SMU Data Science Review

Volume 8
Number 1 *Spring 2024*

Article 4

Game Recommendation Analysis Using Steam Profiles and Reviews

Robert Blue
Southern Methodist University, robert.blue101@gmail.com

Luis Garcia
Southern Methodist University, garcia.luis@mail.smu.edu

Jacob Turner
Southern Methodist University, jturner@mail.smu.edu

Game Recommendation Analysis Using Steam Profiles and Reviews

Robert Blue, Luis Garcia, Jacob Andrew Turner, Ph.D.
Master of Science in Data Science, Southern Methodist University
Dallas, TX 75275 USA
rblue@smu.edu
garcia.luis@smu.edu

Abstract. Smaller game studios are at a disadvantage when it comes to getting their product noticed by users. This study aims to provide insights on how recommendation engines work so that these smaller studios can have their games noticed on Steam. Steam is one of the largest video game distribution services and they have a recommendation engine which promotes games to its user base. This study utilized user information such as number of games played, the type of games, and the hours played and created recommendation engines to identify the qualities in the game that are driving recommendations.

1 Introduction

Video games as a market is expected to have explosive growth in the next 4 years. According to Statistica, by the end of 2023, the market is expected to reach US$249.60 bn. By 2028, that market cap is expected to grow to US$389.70bn. (Statistica, 2023). Fig 1 shows the explosive growth over the last 5 years of the game industry at an annual growth rate of over 9% year.

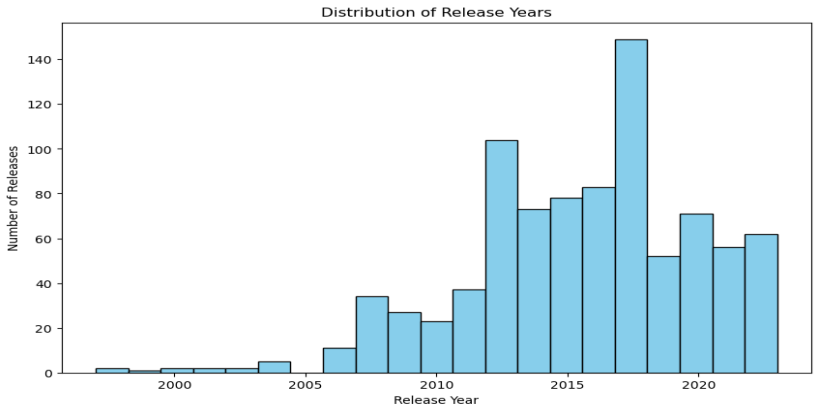

Fig. 1. A histogram of games released since 1995 and into the present day. The game release cycle as exploded since the 2010s with a heavy skew towards the later part of the timeline.

Fig. 2 highlights the top fifteen studios that have released games on Steam since 2019 with their number of releases, total owners, and total ratings. Only one studio that is classified as "indie", Endnight Game Studio, is in the top ten and the list only has four indie studios total including Re-Logic, ConcernedApe, and Klei Entertainment in the top fifteen. Among those, not a single indie publisher has sold more than 25,000,000 units whereas the top publisher, Valve, has sold 143,000,000 units alone.

developer	num_releases	sum_owners	mean_price	mean_rating	total_ratings	score
Valve	18	143000000	11.434444	89.292819	2148228	0.635890
Amazon Games	1	50000000	39.990000	69.516109	269314	-0.087039
Facepunch Studios	2	40000000	24.990000	91.189830	2014409	0.340153
CD PROJEKT RED	4	34000000	32.490000	87.568976	1593492	0.327171
DICE	5	30000000	41.990000	73.500608	635776	0.098603
Ubisoft Montreal	7	27000000	29.990000	81.258350	1411027	0.277905
CAPCOM Co., Ltd.	12	26000000	31.740000	86.585695	826283	0.425525
Larian Studios	3	26000000	48.323333	92.784024	678668	0.350214
Bethesda Game Studios	8	26000000	29.990000	83.703609	1239128	0.329254
Endnight Games Ltd	2	25000000	24.990000	89.365262	680089	0.263979
Klei Entertainment	3	24000000	16.656667	95.090899	646423	0.382189
FromSoftware Inc.	2	22000000	59.990000	90.305067	793373	0.277926
ConcernedApe	1	20000000	14.990000	97.368299	628000	0.339276
Wallpaper Engine Team	1	20000000	3.990000	97.219611	719076	0.340546
Re-Logic	1	20000000	9.990000	96.957636	1220433	0.350539

Fig. 2. Top 15 studios currently on Steam.

Steam has emerged as a dominant platform for gamers, with its recommendation engine playing a crucial role in directing users towards potential game interests. This research delves into the mechanics and intricacies of constructing a recommendation engine, meticulously designed to consider all facets of a Steam user's profile. The quantity of games owned or engaged with in specific genres by a user is a foundational metric, facilitating the comprehension of their gaming predilections. Such information proves instrumental in forecasting subsequent interests which play a pivotal role in which games the recommendation engine populates. An individual's cumulative playtime stands as an incontrovertible testament to a game's staying power and the depth of user engagement. Directing attention to this metric ensures the recommendation engine acknowledges and integrates the fervor users display for specific game categories or titles.

