Data Mining und Maschinelles Lernen

Prof. Kristian Kersting
Zhongjie Yu
Johannes Czech

TECHNISCHE UNIVERSITÄT DARMSTADT

Sommersemester 2020
15. Juli 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 [W]	Nachnahme	Vorname	Matrikelnummer
1	**Cui**	**Yi**	**2758172**
2	**He**	**Ruidi**	**2618034**
3	**Bao**	**Wenhua**	**2512664**

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.

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:

1.1b) Visualisierung (3 Punkte)

Erstellen Sie eine Visualsierung (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.

① erst Abzweigung:

Ausblick

$P(\text{Sonnig}) = \frac{5}{14}$ $P(\text{Bewölkung}) = \frac{4}{14}$ $P(\text{Regen}) = \frac{5}{14}$

Spiel Baseball	Sonnig	Bewölkung	Regen
Nein	3	0	2
Ja	2	4	3

Entropie: Sonnig $I_S(+,-) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5} = 0.971$

Bewölkung $I_B(+,-) = -\frac{4}{4}\log\frac{4}{4} = 0$

Regen $I_R(+,-) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.971$

Infogewinn: $G_{IA} = -\left[\frac{5}{14} I_S(+,-) + \frac{4}{14} I_B(+,-) + \frac{5}{14}\cdot I_R(+,-)\right]$

$$= -\left[\frac{5}{14}\left(-\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5}\right) + \frac{4}{14}\left(-\frac{0}{4}\log\frac{0}{4} - \frac{4}{4}\log\frac{4}{4}\right) + \frac{5}{14}\left(-\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5}\right)\right]$$

$= -0.693$

Temperture.

$P(\text{Warm}) = \frac{4}{14}$ $P(\text{Mild}) = \frac{6}{14}$ $P(\text{Kühl}) = \frac{4}{14}$

Spiel Baseball	Warm	Mild	Kühl
Nein	2	2	1
Ja	2	4	3

Entropie: Warm: $I_W(+,-) = -\frac{2}{4}\log\frac{2}{4} - \frac{2}{4}\log\frac{2}{4} = 1$

Mild: $I_M(+,-) = -\frac{2}{6}\log\frac{2}{6} - \frac{4}{6}\log\frac{4}{6} = 0.918$

Kühl: $I_K(+,-) = -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4} = 0.81$

Infogewinn $G_{IT} = -\left(\frac{4}{14} I_W(+,-) + \frac{6}{14} I_M(+,-) + \frac{4}{14} I_K(+,-)\right)$

$= -0.911$

Abbildung 1: task 1 a

Luftfeuchtigkeit.

$P(\text{Hoch}) = \frac{7}{14}$ $\quad$ $P(\text{Normal}) = \frac{7}{14}$.

Spiel Baseball	Hoch	Normal
Nein	4	1
Ja.	3	6

Entropie: Hoch $I_H(+,-) = -\frac{4}{7}\log\frac{4}{7} - \frac{3}{7}\log\frac{3}{7} = 0.985$

Normal $I_N(+,-) = -\frac{1}{7}\log\frac{1}{7} - \frac{6}{7}\log\frac{6}{7} = 0.591$

Infogewinn $G_L = -\left(\frac{7}{14} I_H(+,-) + \frac{7}{14} I_N(+,-)\right)$

$= -0.788$

Wind

$P(\text{Schwach}) = \frac{8}{14}$ $\quad$ $P(\text{Stark}) = \frac{6}{14}$

Spiel Baseball	Schwach	Stark
Nein	2	3
Ja.	6	3

Entropie Schwach $I_{Sc}(+,-) = -\frac{2}{8}\log\frac{2}{8} - \frac{6}{8}\log\frac{6}{8} = 0.81$

Stark $I_{St}(+,-) = -\frac{3}{6}\log\frac{3}{6} - \frac{3}{6}\log\frac{3}{6} = 1$

Infogewinn $G_W = -\left(\frac{8}{14} I_{Sc}(+,-) + \frac{6}{14} I_{St}(+,-)\right)$

$= -0.892$.

„Ausblick" hat der größt Informationsgewinn. deshalb wählt man „Ausblick" als erst Abzweigungsknote.

Abbildung 2: task 1 a

② Am Knoten "Sonnig":

Temperature.

		$P(\text{Warm}) = \frac{2}{5}$	$P(\text{mild}) = \frac{2}{5}$	$P(\text{kühl}) = \frac{1}{5}$
Spiel Baseball	Nein	2	1	1
	Ja.	0	1	0

Entropie: Warm: $I_W(+,-) = -\frac{2}{2}\log\frac{2}{2} = 0$

Mild: $I_M(+,-) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$

Kühl: $I_K(+,-) = -\frac{1}{1}\log\frac{1}{1} = 0$

Infogewinn $G^S_{\geq T} = -\left(\frac{2}{5} I_W(+,-) + \frac{2}{5} I_M(+,-) + \frac{1}{5} I_K(+,-)\right)$

$= -0.4$

Luftfeuchtigkeit.

		$P(\text{Hoch}) = \frac{3}{5}$	$P(\text{Normal}) = \frac{2}{5}$
Spiel Baseball	Nein	3	0
	Ja.	0	2

Entropie: Hoch $I_H(+,-) = -\frac{3}{3}\log\frac{3}{3} = 0$

Normal $I_N(+,-) = -\frac{2}{2}\log\frac{2}{2} = 0$

Infogewinn $G^S_{\geq L} = -\left(\frac{3}{5} I_H(+,-) + \frac{2}{5} I_N(+,-)\right)$

$= 0$

Abbildung 3: task 1 a

Wind

$P(\text{Schwach}) = \frac{3}{5}$ $P(\text{Stark}) = \frac{2}{5}$

Spiel Baseball	Schwach	Stark
Nein	2	1
Ja.	1	1

Entropie Schwach $I_{Sc}(+,-) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.918$

Stark $I_{St}(+,-) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$

Infogewinn $G^S_{SW} = -\left(\frac{3}{5} I_{Sc}(+,-) + \frac{2}{5} I_{St}(+,-)\right)$

$= -0.967$

„Luftfeuchtigkeit" besitzt der größt Informationsgewinn. deshalb wählt man man „Luftfeuchtigkeit" als die Abzweigungskante. Außerdem sucht man nicht tiefer weil der Informationsgewinn von „Luftfeuchtigkeit" hier null ist.

