

EINFLUSS VON GENERATIVER KI AUF DIE AKADEMISCHE LEISTUNG VON STUDIERENDEN

DataPy 25/26

Sigourney Reipsch 104871

Inhaltsverzeichnis

Kernidee und Datenquelle	1
Datenbereinigung	1
Datenvisualisierung	2
Machine Learning	3
Streamlit-Anwendung	3

Link zum Quellcode: https://github.com/Siney-MI/ai_student_analysis.git

Kernidee und Datenquelle

Das Ziel dieses Projekts ist es, den Zusammenhang zwischen der Nutzung von KI-Tools und dem Studienerfolg zu untersuchen.

Im Zentrum steht dabei die folgende Forschungsfrage: Wie beeinflusst die Nutzung von generativen KI-Tools das akademische Leistungsniveau von Studierenden, und welche Faktoren wirken sich dabei am stärksten positiv oder negativ auf die Noten aus?

Für die Analyse wurde der Datensatz "AI Tools Usage by Indian College Students 2025" verwendet.

Link: <https://www.kaggle.com/datasets/kshitijsaini121/ai-tools-usage-by-indian-college-students-2025>

Ich habe mich für dieses Thema entschieden, da es aktuell sehr relevant ist und mich persönlich interessiert. Im Studium beobachte ich zunehmend, dass immer mehr Studenten KI-Tools für ihre akademischen Aufgaben nutzen. Mich hat daher interessiert, anhand von Daten zu überprüfen, welchen tatsächlichen Einfluss diese Tools auf die Noten haben und ob sie die Leistung verbessern oder verschlechtern. Der gewählte Datensatz hat auf den ersten Blick meine Kriterien diesbezüglich erfüllt. Später musste ich jedoch feststellen, dass es sich mit hoher Wahrscheinlichkeit um einen synthetischen Datensatz handelt.

Datenbereinigung

Probleme in den Rohdaten:

Bei der ersten Inspektion fielen mehrere Probleme auf:

- Die Spalte State wies ca. 44 % fehlende Werte auf.
- Es gab inkonsistente Schreibweisen.
- Es wurden 3 identische Datensätze identifiziert.
- Gleiche Begriffe wurden unterschiedlich benannt, obwohl sie das Gleiche Ausdrücken (z.B. "Exam Prep" und "Exam Preparation").
- In den Spalten „AI_Tools_Used“ und „Use_Cases“ befanden sich mehrere Werte pro Zelle (Listen als Strings), was eine direkte Analyse verhinderte.

Um die Datenqualität für die Analyse sicherzustellen, wurden folgende Schritte unternommen:

- Die 3 Duplikate sowie für die Fragestellung irrelevante imSpalten wurden gelöscht.
- Die Textdaten wurden vereinheitlicht, indem ich alle Einträge in Kleinbuchstaben umgewandelt, Leerzeichen entfernt und Tippfehler korrigiert habe.
- Die Synonyme wurden zu einer Spalte zusammengefasst.
- Die numerischen Spalten wurden auf Ausreißer geprüft, aber glücklicherweise war keine Korrektur notwendig, weil es keine gab.

Herausforderungen:

Eine besondere Herausforderung stellte die Struktur der Listen-Spalten dar. Zunächst übersah ich, dass in „AI_Tools_Used“ und „Use_Cases“ mehrere Werte pro Zelle standen. Dies fiel erst bei der Visualisierung auf, was einen iterativen Rücksprung in die Data-Cleaning-Phase notwendig machte.

Um die Daten auswertbar zu machen, nutzte ich die „explode()-Funktion“. Dies erhöhte die Zeilenanzahl von ca. 3.600 auf über 9.000. Das führte jedoch zu einem neuen Problem. Die Zuordnung zum ursprünglichen Studenten ging verloren. Ein Filtern über „Student_Name“ war nicht möglich, da der Datensatz synthetische Merkmale aufweist (Namen wie "Aarav" kamen bis zu 90-mal für unterschiedliche Personen vor). Ich erstellte daher vor der Transformation eine eindeutige „Student_ID“ basierend auf dem Index. Dies ermöglichte es mir, die explodierten Daten später wieder korrekt zu gruppieren und präzise zwischen personenbasierten Analysen und interaktionsbasierten Analysen zu unterscheiden. Zusätzlich war es notwendig, einen separaten „use_cases“ Datensatz zwischenspeichern. Dies geschah, um sicherzustellen, dass die Anwendungsfälle in den späteren Analysen nicht falsch gewichtet werden.

