Deloitte.

Deloitte Data Science Academy

Lesson 2:

Unsupervised learning

Python data exploration

Features, Clustering, Behavioral segmentation, Anomaly Detection

Hands on training for segmentation in Python

13/11/2019

Today's program

1	Introduction to Data Science, Data Science Tools, Python basics, Git – Code collaboration, Pure Python data exploration	HA1
2	**Features and dataset preparation, Clustering, Behavioral segmentation, Hands on training for data exploration and clustering with Python, Anomaly Detection**	**HA2**
3	Regression and Classification, Hands on training for data exploration, classification and regression	HA3
4	Introduction to NLP and Computer Vision, AutoML	HA4


+8.33
0.13
5,221,600
+2.63
3.9
0
0.00
0
14,400
0.00
29
24,300
-0.45
4.44
5,600
1,220,900
+1.00
6,400
-0.77
4,319,600

Features

Features

= Predictors, independent variable

Parameters specifying the datapoint which are used for prediction / distance metrics.

Supervised learning:

- Discussion: What features would you considered for surviving car accident (Binary classification) (Consider there is no limitation on data availability)

Unsupervised learning:

- Discussion: What features would you use for segmentation of e-shop customers?

Features and their transformation

Numerical values

- Leave as it is – standardize by the training inputs
- Bin into categories (e.g. Citizen number) – this allows the model to learn more individual weigths instead of just one – and then proceed as within categorical values
- Binning can be based either on equidistant intervals, quantiles, visual analytics, …

Categorical values

- One hot encoding {0,1,2}: 0 -> [1,0,0], 1 -> [0,1,0], 2 -> [0,0,1]
- Hashed column – aggregate multiple categories together to provide some separation when number of categories is too high
- Crossed column – combine more features together (lat x long together)
- Embeddings – store to lower-dimensional vector (e.g. NLP)

Binary values

- 0/1 values

Features

Selecting the right features

- **KO criteria** (based on business criteria)
- **Computation of relevant predictors**
  - Each predictor represents a more or less complex calculation that transforms data from source tables into a resulting continuous or categorical variable that describes reality at the level of observation.
  - Ratios, trends, differences, aggregated attributes, etc.
- **Feature selection** (near/zero variance variables, Information Value, correlation, permutation-based variable importance, strong inter-correlation between predictors, etc.) and **dimensionality reduction** (e.g., PCA)
- **Data transformations** (normalization, standardization, lumping, dummy encoding, binning etc.)

Features

Scaling

- If variables are not scaled
    - variable with largest range has most weight
    - distance depends on scale
- Scaling gives every variable equal weight
- Scale if
    - variables measure different units (kg, meter, sec, …)
    - you explicitly want to have equal weight for each variable
- Do not scale if units are the same for all variables
- Most often: better to scale

Feature scaling and normalization

Scaling

- Modifying feature to have a range in [0,1]
- Necessary when features with multiple units exists (e.g. clustering fish type by weight and length)

$$x_{scaled,i} = \frac{x_i - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}$$

Normalization

- Modifying distribution to have 0 mean and 1 standard deviation
- Assumes that underlaying distribution is normal

$$x_{norm,i} = \frac{x_i - mean(x)}{std(x)}$$

Features – empty values

Most datasets are imperfect and contain missing data. Options for dealing with this issues are following:

1. **Drop the entries with missing data**
    - Simplest solution, but feasible only when the reduced dataset is sufficient for the task or when there are only few missing values.
2. **Impute the dataset**
    - Fill the missing value by using information about the feature column containg the missing value (e.g. use mean of the values) or use default value
    - Use multiple feature columns to estimate the missing value
3. **EM algorithm**
    - Use Expectation-maximization algorithm to fill the missing values: The algorithm assumes that the data distribution form is known and iteratively uses two steps to estimate the data:

        E-step: estimate the missing values given the distribution parameters
        M-step: estimate the distribution parameters given the data

Python libraries

Numpy, logical indexing, matplotlib

Python libraries

Open jupyter notebook D1_Python_libraries.ipynb

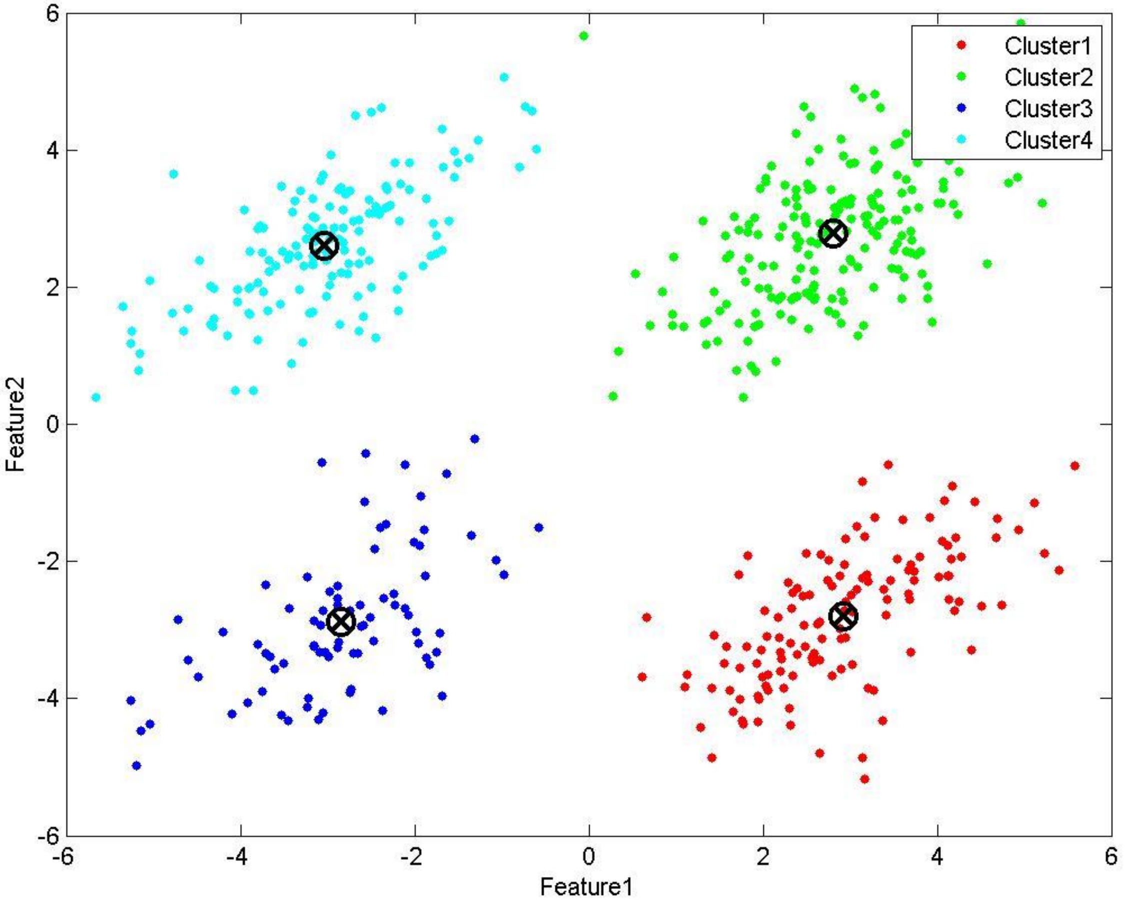
Segmentation

What is segmentation

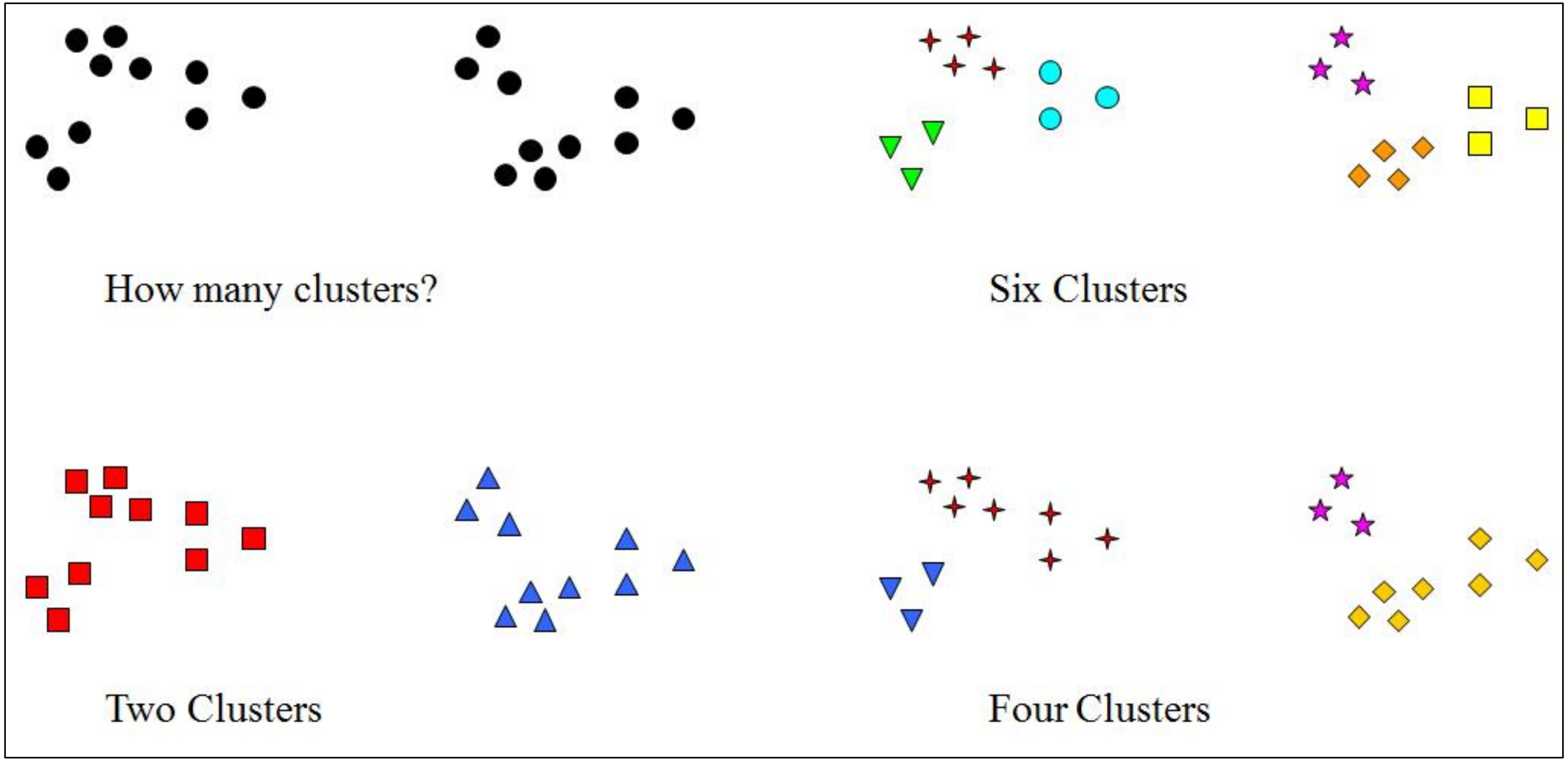
Segmentation (clustering) group large number of data points into segments (clusters) which are

a) homogeneous inside – contain points similar and close to each other,

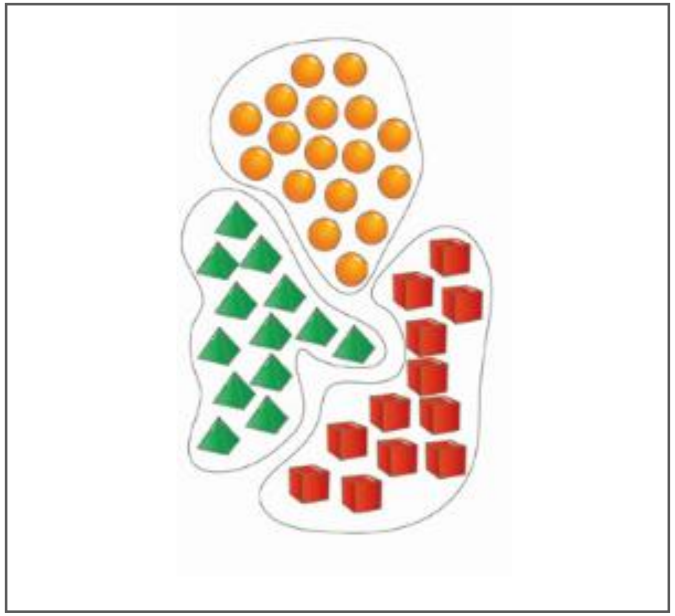
b) different outside – points from different clusters are dissimilar and far from each other.



Clustering can be ambigous

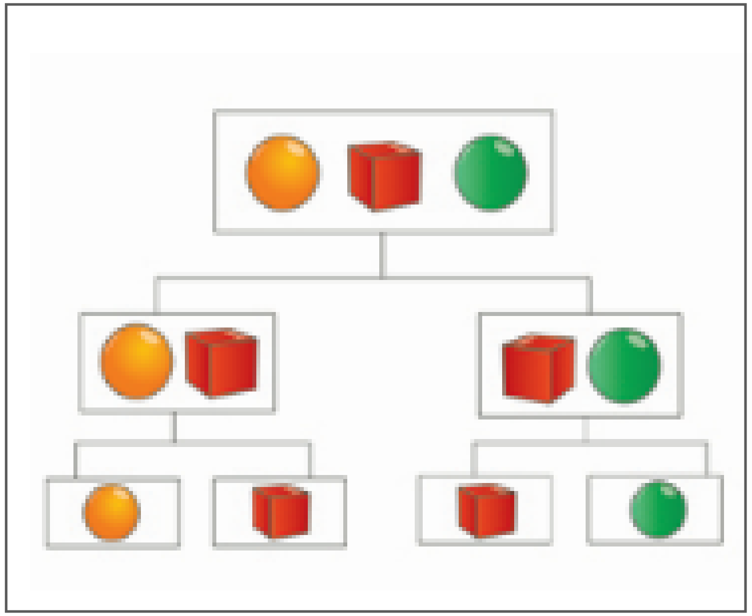


Hierarchical vs non-hierarchical clustering



Non-hierarchical: a division of points into non-overlapping subsets (clusters) such that each point is in exactly one cluster.

Hierarchical: a set of nested clusters organized as a hierarchical tree.

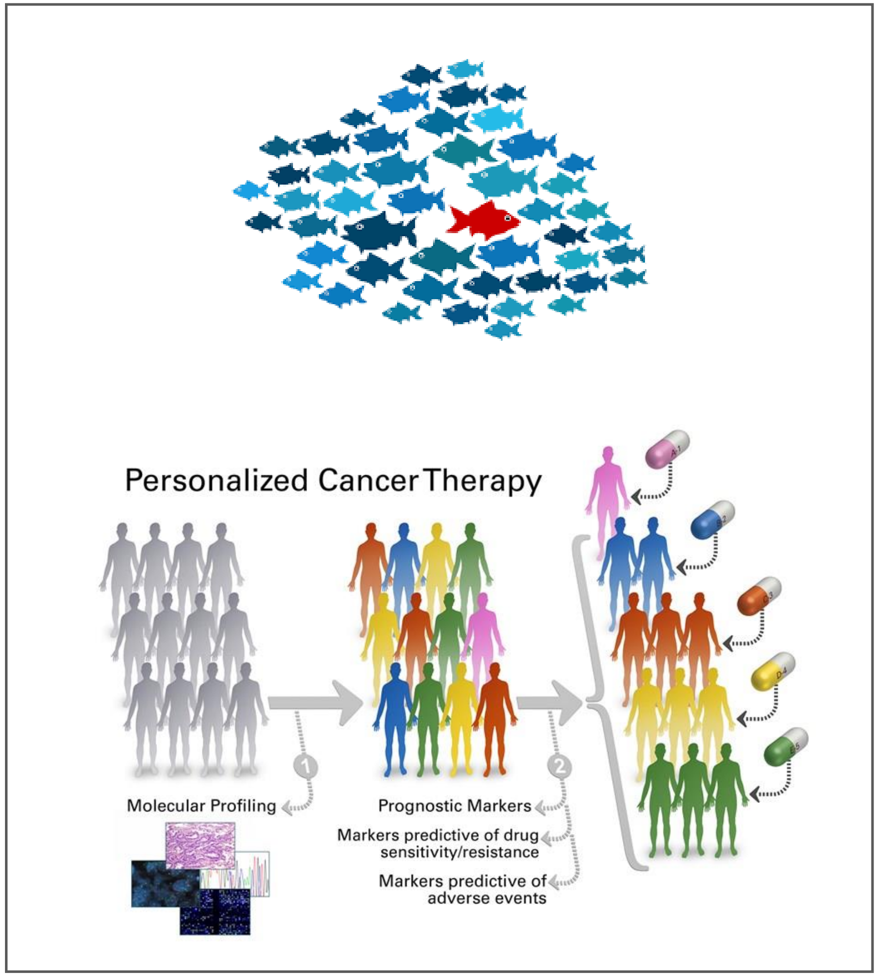


Business use cases

Data exploration	Understand the behavior of your customers through data. Visualization of each segment's behavior helps business make sense of the data.
Service model	Apply different service level by segment based on revenue potential and needs.
Customer lifetime value (CLV)	Markov chain approach to CLV requires granular and actionable segmentation as input. Transitions among segments are at the heart of CLV calculation.
Dynamic pricing	Capturing price sensitivity is key element of good pricing. Elasticity is individual, but it can only be measured on a group of customers. Segmentation is a good compromise.

