Hochschule Bochum
Bochum University
of Applied Sciences

Fachbereich Wirtschaft
Prof. Dr. Christian Bockermann

Data Science 1

Wintersemester 2021/2022

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.

Bearbeitungszeit und Abgabe

Das Thema wird am 9.2.2022 um 8 Uhr im virtuellen Vorlesungsraum vorgestellt. Die Hausarbeit kann dann bis zum 18.2.2022 um 23:59 Uhr erfolgen.
Die Abgabe erfolgt als PDF-Datei (Export des Jupyter-Notebooks) per E-Mail an den Dozenten. Das PDF (bzw. Jupyter-Notebook) muss dazu alle Namen und Matrikelnummern der bearbeitenden Studierenden enthalten. Für Studierende, die die Hausarbeit als Gruppe bearbeiten (maximal 3 Studierende in einer Gruppe), reicht es, **eine** PDF Datei per Mail zu schicken.

Aufgabe 1 (Python Basics)

Als Kontext für die diesjährige Hausarbeit nutzen wir das Thema Corona – im Speziellen: Corona und die Auswirkungen auf die Kinderbetreuung.
Nach den geltenden Regeln müssen für Kinder in KiTas regelmässig Testnachweise abgegeben werden, die von der KiTa-Leitung verwaltet werden müssen. Dazu gibt es einen Datensatz[1] der fiktiven *Kita Regenbogen*, in dem für jeden Nachweis die Nummer, der Name des Kindes, die KiTa-Grupe und das Test-Datum enthalten ist.
Das bedeutet, der Datensatz ist eine Liste von Tupeln, die folgendes Format haben:

```
(testNr, name, gruppe, datum)
```

Die `testNr` ist eine fortlaufende Nummerierung der Tests, `name` ist der Name des Kindes, das getestet wurde. Die `gruppe` ist der Name der Gruppe, in der sich das Kind befindet und `datum` enthält einen String im Format `Jahr-Monat-Tag`.

```
from kita import testNachweise

tests = testNachweise()
tests[4]   #  (4, 'Liam', 'blau', '2022-01-18')
```

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

```
(4, 'Liam', 'blau', '2022-01-18')
```

Für eine derartige Liste sollen Sie die folgenden Aufgaben lösen:

1. Bestimmen Sie die *verschiedenen* Gruppen, die in der KiTa vorhanden sind! Schreiben Sie dazu eine Funktion `gruppen(xs)`, die für die Liste xs von Tests, die Menge der Gruppen zurückgibt, die in der Liste enthalten sind.

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

2. Schreiben Sie eine Funktion `testsInGruppe(xs, gruppe)`, die für die Liste `xs` der Tests und die Gruppe `gruppe`, die Liste der Tests mit der Gruppe `gruppe` zurückliefert.

3. Schreiben Sie eine Funktion `anzahlTestsInGruppen(xs)`, die die Liste `xs` der Testnachweise nach der KiTa-Gruppe gruppiert und für jede Gruppe die Anzahl der Tests berechnet.

   Die Liste soll also folgendes Format haben:

   ```
   [('blau', 214), ('gruen', 321), ('rot', 284),...]
   ```

4. Schreiben Sie eine Funktion `testsInGruppeDatum(xs,gruppe,datum)`, die eine Liste der Tests aus `xs` zurückgibt, die zur Gruppe `gruppe` gehören und das Datum `datum` enthalten.

[1] Die Daten sind zufällig generiert, die Namen kommen aus der Liste der beliebtesten Geburtsnamen von 2017 der Stadt Köln.

5. Das Modul `kita` enthält zudem die Funktion `kinderAusGruppe(gruppe)`, die die Liste der Kinder für eine Gruppe zurückliefert. Diese können Sie mit

   ```
   from kita import kinderAusGruppe
   ```

   importieren. Schreiben Sie eine Funktion `alleGetestet(xs, gruppe, datum)`, die `True` zurückgibt, wenn für alle Kinder der Gruppe `gruppe` ein Test vorliegt, und sonst `False`.

   Hinweis: Sie können der Einfachheit halber annehmen, dass für jedes Kind und Datum maximal ein Test vorliegt. Es reicht also, die Anzahl der Tests für Datum und Gruppe mit der Anzahl der Kinder für die Gruppe zu vergleichen.

   Zum Testen können Sie die Gruppe `blau` und das Datum `2022-01-26` nehmen. Dort sind die Tests vollständig. Für `blau` und `2022-01-27` gibt es fehlende Tests.

