Machine Learning Based Performance Analysis Model for Esports-PUBG

Abstract

In this research, we present a machine learning-based model designed to enhance player performance in esports, specifically focusing on BGMI or popularly known as PUBG. The model begins with comprehensive data collection from Kaggle, followed by the detection and removal of anomalies, particularly inactive or AFK (away-from-keyboard) players, to ensure clean and relevant data. Through detailed Exploratory Data Analysis (EDA), including the use of correlation heatmaps and bar plots, we identify critical patterns that influence gameplay outcomes. Feature selection and engineering are employed to refine the dataset, optimizing it for input into our machine learning algorithm. The model focuses on detecting performance-enhancing anomalies across essential gameplay elements like kills, movement, and resource management. Based on this analysis, we provide personalized, data-driven recommendations to help players improve their in-game performance. The final phase involves deploying the model to deliver insights, assisting players in refining their strategy with actionable, tailored feedback aimed at boosting overall gameplay.

Introduction

The rapid rise of esports, particularly competitive titles like BGMI and PUBG, has emphasized performance optimization for players aiming to improve their strategies. As esports becomes a global phenomenon, the demand for tools that help players enhance their skills through data-driven insights has grown. However, players often face challenges due to the complexity of the game, with variables like kills, movement, resource management, and team coordination affecting performance. Addressing anomalies in gameplay data, such as players going AFK, is crucial for accurate performance analysis. This research develops a machine learning-based model for refining player performance by focusing on key metrics and providing real-time feedback.

Literature review

1. **"Prediction of the Final Rank of Players in PUBG with the Optimal Number of Features" (Sen et al.)**: This paper focuses on predicting the final rank of PUBG players by utilizing various machine learning algorithms on a dataset with 29 features. Through feature selection and engineering, the authors determined that using 8 key features (such as total distance traveled, team kills, and damage normalized) was optimal for balancing accuracy and computation time. Gradient Boosting Regressor (GBR) and LightGBM (LGBM) were found to deliver the best performance with accuracies above 91%, reducing the empirical runtime compared to models using more features.
2. **"Esports Analytics on PlayerUnknown's Battlegrounds Player Placement Prediction using Machine Learning Approach" (Ghazali et al.)**: In this study, the authors aimed to predict player placement in PUBG matches using machine learning models such as Decision Tree Regression, Support Vector Machine (SVM), and Linear Regression. The dataset from Kaggle with 29 attributes was used, and feature selection was performed to identify significant predictors like walk distance and kills. Among the models, SVM performed best in terms of RMSE, while Decision Tree Regression was the fastest(Pubg2).

Comparative Analysis:

Both papers used the Kaggle dataset with 29 attributes for PUBG and emphasized feature selection to improve model efficiency. However, the papers differ in their machine learning techniques. Sen et al. applied a variety of algorithms, with GBR and LGBM emerging as the best-performing models for predicting final rank. In contrast, Ghazali et al. focused on SVM and Decision Tree Regression, with SVM showing superior RMSE values but Decision Tree excelling in speed. Both papers identified similar influential features, like total distance traveled and kills, but Sen et al. reduced features to optimize runtime, whereas Ghazali et al. prioritized the accuracy of SVM.

Methodology

1. Data Sourcing

The data for this study is primarily sourced from publicly available PUBG/BGMI datasets on platforms like Kaggle, which provide comprehensive records of player statistics, match outcomes, and gameplay telemetry. These datasets typically include a wide range of features, such as player survival time, kills, damage dealt, weapon usage, and final match placements. In addition to public datasets, we also collected custom datasets by interfacing with in-game APIs and gathering player telemetry data. This data offers a more granular view of player performance, including real-time stats, movement patterns, and detailed weapon accuracy metrics.

2. Dataset Structure

The dataset consists of numerous features reflecting both player behavior and match dynamics. Key features include:

- **Numerical Features:**
    - *Survival Time:* Duration (in seconds) the player stays alive during the match.
    - *Kills:* Number of enemy players eliminated.
    - *Damage Dealt:* Total damage inflicted on opponents.
    - *Distance Traveled:* Total distance covered by the player during the match.
    - *Heal/Boost Items Used:* Count of health or boost items used.
    - *Weapon Accuracy:* Accuracy percentage of shots fired by the player.
- **Categorical Features:**
    - *Match Type:* Solo, Duo, or Squad.
    - *Final Match Placement:* Ranking of the player or team at the end of the match.
    - *Weapon Types:* Categories of weapons used, such as rifles, shotguns, or snipers.
    - *Region:* Server region (e.g., North America, Asia).

