

1163150 - Übung 1

Name, Vorname: _____

Matrikelnummer: _____

- Abgabetermin der Übung ist der 2. Mai 2018
- Elektronische Abgaben erfolgen grundsätzlich über die Moodle-Plattform.
- Handgeschriebene Lösungsaufgaben können ins Fach eingeworfen werden (WH C Etage 2)
- Für jeden Tag nach Abgabefrist werden 5 Punkte der Maximalpunktzahl abgezogen
- Über alle Übungen hinweg besitzen Sie 3 Bonustage, die Sie für eine verspätete Abgabe ohne Punktabzug verwenden können

Aufgabe:	1	2	Summe:
Punkte:	15	5	20
Ergebnis:			

1. Lineare Algebra

Gegeben sind die folgenden Matrizen A und B sowie die Vektoren \vec{x} und \vec{y} :

$$A = \begin{pmatrix} 4 & 4 & 5 \\ 2 & 1 & 7 \\ 4 & 8 & 3 \end{pmatrix}, B = \begin{pmatrix} 1 & 6 \\ 3 & 1 \\ 5 & 2 \end{pmatrix}, C = \begin{pmatrix} 1 & 4 & 4 \\ 3 & 1 & 2 \\ 6 & 7 & 1 \end{pmatrix}, \vec{x} = \begin{pmatrix} 9 \\ 5 \\ 7 \end{pmatrix}, \vec{y} = \begin{pmatrix} 3 \\ 1 \\ 5 \end{pmatrix}$$

- (a) (1 Punkt) Berechnen Sie das Skalarprodukt $\vec{x} * \vec{y}$. Notieren Sie den vollständigen Rechenweg.
- (b) (0 Punkte) Berechnen Sie das Produkt $\vec{x} * \vec{y}^T$. Notieren Sie den vollständigen Rechenweg.
- (c) (2 Punkte) Berechnen Sie das Produkt $A * B$. Notieren Sie den vollständigen Rechenweg.
- (d) (0 Punkte) Berechnen Sie das Produkt $B * A$. Notieren Sie den vollständigen Rechenweg.
- (e) (0 Punkte) Wie können Sie die Matrizen A, B modifizieren, sodass die Werte des Produkts $B * A$ den Werten des Produkts $A * B$ entsprechen. Notieren Sie den vollständigen Rechenweg.
- (f) (0 Punkte) Berechnen Sie das Frobenius-Produkt $\langle A, C \rangle_F$. Notieren Sie den vollständigen Rechenweg.
- (g) (2 Punkte) Berechnen Sie das Hadamard-Produkt $A \circ C$. Notieren Sie den vollständigen Rechenweg.
- (h) (10 Punkte) Implementieren Sie das Jupyter-Notebook »linear_algebra.ipynb«.

2. k-Nearest-Neighbor (kNN)

- (a) (0 Punkte) Implementieren Sie das Jupyter-Notebook »k_nearest_neighbor_total.ipynb«.
- (b) (5 Punkte) Implementieren Sie die Kreuzvalidierung im Jupyter-Notebook »k_nearest_neighbor_cross.ipynb« sowie die vektorisierte Distanzberechnung im Skript »k_nearest_neighbor_cross.py«