Ergebnis: Nach der Bereinigung standen mir somit der explodierte Hauptdatensatz (9.357 Zeilen) für Detailanalysen sowie der spezialisierte Basis-DataFrame mit einzigartigen Studenten (3.611 Zeilen) für personenzentrierte Auswertungen zur Verfügung.

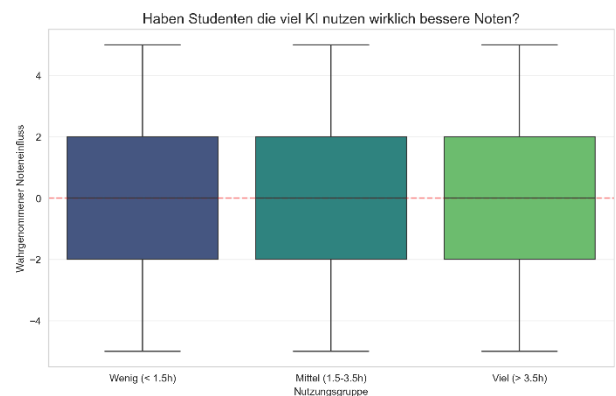
Datenvisualisierung

Um die Forschungsfrage zu beantworten, habe ich verschiedene Visualisierungsarten getestet, darunter Histogramme zur Verteilung, eine Korrelationsmatrix, gruppierte Boxplots für Vergleiche und Balkendiagramme für Kategorien.

Besonders bewährt haben sich Boxplots und Balkendiagramme. Ursprünglich hatte ich Scatterplots in Erwägung gezogen, diese erwiesen sich jedoch als ungeeignet, da die Datenpunkte zu stark überlappten und viele Variablen kategorialer Natur waren. Die Boxplots hingegen konnten die Verteilungen innerhalb der Gruppen sauber trennen. Die Visualisierungen lieferten überraschende Ergebnisse, die meiner ursprünglichen Intuition widersprachen.

- Die gruppierten Boxplots zeigten, dass die Nutzungsdauer keinen direkten Einfluss auf den wahrgenommenen Notenerfolg hat (Abb.1). Die Mediane der Gruppen lagen fast identisch auf der Nulllinie. Eine intensivere Nutzung garantiert also keine besseren Noten.

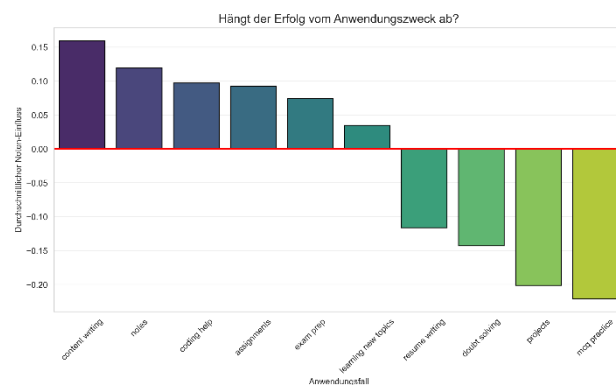
Abbildung 1 KI = bessere Noten



Während die Dauer irrelevant scheint, ist das Wie entscheidend. Die Analyse der Use-Cases zeigte eine Spaltung (Abb.2).

- Studierende, die KI für produktive und organisatorische Aufgaben nutzen, berichten von den deutlichsten Notenverbesserungen. Auch die allgemeine Prüfungsvorbereitung wirkt sich positiv aus.
- Studierende, die KI für spezifische Übungsformate oder das bloße Lösen von Verständnisfragen nutzen, verzeichnen eine Verschlechterung. Überraschenderweise korreliert auch die Nutzung für "Projects" stark negativ mit dem Notenerfolg.

