ML Models of Player Engagement Levels

AI and Machine Learning (ACCA6011)

Jovanna De La Trinidad (2504811)

Abstract

This report looks into different machine learning methods to understand and predict player engagement within a game development studio. Using the Predict Online Gaming Behavior Dataset, a Decision Tree and Support Vector Machine (SVM) model were trained to identify engagement levels based on the datasets demographics and gaming habits, with SVM producing the most accurate predictions. Additionally, K-Mean Clustering Algorithm is implemented to create behavioral groups by analysing features within the database. Overall, the combination of supervised and unsupervised learning methods provide valuable insights into player engagement and behaviors, offering practical guidance for enhancing retention strategies and personalizing game design.

1. Introduction

A game development studio aims to understand its players' engagement. Using the Predict Online Gaming Behavior Dataset on Kaggle (Rabie El Kharoua, 2024), the analytics team has access to data on demographics, gaming habits, and player engagement levels. This data enables an analysis of current behaviors and the development of models to predict future player engagement.

This report applies and evaluates Decision Tree and Support Vector Machine (SVM) models to predict player engagement. It also uses an unsupervised learning technique, the K-Means Clustering Algorithm, to discover behavioral patterns among users. Through this analysis, the gaming studio can identify which behaviors and patterns correlate with higher engagement and which areas of the game need improvement to support better retention and user satisfaction. Machine learning enables companies to understand player behaviours and make informed decisions by analyzing behaviour and predicting future engagement patterns. Companies can use

supervised and unsupervised learning methods to support player engagement and build models to predict future user engagement.

2. Problem Statement

Game companies rely on player engagement to maintain retention, grow their user base, and increase revenues (*Gaming Analytics: Unveiling Insights for Player Retention and Monetization*, 2024). The Predict Online Gaming Behavior Dataset contains information for 40034 users, including their demographics, gaming habits, and engagement levels. The aim is to use the provided data to understand better what drives engagement levels and to build models that accurately predict how engaged a player is likely to be. Customer engagement is one of the most critical factors in determining a company's success because it directly influences retention and long-term revenue (*Deep-Dive into AI/ML to Enhance Customer Engagement*, 2023). Machine learning algorithms support this by identifying predictors of user behaviour and discovering hidden patterns in engagement levels (Necula, 2023).

To address these objectives, a supervised learning model, such as Decision Trees and SVMs, can be used to predict engagement categories (High, Medium, or Low) based on players' demographics and gaming habits. As well, unsupervised learning methods, such as K-Means clustering, can group players autonomously to identify player types and their behaviors. Using supervised and unsupervised learning, the gaming company gains insights that inform game design, retention strategies, and personalized engagement initiatives.

3. Methodology

Data Exploration and Cleaning

The purpose of data exploration is to examine the dataset, identify any discrepancies, and determine what is important and what is not for the ML Models. After identifying what is

necessary and what is unnecessary to optimize space, we remove the columns that serve no purpose.

Decision Tree and SVM Models

The studio needs a supervised learning approach to predict player engagement. A Decision Tree is easy to interpret, flexible, and requires minimal preprocessing (GeeksforGeeks, 2017b). A Support Vector Machine (SVM) model classifies the data by finding the optimal separating hyperplane between the classes (GeeksforGeeks, 2024a). SVMs maximise the distance between the boundary and the closest data point, helping the model see the unseen data (GeeksforGeeks, 2024a). Decision trees help identify factors that influence player engagement, while SVM models achieve high accuracy by capturing complex player behavior patterns. Using both methods allows comparison between a simple, interpretable model and a robust non-linear classifier.

Evaluation Metrics

Analyzing the Decision Tree and SVM models is necessary to assess accuracy, precision, recall, F1-Score, and AUC-ROC. Accuracy shows how often the model correctly predicts the engagement level. Precision measures how many of the players predicted to have high engagement actually did. Recall measures how many of the high-engagement players the model was able to identify. The F1-score combines precision and recall. AUC-ROC shows how well the model separates the engagement classes, and scores closer to 1 indicate stronger performance. Together, these metrics help confirm that the models are reliable and are not missing important patterns.

Hyperparameter Tuning

Hyperparameter tuning is an important part of building the models because the parameters control how the model learns. A hyperparameter is a non-learning parameter that is defined before the learning process starts. For example, in a decision tree, hyperparameters affect the model's accuracy, generalization, and robustness (GeeksforGeeks, 2024b). The performance of SVM also depends on the appropriate choice of hyperparameters, such as the kernel, C, and gamma, because they determine how strictly the model separates the engagement classes and how sensitive it is to data patterns (GeeksforGeeks, 2024a). Tuning these hyperparameters helps the models perform better, reduce overfitting, and generalise well to new players (GeeksforGeeks, 2024b).

