Unsupervised Machine Learning: Behavior-Based and Popularity-Based Clustering of Roblox Games.

Nicole B. Pagkatipunan
College of Computing and Information Technologies
National University, Manila, Philippines
pagkatipunannb@students.national-u.edu.ph

Franc Thomas A. Sia
College of Computing and Information Technologies
National University, Manila, Philippines
siafa@student.national-u.edu.ph

***Abstract*—This paper applies clustering techniques on data from public Roblox games metadata to analyze patterns in game popularity and engagement patterns of players. Based on K-Means, DBSCAN, and the Gaussian Mixture Models, using crafted popularity and behavioral features, the study identifies sets of games that vary according to scale and engagement characteristics, showing how clustering technique can help identify patterns of performance on large-scale gaming platforms.**

***Index Terms*—Clustering, K-Means, Gaussian Mixture Model, PCA, Behavioral Analytics**

I. Introduction

Digital game platforms have become vast ecosystems in which patterns of user engagement, popularity and interaction produce enormous amounts of behavioral data. Roblox in particular is a unique platform where millions of games coexist with extremely heterogeneous audiences, from mainstream to niche communities. As platforms play an increasingly influential role in the way in which content is produced, discovered and consumed, understanding the clustering of games into meaningful behavioural categories is a salient analytical endeavour. A study call this phenomenon the ”platforming of everything,” claiming that digital infrastructures are not only the host for content, but they also provide a platform for structuring user participation and visibility [1].

Despite the fact that there are available performance indicators such as active users, favorites, and visit counts, the raw popularity of games alone is not enough to fully explicate the behavioural identity of games. Some titles maintain long-term loyal audiences, while others have acute viral spikes followed by decay. Detecting these kind of patterns requires unsupervised learning techniques that can group games on behavioural similarity without predefined labels.

Clustering is popularly used for the purpose; but challenge lies in the choice of the best type of clustering model. Different algorithms make different assumptions about the structure, density and distribution of clusters. For example, K-Means is efficient and popular, but it works best if clusters are spherical and of similar size [2]. Gaussian Mixture Models (GMMs) support probabilistic membership and are able to model overlapping behavioural groups [3]. Density based methods like DBSCAN are very good at detecting outliers and irregular cluster shapes, which is very helpful if there are viral or anomalous games in the data set [4].

Consequently, this research is driven by the desire of benchmarking multiple clustering algorithms and identifying the one that produces the most interpretable and meaningful segmentation of Roblox games. Moreover, cluster validity and selection are important issues, especially for large-scale datasets, for which estimating the number of clusters is not trivial. [5] stress the need for robust cluster estimation methods for reliable unsupervised learning results.

This research focuses on the evaluation of clustering approaches based on both raw performance features and engineered behavioural metrics to determine salient game categories, which include mainstream, niche loyal experiences and one hit wonders.

II. Literature Review

A. Clustering in Platform Behavioral Analysis

The main topics of the course evidenced in the learning resources that are required to deliver the course.Digital platforms create behavioral ecosystems that are multidimensional in terms of their behavioral outcomes with user interaction and algorithmic visibility driving content performance. According to [1], platforms define epistemologies of engagement more and more, i.e. popularity measures do not indicate the preference of users, but amplification through the platforms. This renders the clustering methods useful in the revelation of concealed behavioral groupings beyond the superficial rankings.Within the realm of gaming, clustering would allow assisting in the segmentation of experiences into meaningful categories and would allow the researcher to interpret the engagement trends and patterns of behavior in players.

B. K-Means and Improvements

K-Means is still among the widely used clustering algorithms in spite of its simplicity and high-speed computational nature. It divides the data into k clusters at minimum variance within clusters. Nevertheless, this approach is initialisation sensitive, presupposes a spherical cluster structure and in many cases, it fails when uneven or non-linear data distributions appear. Another comparative analysis conducted by [2] presents results of the standard and improved K-Means algorithms, demonstrating that the centroid selection and optimization can be improved to produce the outcomes of clustering that were more stable. Base-line K-Means becomes a significant point of reference in the context of the Roblox games, especially with the row-popularity (visits and favorites) as the metric used to group games together.

