FACULTY OF APPLIED SCIENCES
UNIVERSITY
OF WEST BOHEMIA

Ostbayerische
Technische Hochschule
Amberg-Weiden

PITFALLS AND WEAKNESSES OF PRINCIPAL COMPONENT ANALYSIS

© 2023 OTH Amberg-Weiden
Christian Rute | 92224 Amberg

© 2023 University of West Bohemia
Jan Šimek | 301 00 Pilsen

Agenda

Our plan for today

01 Introduction

02 Understanding PCA

03 Potential Pitfalls and Weaknesses of PCA

04 Practical PCA experiment

05 Conclusion

1.0 INTRODUCTION

1.1 Introduction

How it started

Recap

We were both at the 2nd AI Summer School. After we had both listened to the lectures, we decided to create a small project on one of the topics.

So, one of the possible projects was the pitfalls and weaknesses of PCA –

Principle Component Analysis

1.2 Introduction

What is principal component analysis?

Definition of Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique to simplify high-dimensional data by transforming it into a lower-dimensional representation while retaining most of the explainability of the information.

Purpose and Benefits of PCA

The main purpose of PCA is to identify the most important features or variables that explain the most significant variance in the data. This can help in reducing the dataset and removing noise, it can also help to choose the right features.

2.0 UNDERSTANDING PCA

2.1 Assumptions

Linear relationship between all variables

The PCA algorithm assumes that the relationship between variables are linear. If the relationship is nonlinear, the algorithm may not produce meaningful results.

Sampling adequacy

So that our **PCA** algorithm can deliver appropriate results, we need samples that are not too small -> large enough sample sizes are required.

Suitable data for data reduction

In order for the variables to be reduced to a smaller number of components, there must be sufficient correlations between the variables.
If this condition is not given, this algorithm cannot do much.

2.2 Data scaling

Importance of Scaling Data for PCA:

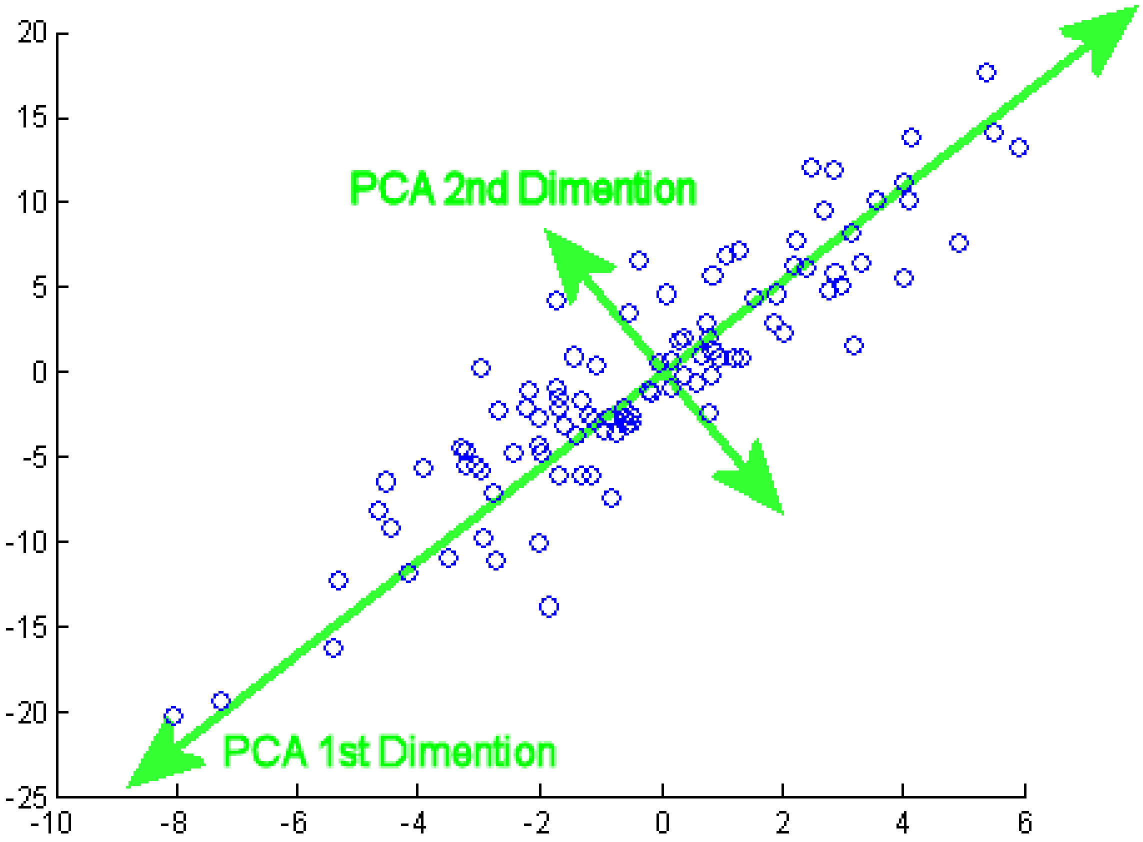
Scaling the data is crucial in PCA to ensure that all variables contribute equally to the analysis. Variables with larger scales can dominate the analysis and obscure the influence of smaller-scale variables.