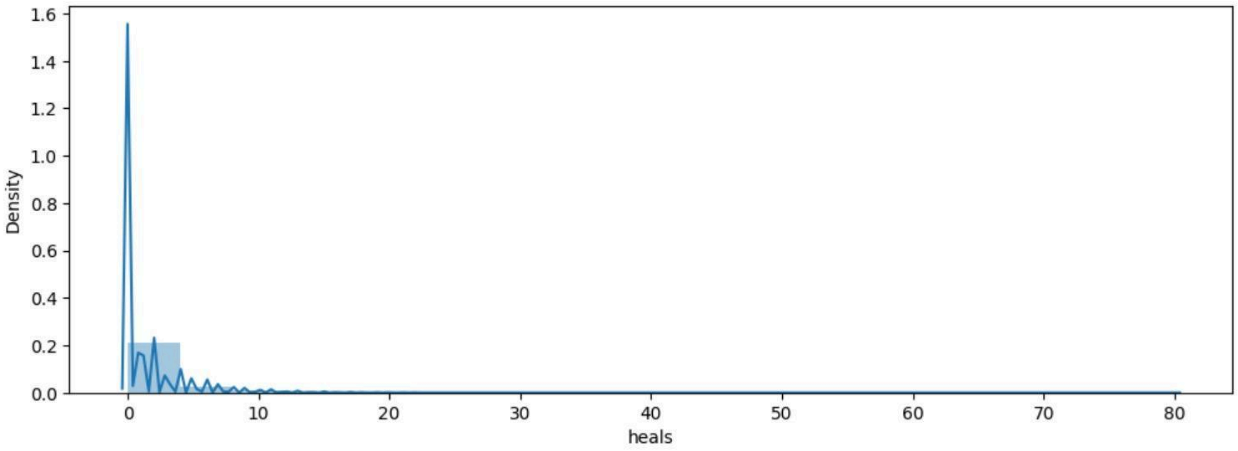
Additional features, like player movement behavior, team communication, and vehicle usage, are also considered, depending on the dataset's availability.

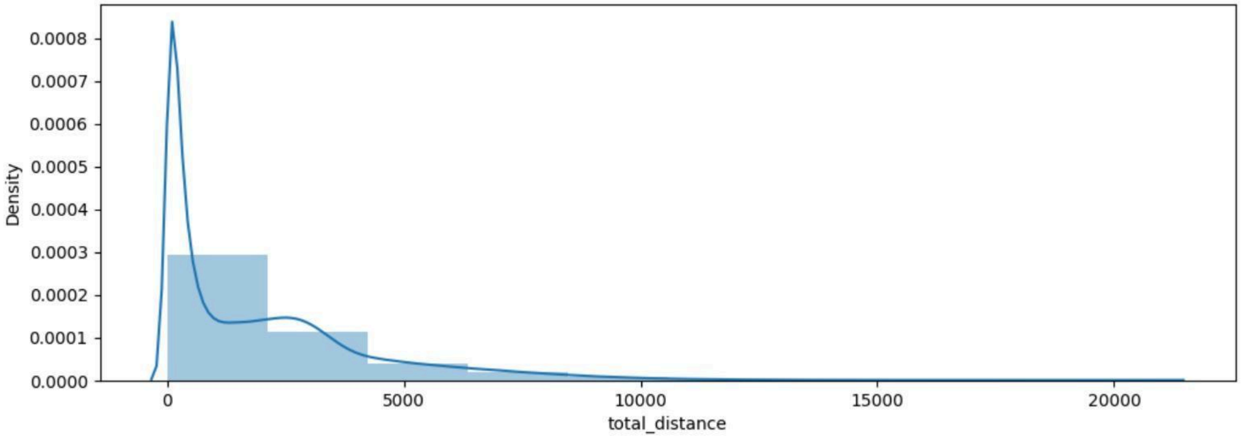
3. Data Preprocessing

Before feeding the data into machine learning models, extensive preprocessing steps are performed to ensure data quality and consistency:

- **Handling Missing Data:** Missing values are imputed using mean or median values for numerical features. For categorical features, missing data is handled by assigning the most frequent category or a distinct "unknown" label.

