Data Mining und Maschinelles Lernen

Prof. Kristian Kersting
Zhongjie Yu
Johannes Czech

TECHNISCHE UNIVERSITÄT DARMSTADT

Sommersemester 2020
30. Juni 2020
Bonusübungsblatt 1

Die **Abgabefrist** dieser Bonusübung ist am **15.07.2020** um **23:59 Uhr**.

Die Bonusübung wird am **16.07.2020** um **13:30 Uhr** besprochen.

Aufgabe	1	2	3	4	5	6	7	8
Maximal Punktzahl	16	4	3	4	15	19	6	7
Erreichte Punktzahl								

Gruppe A	**Nachnahme**	**Vorname**	**Matrikelnummer**
1	Nowak	Patrick	2455200
2	Göller	Nicolas	
3	Helm	Falco	?
4	Cakaj	Rinor	-

Benötigte Dateien
Alle benötigten Datensätze und Skriptvorlagen finden Sie in unserem Moodle-Kurs:
https://moodle.informatik.tu-darmstadt.de/course/view.php?id=937

Gruppeneinteilung
Bearbeiten Sie diese Übung in Dreier- oder Vierergruppen. Es steht Ihnen frei die Gruppen selbst zu bilden. Nutzen Sie hierfür die Gruppeneinteilung und das Forum in Moodle. Sehen Sie bitte aufgrund der aktuellen Coronakrise davon ab, sich lokal zu treffen und nutzen Sie stattdessen digitale Kommunikationskanäle. Zur gemeinsamen Bearbeitung der Abgabe können sie beispielsweise https://overleaf.com nutzen.

Theoretische Aufgaben
Bei theoretischen Übungsaufgaben, sind wir Ihnen sehr dankbar, wenn Sie diese in LaTeX formatieren und als PDF einreichen. Nutzen Sie hierfür die LaTeX-Vorlage und die vorgesehene Blöcke:

```
\begin{solution}
% Geben sie hier ihre Antwort an.
\end{solution}
```

Geben Sie dabei ihre Gruppenmitglieder und Gruppennummber in der Datei `group_members.tex` an.

Wenn Sie mit LaTeX nicht ausreichend vertraut sind, können Sie auch einen hochauflösenden Scan einer handgeschriebenen Lösung einreichen. Bitte schreiben Sie ordentlich und leserlich.

Programmieraufgaben
Bei Aufgaben, die mit einem **</>** versehen sind, handelt es sich um Programmieraufgaben. Bearbeiten Sie in diesem Fall die vorgegebene Programmiervorlage. Verwenden Sie bevorzugt **Python 3.7**, da wir diese Version zum Testen ihrer Lösung benutzen. Benennen Sie die Funktionsdateien nicht um und ändern Sie die angegebenen Funktionssignaturen nicht. Wenn Sie das Gefühl haben, dass es ein Fehler bei den Zuweisungen, fragen Sie uns auf Moodle.

Bonusübungsblatt 1, Data Mining und Maschinelles Lernen
Nachname, Vorname: _____ Matrikelnummer: ⌴⌴⌴⌴⌴⌴⌴

Formalien zur Abgabe
Bitte laden Sie Ihre Lösungen in der entsprechenden Rubrik auf Moodle hoch. Sie müssen **nur eine Lösung pro Gruppe** einreichen. Wenn Sie keinen Zugang zu Moodle haben, setzen Sie sich bitte so schnell wie möglich mit uns in Verbindung. Laden Sie alle Ihre Lösungsdateien (die PDF-Abgabe und .py-Dateien) als eine einzige .zip-Datei hoch. Bitte beachten Sie, dass wir keine anderen als die angegebenen Dateiformate akzeptieren. **Laden Sie den gegeben Datensatz zur Programmieraufgabe nich in der Abgabe hoch.** Nutzen Sie folgende Namensgebung:

```
dmml_bonus1_group<groupid>.zip
├── dmml_bonus1.pdf
├── 02_dt_classification.py
├── 03_dt_regression.py
└── 04_random_forest.py
```

Verspätete Abgaben
Verspätete Abgaben werden akzeptiert, aber für jeden Tag, an dem die Abgabefirst überschritten wird, werden 25 % der insgesamt erreichbaren Punkte abgezogen. Nachdem die Übung offiziell besprochen wurde, können Sie die Aufgabe nicht mehr einreichen.

