Hochschule Bochum
Bochum University
of Applied Sciences

Fachbereich Wirtschaft
Prof. Dr. Christian Bockermann

Data Science 1

Sommersemester 2024

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 muss bis spätestens 23:59 Uhr am 11.8.2024 in der zugehörigen Aufgabe im Moodle Kurs hochgeladen werden.
Andere Formen der Abgabe sind nicht vorgehen.

Die Hausarbeit kann in Gruppenarbeit von bis zu drei Personen bearbeitet werden. In diesem Falle genügt *eine* Abgabe.

In jedem Fall sind in der Hausarbeit am Anfang des Notebooks die Namen und Matrikelnummern aller daran beteiligten Personen zu vermerken (gilt auch für Einzelabgaben).

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.

Die Verwendung von ChatGPT oder ähnlichen Hilfsmitteln ist nicht gestattet. Die Prüfungsordnung sieht für Verdachtsfälle die Möglichkeit mündlicher Nachprüfungen vor.

Aufgabe 1 (Python Basics)

Während eines Studiums hinterläßt jeder Studierende eine Reihe von Spuren an der jeweiligen Hochschule. Dazu gehören unter anderem Prüfungsleistungen, die in Systemen wie HISinOne (Studierendenportal) gespeichert werden.
Der Bereich *Learning Analytics* beschäftigt sich mit der Analyse der Prüfungsdaten um z.B. mögliche Studienabbrecher frühzeitig zu erkennen und Hilfestellungen zu geben. Auch die Lehrenden könnten mit Hilfe solcher Daten erkennen, in welchen Bereichen die eigenen Vorlesungen noch verbessert werden müssen.
Im Folgenden betrachten wir eine Liste von Prüfungsleistungen (Tests), die über das Modul `la` mit der Funktion `pruefungen()` abgerufen werden können. Jedes Element dieser Liste ist ein Tupel mit den folgenden Werten:

```
(Vorlesung, PruefArt, Tag, Gewicht, Matrikelnr, Punkte, AbgabeTag)
```

Hier ist ein kleines Beispiel, wie die Daten zu benutzen sind:

```
import la

ergebnisse = la.pruefungen()

pruefung_nr50 = ergebnisse[50]
#  ("Wirtschaftsinformatik", "Uebungsaufgabe", 19.0, 10.0,
                                  1967783, 70.0, 16)
```

Wie in dem Python Code zu sehen ist, hat die Prüfung am Index 50 der Liste die Werte:

```
("Wirtschaftsinformatik", "Übungsaufgabe", 19.0, 10.0, 1967783, 70.0, 16)
```

Das heisst, dass die Prüfung *Wirtschaftsinformatik* von dem/der Studierenden mit der Matrikelnummer *1967783* am 19ten Tag nach Kursbeginn fällig war. Die Aufgabe (Übungsaufgabe) wurde 16 Tage nach Kursbeginn abgegeben. Die erreichte Punktzahl beträgt 70% und macht einen Anteil von 10% an der Gesamtnote für diesen Kurs aus.

Für eine derartige Liste sollen Sie die folgenden Aufgaben lösen:

1. Schreiben Sie eine Funktion `studierende(liste)`, die als Parameter die obige Liste bekommt und die Menge der Matrikelnummern zurückgibt, für die es Einträge in der Liste gibt. Dabei soll jede Matrikelnummer nur einmal in der Ergebnisliste vorkommen.

2. Schreiben Sie eine Funktion `kurse(liste)`, die als Parameter die obige Liste bekommt und die Menge der Kurse (Namen des Kurses) zurückgibt, die in der Liste auftauchen. Dabei soll jeder Kurs nur einmal auftauchen.

3. Schreiben Sie eine Funktion `kurse_von(liste, matrikelnr)`, die für die gegebene Liste und Matrikelnummer die Menge aller Kurse (Namen) zurückliefert, für die der/die Studierende mit der angegebenen Matrikelnummer Prüfungen/Tests abgelegt hat.

4. Schreiben Sie eine Funktion (Name der Funktion beliebig), die zählt, wie viele Studierende Tests/Übungsaufgaben in mehr als einem Kurs abgelegt haben.

