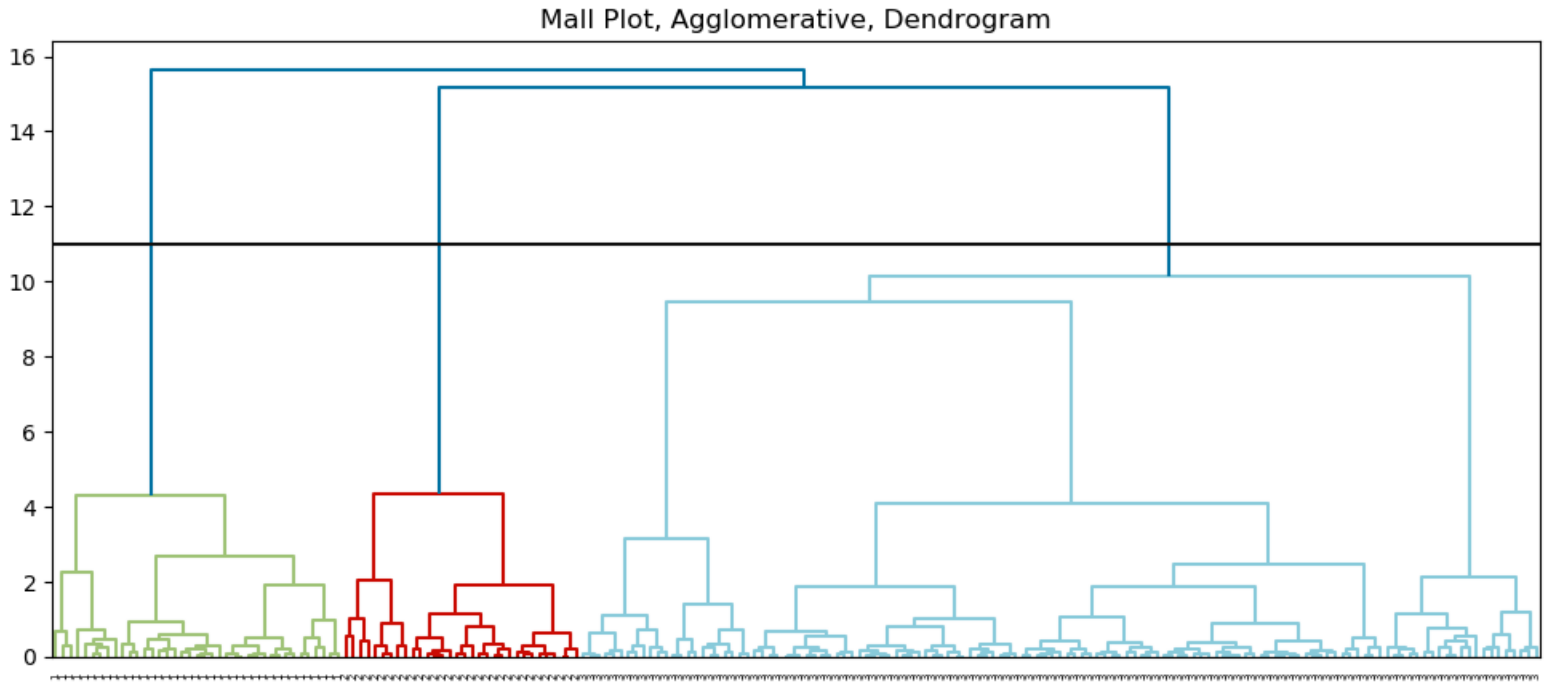


Example: Mall Data

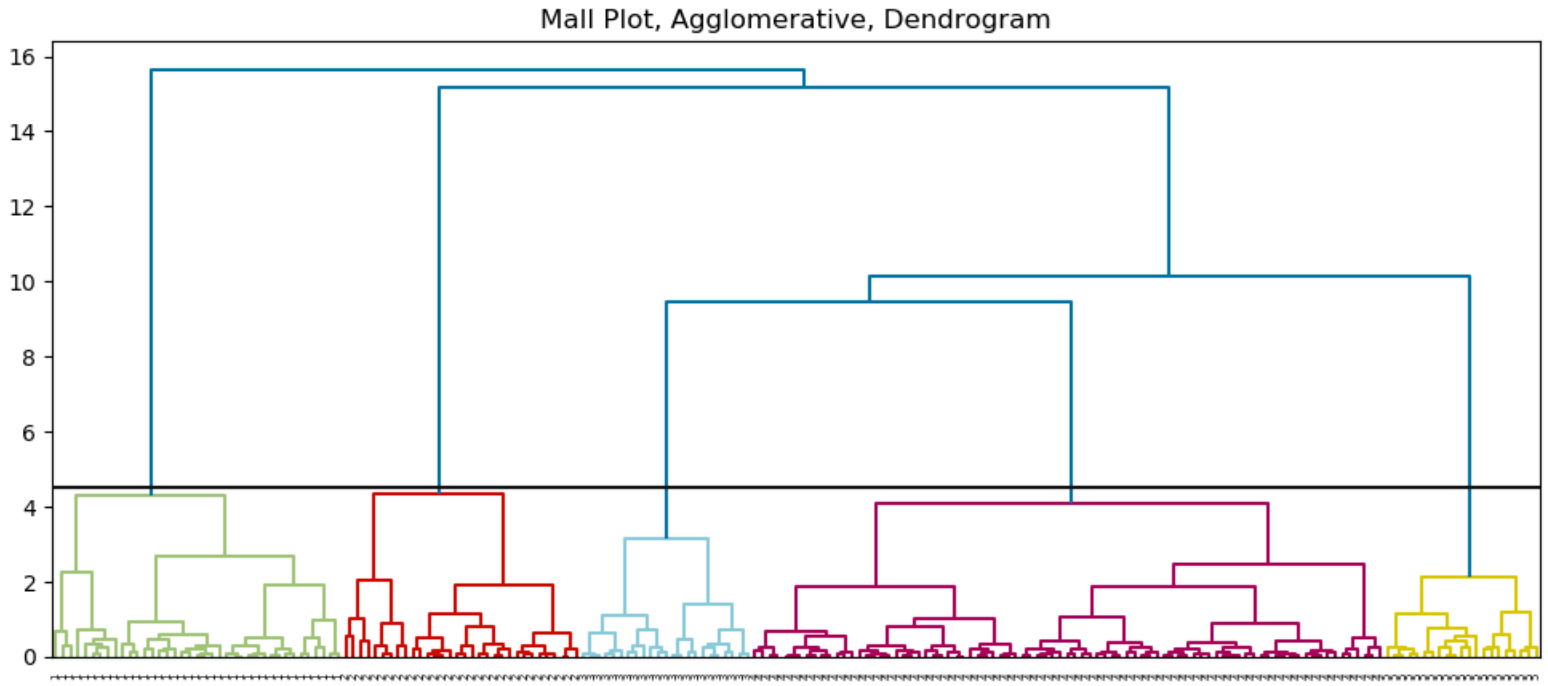
```
In [81]: aggl = scipy.cluster.hierarchy.linkage(X, method='ward', metric='euclidean', optimal_ordering=True)
```



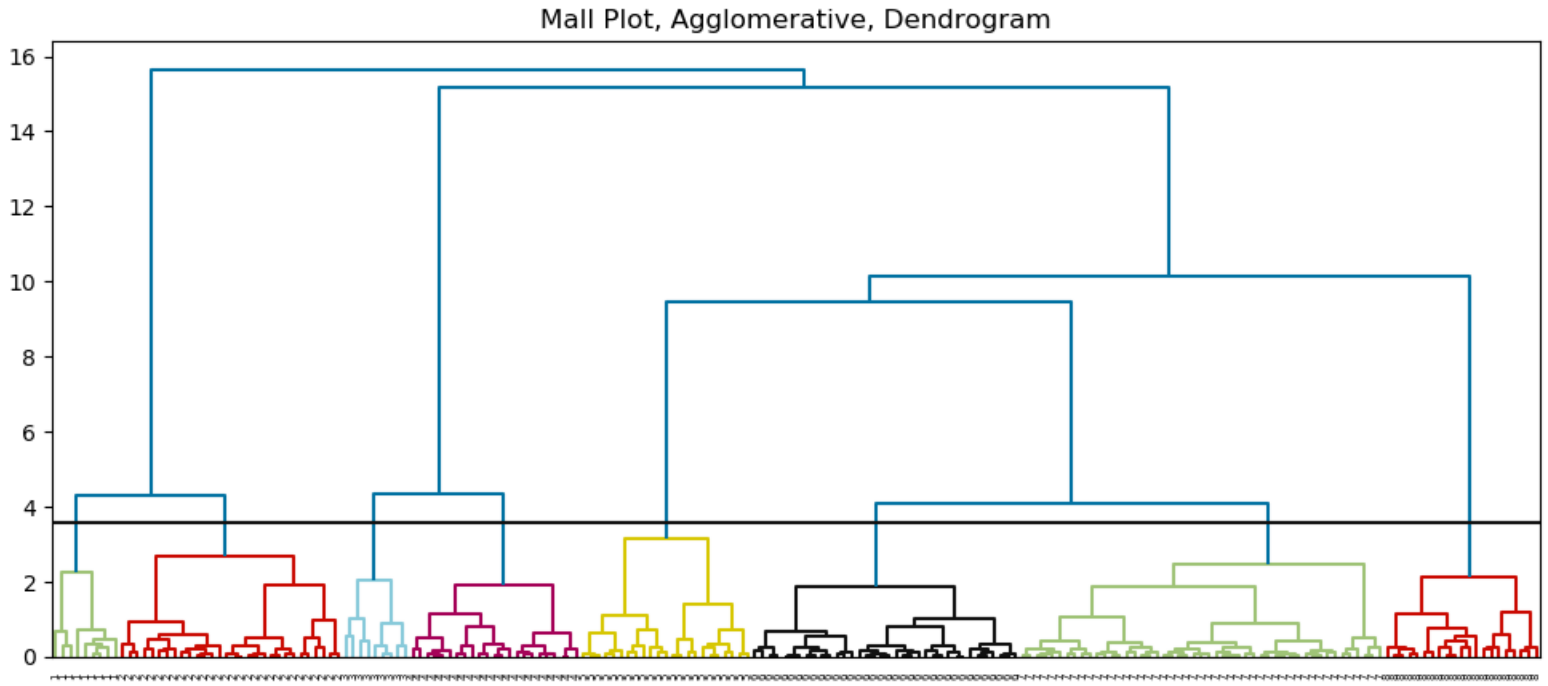
Example: Mall Data



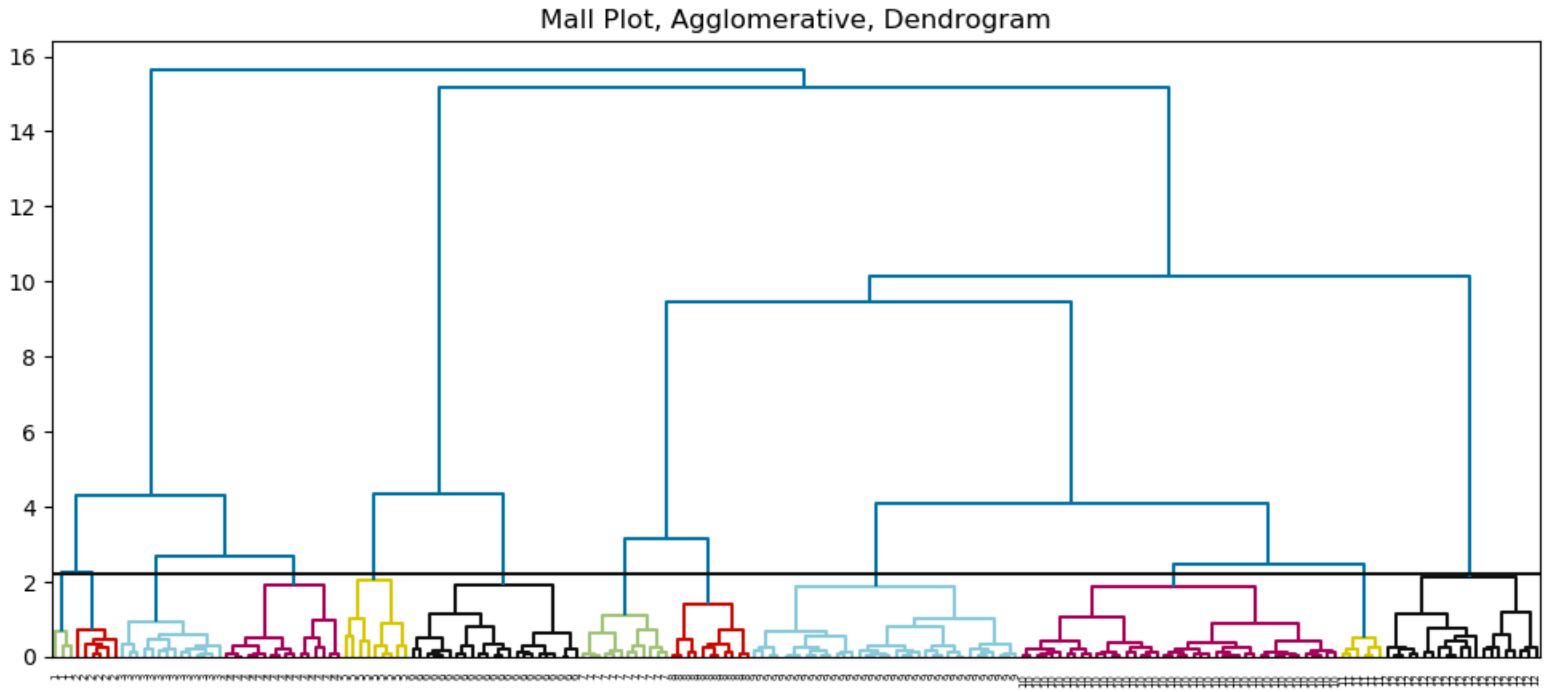
Example: Mall Data



Example: Mall Data



Example: Mall Data

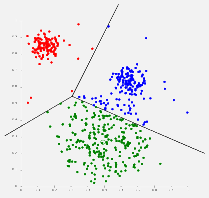
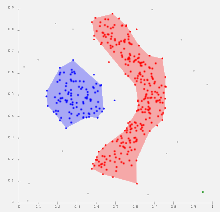
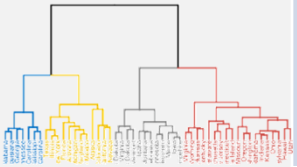
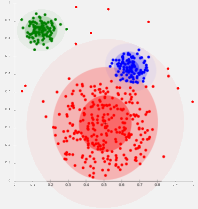


Smith
SCHOOL OF BUSINESS
Queen's University

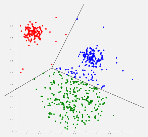
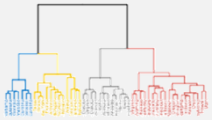
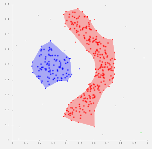
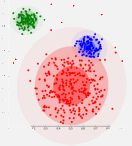
COMPARING ALGORITHMS

Uncle Steve's Clustering Algorithm Guide

Smith School of Business, Queen's University

Model	HParameters	Distance	Pros	Cons
 K-means	• K	Euclidean	✓ Fast ✓ Scalable	× Only works for "well-shaped" clusters × Sensitive to outliers × Sensitive to noise
 DBSCAN	• MinPts • Eps • Distance metric	Any	✓ Can detect arbitrary shapes ✓ Not sensitive to noise ✓ Supports outlier detection	× Hyper parameter selection is hard × Trouble identifying clusters of varying densities
 Hierarchical	• Distance metric • Linkage	Any	✓ Recovers a hierarchy ✓ Can view clusters at various granularities ✓ More flexible	
 GMM	• K	Mahalanobis	✓ Can handle "non-circular" cluster shapes ✓ Provides fuzzy clustering	× Harder to interpret results × Slow on large data

Uncle Steve's Ultimate Comparison Guide

	K-Means	Hierarchical	DBSCAN	GMM
Hyperparameters	K	Linkage, Distance	MinPts, Eps, Distance	K
Distance metric(s)	Euclidean	Any	Any	Mahalanobis
Training speed	Fast	Medium	Fast	Medium
Works well with small data	Yes	Yes	Yes	No
Robust to irrelevant features	No	No	No	No
Detects "circular" shapes	Yes	Yes	Yes	Yes
Detects "non-circular" shapes	No	Yes	Yes	No
Provides fuzzy memberships	No	No	No	Yes
Recovers hierarchy of clusters	No	Yes	Yes*	No
Robust to outliers	No	No	Yes	No