Bewertungsfaktoren
Die Bewertung dieser Übung hängt von den folgenden Faktoren ab:

- Richtigkeit der Antwort
- Klarheit der Präsentation der Ergebnisse
- Schreibstil

Wenn Sie bei einer Aufgabe nicht weiterkommen, versuchen Sie zu erklären warum und beschreiben Sie die Probleme, auf die Sie gestoßen sind, da Sie dafür Teilpunkte erhalten können.

Umgang mit Plagiaten
Sie dürfen gerne kursbezogene Themen in der Vorlesung oder in unseren Moodle Foren diskutieren. Sie sollten allerdings keine Lösungen mit anderen Gruppen teilen, und alles, was Sie einreichen, muss Ihre eigene Arbeit sein. Es ist Ihnen auch nicht gestattet, Material aus dem Internet zu kopieren. Sie sind verpflichtet, jede Informationsquelle, die Sie zur Lösung der Übungsaufgabe verwendet haben (d.h. andere Materialien als die Vorlesungsmaterialien), anzuerkennen. Zitierungen haben keinen Einfluss auf Ihre Note. Nicht anerkennen einer Quelle, die Sie verwendet haben, ist dagegen ein klarer Verstoß gegen die akademische Ethik. Beachten Sie, dass die Universität sehr ernst mit Plagiaten umgeht.

Aufgabe 1.1: Entscheidungsbäume - ID3 Algorithmus (16)

Die folgende Tabelle zeigt die Entscheidung, ob Baseball gespielt wird, basierend auf vier Wetterattributen.

Tabelle 1: Trainingsdatensatz, ob Baseball gespielt wird basierend auf der Wetterlage.

Tag	Ausblick (A)	Temperatur (T)	Luftfeuchtigkeit (L)	Wind (W)	Spielt Baseball (B)
T1	Sonnig	Warm	Hoch	Schwach	Nein
T2	Sonnig	Warm	Hoch	Stark	Nein
T3	Bewölkung	Warm	Hoch	Schwach	Ja
T4	Regen	Mild	Hoch	Schwach	Ja
T5	Regen	Kühl	Normal	Schwach	Ja
T6	Regen	Kühl	Normal	Stark	Nein
T7	Bewölkung	Kühl	Normal	Stark	Ja
T8	Sonnig	Mild	Hoch	Schwach	Nein
T9	Sonnig	Kühl	Normal	Schwach	Ja
T10	Regen	Mild	Normal	Schwach	Ja
T11	Sonnig	Mild	Normal	Stark	Ja
T12	Bewölkung	Mild	Hoch	Stark	Ja
T13	Bewölkung	Warm	Normal	Schwach	Ja
T14	Regen	Mild	Hoch	Stark	Nein

Tabelle 2: Vorhersage-Datensatz, ob Baseball gespielt wird.

Tag	Ausblick (A)	Temperatur (T)	Luftfeuchtigkeit (L)	Wind (W)	Spielt Baseball (B)
T15	Sonnig	Mild	Hoch	Schwach	?
T16	Bewölkung	Mild	Normal	Schwach	?
T17	Regen	Kühl	Normal	Stark	?

Die Aufgabe ist es folgende Frage zu beantworten: *Unter welchen Bedingungen wir Baseball gespielt?*

1.1a) ID3 Algorithmus (10 Punkte)

Erstellen Sie den Entscheidungsbaum mittels des ID3 Algorithmus. Berechnen Sie dabei die **Entropie** und den **Informationsgewinn** (engl. *gain*) der Attribut-Selektion für jeden Schritt.

