Personalised Game Recommendation System

Aditya Sharma(2022038)
Kanishk Kumar Meena(2022233)
Vansh Aggarwal(2022558)
Group - 42

End sem project

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

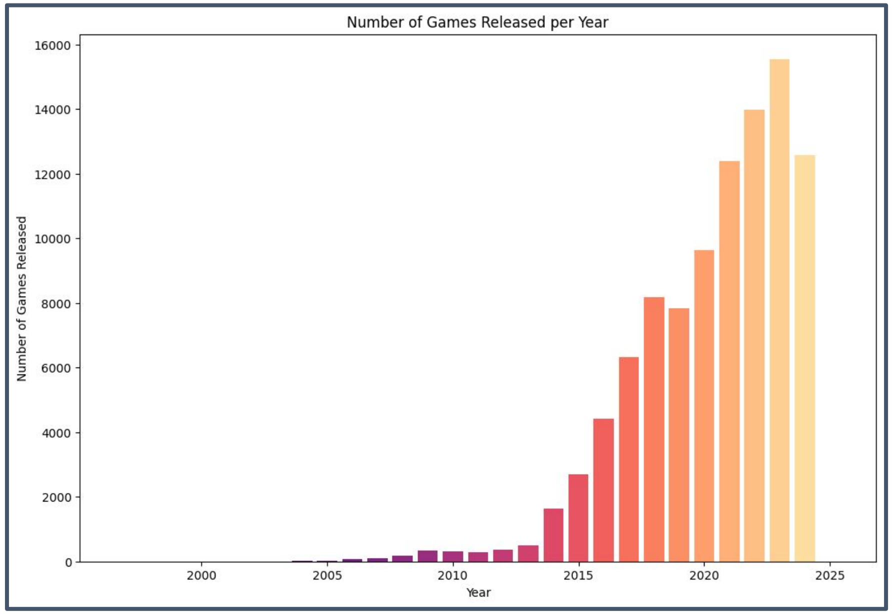
Motivation

Why This Project?

- The vast and ever-growing selection of video games on platforms like Steam can overwhelm players, often resulting in decision paralysis. Our project addresses this issue by developing a personalized game recommendation system, streamlining the game discovery process to help players find titles that align with their preferences and playstyles.

Inspiration

- This project was inspired by the challenges gamers face in navigating extensive game libraries. Observing the struggle to discover new or niche titles that match individual interests, we aimed to leverage machine learning to create a recommendation system that enhances the gaming experience by suggesting both popular and lesser-known titles tailored to user tastes.



Graph of Number Of Games released per year from our Dataset. Show an exponential pattern.
Number of Games being released each year continues to grow rapidly.
Making it difficult to find Games that you may like.

Literature Review

A Machine-Learning Item Recommendation System for Video Games

- *Paul Bertens, Anna Guitart, Pei Pei Chen and Africa Periancz, Paper Link*

- This study explores machine-learning models, such as Extremely Randomized Trees (ERT) and Deep Neural Networks (DNN), to create personalized recommendations in video games. Unlike traditional recommendation systems that rely on static user profiles and collaborative filtering, these models leverage real-time user behavior to dynamically adapt to player preferences. The ERT model demonstrated higher accuracy and scalability, making it well-suited for in-game recommendations in free-to-play games.

A Review of Content-Based and Context-Based Recommendation Systems

- *Umair Javed and Kamran Shaukat , Paper Link*

- This article reviews content-based and context-based recommendation systems, highlighting their techniques and applications across various fields, particularly in e-learning and media recommendations. It discusses the roles of collaborative filtering, semantic reasoning, and ontology representation in enhancing recommendation accuracy. The research emphasizes the significance of contextual information in improving user experience while addressing challenges like information overload and cold-start problems.

Literature Review

Recommender Systems for Online Video Game Platforms: the Case of STEAM

- *Germán Cheuque,José Guzmán and Denis Parra, [Paper Link]()*
- This paper investigates recommender systems for online video game platforms, specifically STEAM, addressing the challenge of information overload in the industry. It tests various state-of-the-art models, including Factorization Machines (FM), Deep Neural Networks (DeepNN), and a hybrid (DeepFM), to improve game recommendations. The study finds that DeepNN performs best in terms of accuracy, novelty, and diversity, while sentiment analysis from game reviews proved less impactful than anticipated.

Game Recommendation System

- *Man-Ching Yuen , Chi-Wai Yung , Wing-Fat Cheng,Hon-Pong Tsang,Chi-Ho Kwan,Chun-Lok Chan andPo-Yi Li, [Paper Link]()*
- A game recommendation system was developed by a team from the Vocational Training Council and Hong Kong Shue Yan University to help users navigate online gaming platforms like Steam. The system employs machine learning and data visualization techniques to enhance user experience and provide personalized game suggestions. Key challenges include data collection and user interface design. The goal is to improve recommendation accuracy and flexibility compared to existing systems.

Concluding Remarks:

- *The literature review provides essential insights for enhancing our game recommendation system.*
- *It highlights key methodologies that will inform our future research directions.*

Dataset Description

We have done both Content-Based Filtering and Collaborative Filtering

- **Data for Content-Based Filtering:**
  - Data sourced from Kaggle, Video Games Recommendation System.
  - The dataset was obtained from Video Game Sales with Ratings in Kaggle, which were web scraped by Gregory Smith from VGChartz Video Games Sales. The collection of data includes details such as the game's title, genre, the platform it runs on, the company that published it, and other relevant information. From year 1980 up to 2020, the dataset includes a wide range of video game releases that spans over four decades. From year 1980 up to 2020, the dataset includes a wide range of video game releases that spans over four decades.

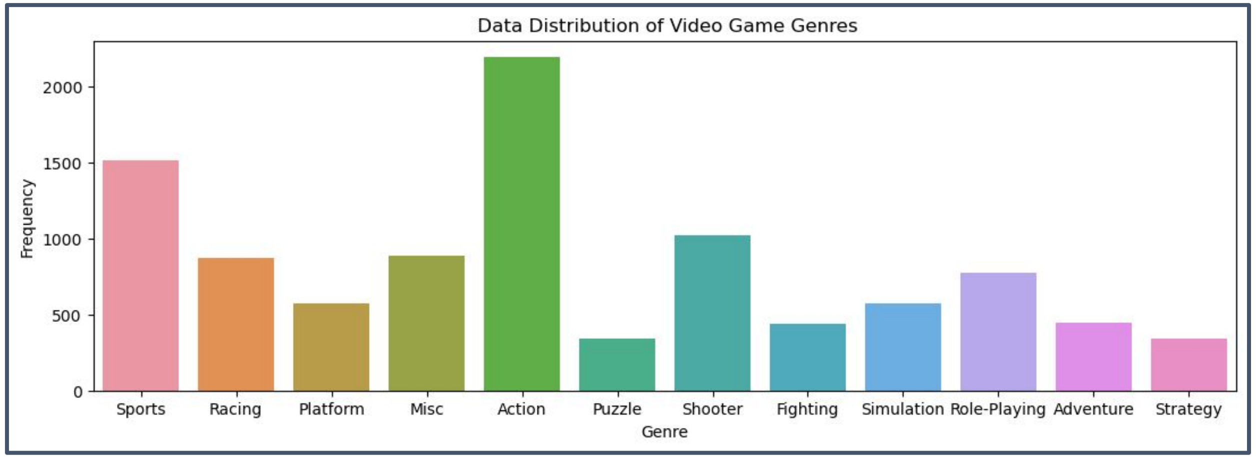
- **Data for Collaborative Filtering:**
  - Data sourced from Kaggle, Game Recommendations on Steam.
  - The dataset contains over 41 million cleaned and preprocessed user recommendations (reviews) from a Steam Store - a leading online platform for purchasing and downloading video games, DLC, and other gaming-related content. Additionally, it contains detailed information about games and add-ons. The dataset was collected from Steam Official Store. All rights on the dataset thumbnail image belong to the Valve Corporation.

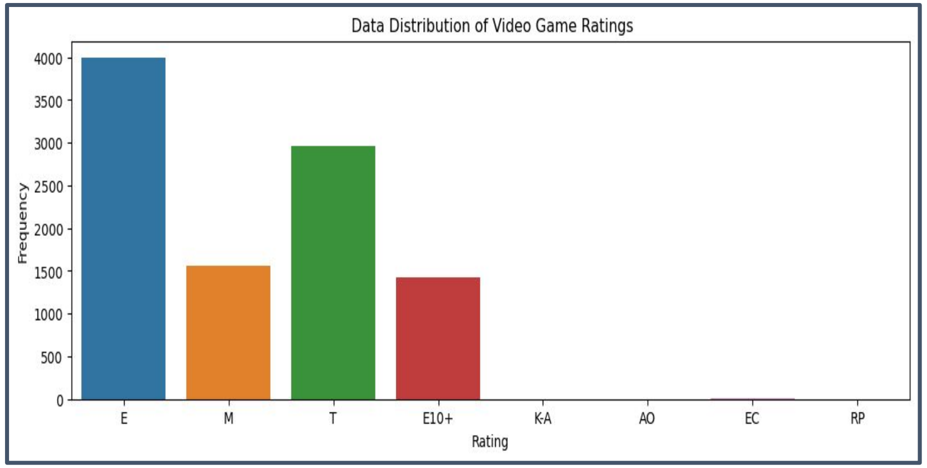
Dataset Description (Content-Based Filtering)

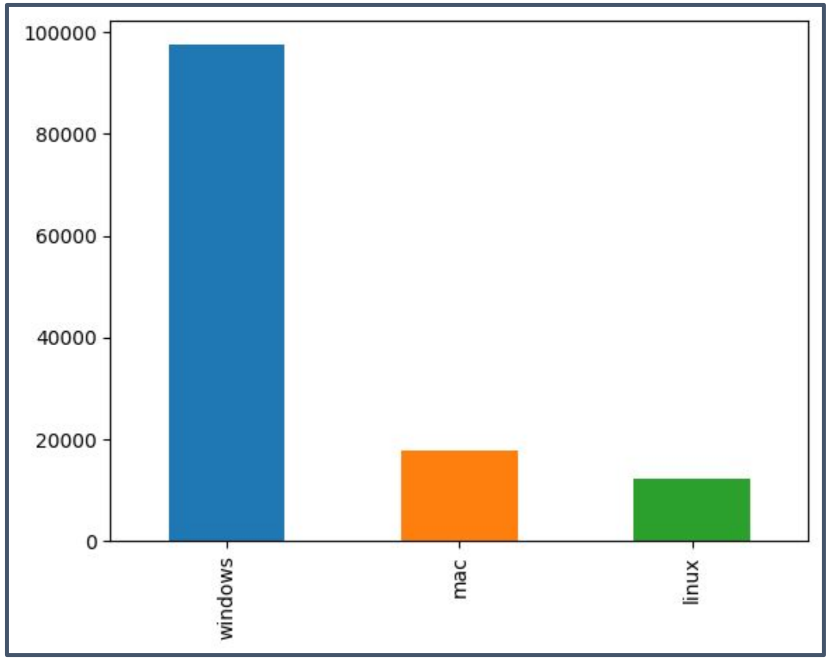
name	release_date	price	dlc_count	detailed_description	about_the_game	short_description	windows	mac	linux	achievements
Object	Object	Float64	Int64	Object	Object	Object	Boolean	Boolean	Boolean	Int64
Name of the video game	Release date of the video game	Price of the video game	Number of downloadable content (DLC) available	Detailed description of the game	Summary of the game	Brief description of the game	Indicates if the game is available on Windows	Indicates if the game is available on macOS	Indicates if the game is available on Linux	Number of achievements available in the game

supported_languages	full_audio_languages	developers	publishers	categories	genres	estimated_owners	average_playtime_forever	average_playtime_2weeks	tags	aggregate_score
Object	Object	Object	Object	Object	Object	Int64	Int64	Int64	Object	Int64
Languages supported by the game	Languages with full audio support	Names of the game developers	Names of the game publishers	Categories the game belongs to (e.g., Action, RPG)	Genres the game belongs to (e.g., Shooter, Adventure)	Estimated number of owners	Average playtime for all players	Average playtime in the last 2 weeks	Tags associated with the game (e.g., "open-world")	Aggregate score of the game

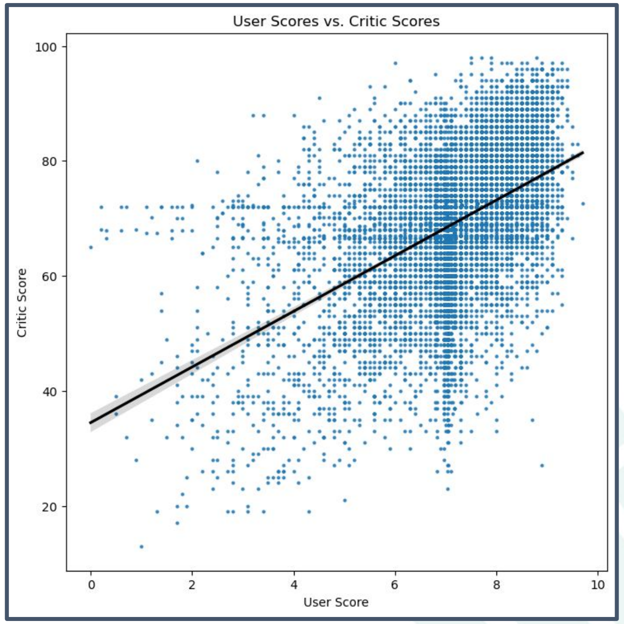
Dataset Description (Content-Based Filtering)







- Windows remains to be dominant platform for gaming
- The distribution of games across different genres in consistent.
- Positive Correlation between User Score & Critic Score can be seen from their pair plot
- Majority of games remain to be "rated for all" to cater to larger market.



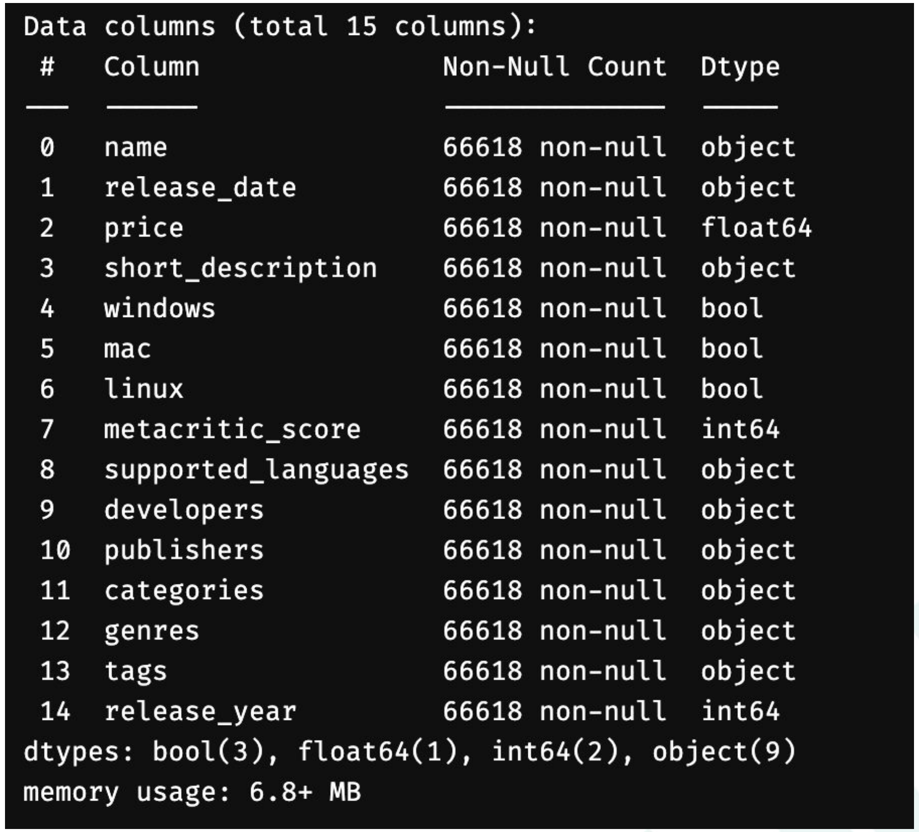
Dataset Description (Content-Based Filtering)

Preprocessing Requirements:

- **Feature Analysis:**
  - Count of games with specific conditions:
    1. Metacritic score ≠ 0
    2. Recommendations ≠ 0
    3. User score ≠ 0
    4. Positive reviews and negative reviews ≠ 0
  - For some games we scraped scores from steam
  - Total of 52,000 games had a score after this.
- **Aggregate Feature Creation:**
  - Added a new column, aggregate_score, calculated as follows:
    1. Metacritic Score: Directly used (0 to 100).
    2. User Score: Directly used (0 to 100).
    3. Positive/Negative Ratio: Normalized to a 0-100 scale, ensuring both counts are ≠ 0.
    4. Recommendations: Normalized using a normal distribution, where mean values receive a score of 50 based on percentile ranking.
    5. Average Playtime: Normalized similarly to ensure consistent scaling.