Clustering

Clustering is an unsupervised learning method that groups data points based on their similarities. K-Means Clustering was applied to identify different engagement levels of each player. K-Means calculates the distance between each data point and a set of centroids, and assigns each point to the closest centroid (Joseph, 2025). The aim is to form k distinct groups and minimise the total distance between each point in a cluster and its corresponding centroid (Joseph, 2025).

4. Experiment

Models

The dataset is loaded into a Jupyter Notebook and explored using .info(), .nunique(), and print(data) to understand the data types and identify unnecessary columns. PlayerID, Gender, Location, and GameGenre do not have a meaningful impact on engagement levels, therefore they were removed. Features such as Age, PlayTimeHours, SessionsPerWeek, AvgSessionDurationMinutes, and PlayerLevel were all loaded in the independent variable X, while EngagementLevel was loaded into the dependent variable y. The dataset was divided into

two sets, with 70% of the data being in the training set and 30% of the data in the test set. As well, the use of stratification maintained the division between classes to balance the engagement levels in each set.

A decision tree classifier was trained on the training set, with a maximum depth of 2 to avoid overfitting, to understand how the model splits the data. To ensure the model is correct, dtree.predict() is used to generate an output of player engagement based on inputs.

StandardScaler standardized the SVM putting all features on the same scale so that variables with large numeric values do not dominate the distance calculations used by SVM. A linear-kernel SVM was then trained on the scaled training data, and the performance was evaluated using accuracy, precision, recall, F1-score, and AUC-ROC.

To improve performance, hyperparameters for both models were tuned using GridSearchCV, which systematically evaluates multiple combinations and selects the best-performing one based on the F1-macro score (GeeksforGeeks, 2024b).

For the decision tree, the hyperparameter testing included max_depth, min_samples_split, min_samples_leaf, and the splitting criterion (gini or entropy).

For SVM, different values of C, gamma, and RBF kernels were evaluated.

The best models were retrieved using best_estimator_, and their parameters were reported.

K-means

To determine the correct number of clusters, two techniques were applied—the Elbow Method and the Silhouette Score. Based on the evaluations, the k that gave the most meaningful structure was selected and used for the final k-means model. Cluster labels were then assigned to each player and added to the dataset. To visualize the clusters, Principal Component Analysis (PCA)

was applied to reduce the data to two dimensions allowing to identify the players based on their behaviors.

5. Results

Decision Tree and SVM Performance

Both decision tree and support vector machine models were evaluated on the test set using accuracy, precision, recall, F1 score, and AUC-ROC.

```
Decision Tree Performance on Test Set:
Accuracy:  0.8105902922321205
Precision:  0.8013195214537182
Recall :  0.8037003292490454
F1 score:  0.8024843484587408
ROC-AUC Score: 0.8519511838205723


SVM Performance on Test Set:
Accuracy:  0.816418283240363
Precision:  0.8220298033853783
Recall :  0.8019175602735086
F1 score:  0.8105999834369545
ROC-AUC Score: 0.8494133781407688
```

Figure 1. Shows the initial untuned model performances

```
Tuned DT Performance on Test Set:
Accuracy:  0.8796936141869952
Precision:  0.8796477458200883
Recall :  0.8708826660626402
F1 score:  0.8749894139232423
ROC-AUC Score: 0.9025635011136641

Tuned SVM Performance on Test Set:
Accuracy:  0.8926817084339356
Precision:  0.896361944110994
Recall :  0.8812845502378036
F1 score:  0.8881749639131904
ROC-AUC Score: 0.910648816611285
```

Figure 2. Shows the model performances after tuning

After tuning the Decision Tree and SVM, there is a difference in performances, noticeably the tuned model performed better.

