Duale Hochschule Baden - Württemberg Mannheim

Seminararbeit

Entscheidungsbäume

Studiengang Angewandte Informatik

Studienrichtung Informatik

Autor:	Martin Pretz
Matrikelnummern:	7060026
Kurs:	TINF18AI1
Bearbeitungszeitraum:	19.05.2021 - 10.06.2021

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Abstract

2 Einführung

3 Was sind Entscheidungsbäume?

Bei Entscheidungsbäumen handelt es sich um eine bestimmte Form von Klassifikationsalgorithmen.

3.1 Motivation und Ziel

3.2 Generischer Aufbau

Im wesentlichen bestehen Entscheidungsbäume aus Knoten und Kanten. Bei einem Knoten handelt es sich um ein zu prüfendes Attribut während es sich bei einer Kante um das Ergebnis dieser Überprüfung handelt. [1] Darüber hinaus können Knoten wiederrum in Entscheidungsknoten, Wahrscheinlichkeitsknoten und Endknoten unterteilt werden.

Entscheidungsbäume bestehen im wesentlichen aus den vier Bestandteilen Wurzel, Knoten, Kante und Blatt. Bei der Wurzel handelt es sich im Grunde um einen Knoten. Bei einem Blatt handelt es sich um eine

3.3 Vorteile

3.4 Nachteile

4 Der ID3 Algorithmus

Bei ID3 (Iterative Dichotomiser 3) handelt es sich um einen Algorithmus zur Erstellung eines Entscheidungsbaumes welcher von Ross Quinlan entwickelt wurde. [2]

4.1 Grundlagen

Der ID3-Algorithmus macht sich zwei Konzepte der Informationstheorie zu nutze. Es handelt sich zum einen um die Entropie und zum anderen um den Informationsgewinn. Beide Konzepte werden im nachfolgenden erläutert. Im Anschhluss daran wird die eigentliche Funktionsweise des ID3 Algorithmus erläutert.

4.1.1 Entropie

In der Informationstheorie wird mit der Entropie H die Sicherheit bzw. Unsicherheit einer Variablen X angegeben. Dementsprechend ist x_i eine mögliche Ausprägung der Variablen X und $P(x_i)$ die Wahrscheinlichkeit mit der die Variable X die Ausprägung x_i hat. [3]

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i)$$

Abbildung 4.1: Definition der Entropie nach Shennon[3]

Beispiel: Sei D ein Datensatz in dem das Attribut X mit den möglichen Ausprägungen x_1, x_2 und x_3 vorkommt. Weiterhin gelte, dass x_1 neun mal, x_2 drei mal und x_3 fünf mal in D vorhanden ist. Zur Bestimmung der Entropie von X egribt sich die nachfolgende Berechnung. Für den ID3-Algorithmus wird üblicherweise der Logarithmus zur Basis *b=2* verwendet. [4]

$$
\begin{aligned}
H(X) &= -\sum_{i=1}^{3} P(x_i)\log_2 P(x_i) \\
&= -(P(x_1)\log_2 P(x_1) + P(x_2)\log_2 P(x_2) + P(x_3)\log_2 P(x_3)) \\
&= -\frac{9}{17}\cdot\log_2\left(\frac{9}{17}\right) - \frac{3}{17}\cdot\log_2\left(\frac{3}{17}\right) - \frac{5}{17}\cdot\log_2\left(\frac{5}{17}\right) \\
&\approx 0,485755 + 0,441618 + 0,519275 \\
&\approx \underline{\underline{1,446648}}
\end{aligned}
$$

Abbildung 4.2: Exmeplarische Berechung der Entropie

4.1.2 Informationsgewinn

In der Informationstheorie beschreibt der Informationsgewinn IG das Maß an Informationen das über eine Zufallsvariablen X durch Beobachtung einer anderen Zufallsvariablen Y gewonnen werden kann. [1] Konkret ergibt sich der Informationsgewinn aus der Differenz der Entropie $H(X)$ und der bedingten Entropie $H(X|Y)$. [5]

$$IG(X,Y) = H(X) - H(X|Y) = H(X) - \sum_{y \in Y} P(y)H(X|Y=y)$$

Abbildung 4.3: Allgemeine Definition des Informationsgewinns [1, 6, 7]

Beispiel: Sei S ein Datensatz mit den in Tabelle 4.1 dargestellten Werten. Außerdem seien A, B, C und T Attribute von S mit den möglichen Ausprägungen *True* und *False*. Sei weiterhin das Attribut T das Zielattribut gegen das der Informationsgewinn der übrigen Attribute ermittelt werden soll.

