A recommendation system based on PCA and K-means Clustering for Steam Video Games

Robert Officer
U20431122
University of Pretoria
Pretoria, South Africa

ABSTRACT

This study presents a recommendation system for video games on extracted from Steam, leveraging Principal Component Analysis (PCA) and K-Means clustering. The system integrates two datasets containing user behaviour and game attributes, further enriching them with inferred features such as average playtime and engagement metrics. Dimensionality reduction via PCA simplifies data complexity by finding latent features in the dataset and thereafter clustering users and games into behavioural and feature-based groups. Recommendations are personalized by identifying games from clusters with high engagement metrics, enhancing user satisfaction. Experimental results demonstrate the system's potential to improve user-game matching, overcoming limitations of the currently implemented Steam interactive recommender.

1. INTRODUCTION

E-commerce systems are those in which the purchase and marketing of goods and/or services is achieved over the internet [5; 6]. Recommendation systems are systems designed to offer suggestions to goods and/or services that prospective customers would be interested in. The main benefits of recommendation systems is in guiding users toward content, products, or services they might find engaging thereby increasing satisfaction of the user experience and and therefore increasing the likely hood or returning customers [8]. Steam is considered to be the largest platform for e-commerce, holding over 34,000 items for sale. The current Steam recommendation system works off of play patterns of each user, looking at the tags of the games the user has played and suggesting games with similar tags. However, this system may overlook deeper behavioral patterns and relationships between users and games. This project leverages advanced techniques such as PCA and K-Means clustering to build a robust, scalable recommendation system. The primary goal is to recommend games that players in the same behavioral cluster have played extensively, thereby personalizing recommendations while considering user-game dynamics.

1.1 Problem statement

The rapid growth of e-commerce platforms has resulted in an overwhelming amount of accessible items and services to be purchased. This increases the challenge for online users in identifying items they would enjoy and is particularly relevant in the gaming industry. The current recommendation system of Steam fails to capture deeper insights that can be provided by the relationships between similar games and users.

This paper aims to address this challenge by making use of inferred user and game features, dimensionality reduction and clustering techniques. By grouping users and games based on behavioural and attribute patterns, the recommender system can find patterns

2. LITERATURE REVIEW

2.1 Recommendation systems

Recommendation systems are well-established, with three categorisations: collaborative filtering, content-based filtering, and hybrid methods. This section briefly describes each category.

2.1.1 Collaborative filtering

Collaborative Filtering (CF) is a widely used method in recommender systems that makes relies on the preferences and behaviours of other users to make recommendations. Neighbourhood-based CF identifies similarities between users or items, recommending items liked by similar users or those similar to the target user's previous choices. Model-based CF uses machine learning techniques, such as matrix factorization, to uncover latent patterns in user-item interactions, offering better scalability and performance. These techniques often exploit information provided by user ratings and item tagging [10]. Unfortunately, collaborative filtering struggles in a number. One major fault is the reliance of a number users in order to perform recommendations. This is commonly known as the cold-start problem, whereby systems need data to provide the service but have not yet received enough.

2.1.2 Content-based filtering

Content-based filtering recommends items to users based on the attributes of the items and the information on the user's profile. Unlike collaborative filtering, it does not rely on the behaviour of other users. Instead, it relies heavily on metadata of both items and the target user such as genre or keywords of the item and past item interactions. However, it faces challenges like overspecialization, where users only receive recommendations similar to what they already know, and difficulty addressing cold-start problems for new users

or items. Despite these limitations, CBF is effective when detailed item information is available, making it suitable for domains with well-structured metadata.[10]

2.1.3 Hybrid methods

Hybrid approaches combine multiple recommendation systems together and attempt to formulate recommendations by combing the results gathered from the different systems implemented. Hybrid systems have also been shown to address a number of shortcomings of both CF and CBF systems by exploiting the advantages of the multiple methods utilised.

2.2 Clustering Algorithms

A major role of data analysis is to be able to understand the relationships that occur throughout a dataset. Clustering algorithms are able to provide a method in finding these relationships using similarity measurements. This unsupervised technique's goal is to ensure that objects within a cluster are highly similar while objects in different clusters are distinct. Some common clustering approaches include, but are not limited to:

- Hierarchical clustering: Tree-like structures are created to represent data relationships
- Partitional clustering: Divisions in the data are created to form non-overlapping groups.
- Graph based clustering: Extends current clustering techniques but maps them to specific data structures and patterns
- Density based clustering: Identifies clusters based on denser regions in distributions of the data.