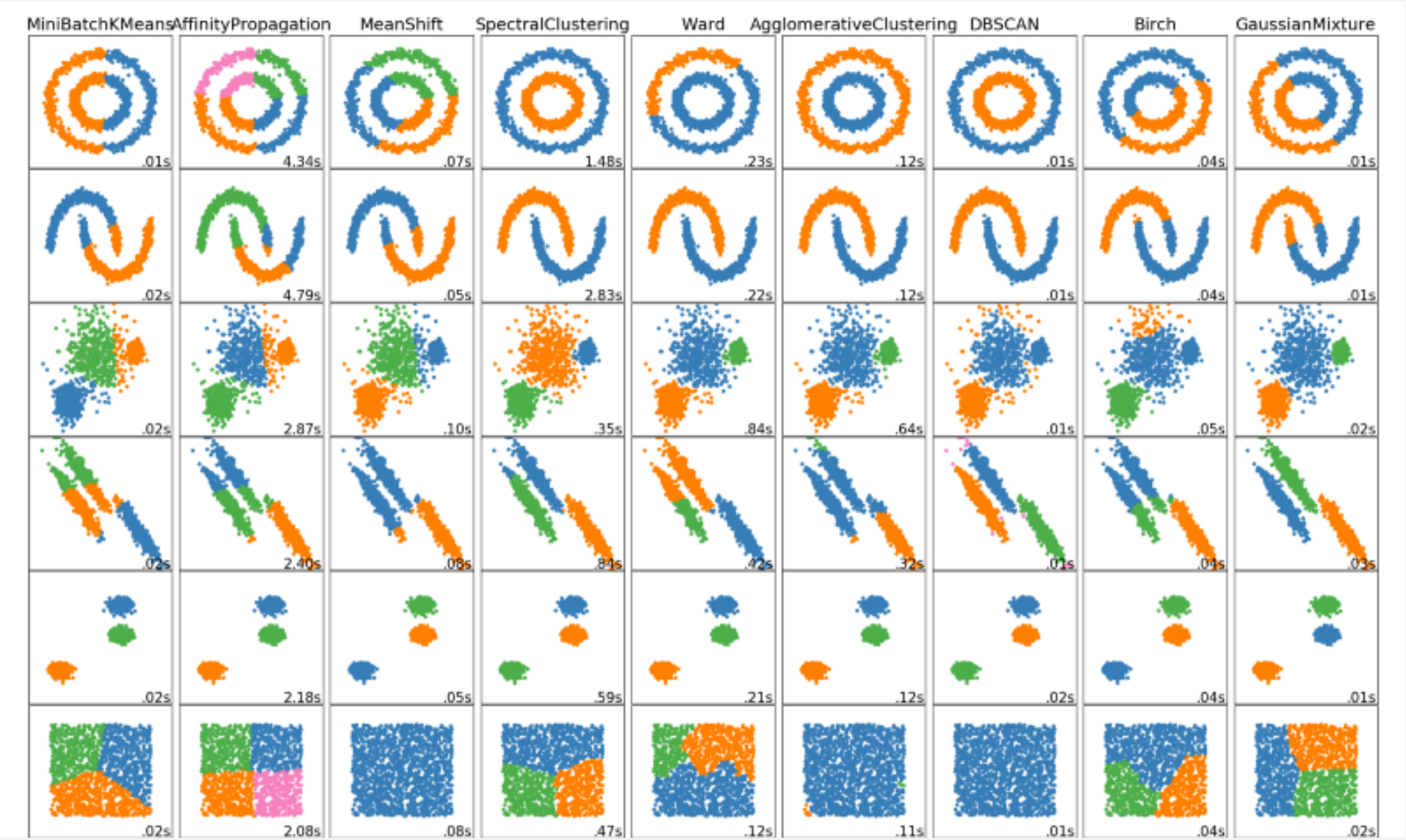
* HDBSCAN can.

Scikit-Learn's Table

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

Source: scikit-learn.org

Scikit-Learn's Visualization of Algorithms



Source: scikit-learn.org

INTERPRETING CLUSTERS

How to Interpret the Clusters

As a human, how do you interpret the clusters?

	Male	Age	AnnualIncome	SpendingScore	Cluster ID
0	True	19	15	39	2
1	True	21	15	81	2
2	False	20	16	6	5
3	False	23	16	77	2
4	False	31	17	40	5
5	False	22	17	76	2
6	False	35	18	6	5
7	False	23	18	94	2
8	True	64	19	3	4

- 4 features, 200 instances
- Hierarchical clustering (Euclidean, Ward, K=5)

- **Solution 1**: Look at cluster means
- **Solution 2**: Relative importance plot
- **Solution 3**: Find exemplars

Exercise: Look at Cluster Means/Stats

Cluster 1:
Number of Instances: 39

	Min	Mean	Max	Variance	Skewness	Kurtosis
Male	0.00	0.46	1.00	0.95	0.52	-0.54
Age	27.00	32.69	40.00	39.85	44.55	25.92
AnnualIncome	69.00	86.54	137.00	70.72	96.41	96.51
SpendingScore	63.00	82.13	97.00	53.60	47.41	19.20

Cluster 2:
Number of Instances: 39

	Min	Mean	Max	Variance	Skewness	Kurtosis
Male	0.00	0.38	1.00	0.92	0.69	-0.42
Age	18.00	26.15	40.00	41.68	46.73	28.13
AnnualIncome	15.00	43.77	76.00	73.21	58.64	25.12
SpendingScore	13.00	58.97	99.00	63.39	57.67	41.55

Cluster 3:
Number of Instances: 61

	Min	Mean	Max	Variance	Skewness	Kurtosis
Male	0.00	0.55	1.00	0.96	0.35	-0.54
Age	19.00	41.45	59.00	47.81	34.55	32.56
AnnualIncome	71.00	89.09	137.00	70.89	93.18	86.22
SpendingScore	1.00	16.18	39.00	53.65	63.71	44.61

Cluster 4:
Number of Instances: 33

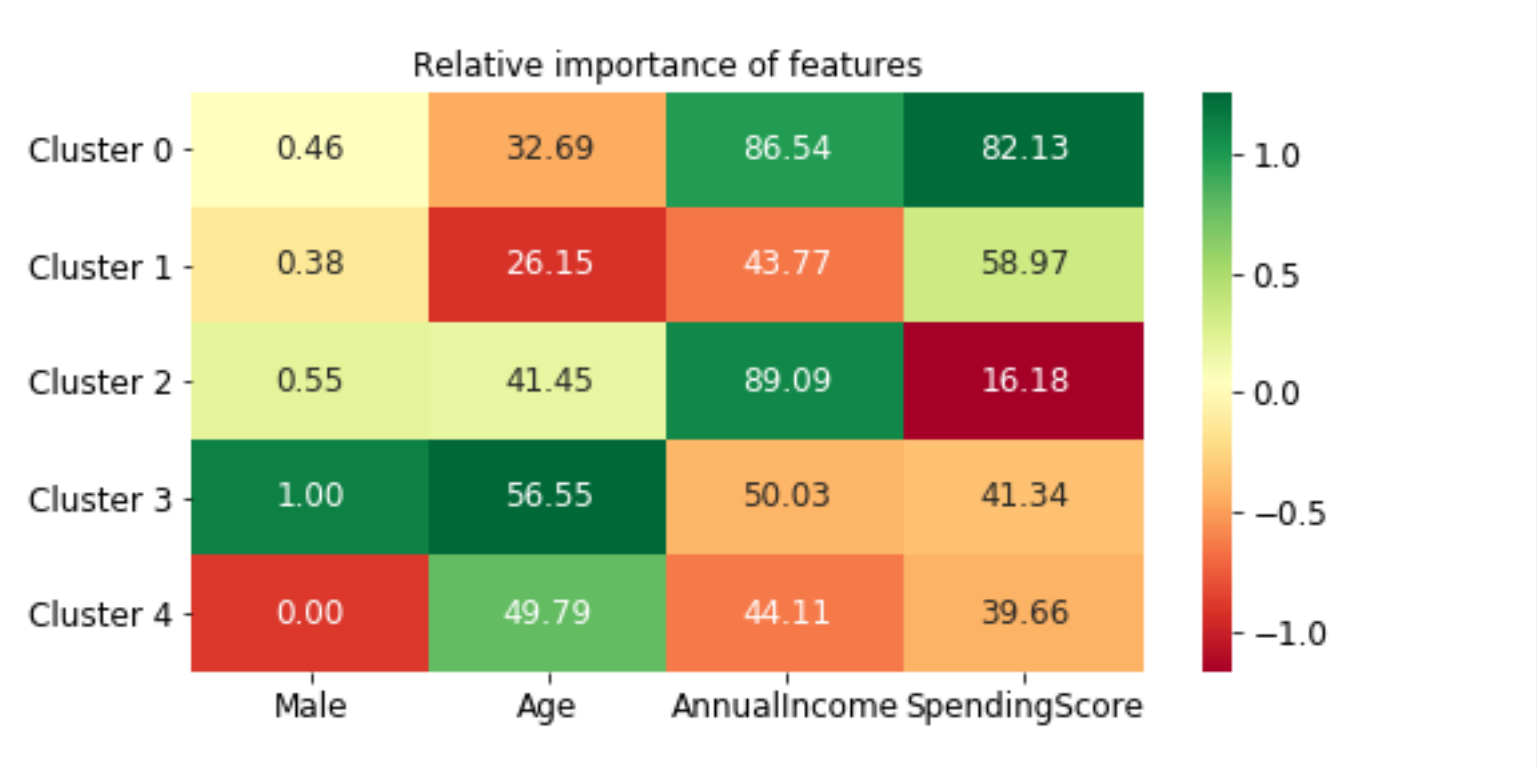
	Min	Mean	Max	Variance	Skewness	Kurtosis
Male	1.00	1.00	1.00	0.44	0.44	-1.05
Age	38.00	56.55	70.00	45.58	36.55	22.07
AnnualIncome	19.00	50.03	77.00	68.85	45.31	55.02
SpendingScore	3.00	41.34	60.00	60.31	15.55	72.17

Cluster 5:
Number of Instances: 29

	Min	Mean	Max	Variance	Skewness	Kurtosis
Male	0.00	0.00	0.00	0.44	-0.06	-0.55
Age	20.00	49.79	68.00	46.87	34.32	45.03
AnnualIncome	16.00	44.11	67.00	69.50	53.00	34.67
SpendingScore	5.00	39.66	59.00	60.41	28.06	38.79

Relative Importance Plots

- Greener = higher value than population
- Redder = lower value than population
- Yellow = same value as population



Exemplars

- Find a representative instance
- Use as exemplar

Cluster 1:

	CustomerID	Male	Age	AnnualIncome	SpendingScore
175	176	False	30	88	86

Cluster 2:

	CustomerID	Male	Age	AnnualIncome	SpendingScore
58	59	False	27	46	51

Cluster 3:

	CustomerID	Male	Age	AnnualIncome	SpendingScore
170	171	True	40	87	13

Cluster 4:

	CustomerID	Male	Age	AnnualIncome	SpendingScore
74	75	True	59	54	47

Cluster 5:

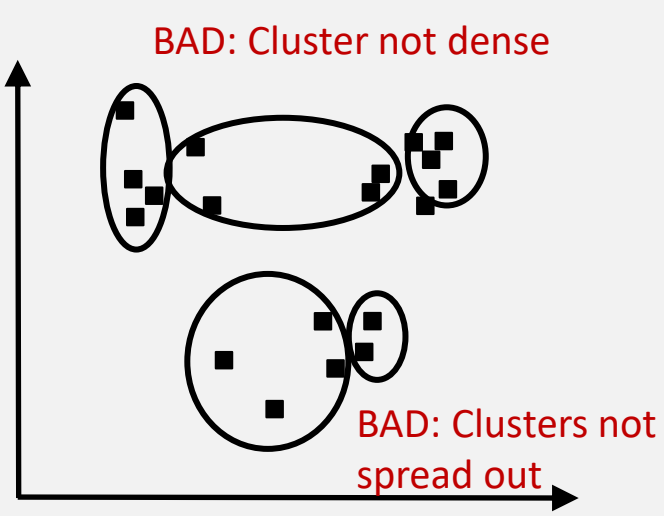
	CustomerID	Male	Age	AnnualIncome	SpendingScore
54	55	False	50	43	45