Abbildung 4: task 1 a

③ Am Knoten "Regen"

Temperture.

		$P(Warm) = 0$	$P(Mild) = \frac{3}{5}$	$P(Kühl) = \frac{2}{5}$
Spiel Baseball	Nein	0	1	1
	Ja	0	2	1

Entropie : Warm : $I_W(+,-) = 0$

Mild : $I_M(+,-) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.918.$

Kühl : $I_K(+,-) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$

Infogewinn $G^S_{2T} = -\left(0 \cdot I_W(+,-) + \frac{3}{5} I_M(+,-) + \frac{2}{5} I_K(+,-)\right)$

$= -0.951$

Luftfeuchtigkeit.

		$P(Hoch) = \frac{2}{5}$	$P(Normal) = \frac{3}{5}$
Spiel Baseball	Nein	1	1
	Ja	1	2

Entropie : Hoch $I_H(+,-) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$

Normal $I_N(+,-) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} \approx 0.918$

Infogewinn $G^S_{2L} = -\left(\frac{3}{5} I_H(+,-) + \frac{2}{5} I_N(+,-)\right)$

$= -0.967$

Abbildung 5: task 1 a

Luftfeuchtigkeit.

$P(\text{Hoch}) = \frac{2}{5}$ $\quad$ $P(\text{Normal}) = \frac{3}{5}$

Spiel Baseball	Hoch	Normal
Nein	1	1
Ja.	1	2

Entropie: Hoch $I_H(+,-) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$

Normal $I_N(+,-) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} \approx 0.918$

Infogewinn $G^S_{2L} = -\left(\frac{3}{5} I_H(+,-) + \frac{2}{5} I_N(+,-)\right)$

$= -0.967$

Wind

$P(\text{Schwach}) = \frac{3}{5}$ $\quad$ $P(\text{Stark}) = \frac{2}{5}$

Spiel Baseball	Schwach	Stark
Nein	0	2
Ja.	3	0

Entropie Schwach $I_{Sc}(+,-) = -\frac{3}{3}\log\frac{3}{3} = 0$

Stark $I_{St}(+,-) = -\frac{2}{2}\log\frac{2}{2} = 0$

Infogewinn $G^S_{2W} = -\left(\frac{3}{5} I_{Sc}(+,-) + \frac{2}{5} I_{St}(+,-)\right)$

$= 0$

"Wild" besitzt der größt Informationsgewinn. deshalb wählt man "Wild" als die Abzweigungsknote. Außerdem sucht man nicht tiefer weil der Informationsgewinn von "Wild" hier null ist.

Abbildung 6: task 1 a

1.1b) Decision Tree

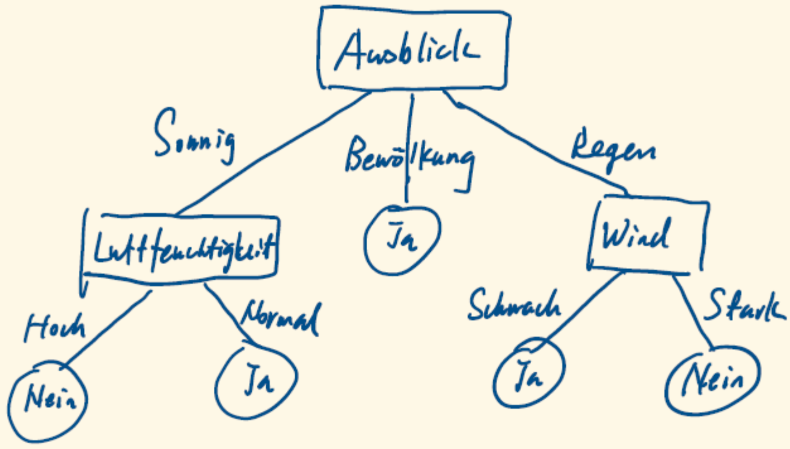


Abbildung 7: task 1 b

Lösungsvorschlag:

1.1c) Vorhersage: T_{15}: Nein
T_{16}: Ja
T_{17}: Nein

Abbildung 8: task 1 c

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 9. 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.

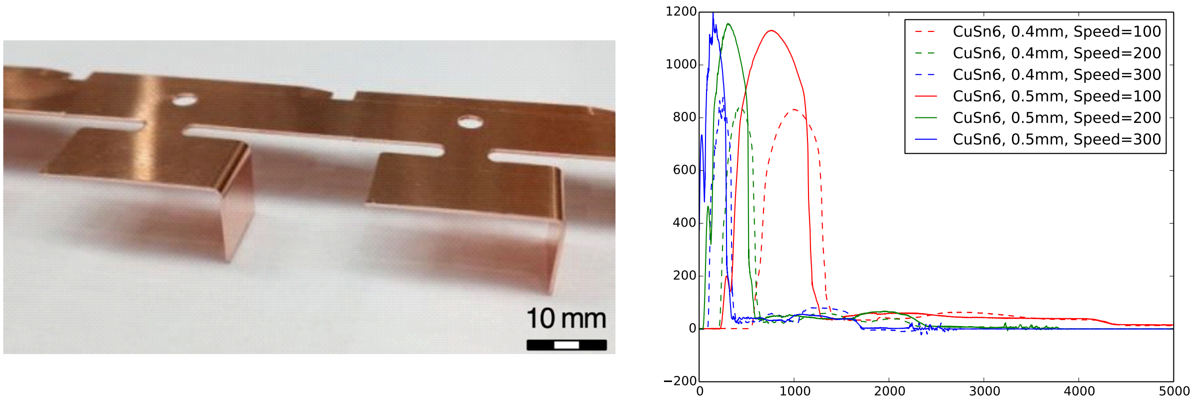


Abbildung 9: Links: Veranschaulichung des PtU Prozess. Rechts: Beispiel des Kraftsignals.

