WiSe 25/26

Data Science Programmierung mit Python

Prof. Amy Siu & Prof. Selcan Ipek-Ugay

Vorhersage von Prüfungsergebnissen

Können wir Prüfungsergebnisse eines Studierenden basierend auf dem akademischen Verhalten, Lerngewohnheiten, Lebensstil und den Prüfungsbedingungen vorhersagen?

GitHub (github.com/luongvkh/exam_score_prediction)

Streamlit (exam-score-prediction-luongvkh.streamlit.app)

Van Luong

102920

valu1138@bht-berlin.de

Inhaltsverzeichnis

Kernidee und Datenquelle

Die Forschungsfrage meines Projekts lautet: “Können wir Prüfungsergebnisse eines Studierenden basierend auf dem akademischen Verhalten, Lerngewohnheiten, Lebensstil und den Prüfungsbedingungen vorhersagen?” Dazu habe ich den Datensatz Exam Score Prediction Dataset des Kaggle-Users “kundanbedmutha” verwendet. Der Datensatz bietet insgesamt 20.000 Zeilen und 13 Spalten.

Für dieses Projekt wollte ich einen Datensatz finden, der nicht nur für mich, sondern auch für andere Personen interessant sein könnte. Meine Idee war es, mit meiner Datenanalyse für mich und mein Umfeld ein nützliches Ergebnis zu erzielen, welches ich am Ende in meine Streamlit-App integrieren kann. Dieses Projekt war also mein Versuch, für Studierende ein Tool zu entwickeln, womit man seine Prüfungsergebnisse vorhersagen kann.

Features

- age, gender, course of study, daily study hours, class attendance, study method, sleep duration, sleep quality, internet availability, facility rating, exam difficulty

Target Variable

- exam score percentage (1-100)

Datenbereinigung

Ein Hauptproblem bei meinem ursprünglichen Datensatz war, dass fast alle Spalten den Datentypen “object” hatten. Um dies zu beheben, habe ich die jeweiligen Spalten *Data Type Conversion* durchgeführt. Somit wurden aus den Spalten "gender", "course", "sleep_quality", "study_method", "facility_rating" und "exam_difficulty" categorical columns und “internet_access” ein Bool.

Zudem hatte die Spalte “course” indische Abschlüsse, die wir nicht gewohnt sind. Das Problem habe ich gelöst, indem ich wie folgt Mapping angewandt habe:

```
# Indian degree → German degree
degree_mapping = {
    'bca': 'B.Sc.',
    'ba': 'B.A.',
    'b.sc': 'B.Sc.',
    'b.com': 'B.A.',
    'bba': 'B.A.',
    'diploma': 'Diploma',
    'b.tech': 'B.Eng.'
}
```

Meine **Data Cleaning Pipeline** sieht folgendermaßen aus:

1. **Datensatz in Pandas laden**
2. **Daten inspizieren**
    - Hier wurde der Datensatz untersucht und die wichtigen Features wurden festgelegt

3. **Fehlende Daten**
    - In diesem Schritt wurden fehlende Werte gesucht und visualisiert. Es stellte sich heraus, dass der Datensatz keine fehlenden Werte hat.

4. **Datentyp-Konvertierung**
    - Hier wurden die Datentypen kontrolliert und bei Bedarf konvertiert.

5. **Duplikate erkennen und entfernen**
    - Wie in Schritt 3 gab es hier keine Duplikate, sodass keine Handlung erfordert wurde.

6. **Outlier behandeln**
    - Hier wurden Outlier gesucht mittels Statistiken, Boxplots und der IQR-Methode. Auch hier wurden keine gefunden, es wurde keine Handlung getätigt.

7. **Konsistenz und Standardisierung**
    - In diesem Schritt wurden alle kategorialen Spalten standardisiert.

8. **Finaler Check**
    - Vor dem letzten Schritt wurden die Dimension und der Aufbau des bereinigten Datensatzes geprüft, sowie erneut nach fehlenden Werten und falschen Datentypen geschaut.

