DATA301 Final Report

Phoebe Zhang 62244228

Abstract/Summary

This project uses the review data & item metadata datasets from Version 2 of the Steam Video Game and Bundle Data. The research question is: Which user defined tags are most commonly associated with high user engagement, and can these tags be used to predict the popularity of emerging games? To answer this question, the A-Priori algorithm and Jaccard Similarity will be used. The intended outcome is to obtain a list of user-defined tag combinations that are linked to popular games and create a tag-based similarity model capable of predicting the potential popularity of emerging games by comparing their user-defined tags to those of previously/currently popular games. The results from this project could assist game developers in designing engaging games and advise them of game features that attract users and guide gamers through Steam's vast game catalogue allowing them to discover high-quality games earlier.

Introduction

Background

The selected datasets, from Version 2 of the Steam Video Game and Bundle Data, contain metadata and user review data for available games on Steam. A key feature in these datasets is the 'tags' attribute which represents a list of user-defined labels applied to the game, chosen by players to reflect their experience, gameplay and the content of the game. The 'genres' feature in the datasets are broad categories applied to the given game, defined by the developers and are not analysed as a part of the user-defined tags. The 'release_date' feature indicates when the game was released allowing the ability to distinguish older games from newly released ones. The 'hours' feature describes the total time a given user has spent playing the specific game, reflecting user engagement.

The A-Priori algorithm is used to find frequent combinations of items (itemsets) and association rules in large datasets. A key concept of the A-Priori algorithm relevant to this project is the support. The support of an itemset measures how frequently it appears in the dataset with respect to the total number of transactions. The Jaccard Similarity of sets is the ratio indicating the similarity of two sets by comparing their intersection size to the size of their union.

Motivation

The research question is personally relevant to me because I am a game enthusiast who frequently uses Steam to discover new games. When looking for a new game to try out, I find that user-defined tags help to guide my decision, especially when I'm looking for something to match a specific genre, mood, or playstyle. However, with Steam's extensive collection of available games, I **often** find it overwhelming and exhausting to search through; oftentimes emerging with no luck finding something that might truly be entertaining and engaging to play, especially games that have not been widely recognised or marketed.

By analysing which user-defined tags are most commonly associated with high user engagement and applying the identified patterns to predict the potential popularity of emerging games, this research forms a direct connection between my own experience as a gamer and Steam user and my academic experience in data science and algorithms. Sharing the results of this research with others in the gaming industry where everyone is always looking for something new could help them to discover hidden gems based on patterns identified in community feedback; not purely through marketing.

Research Question/Hypothesis

The research question is: Which user-defined tags are most commonly associated with high user engagement, and can these tags be used to predict the popularity of emerging games? This question is relevant to the datasets from Version 2 of the Steam Video Game and Bundle Data because user-defined tags, game release dates, game metadata, and user reviews are present in the datasets. The listed features of the dataset enable the analysis of the correlation of user-defined tags and high-engagement games and the identification of newer games which share similar tags.

The A-Priori algorithm is implemented to identify frequent user-defined tag combinations among high-engagement games (those with high number of reviews and playtimes) and the Jaccard similarity is computed to measure the similarity of sets of tags between high-engagement games and newer emerging games.

Experimental Design and Methods

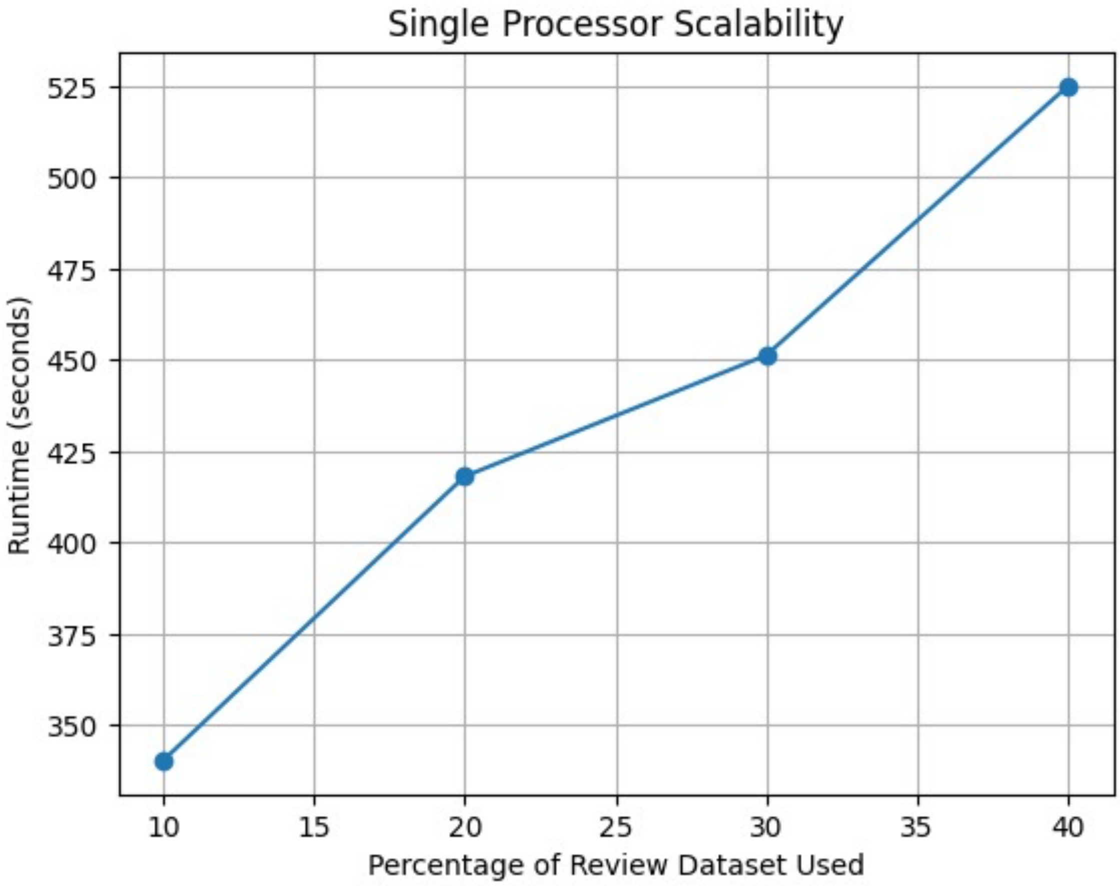
The dataflow process began with data preprocessing: the raw metadata and review datasets were loaded, decompressed, cleaned and filtered to retain relevant columns namely user-defined tags, release_date, product_id, app_name, and user engagement metrics such as hours played and total number of reviews. The tags were normalized to ensure consistent formatting, and the release dates were converted to datetime. The two cleaned datasets were then merged on product_id and aggregated to compute total hours played and the number of user reviews per game.

The threshold for high engagement was defined by the top 75th percentile of total hours and number of reviews. The combined dataset was filtered to only contain games with total hours and number of reviews surpassing the defined thresholds. A second subset was formed by filtering the combined data for games released within the past 2 years relative to the most recent release date in the dataset.

The user-defined tags lists in the high engagement games data were converted into binary arrays, and the A-Priori algorithm was applied to identify frequent combinations of user-defined tags among the popular games. The Jaccard Similarity was then computed on the binary set of user-defined tags from the newer games data and the identified tag combinations from the high-engagement games data to evaluate the tag similarity between the high engagement games and the newer ones. Each game's user-defined tags in the recent games dataset was compared against the frequent tag itemsets to compute its maximum Jaccard Similarity score, quantifying how closely a newer game resembles popular games based on community selected tags. Games with a Jaccard similarity score equal to 1.0 are those with exact tag matches to popular games, indicating potential popularity based on user-defined tag similarity.

- urllib.request: downloads the compressed dataset files from UCSD's Recommender Systems and Personalization Datasets website
- gzip and shutil: decompresses the .gz files into raw JSON data
- ast and json: converts python dict strings to valid JSON strings
- pandas: used for transforming in-memory data
- dask.dataframe and dask.bag: for scalable reading and preprocessing of the large datasets
- normalize_tags(tags): function to normalize user-defined tag lists for each game
- mlxtend.preprocessing.TransactionEncoder: for transforming user-defined tag lists into binary matrices for the A-Priori algorithm
- mlxtend.frequent_patterns.apriori: to identify frequent itemsets in high engagement games
- compute_jaccard_similarity(tags, frequent_sets): function to compute Jaccard similarity between a new game's user-defined tags and the identified frequent tag sets from high engagement games
- apply_jaccard(partition, frequent_sets): function to apply Jaccard similarity scores across partitions of the dataset
- map_partitions(): to apply the Jaccard similarity function across recent games data in parallel

Results



Frequent combinations of user-defined tags among high engagement games defined as those in the top 25% of total hours played and number of reviews were identified. The results showed that there were 38 frequent tag combinations with a support greater than 0.2. The top 5 frequent tag combinations sorted by support are: (singleplayer, action), (adventure, singleplayer), (singleplayer, indie), (adventure, action), and (multiplayer, action).

The Jaccard similarity between the tag sets of newer released games and these frequent itemsets was computed. It was found that 14 games from the recent games subset had a Jaccard similarity score of 1.0, indicating exact matches with tag combinations found in popular games. These games can be interpreted as having high potential success in the future, as their user-defined tags resemble tag combination patterns associated with popular games.

Conclusion

Were you able to answer your hypothesis / research questions?

Yes, I was able to answer the research question regarding which tag combinations are most commonly associated with high engagement games and I used these patterns to predict the potential success of newer emerging games. By implementing the A-Priori algorithm to identify frequent itemsets of tag combinations and applying Jaccard similarity to measure tag similarity, relevant frequent itemsets were extracted and utilised to predict the success of newer released games.

What implications do your results have?

By identifying frequent patterns of tag combinations associated with high engagement games, game developers are able to better design, build and market their games to align with features that are known to perform well, increasing the chances of success in the competitive game market. The results also offer game distribution platforms such as Steam, a way to recommend and promote new games that share features similar or identical to popular games, enabling the ability to enhance recommendation systems and advertise promising new games earlier in their release.

What future questions or directions would you take with your project?

To extend on this project, I would consider incorporating user demographic data such as playtime habits or user region to investigate whether specific tag combinations are associated with certain audiences (e.g. hardcore gamers vs casual gamers). Another direction the project could take is to integrate additional metadata to uncover deeper patterns within the data. This could entail of implementing the A-Priori algorithm and applying Jaccard similarity on not just the ‘tags’ feature but also the ‘genres’ (categories defined by game developers) column of the game metadata dataset to potentially uncover more insights into what drives game popularity, based on both user-defined tags and developer-defined categories. Furthermore, the scalability of the project could be improved by experimenting with systems beyond Dask.

Critique of Design and Project

A part of the project that I think could have been improved with another approach was the data preprocessing stage, specifically the conversion of Python dictionary strings into valid JSON strings. The raw JSON data files (steam_games.json and steam_reviews.json) contained string representations of Python dictionaries, not directly readable by Dask’s JSON readers. To address this problem, the data was converted sequentially into Python dictionaries and then re-encoded into valid JSON format strings. The approach worked, but was inefficient, taking over 10 minutes to process the datasets due to their large size and the sequential conversion implementation.

To improve the runtime of this problem, parallel preprocessing could be implemented. Instead of converting the Python dictionary strings sequentially, multiple lines can be converted simultaneously by distributing the workload across multiple processors. By parallelizing the work done on this step, the runtime can be significantly reduced, making the workflow more efficient for large datasets.

Reflection

- A-Priori algorithm
- Jaccard similarity
- Union & intersection
- Parallelism
- Dask libraries
- Data cleaning and preprocessing
- Python libraries
- Dask documentation
- Transaction encoding

What did you learn from this project?

From this project, I learnt how to implement the A-Priori algorithm to identify frequent itemsets in large, complex datasets and how to apply and interpret Jaccard similarity measures. I acquired practical experience with handling large real-world datasets containing poor formatting, missing columns and differences between expected and actual metadata, highlighting the importance of effective data cleaning. The project emphasized the challenges of dealing with inconsistent, large, and complex real-world data.

References

Apriori Algorithm. (05/04/2025). GeeksforGeeks. Retrieved 2 May from https://www.geeksforgeeks.org/apriori-algorithm/

Apurva Pathak, K. G., Julian McAuley. (2017). *Generating and personalizing bundle recommendations on Steam* SIGIR,

Jure Leskovec, A. R., Jeff Ullman. Finding Similar Items. In *Mining of Massive Datasets* (3rd ed., pp. 73-134). Cambridge University Press. http://infolab.stanford.edu/~ullman/mmds/ch3n.pdf

Jure Leskovec, A. R., Jeff Ullman. Frequent Itemsets. In *Mining of Massive Datasets* (pp. 213-251). Cambridge University Press. http://infolab.stanford.edu/~ullman/mmds/ch6.pdf

Karabiber, F. *Jaccard Similarity*. Retrieved 2 May from https://www.learndatasci.com/glossary/jaccard-similarity/

Mengting Wan, J. M. (2018). *Item recommendation on monotonic behavior chains* RecSys,

Mwiti, D. (2025). *Apriori Algorithm Explained: A Step-by-Step Guide with Python Implementation*. https://www.datacamp.com/tutorial/apriori-algorithm

Raschka, S. *TransactionEncoder: Convert item lists into transaction data for frequent itemset mining*. https://rasbt.github.io/mlxtend/user_guide/preprocessing/TransactionEncoder/

Team, D. D. (2016). *Dask*. https://docs.dask.org/en/stable/

Wang-Cheng Kang, J. M. (2018). *Self-attentive sequential recommendation* ICDM,