9th Lecture: Unsupervised learning

Data Science Foundations Course

27.6.2023

Today’s structure

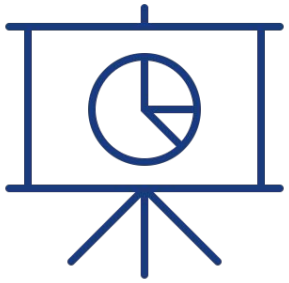
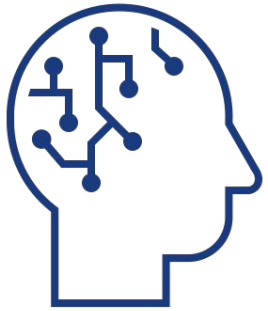
1. Intro to Unsupervised Learning
2. Segmentation (using k-means)
3. Principal Component Analysis (PCA)

Andrea

Michal

Unsupervised Learning Introduction

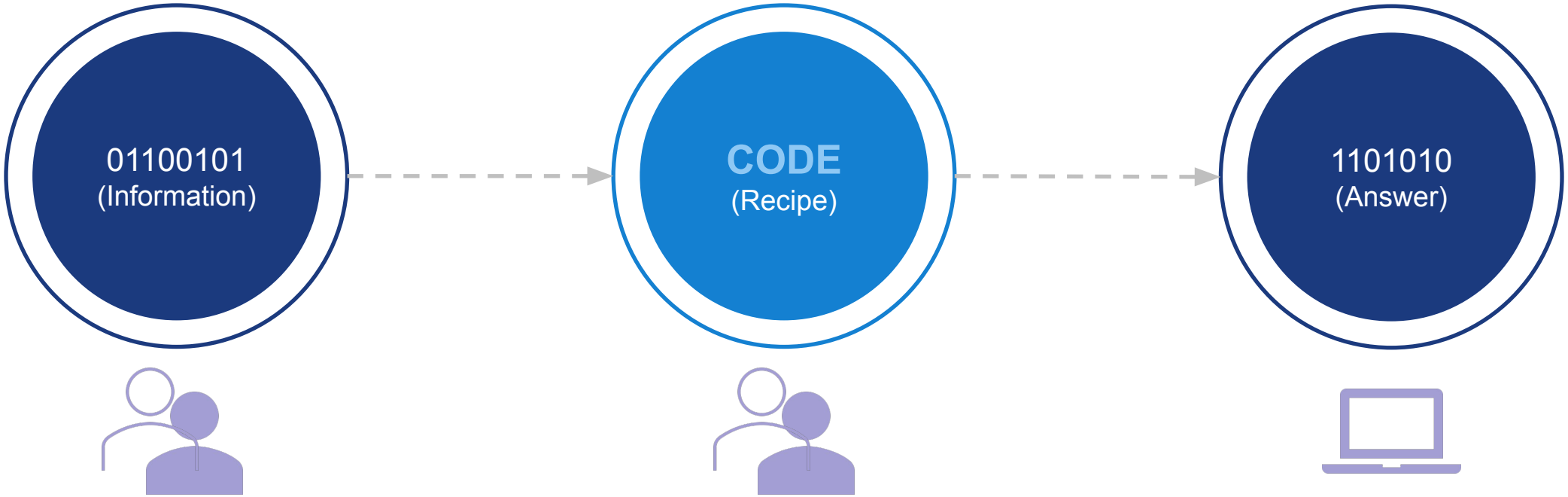
Data Science

None	Many	Few
		
Descriptive Analytics	Machine Learning	Statistical Inference
Get inspired	Make a recipe	Decide wisely

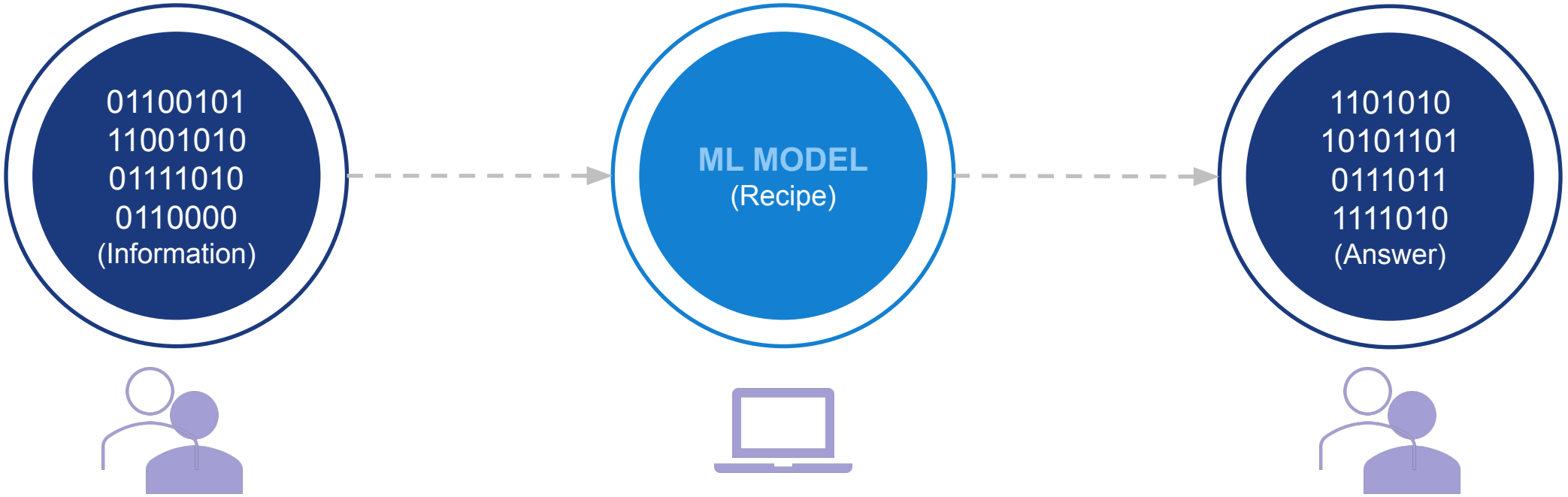
czechitas

Machine learning vs. traditional programming

Machine learning vs. traditional programming



Machine learning vs. traditional programming



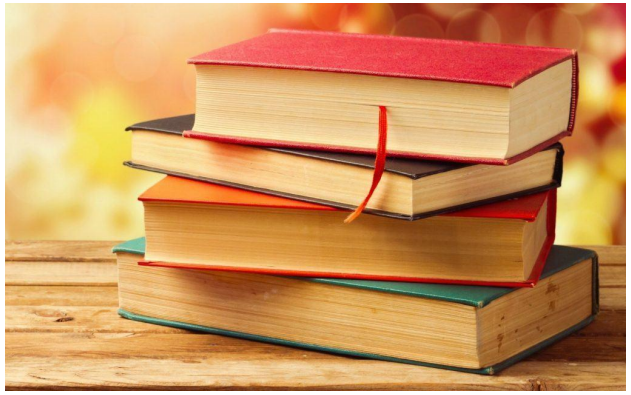
Supervised Learning

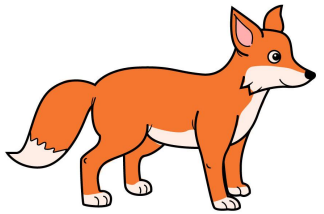
Instance	Label	Feature 1	Feature 2	Feature 3	…
1	Fox	54	123	140	
2	Not fox	78	89	12	

Fox

Not fox

Not fox





Fox

iHeartCraftyThings.com

czechitas

Supervised Learning

Instance	Label	Feature 1	Feature 2	Feature 3	…
1	Fox	54	123	140	
2	Not fox	78	89	12	

Not fox

czechitas

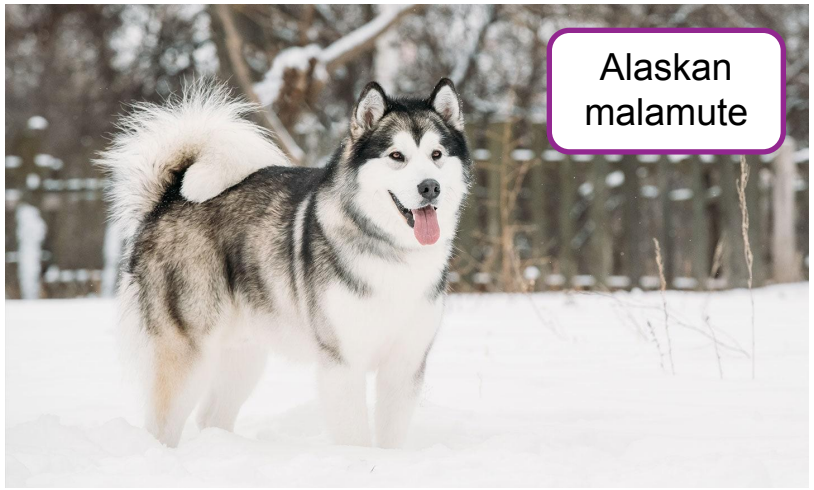
Unsupervised Learning

label 1
label 2
…

1

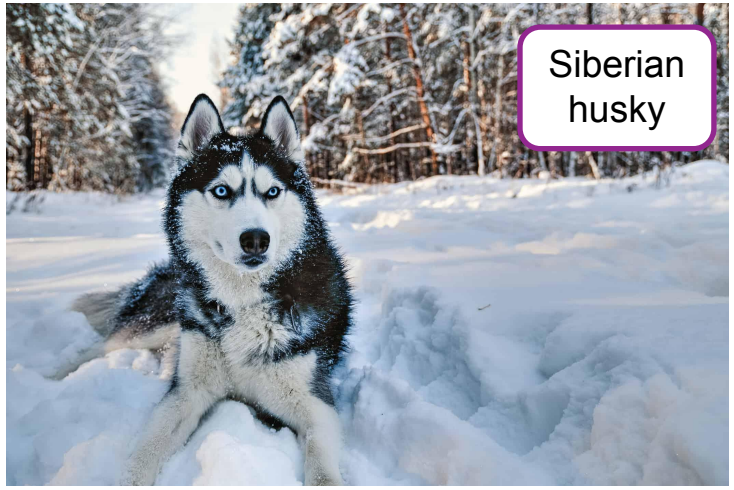
2

3



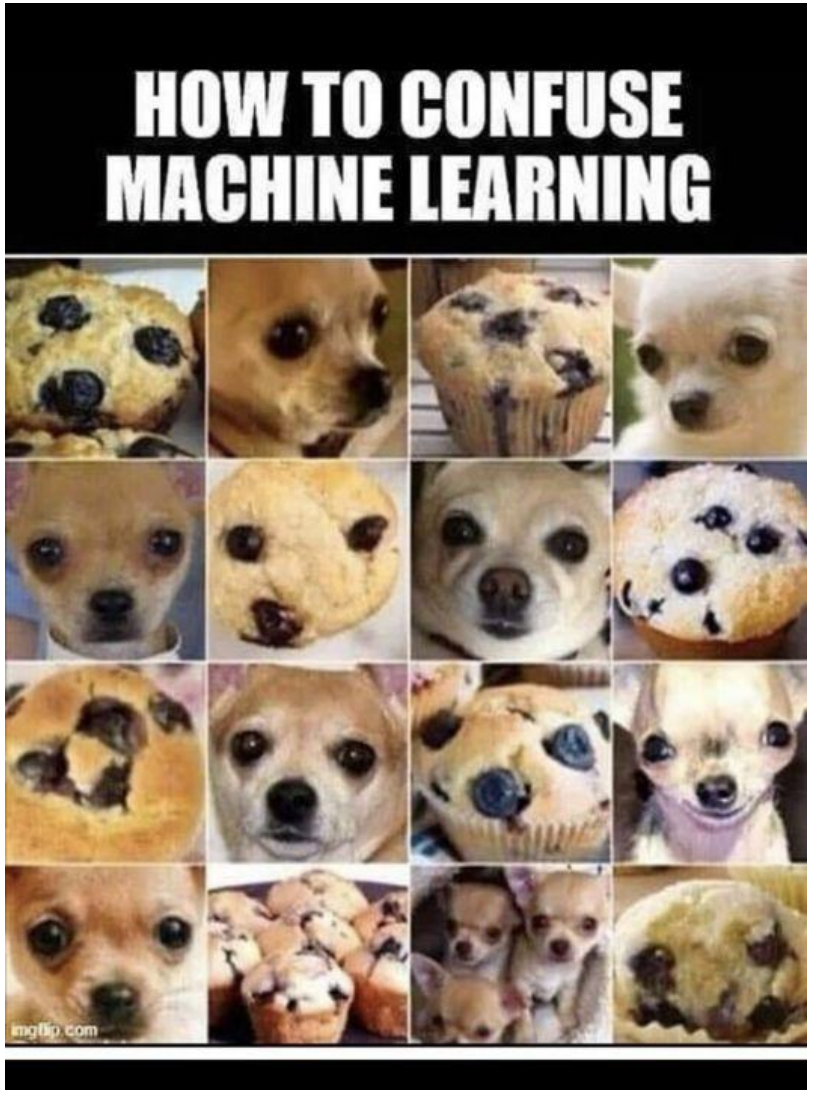
4

5



czechitas

Choice of dataset is important



czechitas

Supervised Learning

For any instance you give the system, you have the correct label on hand

CustomerID	Income	Years Education	Age	Default
2343	50 000	17	35	No
1213	35 000	15	32	Yes
4533	40 000	15	53	No
4563	100 000	19	51	No
7554	50 000	18	28	No
6465	27 500	13	25	Yes
7453	34 000	13	32	No
6775	72 000	18	43	No
4643	50 000	19	47	No
6886	48 000	19	37	?
8668	62 500	21	39	?
8765	78 000	23	46	?
9797	23 000	12	29	?