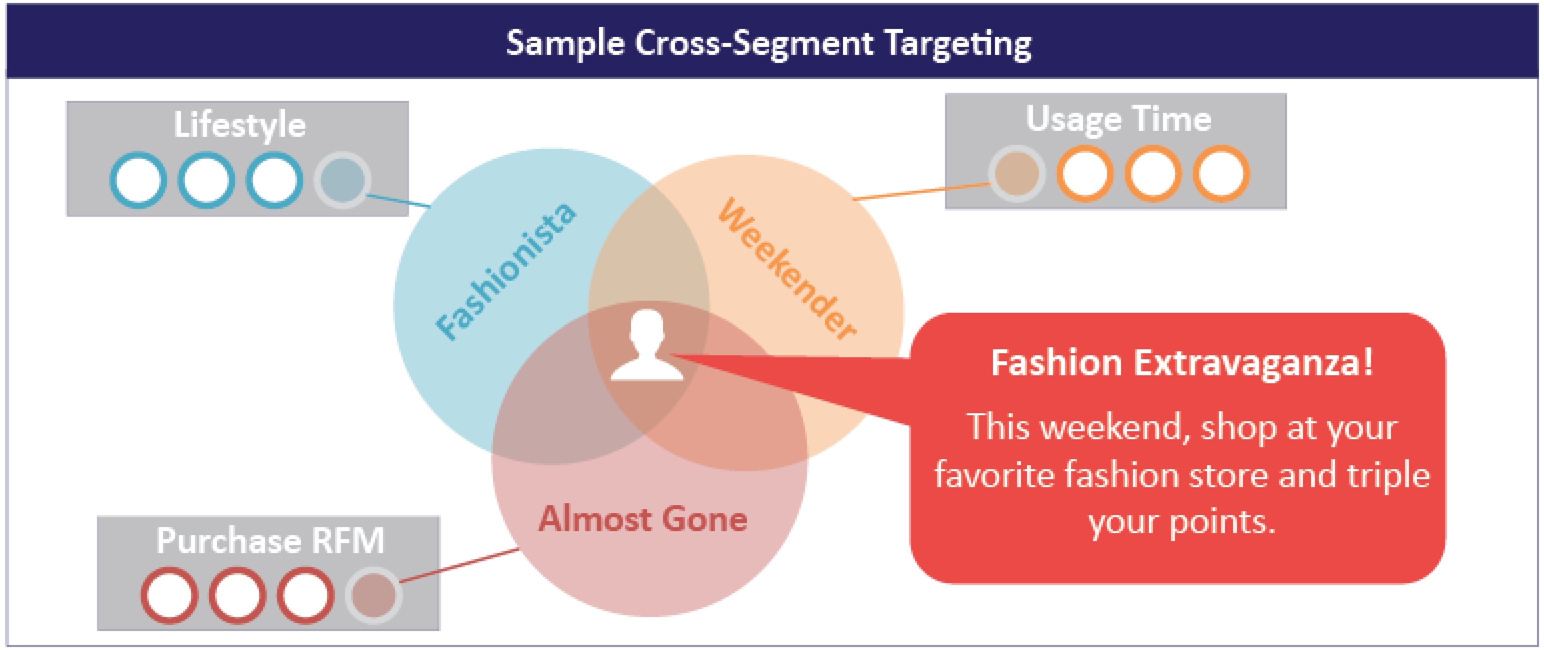
Business use cases

Anomaly detection	Outliers which do not fit into any segment indicate unusual behavior and help to detect fraudsters.
Personalized treatment	In healthcare different patients react in a different way on particular treatment. Finding groups of patients with common treatment profile makes treatment more effective and more personal.



Business use cases

Customer targeting Create personalized offer to each segment.



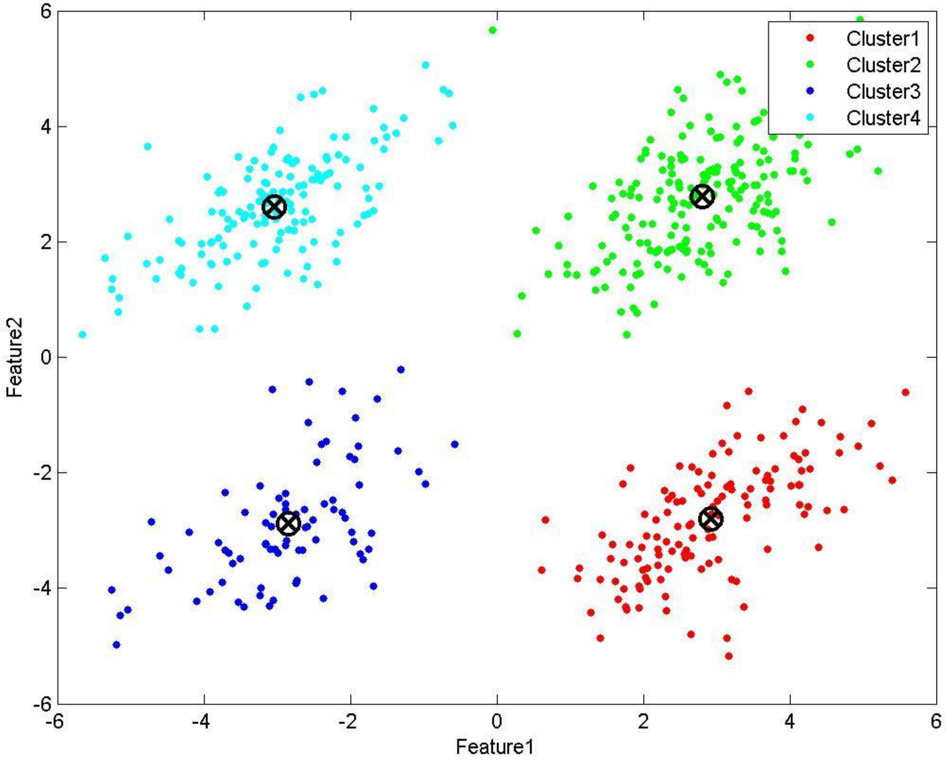
Analytical use cases

Reduce size of your problem	Optimization problem (e.g. marketing optimization) applied on big customer portfolio (e.g. 5 millions) might become unfeasible to solve. Working on 1000 small customer segments with common Probability to Buy and ΔCLV profile makes the problem perfectly feasible and scalable to any portfolio size.
Improve your predictive model	If we have big enough sample for predictive modelling, splitting it into clusters and developing separate model on each cluster gives better predictive performance of the resulting clustered model. Special types of binary classifiers (logistic model tree, LogitBoost) already utilize this opportunity.

Distance metrics

All clustering algorithms rely on a metric in the n-dimensional space

Distance	Note
Euclidean	L2 norm, special case of Minkowski for p=2
Manhattan	L1 norm, special case of Minkowski for p=1
Chebyshev	L∞ norm, special case of Minkowski for $p \to \infty$, also maximum distance
Minkowski	Lp norm. For p<1 the triangle inequality does not hold.
0-1 match	Categorical: 0 iif values match
Normalized Ranks	Ordinal: scale rank to [0, 1] and treat it as interval scaled
Gower	Mixed: Select distance measure for each variable, scale it to [0, 1] and average across variables.



Distance metrics

Gower distance

Gower distance between datapoints i, j is defined as

- $G_{ij} = \frac{\sum_{k=1}^{N} w_{ijk} s_{ijk}}{\sum_{k=1}^{N} w_{ijk}}$,

where

w_{ijk} is the weight of observation k between i and j

s_{ijk} is the distance between i and j on feature k defined as:

$\mathrm{S_{ijk}} = \begin{cases} 0 \ if \ X_{ik} = X_{jk} \\ 1 \ if \ X_{ik} \neq X_{jk} \end{cases}$ for categorical features

and $S_{ijk} = |x_{ik} - x_{jk}|$ for numerical features (assuming the variables are already scaled)

K-means

K-means

Overview

Desired number of clusters K is input. The K-means algorithm aims to choose centroids that minimize the inertia, or within-cluster sum of squared criterion:

$$\sum_{i=1}^{K} \sum_{x \in S_i} \|x - \mu_i\|^2 \to \min_{S}$$

where μ_i is mean of points in S_i and

$$x \in S_i \Leftrightarrow \|x - \mu_i\| = \min_j \|x - \mu_j\|$$

Initial μ_i coordinates are random.

Clusters are Voronoi cells

K-means clustering on the digits dataset (PCA-reduced data)
Centroids are marked with white cross

K-means

Algorithm

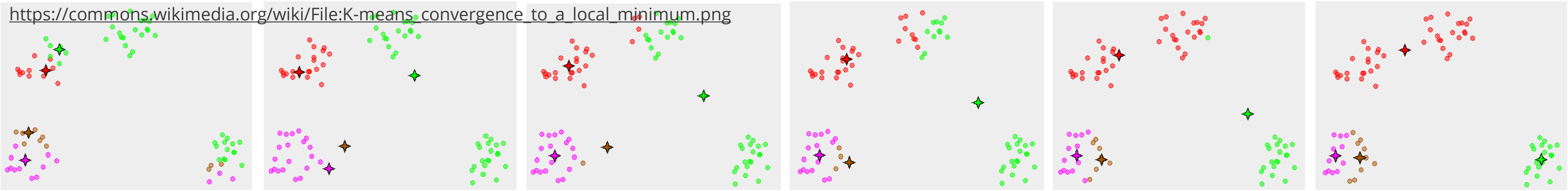
Algorithm steps:

1. Initialize centroid randomly (given |S|)
2. Assign each data point to the nearest centroid
3. Set the centroid to the „middle“ of the assigned points
4. Repeat 2. and 3. until convergence

Initialization:

- Can lead to local optimal solution based on the centroid initialization
- K-means++ (default k-means algorithm in scikit-learn)

https://commons.wikimedia.org/wiki/File:K-means_convergence_to_a_local_minimum.png

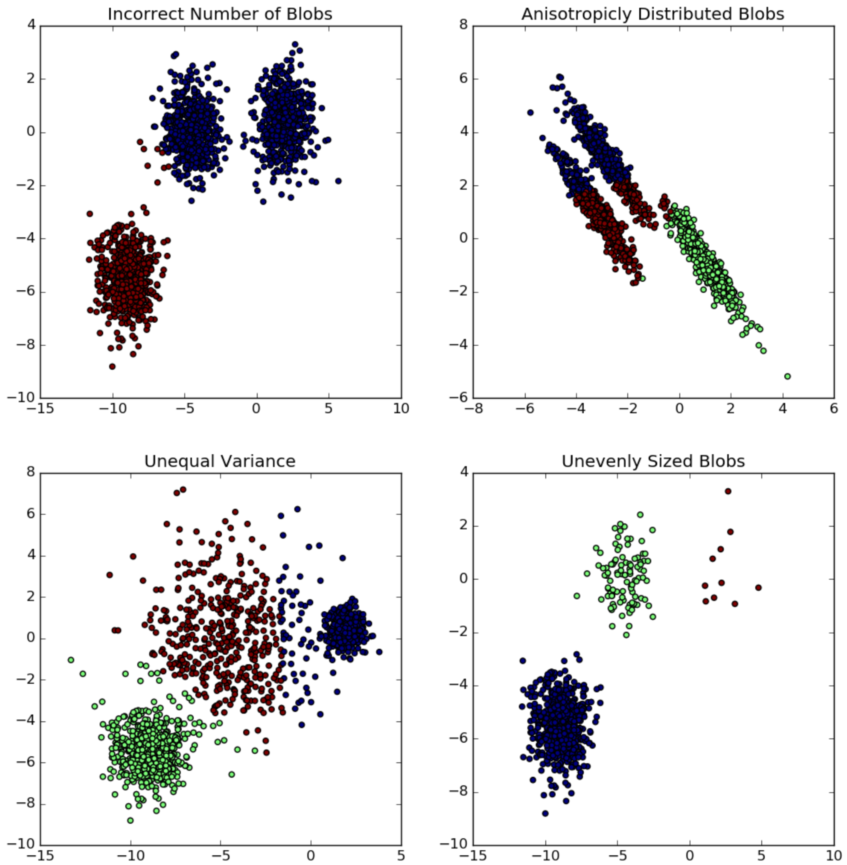


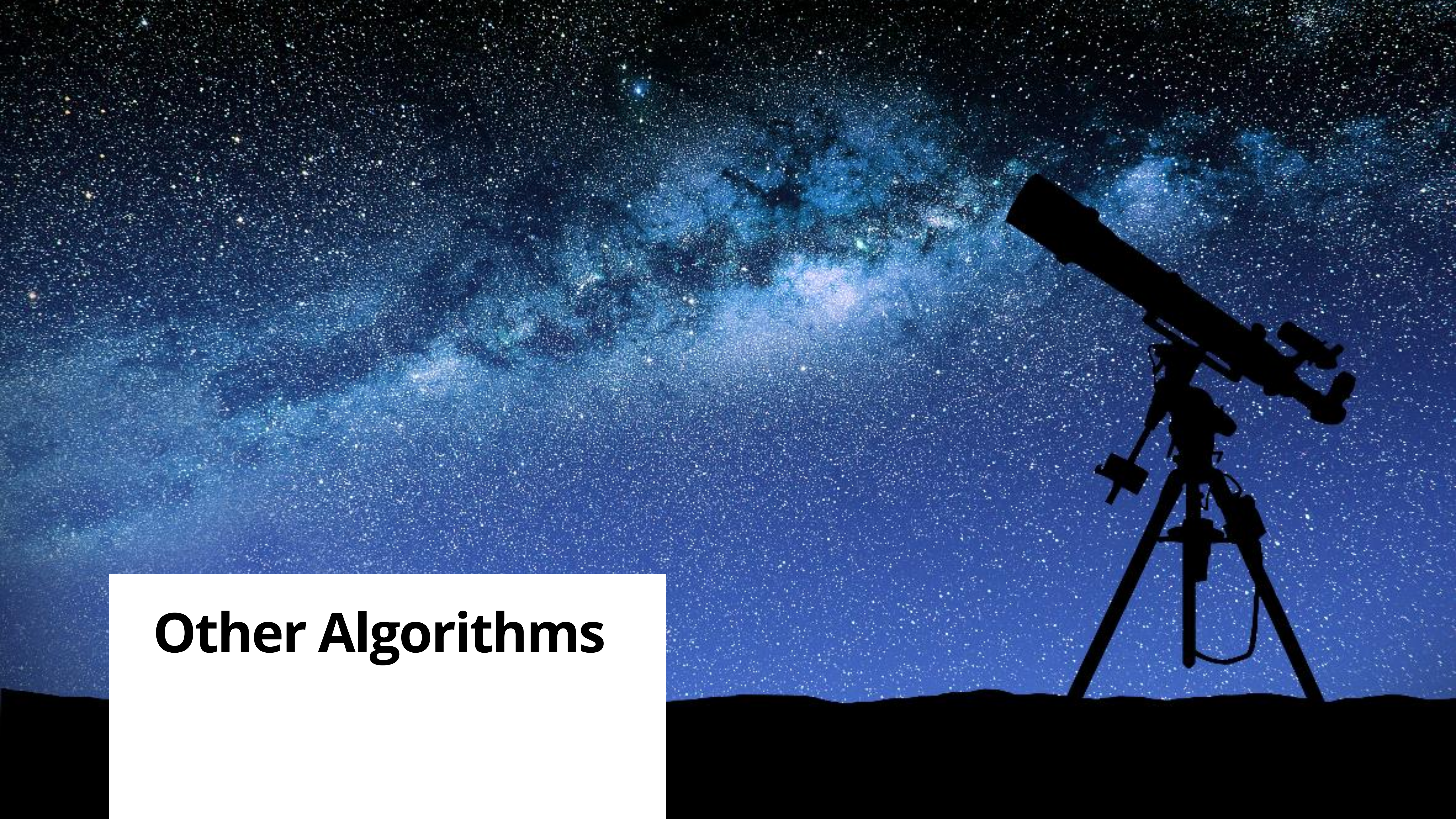
K-means

Dataset properties

Inertia makes the assumption that clusters are convex and isotropic. It responds poorly to elongated clusters, or manifolds with irregular shapes.

