ML-Powered Game Analytics Project

Prepared by: Rama Manna'

Executive Summary

This project focuses on analyzing player behavior using machine learning techniques to gain actionable insights and improve game design. We used real gameplay session data from a simple 2D browser game, applied visualizations, clustering, and prediction models to identify behavioral patterns, anticipate player struggles, and generate automated recommendations to enhance player experience.

Introduction

Problem Statement: Game developers often lack data-driven insights into player behavior, which leads to trial-and-error adjustments in game design.

Project Goal: To build an ML-powered analytics system that collects player data, detects behavior patterns, and produces useful recommendations to improve balance and engagement in the game.

Methodology

Game Design Choice:
We used session logs from a 2D browser-based game. Each session was recorded with key metrics such as the number of moves, session duration, and final result (win or lose).

Data Collection:
The dataset `game_Rama_data.csv` contained:
- Number of player moves
- Game session duration (in seconds)
- Session result (win or lose)

Dashboard Implementation:
Using Python libraries (`matplotlib`, `seaborn`), we created visualizations to display:
- Move and duration distributions by result
- Player clustering using K-Means
- Predictive patterns using Decision Trees

Machine Learning Implementation:
1. Clustering (K-Means): Segmented players into 3 behavior-based groups

based on moves and session duration.
2. Prediction (Decision Tree): Trained a model to predict win/loss outcomes using simple features.
3. Insight Generation: Analyzed the average behavior of losing players and proposed design recommendations accordingly.

Results and Analysis

Key Patterns:
- Losing players generally had lower move counts and shorter gameplay sessions.
- K-Means clustering revealed three distinct behavioral groups.
- The Decision Tree model showed reasonable accuracy in predicting outcomes.

Recommendations:
"Most losing players had around 10 moves and played for approximately 30 seconds. Consider adding a tutorial or visual hint during this time frame to support struggling players."

Technical Challenges

- Required data cleaning and type conversion for compatibility with ML models.
- Feature selection involved trial and error to find meaningful inputs.
- Balancing code simplicity with informative and clear visualizations.

Future Improvements

- Include more session context such as stage level or player decisions.
- Develop an interactive dashboard using Streamlit or Dash.
- Explore more advanced models like Random Forest or Neural Networks.

Conclusion

This project effectively demonstrated the power of ML in game analytics by clustering player behavior, predicting session outcomes, and generating actionable insights. These approaches can significantly enhance game design, improve player retention, and increase satisfaction.