Aufgabe 2 (Pandas und Statistiken)

Nachdem wir uns in Aufgabe 1 mit den Testnachweisen beschäftigt haben, ist es natürlich auch wichtig, sich einen Überblick über die Entwicklung der aktuellen Corona-Lage zu verschaffen. Das RKI veröffentlicht dazu ja täglich die Infektionszahlen.
Die Datei `Kurse/DataScience1/data/rki-nrw-2022.csv` enthält einen aktuellen Datensatz der NRW Landkreise, der in dieser Aufgabe etwas exploriert werden soll. Der Datensatz enthält im Unterschied zu den originalen RKI-Daten lediglich die folgenden Spalten:

Bundesland	Landkreis	Meldedatum	Altersgruppe	Geschlecht	AnzahlFall
Nordrhein-Westfalen	SK Düsseldorf	2020-03-31	A00-A04	W	1
Nordrhein-Westfalen	SK Düsseldorf	2020-04-02	A00-A04	M	2
Nordrhein-Westfalen	SK Düsseldorf	2020-04-05	A00-A04	M	1
Nordrhein-Westfalen	SK Düsseldorf	2020-04-10	A00-A04	M	1
Nordrhein-Westfalen	SK Düsseldorf	2020-04-22	A00-A04	W	1

1. Zunächst sollen ein paar generelle Informationen berechnet werden:
   - Wieviele Einträge enthält der Datensatz insgesamt?
   - Wieviele männliche/weibliche Fälle wurden dem RKI insgesamt gemeldet?
   - Über welchen Zeitraum erstrecken sich die Daten? (Beginn/Ende?)
   - Welche Altersgruppen werden erfasst?

2. Welche Werte sind in der Spalte `Landkreis` enthalten? Welcher Wert gehört zur Stadt Bochum?

   Erstellen Sie einen Dataframe `bo`, der die Spalten `Meldedatum`, `Altersgruppe`, und `AnzahlFall` enthält. Plotten Sie mit Hilfe des Datensatzes den Verlauf der Anzahl der Fälle für die Stadt Bochum.

   Hinweis: Es ist hilfreich, die Spalte `Meldedatum` in einen Datumstyp umzuwandeln. Dies können Sie mit der Pandas Funktion `pd.to_datetime(..)` machen, die eine neue Spalte mit Datumswerten zurückliefert.

   Ausserdem hilft Ihnen die `groupby` Funktion von Pandas in Kombination mit einer Aggregierungsfunktion (`..sum()`) um z.B. die Fälle pro Tag aufzusummieren.

3. Für die *KiTa Regenbogen* ist natürlich insbesondere der Pandemie-Verlauf für die kleinen Kinder interessant. Erstellen Sie einen Plot, der die gemeldeten Fälle pro Tag für die unterschiedlichen Altersgruppen anzeigt.

   Hinweis: Wenn Sie für einen DataFrame mehrere `.plot()` Aufrufe in einer Code-Zelle ausführen, dann werden die Plots in einem Diagramm dargestellt. Ansonsten sind für die Aufgabe auch Einzelplots je Altersgruppe möglich.

4. Vergleichen Sie die Infektionszahlen der Altersgruppe 0-4 der Städte Bochum, Dortmund und Unna. Was fällt Ihnen dabei auf?

Aufgabe 3 (Modell-Training)

In der Vorlesung hatten wir als Anwendungsbeispiel für KI auch die Analyse von Husten-Aufnahmen für die Erkennung von Covid-19 Erkrankungen kurz vorgestellt. Das Verfahren beruht darauf, Eigenschaften des Hustengeräusches zu analysieren und dann das Geräusch als `covid19` oder `nicht-covid19` zu klassifizieren.

Wir betrachten im Folgenden einen fiktiven Datensatz, der Eigenschaften aus den Tonaufnahmen und die zugehörige Klassifikation enthält. Die Spalte `Label` enthält die zugehörige Klasse für die entsprechende Zeile (`covid19` oder `none`).

Sie finden die Datei unter `Kurse/DataScience1/data/covid-audio.csv`

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 `covid19` 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?

7. Trainieren Sie ein Modell mit dem – Ihrer Meinung nach – besten Wert für `max_depth`. Nutzen Sie dafür den Trainingsdatensatz und bestimmen Sie für Ihren gewählten Parameter-Wert den Generalisierungsfehler.

   Berechnen Sie für das trainierte Modell die Confusion Matrix und vergleichen Sie diese mit den Angaben, die in aktuellen Antigen-Tests angegeben werden (siehe Abbildung 1, letzte Seite).

   Wie gut schneidet Ihr Modell im Vergleich dazu ab?

【LEISTUNGSMERKMALE】
Klinische Leistung

Methode		PCR-Komparator (nasopharyngeale Abstrichprobe)		Gesamt
COVID-19 Antigen-Schnelltest-Kit (kolloidales Gold) (Speichelprobe)	Ergebnisse	Positiv	Negativ	
	Positiv	196	0	196
	Negativ	7	474	481
	Gesamt	203	474	677
Sensitivität (Rate der echten positiven Ergebnisse): 96,55 % (95 % CI, 93,05 % ~98,32 %) Spezifität (Rate der echten negativen Ergebnisse):>99 % (95 % CI, 99,20 % ~100,00 %) Genauigkeit (echt positive und negative Rate): 98,97 % (95 % CI, 97.88% ~99.50%)				

Nachweisgrenze: 5×10^2 $TCID_{50}$/ml

Figure 1: Konfusionsmatrix eines COVID-19 Antigen-Schnelltest-Kits.