UNLOCKING BTD6 PLAYER INSIGHTS!

AN ANALYTICAL APPROACH TO ENHANCE THE GAME, BOOST PLAYER ENGAGEMENT, AND OPTIMIZE MONETIZATION

NINJA KIWI

PREPARED BY
VIGNESH JAIRAM #VJAI535

- BUSINFO 718: FINTECH ANALYTICS INDUSTRY PROJECT - PROJECT REPORT

- 5941 WORDS
- 6 JUNE 2024

EXECUTIVE SUMMARY

Purpose

By understanding player behavior and preferences, Ninja Kiwi aims to enhance player engagement and business growth for its flagship game, BTD6. They face challenges in retaining players and converting them into paying customers. The core research questions focus on clustering players based on psychographics and identifying key factors influencing player churn and in-app purchases. These insights are crucial for developing tailored strategies to boost player satisfaction, retention, and revenue.

Methodology

The project employed a comprehensive approach involving exploratory data analysis, feature engineering, and data cleaning in SQL, clustering analysis in SPSS and Python, and predictive modelling – binary classification in Python. Data from multiple tables were pre-processed and analyzed. Feature engineering was performed to create relevant variables to be used as inputs to the machine learning models. After segregating players based on their playing ranks, clustering was conducted using Python K-means and SPSS Two-step clustering algorithms to select the optimal model. Churn and IAP predictions utilized logistic regression, random forest, and gradient boosting models to pick the best-performing model among them.

Key Findings

Clustering Analysis:

- Two-step clustering marginally outperformed K-means clustering in the average Silhouette Score across the models.
- Four distinct clusters emerged among beginners, highlighting varied spending on continues, heroes, and event game modes.
- Intermediate player clusters showed differences in spending on heroes and power abilities, with a focus on boss events.
- Experienced player clusters revealed high spending on heroes and power items, with balanced game mode preferences.

Churn Prediction:

- Gradient boosting outperformed logistic regression and random forest, achieving the highest ROC-AUC score.
- Important variables included session duration, number of maps played, and in-game spending on heroes and power items.

.

IAP Prediction:

- Gradient boosting again showed superior performance in predicting IAP behaviors.
- Critical variables included in-game spending on hero upgrades, power items, and current player rank.

Recommendations

Segmentation Strategies:

- For beginners, offer tutorials on gameplay strategies and introduce a "Continue" package to encourage microtransactions.
- Adjust game difficulty at the intermediate level to improve win rates and promote non-standard game modes.
- Among experienced players, engage explorers with new content and achievers with more power items and upgrades.

Churn Reduction Strategies:

- Enhance session engagement with daily rewards and challenges.
- Diversify game modes and maps to maintain player interest.

IAP Promotion Strategies:

- Promote bundles and limited-time offers to boost in-game earnings and spending.
- Reward diverse gameplay experiences and rank-based achievements.

Implementation

Integrate the chosen two-step clustering and gradient boosting models into BTD6's backend, using batch processing for periodic updates and real-time deployment through APIs. Automated model retraining and analytics dashboards will ensure continuous performance monitoring and actionable insights.

TABLE OF CONTENTS

1. INTRODUCTION

Ninja Kiwi is a leading game development company that has been creating engaging and innovative games for nearly two decades. Known for their emphasis on player enjoyment and long-term engagement, Ninja Kiwi has developed a strong portfolio of popular games, with Bloons TD 6 (BTD6) being their flagship title. BTD6 is a widely recognized tower defense game available on multiple platforms, providing players with various game modes, levels, and in-app purchase (IAP) options. The company's core activities include game development, player experience enhancement, and continuous content updates to maintain player interest and satisfaction. Through fun gameplay mechanics and strategic challenges, Ninja Kiwi has built a dedicated player base and a reputation for high-quality game experiences. The company's commitment to creating games that they would personally enjoy playing has driven their success and positioned them as a significant player in the gaming industry.

The business problem involves understanding the diverse ways players interact with BTD6 without compromising their privacy. Ninja Kiwi does not track demographic data, which makes it challenging to identify different player segments and tailor content to their preferences. The company aims to group players based on their psychographics, which reflect their gameplay behaviors and preferences, rather than demographic information. This project seeks to define these psychographic groups, analyze their behavior concerning IAP and churn, and propose strategies to enhance player satisfaction and business growth. The tasks undertaken to achieve these objectives include clustering players based on their psychographics, developing predictive models to forecast churn and IAP, and identifying influential factors that cause these events. By focusing on psychographics, Ninja Kiwi hopes to understand the motivations and preferences of their players, leading to more targeted and effective game development strategies.

Our project's scope encompasses analyzing player behavior data to answer three critical questions. First, we aim to group players based on their psychographics to facilitate the development of game content that enhances player satisfaction and contributes to business growth. Second, we seek to identify the factors that cause players to churn or become inactive. Understanding these factors will enable Ninja Kiwi to develop strategies to retain players and reduce churn rates. Third, we aim to determine the factors that influence whether a player makes an IAP. This insight will help create more compelling in-game purchases that appeal to different player segments. This report is structured to provide a comprehensive roadmap of our approach, including the project background, methodology, results, findings, recommendations, and conclusion. By addressing these questions, we aim to provide actionable insights to help Ninja Kiwi optimize their game content and improve player retention and monetization.

2. PROJECT BACKGROUND

Ninja Kiwi assigned our team this project to unlock valuable insights into player behavior and drive business growth. Despite its importance, the project had been on the back burner due to the company's focus on other pressing issues. Our team, consisting of five members with varying strengths, was well-suited to tackle this challenge. Though we had no experience with BTD6, we were enthusiastic about learning the game to ensure insightful and actionable results.

2.1 Industry Background and Challenges

The gaming industry is highly competitive and characterized by rapid technological advancements and diverse game genres. With the proliferation of mobile and online gaming, companies must constantly innovate to retain players and monetize their games effectively. Ninja Kiwi operates in the tower defense genre, a niche but popular segment known for its strategic gameplay and replayability.

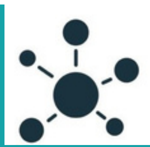
Ninja Kiwi faces competition from other prominent developers in the tower defense and broader mobile gaming markets. Direct competitors include Ironhide Game Studio (known for the Kingdom Rush series) and Supercell (creator of Clash of Clans). These companies share similar business models focusing on free-to-play games with IAP and continuous content updates to engage players.

2.2 Business and Revenue Models

The business models in the gaming industry often revolve around free-to-play games with monetization through IAP, advertisements, and premium content. Ninja Kiwi follows this model (Burrell, 2021), offering BTD6 as a free-to-play game with optional purchases for hero upgrades, unique skins, and other in-game items. This approach is designed to maximize player base growth while generating revenue from dedicated players willing to spend on enhancing their gameplay experience.

2.3 Research Questions and Project Motivations

Understanding the nuances of player behavior is crucial for enhancing the game, engaging players, and optimizing revenue streams. The questions our analysis seeks to answer are as follows:

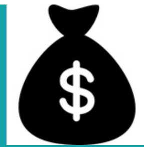


Clustering

1. How do we group/cluster players based on their psychographics, which would facilitate the development of game content that enhances player satisfaction and contributes to business growth?

Binary Classification

2. What are the factors that make players churn/inactive?



Binary Classification

3. What are the factors that determine if a player makes an in-app purchase or not?

Answering these research questions can add significant value to Ninja Kiwi:

- **Psychographic Grouping:** By clustering players based on psychographics, Ninja Kiwi can tailor game content and marketing strategies to specific player segments. This personalization can enhance player engagement and satisfaction, increasing retention and spending.

- **Churn Prediction:** Identifying the factors that cause players to churn allows Ninja Kiwi to implement targeted retention strategies. For example, personalized offers and loyalty rewards can be designed to keep high-risk players engaged, reducing churn rates and maintaining a stable player base.

- **IAP Prediction:** Understanding what drives players to make in-app purchases enables Ninja Kiwi to optimize their IAP offerings. Optimized IAP strategies based on player behavior can increase conversion rates and revenue. By focusing on features that are most likely to convert players into spenders, Ninja Kiwi can increase their revenue without alienating their player base.

2.4 BTD6 Game Mechanics and Key Variables

Bloons TD 6 is a tower defense game where players use monkey-themed towers to prevent balloons ("bloons") from reaching the end of a defined track. Players can choose from various game modes, levels, and difficulties, each providing different challenges and rewards. Key mechanics include:

- Tower Placement: Players strategically place towers to pop bloons.
- Upgrades: Towers can be upgraded using in-game currency for better performance.
- Heroes and Skins: Players can unlock and customize heroes with unique abilities.
- Game Modes: Different modes (e.g., standard, co-op, challenge) offer varied gameplay experiences.

Key variables such as platform, game modes, player level, IAP status, cosmetic interactions, days spent in the game, and game difficulty are crucial in understanding how players engage with BTD6. These variables capture the essence of player behavior and preferences, providing insights into their engagement, spending habits, and overall interaction with the game. Analyzing these variables allows us to tailor strategies that align with player needs and enhance the gaming experience.

2.5 Bartle's Taxonomy

Bartle's Taxonomy, which classifies players into four types (Achievers, Explorers, Socializers, and Killers), isn't ideally suited for BTD6 due to its focus on social interactions and player-vs-player dynamics, which are less prominent in BTD6. However, we have utilized certain aspects of Bartle's Taxonomy to understand player motivations and behaviors. For instance, identifying Achievers who focus on progressing and optimizing gameplay aligns with our spending patterns and engagement analysis. This adapted approach helps in tailoring content and marketing strategies to enhance player satisfaction and retention despite the taxonomy's broader limitations for BTD6.

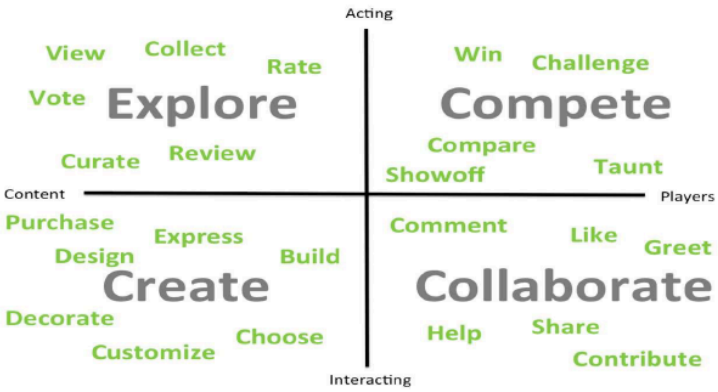


Figure 1: A chart depicting Bartle's Taxonomy

3. METHODOLOGY AND ANALYSIS

3.1 Basic Outline and Steps

Our project followed a meticulously structured approach to address the business problem and achieve the defined objectives. We began by familiarizing ourselves with BTD6, which involved reading about the game mechanics, rules, and various features (McClure, 2022) and playing the game to gain a deeper understanding of player interactions. We then connected to Ninja Kiwi's PostgreSQL database using DBeaver, enabling us to access the necessary data stored in 15 initial tables provided by the company. Our initial task involved conducting Exploratory Data Analysis in SQL on these tables, examining their structure, contents, and relationships to identify key variables and uncover initial insights. This EDA helped us better define the three critical questions our project aimed to answer.

