Evaluating Models Predicting Game Outcomes Based on Game Data and Player-Champion Experience in League of Legends

By: Richard Dimaria and Megh Bhavsar

Description

League of Legends (LoL) stands as a competitive computer game, holding the title of the most largely played Multiplayer Online Battle Arena (MOBA) globally. The ranked competitive gameplay consists of ten players, organized into two teams of comparable skill that the matchmaking algorithm assembles them into. These opposing teams compete in a battle to destroy the rival team's base in order to win the match. At the beginning of each match, players select a champion; this decision is influenced by the player's skill with that champion. The proficiency a player demonstrates with a particular champion becomes an important factor in their performance. Elevated levels of mechanical expertise give players the ability to thrive in a fast-paced gaming environment, making swift and informed decisions. Once the match has started, there are several factors that can swing the game in either direction. For example, if mid-game a team kills a dragon(a non-playable character), then the resulting team will get a team-wide buff, which increases the team's overall power and helps them win the game. Consequently, a player's mechanical skills significantly influence both individual performance and the ultimate outcome of the match. We feel that being able to predict the outcome of such a complicated game with a lot of different factors will give us a lot of insight on making a model efficient and accurate.

Objective

Our main objective for this project is to implement a single model from both papers and then compare and contrast to see which method worked better and which was more accurate and effective at determining the match outcome.

Sub-Objective

Our sub-objective is to use the model we created for the first paper and then improve on it by adding features used in the second paper to create the best possible model.

Tools

For our project, we will be using a multitude of tools to implement these models. We will be coding on Jupyter Notebook. We will use Python as the programming language, and to actually implement the models, get accuracy, and other related data, we will use Pandas, Keras and Scikit-Learn.

Approach

We will implement the algorithms mentioned in both articles and then compare and contrast them to see which one is better and more accurate. We will then try to combine both methodologies into one algorithm that predicts based on player-champion experience along with the first 10 minutes of the dataset of 10,000 diamond games to try to make an algorithm that is extremely efficient at predicting the outcome of the game. We will accomplish this by taking the game ID from the Kaggle dataset. We will then call the riot api to get data based on the game data, then combine the players experience with their champion along with the first 10 minutes of the game in order to create an algorithm that takes from both solutions.

The first paper gets its dataset from the open-sourced Riot API. This dataset contains general information about the match and the mastery the player has with the champion. This is an integer that represents the amount of points the user has accumulated on that champion. Players get these points by playing games with the champion. It also contains the player's win rate with the champion, which is the ratio between the amount of games they have played compared to the total amount of games they have played. It also includes the amount of games that have been played with that champion in the current season. Seasons typically last for about a year, and every new season, player ranks are reset. Last but not least, it includes the total number of recent games they have played with that champion. These stats will help us determine how good the player is with that specific champion. We can use the familiarity of the player with the champion in terms of all of the stats we pulled from the dataset in order to create an accurate K-Nearest-Neighbour model that can accurately predict the outcome of the game.

The second paper's dataset contains 9879 snapshots of what the game looks like at 10 minutes, with features that lead toward a team's victory. This data can be grouped into separate types of data, such as special monsters killed, which include dragons and heralds. Along with any jungle monsters killed. It will also include a vision score, which includes wards places, wards destroyed. This data will also contain kills and assists on other champions, along with first blood. Last but not least, it will also include information about any towers that were destroyed in the first 10 minutes. All of this information from the first 10 minutes will help us predict what the game outcome will be, as that is very useful information. For example, if a team has a lot more kills than the other team, they are way more likely to end up winning the game as they get more gold, which allows them to buy more items. The same goes for dragons killed, as more dragons killed means that team will get more buffs, making them stronger overall. Most importantly, along with the data, each row also has the game ID. We can use the game-id to get more information about the game, like the player-champions data from the first paper, and combine both algorithms to have a better prediction of what the game outcome should be.

From the article that predicts the game outcome based on player and champion experience, we will implement the K-Nearest-Neighbours model using Python and Scikit-Learn. This algorithm takes a single piece of data and compares it with the k closest data from the dataset used, using distance as the closeness measure. In the second article, we will combine the Kaggle dataset with the dataset from the first article and then apply the K-nearest model to the dataset. With more data for the first model, it should have a better prediction of the game outcome. To start, we will first implement the model from the first and second articles. We will then try to adjust the first algorithm so that it takes in more features from the second dataset. Our final goal is to try to get the first model to be extremely effective and accurate in predicting the game outcome.

Progress

So far, we have created a Jupyter notebook using the Python 3 ipykernel. We then attempted to get the player-champion experience data from the riots website mentioned in the first article, although we had some trouble doing this because the riots API has a rate limit, so we could not make roughly 10,000 calls to them. Our solution was to start making calls now and save all the data we needed in a CSV. There was no trouble getting the diamond-ranked data from the Kaggle website. We imported the Kaggle data using Pandas and displayed it to make sure it was being pulled properly. We also implemented the model from the second paper that uses the first 10 minutes of the game to predict the outcome of the game with 72% accuracy. It has very few lines of code but works perfectly as expected.

Remaining Objectives

Download and create our dataset from Riot Game's API.

Build the KNnearest-Neighbours model to predict game outcomes based on the player-champion experience.

Evaluate both models and compare the differences between them to see which is more effective and why.

Create a new model that combines both previous models so that it produces higher accuracy.

Risks

Some issues that may prove to be a problem are that the Riot API only allows 20 requests per second, so this may cause the amount of time needed to import all of the data to be too much. However, there is a premium option available, but we do not want to spend money on this. Our approach around this is to rate limit our calls to the Riot API and parse the response so we get the data we need. Once we do this, we can save each piece of data on our own CSV file that can be combined with our other data from Kaggle.

References

Using machine learning to predict game outcomes based on player ... (n.d.-a). https://arxiv.org/pdf/2108.02799

Lee, J. (2021, June 23). *Using machine learning to Understand League of Legends*. Medium.

https://towardsdatascience.com/the-path-to-a-victorious-league-of-legends-match-40d51a1a089e