## Research Questions and Motivation

Steam, the world's largest digital distribution platform for PC gaming, faces unprecedented scaling challenges as it serves millions of concurrent users across its digital marketplace and social network. Developed by Valve Corporation, Steam has evolved beyond game distribution into a complex ecosystem supporting cloud gaming, virtual reality, and social networking. This exponential growth in both content and users presents unique challenges in data management and user experience optimization.

### Research Questions

- *How can network analysis of user interaction patterns be leveraged to create accurate game recommendations while minimizing latency and computational overhead?*
- *What network architectures and data distribution models would optimize the performance of a large-scale recommendation system serving millions of concurrent Steam users?*
- *How can peer behavior analysis be implemented efficiently across Steam's distributed network to generate personalized recommendations?*

### Motivation

Steam's rapid scaling presents two intersecting challenges: **managing computational resources across a distributed network while delivering increasingly personalized user experiences**. This research addresses the critical balance between recommendation quality and system performance, with implications for large-scale digital platforms facing similar scaling challenges. The findings will contribute to developing network-efficient recommendation systems that maintain personalization quality while optimizing resource utilization.

## Related Work

Previous research in gaming recommendation systems spans from fundamental algorithms to platform-specific implementations, focusing on scalability and user behavior.

**Player Behavior Analysis and Profiling**: *Sifa et al.*'s work on game telemetry and player behavior (*2018*) established fundamental approaches to understanding player patterns through machine learning techniques. Their research demonstrated how behavioral profiling could be used to improve game recommendations, particularly when dealing with large-scale telemetry data from multiple games.

**Cross-Game Player Analysis**: Building on behavioral analysis, *Sifa*, *Drachen*, and *Bauckhage* (*2021*) conducted a comprehensive study of Steam platform data across over 3,000 games and 6 million players. Their analysis of playtime patterns and game ownership revealed crucial insights into how players distribute their time across multiple games, providing valuable foundations for cross-game recommendation systems.

**Large-Scale Recommendation Systems**: *Gomez-Uribe* and *Hunt* (*2015*) detailed Netflix's recommendation system architecture and methodologies, offering valuable insights for gaming platforms. Their work demonstrated how large-scale recommendation systems can balance algorithmic sophistication with system performance, particularly relevant for platforms handling millions of concurrent users.

**Resource Management**: *Delimitrou* and *Kozyrakis* (*2014*) addressed critical infrastructure challenges through their Quasar system, demonstrating methods for improving resource utilization while maintaining service quality. Their findings on resource allocation and workload management are relevant for scaling recommendation systems across distributed networks.

**Social Network Influence**: *Bakshy et al.* (*2012*) examined how social networks impact information diffusion through a large-scale experiment with 253 million subjects. Their findings on the role of weak versus strong ties in information propagation provide valuable insights for understanding how social connections influence game discovery and recommendations on gaming platforms.

## Data

We are primarily using the *Game Recommendations on Steam* dataset from Kaggle. It includes the following three dataframes.

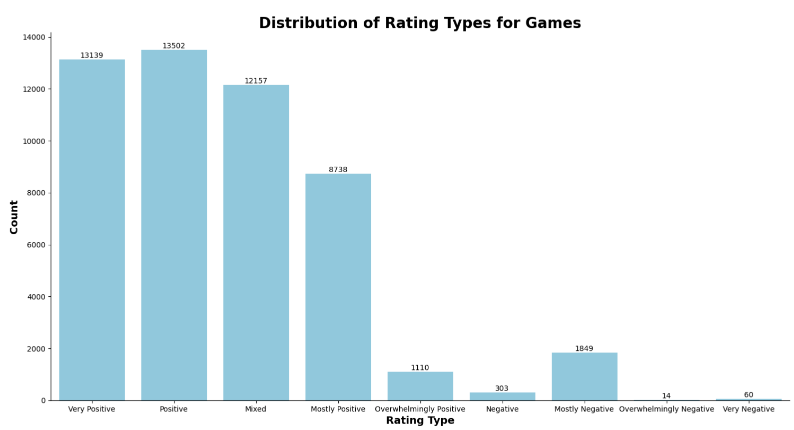
- **Games**: Stores the properties of more than 50,000 Steam games, such as titles, prices and ratings.
    - *Diverse Game Attributes*: This dataframe captures multiple dimensions of each game, including user ratings, positive feedback ratios, user review comments, and pricing. These attributes allow for rich analyses, such as trends in user ratings over time, correlations between positive ratios and game popularity, and price trend analysis.
- **Recommendations**: Detailed data of the users' reviews on the games, including the hours played and the recommendation status in binary format.
    - *Engagement and Sentiment Data*: This dataframe captures valuable insights into user behavior, including the number of hours played, recommendation status (recommended or not), and the helpful count, which shows how many users found each review helpful. These attributes can help understand the relationship between a game entity and a user entity.

