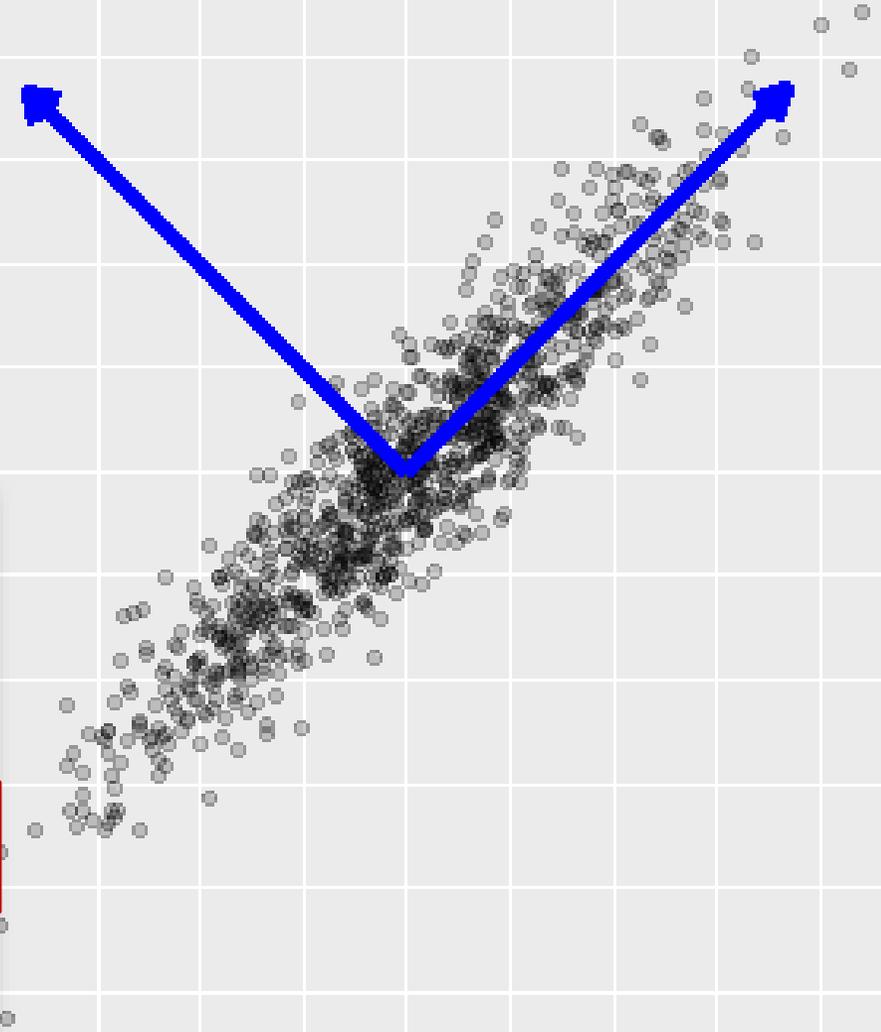
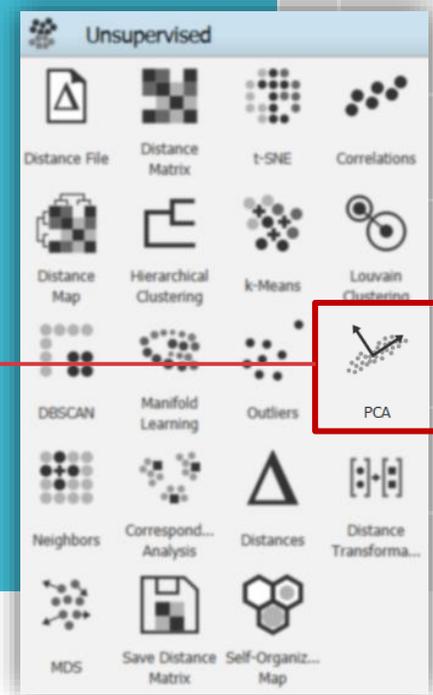


# PCA

Umgang im Unterricht mit komplexen Datensätzen & Orange Data Mining

Workshop LFT KI 24.1.2023:  
Hauptkomponentenanalyse





## Das Problem

erstmal im Unterricht aufgetreten im Hectorseminar-Projekt 2022:  
Diabetes-Datensatz in Orange Data Mining

