h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbmn
FACHBEREICH MATHEMATIK
UND NATURWISSENSCHAFTEN

Gruppe 14
Tamara Döß (767956)
Tobias Rohrer (???)

Data Mining 1

Booklet von Gruppe 14

Inhaltsverzeichnis

Zusammenfassung **ii**

1 Aufgabe 1: Entscheidungsbäume **1**
1.1 Feature Engineering . . . 1
1.1.1 Fehlende Werte . . . 1
1.1.2 Merkmalsaufteilung . . . 1
1.1.3 Merkmalserstellung . . . 1
1.1.4 Diskretisierung . . . 1
1.1.5 Kodierung kategorischer Werte . . . 2
1.1.6 Bereinigung von Ausreißern . . . 2
1.1.7 Variablenselektion . . . 2
1.2 Entscheidungsbäume . . . 2
1.2.1 Default-Einstellungen . . . 2
1.2.2 Variationen . . . 3
1.2.3 Minimal Cost-Complexity-Pruning . . . 3

2 Aufgabe 2: Neuronale Netze **4**
2.1 Unterkapitel 1 . . . 4
2.1.1 Unterkapitel 1.1 . . . 4

3 Aufgabe 2: Ensemblemethoden **5**
3.1 Unterkapitel 1 . . . 5
3.1.1 Unterkapitel 1.1 . . . 5

4 Aufgabe 4: Support Vector Machines **6**
4.1 Unterkapitel 1 . . . 6
4.1.1 Unterkapitel 1.1 . . . 6

Anhang **I**
Quellcode zu Aufgabe 1 . . . I
Quellcode zu Aufgabe 2 . . . II
Quellcode zu Aufgabe 3 . . . III

Quellcode zu Aufgabe 4 IV

Zusammenfassung

1 Aufgabe 1: Entscheidungsbäume

1.1 Feature Engineering

1.1.1 Fehlende Werte

Zunächst wurde der betrachtete Datensatz auf fehlende Werte untersucht. Im ersten Schritt werden 834 Beobachtungen, welche keinen Wert für die Zielvariable *RainTomorrow* aufweisen, aus dem Datensatz entfernt. Damit wurde die Anzahl an Beobachtungen auf 33402 reduziert.

In einem nächsten Schritt werden die Spalten aus dem Datensatz entfernt, in denen mehr als 40% der beinhaltenden Variablen keine validen Werte aufweisen. Namentlich werden somit die Spalten *Evaporation, Sunshine, Cloud9am* sowie *Cloud3pm* aus dem Datensatz entfernt. Der Schwellwert von 40% wurde empirisch festgelegt und hat zu den besten Klassifizierungsergebnissen geführt.

Des Weiteren werden Beobachtungen aus dem Datensatz entfernt, von denen mehr als 50% der Variablen keine validen Werte zeigen. Das betrifft 55 Beobachtungen, sodass die Gesamtanzahl an Beobachtungen 33347 beträgt. Durch die ersten beiden Schritte wurden lediglich ca. 3% der Beobachtungen entfernt.

Die übrig gebliebenen Datensätze sind noch immer nicht frei von fehlenden Werten. Um diese zu ersetzen werden für kategorische und numerische Variablen verschiedene Strategien verfolgt. Für die *Imputation* numerischer Werten wird der Median der jeweiligen Variablen verwendet, da dieser im Vergleich zum Mittelwert robuster gegenüber Ausreißern ist. Für kategorielle Variablen hingegen wird der am häufigsten vorkommende Wert verwendet. Wichtig bei der Ermittlung des Medians bzw. des häufigsten Wertes ist, dass dieser ausschließlich getrennt für Trainigs- und Testdaten ermittelt wird. Die Ermittlung auf Basis des gesamten Datensatzes würde eine *Test-Train-Leakage* darstellen und ist zu vermeiden. Es muss davon ausgegangen werden, dass die Testdaten nicht bekannt sind. Die jeweils ermittelten Werte werden dann auf die Trainings- und Testdaten angewendet.

1.1.2 Merkmalsaufteilung

Das Feld *Datum* wurde in die Merkmale *Year, Month* und *Day* aufgeteilt.

1.1.3 Merkmalserstellung

Eine weit verbreitete Technik des Feature Engineerings ist die Erstellung zusätzlicher Merkmalen. Somit wurde die Variable *MinMaxDiff* erstellt, welche die Differenz zwischen der minimalen und der maximalen Tages-Temperatur angibt. In gleicher Weise wurden die Variablen *PressureDiff, HumidityDiff und WindSpeedDiff* erstellt.

1.1.4 Diskretisierung

Die Diskretisierung eines Merkmals kann eine Überanpassung bei der Erstellung von Modellen verhindern, indem der Wertebereich des Merkmals minimiert und somit generalisiert wird. Es sollte beachtet werden, dass der Informationsverlust durch die Diskretisierung nicht zu hoch ausfällt.

Zu Testzwecken wird das neu erstellte Merkmal *Month* in zwei weiteren Merkmalen diskretisiert. Zum einen wird das kategorische Merkmal *Season* erstellt, welches angibt, in welcher Jahreszeit der Monat liegt. Zum anderen gibt das Boolean Merkmal *RainyMonth* an, ob es sich um einen in Erwartung regenreichen Monat handelt.

1.1.5 Kodierung kategorischer Werte

Um kategorische Werte für weitere Analysen versenden zu können, müssen diese in numerische Werte umkodiert werden. Hierbei werden für die binäre Variablen *RainToday und RainTomorrow* alle Werte in Boolean umgewandelt. Das neu diskretisierte Merkmal *Season* wird mittels *One-Hot Encoding* in drei Boolean Spalten aufgeteilt. Eine einfache Zuteilung eines Zahlenwertes pro auftretender Variablenausprägung hat den Nachteil, dass dadurch eine metrische numerische Variable für ein gegebenenfalls nicht metrisch interpretierbares Merkmal entsteht. Für Entscheidungsbäume wäre dies kein Problem, jedoch sollte das Feature Engineering weitgehend unabhängig von dem verwendeten Klassifizierungsalgorithmus durchgeführt werden.

Mit Hilfe des *Target Encodings* werden die Merkmale *Location, WindGustDir, WindDir9am* und *WindDir3pm* verarbeitet. Dabei werden den jeweiligen Merkmalsausprägungen ihr Einfluss auf die Zielvariable zugeordent, also die bedingte Wahrscheinlichkeit, dass unter der Bedingung, dass z.B. Merkmal *Location* für die Zielvariable (*Target*) den Wert 1 (in unsrem Fall "Yes") annimmt. Zum Beispiel wird die Merkmalsausprägung "Perth"mit dem Wert 0.2 kodiert, wenn 20% der jeweiligen Beobachtungen für die Zielvariable den Wert "Yesännehmen. Somit wird keine zufällige Zuteilung von nummerischen Werten verteilt, sondern direkt einen Verbindung zu Zielvariable hergestellt. Ein Nachteil des Target Encoding ist, dass genau durch diese Verbindung zur Zielvariable die Wahrscheinlichkeit des Overfittings steigt (vgl. https://towardsdatascience.com/why-you-should-try-mean-encoding-17057262cd0 keine Wissenschaftliche Quelle).

1.1.6 Bereinigung von Ausreißern

1.1.7 Variablenselektion

1.2 Entscheidungsbäume

Für die Erstelltung der Entscheidungsbäume wurde der Datensatz in eine Trainings- und eine Testmenge eingeteilt. Dabei beträgt das Verhältnis 8:2. 80% der Daten sind genügend, um ein gutes Modell zu trainieren. 20% sind eine ausreichende Menge, um einen genauen Testfehler und die Modellgüte zu bestimmen.

1.2.1 Default-Einstellungen

Quelle: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
Die Entscheidungsbäume werden mit dem Modul *tree.DecistionTreeClassifier* erstellt. Im ersten Aufgabenteil werden dazu die Default-Einstellungen des Moduls genutzt.

Diese geben an, dass ein Baum immer maximal gebaut wird (*max_ depth = None*). Das heißt, so lange mehr als ein Objekt in einem Knoten vorliegt, wird dieser Knoten weiter aufgespalten (*min_ samples_ split = 2*). Das Ergebnis ist überangepasst, da jeder einzelne Datenpunkt ein eigenes Blatt im Baum bekommt (*min_ samples_ leaf = 1*). Dieser Baum kann somit nicht auf unbekannte Daten generalisiert werden und der Testfehler fällt sehr hoch aus. Die Splits werden mit dem Gini-Wert durchgeführt und nicht mit der Entropie (*citerion=„gini“*). Zudem werden für jedes Spalten alle Merkmale einbezogen, um den besten *Split* zu finden (*max_ features = None*). In Abbildung 1 wird der mit den Default-Einstellungen erzeugte Baum dargestellt. Es ist leicht zu erkennen, dass dieser Baum zu viele Knoten und Blätter enthält. Dem kann mit verschiedenen Einstellungen entgegengewirkt werden.

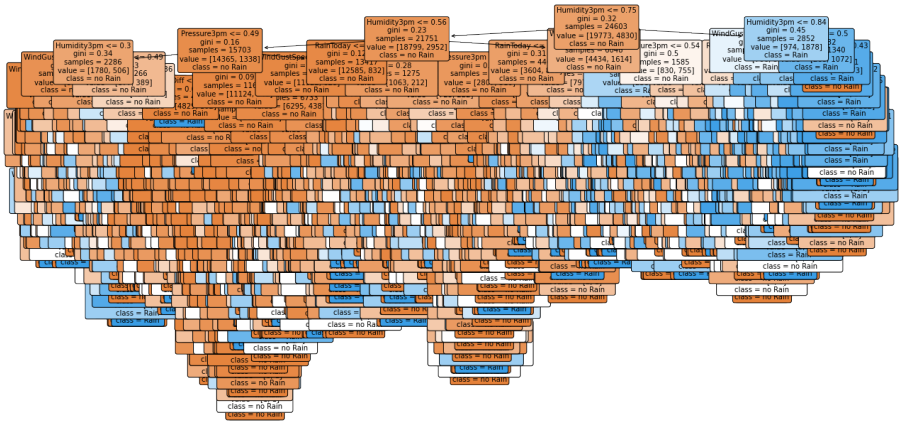

Abbildung 1: Entschidungsbaum mit Default-Einstellungen

1.2.2 Variationen

Um einen übersichtlichen und brauchbaren Entscheidungsbam erzeugen zu können werden verschiedenen Einstellungen angewandt. Im Folgenden werden ausgewählte Variationen dargestellt und kurz besprochen. Bei der Visualisierung liegt der Fokus auf der Struktur des Baumes und nicht darauf, dass die einzelnen Knoten identifiziert werden können.

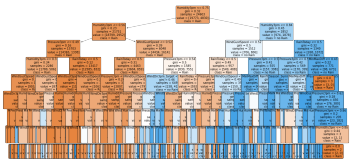

Abbildung 2: Maximale Tiefe von 8

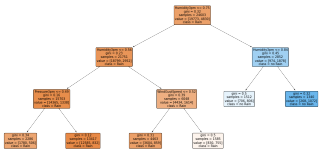

Abbildung 3: Minimale Unschärfe Reduktion von 0.003

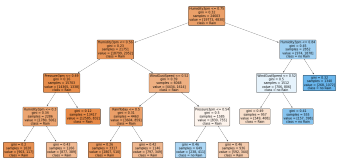

Abbildung 4: Maximale Blatt-Anzahl von 10

In Abbildung 2 wurde eine maximale Tiefe von acht angegeben. Mit dieser Anzahl an Stufen wurde eine bessere Genauigkeit (*accuracy*) und ein geringerer Testfehler erreicht als mit anderen ausprobierten Werten. Trotz der geringen Tiefe ist der Entscheidungsbaum schon sehr unübersichtlich in der Visualisierung. Im Gegensatz zu den anderen Variationen sieht dieser Baum sehr **symmetrisch**, da jeder Ast gleich lang verfolgt wird und der Baum nicht an unterschiedlichen Tiefen abgeschnitten wird.

In den Abbildungen 3 und 4 ist diese Symmetrie nicht mehr zu erkennen. Mit der Einstellung *min_ impurity_ decrease* wird festgelegt, welchen Anteil der Unschärfe ein weiterer Split reduzieren muss. Da das für jeden Knoten einzeln entschieden wird, werden nicht alle Äste bis zur selben Tiefe verfolgt. Bei der Einstellung der maximalen Anzahl an Blatt Knoten ...

1.2.3 Minimal Cost-Complexity-Pruning

2 Aufgabe 2: Neuronale Netze

2.1 Unterkapitel 1

2.1.1 Unterkapitel 1.1

3 Aufgabe 2: Ensemblemethoden

3.1 Unterkapitel 1

3.1.1 Unterkapitel 1.1

4 Aufgabe 4: Support Vector Machines

4.1 Unterkapitel 1

4.1.1 Unterkapitel 1.1

Anhang

Quellcode zu Aufgabe 1

Quellcode zu Aufgabe 2

Quellcode zu Aufgabe 3

Quellcode zu Aufgabe 4