- **Users**: Information about the users themselves, including the number of games owned and reviews posted.
    - *User Profile Insights*: This dataframe provides a snapshot of each user's engagement with the platform, capturing key details such as the total number of games owned and the volume of reviews posted. These attributes can help identify user profiles into certain user categories.

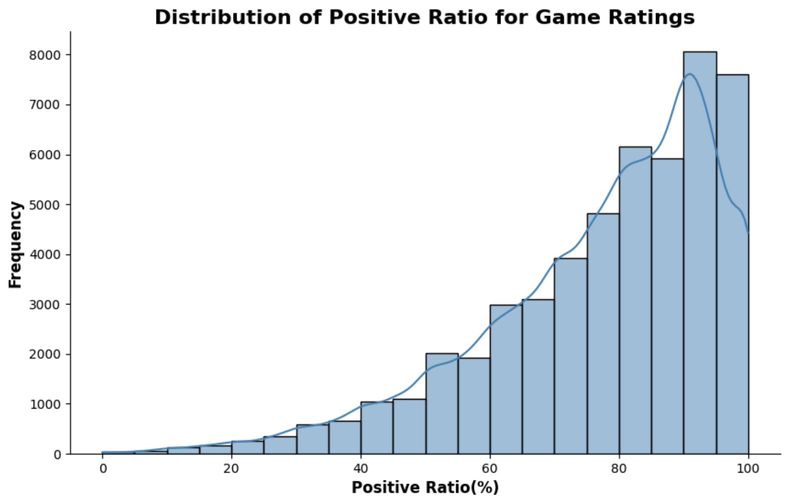
The dataset for this project was sourced from Kaggle, making the data collection straightforward as we simply downloaded the three datasets locally. To streamline our analysis, we decided to merge the user reviews and recommendations datasets based on user ID. This merging approach simplifies the dataset structure, enhancing clarity and facilitating more efficient data analysis.

## Data Exploration

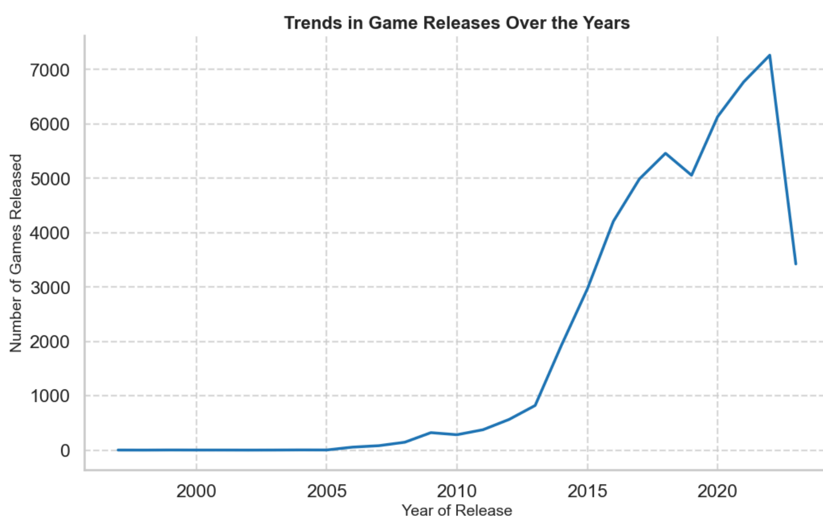
To look deeper into our three datasets: *games.csv*, *recommendations.csv*, and *users.csv*, each offering unique insights into user preferences and game attributes. The *games.csv* dataset encompasses 50,872 games, with a median of 49 user reviews per game and a median positive review ratio of 81%, highlighting a tendency toward positive user feedback. The *recommendations.csv* dataset provides detailed metrics on user engagement, recording an average of 100.6 play hours per user and a total of approximately 4.14 billion play hours across all users. The dataset also includes 35,304,398 positive recommendations and 5,850,396 negative ones. Finally, the *users.csv* dataset includes information on 14,306,064 users, 7,572 unique products, and a total of 41,154,794 reviews, providing a comprehensive view of user activity and product diversity. These datasets together form a robust foundation for analyzing user behaviors and game preferences.