Labeled Data

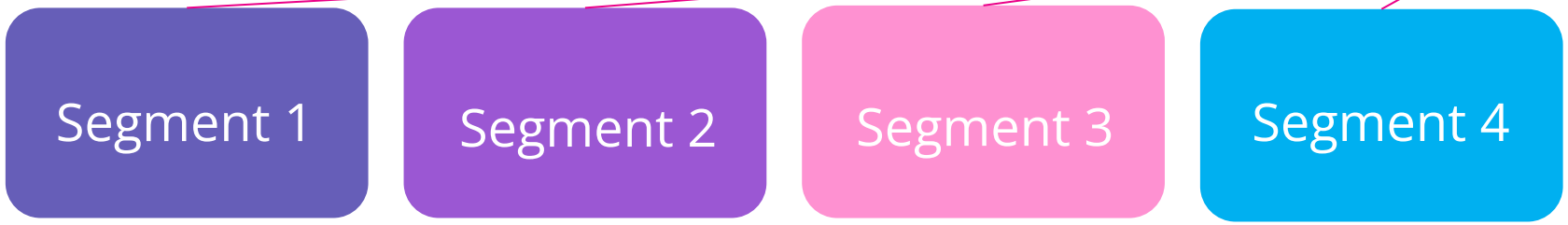
Unlabeled Data

Unsupervised Learning

You don't have any labels, but you search for patterns

CustID	Income	Education	Age	Gender	Last Purchase Amount	Customer Segment
2343	50 000	17	35	M	2500	?
1213	35 000	15	32	F	34000	?
4533	40 000	15	53	F	12000	?
4563	100 000	19	51	M	2100	?
7554	50 000	18	28	M	760	?
6465	27 500	13	25	F	21000	?
7453	34 000	13	32	M	42000	?

Unlabeled Data



Applications of Unsupervised Learning

- Clustering (e.g. of customers, documents, images)
- Understanding underlying patterns in data
- Dimensionality reduction
  - Feature selection
  - Feature extraction
- Anomaly or outlier detection

Validation

Validation is not easy!
No ground truth
to compare against!

- External validation
  - Involves SME knowledge
  - Similar to supervised learning validation
- Internal validation
  - Cohesion within each cluster
  - Separation between different clusters
  - □ can be combined in one number (e.g. Silhouette coefficient, Calisnki-Harabasz coefficient)
- Train/test or twin-sample validation
- Use output of unsupervised model as input to supervised model

czechitas

Summary

- First, define your problem, only then check if you can apply ML, or more specifically, unsupervised modelling
- Learning on unlabeled raw data, i.e., there is no ground truth
- Used for clustering and dimensionality reduction
- You cannot rely on human judgement just as much as you cannot rely on ML algorithms
- Validate model results

Unsupervised Learning Methods

- k-means Clustering
- Principal Component Analysis (PCA)
- Hierarchical Clustering

k-means clustering (segmentation)

What is segmentation?

Process of finding similar segments from entities with many characteristics

- What could be entities that are segmented?
- What could be the “many” characteristics?
- How could the segments be similar?

Why it is important?

Each segment can be approached by different strategy or learnings from one segment can be transferred to the other ones

- Identify how many different entities segments have
- Understand how segments differ and where are similar
- Define different marketing strategy for each segment

How are segments found?

Machine learning methods are used especially when number of characteristics is quite large

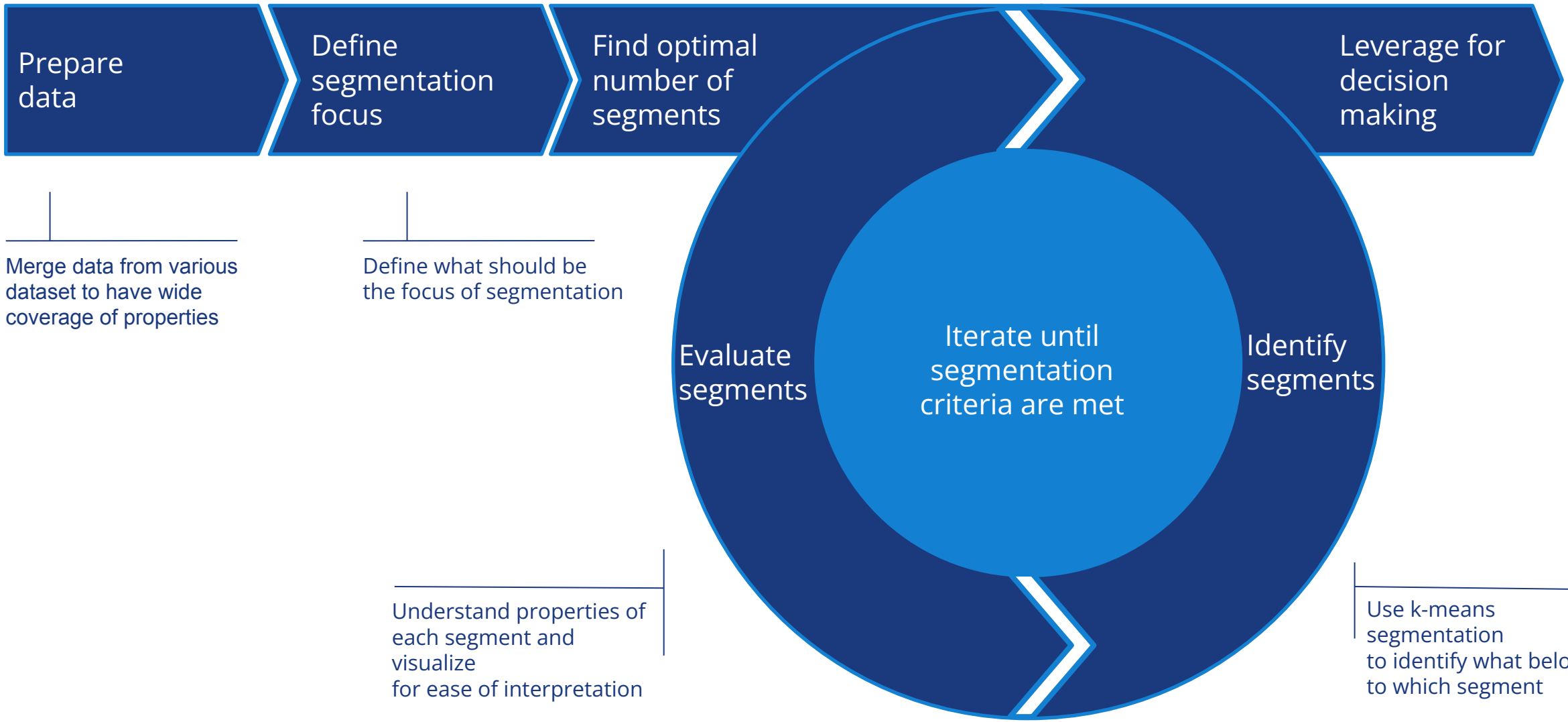
- Simplest approach is to use business rules. How?
- With many characteristics, it is not clear how should be segments defined. Why?
- ML methods are very popular, in particular k-means

There are TWO key challenges in K-MEANS approach to segmentation

With many characteristics in your data, how to ensure segmentation results are tailored to business needs?

k-means approach requires data scientist to specify number of segments manually. How to choose number of segments algorithm searches for?

Segmentation approach



Merge data from various dataset to have wide coverage of properties

Define what should be the focus of segmentation

Leverage Machine Learning to identify number of distinct segments

Define segment specific strategies
Learn from the successful segments

Understand properties of each segment and visualize
for ease of interpretation

Use k-means segmentation
to identify what belongs to which segment

Segmentation approach

Leverage Machine Learning to identify number of distinct segments

Define segment specific strategies
Learn from the successful segments

Prepare data

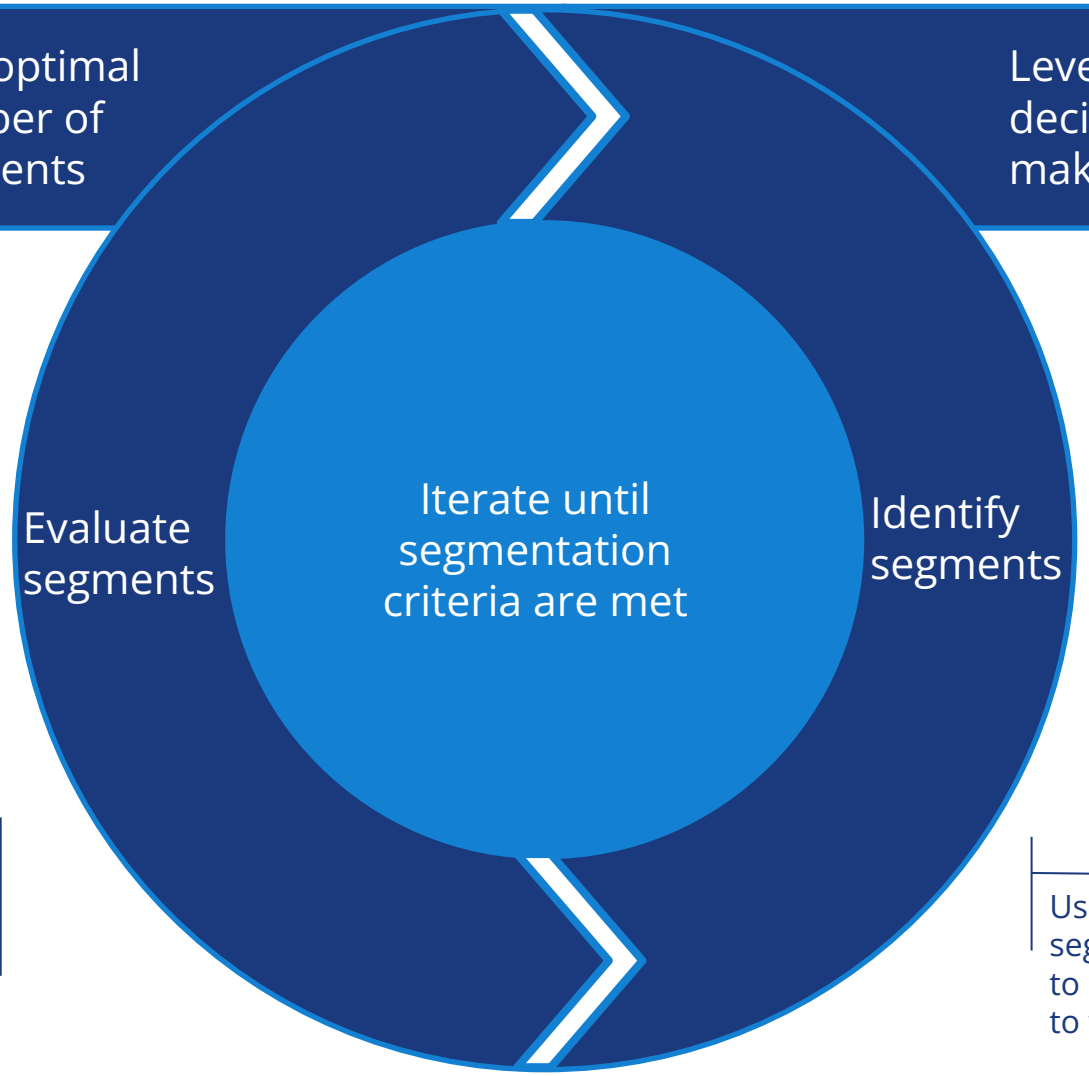
Define segmentation focus

Find optimal number of segments

Leverage for decision making

Merge data from various dataset to have wide coverage of properties

Define what should be the focus of segmentation



Understand properties of each segment and visualize
for ease of interpretation

Use k-means segmentation
to identify what belongs to which segment

k-means requirements for data prep

- Only numerical variables allowed
- No missing variables allowed
- k-means is sensitive to outliers (consider removing them)
- Scale all the variables to the same measure (standardize)
- Remove variables with near 0 variance (not bringing any value)