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:

Test Accuracy: 0.989247311827957

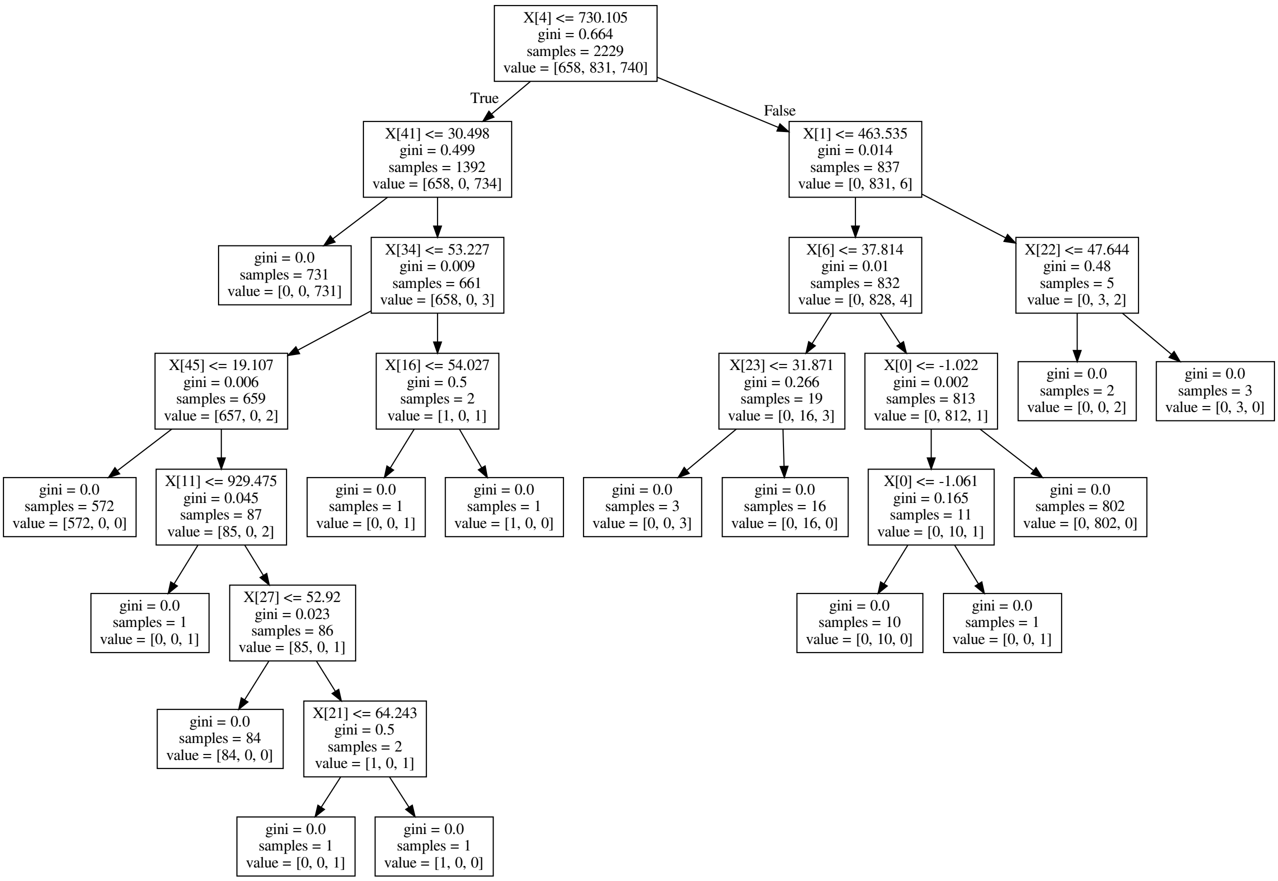


Abbildung 10: Visualisierung des Entscheidungsbaumes zur Klassifikation

```python
import graphviz
import numpy as np
from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split


def fit_dt_classifier(X_train: np.ndarray, y_train: np.ndarray) -> tree.DecisionTreeClassifier:
    """Creates and fits a decision tree classifier on the training data."""
    clf = tree.DecisionTreeClassifier()
    clf = clf.fit(X_train, y_train)
    return clf
```

```
def get_test_accuracy(clf, X_test: np.ndarray, y_test: np.ndarray) -> float:
    """Evaluates the test accuracy for a given classifier and test dataset."""
    pre = clf.predict(X_test)
    acc = accuracy_score(y_test, pre)
    return acc


def export_tree_plot(clf, filename: str) -> None:
    """Exports the tree plot for the given classifier as a pdf with given filename."""
    dot_data = tree.export_graphviz(clf, out_file=None)
    graph = graphviz.Source(dot_data)
    graph.render(filename)


def main():
    # for reproducibility
    np.random.seed(42)

    # load data
    X_data = np.loadtxt(open(’./Data/FileName_Fz_raw.csv’, ’r’), delimiter=",", skiprows=0)
    y_data = np.loadtxt(open(’./Data/FileName_Speed.csv’, ’r’), delimiter=",", skiprows=0)

    # down sample the data
    X_sample = X_data[:, ::100]
    print("X_data.shape:", X_data.shape)
    print("y_data.shape:", y_data.shape)

    # split training and test sets
    X_train, X_test, y_train, y_test = train_test_split(X_sample, y_data, test_size=0.2)

    # train
    clf = fit_dt_classifier(X_train, y_train)
    # predict
    acc = get_test_accuracy(clf, X_test, y_test)
    print(’Test Accuracy:’, acc)

    print("predict_proba:", clf.predict_proba(X_test))

    # plot tree
    export_tree_plot(clf, "classification_tree")


if __name__ == ’__main__’:
    main()
```