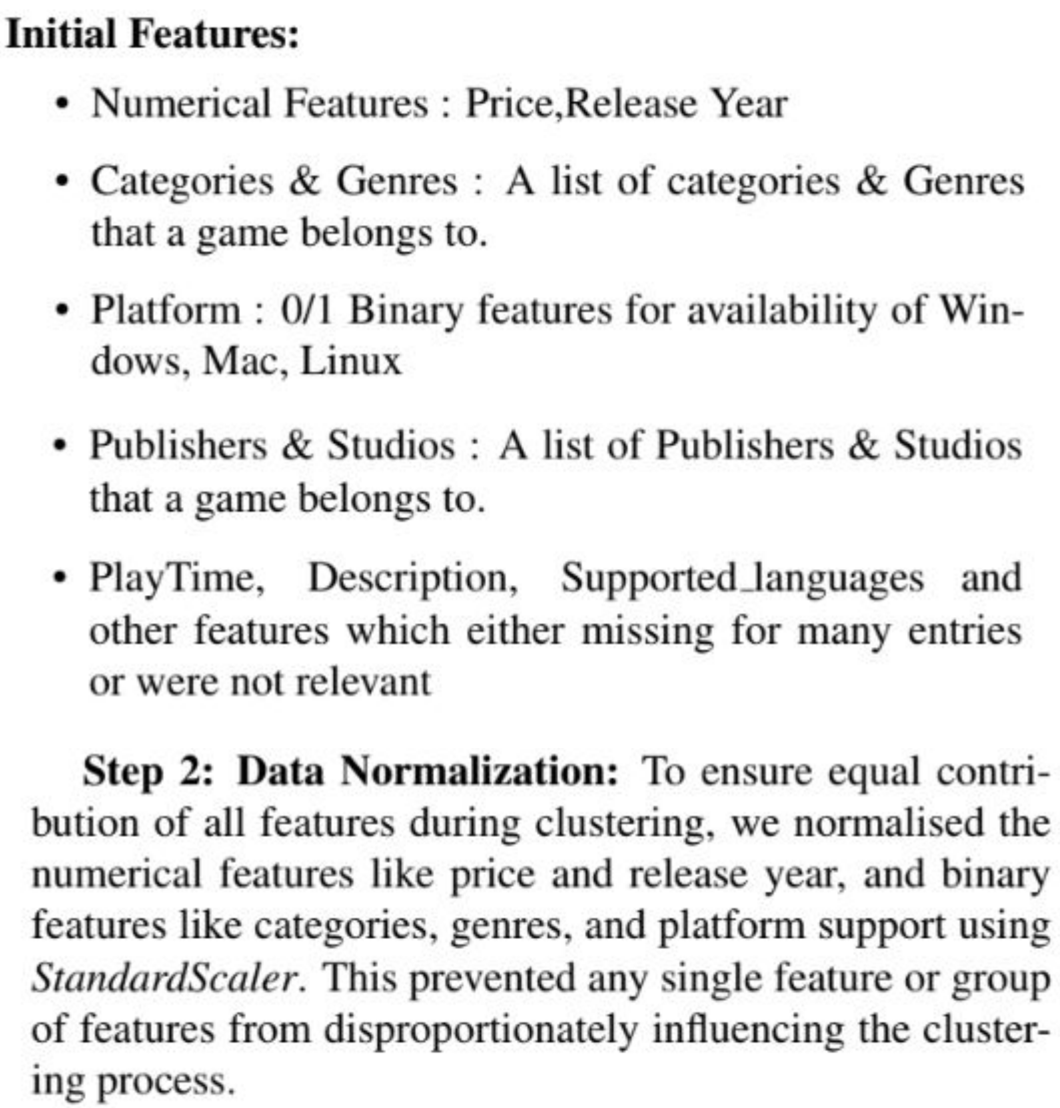
The dataset contains games.csv

- After Addition Of aggregate_score and removal of unnecessary columns we were left with



```
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   name                 66618 non-null  object
 1   release_date         66618 non-null  object
 2   price                66618 non-null  float64
 3   short_description    66618 non-null  object
 4   windows              66618 non-null  bool
 5   mac                  66618 non-null  bool
 6   linux                66618 non-null  bool
 7   metacritic_score     66618 non-null  int64
 8   supported_languages  66618 non-null  object
 9   developers           66618 non-null  object
 10  publishers           66618 non-null  object
 11  categories           66618 non-null  object
 12  genres               66618 non-null  object
 13  tags                 66618 non-null  object
 14  release_year         66618 non-null  int64
dtypes: bool(3), float64(1), int64(2), object(9)
memory usage: 6.8+ MB
```

Data PreProcessing (Content Based Filtering)



Initial Features:

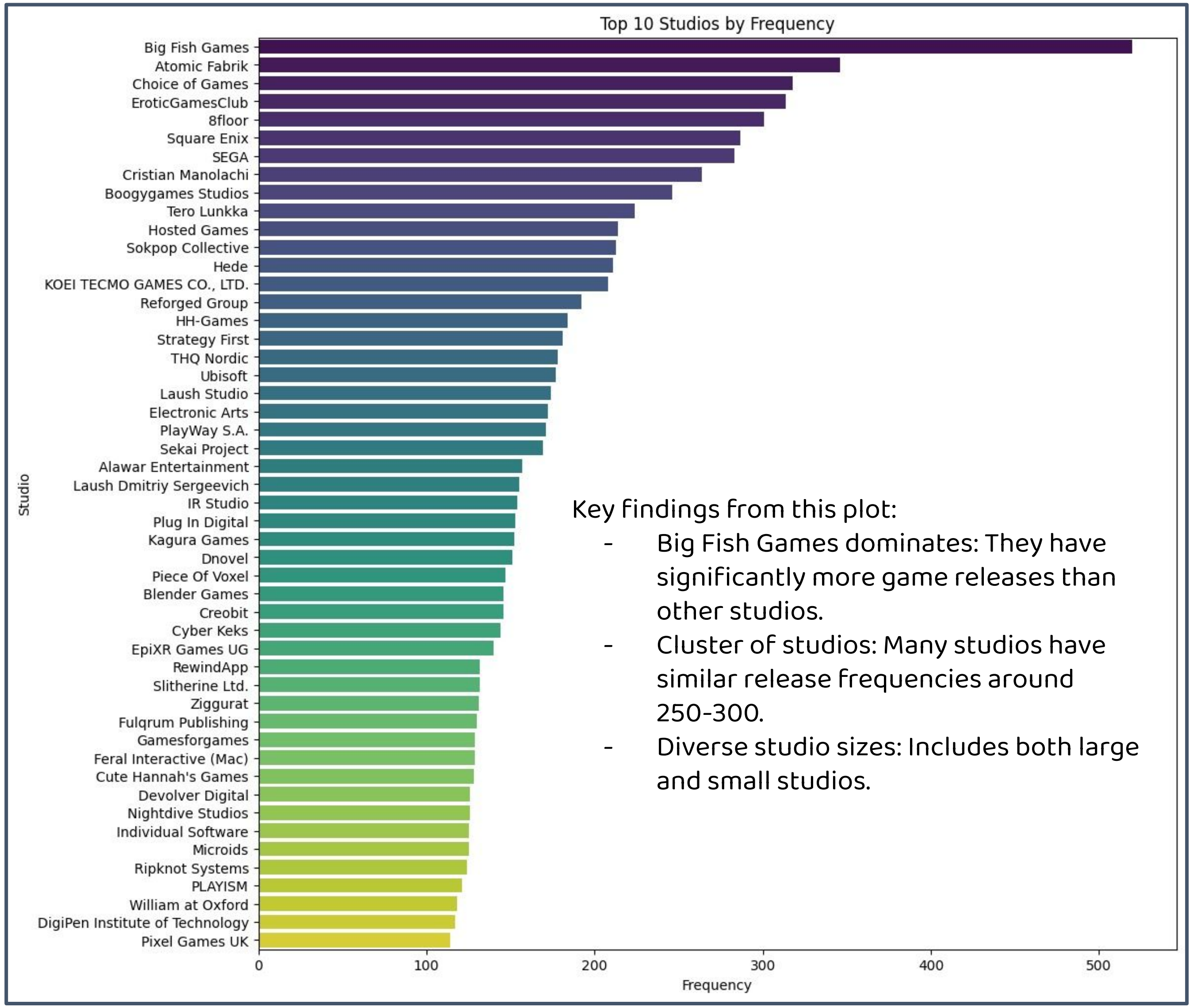
- Numerical Features : Price,Release Year
- Categories & Genres : A list of categories & Genres that a game belongs to.
- Platform : 0/1 Binary features for availability of Windows, Mac, Linux
- Publishers & Studios : A list of Publishers & Studios that a game belongs to.
- PlayTime, Description, Supported_languages and other features which either missing for many entries or were not relevant

Step 2: Data Normalization: To ensure equal contribution of all features during clustering, we normalised the numerical features like price and release year, and binary features like categories, genres, and platform support using *StandardScaler*. This prevented any single feature or group of features from disproportionately influencing the clustering process.

Genres, Categories were separated in 1-hot encoded to separate binary 0/1 columns. Numerical Features when Normalized using Standard Scaler

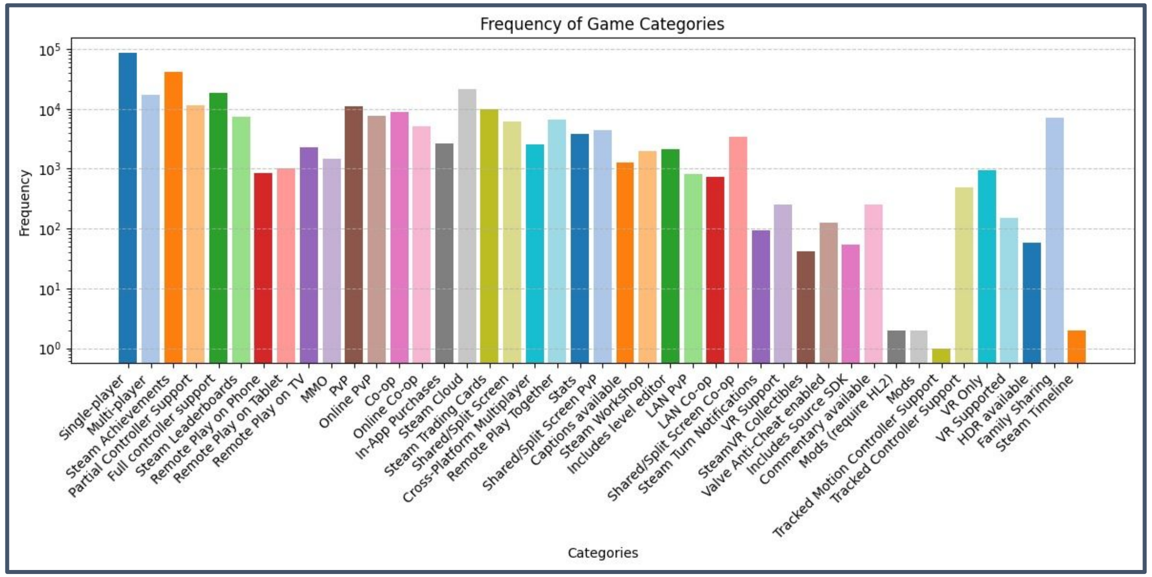
```
#   Column                                            Dtype
---  ------                                 --------  -----
 1   windows                                non-null  int64
 2   mac                                    non-null  int64
 0   release_year                           non-null  int64
 3   linux                                  non-null  int64
 4   price                                  non-null
float64
 5   categories                             non-null  object
 6   genres                                 non-null  object
 7   release_year                           non-null  int64
 8   game_studios                           non-null  object
 9   categories_includes_level_editor       non-null  int64
 10  categories_<category_name>             non-null  int64
......
 52  genres_nudity                          non-null  int64
 53  genres_casual                          non-null  int64
 54  genres_short                           non-null  int64
 55  genres_video_production                non-null  int64
```

Data Analysis



Key findings from this plot:

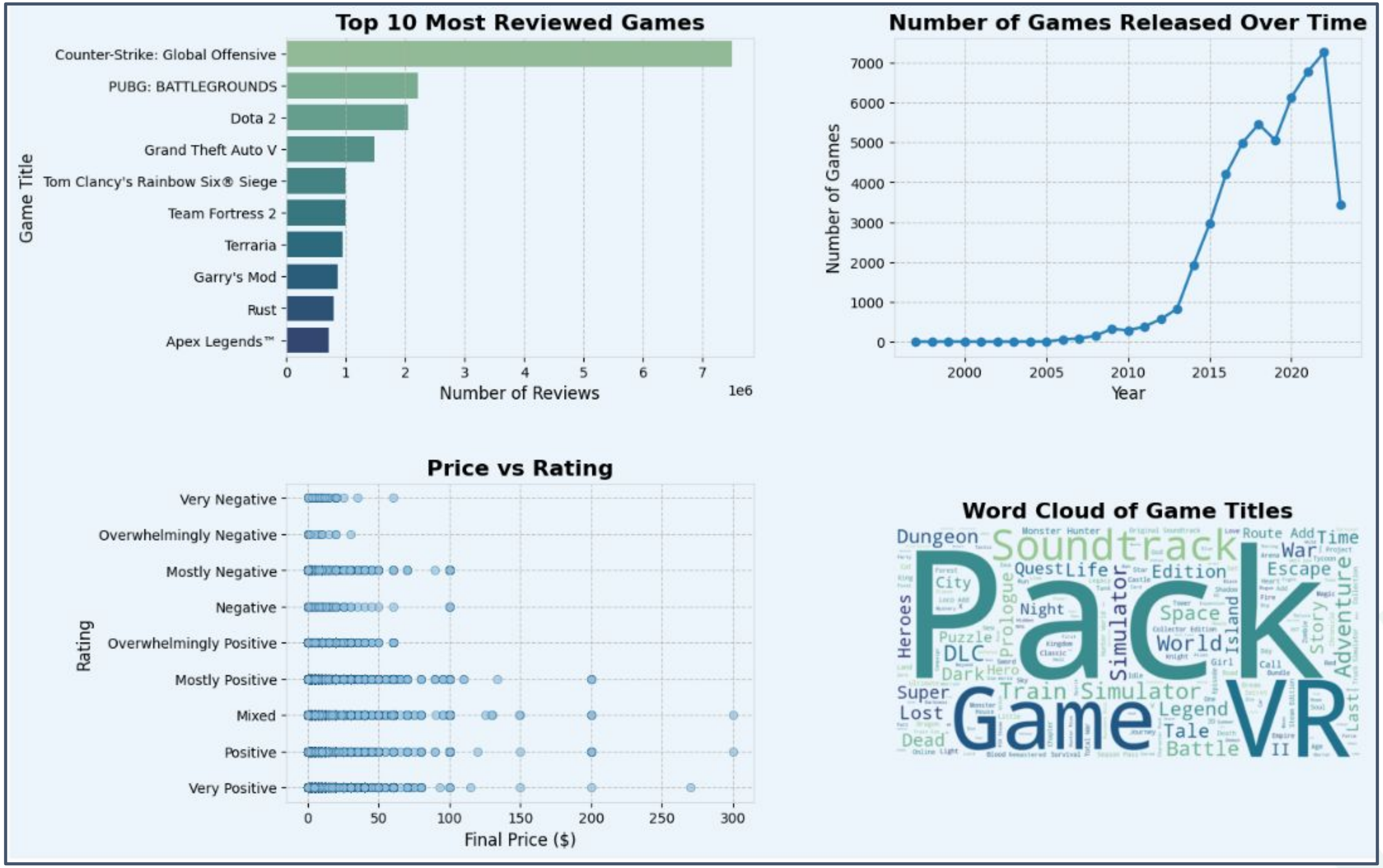
- Big Fish Games dominates: They have significantly more game releases than other studios.
- Cluster of studios: Many studios have similar release frequencies around 250-300.
- Diverse studio sizes: Includes both large and small studios.



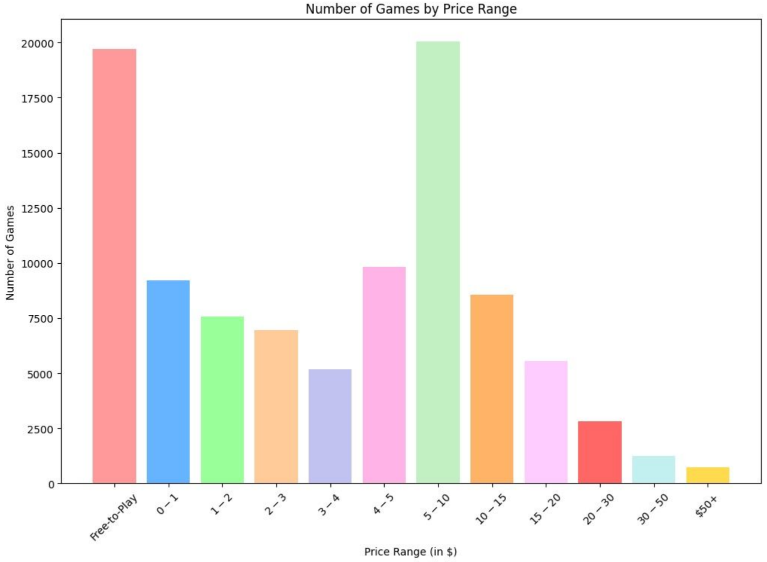
Findings from above plot:

1. Single-player and multiplayer: Most popular categories.
2. Controller support: High frequency, indicating preference for gamepads.
3. Steam features: Popular among gamers.
4. Online features: Reflect growing popularity of multiplayer gaming.
5. VR and AR: Less frequent but present, indicating growing interest.

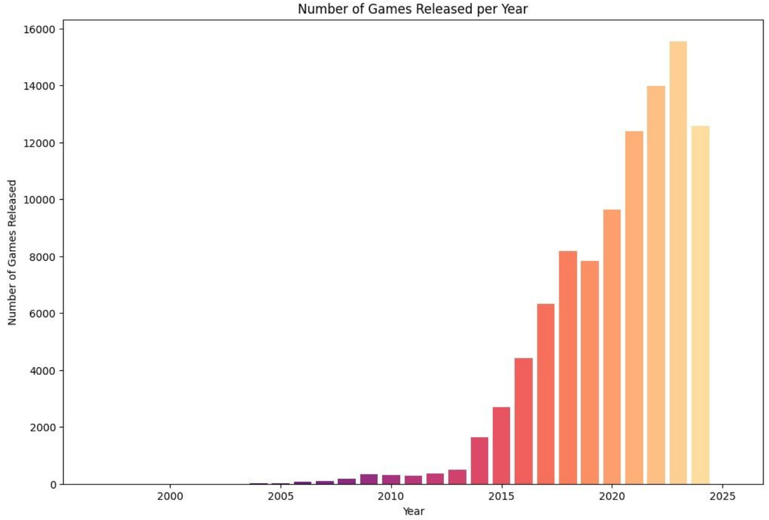
Data Analysis



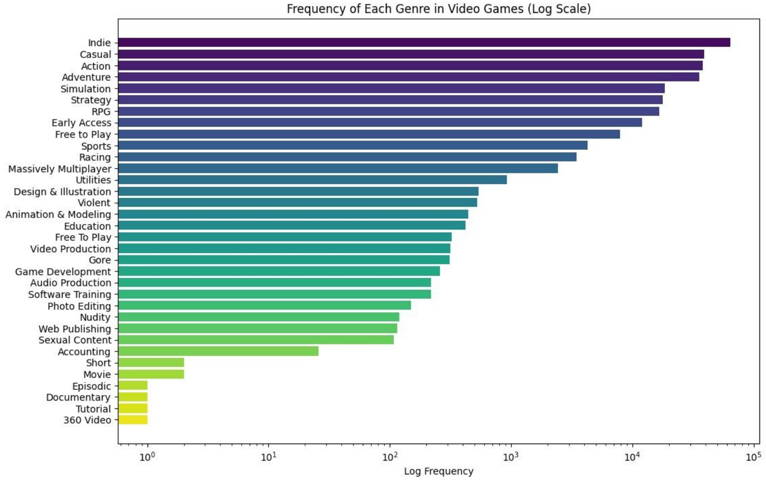
Data Analysis



1. Free-to-play dominance: Most games are free.
2. Price range distribution: Fewer games are priced higher.
3. Price clusters: Games are often priced around certain points.
4. Long tail: A small number of games are priced $50+.



1. Steady growth: Number of games released has consistently increased.
2. Accelerated growth: Growth rate has increased in recent years.
3. Year-over-year fluctuations: Some variation in release numbers.



1. Indie and Casual: Most popular genres.
2. Action and Adventure: Also popular.
3. Diverse genres: Wide range of genres represented.

Methodology (Content Based Filtering)

Objective: Develop a game recommendation system also using data from game studios.
Challenge: Over **50,000 unique values** in game studios data (publishers and developers).
Approach: A systematic method including data combination, encoding, feature selection, normalization, clustering, and model training.
As, direct 1-hot encoding for these many unique values.
Label Encoding would also be problematic as it will assign a sense of order in studios, which may create bias

Step 1: Combining Publishers and Developers

- Created a new column game_studios to merge publishers and developers for each game.
- Resulted in multiple studios per game, forming lists of studios.

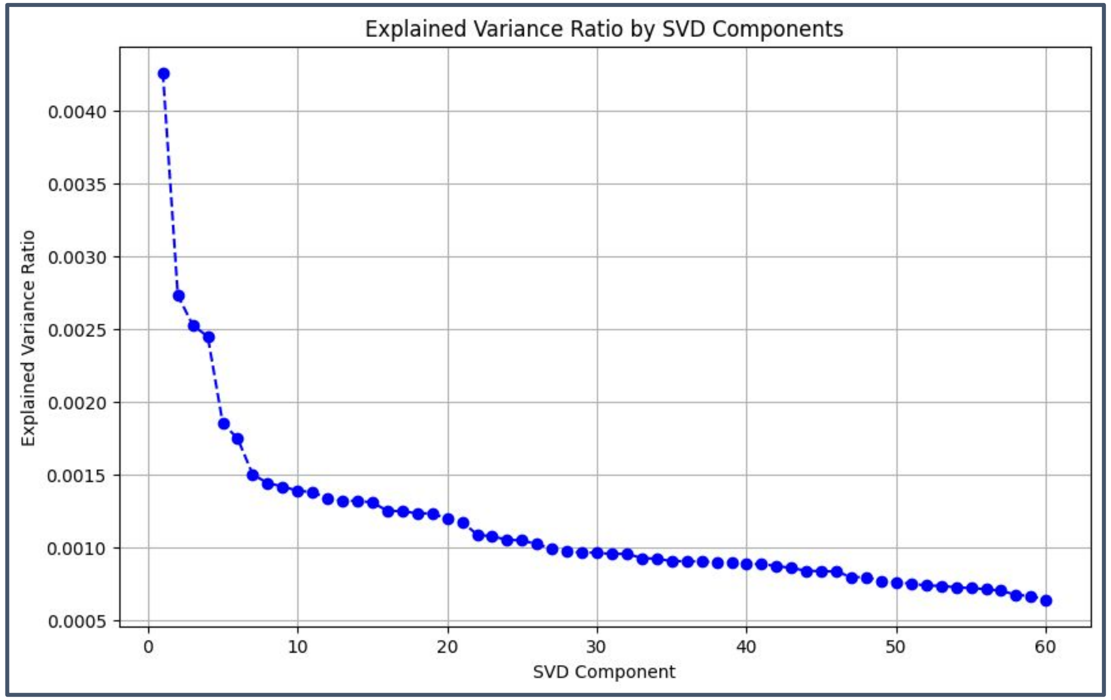
Step 2: Multi-Hot Encoding

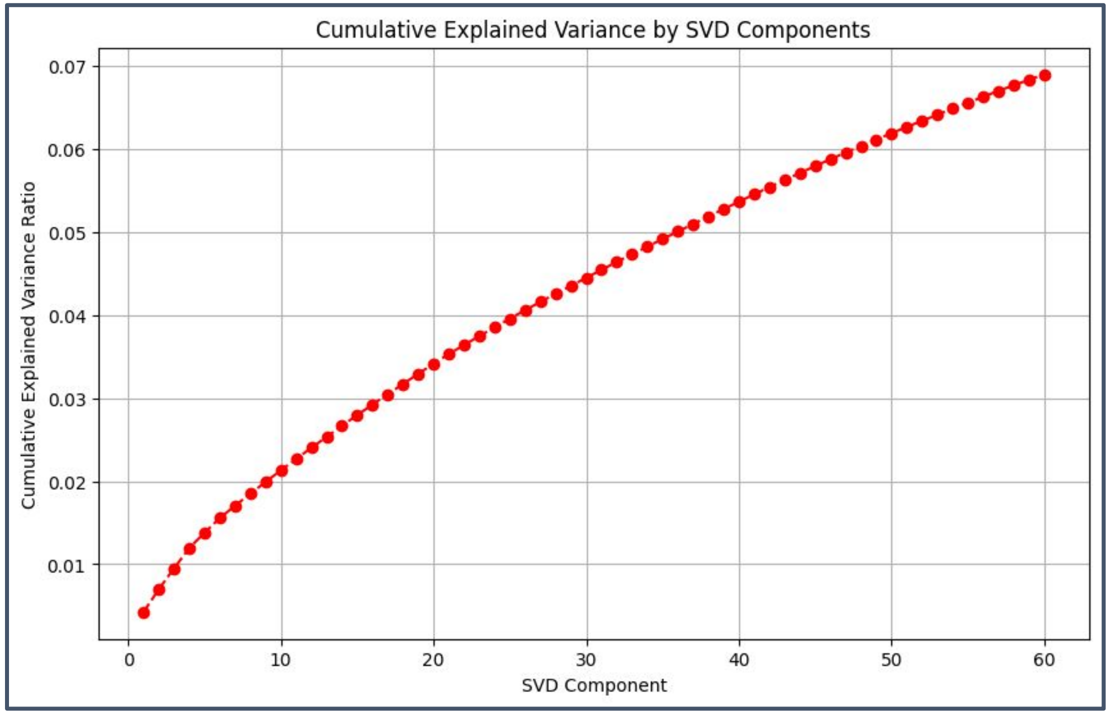
- Applied multi-hot encoding to transform studio lists into binary columns (0 or 1).
- Created a sparse matrix, challenging for clustering due to the high dimensionality.

Step 3: Dimensionality Reduction with Truncated SVD

- Used Truncated SVD to reduce the sparse matrix dimensionality to 60 components.
- Explained variance was low (6.8%), indicating minimal information captured due to sparsity.

Methodology





Doing Dimensionality Reduction for game_studios dataset
For our case, Truncated SVD is the better choice because as our data is sparse:

1. It handles sparse data efficiently, which is crucial when dealing with the large number of unique studios and the resulting sparse multi-hot matrix.
2. It doesn’t require mean-centering the data, making it more suited for binary features.
3. It's computationally more efficient for the large and sparse game studio dataset you’re dealing with.

Not able to get very good reduction. Not able to capture good amount of variance

Methodology – Feature Selection and Normalization

Step 1: Final Feature Set: Composed of:

- Numerical Features:
  1. Price
  2. Release Year
- Binary Features:
  1. One-hot encoded categories (40 features) and genres (32 features)
  2. Example categories: Single Player, Co-op, Steam Achievements.
- Platform Support:
  1. Binary columns for Windows, Mac, and Linux.
- Studio Clusters:
  1. Derived from K-Means clustering, each game assigned to one of 10 clusters (Cluster 0 to Cluster 9).

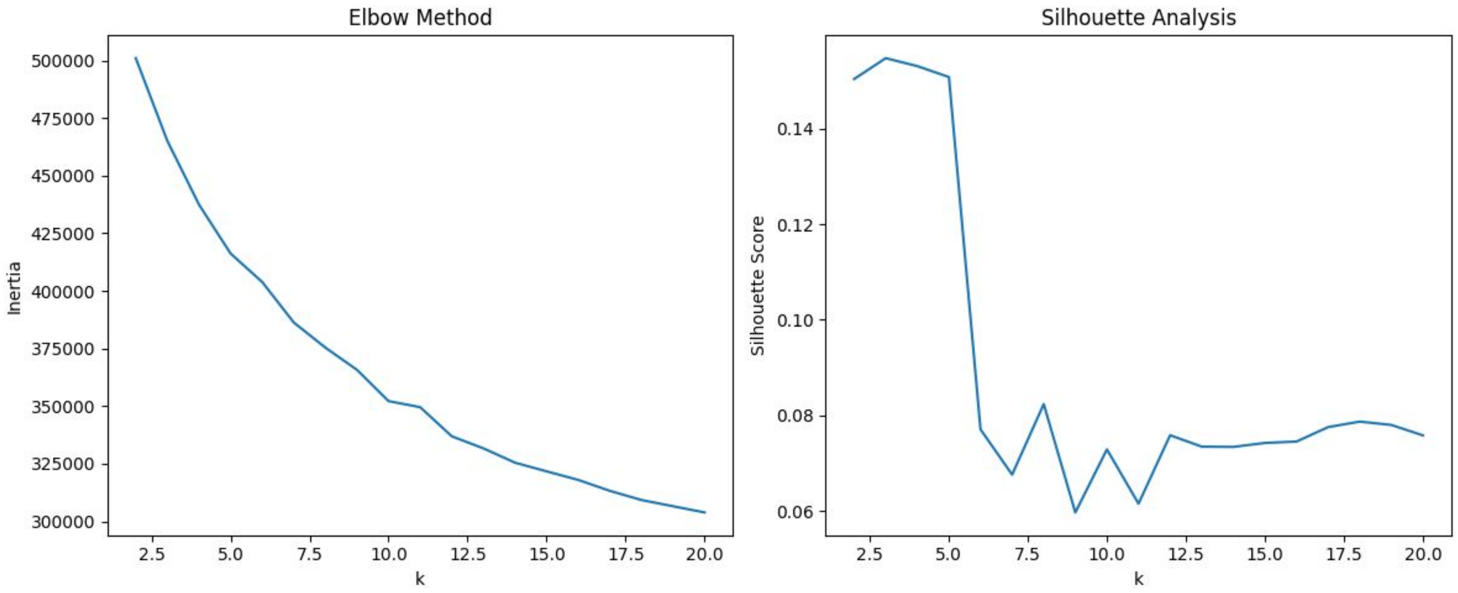
Step 2: Data Normalization

- Standardized features using StandardScaler to ensure equal contribution.
- Both numerical (price, release year) and binary features normalized to prevent bias in clustering.

Step 3: K-Means Clustering

- Applied K-Means clustering on the processed, normalized features.
- Hypothesis: Similar studios produce similar games; thus, they cluster together.

Methodology(Content Based Filtering) -
Feature Selection and Normalization



Step 4: Determining Optimal Clusters

- Utilized the Elbow Method and Silhouette Analysis to find the optimal number of clusters (k).
- Elbow Method indicated k = 5 as optimal, while Silhouette Analysis supported this choice with scores peaking between k = 3 and k = 5.

Step 5: Final Model Training and Cluster Assignment

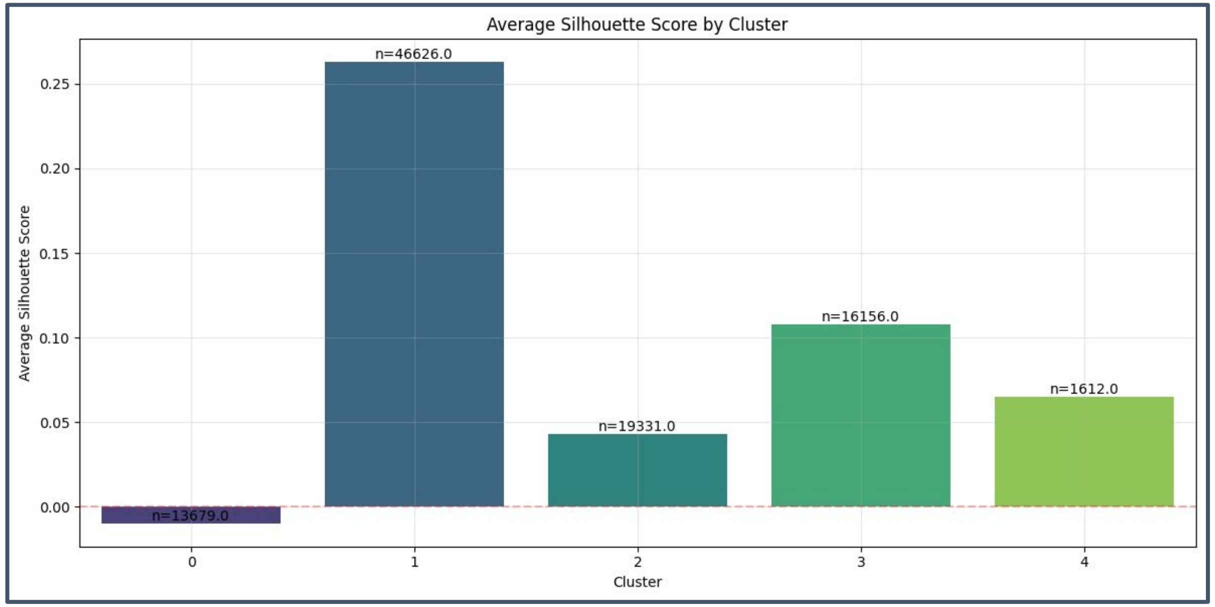
- Trained final K-Means model with k = 5.
- Generated cluster assignments for each game, establishing a robust basis for the recommendation system.

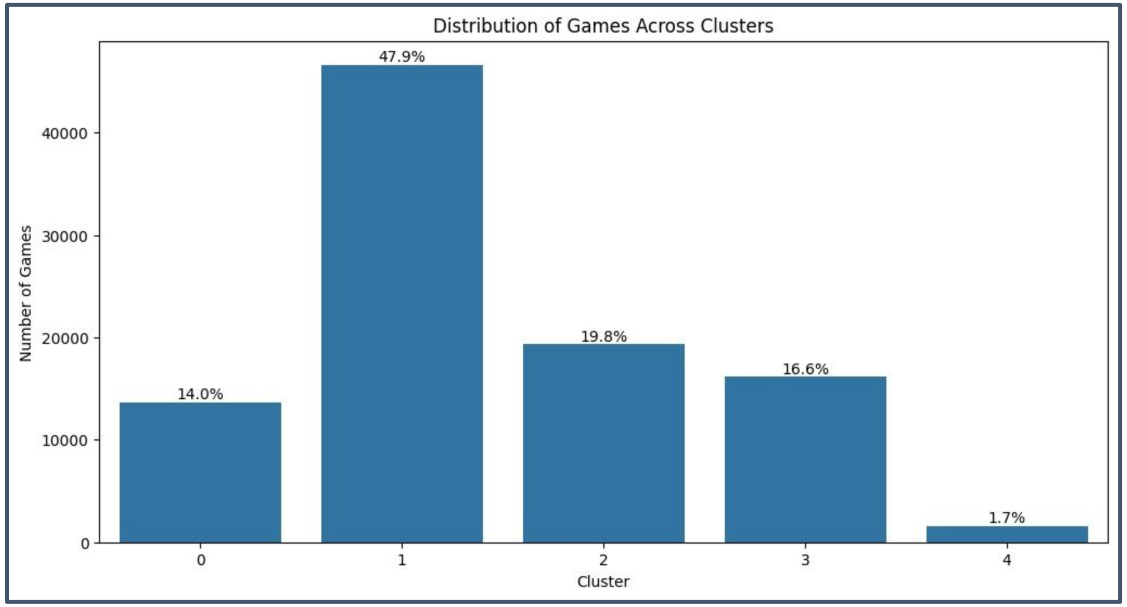
Analysis (Content Based Filtering)

Model Evaluation Metrics:

1. **Elbow Method:**
   - The elbow plot revealed that k = 5 is the optimal number of clusters for K-Means.
   - At k = 5, inertia (within-cluster sum of squares) starts decreasing more slowly, indicating diminishing returns from adding more clusters.
2. **Silhouette Analysis:**
   - Achieved a silhouette score of 0.15 for k = 5.
   - Indicates strong separation between clusters based on game attributes, reflecting well-defined clusters.
3. **Nearest Neighbors Evaluation:**
   - The recommendation system effectively identifies games with similar attributes (genre, platform, price).
   - Results show a strong correlation between recommended games and user preferences, ensuring relevance in suggestions.

Analysis





```
📊 Cluster-wise Silhouette Analysis:
==================================================

Cluster 1.0:
Average Silhouette Score: 0.263
Cluster Size: 46626.0
Quality: Good - Reasonable structure

Cluster 3.0:
Average Silhouette Score: 0.108
Cluster Size: 16156.0
Quality: Fair - Normal structure

Cluster 4.0:
Average Silhouette Score: 0.065
Cluster Size: 1612.0
Quality: Fair - Normal structure

Cluster 2.0:
Average Silhouette Score: 0.043
Cluster Size: 19331.0
Quality: Fair - Weak structure

Cluster 0.0:
Average Silhouette Score: -0.010
Cluster Size: 13679.0
Quality: Poor - Potential
misclassification
```

Analysis

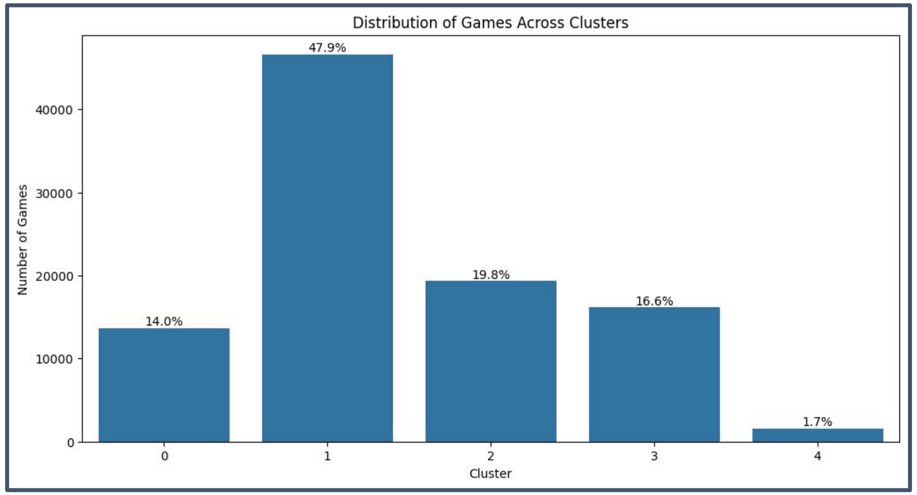
Nearest Neighbors Example - Similar Game Search:

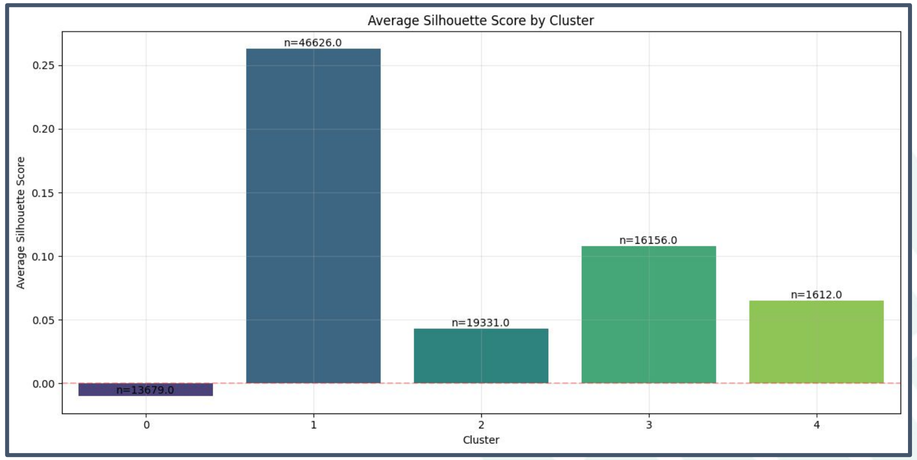
1. **Query Example:** Searching for similar games to "FIFA"
2. **Whoosh Fuzzy Search:**
   - Used fuzzy search to find the top 5 closest game name matches.
   - Displayed as a table with game names and match scores.
3. **Recommendation Generation:**
   - Selected the highest-scoring game from the search results.
   - Generated recommendations based on clustering model and feature similarity.
4. **Feature Matching and Cosine Similarity:**
   - Calculated similarity based on features such as genre, platform, and price.
   - Final recommendations sorted by aggregate score, feature matching score, and cosine similarity score.
5. **Results:**
   - Recommendations provide a list of top similar games.
   - Each recommendation is accompanied by relevant scores and details.
6. **Impact:**
   - Meaningful recommendations are generated even when an exact match is not found in the database, improving user experience.

Analysis

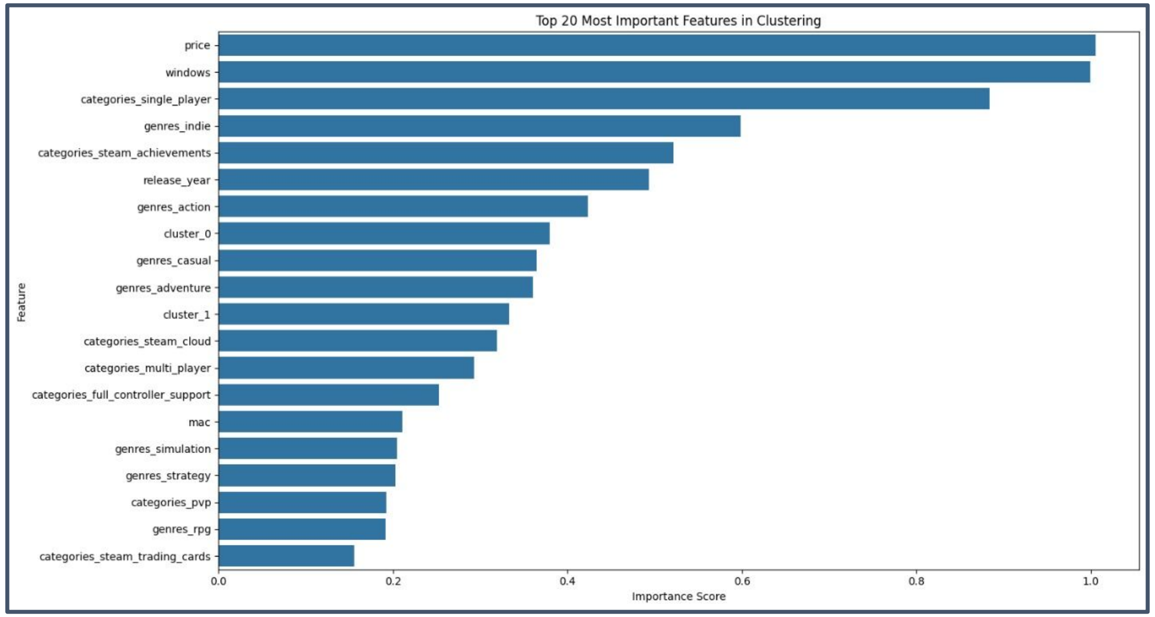
Cluster-wise Game Analysis:

- ➔ Silhouette Score:
  - Achieved a silhouette score of 0.152 and an inertia of 446,447.28, indicating moderate cluster separation.
- ➔ Cluster Insights:
  - Cluster analysis revealed distinct groupings of games based on key features.
- ➔ Cluster 0:
  - Number of games: 13,679
  - Common Features:
    - a. High prevalence of multiplayer and pvp games.
    - b. Key attributes: Windows-based, indie games, with an average price near zero.
- ➔ Cluster 1:
  - Number of games: 46,626
  - Common Features:
    - a. Majority are single-player, indie games.
    - b. Windows-based, with an average price slightly negative (due to promotions/free games).
- ➔ Cluster 2:
  - Number of games: 19,331
  - Common Features:
    - a. Predominantly single-player, indie games with Steam achievements.
    - b. Windows-based with an average price near zero.
- ➔ Cluster 3:
  - Number of games: 16,156
  - Common Features:
    - a. Almost all are single-player games, with Steam achievements.
    - b. Windows-based, indie titles, higher presence of cluster 1 features.
- ➔ Cluster 4:
  - Number of games: 1,612
  - Common Features:
    - a. Higher-priced games (~$4.66), Windows-based with Steam achievements.
    - b. Smaller group, primarily focusing on premium content and more features like Steam cloud.





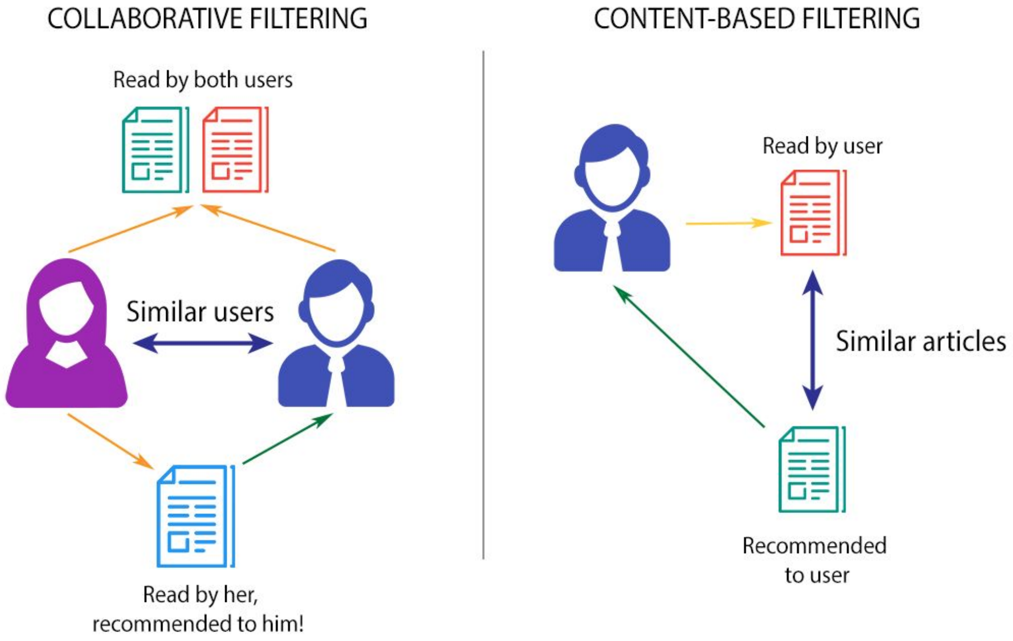
Analysis



- ➔ **Distinct Clusters:** Each cluster has a well-defined set of attributes that differentiate it from the others, e.g., pricing, single-player/multiplayer focus, and platform.
- ➔ **Pricing Trend:** Clusters with higher pricing (like Cluster 4) tend to have more premium features and are smaller in size compared to free/low-cost game clusters.
- ➔ **Platform Consistency:** Across all clusters, games are predominantly available on Windows.
- ➔ **Genre Focus:** Most clusters exhibit a high proportion of indie games, indicating their popularity across multiple feature sets.

Collaborative Filtering

We implemented an Item-Based Collaborative Filtering model to recommend games based on user behavior, utilizing a user-item matrix and cosine similarity to identify similar users and aggregate their recommendations for tailored suggestions.



Dataset Description (Collaborative Filtering)

The dataset consists of three main entities:

1. **games.csv** - a table of games (or add-ons) information on ratings, pricing in US dollars $, release date, etc. A piece of extra non-tabular details on games, such as descriptions and tags, is in a metadata file
2. **users.csv -** a table of user profiles' public information: the number of purchased products and reviews published
3. **recommendations.csv -** a table of user reviews: whether the user recommends a product. The table represents a many-many relation between a game entity and a user entity.

The dataset does not contain any personal information about users on a Steam Platform. A preprocessing pipeline anonymized all user IDs. All collected data is accessible to a member of the general public.

Recommendations and Users merged:

app_id	funny	hours	date	helpful	is_recommended	review_id	user_id	products	reviews
Object	Int	Int	Object	Int	Boolean	Object	Object	Int	Int
Unique Identifier for the apps on SteamGame platform	The number of reviews that are funny	Number of hours spent by the reviewer	Date of review	The number of reviews that are helpful	Whether the game is recommended by the reviewer	Unique identifier for the review	Unique identifier for the user	Number of the products purchased by the user	Number of reviews given by the user

Dataset Description (Collaborative Filtering)

Games and Recommendations merged:

title	date_release	win	mac	linux	positive_ratio	discount	price_final	price_original	rating
Object	Object	Boolean	Boolean	Boolean	Float	Float	Float	Float	Object
Name of the game	Date when the game was released	If the game is compatible with windows OS	If the game is compatible with mac OS	If the game is compatible with linux OS	Positive reviews divided by total reviews	Discount offered for a game	Price after the discount applied	Price before the discount	9 categories of ratings

steam_deck	user_reviews	funny	hours	date	helpful	is_recommended	review_id	user_id
Boolean	Int	Int	Int	Object	Int	Boolean	Object	Object
If the game is available on steam's handheld gaming device called steam deck	Total number of reviews	The number of reviews that are funny	Number of hours spent by the user	Date of review	The number of reviews that are helpful	Whether the game is recommended by the user	Unique identifier for the review	Unique identifier for the user

To facilitate analysis, we intend to merge the three original tables in pairs. The above provides a comprehensive description of the merged tables and their respective columns.

Dataset Description (Collaborative Filtering)

Preprocessing Requirements:

- **Initial Dataset Overview** (games, users, recommendations)**:**
  - Total Entries: 50,796, 14306064, 41154794
  - Columns: 12, 3, 8
  - Data Types:
    - games.csv: 3 integers, 3 objects, 4 boolean, and 3 floats.
    - users.csv: 3 integers.
    - recommendations.csv: 5 integers, 1 objects, 1 boolean, and 1 floats.

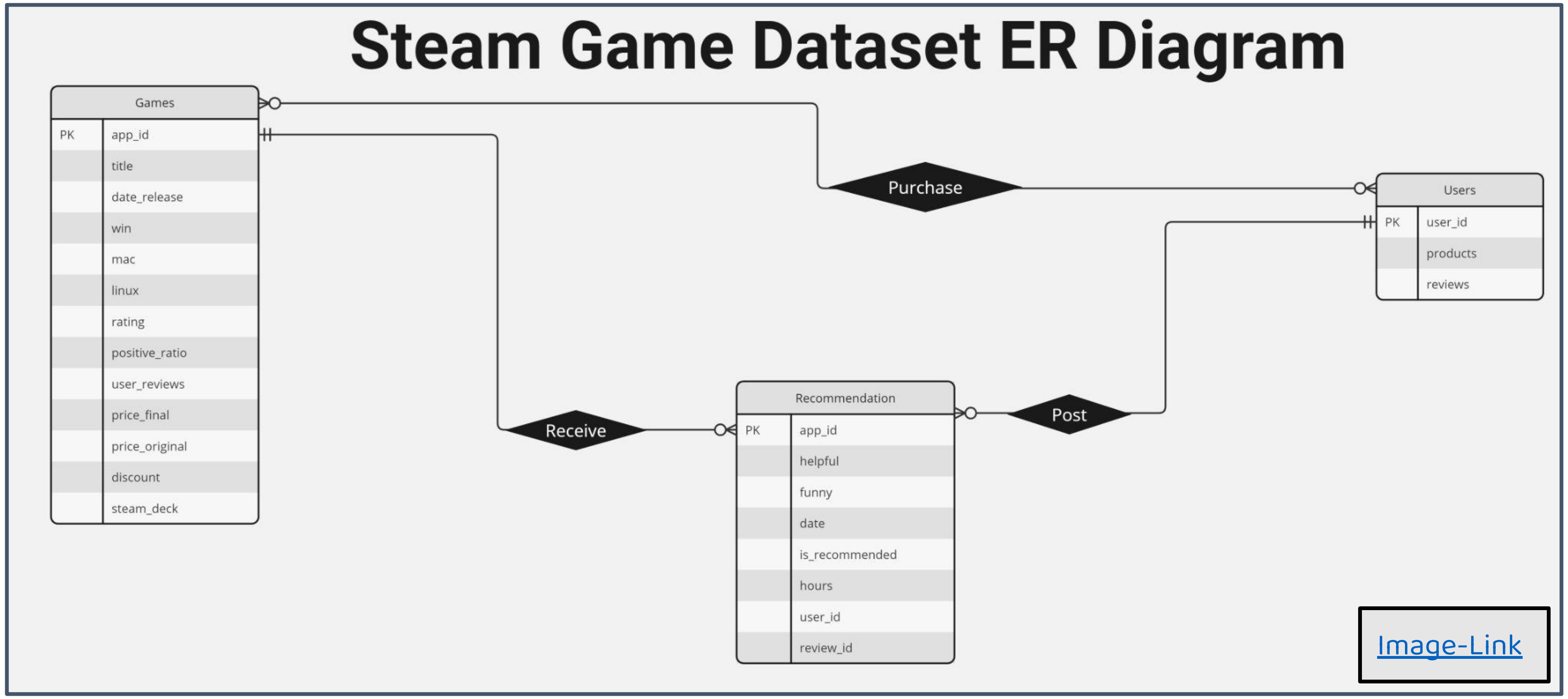
- **Data Type Conversions:**
  - Convert 'object' columns to 'string' type.
  - Convert date_release column to a datetime datatype.

- **Data Quality Checks:**
  - All columns in the dataset have zero missing values, indicating that the data is complete and ready for analysis.

Dataset Description (Collaborative Filtering)

ER Diagram:

- **Games to Recommendations:** Optional-Many, a game can have zero or many recommendations; a recommendation is linked to one game.
- **Recommendations to Users:** Mandatory-One, a recommendation is associated with one user; a user can have zero or many recommendations.
- **Games to Users:** Optional-Many, a game can have zero or many users; a user can purchase zero or many games.



Steam Game Dataset ER Diagram

Games
PK app_id
title
date_release
win
mac
linux
rating
positive_ratio
user_reviews
price_final
price_original
discount
steam_deck

Purchase

Users
PK user_id
products
reviews

Receive

Recommendation
PK app_id
helpful
funny
date
is_recommended
hours
user_id
review_id

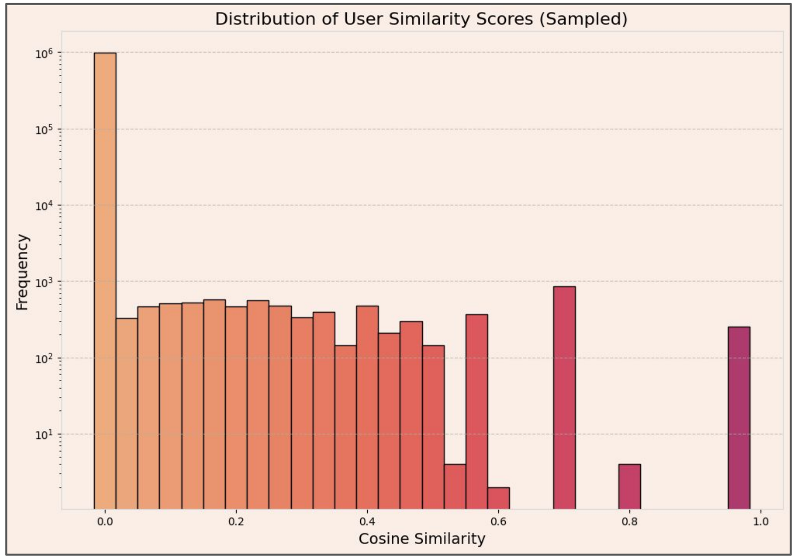
Post

Image-Link

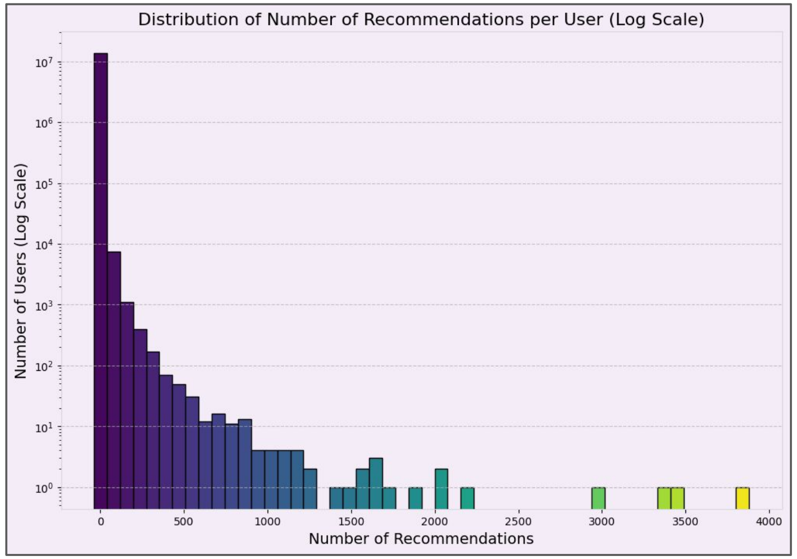
Methodology (Collaborative Filtering)

- Creating User-Item Matrix:
  - Group the reviews DataFrame by user_id and app_id and take the maximum recommendation status (is_recommended).
  - Convert the Dask DataFrame into a Pandas DataFrame.
- Sparse Matrix:
  - Convert the user-item matrix into a sparse matrix using coo_matrix.
  - For further analysis, convert this sparse matrix into CSR format for efficient memory usage.
- User Similarity:
  - Compute cosine similarity between users based on their game recommendations.
  - Identify similar users for a given user_id.
- Game Recommendations:
  - Recommend games based on what similar users have recommended.
  - Collect recommendations from these users and return the top recommended games.
- Visualization:
  - Create visualizations such as heatmaps for user-item matrices and user similarity matrices.
  - Plot the top 10 most recommended games with a bar chart and a custom color palette.

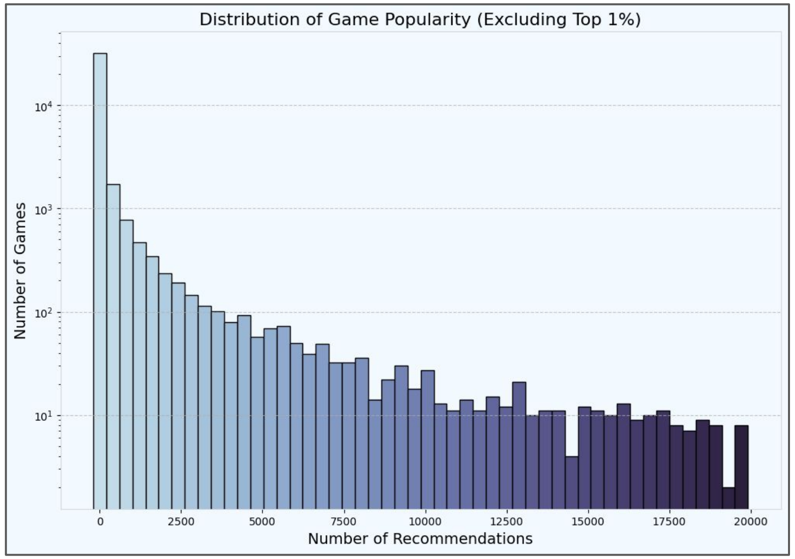
Analysis (Collaborative Filtering)



1. Right-skewed distribution: Most user pairs have low similarity scores.
2. Mode around 0: Many pairs have very low similarity.
3. Long tail: A few pairs have very high similarity.

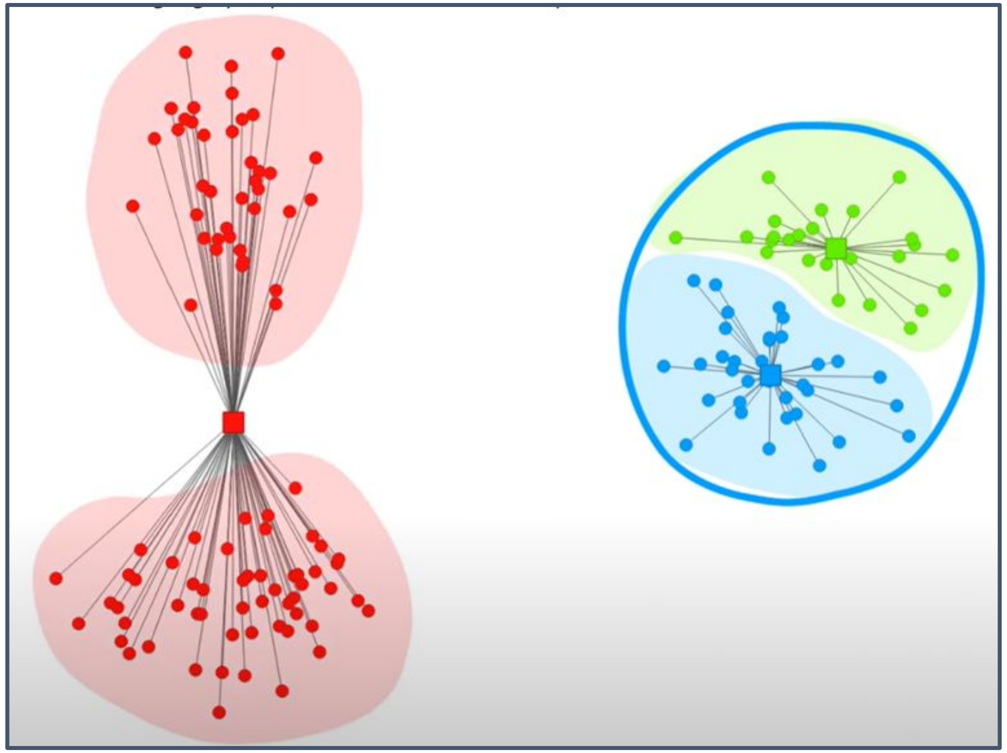


1. Long right tail: A small number of users receive many recommendations.
2. Mode around 0: Many users receive few recommendations.



1. Long right tail: A small number of games receive many recommendations.
2. Rapid decline: Most games receive few recommendations.

Challenges with Random Initialization



Showing Clustering After Random Init of Centroids

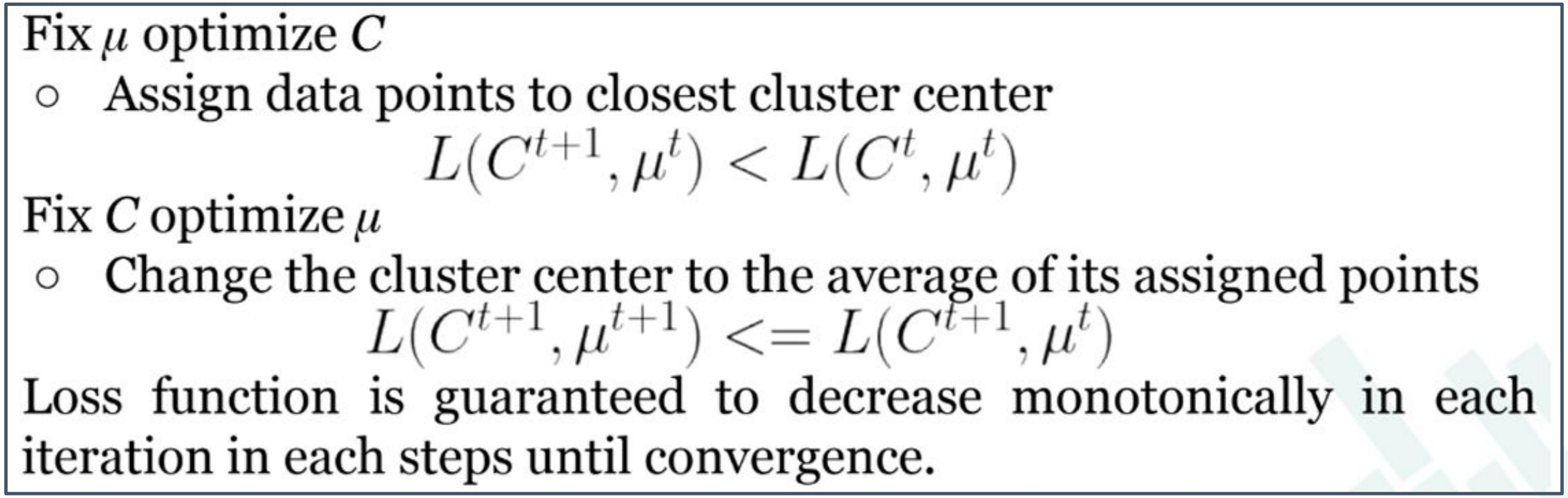
Random centroid initialization when doing clustering has some shortcomings.

Non-Convex Optimization Problem

- Multiple local minima exist
- Final clustering highly dependent on initial centroid positions
- May lead to
  - **Splitting** of a single cluster
  - **Merging** of two cluster

Challenges with Random Initialization

Arriving at global minima through random initialization is not guaranteed, and in most cases, it is **highly unlikely.**



Fix μ optimize C

- ○ Assign data points to closest cluster center

$$L(C^{t+1}, \mu^t) < L(C^t, \mu^t)$$

Fix C optimize μ

- ○ Change the cluster center to the average of its assigned points

$$L(C^{t+1}, \mu^{t+1}) <= L(C^{t+1}, \mu^t)$$

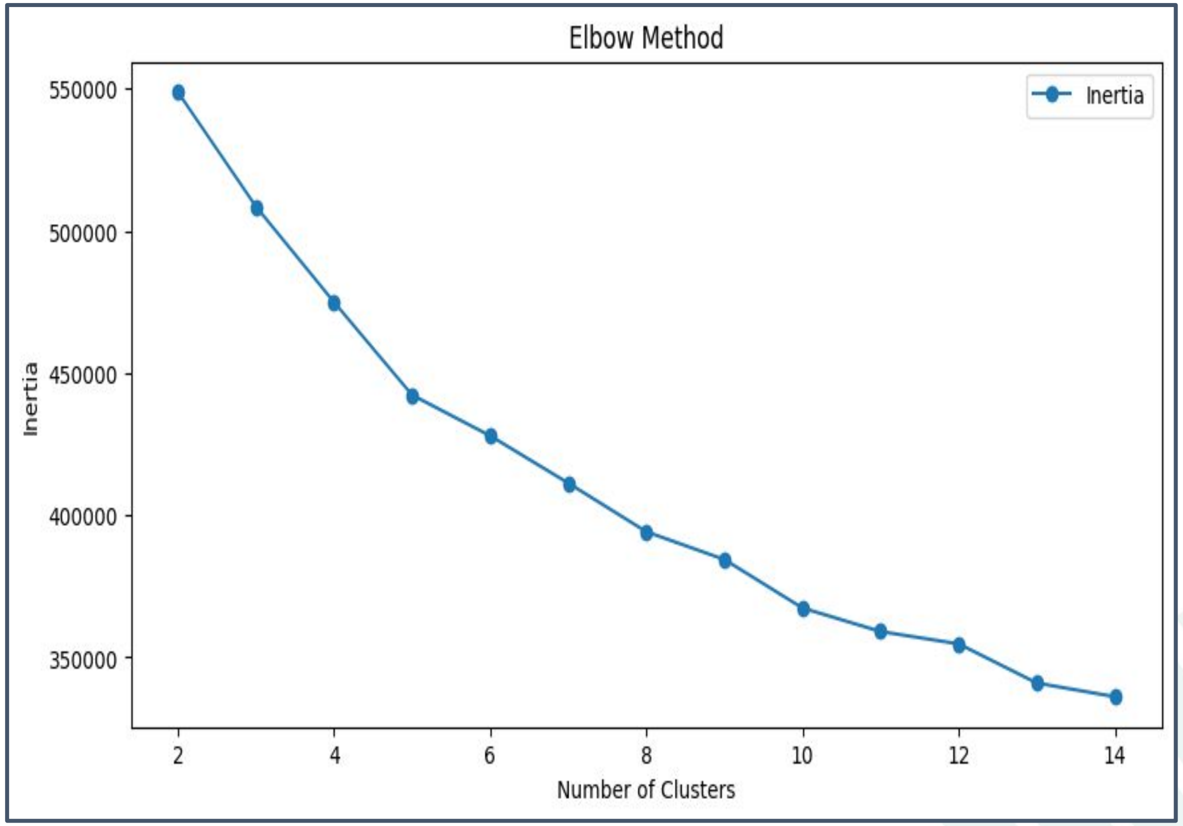
Loss function is guaranteed to decrease monotonically in each iteration in each steps until convergence.

K++ Means Clustering

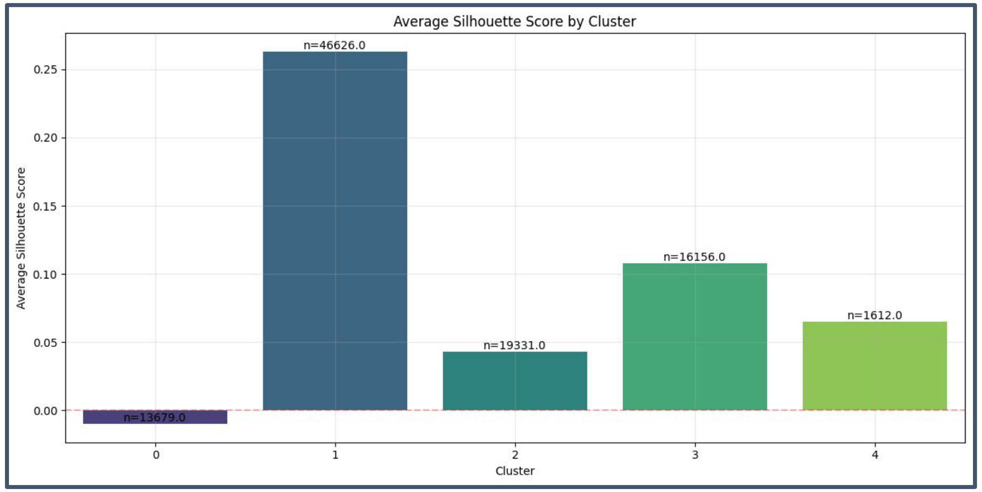
Taking Different Approach **K++ Means Clustering**

1. Select the first centroid randomly from the data points.
2. For each remaining point, compute the distance to the nearest selected centroid.
3. Select the next centroid from the data points with a probability proportional to the squared distance.
4. Repeat until **k** centroids are chosen.

Choosing **Correct** Value of K (i.e number of clusters) using Elbow Method

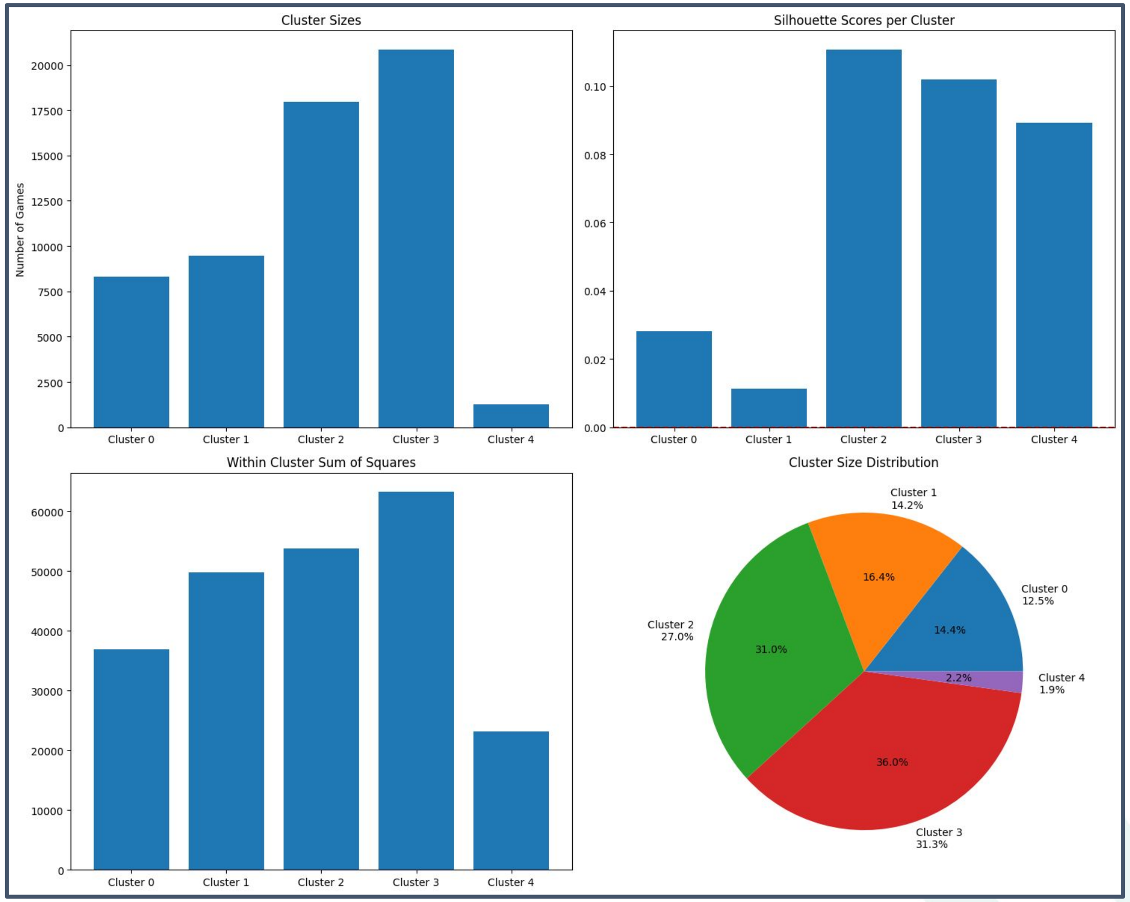


Comparing With Previous Results



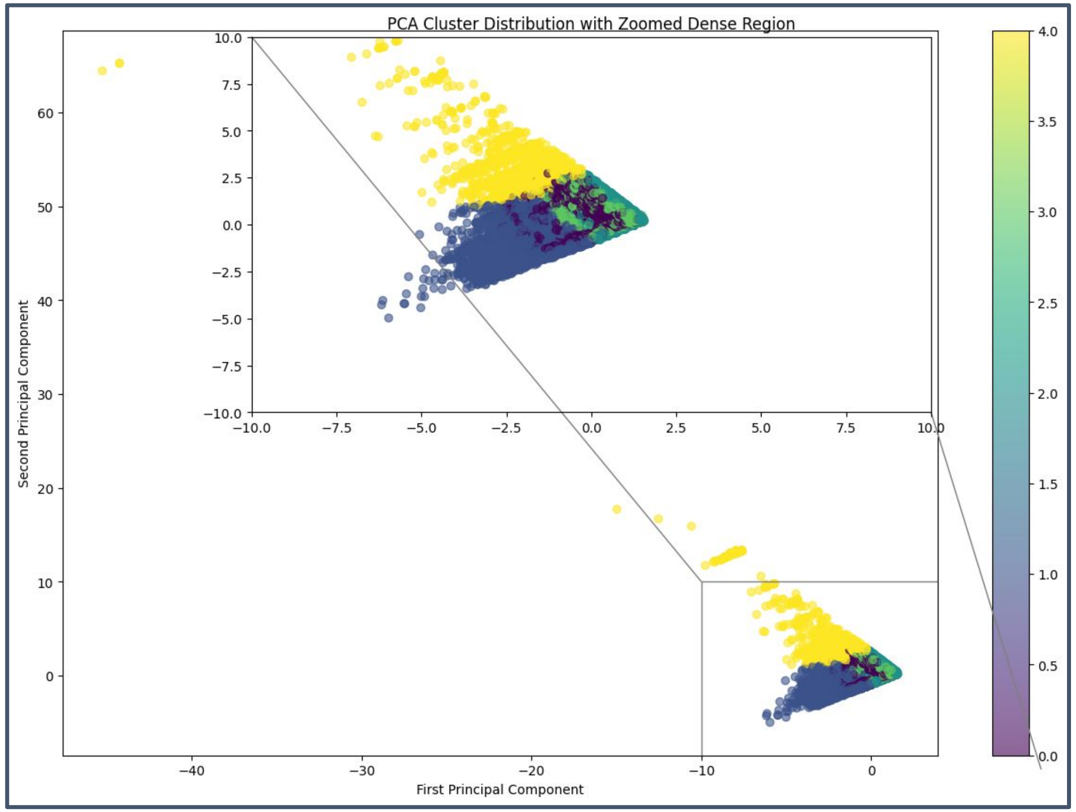
Previously We were experiencing low inter cluster similarity in some clusters. Now we were able to improve

- **Better Silhouette Score** for every cluster
- **Better distribution** of games within each cluster



New Model Performance Metrics

Distance Metric For Clustering



Scatter Plot of Data after **PCA**

The choice of distance metric depends on the nature of the data and the clustering algorithm being used. Some options we had were:

1. **Euclidean Distance**

   $$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

2. Manhattan Distance (L1 norm)

   $$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

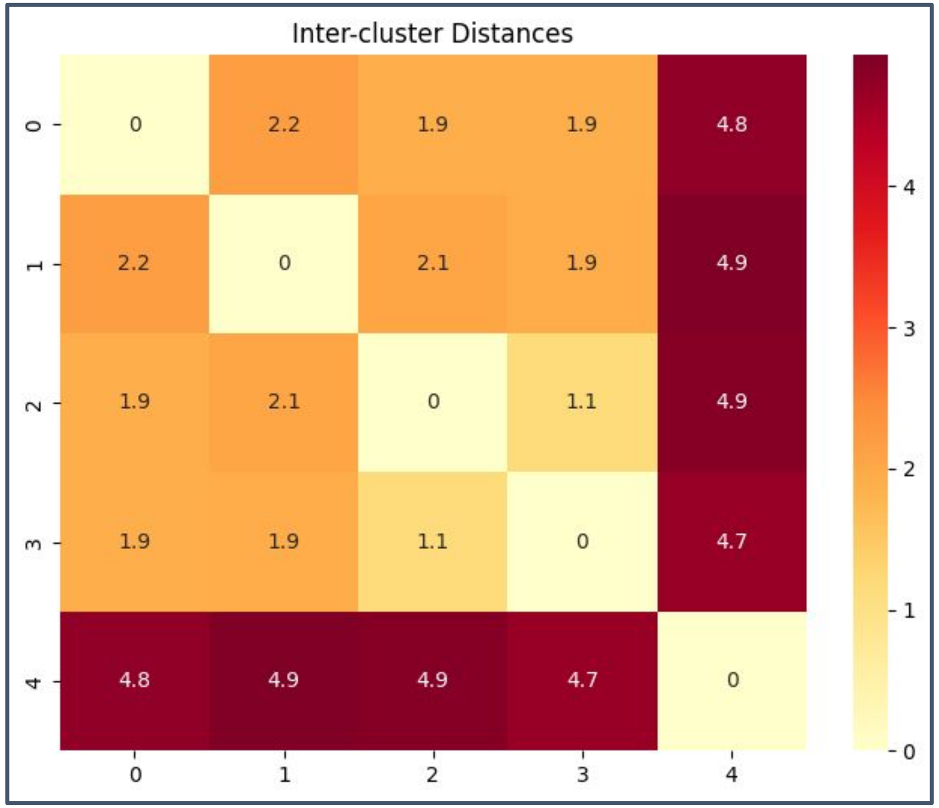
3. Cosine Distance

   $$1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}$$

4. Mahalanobis Distance

   $$d(x,y) = \sqrt{(x-y)^T S^{-1} (x-y)}$$

Best Distance Metric For Clustering



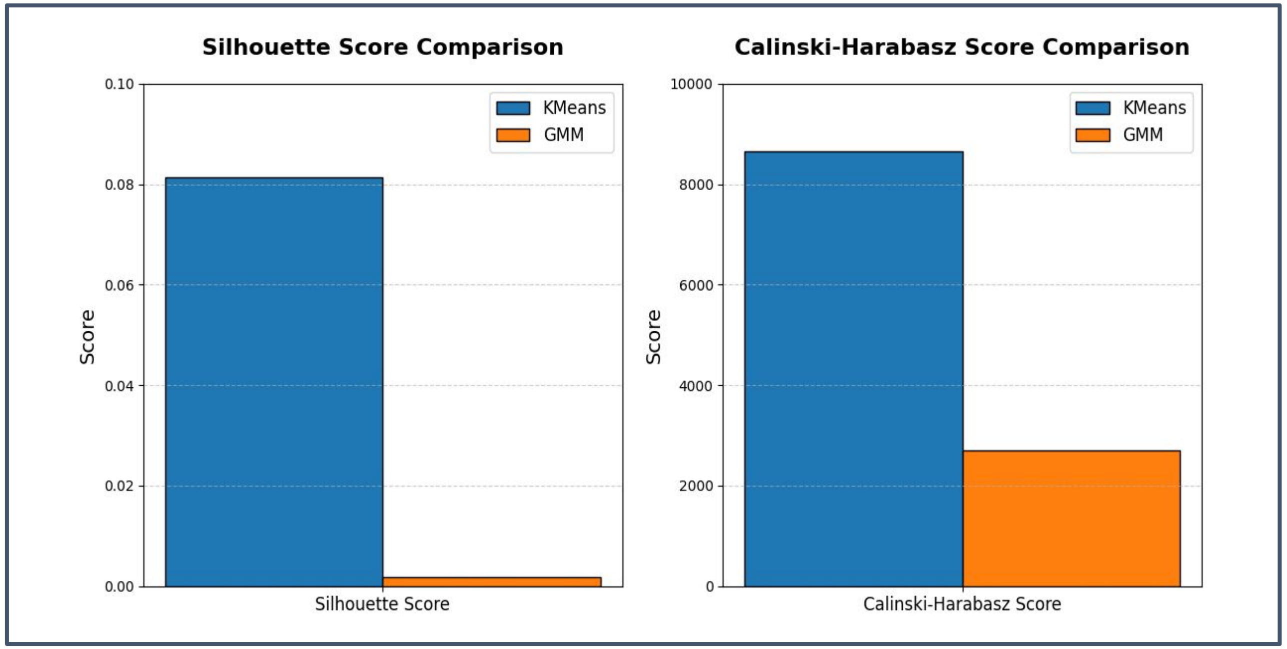
	0	1	2	3	4
0	0	2.2	1.9	1.9	4.8
1	2.2	0	2.1	1.9	4.9
2	1.9	2.1	0	1.1	4.9
3	1.9	1.9	1.1	0	4.7
4	4.8	4.9	4.9	4.7	0

As, the Data seem to be made of **circular clusters that are close together**, instead of following some explicit pattern corresponding to covariance matrix.

Clusters shaped like crescents or concentric circles which is best suited for **Euclidean Distance**

Inter-Cluster Distances After K++ means Clustering. **High Values** correspond to **better separated clusters**.

Comparing with GMM(Gaussian Models)

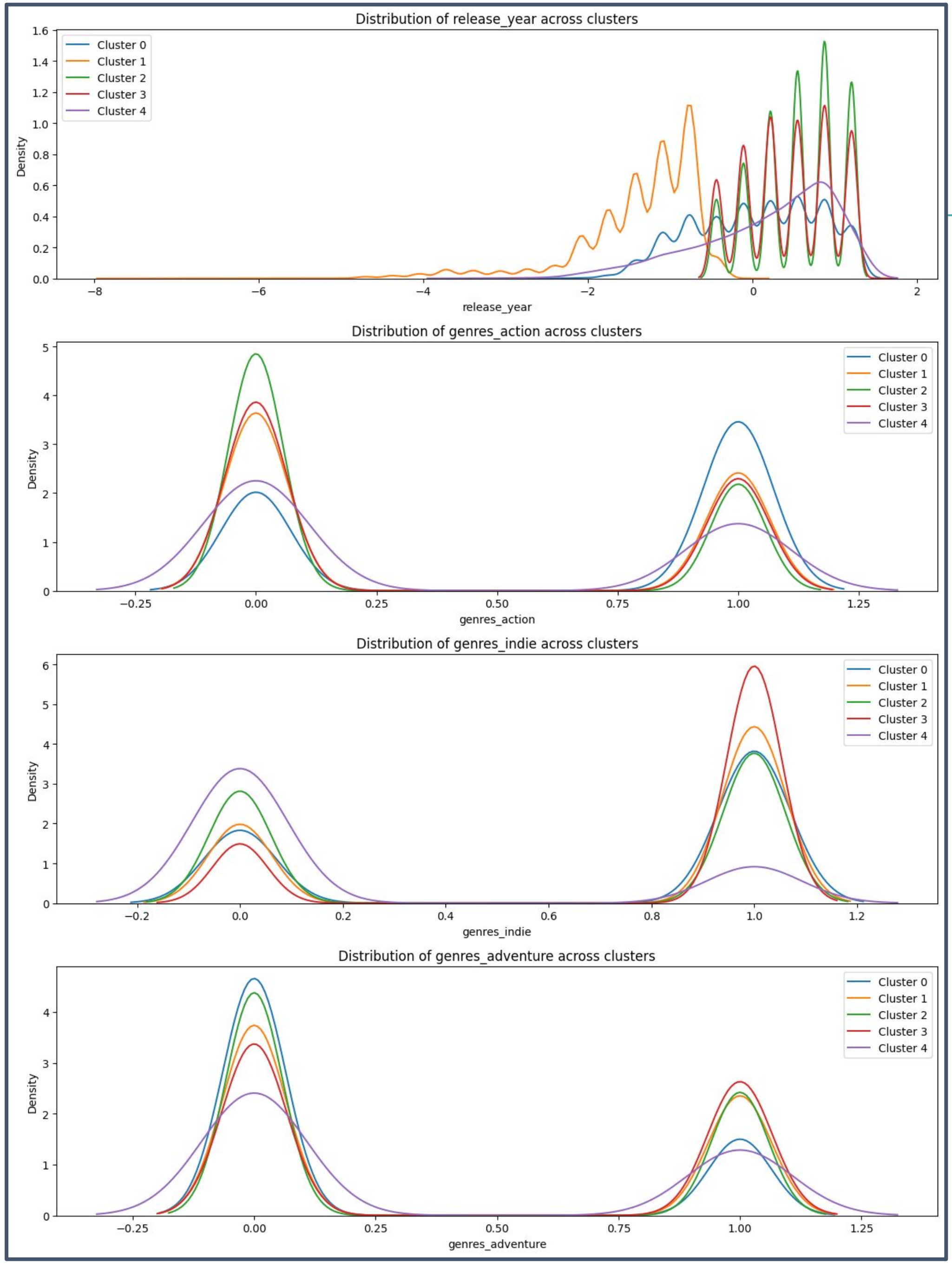


GMM v/s K means++

1. Hard v/s Soft Assignment, GMM models allows for multiple assignment in clusters. i.e partial membership.
2. We noticed that **K means performed much better than GMM** in both Silhouette Score and Calinski-Harabasz Score

This could mean clusters in the dataset **do not** follow this assumption (e.g., they are non-elliptical, irregular, or arbitrary shapes), GMM may struggle.

Some Analysis about Model



The Graphs on left show, shows distribution of values of different attributes, among different clusters of the model.

We see that attributes like release_year, genere_action, genere_indie, genere_adventure are distributed well among the clusters.

Silhouette v/s Cluster Size Analysis

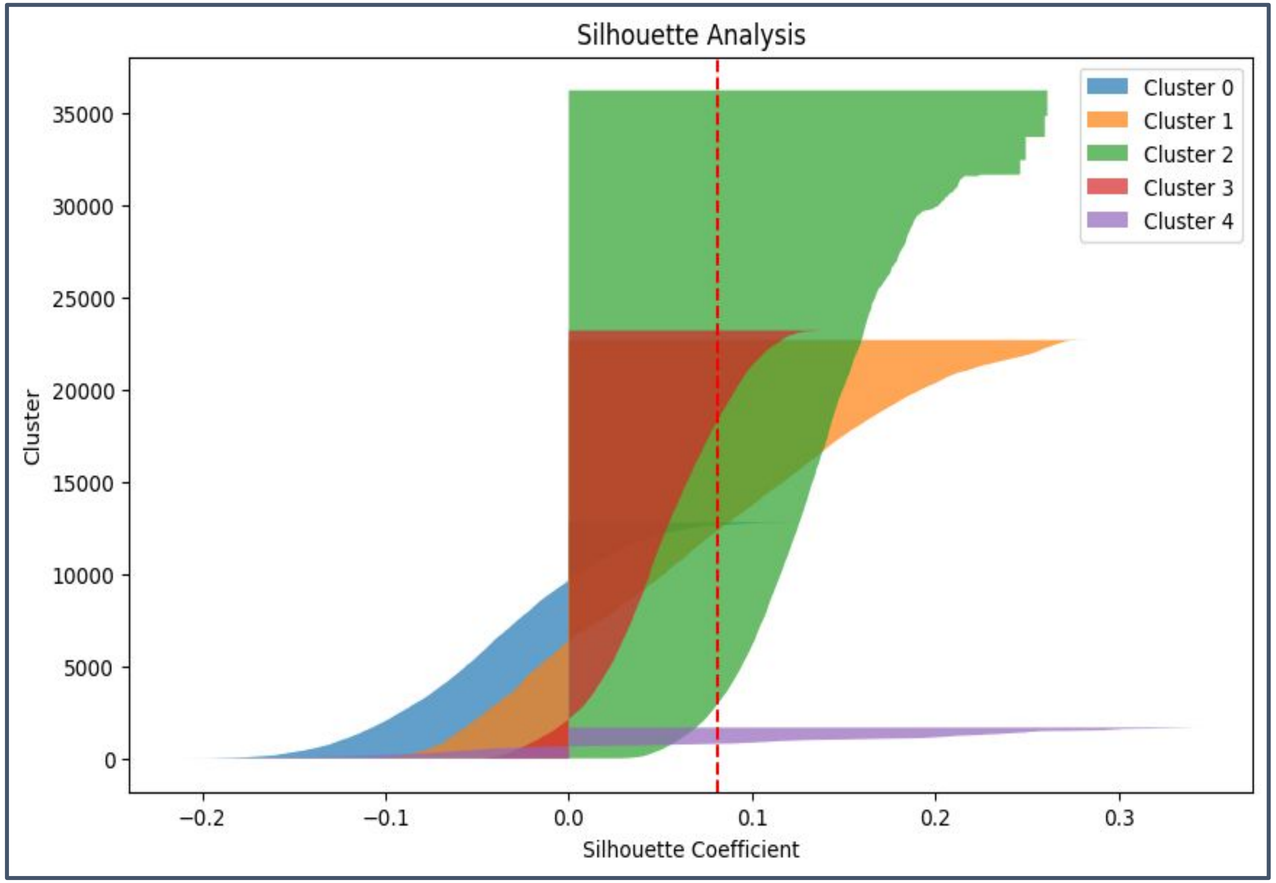
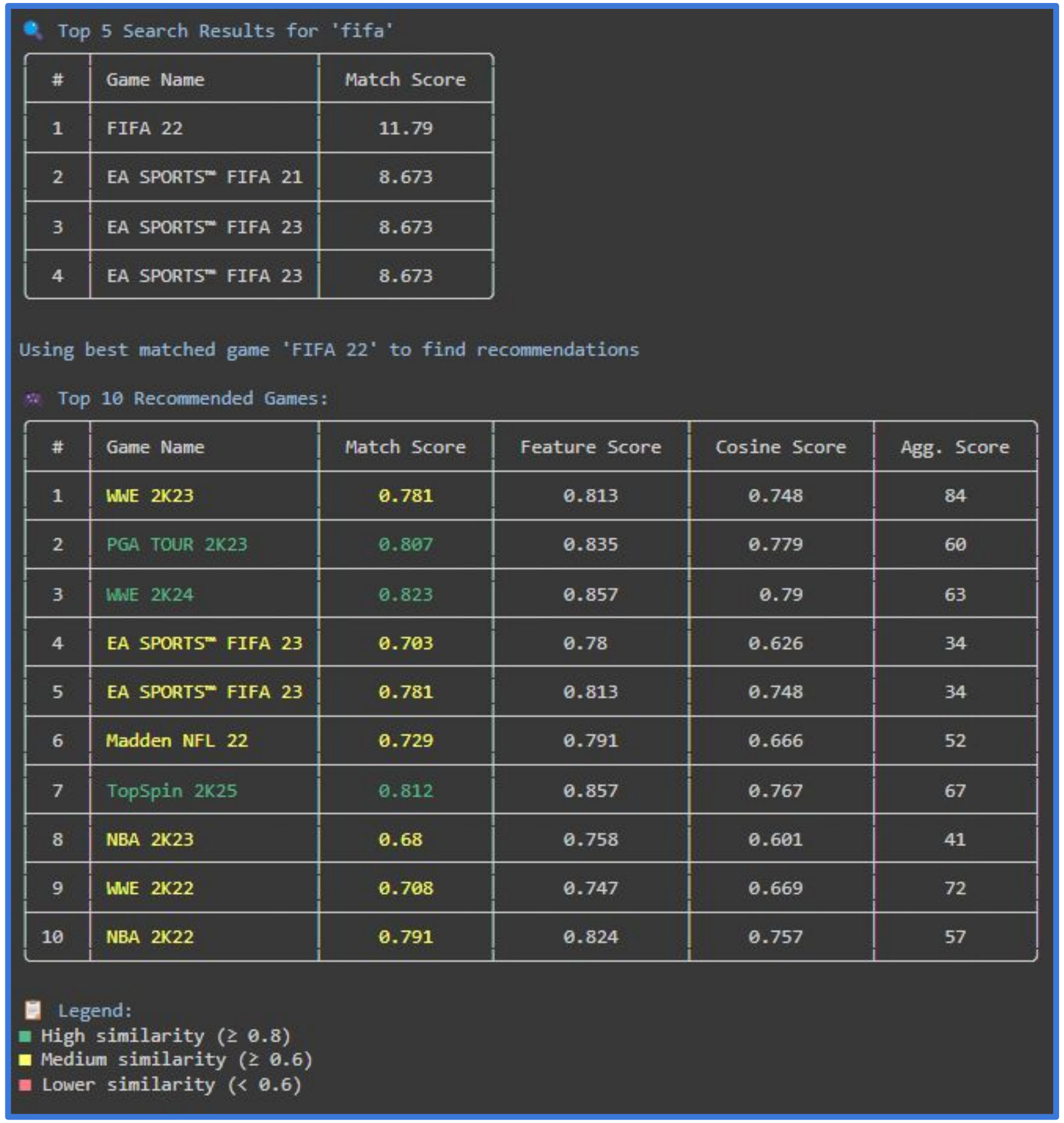


Image of Left show, change in Silhouette coefficient as we increase the number of games in the cluster.

We see that all clusters **achieve positive Silhouette coefficient** after all games have been added.

Analysis



```
🔍 Top 5 Search Results for 'fifa'
┌────┬───────────────────┬─────────────┐
│ #  │ Game Name         │ Match Score │
├────┼───────────────────┼─────────────┤
│ 1  │ FIFA 22           │    11.79    │
├────┼───────────────────┼─────────────┤
│ 2  │ EA SPORTS™ FIFA 21│    8.673    │
├────┼───────────────────┼─────────────┤
│ 3  │ EA SPORTS™ FIFA 23│    8.673    │
├────┼───────────────────┼─────────────┤
│ 4  │ EA SPORTS™ FIFA 23│    8.673    │
└────┴───────────────────┴─────────────┘

Using best matched game 'FIFA 22' to find recommendations

🎮 Top 10 Recommended Games:
┌────┬───────────────────┬─────────────┬───────────────┬──────────────┬────────────┐
│ #  │ Game Name         │ Match Score │ Feature Score │ Cosine Score │ Agg. Score │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 1  │ WWE 2K23          │    0.781    │     0.813     │    0.748     │     84     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 2  │ PGA TOUR 2K23     │    0.807    │     0.835     │    0.779     │     60     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 3  │ WWE 2K24          │    0.823    │     0.857     │     0.79     │     63     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 4  │ EA SPORTS™ FIFA 23│    0.703    │     0.78      │    0.626     │     34     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 5  │ EA SPORTS™ FIFA 23│    0.781    │     0.813     │    0.748     │     34     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 6  │ Madden NFL 22     │    0.729    │     0.791     │    0.666     │     52     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 7  │ TopSpin 2K25      │    0.812    │     0.857     │    0.767     │     67     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 8  │ NBA 2K23          │    0.68     │     0.758     │    0.601     │     41     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 9  │ WWE 2K22          │    0.708    │     0.747     │    0.669     │     72     │
├────┼───────────────────┼─────────────┼───────────────┼──────────────┼────────────┤
│ 10 │ NBA 2K22          │    0.791    │     0.824     │    0.757     │     57     │
└────┴───────────────────┴─────────────┴───────────────┴──────────────┴────────────┘

📋 Legend:
🟩 High similarity (≥ 0.8)
🟨 Medium similarity (≥ 0.6)
🟥 Lower similarity (< 0.6)
```

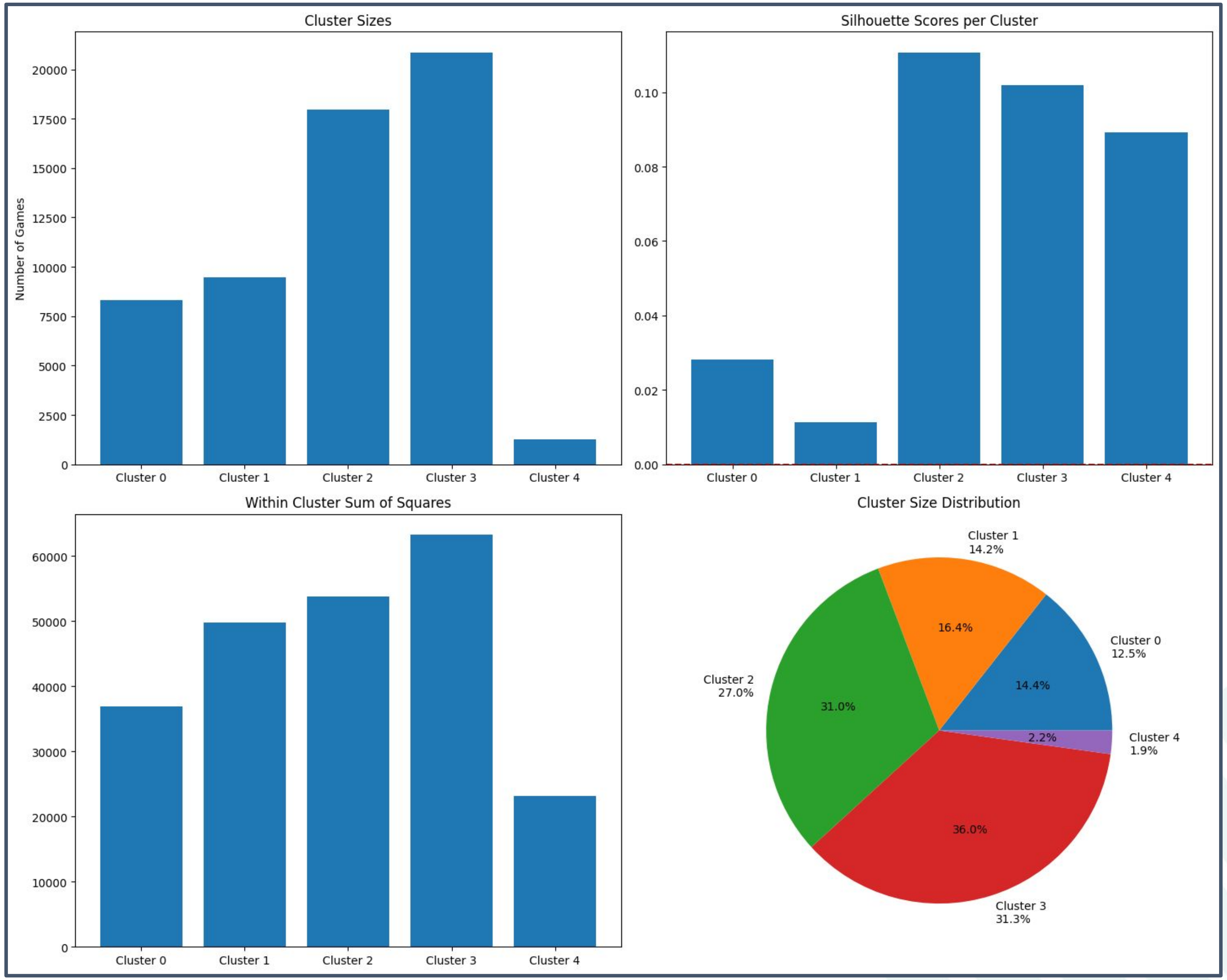
Example Run: For the query "fifa"

1. we use **Whoosh's** fuzzy search to find the closest matching game names and display the top 5 matches with their scores in a table.
2. The highest-scoring game is selected, and we generate a list of recommended games based on
   a. feature score
   b. co-sine similarity scores,
   c. Match Score : Combination of (feature score + cosine score)
   d. sorted by aggregate score (if available).
3. This process enables us to provide meaningful recommendations even when an exact match is not found in the database.
4. The top 10 Recommended games are printed in a tabulated form.

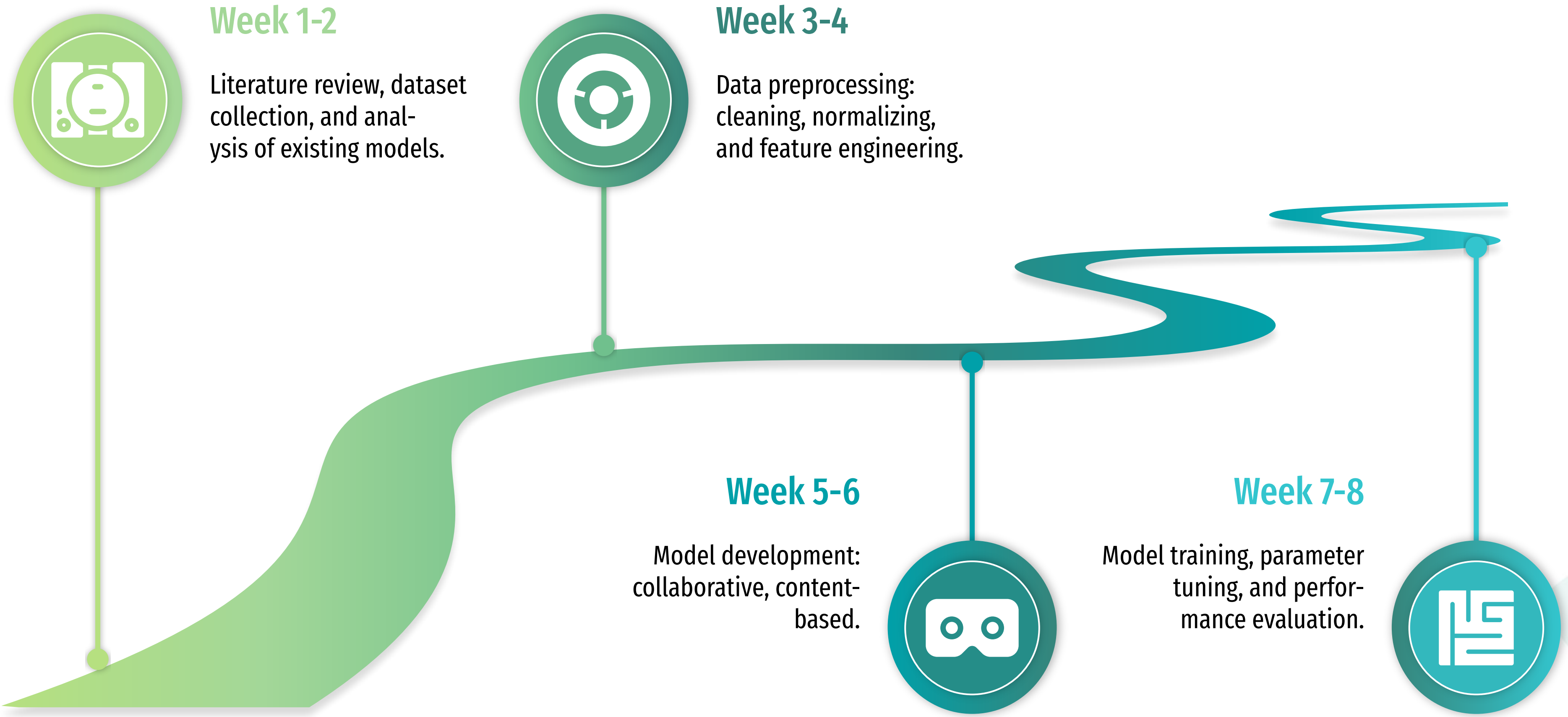
Analysis

- **Nearest Neighbors Evaluation:** The recommendation system successfully identified games with similar genres, platforms, and price ranges, showing a strong alignment between the recommended games and user preferences. As shown in Fig. we can see distribution of games cluster wise. This plot Visualizes how games are distributed across clusters.

- **Observations and Insights:**
  - **Silhouette Scores:** Cluster 2's higher silhouette score reflects well-defined groupings, whereas Cluster 1's low score indicates possible overlap with other clusters.
  - **Within-Cluster Sum of Squares:** Higher within-cluster variance in Cluster 3 suggests potential for further feature refinement or better separation



Timeline



Week 1-2

Literature review, dataset collection, and analysis of existing models.

Week 3-4

Data preprocessing: cleaning, normalizing, and feature engineering.

Week 5-6

Model development: collaborative, content-based.

Week 7-8

Model training, parameter tuning, and performance evaluation.

Individual Member Contributions

All team members contributed equally to the project, assisting one another with code revisions and the writing of this report. Each member also sourced various research papers and datasets. The individual contributions outlined below represent the task assignments for each team member:

- **Aditya Sharma(2022038):** Data preprocessing, Feature engineering, clustering analysis, model development, EDA, SVD, fuzzy search.
- **Kanishk Kumar Meena(2022233):** Data cleaning(user-recommendation data),Collaborative Filtering, Model evaluation.
- **Vansh Agarwal(2022558):** Dataset management and cleaning(Games data), visualization, EDA,Performance Testing

THANK YOU!