5. Schreiben Sie eine Funktion `kurs_ergebnisse1(liste, kurs)`, die eine Liste der Gesamtpunkte für jede/jeden Studierenden ermittelt, die in den Prüfungen zu diesem Kurs gesammelt wurden. Die Liste soll folgendes Format haben:

   ```
   [ (matrikelnr, summePunkte), ...]
   ```

6. Schreiben Sie eine Funktion `kurs_ergebnisse_gewichtet(liste, kurs)`, bei der wieder eine Liste von Matrikelnummern und Punktzahlen ermittelt wird. Allerdings sollen die Punkte dieses Mal nicht einfach aufsummiert werden, sondern jeweils mit dem prozentualen Anteil gewichtet werden. Es soll also die gewichtete Summe von Punkten und Anteil je Prüfungsteilnehmer ermittelt werden.

   Hinweis: Beachten Sie, dass die Gewichte in der obigen Liste in Prozentpunkten angegeben ist.

7. Schreiben Sie eine Funktion `avg_abgabezeit(liste)`, die für alle Studierenden die durchschnittliche Bearbeitungszeit für Übungsaufgaben ermittelt. Die Tests vom Typ `Quiz` sollen dabei nicht betrachtet werden.

   Die Bearbeitungszeit einer Übungsaufgabe ergibt sich aus der Differenz zwischen dem Prüfungstag und dem Abgabetag.

8. Schreiben Sie eine Funktion `kurs_summary(liste, kurs)`, bei der für den angegebenen Kursnamen eine Liste von allen Teilnehmern mit der angegebenen Anzahl von Tests, der durchschnittlichen Punktzahl, sowie der gewichteten Summe der Punkte in diesem Kurs.

   Die Liste soll also folgendes Format haben:

   ```
   [ (matrikelnr, anzahlTests, durchschnittPunktzahl, gewSummePunkte), ...]
   ```

9. Erweitern Sie die Funktion vorherige zur Funktion `kurs_summary2(liste, kurs)`, so dass zusätzlich für jeden Studierenden des Kurses die durchschnittliche Bearbeitungszeit der Übungsaufgaben mit ausgegeben wird.

Aufgabe 2 (Pandas und Statistiken)

In dieser Aufgabe geht es um Statistiken zum Lernverhalten und zu Prüfungsergebnissen. Gibt es Prüfungen, die häufig wiederholt werden? Wie ist das An- und Abmeldeverhalten bei bestimmten Prüfungen?
Prüfungsleistungen können dabei Klausuren, bearbeitete Übungsaufgaben sein oder die Teilnahme an einem Quiz sein.
Sämtliche Dateien zu dieser Aufgaben finden Sie im Verzeichnis

`Kurse/DataScience1/data/learning_analytics`

auf dem Notebook Server. Als Grundlage dient eine Tabelle mit Prüfungsergebnissen, die Sie in der Datei `ergebnisse.csv` finden:

kurs	pruefung	matrikelnr	pruefungsart	datum	abgabedatum	punktzahl	gewichtung
16	34877	596938	Übungsaufgabe	173	173.0	94	25
18	34870	1534560	Quiz	222.0	193.0	86.0	0
18	34863	542506	Übungsaufgabe	131.0	128.0	82.0	25
19	34887	2256318	Übungsaufgabe	52.0	53.0	74.0	12
9	25348	440003	Übungsaufgabe	25.0	24.0	37.0	10
20	37418	590593	Quiz	229.0	91.0	100.0	0
18	34862	554940	Übungsaufgabe	89.0	88.0	75.0	25
6	15019	620507	Quiz	194.0	200.0	100.0	1
5	14989	548617	Übungsaufgabe	187.0	187.0	91.0	18
22	37428	632569	Quiz	222.0	101.0	100.0	0

Table 1: Die Datei `ergebnisse.csv`

Die Spalten *kurs*, *matrikelnr* und *pruefungsart* enthalten die ID des Kurses, in dem die Prüfung abgelegt wurde, die Matrikelnummer des Studierenden und die Art der Prüfungsleistung. *pruefung* ist die ID der Prüfung. Die Spalten *datum* und *abgabedatum* enthalten den Zeitpunkt der Prüfung bzw. das Abgabedatum. Die Daten sind als Anzahl der Tage seit Beginn des Kurses angegeben.
Zusätzlich sind die erreichte *punktzahl* und die *gewichtung* der Prüfung mit vermerkt. Die erste Zeile zeigt also, dass der Student 596938 die Übungsaufgabe in Kurs 16 direkt am Tag der Ausgabe absolviert und dabei 94 Punkte erreicht hat. Die Prüfungsleistung zählt zu 25% in der Gesamtleistung dieses Kurses.
Klausuren werden separat gewertet (siehe Zeile 2) und haben entsprechend eine Gewichtung von 100% zur Notengebung.

Studierenden-Daten

Spannend ist in diesem Szenario natürlich die Frage, ob die Teilnahme an Übungsaufgaben zu besseren Prüfungsleistungen in der Klausur führen. Auch sozio-demographische Faktoren können für die Bewertung und Verbesserung von Lehre intessant sein.
In der Datei `studierende.csv` finden sich Informationen zu den Kursteilnehmern und deren Abschneiden (finale Prüfungsleistung bestanden oder nicht), sowie Daten zum Alter, Wohnort und der Anzahl vorangegangener Versuche.
Der Studierende 278272 aus Zeile 4 z.B. hat den Kurs Nr. 3 nicht bestanden, hat zuvor aber schon einen Versuch unternommen (`versuche = 1`).

matrikelnr	kurs	gender	alter	stadt	versuche	ergebnis
560399	19	M	29	Dortmund	0	bestanden
2652268	19	F	28	Bochum Mitte	0	nicht bestanden
592081	3	F	22	Herne	0	nicht bestanden
278272	3	M	26	Bochum	1	nicht bestanden
1420537	16	M	25	Bochum Mitte	0	bestanden
684944	7	M	20	Bochum Mitte	0	abgemeldet
1895399	17	M	25	Bochum Mitte	0	abgemeldet
2106955	17	F	26	Witten	0	Auszeichnung
624069	6	F	18	Dortmund	0	bestanden
680723	4	F	25	Langendreer	0	Auszeichnung

Table 2: Die Tabellenstruktur in der Datei `studierende.csv`.

Kurs-Daten

Ein *Kurs* ist in diesem Sinne immer ein Modul, das in einem bestimmten Semester durchgeführt wird. In der Datei `kurse.csv` gibt es eine Aufstellung der Kurse, über die die Prüfungsdaten erhoben wurden.

id	modul	name	semester
8	3	Beschaffung und Logistik	S2014
4	2	Strategisches Marketing	W2014
7	3	Beschaffung und Logistik	W2014
16	6	Finanzmanagement	W2013
18	6	Finanzmanagement	S2013
19	6	Finanzmanagement	S2014
17	6	Finanzmanagement	W2014
13	5	Volkswirtschaftslehre	W2013

Table 3: Die Struktur der Tabelle `kurse.csv`.

Prüfungsdaten

Neben den Prüfungsdaten der Studierenden enthält die Datei `pruefungen.csv` eine Aufstellung aller Prüfungen, die in einem Kurs stattfinden. Die nachfolgenden Tabelle zeigt einen kurzen Ausschnitt daraus:

id	kurs	prüfungsleistung	datum	gewichtung
34895	19	Quiz	227.0	0
24284	8	Übungsaufgabe	151.0	22
37425	22	Übungsaufgabe	61.0	0
34911	17	Klausur	241.0	100
34898	19	Klausur	227.0	100
15014	6	Klausur	nan	100

Table 4: Die Struktur der Tabelle `pruefungen.csv`.

Sie sollen sich im Folgenden mit diesen Daten beschäftigen. Dazu ist es natürlich wichtig, sich erstmal einen Überblick zu verschaffen. Die Tabellen selbst sind über verschiedene Spalten miteinander verknüpfbar. So ist beispielsweise die Spalte `kurs` in der Regel ein Fremdschlüssel, der auf die `id` Spalte in der Tabelle `kurse.csv` zeigt. Leitende Fragestellungen im Kontext von Learning Analytics können z.B. sein:

- Führt die konsequente Bearbeitung von Übungsaufgaben zu besseren Lernerfolgen der Studierenden?
- In welchen Kursen werden die meisten Tests absolviert?
- Welche Altersgruppe ist am häufigsten vertreten?