ID	Attribut A	Attribut B	Attribut C	Attribut T
1	True	True	True	False
2	True	False	True	True
3	False	False	True	True
4	False	True	True	False
5	False	True	False	True

Tabelle 4.1: Beispiel Datensatz S zur Berechnung des Informationsgewinns [5]

Im nachfolgenden wird der Informationsgewinn für das Attribut A berechnet. Dabei wird zu erst die Entropie des Datensatz S bestimmt, welche durch die Entropie des Zielattributes T charakterisiert wird. Es gilt also $H(S) = H(T)$.

$$\begin{aligned}
H(S) = H(T) &= -\sum_{i=1}^{2} P(x_i) \log_2 P(x_i) \\
&= -(P(\text{True}) \log_2 P(\text{True}) + P(\text{False}) \log_2 P(\text{False})) \\
&= -\frac{3}{5} \cdot \log_2 \left(\frac{3}{5}\right) - \frac{2}{5} \cdot \log_2 \left(\frac{2}{5}\right) \\
&\approx 0,970951
\end{aligned}$$

Im nächsten Schritt wird die bedingte Entropie für das Attribut A berechnet. [7, 6]

$$\begin{aligned}
H(T|A) &= \sum_{a \in A} P(A = a) \cdot H(T|A = a) \\
&= P(A = True) \cdot H(T|A = True) + P(A = False) \cdot H(T|A = False) \\
&= \frac{2}{5} \cdot \left(-\frac{1}{2} \cdot \log_2 \left(\frac{1}{2}\right) - \frac{1}{2} \cdot \log_2 \left(\frac{1}{2}\right)\right) + \frac{3}{5} \cdot \left(-\frac{1}{3} \cdot \log_2 \left(\frac{1}{3}\right) - \frac{2}{3} \cdot \log_2 \left(\frac{2}{3}\right)\right) \\
&= \frac{2}{5} \cdot 1 + \frac{3}{5} \cdot 0,918296 \\
&\approx 0.950978
\end{aligned}$$

Nachdem nun sowohl die Entropie als auch die bedingte Entropie berechnet sind kann final der Informationsgewinn bestimmt werden.

$$\begin{aligned}
IG(T, A) &= H(T) - H(T|A) \\
&= 0,970951 - 0.950978 \\
&= \underline{\underline{0,019973}}
\end{aligned}$$

Analog können die Informationsgewinne für die Attribute B und C berechnet werden. Diese liegen bei $IG(T, B) = \mathit{0{,}419973}$ und bei $IG(T, C) = \mathit{0{,}170951}$.

4.1.3 Funktionsweise

Der ID3 Algorithmus startet mit einem Datensatz S mit Objekten $O_1, ..., O_n$ welche verschiedenen Attribute $A_1, ..., A_k$ und eine Klassifizierung C besitzen. [4, 2] Die Werteausprägungen der Attribute sind dabei normalerweise endlich und diskret. [8]
Zu Beginn muss zunächst Knoten bestimmt werden. Dazu wird der Informationsgewinn $IG(S)$ für jedes Attribut $A_1, ..., A_k$ berechnet. Das Attribut A_H mit dem höchsten Informationsgewinnungswert wird dann als Knoten gewählt. Dabei gilt das A_H die möglichen Ausprägungen $a_1, ..., a_v$ haben kann. Basierend auf A_H wird S in $S_1, ..., S_v$ Teilmengen zerlegt wobei die Teilmenge S_i die Objekte aus S beinhaltet deren Wert in A_H die Ausprägungen a_i hat. Anders ausgedrückt gilt $S_i = \{S|A_H = a_i\}$. Für jede Teilmenge S_i wird dann rekursiv der vorher beschriebene Prozess für die verleibenden Attribute $\{A_1, ..., A_k\} \setminus A_H$ durchlaufen wodurch entsprechende Teilbäume erzeugt werden. [2] Dabei wird die Rekursion beendet sofern,