Pregnancies	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	Diabetes Pedigree Function	Age	Outcome
6	148	72	35	0	336	627	50	1
1	85	66	29	0	266	351	31	0
8	183	64	0	0	233	672	32	1
1	89	66	23	94	281	167	21	0
0	137	40	35	168	431	2288	33	1
5	116	74	0	0	256	201	30	0
3	78	50	32	88	31	248	26	1
10	115	0	0	0	353	134	29	0
2	197	70	45	543	305	158	53	1
8	125	96	0	0	0	232	54	1

Wie trainiert man da ein Modell, bzw. trifft eine Aussage?

DON'T BE SCARED OF NUMEROUS VARIABLES



PCA IS HERE TO HELP



## Wäre es nicht toll, wenn man Daten ‚komprimieren‘ könnte?

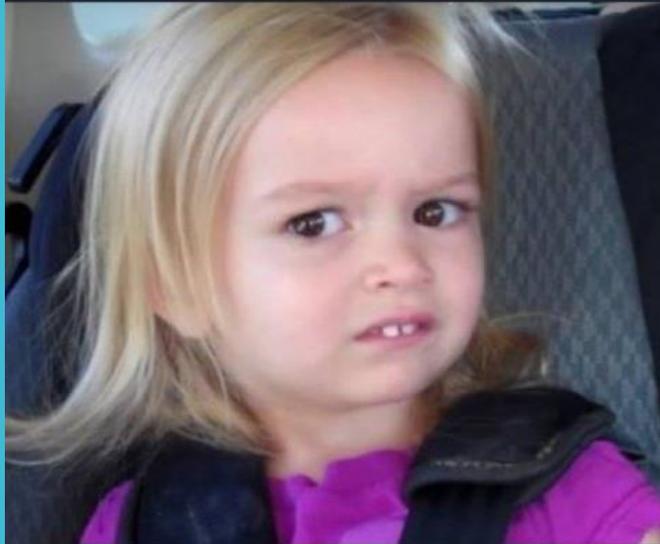
- um die wichtigen von den unwichtigen Features zu unterscheiden?
  - Und wenn man die unwichtigen Features wegnimmt, so hat man die Komplexität der Daten verringert
  - Was gleichbedeutend mit einer „Dimensionsreduktion“ ist.
  - Quasi eine Art „Rauschunterdrückung“ für Daten?
- ... Dann wären Daten leichter interpretierbar / visualisierbar.
- ... Modelle könnten leichter trainiert werden.
- ... und würden auch schneller laufen.
- ... und dabei sogar noch weniger Speicher benötigen.

*Und das alles für numerische, Bild- und Textdaten?*

# „Was die SuS am Ende der Stunde verstanden haben sollten ...“

Sammlung von wichtigen Konzepten und Begriffen:

**ME WHEN**



**MRS. JACOBS USES SO MUCH BIG VOCABULARY WORDS THAT I DON'T UNDERSTAND**

**Hauptkomponente**  
**Principal Component PC1, PC2 ...**

**Singulärwertzerlegung**  
**Singular Value Decomposition**

**Screeplot & kumulierte Varianz**  
**Scree Plot & explained Variance**

**Faktorwerte der Hauptkomponente**  
**loading scores of principal comp.**

**Normalisierung von Variablen**  
**Normalized Variables**

**Kovarianz & Kovarianzmatrix**  
**Covariance & covaricane matrix**

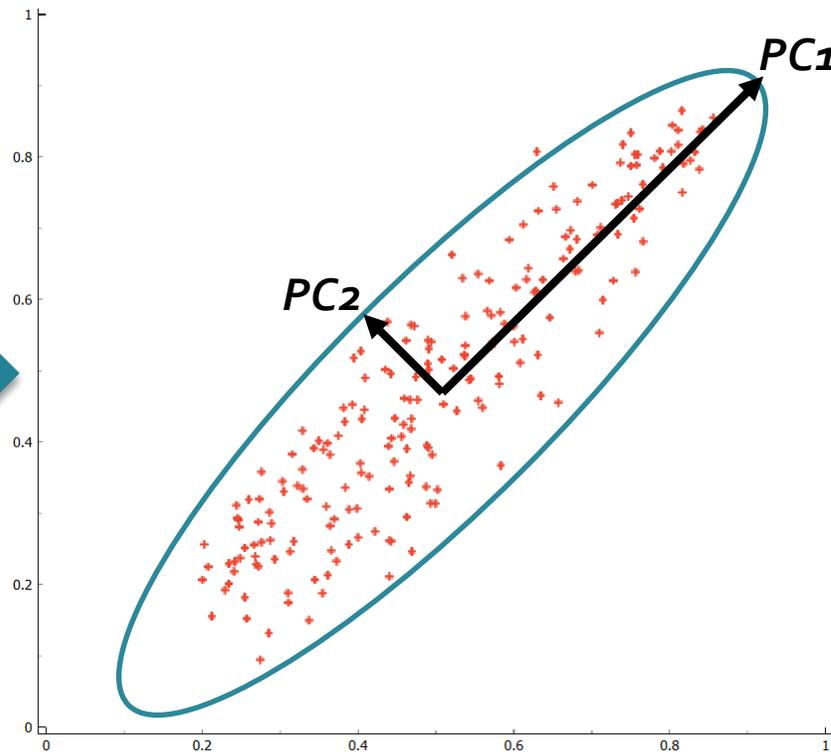
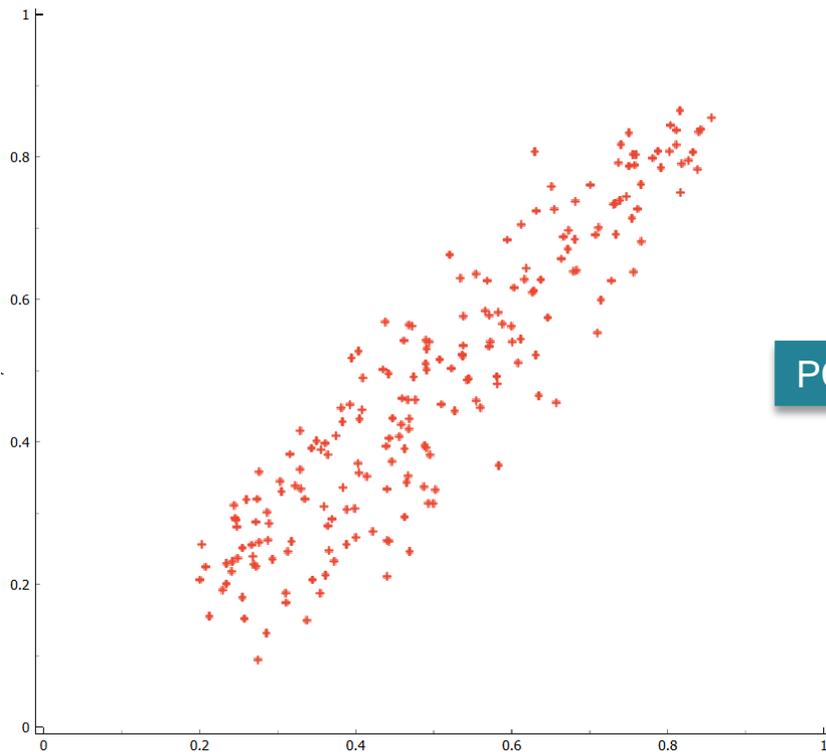
**Eigenvektor / Eigenwert**  
**Eigenvector / Eigenvalue**

Universitätsniveau ... nicht für SuS – aber für Lehrer

## Intuitives Konzept:

Die PCA kann man sich als Anpassung eines ( $n$ -dimensionalen) Ellipsoids an die Daten vorstellen, wobei jede Achse des Ellipsoids eine Hauptkomponente darstellt.