- **Outlier Removal:** Players who exhibit abnormal behavior, such as going AFK (away from keyboard) or being inactive for extended periods, are filtered out. Matches where players leave early or show zero movement are also excluded to reduce noise.





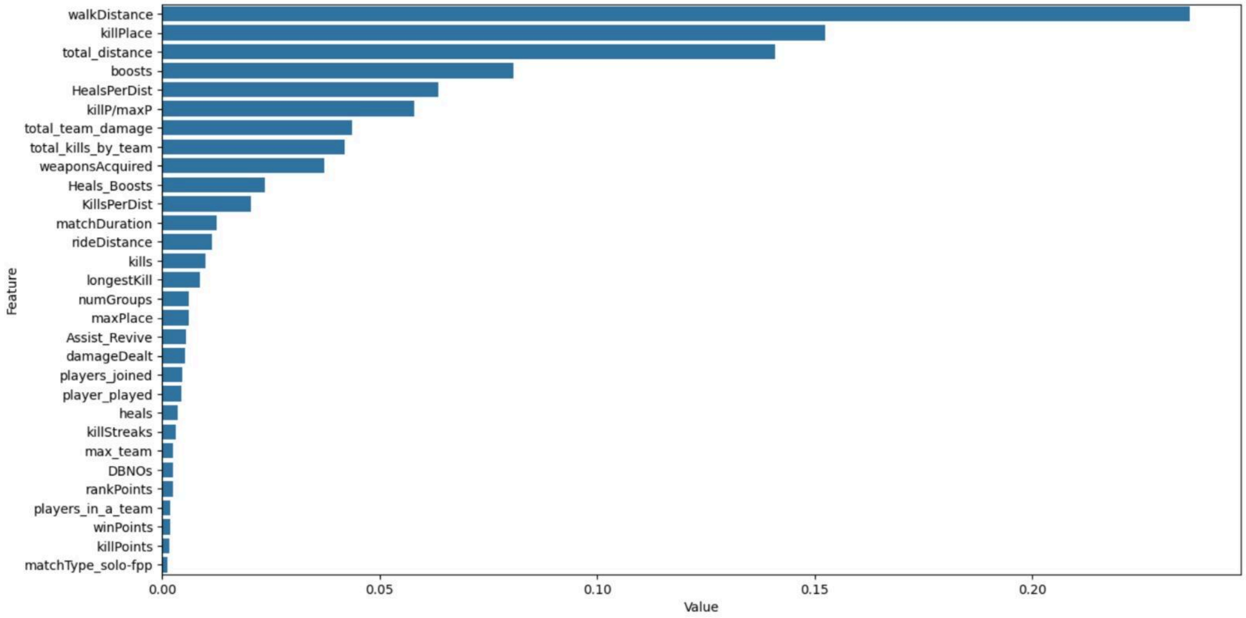
- **Standardization/Normalization:** Features like survival time, damage dealt, and distance traveled are standardized to have zero mean and unit variance. This ensures that features on different scales do not skew the model's learning process.
- **Feature Engineering:** Additional features are created through interactions between existing ones, such as combining kills and assists into a "combat engagement" metric. We also create new variables based on distance covered and accuracy, aiming to better capture player skill and behavior.

4. Feature Description

The following features are identified as the most relevant to player performance in PUBG/BGMI:

- **Survival Time:** The length of time a player remains alive in the match is one of the strongest indicators of skill, as better players tend to survive longer.
- **Kills/Assists:** High numbers of kills or assists are a direct measure of combat proficiency. These metrics reflect the player's offensive capability and contribution to the team's success.
- **Damage Dealt:** This captures the overall impact a player has on the match, as it measures not just eliminations but also engagements with enemies.
- **Distance Travelled:** Players who traverse greater distances may have better positioning and movement strategies, which are key to survival and outflanking opponents.
- **Heal/Boost Items Used:** The effective use of healing and boost items is critical to sustaining performance throughout the match, especially in prolonged encounters.
- **Weapon Accuracy:** Players with higher accuracy are more likely to win engagements, making this a crucial metric for evaluating shooting skills.
- **Match Placement:** Final ranking in the match reflects overall success. High placements correlate with both survival and performance across all other metrics.

Most important features



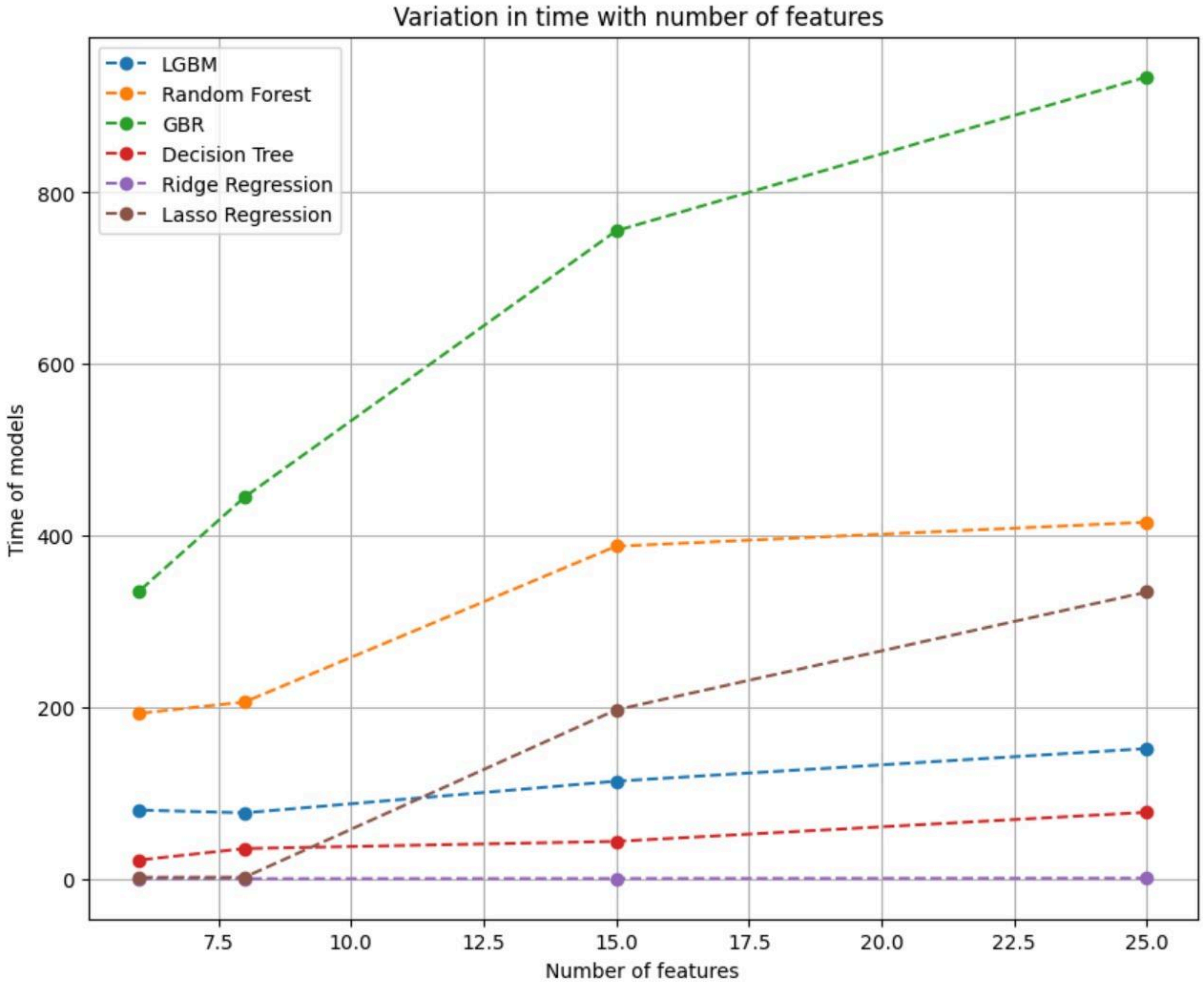
By preprocessing and engineering features in this manner, the dataset is optimized for input into machine learning models, ensuring that all relevant information is captured while minimizing noise and inconsistency.

