ABC Gaming Loyalty Points Analysis

NAVYA MUDGAL

JULY,08,2025

TOOLS USED-

Python (Jupyter Notebook)

INTRODUCTION

The objective of this assignment is to calculate player-wise loyalty points on the ABC gaming platform based on their activity, including deposits, withdrawals, and games played. These loyalty points help measure player engagement and commitment to the platform. Using these points, the goal is to generate actionable insights and propose a fair bonus distribution strategy to reward loyal players and enhance player retention.

DATASET DESCRIPTION

Three datasets were provided for this analysis:

- **Deposits Sheet:** Contains Player ID, deposit amount, and timestamp of each deposit.
- **Withdrawals Sheet:** Contains Player ID, withdrawal amount, and timestamp of each withdrawal.
- **Games Sheet:** Contains Player ID, Game ID, and timestamp of each game played.

All data corresponds to player activities during October, including precise timestamps. These datasets enable a detailed view of both financial transactions and gameplay behavior of each player, which serve as inputs to calculate loyalty points.

Methodology

The loyalty points calculation and analysis were conducted using Python in a Jupyter Notebook environment, leveraging the Pandas library for efficient data processing. The following step-by-step approach was adopted:

- **Data Import:**
  The three datasets — Deposits, Withdrawals, and Games — were imported as Pandas DataFrames with appropriate parsing of date-time columns for accurate timestamp handling.
- **Timestamp Processing:**
  Each transaction or gameplay timestamp was converted to extract the date and assigned to one of two time slots per day:
  - Slot S1: 12:00 AM to 11:59 AM
  - Slot S2: 12:00 PM to 11:59 PM
- **Data Aggregation:**
  For each dataset, data was grouped by Player ID, Date, and Slot to calculate:
  - Total deposit and withdrawal amounts, and their counts
  - Number of games played
- **Data Consolidation:**
  The aggregated deposit, withdrawal, and gameplay summaries were merged into a single DataFrame, ensuring missing values were filled with zero to maintain consistency.

- **Loyalty Points Computation:**
  The loyalty points for each player per date and slot were calculated using the formula:

  LoyaltyPoints=(0.01×DepositAmount)+(0.005×WithdrawalAmount)+(0.001×max⁡(DepositCount−WithdrawalCount,0))+(0.2×GamesPlayed)LoyaltyPoints = (0.01 \times DepositAmount) + (0.005 \times WithdrawalAmount) + (0.001 \times \max(DepositCount - WithdrawalCount, 0)) + (0.2 \times GamesPlayed)LoyaltyPoints=(0.01×DepositAmount)+(0.005×WithdrawalAmount)+(0.001×max(DepositCount−WithdrawalCount,0))+(0.2×GamesPlayed)

- **Monthly Aggregation and Ranking:**
  Player-wise loyalty points were summed over the entire month to compute total monthly points. Players were then ranked based on their total points to identify the top performers.
- **Additional Metrics:**
  Further analysis included calculating average deposit amounts, average deposits per user, and average games played per user to gain broader insights into player behavior.

Loyalty Points

The analysis identified top-performing players on each date and time slot based on their loyalty points. Sample results include:

Date	Slot	Top Player	Loyalty Points
2nd October	S1	Player123	15.5
16th October	S2	Player456	12
...	...	...	...

These rankings highlight the most engaged and valuable players in terms of activity and financial transactions during specific time windows.

Bonus Allocation (Rs. 50,000)

To reward the top 50 players based on their loyalty and engagement, a fair and balanced bonus distribution method was implemented. The total bonus pool of Rs. 50,000 was allocated using a weighted scoring system that considers both financial contributions and gameplay activity.

Approach:

- **Weighted Factors:**
  The bonus allocation formula assigned 70% weight to each player's share of total loyalty points and 30% weight to their share of total games played. This ensures that both heavy spenders and highly active players are rewarded appropriately.
- **Calculation Steps:**
  1. Calculated each player's percentage contribution to the total loyalty points among the top 50 players.
  2. Calculated each player's percentage contribution to the total games played among the same group.
  3. Computed a **Final Score** for each player as:

     Final Score=(0.7×Loyalty Points %)+(0.3×Games Played %)\text{Final Score} = (0.7 \times \text{Loyalty Points \%}) + (0.3 \times \text{Games Played \%})Final Score=(0.7×Loyalty Points %)+(0.3×Games Played %)

4. Distributed the total bonus proportionally based on the Final Score:

   Bonus=(Final Score∑Final Scores)×50,000\text{Bonus} = \left(\frac{\text{Final Score}}{\sum \text{Final Scores}}\right) \times 50,000Bonus=(∑Final ScoresFinal Score)×50,000

Outcome:

This weighted allocation method provides a balanced incentive structure that motivates players to both invest financially and participate actively in gameplay. By doing so, it encourages sustained engagement across multiple dimensions, fostering long-term loyalty on the platform.

Fairness of Loyalty Formula

Key observations about the current formula:

- Players earn points for withdrawals, which may not truly reflect loyalty.
- Players with high deposits gain an outsized advantage.
- Regular players making smaller deposits are undervalued.

Suggestions for improvement:

- Remove points for withdrawals.
- Introduce login streak points to reward consistent engagement.
- Cap daily deposit-based points to avoid excessive advantage.
- Reward players based on win ratio or gameplay quality.
- Implement a player tier/level system for better differentiation.

CONCLUSION

This project successfully demonstrated the use of Python and Pandas in a Jupyter Notebook environment to calculate player-wise loyalty points on the ABC gaming platform by integrating deposit, withdrawal, and gameplay data. The methodology allowed detailed time-slot based analysis and comprehensive aggregation of player activity.

The proposed weighted bonus allocation system effectively balances rewarding both high-value spenders and active players, ensuring fair and motivating incentives. Furthermore, the critical evaluation of the existing loyalty formula highlighted key areas for improvement, such as removing points for withdrawals and incorporating engagement-based rewards like login streaks and gameplay quality.

Overall, this analysis provides actionable insights and a data-driven framework to enhance player retention and satisfaction through a transparent, fair loyalty program.