Loyalty Program Analysis Report

**Prepared by: Samruddhi Gulhane**

# 1. Introduction

This report documents the analysis and implementation of a loyalty program for the ABC Gaming Platform. The primary objective was to design a points-based system that rewards players based on their engagement and monetary transactions, and to develop a fair strategy to distribute bonuses to the top-performing players. The entire analysis, data processing, and leaderboard generation were implemented in Python, as presented in the

# 2. Data Preparation

A synthetic dataset was created to simulate realistic player behavior on the gaming platform for the month of October. Each record represents an individual player and includes the following attributes:

- Deposit Amount: Total funds deposited by the player.  
- Withdrawal Amount: Total funds withdrawn by the player.  
- Number of Deposits: Count of deposit transactions.  
- Number of Withdrawals: Count of withdrawal transactions.  
- Number of Games Played: Total games played during the period.  
- Login Streak: Consecutive days of login activity.

The dataset was generated using the Python pandas and NumPy libraries to simulate variability in player activity.

# 3. Loyalty Points Calculation

The initial loyalty points formula was designed to reward both financial contribution and gaming activity. The formula used is as follows:

Loyalty Points = (0.01 × Deposit) + (0.005 × Withdrawal) + 0.001 × max (0, Number of Deposits − Number of Withdrawals) + 0.2 × Games Played

This formula was chosen to incentivize deposits, discourage excessive withdrawals, and reward active gaming participation.

For each player, the following metrics were computed:  
- Individual loyalty points  
- Average deposit amount  
- Average number of games played per user  
- Total loyalty points accumulated in the month

The results were validated and exported for further analysis.

# 4. Bonus Distribution Strategy

To encourage player retention and reward top contributors, ABC Gaming allocated a total bonus pool of ₹50,000, to be distributed among the top 50 players.

Proposed Hybrid Bonus Model:

Metric | Weight | Bonus Allocation  
Loyalty Points | 70% | ₹35,000  
Games Played | 30% | ₹15,000

The hybrid model ensures that players are rewarded not only for financial contribution but also for active engagement on the platform.

Distribution Method:  
- Players are ranked based on their loyalty points and games played.  
- Each player's share is calculated proportionally to the total loyalty points and total games played among the top 50 players.  
- Final bonus per player is the sum of their loyalty-based bonus and engagement-based bonus.

A comprehensive leaderboard was generated and exported as Top\_50\_Hybrid\_Bonus\_Leaderboard.csv, detailing:  
- Player ID  
- Loyalty Points  
- Games Played  
- Loyalty Points Bonus  
- Games Played Bonus  
- Total Bonus Awarded.

# 5. Evaluation of the Loyalty Points Formula

Strengths:  
- Encourages deposits and active gaming.  
- Provides a simple yet effective metric for ranking players.  
- Easy to interpret and scale.

Limitations:  
- May disproportionately favor players with high deposits but low engagement.  
- Does not reward skill or competitiveness (e.g., games won).  
- Does not fully account for consistent daily activity (login streak).

Proposed Improvements:  
To enhance fairness and promote balanced user behavior, an updated formula was recommended:

Revised Loyalty Points = (0.008 × Deposit) + (0.004 × Withdrawal) + 0.001 × max(0, Number of Deposits − Number of Withdrawals) + 0.15 × Games Played + 0.1 × Games Won + 0.02 × Login Streak

This version:  
- Lowers the weight for deposits to reduce bias toward big spenders.  
- Adds a reward for games won, promoting competitive play.  
- Includes login streaks, encouraging consistent daily engagement.

These refinements provide a more holistic measure of player value and foster healthy gaming habits.

# 6. Technical Implementation

The complete solution was implemented in Python, leveraging the following tools and libraries:  
- pandas: For data manipulation and aggregation.  
- NumPy: For generating random distributions and calculations.  
- CSV Export: To store the final leaderboard and summary reports.

The scripts include:  
- Data generation and preprocessing.  
- Loyalty points computation.  
- Bonus calculation logic.  
- Leaderboard ranking and export.

All scripts and outputs are available in the GitHub repository for reference and reuse.

# 7. Conclusion

This project successfully simulates a loyalty program for a gaming platform, combining data-driven analysis with a practical reward mechanism. The hybrid bonus model aligns monetary contribution with genuine player engagement, ensuring fair distribution of incentives.

With the proposed enhancements to the loyalty formula, this model can be directly adapted for real-world deployment, supporting user retention and encouraging active, healthy gaming behavior.

Repository: https://github.com/SamruddhiGulhane/project3-by-sam