Beyond the dimension of gameplay, users articulate their sentiments pertaining to games via reviews. A meticulous analysis of the linguistic components and overarching sentiments contained within these reviews can shed light on the intricate faces of user contentment or dissatisfaction. This analysis can also be applied to time played and using sentiment analysis on the review itself to determine the legitimacy of said review.

A review with only a handful of hours played will not be as useful as a review with hundreds of hours played. In the same vein of analysis, a review that only critiques or compliments a small fraction of the game will not be as valuable as a review that goes into the specific mechanics or aspects that are positive or negative.

Despite the undeniable prowess of Steam's existing recommendation engine, it is characterized by discernable biases. The prevailing algorithm favors a discernable inclination towards large and already popular studios and widely acclaimed titles. Consequently, this tends to overshadow contributions from smaller studios or independent developers, thereby diminishing their presence and visibility on the platform. While such a strategy proves effective in mitigating the prevalence of "shovelware", or games known more for their volume than virtue – it also raises concerns. There is potential risk of inadvertently curtailing exposure for high-caliber content originating from entities that lack the larger backing or marketing budget which is typical for more mainstream studios and titles.

The importance of recommendations systems is not solely a feature of the gaming market. Across diverse sectors, from digital literature to movie streaming platforms, personalized recommendations exert a profound impact on product uptake. A well-structured recommendation infrastructure not only augments product prominence but also refines the overarching user interaction. Personalized consumer experiences have the potential to uplift sales figures by an estimated 20%. An overwhelming majority, approximately 80% of consumers, demonstrate a preference for e-commerce platforms that offer customized and personalized recommendation experiences (Sahin 2023). In the gaming sector, enthusiasts are inclined to write comprehensive and extended reviews for titles they harbor deep-rooted affinities for and into which they have invested considerable resources, whether it is time, money, or both. In fact, comprehensive reviews are often an outcome of elevated game acquisition costs (Lin, Bezemer, Zou, Hassan 2019).

March 2020 marked a significant juncture with Valve introducing a re-envisioned Interactive Recommendation engine. Using a foundation built on a robust machine learning infrastructure, the system gleans insights from the playtime trajectories of a vast user base on Steam. Rather than using a traditional tagging system, the emphasis is instead focused on player behavior and established patterns, forming the foundation for game recommendations. This new system still shares the same pitfalls as its predecessor, particularly a bias towards games with substantial marketing or widespread recognition (Robertson 2019). Notably, this overhaul appeared, to a significant extent, as a redressal to apprehensions voiced by smaller studios, who felt marginalized during major steam promotional periods like the Steam Summer Sale (Grayson 2019).

Given the current situation and identified gaps, this research paper hopes to answer a simple question: How do large studios get their games recommended on Steam over smaller studios?

2 Literature Review

The Literature Review focuses on three areas: Analysis of Game Reviews, NLP and sentiment analysis methods used on reviews, and matrix factorization.

2.1 Analysis of Game Reviews

Game reviews hold valuable information that can help decipher why a game was liked or disliked by a reviewer (Lin et al. 2019). Prior work analyzing words in reviews yielded associations between negative reviews after many playing hours and bad patches as well as negative reviews with few playing hours and severe bugs or bad game design (Lin et al. 2019). An analysis of player behavior concluded that gamers have little patience when it comes to faulty servers (Chambers et al. 2005), further supporting the previous association between short play hours and bad game design. One contributing factor for effectively addressing bad game designs are the type of update strategies that the developers use, of which updates that happen less than weeks apart from one another tend to have more back-to-back updates (Lin, Bezemer, Hassan 2017). Another finding was that negative reviews are usually posted with less than half of the playing time than those of positive reviews (Lin et al. 2019). Lastly, it would seem like price affects the user's willingness to rate the game differently depending on the tags (Toy et al. 2023)

An analysis of the reviews of games, considering the genre, yielded the results that game reviews vary in length and playtime when reviewed depending on the genre of the game (Guzsvinecz et al. 2023). Another observation was that positive reviews were more prevalent during Early-Release windows so during these times someone can expect the reviews to have positive language (Lin et al. 2018). Building recommendation systems while considering genres allow for recommendations to be better aligned with user's interests (Andersson, J 2022). One can still create simple recommendation systems without knowing the genre and only using data regarding what games the target user is playing (R.R 2021).