Aufgabe 1.3: Entscheidungsbäume - Regression (3)

Ein Entscheidungsbaum zur Regression kann auf den PtU Datensatz (s. Abb. 9) 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:

max depth: default, Test MSE: 3.5449382602670803e-05
max depth: 3, Test MSE: 4.83015794061216e-05

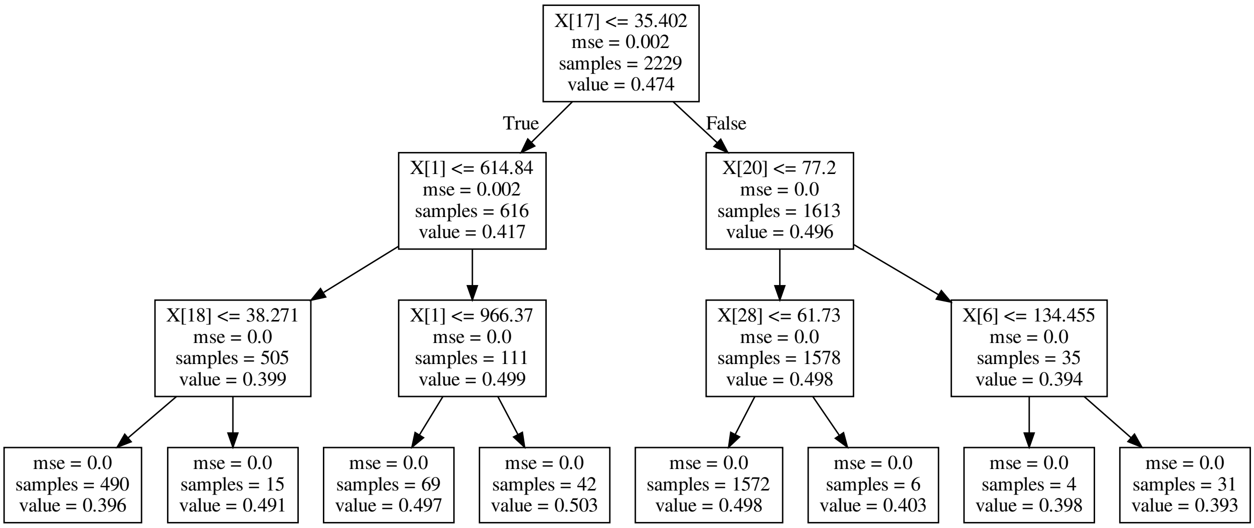


Abbildung 11: Visualisierung des Entscheidungsbaumes zur Regression

```python
from sklearn import tree
from sklearn.model_selection import train_test_split
import numpy as np
import graphviz
from sklearn.metrics import mean_squared_error


def fit_dt_regressor(X_train: np.ndarray, y_train: np.ndarray, max_depth=None) -> tree.DecisionTreeRegressor:
    """Creates and fits a regression tree on the training data."""
    clf = tree.DecisionTreeRegressor()
    clf = clf.fit(X_train, y_train)
    return clf


def get_test_mse(clf, X_test: np.ndarray, y_test: np.ndarray) -> float:
    """Evaluates the test mse for a given classifier and test dataset."""
    pre = clf.predict(X_test)
    mse = mean_squared_error(y_test, pre)
    return mse


def export_tree_plot(clf, filename: str) -> None:
    """Exports the tree plot for the given classifier as a pdf with given filename."""
    dot_data = tree.export_graphviz(clf, out_file=None)
    graph = graphviz.Source(dot_data)
    graph.render(filename)


```

```
def main():
    # for reproducibility
    np.random.seed(42)

    # load data
    X_data = np.loadtxt(open('./Data/FileName_Fz_raw.csv', 'r'), delimiter=",", skiprows=0)
    y_data = np.loadtxt(open('./Data/FileName_thickness.csv', 'r'), delimiter=",", skiprows=0)
    print("X_data.shape:", X_data.shape)
    print("y_data.shape:", y_data.shape)

    # down sample the data
    X_sample = X_data[:, ::100]

    # split training and test sets
    X_train, X_test, y_train, y_test = train_test_split(X_sample, y_data, test_size=0.2)

    # train
    clf = fit_dt_regressor(X_train, y_train)
    # predict % evaluate
    mse = get_test_mse(clf, X_test, y_test)
    print('Test MSE:', mse)

    # change max tree depth
    # train
    clf = fit_dt_regressor(X_train, y_train, max_depth=3)

    # predict & evaluate
    mse = get_test_mse(clf, X_test, y_test)
    print('Test MSE:', mse)

    # plot tree
    export_tree_plot(clf, "regression_tree_d3")


if __name__ == '__main__':
    main()
```

Aufgabe 1.4: </> Zufallswälder (4)

In dieser Aufgabe verwenden wir den PtU Datensatz (s. Abb. 9), 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:

Random Forest Test Accuracy: 0.992831541218638
Tree Stump Tree Test Accuracy: 0.9874551971326165

```
X_data.shape: (2787, 4999)
y_data.shape: (2787,)
y_pred: [100. 300. 300. 100. 200. 100. 200. 300. 200. 300.] ...
y_test: [100. 300. 300. 100. 200. 100. 200. 300. 200. 300.] ...
Train Confusion Matrix:
 [[658   0   0]
 [  0 831   0]
 [  0   0 740]]
Test Confusion Matrix:
 [[166   0   0]
 [  0 207   0]
 [  0   4 181]]
```