Standardization of data is beneficial. Outliers should be capped. Missing values has to be treated. Binning is not recommended.

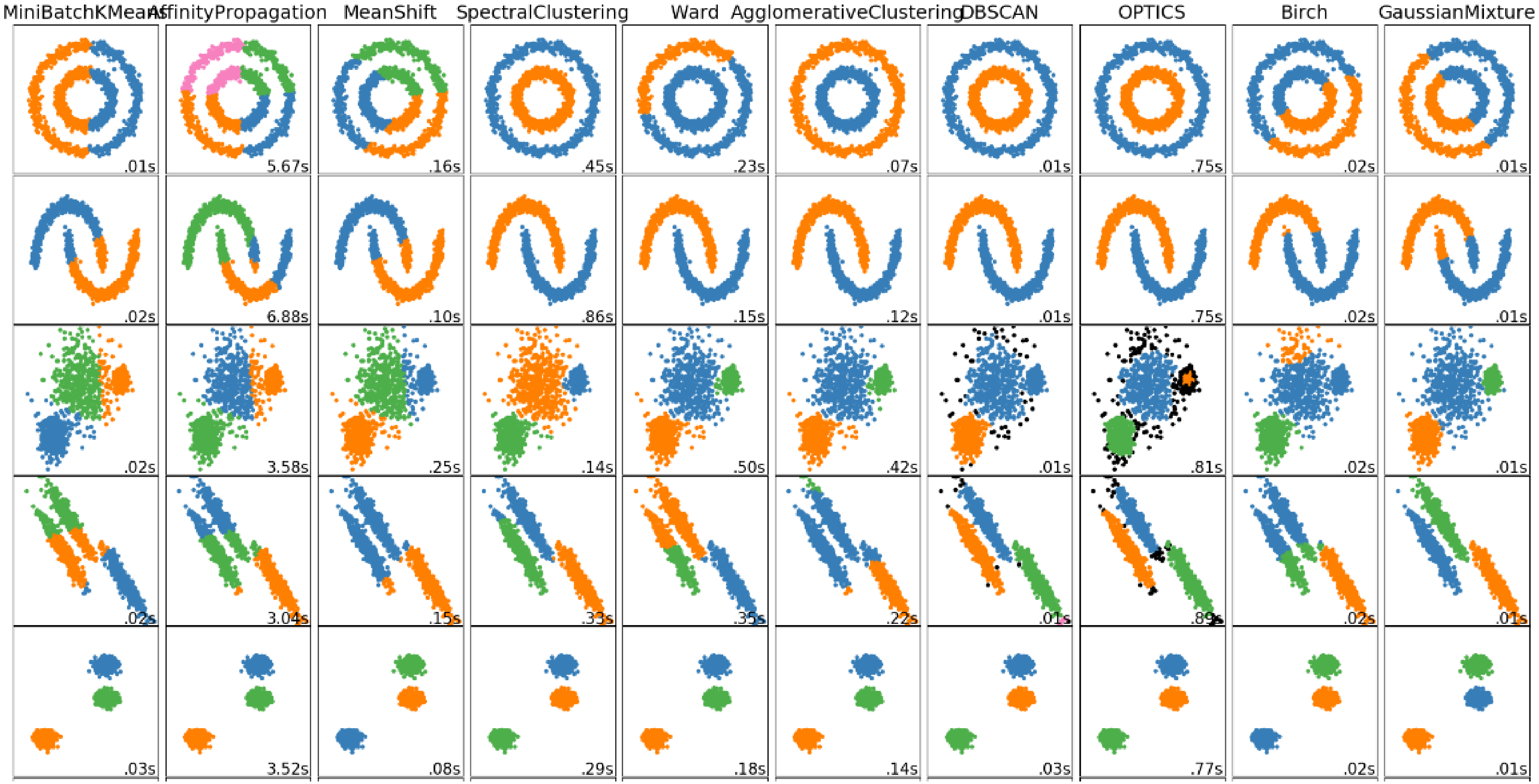




Other Algorithms

Scikit-learn overview

https://scikit-learn.org/stable/modules/clustering.html



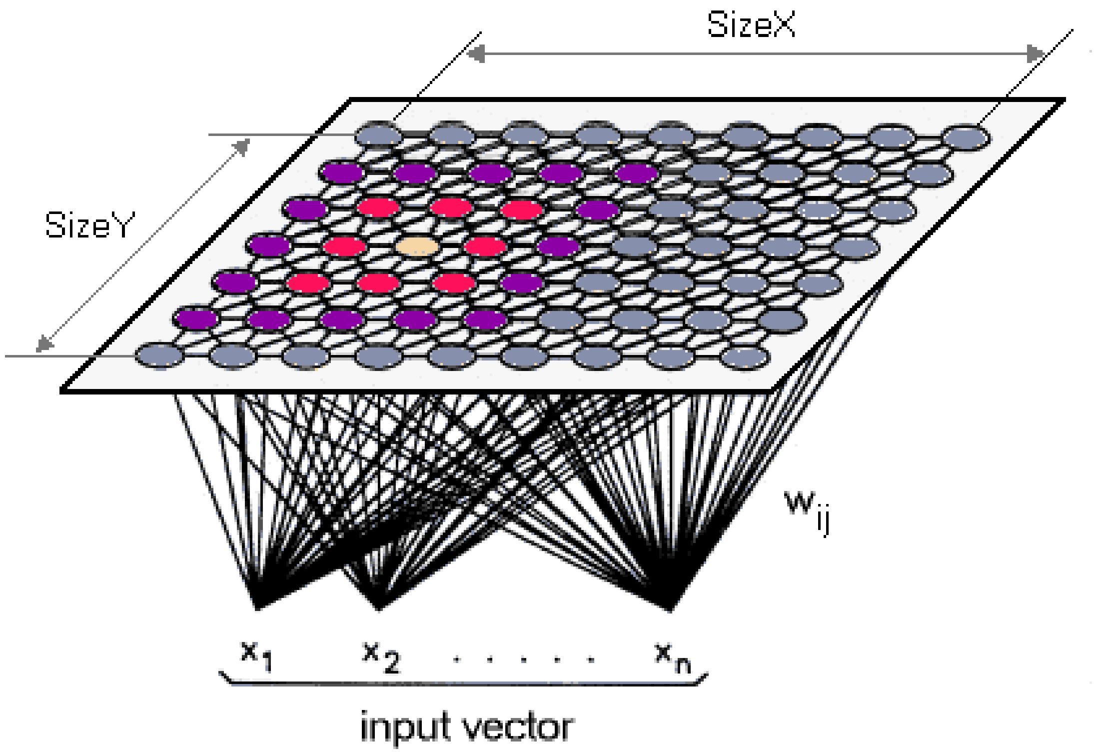
Self-organizing maps (SOM)

Overview

SOM is a type of artificial neural network which provides a way of representing ND data in much lower dimensional spaces - usually 2D, preserving topology. These lower dimension data then can be further used for clustering tasks.

A 1D/2D grid of nodes is the input.

Each node represents cluster centroid, it is initialized randomly and updated in each iteration.



https://github.com/yogonza524/SOM

 |

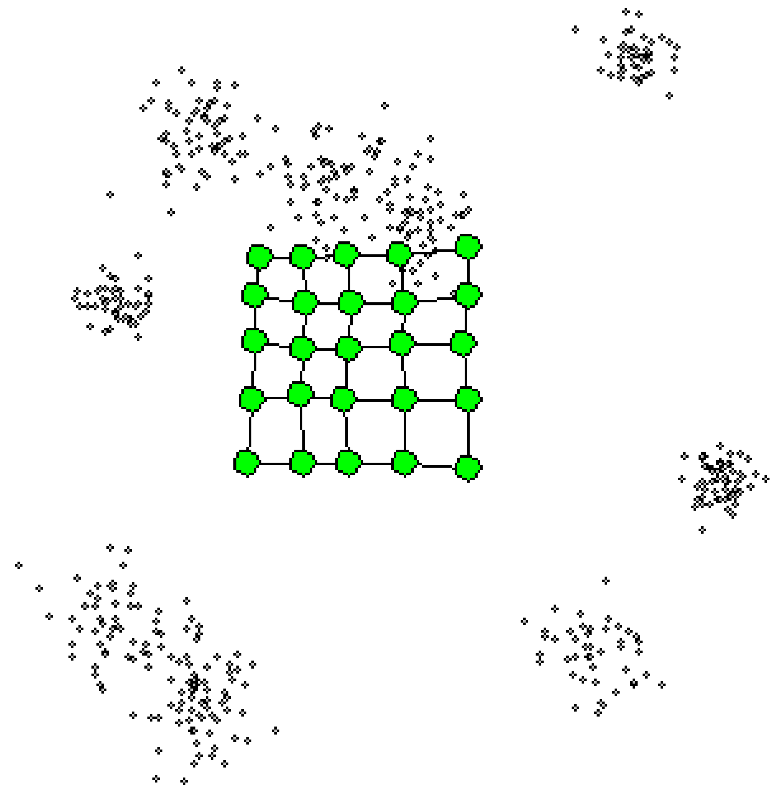
Self-organizing maps (SOM)

Algorithm

In each iteration the cluster centroids are moved closer to the surrounding data. The radius gets smaller following exponential decay.

Steps:

1. Randomize node weight vectors in a map
2. Randomly pick input vector D (datapoint)
3. Use distance metrics to find closest node – best matching unit (BMU)
4. Update weight vectors of the nodes in the neighbourhood of BMU (including BMU) by pulling them closer to D
5. Repeat from 2. until iteration limit reached or convergence



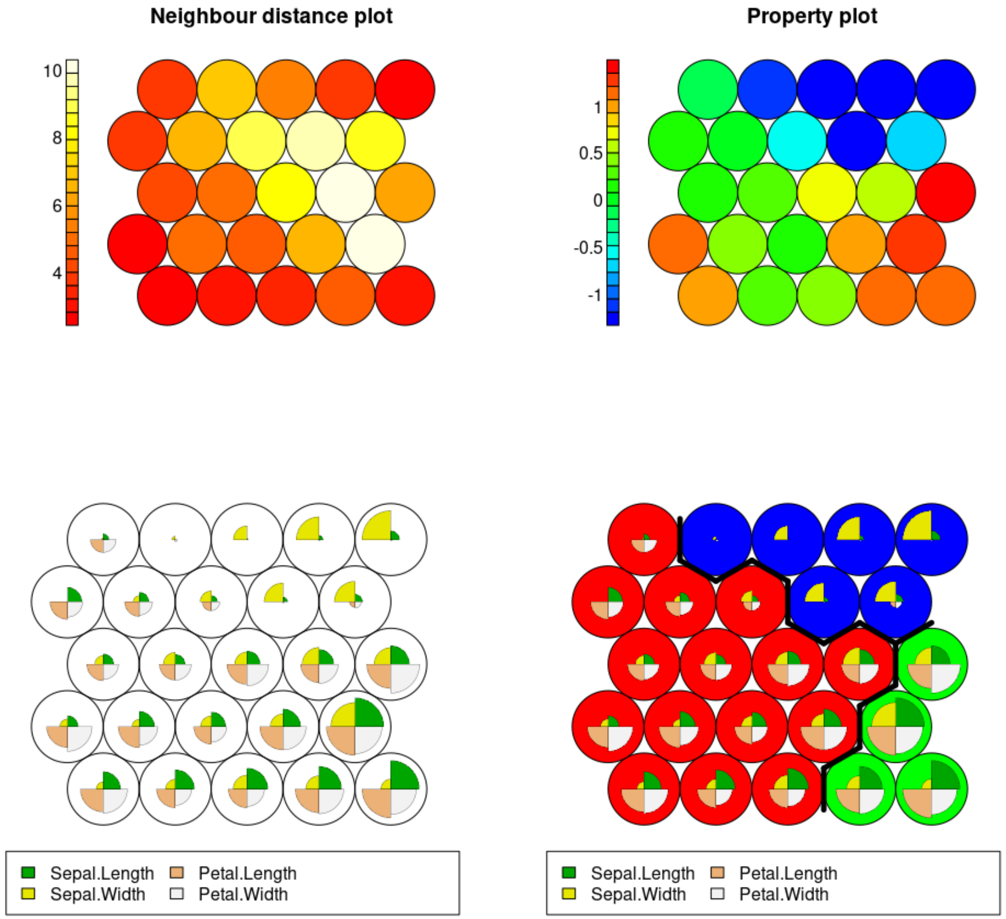
http://http://bioinf.fz-borstel.de/mchips/course/som.gif

Self-organizing maps (SOM)

Clustering

Typical visualization is in the form of a hexagonal grid. Clustering is done on the map after training is performed.

Usually a grid of hundreds or thousands nodes is trained and clustering on the map itself is performed afterwards.



https://en.proft.me/2016/11/29/modeling-self-organising-maps-r/

DBSCAN

Clustering

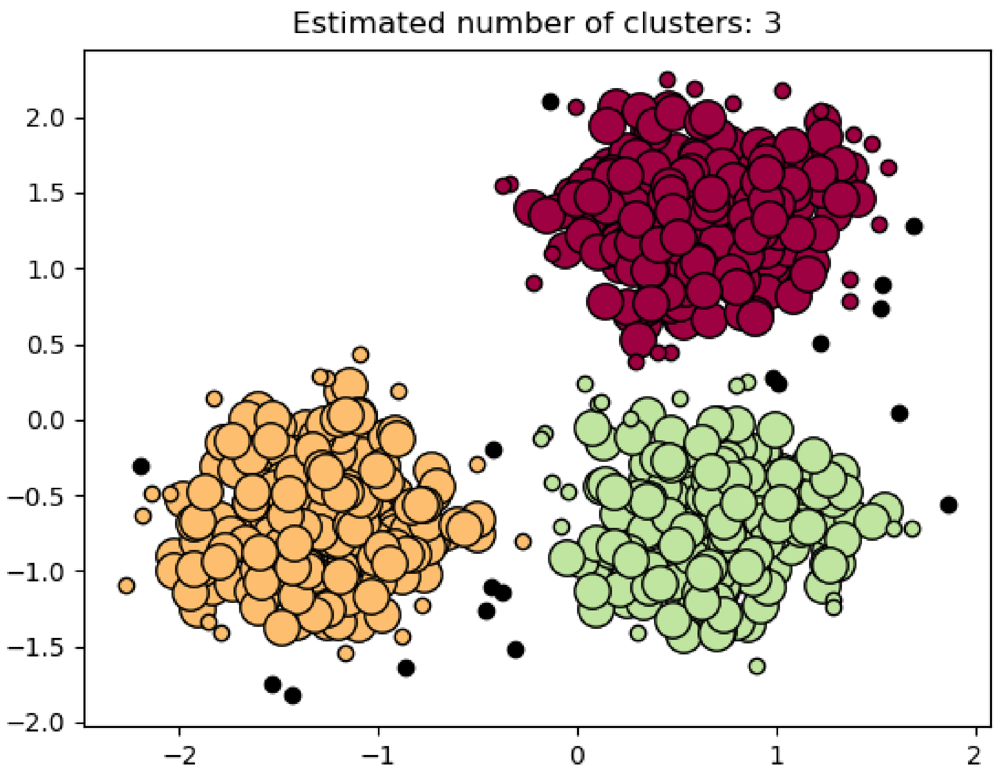
The DBSCAN algorithm views clusters as areas of high **density** separated by areas of low density. Due to this rather generic view, clusters found by DBSCAN can be any shape, as opposed to k-means which assumes that clusters are convex shaped.

Automatically determines best number of clusters.

Special cluster for **outliers**.

Parameters:

- ε –max distance between two samples to be considered in same cluster
- minimum number of points required to form a dense region.