Segmentation approach

Leverage Machine Learning to identify number of distinct segments

Define segment specific strategies
Learn from the successful segments

Prepare data

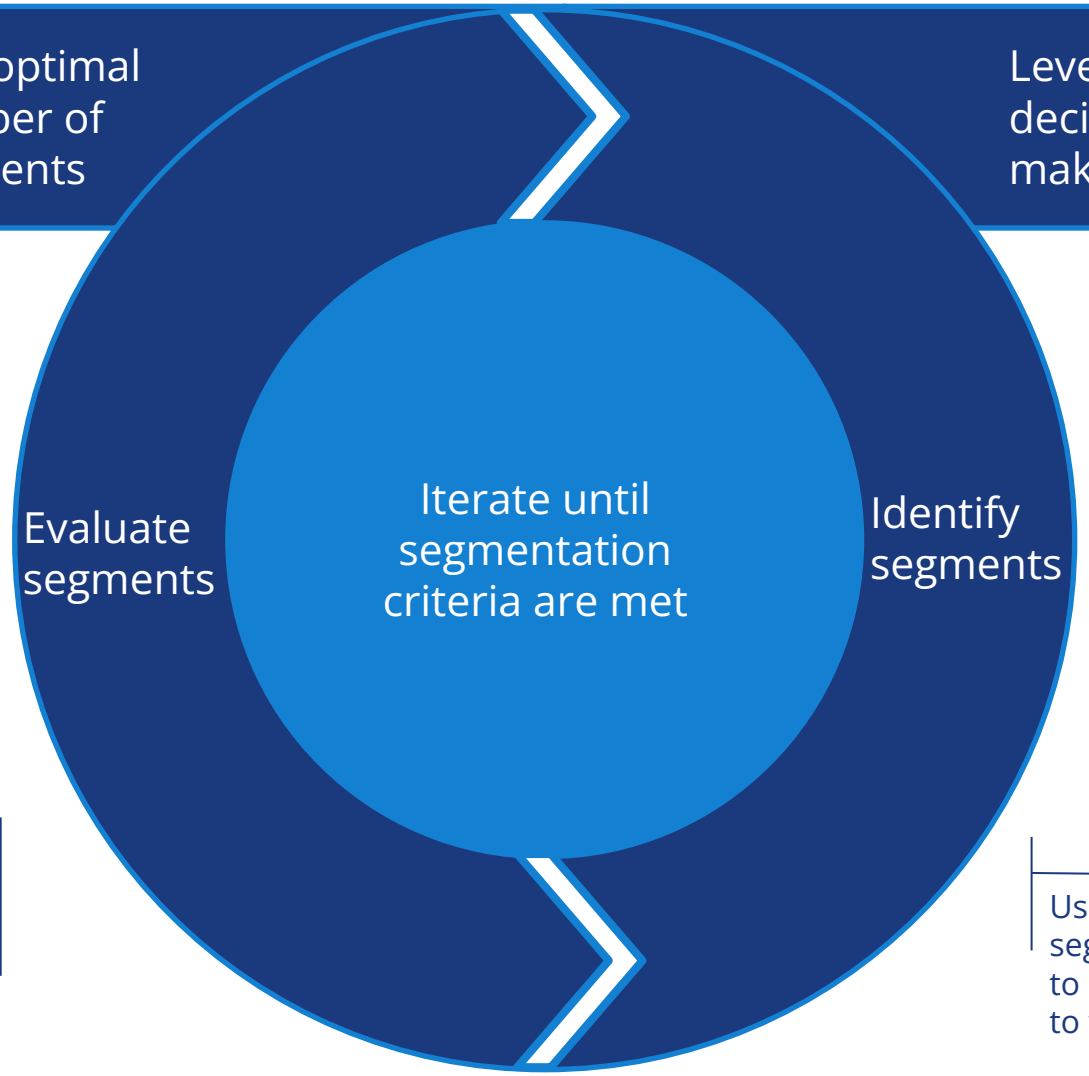
Define segmentation focus

Find optimal number of segments

Leverage for decision making

Merge data from various dataset to have wide coverage of properties

Define what should be the focus of segmentation



Understand properties of each segment and visualize
for ease of interpretation

Use k-means segmentation
to identify what belongs to which segment

Define Segmentation Focus

Case Study

You are doing segmentation for Marketing Director of Albert (Ahold), who is interested to send personalized newsletters to different segments based on their shopping behavior.

How would you define focus of the segmentation? From where would you source your data?

Segmentation approach

Leverage Machine Learning to identify number of distinct segments

Define segment specific strategies
Learn from the successful segments

Prepare data

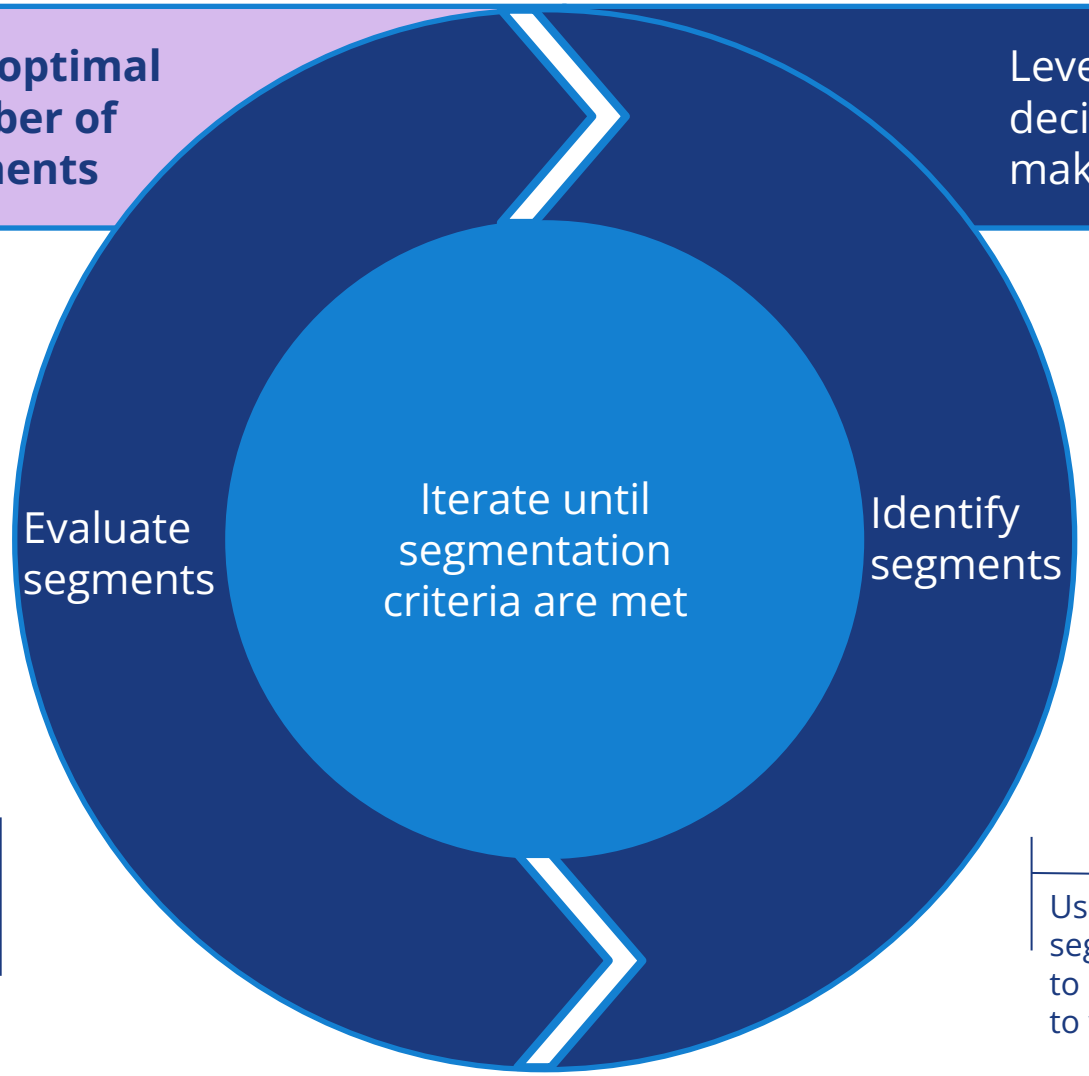
Define segmentation focus

Find optimal number of segments

Leverage for decision making

Merge data from various dataset to have wide coverage of properties

Define what should be the focus of segmentation

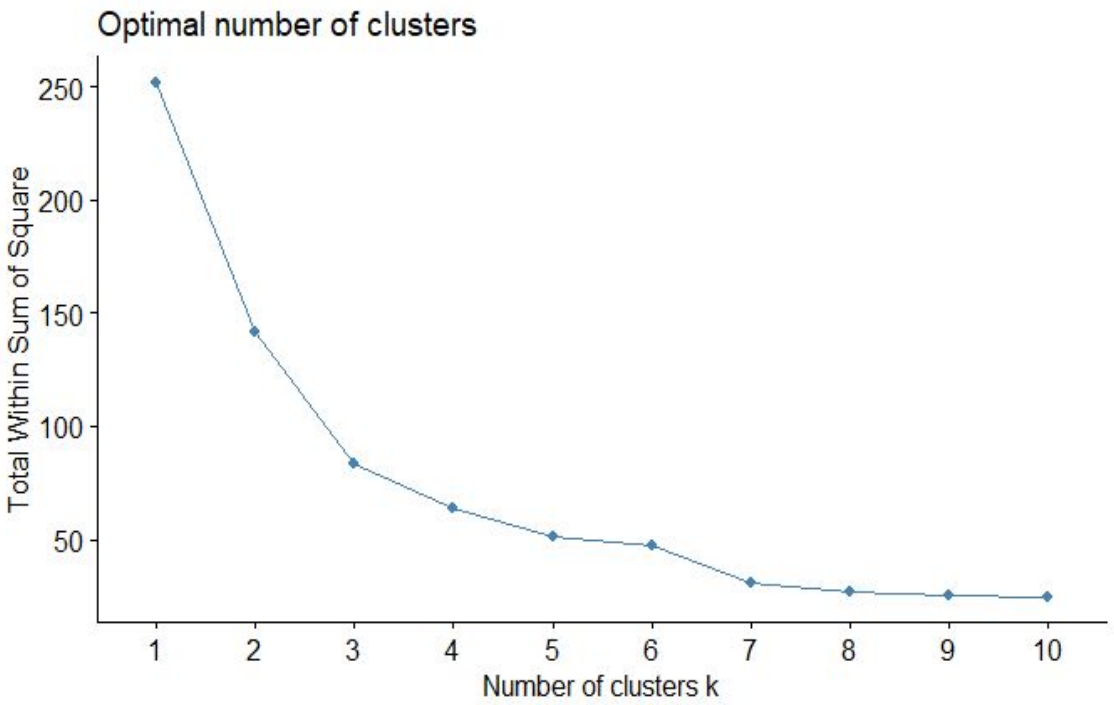


Understand properties of each segment and visualize
for ease of interpretation

Use k-means segmentation
to identify what belongs to which segment

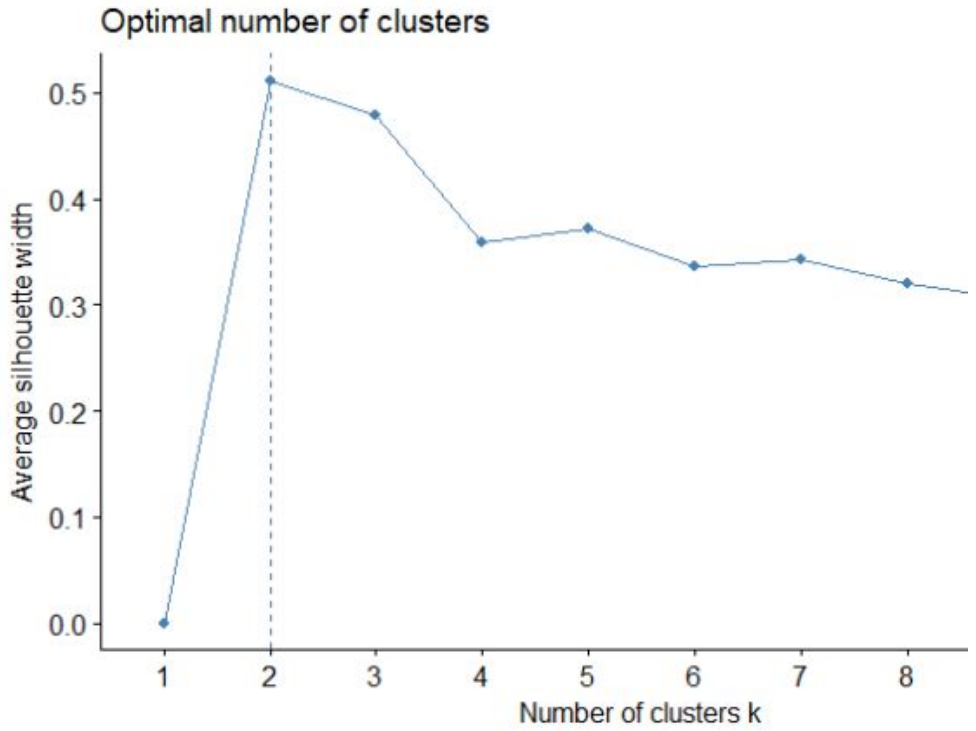
TWO alternatives to find optimal number of segments

Elbow chart



We plot the **percentage of variance explained by the clusters** against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the “elbow criterion”. This “elbow” cannot always be unambiguously identified.

Silhouette Score



Silhouette approach measures the **quality of a clustering**. A high average silhouette width indicates a good clustering. The optimal number of clusters k is the one that maximizes the average silhouette over a range of possible values for number of clusters k.