C. Gaussian Mixture Models of Probabilistic Clustering.

Gaussian Mixture Models (GMMs) are built on the principle of clustering whereby the data points are held to be produced as a mixture of Gaussian distributions. In contrast to K-Means, GMMs provide soft assignments, which is that each game can be in clusters with probability distributions. [3] emphasizes that GMMs prove to be particularly effective when there is an overlap of clusters or when there is no clear-cut pattern of behavior. This renders GMM suitable to Roblox data, with the games potentially having similar traits in each popularity level, the level of engagement, and retention behavior.

D. Dense-Mode Clustering and Outlier Detection.

Methods that are based on density like DBSCAN grouping is the method that uses dense regions in the feature space to label sparse ones as noise. The property can be used in the detection of anomalies such as viral games which do not act like the majority. The article by [4] suggests the DBSCAN+ framework, a more powerful version of the density-based clustering algorithm, which is stronger due to its statistical check. These approaches are especially applicable to behavioral data sets, in which clusters can be non-uniform in shape and have outliers.

E. Cluster Validation/Model Selection.

The main challenge with clustering is finding the right amount of clusters as well as meaning of the results. [5] suggest the use of ensemble-based procedures to estimate the number of clusters in the big data and observe that the choice of the cluster is one of the key factors that influence the reliability. Moreover, deep clustering has been identified recently as a learning method of learning complex representations. [6] are giving an excellent literature review of deep clustering and describe some major challenges, including interpretability, scalability, and evaluation. Although deep clustering shows promise, the more traditional algorithms such as K-Means, GMM, and DBSCAN continue to dominate the use of clustering as their transparency and ease of interpretation is crucial in the benchmarking research.

F. Research Gap

Although extensive research has been done on clustering techniques, little has been done on benchmarking different clustering algorithms specifically to analyze behavioral patterns in large user-generated games platforms like Roblox. Repeated literature typically addresses aspects of algorithmic enhancements, rather than comparative model interpretability. This paper fills this loophole by using and comparing K-Means, GMM, and DBSCAN to Kickstarters game analytics and designed behavioral variables, with the aim of determining the approach of K-Means that offers the most interpretable K-Means users to divide platform-based games.

III. Methodology

A. Dataset Description

The analysis is based on a sample of Roblox games that the researchers have gathered in metadata found publicly on the platform. One game is represented by each record and has such attributes as title, genre, creator, description, server size, the number of active users, the number of favorites, the total number of visits, creation date, and the last update date. Descriptive and non-numeric fields (e.g., Creator, Description and URL) had been eliminated since they are not directly amenable to numerical clustering. The attribute Total Visits that was formatted as strings (1.2M, 500K) was changed to floating-point data by substituting the suffixes (K, M, B) with the corresponding numeric multiplier. The rows with the missing or invalid values of key numerical fields were eliminated to provide the stable model training. The dataset was cleaned after which N games were available. Each experiment used fixed random seeds, thus, being reproducible.

Two preprocessing pipelines have been built:

1) a popularity based pipelining baseline and optimized K-Means clustering, and
2) a pipeline using behavior models of DBSCAN and Gaussian Mixture Models (GMM).

B. Exploratory Data Analysis (EDA)

Exploratory data analysis (EDA) had been performed before the construction of the models to learn about the data structure, uncover inconsistencies, and inform pre-processing and feature engineering choices.