K-Means

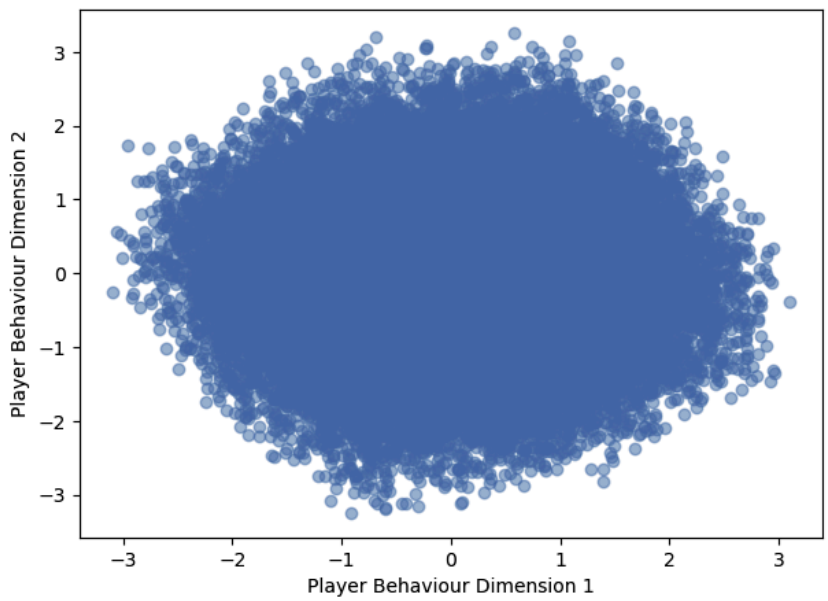

Figure 3. PCA before Clustering

In Figure 3, PCA shows the 2D scatterplot of the behaviors of the players within the dataset.

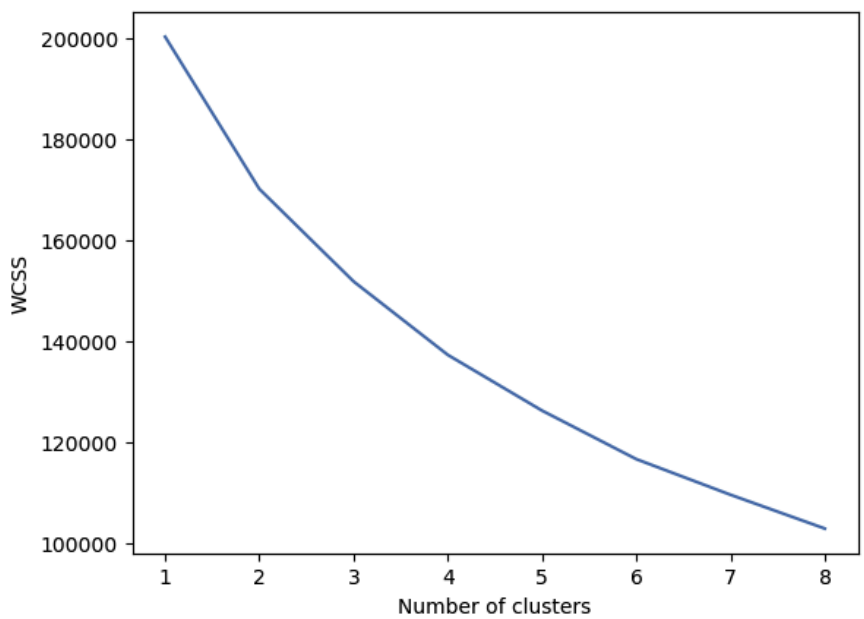

Figure 4. Elbow Method

In Figure 4, the most significant shift is around k = 2 and k = 3, suggesting that two clusters provide the most apparent separation in player behavior patterns.

```
K=2 | Silhouette=0.148
K=3 | Silhouette=0.135
K=4 | Silhouette=0.142
K=5 | Silhouette=0.143
```

Figure 5. Silhouette Method Results

In Figure 5, the silhouette closest to 1 is best, being K = 2.