Issues with Different Scales:

When variables are measured on different scales, PCA can be biased towards variables with larger scales, leading to inaccurate results. Scaling the data to a common scale eliminates this bias.

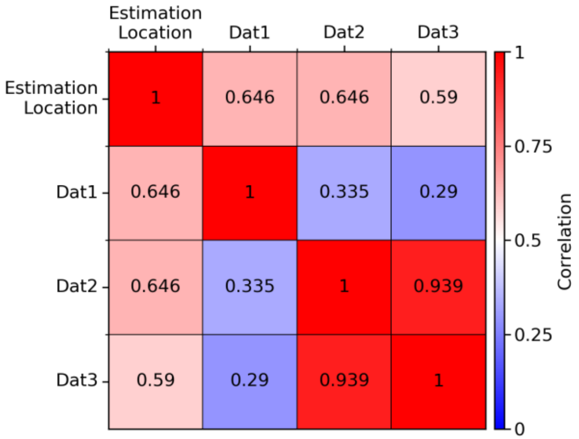
2.2 Data scaling

Visualization



2.3 Correlation structure

So, with the covariance matrix we can revealing the relationships between variables. Eigendecomposition, on the other hand, breaks down a matrix into eigenvalues and eigenvectors, providing critical insights into the data's variance and direction.



	Estimation Location	Dat1	Dat2	Dat3
Estimation Location	1	0.646	0.646	0.59
Dat1	0.646	1	0.335	0.29
Dat2	0.646	0.335	1	0.939
Dat3	0.59	0.29	0.939	1

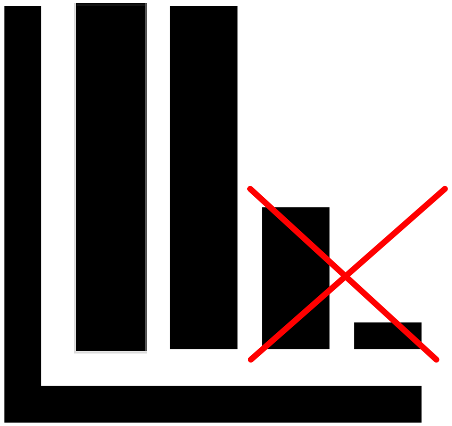
- Covariance matrix: Reveals variable relationships.
- Eigendecomposition: Breaks down a matrix into eigenvalues and eigenvectors.

2.4 Dimensionality reduction

Thus, when applying PCA, the user can specify the number of dimensions to be reduced.

Process of Reducing Dimensions:

PCA reduces the dimensionality of the dataset by transforming the original variables into a new set of *uncorrelated* variables called **principal components**. These principal components are ordered based on their ability to explain the variance of the data.



3.0 POTENTIAL PITFALLS AND WEAKNESSES

3.1 Non-linear relationships

Inability of PCA to capture non-linear relationships:
PCA implies that it assumes the *relationships between variables are linear*. However, in many real-world scenarios, relationships between variables are non-linear.