It is quite easy to draw Elbow or Silhouette charts

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer, SilhouetteVisualizer

# Assuming you have your data stored in a DataFrame called data_all_scaled

# Silhouette chart
silhouette_visualizer = SilhouetteVisualizer(KMeans(), colors='yellowbrick')
silhouette_visualizer.fit(data_all_scaled)
silhouette_visualizer.show()

# Elbow chart
elbow_visualizer = KElbowVisualizer(KMeans(), k=(1, 10), metric='wcss')
elbow_visualizer.fit(data_all_scaled)
elbow_visualizer.show()
```

Segmentation approach

Leverage Machine Learning to identify number of distinct segments

Define segment specific strategies
Learn from the successful segments

Prepare data

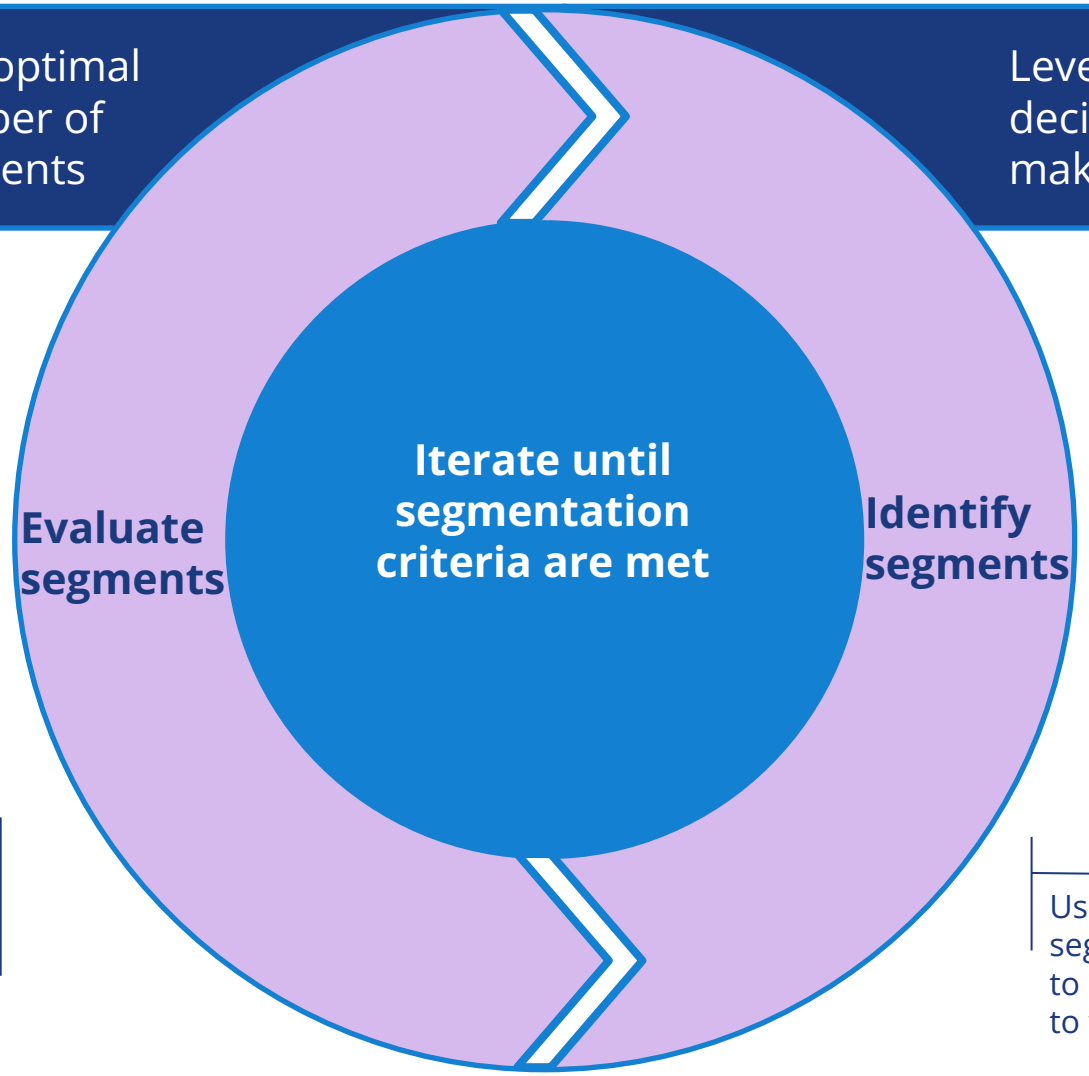
Define segmentation focus

Find optimal number of segments

Leverage for decision making

Merge data from various dataset to have wide coverage of properties

Define what should be the focus of segmentation



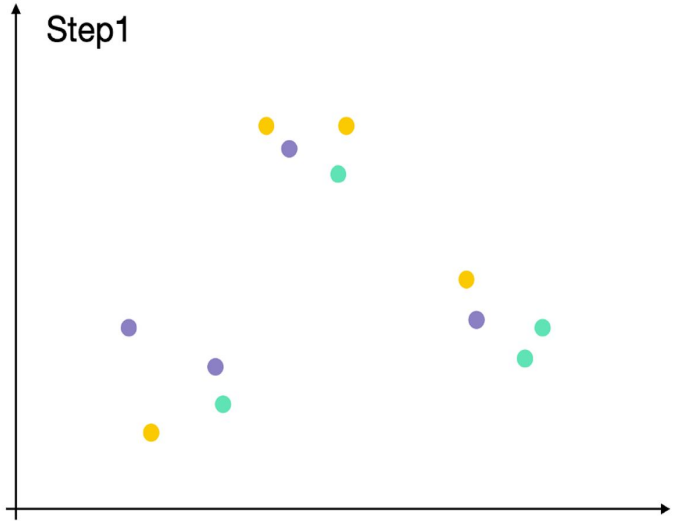
Understand properties of each segment and visualize
for ease of interpretation

Use k-means segmentation
to identify what belongs to which segment

Procedure

EXAMPLE: Clusters = 3

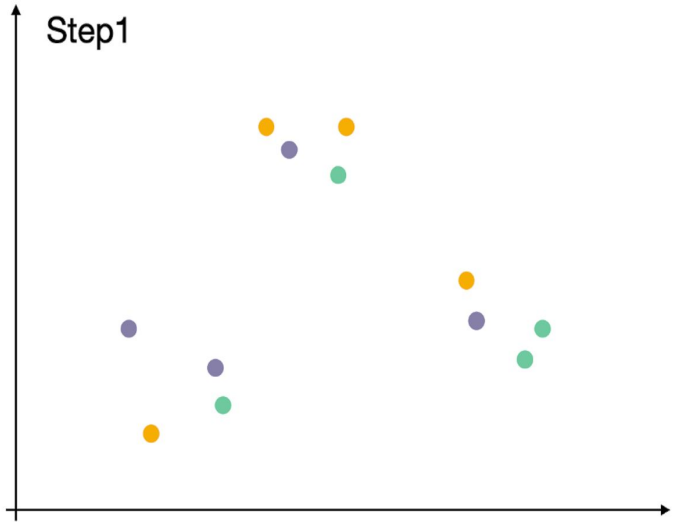
Step 1:
Assign a class (1,2,3) to each observation at random.

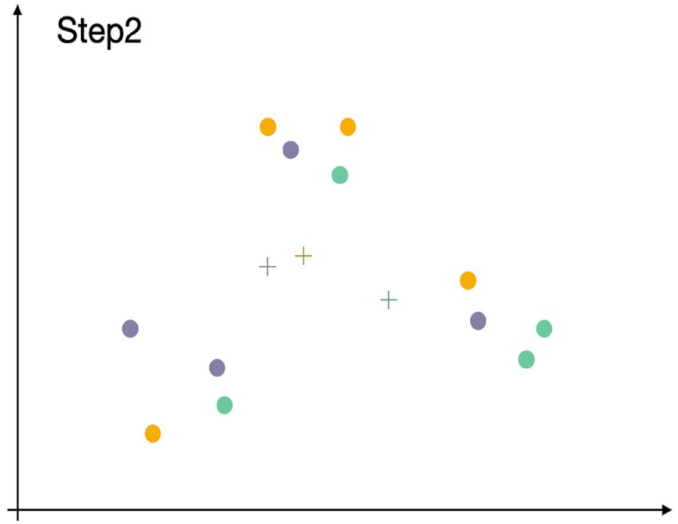


Procedure

EXAMPLE: Clusters = 3

Step 1:
Assign a class (1,2,3) to each observation at random.



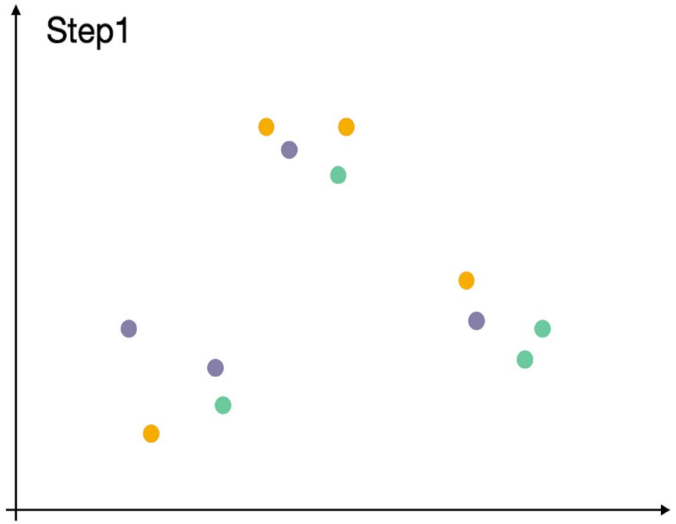


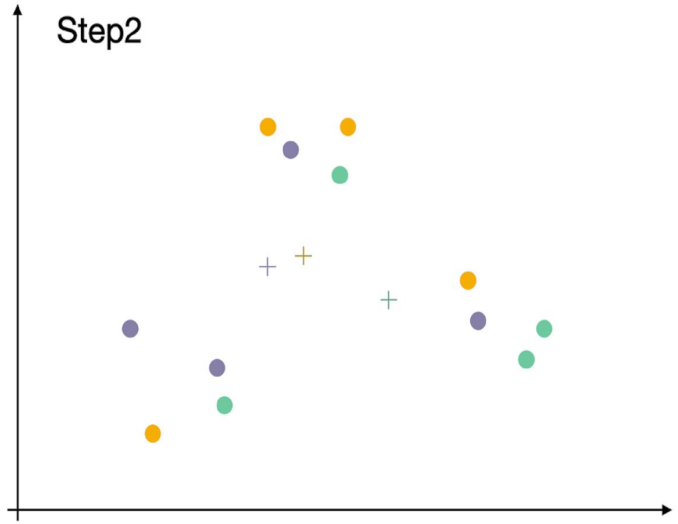
Step 2:
Calculate the centroid of each cluster.

Procedure

EXAMPLE: Clusters = 3

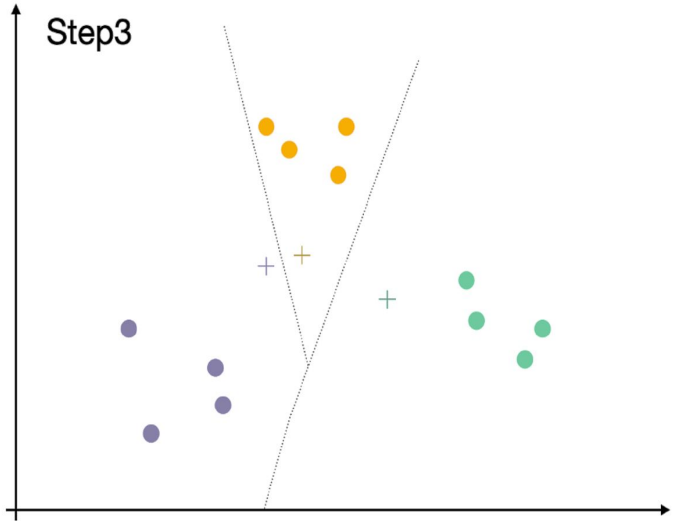
Step 1:
Assign a class (1,2,3) to each observation at random.





Step 2:
Calculate the centroid of each cluster.

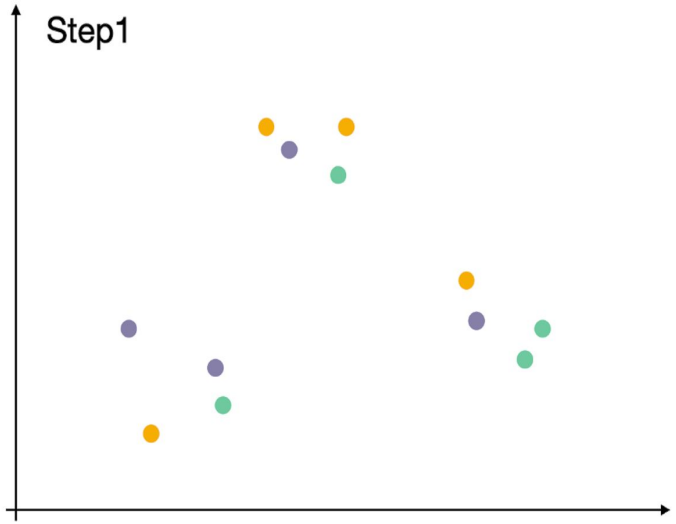
Step 3:
Update class labels for each data point, to its closest centroid.

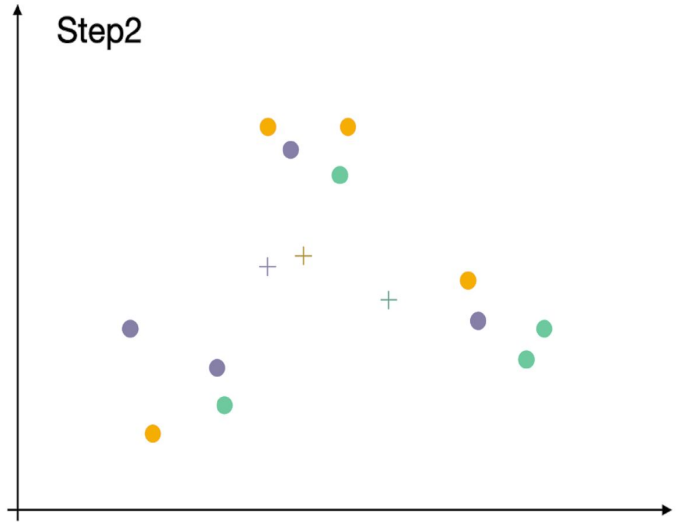


Procedure

EXAMPLE: Clusters = 3

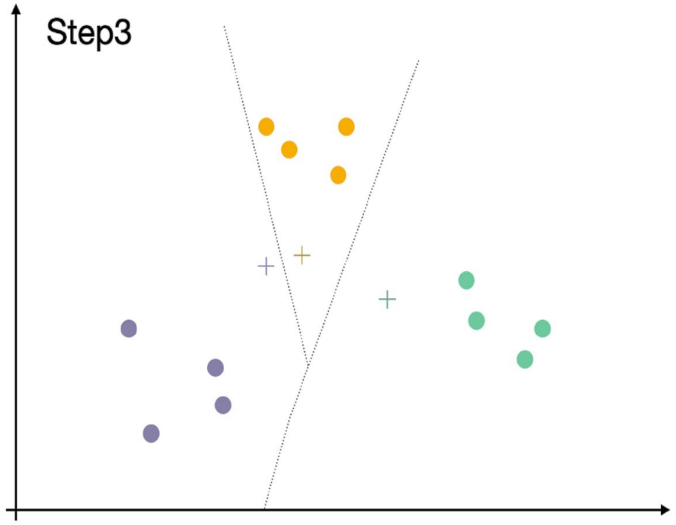
Step 1:
Assign a class (1,2,3) to each observation at random.

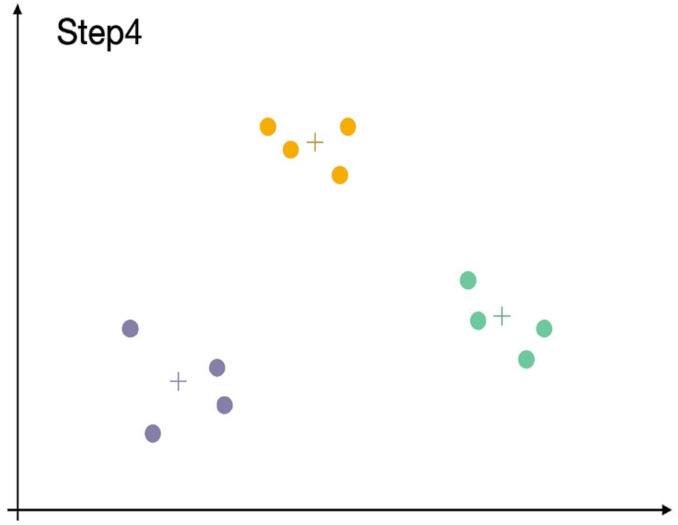




Step 2:
Calculate the centroid of each cluster.

Step 3:
Update class labels for each data point, to its closest centroid.

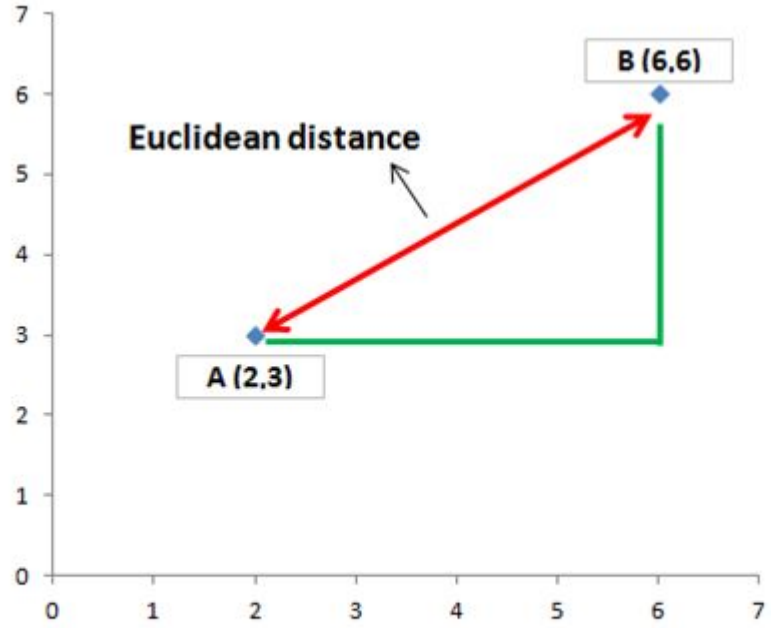




Step 4:
Repeat Steps 2, 3 until the class assignment remains unchanged.

How to compute Distance between points?

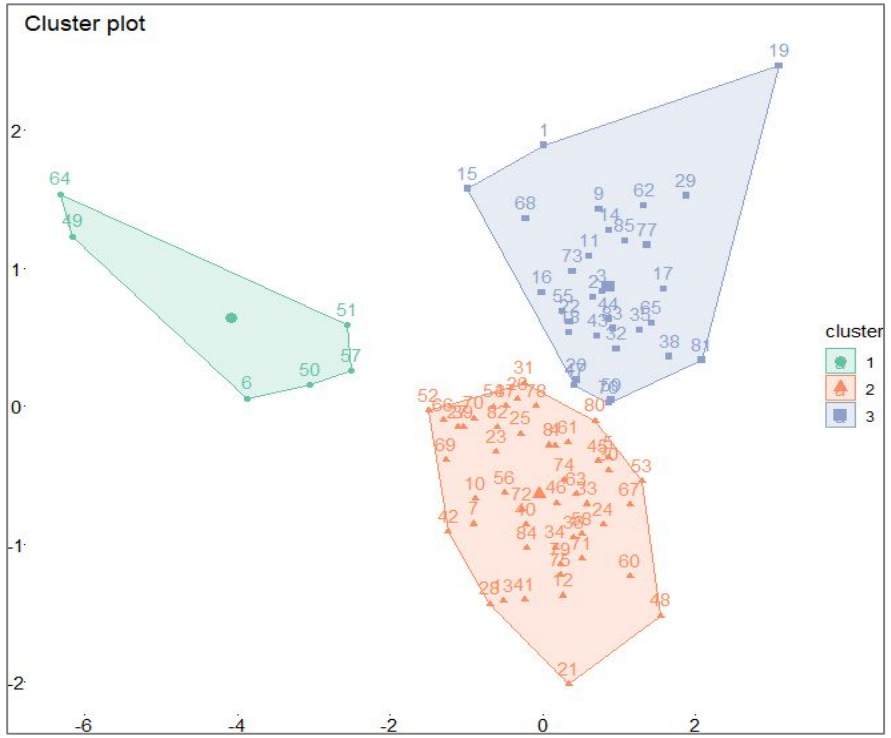
Euclidean Distance is most often used metric



$$Euclidean\ distance\ (a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