Hierarchical clustering – Agglomerative

Clustering

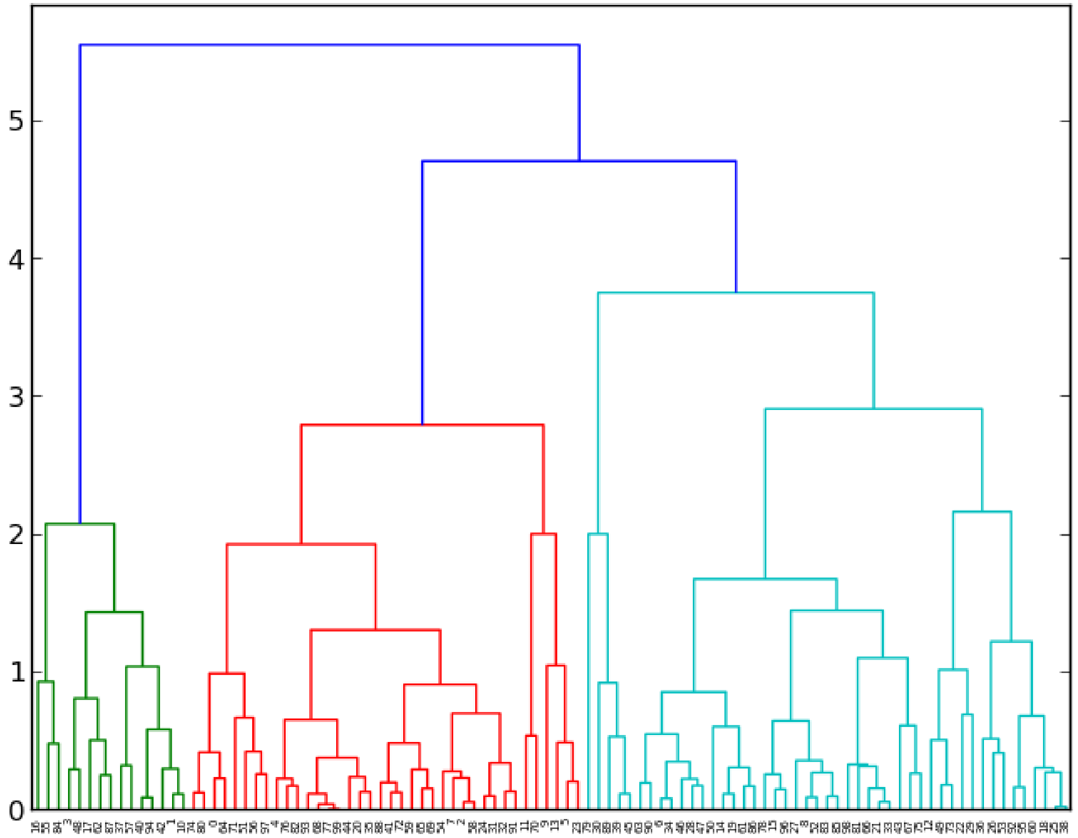
Agglomerative hierarchical clustering performs a bottom up approach: each observation starts in its own cluster, and clusters are successively merged together. In each step one merge is done and the **linkage criteria** determines the metric used for the merge strategy:

Single	Minimizes the distance between the closest observations of pairs of clusters
Average	Minimizes the average of the distances between all observations of pairs of clusters.
Complete	Minimizes the maximum distance between observations of pairs of clusters.
Ward	Minimizes the inertia of the merged clustering (sum of squared differences from cluster centroids). In this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.

Hierarchical clustering – Agglomerative

The hierarchy of clusters is represented as a tree called **dendrogram**. Vertical distances are proportional to objective function changes with respect to the selected linkage.

User decides on **grouping threshold** which yields the final segmentation.



Evaluation and visualization

Clustering performance evaluation

How good is my clustering?

Many performance measures presented in literature are based on comparing cluster labels with "ground truth" class labels, which is **misleading and wrong**. Clustering is not classification!

Measures not assuming labeled dataset is e.g. Silhouette Index, Calinski-Harabaz Index and Davies-Bouldin Index and others.

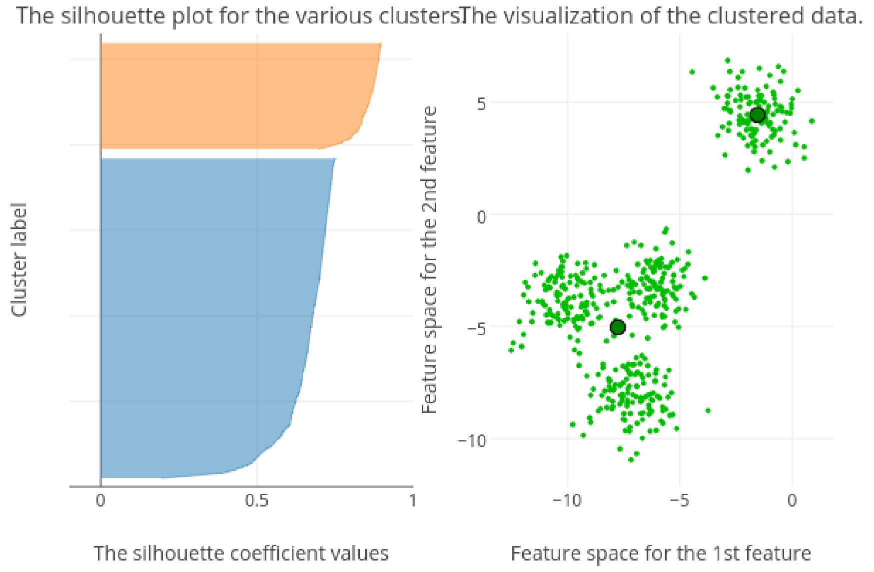
Silhouette coefficient

- *Cohesion* $a(x)$: average distance of x to all other vectors in the same cluster.
- *Separation* $b(x)$: average distance of x to the vectors in other clusters. Find the minimum among the clusters.
- *silhouette* $s(x)$:

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

- $s(x) = [-1, +1]$: -1=bad, 0=indifferent, 1=good
- Silhouette coefficient (SC):

$$SC = \frac{1}{N}\sum_{i=1}^{N} s(x)$$

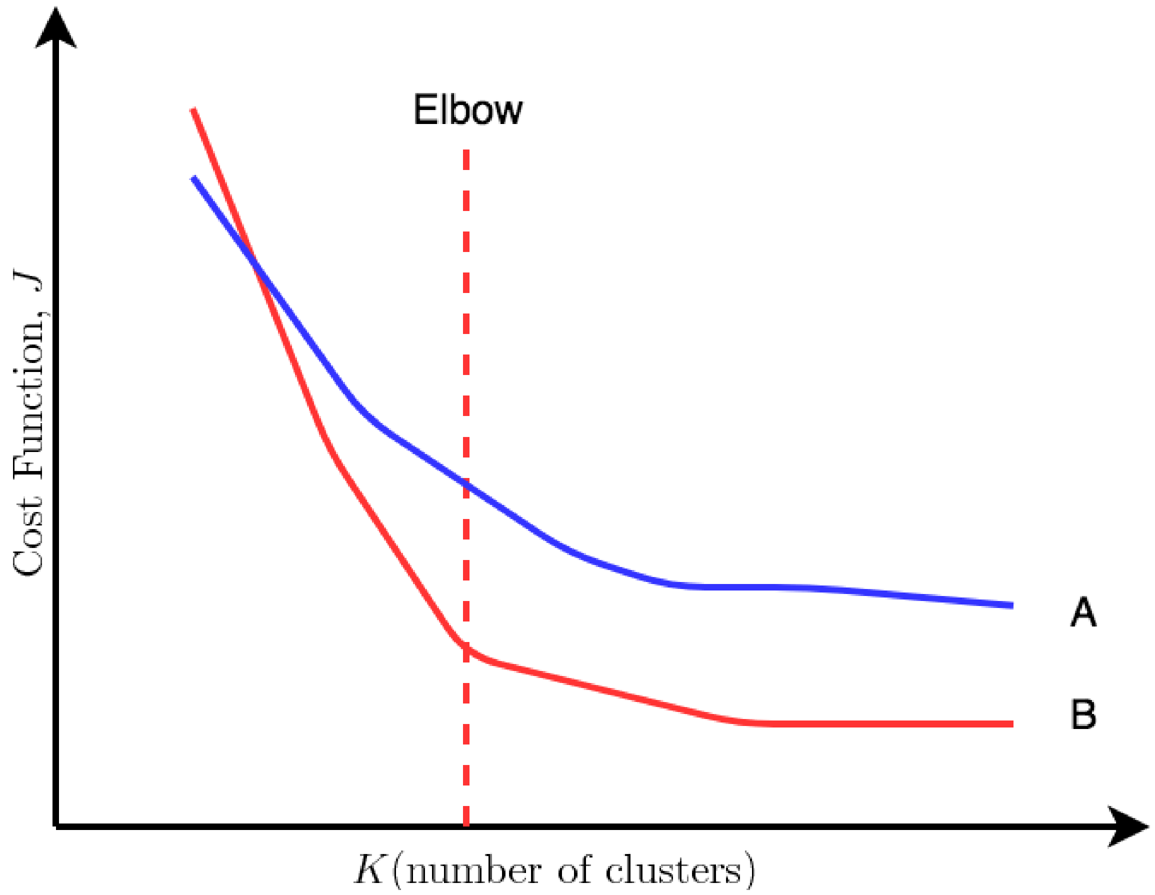


Elbow rule

How many segments should I have

If you are not sure how many segments to have, you can use elbow rule to determine the point where the cost function (e.g. inertia) becomes flat.

Elbow rule was originally designed for k-means, but it is usable for other clustering algorithms as well, also for other cost functions.



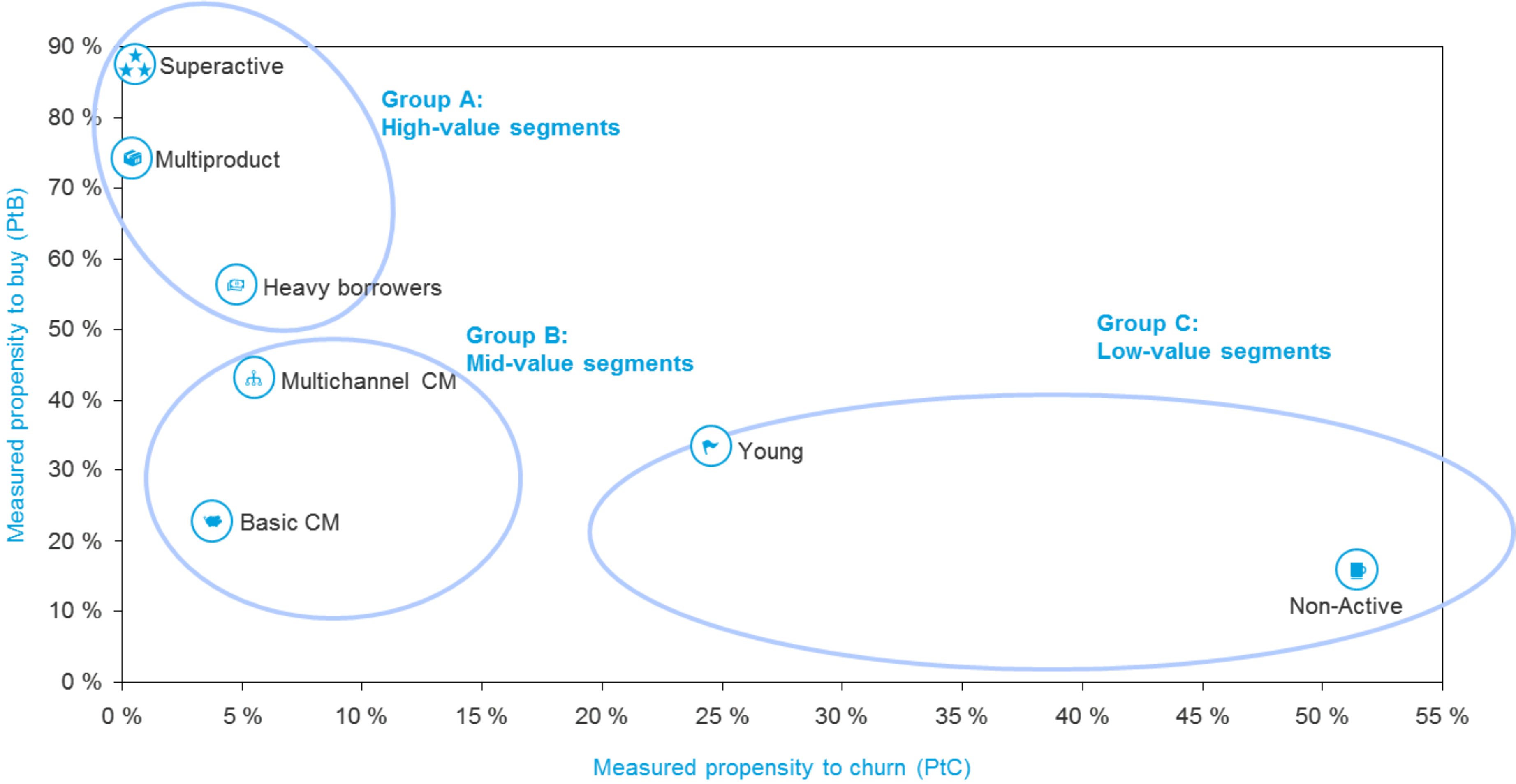
How to present segmentation results

Value of the segment		
Low	**Medium**	**High**
NON-ACTIVE	MULTICHANNEL CM	MULTIPRODUCT
YOUNG	BASIC CM	HEAVY BORROWERS SUPERACTIVE

How to present segmentation results

	MULTIPRODUCT	HEAVY BORROWERS	SUPERACTIVE
Number of customers	23 %	5 %	3 %
Average income	47k PLN	53k PLN	146k PLN
Description	• Generate 53 % of KSS income • 2/3 with 4 or more products	• Nearly 50 % older than 11 years on the market but shorter than 4 years with ING • 65 % penetration on WC loans and 43 % on investment loans • Half with total lending balance above 600k PLN	• 75 % generate above 31k PLN of income p.a. • Oldest companies with the longest relationship with ING • Highest penetration on WC loans, TF and FM products • Most frequent users of BOL
PROPENSITY TO BUY	74,2%	55,9%	87,5 %
PROPENSITY TO CHURN	0,8%	5,0%	1,0%

How to present segmentation results

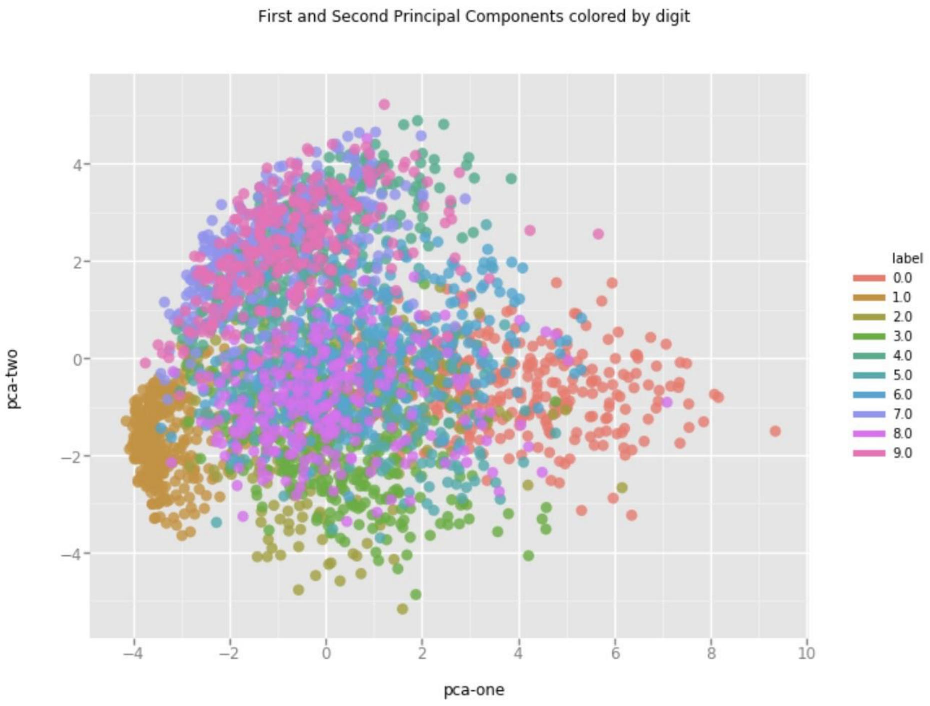


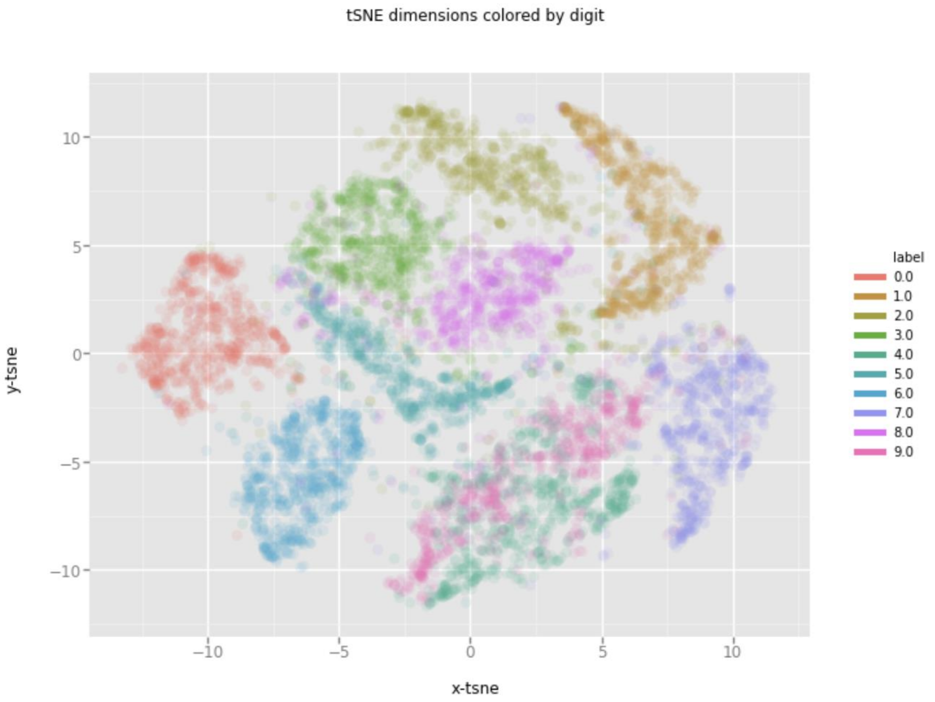
How to present segmentation results

Scatter plot of individual data points colored by segment labels

How to organize data points on 2D plane? One popular option is to use Principal Component Analysis (PCA) and plot against first two components.