2.2 NLP

While previous game review analysis provides trends to look out for when scrapping Steam reviews, deeper dives into the reviews can be done with NLP and Sentiment Analysis for better recommendations just like Gameopedia has been doing with their sentiment analysis tool (Karthikeyan, K. 2021). Another way that NLP methods have been used are to identify nouns in reviews and pair them with adjectives to identify patterns in reviews (Zhu et al. 2015). The analysis of game reviews also used sentiment analysis with classification of words being done in accordance with the NRC Emotion Lexicon which is a list of English words and their association with sentiment and found that the intensity of like and dislike in the reviews varied depending on the genre (Guzsvinecz et al. 2023). Reviews were portioned into equal lengths and sentiment per portion was taken to analyze how sentiment differed from starting a review to ending it (Guzsvinecz et al. 2023). Incorporating sentiment of reviews when making recommendations improves the recommendation accuracy (Roy et al. 2021).

One thing to consider when analyzing sentiment in reviews is that some reviews may be fake reviews, luckily there exist ways to identify fake reviews and fake reviewers which can be used to deduce legitimate reviews (Liu, B 2012). Sentiment polarity categorization, scoring a word from extremely negative to incredibly positive, is an issue that comes with sentiment analysis but a study on Amazon reviews proposed a

three-phase process that helped researchers yield successful results for sentence and review-level sentiment analysis (Fang et al. 2015).

2.3 Collaborative Filtering and Matrix Factorization

Collaborative filtering (CF) is a foundational technique employed by large-scale recommendation systems. CF analyzes engagement data, such as user reviews and item popularity through positive feedback, to connect patterns between users. This analysis helps in the construction of predictive models that suggest items in which a particular user has liked in the past. The process involves two primary methods: neighborhood approaches, which focus on direct interactions between users and items, and matrix factorization techniques, akin to principal component analysis, which reduce data complexity by inferring latent factors that describe user and item interactions (Batra et al. 2023). However, CF is susceptible to certain biases; items with higher engagement levels often gain disproportionate visibility, potentially overshadowing less popular options. This popularity bias can initiate a feedback loop, where popular items are moved towards the top of a recommendation queue. Additionally, as the system scales up to accommodate more users and items, the complexity of these relationships increases, challenging the scalability of the model. For a deeper technical exploration of collaborative filtering, including matrix factorization, readers are directed to additional resources such as the Real Python tutorial on collaborative filtering (Ajisaria 2021) and relevant academic literature. This background can enhance understanding of the data inputs and outputs integral to CF, providing clearer insights into its operational mechanics.

To overcome these limitations, companies often use a technique called Matrix Factorization. The streaming giant Netflix employs this algorithm as do other major companies such as Amazon and Spotify. Matrix factorization is effective in recommendation systems because of many reasons. One of the major reasons for choosing matrix factorization is its ability to identify latent factors. A latent factor is a variable that can only be observed through mathematical means rather than purely observing the data. This is important because the algorithm can detect correlations that might go unnoticed which will build an even more robust user-item interaction. Matrix factorization also handles the issue with sparse data by “filling in” the missing correlations with these latent factors. This provides a robust system of recommendation based on user preferences. Matrix factorization is also scalable by using systems such as ALS (alternating least squares) which can be run in parallel across multiple environments which can optimize the output.

3 Methods

We will be using previous work that has been done in the recommendation engine systems, notably the work of Dr Julian McAuley from the University of California, San Diego (Wan, McAuley 2018). We will also be using the Steam API in order to get user and game information.

The data for this study is sourced from two different APIs as data for users and games are accessible via one single API. The Steamworks Web API contains requests used to get the list of games along with their special keys, get information about how many games a player owns and how often they play those games, and a list of which games have been recently played. The steam recommendation API contains a request that gets the data for the reviews

Tags for the games are extracted manually by using requests.get and BeautifulSoup to scrape a page in the Steam documentation website. Further data cleaning is done to results from page so that we can extract the tags for the genre and sub-genre category only. The json.loads method will be used when calling both APIs.

Before either of the APIs can be used, Steam requires that a Steam account be created and that a Steam Web API Key be generated. Steam accounts have unique steam IDs which link reviews to users and, if user is visible, allows for the Steamworks API to extract information on the user's gaming tendencies.

The first API action to be done is to extract the names of the games as well as their game ID which is saved as appid. These are extracted via the Steamworks API's ISteamApps Interface using the GetAppList method, no additional parameters need to be included. The results of the query will be saved on a Nx2 table so that the names of the games can later be associated via a secondary key to any future table. From here the webpage for the games will need to be scrapped for the tags as no API has this information. Tags from the genre and sub-genre category will be compared and only those tags will be kept.

The second thing that needs to be done with the APIs is to extract the reviews from the games. Games are queried with the getreviews method by feeding it the appid and the language parameter so that only English reviews are returned. The call returns the number of reviews, the user's steamid, the time played at the time of the review, when the review was created, if the review was during Early Access, the review itself, and many other fields that will be taken into consideration when evaluating the validity of the reviews. All games extracted from the previous API calls will be looked up and only the games with more than 100 reviews will be kept for analysis. From these games, only users who have a num_games_owned greater than 0 will be looked at since a 0 in this field indicates that though we can see their review, we do not have access to their library, and these are not users who we want to incorporate into our Matrix Factorization.