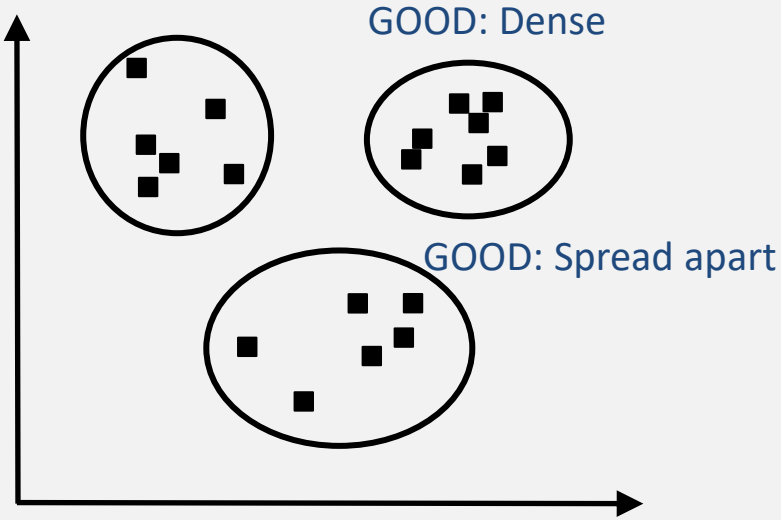
Interpreting Clusters

https://stream.queensu.ca/Watch/a8ACo7t3

EVALUATION

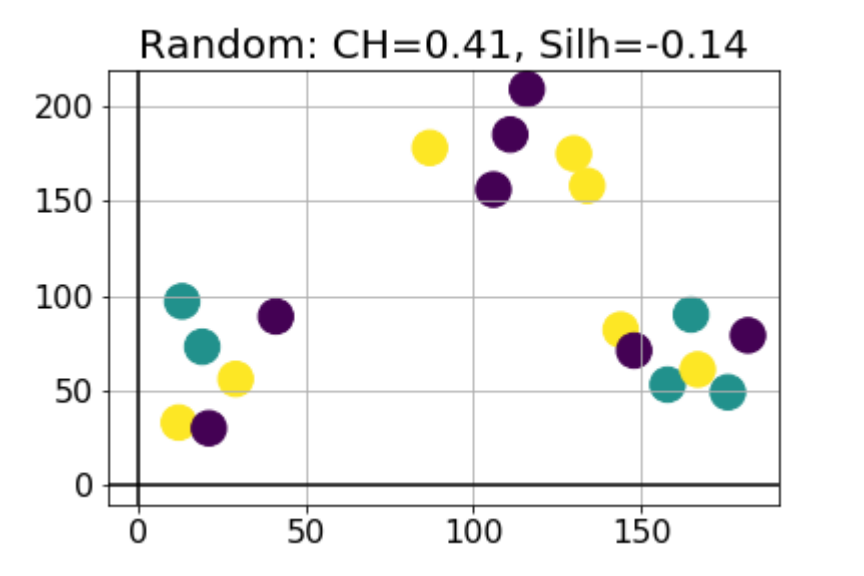
How to Assess/Evaluate Clusters

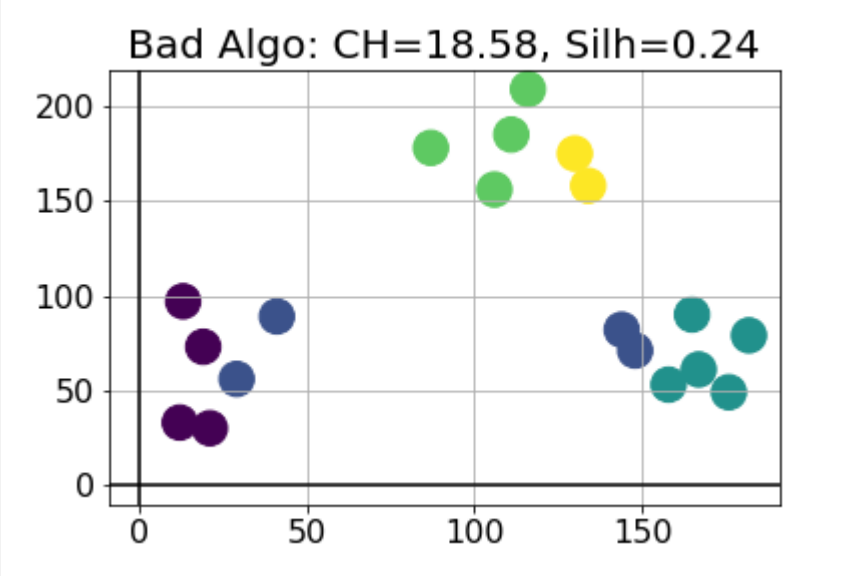


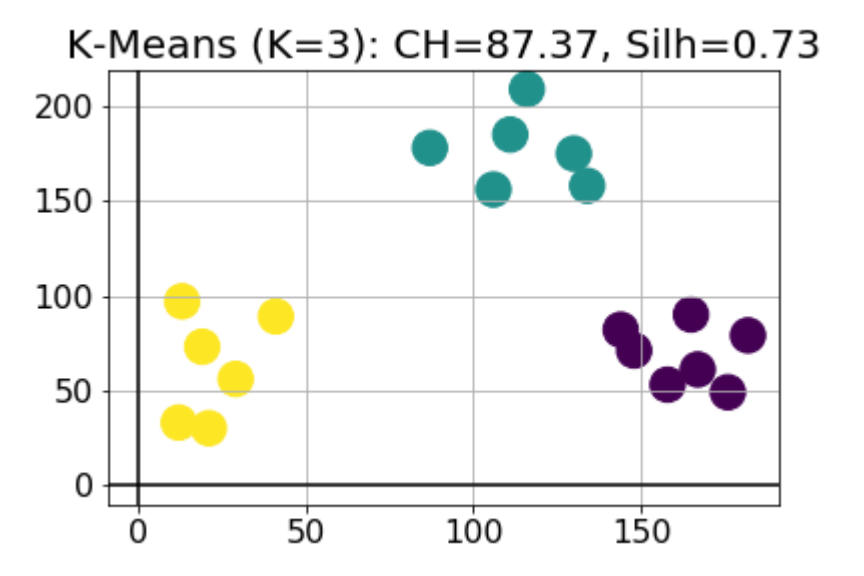


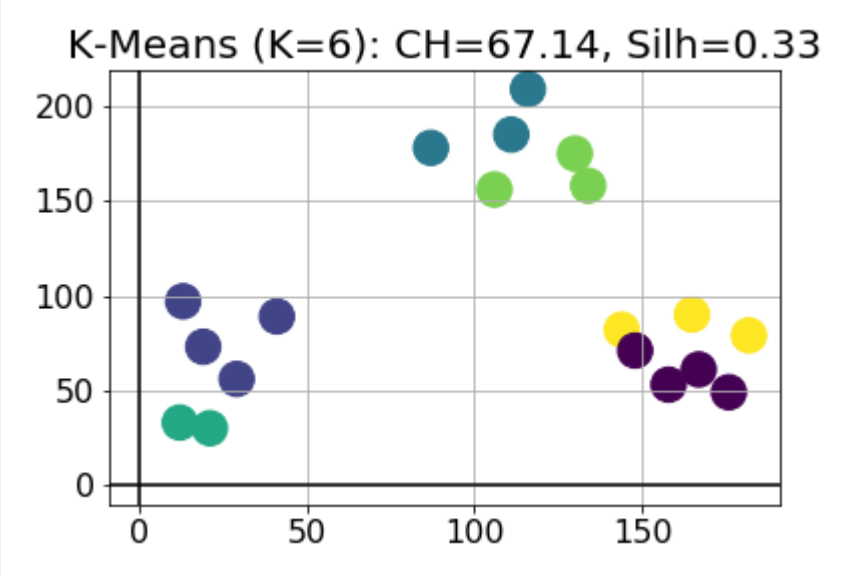
- ***Internal validation metrics:***
  - **Calinski-Harabasz Index**: Measures density of clusters
    - Intra-cluster density
    - Range of 0+
    - Higher = better
  - **Silhouette Coefficient**: Measures distance between clusters
    - Inter-cluster distance
    - Range of [-1, 1]
    - Higher = better; < 0 = bad

Examples


Random: CH=0.41, Silh=-0.14
200
150
100
50
0
0
50
100
150

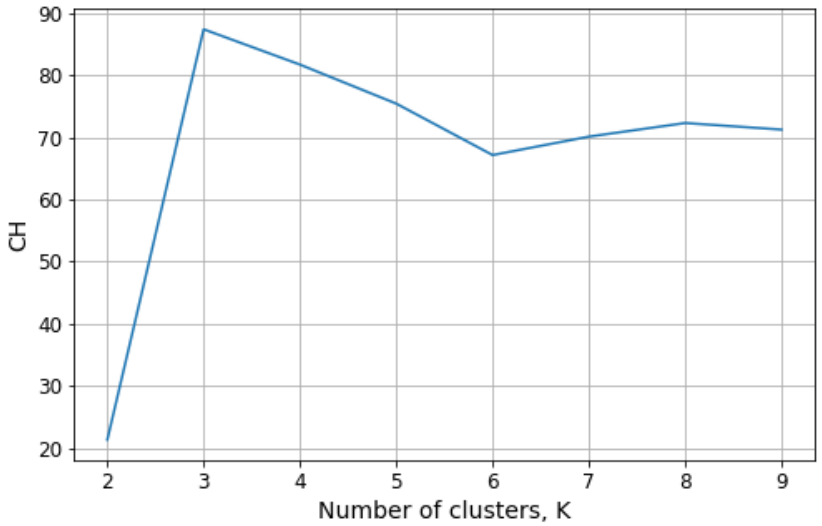

Bad Algo: CH=18.58, Silh=0.24
200
150
100
50
0
0
50
100
150

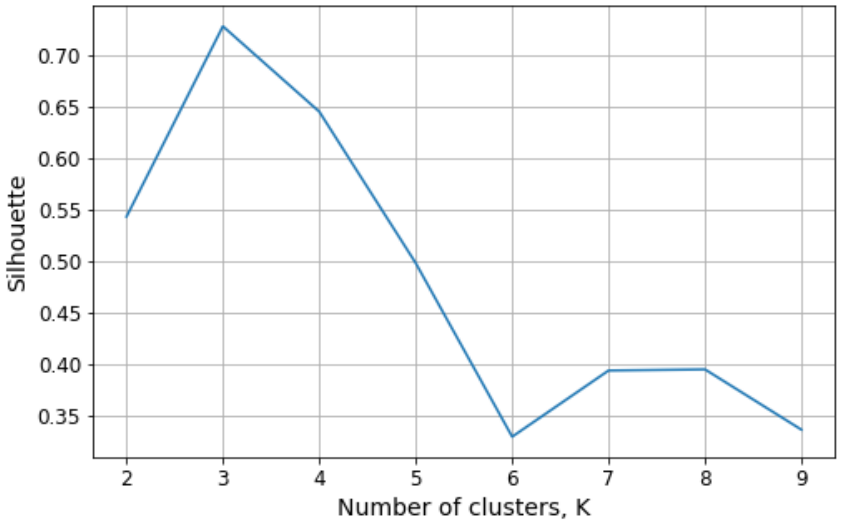

K-Means (K=3): CH=87.37, Silh=0.73
200
150
100
50
0
0
50
100
150


K-Means (K=6): CH=67.14, Silh=0.33
200
150
100
50
0
0
50
100
150

Hyperparameter Tuning

- E.g., finding the best *K* in K-Means:





PRACTICAL ISSUES

Feature Scaling

- Useful to scale/standardize features when using Euclidean
  - Who are the most similar?
- The goal is to make the features comparable
- Generally features are scaled to have:
  - Mean 0, Standard deviation 1

ID	age	income
1	65	$40,000
2	25	$40,100
3	25	$41,000
	…	

ID	age	income
1	0.15	-0.25
2	-0.21	-0.26
3	-0.21	-0.28
	…	

Categorical Features

- How to handle categorical features?
- Can't use standard distance metrics (why not?)
- Solutions:
  - Encode categorical to numeric (OHE, Ordinal, Target, etc.)
  - Use special distance metrics, like Jaccard, Hamming, Dice

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-1	0	unknown
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	339	4	failure
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	330	1	failure
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-1	0	unknown
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-1	0	unknown
5	35	management	single	tertiary	no	747	no	no	cellular	23	feb	141	2	176	3	failure
6	36	self-employed	married	tertiary	no	307	yes	no	cellular	14	may	341	1	330	2	other
7	39	technician	married	secondary	no	147	yes	no	cellular	6	may	151	2	-1	0	unknown
8	41	entrepreneur	married	tertiary	no	221	yes	no	unknown	14	may	57	2	-1	0	unknown
9	43	services	married	primary	no	-88	yes	yes	cellular	17	apr	313	1	147	2	failure

Encoding

Occupation
Management
Services
Technician
Blue-collar
Management
Blue-Collar

Management	Services	Technician	Blue-collar
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0
0	0	0	1

Education
primary
primary
secondary
primary
tertiary
secondary

Education
0
0
1
0
2
1

Categorical-aware Distance Metrics

E.g., Hamming, Jaccard, Simple Matching Coefficient, Dice

$$dist_{ham}(A,B) = \frac{1}{n}\sum_{i=1}^{n}\delta(A_i,B_i) \qquad \text{where} \qquad \delta(A_i,B_i) = \begin{cases} 0, A_i = B_i \\ 1, A_i \neq B_i \end{cases}$$

ID	Age	Gender	Location	Status	Height
A	Young	Male	Rural	Student	Low
B	Young	Male	City	Student	Low
C	Old	Female	City	Prof	High

	A, B	A, C	B, C
Hamming	0.20	1.00	0.80

Other Tidbits

- Dimensionality reduction techniques first?
  - PCA, ICA, NMF
- Some datasets are hard/impossible

RESOURCES

Resources

- Conceptual Videos by Uncle Steve
  - K-means clustering
  - DBSCAN clustering
  - Hierarchical clustering
  - Intepretting Clusters
- Coding Tutorial Videos by Uncle Steve
  - K-means clustering code
  - DBSCAN clustering code
  - Hierarchical clustering code
- Coding Tutorial Files by Uncle Steve
  - slides_clustering.ipynb
  - slides_clustering_interpret.ipynb
  - slides_clustering_mixed.ipynb
  - slides_clustering.Rmd

SUMMARY

Summary

- ***Clustering:***
  - Grouping "similar" instances together into clusters
  - An unsupervised learning technique
- **Applications**
  - Customer segmentation, outlier detection, document clustering, gene research, …
- **Distance metrics**
  - Euclidean, Cosine, Manhattan, Jaccard, Hamming, …
- **Algorithms**
  - Centroid: k-mean, k-medians, k-modes, etc.
  - Distribution: Gaussian Mixture Models
  - Density: DBSCAN, OPTICS
  - Connectivity: Hierarchical
- **Evaluation**
  - Measure internal validation metric, like silhouette/CH

APPENDIX: ALGORITHMS

Mixed Categorical and Numeric Features

- Use numeric distance metrics on the numeric features
- Use categorical distance metrics on categorical features
- Then combine distances
- Note:
  - Gower() is a function that does this (available in R)
  - Nothing in Python (but see slides_clustering_mixed.ipynb)

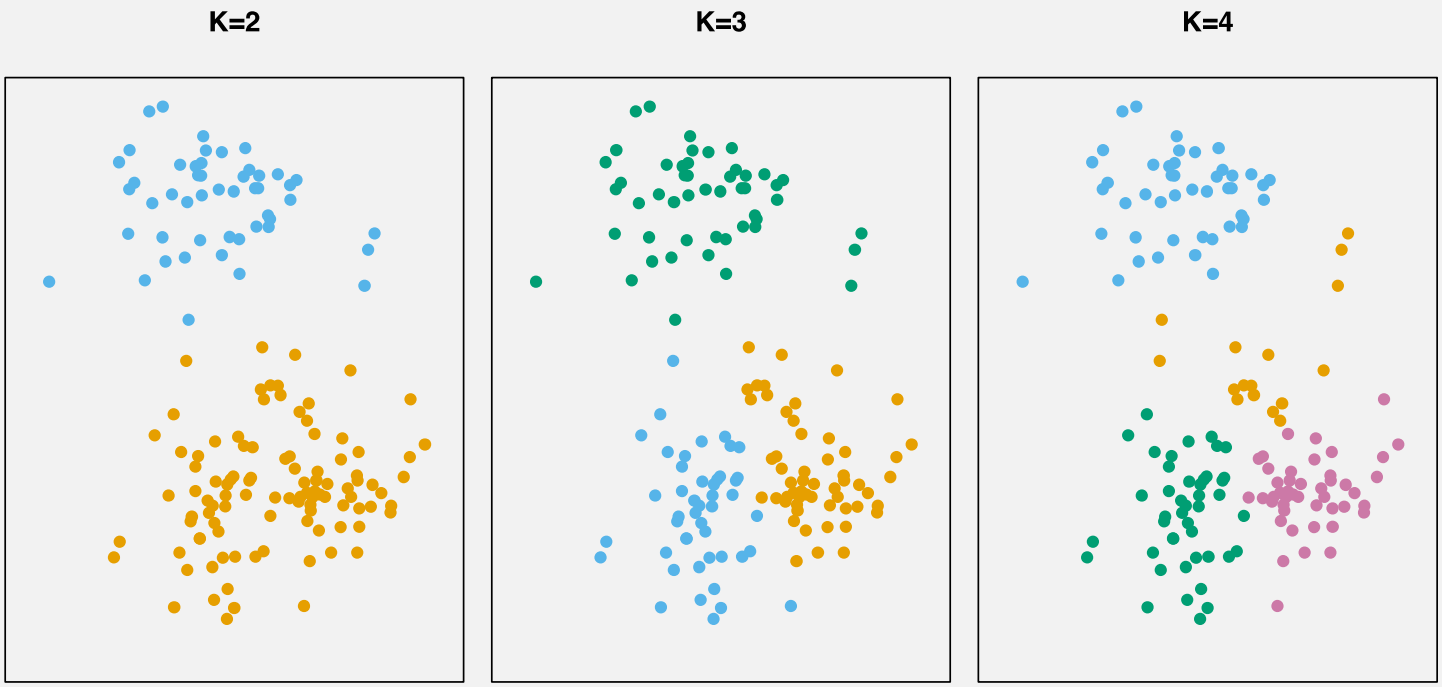
Job	Marital	Housing	Loan	Contact	Month	Default	Age	Day	Campaign	Pdays	Previous	Balance
blue-collar	married	yes	No	unknown	Mar	Unknown	1.6	-0.4	-0.1	-0.5	-0.4	0.3
blue-collar	married	yes	No	Unknown	May	True	1.3	-0.2	-0.1	-2.1	-0.9	1.3

- Euclidean distance = 1.98
- Hamming distance = 2/7 = 0.29
- Weighted average = (0.29*7 + 1.98*6) / 13 = 1.07

