Joyner, Jackson

[Email address]

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

lolesports MVP Prediction Analysis

Table of Contents

**Introduction**

In 2009, Riot Games founded by Brandon Beck and Marc Merrill released a game that would soon become one of the largest video games in the world, League of Legends (commonly referred to as League). While the game was growing to astronomical levels, the interest in a professional level of league was growing at similar rates. This was due to having one of the best in game ranking systems in modern video games. Players joined up and started teams to compete in various tournaments across the world, competing against the best of the best players. As this grassroots tournament scene grew, so did Riot Games’ efforts to support the players in this pro scene. In 2011, the first world championship took place at a gaming convention Dreamhack in Sweden, thus LoLEsports was born.

Throughout the years of LoLesports, there have been 11 different leagues created, with four leagues eventually reaching a franchised league. Each league consists of 8-10 teams that compete throughout the year to earn a spot at the world championship. Throughout the regular season, each league grants a Most Valuable Player (MVP) of their league. In 2021, the North American league, the LCS, granted their MVP to a player named Spica, the Jungler for Team Solo Mid (TSM). A player by the name of Closer, the Jungler for 100 Thieves (100T) was a close second. Personally, I believe that Closer was more deserving of the title. This idea led to finding a statistical method to determine the MVP of a given league for the regular season.

There are some inherent issues when working with League data, as much of the dataset is colinear. As the game progresses, the player passively gains important resources that can be used to gain advantages in damage and damage reduction. This leads to collinearity of Damage, Gold, XP, Total CS and more. While managing the collinearity, is there a statistical way to determine the MVP of a league?

**What is League of Legends?**

League of Legends is an online 5 vs. 5 multiplayer game that was released in 2009 by Riot Games. The objective of the game is to destroy the enemy nexus by dealing damage to it. There are a series of objectives that need to be taken for a team to be able to destroy the enemy base. League of Legends has built a huge following by being a free game, that offers cosmetic transactions to earn revenue. As the game was growing, there was a huge push for this game to be played at a professional level, and in 2011, the first world championship was held, and LoLEsports was born.

Just like traditional sports, League of Legends players play separate positions. The positions are referred to by the lane that the player is in. 

Figure 1: Map of League of Legends

The top lane, the left side of the map, is filled with one player from each team. These players typically play tank or brawlers who have good damage and can take a hit. The mid lane (diagonal lane in the map) player typically plays mages or assassins. These characters have low health and high damage stats. There are two players per team in the bottom lane. One plays a marksman, which is a high-ranged damage dealer and the other plays a support character who is either a healer, or a tank. Then the player who doesn’t have a designated lane to play in is the jungler. In between each lane is the jungle, which is the jungler’s territory. They go from lane to lane helping other teammates. These players typically play high mobility characters that excel in helping others.

As the game progresses, the lanes get waves of minions that grant gold and experience for characters that kill them. The jungle has mobs that grant the same resources in a more condensed fashion. While fighting minions and mobs, the players are also battling their lane opponent, often killing each other players use their gold and experience to buy items and level up. Items grant bonus attributes to their characters, that powers them up. Leveling up grants increased abilities effectiveness. This allows for players to try different strategies, go for different objectives, all effecting the outcome of the game.

**Data Description**

In 2014, there was a huge push for LoLEsports data to be released to the public, and Riot Games agreed to do so. They released an API that allowed people to request data for researching and analysis. This was primarily used by teams and organizations trying to push into the top leagues. Around this time, Tim Sevenhuysen was working with a team in Europe named Fnatic as a data analyst. He used this API to create a website for the public to use called Oracleselixer.com. This website would become the go-to resource for all LoLEsports statistics for fans and up-and-coming analysts alike. In January, I pulled match data from the last 8 years of LoLEsports games from oracleselixers.com to begin this analysis. This match data contains 12 lines of statistics per game; 5 lines for the blue side team players, 5 lines for the red side team players, one line for the blue side team, and one line for the red side team.

This data ended up being 400,000+ rows of data from the years 2014 to 2022. I wanted to only use the data from the top four leagues in this analysis, so the 400,000+ dropped down to (Need to pull the 2022 data still so this will be updated at a later date). From this, I needed to drop any rows with null valuables. This is ok with this data, as there are many more instances of that player playing games. This could be an issue with low game count players, but not enough to break the analysis.