Modern (2012) approach is to use t-SNE algorithm (winner of the Merck Viz Challenge) specifically designed for this purpose.



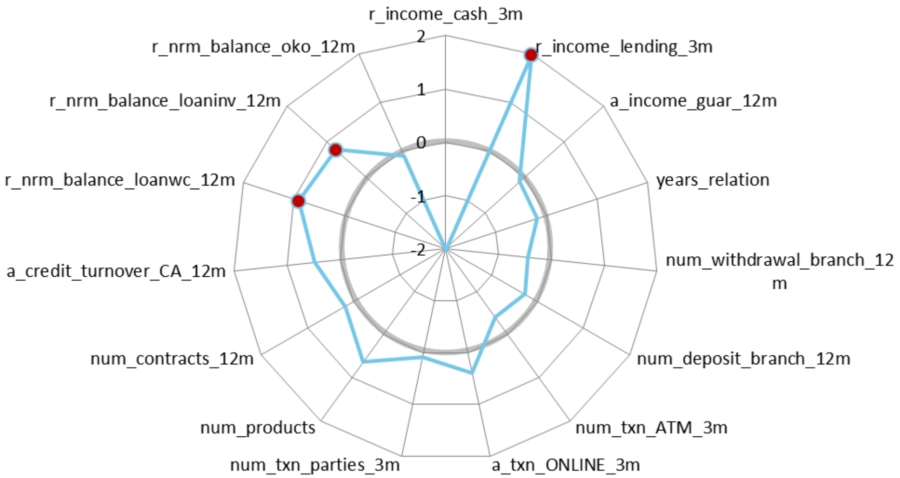


How to present segmentation results

Radar plot

Heavy borrowers (4%)

- Companies from this segment have a combination of working capital and investment loans
- Although they have a number of other products they bring most of the profit for ING on these loans
- They have high propensity to buy new products, especially loans and cash management
- Among profitable segments they have relatively higher churn rate



 |

How to present segmentation results

Transition matrix

Show **segmentation dynamics**. With a fixed **segmentation pipeline** you score the population in two or more points in time and measure the transitions between segments. Can be used for determining marketing campaing targets, CLV, etc.

Segmentation pipeline means function which takes input raw segmentation variables and returns segment label, including all transformations such as imputation, capping or scaling.

Segment(t)	Segment (t+1)		
	0	1	2
0	0.550	0.250	0.200
1	0.050	0.870	0.080
2	0.000	0.000	1.000

$$TM(i,j) = P[Segment(t+1)=j \mid Segment(t)=i]$$

Python clustering

scikit-learn, pandas

Python clustering

Scikit-learn, pandas, dataset preparation and clustering

Open jupyter notebook D2_Clustering_penguins.ipynb

Outliers Detection

Discussion

- What is an outlier (anomaly)?
- Name situation in which anomaly should be removed during pre-processing of the dataset and one in which anomaly carries informantion.
- Try to come up with some ways/concepts how outliers can be detected.
- What is the difference between supervised and unsupervised anomaly detection?
- Can you think of visualization that would show outliers?

Outliers Detection

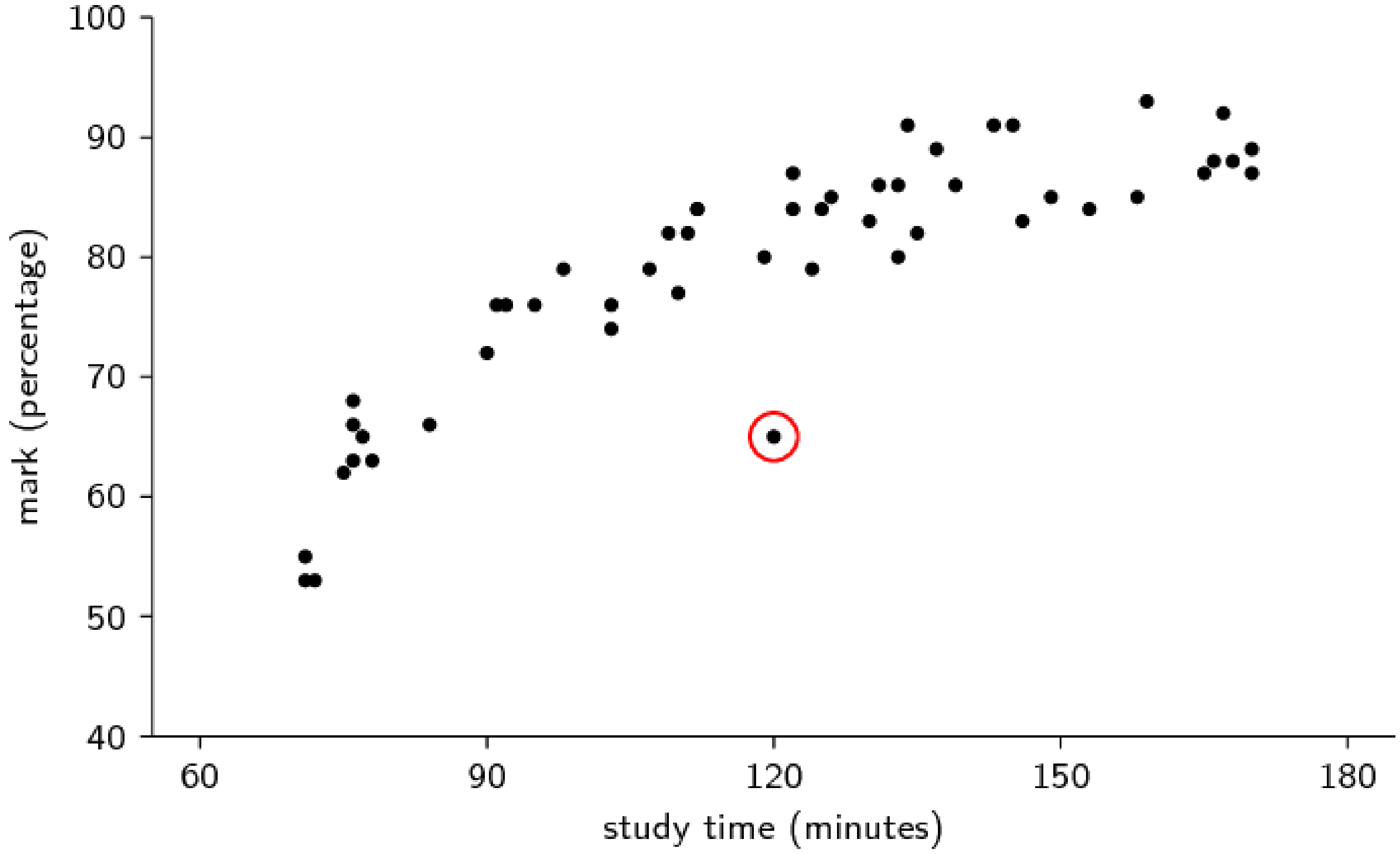
Introduction

- Outliers/anomalies
  - Set of data points that are considerably different than the remainder of the data
  - Definition and common sense implies that anomalies are rare

- Sources
  - Data errors
  - Natural variation

- Applications:
  - Credit card fraud detection, Stolen accounts (google), Network intrusion detection, Fault detection, Data cleansing and exploration, Predictive maintanance

Outliers Detection



https://www.siyavula.com/read/maths/grade-11/statistics/11-statistics-06

Challanges

- **Number of attributes**
  - Some anomalies are defined in terms of a single attribute
  - However, object may not be anomalous in any one attribute
  - Finding anomalies using all atributes is hard – noisy or irrelevant attributes or object is only anomalous w.r.t. other attributes

- **Anomaly scoring**
  - Many techniques provide only binary categorization (e.g. classification based approaches)
  - Other approaches assign scores to all points – there is a need to figure out correct threshold
  - How many anomalies are there in the data?

Challanges

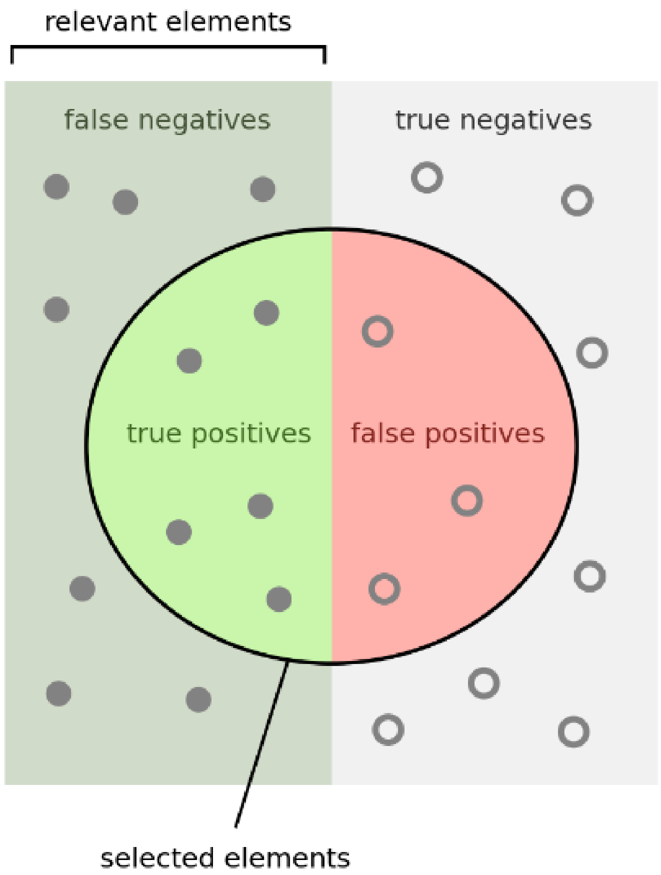
- **Evaluation**
  - How to measure performance of selected approach?
  - Quiz: Define following metrics according to their definitions:
    - Accuracy = Number of correct predictions
    - Precision = How many selected items are relevant?
    - Recall = How many relevant items are selected?

cnt		Predicted	
		1	**0**
Actual	**1**	tp	fn
	0	fp	tn

- **Efficiency**
  - Many techniques provide only binary categorization (e.g. classification based approaches)
  - Other approaches assign scores to all points – there is a need to figure out correct threshold
  - How many anomalies are there in the data?

Evaluation

- Accuracy = Number of correct predictions
  - (tp + tn) / (tp + fp + tn + fn)
- Precision = How many selected items are relevant?
  - tp / (tp + fp)
- Recall = How many relevant items are selected?
  - tp / (tp + fn)

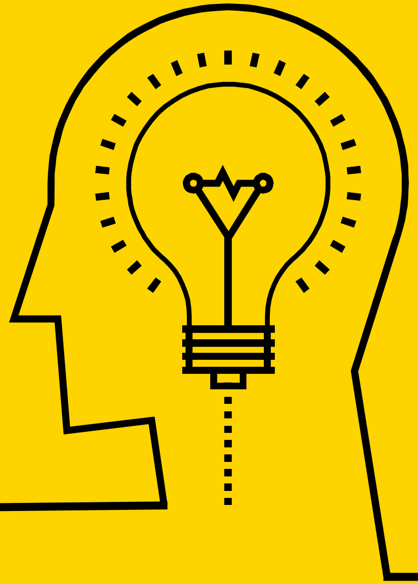


https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

Anomaly Detection Methods

1. **(Graphical methods)**
2. **Model-based methods**
   - unsupervised (points not fitting well or points distorting the model)
     - statistical distribution, clusters, regression, geometric, …
   - supervised (anomalies as a rare class)
3. **Proximity-based**
   - anomalies are points far away from other points
4. **Density-based**
   - low density points are outliers
5. **Pattern matching**
   - profiles/templates of atypical but important events defined beforehand