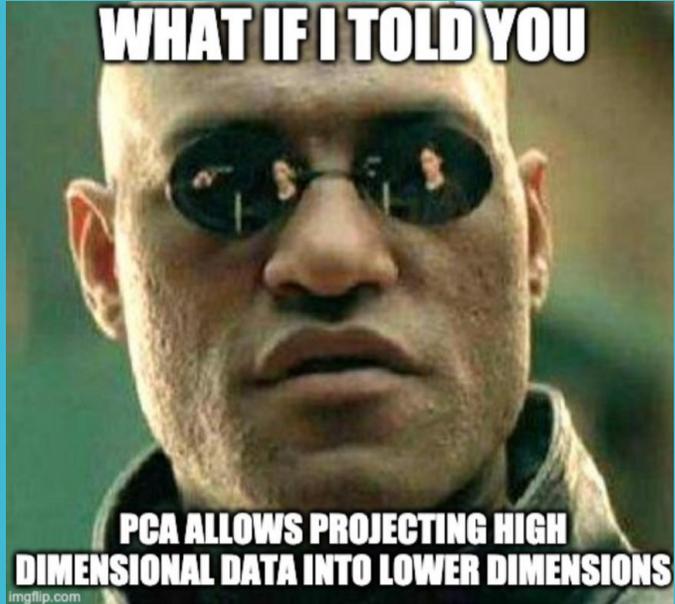
Wenn eine Achse des Ellipsoids klein ist, dann ist auch die Varianz entlang dieser Achse klein.



*PC1, PC2 sind die Eigenvektoren der Kovarianzmatrix.*

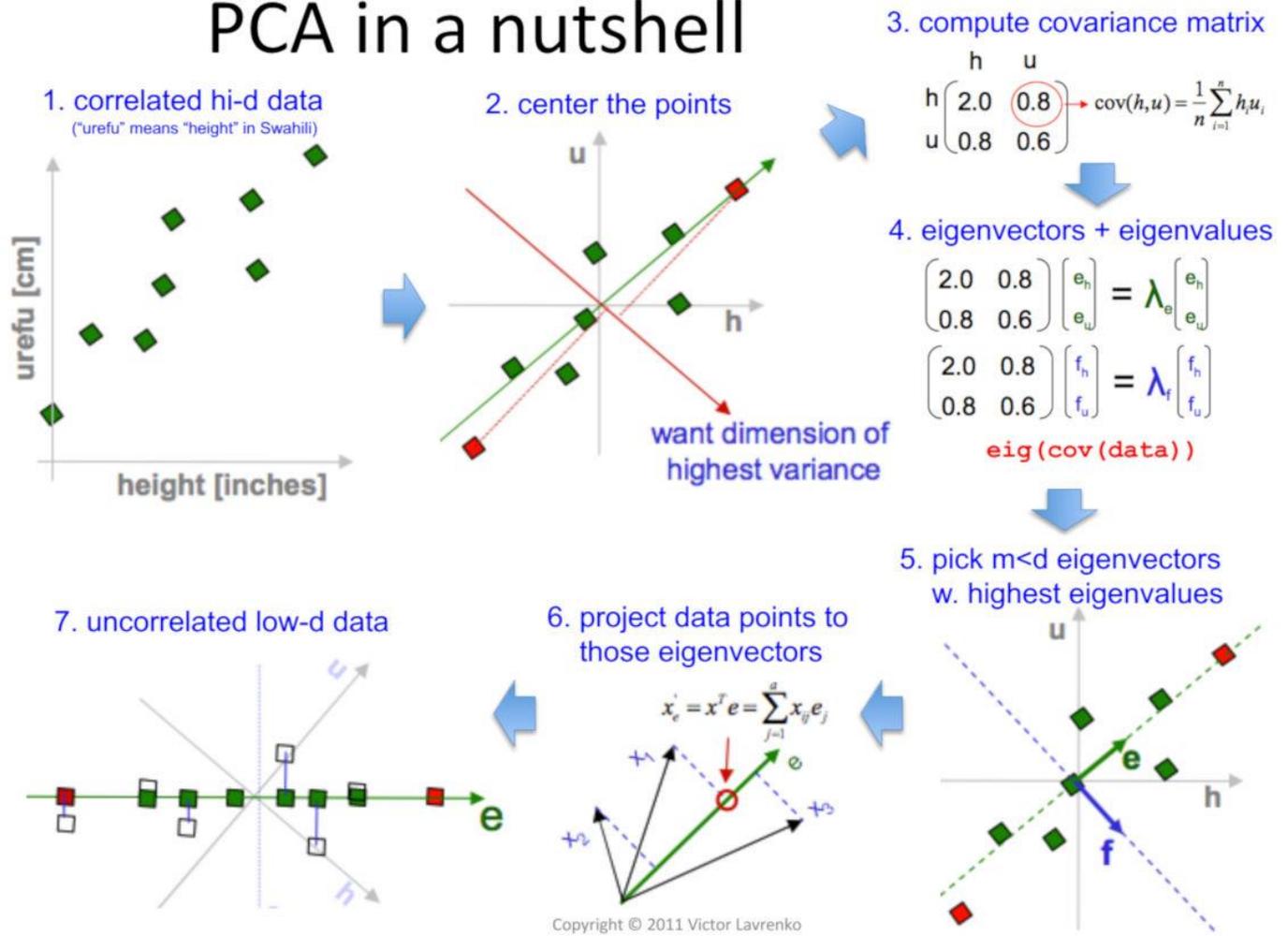
*Die Länge der Eigenvektoren entspricht den Eigenwerten.*