Each algorithm comes with its strengths, limitations, and applications, often requiring a tailored approach based on the dataset and problem at hand [13].

2.2.1 K-means clustering

K-means is one of the most widely used clustering algorithms, designed for partitional clustering. In an iterative process, a dataset is partitioned into a pre-defined number of clusters by minimizing intra-cluster variance. Initial cluster centroids are selected either randomly or using an optimised technique (i.e. Mean, median and mode centroid [2]), thereafter each each data point is assigned to the nearest centroid. The centroids are then recalculated as the mean of the points in each cluster, and the process repeats until minimal change between the data points occurs in the centroids. Despite its simplicity and efficiency, K-means assumes clusters are spherical and of similar size, making it less effective for datasets with irregularly shaped or imbalanced clusters. It is computationally efficient but sensitive to initial centroid placement and outliers. Some variations of K-means clustering such as K-medoids and fuzzy K-means, aim to improve upon k-means whilst addressing some of these challenges. [13].

2.3 Pricipal component analysis

Principal Component Analysis (PCA) is a multivariate statistical technique that has become widely-used in simplifying complex datasets. PCA identifies new orthogonal variables, called principal components, that represent the most significant information in the dataset by analysing where observations are described by inter-correlated quantitative variables, t. The first principal component captures the maximum variance in the data, with each subsequent component capturing the next highest variance while remaining orthogonal to the others. This process reduces dimensionality while preserving the underlying structure of the dataset.[1].

2.4 Examples of recommendation systems

Bandyopadhyay et al. in [4] made use of PCA and k-means clustering to improve upon e-commerce recommendations. PCA is utilised to reduce the dimensionality of both consumer and product datasets by identifying the principal components. K-means clustering is used to cluster customers based on monthly income and previous purchasing history, whilst products are clustered using attributes such as price, sales volume and revenue.

Ahuja et al. made use of K-means and K-nearest neighbour (KNN) to form recommendations for moves. Movies are clustered based on their genres using Within-cluster sum of squares in order to optimise the number of clusters. Thereafter, KNN is then aided by a clustered utility matrix calculated with Pearson correlation to generate personalised movie recommendations. This system made use of the Root Mean Squared Error metric in order to show the improvement that was found in a previously used technique based on the MovieLens 100K dataset found on Kaggle [7] [3].

Yadav et al., similarly to Bandyopadhyay et al., make use of K-means clustering along with PCA in order to form a recommendation system. However, the system developed is applied to movie recommendations found in the MovieLens 100K dataset [7]. Using this technique, improvement from previous techniques in terms of root mean squared error (RMSE), mean absolute error (MAE) as well as the Dunn index (a metric used for evaluating clustering techniques) were found [14].

2.5 Critical analysis

Although the proposed technique is not a new technique, through extensive research it was found that K-means clustering in combination with PCA has not been used in recommendation systems for video games. This paper will therefore aim to bridge the gap in this research.

3. METHODOLGY

This system is written in python making use of multiple skikit-learn libraries for both the K-means clustering as well as the PCA. This section will run through a number of processes performed in both preparation of the dataset and implementation of the system. This research followed the CRISP-DM process proposed in [12] that covers the following structure: business understanding, data understanding, data preparation, data modelling, data evaluation and deployment.

3.1 Dataset formation

This section describes the collection step of the data understanding heading of the CRISP-DM methodology. The system integrates two datasets:

- Steam Video Games: A dataset that contains the behaviours of a number of users using the following attributes: Game title, behaviour name (play or purchase), total play time [11].
- Steam Games Dataset: This dataset contains information about games available on the Steam platform. Each entry includes details like the game's name, release date, price, required age, and a description. The dataset also includes data on supported platforms (Windows, macOS, Linux), game genres, developers and publishers. Additionally it provides some numerical data such as the number of achievements, user ratings, estimated owners, average playtime [9].

The information obtained by these two datasets where transferred to two new datasets known as the User and Game features datasets. In addition to the provided information, these datasets were enriched with inferred features, such as average playtime per game for each user, total playtime per game, and the number of players engaging with a game.

