**MINI PROJECT REPORT**

**Recommending Champion Roles Based on Metrics**

TABLE CONTENT

TASK

ABOUT REASEARCHER

1. OVERVIEW
2. Exploratory Data Analysis
   1. Data Preprocessing
      1. Handle missing Data
      2. FeatureExtraction & Selection

2.1.3 Data Storage

* 1. Data Analysis and Evaluation

2.1.1 Choosing important attributes

2.2.2 Statistic

III. FORECAST

IV.Insights & Results

V . Conclusion

VI. References

ABOUT RESEARCHER

***Researcher Background***

My name is Phong , I was born in Phu Yen, with a passion for analysis and learning to code since I was young, I want to become a data analyst to improve my skills.

***Reason for Choosing This Topic***

I have been passionate about gaming since I was a child, especially competitive and strategic games. I have achieved the highest rank of League of Legends and have been a semi-professional player.

A screenshot of a video game

AI-generated content may be incorrect.

*Figure 1 : I got a highest rank for 9 years*

A group of people in a room

AI-generated content may be incorrect.

*Figure 2 : I joined SBTC Academy*

Instead of pursuing a professional career, I wanted to use my knowledge and experience to explore further how data can be used to analyze and optimize in-game performance. With the desire to combine my passion with my data analysis skills, I took on this project to research and better understand the behavior of players at the highest ranks of League of Legends. This was not only an opportunity to improve my data analysis skills but also an important step on the path to becoming a data analyst in the gaming industry.

**I. OVERVIEW**

As players develop a deeper understanding of League of Legends, analyzing performance metrics becomes essential beyond just knowing each champion's characteristics. Evaluating statistical indicators helps players refine their tactics, enhance their skills, and make more informed decisions in every match.

This project focuses on analyzing in-game statistics to recommend suitable champion roles based on key performance metrics, including:

* Gold Per Minute (GPM)
* Damage Per Minute (DPM)
* Vision Score
* Kills
* Assists
* Total Minions
* Longest Time Spent Living
* Total Damage Taken

The project utilizes data analytics techniques, statistical analysis, and visualization to extract insights from match data .By leveraging these analyses, the model suggests champion roles that align with players' strengths and playstyles. This not only helps improve their overall performance but also enhances their gaming experience by making champion selection more intuitive and strategic.

**III.EXPLORATORY DATA ANALYSIS**

**2.1 DATA PREPROCESSING**

Data Preprocessing is an important step to ensure the best quality data before using it in player data analysis.

**2.1.1. Data Source**

The data is obtained from the Riot API, specifically from three ranked tiers:

* Challenger
* Grandmaster
* Master

The API endpoints used include:

* /lol/league/v4/challengerleagues/by-queue/RANKED\_SOLO\_5x5
* /lol/league/v4/grandmasterleagues/by-queue/RANKED\_SOLO\_5x5
* /lol/league/v4/masterleagues/by-queue/RANKED\_SOLO\_5x5

I focused on only the three highest ranks—Challenger, Grandmaster, and Master—because the quality of data at these ranks is higher, with players of consistent skill and strategy, allowing for more accurate performance analysis. Additionally, removing data from lower ranks (Diamond and below) helps limit noise from random factors such as inconsistent players or trolling. Furthermore, the highest ranks reflect the most effective metagame and strategies, providing valuable data for performance and gameplay research. Finally, limiting the scope of the study to these three ranks optimizes resources and processing time, rather than collecting data from the entire vast rank system. All the above data is based on more than 1300 matches of players. Because of the limited performance of personal computers, I only collected people in MID position in North America .

**2.1.2 Data Cleaning**

• Convert JSON data into Pandas DataFrame for easier processing.

• Keep only important columns like puuid (Player Unique ID).

• Sort players by leaguePoints (ranking points) from high to low to select the top players with high rank.

• Combine data from different ranks into a single dataset.

**2.1.3. Handling Missing Data**

During the data fetching process, if the API does not return the match ID (match\_id) or player information (player\_info), the program will ignore those cases and log the error.

Error data is handled with try-except to avoid the program from suddenly stopping.

**2.1.4 Feature Extraction & Selection**

• For each puuid, send an API request to get a list of match\_ids

• Filter players with the MIDDLE position from the match data.

• Generate a list of information about the player, including:

• Personal stats in the match (KDA, kills, assists, damage ,..)

• Champion name used

• Farm stats (CS), gold earned

• Match results (Win/Loss)

• The goal of the project is to analyze roles, I will group the champions according to their respective roles:

•**Mage – Mid** : Use skills to deal damage ( Ability Power )

• **Assassin** : Highly mobile, deals quick damage to take down targets . (e.g : ADC )

• **Fighter** : Combination of damage and resistance, good for prolonged combat

• **Marksman** : Ranged damage dealer, dependent on basic attacks

•**Tank** : High resistance, good at initiating fights, usually goes top or support

• **Mage – Support(\*)** : Support teammates with magical skills, can deal damage or control opponents .

(\*) : A champion I consider a mage-mid if they are a main damage dealer, there are some champions that don't have that ability, only support teammates (e.g Lulu mid,..) then I will consider them a mage-support .

**2.1.5 Data Storage**

After processing the data, I will save them into an excel file for analysis.

**2.2 DATA ANALYSIS AND EVALUATION**

**2.2.1 Choosing important attributes**

Because the dataset has a very large number of columns, based on my experience, I will choose the columns that are really important for the analysis.

More specifically, I will choose the attributes 'visionScore', 'wardsPlaced','wardsKilled','kills', 'deaths', 'totalMinionsKilled', 'neutralMinionsKilled','longestTimeSpentLiving',

'totalDamageTaken' and calculate the columns GPM (Gold Per Minute) and DPM (Damage Per Minute).

Formula: GPM = goldEarned/timePlayed \* 60

DPM = totalDamageDealtTochampions/timplayed \* 60

Explain some attributes :

**Control map :**

**-visionScore**: The player's overall vision score throughout the match. This score is calculated based on the number of wards placed, destroyed wards, the amount of time controlling vision, and other factors.

**-wardsPlaced**: The number of wards the player has placed throughout the match.

**-wardsKilled**: The number of wards the player has destroyed from the enemy.

**Combat & Survival Stats:**

* + **kills:** The number of enemy champions the player has killed during the match.
  + **deaths:** The number of times the player has been killed.
  + **longestTimeSpentLiving:** The longest time the player has survived without being killed in a match.

Farm & Resource Stats:

* + **totalMinionsKilled:** The total number of minions the player has killed (cs - creep score).
  + **neutralMinionsKilled:** The total number of jungle monsters the player has killed, including both the enemy jungle and the team's jungle.

**Damage & Taken Stats:**

* + **totalDamageTaken:** The total amount of damage the player takes from all sources (champions, minions, jungle monsters, towers).

**Player performance :**

**Gold Per Minute (GPM) :** Evaluate a player's ability to earn gold effectively over time

**Damage Per Minute (DPM) :** Measures a player's average damage output per minute, helping to gauge how well they contribute to combat.

**2.2.2 Statistics**

After identifying the attributes to analyze, I will use Correlation to see the relationship between the attributes.

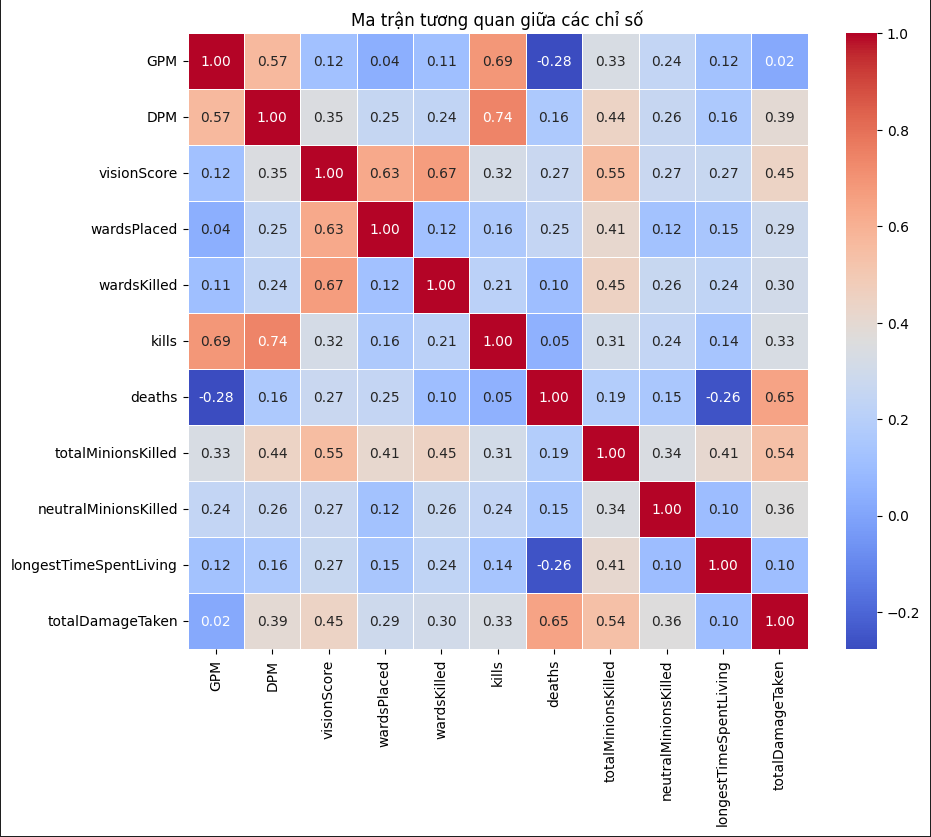


Figure 3 : Correlation attributes

According to the figure , some important information include :

**GPM (Gold Per Minute)**

• High correlation with kills (+0.69): Players with more kills tend to earn gold faster.

• Moderate correlation with DPM (+0.57): Players who deal a lot of damage also earn gold well.

• Negative correlation with deaths (-0.28): Too many deaths reduce gold earning ability due to loss of respawn time

**DPM (Damage Per Minute)**

• High correlation with kills (+0.74): Players who deal high damage tend to have higher kills.

• Moderate correlation with totalMinionsKilled (+0.44): Players who farm a lot may have better damage output.

• Correlation with totalDamageTaken (+0.39): Players who deal high damage tend to also take more damage (due to fighting).

**visionScore**

• Highly correlated with wardsPlaced (+0.63) and wardsKilled (+0.67): Vision points come mainly from the number of wards placed and destroyed by the player.

• Moderately correlated with totalMinionsKilled (+0.55): Players who control vision well tend to farm better.

**kills**

• Highly correlated with DPM (+0.74) and GPM (+0.69): Players with more kills tend to deal more damage and earn gold faster.

• Lowly correlated with totalMinionsKilled (+0.31): Players with more kills don't necessarily farm more.

**deaths**

• Highly correlated with totalDamageTaken (+0.65): Players who take more damage are more likely to die.

• Lowly correlated with kills (+0.05): Deaths don't reflect kills.

**totalMinionsKilled**

• Moderate correlation with DPM (+0.44) and GPM (+0.33): Farming more increases gold generation speed but does not affect damage too much.

**totalDamageTaken**

• High correlation with deaths (+0.65): Taking more damage increases the risk of being killed.

• Correlation with DPM (+0.39): Players who deal a lot of damage often also take a lot of damage from participating in fights.

Analyze the performance of each champion role in the game based on indicators such as GPM (Gold Per Minute), DPM (Damage Per Minute), visionScore, kills, deaths, minions killed, etc.

Calculated statistical values:

mean: Average.

median: Median (middle value).

std: Standard deviation (indicates the level of data volatility).

Conclusion :

**Fighters** have the highest GPM (425.85), indicating their ability to farm gold consistently.

**Mage - Support** has the lowest GPM (363.77), reflecting a support role that is not focused on gold farming.

**Assassin, Mage - Mid,** and **Marksman** have similar GPM (~415-420), which is suitable for resource farming and damage dealing playstyles.

**Mage - Mid** has the highest DPM (846.28), reflecting their role as a strong magic damage dealer.

**Marksman** (819.55) **and Assassin** (809.67)also have high DPM, showing their ability to deal large amounts of damage over time.

**Mage - Support** has the lowest DPM (519.20), as they focus on control and support rather than dealing damage.

**Mage - Mid, Marksman** have the most farmed minions (~55 minions/game).

**Fighter** has the highest neutral minion stat (7.63), due to frequently farming jungle monsters while laning or switching between lane and jungle.

**Mage - Support** has the lowest farm stat (~40), because their role is not focused on resource gathering.

The role with the highest KDA: **Mage - Support** with 2.37 KDA.

Mage - Mid is identified as the role with the highest efficiency.

The role with the highest number of wins:

Mage - Mid leads with 330 wins (46.54%).

Mage - Support ranks second with 207 wins (29.2%).

Assassin has only 101 wins (14.2%), which is significantly lower.

Tank has 39 wins (5.5%), which is quite low.

=> Mage - Mid is not only highly efficient but also has the highest win rate.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4 : analyzed data



Figure 5 : Dashboard

1. **FORECAST**

In this project, I use 3 main models to predict player trends :

• RandomForestClassifier

• GradientBoostingClassifier

• KNeighborsClassifier (KNN)

With the right role for the player, using Classification models, and having Hyperparameters (important to tune for optimal performance) might be the best option.

The prediction will be done by asking the user to enter gameName and ID, based on the available RIOT API. I will create a function that outputs puuid, each puuid will have a MATCH ID containing many important parameters of the player and I will use that data to analyze to give a more suitable role and champion based on the analyzed data.

1. **Results & Insights**

|  |  |
| --- | --- |
| **Model** | **Prediction** |
| RandomForestClassifier | **69.57%** |
| GradientBoostingClassifier | **69.57%** |
| KNeighborsClassifier (KNN) | **59.78%** |

Both Random Forest and Gradient Boosting achieve accuracy close to 70%, while KNN is lower (~60%).

RandomForest vs GradientBoosting: These two models have similar performance, but GradientBoosting has better avg precision macro.

KNN is not a good choice because of low accuracy and poor F1-score for some groups.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 6.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 6.1

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 6.2

**Choosing the best model:**

**GradientBoostingClassifier** is the best choice because of high accuracy and better avg precision macro.

**RandomForest** is also a good choice if faster training speed is a priority.

**Testing Result :**

**A screenshot of a computer

AI-generated content may be incorrect.**

Figure 7

We use the player : ToastyAlex#NA1 (top 1 challenger in NA ) to predict player behavior .

According to the statistics , GPM and DPM is high , It seems that this player has good performance.However, minion is very low , total damage taken and death is quite high , this is also understandable because in addition to focusing on laning, high-rank players often watch the mini map to be able to roam,support jungle to take target ,.., thereby increasing the distance with the mid laner on the other side as well as creating favorable conditions for the lane where they have successfully roamed , and this number can increase with some servers like China , VietNam ,… where early mutation is always most important .

Although the limited amount of data leads to a decrease in the accuracy of the models, the accuracy of this model can be referenced because in reality, this player always prioritizes high mutation generals, in the Assassin class, these comments can be passed through the streams and records of this player. From there, this model can be used with other players for reference.

**V . Conclusion**

The above models need to be improved, adding more independent attributes, as well as adding more samples to make the model complete.

Testing many other servers to give different results, so that the training can be more convenient.

The completed model can create an app or website for players to refer to.

**VI.References**

[Riot Developer Portal](https://developer.riotgames.com/apis)

[GitHub - vkaelin/LeagueStats: 📈 League of Legends Stats Web App](https://github.com/vkaelin/LeagueStats)

[K-Nearest Neighbor(KNN) Algorithm - GeeksforGeeks](https://www.geeksforgeeks.org/k-nearest-neighbours/)

[Gradient Boosting in ML - GeeksforGeeks](https://www.geeksforgeeks.org/ml-gradient-boosting/#:~:text=Gradient%20Boosting%20is%20a%20powerful,previous%20model%20using%20gradient%20descent.)

[Random Forest Algorithm in Machine Learning - GeeksforGeeks](https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/)