Hochschule Bochum
Bochum University
of Applied Sciences

Fachbereich Wirtschaft
Prof. Dr. Christian Bockermann

Data Science 1

Sommersemester 2021

Hausarbeit

Die Prüfungsleistung zum Modul *Data Science 1* findet als Hausarbeit statt. Die Aufgabenstellung zur Hausarbeit finden Sie in diesem Dokument.

Für die Bearbeitung der Aufgabenstellung und die Erstellung Ihrer Hausarbeit steht wieder der Jupyter-Notebook Server zu Verfügung. Die Abgabe der Hausarbeit erfolgt dann als PDF-Export Ihres Jupyter-Notebooks. Das PDF Ihres Notebooks laden Sie als Lösung in der zugehörigen Aufgabe im Moodle Kurs hoch.
Andere Formen der Abgabe sind nicht vorgehen.

Für die Bearbeitung der Aufgabenstellung haben Sie ab Themenausgabe eine Woche Zeit. Der exakte Zeitraum wird in der zugehörigen Aufgabe im Moodle Kurs vermerkt.

Als Materialien können Sie sämtliche Unterlagen aus der Vorlesung und den Übungen mit benutzen, im Internet recherchieren oder weitere Bücher/Kurse mit verwenden. Geben Sie bitte bei Verwendung von umfangreicherem Programm-Code aus dem Netz (mehr als 3-4 Zeilen) die Quelle kurz mit an.

Aufgabe 1 (Python Basics)

Passend zu Tokio-2020 sind das Thema der diesjährigen Hausarbeit die Olympischen Spiele. Wir betrachten dazu eine Liste von Athleten der Olympischen Spiele von Rio 2016, die in folgendem Format vorliegt:

```
(id, name, sportart, geschlecht)
```

Die `id` ist die eindeutige Nummer für jeden Athleten, `name`, `sportart` und `geschlecht` sollten ja klar sein. Die Liste der Sportler bekommen Sie über die Funktion `athleten()` aus dem Paket `olympia`:

```
from olympia import athletes

athletes = athletes()
athletes[4]   #  (33922579, 'Aaron Gate', 'cycling', 'male')
```

Wie in dem Python Code zu sehen ist, hat beispielsweise der Athelete am Index 4 der Liste die folgenden Werte:

```
(33922579, 'Aaron Gate', 'cycling', 'male')
```

Für eine derartige Liste sollen Sie die folgenden Aufgaben lösen:

1. Bestimmen Sie die *verschiedenen* Sportarten in der Athleten-Liste!
   Schreiben Sie dazu eine Funktion `sportarten(xs)`, die für die Liste xs von Athleten, die Menge der Sportarten zurückgibt, die in der Liste enthalten sind.

   Das Ergebnis der Funktion soll also ein Liste oder Menge (`set`) sein, die jede Sportart nur einmal enthält.

2. Schreiben Sie eine Funktion `athletenFuerSportart(xs, sport)`, die für die Liste `xs` der Sportler und die Sportart `sport`, die Liste der Sportler mit der Sportart `sport` zurückliefert.

3. Schreiben Sie eine Funktion `anzahlNachSportart(xs)`, die die Liste `xs` der Sportler nach ihrer Sportart gruppiert und für jede Sportart die Anzahl der Athleten mit dieser Sportart berechnet.

   Die Liste soll also folgendes Format haben:

   ```
   [('aquatics', 1445), ('archery', 128), ('athletics', 2363),...]
   ```

4. Schreiben Sie eine Funktion `frauenanteil(xs, sport)`, die für die Liste `xs` und die Sportart `sport` den Frauenanteil berechnet.

5. Schreiben Sie eine Funktion `frauenanteilGesamt(xs)`, die für die Liste der Athleten den Frauenanteil für jede Sportart berechnet.

   Die Liste sieht also genauso aus wie in Aufgabe 1.3, enthält aber statt der Anzahl jetzt den prozentualen Anteil der Frauen in der jeweiligen Sportart.

Aufgabe 2 (Pandas und Statistiken)

In der Datei `Kurse/DataScience1/data/olympics-history.csv` sind die Teilnahmen von Sportlern an Olympischen Spielen seit 1896 enthalten. Dieser Datensatz soll mit Hilfe von Pandas im Folgenden untersucht werden.
Der Datensatz ist im wesentlichen selbsterklärend. In der folgenden Tabelle sind einige der wichtigsten Spalten exemplarisch dargestellt:

ID	Name	Sex	Age	NOC	Year	Season	Sport	Medal
1	A Dijlang	M	24.0	CHN	1992	Summer	Basketball	NaN
2	A Lamusi	M	23.0	CHN	2012	Summer	Judo	NaN
3	Gunnar Nielsen Aaby	M	24.0	DEN	1920	Summer	Football	NaN
4	Edgar Lindenau Aabye	M	34.0	DEN	1900	Summer	Tug-Of-War	Gold
5	Christine Jacoba Aaftink	F	21.0	NED	1988	Winter	Speed Skating	NaN
5	Christine Jacoba Aaftink	F	21.0	NED	1988	Winter	Speed Skating	NaN

Wie man in der letzten Zeile der Tabelle sieht, sind Sportler, die in einer Sportart an mehreren Disziplinen teilgenommen haben mehrfach enthalten. D.h. jede Zeile beschreibt die Teilnahme ein einer Disziplin. Die Spalte `NOC` enthält die Nation, für die der Athlet an dem jeweiligen Wettbewerb teilgenommen hat.
Die Spalte `Medal` gibt an, welche Medaille der Sportler bei der Teilnahme gewonnen hat. Bei keinem Medaillengewinn enthält die Spalte den Wert NaN (*not a number*).

