Introduction to Steam-Games-Recommender-System

Zeyu Liang

Video games have become a big part of our lives, with platforms like Steam offering thousands of different games to play. With so many options, it can be tough for people to find the games that match their interests. This is why game recommendation systems are really important. These systems help users find games they might enjoy, making it easier to decide what to play next. When users find games they like more easily, they end up spending more time on the platform, which is great for both the game developers and the platform itself.

The goal of this project is to create a game recommendation system for Steam users that works by predicting what games they might like based on their past ratings. By using a dataset with user ratings for different games, we can build a system that learns from these patterns and recommends new games that users haven’t tried yet. This approach helps make browsing huge game libraries less overwhelming and more fun, as users get personalized suggestions that fit their preferences.

In this report, I’ll go over how I prepared and transformed the data. The original dataset had over 4.4 million rows, where each row showed a user rating a game. I restructured this data into a matrix where each row represents a user and each column represents a game. If a user hasn’t rated a game, the cell stays blank (or NaN). This helps the model understand which games have been rated and which haven’t.

The next step was to split the data into training and testing sets. The top 80,000 user records were used for training, and the rest were for training. I also made sure that the user\_id and app\_id columns were re-indexed starting from 1 so that the data was more organized and easier to work with. This report will discuss different techniques for building recommendation systems, like collaborative filtering and machine learning methods, and how they perform on this dataset.

There are some limitations to this project that are worth mentioning. One limitation is that the system only uses past ratings to make predictions, without considering other useful information like user age, location, or game genres. This means that the recommendations might not be as accurate as they could be if more data were included. Also, because the dataset is so large, I eliminate a lot of the data so that I can train my model faster on my CPU. If I have a GPU, I can then incorporate more data and train a more accurate model that prevents overfitting.