```python
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import seaborn as sns
import matplotlib.pyplot as plt

# KMeans segmentation (clustering)
k = 3
kmeans = KMeans(n_clusters=k, n_init=25)
kmeans.fit(data_all_scaled)
cluster_labels = kmeans.labels_
cluster_centers = kmeans.cluster_centers_
```




```python
# Visualize segments
plt.figure(figsize=(8, 6))
sns.scatterplot(x=data_all_scaled[:, 0], y=data_all_scaled[:, 1], hue=cluster_labels, palette='Set2')
plt.title('KMeans Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```

Segmentation approach

Leverage Machine Learning to identify number of distinct segments

Define segment specific strategies
Learn from the successful segments

Prepare data

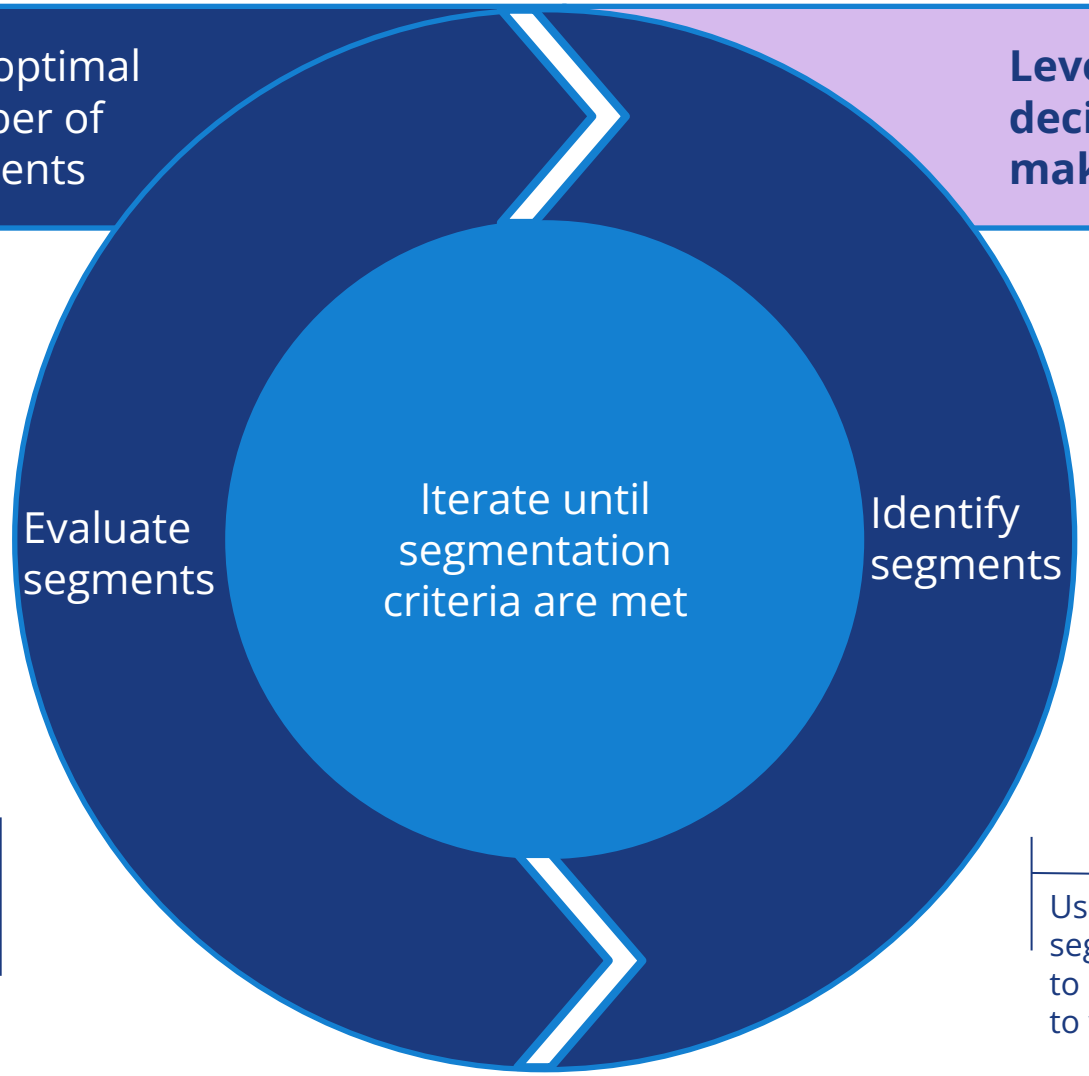
Define segmentation focus

Find optimal number of segments

Leverage for decision making

Merge data from various dataset to have wide coverage of properties

Define what should be the focus of segmentation



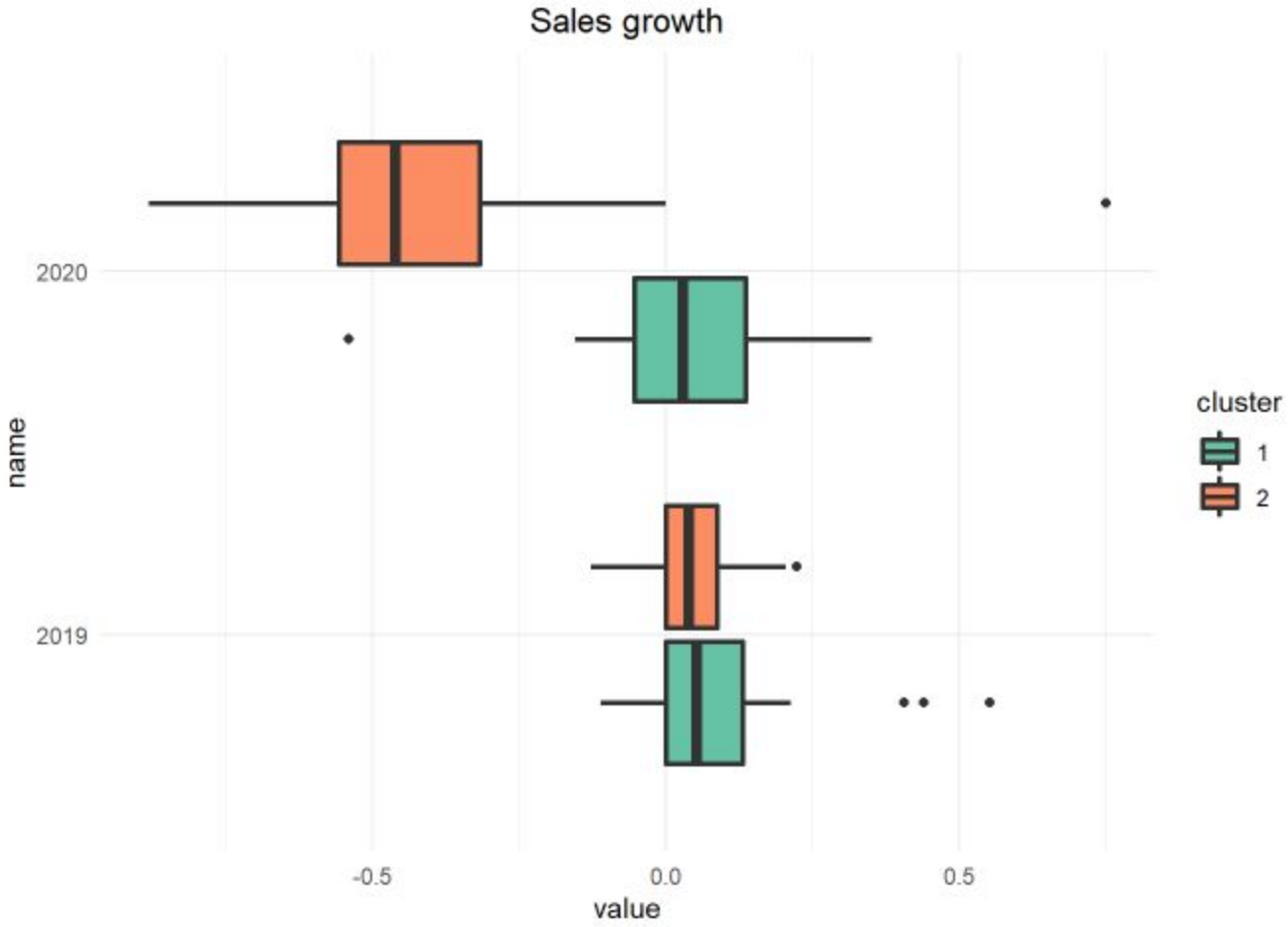
Understand properties of each segment and visualize
for ease of interpretation