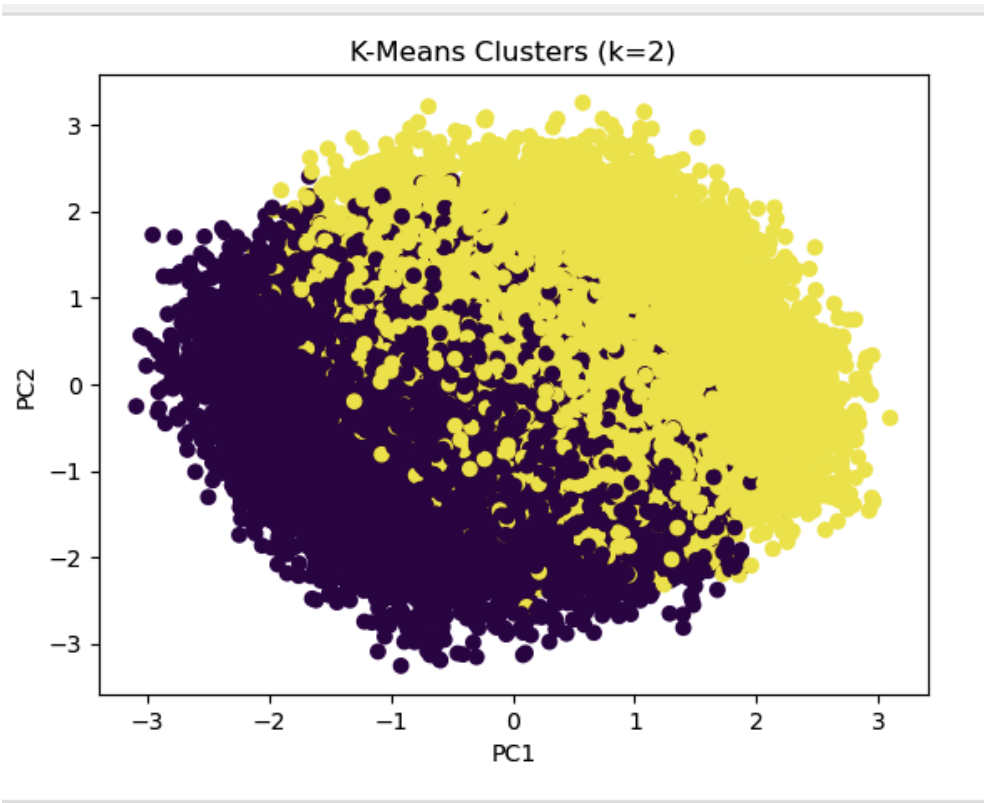

Figure 6. PCA after K-Mean

Figure 6 shows the PCA representation of K-means clustering with k = 2. The PCA scatterplot shows two overlapping but distinct behavioral groups, suggesting two primary player types based on playtime patterns and session activity.

```
Cluster counts:
Cluster
0    20088
1    19946
Name: count, dtype: int64
```

Figure 6 Cluster Count

- Cluster 0 : Higher playtime, more frequent sessions, longer session duration
- Cluster 1 : Lower activity levels and shorter sessions

These clusters show the differences in player behaviour independent of the engagement labels.

6. Discussion

The Decision Tree and SVM models both performed well at predicting player engagement, but the tuned SVM performs the best. This suggests that player engagement patterns in the dataset are not linear and benefit from a SVM model to find stronger separation boundaries between classes.

K-means clustering identified two main behavioral groups among players. Although the original labelling was low, medium, and high engagement, the unsupervised method naturally grouped players into two clusters based on similarities in playtime and activity features. This shows that player behavior forms two dominant patterns in the database, highlighting qualities that the supervised models were unable to capture.

7. Conclusion

This project explored how machine learning algorithms can help a gaming studio understand and predict player engagement. Decision Tree and SVM were trained on demographic and game player features. After tuning, the SVM delivered the best performance. K-means clustering identified two main behavioral groups of players. Overall, machine learning is effective in

supporting player retention strategies by identifying what behaviors drive engagement and how different types of players interact with the game.

8. Recommendations

1. Use SVM for engagement prediction, as it performed the best in identifying player engagement levels.
2. Player Behavior may change over time so make sure to re-run clustering.
3. Use Decision Trees for simple data points.

References

Ahiabah-Quarshie, S. (2024, March 25). *Analyzing and categorizing football player roles through clustering*. An Educational Football Data Science Blog and Analytics Platform; PITCH IQ. https://steveaq.github.io/Player-Roles-Clustering/

Deep-Dive into AI/ML to Enhance Customer Engagement. (2023, October 9). https://webengage.com/blog/ai-ml-to-enhance-customer-engagement/

Gaming Analytics: Unveiling Insights for Player Retention and Monetization. (2024). Castordoc.com. https://www.castordoc.com/ai-strategy/gaming-analytics-unveiling-insights-for-player-retention

GeeksforGeeks. (2017a, May 2). *K means Clustering – Introduction*. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/k-means-clustering-introduction/

GeeksforGeeks. (2017b, December 16). *Decision Tree in Machine Learning*. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/decision-tree-introduction-example/

GeeksforGeeks. (2024a, April 12). *How SVM constructs boundaries?* GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/how-svm-constructs-boundaries/

GeeksforGeeks. (2024b, April 16). *How to tune a Decision Tree in Hyperparameter tuning*. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/how-to-tune-a-decision-tree-in-hyperparameter-tuning/

Joseph, S. (2025). *Unsupervised learning* [PowerPoint slides]. Artificial Intelligence and Machine Learning, University of Wales Trinity Saint David.

Necula, S.-C. (2023). Exploring the Impact of Time Spent Reading Product Information on E-Commerce Websites: A Machine Learning Approach to Analyze Consumer Behavior. *Behavioral Sciences*, *13*(6), 439. https://doi.org/10.3390/bs13060439

Rabie El Kharoua. (2024). *Predict Online Gaming Behavior Dataset*. Kaggle.com. https://www.kaggle.com/datasets/rabieelkharoua/predict-online-gaming-behavior-dataset?resource=download