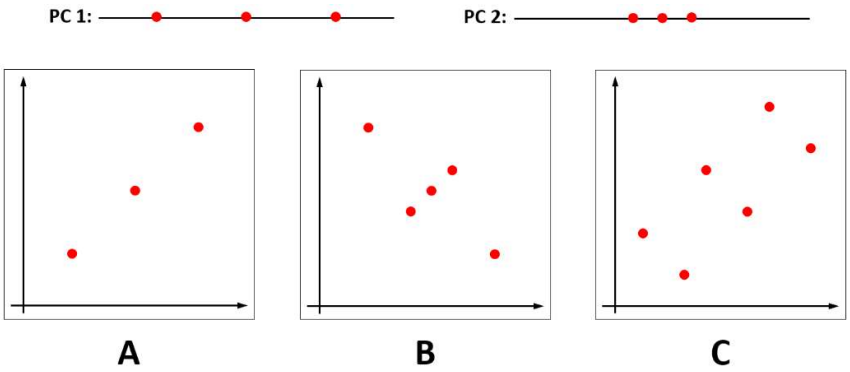
Introduction to Data Science HS 2021

Aufgabenblatt 3: Hauptkomponentenzerlegung

Die Bearbeitung der Aufgaben ist freiwillig; es erfolgt keine Bewertung.

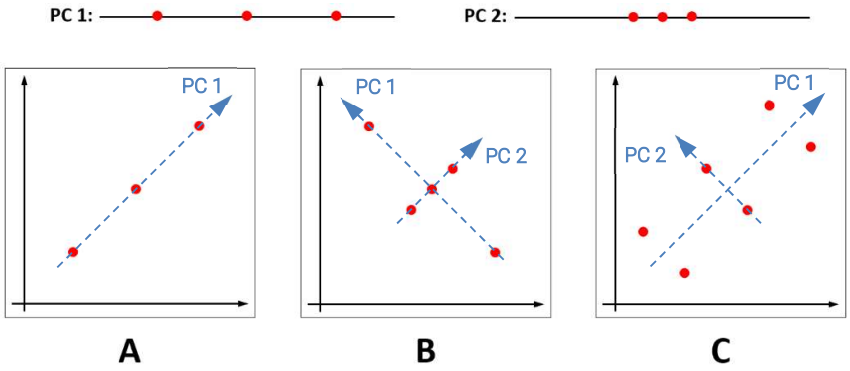
Aufgabe 1

Gegeben sind folgende Datensätze A, B und C sowie die ersten beiden (nicht massstabsgetreuen) Komponenten PC 1 und PC 2 einer Hauptkomponentenzerlegung. Welche der drei Datensätze liefern mittels *Principal Component Analysis* beide Hauptkomponenten?



– Lösung –

Nur Datensatz B kann beide Hauptkomponenten PC 1 und PC 2 liefern.



Aufgabe 2

Gegeben ist folgende prozentuale Verteilung der Varianz auf die Hauptkomponenten (PC) eines 8-dimensionalen Datensatzes gemäss *Principal Component Analysis*. Zeichnen Sie die

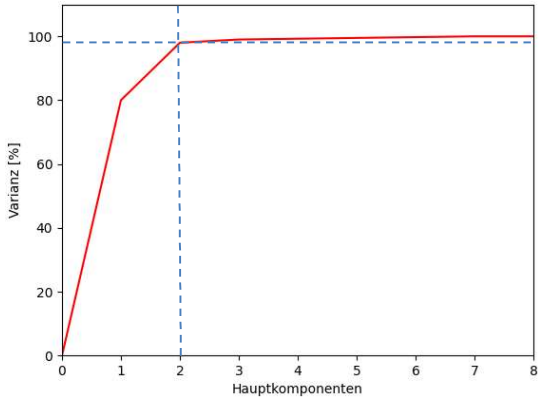
Verteilung der Varianz in Python als kumulative Summe (d.h. aufsummiert) in ein Diagramm (*x*-Achse: Hauptkomponenten; *y*-Achse: Varianz [%]). Überlegen Sie ausserdem, auf wie viele Dimensionen sich der Datensatz praktisch ohne Verlust reduzieren lässt.

PC	1	2	3	4	5	6	7	8
Varianz (in [%])	80	18	1	0.25	0.25	0.25	0.25	0

– Lösung –

Vorbemerkung: Damit die Kurve als kumulative Summe bei 0.0 beginnt, muss am Anfang noch ein zusätzlicher Punkt 0.0 (quasi als Hauptkomponente 0) hinzugefügt werden.

```
>>> varianz = [0., 80., 18., 1., 0.25, 0.25, 0.25, 0.25, 0.]
>>> x = np.linspace(0, 8, 9)
>>> plt.xlabel('Hauptkomponenten')
>>> plt.ylabel('Varianz [%]')
>>> plt.axis([0,8,0,110])
>>> plt.plot(x, np.cumsum(varianz), 'r-')
>>> plt.show()
```



Da 99 % der Varianz auf die ersten beiden Hauptkomponenten verteilt ist, lässt sich der Datensatz praktisch verlustfrei auf zwei Dimensionen reduzieren.