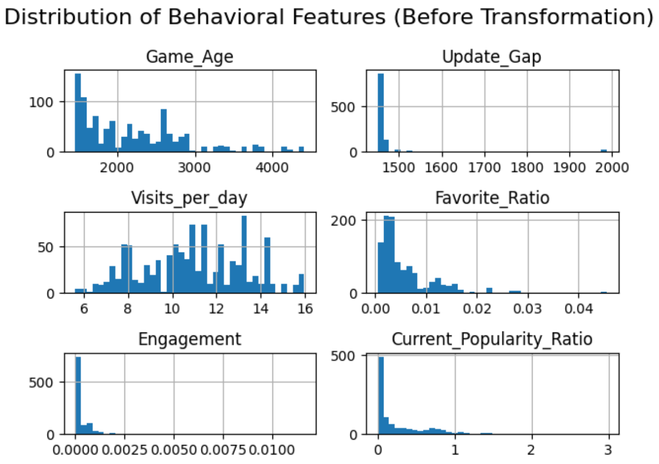

Fig. 1. PCA component contributions for behavioral features

To analyze all the numerical variables, such as Total Visits, Favorites, Active Users, and Server Size first, summary statistics and feature distributions were analysed. The distributions of these variables were incredibly skewed as the number of games with very large popularity values was very small. This observation inspired the subsequent normalizations and log based transformations that were applied later in pipelines.

Second, data quality checks were done to determine missing, invalid or improperly formatted data. The records that had missing or non-numeric values in the critical fields of popularity were deleted to make the records stable during clustering. It was also confirmed during the stage that conversion of Total Visits values of abbreviated strings (ex: K, M, B suffixes) to the number form.

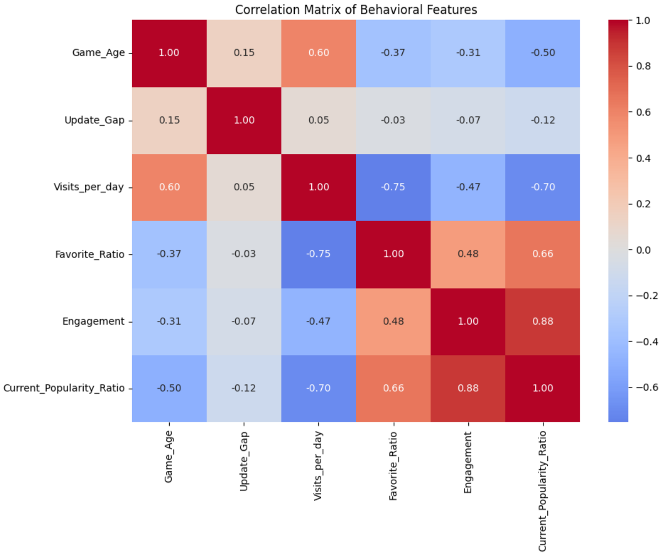

Fig. 2. PCA component contributions for behavioral features

Third, the correlation analysis and visualization of the relationships between popularity metrics was inspected. The good correlations among visits, favorites, and active users proved that these variables reflect various faces of popularity of the game, and the distinction between popularity-based and engagement-based clustering pipelines will be achieved.

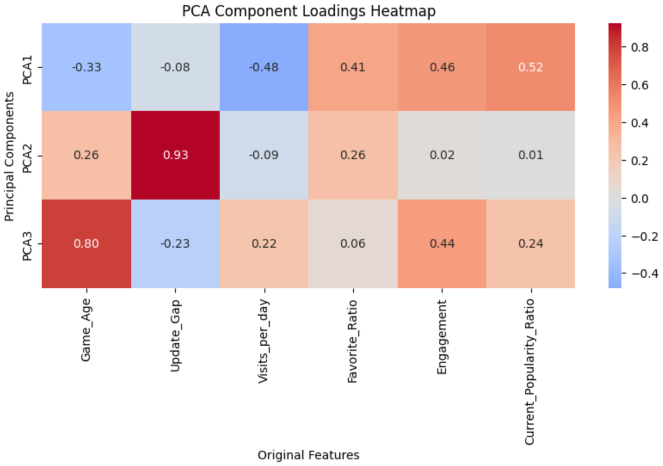

Fig. 3. PCA component contributions for behavioral features

Lastly, a Principal Component Analysis (PCA) of the data at the global level was performed to explore the data structure. It has been demonstrated by PCA projections that variation was mainly controlled by the differences in popularity scales, which supported the necessity to include ratio-based and behavioral characteristics in the subsequent pipelines to be able to capture the patterns of engagement, as opposed to raw exposure.