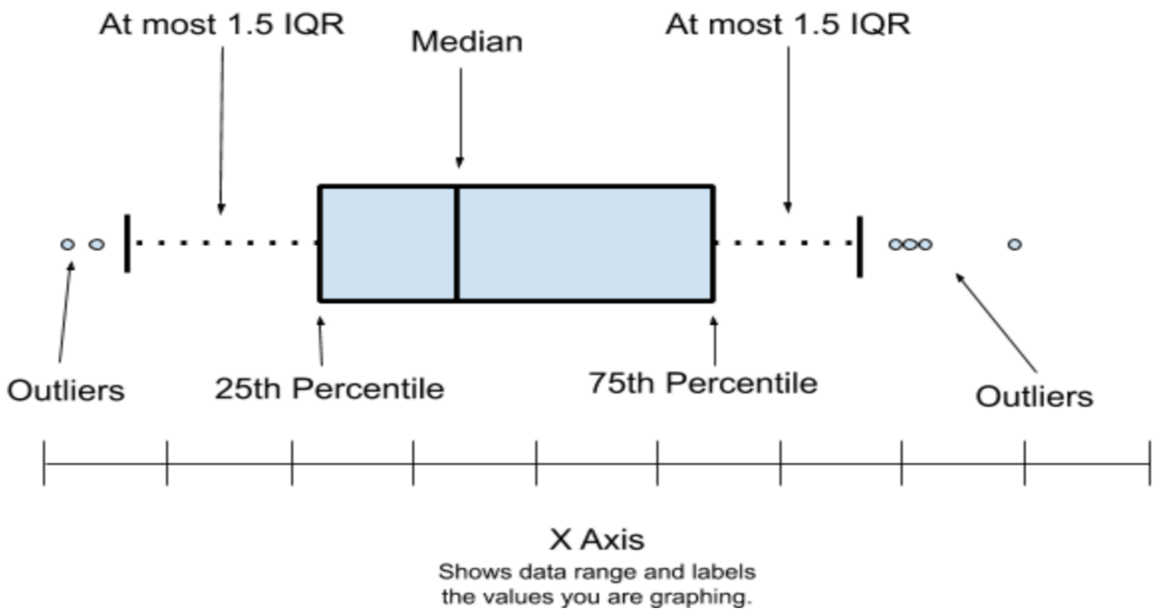
Graphical methods

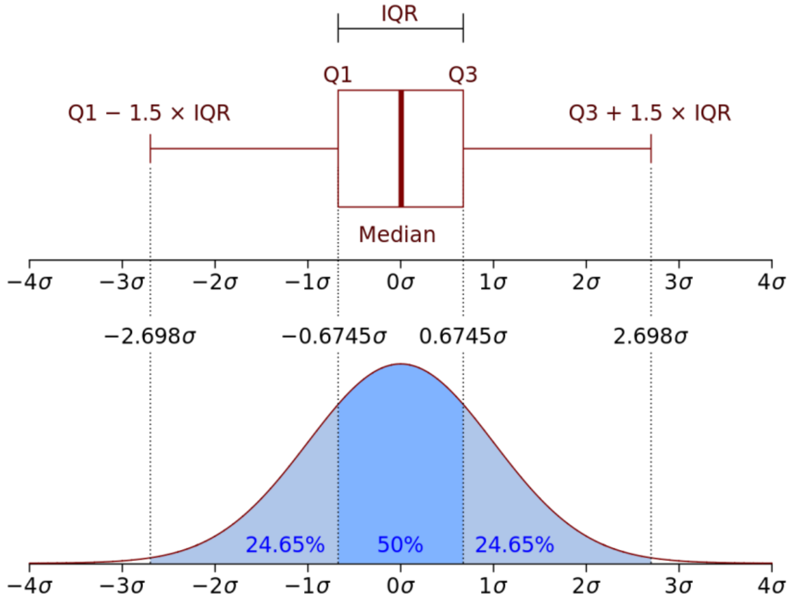


Graphical Approaches

Box plot

- Carries a lot of information – great for data exploration purposes
- 1-D

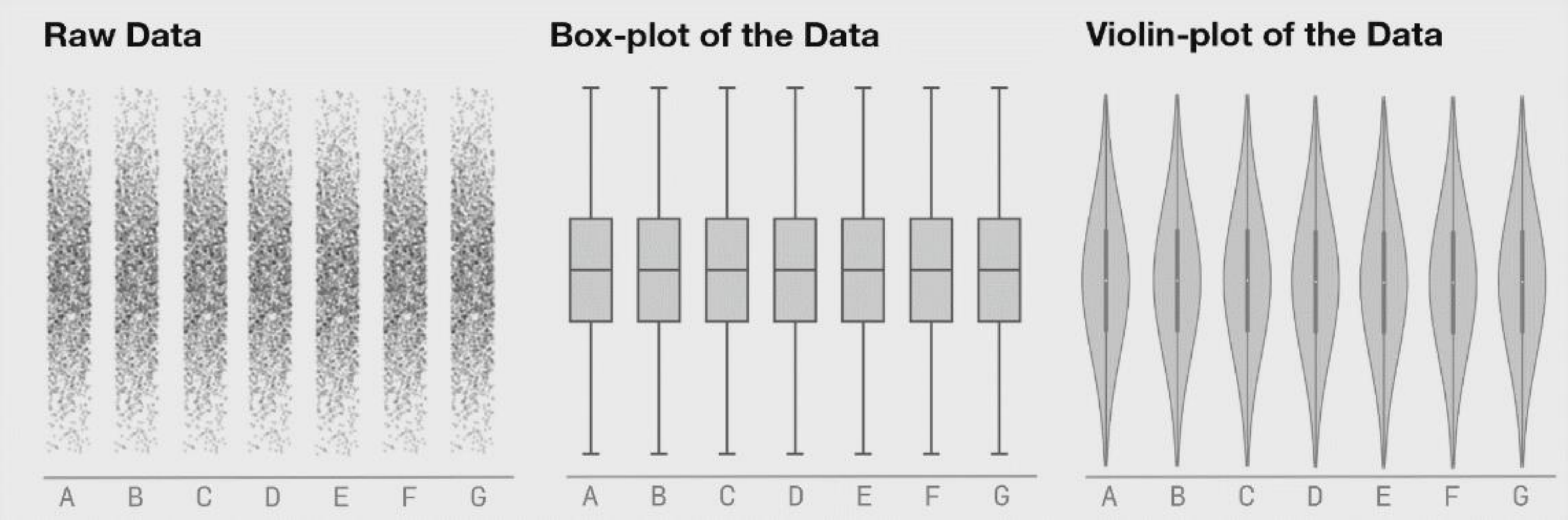




https://publiclab.org/notes/mimiss/06-18-2019/creating-a-boxplot-to-identify-outliers-using-codap
https://en.wikipedia.org/wiki/Interquartile_range

Graphical Approaches

Violin plot

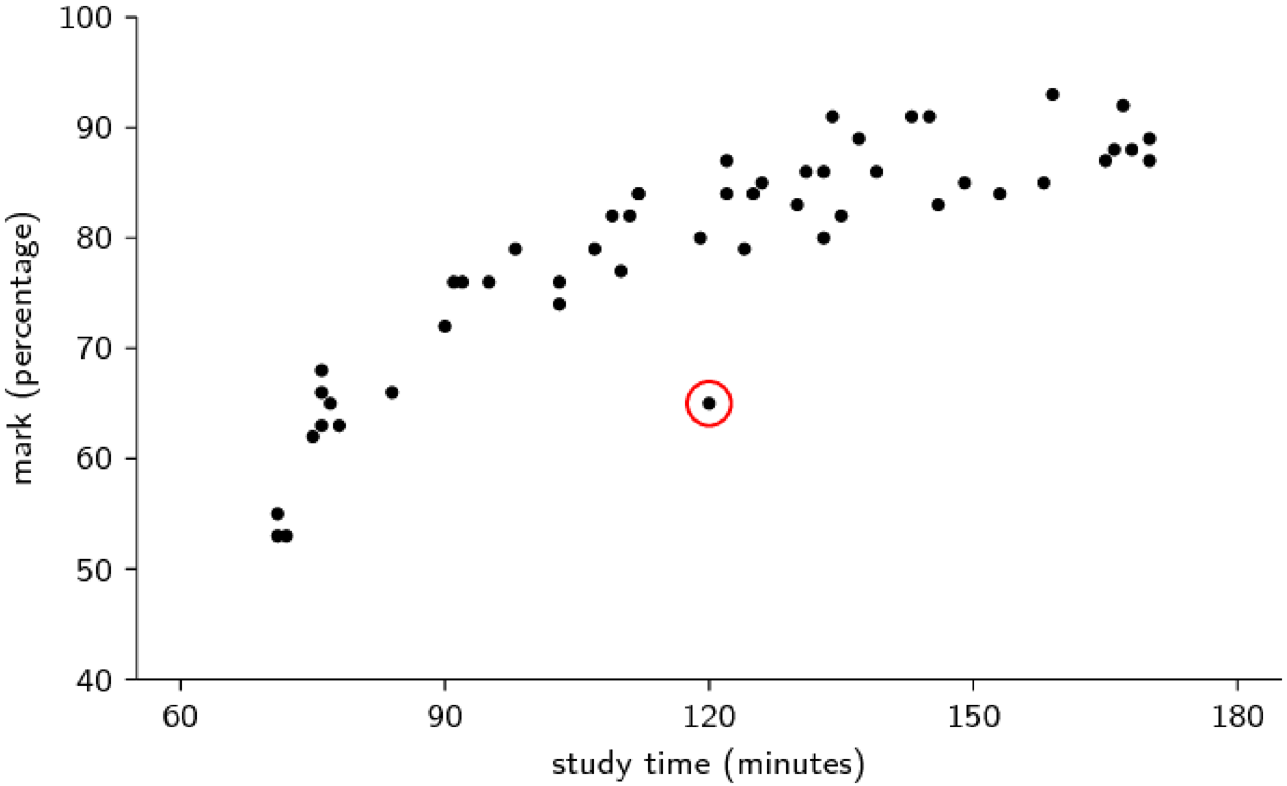


https://blog.bioturing.com/2018/05/16/5-reasons-you-should-use-a-violin-graph/

Graphical Approaches

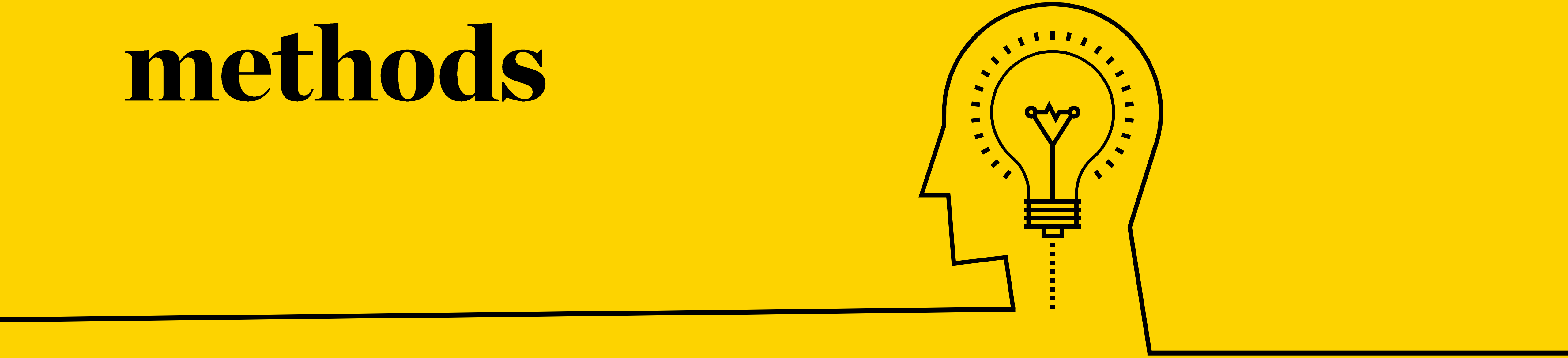
Scatterplot

- Subjective
- Manual
- 2D/3D



https://www.siyavula.com/read/maths/grade-11/statistics/11-statistics-06

Model based methods



Model-based Methods

Statistical Approach

Definition:

- Outlier is an object that a low probability w.r.t. a probability distribution model of the data.

Anomaly score function:

- Given a data instance x from a dataset D: $f(x) = 1/P(x|D)$

General flow:

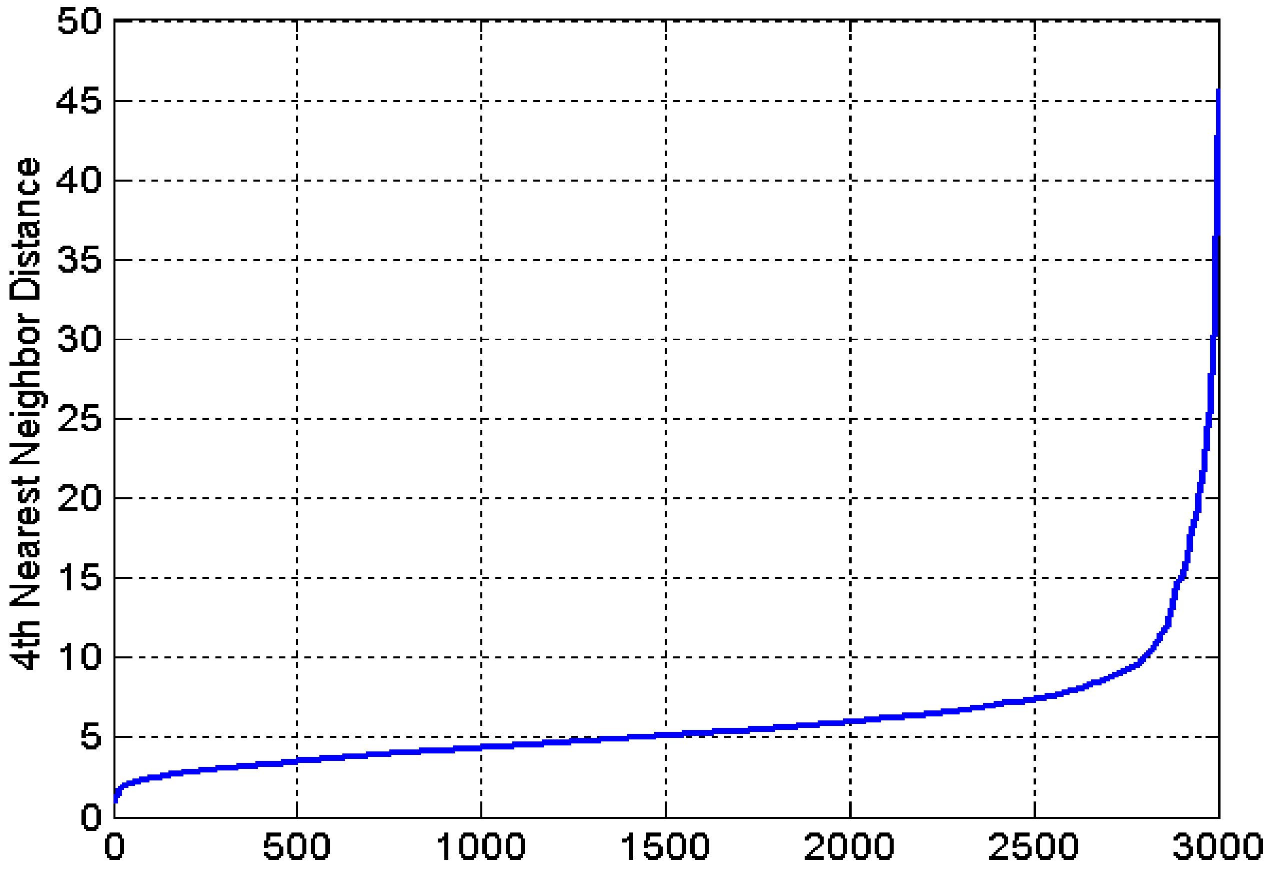
1. Calculate anomaly score for each data point in the dataset.
2. Use threshold t on the score to determine outliers.

Q: How to determine the threshold?

For determining the right threshold, same idea as in PCA (Principal Component Analysis) can be used: the elbow rule. Sort the values f(x) and plot them in order to see the ideal threshold. (Example on next slide)

Model-based Methods

Determining threshold for statistical approach

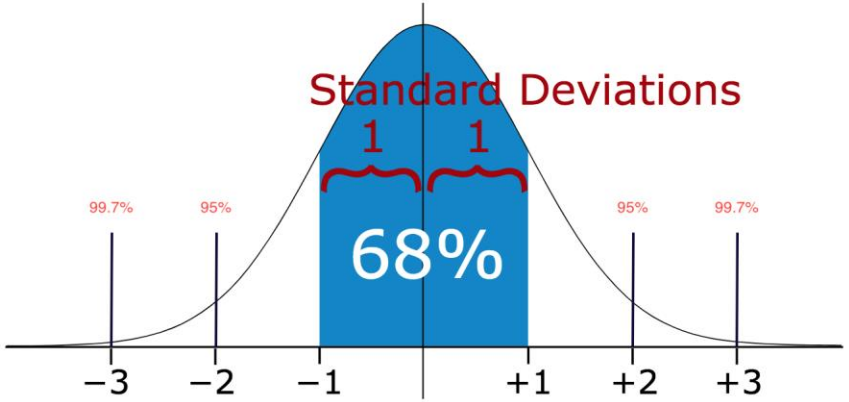


http://web.cs.wpi.edu/~ruiz/KDDRG/Resources/AnomalyDetection/Ruiz_Slides_Anomaly_Detection.pptx

Model-based Methods

Normal Distribution – Stardard Deviation and z-score

- If the distribution as approx. **normal**, then 68% of the data lie within 1 std (standardad devation), 95% within 2 stds and 99.7% within 3 stds.
- **z-score** is a metric giving you an idea how far from the mean a data point is in terms of stds
- $\mu = \frac{\sum x}{n}$
- $\sigma(std) = \sqrt{\frac{\sum (x-\mu)^2}{n-1}}$
- $z = \frac{x - \mu}{\sigma}$
- Multivariate: Mahalanobis distance



https://towardsdatascience.com/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623

Model-based Methods

Normal Distribution – Grubb's test

- Used to detect a single outlier in a univariate data that follows approx. normal distribution.
- Hypothesis: H_0 – no outliers, H_a – exactly one outlier in the dataset

$$G = \frac{\max |Y_i - Y|}{std}$$

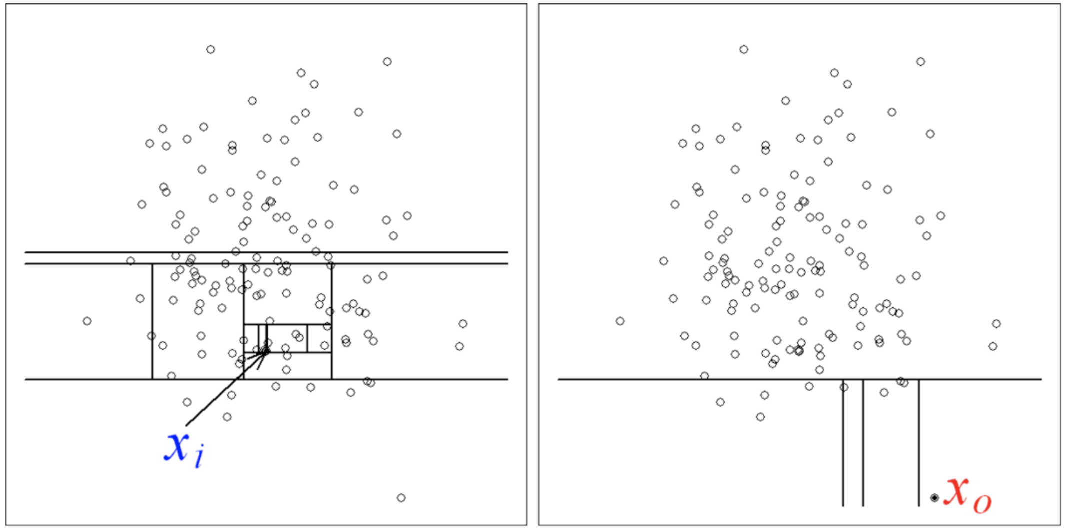
- Reject H_0 if:

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/N, N-2)}}{N-2+t^2_{(\alpha/N, N-2)}}}$$

https://towardsdatascience.com/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623

Isolation forest

- Build on Random trees
- Partitions are created by selecting a feature and then selecting a random split value.
- Outliers are lessfrequent than regular observation – that is why using random partitioniong outliers should lie closer to the root.
- Scalable to higher-dimensional spaces
- scikit-learn (Python) x solutide (R)



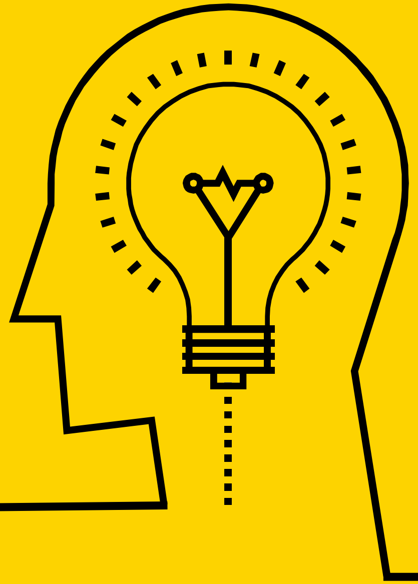
 |

Model-based Methods

Summary

- Pros/Cons:
  - Identifying the distribution in a dataset. (Multivariate case is hard)
  - Firm mathematical foundation.
  - Can be efficient
  - Good results if distribution is known.
  - Anomalies can distort the parameters of the fistribution.
- Mostly unsupervised methods are used
- Z-score, Probability scores, Grubb's test, Isolation forests

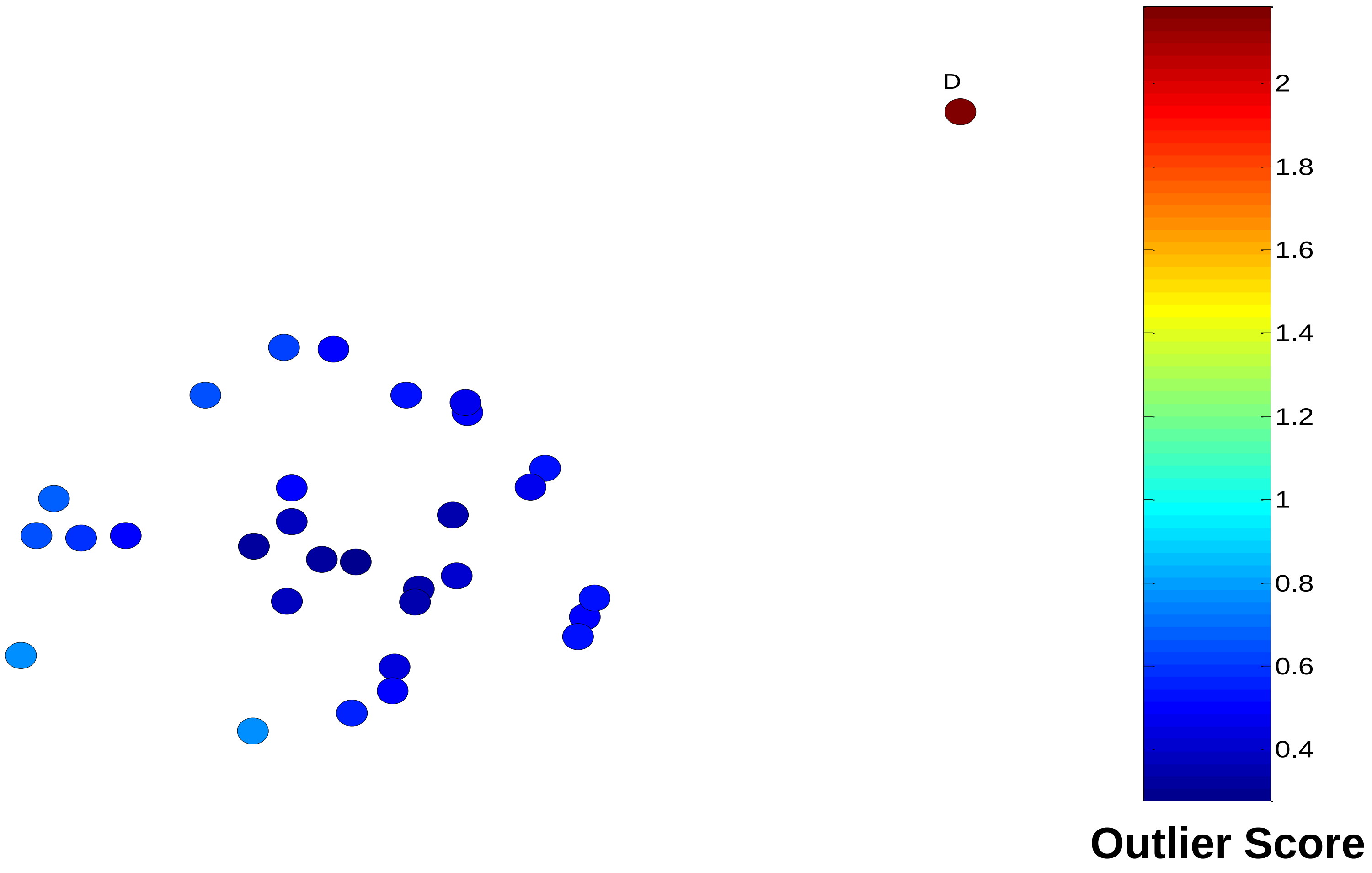
Proximity based methods



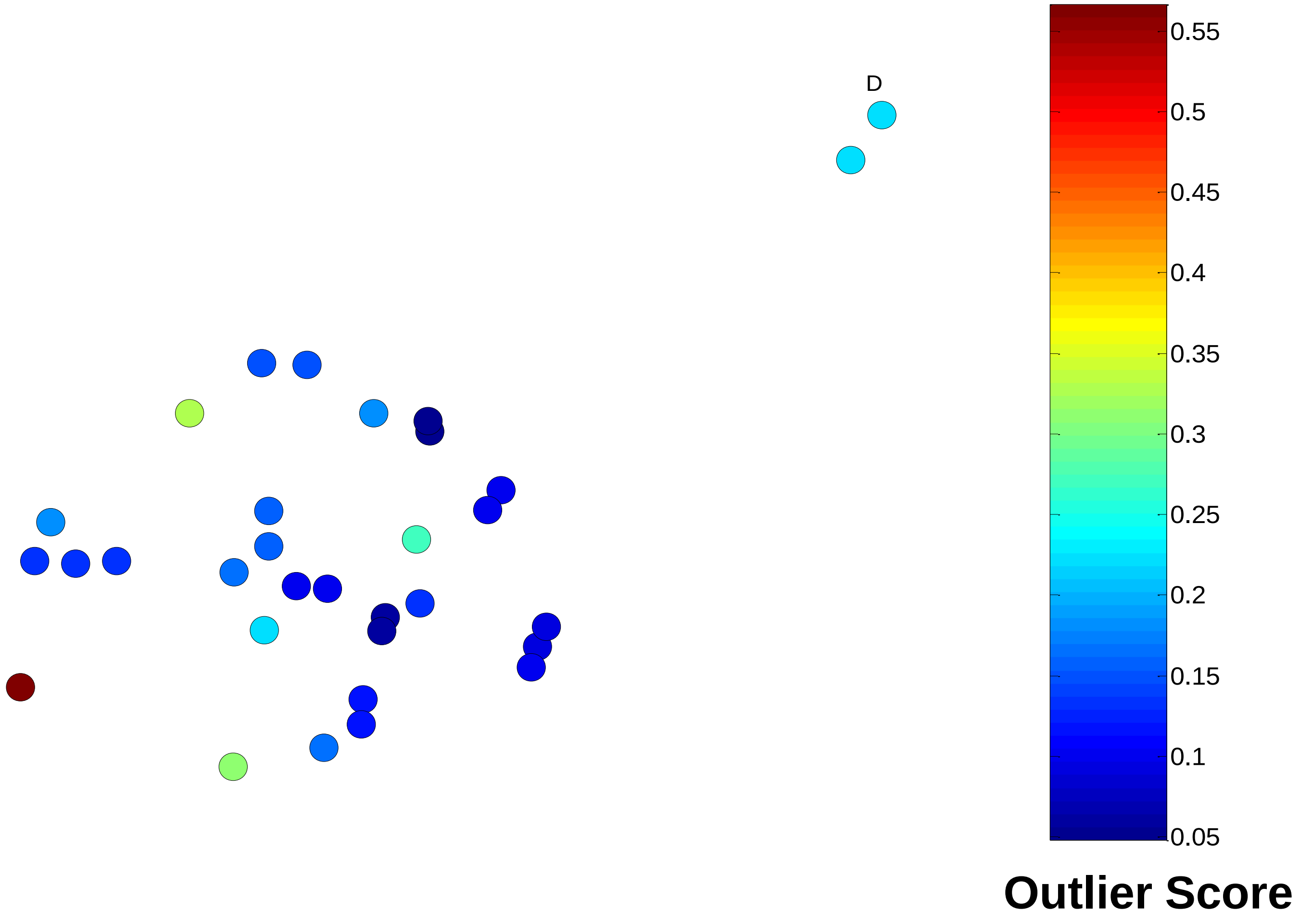
Proximity Approach

- Based on idea that object is an outlier if a specified fraction of the objects is more than a specified distance away.
- Distance needs to be defined beforehand (can be tricky with mixed-type data – remember clustering)
- The outlier score of an object is the distance to its **k-th nearest neighbour**.