Now that the game reviews, game tags, and users with public libraries have been extracted, the user's library information can be queried. Using the IPlayerService Interface's GetOwnedGames method, the games for a user can be pulled by providing the user's unique steamid which was pulled from the reviews. This returns the total gametime for a game, the time played during the last two weeks, as well as the last time a game was played, and it includes the steam_appid. The information for the last time played is returned in Epoch format and must be converted into datetime to make sense of it and playtime is converted from minutes to hours to be more concise.

The query for the reviews disregards off-topic reviews and returns a score for the review based on usefulness which can be used to identify how thoughtful their reviews are. Intensity of the reviews will be analyzed via sentiment analysis to further give weight to them. Statistical methods will be implemented to the reviews to make sure that they have statistical significance after determining what distribution the reviews

follow as to meet the assumptions of the many models that can be used (Dror et. al 2020).

In the development of our recommendation systems, both user-based and item-based approaches were employed using data accessed via an API. Specifically, for the item-based system, the "Engagement Factor" was determined by the total time users spent playing the games, while the "Popularity Factor" was based on the number of positive ratings each game received. This data was organized into a structured dense matrix, wherein the rows represented individual games and the columns encapsulated these key metrics, alongside total downloads which aided in category differentiation and clustering. This matrix format facilitates the application of Euclidean distance measurements to identify and group games with similar attributes based on their engagement and popularity. In essence, this matrix is not merely a collection of raw data but a transformed entity that serves as a similarity matrix, enabling the efficient clustering of games sharing comparable characteristics. This approach underscores the nuanced interplay between different types of data in our system and highlights the methodological rigor in segmenting and analyzing game attributes.

Similarly, the user-based system leveraged Engagement and Popularity latent factors which were derived from individual user behaviors. Instead of using overall playtime and reviews, this system analyzed over 800 users' Steam libraries and focused only on their own respective playtimes along with the total number of positive ratings each game received. This detailed analysis helped us discover the latent factors that served as key patterns and preferences in individuals' gaming preferences and enabled us to create a predictive model. Specifically, we used a model that employs singular value decomposition (SVD), a matrix factorization technique that decomposes data into singular vectors and values. This method effectively identifies and quantifies the underlying structures in user-game interactions and builds a prediction based on these factors. We then use the predicted playtime that the model outputs to create a robust recommendation system. This predictive approach used the strengths of matrix factorization to offer personalized game recommendations, enhancing user engagement by custom-fitting suggestions to an individual's preferences and gaming habits.

4 Results

Though we limited the scope of the games being queried to games that had more than 100 reviews, the engagement that these games have vary vastly. Figure 3 below highlights the fact that though we made a deliberate choice to choose games that were getting feedback from users, the magnitude of it is not uniform.

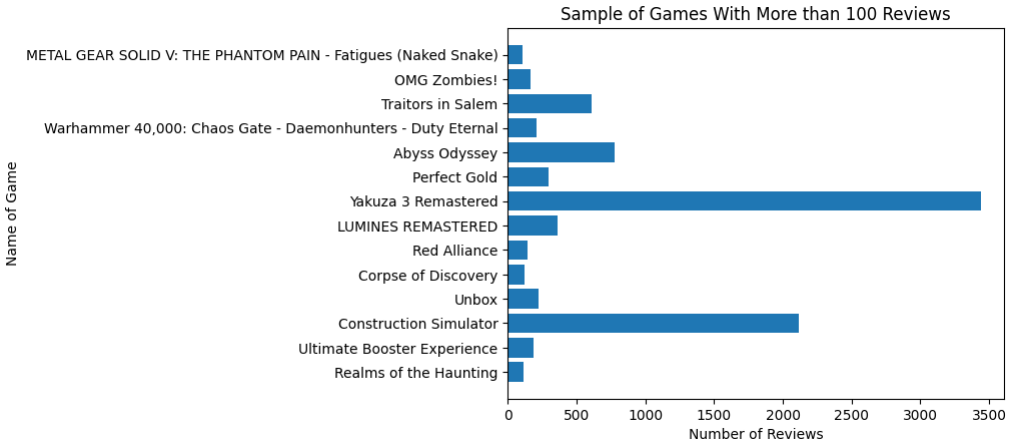

Fig. 3. Though appids have more than 100 reviews, the actual number of reviews varies.

The games in Steam are tagged with top-level genre, genre, sub-genre, and many more diverse types of tags. Figure 4 below shows 10 of the top-level genre tags, many of which were the tags present in the item matrix created to give out item-based recommendations.