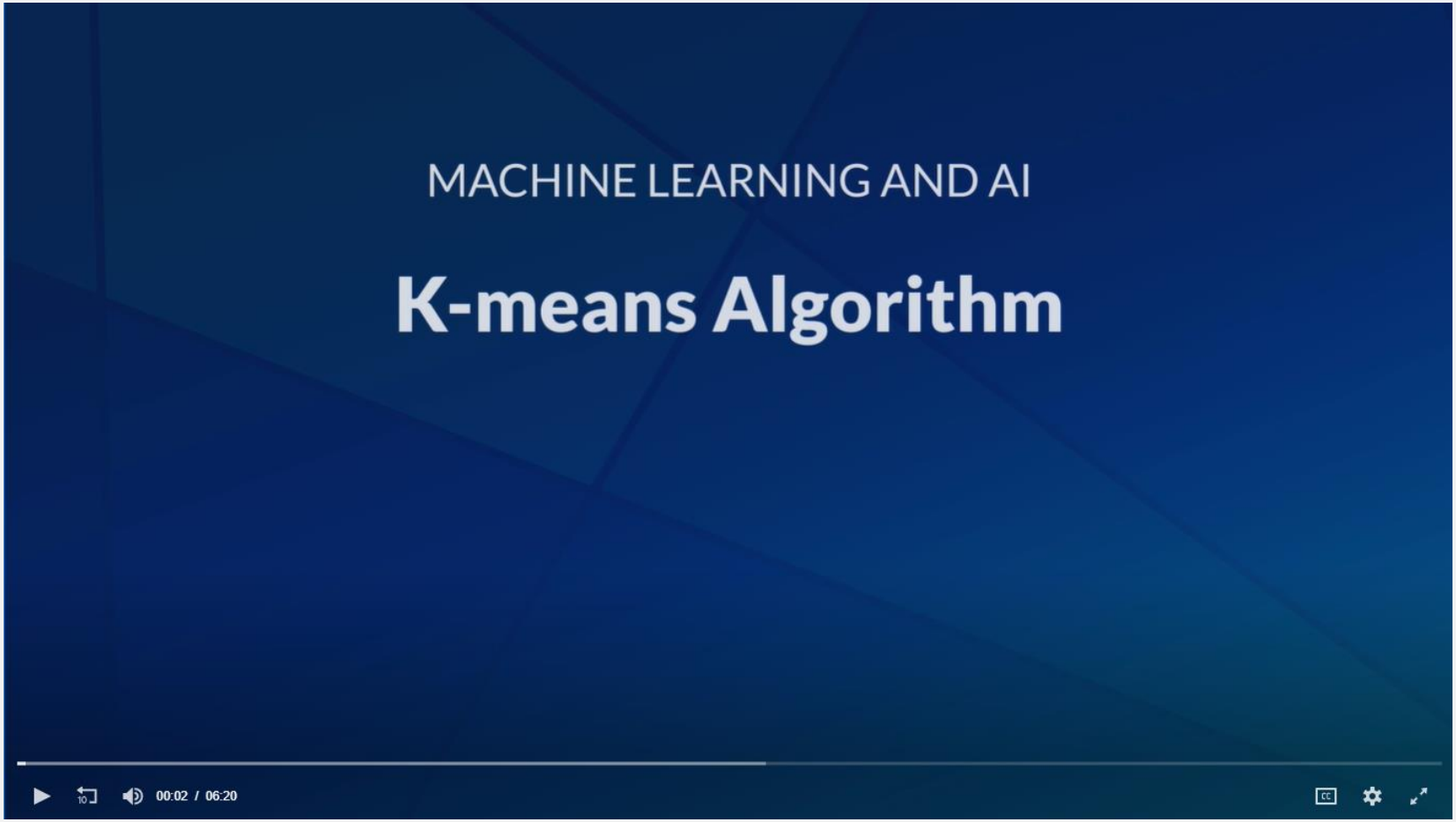
K-MEANS (A CENTROID MODEL)

K-Means

- One of the most popular clustering algorithms
  - Fast, easy to understand
- User specifies the desired number of clusters, K
- Algorithm assigns each instance to exactly one of the K clusters



How Does it Work?



https://stream.queensu.ca/Watch/e6HKg75W

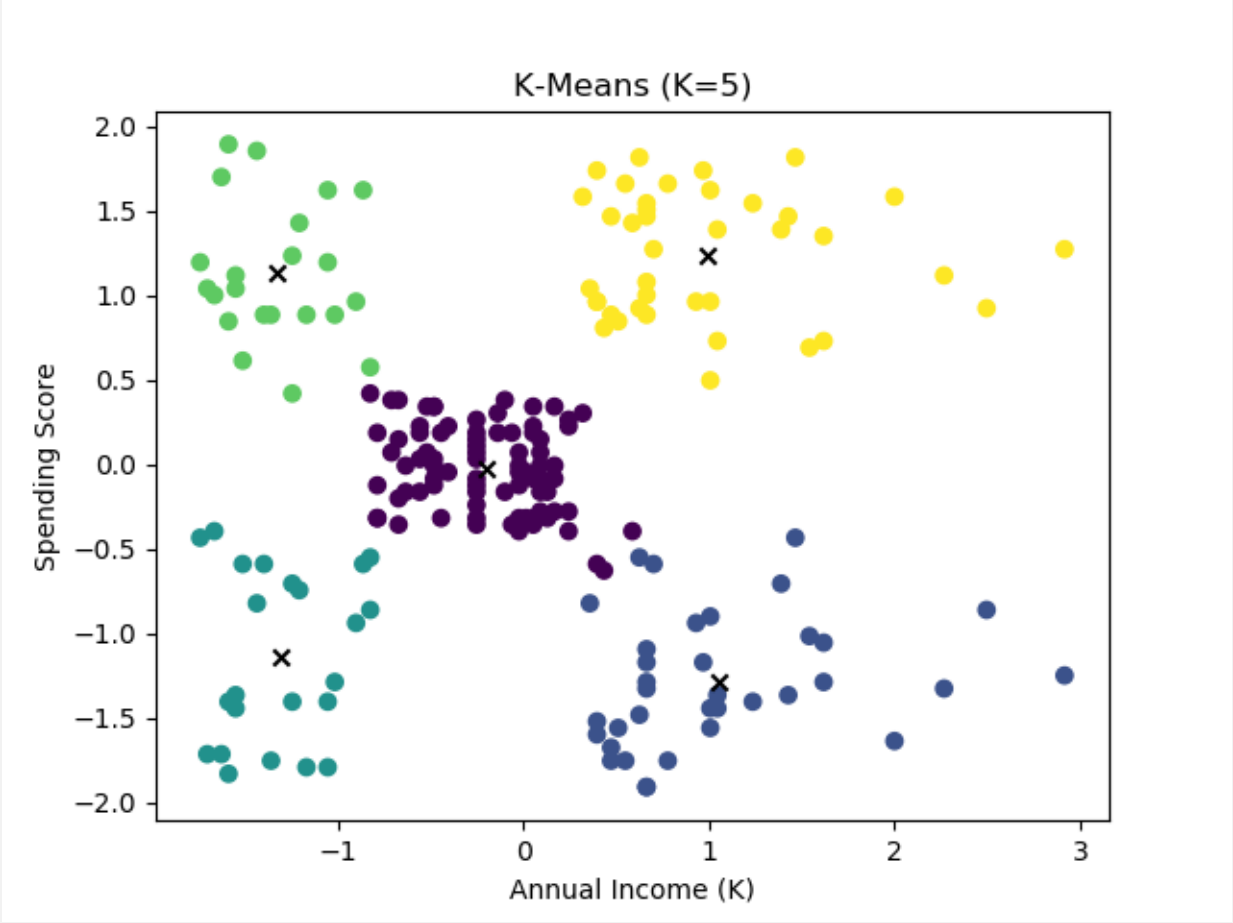
Example: Mall Data

```
k_means = KMeans(n_clusters=5, random_state=42)
k_means.fit(X)
```

```
k_means.labels_
```

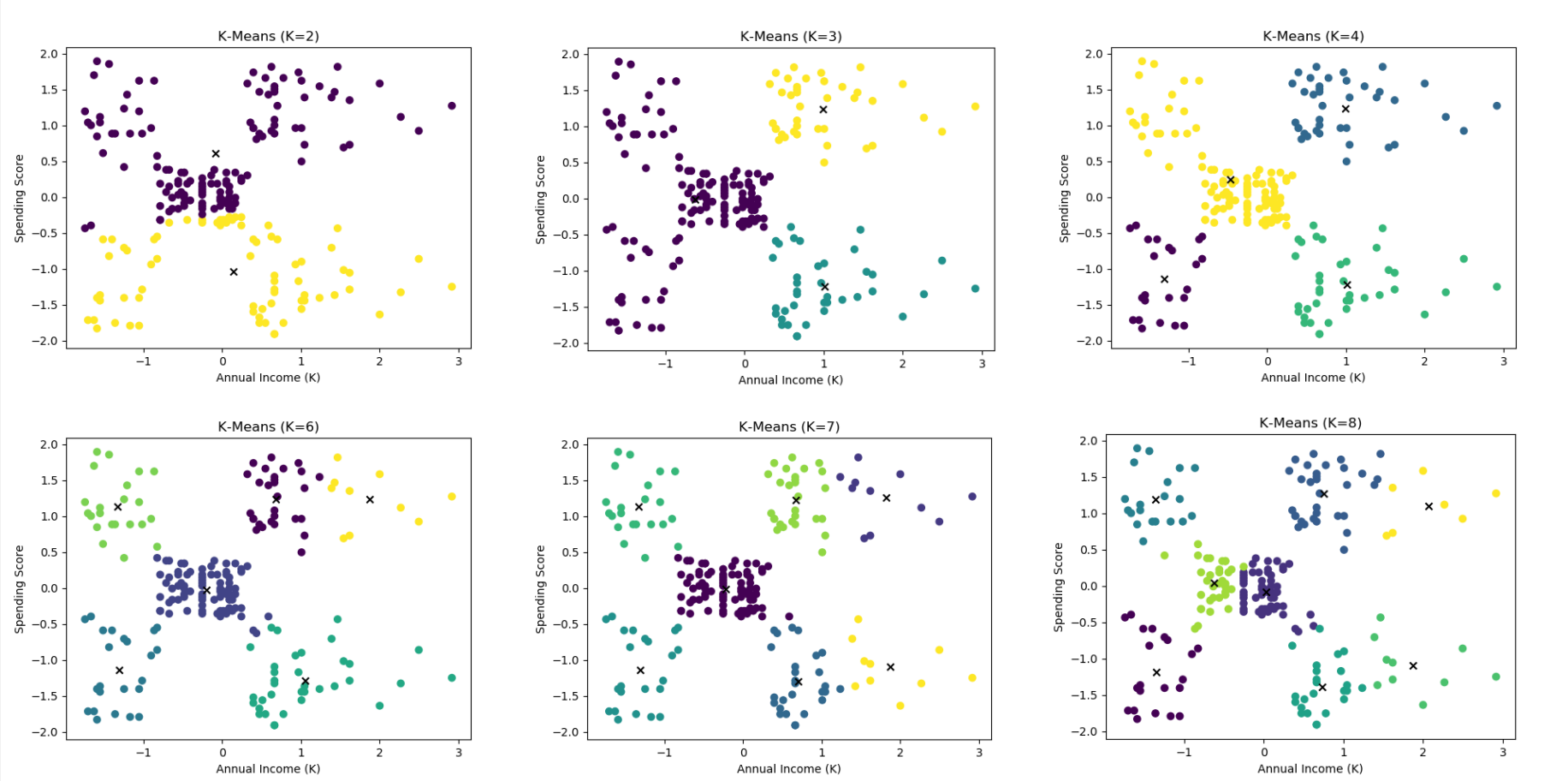
```
array([2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 0,
       2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 1, 4, 0, 4, 1, 4, 1, 4,
       0, 4, 1, 4, 1, 4, 1, 4, 1, 4, 0, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4])
```

Example: Mall Data



How to Determine K?

- Trial and Error (*Hyperparameter Tuning*)
  - Try different Ks, see which "works best"
    - Better internal validation metrics (discussed later)
    - Better for business problem

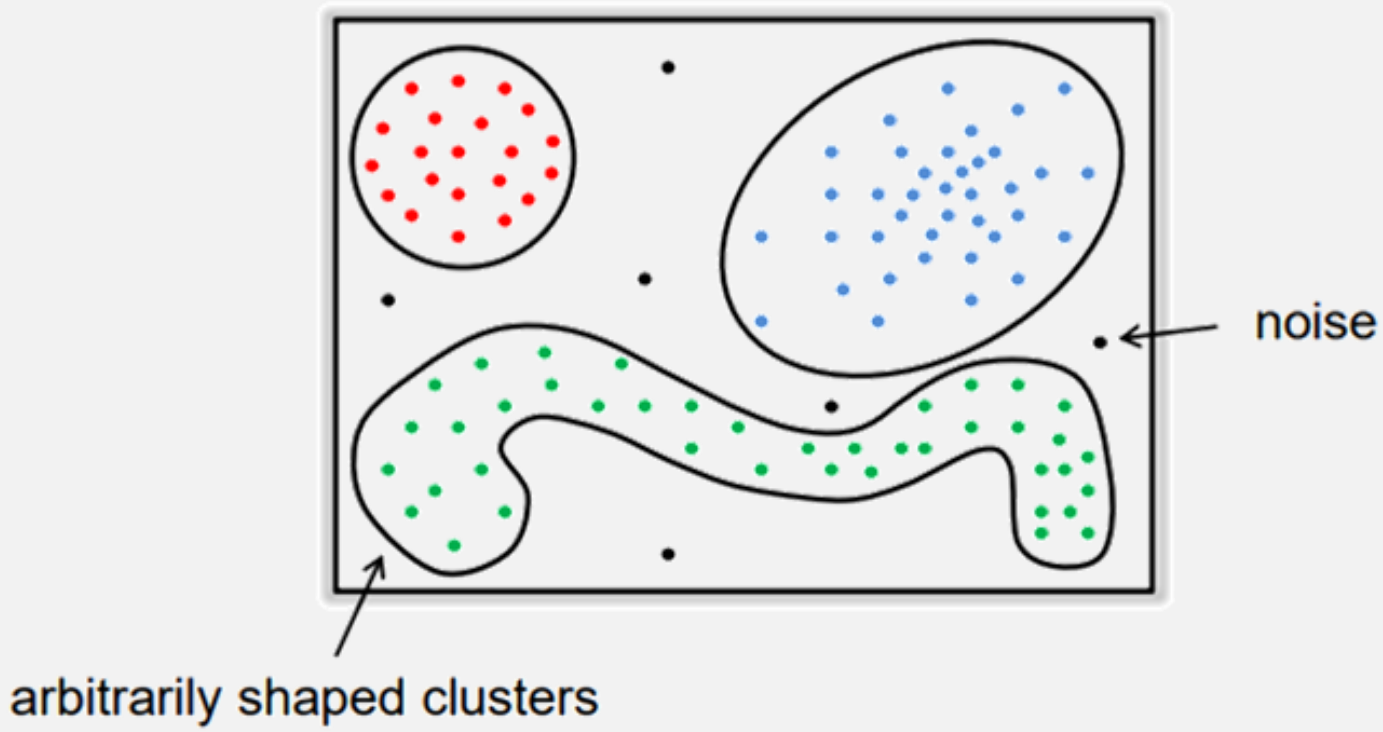


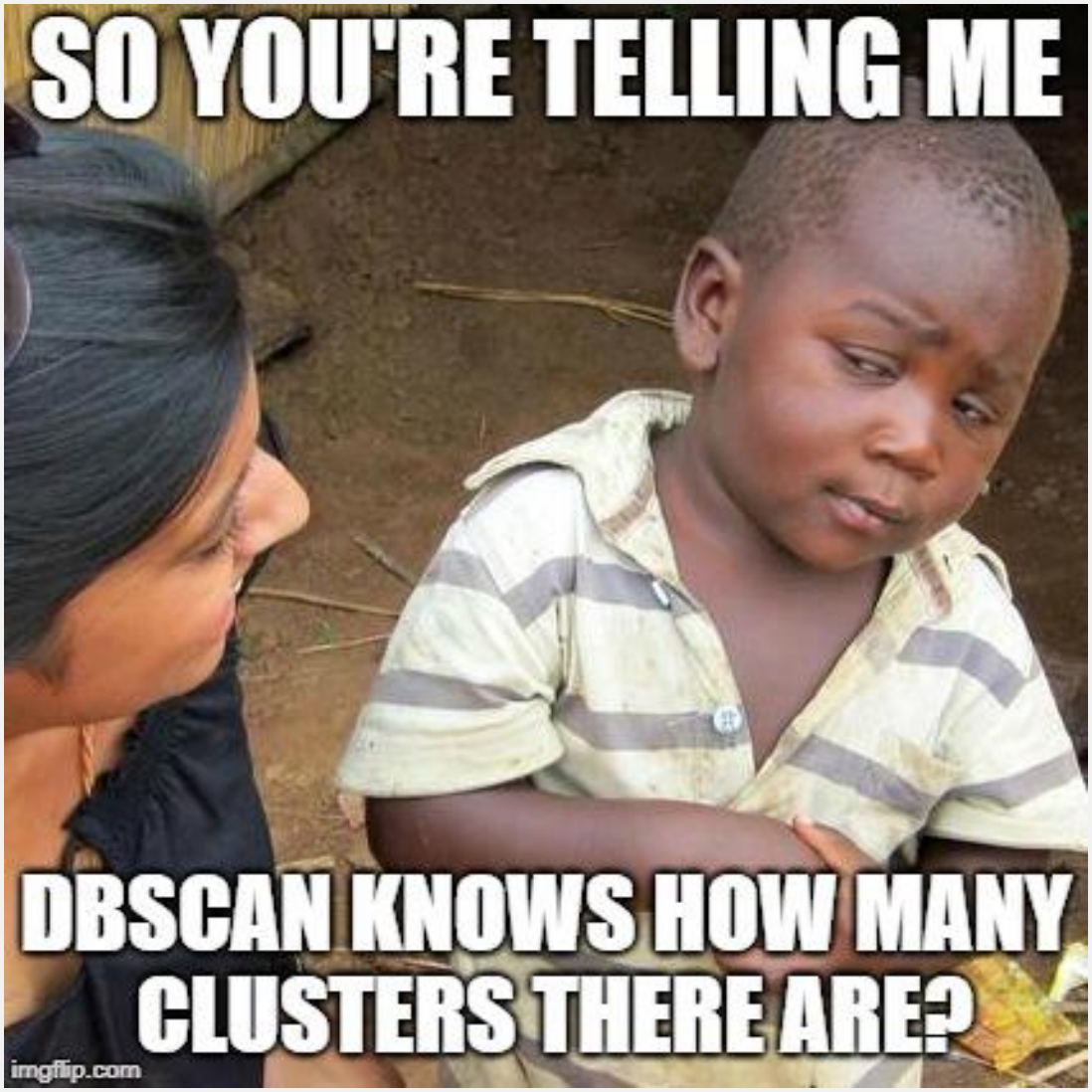
DBSCAN (A DENSITY MODEL)