*Je größer der Eigenwert, desto 'bedeutsamer' Eigenvektor.*



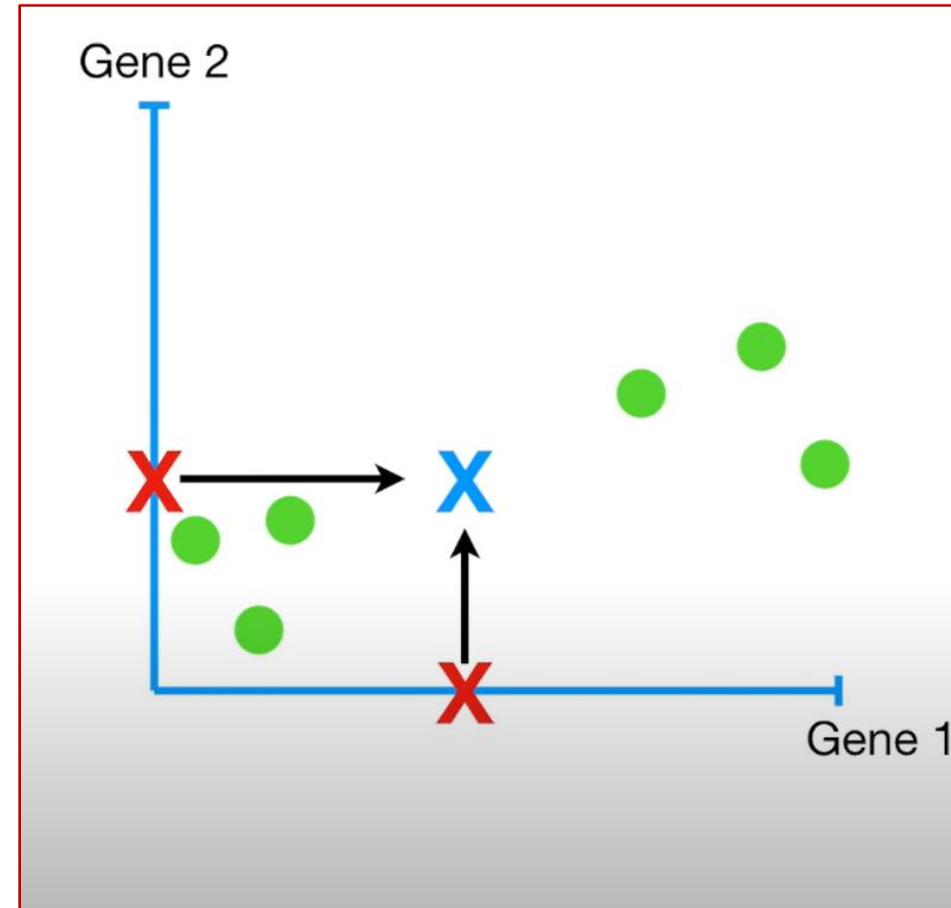
# Wie funktioniert PCA?

## PCA in a nutshell



Josh Starmer on Statquest explaining the main idea of pca:

Range of  
4:05 – 9:25  
(1.25x speed)



<https://www.youtube.com/watch?v=FgakZw6K1QQ&t=1086s>





<https://chat.openai.com/chat>

## Fragen wir das Orakel!



erkläre das prinzip der hauptkomponentenanalyse



Die Hauptkomponentenanalyse (PCA) ist eine statistische Methode zur Untersuchung von multivariaten Daten. Sie dient dazu, die Struktur der Variabilität in den Daten zu erfassen, indem die Dimensionen (Komponenten) identifiziert werden, die die größte Varianz aufweisen. Diese Komponenten können dann verwendet werden, um die Daten zu projizieren, um die wichtigsten Merkmale der Daten zu visualisieren oder um die Daten zu reduzieren. Im Grunde genommen, ist PCA eine Methode, um die wichtigsten Eigenschaften in einer Multivariablen Datenmenge zu finden und diese in eine neue Darstellung mit weniger Dimensionen umzuwandeln.