Conclusion

The analysis of training **time vs. the number of features** shows:

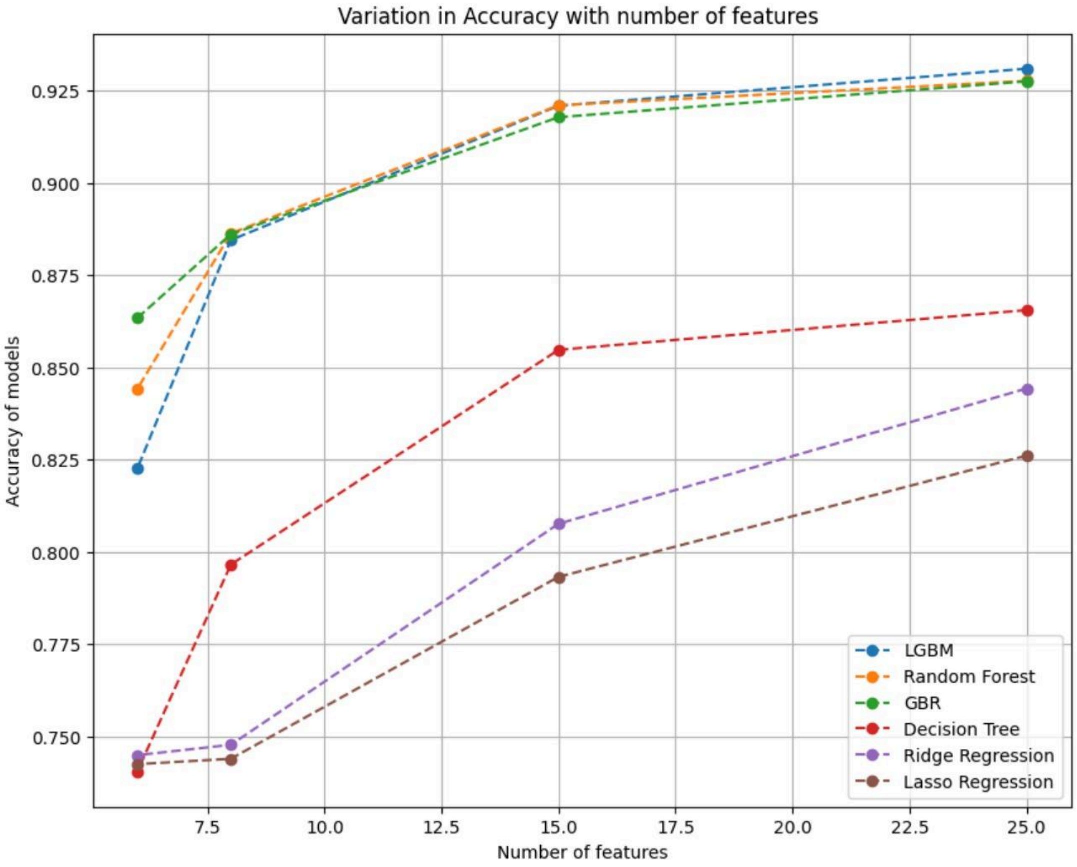
Time vs the number of features

- **LGBM** is the most computationally efficient model, maintaining low training time even with more features.
- **Random Forest** shows stable training times but slightly increases after 15 features.
- **GBR** and **Decision Tree** become slower with more features, making them less suitable for high-dimensional datasets.
- **Ridge** and **Lasso Regression** are computationally efficient, with minimal sensitivity to feature count.



For **accuracy vs. features**:

- **LGBM** and **Random Forest** typically maintain high accuracy as features increase, though accuracy plateaus with too many features.
- **GBR** performs well initially but risks overfitting with too many features.
- **Ridge** and **Lasso Regression** are stable but may underperform in complex feature sets.



Final Takeaway:

LGBM offers the best balance of accuracy and efficiency, making it ideal for large feature sets. **Ridge** and **Lasso** are computationally fast, while **GBR** and **Decision Tree** models become inefficient with higher dimensions. Model selection should balance accuracy needs with computational constraints.