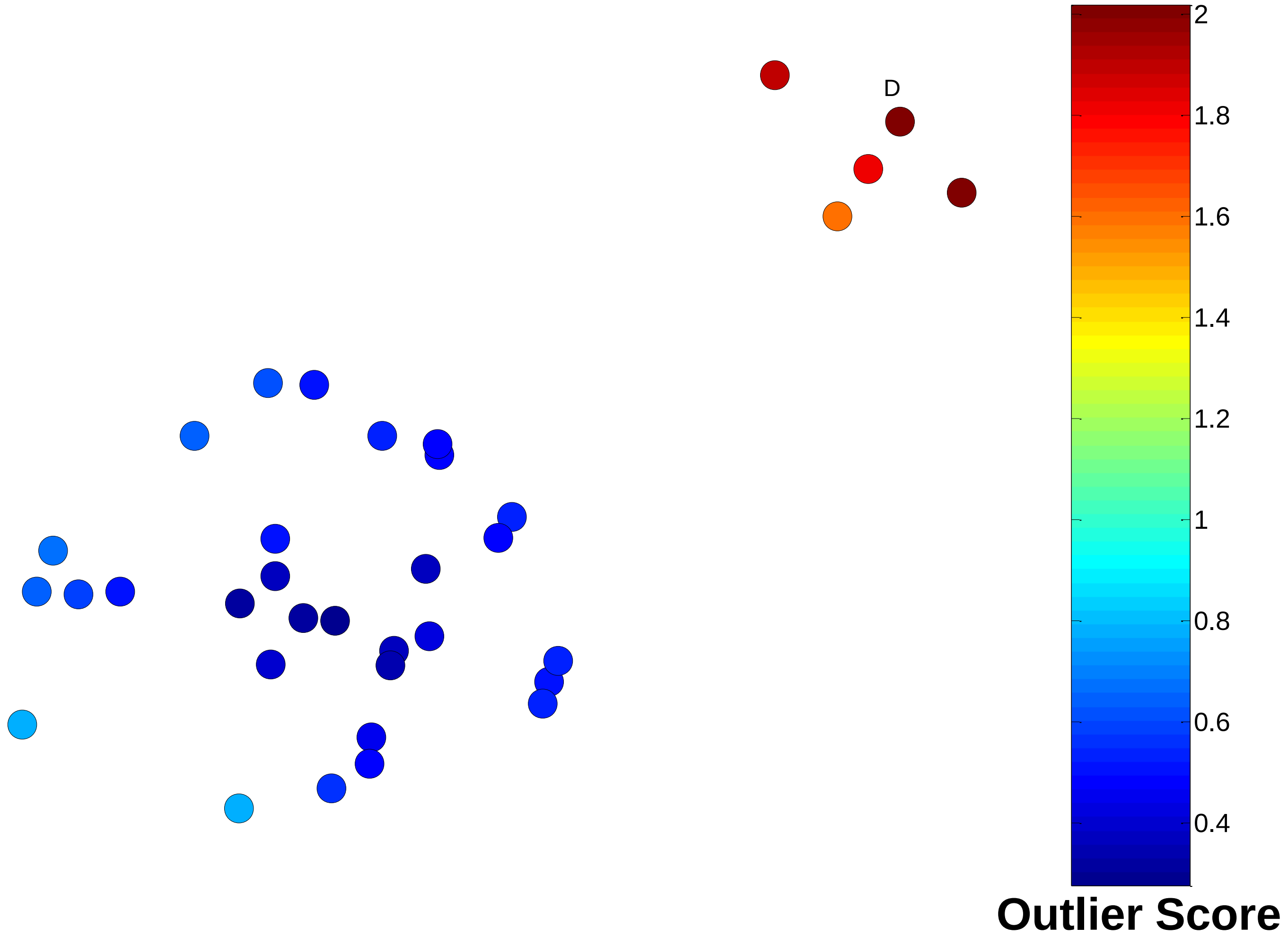
1-Nearest Neighbour



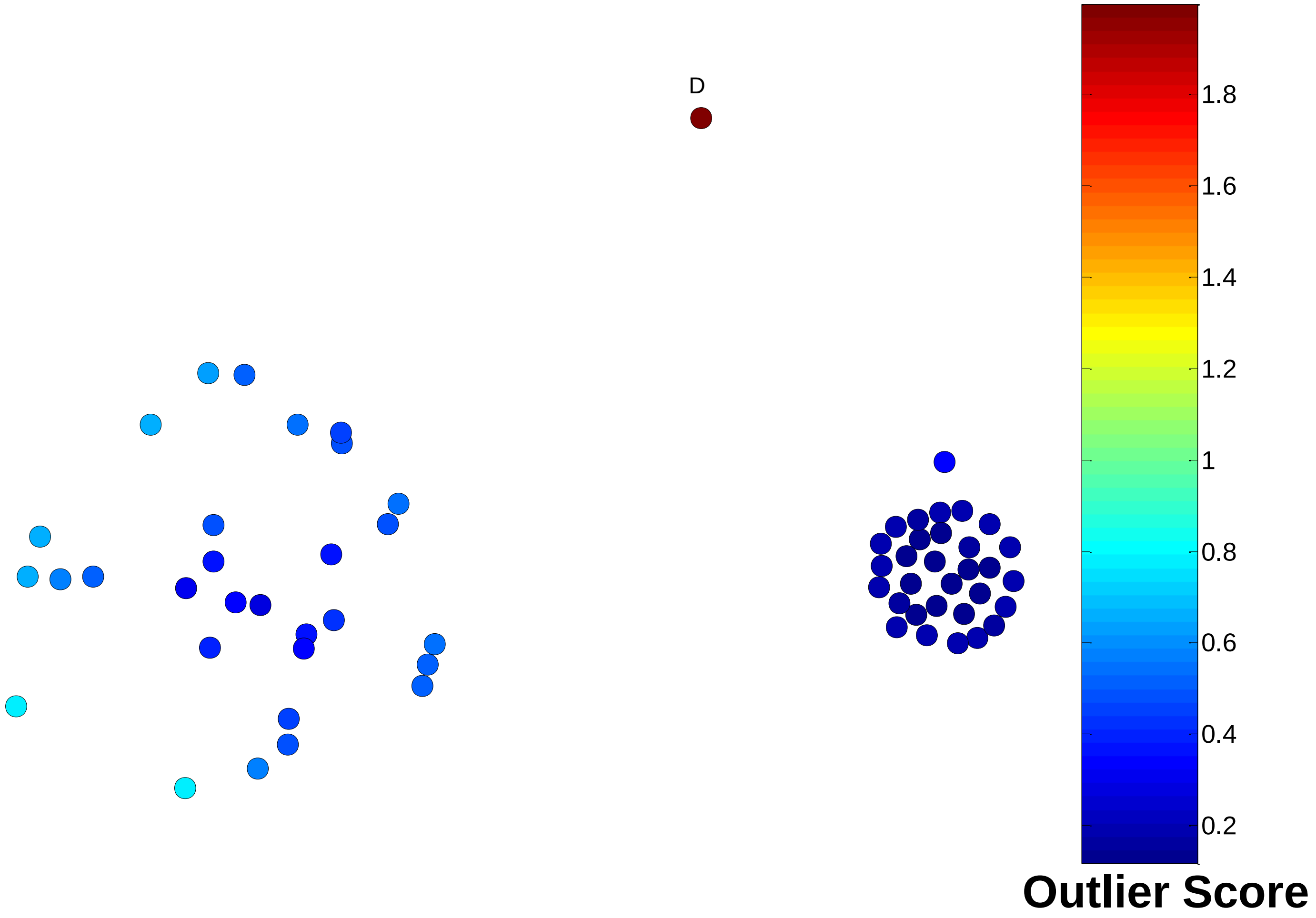
1-Nearest Neighbour



5-Nearest Neighbour



5-Nearest Neighbour

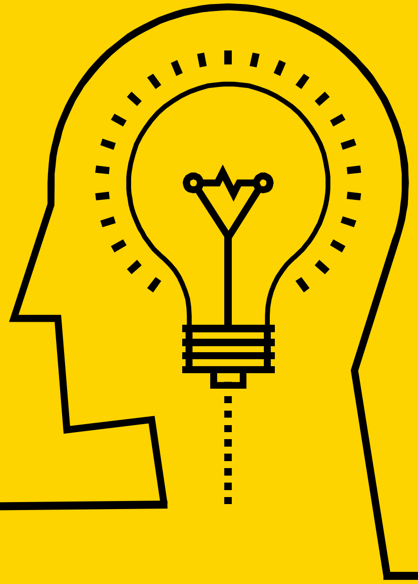


Proximity-based Methods

Summary

- Simple
- Distance metrics needed (not always straight-forward and tricky in high-dimensional space)
- Sensitive to parameters (outliers x dislocated cluster)
- Sensitive to variations in density
- Computationally demanding - $O(n^2)$

Density based methods



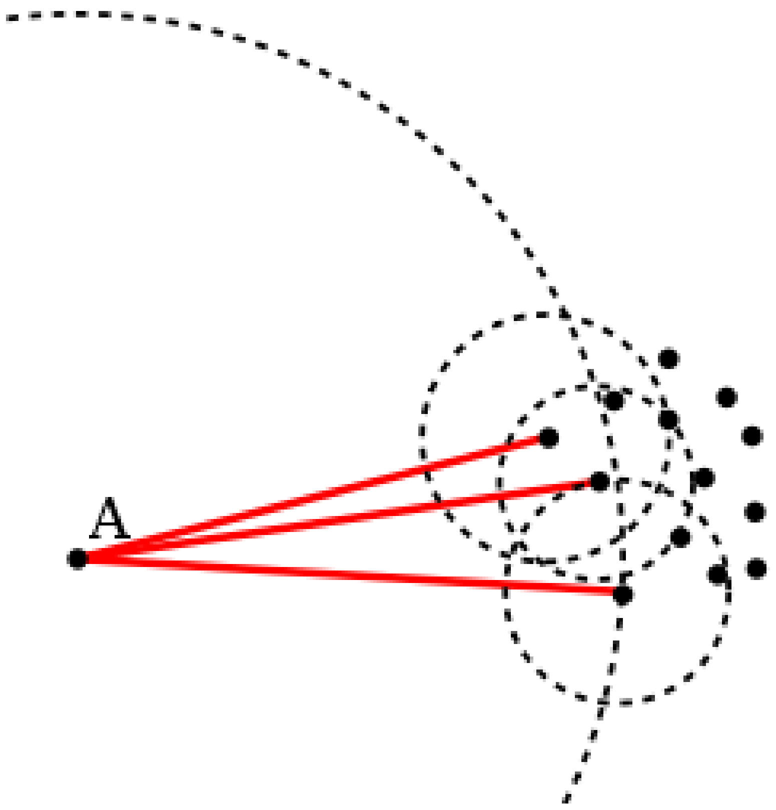
Density-based Approach

- The outlier score of an object is the inverse of density around the object – objects with low density are treated as outliers.
- Inverse of distance to k-nearest neighbour
- Inverse of average distance to k neighbours
- Local outlier factor (**LOF algorithm**)

Density-based Approach

LOF

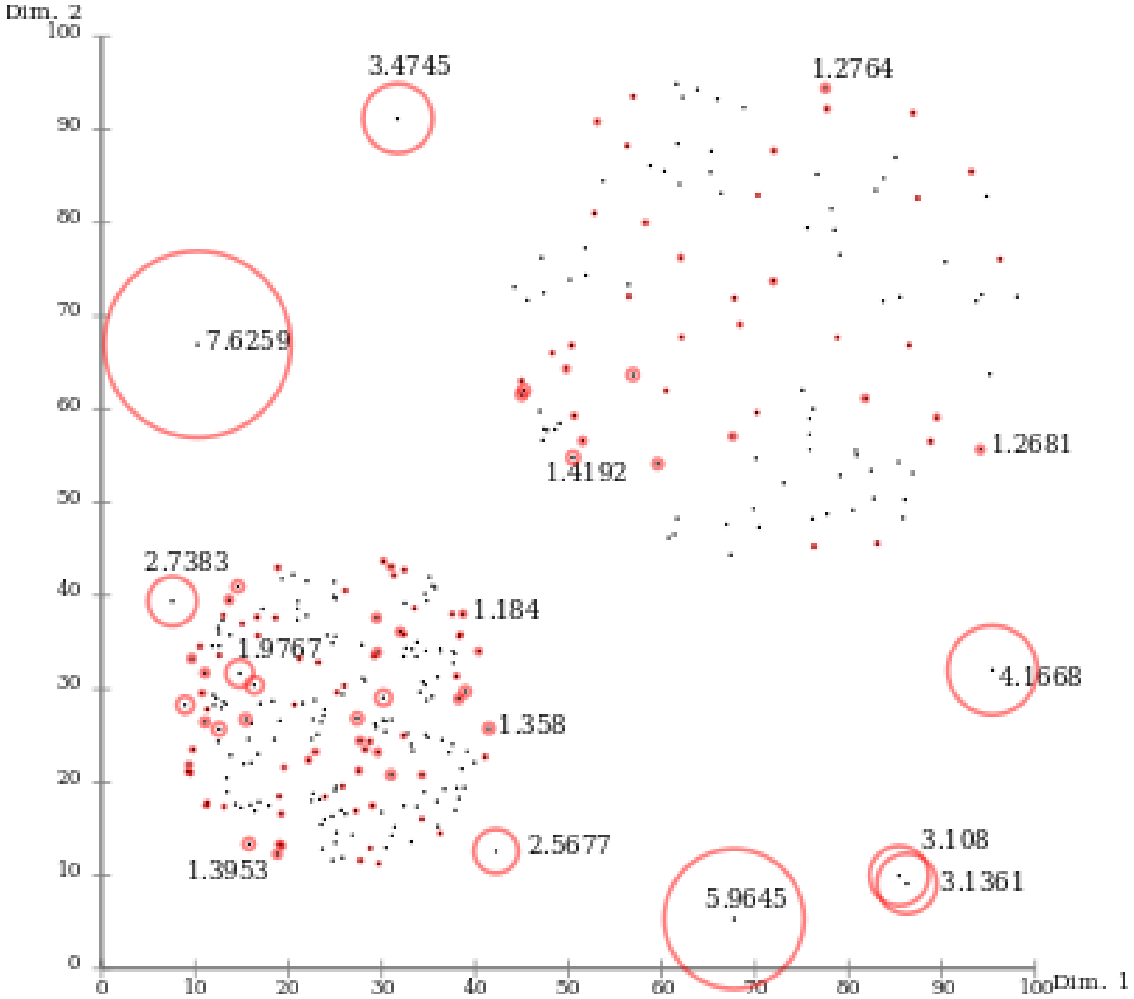
- Measures local deviation of a given data point w.r.t. its neighbours.
- Compare local density of a point with the densities of its neighbours. (A has much lower density than its neighbours)



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

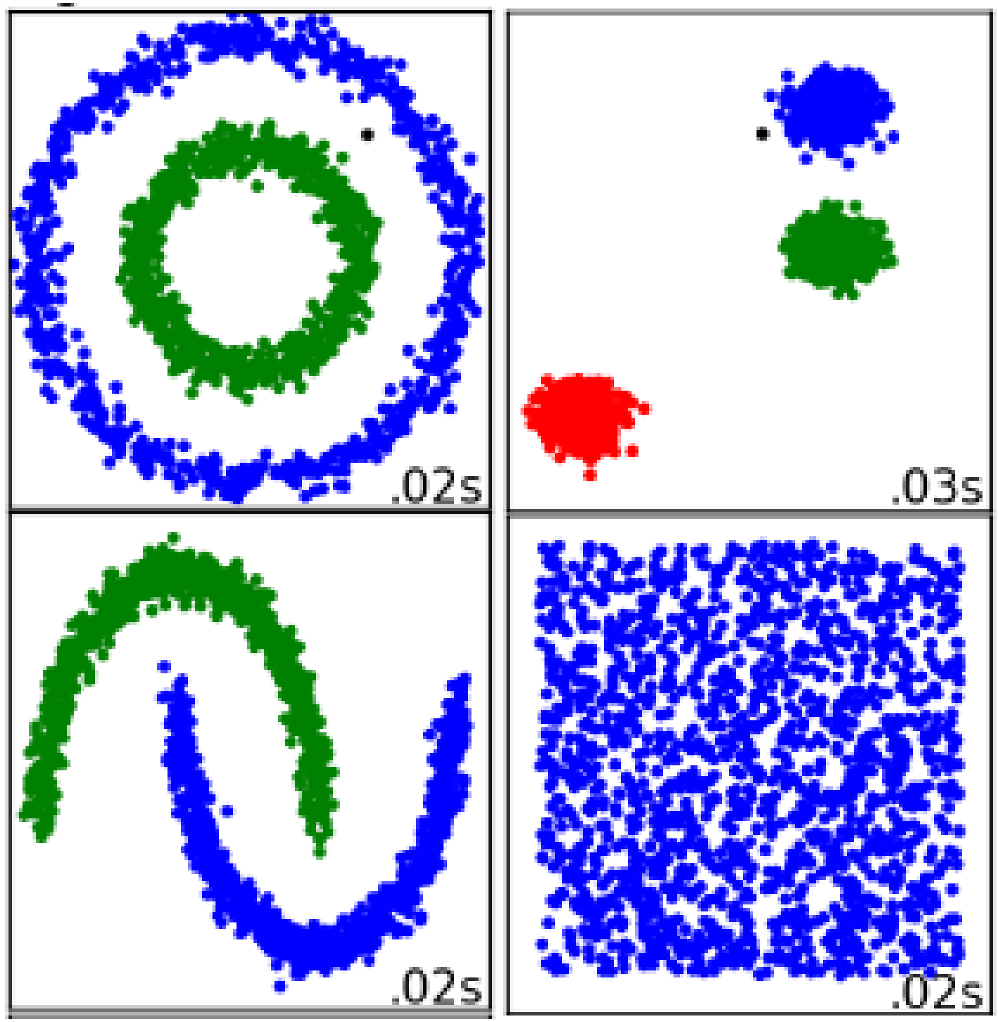
Density-based Approach

LOF



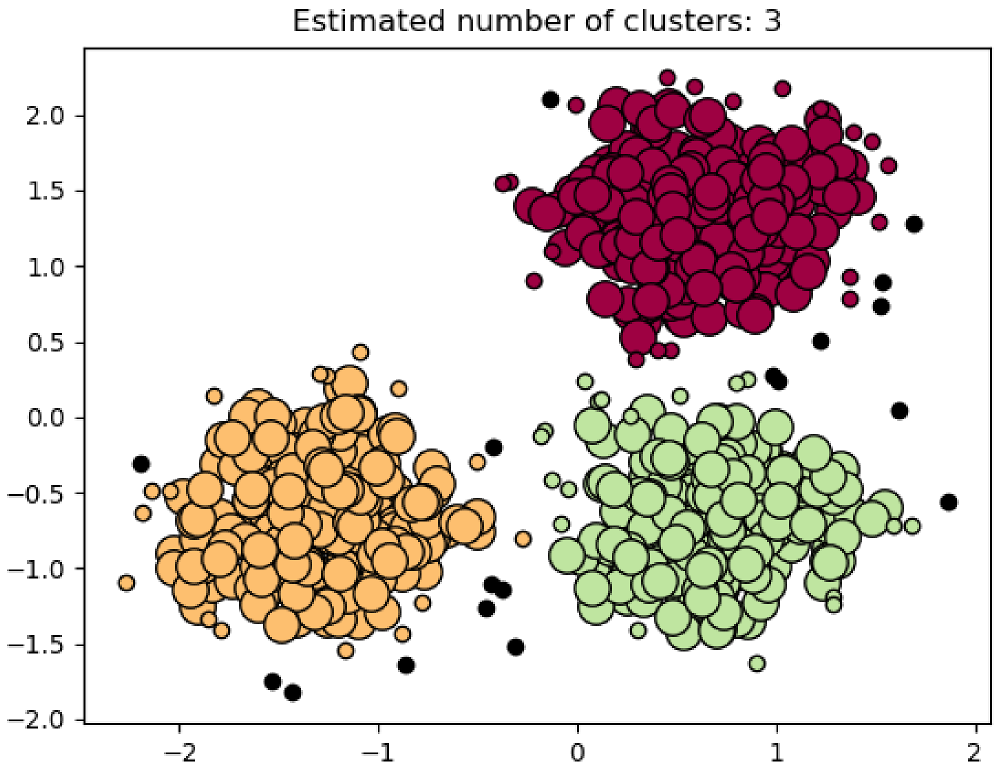
DBSCAN

- The DBSCAN algorithm views clusters as areas of high density separated by areas of low density.
- Due to this rather generic view, clusters found by DBSCAN can be any shape, as opposed to k-means which assumes that clusters are convex shaped.
- Automatically determines best number of clusters.
- It is robust to outliers.



DBSCAN

- Needs parameter tuning:
  - eps – maximum distance between two samples for one to be considered the neighbourhood of the other.
  - min_samples – number of samples in a neighbourhood for a point to be considered core point
- Proper distance neccessary
- Outputs clustered points + outliers as a special class



https://scikit-learn.org/stable/auto_examples/cluster/plot_dbscan.html

Summary

- There are different options for detecting outliers each having own pros and cons.
- There is a big difference between identifying outliers in one variable and in whole dataset – multivariate outlier detection carries issues like distance metrics selection, inability to use statistical approaches and inability to visualize the results.
- Methods for tackling outliers are:
  - Graphical
  - Model-based
  - Distance-based
  - Density-based
- Usually, outliers detection is an unsupervised task.

References

- Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. Introduction to Data Mining (2nd Edition) (2nd. ed.). Pearson. https://www-users.cs.umn.edu/~kumar001/dmbook/index.php#item4

- Worcester Polytechnic Institute - Prof. Carolina Ruiz - lecture on Anomaly detection http://web.cs.wpi.edu/~ruiz/KDDRG/Resources/AnomalyDetection/Ruiz_Slides_Anomaly_Detection.pptx

- https://towardsdatascience.com/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623
- https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e

Home Assignment

Home assignment HA2

Develop segmentation of Airbnb hosts using predictors and algorithm of your choice in Python. Name your segments and provide "business" interpretation of each segment. Pick a good example listing who represent their respective segment.

Alternatively you can develop a segmentation on your real business use-case given that you can share the data with the tutors.

Save your notebook and send it to *pmilicka@deloitteCE.com* with subject **DSI_HA02_surname** for review.

- The notebook needs to be easily runnable (dataset preparation is inside the jupyter notebook on top of the base airbnb tables.

