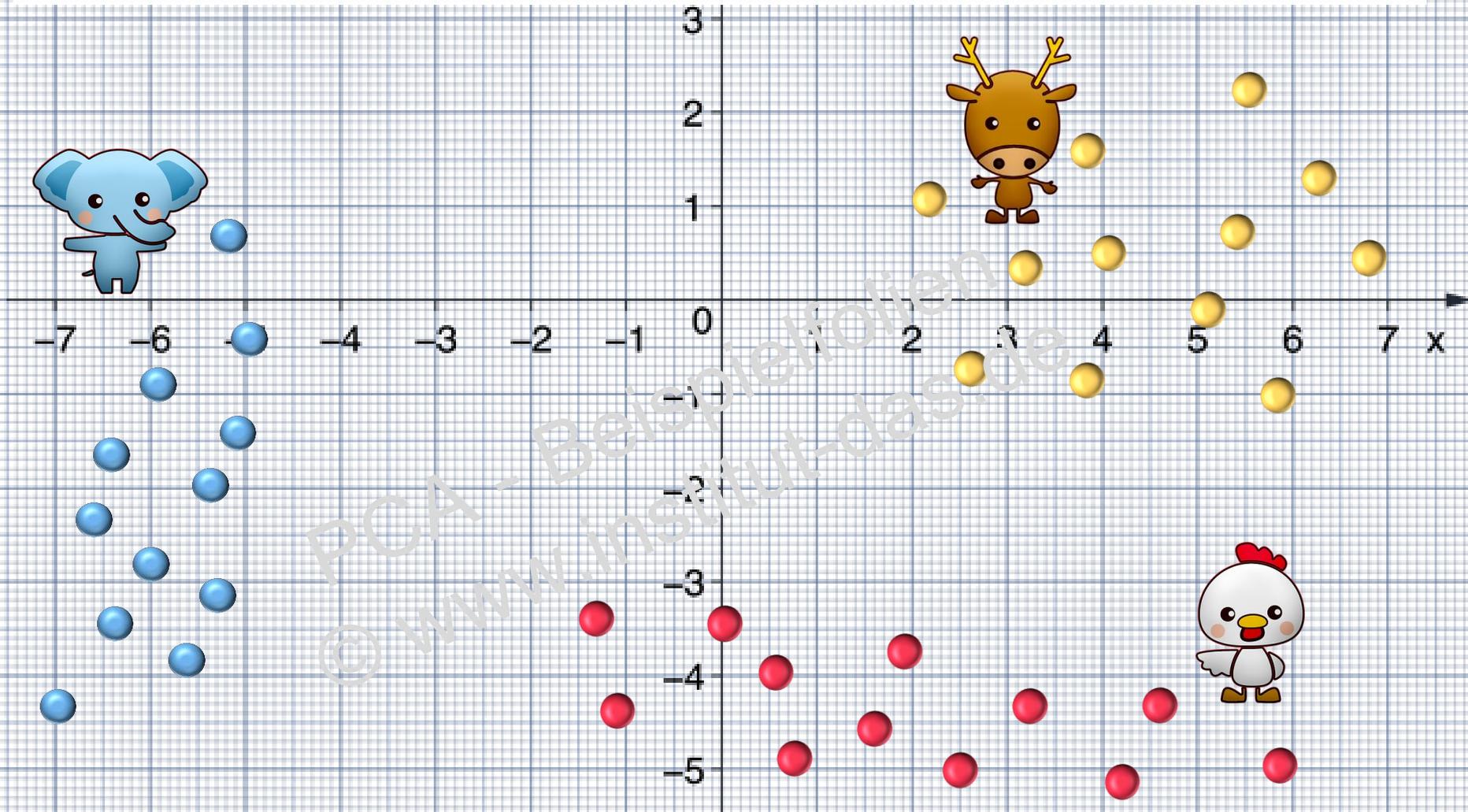


Data Sciences

Prof. Dr. Martin Przewloka
Institut für Digitale Assistenzsysteme e.V.
martin.przewloka@institut-das.de

PCA

Fokus: Python Framework



Inhalte

1. PCA – Principal Component Analysis - was ist das?

2. Etwas Theorie

3. PCA in der Praxis

4. Lernkontrolle

Nicht notwendig:
Programmierkenntnisse!

Hilfreich:

Installierte Python-Umgebung (idealerweise Python 3)

PCA - was ist das?

PCA = Principal Component Analysis = Hauptkomponentenanalyse gehört zur Kategorie der multivariaten, qualitativen Verfahren und ist Teil des „unsupervised Learnings“

Ziel ist es, aus einem hochdimensionalen Datensatz die Komplexität zu reduzieren, indem die wesentlichen, niederdimensionalen Einflussfaktoren identifiziert werden, die sogenannten Hauptkomponenten. Mathematisch handelt es sich um ein Projektionsverfahren.

Ein erstes Beispiel:

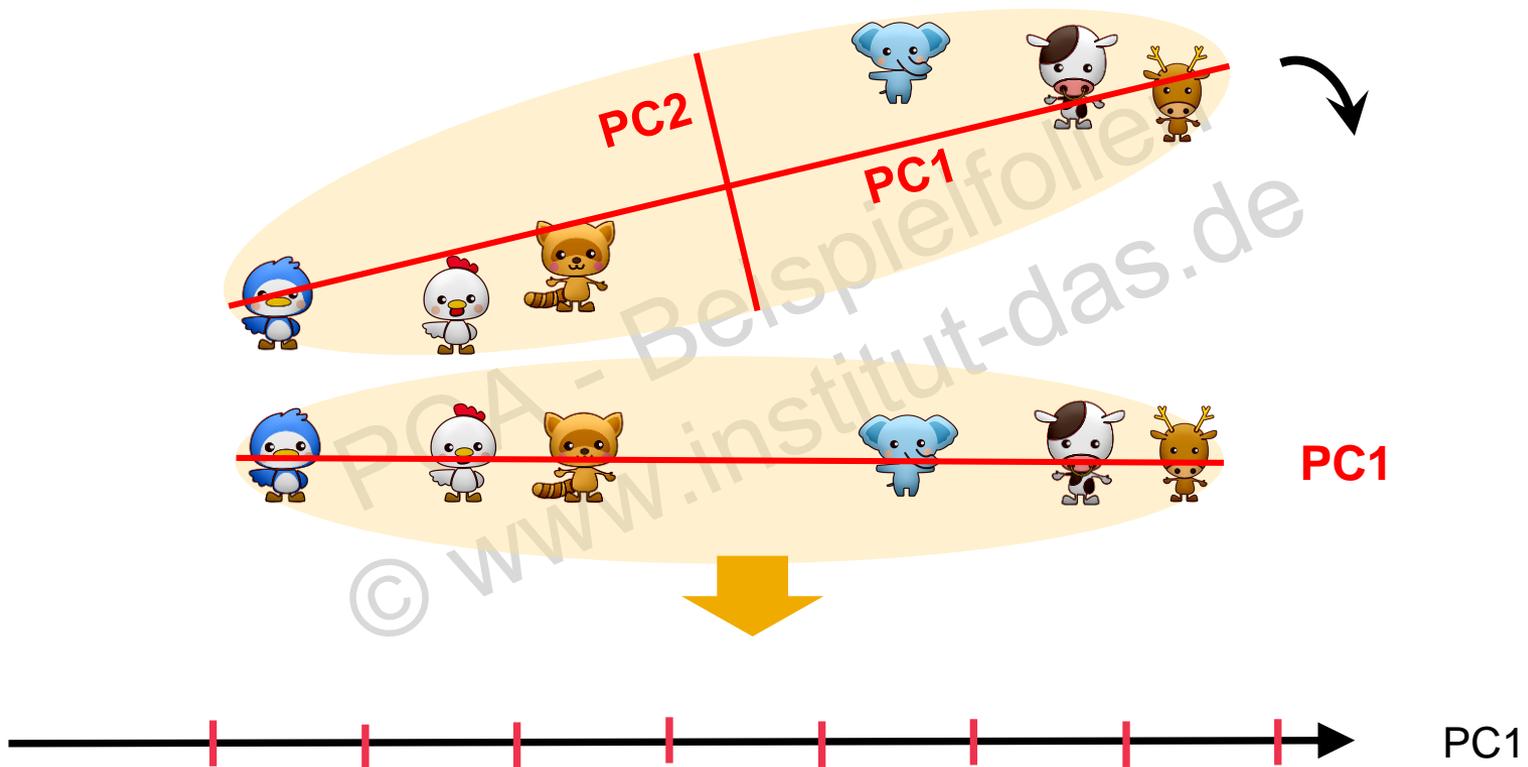
Wir alle schauen heute noch 2-dimensional Fernsehen/Filme. 3D-Verfahren haben nur ein Nischendasein. Die 3. Dimension liefert keine entscheidende Information mehr. Sie ist zwar nicht irrelevant, die Informationsübertragung für diesen Anwendungsfall kann aber auf 2 Dimensionen reduziert werden.

Konkrete Beispielfragestellung:

Eine Maschine wird von 100 Sensoren überwacht. Welche Sensoren sind für die Vorhersage der Restlebensdauer am relevantesten?