DBSCAN

Density-**b**ased **s**patial **c**lustering of **a**pplications with **n**oise

- Discovers of clusters with arbitrary shape
- No prior knowledge of the number of clusters required
- Good efficiency on large data sets
- Two hyper parameters : MinPts and Eps



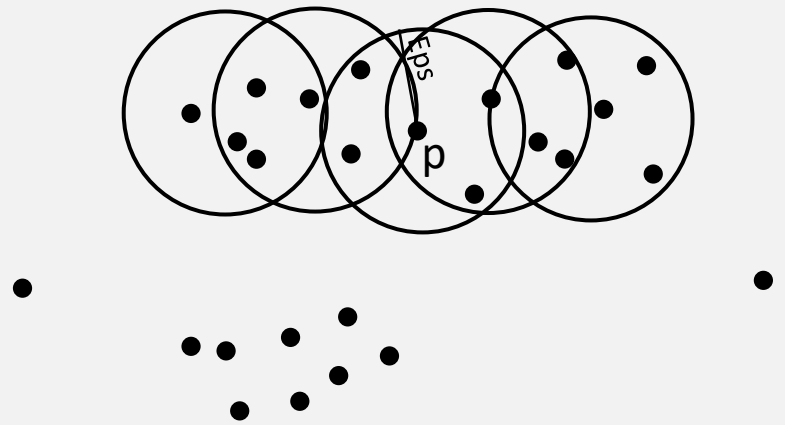

SO YOU'RE TELLING ME
DBSCAN KNOWS HOW MANY CLUSTERS THERE ARE?
imgflip.com

How Does It Work?

https://stream.queensu.ca/Watch/Lk28BmZb

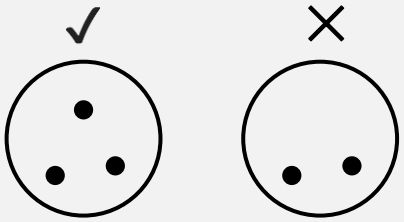
DBSCAN

- Main Idea:
  - Clusters have a higher density of points than noise
  - Noise has lower density than clusters
- Algorithm looks for sets of points that are "dense"

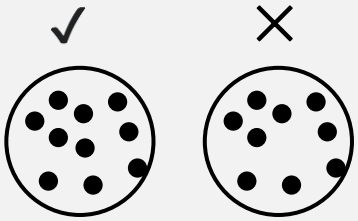


- *Eps: Radius of circle*
- *MinPts: Number of instances inside circle*

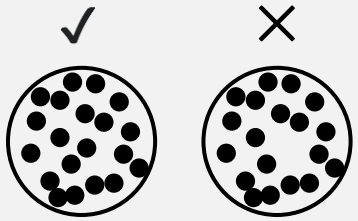
Effect of MinPts and Epsilon



MinPts = 3

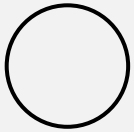


MinPts = 10

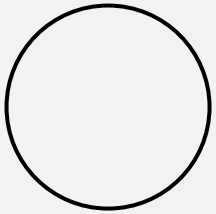


MinPts = 20

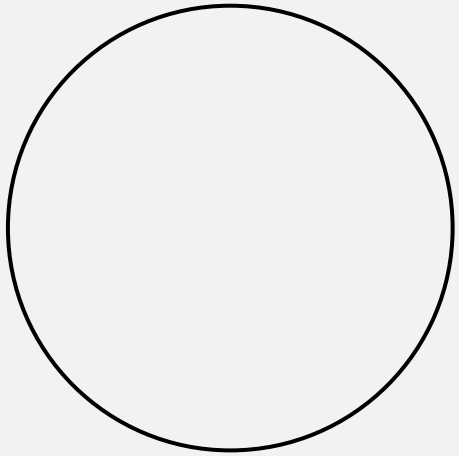
Small epsilon



Medium epsilon



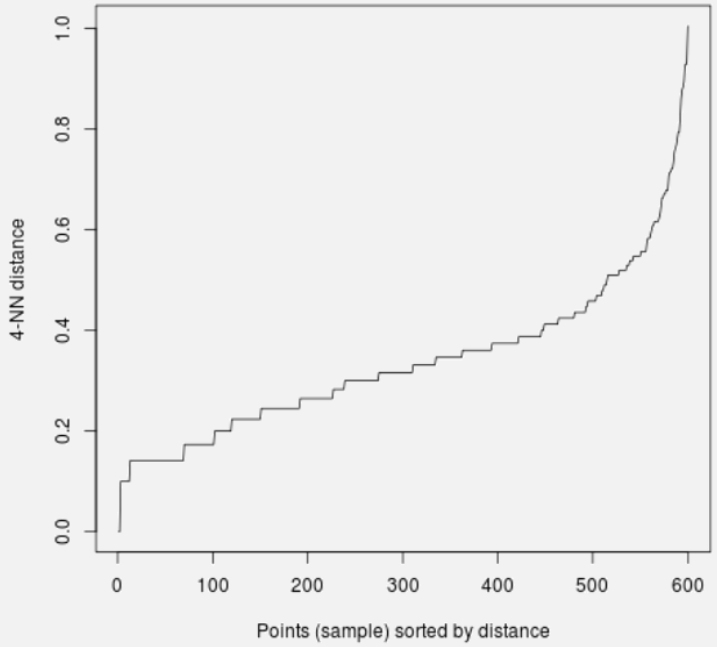
Large epsilon



Huge epsilon

Choosing Hyperparameter Values

- Trial and error (*Hyperparameter Tuning*)
- Best practices:
  - MinPts
    - At least 3
    - At least the number of features + 1
  - Eps
    - Using the elbow in a k-distance graph

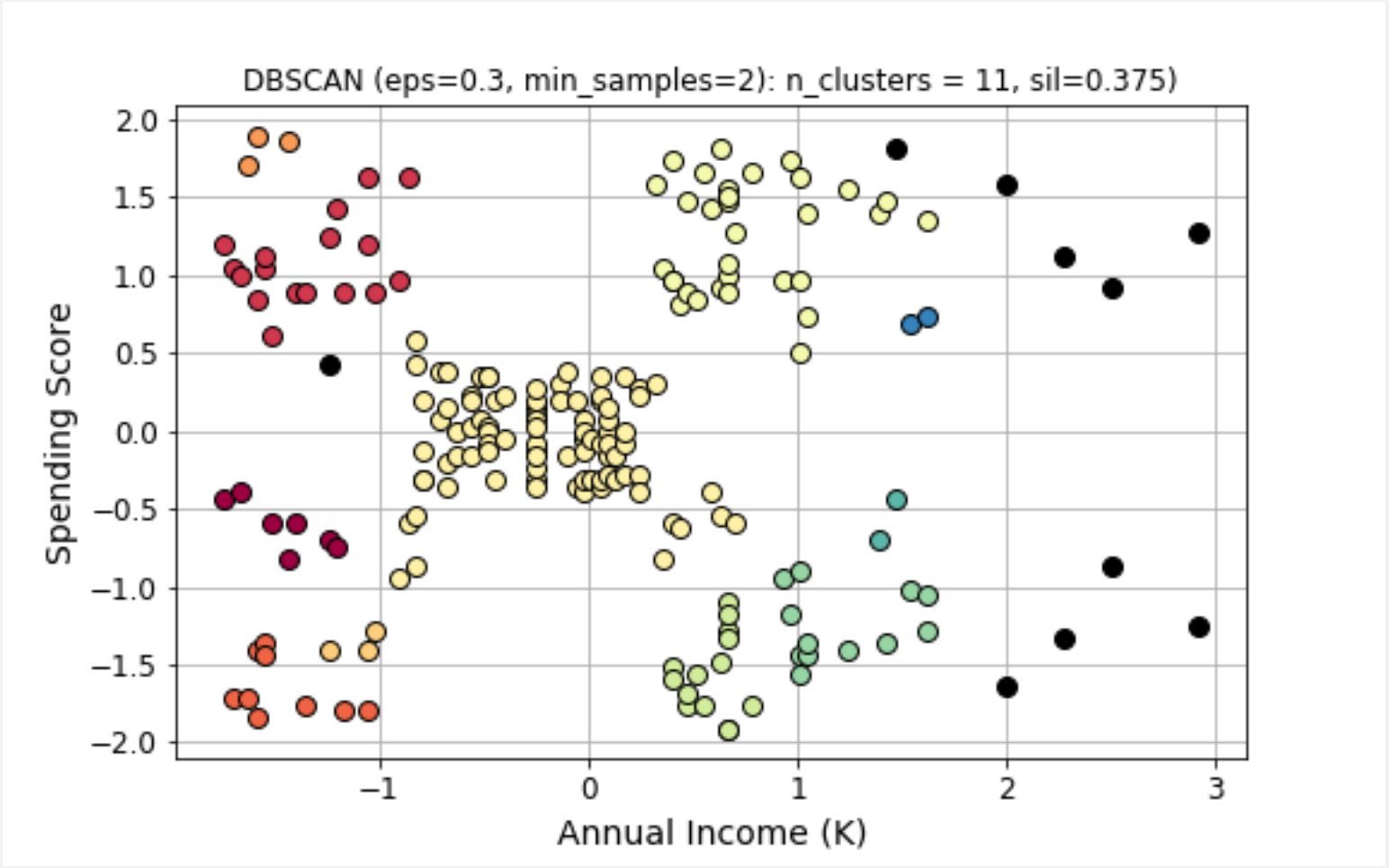


Example: Mall Data

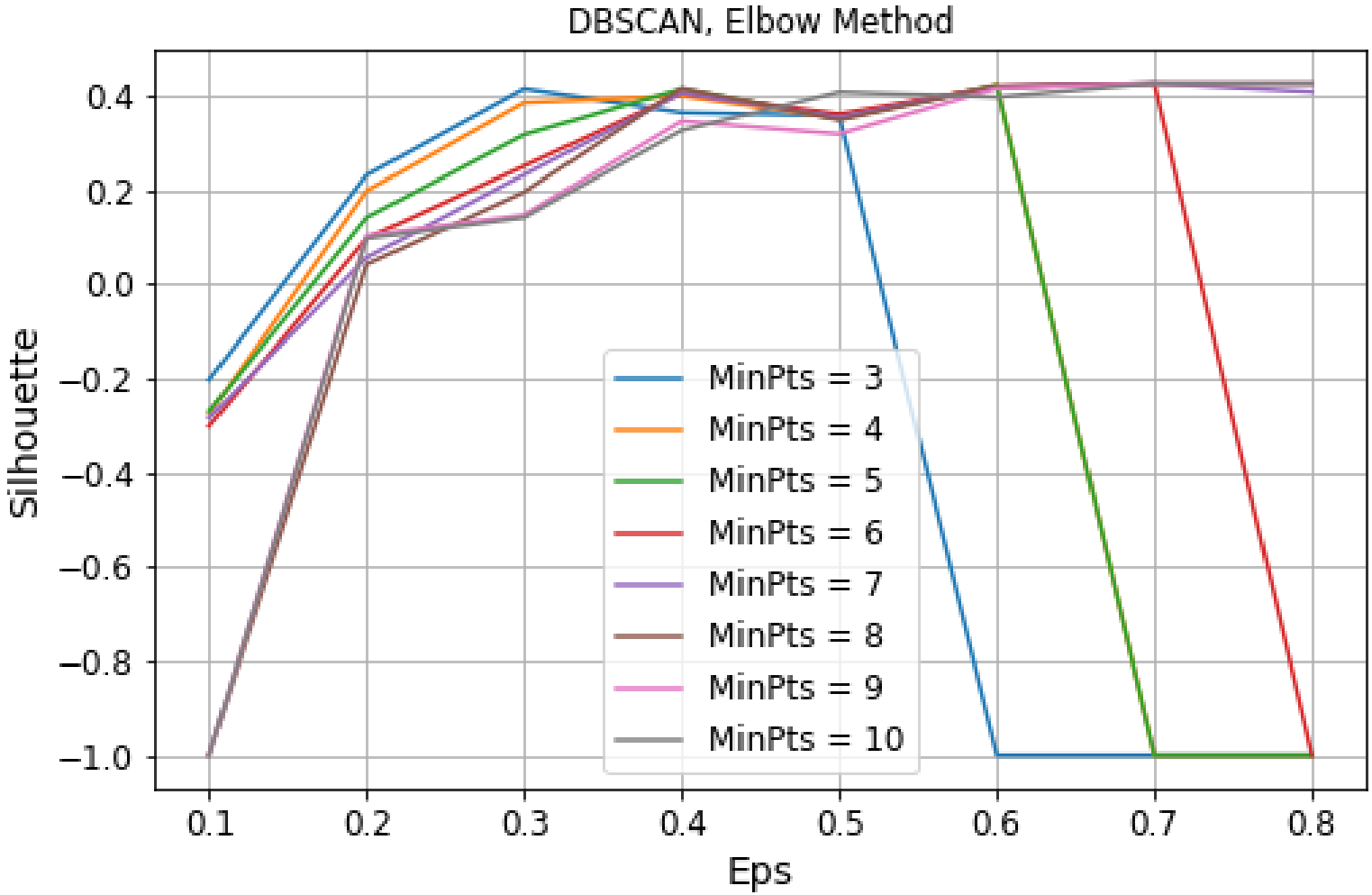
```
db = DBSCAN(eps=0.3, min_samples=2)
db.fit(X)
```

```
db.labels_
```