9. **Bereinigten Datensatz speichern**

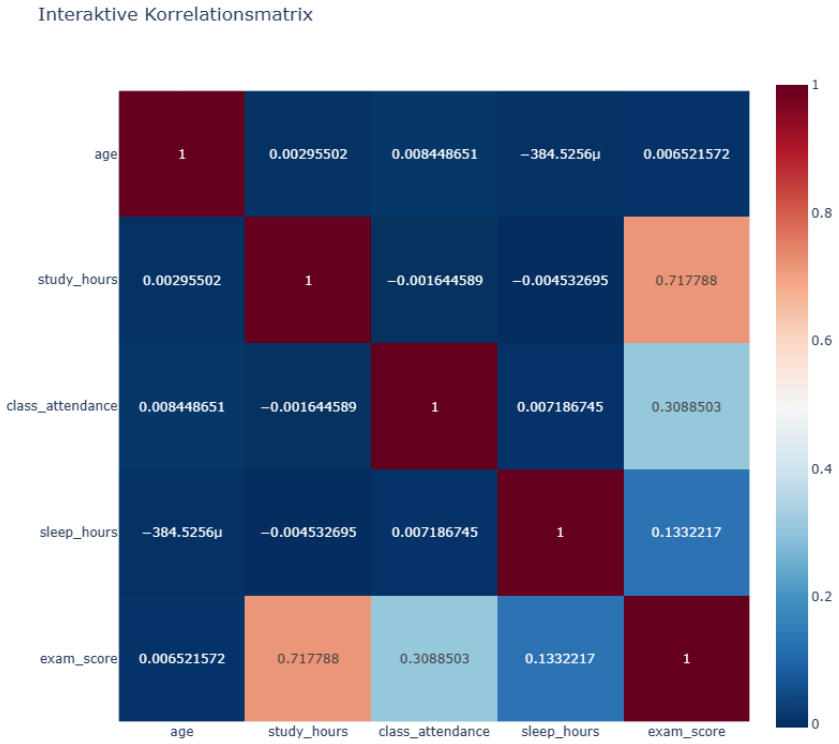
Insgesamt wurden keine Zeilen entfernt, da keine Zeile fehlerhaft oder ein Outlier ist. Der Datensatz war von Anfang an sehr sauber. Allerdings habe ich die Spalte “student_id” entfernt, da sie weder einen wirklichen Zweck für die Research Question geboten hat, noch unique war, weshalb sie nicht als Index dienen konnte.

Datenvisualisierung

Zur Datenvisualisierung habe ich Histogramme, Boxplots, Count Plots, Scatter Plots mit und ohne Farbkodierung, genau so wie Grouped Bar Charts, Korrelationsmatrizen, Heatmaps, Pairplots und Seaborn FacetGrids verwendet. Besonders aussagekräftig sind die interaktive Korrelationsmatrix und der interaktive Scatter Plot, da man daraus die wichtigsten Features extrahieren kann. Interessant ist aber auch das interaktive Histogramm, welches verschiedene Lernmethoden miteinander vergleicht.

Interaktive Korrelationsmatrix

Dieses Diagramm zeigt die Korrelationen zwischen den numerischen Attributen meines Datensatzes. Bezogen auf meine Target Variable "exam_score" kann man hieraus ziehen, dass nur study_hours, class_attendance und sleep_hours einen bedeutenden Einfluss auf den exam_score haben.

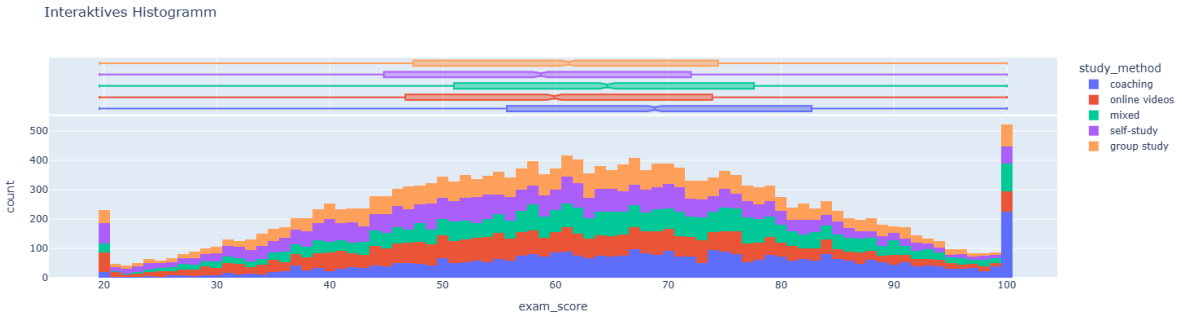


Interaktiver Scatter Plot

Hier sieht man den interaktiven Scatter Plot der Attribute study_hours, sleep_quality und exam_scores. Das Diagramm zeigt, dass die exam_scores besser werden, je mehr Stunden der Student lernt (study_hours) und je besser die Schlafqualität ist (sleep_quality).

Interaktives Histogramm

Im Bild sieht man das Interaktive Histogramm, welches die Verbindung zwischen study_method und exam_score zeigt. Zu erkennen ist, dass Coaching den besten durchschnittlichen Exam Score erbringt. Bei den Einträgen im Datensatz mit voller Punktzahl (100%) ist die study_method "coaching" am meisten vertreten, während diese Methode bei den Klausuren mit der niedrigsten Punktzahl (ca. 20%) am wenigsten vertreten ist.



Insgesamt ergaben sich folgende Erkenntnisse aus der Datenvisualisierung:

- Nur die numerischen Attribute study_hours, class_attendance und sleep_hours haben einen nennenswerten Einfluss auf die Target Variable exam_score.
- Aus den Boxplots nach Kategorien ergab sich, dass gender, course, internet_access und erstaunlicherweise die exam_difficulty kaum Einfluss auf den exam_score haben.
- Sleep_quality, study_method und facility_rating zeigen zwar eine kleine Korrelation zur Target Variable, aber jedoch in keiner bedeutenden Menge.

Machine Learning

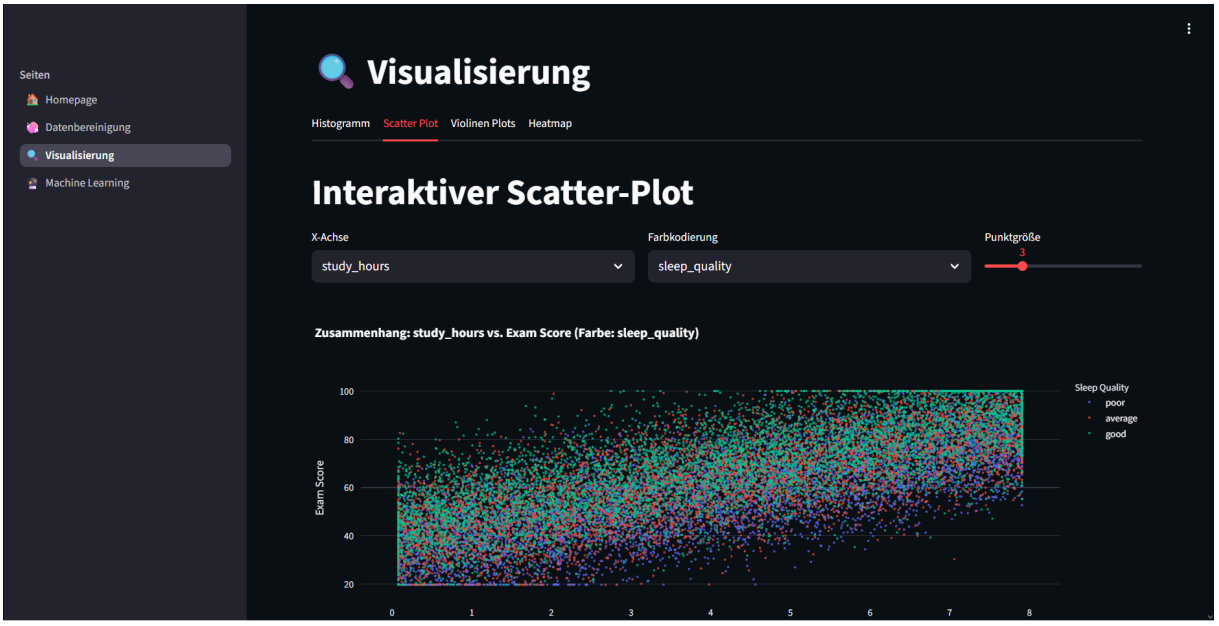
In meinem Projekt habe ich ein Linear-Regression Modell verwendet, um aus den Daten eines Studierenden eine möglichst akkurate Vorhersage seines/ihres Prüfungsergebnisses zu erzielen.