C. *Popularity-Based Feature (K-Means) Pipeline.*

1) Feature Selection: In the case of the baseline K-Means experiment, the following four popularity and capacity features were chosen:
   - Active Users
   - Favorites
   - Total Visits
   - Server Size
2) Scaling: All the features were normalized using z-score normalization through StandardScaler since K-Means is based on Euclidean distance, so that each feature has zero mean and unit variance. This helps to avoid the large-scale variables like Total Visits dominating distance calculations.
3) Baseline K-Means Training: K-Means clustering was tried with cluster counts kkk ranging between 2 and 6. The within-Cluster sum of squares (WCSS) was calculated on each value and elbow technique was applied to find out a suitable number of clusters. The elbow point showed that k 3 gave an appropriate tradeoff between compactness and simplicity. The last potential combination of configuration:
   - Initialization: k-means++
   - Number of initializations: 10
   - Random seed: 42

   This was done by assigning each game to the closest centroid in the normalized feature space.
4) Evaluation: To determine the quality of clustering, the silhouette score was used which is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \quad (1)$$

   Where a(i) is the average distance of sample iii to other points within the same cluster and b(i) is the minimum of the average distance to others within different clusters. The interpretation of cluster profiles involved the calculation of means features when per cluster. Visualization was done by Principal Component Analysis (PCA) into two dimensions of the projected standardized features. PCA was not employed to determine outcomes but was only used to visualize.

D. *Pre-processing Pipeline II: Engagement-Enhanced Features (Optimized K-Means)*

1) Feature Engineering: In order to introduce an aspect of engagement level beyond the crude popularity two ratio based features were added:

$$R_{\text{fav/visit}} = \frac{\text{Number of Favorites}}{\text{Number of Visits}} \quad (2)$$

$$R_{\text{active/visit}} = \frac{\text{Number of Active Users}}{\text{Number of Visits}} \quad (3)$$

These ratios result in user loyalty and interaction as compared to exposure. The end feature set was hence comprised of Active User, Favorites, Total Visits, Server Size, Favorites per Visit, Active Users per Visit. Before training, infinite or undefined values which occurred as a result of an operation of dividing by extremely small or zero counts of visits were eliminated.

2) Scaling and Training: All six characteristics were normalized on z. Cluster sizes of 2-6 were also tested with k of 3 again, as it was more interpretable and comparable to the baseline. The configuration of training was the same as the baseline K-Means configuration.
3) Evaluation: The computation of silhouette scores was done to compare the improved representation to the baseline model. Mean feature statistics were used to describe the engagement behavior by analyzing the cluster centroids. Clusters were characterized by PCA and visualized in three dimensional space with three components where the mainstream, mid-tier, and niche categories were defined using the characteristics. Heatmaps of PCA input were analyzed to understand the contribution of features as well.

E. Preprocessing Pipeline III: Behaviour-Based Features (DBSCAN and GMM)

1) Behavioral and Temporal Characteristics Feature Engineering. A behavior space was built to represent lifecycle and engagement behavior:
   - Game Age (days since creation, cut down to 7 days at least)
   - Update Gap (they accelerate as the days pass between updates)

   A number of engagement measures were then calculated:

$$\text{Visits per Day} = \frac{\text{Total Visits}}{\text{Game Age}} \quad (4)$$

$$\text{Favorite Ratio} = \frac{\text{Favorites}}{\text{Total Visits}} \quad (5)$$

$$\text{Engagement} = \frac{\text{Active Users}}{\text{Total Visits}} \quad (6)$$

$$\text{Current Popularity Ratio} = \frac{\text{Active Users}}{\text{Visits per Day}} \quad (7)$$

   To minimize the effect of the extreme outliers, the feature ratio was clipped to percentiles range that was determined empirically. Log-transformed highly skewed variables were used.

$$x' = \log(x + 1) \quad (8)$$

   The remaining values that were either infinite or missing were eliminated before clustering.
2) Scaling: All the behavioral features were normalized before clustering since both the DBSCAN distance computation and GMM covariance determination is related to the scale of features.