Lösungsvorschlag:

Wir filtern im ersten Schritt nach Attributen und Schätzen p_+ und p_- für jede Ausprägung des Attributs durch Zählen. Es ergibt sich

Tabelle 3: Attribut: Ausblick

Ausprägung	Anzahl	davon +	davon -	Entropy
Sonnig	5	2	3	0.971
Bewölkt	4	4	0	0
Regen	5	3	2	0.971

Tabelle 4: Attribut: Temperatur

Ausprägung	**Anzahl**	**davon +**	**davon -**	**Entropy**
Warm	4	2	2	1
Mild	6	4	2	0.918
Kühl	4	3	1	0.811

Tabelle 5: Attribut: Luftfeuchtigkeit

Ausprägung	**Anzahl**	**davon +**	**davon -**	**Entropy**
Hoch	7	3	4	0.985
Normal	7	6	1	0.591

Tabelle 6: Attribut: Wind

Ausprägung	**Anzahl**	**davon +**	**davon -**	**Entropy**
Stark	6	3	3	1
Schwach	8	6	2	0.811

Pro Ausprägung setzen wir nun

$$p_+ = \frac{\text{davon +}}{\text{Anzahl}} \quad \text{und} \quad p_- = \frac{\text{davon -}}{\text{Anzahl}}$$

und berechnen die Entropien mittels $I(p_+, p_-) = (-p_+ \cdot \log p_+) + (-p_- \cdot \log p_-)$. Wobei der Logarithmus zur Basis 2 benutzt wurde. Die Ergebnisse haben wir an die Tabelle angefügt. Letztlich finden wir für jedes Attribut Att. den Informationsgehalt mittels

$$\text{Information(Att.)} = -\sum_{\text{Ausprägung A von Att.}} \frac{\text{Anzahl(A)}}{\text{ges}} \cdot \text{Entropy(A)}$$

Es ergibt sich (ges=14),

Tabelle 7: Informationsgehalt

Attribut	**Information**
Ausblick	-0.694
Temperatur	-0.911
Luftfeuchtigkeit	-0.788
Wind	-0.892

, sodass wir uns als erstes Kriterium für das Attribut Ausblick entscheiden. Bei Bewölkung sind bereits alle Tage in einer Kategorie (+), weshalb wir hier ein Blatt mit einem + erstellen. Betrachten wir nun also alle Tage an denen es Sonnig ist und erstellen hier rekursiv einen Teilbaum. Wir fassen nochmal alle Daten, welche wir in diesem Knoten gegeben haben zusammen:

Tabelle 8: Trainingsdatensatz, nur Sonnig

Tag	Ausblick (A)	Temperatur (T)	Luftfeuchtigkeit (L)	Wind (W)	Spielt Baseball (B)
T1	Sonnig	Warm	Hoch	Schwach	Nein
T2	Sonnig	Warm	Hoch	Stark	Nein
T8	Sonnig	Mild	Hoch	Schwach	Nein
T9	Sonnig	Kühl	Normal	Schwach	Ja
T11	Sonnig	Mild	Normal	Stark	Ja

Wir zählen, so wie oben und erhalten:

Tabelle 9: nur Sonnig, Attribut: Temperatur

Ausprägung	Anzahl	davon +	davon -	Entropy
Warm	2	0	2	0
Mild	2	1	1	1
Kühl	1	1	0	0

Tabelle 10: nur Sonnig, Attribut: Luftfeuchtigkeit

Ausprägung	Anzahl	davon +	davon -	Entropy
Hoch	3	0	3	0
Normal	2	2	0	0

Tabelle 11: nur Sonnig, Attribut: Wind

Ausprägung	Anzahl	davon +	davon -	Entropy
Schwach	3	1	2	0.918
Stark	2	1	1	1

Man sieht direkt, dass Attribut L auf den Trainingsdaten eine optimalen Informationsgewinn hat. Dies verifiziert man durch Berechnen der Informationsgehalte(ges=5)):

Tabelle 12: Informationsgehalt, nur Sonnig

Attribut	Information
Temperatur	-0.4
Luftfeuchtigkeit	0
Wind	-0.95

Unter dem Sonnig-Ast erzeugen wir also einen Knoten 'Luftfeuchtigkeit' mit 2 Blättern. Ast 'hoch' führt zu - und Ast 'normal' zu +.
Nun müssen wir noch den Teilbaum finden, der an den Ast 'Regen' anhängt. Filtern wir also nach diesem Attribut und zählen dann wieder, so erhalten wir:

Tabelle 13: Trainingsdatensatz, nur Regen

Tag	Ausblick (A)	Temperatur (T)	Luftfeuchtigkeit (L)	Wind (W)	Spielt Baseball (B)
T4	Regen	Mild	Hoch	Schwach	Ja
T5	Regen	Kühl	Normal	Schwach	Ja
T6	Regen	Kühl	Normal	Stark	Nein
T10	Regen	Mild	Normal	Schwach	Ja
T14	Regen	Mild	Hoch	Stark	Nein

Tabelle 14: nur Regen, Attribut: Temperatur

Ausprägung	Anzahl	davon +	davon -	Entropy
Mild	3	2	1	0.918
Kühl	2	1	1	1

Tabelle 15: nur Regen, Attribut: Luftfeuchtigkeit

Ausprägung	Anzahl	davon +	davon -	Entropy
Hoch	2	1	1	1
Normal	3	2	1	0.918

Tabelle 16: nur Regen, Attribut: Wind

Ausprägung	Anzahl	davon +	davon -	Entropy
Schwach	3	3	0	0
Stark	2	0	2	0

Wir sehen wieder direkt die perfekte Klassifizierung beim Attribut Wind(Informationswert=0). Die anderen beiden haben jeweils einen Informationswert von -0.951. Wir erstellen hier also einen Knoten 'Wind' mit 2 Blättern '+' für Ausprägung 'schwach' und '-' für Ausprägung 'stark'. Es ergibt sich insgesamt der folgende Entscheidungsbaum:

1.1b) Visualisierung (3 Punkte)

Erstellen Sie eine Visualisierung (Plot oder eingefügte Zeichnung) des Entscheidungsbaumes aus Aufgabenteil a).

Lösungsvorschlag:

1.1c) Vorhersage (3 Punkte)

Geben Sie anhand ihres Entscheidungsbaumes eine Vorhersage für die Tage 15 bis 17 aus Tabelle 2, ob Baseball gespielt wird.

Lösungsvorschlag:

Aufgabe 1.2: </> Entscheidungsbäume - Klassifikation (4)

Im Institut Institut für Produktionstechnik und Umformmaschinen, kurz PtU[1], der TU Darmstadt gibt es die Aufgabe eines Scherschneideverfahrens. Dabei wird mit Hilfe eines Stempels ein Loch in einen metallischen Werkstoff gestanzt, siehe Abbildung 1. Es gibt zwei Werkstoffe im Versuch: `CuSn6` und `16MnCr5`. Die Dicke des Materials beträgt entweder 0.4 mm oder 0.5 mm. Die Geschwindigkeit des Stempels liegt in den drei Stufen 100, 200 und 300 vor.

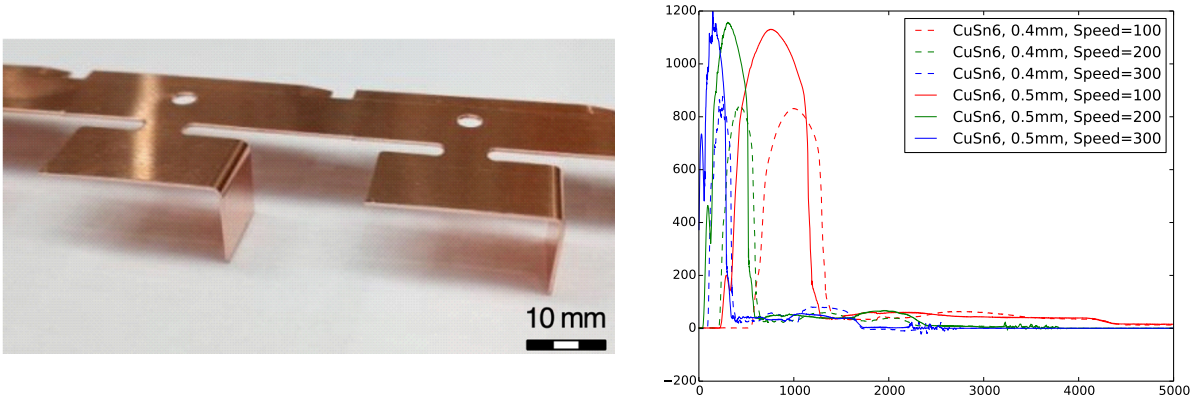


Abbildung 1: Links: Veranschaulichung des PtU Prozess. Rechts: Beispiel des Kraftsignals.

Es gibt mehrere Sensoren, die den Status des Stanzen messen:

- Kraft Sensor / Force sensor (Fz)
- Verschiebungssensor / Displacement sensor (w)
- Beschleunigungssensor / Acceleration sensor (acc)

Im Experiment messen die Sensoren den Status des Stanzen von Beginn bis Ende jedes Prozesses. Die Zeitreihendaten von jedem Sensor werden als 1D-Vektor dargestellt. Das Interessante an dieser Aufgabe ist, dass der Status des Stanzen von der Art und Dicke des Materials abhängt. Andererseits ist es uns möglich, aus den Zeitreiheninformationen von den Sensoren auf die Art und Dicke des Materials und die Geschwindigkeit des Stanzen zu schließen.

- **`Filename_Fz_raw.csv`**: Enthält Daten aus dem Kraftsensor. Jeder Zeilenvektor repräsentiert das Kraftsignal. Die Anzahl der Zeilen in der Datei beträgt 2787, was der Anzahl der Experimente im Datensatz entspricht.
- **`Filename_Speed.csv`**: Enthält die Geschwindigkeit des Schlags der 2787 Proben.
- **`Filename_thickness.csv`**: Enthält die Materialstärke der 2787 Proben.

In dieser Übung verwenden wir die Daten des Kraftsensors, um die Proben nach der Geschwindigkeit des Stempels zu klassifizieren.

1.2a) </> `02_dt_classification.py` (3 Punkte)

Laden Sie die Daten des Kraftsensor aus `Filename_Fz_raw.csv` und die Geschwindigkeit des Stanzen aus `Filename_Speed.csv` und vervollständigen Sie den Code in `02_dt_classification.py`:

</> Trainieren Sie einen Entscheidungsbaum zur Klassifikation in der Methode `fit_dt_classifier()` mithilfe von `sklearn` unter der Verwendung der Standardparameter.

</> Ermitteln Sie die Testgenauigkeit in der Methode `get_test_accuracy()`.

[1] `https://www.ptu.tu-darmstadt.de/`

</> Plotten Sie den resultierenden Entscheidungsbaum in `export_tree_plot()`. Sie können dabei die Funktion `tree.export_graphviz()` verwenden.

1.2b) Visualisierung (1 Punkt)

Zeigen Sie die erstelle Visualisierung des Entscheidungsbaumes zur Klassifikation aus Unteraufgabe a).

Lösungsvorschlag:

Aufgabe 1.3: Entscheidungsbäume - Regression (3)

Ein Entscheidungsbaum zur Regression kann auf den PtU Datensatz (s. Abb. 1) angewendet werden, um auf die Dicke des Materials zu schließen, welche ein kontinuierlicher Wert ist.

1.3a) **</>** `03_dt_regression.py` (2 Punkte)

Laden Sie die Daten des Kraftsensor aus `Filename_Fz_raw.csv` und die Materialstärken aus `Filename_thickness.csv` und vervollständigen Sie den Code in `03_dt_regression.py`:

</> Trainieren Sie einen Entscheidungsbaum zur Regression in `fit_dt_regressor()`.

</> Berechnen Sie den mittleren quadratischen Fehler für den Testdatensatz in der Funktion `get_test_mse()`.

1.3b) Visualisierung (1 Punkt)

Zeigen Sie die erstelle Visualisierung des Entscheidungsbaumes zur Regression aus Unteraufgabe a).

Lösungsvorschlag:

Aufgabe 1.4: </> Zufallswälder (4)

In dieser Aufgabe verwenden wir den PtU Datensatz (s. Abb. 1), um mit einem Zufallswald (engl. Random Forest) die Geschwindigkeit des Einschlags aus den Zeitreihen zu klassifizieren.

1.4a) **</>** `04_random_forest.py` (2 Punkte)

Laden Sie die Daten des Kraftsensor aus `Filename_Fz_raw.csv` und die Geschwindigkeit des Stanzen aus `Filename_Speed.csv` und vervollständigen Sie den Code in `04_random_forest.py`:

</> Trainieren Sie einen Zufallswald aus 100 Entscheidungsstümpfen in der Methode `fit_tree_stump_forest()`.

</> Trainieren Sie einen einzelnen Entscheidungsstumpf in der Methode `fit_tree_stump()`.

1.4b) Konfusionsmatrix (2 Punkte)

Geben Sie die Konfusionsmatrizen des Trainings- und Testdatensatzes mit absoluten Werten für den Zufallswald an.

Lösungsvorschlag:

Aufgabe 1.5: AdaBoost (15)

In dieser Aufgabe werden Sie AdaBoost auf die gegebenen Trainingsbeispiele aus der Tabelle 17 anwenden.

Tabelle 17: Datensatz mit zwei Merkmalen und zwei Zielklassen.

$\mathbf{x_1}$	$\mathbf{x_2}$	**Klasse**
1	5	+
2	2	+
5	8	+
6	10	+
8	7	+
3	1	-
4	6	-
7	4	-
9	3	-
10	9	-

Entscheidungsstümpfe mit ganzzahligem Schwellwert (z.B. $\mathbf{x_1} \leq T \Rightarrow +$ oder $\mathbf{x_1} > T \Rightarrow +$) sollen als Basis-Lerner verwendet werden. Der Basis-Lerner minimiert die Summe der Gewichtungen der falsch klassifizierten Beispiele aus allen möglichen Aufteilungen. Für ein Unentschieden wählen Sie die erste gefundene Übereinstimmung, beginnend mit Entscheidungsstümpfen für $\mathbf{x_1}$ und dann $\mathbf{x_2}$.

Verwenden Sie die Formel:

$$\alpha_i = \frac{1}{2} \log \left(\frac{1 - err_i}{err_i} \right) \tag{1}$$

zur Berechnung von α_i.

1.5a) Algorithmus (12 Punkte)

Zeigen Sie die Ausführung des Adaboost Algorithmus für die **ersten beiden** Iterationen. Geben Sie dabei die **Fehler** (Summe der Gewichtungen der falsch klassifizierten Beispiele) für die möglichen Entscheidungsgrenzen von 1 bis 10 an, sowie die **Gewichtung** jedes Datenpunktes vor und nach Normalisierung an.

Lösungsvorschlag:

1.5b) Gesamtmodell (3 Punkte)

Geben Sie das Gesamtmodell $f(x)$ nach zwei Iterationen an.

Lösungsvorschlag:

Aufgabe 1.6: Naïve Bayes (19)

In dieser Aufgabe verwenden wir wieder den Baseball-Datensatz (s. Tabelle 1 und 2) und einen Naïve Bayes Klassifikator, um zu entscheiden ob Baseball gespielt wird oder nicht.

1.6a) Formel für Merkmalsausprägung (4 Punkte)

Zeigen Sie die Formel für $P(B = Ja \mid Merkmal)$ und $P(B = Nein \mid Merkmal)$ für den gegebenen Datensatz.

Lösungsvorschlag:

1.6b) Wahrscheinlichkeiten (6 Punkte)

Bestimmen Sie angesichts der oben genannten Trainingsdaten alle Wahrscheinlichkeiten, die erforderlich sind, um den Naïve Bayes Klassifikator für beliebige Vorhersagen, ob Baseball gespielt wird, anzuwenden.

Lösungsvorschlag:

1.6c) Vorhersage (9 Punkte)

Treffen Sie Vorhersagen nach Naïve Bayes für die Tage 15 bis 17 aus Tabelle 2, ob Baseball gespielt wird. Geben Sie dabei den Rechenweg an.

Lösungsvorschlag:

Aufgabe 1.7: K-Means (6)

Folgender Datensatz besteht aus 8 Punkten:

$$\begin{aligned} x_1 &= (2,8), \quad x_2 = (2,5), \quad x_3 = (1,2), \quad x_4 = (5,8), \\ x_5 &= (7,3), \quad x_6 = (6,4), \quad x_7 = (8,4), \quad x_8 = (4,7). \end{aligned} \tag{2}$$

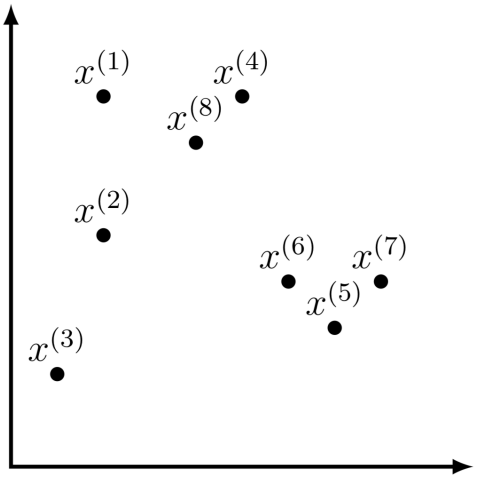


Abbildung 2: Visualisierung des K-Means Datensatzes

1.7a) K-Means Algorithmus (6 Punkte)

Benutzen Sie den K-Means Algorithmus mit der Euklidischen Distanz um diese 8 Datenpunkte in $K = 3$ Cluster einzuteilen. Nehmen Sie dabei an, dass die Clusterzentren mit den Punkten x_2, x_4 und x_8 initialisiert sind. Führen Sie zwei Iterationen des K-Means Algorithmus durch und geben Sie die Koordinaten der Zentroide der Cluster an.

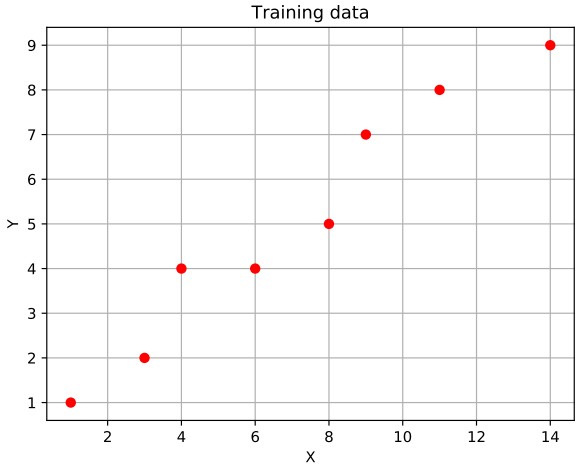
Lösungsvorschlag:

Aufgabe 1.8: Regressionsanalyse (7)

Gegeben sind folgende Datenpunkte:

x	1	3	4	6	8	9	11	14
y	1	2	4	4	5	7	8	9

Abbildung 3: Veranschaulichung der Datenpunkte zur linearen Regression.



Wir möchten eine Regression nach dem Prinzip der kleinsten Fehlrequadrate erstellen:

$$y = f(x) = \langle W, x \rangle + b\,. \tag{3}$$

Mit der Hilfe eines $(p+1)$-dimensionalen Vektors $\vec{x} = (1, x_1, \cdots, x_p)$ und $x \in \mathbb{R}^{1\times p}$, können wir b in dem Vektor W codieren:

$$y = f(x) = \left\langle W^{'}, \vec{x}^{T} \right\rangle, \tag{4}$$

wobei hier $W^{'} \in \mathbb{R}^{2\times 1}$ und $\vec{x} \in \mathbb{R}^{1\times 2}$.

1.8a) Herleitung (5 Punkte)

Zeigen Sie, dass das optimale $W^{'}$:

$$W^{'} = (\vec{X}^{T}\vec{X})^{-1}\vec{X}^{T}Y, \tag{5}$$

entspricht, wobei $\vec{X} \in \mathbb{R}^{n\times 2}$ und $Y \in \mathbb{R}^{n\times 1}$.

Lösungsvorschlag:

1.8b) Parameterbestimmung (2 Punkte)

Berechnen Sie W und b für den gegebenen Punktdatensatz. Die Inverse $(\vec{X}^{T}\vec{X})^{-1}$ muss dabei nicht manuell berechnet werden.

Bonusübungsblatt 1, Data Mining und Maschinelles Lernen
Nachname, Vorname: _____ Matrikelnummer: ⌊_⌋⌊_⌋⌊_⌋⌊_⌋⌊_⌋⌊_⌋⌊_⌋

Lösungsvorschlag: