UNIVERSITY OF EXETER

Computer Science Department

B.Sc. Computer Science and Mathematics
Computer Science Department

Speedrunning Under The Eyes Of Machine Learning

Candidate

Alexander Jack Christopher Merren

Student ID 227696

Supervisor

Dr. Chico Camargo

University of Exeter

Academic Year
2022/2023

Abstract

The internet has allowed the transformation of real-life social circles to online social platforms, which aim to connect users similar to real-life interactions, but at a grand scale. This scale allows people to connect on previously uncommon behaviour. A recent phenomenon that can studied from this perspective is speedrunning: the practice of completing a videogame as quick as possible without cheating. Few studies quantitatively assess the act of speedrunning and the community surrounding the practice. We introduce a first perspective on investigating and interrogating the behaviour of users on speedrun.com. This project aims to determine communities of users on speedrun.com, and to create a game recommendation system using behavioural data from speedrun.com. We start by investigating community detection methods on a user-user and a user-game network, and determine the similarities between users and games within these communities. Second, this project implements content-based and collaborative filtering on users' game choice to identify trends and patterns in user behaviour. This project produces clusters of users and games, and the similarity between these clusters. Likewise, we create a game recommendation system, and a novel dataset of users and games from speedrun.com. This project finds users usually speedrun games to based on their platform of release, game genre, game mechanics, and specific game franchise. Similarly, accurate game recommendations can be made using these same criteria. This study provides an original application of community detection and recommendation system methods to the videogame community.

	Yes	No
I certify that all material in this dissertation which is not my own work has been identified.	☑	☐
I give the permission to the Department of Computer Science of the University of Exeter to include this manuscript in the institutional repository, exclusively for academic purposes.	☑	☐

Contents

1 Introduction

In February 2022, FromSoftware released Elden Ring: the highly anticipated next installment of the amazingly popular franchise, Dark Souls. Boasting an impressive 57 hours of content to reach the end credits, it is entirely unexpected that six months later, the game was completed in just under four minutes — without cheats of any kind. The people responsible for this achievement are known as speedrunners: players that aim to complete a videogame as quickly as possible without cheating.

This style of playing games has become a sensation within the entertainment and videogame industries, and has consequently created many online communities where players gather to compete and collaborate with each other. The largest of these communities is speedrun.com, a platform that supports over 1.5 million of these players. Viewing these online communities from the perspective of machine learning and data science, they are a heretofore unexplored source of user behavioural data. Therefore, this dissertation aims to investigate the user behaviour and game choices of users on the largest community, speedrun.com, using artificial intelligence and data science.

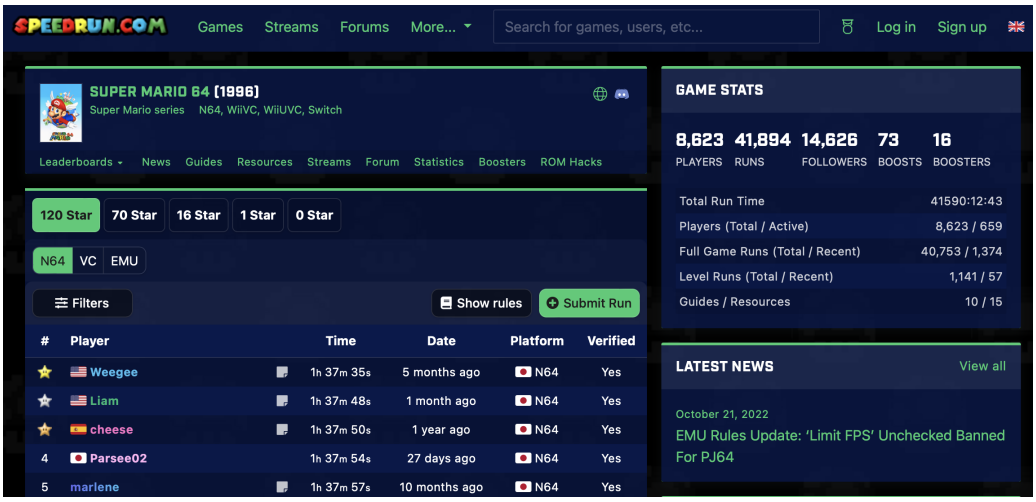


Figure 1: The leaderboard page for Super Mario 64's 120 star category, filtered by runs on the N64.

1.1 Motivation

There is a broad literature of analysing social networks with data science and machine learning methods. Methods such as community detection and recommendation systems are designed to investigate and provide a practical application of user behavioural data. However, speedrunning has not typically been the subject of machine learning methods. Current literature focuses on elucidating the ethos and principles behind the practice of speedrunning, rather than data analysis [20, 38, 12]. Previous attempts at analysis have focused on the evolution and growth of speedrunning community, rather than understanding the user's behaviour [36].

The videogame industry is currently worth approximately $200 billion, more than all other types of media combined [32]. User behaviour analysis, such as community detection and recommendation systems, provide insight on the trends and demands of the videogame community [41]. Publishers and developers may then use these insights to make data-driven decisions.

This project aims to contribute to the wider literature by analysing the user behaviour on speedrun.com in an attempt to understand *why* users choose the games they play. This will be partly determined via community detection methods to identify the similarities between users on speedrun.com. Recommendation systems will help determine if this data is suitable for identifying patterns in user behaviour.

1.2 Aims and Objectives

The aim of this project is to study the users and games of speedrun.com. Specifically, to investigate and understand the behaviour of users on speedrun.com, and to determine patterns in the choice of games that users play. Additionally, the project aims to make speedrunning data available to enable further research in this topic. This project uses machine learning methods to cluster and compare user behaviour data to determine if user behaviour can be characterised. Online version control systems are used curate and distribute the data source to encourage further research.

There are a few objectives that are necessary to satisfy the aims of the project. These objectives are formatted as research questions, that help structure the results and discussion in this dissertation:

1. Are there communities of users and games on speedrun.com? If users and games are clustered in distinct communities, then community detection methods will be able to identify these communities accurately.
2. What do the users and games in each community have in common? If the features of both users and games are consistent in an entire community, then different patterns of behaviour can be identified in each community.
3. Can a game recommendation system be created using speedrunning data? If user preferences can be determined, then the games that users may play will be determined using user-user or game-game similarity.
4. Are game recommendation systems using this data accurate? Moreover, what methods of recommendation perform better? If a game recommendation system has a reasonable accuracy, then different clusters of users will be determined from their game preferences.
5. What do games and users on speedrun.com look like? If the features of users and games are analysed, then both can be characterised and contribute to understanding the overall community of speedrun.com.
6. Can this project produce a publishable data source for future investigation? If the data can be structured to enable distribution, will this entice further investigation into the subject of speedrunning?

These objectives will be met if the research questions can be answered and accompanied with sufficient evidence. First, a list of communities and the trends within the games and users of each community must be produced. Likewise, the recommendation engine must reach a satisfactory accuracy level and produce meaningful recommendations. Finally, the data used in this research must be published online and available for download.

First, the report provides the necessary context to understand the current literature surrounding speedrunning, community detection, and recommendation systems. This is followed by the Design chapter, which describes the resources used and created to ensure the success of the project. The fourth chapter describes the specific implementation of the systems described in the design section, and discusses how these implementations are evaluated and tested. The results are presented and discussed in the fifth and sixth chapters respectively. To conclude, the limitations of the report are discussed, and further avenues for research are proposed.

2 Background Context

To understand the proposed research in this project we must discuss the history, purpose, and concepts of speedrunning. Furthermore, to evaluate the performance of community detection algorithms and recommendation system methods, it is necessary to understand the underlying functionality and the evaluation metrics of each technique.

2.1 Speedrunning

The fundamental purpose of speedrunning is to 'complete' a videogame as fast as possible. To achieve this, players use specific strategies to reach a defined point of completion. Some strategies follow unintended routes to save time, or exploit glitches to skip sections of a videogame's story. Videogames are divided into different categories where the definition of 'completion' changes [20]. Levels are smaller subdivisions of categories and apply a different objective to a segment of a full videogame. Some common definitions of completing a videogame are to reach the end credits (Any%) or to collect all items (100%). Any given attempt to speedrun a game is also known as a 'run'. All players are competing with each other on leaderboards, which are organised by game, category, and level. The top run by every user in a combination of game, category, and level are displayed on the leaderboards, and the fastest run on any leaderboard is called a 'world record'. The discovery and and perfection of speedrunning strategies is a form of cumulative evolution in the community, and is represented by the history of a world record.

Speedrunning is performative in nature, as video evidence is required to submit a run on any leaderboard [38]. Previous research agrees that speedrunning has three components: physical mastery, collective knowledge, and subversion of expectations [20, 38]. This ethos of speedrunning attracts many videogame players, as it is a natural consequence of playing videogames. The physical mastery and collective knowledge components inspire the community of speedrunners to be competitive and collaborative. All game-breaking discoveries are shared and celebrated, unlike other communities where glitches are withheld to gain a competitive advantage [38]. Most prior research has focused on the qualitative aspects of speedrunning; literature has determined the philosophy of speedrunning, analysed how speedrunning has grown since it's conception in the 1990's, and classified different types of speedrun [38, 20, 37]. However, research has rarely performed quantitative analyses of the online communities of speedrunners. Additionally, previous studies surrounding the speedrunning community are limited in scale and only concern a small portion of speedrunners.

2.2 Network Analysis and Community Detection

Online social networks have become more and more popular and facilitate user interaction. Traditionally, social networks represent the relationships between entities where entities may be 'people' and the relationships are 'friendship' [28]. Real-world social networks naturally exhibit a community structure— the process for discovering this structure is known as community detection [3]. Communities are defined as groups of entities that are closer to each other than other entities in the network. This property is called homophily, where contact between similar people occurs at a higher rate than among dissimilar people [26]. This characteristic may be exploited to discover trends within communities or to group customers with similar interests to increase sales [41].

Community detection methods are either supervised or unsupervised, depending if some or all nodes are labelled [7]. Unsupervised learning methods aim to group similar objects without prior knowledge of them [3]. Three examples of unsupervised community detection algorithms are Louvain, Clauset-Newman-Moore, and Infomap. Community detection algorithms aim to maximise the modularity of the communities: a scalar value between 1 and -1 that measures the density of links inside communities as compared to links between communities [27]. Modularity is both a method of comparing the quality of communities, and an objective function to optimise [4].

The Louvain algorithm is a two-phase agglomerative algorithm: separating each node into a single community and merging communities if there is an increase in overall modularity. This is

repeated until there is no improvement in modularity by combining communities. The second phase of the algorithm is to create a meta-network so communities are a single node. The weighted edges between these new nodes are the sum of the weighted edges of the original community. These phases are repeated iteratively until a maximum modularity is obtained [4]. The time complexity of this algorithm is unknown. However, $O(m)$ and $O(n \log^2 n)$ are observed empirically, where n is the number of nodes and m is the number of edges. [3, 25].

Clauset et al. developed the Clauset-Newman-Moore algorithm as a further optimisation of a previous algorithm [8]. It uses a greedy modularity heuristic and operates like the first phase of the Louvain algorithm. Each node is part of a community, and communities are merged if they produce the largest increase in modularity. This is repeated until a single, large community is produced. The algorithm returns the communities with the largest modularity value. The time complexity is $O(n \log^2 n)$, equal to the Louvain algorithm. [8]

The Infomap algorithm by Rosvall et al. uses a separate methodology from the Clauset-Newman-Moore and Louvain algorithms. The method uses a series of random walks to generate a 'cookbook' of codes to represent nodes in the network via Huffman coding. This encoding of nodes is then the input of hierarchical clustering to identify clusters of nodes [35]. This allows the algorithm to describe the network with as few bits as possible [25]. The time complexity is $O(n \log n)$ on average [35].

The performance of community detection methods is measured via quality functions which sum the qualities of nodes within a partition. Examples of these quality functions are modularity, coverage, and performance. Modularity has been defined earlier as the density of links inside communities compared to links between communities. Performance measures how many pairs of nodes are within the same community, and how many non-pairs are within different communities. Coverage measures how disconnected each of the clusters are by the ratio of number of edges in communities to the total number of edges [14].

Centrality analysis is another branch of network analysis aiming to identify the flow of traffic through a network [6]. Various centralities have been devised to identify different structures within a network such as betweenness, degree, PageRank, and Hubs and Authorities (HITS). Degree centrality measures the popularity of a node based on it's degree. HITS uses the degree of nodes to define hubs and authorities, where authorities are the most relevant nodes in a network, and hubs are connected to many authorities. PageRank measures centrality by how often a simulated 'user' will visit a node, and betweenness centrality determines how many times a node is used in other nodes' shortest paths [6, 31].

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j) \qquad P(P) = \frac{|\{(i,j) \in E, C_i = C_j\}| + |\{(i,j) \notin E, C_i \neq C_j\}|}{n(n-1)/2}$$

(a) Modularity (b) Performance

Figure 2: The equations for modularity and performance of communities.

2.3 Recommendation Systems

To cope with the velocity, variety, and volatility of data on online social networks, recommendation systems have been employed to improve user experience by personalising content or creating new opportunities for marketing strategies. There are two popular methods used in recommendation systems: content-based filtering and collaborative filtering [11]. Each user can be represented by the items that they have interacted with previously. Collaborative filtering recommends items to users by

recommending items that are highly rated by similar users. Content-based filtering methods recommend items that are similar to items a user has interacted with previously [24]. The recommendation functions within these methods range from the using neighbours of a bipartite user-item graph, or similarity functions using the features of user-item matrices [11].

Recommendation systems, particularly content-based and collaborative filtering, suffer from data sparsity and cold-start problems. Data sparsity is the issue of finding a reliable set of users who are similar to a target user. The cold-start problem refers to the issue of generating accurate recommendations for new users that do not have previous interests. Extra information is required to fix these problems, such as modelling social trust between users [16].

Previous implementations of recommendation systems aim to predict a specific rating for a given user, not a binary categorisation of *will play* or *will not play*. Therefore, previous implementations use evaluation metrics such as coverage and Mean Average Error (MAE). In this context, coverage is the percentage of recommendations that are relevant to the users. MAE is the mean absolute difference between predicted ratings and an actual rating [17]. Classification recommendation systems use metrics that determine the 'relevance' of recommendations. Accuracy and recall are both measures of relevance. Accuracy measures the number of recommendations that a user has chosen or selected, and recall is the percentage of the total number of games a recommendation system can recommend [21].

3 Design

This section focuses on formulating an experimental design that will fulfil the aims and objectives of this project: to study the behaviour of speedrunners on speedrun.com. This section initially describes the data source used in this project, and justifies the use of specific technologies to interpret the data. The latter half describes the research questions and the evaluation critera related to the aims and scope of the project.

3.1 Data Source

The chosen data source is the publicly available speedrun.com API. The owners of this API, Elo Entertainment, report a total of 1,456,276 users that have submitted 3,530,637 runs on 32,089 games [23]. This data source is accessed via REST and formatted using JSON. This source has the easiest method of access and the largest amount of data available. Other online speedrunning communities, such as TwinGalaxies [15] and SpeedDemosArchive [1] are available, but they may not suitable for collection as these communities have a small amount of data compared to speedrun.com. Additionally, they do not have public REST APIs and would require invasive forms of data collection such as web-scraping. These platforms have a reasonable number of users, but were not studied in the scope of this project.

The API is separated into several endpoints, each representing a different entity of the speedrunning community. The core aspect of speed running — the runs — are available on their own endpoint. These are filtered by user ID, and lists all verified, submitted, and rejected runs by each user. Moreover, the leaderboards for individual games are available, which can be filtered by category and level IDs. By definition, the number of runs by each user is greater than the number of runs on the leaderboards. There are also individual endpoints for metadata of games, user profiles, categories, levels, and game developers. Some endpoints require authorisation to access, but only for resource creation, not resource retrieval. This authorisation is implemented using an API key that must be included in the request header. For all endpoints there is a rate limit enforced: only 100 requests

can be made per second and any requests that are sent over the limit will have a response code of 420. No personal data is available through speedrun.com's API. All information on the platform is non-identifiable as defined by the current standards (GDPR), and sensitive user information, such as location data, is voluntary.

This data source does lack resources that may have contributed to the success of the project. Notably, there is not information regarding the interests of the users, or what communities they may identify with. The community data could be used to create base truth communities, where generated clusters of users and games could be compared to the base truth to judge their accuracy. The patterns of user behaviour could be compared to their interests to judge how similar their patterns are to their reported interests.

3.2 System Design

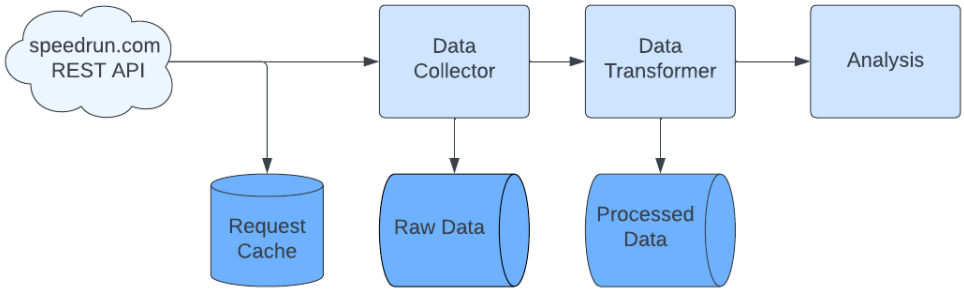


Figure 3: Flow of data through the collection, transformation, and analysis systems.

To satisfy the objectives of the project, data from the speedrun.com API must be collected, transformed, and analysed. There is a vast amount of data available through the speedrun.com REST API, so robust systems of collection, processing, and analysis were required to ensure the accuracy and completeness of the data, and the correctness of the analysis.

A modular design of the three systems was chosen due to the outputs desired from each stage of the process. Specifically, the collection system requests and formats data from the speedrun.com API to be easily distributable. The transformation system processes the raw data to produce particular formats of the raw data for easy analysis. Finally, the analysis system produces data and insights from the previous stages to present and discuss in this report.

3.3 Technology Choice

All code from the collection to the analysis stages was written in Python. Python has a large community of users which maintain libraries that are essential for data analysis and data science. It also allows for rapid prototyping and development as it is an interpreted language with an extremely simple syntax.

The data management and exploratory analysis used particular modules such as pandas, numpy, and matplotlib [42, 19, 22]. These packages greatly improve the developer experience of data analysis by providing fast easy data management, mathematical operations, and chart drawing capabilities respectively. They are incredibly popular packages with a combined 291,914,314 downloads in April 2023 alone [13].

The network analysis used the networkx, graph_tool, cdlib, and sknetwork [18, 34, 30, 5] modules. Networkx provided many base functions such as community detection methods and basic graph analysis tools. Graph_tool had similar functionality with increased speed, as it is implemented in

C++. This package was used for particularly expensive operations, such as centrality analysis. Both cdlib and sknetwork were used for other implementations of community detection algorithms, to verify and contrast the communities generated by each implementation.

For the game recommendation engine, sklearn was used to generate train-test splits of the transformed data due to its simple developer interface [29].

3.4 Success Criteria

The success of the data collection system will be judged on the volume and value of the data collected from the speedrun.com API. If there is a large amount of data that is comparably accurate to the leaderboards on speedrun.com, then the collection system will be deemed successful. Likewise, the transformation and analysis systems will be successful if they provide an easy method to transform the raw data so that insights can be gathered. The transformation system must be able to create reproducible data, which is then used by the analysis system to produce valuable insights.

The success of community detection methods will be determined both quantitatively and qualitatively. Intrinsic metrics such as modularity, performance, and coverage quantitatively determine the similarity or dissimilarity between users and games. Due to a lack of base truth for the communities of speedrun.com, extrinsic methods of evaluation cannot be used. Modularity, performance, and coverage provide a value between 1 and 0, which are their best and worst values respectively. Additionally, the contents of the communities will determine how focused the clusters are and help determine what their commonality is.

The efficacy of the recommendation systems will also be determined both quantitatively and qualitatively. In real-world applications, metrics such as click-through rate (how many recommendations are clicked) and conversion rate (how many recommendations lead to a submitted run) determine the accuracy of these systems [10]. Creating and deploying such a system is not within the scope of this project, so we must rely on other metrics such as predicting what other games a user will play given a single game. Precision, recall, and F1-score will be used to examine how relevant the recommendations are to the users of speedrun.com. Precision measures how similar the set of recommended games and the set of played games are. Furthermore, recall measures how many games from the set of recommended games have been played by the users. The F1-score is a balanced measure between both the recall and precision. The recommendations will also be analysed to examine the diversity, accuracy, and relevance of each recommendation. Recommendation systems suffer from the "cold-start" problem, where lesser-known items are recommended less [11]. A more diverse set of recommendations may decrease the effect of this problem.

3.5 Limitations of the Design

There are several limitations to the design of the project. First, a single data source is being used where there are multiple online communities of speedrunners. These other data sources may contradict each other in their community structure or user behaviour. In any case, this project provides a first impression of research in this topic. Another limitation is the lack of base truth communities of the users and games of speedrun.com. The accuracy and validity of communities must then be evaluated using internal indices, rather than comparing them to pre-labelled communities. Likewise, this project will derive hypotheses on the formation of these communities, which can be confirmed with further research. Finally, there are time and budget constraints on this project: it may not be possible to collect all necessary data as thousands of requests to the data source are needed. The design of the collection system is curated to mitigate this risk by caching requests. By publishing the data, it can be expanded upon to be included in further research.

4 Methods and Implementation

4.1 Data Collection

The data collection system was written in Python, using the srcomapi [39] module for interacting with the speedrun.com. The srcomapi module encapsulates the structures and endpoints of the REST API and allows an easier developer experience. The responses from the REST API were stored in a SQLite database via the requests_cache [33] module. The requests to the REST API were only processed if the response code was 200 (a successful request). Any other errors from the request— apart from a Not Found error— triggered repeated requests until the original request was successful. The requests were repeated to automate as much of the collection process as possible. Likewise, responses were cached to reduce the effect of repeated requests on the rate limit of the API. Subsequent requests could be fulfilled using a cached response instead of sending redundant requests that were previously successful. Individual 'feature' classes were responsible for processing and storing a single type of data, using composition for utility classes such as requesting logic. The collection system wrote directly to a raw data directory to enable distribution for further research and analysis.

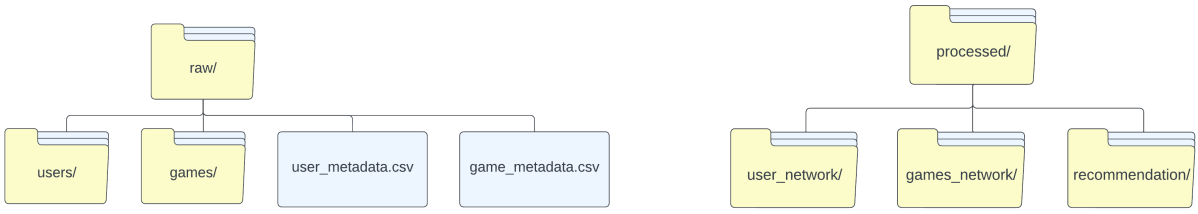
To ensure that all the collected data was accurate, all endpoints that allowed filtering by date set a maximum date of January 1st, 2023. This was done throughout all the data in this project, as to ensure the veracity of data. The date filtering option was not available for all endpoints so inaccuracies were present in the collected data. These inaccuracies were adjusted during the cleansing of the data.

Games metadata was initially retrieved using a bulk method, which provided a large amount of games with limited metadata. Subsequent requests to the games endpoint with each games' internal ID expanded upon the limited metadata. However, this endpoint did not contain information about the total number of runs, or the number of users and guests that have played a game. To collect this information, the collection system scraped a statistics webpage. For certain games this webpage was not functional, so these values were set to a special value to be filtered out or supplemented during the cleaning process. All games metadata was written to a single CSV file.

The leaderboard endpoint of the REST API can be filtered by game ID, category ID, and level ID. The collector class iterated through the leaderboards for each combination of game, category, and level and stored all unique user IDs in those leaderboards. Another collector class then wrote the personal best(s) for every user in every leaderboard of a game to a corresponding file in the `games` directory. Due to the method of collection, games with non-functional leaderboards could not be collected. The videogame Subway Surfers could not be collected at this stage as it did not have a single functional leaderboard. This is a notable example as it is reported to have the largest number of runs and users on speedrun.com [40]. Therefore, Subway Surfers is not present in further processes or analysis. This technique was used to collect information on the games that users play as there is no direct method to obtain a list of all user IDs. The individual runs of all the users, and metadata about each user were also collected using the user IDs from the previous method. Only verified runs were collected by filtering the requests by the run status as 'verified'. The individual runs of each user were saved to a corresponding file in the `users` directory. All user metadata was written into a single CSV file.

All the collected data was tested to ensure that the data was accurate. Specifically, the metadata, user runs data, and leaderboard data were tested by comparing the statistics of the collected data to the About page of speedrun.com. This page lists the amount of runs, users, and games on the

website. The user runs and leaderboard data were also tested by comparing users in the dataset and their corresponding webpage. This verified that the information was correct. These tests corroborated each other determining that the data is mostly correct prior to cleansing and transformation.



(a) Raw data.

(b) Cleaned and transformed data.

Figure 4: Layout of data.

4.2 Data Cleansing

Feature Name	Data Type	Example
game_id	String	"k6q4rqzd"
game_name	String	"Seterra"
game_created_date	Datetime	2018-09-27T08:29:37Z
game_release_date	Date	1997-01-01
game_developers	Comma-separated string	"4eppvoer,ne410dem"
game_num_categories	Integer	29
game_num_levels	Integer	903
game_num_runs	Integer	61962
game_num_users	Integer	7168
game_num_guests	Integer	36
user_id	String	"e8e52yp8"
user_signup_date	Datetime	2017-12-08T03:05:55Z
user_location	String	"br"
user_num_games	Integer	2
user_num_runs	Integer	5
user_games	Comma-separated string	"j1lq8v6g,m1zklz1"

Table 1: Features of user and game metadata.

Due to the method of collection, there were some remaining inaccuracies in the games metadata. Some features of this data were clean (`game_id` and `game_name`), as they were required to be a valid game on speedrun.com. All other features of this data set were either incomplete or inaccurate. The `release_date` and `created_date` features contained date values in different formats and had many null values. These features were converted to a single format and the null values were separated into another data source. The date features were also filtered so the games with created or release dates after January 1st, 2023 were removed. This step removed 786 games. The `num_categories` and `num_levels` features were collected during the bulk collection phase and contained no invalid values. In contrast, the `num_runs`, `num_users`, and `num_guests` fields contained purposely invalid values from the collection phase (specifically -999) to indicate that the information could not be collected. These values were either updated individually or removed depending if the correct information could be collected. This removed a further 803 games. In total 1,589 games were removed during cleaning.

The leaderboard data was cleaned by filtering the raw data using the game IDs from the cleaned games metadata step. Furthermore, games were removed if they had no runs or users. This further reduced the number of games present in the leaderboard data from 31,405 games to 30,433.

The user metadata was incredibly accurate for the number of users in the dataset, but still contained invalid values. All `user` values were accurate as they are the internal ID for each user. Some users could not be found by the REST API, so their `signup_date` was left as null. These values were removed to have a completely accurate dataset of active users and their sign-up dates. This step removed 74 users. The `signup_date` feature was then filtered so that no users past January 1st, 2023 were considered for the analysis. This step removed 5,622 users. The `num_games` and `games` fields were collected from the previously stored leaderboard data, and the location data contained no invalid values was previously cleaned when removing the null `signup_date` values. The `num_runs` feature was collected for a subset of all active users, so this was the final size of the user metadata data set. The cleaning process removed 8,121 users, resulting in complete information for 332,897 active users.

Overall, this step was incredibly straightforward, and the only issue encountered was games with non-functional leaderboard endpoints. The cleaned data was then the input to the data transformation stage. The transformation stage aims to prepare the cleaned data for analysis and machine learning applications.

4.3 Data Transformation

To fulfil the aims of the project, the data must be formatted for further analysis. The community detection processes require graphs with suitable nodes and edges. The game recommendation system requires data that is easy to compare both items and users with each other. Each piece of transformed data was written to a separate `processed` directory to identify which data is the result of analysis and not collection.

Two graphs were created from the cleaned data to analyse using both exploratory and community detection methods. First, a game-game network where nodes are game IDs, and weighted edges that represent the number of users that appear in a pair of games' leaderboards. The data source for this graph was the list of personal bests for each user in the data set. A transformation function iterated through each game in the `games` directory and tallied the number of users that appeared in a pair of games' leaderboards. This created a list of edges for the game-game network. Self-loops were removed at this stage, and pairs of games with zero shared users were not recorded. These edges were stored in a tab separated values (TSV) file, which is the default input format for graph analysis tools like Gephi. The game-game graph was tested by selecting a random sample of connected games, and determining if the edges between these games were valid in the list of all user runs.

The second graph is a user-game network with both games and users as nodes, and an edge exists if a user has played a game. The data source for this graph was the user metadata CSV which contained the game IDs of each game a user has played in a comma separated list. Each individual game ID was extracted from the comma separated list and formatted so that an individual entry was a single user ID and a single game ID. This is an edge list for a bipartite user-game graph, which was stored in a similar TSV format. These edges did not have weights, as the frequency of the number of runs a user has submitted for a single game was not collected until this analysis was completed. Similar to the game-game graph, the accuracy of the user-game graph was tested by selecting a random sample of users, and checking if the list of verified runs was the same as the neighbours of the user-game graph.

The transformed data for the recommendation system used the same data source as the bipartite

user-game network. Similarly, each game ID was extracted from the comma separated list of games. These game IDs were formatted so that a single entry was a user ID, followed by a game ID, and a binary variable representing if a user had played that game. This list of each user ID and game ID was then pivoted to create a sparse matrix M of size $n_{\text{users}} \times n_{\text{games}}$. This matrix records if a user i has played a game j with $M_{ij} = 1$, and $M_{ij} = 0$ if not. The dot product of M and M^T produces the cosine similarity matrix of M. Unfortunately, if the whole data set was included in this process, it would require over 32 GB of memory to store a single matrix. Therefore, a smaller sample of users was chosen to create the recommendation system. Two samples were taken: one containing a number of the top users ranked by the number of games played, and another of a random sample of users. Each sample was split into a training and test splits, with an 80:20 train-to-test ratio. The first sample contained 5600 users after splitting the data, resulting in 23,930 games in the similarity matrix. The second sample contained 16,000 users after splitting the data and contained only 9,650 games. For further analyses, the sample of top users was chosen to include the maximum number of games possible in the final recommendation system.

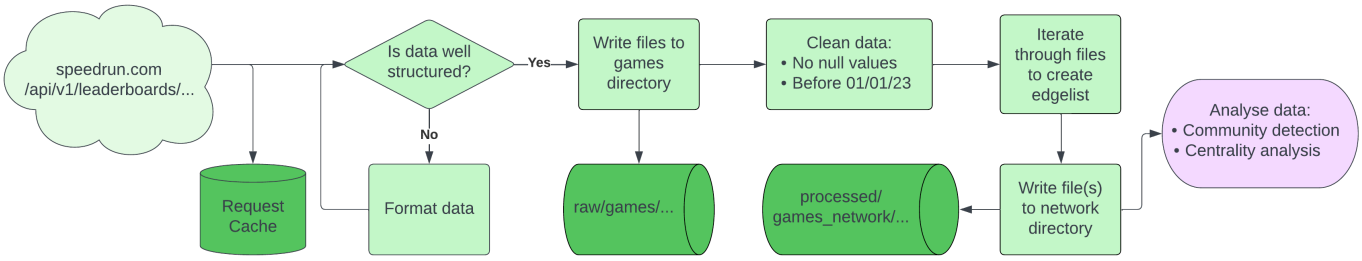


Figure 5: Flow from data source to analysis for the game-game network using leaderboard data.

4.4 Methods of Data Analysis

From prior research, the Louvain, Infomap, and Clauset-Newman-Moore community detection algorithms were chosen to run on the game-game and user-game networks. Only two of these algorithms (Louvain and Clauset-Newman-Moore) were implemented in the networkx module for Python. The implementation of Infomap was taken from the cdlib module. A secondary version of the Louvain algorithm was also necessary to run on bipartite networks. This secondary version was implemented in the sknetwork module. The Louvain algorithm has two hyperparameters available: `resolution` and `threshold`. Similarly, the Clauset-Newman-Moore algorithm accepted hyperparameters of `resolution`, `cutoff`, and `best_n`. The Infomap algorithm did not accept any hyperparameters. To evaluate the performance of each of these methods, modularity, coverage, and performance were measured.

Two different methods of recommendation systems are used to create recommendations for the games of speedrun.com: content and collaborative filtering. These methods rely on assigning a similarity value between comparable items.

The first method, collaborative filtering, uses the similarity between users to define which games to recommend. The formula for similarity between users is the Jaccard index, which measures the similarity between two sets by the ratio between their intersection and union. In this context, the two sets are the games that two users have played. The Jaccard index has been modified to use the total number of games available instead of the union of both sets of games. This was done to improve the precision of the recommendations and to not always recommend the most popular games. The Jaccard index was chosen as it directly measures the similarity between sets of games — the only data

we have for each user's preferences. Due to the nature of the collaborative filtering method it cannot be evaluated via a train/test split. The users that are part of the test split are not in the train split, and cannot have a similarity metric to the users in the training data.

The content filtering method uses similarity between games to calculate recommendations. The cosine similarity matrix was used to calculate the similarity between a sample of games in the dataset, and provides a method to lookup all similarity values for a given game. There is a single hyperparameter for both recommendation systems, which is the number of output recommendations. The number of recommendations may change depending on the context of the use of the system. The content filtering method can be evaluated with both quantitative and qualitative methods, using metrics such as precision, recall, and F1-score.

5 Results

This section starts with the results of exploratory analysis of the data and subsequently details the results of the two aims of this project: understanding the behaviour of communities and users of speedrun.com, and creating a game recommendation system for speedrunners.

5.1 Exploratory Analysis

The cleansed data from speedrun.com was analysed to describe the characteristics and behaviour of the users on the platform. The data describes 332,746 users who had submitted 3,086,954 runs for 31,416 games. On average, a user on speedrun.com has 2.74 verified runs on only one game. 246,995 of these users (or 73.66%) have only played one game, and similarly 138,300 users (41.54%) have only achieved a single verified run.

Metadata was collected for a total of 32,994 games from speedrun.com. The cleansed data set contained 31,405 games — all released before 2023 and contained no null values. The average game within this clean data has 4 categories, 7 levels, 88 runs, 19 users, and 2 guests. Ranking all games by the number of runs and number of users, the top game in both metrics is Seterra with 61,962 runs by 7,168 users. The data is very imbalanced, with 50% of all games on speedrun.com having less than 10 runs, and the top 519 (or 1.65% of) games have an equal number of runs to the bottom 30,886 (98.35%) games. The game with the largest number of guests is Minecraft: Legacy Console Edition with 15,646 guests and 155 users; a highly unusual number of guests considering most games have higher numbers of users compared to guests.

The top three users have played 2059, 1884, and 916 different games respectively. This an unusually high number of unique games played with z-scores of 277.6, 254, and 123.3 respectively. Furthermore, ranking all users by the number of verified runs shows that the new top three users have 9800, 8822, and 7598 runs. These new top users do not necessarily have an abnormally high number of games played, as a few of these top users have a z-score of the number of games played < 1.

The most popular game for users that have played a single game is Seterra, an online map quiz game. The next largest games are Minecraft: Java Edition, ROBLOX: Speed Run 4, Hypixel BedWars, Celeste, ROBLOX: Doors, and Google Snake. Analysing the users that have only played two games, the pair of games that a user chooses to play seem to be related. Users play a combination of: a game and it's category extensions, a prequel/sequel to the first game, or a mini-game within the first game. For example, users that play Minecraft: Java Edition are likely to play games such as Minecraft: Java Edition Category Extensions (category extensions), Minecraft: Bedrock Edition (sequel), or any Hypixel game (mini-games).

The first user created their account on January 6th, 2014, and half of all users had created their

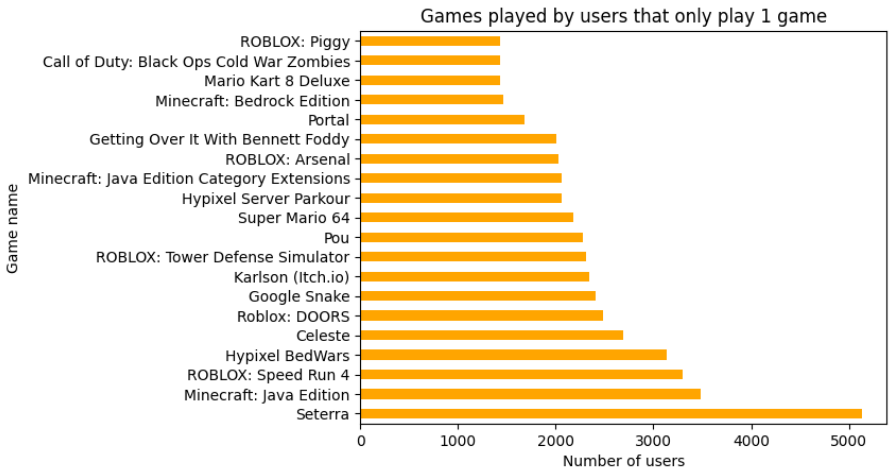
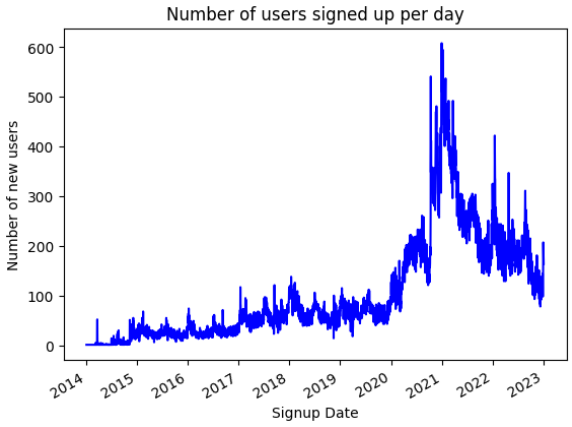
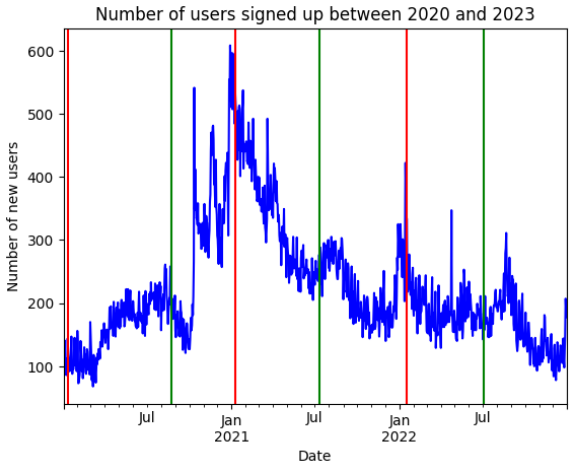


Figure 6: Games played by users on speedrun.com.



(a) The number of new active users per day between 2014 and 2023.



(b) The number of new active users per day between 2020 and 2023. Dates of GDQ events highlighted in red and green.

Figure 7: Rate of new users joining speedrun.com and demographics of all users.

accounts before January 4th, 2021. It took only two years to double the number of active users on speedrun.com. There is a massive increase in sign-ups starting around January 2020 and ending around January 2021. The average number of sign-ups per day is higher after the peak (232.2) than before (43.8). There are also short-term rises and falls in the number of sign-ups per year, with at least two specific increases in sign-ups.

Focusing on the demographic metadata of the users, 101,439 users or (29.96%) have submitted their location as the United States, making it the largest non-null demographic. The next largest demographic is 'No Country Specified' with 58,107 (17.16%) users, followed by the United Kingdom with 20,030 users (5.92%). The subsequent demographics such as Canada, France, Germany, and Australia slowly decrease in the number of users.

5.2 Network Analysis

From the data collected two networks were created: a weighted and directed game-to-game network, and a directed bipartite user-to-game network. The game-game network has games as nodes, and weighted edges that are determined by the number of users that appear in a pair of

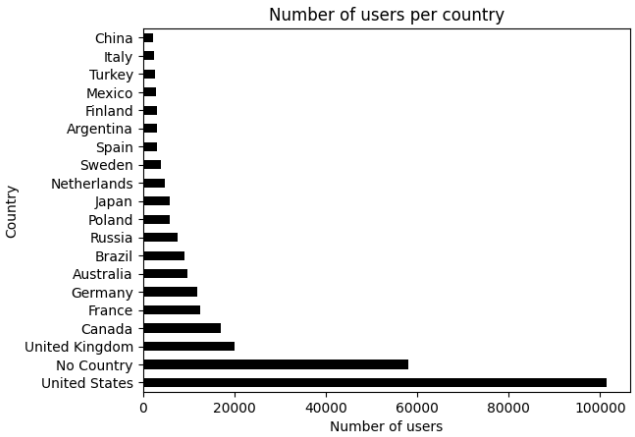
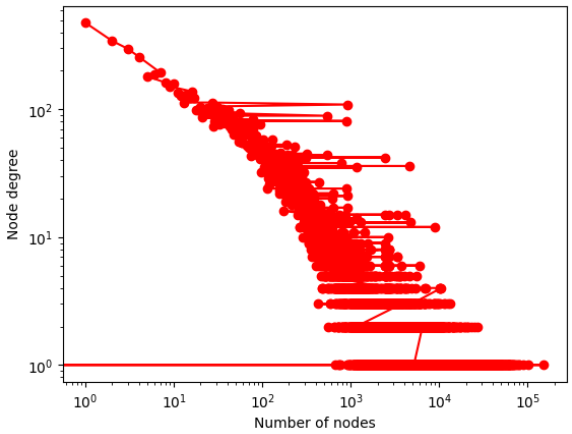


Figure 8: The demographics of all active users of speedrun.com.

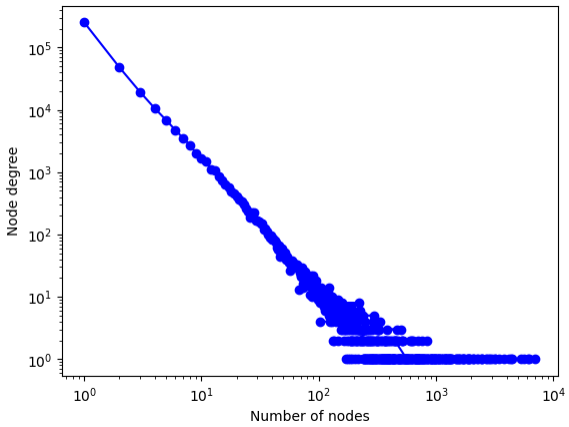
Network Name	Number of Nodes	Number of Edges	Density	Clustering Coefficient
Game-Game	30,433	14,739,311	1.59×10^{-2}	0.656
User-Game	366,747	668,788	9.94×10^{-6}	0.496

Table 2: Comparison of the Game-Game and User-Game networks.

games' leaderboards. This creates a network with 30,433 nodes and 14,739,311 edges. This network has a density of 0.0159, and a clustering coefficient of 0.657. There is a high degree distribution in the game-game network with a few very dominant nodes.



(a) Log-log scale degree distribution for game-game network.



(b) Log-log scale degree distribution for user-game network.

Figure 9: Node degree distribution for the game-game and user-game networks.

Four different variations of centrality were measured for the game-game network: degree, PageRank, hubs and authorities (HITS), and betweenness. Analysing the degree centrality of each of the nodes reveals Seterra is the most popular game using both in-degree and out-degree. This is followed by Celeste, Super Mario 64, Minecraft (Classic), Super Mario Bros., Mario Kart 8 Deluxe, Bee Movie Game (DS), New Super Mario Bros., Super Mario World, and Pac-Man. The game with the highest betweenness centrality is Celeste. The developer with the highest average betweenness centrality is Matt Makes Games Inc., whom are famous for developing Celeste. Furthermore the developers for Seterra, Outlast, Super Smash Bros. Ultimate, and Pringles have the highest average betweenness

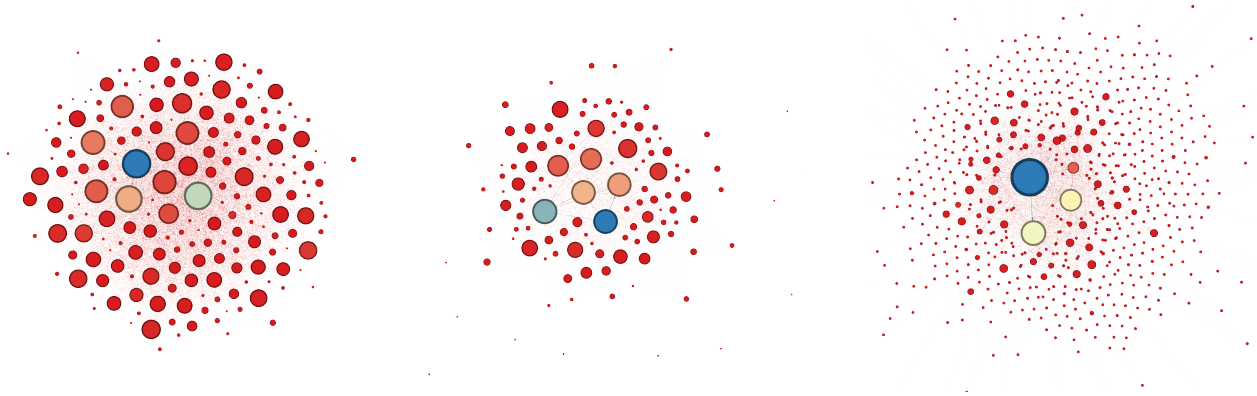
centrality. Other types of centrality produce similar results to the degree centrality.

The user-game graph provides a secondary view on the games and users of speedrun.com. A directed bipartite graph was created with games and users as nodes, and edges determined by if a user has played a given game. This network has a total of 366,747 nodes and 668,788 edges. This bipartite graph has a density of 9.94×10^{-6} and an average clustering coefficient of 0.496. There is a similar high degree distribution to the game-game network, but the user-game network has a straighter line in the log-log graph.

5.3 Community Detection and Evaluation

Network Name	Algorithm Name	Modularity	Performance	Coverage
Game-Game	Louvain	0.400	0.697	0.470
	Infomap	0.312	0.947	0.703
	CNM	0.275	0.360	0.845
User-Game	Louvain	0.716	0.965	0.772
	Infomap	0.682	0.868	0.812
	CNM	0.682	0.911	0.827

Table 3: Comparison of modularity, performance, and coverage on Game-Game and User-Game networks.



(a) Louvain algorithm (b) Infomap algorithm (c) CNM algorithm

Figure 10: Meta-networks of the Louvain, Infomap, and CNM algorithms on the user-game network.

The Louvain, Clauset Newman Moore (CNM), and Infomap community detection algorithms provided mixed results on the game-game network. With an average modularity of 0.329. Louvain produces the highest modularity score of 0.4, whereas Clauset-Newman-Moore and Infomap algorithms produce a modularity score of 0.275 and 0.312 respectively. The Clauset-Newman-Moore algorithm produced the highest coverage of 0.845 and the lowest performance of 0.360. Infomap produces by far the best performance of 0.947 and an intermediate value of coverage and modularity when compared to the other algorithms.

Investigating the communities produced by the Louvain algorithms shows that the contents are somewhat dissimilar. Community number 23 has a number of games z-score of 16.5 with the next largest community having a z-score of 1.84. Community 23 contains over half of all games on speedrun.com. The games within this community are not very similar, ranging over many game series, platforms, decades, and genres. Community 8 of the game-game network could be categorised as video games that are available on web/mobile platforms. However, this not consistent upon including less popular games. This is a trend for most of the communities, especially for those with games that

are massively popular. Some communities differ from the majority, containing a focused set of games. For example, community 27 contains 168 games with most related to the LEGO franchise or science fiction franchises such as Spider-Man, Batman, The Matrix, or Ben-10. Overall most communities in the game-game network found via the Louvain algorithm contain weakly linked video games.

Community detection algorithms on the bipartite user-game network produced significantly better performance in all metrics. The Louvain algorithm produced 1,234 communities with a modularity of 0.716. The number of communities was reduced by removing all but the largest connected component. The largest connected component contained 364,123 out of 367,747 nodes, or 99% of the total. Other connected components did not contain many users, with an average 2.53 nodes per connected component. By removing 2,624 nodes from the user-game network, the number of communities reduced from 1,234 to 198, while the modularity score remained at 0.716. Communities in the user-game network have on average 1,685 users and 153 games. The largest community (number zero) has 30,327 users and 2,362 games. Similarly, community 2 contains the highest number of games with 7,388 games and 20,530 users. Although the majority of communities have a consistent ratio of users to games, some communities have a low ratio considering the number of users within that community.

The largest community in the user-game network (community one) appears to be organised around games that were originally released on platforms like the Nintendo Wii, Nintendo Switch, and Nintendo 64. Community zero follows this trend as most games were released for the Nintendo Entertainment System and Super Nintendo Entertainment System. Community two is another example of games categorised by their platform, with most video games exclusively available on the web or mobile. Several communities appear to be each related to a single entity or game franchise. For example, community four almost entirely contains games from the ROBLOX franchise, and communities six, eight, and ten containing games related to Minecraft. Some communities appear to be characterised by the mechanics of the games therein, for example community seven containing games like Portal, Refunct, Mirror's Edge, and Half-Life. Open-world video games such as Grand Theft Auto V, Need for Speed: Carbon, and Just Cause 3 are categorised together. Similarly, horror games like Five Nights at Freddy's, Hello Neighbour, and Poppy Playtime are grouped together in community 13. Some communities have a small number of games with a single incredibly popular game. Community 26 contains only 17 games and 3,495 users with the most popular game being Pou with 2,384 players compared to 889 players for Temple Run 2. The games with the highest betweenness centrality of a community are usually the most popular games in that community.

Most communities have similar demographics to the larger network. There are only 29 communities where the top demographic is not the United States or No Country Specified. All of these communities have a negative z-score of number of users. In most communities the demographic that has the largest difference between their representation in the overall game-user network and their representation in a community is usually the most popular demographic.

5.4 Game Recommendation System

The content filtering recommendation method produced results with a maximum F1-score of 0.557 at 20 recommendations. Recall increases linearly with the number of recommendations, and the best combination of precision and recall occurs at 5 recommendations. Requesting game recommendations using a game in a well-known game series returns other entries of the original game series. The recommended games also usually contains games with similar game mechanics and genres. For example, requesting recommendations for Call of Duty: Black Ops III Zombies produces

recommendations for games within the Call of Duty franchise (Call of Duty: Black Ops II Zombies), games within the genre of Zombies (Resident Evil 2, Outlast), and overall popular games (Grand Theft Auto V).

The collaborative filtering method returns similar recommendations to the content filtering method. There is empirically higher diversity in the recommendations from the collaborative method compared to the content filtering method. For example, for a user that has played The Legend of Zelda: A Link to the Past, the collaborative filtering method recommends Nintendo games from most decades (such as Super Mario 64 for the N64, Mario Kart 8 Deluxe for the Nintendo Switch), and popular games (like Celeste). When inputting the same game to the content filtering method, the recommendations are focused on Nintendo games from the same era (such as Super Mario Bros. and The Legend of Zelda.

6 Discussion

The analysis answers all the research questions stated earlier. Several different communities were found using the Louvain, CNM, and Infomap algorithms, with the Louvain algorithm performing the best on the user-game network with a modularity of 0.716. Most communities found using this method are extremely focused and most games within each share a distinct commonality. Two methods of recommendation systems were implemented, and both produced quantitatively and qualitatively good results. The collaborative filtering method produced only qualitative results, while the content filtering method produced a F1-score of 0.55 at 5 recommendations. The analysis of users of speedrun.com provided an insight into user behaviour and demographics, and the popularity of speedrun.com over time. The games of speedrun.com have a heavy imbalance with a few games being massively popular in most metrics. Finally, the collection process produced a lot of raw data from speedrun.com, and is available for distribution on GitHub.com.

6.1 Exploratory Analysis

Focusing on the exploratory analysis, it is not surprising that there is a heavy imbalance in the data with respects with to the number of users and number of runs for each game. There are approximately 32,000 games on speedrun.com, and most of them are extremely unpopular. Only a few games share most of the users, which can be defined as the most 'speedrunnable' games. These popular games are expected as they are either incredibly influential games such as Super Mario 64, or games with low barriers to entry and short speedrun times such as Seterra. These popular games remain similarly popular within the users that have only played a single game, indicating that most users will initially choose either a short or famous game to speedrun.

The number of runs and games played per user is not surprising. It was expected that most users have only played one game, but it is surprising that these users have more than one verified run. The expected user behaviour is that a user that tries speedrunning will only do so once, and then stop if they find it is too difficult or they are satisfied with their time. This is not the case, as users seem to submit a few times before achieving a desired time. The top users ranked by unique games played shows us that there are a few users that aim to play as many games as possible. There was an explicit competition between users named "Gotta Run 'Em All", with the explicit objective of playing as many games as possible. It is expected that the top users ranked by number of runs will be incredibly dedicated to a subset of games, submitting hundreds or thousands of runs for a single game. This is repeated for possibly all the games they play, as they have on average a low z-score of unique games played.

The popularity of the speedrun.com platform is completely unpredictable. It does have a slow increase in the number of new users joining the platform, but the increase from January 2020 to January 2021 is much higher than expected. The short-term trends within each year is correlated with the dates of Games Done Quick (GDQ) events, and the rapid increase in popularity in 2020 is most likely due to the COVID-19 pandemic. A study focusing on the amount of time spent playing video games both pre and post-pandemic shows that most respondents played far more video games post-pandemic [2]. This increase in the frequency of playing videogames would inevitably lead some to explore other ways of playing. The sharpest increase of new users occurs in October 2020. Unfortunately, no single event could be found to explain this increase.

Speedrunning is mainly a phenomenon in the United States. No other demographic (apart from users that failed to submit their location) approaches the United States in size. It contains approximately 30% of all users, and possibly more considering a portion of the 'None' demographic are most likely from the US, but did not fill in their location. The other top demographics share many similarities, so finding a single contributing factor is incredibly difficult.

6.2 Community Detection

Although similar methods were applied to the user-game and the game-game graph, the results are incredibly different. The Louvain algorithm on the user-game graph had a 78.75% increase in modularity over the game-game graph. This suggests that the communities on the user-game graph are well-defined compare to the game-game graph. This is true empirically, as the communities in the user-game graph tend to be more focused and meaningful.

Communities zero and one both contain mostly Nintendo games, but are separated by the original platform of release. This separation suggests that users choose games based on the platform of release, or how old a videogame is. Community two contains games mostly for web or mobile, but investigating closely, it seems that these games have extremely low barriers for entry. This could mean that games are free-to-play, or only requiring a web-browser to play. Some communities are characterised by genre of games, such as community five containing platformers such as Celeste, community seven containing first-person perspective games like Half-Life, and community zero containing open-world games such as Grand Theft Auto 5. This suggests that users pick games based on the genre or underlying mechanics of the games. We also see communites based on a single game franchise such as communities six and 18. These communities are focused on Minecraft or Sonic games respectively, indicating that users will explore the same game franchises and the different speedruns they offer.

The demographics within the communities does not change often. It was expected that some communities could contain users sharing a common language, or playing games from a certain region. Some communities do have these properties, but they are not large enough to characterise user behaviour as a collective.

The centrality analysis of the user-game and game-game graphs give some unexpected results. The degree centrality of games provides no interesting results, with the most popular games having the highest in-degree and out-degree. The Hubs and Authorities (HITS) centrality presents Authority nodes that tend to be those with high degree centrality, and Hub nodes that tend to be web or mobile games. PageRank offers the same results as the degree centrality. The betweenness centrality offers the most interesting results, with a higher betweenness centrality acting as bridges between communities. The most popular games within each community are usually those with the highest betweenness centrality. For example, community zero's highest betweenness centrality is Super

Mario 64. Similarly, The Legend of Zelda: A Link to the Past has the highest betweenness centrality in community one. This trend is repeated through most of the communities, but notably missing in community two, which contains mostly web or mobile games. The highest betweenness centrality belongs to the Bee Movie Game (DS), which is unrelated to the other games in its community. This cannot be explained currently. These highest betweenness centrality games suggests that users explore new kinds of game or communities by playing the most popular game of that kind. The users then either explore this new community or return to their original preferences.

6.3 Game Recommendation System

Both methods of game recommendation systems produced anecdotally good recommendations. The recommendations were unexpectedly accurate, and were good at recommending games from the same game franchises. This suggests that users tend to play games within game franchises, and not random samples of the available games. Likewise, games with similar in-game mechanics or genre are also recommended, so users are likely to play these games together as well.

The collaborative filtering method produced empirically more diverse recommendations. This may be advantageous when considering the cold-start problem of recommendation systems. The collaborative filtering method also takes users as input, rather than an individual game. This may be preferred when implementing and deploying a real-world game recommendation system, using only one request compared to several for each game a user plays, and ranking after considering their similarities.

6.4 Implications

Recalling the research questions stated earlier, it has been proven that there are communities of users of speedrun.com whose videogame preferences align with each other. Furthermore, these similarity of these users are reinforced with the accuracy of a game recommendation system with a reasonable accuracy and diversity of recommendations. This project provides concrete communities of users, and theories of the commonality between the users and games in these communities. Likewise, it analyses the users and games of speedrun.com and describes why they choose the games they play.

7 Conclusion

This project uses machine learning and data science techniques to understand the user behaviour of speedrunners. This network of speedrunners show a heavy imbalance between the games played, with the top 1.65% of games being as popular as the bottom 98.35%. The community detection methods found strong evidence of the existence of communities of users of speedrun.com whose video game preference align with each other. These communities shared preferences for specific game genres, franchises, mechanics, or platform. These communities are typically accessed by their most popular game. A couple of implementations of a game recommendation system found patterns of behaviour over most users. Most users tend to play games that are within a single game franchise, share similar mechanics, on a single platform, or are incredibly popular.

The community detection results imply there are types of user that are defined by the games they play. These results could be utilised to enable data-driven decisions, particularly by video game developers or marketers. The game recommendation system results indicate that the developers of speedrun.com could implement a similar system using the data they have, and that it would be reasonably successful within the community. In this context, a game recommendation could improve

the user engagement on the platform, and help developers anticipate the demands of the consumers.

This project contributes to the literature on the applications of community detection and recommendation systems, and under-investigated topics such as videogames. This study has labelled communities of users on speedrun.com, and created a game recommendation engine tailored towards speedrunners. A dataset of the games and users of speedrun.com was also curated and published on GitHub.com [9]. This research also addresses the gap in research surrounding speedrunning, and hopefully prompts further investigation into this topic.

7.1 Critical Reflection

There are several limitations of the the research within this project. First, there was limited access to the data source of speedrun.com. Only users that have contributed to the leaderboards were included in the research. The speedrun.com REST API does not have functionality for listing all the users on their platform, so only a subset of all their users could be processed. The remaining users could give a broader picture of the speedrunning community as they may contribute through other means and not the leaderboards.

Second, other speedrunning platforms could have been considered if more time was available to complete the research. Other online communities of speedrunners such as TwinGalaxies and SpeedDemosArchive contain important information about how the community interacts; more speedrunning communities could ratify the results found for the research on speedrun.com.

Another constraint of this research is the amount of metadata available for each user on speedrun.com. A limited amount of metadata was collected for each user on the speedrun.com platform as all information from the data source had to be non-identifiable. User metadata beyond what was supplied could help both the community detection and game recommendation methods. The community detection could use extra user metadata to find more nuances between the communities, and perhaps find patterns between personal characteristics and the games played by a community. Recommendation engines could use concepts such as interests and personal characteristics to provide recommendations to groups of users that share these traits.

Other methods of community detection and recommendation systems could have been explored in this research. Time constraints and lack of experience in the relevant fields of research contributed to a lack of diversity of methods used to fulfil the aims of the project. Other methods of community detection could have found communities with higher modularity, and corroborated the communities found within this project. Likewise, other types of recommendation systems could improve on the methods used with respect to the scalability and variety of data used to produce recommendations.

7.2 Further Research

The limitations of this project provide several avenues for further research: investigating other online speedrunning communities, improving the methods of community detection and recommendation systems, or using other methods of analysis on the speedrun.com dataset. In particular, a rigorous testing of the produced recommendation system via click-through rate and conversion rate may be implemented by speedrun.com themselves. This project also raised some further questions that could be investigated further. For instance, how have the communities evolved through time? Every run in the dataset has a timestamp related to it — these could be used to track the evolution of the communities found in this project. Also, this study does not ask: What effect do influencers have on the speedrunning communities? This project has created communities of both users and games, but has not investigated the effect these users have on each other.

References

[1] Speed Demos Archive. *SpeedDemosArchive: Playing games quickly, skillfully and legitimately*. URL: https://speeddemosarchive.com (visited on 04/17/2023).

[2] Matthew Barr and Alicia Copeland-Stewart. "Playing Video Games During the COVID-19 Pandemic and Effects on Players' Well-Being". In: *Games and Culture* 17.1 (2022), pp. 122–139. DOI: 10.1177/15554120211017036. eprint: https://doi.org/10.1177/15554120211017036. URL: https://doi.org/10.1177/15554120211017036.

[3] Punam Bedi and Chhavi Sharma. "Community detection in social networks: Community detection in social networks". en. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 6.3 (May 2016), pp. 115–135. ISSN: 19424787. DOI: 10.1002/widm.1178.

[4] Vincent D Blondel et al. "Fast unfolding of communities in large networks". en. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008. ISSN: 1742-5468. DOI: 10.1088/1742-5468/2008/10/P10008.

[5] Thomas Bonald et al. "Scikit-network: Graph Analysis in Python". In: *Journal of Machine Learning Research* 21.185 (2020), pp. 1–6. URL: http://jmlr.org/papers/v21/20-412.html.

[6] Stephen P. Borgatti. "Centrality and network flow". In: *Social Networks* 27.1 (2005), pp. 55–71. ISSN: 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2004.11.008.

[7] Zhengdao Chen, Xiang Li, and Joan Bruna. *Supervised Community Detection with Line Graph Neural Networks*. 2020. arXiv: 1705.08415 [stat.ML].

[8] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. "Finding community structure in very large networks". en. In: *Physical Review E* 70.6 (Dec. 2004), p. 066111. ISSN: 1539-3755, 1550-2376. DOI: 10.1103/PhysRevE.70.066111.

[9] *Collection and analysis of data from Speedrun.com*. URL: https://github.com/alexmerren/speedruncom-data (visited on 05/01/2023).

[10] Yimin Cui. "Intelligent Recommendation System Based on Mathematical Modeling in Personalized Data Mining". In: *Mathematical Problems in Engineering* 2021 (Feb. 2021). Ed. by Sang-Bing Tsai, p. 6672036. ISSN: 1024-123X. DOI: 10.1155/2021/6672036.

[11] Magdalini Eirinaki et al. "Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions". en. In: *Future Generation Computer Systems* 78 (Jan. 2018), pp. 413–418. ISSN: 0167739X. DOI: 10.1016/j.future.2017.09.015.

[12] Juan Escobar-Lamanna. "Why Speed Matters: Collective Action and Participation in Speedrunning Groups". MA thesis. OCAD University, Apr. 2019. URL: http://openresearch.ocadu.ca/id/eprint/2474/.

[13] Christopher Flynn. *Python Package Index Statistics*. Version latest. URL: https://pypistats.org/about (visited on 12/25/2022).

[14] Santo Fortunato. "Community detection in graphs". In: *Physics Reports* 486.3-5 (Feb. 2010), pp. 75–174. DOI: 10.1016/j.physrep.2009.11.002. URL: https://doi.org/10.1016%5C%2Fj.physrep.2009.11.002.

[15] Twin Galaxies. *TwinGalaxies Speedrun Submissions*. URL: https://www.twingalaxies.com (visited on 04/17/2023).

[16] Guibing Guo, Jie Zhang, and Daniel Thalmann. "Merging trust in collaborative filtering to alleviate data sparsity and cold start". en. In: *Knowledge-Based Systems* 57 (Feb. 2014), pp. 57–68. ISSN: 09507051. DOI: 10.1016/j.knosys.2013.12.007.

[17] Jyoti Gupta and Jayant Gadge. "Performance analysis of recommendation system based on collaborative filtering and demographics". In: *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*. IEEE. 2015, pp. 1–6.

[18] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15. URL: http://conference.scipy.org/proceedings/SciPy2008/paper_2/.

[19] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[20] Michael Hemmingsen. "Code is Law: Subversion and Collective Knowledge in the Ethos of Video Game Speedrunning". In: *Sport, Ethics and Philosophy* 15.3 (2020), pp. 435–460. DOI: 10.1080/17511321.2020.1796773.

[21] Jonathan L Herlocker et al. "Evaluating collaborative filtering recommender systems". In: *ACM Transactions on Information Systems (TOIS)* 22.1 (2004), pp. 5–53.

[22] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95.

[23] Elo Entertainment Inc. *About Speedrunning and Statistics*. Accessed on: Dec. 25 2022. URL: https://web.archive.org/web/20221128161434/https://www.speedrun.com/knowledgebase/about (visited on 12/25/2022).