F. DBSCAN Model

1) Model Architecture: The density based clustering algorithm known as DBSCAN is controlled by; Epsilon refers to neighborhood radius, while MinPts is the minimum amount of points needed to create a thick area. DBSCAN is able to find arbitrary shaped clusters and determine sparse observations as noise unlike K-Means.
2) Training and Value choice: A search of k distance parameter settings was made before building a k-distance graph with a 10th nearest neighbor. An optimum value of epsilon= 0.85 was represented by the elbow point of this graph. MinPts of 10 were used to train the final model.
3) Evaluation: As the DBSCAN marks noise points as -1, silhouette scores were calculated only on non-noise samples when there were found more than one cluster. The outliers and the number of clusters that were identified were also reported.

G. Gaussian Mixture Model (GMM)

1) Model Architecture: GMM models represent the data as a combination of multivariate global mean distributions. The components are defined by the following aspects:
   - A mean vector
   - A full covariance matrix
   - A mixing weight

   Complete covariance matrices permit clusters to be elliptic in shape. The model gives soft cluster probabilities to every sample.
2) Training Procedure: Models of 1 to 8 components were (EM) trained through Expectation-Maximization algorithm. The selection of a model was done by the use of the Bayesian Information Criterion (BIC) and Akaike Information Criterion (AIC); k = 3 resulted in the lowest values of both criteria. The last model had 20 random initializations to provide stability of convergence. Contemporary times have placed increased demands on teachers to assess students' performance and share the results they gather with teachers.
3) Evaluation: The current times are emphasizing more on the performance of the students and giving teachers the outcomes of the evaluations they provide. Maximum posterior probability was used to achieve hard cluster assignments which were used to get label assignments and soft assignments were held back to be interpreted. Silhouette was calculated on hard labels. Mean behavioral features revealed clusters that were used to describe a cluster profile and interpreted as one of Mainstream, Average or One-Hit Wonder. Both DBSCAN and GMM cluster assignments were visualised by PCA projections

into three dimensions and PCA loadings were analysed to understand the major behavioural patterns.

IV. Results

A. K-Means Clustering Baseline

A K-Means baseline clustering was done based on four numerical characteristics, which are Active Users, Favorites, Total Visits and Server Size. Before clustering all features were standardized. The elbow method was tested to get the number of clusters between $k = 2$ and $k = 6$ and $k = 3$ was determined using the elbow point. The resulting clustering had a silhouette score of 0.768, which signified strong separation among clusters.

Means on the cluster level show three levels of gameplay popularity:

- A large mainstream cluster where visits, favorites and active users are extremely high.
- A middle range cluster of games with moderate popularity.
- A cluster with relatively low visits but unusually high niche-like server counts.

Two-dimensional PCA projection confirms visible clustering in reduced dimension, supporting the appropriateness of K-Means for baseline segmentation.

B. Behavioral Ratios K-Means Optimized

To enhance behavioral differentiation, two engagement ratios were added:

- Favorites per visit
- Active users per visit

After removing invalid ratio values and rescaling the data, K-Means was applied again with $k = 3$ clusters.