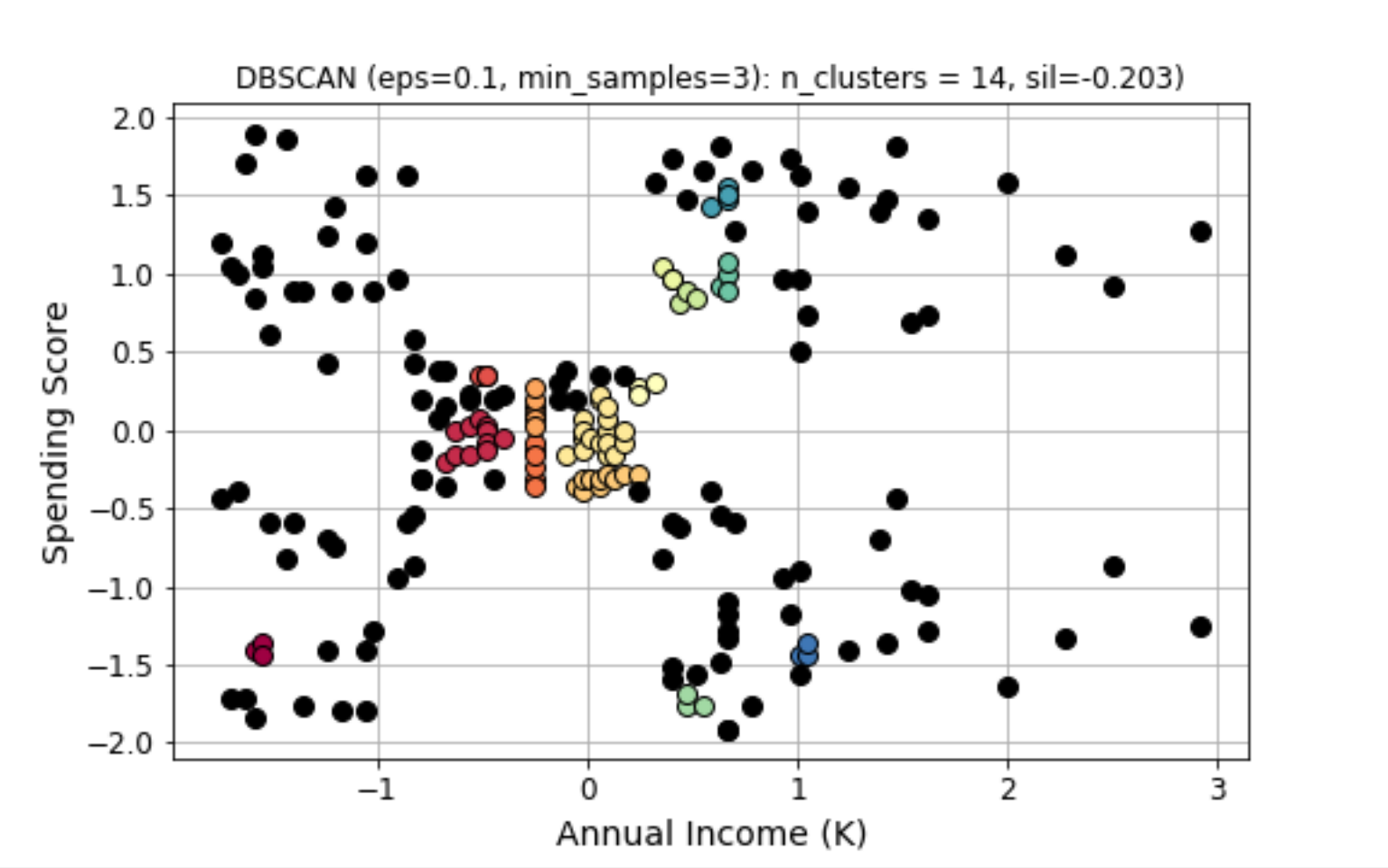
```
array([ 0,  1,  2,  1,  0,  1,  2,  3,  2,  1,  2,  3,  2,  1,  2,  1,  0,
        1,  0,  3,  0,  1,  2,  1,  4,  1,  0, -1,  0,  1,  2,  1,  2,  1,
        4,  1,  4,  1,  5,  1,  5,  1,  5,  5,  5,  5,  5,  5,  5,  5,  5,
        5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,
        5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,
        5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,
        5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,  5,
        5,  5,  5,  5,  6,  5,  6,  5,  6,  7,  6,  7,  6,  5,  6,  7,  6,
        7,  6,  7,  6,  7,  6,  5,  6,  7,  6,  5,  6,  7,  6,  7,  6,  7,
        6,  7,  6,  7,  6,  7,  6,  5,  6,  7,  6,  8,  6,  8,  6,  8,  6,
        8,  6,  8,  6,  8,  6,  8,  6,  8,  6,  9,  6,  8,  6,  9, -1,  8,
       10,  8,  6,  8, 10, -1, -1, -1, -1, -1, -1, -1, -1], dtype=int64)
```


DBSCAN (eps=0.3, min_samples=2): n_clusters = 11, sil=0.375)
Spending Score
Annual Income (K)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1
0
1
2
3

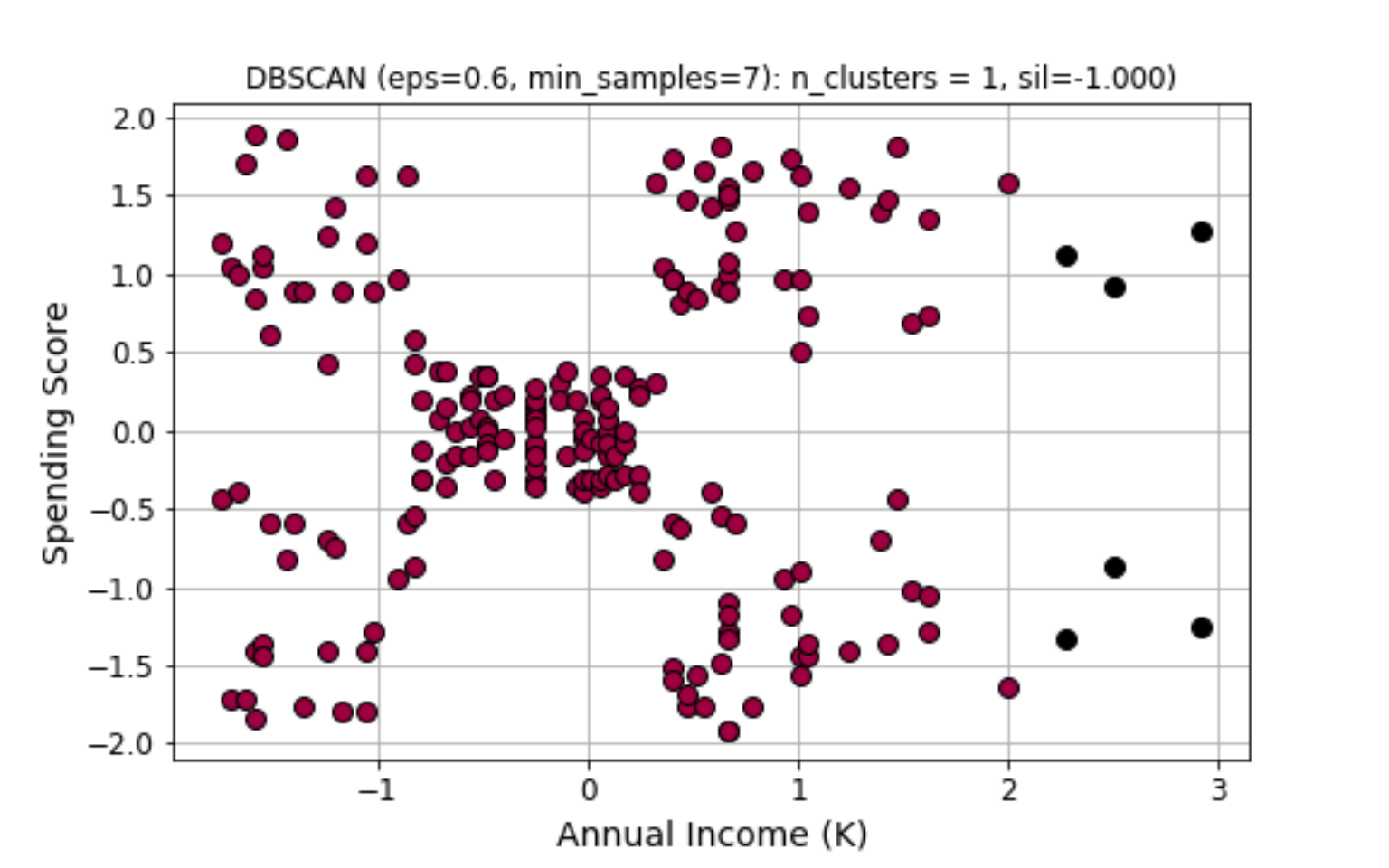
Example: Mall Data

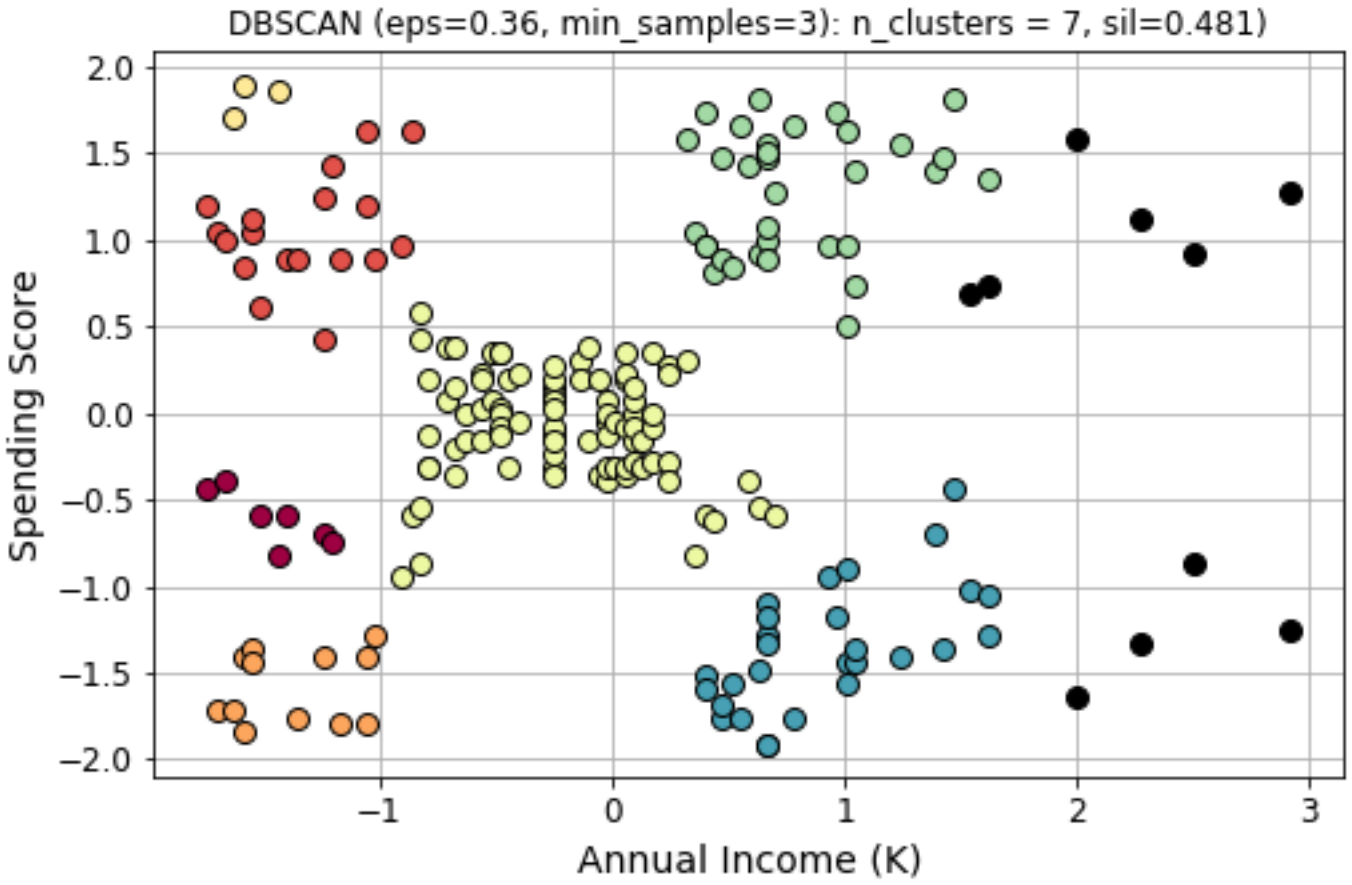


Example: Mall Data



Example: Mall Data



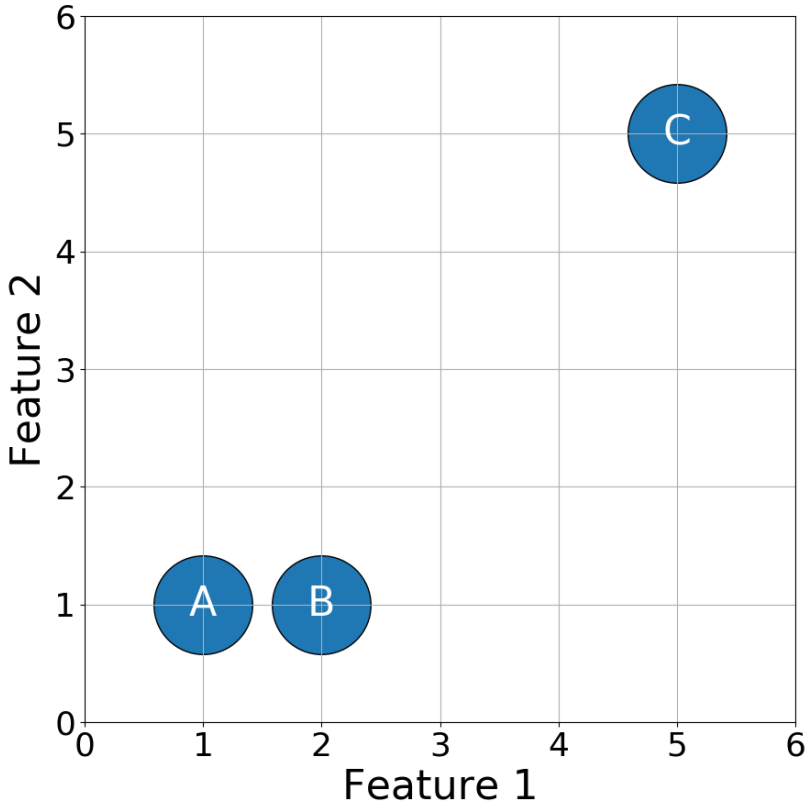

DBSCAN (eps=0.36, min_samples=3): n_clusters = 7, sil=0.481)
Spending Score
Annual Income (K)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1
0
1
2
3

APPENDIX

Chebyshev Distance

Greatest of the distances along any dimension

$$dist_{cheby}(A, B) = \max_i |A_i - B_i|$$



ID	Feature 1	Feature 2
A	1.0	1.0
B	2.0	1.0
C	5.0	5.0

	A, B	A, C	B, C
Euclidean	1.00	5.66	5.00
Cosine	0.05	0.00	0.05
Manhattan	1.00	8.00	7.00
Chebychev	1.00	4.00	4.00

Example: Mall Customer Data

Toy dataset representing customers at a mall

- 200 instances, three features

ID	Age	Annual Income	Spend Score
1	0.02	0.00	0.39
2	0.06	0.00	0.82
3	0.04	0.01	0.05
4	0.10	0.01	0.78
5	0.25	0.02	0.40
6	0.08	0.02	0.77
7	0.33	0.02	0.05
8	0.10	0.02	0.95
9	0.88	0.03	0.02
10	0.23	0.03	0.72
11	0.94	0.03	0.13
12	0.33	0.03	1.00
13	0.77	0.04	0.14
14	0.12	0.04	0.78
15	0.37	0.04	0.12

Euclidean, most similar

ID	Age	Annual Income	Spend Score	Distance
5	0.25	0.02	0.40	0.23
48	0.17	0.20	0.47	0.27
18	0.04	0.05	0.66	0.28

Euclidean, least similar

ID	Age	Annual Income	Spend Score	Distance
197	0.52	0.91	0.27	1.04
199	0.27	1.00	0.17	1.05
200	0.23	1.00	0.83	1.12

Manhattan, most similar

ID	Age	Annual Income	Spend Score	Distance
5	0.25	0.02	0.40	0.25
18	0.04	0.05	0.66	0.35
3	0.04	0.01	0.05	0.36

Manhattan, least similar

ID	Age	Annual Income	Spend Score	Distance
200	0.23	1.0	0.83	1.66
179	0.78	0.64	0.13	1.66
194	0.38	0.80	0.92	1.70