[24] Przemysław Kazienko, Katarzyna Musial, and Tomasz Kajdanowicz. "Multidimensional Social Network in the Social Recommender System". en. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 41.4 (July 2011), pp. 746–759. ISSN: 1083-4427, 1558-2426. DOI: 10.1109/TSMCA.2011.2132707.

[25] Andrea Lancichinetti and Santo Fortunato. "Community detection algorithms: A comparative analysis". In: *Phys. Rev. E* 80 (5 Nov. 2009), p. 056117. DOI: 10.1103/PhysRevE.80.056117. URL: https://link.aps.org/doi/10.1103/PhysRevE.80.056117.

[26] Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a Feather: Homophily in Social Networks". In: *Annual Review of Sociology* 27.1 (2001), pp. 415–444. DOI: 10.1146/annurev.soc.27.1.415. eprint: https://doi.org/10.1146/annurev.soc.27.1.415. URL: https://doi.org/10.1146/annurev.soc.27.1.415.

[27] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks". In: *Phys. Rev. E* 69 (2 Feb. 2004), p. 026113. DOI: 10.1103/PhysRevE.69.026113. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

[28] Koray Öztürk. "Community detection in social networks". MA thesis. Middle East Technical University, 2014.

[29] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[30] Tiago P. Peixoto. "The graph-tool python library". In: *figshare* (2014). DOI: 10.6084/m9.figshare.1164194. URL: http://figshare.com/articles/graph_tool/1164194 (visited on 09/10/2014).

[31] Nicola Perra and Santo Fortunato. "Spectral centrality measures in complex networks". In: *Phys. Rev. E* 78 (3 Sept. 2008), p. 036107. DOI: 10.1103/PhysRevE.78.036107. URL: https://link.aps.org/doi/10.1103/PhysRevE.78.036107.

[32] *PwC Global Entertainment and Media Outlook 2022-2026*. URL: https://www.pwc.com/gx/en/news-room/press-releases/2022/global-entertainment-and-media-outlook-2022-2026.html (visited on 04/24/2023).

[33] *requests-cache: Persistent HTTP cache for python requests*. URL: https://github.com/requests-cache/requests-cache (visited on 04/17/2023).

[34] Giulio Rossetti, Letizia Milli, and Rémy Cazabet. "CDLIB: a python library to extract, compare and evaluate communities from complex networks". In: *Applied Network Science* 4.1 (July 2019), p. 52. ISSN: 2364-8228. DOI: 10.1007/s41109-019-0165-9.

[35] M. Rosvall, D. Axelsson, and C. T. Bergstrom. "The map equation". en. In: *The European Physical Journal Special Topics* 178.1 (Nov. 2009), pp. 13–23. ISSN: 1951-6355, 1951-6401. DOI: 10.1140/epjst/e2010-01179-1.

[36] Ivan Dmitriy Ortiz Sánchez. "Mode and tempo of cultural evolution in video games". Unpublished. Bachelor's Thesis. Universitat Pompeu Fabra, 2021.

[37] Rainforest Scully-Blaker. "A practiced practice: Speedrunning through space with de certeau and virilio". In: *Game Studies* 14.1 (2014).

[38] Rainforest Scully-Blaker. "Re-curating the Accident: Speedrunning as Community and Practice". Unpublished. MA thesis. Concordia University, Sept. 2016. URL: https://spectrum.library.concordia.ca/id/eprint/982159/.

[39] *Srcomapi: A Python implementation of the speedrun.com REST API*. URL: https://github.com/blha303/srcomapi (visited on 04/17/2023).

[40] *Subway Surfers - speedrun.com*. URL: https://www.speedrun.com/subsurf (visited on 04/18/2023).

[41] Lei Tang and Huan Liu. "Community Detection and Mining in Social Media". In: *Community Detection and Mining in Social Media*. 2010.

[42] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

Appendix

Appendix 1: Description of the 40 largest communities found on the user-game network via the Louvain method

Community Number	Number of Users	Number of Games	Description of Games in Community
0	30327	2362	The most popular games in this community are Nintendo games from the Nintendo 64 (N64), Nintendo Wii, and Nintendo Switch. Another focus is on games that are released from the release of the N64 in 1996.
1	24432	4796	This community contains games that are available on the web/-mobile. It also contains many ROBLOX games, but they are not frequent enough to define this community by this game series.
2	20530	7388	The most frequent games in this community are Nintendo games from Nintendo Entertainment System (NES), Super NES (SNES), and GameBoy Advance (GBA), and fan-made games for the N64. This community is related to community zero, but the difference is the age of the games.
3	14490	1562	There are many incredibly popular games in this community, so it is hard to pin the exact similarity between the items. It is possible that these games are similar in their in-game mechanics of fighting/shooting. However, some games are in the genre of episodic graphic games that lack fighting/shooting mechanics, so the similarity is relatively weak.
4	15247	134	Since there are few games in this community, they are very similar. These games are overwhelmingly from the ROBLOX franchise. Other games are only available on the web.
5	12509	738	The games in this community are all developed and published by independent developers.
6	10721	237	This community has mostly games related to the Minecraft franchise, with some outliers such as the Elder Scrolls Online. There is no discernable similarity between *all* games.
7	9920	785	The games in this community are all traditionally played on PC. This is to gain a greater control over the in-game mechanics, commonly in games with first-person perspectives.
8	10003	89	The most popular games are related to the Hypixel Minecraft server, or games that are exclusively available for PC.

Continued on the next page

Community Number	Number of Users	Number of Games	Description of Games in Community
9	8348	1093	All games in this community are related to the open-world, racing, or sport genres. The most popular games are in the Grand Theft Auto franchise, and also contains the F1, FIFA, and NBA franchises.
10	7731	147	The overwhelming majority of games in this community are from the mobile version of Minecraft (Minecraft: Bedrock Edition), and other mini-games within this game.
11	7052	363	The popular games in this community are from the Call of Duty franchise. This genre could be classified as first/third-person action games.
12	6128	1103	These games were originally released on either the PlayStation one, two, or three; other games were released for the GBA, or GameBoy Color (GBC).
13	6589	228	Almost all games in this community are categorised in the horror genre. There are a few outliers but they have very few players compared to the top games.
14	6002	115	The games in this community are all published independently, and specifically are available on the website `itch.io`.
15	5259	804	Most games in this community are fantasy role-playing games, such as Xenoblade Chronicles, Final Fantasy, and Warhammer.
16	5687	28	This community is constructed primarily of users that play Seterra, with a few other web-exclusive games.
17	4461	23	These games are mostly part of the ROBLOX franchise, with a few extremely unpopular games that do not match this criteria.
18	4156	304	The most popular games in this community are from the Sonic franchise. Including all games show that it is apt to characterise these games as racing/platformer games, which are Sonic-like.
19	4409	49	Similar to community 17, most games are part of the ROBLOX franchise. It is not clear why these communities are separate.
20	4359	79	There is a single incredibly popular game which is Google Snake. Other games are usually web-exclusive. This community is similar in structure to community 16.
21	3959	224	The popular games in this community are web-only games from the 1990's to the 2010's, with the less popular games being a mix of PC, GBA, and PS1 games. There is no discernable similarity between the less popular games.
22	4141	28	Top games are massively-multiplayer shooter games, with less popular games being web-only.
23	3935	74	This community is primarily characterised by games such as Getting Over It With Bennett Foddy and Jump King. These games rely on realistic physics for their in-game mechanics.
24	3324	434	This community is comprised of games related to animated films or TV series such as SpongeBob SquarePants or Madagascar. Generally, this community are characterised as cartoon videogames.

Continued on the next page

Community Number	Number of Users	Number of Games	Description of Games in Community .
25	3439	141	This community is characterised by competitive multiplayer games, with some having large Esports communities such as Overwatch and Valorant.
26	3495	17	These games are identified by their casual style of gameplay, and being available on mobile platforms. This community is similar to number 16 and 20 where the most popular game, Pou, is overwhelmingly popular.
27	2983	511	This community is mainly games from the Pokémon franchise, or other popular franchises such as Mega Man. Less popular games could be characterised as visual novel adventure games, such as Phoenix Wright: Ace Attorney, and the Nancy Drew series.
28	3304	57	Similar to community 26, these games have casual style of gamepaly, and could be defined as single-player puzzle games.
29	3185	88	Most popular games in this community are mobile-only games.
30	2829	294	The is no obvious connection between the games in this community, but the top four games of Deltarune, Undertale, Helltaker, and Genshin Impact are incredibly more popular than the others.
31	2960	82	The games in this community are singleplayer comedy point-and-click games. The top game, There Is No Game, is undoubtedly the most popular.
32	2578	238	This community contains games predominantly from the LEGO franchise, with other science fiction franchises such as Batman, Spider-Man, and Transformers. Overall, this community could be characterised by LEGO games and games from comic franchises.
33	2180	565	The games within this community could be defined as difficult platformer games.
34	2543	59	This community contains various games that could be characterised as simulation and action games.
35	2480	110	Most of these games are from the Hitman franchise. This community could be characterised as action and simulation games.
36	2314	242	This community could be defined as role-playing action games, including games such as Fallout 4, and Borderlands 3. However, there are many point-and-click that do not fit this charaterisation.
37	2355	81	Similar to community 36, these games are characterised by being fast-paced first-person shooter games.
38	2343	92	The top game in this community, PAYDAY 2, is overhwlmingly popular. The games in this community are categorised by their focus on violence.
39	2296	57	These games are usually from the 1990's, such as Doom or Quake, and may have a cult-like following from users that enjoy playing older games.
40	2152	227	This community is particularly unfocused, containing games from the horror genre, Minecraft spin-off games, and old PC games.