Tags found in game's store page
Action
Adventure
Casual
Experimental
Puzzle
Racing
RPG
Simulation
Sports
Strategy

Fig. 4. 10 of the 423 possible tags a game could have

The SteamAPI allows users to investigate a Steam user's library and see which games they are playing and how much, but only if they share that information publicly. Figure 5 shows that the playtime for a game can be pulled from a user's library by providing the game of interest. This game to playtime pipeline was used to get the total playtime that a game had across the sample of users that had their Steam library's queried via the SteamAPI as part of this study.

```
$response
$response$game_count
[1] 1

$response$games
$response$games[[1]]
$response$games[[1]]$appid
[1] 1971870

$response$games[[1]]$playtime_forever
[1] 4785

$response$games[[1]]$playtime_windows_forever
[1] 4785

$response$games[[1]]$playtime_mac_forever
[1] 0

$response$games[[1]]$playtime_linux_forever
[1] 0

$response$games[[1]]$rtime_last_played
[1] 1700247010

$response$games[[1]]$playtime_disconnected
[1] 0
```

Fig. 5. This figure shows the output in R of the results we get when querying a specific appid, we can get all the above user information with relationship to a game that they own.

The group of interest for this study was users who had a decent number of games and playtime. Figure 6 shows what a distribution of games and playtime that they could have. Though Figure 5 above shows that the number of games a user has is provided by a request of the SteamAPI, this field was disregarded as games with 0 playtime were still being counted towards their number of games.

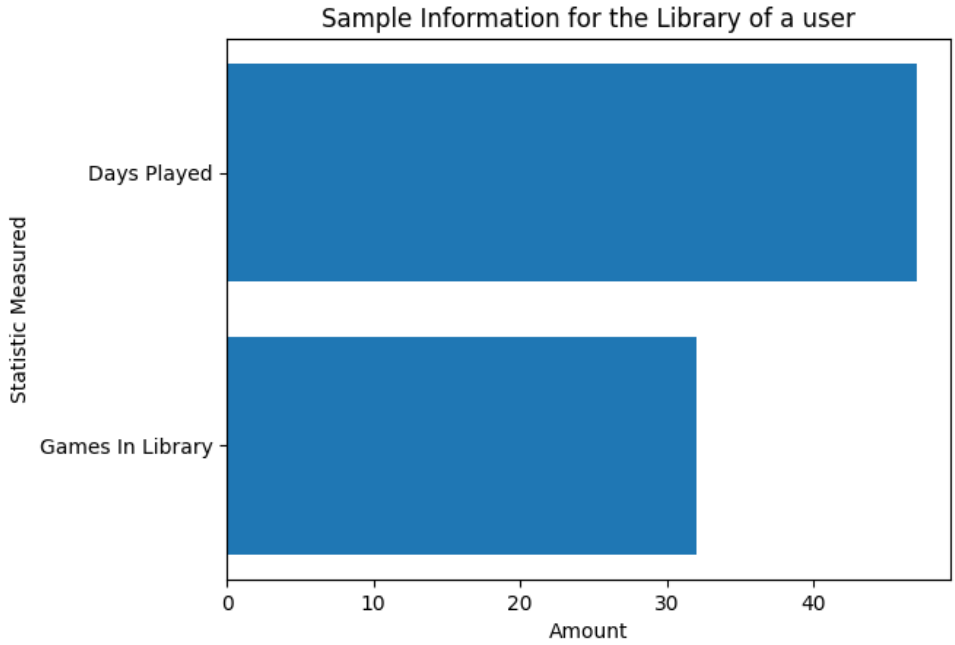

Fig. 6. This figure shows that we can find users who have public profiles and a sizable number of games played as well as a good amount of time spent playing those games. These are the users we want to target.

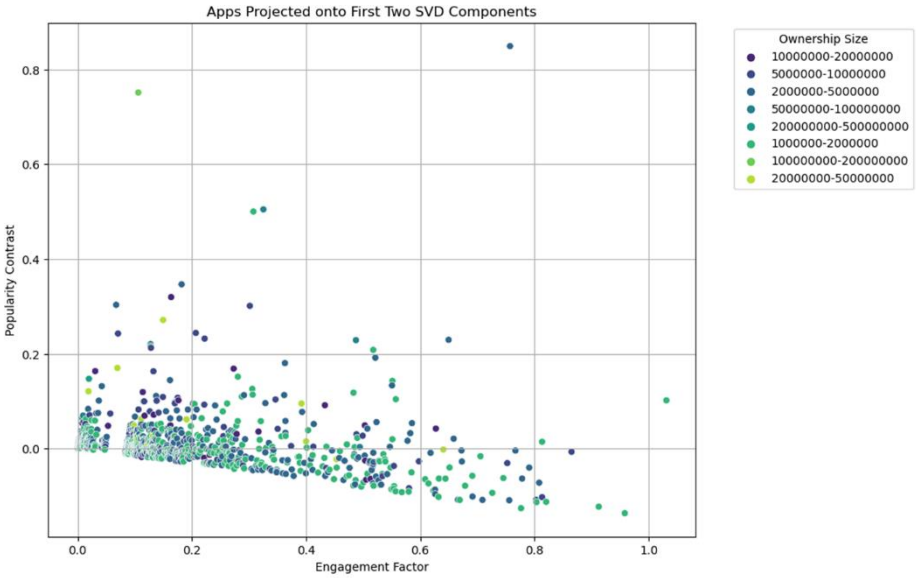

Fig. 7. This figure shows the two latent factors, engagement factor and popularity content, that

were derived after performing matrix factorization on the item-matrix that contained game information. Every *dot* is a game, and the dots are color-coded based on which cluster of *Ownership Size* it falls under to help visualize how similar games with the same number of downloads are. By using the values of the latent factors, distances can be derived between any game

From the games shown in Figure 7, 'Counter-Strike' was used as a baseline and the 5 closest games to it were derived. Table 1 below shows the games that had the lowest Euclidian Distance to 'Counter-Strike' and what that distance is. Given that the playtime and number of positive ratings were the latent factors, these recommendations are driven by user interaction as opposed to item-similarity, especially since its sequel, 'Counter-Strike 2', is not a recommendation.

Table 1. This table shows the 5 games wit the lowest Euclidean Distance from 'Counter-Strike'. The Euclidean Distance was derived using Engagement Factor and Popularity Content values from Figure F as inputs.

appid	name	Distance
400	Portal	.002737
320	Half-Life 2: Deathmatch	.018651
620	Portal 2	.033383
108600	Project Zomboid	.057210
272060	Serena	.115725

While the previous recommendations, the item-based recommendations, were derived by taking a game's data and comparing it to other games based on how similar game's total playtime and positive ratings were, the user-based recommendations leverage user data to give predictions. A user-item matrix is created based on user playtime statistics. Every row in the matrix is a specific user and every column in the matrix is a specific game. The entries in the matrix are that user's total playtime for a game. The result of performing matrix factorization on this is two new matrices, one which holds the latent factors for the user and another that holds the latent factors for the games. These matrices were used to generate the predicted playtime for games that a user had not played. Using the user specific predicted playtimes that the SVD helped derive and the positive ratings for a game, users were given catered recommendations. Table 2 and Table 3 show Heart of Iron IV as a recommendation but the actual predicted playtime for it varies because the user playtime statistics differ. It is evident that a game's popularity is driving recommendations as most games that are being recommended were either critically acclaimed at one point or have an active player

base, but the games here have varying popularity as opposed to the ones from the previous model.

Table 2. This table shows the recommended games, indicated by the appid, that user ####6393 received. These recommendations were heavily influenced by how much a game was played and how positive the ratings for said game were. A notable observation is that the game in row 2 has fewer positive ratings and more predicted playtime than the games in row 1 and row 3.

Top 5 Recommended Games for User with Steam ID :####6393			
Appid	AppName	Predicted Playtime	Positive Ratings
218620	Payday 2	2857.33061	576695
394360	Hearts of Iron IV	4558.112466	238180
72850	Elder Scrolls V: Sl	1079.144022	298564
8930	Sid Meier's Civiliz	1688.047861	187555
220200	Kerbal Space Pro	2231.55408	112151

Table 3. This table shows the recommended games, indicated by the appid, that user ####1332 received. Comparing this to Table 2, the game with appid 394360 is seen in both but the predicted playtime and rank of the recommendation vary between us

Top 5 Recommended Games for User with Steam ID :####1332			
Appid	AppName	Predicted Playtime	Positive Ratings
730	Counter Strike 2	917.73233	6816643
236390	War Thunder	1540.872027	346183
107410	Arma 3	2230.816057	232581
381210	Dead by Daylight	747.028905	563143
394360	Hearts of Iron IV	887.495217	238180

5 Discussion

The analysis explores the effectiveness of a recommendation system compared to Steam's existing mechanisms, focusing on the representation of indie studios. It raises the question of whether companies featured on Steam should utilize public information to target potential customers, weighing the risks and benefits of such strategies. The study also highlights unique findings and challenges encountered, particularly the computational difficulties in processing Steam's extensive game catalog and user reviews. With almost 200,000 games and some having over 100,000 reviews, the task was daunting and time-consuming.

A deliberate decision was made to exclude users with limited game libraries or playtime, acknowledging that this choice omits a segment of the Steam population. The study faced limitations in data querying, especially with the Steam API's restriction of displaying only 100 reviews at a time, necessitating multiple requests. The methodology involved using a relatively small sample size—860 games and 1,000 users' library information—deemed sufficient for understanding and developing the recommendation engine despite its minor representation of the vast Steam ecosystem.

The focus was primarily on users with extensive playtime and large game collections, specifically filtering for games with 100 or more reviews to exclude niche or downloadable content only games. This approach ensured the study concentrated on games within the indie category rather than a broader, less defined game set. The limitations extend to the consideration of only games with numerous reviews, inadvertently excluding new releases and requiring manual efforts to gather game tags due to API constraints. Additionally, only English language reviews were analyzed, resulting in a significant data exclusion.

5.1 Ethics

From an ethical standpoint, the study aims to support indie studios overshadowed by larger entities by and maintains user privacy through partial omission of Steam IDs. This measure respects user privacy while acknowledging the public availability of such data. There are many ethical concerns with being able to identify users that may be drawn to a product.

As shown with the item-based recommendation which was created from the matrix factorization of the game information of 19 features were broken down into latent factors, items can be used to recommend other items, so by using different inputs one could easily use users to recommend other users. The information available via the Steam API allows companies the chance to find users that have not touched their product and find their Steam pages. Companies can easily find the users that they have provided the most profit to them, be it through seeing how much they've spent on microtransactions or how many games of theirs they have purchased. Once these users have been identified, they could have their libraries queried to create a user-item matrix that would later be factorized. From here all that would need to be done would be for the companies to sample random user's libraries in a similar fashion to find out which users who have not purchased their product are most like the users who are generating much revenue for their product. As of now, there does not appear to be any indicator that Steam allows companies to directly promote their product to specific users but if that feature is ever added then it would easily be exploited through the method explained above.

5.2 Future Research Opportunities

Though videogames are a good way to pass the time, using the amount of time that someone would spend playing a videogame to recommend a game to them could be detrimental. Say the predicted playtime for a user is 18 hours a day for a game that they do not own. If a recommendation of said game is given to the user and they happen to

abide by the predicted behavior, then that is unethical to do. The deliberate recommendation of a pass time that would consume most of someone's day with the sole aim of company revenue is purely greed. Users should be recommended items with some limitation of predicted playtime to not have negative impact on their productivity or their ability to conduct other standard daily routines such as eating, sleeping, and showering.

Future research directions include expanding the dataset to encompass games with few or no reviews, unreleased titles, and leveraging information from reviews more comprehensively, such as timing and content update item

6 Conclusion

In our analysis, we observed that the Popularity latent factor was more heavily weighted compared to the Engagement factor in both the item-based and user-based recommendation systems. This trend underscores a notable challenge in large-scale matrix factorization recommendation systems, where games that are made by studios with higher marketing budgets and higher visibility tend to be favored. This bias puts smaller and independent studios at a disadvantage because their games may become overshadowed by the larger studios and may not even be featured in the store front. The lower visibility will in turn lead to lower sales and will drive the demand further down.

However, this analysis also presents a potential strategic opportunity for these smaller studios. By understanding the mechanics and makeup of how Steam's recommendation system works where popularity highly influences game sales and recommendations, developers and marketing teams can target strategies and broader outreach efforts to improve their visibility. Tactics such as social media outreach, engagement with prominent streamers and other video game influencers, or increasing their social footprint can boost their engagement and help the studios focus their efforts on the right areas. By using these tactics, they can potentially work around the recommendation algorithms and help level the playing field.

Acknowledgments.

Jacob Andrew Turner, Ph.D. – Advisor

7 References

1. Rizani, M. N., Khalid, M. N. A., & Iida, H. (2023). Application of Meta-Gaming Concept to the Publishing Platform: Analysis of the Steam Games Platform. Information, 14(2), 110. https://doi.org/10.3390/info14020110

2. Sahin,B. (February 2023). E-Commerce Personalization: Your Complete Guide. Bloomreach. https://www.bloomreach.com/en/blog/2017/ecommerce-personalization.

3. Introducing The Steam Interactive Recommender. (March 2020). https://steamcommunity.com/games/593110/announcements/detail/1716373422378712841

4. Robertson, A. (July 2019). Steam's new Interactive Recommender is built for finding 'hidden gems'. The Verge. https://www.theverge.com/2019/7/11/20690231/valve-steam-labs-interactive-recommender-game-recommendation-machine-learning-tool.

5. Grayson, N. (July 2019). This Year's Steam Summer Sale Was A Mess, Game Developers Say Kotoku. https://kotaku.com/this-years-steam-summer-sale-was-a-mess-game-developer-1836215859

6. Lin, Dayi & Bezemer, Cor-Paul & Hassan, Ahmed E.. (2018). An Empirical Study of Early Access Games on the Steam Platform. Empirical Software Engineering. 23. https://doi.org/10.1007/s10664-017-9531-3

7. Lin, D., Bezemer, CP. & Hassan, A.E. Studying the urgent updates of popular games on the Steam platform. Empir Software Eng 22, 2095–2126 (2017). https://doi.org/10.1007/s10664-016-9480-2

8. Seif El-Nasr, M., Drachen, A., & Canossa, A. (Eds.) (2013). Game Analytics: Maximizing the Value of Player Data. Springer. https://doi.org/10.1007/978-1-4471-4769-5

9. Ajisaria, Abinava. "Build a Recommendation Engine With Collaborative Filtering". https://realpython.com/build-recommendation-engine-collaborative-filtering/. 2021

10. Lin, Dayi & Bezemer, Cor-Paul & Zou, Ying & Hassan, Ahmed E.. (2019). An Empirical Study of Game Reviews on the Steam Platform. Empirical Software Engineering. 24. https://doi.org/10.1007/s10664-018-9627-4