Use k-means segmentation
to identify what belongs to which segment

Presenting Segmentation Results

- You need to show business stakeholders that there is indeed difference between the segments
- You need to choose visualization form that will suit business audience and will be understandable
- You need to show business stakeholders that these differences are such that you can act on them (have different strategy for different segments)

More technical type of visualization (Boxplots for distribution)



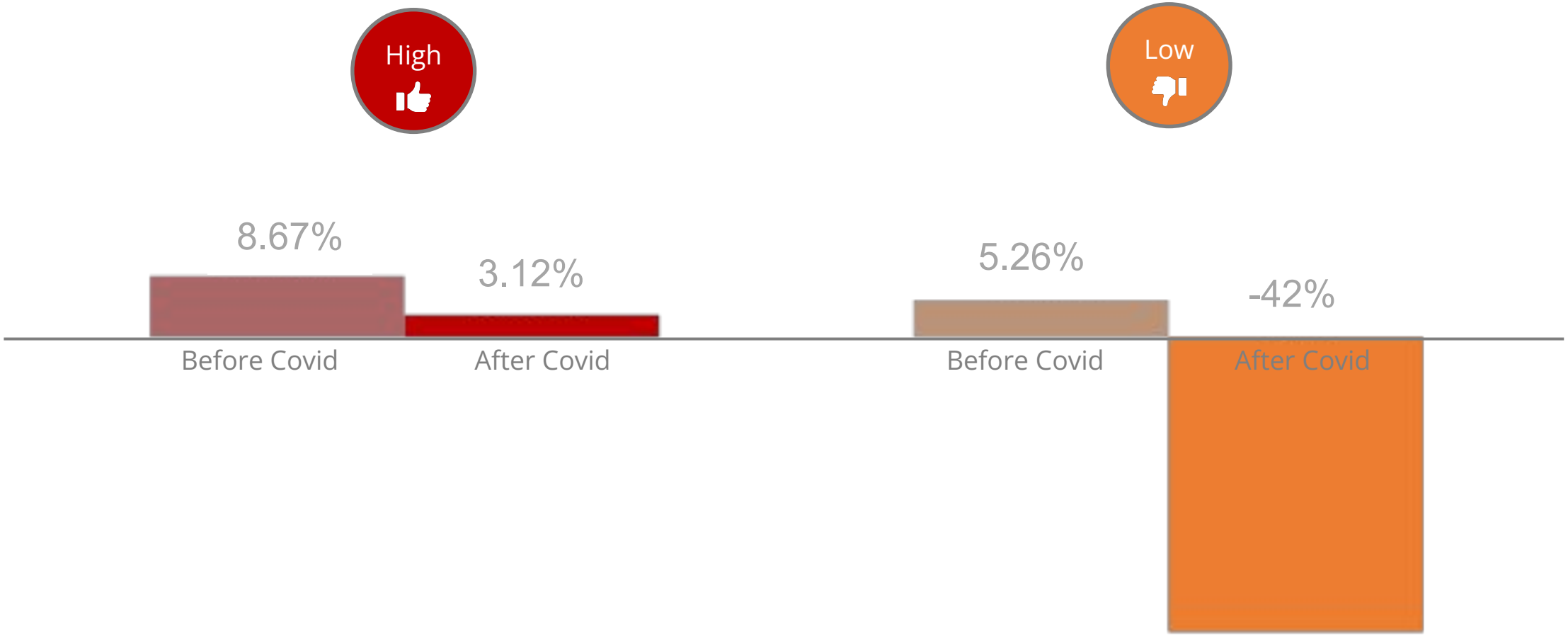
Case Explained:

Segmentation uncovered to segments of stores

1st segment was able to keep sales performance quite similar to pre-covid times. Some stores experienced decline but not big one, and some stores grew even quicker than before

2nd segment was hit significantly by Covid and majority of stores in the second segment encountered sales decline

Less technical type of visualization (Bars for averages)



Summary

k-means is typically used for segmentation

Important to define success and characteristics describing success criteria

Pick appropriate number of clusters

Validate and interpret model results

Principal Component Analysis

Principal Component Analysis (PCA)

Idea

Highly correlated variables are measuring same or very similar characteristics

Example: height VS leg length

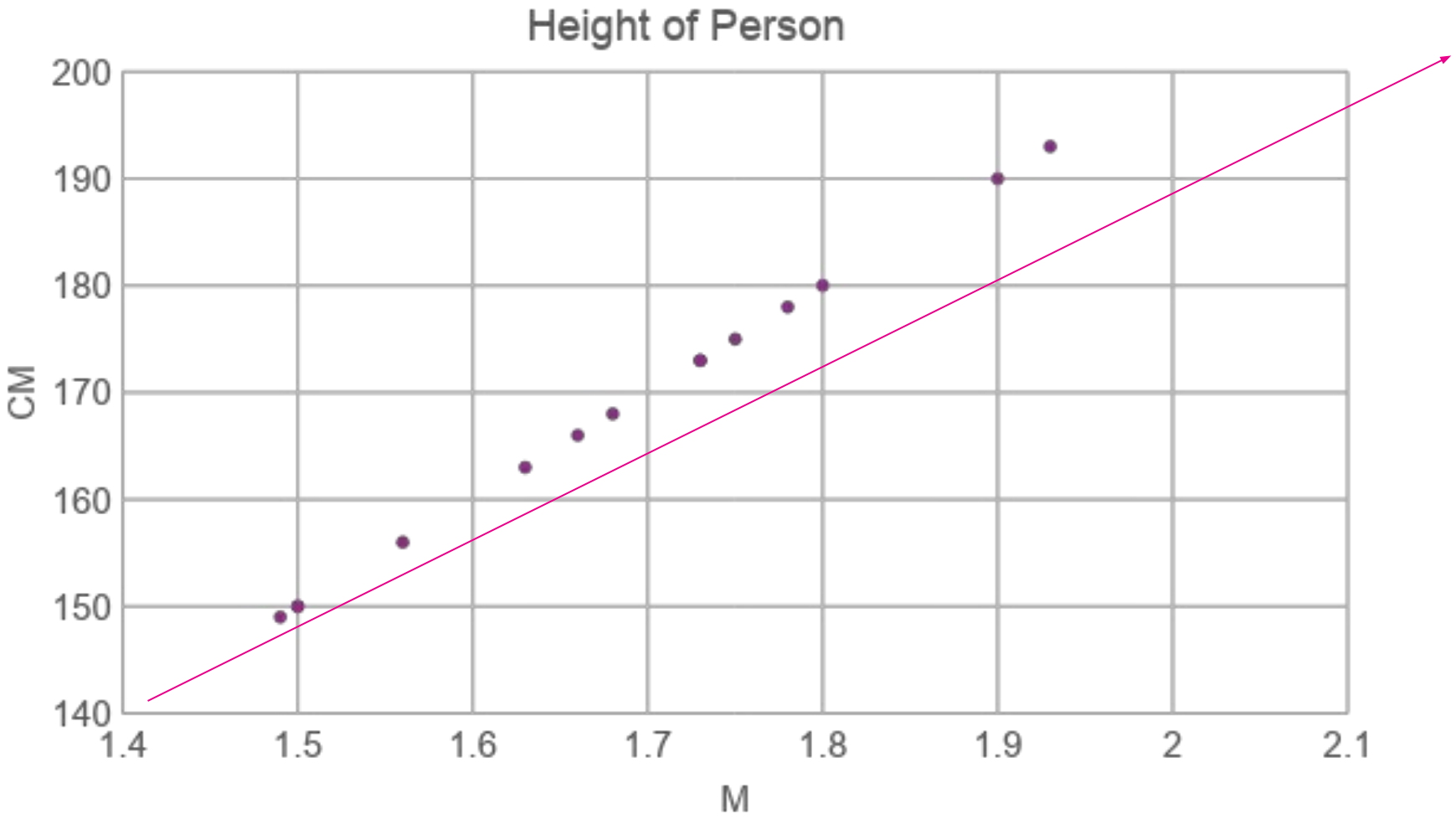
PCA

Combines shared variability of highly correlated variables into new variables
Principal Components

Goals

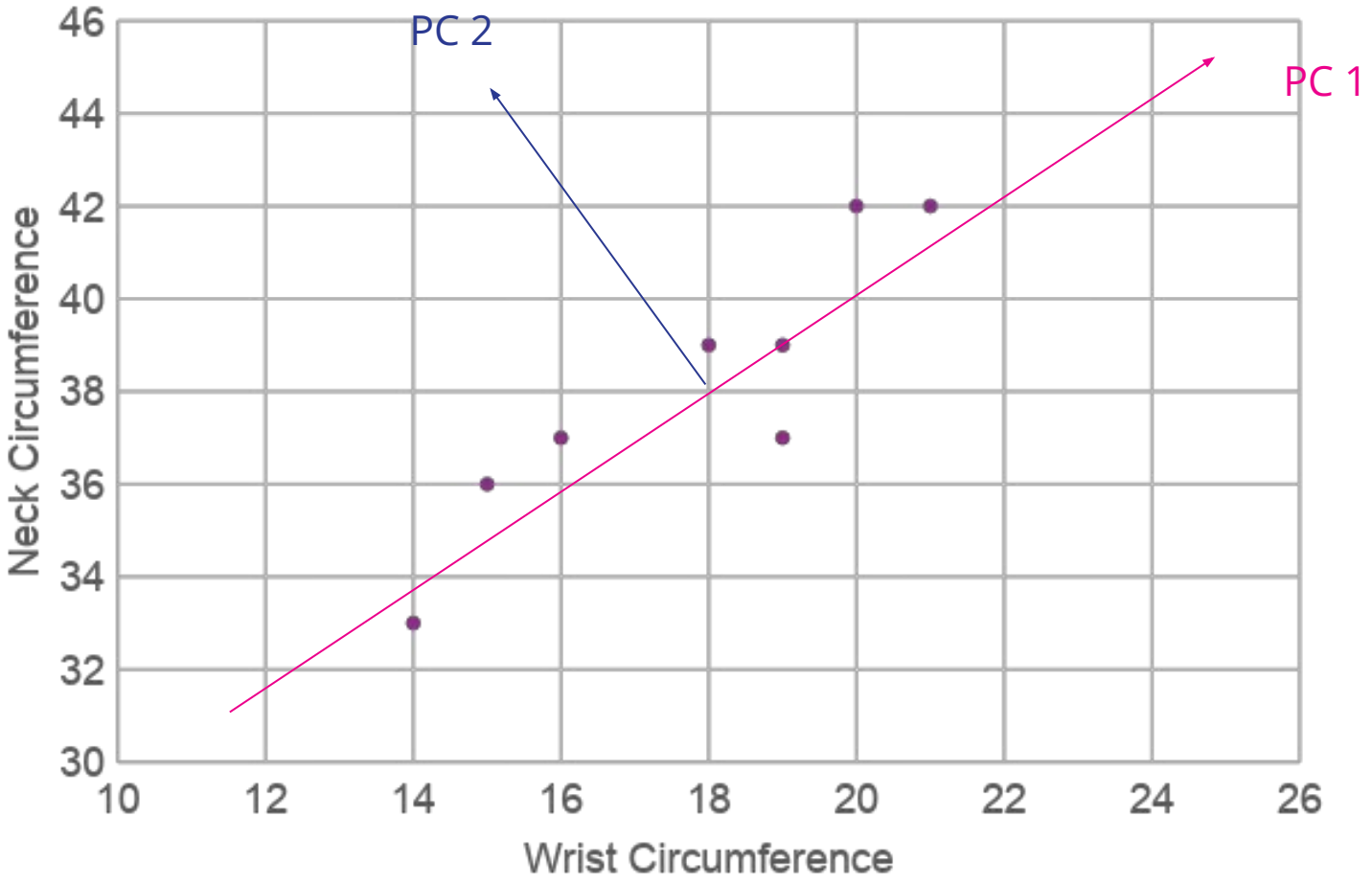
- Reduce dimensionality – replace **many** original variables with **few** principal components
- Identifying underlying factors of data