The visualization shows the distribution of rating types for games in the *games.csv* dataset, with most games receiving positive ratings. "Positive" (13,502 games) and "Very Positive" (13,139 games) are the most common, followed by "Mixed" (12,157 games). Negative ratings are rare, with categories like "Negative" (303 games) and "Overwhelmingly Negative" (14 games) comprising a small fraction. Overall, the data suggests a tendency toward favorable game ratings.




The dominance of positive ratings (**Positive** and **Very Positive**) suggests that most games in the dataset are generally well-received by players, which could indicate that users are more likely to leave reviews for games they enjoy or that the market tends to favor higher-quality games.




The visualization highlights the trends in the number of game releases over the years, showing a steady output from 2000 to 2010, followed by a rapid surge after 2010, peaking at over 7,000 games around 2020. This growth aligns with key industry shifts, such as the rise of digital distribution platforms (e.g., Steam, Epic Games Store) and the increasing accessibility of game development tools like Unity and Unreal Engine, which have empowered indie developers. The sharp decline after 2020 may reflect incomplete data for recent years, disruptions such as the

COVID-19 pandemic, or a potential shift in production focus toward fewer, higher-quality titles or live-service games. This trend underscores the dynamic evolution of the gaming industry, shaped by technological advancements and market demands.

**Initial Setup for Network Analysis**

To create the bipartite graph between games and users, we combined the *recommendations.csv* and *users.csv* datasets by first standardizing the “user_id” column, converting it to a string format prefixed with "User_" in both datasets. The datasets were then merged on the “user_id” column to create the *user_reviews* dataset. To focus on positive interactions, rows where “is_recommended” was *False* were removed, leaving only reviews where users recommended the games. This refined dataset serves as the foundation for constructing the bipartite graph.

Due to the large size of the original *recommendations.csv* dataset, processing the entire combined dataset would be computationally expensive and time-consuming. To address this, we selected a sample of 10,000 rows from the combined *user_reviews* dataset. This sampling approach ensures that our analysis remains efficient while still capturing sufficient diversity and representation of the data for meaningful insights.

**Bipartite Graph**

A bipartite graph is a type of graph where nodes can be divided into two distinct sets, and edges only exist between nodes from different sets. In this context, the graph connects users and games, representing user interactions and recommendations. Using a bipartite graph allows us to analyze relationships and patterns, such as which games are popular among certain user groups or how users with similar preferences interact with games.

Here we used the sample dataset with 10,000 rows. first adds all game IDs (“app_id”) and user IDs (“user_id”) as nodes in two separate sets (bipartite = 0 for games and bipartite =1 for users). Then, edges are added between games and users based on the dataset, where each edge represents a user recommending a game. This graph is crucial for modeling user-game interactions and enabling network-based analyses, such as game recommendations or community detection.

## Graph Exploration:

**Edges and Nodes**

Basic graph metrics were then calculated to gain insights into the graph's properties. The graph contains 18,474 nodes and 51,742 edges, where each edge represents a user recommending a game. The average degree, calculated as the total number of connections divided by the total nodes, is approximately 5.6, indicating that, on average, each node (game or user) is connected to about 5.6 other nodes. These metrics provide an overview of the graph's scale and connectivity, offering a foundation for further analysis.

**Analysis of Projected Graphs (including Centrality Measures, Clustering Coefficients)**

Projections are utilized to analyze the characteristics of game nodes and user nodes independently. The rationale for using separate projections is to avoid direct comparisons between different types of nodes, ensuring a more focused and meaningful analysis. These projections play a critical role in facilitating tasks such as link prediction, which is discussed in detail later in the report.

**a) Projected Graphs**

Projected graphs are essential for simplifying bipartite networks by focusing on one type of node. In this case, the **game-game projection** connects games that share common users, while the **user-user projection** connects users who recommended the same games. These projections allow us to analyze relationships and patterns within each specific group without the added complexity of bipartite structure. For example, the game projection helps identify closely related games based on shared audience preferences.

```
Game-Game Projection - Number of nodes: 9273
Game-Game Projection - Number of edges: 766590
User-User Projection - Number of nodes: 9201
User-User Projection - Number of edges: 919538
```

The game-game projection consists of 9,273 distinct games, represented as nodes, that share at least one user, and 766,590 edges, indicating pairs of games reviewed or recommended by the same user. This projection highlights the rich connections between games, where overlapping user preferences lead to frequent co-recommendations or reviews, reflecting related genres or themes. Similarly, the user-user projection comprises 9,201 distinct users as nodes and 919,538 edges, representing pairs of users who have reviewed or recommended the same game(s). This projection reveals significant user overlap, suggesting clusters of shared interests and behaviors among users who engage with similar types of games.

**b) Degree Centrality & Clustering Coefficient**

**Degree Centrality** Analysis was conducted on the original bipartite graph to measure the importance of nodes. The degree of similarity quantifies how well-connected a node is with higher values indicating more connections. For games, this can highlight popular titles with wide user reach, while for users, it identifies individuals with diverse recommendations.

```
Top 5 Games by Degree Centrality: [(1091500, 0.026518856646016737), (292030, 0.025105966742745355), (275850, 0.024127812194326706), (440, 0.02271492229105532), (374320, 0.0
Top 5 Users by Degree Centrality: [('User_12463171', 0.027175671303785183), ('User_6714378', 0.02577375175239944), ('User_10239050', 0.025558071821417016), ('User_6808967',
```

The top games by degree centrality, such as 1091500 (Cyberpunk 2077) and 292030 (The Witcher® 3: Wild Hunt), have the highest centrality scores, meaning they are connected to many users and are highly popular or widely reviewed. These games likely have broad appeal, attracting a diverse group of players. On the other hand, the top users by degree centrality, including *User_12463171* and *User_6714378*, are highly active and engaged with a wide range of games, indicating they have diverse preferences or contribute significantly to the game ecosystem. These users are key influencers, as they interact with many games, helping to shape the gaming community's tastes.

The **Clustering Coefficient** measures the tendency of nodes to form tightly connected groups. For games, a higher clustering coefficient suggests a group of games frequently recommended together by users, potentially revealing similar genres or attributes. For users, it indicates clusters of like-minded individuals with similar preferences. These metrics provide insights into both network structure and behavioral patterns in the data.

```
Top 5 Games by Clustering Coefficient: [(1146880, 1.0), (491520, 1.0), (458770, 1.0), (30, 1.0), (819230, 1.0)]
Top 5 Users by Clustering Coefficient: [('User_12944107', 1.0), ('User_4769082', 1.0), ('User_7912946', 1.0), ('User_10821225', 1.0), ('User_13285151', 1.0)]
```

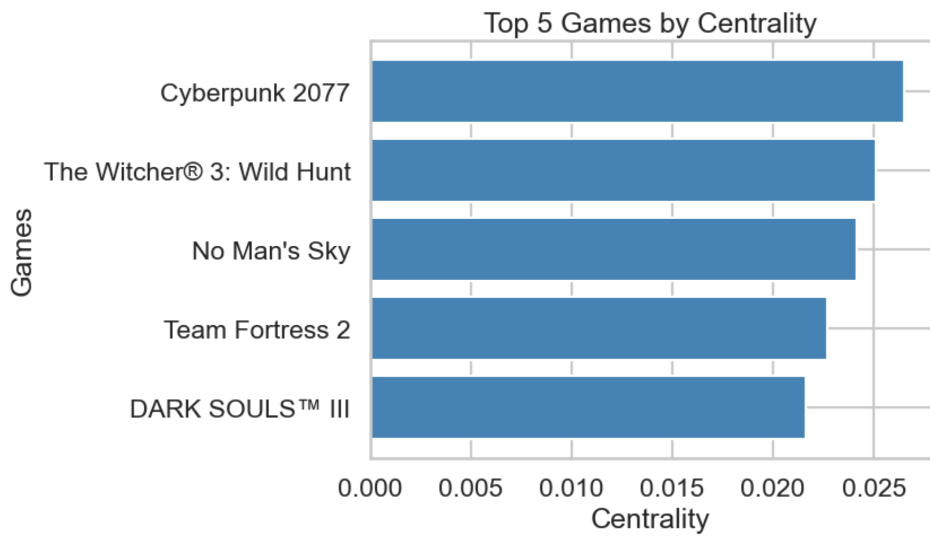
Games with a clustering coefficient of 1.0, such as 1146880 (RRRR3), are part of tightly-knit communities where every game connected to them is also connected to each other, suggesting they belong to niche genres or have strong user preferences. Similarly, users like *User_12944107*, with a clustering coefficient of 1.0, tend to review a specific set of related games, indicating their focus on particular game types or genres.

**c) Betweenness Centrality & Closeness Centrality**

For the calculations of **betweenness** and **closeness centrality**, which are computationally intensive and time-consuming, we plan to use a random sample of 5,000 records from the merged dataset. This sample is selected using scaled player hours as weights to prioritize more active users, while ensuring that highly active users had a higher likelihood of being included multiple times. This approach prioritizes computational feasibility while acknowledging the challenges of ensuring representativeness. By focusing on scaled player hours, we aim to reflect user activity levels in the sample.

**Analysis of Top 5 Games by Centrality**

We analyzed the top 5 games based on their centrality measures, ranking them to highlight their relative importance within the network.




## Edge Weight Analysis

Edge weight analysis examines the strength of connections between nodes in the game-game projection graph. In this context, edges between two game nodes are weighted by the number of users who interacted with both games. This metric highlights pairs of games with the most overlapping user bases, offering valuable insights for recommendation systems, user behavior studies, and market segmentation.

By focusing on the highest-weight edges, the analysis uncovers meaningful relationships between games, revealing patterns in user behavior and preferences. For instance, users who play RPG games might also enjoy open-world exploration games, demonstrating cross-genre appeal. Similarly, connections within franchises emphasize the loyalty of users to specific game series (Results are shown below).

```
Games: Cyberpunk 2077 ↔ The Witcher® 3: Wild Hunt (Shared Users: 29)
Shared Users: User_11264286, User_4781319, User_544393, User_479731, User_11522216, User_6131555, User_6808967, User_9617880, User_117278
Games: DARK SOULS™ III ↔ DARK SOULS™: REMASTERED (Shared Users: 23)
Shared Users: User_12772619, User_6445167, User_7252126, User_12914842, User_11464734, User_8836577, User_11526975, User_8843340, User_45
Games: DARK SOULS™ II: Scholar of the First Sin ↔ DARK SOULS™ III (Shared Users: 22)
Shared Users: User_6445167, User_12914842, User_11464734, User_5394724, User_4752351, User_4319156, User_7407706, User_6154810, User_1306
Games: Cyberpunk 2077 ↔ No Man's Sky (Shared Users: 22)
Shared Users: User_9617880, User_6808967, User_2790967, User_11727843, User_8069840, User_7201116, User_4537631, User_6264037, User_59477
Games: DARK SOULS™ III ↔ Sekiro™: Shadows Die Twice - GOTY Edition (Shared Users: 21)
Shared Users: User_6131555, User_7252126, User_1840094, User_11526975, User_2565800, User_10006340, User_5628313, User_4319156, User_6009
```

The analysis is based on a dataset sampled from 10,000 user-game interactions, introducing a degree of sparsity. This sparsity reflects the limited fraction of the overall user base captured in the sample. As a result, shared user counts between games are modest, ranging from 21 to 29 users for the top edges. While these counts may underestimate the true extent of overlaps in larger datasets, they provide a representative view of broader trends.

Despite these limitations, the results highlight key aspects of user behavior, including franchise loyalty, cross-genre engagement, and developer influence. Scaling this analysis to larger datasets could uncover additional patterns, validate these insights, and further refine applications such as recommendation systems and market strategies.

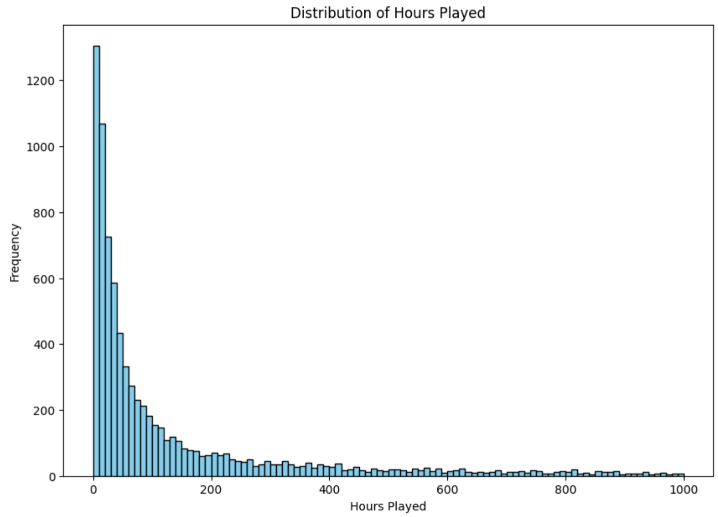
## Link Prediction Method: Weighted Random Walk

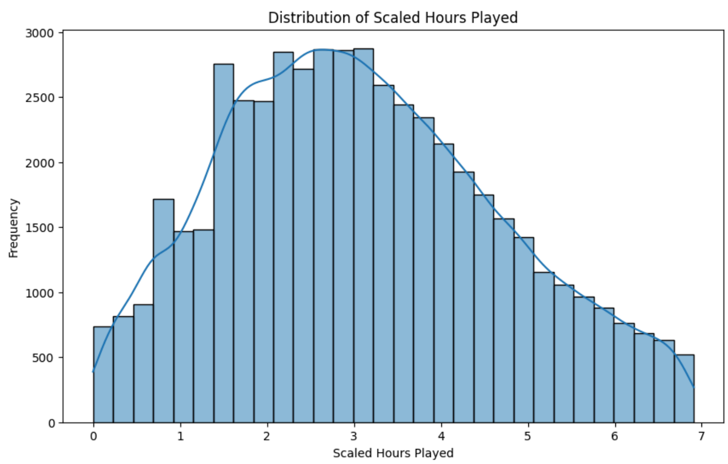
### Introduction

Most link prediction research focused on unipartite networks, such as social networks, Webpages, and email networks. Compared with those cases, link prediction in bipartite graphs presents unique challenges due to the inherent structural constraints and sparsity of connections between two different types of sets of nodes. Unlike general graphs, bipartite graphs lack direct connections within each set, making it harder to infer relationships based solely on proximity or common neighbors. Additionally, real-world bipartite graphs, such as game-user networks, often exhibit extreme sparsity, where most nodes are connected to only a few others, limiting the available information for prediction. Random walk-based methods are well-suited for addressing these challenges because they leverage the graph's global structure by simulating exploration across multiple paths. This approach allows for the discovery of indirect relationships, enabling the model to capture latent patterns and recommend connections that are not immediately obvious. Furthermore, random walks can be easily extended to incorporate edge weights, adding flexibility to prioritize certain relationships based on contextual importance, such as user engagement or preference strength. These advantages make random walks a compelling choice for link prediction in bipartite graphs.

### Step 1: Edge Weight Preparation - Data Scaling

We plan to use play hours as the edge weights to represent the users' strength of preference on the games. Scaling the play hours as edge weights is crucial for accurately reflecting the strength of relationships between users and games. Original play hours exhibit significant variability, with some users spending exponentially more time on certain games than others. Without scaling, this variability can disproportionately influence the random walk, skewing recommendations towards a few games with extremely high play times. Using a logarithmic transformation, such as scaling play hours with log(hours+1), reduces the impact of outliers while preserving the relative differences between edges. This ensures that the model captures meaningful interactions and reduces the influence of extreme values, leading to more balanced and fair predictions. After scaling the play hours, we extracted a smaller sample of 8000 rows for prediction,while keeping the distribution of scaled hours.