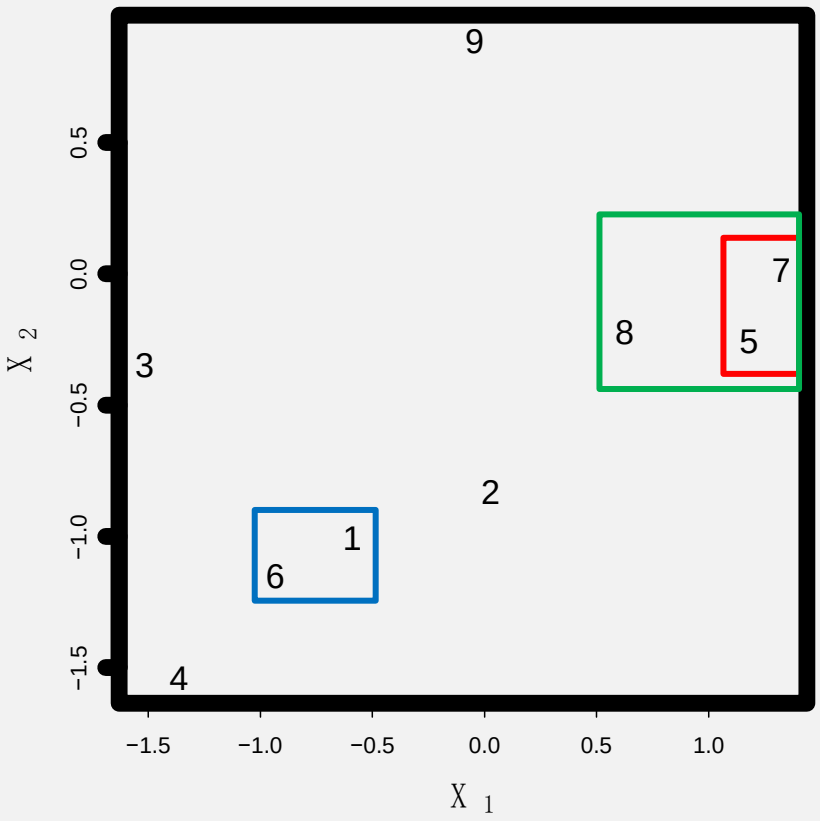
Cosine, most similar

ID	Age	Annual Income	Spend Score	Distance
2	0.06	0.00	0.82	0.0002
6	0.08	0.02	0.77	0.0015
8	0.10	0.02	0.95	0.0017

Cosine, least similar

ID	Age	Annual Income	Spend Score	Distance
9	0.88	0.03	0.02	0.93
157	0.37	0.52	0.00	0.97
159	0.31	0.52	0.00	0.97

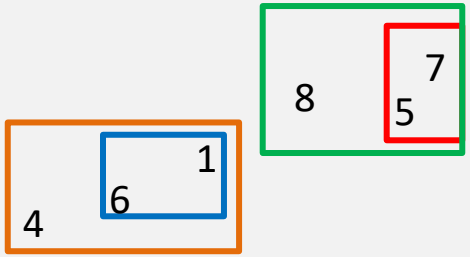
How Does it Work?

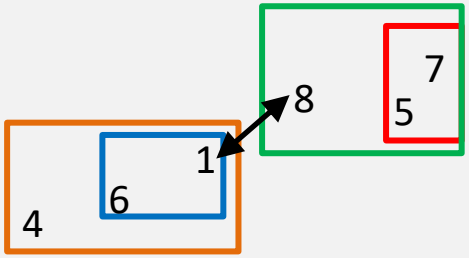


- Say we have two features and 9 instances
- Initially, all 9 points are a cluster
- Closest two points are fused (5, 7)
- Next closest two points are fused (6, 1)
- Next closest two points are fused (8, (5, 7))
- …
- Continue until all instances are fused

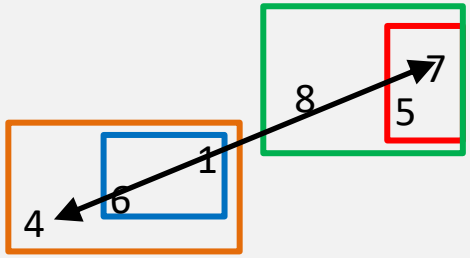
Linkages

- How do we define the distance (*linkage*) between clusters?

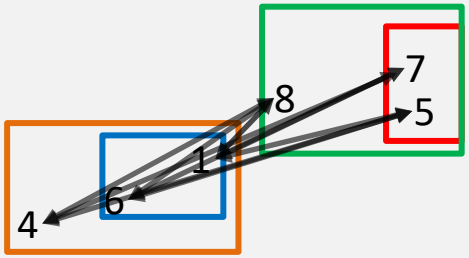




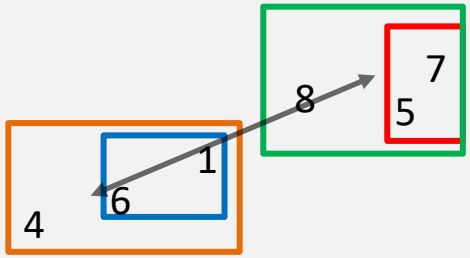
Single Linkage
Smallest distance between instances



Maximum (Complete) Linkage
Largest distance between instances



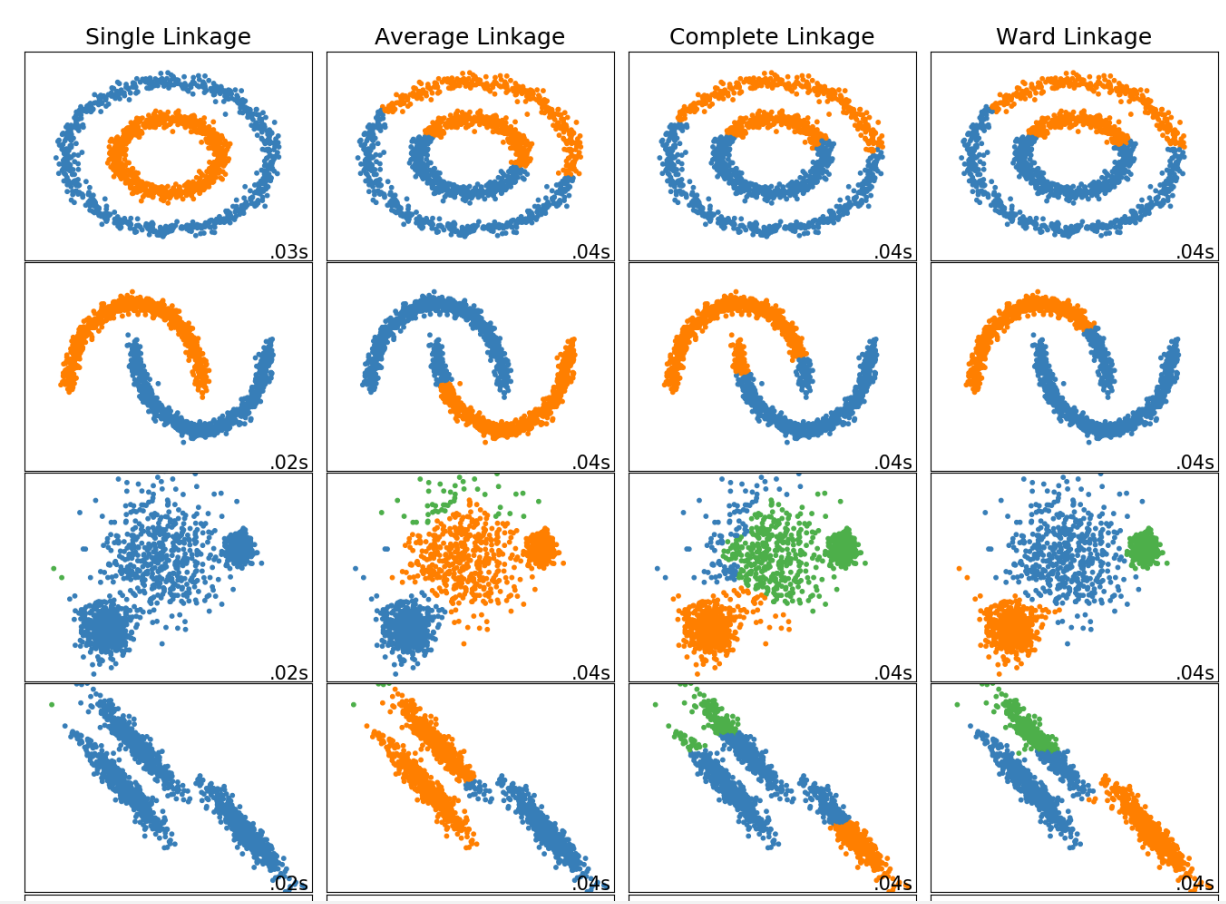
Average Linkage
Average distance between all instances



Ward Linkage
Distance between cluster centroids

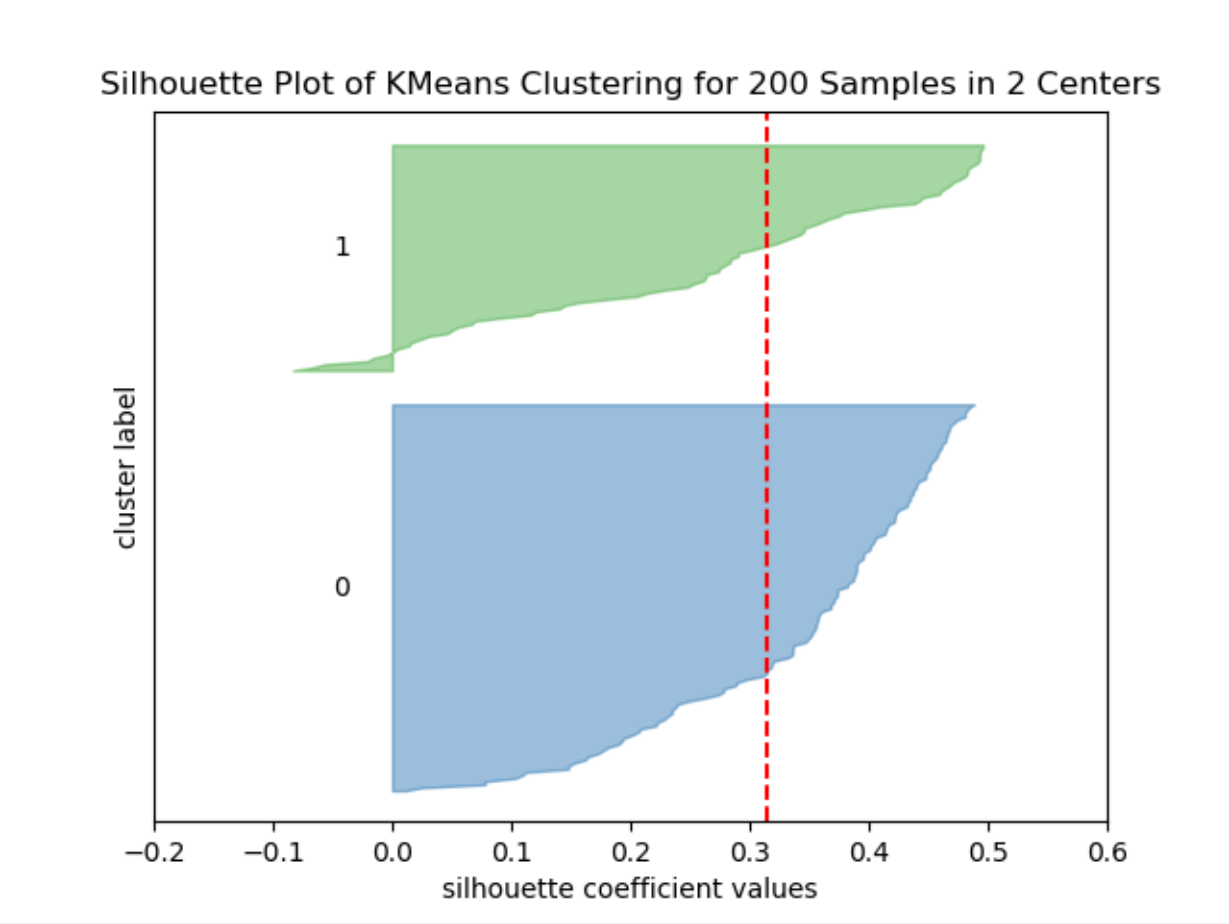
Linkage Can Be Important

- Single linkage is very fast, and good for non-globular data, but bad for globular
- Average, Complete, and Ward create evenly-sized clusters
  - Ward is best for globular data, but can only use Euclidean
  - Complete and Average can use any distance metric

