Project scope update:

I slightly refined the scope. Originally, I planned to use three sources that all contained some overlapping information. I replaced one source with a sales dataset to enable a cross-dataset comparison (ratings vs. sales). The final goal is unchanged: analyze game and predict video game ratings. For the rating metric, I used the Steam positive rating ratio. I standardized genres across sources (a canonical genre_std), filtered to 2000–2020, removed NSFW/non-game rows, and required at least 10 reviews for Steam when computing ratios and running the regression. Full pipeline and running instructions are in README.md.

Data sources:

Collected in code (src/data_collection.py) and saved into data/ (but not committed).

- RAWG API → rawg_10000_unfiltered.csv: name, released, rating, platforms, genres. Used for Q1–Q2. (10,000 raws, 3591 left after cleaning)
- Steam (Kaggle: nikdavis/steam-store-games) → steam.csv: positive_ratings, negative_ratings, price, average_playtime, owners, categories, platforms, publisher, release_date. Derived: total_reviews, positive_rating_ratio, owners_mid. Used for Q3–Q5 and prediction. (27,075 raws, 6,097 after filters)
- VG Sales (Kaggle: gregorut/videogamesales) → vgsales.csv: Name, Platform, Year, Genre, Global_Sales. Used for Q6 (ratings vs sales by genre). (16,958 raws)

API used / how it's demonstrated

- **RAWG API**: I keep my RAWG_API_KEY in a local .env file (not committed; .env is in .gitignore). src/data_collection.py loads it with python-dotenv and calls https://api.rawg.io/api/games.
- **Kaggle API**: I authenticate **either** with environment variables KAGGLE_USERNAME/KAGGLE_KEY **or** a local kaggle.json file.
  kaggle.json is copied to ~/.kaggle/ when running src/data_collection.py. It is also git-ignored.

Issues / difficulties (till now):

- Rate limits/latency when collecting RAWG; solved by paging sequentially, light sleeps, and caching CSVs so re-runs skip collection.
- Schema/label mismatch (genres differ across sources); solved with the canonical genre_std and lower-casing/stripping.
- Model performance: baseline linear regression has modest R2R.