Considering these questions, we brainstormed and conceptualized a template for the final table most appropriate for our analysis, including variables deemed crucial based on our EDA findings. Using SQL, we performed feature engineering to populate this final table, creating new features such as average playtime, purchase frequency, and interaction with game modes. We then carried out data cleaning to handle missing values and remove duplicates. After dealing with outliers, the final table in the SQL database was downloaded as a CSV file. This CSV file was exported into SPSS Statistics to perform clustering. From the final table in the SQL database, we loaded a representative random sample of 100,000 instances into Python for machine learning, making the data manageable for intensive computational tasks. Another round of EDA was performed in Python to explore the sampled data and to check if it accurately reflects the characteristics of the population, including visualizing distributions and identifying correlations. Data preprocessing followed, involving the removal of unnecessary columns and scaling the data to prepare it for machine learning algorithms.

To address the first research question on clustering players based on psychographics, we evaluated two clustering algorithms, one in SPSS and the other in Python, to select the best-performing one. For the chosen cluster model, detailed data analysis and visualization were conducted for each cluster to provide actionable insights. In the Python cluster model, the resulting cluster labels were used to fit a Random Forest Classifier, allowing us to obtain feature importance and differentiate the clusters. To predict if a player would churn or make an IAP, we developed binary classification models for each question in Python by trying three different algorithms and then selected the best ones based on evaluation metrics. Recursive Feature Elimination (RFE), hyper-parameter tuning, and cross-validation were used to refine the chosen models and finally check for their performance.

3.2 Data Sources

For our analysis of BTD6, the company provided us with a comprehensive dataset comprising 15 tables, each detailing various aspects of player interactions and game mechanics. These tables are foundational to our project, allowing us to perform in-depth analyses and derive insights that address the company's objectives. Each table offers unique insights into different aspects of player behavior, from initial engagement and gameplay strategies to spending habits and progression.

The btd6_startsession table includes information on the platform and game version, capturing the initial conditions under which players begin their gaming sessions. The btd6_startrack and btd6_endtrack tables log each instance in which a player starts a new map and the results of their gameplay, respectively. These tables are crucial for understanding player engagement and performance across different sessions. The btd6_placetower table records every instance of tower placement, providing insights into players' strategic decisions during gameplay.

Advanced gameplay actions are captured in the btd6_made5 and btd6_madeparagon tables, which track the creation of tier 5 and paragon towers, respectively. These tables help identify skilled players and their progression. The btd6_useactivatedability and btd6_usepower tables log the use of abilities and powers within the game, shedding light on player strategies and resource usage.

Monetary transactions within the game are tracked in the btd6_gainmonkeymoney and btd6_spendmonkeymoney tables, which record earnings and expenditures of in-game currency. Similarly, the btd6_maintrophies and btd6_spendtrophies tables track the gaining and spending of trophies, which unlock specific game features. The btd6_unlockhero and btd6_unlockheroskin tables log the unlocking of new heroes and hero skins, providing insights into player progression and aesthetic preferences. Finally, the btd6_buyiap table records real-world money purchases made within the game, which is critical for analyzing player spending patterns and monetization strategies.

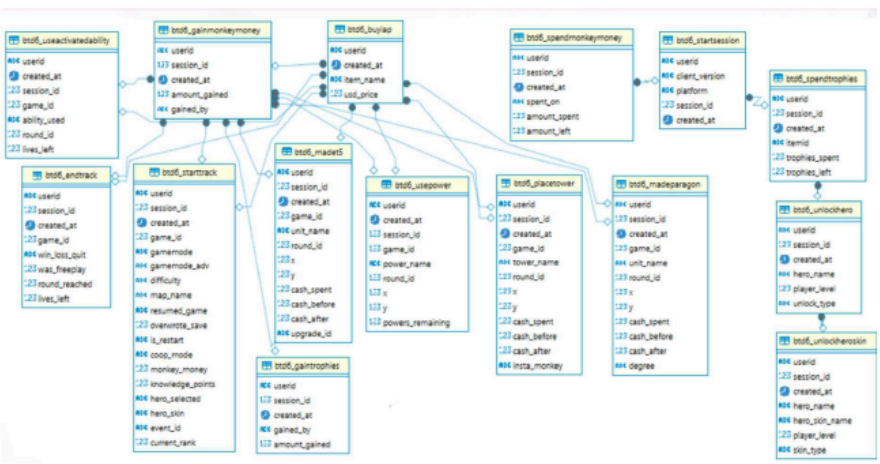


Figure 2: The Entity Relationship Diagram for the 15 data tables provided by Ninja Kiwi (please zoom in to see the columns in each table)

3.3 Feature Engineering and Data Cleaning

Feature engineering involves creating new variables that can enhance the predictive power of machine learning models (please refer to Appendix A1). We utilized the 15 tables provided by Ninja Kiwi to engineer features for the final table. This table served as the basis for our subsequent analyses. The process involved thorough analysis, data manipulation, and the creation of meaningful variables that capture various aspects of player behavior and game mechanics.

We began by extracting relevant data from the 15 tables using SQL. This involved writing complex queries to aggregate, join, and filter data to form the foundation of our feature engineering process. For instance, the btd6_startsession table was used to identify all user IDs and session IDs, providing a basis for linking different aspects of player activity. Similarly, the btd6_startrack and btd6_endtrack tables were utilized to capture the start and end of gaming sessions, essential for calculating session-specific metrics.

The final table has 48 variables, many designed to capture detailed player data. Additionally, we engineered more complex features to capture nuanced aspects of player behavior like transaction proportions, gameplay metrics, and engagement metrics. For example, below are the two target variables for the binary classification problems and the methods used to engineer them:

played_within_30days: Analysis revealed that 7% of users interacted solely within the last 60 days and not in the last 30 days, while 11.4% engaged within the last 30 days. Based on these findings and consultation with the client, we determined the 30-day threshold as the criterion for active users, with the remaining users categorized as churned.

```sql
case

      when (select '2024-02-10'::date - a.max_sessiondate::date) <= 30 then
1

      else 0

end -- 2024-02-10 is the last day we have data for
```

Figure 3: SQL code snippet for feature engineering 'played_within_30days'

iap_flag: This variable was created to indicate whether a player made an in-app purchase (1) or not (0).

```sql
case
    when iap_spend is not null then 1
    else 0
end as iap_flag
```

Figure 4: SQL code snippet for feature engineering 'iap_flag

After engineering these features, we assembled the final table by joining the individual tables on common keys like userid and session_id. This step ensured that each row in the final table represented a comprehensive profile of a player's activity and behavior.

Data cleaning steps, such as handling missing values, dealing with outliers, and scaling, were performed to maintain data integrity. For features where a missing value indicates the absence of an event or condition, zeros were used. For example, for features like num_free_heroes, num_paid_heroes, and trophies_spent, missing values were replaced with 0. To handle outliers in continuous numeric variables, we limit extreme values to the 1st and 99th percentiles. Then, we standardize these variables to ensure that all features contribute equally to the process. The final table thus contained a rich set of features that encapsulate various dimensions of player behavior, engagement, and spending patterns.

```
#Defining numeric variables to deal with outliers and scaling
continuous_numeric_variables=['num_sessions','monkeymoney_gained_excl_iap','avg_moneygained_persession','tropheymoney_gained','avg_trophiesamtgained_persession','avg_days_per_week

#Handling outliers: capping and flooring
for col in continuous_numeric_variables:
    low = df[col].quantile(0.01)
    high = df[col].quantile(0.99)
    df[col] = df[col].clip(lower=low, upper=high)

#Scaling the numeric variables
scaler = StandardScaler()
df[continuous_numeric_variables] = scaler.fit_transform(df[continuous_numeric_variables])
```

Figure 5: Code snippet dealing with outliers and scaling

3.4 Machine Learning

The machine learning process for our project involved several steps, each executed to address the three key questions defined in the project scope. The analysis was performed using SPSS Statistics and Python, leveraging various libraries such as Pandas, Scikit-Learn, and Matplotlib to pre-process the data, build models, and visualize the results.

3.4.1 Clustering in SPSS Statistics

Two-step clustering in SPSS was chosen for its ability to handle mixed data types and large datasets. Players with current rank below 30 were deemed futile to cluster and were excluded. All the others were categorized into three distinct groups based on their current rank as follows:

- Beginners: Players with current rank from 30 to 60.
- Intermediate Players with current rank from 61 to 100.
- Experienced Players: Players with current rank from 101 to 155.

Clustering was initially performed on all players, following which clustering was done for each player category to attain more meaningful segmentation, detailed analysis, and actionable insights. After several trials, we identified seven optimal variables to be used as inputs to cluster:

- Boss_event_prop - Proportion of Boss Event Games out of all games played.
- Event_game_prop - Proportion of Event Games out of all games played.
- Normal_games_prop - Proportion of Standard Games out of all games played.
- Continue_moneyspent_prop - Proportion of spending on Continue out of all monkey money spent.
- Hero_moneyspent_prop - Proportion of spending on Hero out of all monkey money spent.
- Knowledge_moneyspent_prop - Proportion of spending on Knowledge out of all monkey money spent.
- Power_moneyspent_prop - Proportion of spending on Power ability out of all monkey money spent.

3.4.2 Clustering in Python

To make the models comparable, the clustering performed in Python followed the same template as the clustering in SPSS in the sense that it used the same player categories and input variables to cluster. K-means clustering was picked to cluster our data due to its simplicity, efficiency, and scalability in handling large datasets. We initiated the process by selecting a range of cluster numbers and using the ‘Elbow Method’ to determine the optimal number of clusters. After assigning cluster labels, a Random Forest Classifier was fitted to the data to determine input/feature importance. This helped in understanding the key variables that defined each cluster and differentiated one from the other.

3.4.3 Churn Prediction and In-App Purchase Prediction in Python

The two binary classification models empower Ninja Kiwi to proactively address player churn and optimize in-app purchases by identifying at-risk players and potential spenders. Both these classification problems followed a similar approach. We experimented with the three classification algorithms (Demir, 2015) as follows:

Logistic Regression

Selected for its interpretability and efficiency in binary classification problems

Random Forest Classifier

Chosen for its ability to handle non-linear relationships and interactions between variables

Gradient Boosting Classifier

Utilized for its strong predictive performance and ability to optimize errors iteratively.

The target/dependent variable and the independent variables were defined to ensure that no data leakage happened. Played_within_30days and iap_flag were the dependent variables for questions 2 and 3 respectively, with all relevant variables used as independent variables for the three trial models. Afterward, the dataset was split into training and testing sets using a 75-25 split to validate model performance. Each model was trained on the training set using the default hyper-parameters. Finally, the models were evaluated on the test set with the chosen metrics. Keeping in mind our computational limitation, the best-performing models were chosen to further improve upon the result. The following two processes were performed on the best-performing models:

Feature Selection: Recursive Feature Elimination was used to select the 10 most important features to be used as input variables for each model. This step was crucial for improving model performance and interpretability. RFE works by recursively removing the least important features based on model accuracy, building a model on the remaining features, and repeating this process until the desired number of features is selected.

Hyper-parameter Tuning and Cross-validation: GridSearchCV was employed to tune hyper-parameters and perform 2-fold cross-validation (Sharma, 2023), ensuring the models were optimized and generalized. GridSearchCV optimizes hyper-parameters by exhaustively searching through a specified grid of parameter combinations, evaluating each combination using cross-validation, and selecting the best-performing set for the models.

Predictions are made on the test set using the best model, followed by the model being evaluated using the chosen metrics to assess model performance. By following these steps, we ensured a comprehensive and methodical approach to building robust machine learning models that address Ninja Kiwi's business objectives. This process laid the foundation for deriving actionable insights and recommendations based on the model results.