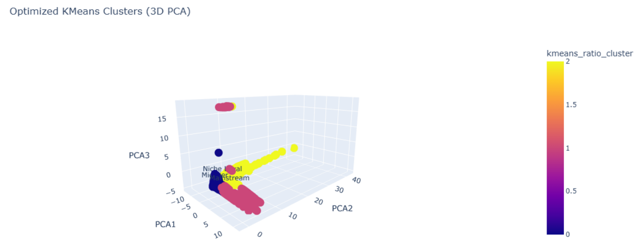

Fig. 4. Optimized K-Means clustering visualized using PCA projection

The optimized model produced a silhouette score of 0.589, smaller than the baseline but yielding more behaviorally meaningful clusters:

1) Mainstream games: very large traffic with relatively low engagement ratios.
2) Mid-level games: average traffic and average loyalty.
3) Niche loyal games: lower visitors but significantly higher engagement and loyalty rates.

Three-dimensional PCA visualization shows separation mainly along engagement-related axes, indicating behavioral clustering rather than scale-based clustering. PCA component loadings also show that engagement ratios and popularity measures strongly influence the principal components.

C. Behavioral Feature Modeling Using Density and Mixture Approaches

A second experiment focused on behavioral dynamics rather than raw scale. Engineered features included:

- Game age
- Update gap
- Visits per day
- Favorite ratio
- Engagement ratio
- Current popularity ratio

Skewed variables were log-transformed and extreme ratio values were capped to stabilize distributions. DBSCAN and GMM were then applied after preprocessing and scaling.

Exploratory analysis showed correlations between engagement and popularity measures, supporting the behavioral feature design.

D. DBSCAN Clustering Results

The initial DBSCAN configuration ($\varepsilon = 0.9$, $min\_samples = 20$) produced:

- 937 games assigned to one cluster
- 93 games labeled as noise

This indicated low separation. When $min\_samples = 10$ and $\varepsilon = 0.85$, the silhouette score was 0.253. Although slightly improved, DBSCAN still grouped most games into a single dense population with scattered outliers.

Therefore, DBSCAN primarily functioned as an outlier detector, identifying games that do not follow normal engagement behavior.

E. Gaussian Mixture Model (GMM) Results

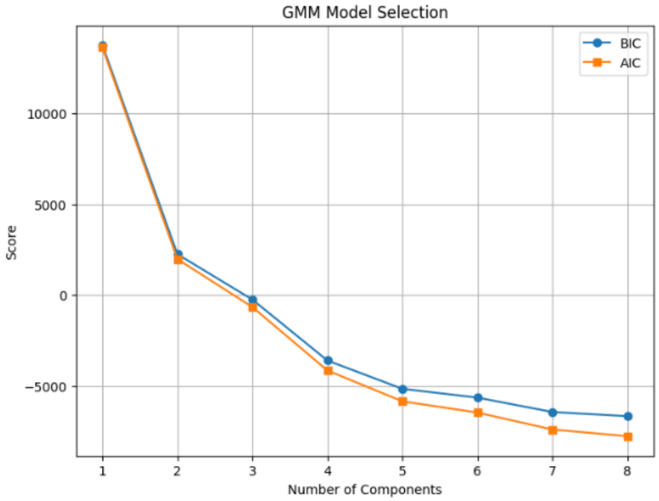

Fig. 5. GMM model selection using AIC and BIC scores

AIC and BIC comparisons for one to eight components indicated that three components provided the best balance between fit and complexity.

The optimized GMM model used:

- 3 mixture components
- Full covariance matrices
- Multiple initialization runs

Approximate mixing weights:

- Cluster A: 42%
- Cluster B: 51%
- Cluster C: 6%

Behavioral interpretation:

- Cluster 1 (Mainstream): high visits per day and sustained popularity.
- Cluster 0 (Average): moderate traffic and engagement.
- Cluster 2 (One-Hit Wonder): historically popular but low long-term engagement.

Soft clustering probabilities also showed uncertainty for some games, indicating transitional titles between popularity levels. PCA visualization confirmed cluster separation, with engagement and popularity ratios dominating component variance, while update frequency formed an independent variation axis.

F. Comparative Observations

Across all methods:

- Best separation by scale: baseline K-Means.
- Optimized K-Means improved behavioral segmentation but reduced compactness.
- DBSCAN primarily detected anomalies rather than forming clear groups.
- GMM produced the most interpretable behavioral clusters through probabilistic modeling.

GMM ultimately identified mainstream, average, and one-hit-wonder game types, aligning well with platform dynamics.

V. Discussions

A. Experimentation and Exploration of Solutions

This study used a systematic experimentation of various clustering algorithms, feature representations, and parameter settings to evaluate unsupervised methods of behavioral segmentation of Roblox games. Three families of algorithms were tested: partition-based clustering using K-Means, density-based clustering using DBSCAN and probabilistic clustering using Gaussian Mixture Models (GMM). Structured experimentation was done on each of the models rather than depending on a single-configuration setting. K-Means clustering was tested on a variety of cluster numbers through the elbow method, and more useful features based on engagement were implemented to to examine how engagement-based features influence segmentation. K-distance graphs were then used to adjust DBSCAN parameters to find a good neighborhood radius and minimum point threshold. GMM models have been tested on Bayesian and Akaike information criteria across various numbers of components, which ensures principled model selection. Stochastic algorithms were initialised repeatedly in order to increase the stability of training. The experiment proves that the clustering outcomes are very sensitive to feature representation. Magnitude-based segmentation is created by popular features and behavioral discrepancy (user loyalty and retention) is discovered through engagement-balanced features. This meant that it was necessary to experiment with various settings to obtain various structural views of the data.

B. Training Strategy and Hyperparameter Optimization

The training strategies of the models were well calculated so that there would be stable and fair comparisons among algorithms. In K-Means, the number of clusters was measured in a systematic way and to mitigate centroid instability, k-means++ was initiated. Several initializations alleviated the convergence to local optima of low quality. The parameters of the DBSCAN were optimized in terms of k-distance analysis, giving e and MinPts the best values. Although parameter tuned, behavioral feature space did not show sharp transitions between clusters but instead smooth transitions between densities, which prevented global separation but permitted effective anomaly detection. In case of GMM, the number of components was determined based on BIC and AIC statistics, and several attempts of running the Expectation-Maximization process would provide the stability of convergence. This probabilistic modeling enabled arising of overlapping behavioral identities due to the continuous engagement patterns. These training methods guaranteed strong performance of clustering and reduced instability due to random initialization.

C. Evaluation and Interpretation of Results

Assessment was done using silhouette scores, cluster distributions, model selection criteria, and PCA-based visual analysis. Findings indicate that there is a significant trade-off between behavioural interpretability and geometric separation. Baseline K-Means got the largest silhouette score which means good separation according to the magnitude of popularity. These clusters are however, more of exposure tiers than engagement behavior. The optimized K-Means model used engagement ratios, and clusters of niche and loyal games that were only niche but not all the time appeared, but the geometric separation was slightly diminished because of overlapping behavioral features. DBSCAN identified a subset of games as noise points which proved the presence of the games which do not follow the dominant patterns of engagement. Although the silhouette scores were small, DBSCAN was successful in isolating the anomalous or viral games that do not follow any general behavioral pattern. GMM presented average silhouette performance, but offered most behaviorally meaningful segmentation in terms of probabilistic cluster assignments. Most games had transitional engagement properties, which is in line with the arguments that platform engagement can be observed in a continuum as opposed to discrete categories. In this way, the clustering effectiveness relies on the analytical goal. K-Means is good at categorizing games with equal or similar magnitude of engagement, DBSCAN is good at detecting anomalies and GMM is good at capturing behavior overlap and transitional identities.

D. Reproducibility and Implementation Considerations

The pipeline of the experiment was the reproducible and modular design. Seed randomization was fixed, preprocess measures were always being used and model settings were always being reported. While feature engineering, scaling, clustering and evaluation phases were decoupled to enable their independent reproduction and expansion. Libraries of standard machine learning are used, which guarantees cross-environment reproducibility. The designed pipeline enables further researchers or developers to add new functionality, algorithms, or validation strategies without compromising comparability with existing results.

E. Alignment with Research Objectives and Model Selection

The findings respond directly to the objectives of the study. Learning without supervision was able to categorize games in the Roblox world based on meaningful engagement, making it possible to identify the mainstream, mid-tier, and niche loyal experiences. Games with comparable levels of engagement were grouped and weird titles that did not follow the general trends were singled out. Various clustering methods showed various structural patterns in the data. K-Means will be effective in clustering games based on size of engagement, DBScan will help identify outliers and GMM will help in capturing behavioral patterns. Therefore, there is no universal model that is predominant in every analysis undertaking. Nevertheless, under conditions of more emphasis on behavioral interpretation and platform ecosystem insight, probabilistic clustering yields the most informative division. Thus, although model selection must be based on the purposes of analysis, the best trade off between interpretability and behavioral realism is provided by the Gaussian Mixture Models.

VI. Conclusion

This study has compared K-Means, DBSCAN, and Gaussian Mixture Models as benchmarks to identify appropriate models of unsupervised learning in segmenting behavioral patterns of the games in Roblox according to popularity and engagement aspects. The results show that unsupervised learning methods could be useful in the segmentation of the games into meaningful engagement-based categories that would permit mainstream and niche experience to be identified. Similar magnitude engagement games were clustered effectively, and the odd games that do not follow the dominant patterns were pointed out using density-based clustering. As a comparative analysis reveals, both clustering techniques focus on the various structural features of the data. K-Means does very well at magnitude based grouping, DBSCAN does very well at identifying engagement outliers and Gaussian Mixture Models do very well at probabilistic segmentation and this can be used to model overlapping behavioral tendencies. When the interpretation of engagement is important, GMM provides the most realistic behavior description of the methods that are evaluated. However, the ultimate choice of model must be a result of the analytic objectives; either scale-based clustering, aberration identification or an interpretation of behavior.

A. Limitations

There are a number of limitations that should be recognized. The data set is a one-time look into the past, which excludes longitudinal analysis of engagement change. The effects of platform algorithms on visibility were not modeled directly and other variables like genre, monetization strategy or social interaction networks were not taken into consideration. Moreover, deep clustering techniques were not tested.

B. Future Work

The future study can further develop this study by adding time-based data to monitor the lifecycle transitions, deep representation learning framework to learn nonlinear behavioral patterns, ensemble clustering strategy evaluation, and other platform interaction measures. Longitudinal analysis could also indicate the way games move between segments of engagement as time goes by.

References

[1] S. Maher and S. Monaci, “Media technologies and epistemologies: The platforming of everything—introduction,” *International Journal of Communication*, Sep. 2024. [Online]. Available: https://ijoc.org/index.php/ijoc/article/view/23795

[2] M. Pokharel, J. Bhatta, and N. Paudel, “Comparative analysis of k-means and enhanced k-means algorithms for clustering,” *NUTA Journal*, vol. 8, no. 1–2, pp. 79–87, Dec. 2021.

[3] P. Sarang, “Gaussian mixture model,” in *Springer Series in Applied Machine Learning*. Springer, 2023, pp. 197–207.

[4] Y. Xie, X. Jia, S. Shekhar, H. Bao, and X. Zhou, “Significant dbscan+: Statistically robust density-based clustering,” *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 5, pp. 1–26, 2021.

[5] M. S. Mahmud, J. Z. Huang, R. Ruby, and K. Wu, “An ensemble method for estimating the number of clusters in a big data set using multiple random samples,” *Journal of Big Data*, vol. 10, no. 1, Apr. 2023.

[6] S. Zhou, H. Xu, Z. Zheng, J. Chen, Z. Li, J. Bu, J. Wu, X. Wang, W. Zhu, and M. Ester, “A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions,” *ACM Computing Surveys*, vol. 57, no. 3, pp. 1–38, 2024.