Dieses Modell kann man in der Streamlit App nutzen.

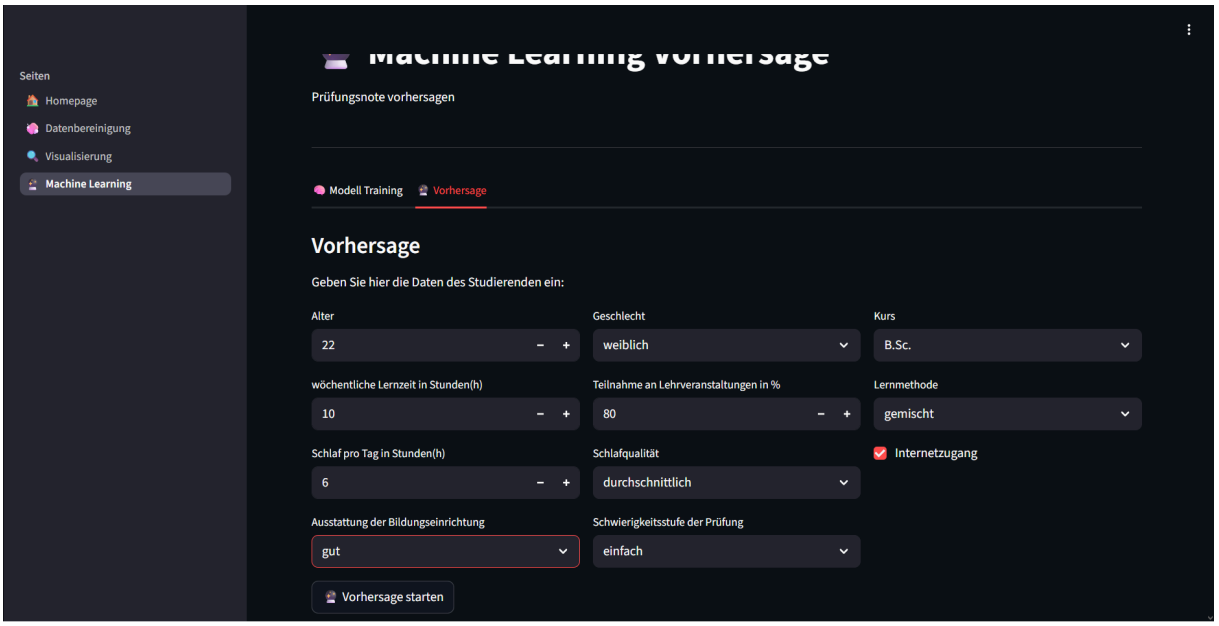
Streamlit-Anwendung

In meiner Streamlit-Anwendung erhält der Nutzer einen detailreichen Überblick über den Datensatz mitsamt den wichtigsten Infos, den Schritten aus der Datenbereinigung sowie auch einer umfangreichen Visualisierung mit interaktiven Histogrammen, Scatter Plots, Violinen Plots und Heatmaps. Auf der Machine-Learning-Seite kann der Benutzer Daten eingeben, um daraus ein Prüfungsergebnis vorhersagen zu lassen.

Zu den interaktiven Elementen meiner Anwendung zählen eine Sidebar, Slider, Dropdown-Menüs, Filter, Tabs, Selectboxes und einige der Visualisierung.



Das Bild zeigt die Visualisierungs-Page. Hier kann der Nutzer anhand von interaktiven Histogrammen, Scatter Plots, Violinen Plots und Heatmaps einen Einblick in den Datensatz schaffen.



Das Bild zeigt die Machine-Learning-Page. Hier kann der Nutzer Daten passend zu den Features des Datensatzes eingeben, um daraus mit Hilfe meines Linear-Regression-Models ein Prüfungsergebnis vorhersagen zu lassen.