4. RESULTS, FINDINGS AND DISCUSSION

4.1 Exploratory Data Analysis

Most Valuable Item
The in-game currency Monkey Money was bought 1.4 million times, generating a revenue of 7.4 million NZD

Favourite Hero
Quincy leads the way with him being chosen 140 million times

Game Results
35 out of 100 games ended in a loss

Most Played Game Mode
Standard Mode dominates with a 32% share

Our EDA was essential for understanding patterns in player preferences, gameplay styles, session durations, and win rates. This foundational work guided our deeper analyses. Visualizations were pivotal in transforming raw data into a coherent story of player behaviors.

Figures 6 and 7 reveal insights into IAP. The first figure here highlights the spending behaviors of the top 10 players, with expenditures ranging from $2,257 to $4,777 on items such as "vaultmonkeymoney" and "mountainmonkeymoney." The second figure here provides a breakdown of IAP spending and frequency by item category. "Monkey Money" tops total spending at $7,446,473, while "Tower Related" items are the most frequently bought, with 801,480 purchases. These figures underscore the financial commitment players make to specific items, emphasizing their popularity and economic significance.

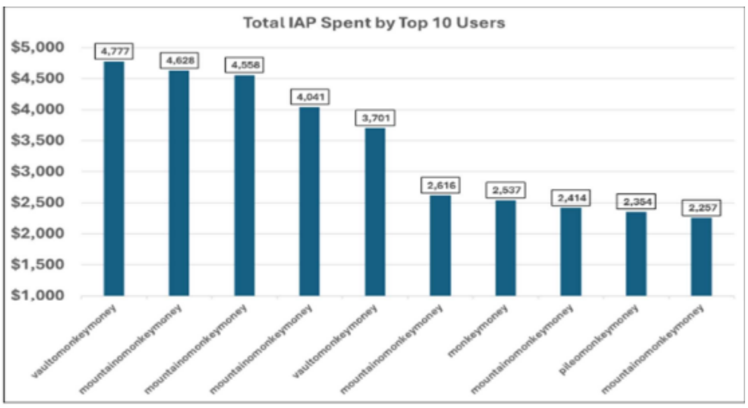


Figure 6: Total IAP spend by top 10 users

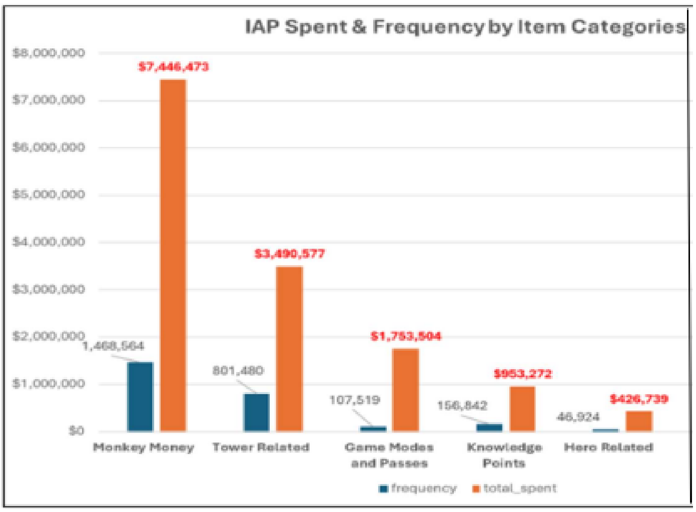


Figure 7: IAP spend and frequency by item categories (please zoom in)

Figure 8 examines game outcomes using data from the endtrack table. Results show that 35.6% of games end in a loss, 27.3% in a win, 25.4% in quitting, and 11.7% in restarting. This indicates that many players experience frequent losses, with a substantial number choosing to quit or restart, pointing to areas for potential enhancement in game design and retention strategies.

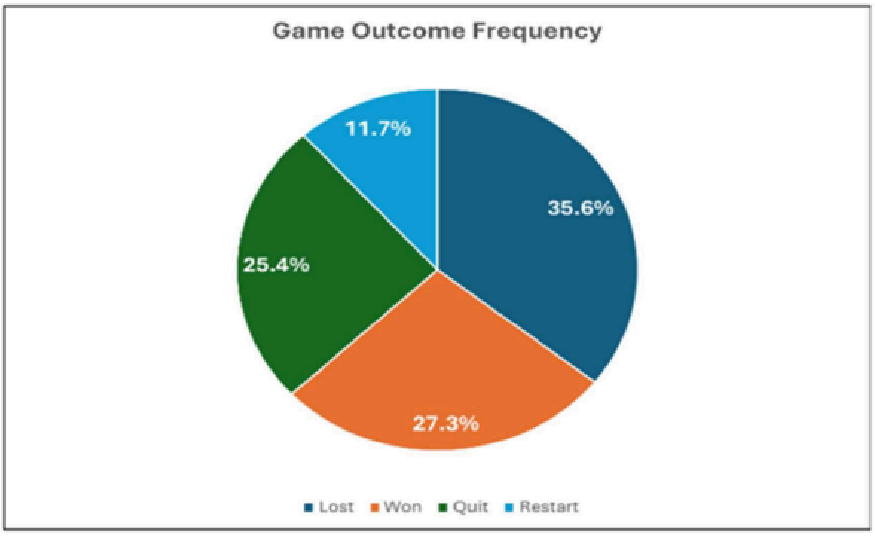


Figure 8: Game outcome frequency

Figures 9 and 10 delve into hero selection and game mode preferences, drawing from the starttrack table. The "Hero Chosen Frequency" chart shows Quincy and ObynGreenfoot as the most selected heroes, followed by Benjamin, Sauda, and Gwendolin. The "Game Mode Played Frequency" chart reveals that Standard mode is the most popular at 31.6%, followed by Modded Standard at 25.1%, and Daily Challenge at 14.8%. Other modes like Player Challenge and Quest also show considerable engagement. These insights highlight player preferences, aiding in the development of targeted content.

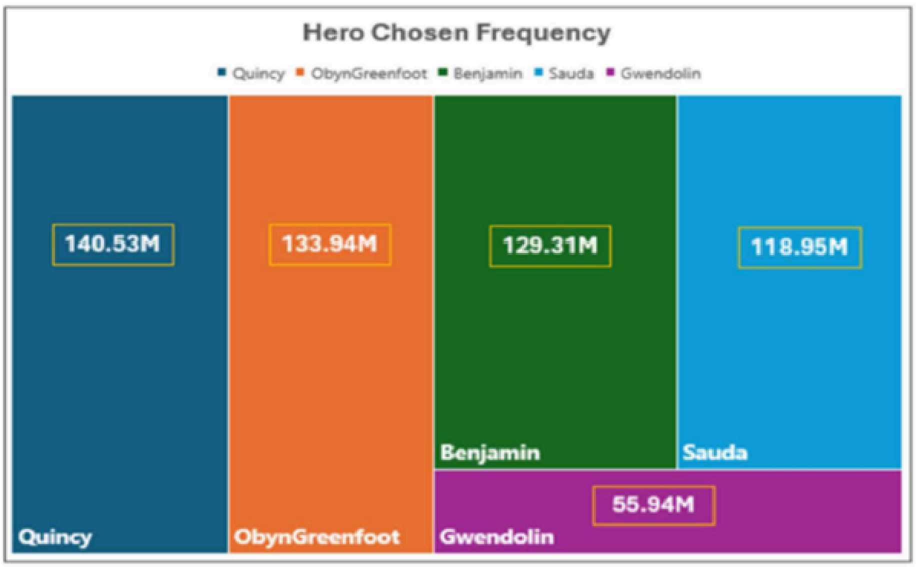


Figure 9: Hero chosen frequency

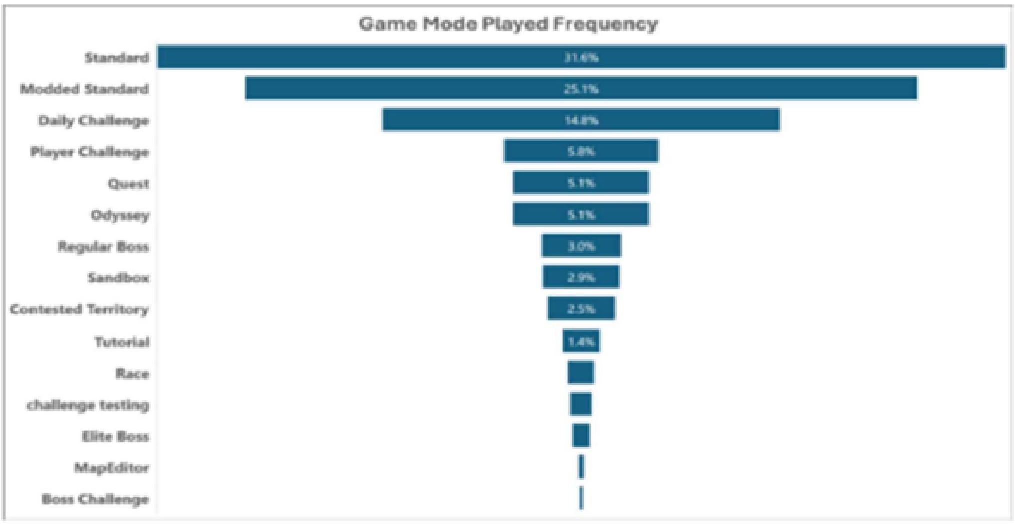


Figure 10: Game mode played frequency (please zoom in)

By analyzing the starttrack and gainmonkeymoney tables (Figure 11), we identified which game modes are the most profitable. Modded Standard leads with 1,907,286 Monkey Money, followed by Standard with 1,124,021, and Daily Challenge with 612,989. This analysis provides clarity on which game modes generate the most revenue, guiding future content updates and monetization efforts.

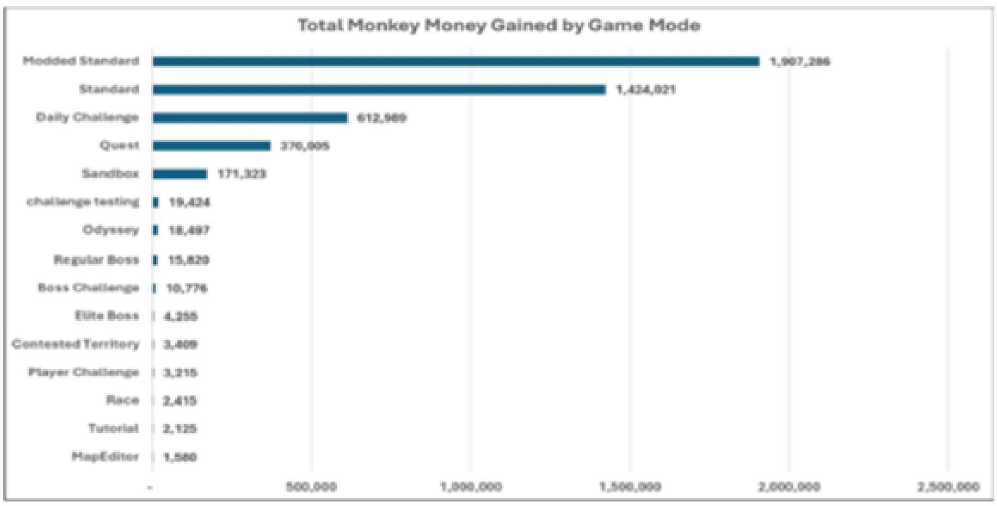


Figure 11: Total Monkey Money gained by game modes (please zoom in)

Further visualizations offered additional insights into player distributions by rank, win rates, and spending patterns. Active players and paying customers tend to cluster around mid-rank levels, with higher ranks showing increased hero skin purchases and steady investment in knowledge points. While normal games dominate, participation in event and boss games is steadily rising across ranks.