1. Zunächst sollen ein paar generelle Informationen berechnet werden:
   - Wieviele Sportler enthält der Datensatz?
   - Wieviel Ausgaben Olympischer Sommerspiele enthält der Datensatz?
   - Wie hoch ist die Frauenquote der Winterspiele von 1988?
   - Welcher Athlet ist der Jüngste Teilnehmer über alle Spiel? Welcher der Älteste?
   - Sind die Athleten der Winterspiele im Schnitt jünger als die der Sommerspiele?

2. Erstellen Sie einen Datensatz/Dataframe `teilnehmerAnzahl`, der die Spalten `NOC`, `Year`, `Season` und `AnzahlAthleten` enthält.

   Plotten Sie mit Hilfe des Datensatzes den Verlauf der Anzahl der Athleten für die Sommer und Winterspiele.

   Hinweis: Hier hilft Ihnen die `groupby` Funktion von Pandas in Kombination mit einer Aggregierungsfunktion. Zum aggregierten Zählen eindeutiger Werte (z.B. der Spalte `ID`), bietet Pandas für Spalten die `nunique()` Methode an.

3. Mit der Pandas-Funktion `pd.get_dummies(df, columns=Spalten)` kann eine nicht-numerische Spalte (Liste von Spalten) in 0/1-Spalten überführt werden. Das Ergebnis ist ein neuer DataFrame.

   Wandeln Sie mit `pd.get_dummies(..)` die Spalte `Medal` in 0/1-Spalten um. Berechnen mit dem daraus entstandenen DataFrame den Länder-Medallienspiegel für die Olympischen Sommerspiele von 2016.

4. Um den Länder-Medallienspiegel zu sortieren, gibt es die DataFrame-Method `sort_values(spalte)`. Um nach Gold/Silber/Bronze zu sortieren, können wir eine neue Spalte `Rang` berechnen, die die gewichtete Summe der Anzahl der Bronze-, Silber- und Gold-Medallien darstellt.

   Nehmen Sie als Gewicht für Gold den Wert 10000, für Silber 100 und für Bronze das Gewicht 1. Ordnen Sie den Medallienspiegel anschließend nach ihem neu-berechneten Rang.

Aufgabe 3 (Modell-Training)

Eine Schattenseite von großen Sportereignissen sind leider immer wieder vorkommende Dopingfälle. Dazu werden die Athleten auf die Einnahme verbotener Substanzen kontrolliert. Die Kontrollen führt die WADA (Welt Anti Doping Agentur) mit akkreditierten Laboren durch.
In der Datei `Kurse/DataScience1/data/doping-tests.csv` finden Sie einen **fiktiven** Datensatz mit den Ergebnissen von Urin- und Blutproben. Die Daten enthalten Meßwerte zu verschiedenen Aspekten einer Probe.
Die Spalte `Ergebnis` enthält die Werte `POS` oder `NEG` und gibt den Status der Probe an. Sie sollen im Folgenden ein Modell trainieren, dass anhand der gemessenen Werte einer Probe die Vorhersage der Spalte `Ergebnis` ermöglicht.

1. Laden Sie die Daten in einen DataFrame und geben Sie die Anzahl der Datensätze, sowie die Anzahl der positiven und negativen Proben an.

   Betrachten Sie die Spalten und Datentypen der Spalten und überlegen Sie sich, welche Spalten Sie für ein Vorhersagemodell verwenden wollen.

2. Wie hoch ist der Fehler eines Modells, das immer nur die Klasse `NEG` vorhersagt?

3. Trainieren Sie ein Entscheidungsbaum-Modell auf dem vollständigen Datensatz. Welchen Trainingsfehler erreicht ihr Modell?

4. Teilen Sie die Daten in Trainings- und Test-Daten auf und verwenden Sie dabei 80% der Daten zum Training und den restlichen Teil für das Testen.

5. Bestimmen Sie den Parameter `max_depth`, der auf den Daten für ein Entscheidungsbaum-Modell den besten Generalisierungsfehler liefert. Testen Sie für die Bestimmung des Parameters `max_depth` die Werte im Bereich von 1 bis 20.

   Erzeugen Sie dazu einen DataFrame, der den Parameter `max_depth`, den Trainings- und den Test-Fehler enthält.

6. Erstellen Sie einen Plot mit dem Parameter `max_depth` auf der x-Achse, der den Trainings- und Test-Fehler für die verschiedenen Werte von `max_depth` zeigt. Für welches `max_depth` bekommen Sie das beste Modell?