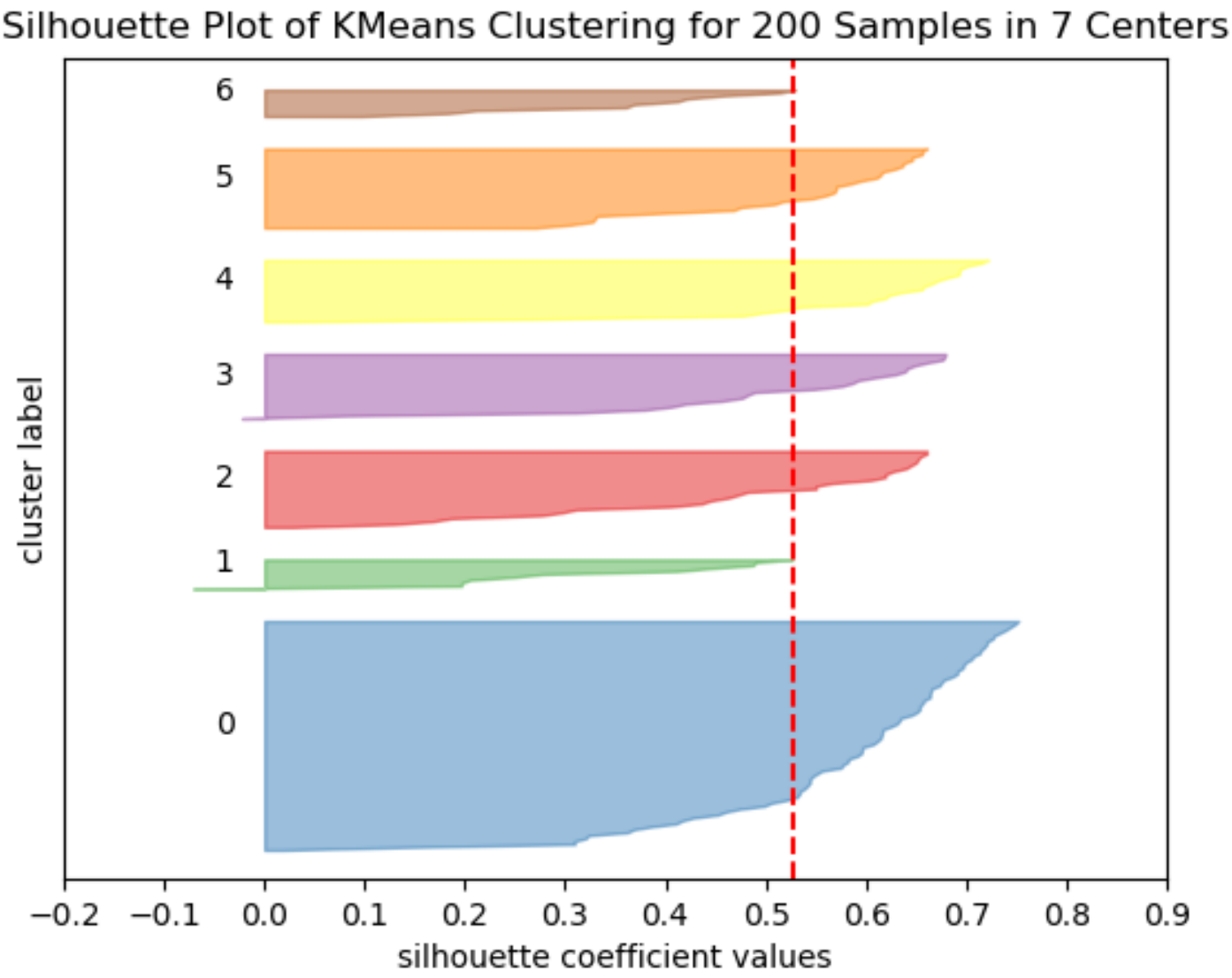


Silhouette Plot

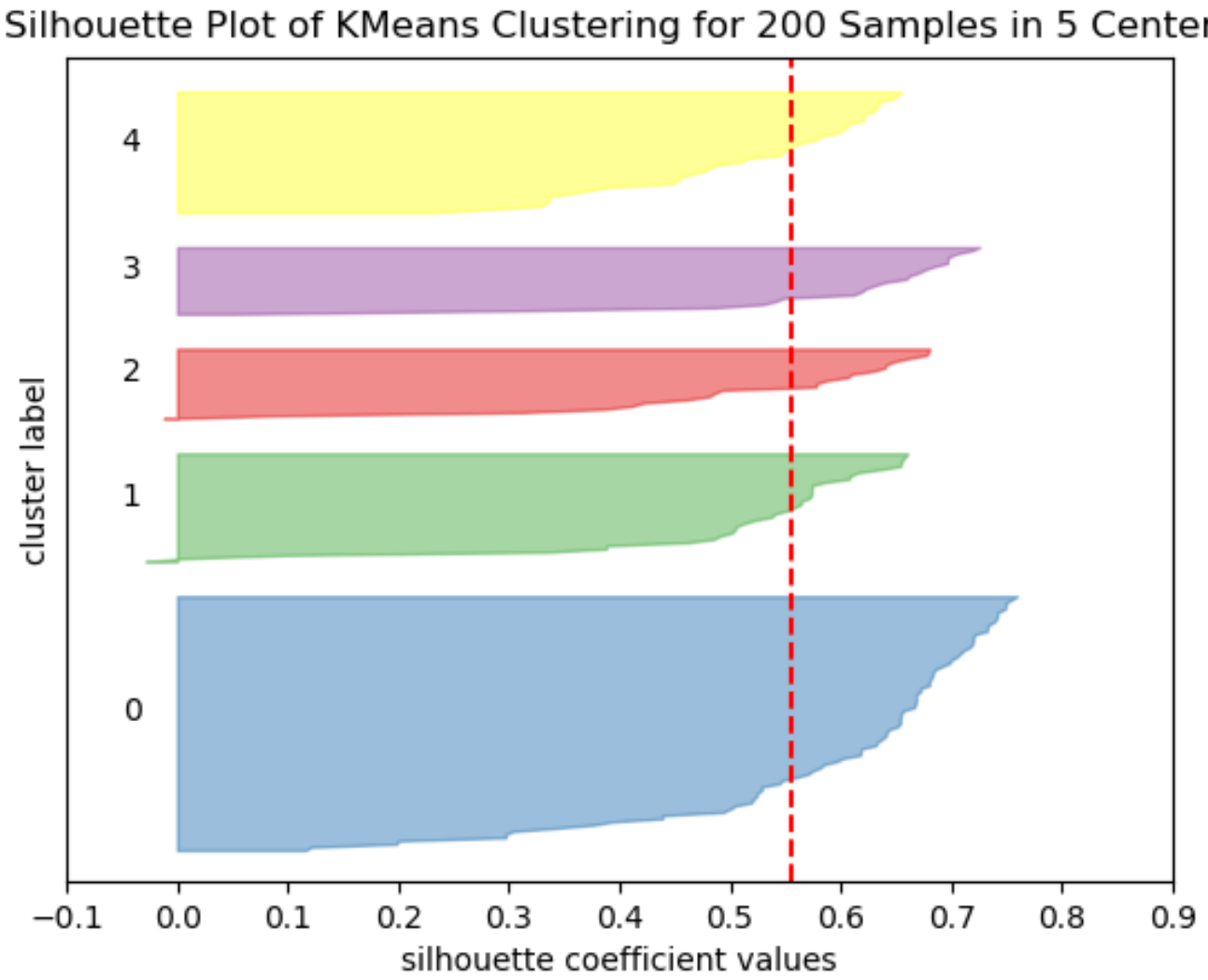
- Shows the silhouette coefficients for each instance
- Allows you to quickly see if there are any "bad" clusters



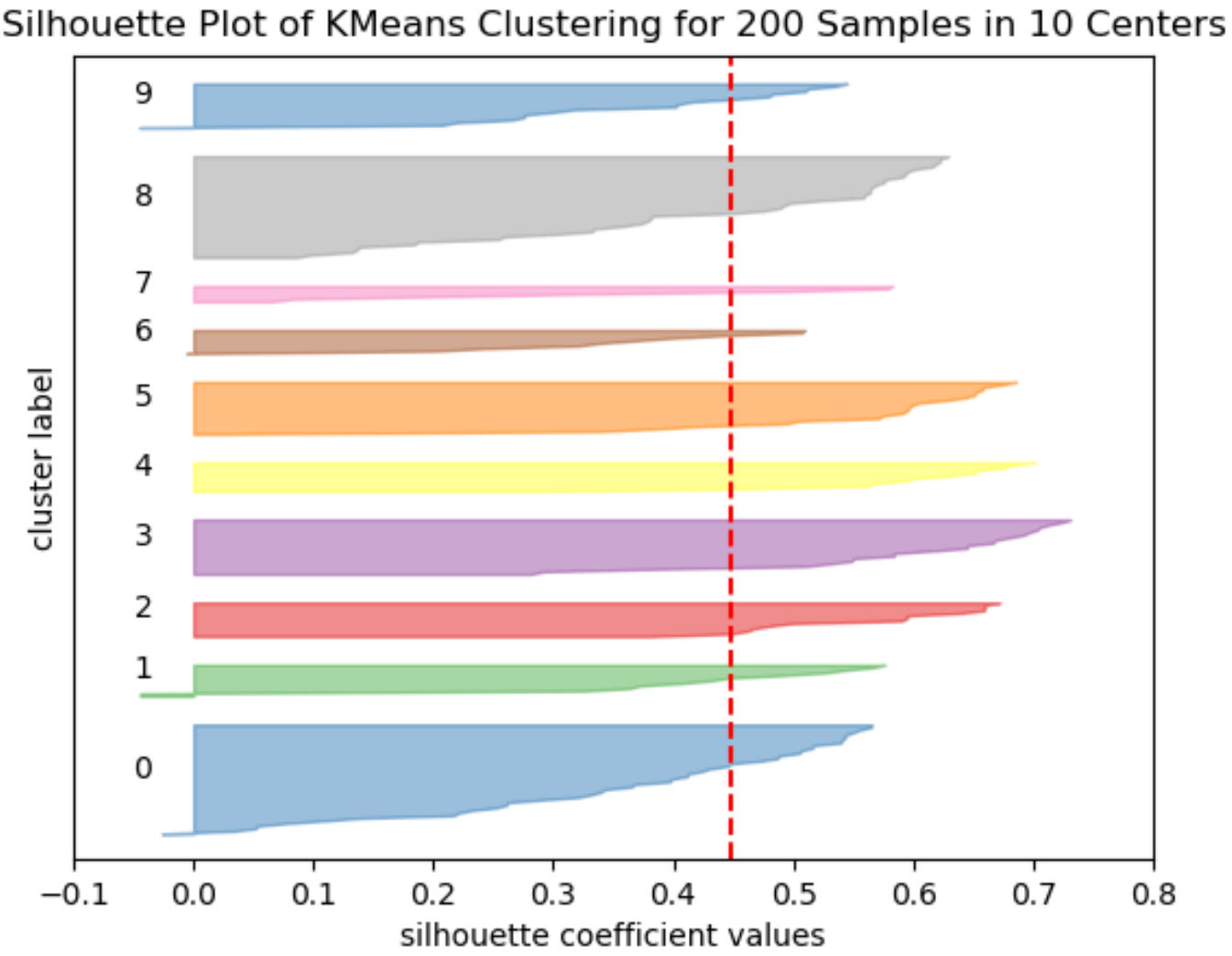
Example: Mall Data



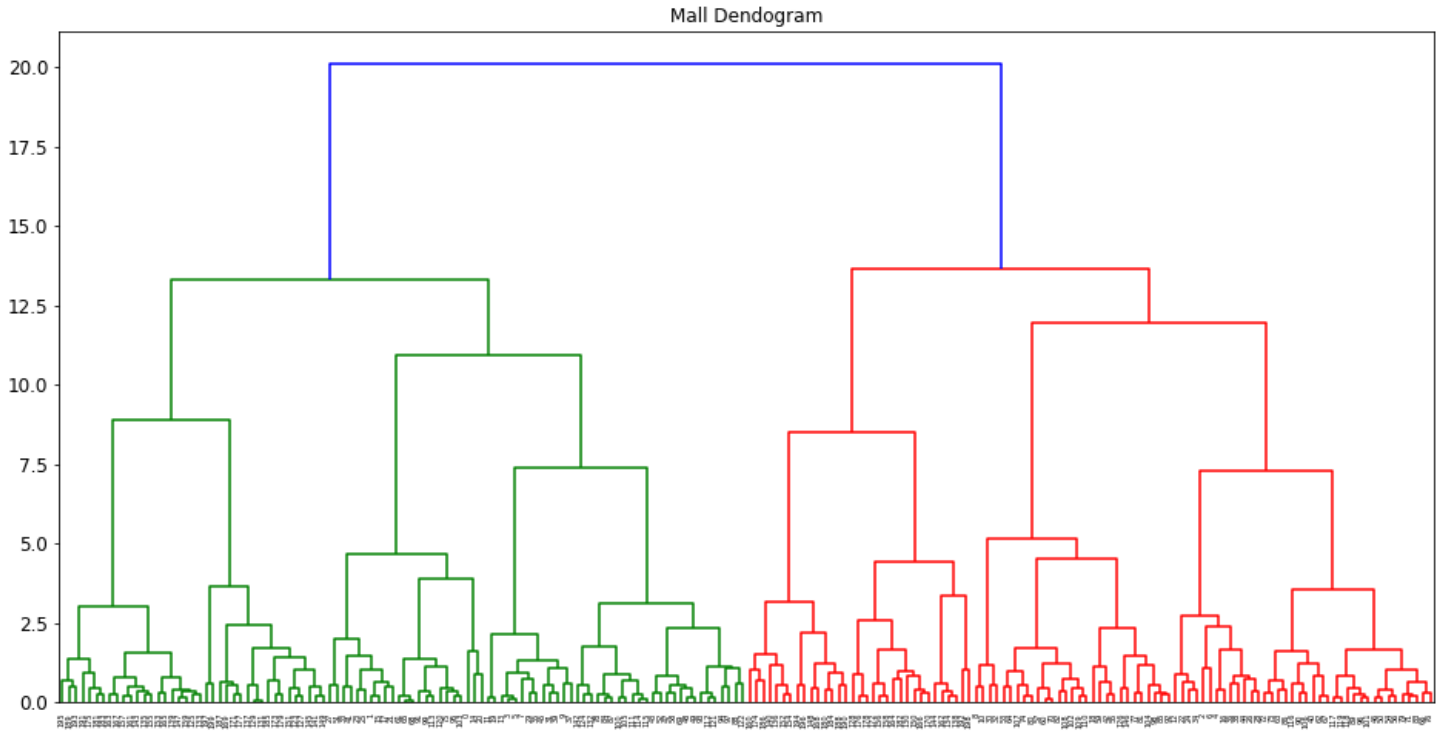
Example: Mall Data



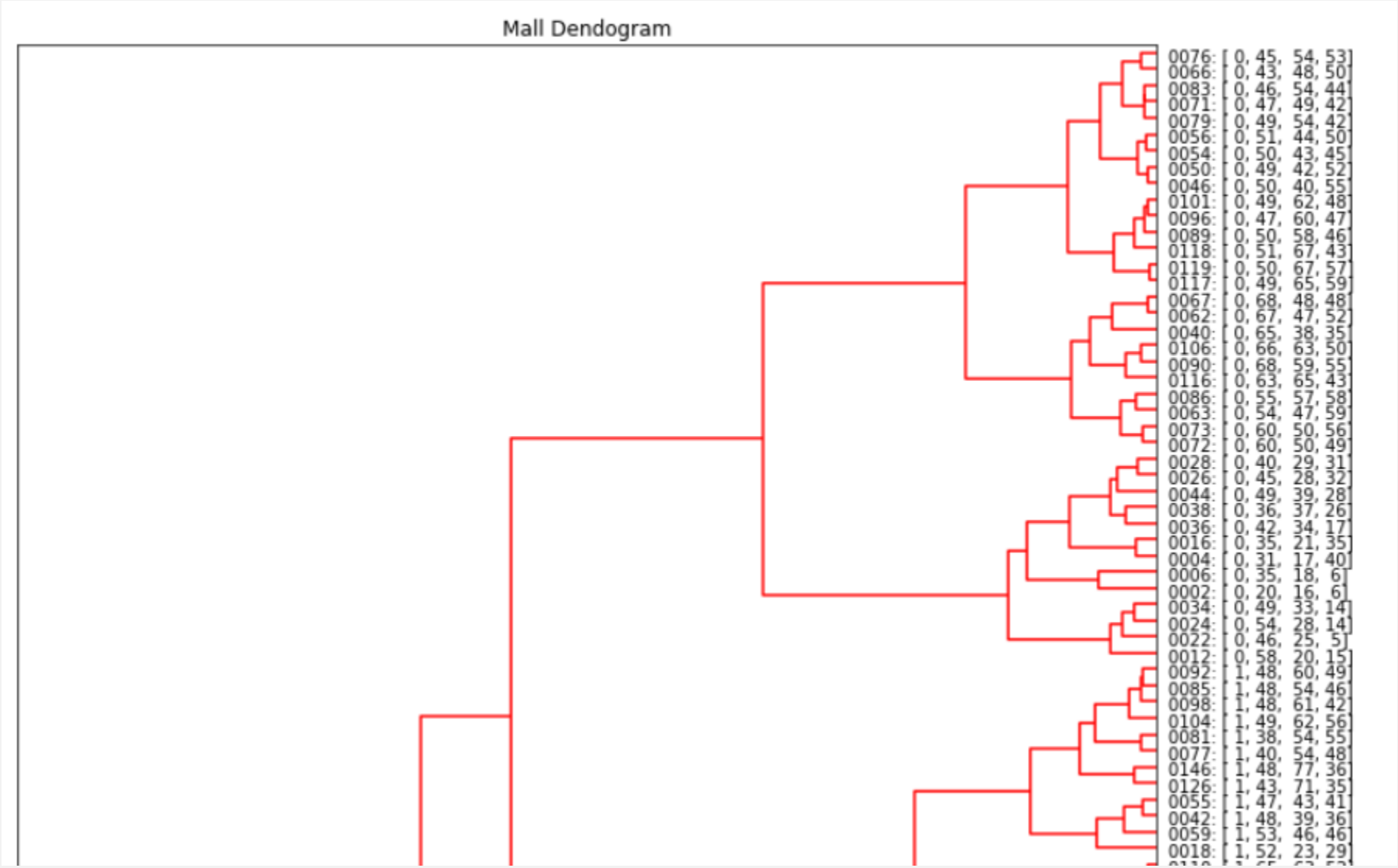
Example: Mall Data



Dendrogram



Dendrogram, On Side, Zoomed, Better Labels



Within-cluster Sum of Squares

- Called ***inertia*** in scikit-learn
- Measures "compactness" of clusters
  - Range of 0, infinite
  - Lower WCSS = more compact cluster = good
  - Higher WCSS = less compact cluster = bad
- The sum of the squared deviations from each instance to the cluster center

$$\mathbf{wcss}(\boldsymbol{c}) = \sum_{\boldsymbol{x_i} \in \boldsymbol{c}} (\boldsymbol{x_i} - \overline{\overline{\boldsymbol{x}}})^{\mathbf{2}}$$

Silhouette Coefficient

- Measures distance between clusters
- Calculates the distance between each instance to the center of the nearest neighbouring cluster
- Range of [-1, 1]
  - 1: instance is far away from the closest neighbouring cluster
  - <1: instance is close to the closest neighbouring cluster
  - 0: instance is on or very close to the decision boundary
- Take average of all instance's silhouette coefficients

Example: Mall Data

```
scaler = StandardScaler()
features = ['AnnualIncome', 'SpendingScore']
X[features] = scaler.fit_transform(X[features])
```

Raw

	AnnualIncome	SpendingScore
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
5	17	76
6	18	6
7	18	94
8	19	3
9	19	72

Scaled

	AnnualIncome	SpendingScore
0	-1.738999	-0.434801
1	-1.738999	1.195704
2	-1.700830	-1.715913
3	-1.700830	1.040418
4	-1.662660	-0.395980
5	-1.662660	1.001596
6	-1.624491	-1.715913
7	-1.624491	1.700384
8	-1.586321	-1.832378
9	-1.586321	0.846310