1 Preface

1.1 Standardimporte

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

2 Supervised Learning

2.1 Lineare Regression

Optimale Anpassung einer Geraden an eine gegebene Menge an Punkten, d.h. für eine Funktion

$$h_\Theta(x) = \Theta_0 + \Theta_1 x_1 + \cdots + \Theta_n x_n$$

soll der Parametervektor Θ gefunden werden (mit Θ_0 als Konstante), der die Summe der quadrierten Abweichungen der Funktionswerte $h_\Theta(x)$ von den tatsächlichen Werten y minimiert (Methode der kleinsten Quadrate):

$$\min_\Theta L(D, f) = \sum_{i=1}^{m} (f(x^{(i)}) - y^{(i)})^2$$
$$\min_\Theta L(D, \Theta) = \|X_D \Theta - y_D\|^2$$

wobei X_D eine Matrix mit den Eingabedaten (zzgl. führende 1-Spalte) ist und y_D der Vektor der tatsächlichen Werte ist:

$$X_D = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix}, \qquad y_D = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```

2.2 Logistische Regression

Test

2.3 Support Vector Machines

Test

2.4 K-Nearest Neighbours

Test

2.5 Bayes-Klassifikator

Test

2.6 Entscheidungsbäume

Test

3 Unsupervised Learning

3.1 K-Means Clustering

Test

3.2 Hierarchisches Clustering

Test

3.3 Assoziationsregeln

Test

3.4 Anomalieerkennung

Test

3.5 Hauptkomponentenanalyse/Principal Component Analysis (PCA)

Test

4 Reinforcement Learning

4.1 Markov-Entscheidungsprozesse

Test

4.2 Passives Reinforcement-Learning

Test

4.3 Aktives Reinforcement-Learning

Test

5 Deep Learning

5.1 Künstliche Neuronale Netze

Test

5.2 Convolutional Neutral Networks

Test

5.3 Recurrent Neutral Networks

Test

5.4 Recurrent Neutral Networks

Test