3.2 Exploratory data analysis

This section will run through the characteristics of both the collected and created datasets. The information collected from this EDA was used in determining what steps were needed to prepare the data in addition to helping find any obvious trends that might be present in the datasets.

The tables and graphs collected from the EDA performed on the datasets can be found in the Appendix, Sections A and B.

3.2.1 User dataset EDA

The EDA performed on this dataset revealed that the distribution of playtime for players varies quite widely throughout the dataset, with only small percentage of players playing more than 4 hours on average per game. This distribution can be seen in figure 5. In addition to this, we can see that the relationship between the number of games a player has bought and the number of games they have not played is directly proportional. This relationship can be seen in figure 6.

3.2.2 Game features dataset

The EDA on this dataset exposed that the distribution within the data was quite wide in terms of playtime (both total and average shown in figures 1 and 4) as well as the number of times a game was purchased.

3.3 Data preparation

A number of different steps were required to get the datasets to a state in which both K-means clustering and PCA could be applied.

The high distribution found in the game features dataset could have potentially led issues during the clustering by focusing too much on outlying data. To solve this it was decided to normalise the data using the equation 1.

In addition to this it was found that a column contained no information, labelled 'extra' and as a result this column was dropped.

The game features data originally contained a column that held an array of objects containing user given tags in addition to the number of times these tags were set by users. These tags were unwounded into columns with the value in the column being the number of times the tag was applied.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

In addition to normalisation of numerical columns, the game titles where encoded using Scikit-learn LabelEncoder library.

3.4 K-means with PCA recommendation system

Both users and games are assigned to clusters based on PCA-transformed features. For K-means clustering, 10 clusters were decided based on experimental outcomes from testing the following cluster values: 5, 10, 15, 20. PCA determines 4 principle components for both users and games. After which, these components are passed in the K-means clustering algorithm to identify the clusters into which users and games fall into separately. For each user, the algorithm identifies games played by others in the same cluster. Recommendations prioritize games with high engagement metrics, ensuring relevance and quality. Based on this description, the implemented system is classified as a Content-based recommendation system.

3.5 Regression model

A regression model, trained on normalized datasets, was also created as an additional method for recommendations. This model was trained in order to predict playtimes with reasonable accuracy. Based on this description, the implemented system is classified as a Content-based recommendation system.

A number of models and parameters where tested using a grid search method provided by the Scikit-learn library. The models tested included: Linear Regression, Random Forest Regression, Ridge Regression. This model was evaluated using the Root-mean squared error (RMSE) equation in addition to the Mean Absolute error (MAE) metrics shown in equations 2 and 3 respectively.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

Where:

- n is the number of data points,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

Where:

- n is the number of data points,
- y_i is the actual value,
- $\hat{y}_i$ is the predicted value.

3.6 Experimental setup

The systems implemented were coded using Python. Google Collab was utilised due to the limited public access to the Tensor Processing Unit v2-8. This increased the efficiency by which the data transformation and training of the regression model took place. An implementation of the code can be found in a GitHub repository located here[1].

4. EXPERIMENTAL RESULTS

4.1 PCA Visualisation

The scatter plots for both the user and game PCA values can be found in the Appendix, Sections C.1 and C.2 respectively. From these we can see some interesting relationships form between the determined principle components. The relationships between all four principle components are lain out in their own individual scatter plots. demonstrated clear separations between clusters, validating PCA's role in dimensionality reduction.

4.2 Cluster Visualisations

Games are recommended by finding users in the same cluster, thereafter games which the current user has not played in addition to games with median playtimes higher than a user's typical playtime were prominently suggested. Users were segmented into 10 clusters, and games into another 10, showcasing distinct groupings of behaviours and game features.

The system successfully recommended games based on cluster behaviors. Examples of recommendations for users found in cluster 3 are shown in table 3

4.3 Regression model

As stated, the root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics evaluated the performance. The results are shown in table 4.

5. RESULTS DISCUSSION

5.1 PCA Visualization and Clustering performances

The integration of PCA and K-Means clustering provided meaningful insights into user and game characteristics, enabling the identification of clusters that reflect distinct behavioral and attribute patterns and thereby allowing for game recommendations based on similar players to be made.