Abbildung 2 Anwendungszweck



Was mich auch überraschte waren die fehlenden Korrelationen. Die Korrelationsmatrix zeigte keine signifikanten linearen Zusammenhänge zwischen Demografie (z. B. Studiengang) und Erfolg. Auch die

Wahl des KI-Tools (ChatGPT vs. Gemini vs. Claude) spielt eine untergeordnete Rolle, die Unterschiede sind minimal.

Herausforderungen:

Eine Herausforderung war die Visualisierung der Studiengänge und Anwendungsgründe, da diese keine klaren Muster erkennen ließen. Dies war der Auslöser für den oben beschriebenen „explode()-Schritt“. Kritisches Problem dabei war, dass dadurch eine statistische Verzerrung entstand. Ein Student, der 3 Tools für 3 Zwecke nutzte, tauchte 9-mal auf, wodurch seine Note fälschlicherweise 9-fach gewichtet wurde. Ich korrigierte dies, indem ich einen spezifischen Use-Case-DataFrame erstellte. In diesem wurde nur die Spalte Use_Cases aufgeteilt, während die Tools gruppiert blieben. Erst durch diesen methodischen Zwischenschritt konnte sichergestellt werden, dass jeder Anwendungsfall korrekt gewichtet in die Analyse eingeht und das oben beschriebene Muster valide sichtbar wird.

Machine Learning

Ursprünglich war geplant, den genauen Einfluss auf die Noten (Skala -5 bis +5) mittels Regression vorherzusagen. Da die explorative Datenanalyse jedoch zeigte, dass keine starken linearen Korrelationen vorliegen, habe ich meinen Ansatz zur binären Klassifikation geändert, um zu überprüfen ob sich Studierende anhand ihres Nutzungsverhaltens vorhersagen lassen, und ob sich ihre Noten verbessern (Klasse 1) oder nicht verbessern/verschlechtern (Klasse 0).

Das beste Ergebnis lieferte der K-Nearest Neighbors (KNN) Algorithmus, da das Modell eine Genauigkeit von 66,11 % erreichte.

Anhand der Confusion Matrix (Abb.3) erkennt man, dass das Modell Studierende ohne Leistungsverbesserung (Klasse 0) sehr zuverlässig (307 korrekte Treffer) erkennt. Allerdings gibt es eine hohe Anzahl an falschen Negativwerten (150). Das Modell übersieht häufig Studierende, deren Noten sich tatsächlich verbessert haben, und stuft sie fälschlicherweise als "nicht verbessert" ein.

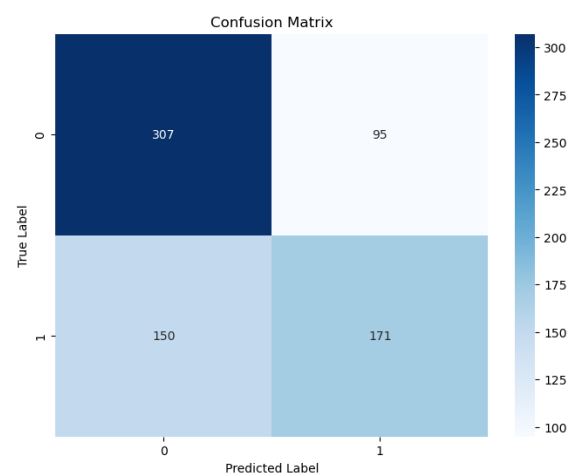


Abbildung 3 Confusions Matrix

Streamlit-Anwendung

Meine Streamlit-Anwendung ist in vier logische Bereiche unterteilt:

Startbildschirm: Dieser Bereich bietet Transparenz über die Datengrundlage. Nutzer erhalten einen direkten Einblick in die Struktur der Daten, indem die ersten 5 Zeilen der verschiedenen Verarbeitungsstufen (Original-Rohdaten, bereinigter Datensatz, explodierter Datensatz) gegenübergestellt werden. Zusätzlich werden wichtige Kennzahlen wie die durchschnittliche tägliche KI-Nutzungsdauer prominent angezeigt.

Daten-Exploration (Abb.4): Hier können Nutzer interaktiv mit dem Datensatz arbeiten. Über die Sidebar stehen globale Filter zur Verfügung: Ein Slider grenzt den Notenbereich ein, während ein Multiselect-Feld die Auswahl spezifischer Studiengänge erlaubt. Die gefilterte Tabelle aktualisiert sich automatisch und kann anschließend über einen Download-Button als CSV-Datei für die lokale Weiterverarbeitung exportiert werden.