**Distribution of the Original Play Hours and Scaled hours**

**Step2: Train-Test Data Preparation**

The next step in creating a link prediction model involves splitting the graph into training and testing datasets while ensuring the graph remains connected. This step is crucial because removing edges arbitrarily may create disconnected components, reducing the validity of the training graph. We first identified the largest connected component of the graph to ensure we are working with a single, cohesive subgraph and then iteratively removed edges at random. After removing each edge, it was checked if the graph remained connected. As a result, 5% of the edges were removed as the test set. There were 6688 edges in the original graph, 6354 edges in the train set, and 334 test edges for evaluating the prediction.

**Step 3: Naive Random Walk**

The naive random walk serves as the baseline for predicting links by simulating simple, random exploration of the graph. It is implemented by simulating random transitions within the graph to explore potential connections. Starting from a given user node, the algorithm alternates between users and games for a specified number of steps. At each step, the algorithm randomly selects a neighbor from the current node's connections, reflecting the exploration of possible links within the graph's structure. The process repeats for multiple independent runs, and the number of visits to each game node is tracked to generate recommendations. The implementation efficiently handles the graph traversal by iterating through neighbors and avoids termination errors by checking for the presence of neighbors at each step. While simple, this approach ignores the weights of edges, treating all connections equally, which limits its ability to prioritize stronger relationships.

**Step 4: Weighted Random Walk**

The weighted random walk enhances the naive version by incorporating edge weights into the algorithm, allowing for more targeted exploration. In the implementation, at each step, the algorithm calculates the transition probabilities by calculating the edge weights of the current node's neighbors. This ensures that nodes with higher weights are more likely to be visited. These edges are likely to represent the users' stronger preference on the connected game. If the edge weights sum to zero, a neighbor will be chosen uniformly at random to avoid errors. The algorithm tracks the visit counts of game nodes across multiple independent runs and excludes already played games from the recommendations. By integrating weights, the code adapts to the graph's contextual information, making the predictions more relevant and precise.

**Step 5: Evaluation**

The evaluation process measures the predictive performance of the random walk models by comparing predicted links to ground truth connections from the test set. After splitting the graph, the test set consists of removed edges representing real connections, while negative samples are generated by pairing users and games that are not connected in the training graph, with a balanced number of positive and negative samples. For each edge, the random walk model generates a ranked list of game nodes based on visit counts, where the count serves as the predicted score for each game. Predictions are evaluated using two metrics: accuracy, which calculates the proportion of correctly predicted edges among all predictions, and Area Under the Curve (AUC), which assesses the model's ability to rank true edges higher than false edges. The predict_edges function implements this process, ensuring stable performance with solutions for edge cases like missing nodes or zero weights. The framework provides a robust measure of the model's precision and ranking ability, enabling direct comparisons between the naive and weighted random walk methods.

## Results

The results of the prediction demonstrate the advantages of incorporating edge weights into the random walk process. While both the naive and weighted random walk models effectively explore the graph and generate meaningful recommendations, the weighted random walk consistently achieves higher accuracy and AUC scores. This improvement highlights the value of using edge weights to prioritize stronger relationships, such as games a user has spent more time playing. It was also found that the accuracy increases as the steps and number of rounds increase, but also makes it more time consuming. Taking the efficiency and accuracy into consideration, we finally chose the weighted random walk with 30 steps and 100 runs.

| | Random Walk | Accuracy | AUC |
|---|---|---|---|
| 0 | Naive Random Walk | 0.551 | 0.663 |
| 1 | Weighted Random Walk | 0.644 | 0.714 |

**Evaluation Results of the Two Models**

While the weighted random walk demonstrates improved accuracy over the naive approach, there are several approaches to further enhance prediction performance. First, incorporating additional features, such as node attributes, can provide richer information for the model. This can be achieved by extending the random walk to include biased transitions based on similarity between node attributes, which may be achieved by collaborative filtering. Second, the edges weights can be refined to reflect more detailed interactions, such as the users' rating on the games. These enhancements would make the model more robust and better suited for complex, real-world link prediction scenarios.

## Challenges

Our research encountered several significant challenges in analyzing Steam's network data and implementing link prediction algorithms. The sheer scale of the dataset presented substantial computational constraints, necessitating careful sampling strategies that balanced representativeness with tractability. While our 10,000-row sample provided valuable insights, it potentially undersold the true complexity of user-game interactions in Steam's complete ecosystem. The extreme sparsity of the bipartite graph posed another significant challenge, as most users interact with only a small fraction of available games, making it difficult to establish reliable connection patterns.

The heterogeneity of gaming behavior complicated our analysis, as play time distributions showed extreme variance – some users accumulated thousands of hours in single games while others spread limited time across many titles. This required careful consideration in edge weight scaling to prevent highly engaged users from dominating the random walk predictions. Additionally, the temporal nature of gaming interactions presented challenges, as user preferences and game popularity evolve over time, but our static graph representation couldn't capture these dynamics.

Implementation of the weighted random walk algorithm faced technical hurdles in maintaining graph connectivity during train-test splits while preserving meaningful edge weights. The computational complexity of calculating centrality measures for the full network necessitated further sampling, potentially missing important structural properties. Finally, the lack of additional context about games (such as genres, release dates, or price points) and users (such as geographic location or platform preferences) limited our ability to validate and interpret the discovered network patterns.

## Conclusions

This research provides valuable insights into the application of network analysis and link prediction for large-scale gaming platforms. Our analysis of Steam's user-game interactions through bipartite network modeling revealed significant patterns in gaming preferences and community structure. The implementation of weighted random walk algorithms demonstrated meaningful improvements over naive approaches in predicting user-game connections, achieving superior accuracy and AUC scores by incorporating play time as edge weights.

The network metrics analysis revealed important structural properties of the gaming ecosystem. High clustering coefficients in both game and user projections indicated strong community formation around particular game types or user preferences. Centrality measures successfully identified influential games and users within the network, providing potential leverage points for recommendation systems. The edge weight analysis uncovered meaningful relationships between games, highlighting both expected connections (within franchises) and potentially valuable cross-genre relationships.

Our findings have significant implications for gaming platforms and digital marketplaces. The success of weighted random walks in link prediction suggests that incorporating user engagement metrics (like play time) can substantially improve recommendation accuracy. The discovered network structure provides insights for content discovery and community building, while the challenges encountered offer valuable lessons for scaling recommendation systems.

Future work could extend this research by incorporating temporal dynamics, additional context features, and more sophisticated random walk variants. Integration with collaborative filtering techniques and exploration of other network algorithms could further improve prediction accuracy. Additionally, investigating the impact of network structure on game success and user retention could provide valuable insights for platform operators and game developers.

This study contributes to the growing body of research on large-scale digital platforms and demonstrates the potential of network analysis in understanding and optimizing user experiences in gaming ecosystems. The methodologies developed here could be adapted for other digital marketplaces facing similar challenges in content discovery and recommendation.

## References:

Allali, O., Magnien, C., & Latapy, M. (2011). Link prediction in bipartite graphs using internal links and weighted projection. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 936-941). IEEE. https://doi.org/10.1109/INFCOMW.2011.5928947

Bakshy, E., Rosenn, I., Marlow, C., & Adamic, L. (2012). The role of social networks in information diffusion. In *Proceedings of the 21st International Conference on World Wide Web* (pp. 519-528). ACM. https://doi.org/10.1145/2187836.2187907

Delimitrou, C., & Kozyrakis, C. (2014). Quasar: Resource-efficient and QoS-aware cluster management. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems* (pp. 127-144). ACM. https://doi.org/10.1145/2541940.2541941

Gomez-Uribe, C. A., & Hunt, N. (2015). The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems, 6*(4), Article 13. https://doi.org/10.1145/2843948

Li, X., & Chen, H. (2013). Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems, 54*(2), 880-890. https://doi.org/10.1016/j.dss.2012.09.019

Sifa, R., Drachen, A., & Bauckhage, C. (2018). Profiling in Games: Understanding Behavior from Telemetry. In K. Lakkaraju, G. Sukthankar, & R. T. Wigand (Eds.), *Social Interactions in Virtual Worlds: An Interdisciplinary Perspective*(pp. 337-374). Cambridge University Press.

Sifa, R., Drachen, A., & Bauckhage, C. (2021). Large-scale cross-game player behavior analysis on Steam. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 11*(1), 198-204. https://doi.org/10.1609/aiide.v11i1.12804