Although the scatter plots of principal components for both users and games did not reveal clear separations between clusters, both users and games were segmented into distinct clusters, validating PCA's utility in dimensionality reduction and aiding in uncovering latent patterns within the datasets. A few potential issues arose whilst implementing this system. The first being the dimensionality of the created datasets which also resulted in a sparsely populated dataset. However, this is where PCA comes in and helps reduce the highly dimensional data into a smaller dataset, focusing on the most important latent features.

[1]https://github.com/Rob-0ff/Honours_project/tree/HHdenoiser_v4.1

5.2 Regression analysis

The supplementary regression model showed predictive capabilities for playtime with RMSE and MAE values of 0.0389 and 0.1374, respectively. While the regression model provided accurate predictions, the clustering-based method was more effective in addressing user-specific recommendation needs as an optimal threshold based on the value produced by the regression model was unable to be determined. At present the normalised average playtime is utilised to determine if a game should be played by seeing if the predicted playtime is greater than the average.

5.3 Recommendations analysis

The system successfully recommended games that users in the same cluster frequently played but the target user had not. Recommendations such as Anguished 9 for cluster 3 users validated the system's capacity to enhance user engagement with high-quality suggestions.

6. CONCLUSION

This research demonstrates the efficacy of combining PCA and K-Means clustering in developing a recommendation system tailored for a subset of Steam's extensive game library. By segmenting users and games into clusters based on behavioural and feature-based patterns, the system offers scalable, personalized suggestions. The approach overcomes limitations of traditional systems by integrating inferred features and emphasizing user-specific engagement. Unfortunately, the system also struggles with well known issues of using these forms of recommendations such as the cold-start problem, by which a system struggles to provide recommendations due to little to no data being present. A potential real-world solution to this would be to use a questionnaire to assess a users likes and dislikes and recommend games that match these tags.

Future work could explore incorporating deep learning models to refine clustering and prediction accuracy further. Expanding contextual features, such as in-game achievements or dynamic play patterns, may enhance recommendation quality.

7. REFERENCES

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] A. Abdulnassar and L. R. Nair. Performance analysis of kmeans with modified initial centroid selection algorithms and developed kmeans9+ model. *Measurement: Sensors*, 25:100666, 2023.

[3] R. Ahuja, A. Solanki, and A. Nayyar. Movie recommender system using k-means clustering and k-nearest neighbor. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 263–268. IEEE, 2019.

[4] S. Bandyopadhyay, S. Thakur, and J. Mandal. Product recommendation for e-commerce business by applying principal component analysis (pca) and k-means clustering: benefit for the society. *Innovations in Systems and Software Engineering*, 17(1):45–52, 2021.

[5] B. Cui, H. Feng, S. Li, and L. Liu. The recommendation service of the shareholding for fund companies based on improved collaborative filtering method. *Procedia Computer Science*, 162:68–75, 2019.

[6] F. T. A. Hussien, A. M. S. Rahma, and H. B. A. Wahab. Recommendation systems for e-commerce systems an overview. In *Journal of Physics: Conference Series*, volume 1897, page 012024. IOP Publishing, 2021.

[7] A. Jha. Movielens 100k. Mendeley Data, V1, 2019.

[8] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134, 2002.

[9] M. B. Roman. Steam games dataset. `https://www.kaggle.com/datasets/fronkongames/steam-games-dataset?select=games.json`, 2023. Retrieved 10/15/2024.

[10] S. S. Sohail, J. Siddiqui, and R. Ali. Classifications of recommender systems: A review. *Journal of Engineering Science & Technology Review*, 10(4), 2017.

[11] A. R. Tamber. Steam video games. `https://www.kaggle.com/datasets/tamber/steam-video-games?resource=download`, 2016. Retrieved 10/07/2024.

[12] R. Wirth and J. Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1, pages 29–39. Manchester, 2000.

[13] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[14] V. Yadav, R. Shukla, A. Tripathi, A. Maurya, et al. A new approach for movie recommender system using k-means clustering and pca. *Journal of Scientific & Industrial Research*, 80(02):159–165, 2021.

