#### **Ziel des Projekts**

Entwicklung eines datengetriebenen Prognosesystems für den Buchmarkt. Das System soll mithilfe von Machine Learning (ML) auf Basis umfangreicher Buchdaten (über 1.000 reale Titel + ggf. synthetische Daten) zwei Kernfragen beantworten:

1. **Wird ein neues Buch ein Bestseller?**
2. **Wird ein Buch verfilmt werden?**

### **Teilprojekt 1: Bestseller-Prognose-Modell**

#### **Beschreibung**

Ein ML-Modell wird mit einer Datenbank aus über 1.000 Büchern trainiert – bestehend aus echten Bestsellern (z. B. Top 100 pro Jahr aus den letzten 25 Jahren weltweit) sowie ergänzend synthetisch generierten Beispielen für Nicht-Bestseller. Jedes Buch ist durch Features wie Sprache, Genre, Verkaufszahlen, Bewertung, Verlagsumsatz, Preis, Veröffentlichungsjahr, Filmadaption etc. beschrieben.

#### **Ziel**

Vorhersage, ob ein neues Buch basierend auf seinen Merkmalen (z. B. Genre, Sprache, Preis, Autor-Rating, etc.) Bestseller-Potenzial besitzt.

#### **Anwendungsfälle**

* Unterstützung für Verlage bei Investitionsentscheidungen.
* Marktanalyse für Selfpublisher oder Plattformen wie Amazon KDP.

### **Teilprojekt 2: Filmadaption-Prognose-Modell**

#### **Beschreibung**

Dieses Modell fokussiert sich ausschließlich auf reale Bücher, die nachweislich hohe Reichweite oder Verkaufserfolg haben (z. B. >100.000 verkaufte Exemplare). Ziel ist die Vorhersage, ob ein Buch mit hoher Wahrscheinlichkeit verfilmt wird. Die Trainingsdaten enthalten das Label „Verfilmt“ oder „Nicht verfilmt“.

#### **Features**

* Genre (Fantasy, Drama, Thriller etc.)
* Sprache & Herkunftsland
* Autoren- & Buchbewertung
* Verkaufszahlen & Popularität
* Themenrelevanz & emotionale Tiefe
* Vorhandene Verfilmung ähnlicher Bücher

#### **Ziel**

Ein Prognosetool, das z. B. Produzenten oder Streamingdienste frühzeitig Hinweise gibt, welche Bücher sich für eine Adaption eignen könnten.

### **Nächste Schritte**

* Zusammenstellung und Normalisierung des Buchdatensatzes.
* Feature-Engineering und Auswahl geeigneter Modelle (z. B. Random Forest, XGBoost, Neural Network).
* Modelltraining und Validierung.
* Evaluation anhand echter aktueller Bestseller/Verfilmungen.
* ggf. UI-Prototyp mit Streamlit oder als API.

### **Teilprojekt 3 (Erweiterung): Piraterie-Auswertung bei Bestsellern**

#### **Kurzbeschreibung**

Untersuchung von 50 internationalen Bestsellern der letzten Jahre hinsichtlich ihrer Verbreitung über illegale Plattformen (z. B. Torrents, PDF-Downloadseiten, Darknet, Foren).

#### **Ziel**

* Quantitative Analyse: Wie viele Raubkopien zirkulieren (geschätzt)?
* Vergleich: Welche Genres oder Sprachen sind besonders betroffen?
* Korrelation: Gibt es Zusammenhang zwischen Verkaufszahlen, Popularität und Piraterie-Risiko?

Projektidee: Analyse und Vorhersage des Erfolgs berühmter Bücher

Ziel des Teils-Projekts

# Erstelle eine interaktive Web-App mit Streamlit, um:

Daten über berühmte Bücher zu analysieren (z. B. Sprache, Genre, Jahr, Verkäufe usw.)

Eine eigene Buchidee einzugeben und mithilfe von Machine Learning vorherzusagen, ob das Buch erfolgreich sein könnte (z. B. Verkaufszahlen oder Bestsellerstatus).

? SQL-Datenbank zur Speicherung und Abfrage der Bücher zu nutzen

Python – Datenverarbeitung und Logik

Pandas / NumPy – Datenanalyse

Plotly /Seaborn – Visualisierungen

Scikit-learn oder XGBoost – ML-Modell zur Vorhersage

SQLite oder PostgreSQL – relationale Datenbank mit SQL-Abfragen

Funktionen der Streamlit-Web-App

1. Datenanalyse & Visualisierung

Filter nach Sprache, Genre, Jahr

Diagramme:

Verkäufe pro Genre

Top-Autoren nach Verkaufszahlen

Entwicklung der Bestseller über Jahrzehnte

Heatmap: "Genre × Jahr"

2. Machine Learning: Erfolgs-Vorhersage

Benutzer gibt Buchdaten ein (Genre, Sprache, Seitenanzahl, Verfilmung usw.)

Modell prognostiziert:

Verkaufszahlen (Regression)

Oder: Bestseller (ja/nein) – Klassifikation

Erklärung der Einflussfaktoren (Feature-Importance)

3. ? SQLDatenbank & Abfragen

Eine Datenbankseite mit:

Filterfunktionen

Manuelle SQL-Abfragen (z. B. alle Bücher nach 2000 mit mehr als 10 Mio Verkäufen)

Beispielabfragen als Buttons

4. Eigene Bücher hinzufügen

Benutzerformular zum Hinzufügen eigener Bücher

Neue Bücher werden in SQL-Datenbank gespeichert

Live-Aktualisierung der Visualisierungen

ML-Ideen im Detail

Regression: Verkäufe (in Millionen) vorhersagen

Klassifikation: Bestseller ja/nein

Clustering: Bücher nach Ähnlichkeit gruppieren (z. B. K-Means)

# Projektstruktur

bash

Code kopieren

buchanalyse-projekt/

│

├── data/

│ └── books.csv

│

├── app/

│ ├── main.py # Streamlit-App

│ ├── ml\_model.py # ML-Modelle & Vorhersage

│ ├── database.py # SQL-Logik

│ ├── visualizations.py # Diagramme & Plots

│ └── utils.py # Hilfsfunktionen

│

├── requirements.txt

└── README.md

??? Mögliche Erweiterungen

Verbindung zu externen Buchdaten (z. B. Google Books API)

Vergleich mit Verfilmungen (Box Office vs. Buchverkäufe)

Nutzerkonten mit Favoriten & Notizen

Dashboard für Verlage zur Manuskript-Einschätzung