Hintergrund dieser Aufgabe ist es, dass Sie sich mit einem unbekannten Datensatz vertraut machen und mit Hilfe von Pandas untersuchen, welche Informationen aus den Daten herausgesucht werden können.

Die Aufgaben:

1. Zunächst sollen ein paar generelle Informationen berechnet werden:
   - Welche Spalten/Datentypen haben die Datensätze?
   - Wieviele Prüfungen werden jedes Semester abgelegt?
   - Wie ist die durchschnittliche Punktzahl für die Klausuren für die Kurse?
   - Wie ist die Altersstruktur der teilnehmenden Studierenden?
   - Wie ist die Verteilung der Geschlechter bzgl. der Kurse? Hat jeder Kurs das gleiche männlich/weiblich Verhältnis?
2. Teilen Sie die Studierenden in Altersgruppen ein. Nutzen Sie dafür die Gruppierungen 10-20, 20-30, 30-40, usw. (alternativ können Sie auch eine eigene Altersgruppierung wählen).

   Wie ist die Altersstruktur der Studierenden? Erstellen Sie einen Histogram-Plot dazu!
3. Wie ist die durchschnittliche Bearbeitungsdauer der Übungsaufgaben je Kurs? Gibt es Unterschiede bzgl. der Bearbeitungsdauer zwischen den Altersgruppen?

   Hinweis: Schauen Sie sich dazu z.B. die Pandas Funktion `cut` an (siehe Pandas Dokumentation).
4. Wir haben in der Vorlesung verschiedene Lernverfahren kennengelernt. In der Tabelle `studierende.csv` ist vermerkt, welche Teilnehmer einen Kurs bestanden haben oder durchgefallen sind. Der Wert "Auszeichnung" steht dabei für eine sehr gute Bestehensleistung.

   Überlegen Sie sich, wie Sie aus den verschiedenen Tabellen einen Datensatz bauen können, der das Lernverhalten der Studierenden, sowie deren soziodemographische Daten enthält und der für die Vorhersage, ob ein Kurs bestanden wird oder nicht, genutzt werden kann.

Schreiben Sie dazu zunächst auf, welche Idee Sie dafür haben, und wie Sie vorgehen wollen – unabhängig von der eigentlichen Implementierung.

5. Entwickeln Sie ein Notebook, dass ihr Vorgehen zur Vorhersage mit Hilfe eines SciKit-Learn Modells umsetzt. Welches Lernverfahren wollen Sie nutzen? Begründen Sie Ihre Auswahl.

Hinweis zur Bearbeitung

Die Vorlesung hat einige der Grundlagen zu Pandas vermittelt. Natürlich ist Pandas deutlich umfangreicher, als man es innerhalb einer 1-semestrigen Vorlesung vermitteln kann. Für die Lösung einiger dieser Aufgaben ist es daher erforderlich, sich weiter mit Pandas zu beschäftigen. Dazu gehört u.a. die Verbindung mehrerer Tabellen (JOIN), was Sie beispielsweise aus dem Bereich der Datenbanken in Wirtschaftsinformatik 2 bereits kennen sollten.
Die Dokumentation für den JOIN findet sich z.B. unter

`https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.join.html`

Es sei hier noch angemerkt, dass es hilfreich ist, wenn der *index* des DataFrames, den man an einen bestehenden DataFrame heften möchte für den JOIN relevant ist. So sollte z.B. der DataFrame für die Kunden als Index am besten die Kundennummer enthalten, bevor dieser an die Bestellungen ge-joint wird.

Es geht bei der Bearbeitung dieser Aufgaben nicht nur um die reine Programmierung in Python. Ziel ist es, die Daten entlang der Teilaufgaben zu analysieren und die Ergebnisse in einem gewissen Rahmen zu interpretieren.
Dazu gehört zu jeder Teilaufgabe, dass Sie kurz skizzieren, wie Sie vorgehen wollen, welche Teil-DataFrames sie ggf. berechnen wollen und was Sie am Ergebnis ggf. kritisch betrachten (z.B. Datenqualität, etc.). Auch dafür haben Sie in Data Science und Kursen wie Wirtschaftsstatistik Methoden und Werkzeuge kennengelernt.