Personalizing Recommendations on *Steam*

1 Dataset Exploratory Analysis

The dataset we chose to use was the Steam video game reviews and bundles dataset provided by Julian McAuley. This dataset consists of four types of metadata, reviews associated with users, product bundles, game details, and games owned per user.

The reviews associated with users metadata consists of a 'user_id' and a list of reviews, with each review containing the date it was posted, the game ID, if the user recommends the game, and the text review the user left for that game.

The product bundles metadata consists of the bundle's price, its name, the discount provided, its id, and the games provided with the bundle while the game details metadata consists of the game's publisher, genres, app name, title, tags, price, and ID.

The games owned per user dataset consists of the user's id, the number of games the user owns, the user's steam id, and the list of games the user owns, with each item in the list containing the game's id, the game's name, and how many hours the user has played the game in the past 2 weeks and forever.

After analyzing the data, we found no clear trend between a game's recommended percentage and its popularity. As seen in **Figure 1**, the top 20 games ranked by number of purchases all have roughly the same recommended percentage, and this trend of games being recommended by between 80% to 100% of its reviewers is consistent throughout the entire dataset.

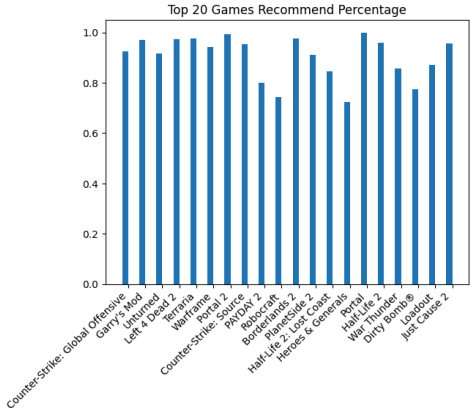

Figure 1: **Top 20 Games' Recommend Percentage**

However, **Figure 2** demonstrates that the popularity of a game is not necessarily representative of how much a user enjoyed the game, as Payday 2 has a higher number of recommends compared to some higher ranking games, despite having less purchases.

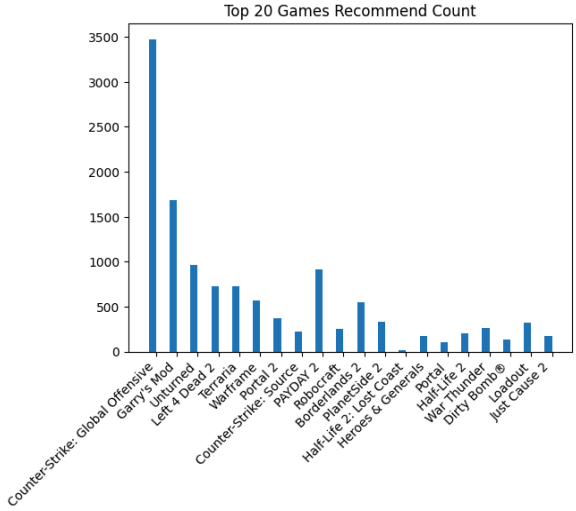

Figure 2: **Top 20 Games' Number of Recommends**

One metric we found interesting was that the percentage of games produced within a specific price range (the ranges being: Free, Under $10, Between $10 and $20, and Over $20) closely matched that of the percentage distribution of games purchased within those same ranges. This implies that game producers understand customers' willingness to pay, so our team placed less emphasis on price as a feature.

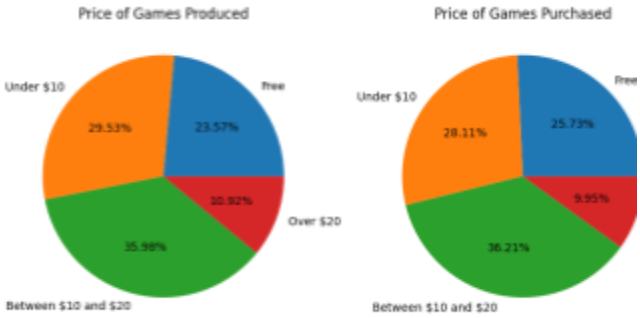

Figure 3: **Distribution of Games by Price**

In terms of overall playtime, Counter-Strike: Global Offensive by far has the most playtime, nearly

twice as much as the next most played game, Garry's Mod. This finding is consistent with the popularity of individual genres pictured in **Figure 5** as the genres of Counter-Strike: Global Offensive include action, single-player, multiplayer, Co-Op, Shooter, and more that are in the top 20 most popular genres.

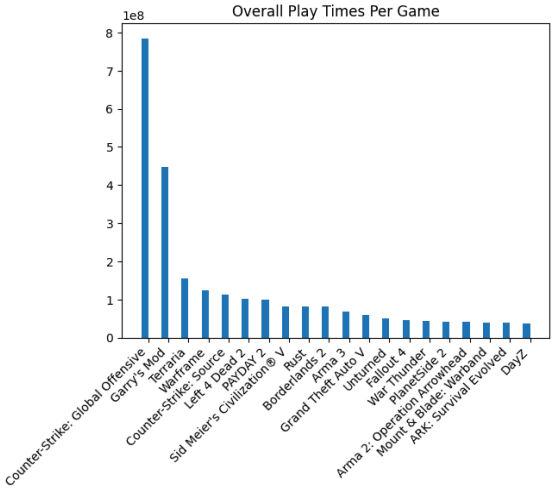

Figure 4: **Top 20 Games by Overall Playtime**

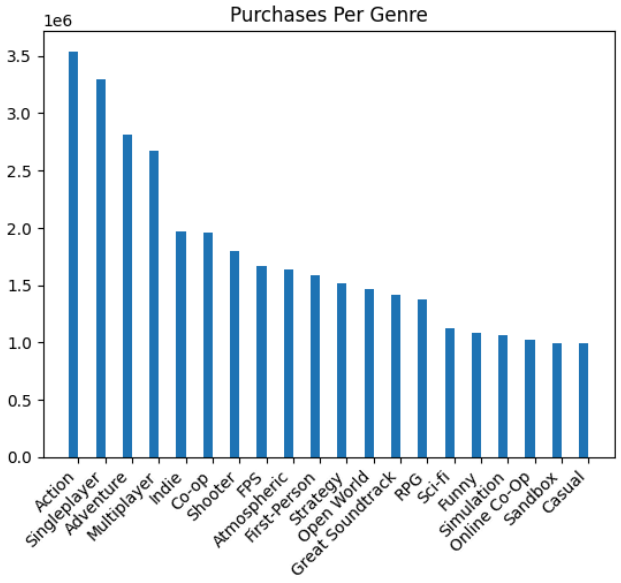

Figure 5: **Top 20 Genres by Game Purchases**

2 Predictive Task

With personalized recommendations, understanding user preferences and predicting relevant content are crucial for enhancing user experience. For our predictive task, we aimed to develop a recommendation system that could predict games a user is likely to buy or play based on their existing library and interactions with these games (in this case, the interactions were reviews and recommendations). While we explored various approaches, we ultimately prioritized a balance between model complexity, runtime efficiency, and interpretability.

We brainstormed several ideas, each addressing different aspects of user behavior and game characteristics. However, some ideas were discarded due to challenges in computational overhead, runtime, or data availability:

1. Discount Optimization:
   Predicting the optimal percentage discount for enticing users to purchase a game while maximizing revenue. This approach required detailed transactional data and dynamic pricing strategies, which added significant complexity.
2. Library-Based Prediction:
   Using a user's current game library to predict whether they would be interested in related games. This method relied heavily on semantic relationships between games, which were challenging to quantify without extensive metadata or external content features.

After considering these factors, we decided to focus on building a recommendation system centered on game interactions and user preferences.

Initially, we attempted to build a model based on **Jaccard Similarity**, where we compare the set of users for a given game to the set of users for each game that a user doesn't own. If no similarity existed between the items, the model would recommend a game based on the user's most played genre and the game with the highest number of recommendations. These recommendations are based on the first half of the game dataset. The second half of the dataset is then used to validate the recommendations by checking if the recommended games are part of the user's game dataset in the second half.

The model would incorporate features such as genre, the user's playtime for each genre (to determine a preferred genre), the number of recommendations a game has, and the price of the game.

3 Model

Initially, we attempted to use a **Jaccard Similarity based model**, but the model had significant issues with time complexity and took too long to run without producing results. We encountered similar challenges using other similarity metrics, where the prediction accuracy was low and the time complexity was a major limitation.

We also considered using a **logistic regression model**, but the size of the dataset—due to the large number of features— again, created significant overhead, resulting in severe runtime issues. It also took too much time to produce any results.

We decided to go with the **Bayesian Ranking Model** (from Chapter 5), which provided somewhat accurate results within a reasonable time frame. The primary model is a Bayesian Personalized Ranking (BPR), a pairwise ranking algorithm that optimizes the likelihood that a user prefers a positive interaction (e.g., a played game) over a negative interaction (e.g., an unplayed game).

The model predicts a score for each user-game pair using the BPR objective:

- Higher scores indicate stronger likelihood of interaction (e.g., a user playing or liking a game).
- Lower scores suggest the user is less likely to interact with the game.

The goal is to rank true user-game pairs higher than false pairs.

1. Test Dataset:
    - The test set consists of:
        - True examples: Games the user has interacted with (e.g., played or reviewed).
        - False examples: Games the user has not interacted with (randomly sampled).
    - Each user has an equal number of true and false examples in the test set.
2. Predictions:
    - The model predicts scores for all test examples (true and false) for each user.
    - The examples are sorted by their predicted scores in descending order.
3. Top-half Prediction:
    - For each user, the top half of the sorted examples is predicted as "true" interactions (positive class).
    - The bottom half is predicted as "false" interactions (negative class).
4. Comparison:
    - The predictions are compared against the actual user-game interactions in the test set.
    - True positives are correctly predicted interactions in the top half.
5. Accuracy Calculation:
    - The total number of correct predictions (true positives) is divided by the total number of predictions.

The accuracy measures how effectively the BPR model prioritizes true user-game interactions over false ones. It reflects the model's ability to:

- Learn user preferences from the training data.
- Generalize these preferences to unseen data (test set).

BPR is well-suited for implicit feedback datasets because it directly optimizes ranking performance rather than predicting absolute scores. It is computationally efficient for sparse datasets and requires only user-item interactions, making it ideal for our task.

It turns out that the Bayesian Ranking Model can make accurate predictions about whether a game should be recommended to a user. However, scalability of the dataset remained a major obstacle, so I split the dataset into training and testing sets.

Using the Bayesian Ranking Model was most optimal due to its efficiency and simplicity. This model only focuses on the interactions between users and games, rather than on the features of the games like genre and pricing.

To further enhance the BPR model, we planned to incorporate weighted features such as game price, tags, and genre. Assigning higher weights to games with higher playtime or positive recommendations could personalize rankings more effectively. This approach would refine the model's

ability to capture user preferences, particularly for their favorite genres or game types. However, these ideas never came to fruition as we struggled with manipulating the other attributes.

4 Related Literature

Our team decided to use the existing Steam Video Game Dataset provided by Julian McAuley for his CSE 158R class. It originated from the **Generating and personalizing bundle recommendations on Steam** paper, co-authored by McAuley. This dataset was used in homeworks to teach students in the class and give them practice implementing different recommender systems. Additionally, this dataset has been used in the following papers:

1. **Self-attentive sequential recommendation**
   Wang-Cheng Kang, Julian McAuley
   ICDM, 2018
2. **Item recommendation on monotonic behavior chains**
   Mengting Wan, Julian McAuley
   RecSys, 2018
3. **Generating and personalizing bundle recommendations on Steam**
   Apurva Pathak, Kshitiz Gupta, Julian McAuley
   SIGIR, 2017

In the **Self-attentive sequential recommendation** paper, the authors, Kang and McAuley, proposed a self-attention based sequential model (SASRec) that outperforms various state-of-the-art sequential models (including MC/CNN/RNN-based approaches) on both sparse and dense datasets. To prove this, they ran their SASRec model on multiple datasets, one of which was the Steam dataset my team plans to use.

In the **Item recommendation on monotonic behavior chains** paper, the authors, Wan and McAuley, propose an item recommendation framework which jointly models the full spectrum of interactions between implicit and explicit signals. To demonstrate that this framework is effective, the authors developed a new recommendation algorithm based on it and ran it on multiple datasets, ranging from sparse to dense, one of which is the Steam dataset my team plans to use.

In the **Generating and personalizing bundle recommendations on Steam** paper, the authors, Pathak, Gupta, and McAuley, seek to understand the semantics of what constitutes a ‘good’ bundle in order to recommend the best bundle for a user. To do this, they extracted a dataset from the Steam video game distribution platform, which is the dataset my team plans to use.

Similar datasets to the Steam dataset my team is using all extract their data from Steam, an online game distribution platform, but they extract different types of data. For example, one dataset, "*Steam Games Dataset : Player count history, Price history and data about games*," focuses on how many players are playing a game at a time and was used in a paper titled, **Weekly Seasonal Player Population Patterns in Online Games: A Time Series Clustering Approac**h. This paper sought to discover patterns of weekly player population fluctuations.

Unfortunately, we did not apply any state-of-the-art methods mentioned in related papers for our prediction task due to a lack of understanding on how to implement and use those methods as well as time constraints. Our model used a basic implementation of stochastic gradient descent, and no advanced methods of optimization were used.

All of the listed papers by McAuley support our conclusion that Bayesian Personalized Ranking (BPR) is one of the more effective models for item recommendation. However, each of the papers expand on BPR to improve its accuracy or applicability.

5 Conclusion

The Bayesian Personalized Ranking (BPR) model achieved a 72% prediction accuracy when tested on the full dataset. This suggests that the model is highly effective at predicting whether a user will interact with a game, based on their historical interactions and the preferences of similar users. A 72% accuracy is a strong indicator that the model successfully captures user preferences, which is a fundamental aspect of personalized recommendation systems.

When compared to simpler baseline models or heuristics—such as random guessing or popularity-based recommendations—the BPR model

clearly outperforms. These baseline models do not consider user-specific preferences, and as such, their predictions are much less accurate. BPR, on the other hand, benefits from latent user and item representations, which allow it to detect complex patterns in user behavior and make much more relevant recommendations.

Model Parameters

- Latent Factors (gammaU and gammaI): These represent the user and item embeddings in a shared latent space. A higher similarity between a user's embedding and an item's embedding suggests a stronger likelihood of interaction, which directly improves the quality of the recommendations.
- Bias Terms (betaI): These account for the global popularity of items, ensuring that frequently interacted-with items are recommended, even if they don't align perfectly with a user's specific tastes.
- Regularization Coefficient (lambda): This prevents overfitting by penalizing large weights in the embeddings, helping the model generalize to unseen data and remain robust.

Improvements and Future Work

While the current model performs well, it could be further enhanced by incorporating additional features, such as game metadata (genre, tags, pricing, etc). These features could provide more context for making recommendations and help capture user preferences that go beyond their direct interactions with games. However, due to time constraints, we were unable to implement these additional features in our model.

If these features had been incorporated, we expect that prediction accuracy could be further improved, providing even more personalized and relevant recommendations for users.

Compared to simpler models like random guessing or popularity-based recommendations, BPR's performance is significantly better, as it leverages user-item interactions and latent factors to personalize the recommendations. While matrix factorization methods might perform similarly in predicting absolute ratings, they are typically less suited for ranking tasks compared to BPR, which is optimized for pairwise ranking.

In conclusion, the proposed BPR model succeeded by capturing the nuanced relationships between users and items through latent representations. Despite the time constraints that prevented us from incorporating additional features, BPR's strong performance highlights its effectiveness in personalized recommendation tasks. Moving forward, adding more diverse features could further improve the model, but even in its current form, BPR offers a substantial improvement over simpler approaches.