

# Final Report: League of Legends Pro Player Performance Analysis

Faris Aleem Ahmad – 33570  
DSA210 – Spring 2025 – Term Project

---

## Motivation

The goal of this project is to analyze how professional League of Legends players perform in two distinct competitive environments: off-season solo queue matches and official tournament games. By comparing measurable gameplay metrics, this project investigates whether the match setting influences player behavior, risk-taking, and performance consistency.

This topic was chosen to validate / inquire upon the patterns I have noticed personally through my years of viewing these professionals and playing the game myself.

---

## Data Sources

The dataset was compiled using publicly available match history and performance statistics from the following sources:

- [OP.GG](#) – Off-season solo queue match logs
- [Trackingthepros](#) – Player match snapshots
- [LoL Fandom Wiki](#) – Tournament match records
- **Riot Games Developer API** – Used to verify and enrich match details

Each of the 7 selected pro players had 10 tournament games and 10 off-season games collected, for a total of 140 matches stored across 14 CSV files.

---

## Methodology

### Data Preparation

- Merged and enriched all CSVs with calculated performance metrics: KDA, CS per minute, Gold per minute, Damage per minute, and Win/Loss.
- Created a **custom performance score** combining weighted gameplay metrics.

### Exploratory Data Analysis (EDA)

- Used summary statistics and visualizations to explore overall trends.

- Noticed patterns suggesting safer and more calculated play during tournaments (e.g., fewer deaths, higher CS/min).

## Hypothesis Testing

Three statistical hypothesis tests (independent two-sample t-tests) were conducted:

1. **Performance Score** — significantly higher in off-season matches
2. **Deaths per Match** — significantly lower in tournaments
3. **CS per Minute** — significantly higher in tournaments

## Machine Learning

### 1. Win/Loss Prediction

- Applied **Logistic Regression** and **Random Forest** to predict match outcome based on gameplay stats.
- Both models achieved an accuracy of ~85.7%.
- Most influential features: KDA, Gold/min, and Deaths.

### 2. Match Type Classification (Tournament vs Offseason)

- Second ML task trained on the same features.
- Logistic Regression reached 92.9% accuracy with high recall for tournament matches.
- Random Forest achieved 100% test accuracy (but slightly lower CV score).
- Key features: CS, Deaths, and Damage/min.

These results suggest that not only does performance vary between match types, but machine learning models can detect and learn those behavioral patterns.

---

## Key Findings

- Pro players exhibit **safer, more efficient gameplay in tournaments** (higher CS, lower deaths).
  - Off-season games show **more aggressive and variable performance**, likely due to relaxed stakes.
  - Both match outcome and match type can be predicted using in-game stats, confirming the presence of consistent behavioral differences.
- 

## Limitations and Future Work

- The dataset size (140 games) limits generalization across all pro players and metas.
- Tournament games were limited to one per series to reduce matchup bias, but this reduced data volume.
- Future versions of this project could:
  - Include more players across additional roles
  - Track time-based stat progression (e.g., early vs late game)

- Use unsupervised learning to cluster players by playstyle
- 

## Conclusion

This project successfully applied data science techniques—including data cleaning, EDA, hypothesis testing, and machine learning—to investigate how pro League of Legends players adapt to different match environments. The models and results support the hypothesis that match setting influences player behavior in statistically and predictively meaningful ways.

---

## AI Disclosure

- Some summary phrasing, markdown formatting, and report restructuring were supported with AI assistance (ChatGPT), as permitted under the guidelines.
- The method to get the riot API key through their developer website was also assisted by AI assistance(ChatGPT).
- All the codes and decisions to apply the methods were unsupported by AI and purely mine.