## 1. 🗃️ Datensatz‑Vorschlag

Basierend auf eurer Zielsetzung wäre ein guter Einstieg:

* **"Amazon Top 50 Bestselling Books (2009–2019)"** – ~550 Einträge mit Features wie Titel, Autor, Verkaufsrang, Genre, Preis, Veröffentlichung [kaggle.com](https://www.kaggle.com/datasets/ashishjangra27/books-dataset?utm_source=chatgpt.com)[news.ycombinator.com+2openlibrary.org+2openlibrary.org+2](https://openlibrary.org/help/faq/using?utm_source=chatgpt.com)[kaggle.com+5kaggle.com+5kaggle.com+5](https://www.kaggle.com/datasets/sootersaalu/amazon-top-50-bestselling-books-2009-2019?utm_source=chatgpt.com)
* Alternativ: **"Best-selling books dataset"** (Wikipedia-Scraping) oder umfassenders Kaggle-Set  
  [kaggle.com+3kaggle.com+3kaggle.com+3](https://www.kaggle.com/datasets/mansivasantpatel/best-selling-books-dataset?utm_source=chatgpt.com)

➡️ Diese Sets sind überschaubar, gut dokumentiert und reichen für Start‑Analysen (Genre‑Verteilung, Trends über Zeit).

## 2. 🎯 **Sprint‑Ziel 1 (2 Wochen)**

**"Bis zum Ende des Sprint 1 steht ein funktionales Prototype‑Dashboard: Daten sind geladen, bereinigt, und erste visuelle Analysen (z. B. Verkaufsrang nach Genre oder Jahr) laufen in Streamlit."**

## 3. 🛠️ Sprint 1 – Aufgaben & Rollen

| **Task** | **Verantwortlicher** | **Aufwand** |
| --- | --- | --- |
| Repo+Ordnerstruktur+Coding Conventions (z. B. z.B. /data, /src, /app) | Dev A | 2 h |
| Dataset herunterladen & als CSV in /data speichern | Dev B | 1 h |
| Daten einlesen & typischen EDA: dtypes, fehlende Werte, Duplikate, Basisstatistiken | Dev C | 3 h |
| Plot 1: Genre-Verteilung (Bar Chart) in Streamlit anzeigen | Dev A | 2 h |
| Plot 2: Zeitverlauf der Verkaufsrank‑Entwicklung (Line Chart) | Dev B | 3 h |
| Streamlit-Grundstruktur aufsetzen (App-Layout, Seitenstruktur) | Dev C | 2 h |
| Git-Workflow strukturieren (Branches + PR) & Initial Commit | Dev A + B + C | 1 h |
| Daily Scrums (15 min täglich) & Dokumentation starten (README + Sprint-Ziel) | Scrum Master | 0.25 h/day |
| Test & Review der bisherigen Funktionalität | alle | 2 h |
| Sprint Review & Retrospektive (Demo + Feedback) | alle | 2 h |

## 4. 🧰 Beispiel‑Trello‑Board / Product Backlog

**Listen**:

* **Backlog**: Ideen & Stories für Sprint 1+
* **Sprint 1 – To Do**: ausgewählte Tasks
* **Doing**: aktuell bearbeitet
* **Done**: abgeschlossene Tasks
* **Impediments**: Blocker

**Karten‑Beispiele (To Do)**:

* „Setup Git-Repo + Ordnerstruktur“
* „Dataset aus Kaggle herunterladen & speichern“
* „EDA: dtypes, missing, Duplikate, Basis-Statistiken“
* „Streamlit App-Grundgerüst implementieren (App starten, Sidebar, Layout)“
* „Visualisierung: Genre-Verteilung“
* „Visualisierung: Verkaufsrank-Entwicklung im Zeitverlauf“
* „Tägliches Scrum + README verfassen“

## 5. ✔️ Definition of Done (Sprint 1)

* ✅ Daten geladen & bereinigt, ohne Fehler
* ✅ Mindestens zwei visualisierbare Analysen implementiert
* ✅ Streamlit‑App funktioniert korrekt
* ✅ Alle Änderungen sind committed + PRs gemerged
* ✅ Kurze README + Sprint‑Ziel dokumentiert

## 📅 Nächste Schritte

1. Wählt euren Datensatz (z. B. Amazon Top 50, Kaggle Bestseller).
2. Entscheidet, wer von euch welche Rolle übernimmt.
3. Erstellt euer Trello-Board und packt obige Tasks ins „To Do“.
4. Plant das erste Sprint Planning + Daily Scrums.

Wenn du willst, erstelle ich auch gerne ein **fertiges Trello‑Template** mit diesen Karten!

## 2. 🎯 **Sprint‑Ziel 1 (2 Wochen)**

**"Bis zum Ende des Sprint 1 steht ein funktionales Prototype‑Dashboard: Daten sind geladen, bereinigt, und erste visuelle Analysen (z. B. Verkaufsrang nach Genre oder Jahr) laufen in Streamlit."**

## 3. 🛠️ Sprint 1 – Aufgaben & Rollen

| **Task** | **Verantwortlicher** | **Aufwand** |
| --- | --- | --- |
| Repo+Ordnerstruktur+Coding Conventions (z. B. z.B. /data, /src, /app) | Dev A | 2 h |
| Dataset herunterladen & als CSV in /data speichern | Dev B | 1 h |
| Daten einlesen & typischen EDA: dtypes, fehlende Werte, Duplikate, Basisstatistiken | Dev C | 3 h |
| Plot 1: Genre-Verteilung (Bar Chart) in Streamlit anzeigen | Dev A | 2 h |
| Plot 2: Zeitverlauf der Verkaufsrank‑Entwicklung (Line Chart) | Dev B | 3 h |
| Streamlit-Grundstruktur aufsetzen (App-Layout, Seitenstruktur) | Dev C | 2 h |
| Git-Workflow strukturieren (Branches + PR) & Initial Commit | Dev A + B + C | 1 h |
| Daily Scrums (15 min täglich) & Dokumentation starten (README + Sprint-Ziel) | Scrum Master | 0.25 h/day |
| Test & Review der bisherigen Funktionalität | alle | 2 h |
| Sprint Review & Retrospektive (Demo + Feedback) | alle | 2 h |

## 4. 🧰 Beispiel‑Trello‑Board / Product Backlog

**Listen**:

* **Backlog**: Ideen & Stories für Sprint 1+
* **Sprint 1 – To Do**: ausgewählte Tasks
* **Doing**: aktuell bearbeitet
* **Done**: abgeschlossene Tasks
* **Impediments**: Blocker

**Karten‑Beispiele (To Do)**:

* „Setup Git-Repo + Ordnerstruktur“
* „Dataset aus Kaggle herunterladen & speichern“
* „EDA: dtypes, missing, Duplikate, Basis-Statistiken“
* „Streamlit App-Grundgerüst implementieren (App starten, Sidebar, Layout)“
* „Visualisierung: Genre-Verteilung“
* „Visualisierung: Verkaufsrank-Entwicklung im Zeitverlauf“
* „Tägliches Scrum + README verfassen“

## 5. ✔️ Definition of Done (Sprint 1)

* ✅ Daten geladen & bereinigt, ohne Fehler
* ✅ Mindestens zwei visualisierbare Analysen implementiert
* ✅ Streamlit‑App funktioniert korrekt
* ✅ Alle Änderungen sind committed + PRs gemerged
* ✅ Kurze README + Sprint‑Ziel dokumentiert

## 📅 Nächste Schritte

1. Wählt euren Datensatz (z. B. Amazon Top 50, Kaggle Bestseller).
2. Entscheidet, wer von euch welche Rolle übernimmt.
3. Erstellt euer Trello-Board und packt obige Tasks ins „To Do“.
4. Plant das erste Sprint Planning + Daily Scrums.

Wenn du willst, erstelle ich auch gerne ein **fertiges Trello‑Template** mit diesen Karten!