Abbildung 12: task 4 a

```
from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix


def fit_tree_stump_forest(X_train: np.ndarray, y_train: np.ndarray, n_estimators: int) ->
    RandomForestClassifier:
    """Creates and fits a random forrest of tree stumps on the training data."""
    clf = RandomForestClassifier(n_estimators=n_estimators)
    clf = clf.fit(X_train, y_train)
    return clf


def fit_tree_stump(X_train: np.ndarray, y_train: np.ndarray) -> tree.DecisionTreeClassifier:
    """Creates and fits a tree stump on the training data."""
    clf = tree.DecisionTreeClassifier()
    clf = clf.fit(X_train, y_train)
    return clf


def main():
    # for reproducibility
    np.random.seed(42)

    # load data
    X_data = np.loadtxt(open('./Data/FileName_Fz_raw.csv', 'r'), delimiter=",", skiprows=0)
    y_data = np.loadtxt(open('./Data/FileName_Speed.csv', 'r'), delimiter=",", skiprows=0)
    print("X_data.shape:", X_data.shape)
    print("y_data.shape:", y_data.shape)

    # down sample the data
    X_sample = X_data[:, ::100]

    # split training and test sets
    X_train, X_test, y_train, y_test = train_test_split(X_sample, y_data, test_size=0.2)

```

```
    # train
    clf = fit_tree_stump_forest(X_train, y_train, n_estimators=100)
    # predict
    y_pred = clf.predict(X_test)
    print("y_pred:", y_pred[:10], "...")
    print("y_test:", y_test[:10], '...')

    # show confusion matrix
    print("Train Confusion Matrix:\n", confusion_matrix(y_train, clf.predict(X_train)))
    print("Test Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

    # evaluate
    acc = accuracy_score(y_test, y_pred)
    print('Random Forest Test Accuracy:', acc)

    # compare with decision tree, max_depth=1, cannot classify 3-class data
    clf = fit_tree_stump(X_train, y_train)
    y_pred = clf.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print('Tree Stump Tree Test Accuracy:', acc)


if __name__ == '__main__':
    main()
```

Aufgabe 1.5: AdaBoost (15)

In dieser Aufgabe werden Sie AdaBoost auf die gegebenen Trainingsbeispiele aus der Tabelle 3 anwenden.

Tabelle 3: 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:

1.5 a) Step 1:

Anfangs sind alle Beispiele gleich wichtig

$W_i = 0.1 \qquad i = 1, \dots, 10$

Samples	1	2	3	4	5	6	7	8	9	10
X1	1	2	5	6	8	3	4	7	9	10
X2	5	2	8	10	7	1	6	4	3	9
Y	+	+	+	+	+	-	-	-	-	-
w	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

$H_1 = \begin{cases} + = 1 & x_1 \leq i \quad i = 1, \dots, 10 \\ - = -1 & x > i \end{cases} \qquad H_2 = \begin{cases} + = 1 & x_2 \leq i \quad i = 1, \dots, 10 \\ - = -1 & x_2 > i \end{cases}$

Abbildung 13: task 5 a

after H_1:

i	1	2	3	4	5	6	7	8	9	10
err	0.4	0.3	0.4	0.5	0.4	0.3	0.4	0.3	0.4	0.5
alpha	0.203	0.423	0.423	0.203	0.203	0.423	0.203	0.423	0.203	0

the weight values of H_1: $\qquad Z = 2 \times \sqrt{0.4 \times 0.6} = 0.980$

Vor normal: $D_{i+1} = D_i(i) \exp(-\alpha_i Y_i H_i(x_i))$

$= 0.1 \times \exp(0.423 \times 1 \times 1) = 0.081$ true classified dertas

$D_{i+1} = 0.1225$

Nach normal $D_1 = 0.083$ true classified dertas

$D = 0.125$ false classified dertas

if we choose $x_1 \leq 2$, after H_2

i	1	2	3	4	5	6	7	8	9	10
err	0.2	0.1	0.1	0.1	1	1	1	1	1	1
alpha	0.693	1.099	1.099	1.099						

if we chose $x_1 \leq 8$, after H_2

i	1	2	3	4	5	6	7	8	9	10
err	0.6	0.5	0.5	0.6	0.5	0.6	0.5	0.4	0.4	0.3
alpha	-0.203	0	0	-0.203	0	-0.203	0	0.203	0.203	0.423

So we choose $x_1 \leq 2$, $x_2 \leq 2$

Nach Normal $= \frac{0.125}{0.2} = 0.625 \rightarrow$ false classified dertas

Normal $= \frac{0.083}{1.8} = 0.046 \rightarrow$ true classified dertas

1.5b) $f_2(x) = 0.423\, H_1(x) + 1.099\, H_2(x)$

Abbildung 14: task 5 b

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:

1.6) a

$$P(\text{Klasse} \mid \text{Merkmal}) = \frac{\text{Likelihood} \cdot \text{Prior}}{P(M)}$$

ohne Normalisierung

$$\Rightarrow P(B = \text{Ja} \mid \text{Merkmal}) = \frac{P(\text{Merkmal} \mid B = \text{Ja}) \cdot P(B = \text{Ja})}{P(\text{Merkmal})}$$

$$P(B = \text{Nein} \mid \text{Merkmal}) = \frac{P(\text{Merkmal} \mid B = \text{Nein}) \cdot P(B = \text{Nein})}{P(\text{Merkmal})}$$

mit Normalisierung:

$$P(B = \text{Ja} \mid \text{Merkmal})^* = \frac{P(B = \text{Ja} \mid \text{Merkmal})}{P(B = \text{Ja} \mid \text{Merkmal}) + P(B = \text{Nein} \mid \text{Merkmal})}$$

$$P(B = \text{Nein} \mid \text{Merkmal})^* = \frac{P(B = \text{Nein} \mid \text{Merkmal})}{P(B = \text{Ja} \mid \text{Merkmal}) + P(B = \text{Nein} \mid \text{Merkmal})}$$

Abbildung 15: task 6 a

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.

