210 Final Project

A. Project Overview

Goal: The project investigates what characteristics will make a video game successful and a bestseller. A gamer is a “bestseller” if it has sold over 1 million copies globally. This project aims to predict whether genre, platform, publisher, or year of release has an impact on the games commercial success.

Dataset: The project uses the Video Game Sales dataset from kaggle, which includes over 16,000 video game entries with columns such as name, platform, release year, genre, publisher, and global sales. The dataset was loaded into Rust using the csv and verde crates.

B. Data Processing

The csv file was parsed into rust using csv crate and serde crate which maps each row of the file to a rust struct. I created a struct called Game, and defined the fields to match the column names in the dataset. When I cleaned this dataset, I excluded any game with a missing global sales field from the best seller related analysis. Any invalid years were excluded from the year based trends.

C. Code Structure

Modules:

- `data.rs`: I loaded the dataset and parsed it here. .
- `analysis.rs`: Used for performing analyses and printing in a certain formatting.
- `main.rs`: Ran all the analysis here and printed output.

Key Functions and Types:

Here are the main types and functions I used:

- **`Game` struct**: This holds the data for each video game, like the name, platform, genre, publisher, release year, and global sales. Some fields are optional because the data might be missing or invalid.

Functions:

- **`load_games`** (in `data.rs`): Takes a CSV file path and returns a list of `Game` entries. It reads the file and uses `serde` to turn each row into a `Game` object, skipping anything that can't be parsed properly.
- **`analyze_category`**: Takes a list of games and groups them by something like genre or platform. Then it prints how many games in each group were bestsellers (sold more than 1 million units).
- **`average_sales_by_category`**: Similar to `analyze_category`, but instead of counting bestsellers, it calculates the average sales for each group.
- **`genre_predictor`**: Applies a basic rule that says "all games of this genre are bestsellers" and checks how accurate that guess is using a confusion matrix.
- **`yearly_bestseller_trend_with_bar`**: Counts how many bestsellers were released each year and shows it with a visual bar chart using characters.
- **`top_games_by_sales`**: Prints the top-selling games based on total global sales.

Main Workflow:

The `main.rs` module executes the full analysis pipeline in the following order:

1. Load and parse the CSV file.
2. Print bestseller rates by category.
3. Evaluate prediction heuristics.
4. Visualize annual trends.
5. Show highest selling games.

D. Tests

I added a `tests.rs` module to check that the main functions work correctly. It uses a few sample games to test each function and make sure nothing crashes. The tests include checks for `genre_predictor`, `analyze_category`, `average_sales_by_category`, and `top_games_by_sales`.

These tests don't check for exact output, but they make sure the functions run without errors. This helps confirm that the main parts of the program are working as expected.

I ran the tests using a cargo `test`, and everything passed without any problems. using `cargo test`, and the output confirmed that there were no panics or unexpected behavior. using `cargo test`. These checks help ensure the integrity of the core logic and satisfy the project requirement for reproducible test coverage.Spot checks were performed on genre counts, average calculations, and confusion matrix results. No automated test suite was used.

E. Results

Here are some of the key things I found based on the actual output:

- **Genre Bestsellers**: Platform games had the highest hit rate (22.01%), followed by Shooter (18.93%) and Sports (12.70%).
- **Platform Bestsellers**: NES had the highest hit rate at 73.47%, but it had fewer total games. PS4 also had a strong rate (21.73%).
- **Publisher Bestsellers**: Nintendo had the highest ratio with 47.80% of its games being bestsellers, followed by Electronic Arts at 25.02%.
- **Bestseller Years**: 2008, 2007, and 2010 had the most bestsellers. There's a noticeable peak between 2004–2011.
- **Prediction Accuracy**: Predicting all Shooter games as bestsellers had an accuracy of 82.72%. Predicting all Sports games had 77.08% accuracy.
- **Yearly Trends**: The ASCII bar chart showed a clear rise in bestsellers from 1995 to around 2011, with peaks in 2007 and 2008.
- **Top Games by Sales**: The top-selling game was *Wii Sports* with 82.74M units. Others in the top 10 included *Super Mario Bros.*, *Mario Kart Wii*, and *Duck Hunt*.
- **Average Sales by Genre**: Platform games had the highest average sales per title (0.94M), followed by Shooter (0.79M) and Role-Playing (0.62M).
- **Top Publishers by Average Sales**: Nintendo led by far with an average of 2.54M units per game, followed by Take-Two Interactive (0.97M) and Sony (0.89M).

F. Usage Instructions

To run the program, follow these steps:

- Make sure `vgsales.csv` is in the project folder.
- Open a terminal and run:
  cargo run
- The results will show up in the terminal.
- It usually runs in under a second. `vgsales.csv` in the root directory.
- Run the program using:
  cargo run
- Output is printed directly to the terminal.
- Execution time is typically under 1 second.

G. AI-Assistance Disclosure

I used ChatGPT to help with generating ideas for the project, explaining concepts in Rust, and giving examples on how to write some of the functions. I made sure I understood all the materials before starting on the project. Only standard libraries (`csv` and `serde`) were used outside of Rust's core tools.