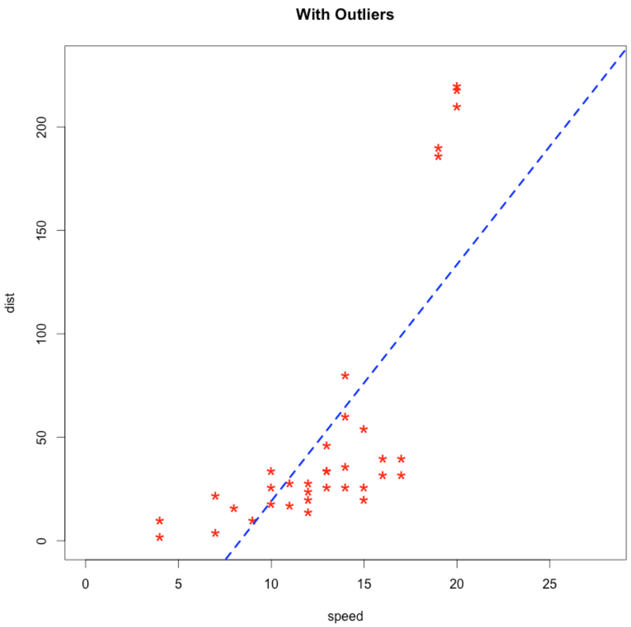
Alternative methods for non-linear data:
Alternative dimensionality reduction techniques may be more suitable. Methods such as Kernel PCA and Non-linear Dimensionality Reduction (NLDR) algorithms are designed to capture non-linear relationships in the data, providing better solutions for these situations.

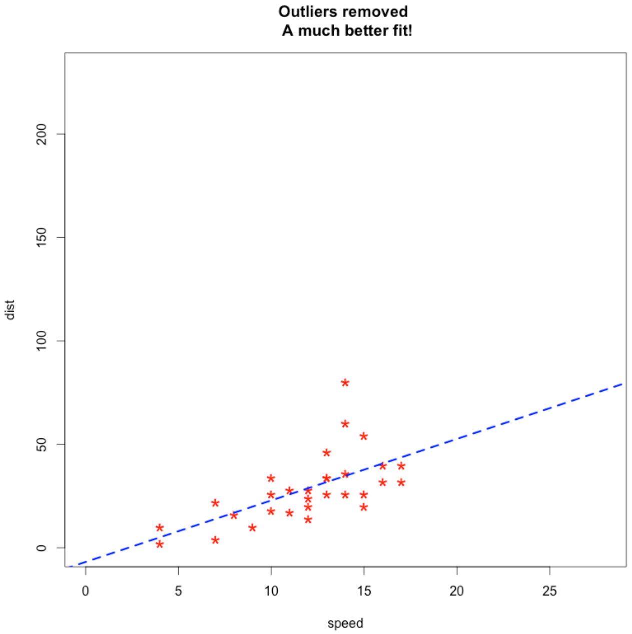
3.2 Pitfalls in Scaling

Pitfalls in Scaling Data:

Incorrect Scaling Methods: Choosing the wrong scaling method for your data can lead to inappropriate results.

Data Quality: If outliers in the data are not removed, they can have a major impact on the data when scaling is applied.





3.3 Beware of Outliers

Impact of Outliers on PCA Results:
They can distort the variance explained by the principal components, thereby affecting the recognition of meaningful data patterns.

Possible solution:
We can simply remove our Outliers or replace them with more representative values.

But we have to consider some consequences:
Data Loss: The removal of outliers can lead to data loss and replacing them may introduce bias into the analysis.

3.5 Interpretability

Difficulties in Interpreting Principal Components:

Principal components are linear combinations of the original variables, making them challenging to interpret in terms of the original data. They *represent patterns in the data* but may not have direct and intuitive meanings, requiring careful analysis and context.

?

Lack of Transparency in Feature Contribution:

PCA does not explicitly provide information about the contribution of each original variable to the principal components. This lack of transparency can make it difficult to interpret the importance of individual variables, and additional analysis may be required to understand their influence on the derived components.

3.6 Common mistakes

Common mistakes:

1. Failing to Standardize or Scale Data
2. Misinterpreting Principal Components as Causal Factors
3. Incorrectly Assuming Component Importance
4. Overlooking PCA's Limitations

Avoidance tips:

1. Always Standardize and Scale Data appropriate
2. Understand the Nature of Principal Components
3. Evaluate Component Contribution and Cumulative Explained Variance
4. Be Mindful of PCA Assumptions and Limitations

4.0 PRACTICAL PCA EXPERIMENT

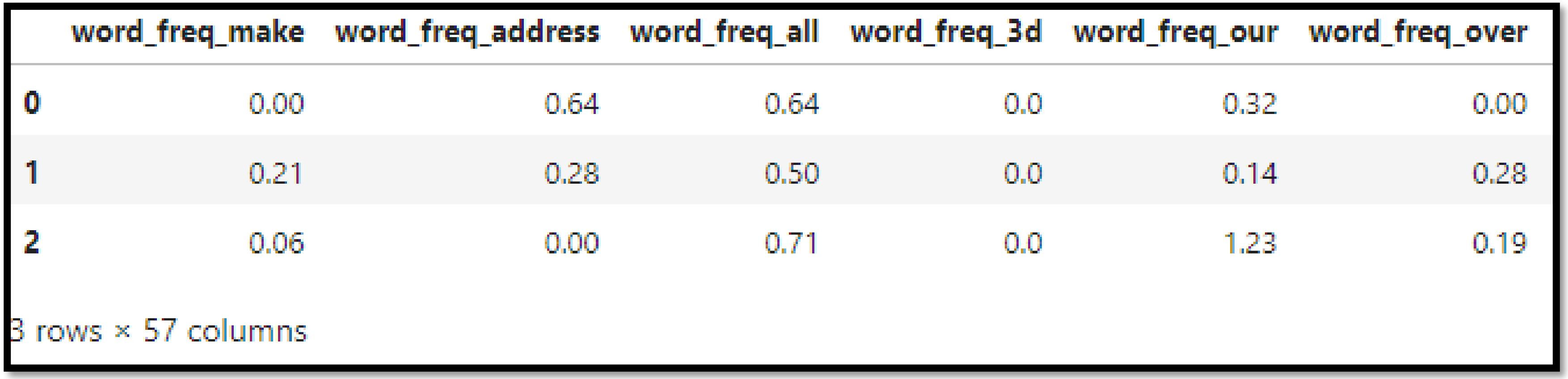
4.1 Aim of our experiment

- Apply PCA on real data
- Compare classification of original and reduced-by-PCA data
- Identify some pitfalls and weaknesses of PCA in problem of classification

- Implementation: Python
  - Google Colab/Jupyter Notebook, Miniconda

4.2 Input data

- Chosen dataset: Spambase - Classifying emails as Spam or Non-Spam
- 4601 instances
- Each instance: 57 features + label (0 – not spam, 1 – spam)
- Features represented by float numbers rounded by 2-3 decimal places



	word_freq_make	word_freq_address	word_freq_all	word_freq_3d	word_freq_our	word_freq_over
0	0.00	0.64	0.64	0.0	0.32	0.00
1	0.21	0.28	0.50	0.0	0.14	0.28
2	0.06	0.00	0.71	0.0	1.23	0.19

3 rows × 57 columns

- Motivation: develop a precious spam filter for an e-mail box.

Spambase - UCI Machine Learning Repository

4.3 Chosen type of classifier

- K-NEAREST NEIGHBORS CLASSIFIER
- Classifier consists of real data sample with labels
- For item classification, find k-nearest data in the classifier set
  - Based on Euclidian distance
- Check labels of the neighbors and assign the major label to the item
- Easy implementation
- Acceptable precision (if we have "enough" data)
- Memory requirements

4.4 Procedure

1. Shuffle data randomly
2. Split the data for classification set and testing set
3. Classify the testing set and measure duration and precision
4. Apply PCA on the data
5. Classify the reduced-by-PCA testing set and measure duration and precision
6. Repeat points 1-5 10times
7. Compare results

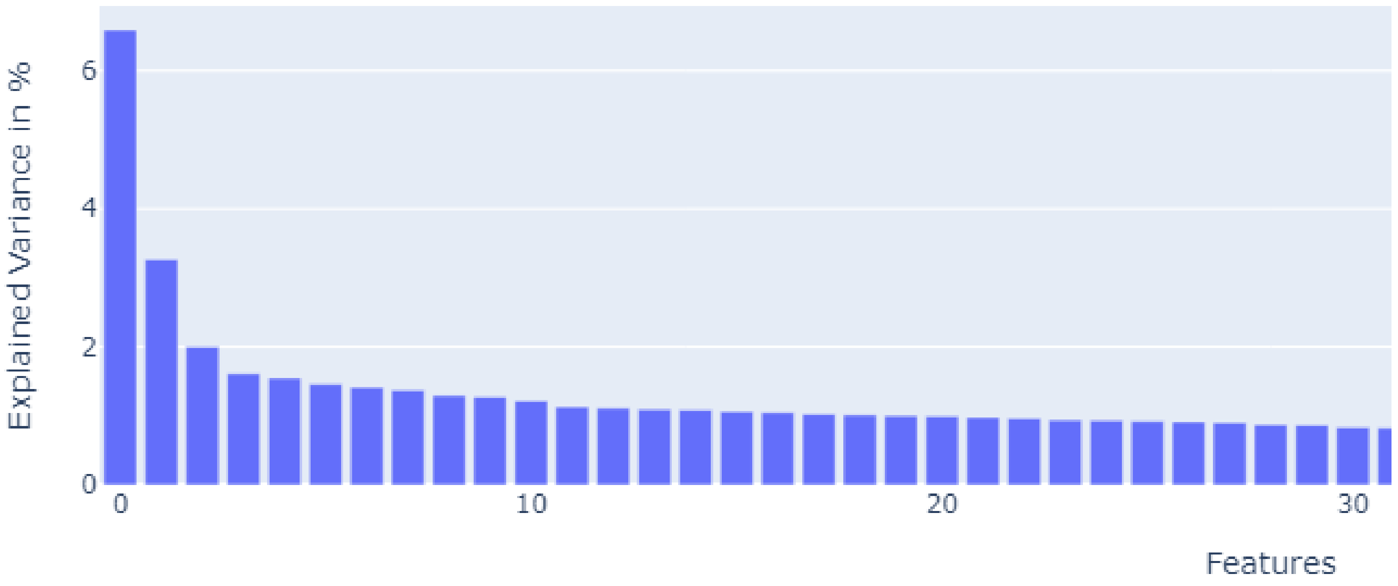
4.5 Observes before PCA

- Data shuffling and classification algorithm were executed 10times in a row
  - On the same machine
- Classification time and precision were measured
- Mean and variance were computed

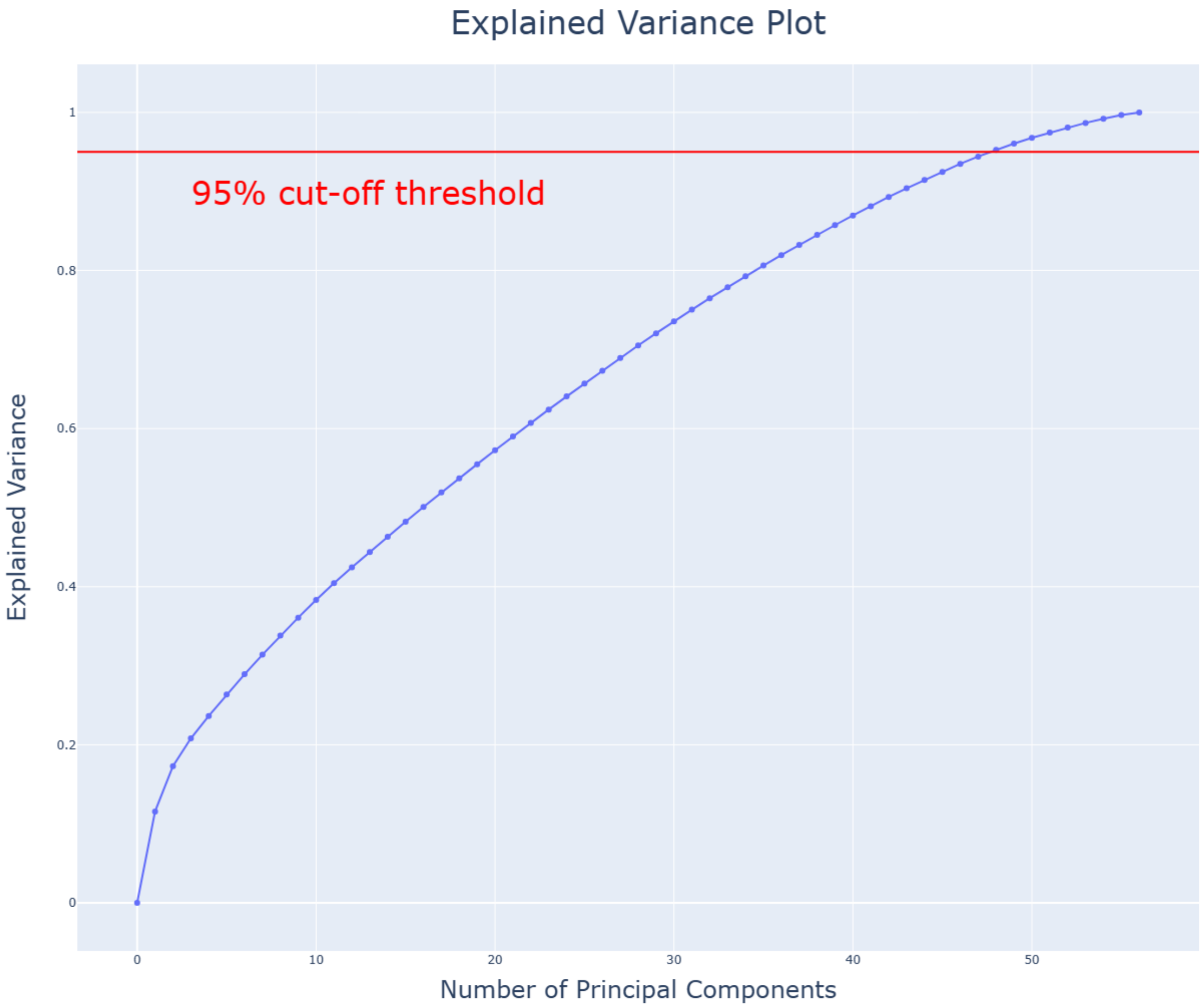
Input data:	Original data	
Measurement:	Classification time [s]	Precision [%]
Mean of 10 executions:	**42,34**	**80,84**
Variance of 10 executions:	9,44	0,94

4.6 PCA Analysis

Bar plot



4.6 PCA Analysis



4.7 Observes and results

- By applying PCA, the dimension of the data was reduced from 57 to 48
  - With 5% information loss

Input data:	Original data		PCA data		Comparsion	
Measurement:	Classification time [s]	Precision [%]	Classification time [s]	Precision [%]	Time diff [s]	Precision diff [%]
Mean of 10 executions:	42,34	80,84	48,31	90,98	5,97	10,15
Variance of 10 executions:	9,44	0,94	6,20	1,87	6,29	2,24

- <u>After PCA application:</u>
- Precision of classification is about 10% higher
- Time duration of classification is about 6 seconds longer

4.8 Possible explanation of results

- PCA can emphasize important properties of the dataset and suppress the pointless ones

- Even though the dimension reduction, original data would be represented better for computing.
  - We explain the duration increase with a lot of zero values in original dataset
  - About 77% of values in original dataset are zeros

5.0 CONCLUSION

5.1 Conclusion

- **PCA Benefits**
- Data dimension reduction
- Emphasizing key data properties
- Suppressing redundant data properties
- Noise reduction

- **PCA Pitfalls and weaknesses**
- Lossy compression
- Optimal cut-off threshold selection (95% ?)
- Solution efficiency is not guaranteed
- Results may differ on various data
- Continuous data representation (float)

5.2 What we learned

- Always agree on data types before start coding parallelly!
  - Merging will be much easier :-)
- You can always impress someone by coding without libraries
  - Except your girlfriend :-)
- PCA is a powerful method, but:
  - You should know (at least abstractly) how it works
  - For non-demonstration usage it is better to use optimized libraries
  - Better results are not always guaranteed

Feel free to ask:

c.rute@oth-aw.de

simio@students.zcu.cz

THANK YOU FOR YOUR ATTENTION

Quellen:

- How to perform a principal components analysis (PCA) in SPSS Statistics | Laerd Statistics
- These kriging weights are really, really ridiculously good looking... (lazymodellingcrew.com)
- Spambase - UCI Machine Learning Repository
- https://medium.com/@raghavan99o/principal-component-analysis-pca-explained-and-implemented-eeab7cb73b72

Quellen: