#### Challenge Proposal

League of Legends Match Prediction

# 1. What are you going to do?

I will develop an AI-driven predictive model that predicts the outcome of **League of Legends ranked matches**. The model will use both **pre-game data** (such as champion selections, roles, and team composition strength) and **early in-game features** (such as first objectives, early kills, and gold differences).  
  
The system will take real match data collected directly from the **Riot Games API**, stored in a csv/local database and then transformed into datasets. The outcome will be a prediction model that estimates the probability of each team winning, similar to win-rate trackers used in esports.

# 2. Why are you going to do that?

Currently, most win predictions in League of Legends are based only on **raw champion win rates** or **expert coaches analysis**, which limits accuracy. By introducing a **machine-learning solution** that integrates both **pre-game** and **early game data**, I can:  
• Improve prediction accuracy by leveraging thousands of historical matches.  
• Analyze champion strengths and synergies beyond simple win rates.  
• Demonstrate how early-game advantages (objectives, kills, towers) affect win probability.  
• Create an engaging AI project that combines real-world data, machine learning, and esports.  
  
This project is made to show how AI can be applied to competitive gaming, with potential uses in coaching, strategy analysis, and fan engagement.

# 3. Who is involved?

• Riot Games (data provider via API)   
• Myself (student developer)  
• League of Legends players & community (indirect stakeholders)  
• Esports coaches (potential application of results)

# 4. When is it happening?

Planned timeline (10 weeks):

|  |  |
| --- | --- |
| **Timeline** | **Activities** |
| Weeks 1 - 3 | Domain Understanding, Analytic Approach, Data Requirements |
| Weeks 4 - 5 | Data Collection, Data Understanding, Data Preparation |
| Weeks 6 - 7 | Preprocessing, Pre-game Model Training, Evaluation |
| Week 8 | In-game Model Training, Evaluation |
| Week 9 | Demonstration (dashboard, predictions) |
| Week 10 | Feedback, Refinement, Final Report |

# 5. Data Requirements

# To successfully train and evaluate the AI model, I will need to collect and process the following data from the Riot API. These requirements include both pre-game information, lane-level matchups, early-game statistics at 10 minutes, and team objective data.

# Match Metadata

# Match ID: Unique identifier for each match, used for indexing and referencing.

# Queue ID: Specifies the game type (e.g., ranked solo/duo). Ensures only competitive matches are included.

# Game Duration: Total length of the match, used to filter out remakes or very short games.

# Champion & Role Data

# Champion IDs per lane: Records the five champions chosen by each team, with their assigned roles (Top, Jungle, Mid, ADC, Support). This allows analysis of lane matchups and composition strength.

# Lane Metrics at 10 minutes

# Gold Difference per lane: The gold advantage/disadvantage per lane at 10 minutes, a strong indicator of early advantage.

# CS Difference per lane: The difference in minion kills per lane at 10 minutes, reflecting lane efficiency and farming.

# XP Difference per lane: The level/experience difference at 10 minutes, showing who reached power spikes earlier.

# K/D/A per lane: Lane-specific kills, deaths, and assists at 10 minutes, indicating lane dominance.

# Team Objectives at 10 minutes

# First Blood: Which team secured the first kill, reflecting early aggression.

# First Tower: Which team destroyed the first tower, an indicator of map control.

# First Dragon (with type): Which team secured the first dragon, providing early buffs.

# First Rift Herald: Which team secured the first Herald, often leading to tower pressure.

# Tower/Dragon Kills (totals): Cumulative objectives taken by each team at 10 minutes, showing overall control.

# Target Variables

# Match Outcome: Binary label indicating Win/Loss from the Blue side perspective.

# Win Probability: Continuous regression output estimating the probability of winning.

# These requirements will guide the data collection and preprocessing, ensuring that the dataset captures both composition-level and early performance-level information, which together provide a strong basis for predicting match outcomes.

# 6. How am I going to do that?

I will follow the AI Project Methodology:

* **Domain Understanding**: Research champion stats, match objectives, and competitive strategies.
* **Analytic Approach**: Define how regression-based models will be used to estimate win probabilities.
* **Data Requirements**: Specify exactly what data I need from the Riot API (champion IDs, objectives, gold difference, etc.).
* **Data Collection**: Use Riot API to gather real matches and timelines.
* **Data Understanding**: Explore and analyze the data to check consistency, missing values, and patterns.
* **Data Preparation**: Clean, structure, and engineer features for modeling.
* **Preprocessing**: Finalize dataset transformations, scaling, and splits.
* **Modeling**: Train ML model for win prediction, starting with pre-game compositions and early-game features.
* **Evaluation**: Compare model performance, highlight important features, and refine based on results.
* **Delivery**: Provide a working AI prototype that shows win predictions at different stages (draft phase and 10 minutes into the match).
* **Feedback**: Present results, gather input from teachers, and iterate to improve the model and dataset in the next cycle.