1. alle Objekte aus S_i die gleiche Ausprägung C_i der Klassifizierung C besitzen. In diesem Fall wird der zu diesem Zeitpunkt ausgewählte Knoten zu einem Blatt mit der Ausprägung C_i. [2, 9]
2. bereits alle Attribute $A_1, ..., A_k$ für den Entscheidungsbaum verwendet worden sind, es also keine Attribute mehr gibt mittels derer ein Objekt aus S_i klassifiziert werden könnte, aber nicht alle Objekte der selben Klassifizierung C_i zugeordnet werden können. In diesem Fall wird der aktuelle Knoten zu einem Blatt, welchem der Modalwert von C zugeordnet wird.[2, 9]
3. die Teilmenge S_i keine Objekte mehr beinhaltet, sie also leer ist. Dies tritt ein wenn sich in der übergeordneten Menge keine Objekte befinden welche die Ausprägung a_i eines Attributes A_M besitzen. In diesem Fall wird ein Blatt erzeugt welchem der Modalwert von C aus der übergeordneten Menge zugeordnet wird. [9]

4.1.4 Eigenschaften von ID3

Der ID3-Algorithmus führt eine Best-First-Search für lokale Optima durch. Außerdem ist ID3 ein gieriger Algorithmus, da er stets das Attribut auswählt welches lokal den besten Informationsgewinn aufweist.

TODO: Vervollständigen

4.2 Verwendeter Beispiel Datensatz

Für diese Arbeit wurde ein Datensatz verwendet welcher auf "RiskSample.csv" basiert. [10] In diesem Datensatz werden verschiedene Attribute im Zusammenhang mit einer Kreditvergabe erfasst. Das Ziel ist es anhand von bestimmten Attributen das Kreditrisiko zu klassifizieren welches in dem Attribut *RISK* erfasst wird. Dabei wird zwischen hohem Risiko (*good risk*), schlechtem Profit (*bad profit*) und schwerem Verlust (*bad loss*) unterschieden. Für die Zwecke dieser Arbeit wird nur der in Tabelle 4.2 dargestellte Auszug des originalen Datensatzes verwendet.

AGE	INCOME	NUMKIDS	MORTGAGE	LOANS	RISK
42.0	29540.0	3.0	Yes	2.0	bad loss
28.0	24332.0	1.0	No	1.0	bad loss
36.0	44048.0	1.0	Yes	0.0	good risk
44.0	25971.0	4.0	Yes	3.0	bad loss
35.0	41132.0	0.0	Yes	1.0	good risk
22.0	15279.0	0.0	No	1.0	bad profit
44.0	16494.0	2.0	Yes	1.0	bad loss
24.0	19782.0	0.0	No	0.0	bad loss
21.0	53402.0	2.0	Yes	2.0	bad loss
28.0	22070.0	1.0	Yes	1.0	bad profit

Tabelle 4.2: Originaler Beispiel Datensatz

Der originale Datensatz erfasst insgesamt 11 Attribute. Diese sind *AGE*, *INCOME*, *GENDER*, *MARTIAL*, *NUMKIDS*, *NUMCARDS*, *HOWPAID*, *MORTGAGE*, *STORECAR*, *LOANS* und *ID*. Der Beispiel-Datensatz für den ID3 Algorithmus berücksichtigt davon nur noch fünf Attribute, nämlich *AGE*, *INCOME*, *NUMKIDS*, *MORTGAGE* und *LOANS*.

4.2.1 Transformation

Bevor die Daten verwendet werden, müssen sie zunächst eine Transformation durchlaufen, wobei diese "bereiningt" werden. Im nachfolgenden werden daher die Transformationen der betroffenen Attribute dargelegt.

Bei diesem Attribut *AGE* handelt es sich um das Alter einer Person welches im originalen Datensatz als Integer vorliegt. Im Zuge der Diskretisierung dieses Attributes wird das Alter in drei Kategorien eingeteilt. Dies sind *Young* (unter 30 Jahren), *Middle* (zwischen 30 und 50 Jahren) und *Old* (über 50 Jahre). Hierbei ist zu beachten dass das Alter im Datensatz lediglich zwischen minimal 18 und maximal 60 Jahren liegt.

Das Attribut *INCOME* liegt in originalen Datensatz als Integer vor und beziffert das jährliche Einkommen einer Person. Auch dieses Attribut wird diskretisiert und in vier Kategorien eingeteilt. Diese sind *Low* (unter 20.000 Euro), *Middle* (zwischen 20.000 und 30.000 Euro), *High* (zwischen 30.000 und 50.000 Euro) und *Very High* (über 50.000 Euro).

NUMKIDS erfasst im originalen Datensatz die Anzahl der Kinder einer Person. Allerdings wird dies im Ziel-Datensatz nicht länger berücksichtigt. Stattdessen gibt es nur eine Unterscheidung ob eine Person ein Kind hat oder nicht, also zwischen den beiden Zuständen *Yes* (Person hat Kinder) und *No* (Person hat keine Kinder).

Im originalen Datensatz wird mit dem Attribut *LOANS* die Anzahl der Darlehen erfasst während in dem transformierten Datensatz nur das Vorhandensein eventueller Darlehen, also nur die Zustände *Yes* (Person hat bereits Darlehen) oder *No* (Person hat aktuell kein Darlehen) erfasst werden.

4.2.2 Finaler Datensatz

Nachdem alle Transformation duchgeführt wurden, ergibt sich für die Tabelle 4.2 nun folgende Struktur.

AGE	INCOME	NUMKIDS	MORTGAGE	LOANS	RISK
Old	Middle	Yes	Yes	Yes	bad loss
Old	Middle	Yes	No	Yes	bad loss
Middle	High	Yes	Yes	Yes	good risk
Middle	Middle	Yes	Yes	Yes	bad loss
Old	High	No	Yes	Yes	good risk
Young	Low	No	No	Yes	bad profit
Old	Low	Yes	Yes	Yes	bad loss
Old	Low	Yes	No	No	bad loss
Old	Very High	Yes	Yes	Yes	bad loss
Old	Middle	Yes	Yes	Yes	bad profit

Tabelle 4.3: Transformierter Beispiel Datensatz

4.3 Implementation

Diese persönliche Implementation besteht im wesentlichen aus vier Funktionen. Zum einen aus Nebenfunktionen wie der Berechnung des Informationsgewinns, der Entropie und des Modalwerts. Zum anderen besteht die Implementation aus der Hauptfunktion, dem eigentlichen ID3 Algorithmus.

4.3.1 Berechnung der Entropie

Die hier vorliegende Implementation zur Berechnung der Entropie erwartet als Eingabeparameter ein Attribut eines Datensatzes. Im Anschluss daran wird über die Zuweisung in Zeile 3 die Anzahl aller Ausprägungne des Attributes ermittelt. Dabei kommt die Bibliotheksfunktion `np.unique` zum Einsatz. Diese Funktion gibt zwei Listen zurück. Die erste Liste beinhaltet dabei alle möglichen Ausprägungen des Attributes während die zweite Liste die Anzahl eben jener Ausprägungen beinhaltet. Im Anschluss daran wird über die Länge der Liste `count` iteriert. Dabei wird in jedem Durchlauf die Wahrscheinlichkeit einer Ausprägung des Attributes berechnet. Außerdem wird die Entropie partiell für die vorliegende Ausprägung berechnet und mit dem vorherigen partiellen Entropiewert addiert.

```python
1 def entropy(attribute):
2     entropy = 0.0
3     values, count = np.unique(attribute, return_counts=True)
4     for index in range(len(values)):
5         probablility = count[index] / sum(count)
6         entropy += (-probablility * np.log2(probablility))
7     return entropy
```

Abbildung 4.4: Funktion zu Berechnung der Entropie eines Attributes[11, 4]

4.3.2 Berechnung des Informationsgewinns

Die vorliegende Implementation des Informationsgewinns erwartet drei Eingabeparameter. Diese sind der Datensatz, ein Attribut dieses Datensatzes und das Zielattribute gegen welches der Informationsgewinns bestimmt werden soll. Zu Beginn werden unter Zuhilfenahme der Funktion `np.unique` alle möglichen Ausprägungen des Attributes sowie deren Anzahl bestimmt.
Im Hauptteil dieser Funktion wird über die möglichen Ausprägungen `values` des Attribute iteriert. Dabei wird in Zeile 5 zunächst eine Teilmenge `subdata` des ursprünglichen Datensatzes gebildet (vgl. Kapitel 4.1.2). Im Anschluss daran wird die Wahrscheinlichkeit der Ausprägung `value`, sowie die Entropie der Teilmenge `subdata` berechnet. Dabei werden die Entropieen der verschiedenen Ausprägungen addiert. Im letzen Schritt wird der Informationsgewinn aus der Differenz der Entropie des Zielattributes und der bedingten Entropie berechnet (vgl. Kapitel 4.1.2).

4.3.3 Berechnung des Modalwertes

Die Funktion `modal` erwartet eine Liste von Elementen als Eingabeparameter. Zu Beginn der Funktion wird zunächst mit Hilfe der Funktion `np.unique` die Menge aller vorkommenden Werte sowie deren Anzahl in der Anfangsliste bestimmt. Im Anschluss daran wird aus beiden Informationen ein Dictionary erstellt. Im letzten Schritt wird

```python
def information_gain(data, attribute, target_attribute):
    values, count = np.unique(data[attribute], return_counts=True)
    entropy_val = 0.0
    for index, value in enumerate(values):
        subdata = data[data[attribute] == value][target_attribute]
        probablility = count[index] / sum(count)
        entropy_val += ( probablility * entropy(subdata) )
    target_entropy = entropy(data[target_attribute])
    information_gain = target_entropy - entropy_val
    return information_gain
```

Abbildung 4.5: Funktion zur Berechnung des Informationsgewinns [11, 4]

mittels einer Lambda Funktion der Schlüssel des Dictionary ermittelt dessen Wert am höchsten ist.

```python
def modal(attribute):
    values, count = np.unique(attribute, return_counts=True)
    total = dict(zip(values, count))
    return max(total, key=lambda k: total[k])
```

Abbildung 4.6: Berechnung des Modalwertes [12]

4.3.4 ID3 - Hauptfunktion

Die Hauptfunktion erwartet als Eingabeparameter den Ursprungsdatensatz, das Zielattribut und eine Liste aller möglichen Attribute des Datensatzes. Bei dem Zielattribut handelt es sich um die Klassifizierung.
Zu Beginn der Funktion werden zwei Abbruchbedingungen geprüft (vgl. Kapitel 4.1). Als erstes wird untersucht, ob die sich in dem Datensatz `data_set` auschließlich Objekte mit der gleichen Klassifizierung befinden. Wenn dem so ist, wird genau diese Klassifizierung zurück gegeben. Als zweites wird untersucht, ob bereits alle Attribute verwendet worden sind. Sofern dies zutrifft wird der Modalwert des Zielattributes zurück gegeben.
Im Anschluss daran wird mit der Funktion `calculate_all_IG` der Informationsgewinn für jedes Attribut berechnet, wobei das Attribut mit dem höchsten Wert (`best_attribute`) als (Wurzel-) Knoten gewählt wird. Im Anschluss daran wird in Zeile 10 `best_attribute` aus der Menge aller Attribute entfernt.
Nun beginnt der iterative Teil der Funktion. Dabei wird über alle Ausprägungen des Attributes `best_attribute` eine Teilmenge `subset` des ursprünglichen Datensatzes erstellt. Danach wird die dritte Abbruchbedingung für die Rekursion geprüft, nämlich ab die Teilmenge `subset` leer ist. In diesem Fall wird ein Blatt erstellt welchem der

Modalwert `modal_value` des Zielattributes zugeordnet wird. Wenn allerdings die Teilmenge `subset` nicht leer ist, so wird in Zeile 18 rekursiv ein Teilbaum erstellt. Dieser wird in Zeile 19 der Ausprägung `value` des Attributes `best_attribute` zugewiesen.

```
1  def ID3(data_set, target_attribute, attributes):
2      if len(np.unique(data_set[target_attribute])) <= 1:
3          return np.unique(data_set[target_attribute])[0]
4      if len(attributes) <= 1:
5          return modal(data_set[target_attribute])
6
7      IG = calculate_all_IG(data_set, target_attribute, attributes)
8      best_attribute = max(IG, key=lambda k: IG[k])
9      tree = {best_attribute: {}}
10     attributes = [x for x in attributes if x != best_attribute]
11
12     for value in np.unique(data_set[best_attribute]):
13         subset = data_set[data_set[best_attribute] == value]
14         if subset.empty:
15             modal_value = modal(data_set[target_attribute])
16             tree[best_attribute][value] = modal_value
17         else:
18             subtree = ID3(subset, target_attribute, attributes)
19             tree[best_attribute][value] = subtree
20     return tree
```

Abbildung 4.7: Hauptfunktion des ID3 Algorithmus [12, 9, 11]

4.4 Anwendung

In diesem Kapitel wird erläutert wie die oben beschriebene Implementation aus dem Beispiel-Datensatz (vgl. Kapitel 4.2) einen Entscheidungsbaum erzeugt. Die Erstellung des Entscheidungsbaumes beginnt mit dem Aufruf `ID3(data, "RISK", attributes)`. Der Datensatz `data` ist dabei der Datensatz aus Kapitel 4.2.2. Dabei beinhaltet `attributes` eine Liste mit den Attributen *AGE*, *INCOME*, *NUMKIDS*, *MORTGAGE*, *LOANS*.

Zu Beginn werden die Abbruchbedingungen überprüft. Da die Objekte im Datensatz nicht alle die selbe Klassifizierung besitzen und noch Attribute zur Verfügung stehen fallen die Abbruchbedingung negativ aus. Daher wird als nächstes der Informationsgewinn für die Attribute berechnet. Dabei ergeben sich folgende Werte.

Das Attribut *INCOME* besitzt den höchsten Informationsgewinn und wird daher als Knoten gewählt aus `attributes` entfernt. Die erste Iteration erfolgt für die Aus-

$$
\begin{aligned}
&IG(data,\ AGE) && = 0{,}36677\\
&IG(data,\ INCOME) && = 0{,}77095\\
&IG(data,\ NUMKIDS) && = 0{,}32192\\
&IG(data,\ MORTGAGE) && = 0{,}13031\\
&IG(data,\ LOANS) && = 0{,}07898
\end{aligned}
$$

prägung *High* wobei die Teilmenge `subdata` angelegt welche nun folgende Struktur hat.

AGE	**INCOME**	**NUMKIDS**	**MORTGAGE**	**LOANS**	**RISK**
Middle	High	Yes	Yes	Yes	good risk
Old	High	No	Yes	Yes	good risk

Tabelle 4.4: Teilmenge `subdata` der Ausprägung *High*

Da `subdata` nicht leer ist, wird die dritte Abbruchbedingung nicht ausgelöst. Stattdessen wird nun rekursiv ein Teilbaum für `subdata` angelegt. Dabei werden zunächst wieder die Abbruchbedingung der Rekursion geprüft. Dabei fällt mit Bezug auf die Tabelle 4.4 auf, dass eine Abbruchbedingung der Rekursion erfüllt ist da alle Objekte aus `subdata` die selbe Klassifizierung aufweisen, nämlich *good risk*. Daher wird nun ein Blatt mit dieser Klassifizierung erzeugt.

Die zweite Iteration erfolgt mit der Ausprägung *Middle*. Auch hier wird erneut eine Teilmenge `subdata` angelegt welche wie folgt aussieht.

AGE	**INCOME**	**NUMKIDS**	**MORTGAGE**	**LOANS**	**RISK**
Old	Middle	Yes	Yes	Yes	bad loss
Old	Middle	Yes	No	Yes	bad loss
Middle	Middle	Yes	Yes	Yes	bad loss
Old	Middle	Yes	Yes	Yes	bad profit

Tabelle 4.5: Teilmenge `subdata` der Ausprägung *Middle*

5 Zusammenfassung

Literatur

[1] M. Schinck, *Data Mining Vorlesung 2: Classification 1, Einführung, Validierung und Decision Trees*, Mannheim, 2021.

[2] J. Quinlan, *Induction of Decision Trees*, 1986.

[3] „Entropy (information theory) - Wikipedia.“ (), Adresse: `https://en.wikipedia.org/wiki/Entropy_(information_theory)`.

[4] S. V. Rupali Bhardwaj, *Implementation of ID3 Algorithm*, CSE, Bahra Univerity India, 2013.

[5] *Information gain in decision trees - Wikipedia*, (Accessed on 05/27/2021). Adresse: `https://en.wikipedia.org/wiki/Information_gain_in_decision_trees`.

[6] *Conditional entropy - Wikipedia*, `https://en.wikipedia.org/wiki/Conditional_entropy`, (Accessed on 05/27/2021).

[7] „Bedingte Entropie – Wikipedia.“ (), Adresse: `https://de.wikipedia.org/wiki/Bedingte_Entropie`.

[8] K. JEARANAITANAKIJ, *Classifying Continuous Data Set by ID3 Algorithm.*

[9] „ID3 algorithm - Wikipedia.“ (), Adresse: `https://en.wikipedia.org/wiki/ID3_algorithm`.

[10] M. Schinck, *Datamining Vorlesung, RiskSample.csv*, Mannheim, 2021.

[11] *Machine Learning with Python: Decision Trees in Python*, `https://www.python-course.eu/Decision_Trees.php`, (Accessed on 05/21/2021).

[12] *Python max()*, `https://www.programiz.com/python-programming/methods/built-in/max`, (Accessed on 05/21/2021).