Data Dictionary

Below is a data dictionary for the LoLEsports statistics used in this analysis

|  |  |
| --- | --- |
| GameID | Unique ID for each match |
| Datacompleteness | Identifier if there are null values |
| League | Identifier of which league the match took place in |
| Year | Year in which the game was played |
| Split | Which split of the season the match was played |
| Side | Which side of the map the player started on |
| Player | Players in game name |
| Team | Team name |
| Pos | Position |
| GP | Games Played |
| W% | Win Percentage |
| CTR% | Counter Pick rate: percentage of games in which this player/champion was picked after their lane opponent |
| K | Kills |
| D | Deaths |
| A | Assists |
| KDA | Kill Death Assist Ratio |
| KP | Kill Participation: percentage of teams kills in which player earned a Kill or Assist |
| KS% | Kill Share: players percentage of their teams total kills |
| DTH% | Average share of teams deaths |
| FB% | First Blood rate: for players/champions, percent of games earning a First Blood participation |
| Result | Result of the game: 1 = win 0 = loss |
| GD10 | Average gold difference at 10 minutes |
| XPD10 | Average experience difference at 10 minutes |
| CSD10 | Average creep score difference at 10 minutes |
| CSPM | Average monster + minions killed per minute |
| CS%P15 | Average share of teams total CS post 15 minutes |
| DPM | Average damage to champions per minute |
| DMG% | Damage Share: Average share of teams total damage to champions |
| D%P15 | Share of teams damage to champions post 15-minutes |
| EGPM | Earned gold per minute - excludes starting gold and inherent gold generation |
| GOLD% | Gold Share: Average share of teams total gold earned - excludes starting gold and inherent gold generation |
| STL | Neutral Objectives Stolen |
| WPM | Wards placed per minute |
| WCPM | Wards cleared per minute |
| Double Kills | Amount of double kills in the match |
| Triple Kills | Amount of triple kills in the match |
| Quadra Kills | Amount of quadra kills in the match |
| Penta Kills | Amount of penta kills in the match |
| Total CS | Total creep score for the match |
| GD15 | Average gold difference at 15 minutes |
| XPD15 | Average experience difference at 15 minutes |
| CSD15 | Average creep score difference at 15 minutes |

**Descriptive Stats**

When looking at League data, there are hard floors to most statistics, as a player can’t have negative stats like gold, xp, damage, kills, deaths, assists and more. Items like these are at a baseline of 0 at the start of the game and only go up as the game progresses. With that in mind, there are some interesting stats to look at when it comes to League of Legends.

Typically, when reporting statistics in League of Legends, it is common practice to compare the results of players against other players of the same position. All visualizations will be done in this fashion, as each role has a different purpose, and each position may have varying degrees of utilization of these statistics.

KDA is a ratio of kills and assists to deaths per game. This is a common statistic used to discuss the usefulness of a player in a game, calculated by (kills + assists) / deaths. Sometimes, when a player doesn’t die in game, the ratio would report as an infinite number, as you are trying to divide by zero, in these specific cases, you would just report kills + assists as the KDA ratio. One issue with KDA is it does not indicate if a player is having a bad game vs a terrible game. If a player has a 0/0/0 score line, the KDA will read the same as someone who has a score line of 0/0/12, 0.

Chart, box and whisker chart

Description automatically generated

Figure 2: KDA Box Plot

As Figure 2 shows, positions like jungle, mid and bot have a higher average KDA than top and support players. Bot lane has the largest IQR, as it typically is a more kill and assist position, where supports and top lane players are typically more supportive and die more frequently. With that in mind, it is important to note that supports have the highest KDA at 33. This performance came from the European support, Kold who played for Splyce in 2017. He sported a score line of 5/1/28 (Kills/Deaths/Assists) in a playoff game against Vitality.

One of the best statistics to determine performance in LoLEsports is Earned GPM. Gold is one of the most important factors of player performance in a game of League. Gold allows for players to become more powerful in the game through the purchase of items, which give statistical advantages to the players. Items can range in price from 400 gold to 3600 for a full item. Having a higher Earned GPM gives you earlier access to these important items. With an overall mean of 223.21, standard deviation of 78.92, and a max of 556.56, Earned GPM showcases the importance of gold in League of Legends.

Chart, box and whisker chart

Description automatically generated

Figure 3: Earned GPM by Position

While Earned GPM Is valuable for looking at the game as a whole, the variable Goldat15 takes a look at the early game performance of a player. The early game is the first 15 minutes of the game, typically when the largest advantages are created. Figure 4 showcases how this variable is impactful.

Chart, box and whisker chart

Description automatically generated

Figure 4: Gold at 15 by Position

At first glance, Figure 4 looks very similar to Figure 3. The main differences in these figures are the range of the values, and overall position of each box plot. There is less variability in this metric than in the whole game. Games can take anywhere from 20 – 80 minutes. Having to look at the data as a whole vs at a set point in time allows for more variability in Earned GPM vs Goldat15. This shows by each position have a more squished box in their box plot in Figure 4.