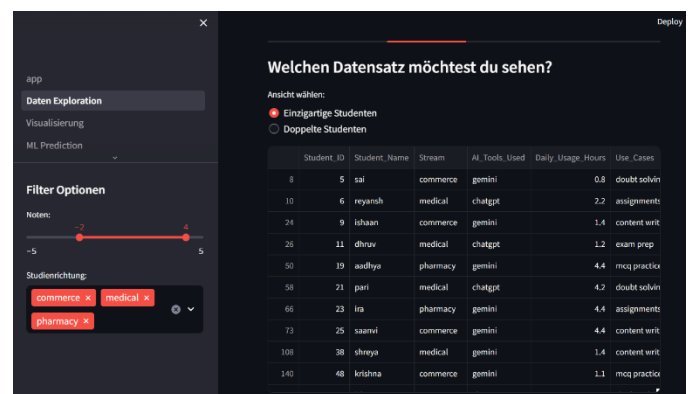


Abbildung 4 Daten Explorations Seite

Visualisierung (Abb. 5): Dieser Tab präsentiert die grafischen Auswertungen. Anstatt alle Grafiken gleichzeitig zu zeigen, wählt der Nutzer über ein Dropdown-Menü die gewünschte Analyse aus (z. B. Verteilungen oder Einflussfaktoren). Zu jeder Grafik gibt es eine kurze Analyse, die platzsparend in einem Expander verborgen ist und bei Bedarf aufgeklappt werden kann.

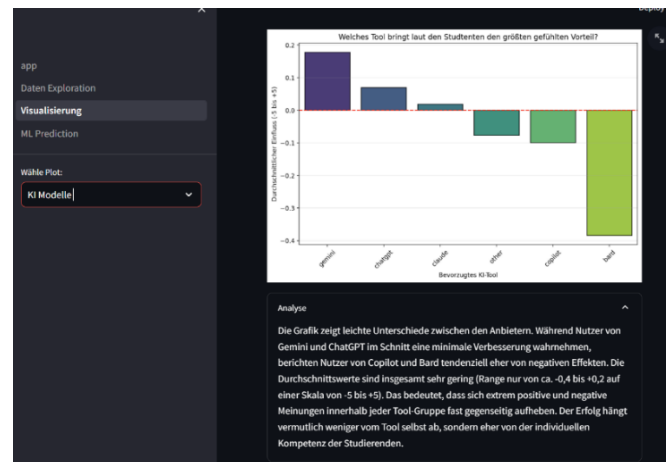


Abbildung 5 Visualisierungs Seite

ML Prediction Dieser Bereich macht das trainierte Machine-Learning-Modell (KNN) nutzbar. Nutzer können das Modell neu trainieren und sehen anhand einer Confusion Matrix sofort die Statistiken der Klassifikation (Abb. 6).

In einem Simulator können Parameter für ein hypothetisches Studierenden-Profil durch Slider, SelectBoxen und Number-Inputs, eingegeben werden (Tägliche Nutzung, Studiengang, Studienjahr, Vertrauen in KI). Die App berechnet live, ob eine Notenverbesserung vorhergesagt wird und mit welcher Wahrscheinlichkeit dies geschieht (Abb.7).

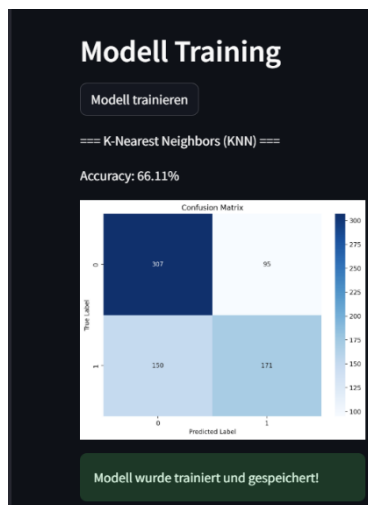


Abbildung 6 Modell Trainings Ansicht

Abbildung 7 Vorhersage