PCA - Motivation



czechitas

PCA - Motivation



Principal Component 1

$$PC_1 = v_{11}Wrist + v_{12}Neck$$

- Replicate as much variability as possible

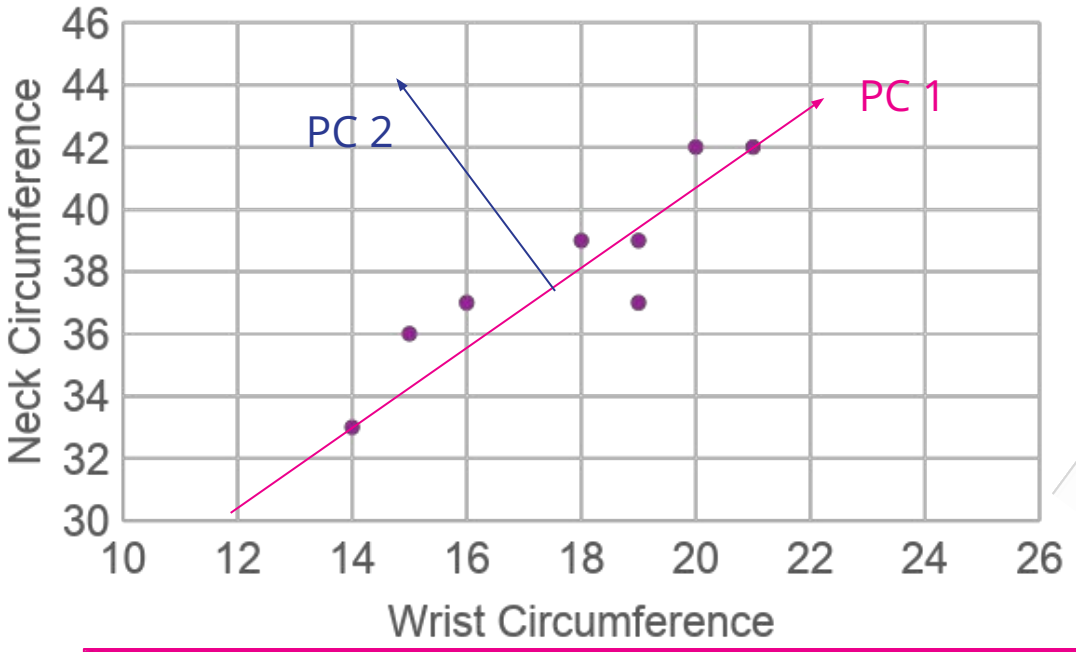
Principal Component 2

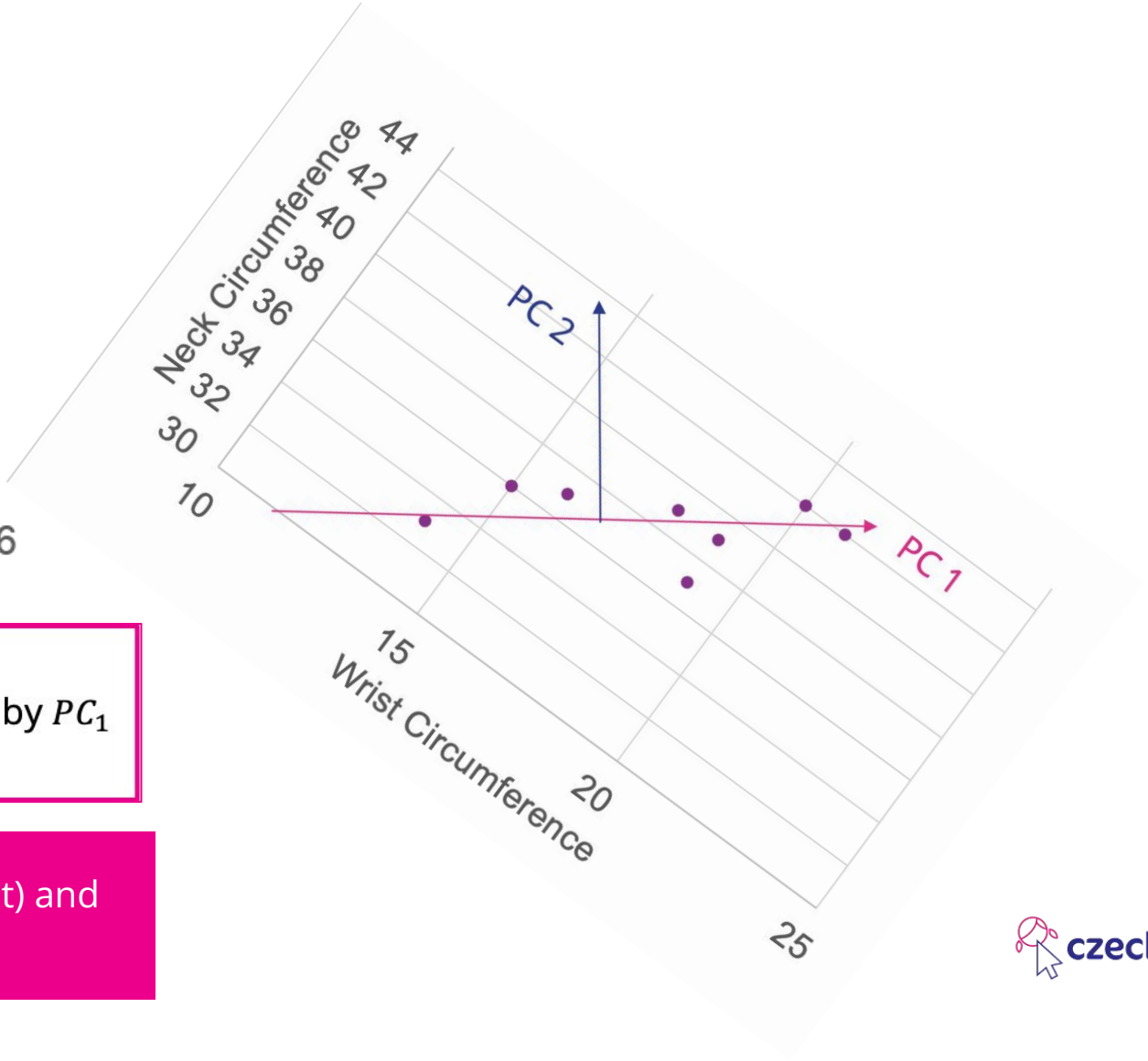
$$PC_2 = v_{21}Wrist + v_{22}Neck$$

- Replicate as much variability as possible that is not captured by PC_2

czechitas

PCA - Visualisation





PC_2 captures only variability that was not captured by PC_1

Principal Components are orthogonal (independent) and can be interpreted as **new X and Y axis**

czechitas

PCA – Sequential Optimization Problem

Principal Component 1

$$PC_1 = v_{11}X_1 + v_{12}X_2 + \ldots + v_{1k}X_k$$

Maximize

$$Var[v_{11}X_1 + v_{12}X_2 + \ldots + v_{1k}X_k]$$

Subject to

- Vector of coefficients v_1 has length 1

Principal Component 2

$$PC_2 = v_{21}X_1 + v_{22}X_2 + \ldots + v_{2k}X_k$$

Maximize

$$Var[v_{21}X_1 + v_{22}X_2 + \ldots + v_{2k}X_k]$$

Subject to

- Vector of coefficients v_1 has length 1
- PC_2 is independent of PC_1

...

Principal Component k

$$PC_1 = v_{k1}X_1 + v_{k2}X_2 + \ldots + v_{kk}X_k$$

Maximize

$$Var[v_{k1}X_1 + v_{k2}X_2 + \ldots + v_{kk}X_k]$$

Subject to

- Vector of coefficients v_1 has length 1
- PC_2 is independent of $PC_1, PC_2, \ldots, PC_{k-1}$

PCA – Dimensionality reduction

- **k variables** is transformed to **k principal** components
- Sequential optimization
  1) 1st PC explains as much variability as possible
  2) 2nd PC explains as much variability as possible that was not explained by 1st PC
  3) 3rd PC explains as much variability as possible that was not explained by 2nd and 3rd PC.

  ...

Dimensionality Reduction	Replace all k variables by first few principal components.

czechitas

Practical Application of PCA

1) Decide whether data are suitable for PCA
2) Scaling
3) Calculate PCs and decide how many include in the new data set
4) Usage
    - Use PCs for modeling
      (do not apply PCA to dataset including explained variable!)
    - Try interpret meaning of PC
    - “Compression”

PCA – Suitability

- Dimension reduction possible only if there are some strongly correlated variables
- Tools for checking suitability

Correlation Matrix

- Look for high correlations, e.g., $|corr| > 0.7$
- Potentially exclude variables that are not correlated with anything

Kaiser-Meyer-Olkin KMO Test

- Metric based on correlation and partial correlation (not formal test)
- KMO is from interval $(0,1)$

KMO	(0,0.49)	(0.5,0.59)	(0.6, 0.69)	(0.7, 0.79)	(0.8, 0.89)	(0.9, 1)
Interpretation	unacceptable	miserable	mediocre	middling	meritorious	marvelous

czechitas

PCA - Scaling

Principal Component 1

$$PC_1 = v_{11}X_1 + v_{12}X_2 + \ldots + v_{1k}X_k$$

Maximize

$$Var[v_{11}X_1 + v_{12}X_2 + \ldots + v_{1k}X_k]$$

Subject to

- Vector of coefficients v_1 has length 1

- Variables have to be demeaned
- The optimization favors variables with higher variance
- X_1 having more variance than X_2 does not mean it has more valuable information
- Apply standardization (scaling)

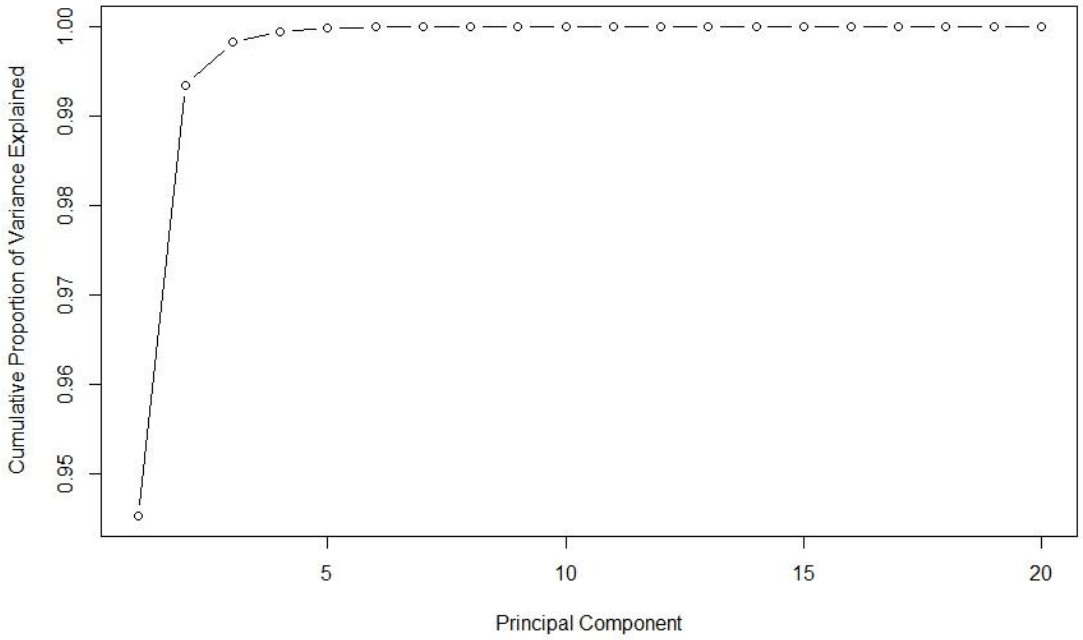
$$Z_j = \frac{X_j - \mathbb{E}[X_j]}{\sqrt{Var[X_j]}}$$

- Note: using common packages we usually do not have apply this, since the PCA implementation allows to use correlation matrix instead covariance matrix or apply scaling

PCA – Number of Components

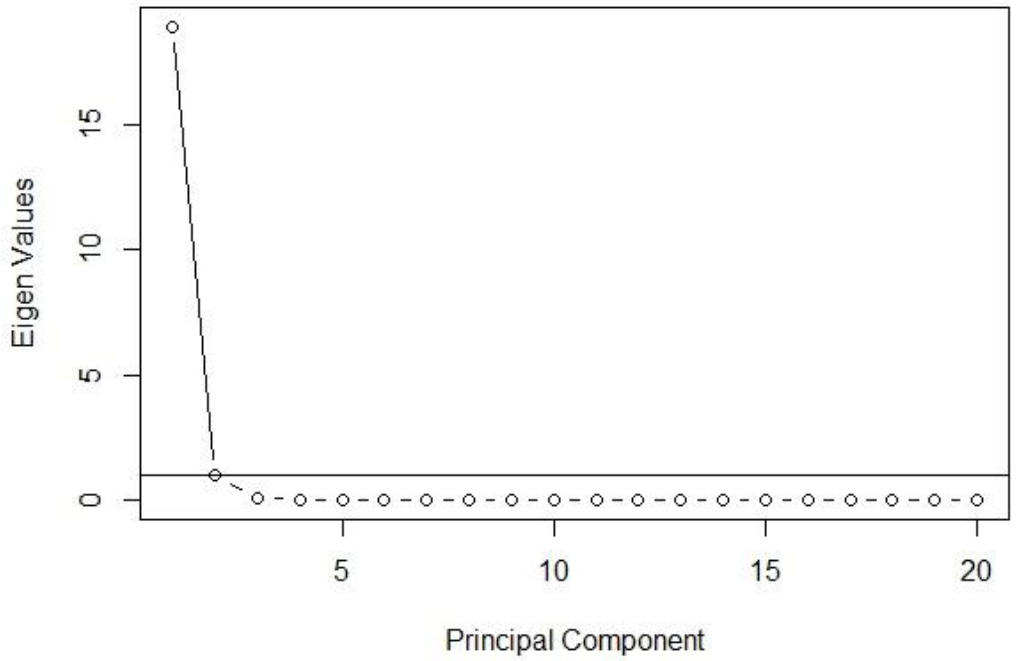
Cumulative Percentage of Explained Variance