Appendices

A. GAME FEATURES EDA

Column	Count	Mean	Std Dev	Min	25%	50%	75%	Max
avg_playtime	3629	1.44	0.92	0.10	0.78	1.30	1.93	7.17
purchase_no_play	3629	32.65	119.23	1.00	3.00	8.00	26.00	4841.00
total_playtime	3629	39.93	277.54	0.10	1.55	4.41	16.53	12929.23
total_players	3629	32.46	119.01	1.00	3.00	8.00	26.00	4841.00
release_date	3629	2020.11	3.09	2006	2018	2021	2023	2024

Table 1: Summary Statistics of Key Columns for the Game features dataset

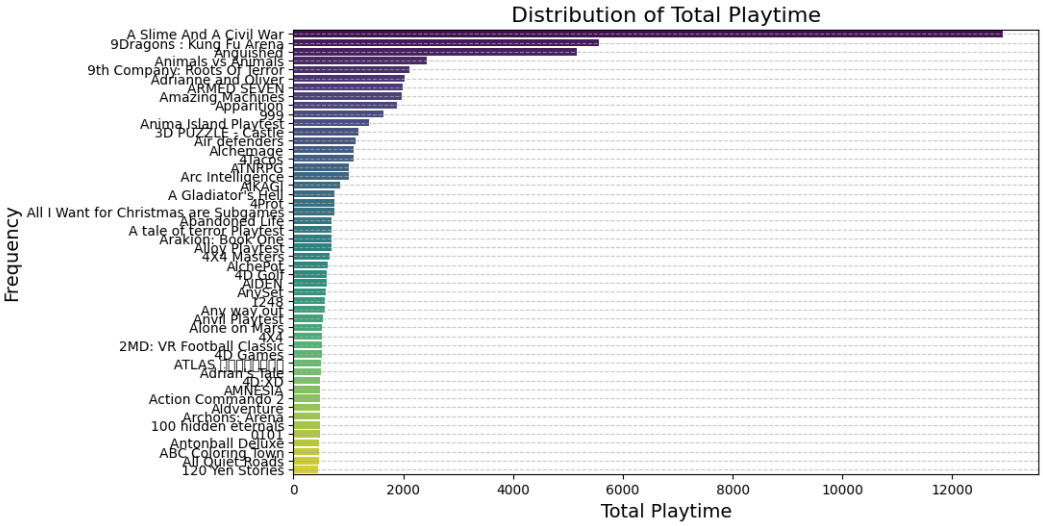


Figure 1: Total playtime (hours) of the top 50 games (Pre-normalisation)

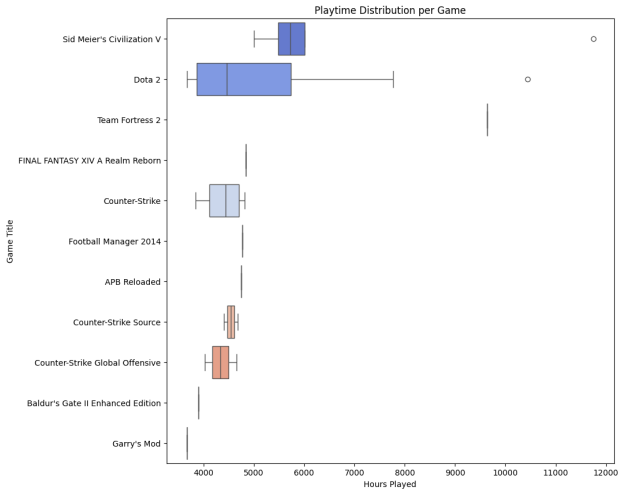


Figure 2: Playtime distribution (hours) of the top 10 games (Pre-normalisation)

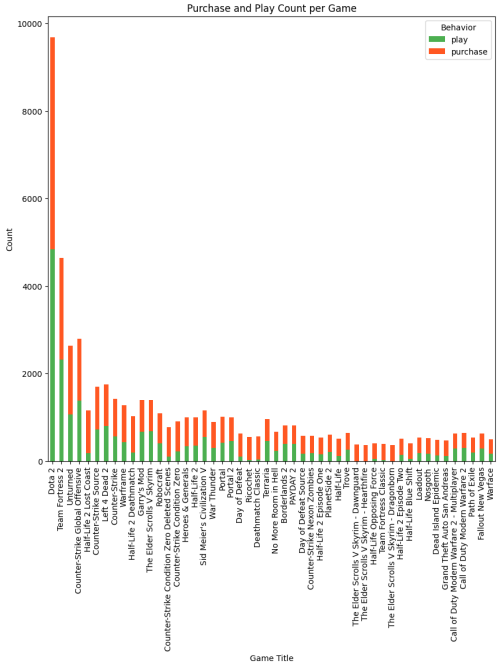


Figure 3: Instances of actual playtime vs purchase with no playtime for the top 50 games (Pre-normalisation)