Another metric you can judge early game performance of a player on is the XP at 15 variable. Where gold allows for players to buy items to increase their power, XP allows for players to upgrade their characters in increase their power. One thing to keep in mind with XP is that there is a hard cap on XP unlike gold, where there are unlimited gold opportunities.

Chart, box and whisker chart

Description automatically generated

Figure 5: XP at 15 by Position

Figure 5 showcases some interesting interactions between positions. We have seen the bot position be dominant in other in game performance metrics, and now they are the second lowest in this metric. This is due to the nature of the game where they are paired up with supports in lane, thus reducing their XP gains. The Jungle position is lower ins XP as well, as they get more condensed gains, with lower overall CS.

The final metric that we will use in the analysis is the Earned Gold Per Minute variable. This variable is looking at how efficient a player is with their in-game time. The higher the value, the more efficient they are.

Chart, box and whisker chart

Description automatically generated

Figure 6: Earned Gold per Minute

Positions like Bot and Mid are highly efficient in Earned GPM, as the team usually funnels resources into them. Jungle and support struggle a bit more in this category but are still highly productive. While it would seem that the MVP should just be a Bot position player, as they are the most dominant in most of the variables listed, we actually see something quite interesting.

**Analysis**

For the analysis of LoLEsports data, I decided to run a logistic regression, as I was going to be testing against a binary variable against continuous variables.

Part of the research into how MVP models are created was looking into impact scores, and how different variables affect the score. One way to get an impact score would be to use a fantasy score, like you find in fantasy football. Luckily, there is a fantasy game mode for League called Fantasy LCS. While the dataset didn’t have this statistic already in it, it’s an easy calculation to make. In a Medium article [1], Jonathan Hockey explains that the scoring system for Fantasy LCS is this: (3\*Kills) + (1.5 \* Assists) + (.02 \* Total CS) – Deaths. While this would seem like a good place to start, it was far from the end results. This impact score was only using variables that positions like Bot and Mid thrive in, thus creates a bias in the analysis.

With that bias in mind, the final variables I used in the logistic regression are: Kills, Deaths, Assists, Gold at 15, XP at 15, Earned Gold Share, and Earned Gold Per Minute. As discussed in the data descriptions, these variables give a more comprehensive look at player performance, allowing for positions like Support and Jungle to be able to compete for the MVP.

One thing to note is that I scaled the earned gold share by 100. Before I did this, the coefficients were wildly high, so scaling the variable allowed for easier interpretation of the variables. To produce this model, I used the Stats Models API package in Python instead of SciKit Learn. I chose this as Stats Models tends to be more accurate and gives a better insight into the full stats behind the model. Below is the logistic regression summary table.

Logit Regression Results

==============================================================================

Dep. Variable: result No. Observations: 1600

Model: Logit Df Residuals: 1592

Method: MLE Df Model: 7

Date: Fri, 29 Apr 2022 Pseudo R-squ.: 0.8221

Time: 16:01:13 Log-Likelihood: -197.29

converged: True LL-Null: -1109.0

Covariance Type: nonrobust LLR p-value: 0.000

==============================================================================

coef std err z P>|z| [0.025 0.975]

------------------------------------------------------------------------------

const 1.7328 0.842 2.058 0.040 0.083 3.383

x1 0.2167 0.095 2.284 0.022 0.031 0.403

x2 -1.0179 0.108 -9.405 0.000 -1.230 -0.806

x3 0.3166 0.046 6.872 0.000 0.226 0.407

x4 -0.0009 0.000 -2.888 0.004 -0.001 -0.000

x5 0.0004 0.000 2.563 0.010 9.2e-05 0.001

x6 -0.0155 0.001 -12.330 0.000 -0.018 -0.013

x7 0.1379 0.011 12.678 0.000 0.117 0.159

==============================================================================

Table 1: Logistic Regression Summary Table

Since this model is a logistic regression, the coefficients and confidence intervals need to be exponentiated, which you can see in Table X. To produce this table I used the Numpy package in Python to make an array of the coefficients and then exponentiate that array. I used the same process for the confidence intervals as well.

Table 2: Odds Multipliers of Model Coefficients

|  |
| --- |
| const 5.656599 |
| Kills 1.242022 |
| Deaths 0.361365 |
| Assists 1.372509 |
| Gold at 15 0.999116 |
| XP at 15 1.000391 |
| Earned Gold Share 0.213049 |
| Earned GPM 1.147847 |

Table 3: Odds Multipliers of Confidence Intervals

|  |  |  |
| --- | --- | --- |
| **const** | 1.086094 | 29.460711 |
| **Kills** | 1.031238 | 1.495890 |
| **Deaths** | 0.292295 | 0.446758 |
| **Assists** | 1.253994 | 1.502225 |
| **Gold at 15** | 0.998516 | 0.999716 |
| **XP at 15** | 1.000092 | 1.000690 |
| **Earned Gold Share** | 0.166621 | 0.272412 |
| **Earned GPM** | 1.123638 | 1.172578 |

A logistic regression was performed to determine the effect of Kills, Deaths, Assists, Gold at 15, XP at 15, Earned Gold Share and Earned GPM on the likelihood of winning a professional League of Legends game. An increase of one kill increased the likely hood of winning by 24% with 95% confidence between 3 and 50% increased odds. Each death decreases the odds of winning by 64% with 95% confidence between 71 and 45% decreased odds. Earning an assist increases your odds of winning by 37% with 95% confidence of increasing those odds between 25 and 50%.

Increasing your gold at 15 by one will decrease your odds of winning by .1% with 95%confidence in decreasing the odds of winning between .2 and .03%. Increasing your XP at 15 increases the likelihood of winning by .03% with 95% confidence of .009 and .06%. Increasing your earned gold share by 1 will decrease the likelihood of winning by 79%, with 95% confidence between 84 and 73% decreased odds of winning. Finally, increasing your Earned GPM by 1, increases the odds of winning by 15% with 95% confidence between 12 and 17%.

I found that increasing your Earned Gold Share would decrease your likelihood of winning quite surprising at first, but after a lot of deliberation, I understood why this would happen. Having one person who is performing way above average is not as good for a team, as it means they are taking resources from players, causing them to slump in their performance and not play well. It gives the enemy team an advantage, as they can pick on the underperforming player and exploit that weakness within the team.

This model when trained on the 2014-2021 data from the four main regions of professional League of Legends, tested on the 2022 North American data produces a 94% accurate model.

|  |  |  |
| --- | --- | --- |
|  | Loss | Win |
| Loss | 759 | 41 |
| Win | 48 | 752 |

Table 4: Confusion Matrix for Logistic Regression.

With a model this accurate, it is important to make sure that it is not over fit. To determine this, we look at the ROC Curve of the model. Figure 7 is the ROC curve for this model.

Shape, rectangle, square

Description automatically generated

Figure 7: Roc Curve

For ROC curves, a perfect Area Under the Curve (AUC) would be a 1. This model produces an AUC of .989. This caused some internal alarms, has this is an extremely high AUC. I cross validated my accuracy so I could determine if there was overfitting of the model. In that cross validation, I found the minimum score to be 92%, the mean to be 94% and the max to be 95% accurate. While there is no end all be all rule for overfitting. Between the cross-validation results and the AUC being this high, I would not consider this model to be overfit.

With the model not being overfit, we can finally get to the MVP of the 2022 LCS season. Riot games has a set of rules that comes with being the MVP. You must have played at least half of the season to be up for consideration. For the regular season, that would be 9 games. I calculated the number of games played for each player, and excluded anyone who played less than 9 games this season. To determine who would be MVP, I took the average of each players values predicted by the model for win vs loss. The player with the highest average predicted score would be the MVP for the season.

Table 5: MVP Prediction

|  |  |
| --- | --- |
| Player Name | Predicted Score |
| Eyla | .73 |
| Bjergsen | .72 |
| Santorin | .68 |
| Insipired | .67 |
| Hans Sama | .67 |
| Vulcan | .67 |
| Winsome | .67 |
| Bwipo | .66 |
| Danny | .66 |
| Impact | .64 |

According to the model, the player named Eyla should have been the MVP for the 2022 season. However, Eyla was disqualified from contention, as he didn’t play enough games in the regular season. My model is using pre, regular and post season data to determine the MVP, as I think it gives a better representation of players, as the whole picture is being looked at, instead of just a smaller lens. The Actual MVP Summit is not even in this top 10 list. This is due to his poor performance in the post season dragging his overall stats down. In fact, in the mid-season break, he was dropped from his team because of that poor performance.

Due to the differences in samples, I am proposing a change to the MVP titles in LoLesports. Keep the per split MVP category, but include a season as a whole MVP given after the last international tournament. It would allow for consistent players to be rewarded for their performance.

Add the Appendix, abstract, conclusion, references