Pick threshold for % of explained variance



Eigen Values

Rule-of-thumb eigenvalue > 1

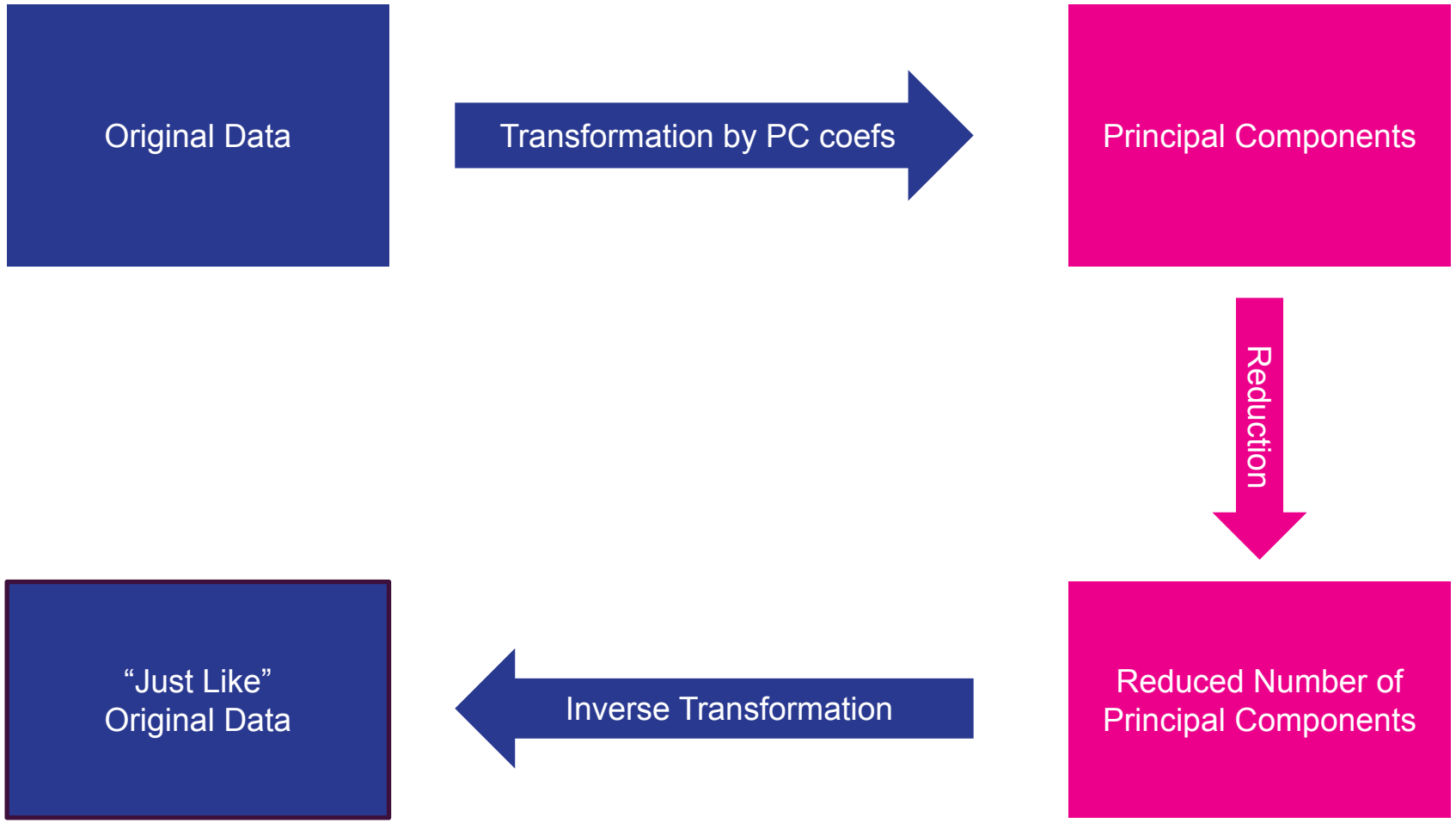


Note: These types of plots are called **scree plots**

czechitas

PCA – Inverse Transform

- Sometimes we want to reconstruct original data from principal components



czechitas

Yield Curve Demo

Other Applications

Image Compression

- Bitmap Images
  - 3 matrices – red, blue, green
  - each element represents pixel
- PCA Compression
  1) Apply PCA to each matrix
  2) Store reduced number principal components with coefficients
  3) Reconstruct original image from principal components when needed
- Note: algorithm above reduces number of columns

PCA usage example: picture compression

https://a.disquscdn.com/uploads/mediaembed/images/3608/625/original.jpg

PCA usage example: picture compression

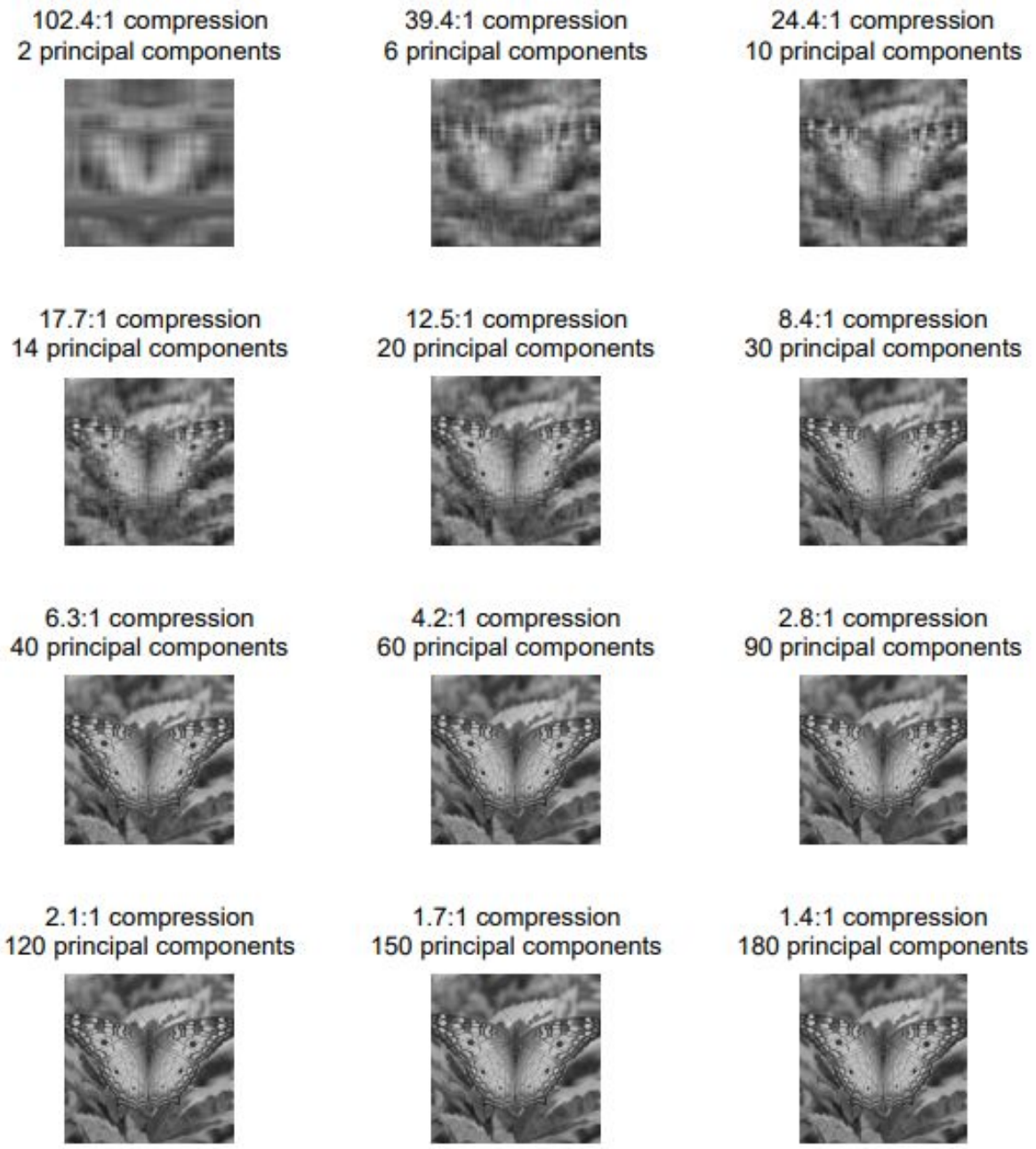


Figure 10: The visual effect of retaining principal components

http://www.dsc.ufcg.edu.br/~hmg/disciplinas/posgraduacao/rn-copin-2014.3/material/SignalProcPCA.pdf

PCA usage example: signal processing

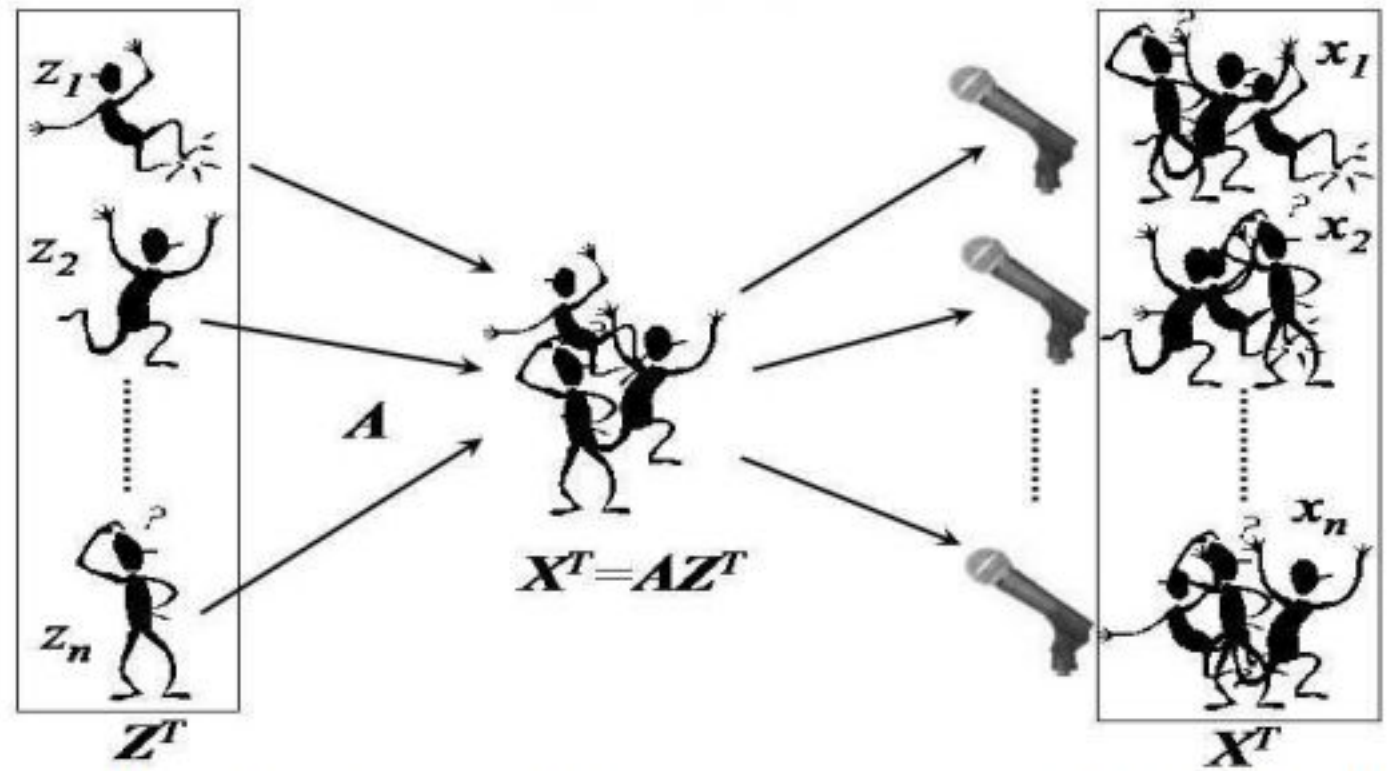


Figure 11: The cocktail party problem (image courtesy of Gari Clifford, MIT)

- de-noise the group signal
- separate out the original sources.

http://www.dsc.ufcg.edu.br/~hmg/disciplinas/posgraduacao/rn-copin-2014.3/material/SignalProcPCA.pdf

PCA - Summary

- Idea: replace dataset with many variables by lower number of principal components
- Properties
  - components are constructed linearly from the original data
  - applicable only to strongly correlated variables
- Applications
  - dimensionality reduction
  - multicolinearity (using LASSO regression instead PCA recommended)
  - compression
  - signal processing

We are looking forward to the next lecture!

Thank you for your attention.