worin liegt der vorteil einer dimensionsreduktion bei der pca



Ein Vorteil der Dimensionenreduktion bei der PCA besteht darin, dass sie die Daten kompakter und übersichtlicher darstellen kann. Indem man die Dimensionen reduziert, auf denen die meiste Varianz liegt, kann man die wichtigsten Merkmale der Daten erhalten und gleichzeitig diejenigen Dimensionen entfernen, die wenig zur Struktur der Daten beitragen.



## Fragen wir das Orakel!

---



<https://chat.openai.com/chat>

TH

schreibe in python einen algorithmus für eine hauptkomponentenanalyse



Ein Algorithmus für die Hauptkomponentenanalyse (PCA) kann in Python wie folgt implementiert werden:





```
import numpy as np

def PCA(X, n_components):
    # Zentriere die Daten
    X = X - np.mean(X, axis=0)

    # Berechne die Kovarianzmatrix
    cov_mat = np.cov(X.T)

    # Berechne die Eigenvektoren und Eigenwerte der Kovarianzmatrix
    eig_vals, eig_vecs = np.linalg.eig(cov_mat)

    # Sortiere die Eigenvektoren und Eigenwerte in absteigender Reihenfolge nach
    # Eigenwerten
    eig_pairs = [(np.abs(eig_vals[i]), eig_vecs[:,i]) for i in
range(len(eig_vals))]
    eig_pairs.sort(key=lambda x: x[0], reverse=True)

    # Wähle die ersten n_components Eigenvektoren aus
    W = np.hstack([eig_pairs[i][1].reshape(X.shape[1], 1) for i in
range(n_components)])

    # Transformiere die Daten mit den ausgewählten Eigenvektoren
    Y = X @ W

    return Y
```

In diesem Algorithmus werden zunächst die Daten zentriert, indem der Mittelwert jeder Spalte subtrahiert wird.

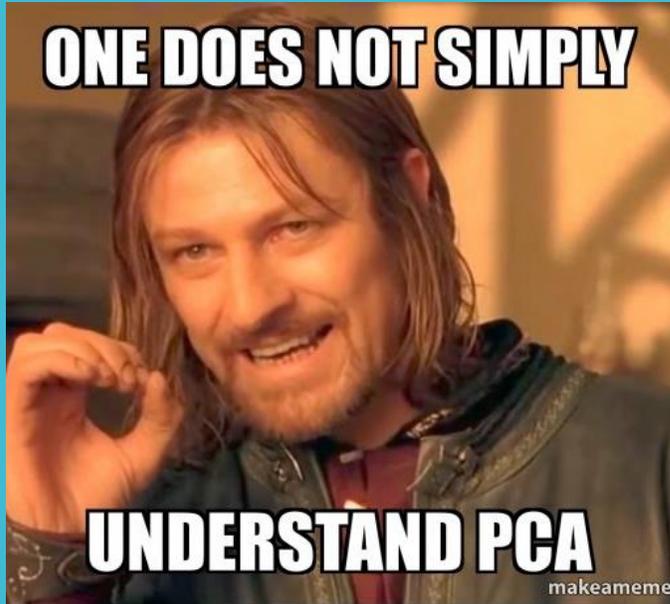
Anschließend wird die Kovarianzmatrix berechnet und die Eigenvektoren und Eigenwerte mit der Funktion `np.linalg.eig()` bestimmt.

Die Eigenvektoren und Eigenwerte werden dann in absteigender Reihenfolge nach Eigenwerten sortiert.

Die ersten `n_components` Eigenvektoren werden ausgewählt und verwendet, um die Daten zu transformieren.

Das resultierende Array `Y` enthält die Hauptkomponenten der ursprünglichen Daten.

Und jetzt? Selber machen!



- Schulnoten
- Diabetes
- Weinsorten
- Iris (wie immer...)
- Selbst gemalte Daten
- Pizza!!!