Etwas Theorie - einfach dargestellt

Nun reduzieren wir die Dimensionen, indem wir PC2 unberücksichtigt lassen, und PC1 auf die Grundachse abbilden (entspricht einer Rechts-Drehung)



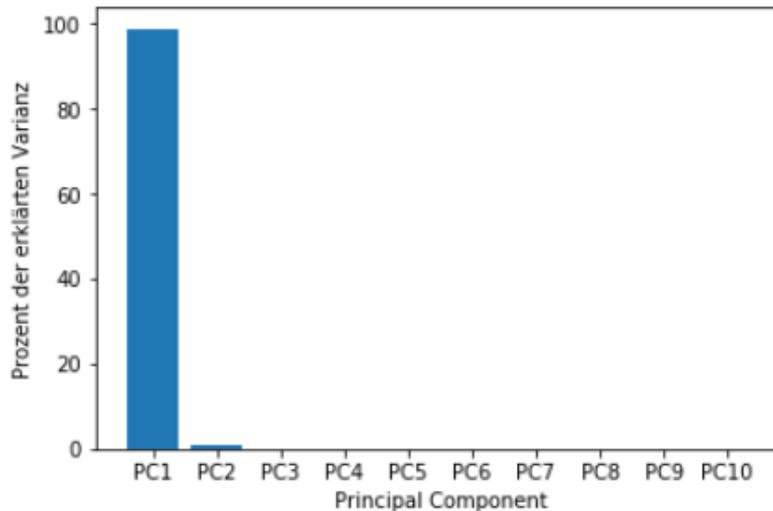
PCA - Praxis

10.) Graphische Ausgabe des Ergebnisses in einem Balkendiagramm

```
plt.bar(x=range(1, len(per_var) +1), height = per_var,  
        tick_label = labels)  
plt.xlabel('Principal Component')  
plt.ylabel('Prozent der erklärten Varianz')
```

```
In [71]: plt.bar(x=range(1, len(per_var)+1), height = per_var, tick_label = labels)  
plt.ylabel('Prozent der erklärten Varianz')  
plt.xlabel('Principal Component')
```

```
Out[71]: Text(0.5, 0, 'Principal Component')
```

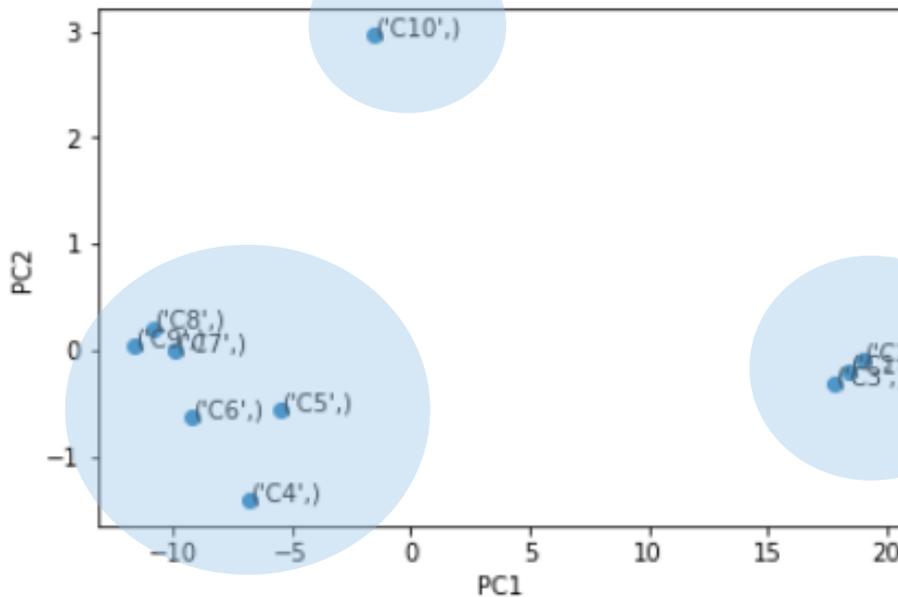


PCA - Praxis

12.) Graphische Ausgabe des Ergebnisses in einem Scatterplot

```
In [88]: plt.scatter(pca_df.PC1, pca_df.PC2)
         for sample in pca_df.index:
             plt.annotate(sample, (pca_df.PC1.loc[sample], pca_df.PC2.loc[sample]))
         plt.ylabel('PC2')
         plt.xlabel('PC1')
```

Out[88]: Text(0.5, 0, 'PC1')



C1 = ,Happiness.Score'
C2 = ,Whisker.high'
C3 = ,Whisker.low'

**Interpretieren
Sie
das Ergebnis!**