1.6) b.

$P(\text{Sonnig}) = \frac{5}{14}$ $P(\text{Bewölkung}) = \frac{4}{14}$ $P(\text{Regen}) = \frac{5}{14}$

$P(\text{Mild}) = \frac{6}{14}$ $P(\text{kühl}) = \frac{4}{14}$ $P(\text{Warm}) = \frac{4}{14}$

$P(\text{Hoch}) = \frac{1}{2}$ $P(\text{Normal}) = \frac{1}{2}$

$P(\text{Schwach}) = \frac{8}{14}$ $P(\text{Stärk}) = \frac{6}{14}$

$P(\text{Ja}) = \frac{9}{14}$ $P(\text{Nein}) = \frac{5}{14}$

$P(\text{Sonnig} \mid \text{Ja}) = \frac{2}{9}$ $P(\text{Bewölkung} \mid \text{Ja}) = \frac{4}{9}$ $P(\text{Regen} \mid \text{Ja}) = \frac{3}{9}$

$P(\text{Mild} \mid \text{Ja}) = \frac{4}{9}$ $P(\text{kühl} \mid \text{Ja}) = \frac{3}{9}$ $P(\text{Warm} \mid \text{Ja}) = \frac{2}{9}$

$P(\text{Hoch} \mid \text{Ja}) = \frac{3}{9}$ $P(\text{Normal} \mid \text{Ja}) = \frac{6}{9}$

$P(\text{Schwach} \mid \text{Ja}) = \frac{6}{9}$ $P(\text{Stärk} \mid \text{Ja}) = \frac{3}{9}$

$P(\text{Sonnig} \mid \text{Nein}) = \frac{3}{5}$ $P(\text{Bewölkung} \mid \text{Nein}) = 0$ $P(\text{Regen} \mid \text{Nein}) = \frac{2}{5}$

$P(\text{Mild} \mid \text{Nein}) = \frac{2}{5}$ $P(\text{kühl} \mid \text{Nein}) = \frac{1}{5}$ $P(\text{Warm} \mid \text{Nein}) = \frac{2}{5}$

$P(\text{Hoch} \mid \text{Nein}) = \frac{4}{5}$ $P(\text{Normal} \mid \text{Nein}) = \frac{1}{5}$

$P(\text{Schwach} \mid \text{Nein}) = \frac{2}{5}$ $P(\text{Stärk} \mid \text{Nein}) = \frac{3}{5}$

Abbildung 16: task 6 b

1.6) c :

$$P(\text{Ja} \mid \text{Sonnig, Mild, Hoch, Schwach}) = \frac{P(\text{Sonnig, Mild, Hoch, Schwach} \mid \text{Ja}) \cdot P(\text{Ja})}{P(\text{Sonnig, Mild, Hoch, Schwach})}$$

$$= \frac{\frac{2}{9} \cdot \frac{4}{9} \cdot \frac{3}{9} \cdot \frac{6}{9} \cdot \frac{9}{14}}{\frac{5}{14} \cdot \frac{6}{14} \cdot \frac{1}{2} \cdot \frac{8}{14}} = \frac{392}{1215}$$

$$P(\text{Nein} \mid \text{Sonnig, Mild, Hoch, Schwach}) = \frac{P(\text{Sonnig, Mild, Hoch, Schwach} \mid \text{Nein}) \cdot P(\text{Nein})}{P(\text{Sonnig, Mild, Hoch, Schwach})}$$

$$= \frac{\frac{3}{5} \cdot \frac{2}{5} \cdot \frac{4}{5} \cdot \frac{2}{5} \cdot \frac{5}{14}}{\frac{5}{14} \cdot \frac{6}{14} \cdot \frac{1}{2} \cdot \frac{8}{14}} = \frac{392}{625}$$

$$\Rightarrow P(\text{Ja} \mid \text{Sonnig, Mild, Hoch, Schwach})^* = \frac{\frac{392}{1215}}{\frac{392}{1215} + \frac{392}{625}} = \frac{125}{368} \approx 0.3397$$

$$P(\text{Nein} \mid \text{Sonnig, Mild, Hoch, Schwach})^* = \frac{\frac{392}{625}}{\frac{392}{1215} + \frac{392}{625}} = \frac{243}{368} \approx 0.6603$$

Abbildung 17: task 6 c

Lösungsvorschlag:

$$P(\text{Ja} \mid \text{Bewölkung, Mild, Normal, Schwach}) = \frac{P(\text{Bewölkung, Mild, Normal, Schwach} \mid \text{Ja}) \cdot P(\text{Ja})}{P(\text{Bewölkung, Mild, Normal, Schwach})}$$

$$= \frac{\frac{4}{9} \cdot \frac{4}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} \cdot \frac{9}{14}}{\frac{4}{14} \cdot \frac{6}{14} \cdot \frac{1}{2} \cdot \frac{8}{14}} = \frac{329}{243}$$

$$P(\text{Nein} \mid \text{Bewölkung, Mild, Normal, Schwach}) = \frac{P(\text{Bewölkung, Mild, Normal, Schwach} \mid \text{Nein}) \cdot P(\text{Nein})}{P(\text{Bewölkung, Mild, Normal, Schwach})}$$

$$= \frac{0 \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} \cdot \frac{5}{14}}{\frac{4}{14} \cdot \frac{6}{14} \cdot \frac{1}{2} \cdot \frac{8}{14}} = 0$$

$$\Rightarrow P(\text{Ja} \mid \text{Bewölkung, Mild, Normal, Schwach})^{*} = \frac{\frac{329}{243}}{\frac{329}{243} + 0} = 100\%$$

$$P(\text{Nein} \mid \text{Bewölkung, Mild, Normal, Schwach})^{*} = \frac{0}{\frac{329}{243} + 0} = 0$$

Abbildung 18: task 6 c

$$P(\text{Ja} \mid \text{Regen, Kühl, Normal, Stark}) = \frac{P(\text{Regen, Kühl, Normal, Stark} \mid \text{Ja}) \cdot P(\text{Ja})}{P(\text{Regen, Kühl, Normal, Stark})}$$

$$= \frac{\frac{3}{9} \cdot \frac{3}{9} \cdot \frac{6}{9} \cdot \frac{3}{9} \cdot \frac{9}{14}}{\frac{5}{14} \cdot \frac{4}{14} \cdot \frac{1}{2} \cdot \frac{6}{14}} = \frac{98}{135}$$

$$P(\text{Nein} \mid \text{Regen, Kühl, Normal, Stark}) = \frac{P(\text{Regen, Kühl, Normal, Stark} \mid \text{Nein}) \cdot P(\text{Nein})}{P(\text{Regen, Kühl, Normal, Stark})}$$

$$= \frac{\frac{2}{5} \cdot \frac{1}{5} \cdot \frac{1}{5} \cdot \frac{3}{5} \cdot \frac{5}{14}}{\frac{5}{14} \cdot \frac{4}{14} \cdot \frac{1}{2} \cdot \frac{6}{14}} = \frac{98}{625}$$

$$\Rightarrow P(\text{Ja} \mid \text{Regen, Kühl, Normal, Stark})^* = \frac{\frac{98}{135}}{\frac{98}{135} + \frac{98}{625}} = \frac{125}{152} \approx 0.8224$$

$$P(\text{Nein} \mid \text{Regen, Kühl, Normal, Stark})^* = \frac{\frac{98}{625}}{\frac{98}{135} + \frac{98}{625}} = \frac{27}{152} \approx 0.1776$$

Abbildung 19: task 6 c

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	Nein
T16	Bewölkung	Mild	Normal	Schwach	Ja
T17	Regen	Kühl	Normal	Stark	Ja

Abbildung 20: task 6 c

Aufgabe 1.7: K-Means (6)

Folgender Datensatz besteht aus 8 Punkten:

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

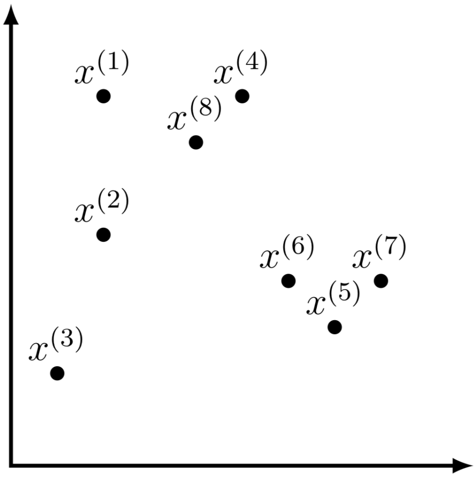
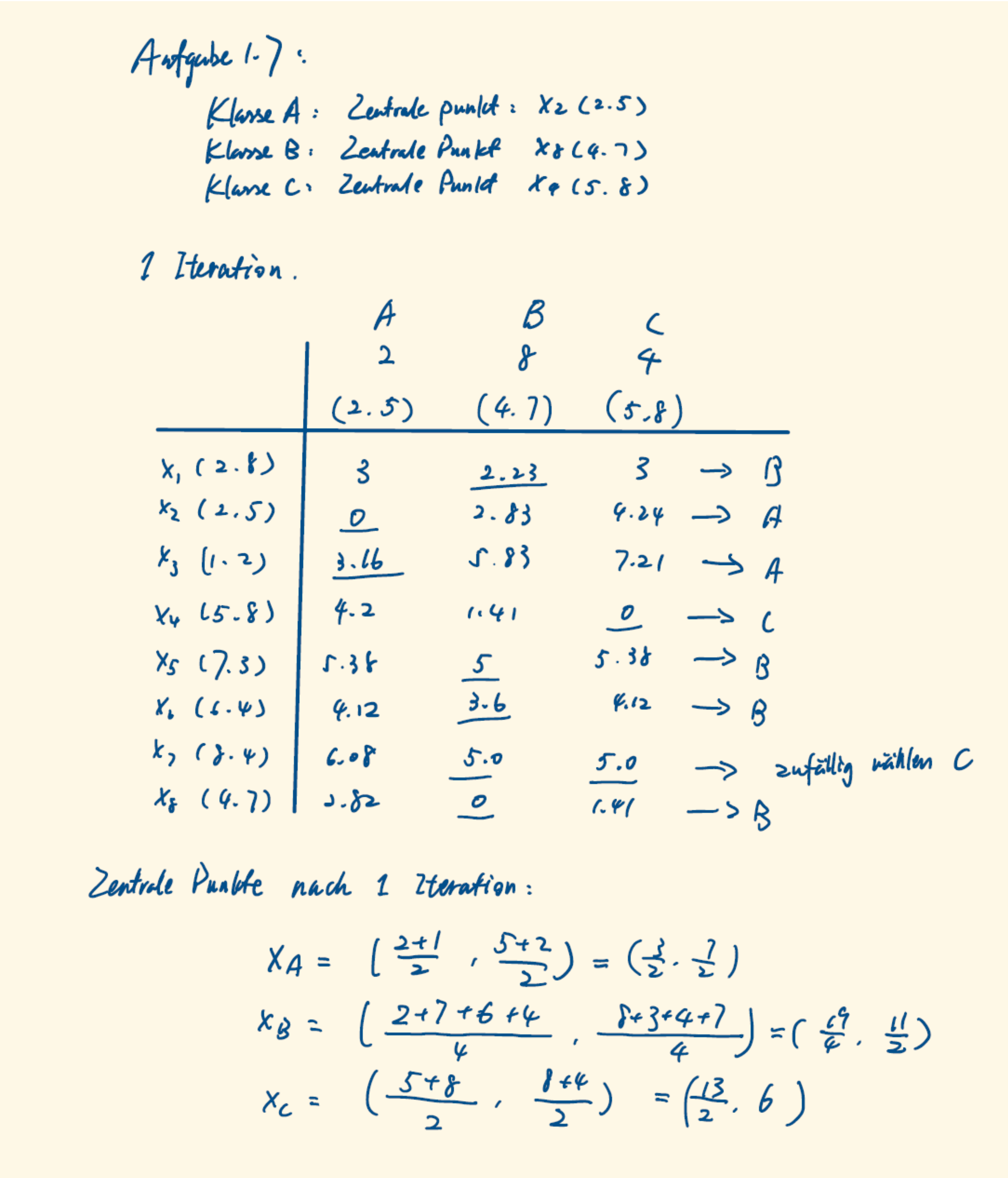