Aufgabe 3

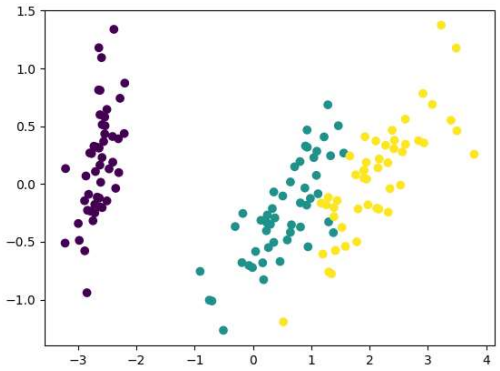
Der Iris-Datensatz aus dem Paket **`sklearn.datasets`** beinhaltet 150 Schwertlilien, die nach vier unterschiedlichen Merkmalen klassifiziert wurden. Laden Sie zunächst den Datensatz **`iris`**, der u.a. die Teile **`iris.data`** (Grösse 150×4) mit den Merkmalswerten sowie **`iris.target`** (Grösse 150×1) mit den korrekten Klassifikationen enthält.

```
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
```

a) Führen Sie mittels PCA eine Reduktion des vierdimensionalen Merkmalraumes (**`iris.data`**) auf zwei Dimensionen durch und visualisieren Sie das Ergebnis als Streudiagramm (Scatter Plot).

– Lösung –

```
>>> from sklearn.decomposition import PCA
>>> model_pca = PCA(2)
>>> data_proj = model_pca.fit_transform(iris.data)
>>> plt.scatter(data_proj[:,0], data_proj[:,1], c=iris.target)
>>> plt.show()
```



b) Wie verteilt sich die Varianz des dimensionsreduzierten Datensatzes auf die beiden Hauptkomponenten? Ist das ein brauchbares Ergebnis?

– Lösung –

```
>>> model_pca.explained_variance_ratio_
0.92461872, 0.05306648
```

In Summe wurden 97.8 % der Varianz erhalten, d.h. der durch die Dimensionsreduktion einhergehende Verlust ist minimal.

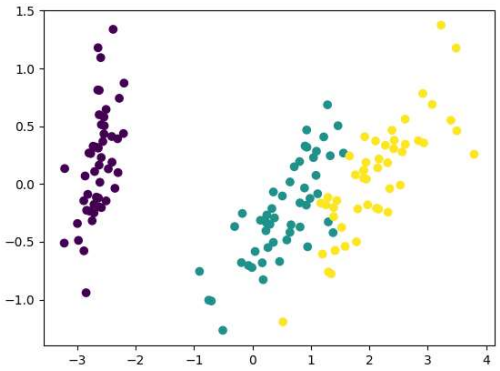
c) Bestimmen Sie mittels *k*-Mitten-Algorithmus die zugehörigen Cluster der dimensionsreduzierten Daten und visualisieren Sie das Ergebnis als Streudiagramm. Wie gut stimmt die vorhergesagte Klassifikation – rein optisch betrachtet – mit dem tatsächlichen Ergebnis aus a) überein?

– Lösung –

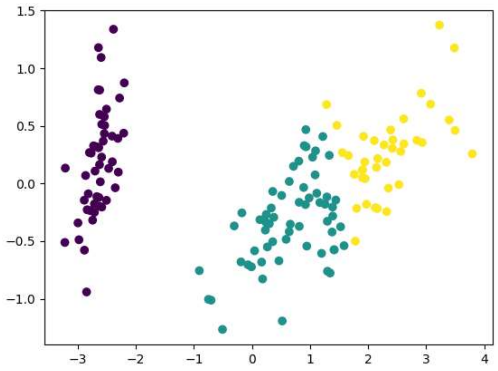
Vorbemerkung: Das Streudiagramm aus Aufgabe a) enthält offenkundig $k = 3$ Cluster.

```
>>> from sklearn.cluster import KMeans
>>> model_km = KMeans(3)
```

```
>>> y_pred = model_km.fit_predict(data_proj)
>>> plt.scatter(data_proj[:,0], data_proj[:,1], c=y_pred)
>>> plt.show()
```



tatsächliche Klassifikation aus a)



vorhergesagte Klassifikation ($k = 3$)

Rein optisch betrachtet, ist die Klassifikation durchaus als sehr gut zu bewerten (im rechten Bild wurde die Farbskala für eine bessere Vergleichbarkeit angepasst), auch wenn einzelne Punkte erkennbar falsch klassifiziert wurden.