11. Karthikeyan, K. (2021, November 2). Gameopedia. Improving Video Game Recommendations: Addressing Challenges and Opportunities in E-Commerce. https://www.gameopedia.com/problems-with-game-recommendations/

12. Grayson, Nathan. (2019, September 12). Kotaku. Steam's Recommendations Will Now Show Popular Games Less Often. https://kotaku.com/steams-recommendations-will-now-show-popular-games-less-1838073592

13. SirTapTap. (2016, September 16). SirTapTap. Steam's New Review Policy Causes More Problems Than It Solves. https://sirtaptap.com/articles/steams-new-review-policy-causes-more-problems-than-it-solves/#google_vignette

14. R.R. (2021, May 9). Using Steam's data to find and recommend similar games. Video Game Recommendation System. https://medium.com/web-mining-is688-spring-2021/video-game-recommendation-system-b9bcb306bf16

15. Singhal, A., Sinha, P., & Pant, S. (2017, December) Use of Deep Learning in Modern Recommendation Systems: A Summary of Recent Works. https://www.researchgate.net/publication/321846860_Use_of_Deep_Learning_in_Modern_Recommendation_System_A_Summary_of_Recent_Works

16. Guzsvinecz, Tibor & Szűcs, Judit. (2023). Length and sentiment analysis of reviews about top-level video game genres on the steam platform. Computers in Human Behavior. 149. https://www.researchgate.net/publication/373882591_Length_and_sentiment_analysis_of_reviews_about_top-level_video_game_genres_on_the_steam_platform

17. Chambers, Chris & Feng, Wu-chang & Sahu, Sambit & Saha, Debanjan. (2005). Measurement-based Characterization of a Collection of On-line Games. https://www.researchgate.net/publication/228956555_Measurement-based_Characterization_of_a_Collection_of_On-line_Games

18. Sahin, B. (2023, February 2). E-Commerce Personalization: Your Complete Guide. Real-Time Personalization. https://www.bloomreach.com/en/blog/2017/ecommerce-personalization

19. Dror, R., Peled-Cohen, L., Shlomov, S. & Reichart, R. (2020). Statistical Significance Testing for Natural Language Processing. Springer. https://doi.org/10.1007/978-3-031-02174-9

20. Cohen, S. (2019). Bayesian Analysis in Natural Language Processing, Second Edition. Springer. https://doi.org/10.1007/978-3-031-02170-1

21. Liu, B. (2012). Sentiment Analysis and Opinion Mining. Springer. https://doi.org/10.1007/978-3-031-02145-9

22. Jackson, D. (2017, July 7). The Netflix Prize: How a $1 Million Contest Changed Binge-Watching Forever. Thrillist. https://www.thrillist.com/entertainment/nation/the-netflix-prize

23. Roy, D., Ding, C. (2021). Multi-source based movie recommendation with ratings and the side information. Soc. Netw. Anal. Min. 11, Article 76. https://doi.org/10.1007/s13278-021-00785-5

24. Zhu, M., Fang, X. (2015). A Lexical Analysis of Nouns and Adjectives from Online Game Reviews. In: Kurosu, M. (eds) Human-Computer Interaction: Interaction Technologies. HCI 2015. Lecture Notes in Computer Science(), vol 9170. Springer, Cham. https://doi.org/10.1007/978-3-319-20916-6_62

25. Fang, X., Zhan, J. (2015) Sentiment analysis using product review data. Journal of Big Data 2, Article 5. https://doi.org/10.1186/s40537-015-0015-2

26. Andersson, J. (2022). Statistical Methods in Recommender Systems. https://doi.org/10.1007/978-1-4842-7765-2_7

27. A. K. Balazs Hidasi, "Session-based Recommendation with Recurrent Neural Networks," ICLR, pp. 1–10, 2016. https://www.researchgate.net/publication/284579100_Session-based_Recommendations_with_Recurrent_Neural_Networks

28. Davis, N., Steam Data Exploration (2019), GitHub, https://github.com/nik-davis/steam-data-science-project

29. *Video games - worldwide: Statista market forecast*. Statista. (Dec 2023). https://www.statista.com/outlook/dmo/digital-media/video-games/worldwide

30. M. Wan, J. McAuley, "Self-attentive sequential recommendation" (2018), *ICDM,* https://cseweb.ucsd.edu/~jmcauley/pdfs/icdm18.pdf

31. Batra, S., Sharma, V., Sun, Y., Wang, X., & Wang, Y. (2023). Steam Recommendation System. *arXiv preprint arXiv:2305.04890*

32. W. Kang, J. McAuley, "Self-attentive sequential recommendation" (2018) *ICDM,* https://cseweb.ucsd.edu/~jmcauley/pdfs/icdm18.pdf

33. A. Pathak, K. Gupta, J. McAuley, "Generating and personalizing bundle recommendations on Steam" (2017), *SIGIR*, https://cseweb.ucsd.edu/~jmcauley/pdfs/sigir17.pdf