Abbildung 21: Visualisierung des K-Means Datensatzes

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 1.7:

Klasse A: Zentrale punkt: x_2 (2.5)
Klasse B: Zentrale Punkt x_8 (4.7)
Klasse C: Zentrale Punkt x_4 (5.8)

1 Iteration.

	A 2 (2.5)	B 8 (4.7)	C 4 (5.8)		
x_1 (2.8)	3	2.23	3	→	B
x_2 (2.5)	0	2.83	4.24	→	A
x_3 (1.2)	3.16	5.83	7.21	→	A
x_4 (5.8)	4.2	1.41	0	→	C
x_5 (7.3)	5.38	5	5.38	→	B
x_6 (6.4)	4.12	3.6	4.12	→	B
x_7 (8.4)	6.08	5.0	5.0	→	zufällig wählen C
x_8 (4.7)	2.82	0	1.41	→	B

Zentrale Punkte nach 1 Iteration:

$$X_A = \left(\frac{2+1}{2}, \frac{5+2}{2}\right) = \left(\frac{3}{2}, \frac{7}{2}\right)$$

$$X_B = \left(\frac{2+7+6+4}{4}, \frac{8+3+4+7}{4}\right) = \left(\frac{19}{4}, \frac{11}{2}\right)$$

$$X_C = \left(\frac{5+8}{2}, \frac{8+4}{2}\right) = \left(\frac{13}{2}, 6\right)$$

Abbildung 22: task 7

2. Iteration:

	A $(\frac{3}{2}, \frac{7}{2})$	B $(\frac{19}{4}, \frac{11}{2})$	C $(\frac{13}{2}, 6)$	
x_1 (2,8)	4.52	3.72	4.92	→ B
x_2 (2,5)	1.58	2.80	4.61	→ A
x_3 (1,2)	1.58	5.13	6.80	→ A
x_4 (5,8)	5.70	2.51	2.5	→ C
x_5 (7,3)	5.52	3.36	3.04	→ C
x_6 (6,4)	4.52	1.95	2.06	→ B
x_7 (8,4)	6.52	3.58	2.5	→ C
x_8 (4,7)	4.30	1.68	2.69	→ B

Zentrale Punkte nach 2 iteration:

$$X_A = \left(\frac{2+1}{2}, \frac{5+2}{2}\right) = \left(\frac{3}{2}, \frac{7}{2}\right)$$

$$X_B = \left(\frac{2+6+4}{3}, \frac{8+4+7}{3}\right) = \left(4, \frac{19}{3}\right)$$

$$X_C = \left(\frac{5+7+8}{3}, \frac{8+3+4}{3}\right) = \left(\frac{20}{3}, 5\right)$$

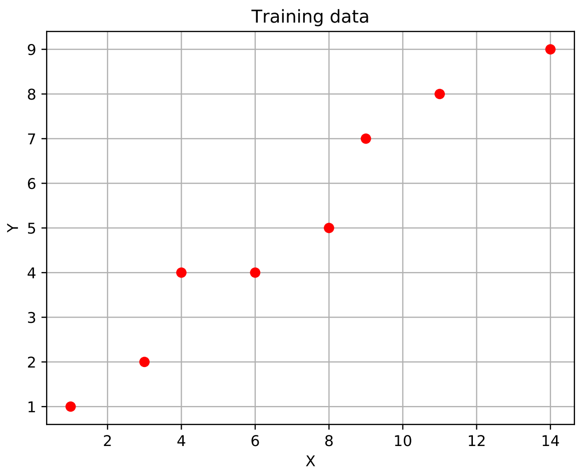
Abbildung 23: task 7

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 24: 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.

Aufgabe 1.8:

1.8a) setzen $K = XW' - y$

$J(w) = K^T \cdot K$

$J'(w) = (K^T \cdot K) \cdot K'$

$= (K^T \cdot K) \cdot (XW' - y)'$

$= (K^T \cdot K)(XW')'$

mit $(Ax)' = A^T$

$\Rightarrow J'(w) = (K^T \cdot K) \cdot X^T$

und $J'(w) = (K^T \cdot I \cdot K) \cdot X^T$ I ist einheitmatrix

mit $(x^T A x) = (A^T + A)x$

$\Rightarrow J'(w) = (I^T + I) K \cdot X^T$

$J'(w) = 2K \cdot X^T$

$= 2(XW' - y) \cdot X^T$

$= 2(X^T X W' - X^T y)$

um W' zu berechnen, $J'(w)$ muss gleich null.

$\Rightarrow 2(X^T X W' - X^T y) = 0$

$\Rightarrow X^T X W' - X^T y$

$W' = (X^T X)^{-1} \cdot X^T y$

Abbildung 25: task 8 a

1.8b) Das Ergebnis nach Code ist $W' = [0.3454, 0.6363]$

Abbildung 26: task 8 b

Lösungsvorschlag: