

# Einführung in die künstliche Intelligenz

## Aufgabe “Unsupervised Learning”

Prof. Dr. A. del Pino

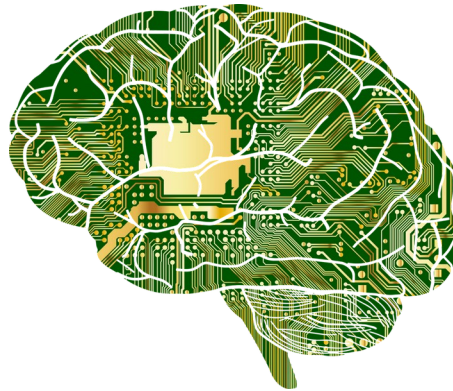


Bild: [https://en.wikipedia.org/wiki/Machine\\_learning#/media/File:Anatomy-1751201\\_1280.png](https://en.wikipedia.org/wiki/Machine_learning#/media/File:Anatomy-1751201_1280.png)

# Aufgabe “Unsupervised Learning”

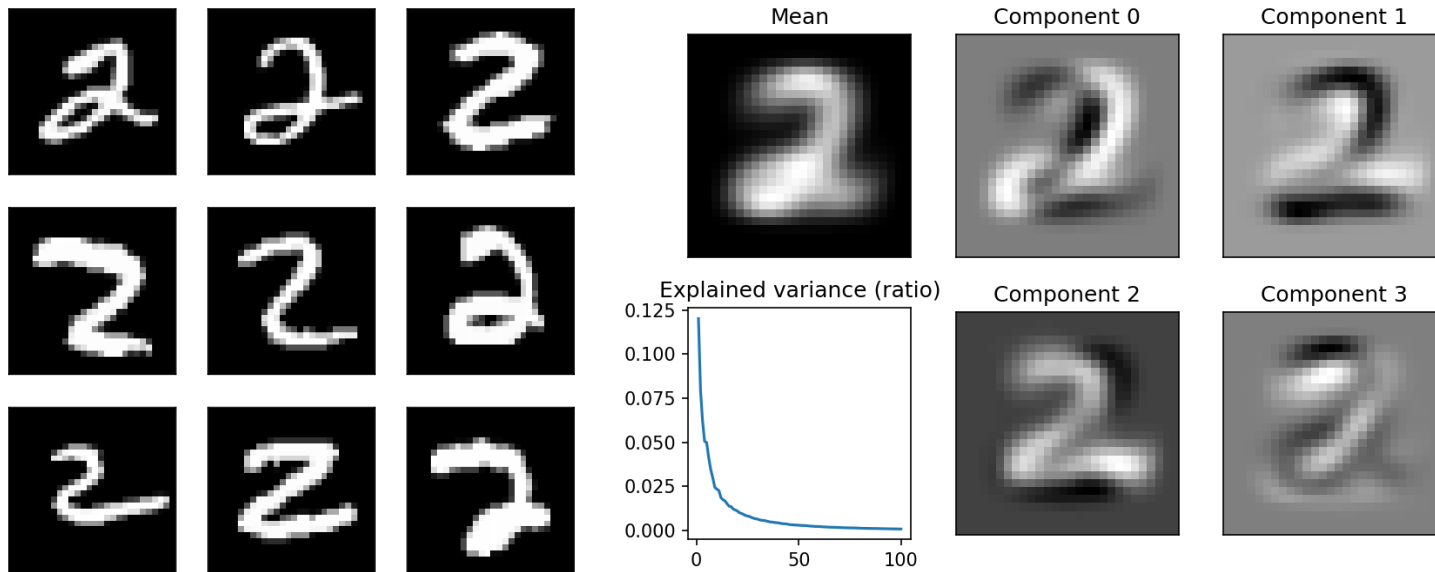
In dieser Aufgabe machen Sie sich mit der Hauptkomponentenanalyse (PCA) in scikit-learn vertraut.

- Beantworten Sie die Fragen zu Mini-Test “Unsupervised Learning” auf Moodle
- Bearbeiten Sie die Aufgaben der nächsten Folien in einem Jupyter Notebook “pca.ipynb”
- Laden Sie wie gewohnt Ihr Jupyter Notebook in Moodle hoch

# PCA – Dimensionsreduktion

Nutzen Sie die Hauptkomponentenanalyse, um die Dimension des MNIST-Datensatzes zu reduzieren:

1. Filtern Sie aus dem MNIST-Datensatz die Bilder, die eine „2“ enthalten heraus.
2. Visualisieren Sie die ersten 9 dieser Bilder.
3. Berechnen und visualisieren Sie den Mittelwert aller „2“-Bilder.
4. Wenden Sie die PCA auf den Datensatz an und visualisieren die erklärte Varianz (als Funktion der Anzahl der Komponenten).
5. Visualisieren Sie die ersten 4 Hauptkomponenten.



# PCA – Klassifikation

Nutzen Sie die Hauptkomponentenanalyse zum Preprocessing vor der Klassifizierung:

6. Projizieren Sie den MNIST Datensatz auf die ersten Hauptkomponenten der „2“ und nutzen Sie anschließend -nearest neighbors (mit) zur binären Klassifikation „2“ vs. „nicht-2“
7. Visualisieren Sie die Genauigkeit/Accuracy (als Funktion der Anzahl der Komponenten)
8. Für wie viele Hauptkomponenten erhalten Sie das beste Ergebnis?
9. Wie passt das Ergebnis mit dem Graphen zur erklärten Varianz zusammen?

# PCA – Datengenerierung

Nutzen Sie die Hauptkomponentenanalyse, um neue Bilder zu generieren:

10. Generieren Sie Faktoren mit einer ähnlichen Verteilung wie die Projektionen und rekonstruieren Sie daraus neue „2“-Bilder.
11. Visualisieren Sie die neu erstellten Bilder.

```
model = PCA(n_components=4)
X_proj = model.fit_transform(X)
X_gen = (np.random.randn(20, 4) * np.std(X_proj, axis=0, keepdims=True)
        + np.mean(X_proj, axis=0, keepdims=True))
X_reconstructed = model.inverse_transform(X_gen)
```

# PCA – Datengenerierung

Erzeugte Bilder:



4 Hauptkomponenten



20 Hauptkomponenten