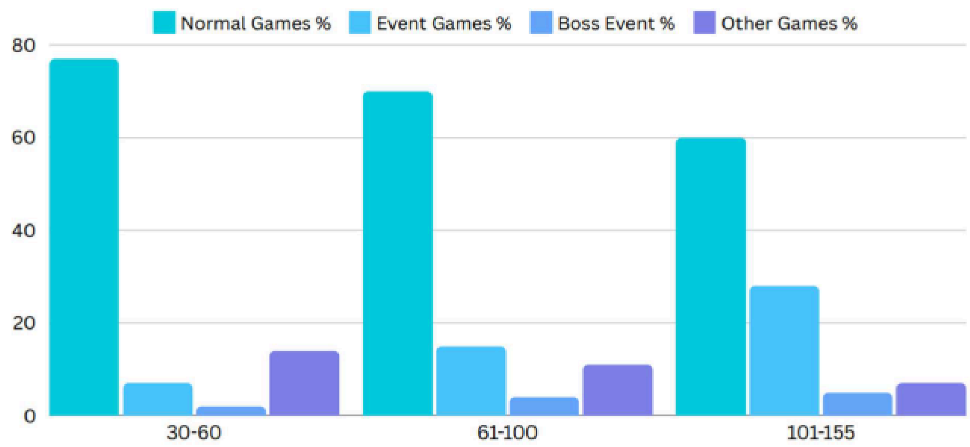


Figure 12: Proportion of game modes by rank

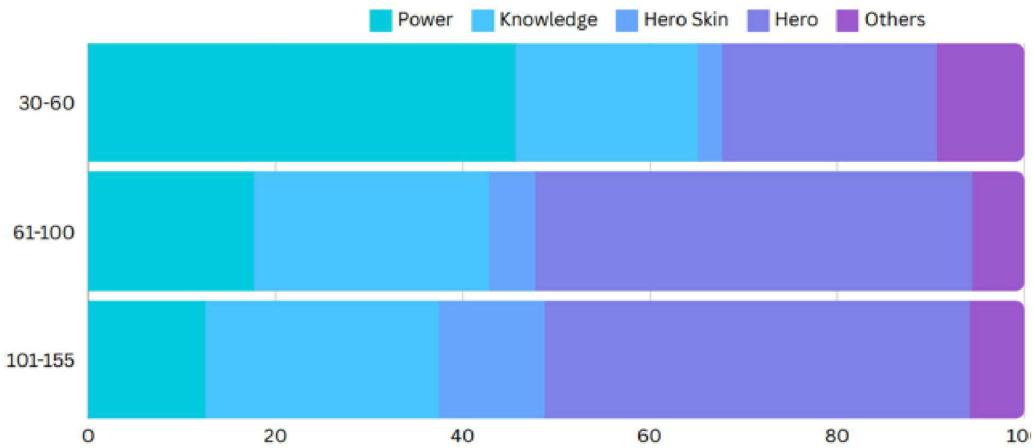


Figure 13: Proportion of Monkey Money spent by rank

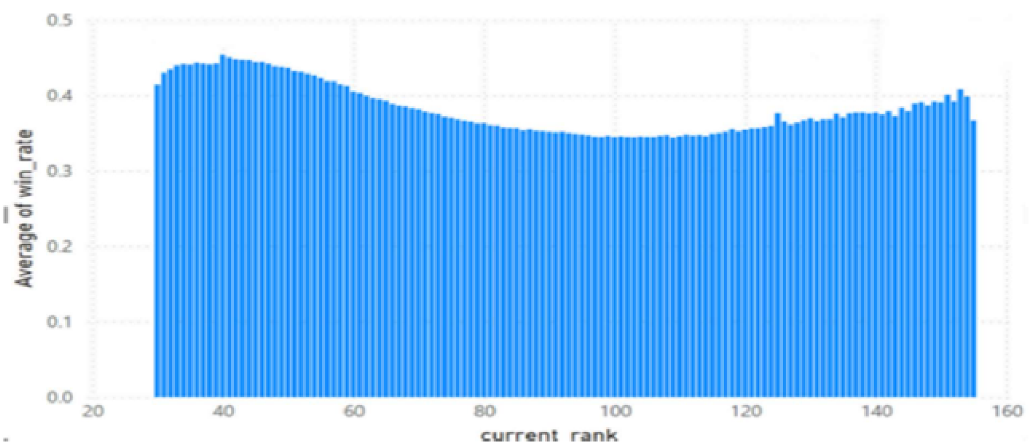


Figure 14: Average win rate by rank

4.2 Clustering Analysis

We performed a detailed clustering analysis comparing the effectiveness of Python's K-means clustering with SPSS's Two-step clustering, focusing on their Silhouette Scores across different player levels. For all players, both methods achieved a Silhouette Score of 0.4, indicating moderate cluster cohesion and separation. However, when broken down by player levels, notable differences emerged. At the beginner level, Python's K-means clustering outperformed SPSS, achieving a Silhouette Score of 0.5 compared to SPSS's 0.4. This suggests that K-means provides more distinct and well-defined clusters for newer players. For intermediate players, SPSS's Two-step clustering had an edge, with a Silhouette Score of 0.4, while Python's K-means scored 0.3. This indicates that SPSS handles the data variability in this player segment better, forming more cohesive clusters. Among experienced players, both methods yielded similar Silhouette Scores of 0.3, reflecting the challenge of clustering this group due to increased variability in player behavior.

These variations can be attributed to the intrinsic characteristics of each clustering algorithm. K-means is effective in minimizing variance within clusters, performing well with well-defined and homogeneous groups. Conversely, SPSS's Two-step clustering excels in managing mixed data types and large datasets, though it may encounter difficulties with noise and outliers. This comparison underscores the importance of selecting an appropriate clustering technique based on the specific characteristics and distribution of the data. The average Silhouette Score of the Two-step clustering marginally outdoes the average of the K-means clustering, leading us to pick the Two-step clustering as the recommended model to be used for further analysis.

Level	Python : K-means Silhouette Score	SPSS: Two-step Clustering Silhouette Score
All	0.4	0.4
30 - 60	0.5	0.4
61 - 100	0.3	0.4
101 - 155	0.3	0.3

Table 1: A table showing the Silhouette Scores of the different cluster models

On performing detailed data analysis of clusters for each player level (please refer to Appendix A2-A9), these are the interesting observations:

All players:

High Spending and Engagement: Players in Clusters 1 and 2 consistently exhibit high real money spending rates and lower churn rates. These players are more involved in non-standard game modes, with Cluster 1 particularly active in event game modes and spending significant in-game currency - monkey money (MM) on Heroes, Knowledge, and Continues.

Evolving Player Behavior: Player behavior evolves as they gain experience. Initially, they focus on standard games to build up their heroes and knowledge. As they advance, they explore a wider variety of game modes, which is reflected in their shifting MM spending habits.

Beginner Players:

Cluster 1 (53.3%): Primarily plays normal games (0.83) and spends significant MM on continues (0.28), suggesting struggles with game difficulty. Low engagement with event and boss events indicates familiarization with basic mechanics.

Cluster 2 (19.1%): Focuses on character development, spending MM on heroes (0.48) and knowledge (0.38), with moderate engagement in event games (0.05).

Cluster 3 (17%): Significant MM spending on power abilities (0.92), primarily playing normal games (0.8) with low event game engagement (0.04).

Cluster 4 (10.5%): Engages more with event games (0.38) and boss events (0.1), indicating early exploration beyond basic gameplay.

I**ntermediate Players:**

Cluster 1 (23.7%): High participation in boss events (0.07) and event games (0.42), with significant MM spending on heroes (0.36) and power (0.14). Lower win rate of 26% suggests prioritizing participation over competitive success.

Cluster 2 (10.9%): Focuses on normal games (0.82) with low IAP spend (1.83) and moderate win rate (43%).

Cluster 3 (37.1%): High MM spending on heroes (0.66) and knowledge (0.14), indicating a balanced approach to game modes and strategic development.

Cluster 4 (28.3%): Significant MM spending on continues (0.3) and participation in normal games (0.81) with a moderate win rate (40%).

Experienced Players:

Cluster 1 (16.5%): Includes the biggest IAP spenders, with balanced game mode preferences and high MM spending on power (0.37).

Cluster 2 (31.8%): Focuses on heroes (0.51), enjoying both strategic and character development aspects.

Cluster 3 (20.5%): Significant MM spending on knowledge (0.23) and heroes (0.51), indicating a developmental focus.

Cluster 4 (31.2%): Moderate MM spending across all categories, high engagement and participation rates, but lower win rate (28%), focusing on enjoying game content rather than achieving competitive success.

Player Level	Beginner (Lvl 30-60)				Intermediate (Lvl 61-100)				Experienced (Lvl 101-155)			
Silhouette Score	0.4 (Fair)				0.4 (Fair)				0.3 (Fair)			
Cluster	1	2	3	4	1	2	3	4	1	2	3	4
Cluster Size	53.3%	19.1%	17.0%	10.5%	23.7%	10.9%	37.1%	28.3%	16.5%	31.8%	20.5%	31.2%
Input												
boss_event_prop	0.01	0.01	0.01	0.1	0.07	0.02	0.02	0.02	0.13	0.03	0.02	0.04
continue_moneyspent_prop	0.28	0.05	0.04	0.17	0.16	0.09	0.08	0.3	0.15	0.1	0.36	0.14
event_games_prop	0.03	0.05	0.04	0.38	0.42	0.07	0.08	0.05	0.25	0.11	0.18	0.55
hero_moneyspent_prop	0	0.48	0	0.15	0.36	0.1	0.66	0.06	0.22	0.51	0.11	0.44
knowledge_moneyspent_prop	0.02	0.38	0.02	0.07	0.16	0.72	0.14	0.08	0.13	0.22	0.23	0.19
normal_games_prop	0.83	0.81	0.8	0.36	0.38	0.82	0.79	0.81	0.54	0.8	0.71	0.35
power_moneyspent_prop	0.06	0.06	0.92	0.17	0.14	0.04	0.05	0.28	0.37	0.04	0.06	0.05
Evaluation Fields												
iap_flag	3.5%	13.9%	5.0%	6.4%	18.7%	8.8%	19.4%	16.1%	38.8%	24.8%	23.0%	24.0%
iap_spend	0.44	2.09	0.91	1.07	4.8	1.83	3.82	4.42	24.26	6.4	10.14	6.64
played_within_30days	14.6%	11.0%	11.4%	12.5%	30.6%	24.7%	28.7%	31.8%	50.2%	50.3%	45.4%	53.5%
avg_games_per_week	9.41	14.58	9.66	15.8	33.81	22.12	22.84	16.48	36.94	38.3	35.94	61.68
current_rank	41.26	49.99	41.44	45.23	77.65	73.05	75.3	72.44	120.48	114.49	120.57	118.33
win_rate	44%	48%	45%	32%	26%	43%	41%	40%	33%	41%	40%	28%

Table 2: A summary of the Two-step cluster models' analysis (please zoom in)

4.3 Churn Prediction

Logistic Regression had lower precision and recall, resulting in a low F1 score and a moderate ROC-AUC score. This is due to its linear nature, which struggles with capturing complex patterns in the data. Random Forest performed better, leveraging multiple decision trees to improve accuracy and precision. However, its recall and F1 score were only slightly improved, indicating it could still miss some patterns due to overfitting on the training data, despite being more robust against overfitting than Logistic Regression. Gradient Boosting emerged as the best, showing superior performance across all metrics. This model effectively handles complex data patterns through iterative improvements, which helps in capturing subtle patterns that other models might miss.

The ROC-AUC score is essential for our specific binary classification tasks as it effectively balances true positive and false positive rates, ensuring robust performance across all decision thresholds. Despite a decrease in the ROC-AUC score to 0.640518 after feature selection, hyper-parameter tuning, and cross-validation, Gradient Boosting was chosen due to its overall balance and robustness. The slight performance drop is attributed to the exclusion of some important variables during feature selection and the specific hyper-parameter values used. The detailed tuning and validation process, including grid search and cross-validation, ensured the model's robustness by minimising overfitting and enhancing generalisation. This comprehensive approach confirmed Gradient Boosting as the best choice for churn prediction in this context, balancing complexity and interpretability with robust performance.

	Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC Score
0	Logistic Regression	0.82692	0.618009	0.270259	0.376063	0.788716
1	Random Forest	0.83772	0.694135	0.284560	0.403645	0.806657
2	Gradient Boosting	0.83812	0.693340	0.289119	0.408074	0.815608

Table 3: Evaluation metrics of the three models for churn Prediction

```
Gradient Boosting Classifier Evaluation Metrics:
Accuracy: 0.83948
Precision: 0.6810883140053524
Recall: 0.31647668393782385
F1 Score: 0.43214942691382485
ROC-AUC Score: 0.6405183915352068
```

Figure 15: Evaluation metrics of the fine-tuned Gradient Boosting Classifier model for churn prediction

Identifying the key variables for churn prediction allows for the development of precise, targeted retention strategies, such as personalized incentives and exclusive in-game rewards to keep players engaged. Understanding these critical factors helps optimize resource allocation, ensuring focused investments in-game features and improvements that directly address the reasons for player churn. Figure 16 shows the 10 most important variables used as input to the churn classification model.

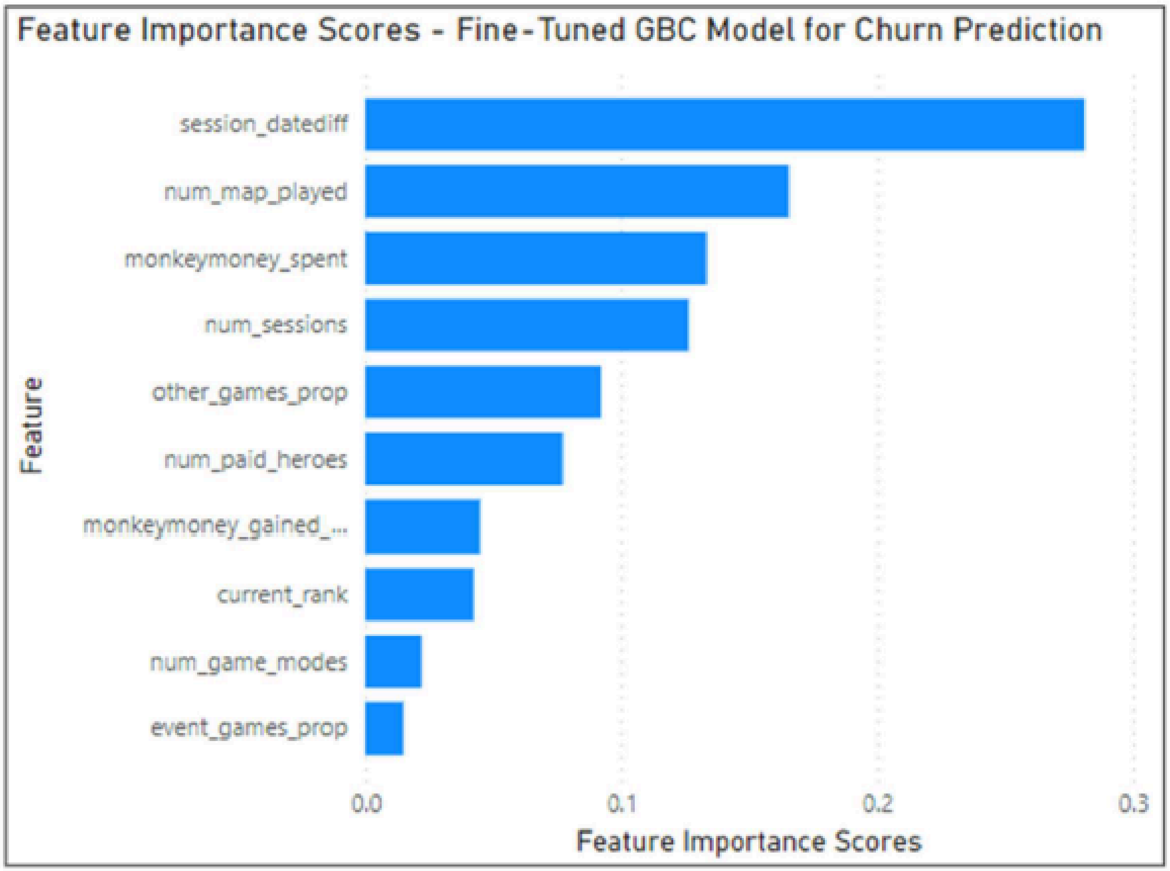


Figure 16: Top 10 most important variables used as input to predict player churn and their importance scores

4.4 IAP Prediction:

Despite its simplicity and efficiency, Logistic Regression's low recall and F1 score indicate it struggles with the imbalanced nature of the dataset, failing to identify a significant number of positive instances (players making an IAP). The Random Forest classifier showed an improved performance, The higher precision and recall demonstrate Random Forest's robustness in handling class imbalance by effectively capturing the minority class (IAP instances). The initial Gradient Boosting model without hyperparameter tuning indicates a strong balance between precision and recall.

After feature selection, hyperparameter tuning, and cross-validation, the chosen Gradient Boosting model's performance showed some decline in ROC-AUC to 0.587963 but improved recall and F1 score, suggesting the model became better at identifying positive instances at the cost of some overall discriminatory power. This decline can be attributed to the complexity of the hyperparameter space and the stringent feature selection process, which might have omitted some influential variables, highlighting the trade-off between model complexity and interpretability.

	Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC Score
0	Logistic Regression	0.90000	0.495614	0.045236	0.082905	0.787974
1	Random Forest	0.91040	0.728723	0.164532	0.268452	0.834703
2	Gradient Boosting	0.90928	0.695578	0.163731	0.265068	0.838769

Table 4: Evaluation metrics of the three models for IAP prediction

```
Gradient Boosting Classifier Evaluation Metrics:
Accuracy: 0.90948
Precision: 0.6690647482014388
Recall: 0.18614891913530823
F1 Score: 0.291262135922330l
ROC-AUC Score: 0.5879638027371503
```

Figure 17: Evaluation metrics of the fine-tuned Gradient Boosting Classifier model for IAP prediction

Knowing the most important variables for IAP prediction enables the creation of targeted marketing campaigns, personalized offers, and promotions that effectively convert non-spenders into spenders, maximizing revenue. These insights help optimize strategic planning and resource allocation, allowing for the development of in-game content and offers that resonate with players most likely to make purchases, driving overall business growth. Figure 18 shows the 10 most important variables used as input to the IAP classification model.

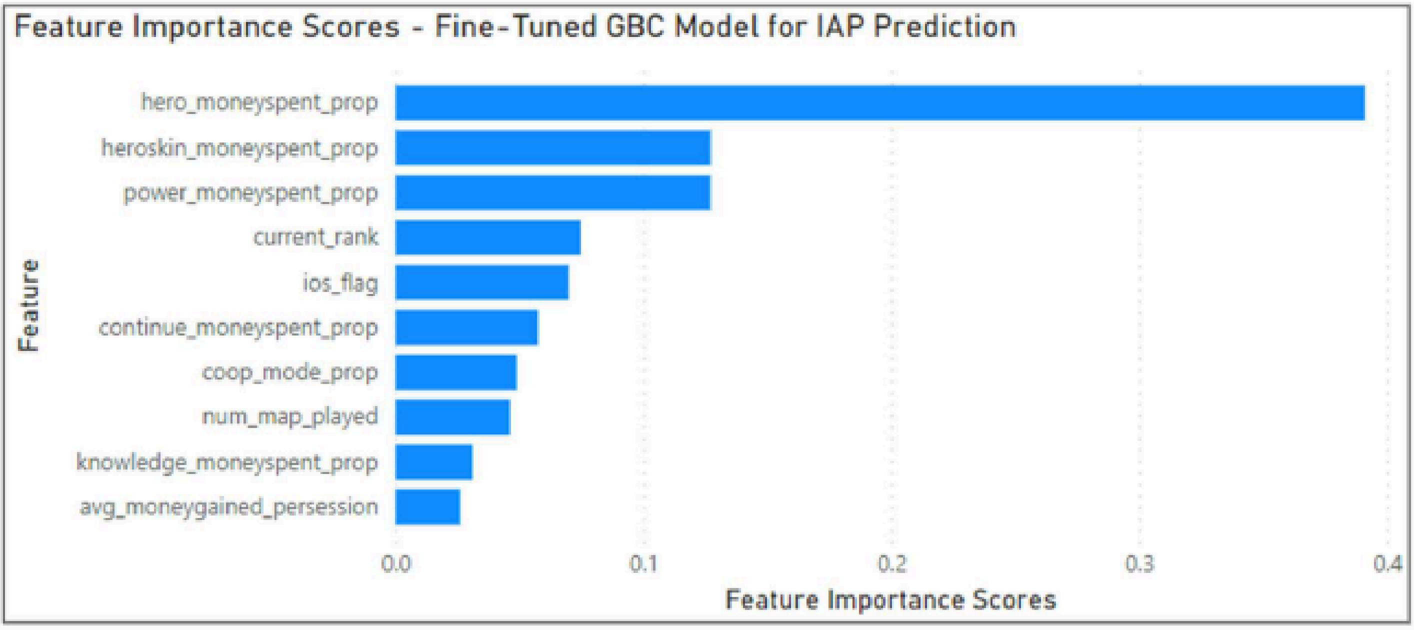


Figure 18: Top 10 most important variables used as input to predict IAP and their importance scores

5. RECOMMENDATIONS AND CONCLUSION

5.1 Recommendations

The table below presents the recommendations for Ninja Kiwi based on our findings:

Basis	Recommendations	Rationale
All Player Clusters	Focus on Clusters 1 and 2, which demonstrate high real-money spending and lower churn rates. Develop personalized retention campaigns featuring exclusive in-game rewards, tailored challenges, and special events to maintain engagement among these high-value players.	Clusters 1 and 2 consistently show higher spending and lower churn rates, indicating their higher engagement and value to the game. Targeting these clusters with personalized retention campaigns will help sustain their activity and spending, leveraging their existing high engagement levels. The data shows these clusters are more involved in non-standard game modes and spend significantly on Heroes, Knowledge, and Continues, supporting the idea of offering tailored challenges and rewards.
Beginner Player Clusters	Support new players, especially in clusters 1 and 2, by offering comprehensive tutorials on gameplay strategies to help them understand game mechanics and improve their skills. Introduce a "Continue" package (e.g., 50 Continues for $1.00) to encourage IAP spending, as beginners have the highest MM spending ratio on Continues.	Beginners in Cluster 1 primarily play normal games and spend MM significantly on continues, indicating struggles with game difficulty. Providing tutorials can help them overcome these challenges and improve their gameplay experience. Introducing a "Continue" package leverages their existing MM spending behavior, encouraging further engagement and IAP spending.

Basis	Recommendations	Rationale
Intermediate Player Clusters	Adjust game difficulty at this level to improve win rates and retain players who might be discouraged by frequent losses. Promote non-standard game modes to keep gameplay fresh and engaging, encouraging players to explore different aspects of the game. Specifically target clusters 3 and 4 as they show significant engagement in event games and strategic spending.	Intermediate players, especially in Cluster 1, have low win rates despite significant participation in boss events and MM spending on heroes and power. As seen from EDA, the number of active players, IAP spending players, and average win rate all peak around rank 40 before there is a drop-off. This suggests that game difficulty at this level might be discouraging. Adjusting the difficulty and promoting non-standard game modes can help retain these players by providing a more balanced and engaging experience. Clusters 3 and 4 data show engagement in event games and strategic spending, indicating a potential for higher retention with targeted content.
Experienced Player Clusters	With attention to clusters 3 and 4, engage explorer players who enjoy playing but have stopped spending on IAP by introducing new content, such as special events or exclusive challenges. Introduce more power items and upgrades, as achiever players are willing to spend more on IAP to enhance their gameplay experience.	Findings show that experienced players in Clusters 3 and 4 exhibit high engagement and participation rates but lower win rates and IAP spending. By introducing new content and power items, Ninja Kiwi can cater to these players' preferences and encourage further spending. Data shows that these players enjoy both strategic and character development aspects, making them ideal targets for new content and upgrades.

Basis	Recommendations	Rationale
Churn Prediction	To reduce player churn, implement daily login rewards and engaging in-game events to boost session engagement and retain players. Develop longer gameplay content and complex storylines to increase session duration, keeping players invested over multiple sessions. Expand the roster of paid heroes and diversify game modes, regularly rotating featured modes and maps to maintain player interest.	The recommendations are based on the 10 most important variables attained through Recursive Feature Elimination.
IAP Prediction	To increase the chances of an IAP, promote bundles including knowledge points, hero upgrades, and skins, emphasizing their benefits through notifications and limited-time offers to enhance in-game earnings and IAP spending. Leverage current player ranks by providing rank-based incentives such as exclusive items, discounts, or special event access to motivate continued playing and IAP spending.	The recommendations are based on the 10 most important variables attained through Recursive Feature Elimination.

Table 5: Recommendations and reasoning

5.2 Conclusion

This analysis provides Ninja Kiwi with a robust understanding of player behavior in BTD6, enabling data-driven decisions to enhance player engagement and business growth. It is suggested that the company integrates the two-step clustering models and the fine-tuned Gradient Boosting models for churn and IAP prediction into the game's backend so that it can apply targeted retention and marketing strategies. These models highlight key factors influencing player retention and IAP spending, offering insights for personalized content and promotional campaigns. Implementing these insights can improve player retention, increase in-app purchases, and drive revenue growth while maintaining player satisfaction and loyalty. Automated model retraining and an analytics dashboard will ensure ongoing performance monitoring and actionable insights.

References

Henry Burrell. (2021). How Ninja Kiwi fought its way to the top. BusinessDesk.
https://businessdesk.co.nz/article/technology/how-ninja-kiwi-fought-its-way-to-the-top

Eric McClure. (2022). How To Master Bloons TD6: The Ultimate Strategy Guide. wikiHow.
https://www.wikihow.com/Bloons-TD-6-Strategy

Necati Demir. (2015). Ensemble Methods: Elegant Techniques to Produce Improved Machine Learning Results. Developers. Toptal.
https://www.toptal.com/machine-learning/ensemble-methods-machine-learning

Abhishek Sharma. (2023). Cross Validation in Machine Learning. GeeksforGeeks
https://www.geeksforgeeks.org/cross-validation-machine-learning/

Appendix

1	column	definition
2	userid	all user id's from the start session table - 14,570,942 users
3	min_sessiondate	the minimum date when a session was started by the user (first session)
4	max_sessiondate	the maximum date when a session was started by the user (last session)
5	num_sessions	number of distinct session_id's
6	ios_flag	flag = 1 when ios is user, else 0
7	android_flag	flag = 1 when android is user, else 0
8	steam_flag	flag = 1 when steam is user, else 0
9	monkeymoney_gained_excl_iap	total monkey money gained by every way except IAP
10	avg_moneygained_persession	total money gained / number of distinct session_id's
11	tropheymoney_gained	number of trophies gained
12	avg_trophiesaamtgained_persession	number of trophies gained / number of distinct session id's
13	avg_days_per_week	average number of days per week a session was started
14	session_datediff	the different between the first sessions date and last sessions date
15	played_within_30days	the flag is 1 if the user used the app in the last 30 days
16	num_free_heroes	number of free heroes earned
17	num_paid_heroes	number of paid heroes earned
18	num_default_skin	number of default skins earned
19	num_new_skin	number of new skins earned
20	trophies_spent	total number of trophies spent
21	monkeymoney_spent	total amount of monkey money spent
22	iap_spend	total money spent on in-app purchases
23	monkeymoney_purchased_prop	proportion of transactions where item_name in ('btd6_pileomonkeymoney_sale', 'btd6_vaultomonkeymoney', 'btd6_vaultomonkeymoney_sale', 'btd6_mountainomonkeymoney', 'btd6_pileomonkeymoney', 'btd6_monkeymoney', 'btd6_extramonkeymoney')
24	cashmode_purchased_prop	proportion of transactions where item_name in ('btd6_doublecashmode')
25	tower_paragon_update_purchased_prop	proportion of transactions where item_name in ('btd6_fulltowerunlock', 'btd6_towerandparagonunlock', 'btd6_paragonunlock', 'btd6_tier4instatowerspack', 'btd6_tier4instatowerspack_sale', 'btd6_towerxp', 'btd6_tier3instatowerspack_sale', 'btd6_tier5instatowerpack', 'btd6_tier3instatowerspack')
26	knowledge_purchased_prop	proportion of transactions where item_name in ('btd6_knowledgepoints', 'btd6_starterheroskinbundle', 'btd6_knowledgeunlocked', 'btd6_knowledgepoint')
27	upgradehero_purchased_prop	proportion of transactions where item_name in ('btd6_benjaminbundle', 'btd6_patfustybundle', 'btd6_ezilibundle', 'btd6_churchillbundle')
28	other_items_purchased_prop	proportion of transactions where all the other items are purchased
29	distinct_items_purchased	total number of distinct iap items bought
30	avg_txn_value	average amount of money spent in each transaction
31	knowledge_moneyspent_prop	the proportion of times the monkey money was spent on 'PurchasingMonkeyKnowledge'
32	continue_moneyspent_prop	the proportion of times the monkey money was spent on anything that starts with 'Continue'
33	hero_moneyspent_prop	the proportion of times the monkey money was spent on anything that starts with 'Hero:'
34	heroskin_moneyspent_prop	the proportion of times the monkey money was spent on anything that starts with 'Hero Skin:'
35	power_moneyspent_prop	the proportion of times the monkey money was spent on anything that starts with 'Power'
36	other_moneyspent_prop	the proportion of times the monkey money was spent on any items not mentioned above
37	max_game_id	maximum game_id for each user
38	min_game_date	first time a game was started - no matter what the first game_id was. I get the min(game_id)
39	num_game_modes	number of distinct game modes played
40	num_map_played	number of distinct maps played
41	coop_mode_prop	Proportion of times coop mode was used while playing
42	boss_event_prop	Proportion of 'Elite Boss', 'Regular Boss' played
43	normal_games_prop	Proportion of 'Modded Standard', 'Standard' played
44	event_games_prop	Proportion of 'Boss Challenge', 'Contested Territory', 'Daily Challenge', 'Odyssey', 'Player Challenge', 'Race' played
45	other_games_prop	Proportion of 'MapEditor', 'Sandbox', 'Tutorial', 'Quest', 'challenge testing' played
46	avg_games_per_week	average number of distinct game_id's played each week
47	current_rank	the current_rank after the most recent game
48	win_rate	(# of times the game was won) divided by (# of times the game was won or lost - nulls in starttrack and not in endtrack are also counted as a loss) and where was_freeplay = 0
49	iap_flag	flag = 1 when a user made an in-app purchase

Figure A1: Descriptions of the feature engineered variables in the final table (please zoom in)

Clusters

Input (Predictor) Importance
■ 1.0 ■ 0.8 ■ 0.6 ■ 0.4 ■ 0.2 □ 0.0

Cluster	1	2	3	4
Label				
Description				
Size	15.6% (921576)	27.9% (1646483)	42.9% (2530932)	13.6% (803143)
Inputs	boss_event_prop 0.09	boss_event_prop 0.02	boss_event_prop 0.01	boss_event_prop 0.01
	continue_ moneyspent_prop	continue_ moneyspent_prop	continue_ moneyspent_prop	continue_ moneyspent_prop
	event_games_prop 0.42	event_games_prop 0.07	event_games_prop 0.03	event_games_prop 0.05
	hero_moneyspent_ prop	hero_moneyspent_ prop	hero_moneyspent_ prop	hero_moneyspent_ prop
	knowledge_ moneyspent_prop	knowledge_ moneyspent_prop	knowledge_ moneyspent_prop	knowledge_ moneyspent_prop
	normal_games_prop 0.36	normal_games_prop 0.80	normal_games_prop 0.83	normal_games_prop 0.78
	power_moneyspent_ prop	power_moneyspent_ prop	power_moneyspent_ prop	power_moneyspent_ prop
Evaluation Fields	iap_flag 0 (84.9%)	iap_flag 0 (83.8%)	iap_flag 0 (94.7%)	iap_flag 0 (92.9%)
	iap_spend 4.28	iap_spend 3.06	iap_spend 1.03	iap_spend 2.08
	played_within_30day s	played_within_30day s	played_within_30day s	played_within_30day s
	avg_games_per_ week	avg_games_per_ week	avg_games_per_ week	avg_games_per_ week
	current_rank 71.00	current_rank 66.20	current_rank 46.73	current_rank 45.96
	win_rate 0.29	win_rate 0.44	win_rate 0.44	win_rate 0.44

Figure A2: Two-step clustering results for all players clusters (please zoom in)

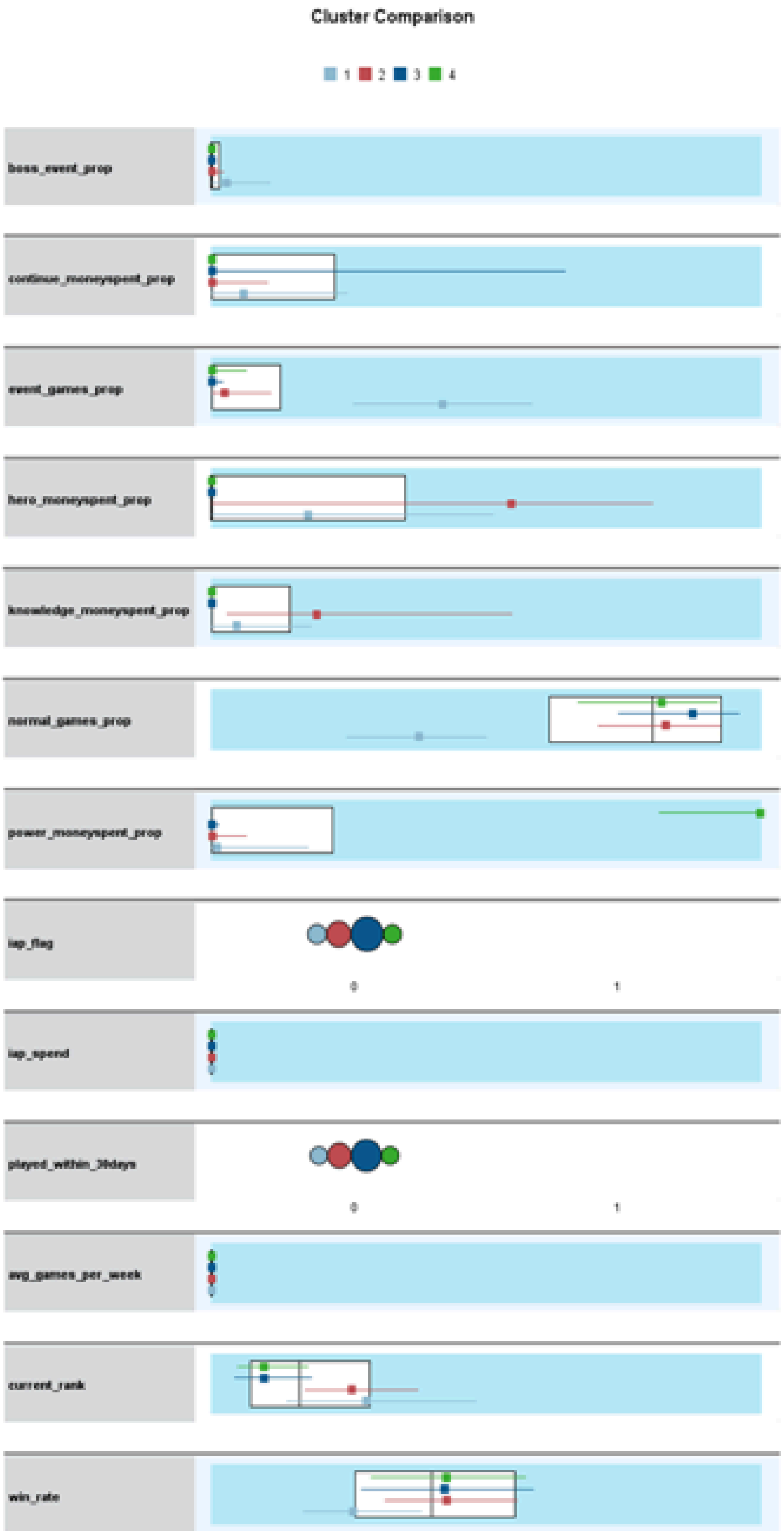


Figure A3: Two-step clustering analysis for all players clusters (please zoom in)

Clusters

Input (Predictor) Importance

1.0 0.8 0.6 0.4 0.2 0.0

Cluster	1	2	3	4
Label				
Description				
Size	53.3% (2150158)	19.1% (770533)	17.0% (686897)	10.5% (423114)
Inputs	boss_event_prop 0.01	boss_event_prop 0.01	boss_event_prop 0.01	boss_event_prop 0.10
	continue_moneyspent_prop	continue_moneyspent_prop	continue_moneyspent_prop	continue_moneyspent_prop
	event_games_prop 0.03	event_games_prop 0.05	event_games_prop 0.04	event_games_prop 0.38
	hero_moneyspent_prop	hero_moneyspent_prop	hero_moneyspent_prop	hero_moneyspent_prop
	knowledge_moneyspent_prop	knowledge_moneyspent_prop	knowledge_moneyspent_prop	knowledge_moneyspent_prop
	normal_games_prop 0.83	normal_games_prop 0.81	normal_games_prop 0.80	normal_games_prop 0.36
	power_moneyspent_prop	power_moneyspent_prop	power_moneyspent_prop	power_moneyspent_prop
Evaluation Fields	iap_flag 0 (96.5%)	iap_flag 0 (88.1%)	iap_flag 0 (95.0%)	iap_flag 0 (93.6%)
	iap_spend 0.44	iap_spend 2.09	iap_spend 0.91	iap_spend 1.07
	played_within_30days	played_within_30days	played_within_30days	played_within_30days
	avg_games_per_week	avg_games_per_week	avg_games_per_week	avg_games_per_week
	current_rank 41.26	current_rank 49.99	current_rank 41.44	current_rank 45.23
	win_rate 0.44	win_rate 0.48	win_rate 0.45	win_rate 0.32

Figure A4: Two-step clustering results for beginners players clusters (please zoom in)

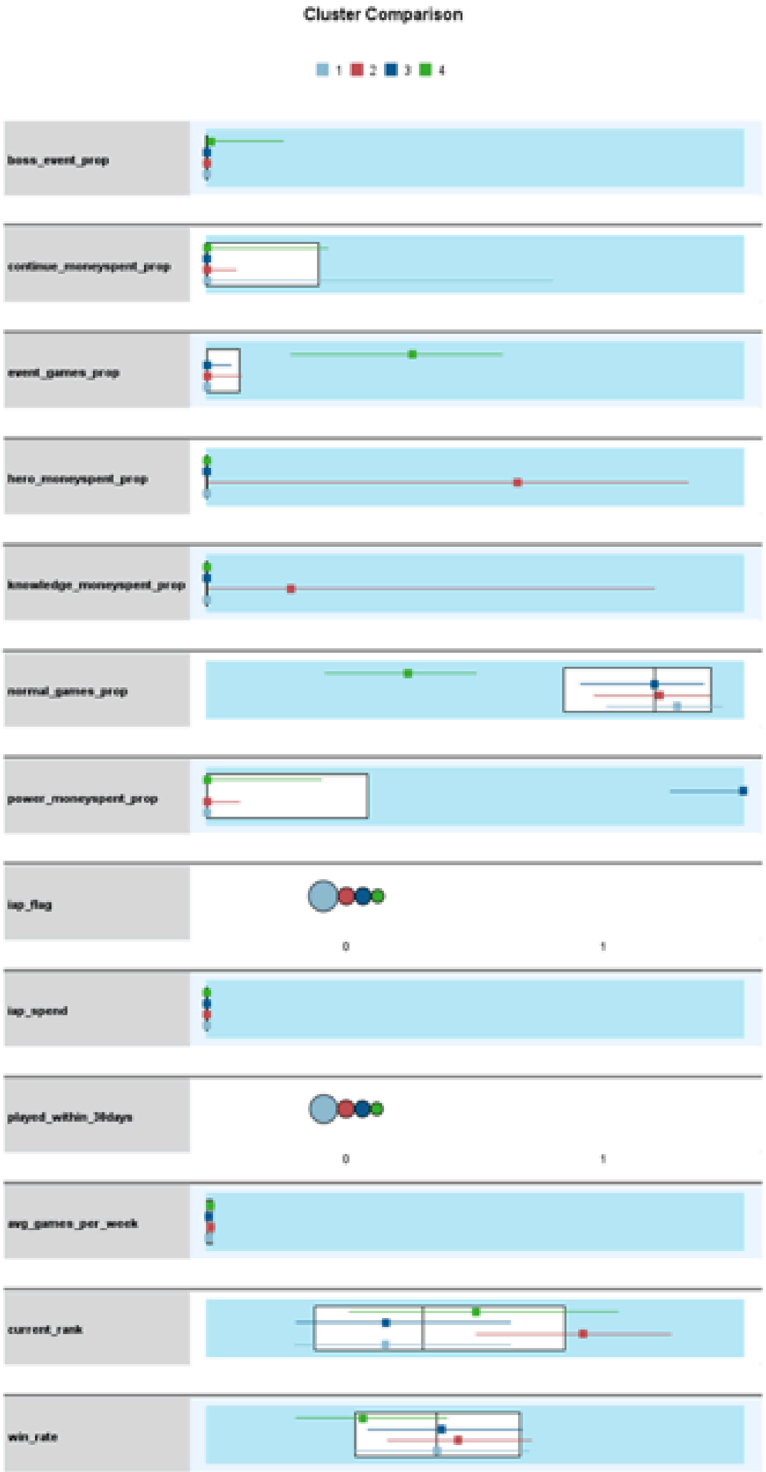
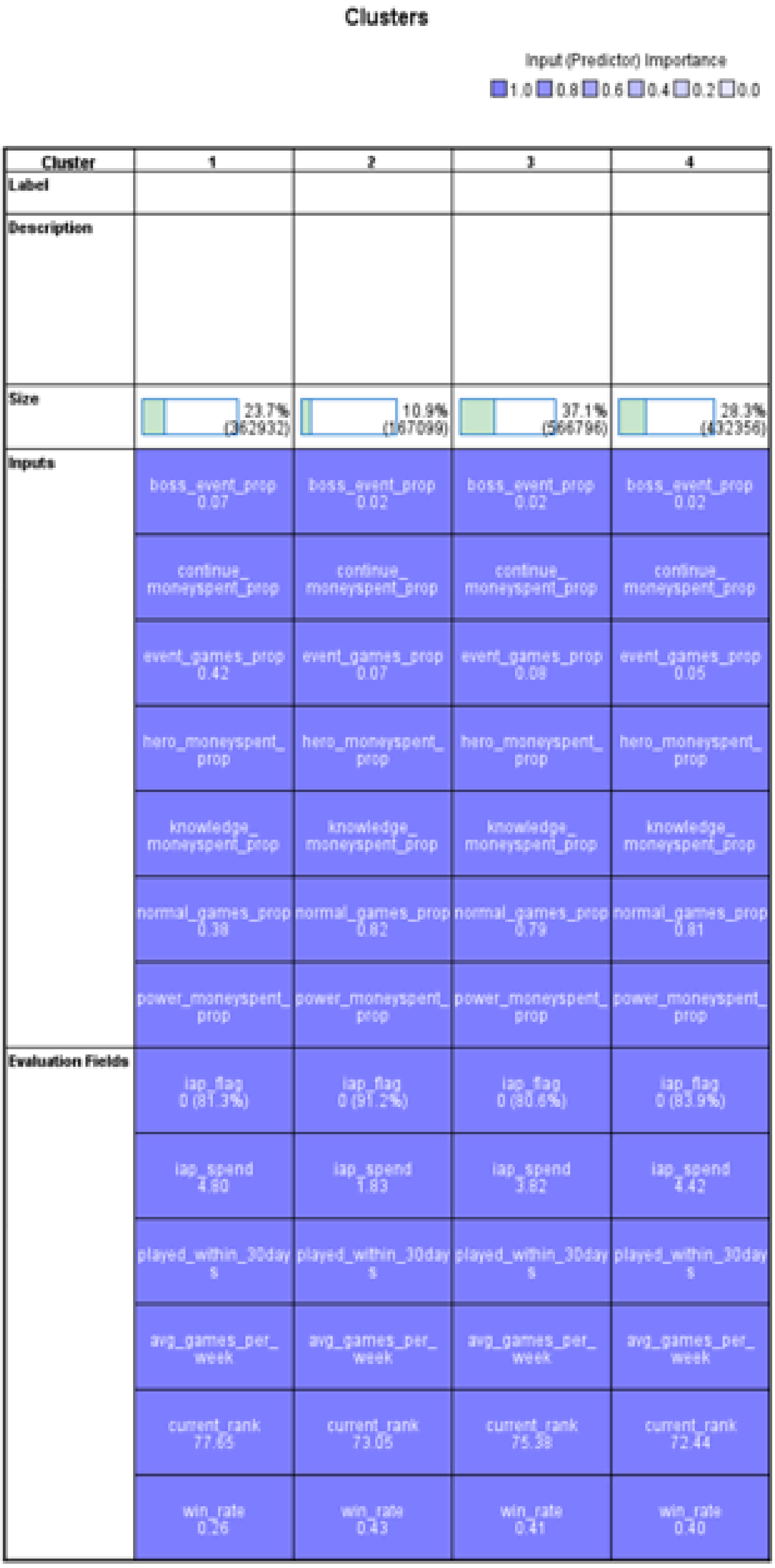


Figure A5: Two-step clustering analysis for beginners players clusters (please zoom in)



Clusters

Input (Predictor) Importance
1.0 0.8 0.6 0.4 0.2 0.0

Cluster	1	2	3	4
Label				
Description				
Size	23.7% (362932)	10.9% (167099)	37.1% (566796)	28.3% (432356)
Inputs	boss_event_prop 0.07	boss_event_prop 0.02	boss_event_prop 0.02	boss_event_prop 0.02
	continue_ moneyspent_prop	continue_ moneyspent_prop	continue_ moneyspent_prop	continue_ moneyspent_prop
	event_games_prop 0.42	event_games_prop 0.07	event_games_prop 0.08	event_games_prop 0.05
	hero_moneyspent_ prop	hero_moneyspent_ prop	hero_moneyspent_ prop	hero_moneyspent_ prop
	knowledge_ moneyspent_prop	knowledge_ moneyspent_prop	knowledge_ moneyspent_prop	knowledge_ moneyspent_prop
	normal_games_prop 0.38	normal_games_prop 0.82	normal_games_prop 0.79	normal_games_prop 0.81
	power_moneyspent_ prop	power_moneyspent_ prop	power_moneyspent_ prop	power_moneyspent_ prop
Evaluation Fields	iap_flag 0 (81.3%)	iap_flag 0 (91.2%)	iap_flag 0 (80.6%)	iap_flag 0 (83.9%)
	iap_spend 4.80	iap_spend 1.83	iap_spend 3.82	iap_spend 4.42
	played_within_30day s	played_within_30day s	played_within_30day s	played_within_30day s
	avg_games_per_ week	avg_games_per_ week	avg_games_per_ week	avg_games_per_ week
	current_rank 77.65	current_rank 73.05	current_rank 75.38	current_rank 72.44
	win_rate 0.26	win_rate 0.43	win_rate 0.41	win_rate 0.40

Figure A6: Two-step clustering results for intermediate players clusters (please zoom in)

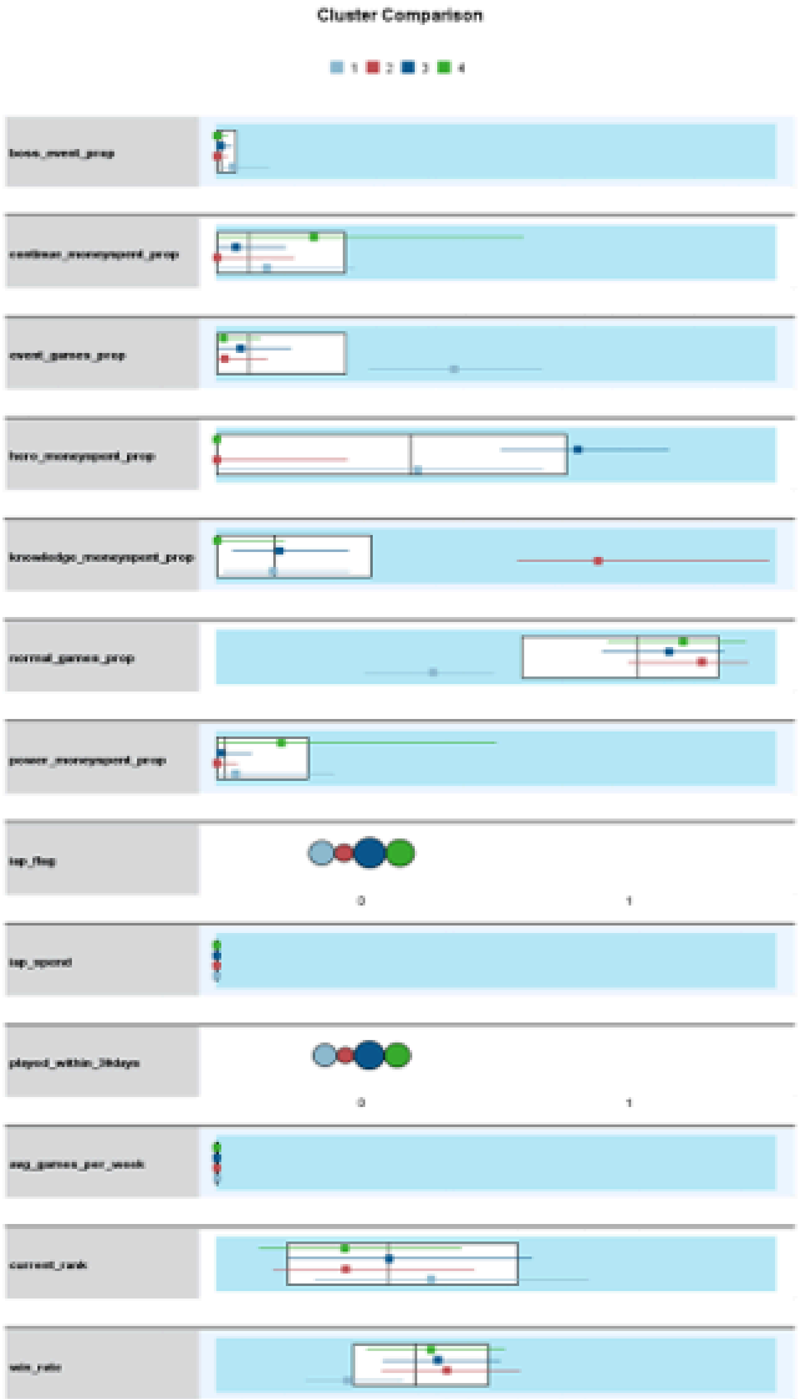


Figure A7: Two-step clustering analysis for intermediate players clusters (please zoom in)

Clusters

Input (Predictor) Importance

1.0 0.8 0.6 0.4 0.2 0.0

Cluster	1	2	3	4
Label				
Description				
Size	16.5% (56616)	31.8% (108687)	20.5% (70165)	31.2% (106781)
Inputs	boss_event_prop 0.13	boss_event_prop 0.03	boss_event_prop 0.02	boss_event_prop 0.04
	continue_moneyspent_prop	continue_moneyspent_prop	continue_moneyspent_prop	continue_moneyspent_prop
	event_games_prop 0.25	event_games_prop 0.11	event_games_prop 0.18	event_games_prop 0.55
	hero_moneyspent_prop	hero_moneyspent_prop	hero_moneyspent_prop	hero_moneyspent_prop
	knowledge_moneyspent_prop	knowledge_moneyspent_prop	knowledge_moneyspent_prop	knowledge_moneyspent_prop
	normal_games_prop 0.54	normal_games_prop 0.80	normal_games_prop 0.71	normal_games_prop 0.35
	power_moneyspent_prop	power_moneyspent_prop	power_moneyspent_prop	power_moneyspent_prop
Evaluation Fields	iap_flag 0 (67.2%)	iap_flag 0 (75.2%)	iap_flag 0 (77.0%)	iap_flag 0 (76.0%)
	iap_spend 24.26	iap_spend 8.40	iap_spend 10.14	iap_spend 6.64
	played_within_30days	played_within_30days	played_within_30days	played_within_30days
	avg_games_per_week	avg_games_per_week	avg_games_per_week	avg_games_per_week
	current_rank 120.48	current_rank 114.49	current_rank 120.57	current_rank 118.33
	win_rate 0.33	win_rate 0.41	win_rate 0.40	win_rate 0.28

Figure A8: Two-step clustering results for experienced players clusters (please zoom in)

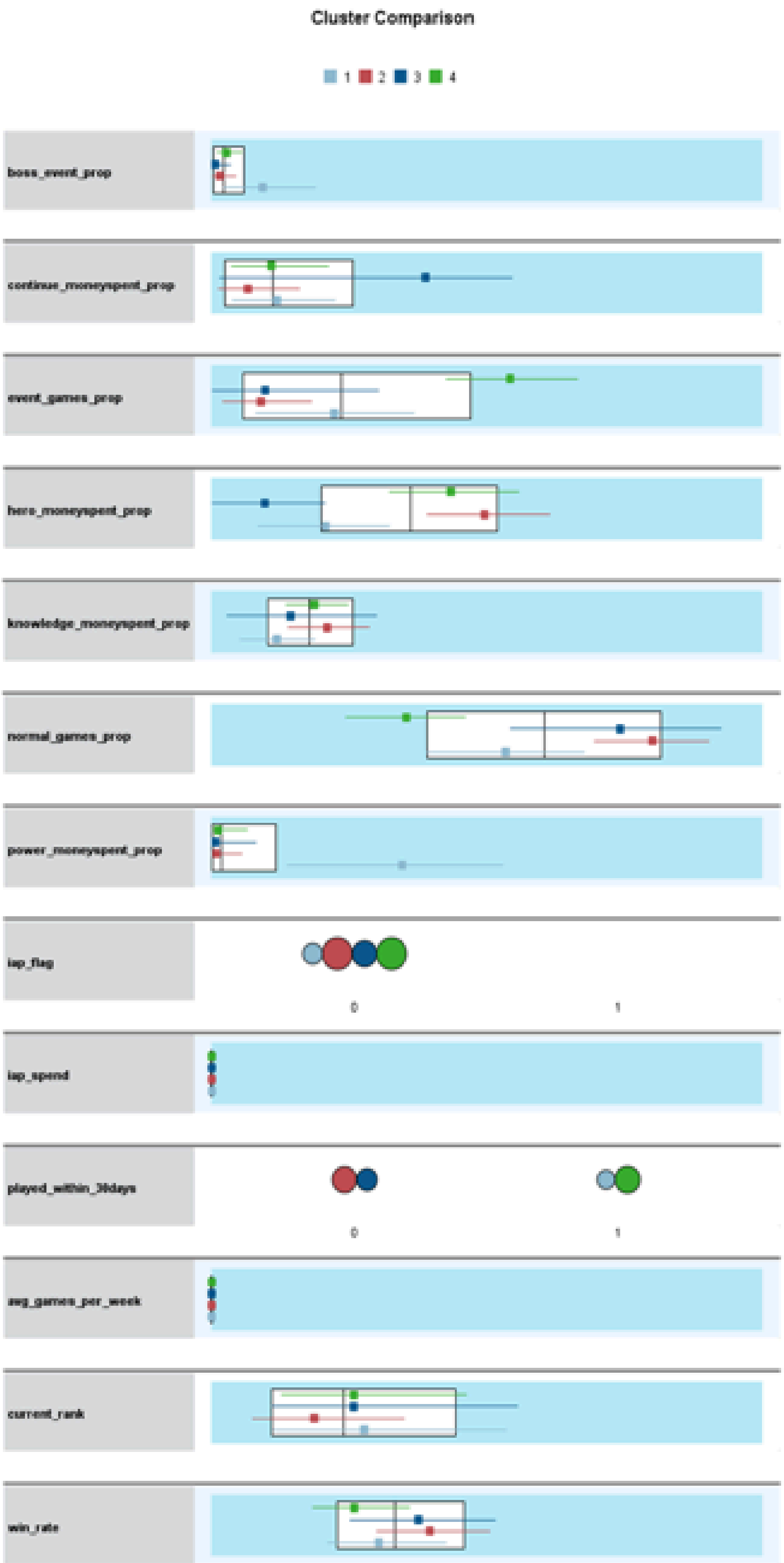


Figure A9: Two-step clustering analysis for experienced players clusters (please zoom in)