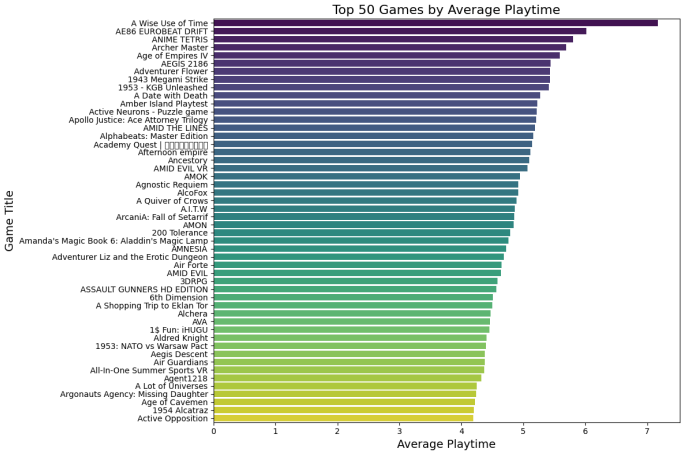


Figure 4: Average playtime (hours) of the top 50 games (Pre-normalisation)

B. USER FEATURES EDA

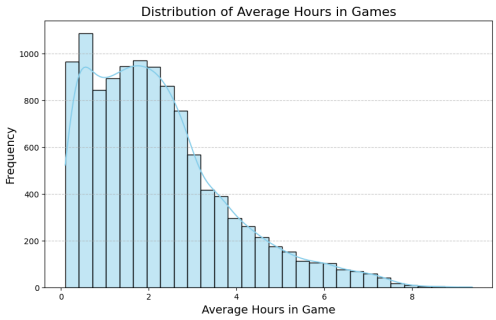


Figure 5: Average hours played per game per player (Pre-normalisation)

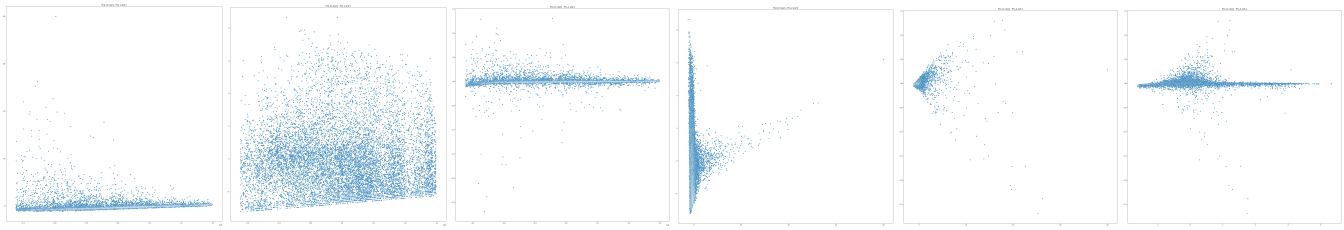
Figure 6: Scatter plot showing the relationship between total games purchased vs not played by users (Pre-normalisation)

Statistic	**user_id**	**avg_hours_in_game**	**purchase_no_play**	**num_bought**
Count	11,350	11,350	11,350	11,350
Mean	160,862,000	2.216	11.181	17.391
Standard Deviation (Std Dev)	78,473,510	1.585	37.369	54.023
Min	5,250	0.095	1.000	2.000
25th Percentile (25%)	99,257,870	0.993	1.000	2.000
Median (50%)	163,514,800	1.933	2.000	3.000
75th Percentile (75%)	217,791,600	3.001	7.000	10.000
Max	309,903,100	9.372	1,075.000	1,573.000
Unique Values	11,350	5,386	244	316
Null Values	0	0	0	0
Duplicated Rows	0	0	0	0

Table 2: Summary of Data Statistics of the User features dataset

C. EXPERIMENTAL RESULTS

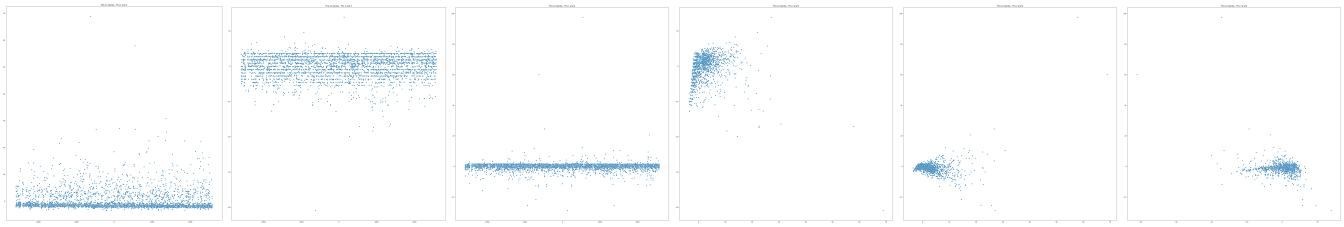
C.1 User results



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4 (e) Image 5 (f) Image 6

Figure 7: Images showing the relationships between different principle components calculated [(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)] of the user features dataset

C.2 Game results



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4 (e) Image 5 (f) Image 6

Figure 8: Images showing the relationships between different principle components calculated [(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)] of Game features dataset

C.3 Clustering Results

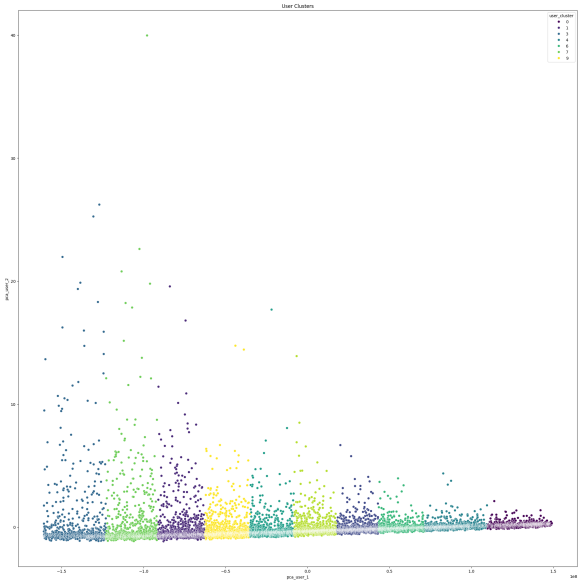


Figure 9: Scatter plot showing the User clusters generated after performing PCA

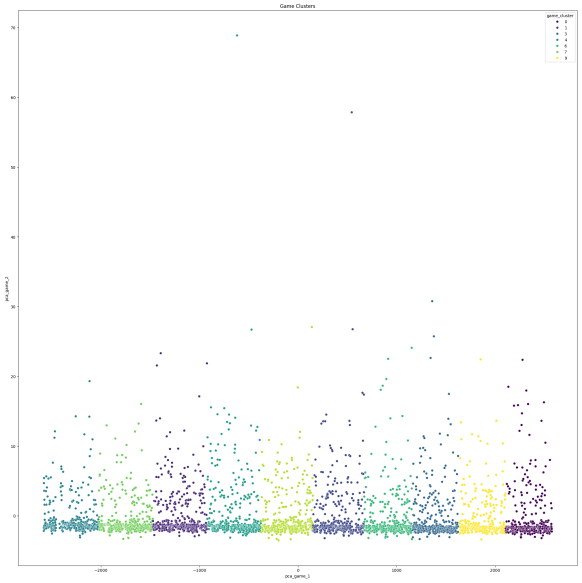


Figure 10: Scatter plot showing the Game clusters generated after performing PCA

User ID	**User cluster**	**Recommended Games**	**Game Clusters**
298950	3	Anguished	9
		9Dragons : Kung Fu Arena	7
		Adrianne and Oliver	8
		Alchemage	2
140825164	3	Anguished	9
		'A Slime And A Civil War	6
17530772	3	Adrianne and Oliver	8

Table 3: Recommended Games for multiple Users

Table 4: Model Performance Metrics

Metric	**Train**	**Test**
Root Mean Squared Error (RMSE)	0.0389	0.0389
Mean Absolute Error (MAE)	0.1369	0.1374