Logistic Regression applied on the game "League of Legends" by Chue Zhang

What is this project about?

This project is about the analysis of the different variables in the video game, "League of Legends" and how different sets of variables compare against each other. These variables range from things such as total kills to total deaths with the goal of trying to understand which variables yield the greatest accuracy when trying to predict the outcome of a match and why.

Why this project?

This project was done to analyze a specific player match history in hopes of being able to show that player(Me) what variables contribute the most in predicting the outcome of the game.

What is League of Legends?

League of legends is a 5 vs 5 arena game where those 5 players have the same amount of resources to use, different champions to pick and must use different strategies in order to win. Games generally last between 20 minutes to over an hour and each of those different champions that the player chooses to have unique abilities that may change battles completely or may contribute to nothing. It is also important to understand there are different roles in the game that each person chooses to play and must stick to during the entirety of that match they are playing.

How do you win?

You win the game by destroying the enemies nexus Each team has a nexus with the pre-requisites of having to destroy the enemies inhibitors inhibitors and nexus turrets before being granted the ability to destroy the enemy nexus. Both teams have these structures that start with the same health points and the same attack points, so fairness is preserved for each game regardless of whichever side of the map you end up spawning in. There are different objectives in the game such as Dragons that help players get certain buffs and gold which help players buy items to strengthen themselves.

Constraints

<u>Data</u>

1Rather than gathering the data of every single player present in the game, this data is only for one specific player. This is important to note since teammates are variables that contribute to the win/loss rate as every player has a different mindset and skill level.

<u>Time</u>

Time is integral in this game as many objectives such as mobs have different spawn/respawn timers and depending on when a player acquires certain objectives boosts overall speed and efficiency. It is also important to note that some champions may perform better later into the game (I.E: Past 25 minutes) and some may perform better earlier in the game, performing worse later in the game.

<u>Champions and Lane</u>

In this game there are different lanes and with different lanes there are different roles and each champion play towards a different role. Most of the data retrieved was for a special gamemode that disregarded the lane and everyones champion was the same.

Why the Constraints important

The importance of listing constraints is so that this proejct could be taken with a grain of salt, this data may not be completely accurate. Incorporating those constaints into the project would most definitely improve the accuracy the predictions however, in order to maintain a level of simplicity, these variables were not used for comparison.

What are the dependencies of this project?

Riot API

- What is Riot API?

    1. Riot API is the interface used to retrieve data for this project. None of the data used was fabricated as the player in question has legitamately played through 100 different rounds of the game

Riot watcher

- What is Riot watcher?

    1. Riot watcher is the module used in this project to simply the methods of retrieving data

Pandas

- What is pandas?

    1. Pandas is a library used for data maniuplation and analysis, specifically it was used in this proeject for creating dataframes and exporting data as .csv files

Code For retrieving Data

```
def matchHistory(num):
    #==== just getting the match details, ranges from 0th - 99th match hen
ce "num" ====#
    last_match = my_matches['matches'][num]
    match_detail = watcher.match.by_id(my_region, last_match['gameId'])
    #==== just getting the match details, ranges from 0th - 99th match hen
ce "num" ====#

    participantsID = []

    #==== match_detail has to be looked through in a for loop because it i
s a list ====#
    for r_name in match_detail['participantIdentities']:
    #==== match_detail has to be looked through in a for loop because it i
s a list ====#

        participants = []
        for row in match_detail['participants']:
            #==== Im doing this so sift out my play info only, not the ent
ire teams ====#
            if(r_name['player']['summonerName'] == "Desuuuuuuuuuu"):
            #==== Im doing this so sift out my play info only, not the ent
ire teams ====#
                p_name = {}
                p_name['name'] = r_name['player']['summonerName']

                participants_row = {}

                #==== from match_details, i am using teams and inside team
s, the amount of dragons killed====#
                for temp in match_detail['teams']:
                    if(r_name['player']['summonerName'] == "Desuuuuuuuuuu"
):
                        participants_row['dragon kills'] = temp['dragonKil
ls']
                 #==== from match_details, i am using teams and inside tea
ms, the amount of dragons killed====#

                #==== from match_details, i am using gameDuration ====#
                g = match_detail['gameDuration']/60
                participants_row['game duration'] = "{:.2f}".format(g)
                #==== from match_details, i am using gameDuration ====#

                #==== getting info from match_detail['stats'] ====#
                participants_row['turret kills'] = row['stats']['turretKil
ls']
                participants_row['inhibitor kills'] = row['stats']['inhibi
torKills']
                participants_row['gold earned'] = row['stats']['goldEarned
']
                participants_row['first blood'] = row['stats']['firstBlood
Kill']
                participants_row['champion'] = row['championId']
                participants_row['kills'] = row['stats']['kills']
```

Sample data for one match

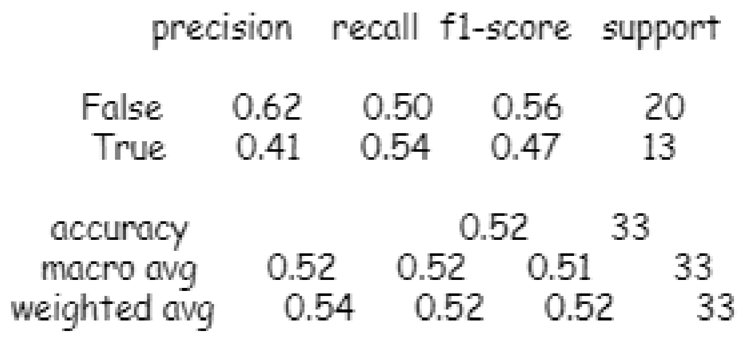
Kills	kill assists	Deaths	Dragons	first blood	game duration	gold earned	inhibitor kills	total Damage Dealt	turret kills	Win
2	1	2	1	False	11.80	5201	0	14541	0	False

Logistic Regression on League of Legends

Logistic Regression was chosen as the base model for predictions as there are many different variables that require comparison in order to determine the importance of each variable. As described in Statistics Soltuion, "Logistic regression was made to describe data and to explain the relationship between one or more variable" which fits the criteria for what is trying to be achieved in this project.

Code for training the dataset

```
X = data[['kills', 'kill assists']]
y = data['win']
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.33, r
andom_state=101)
logmodel = LogisticRegression(solver = 'lbfgs')
logmodel.fit(X_train, y_train)
predictions = logmodel.predict(X_test)
print(classification_report(y_test,predictions))
```



```
              precision    recall  f1-score   support

       False       0.62      0.50      0.56        20
        True       0.41      0.54      0.47        13

    accuracy                           0.52        33
   macro avg       0.52      0.52      0.51        33
weighted avg       0.54      0.52      0.52        33
```

"X" can be any of the 10 variables [kills, kill assists, deaths, gold earned, first blood, dragon kills, game duration, turret kills, inhibitor kills] "y" is always [win]

Table for all tests done

Kills	Kill Assists	Deaths	Damage dealt	gold earned	first blood	dragon kills	game duration	turret kills	Inhibitor kills	Lose	Win	Accuracy
✓	✓									0.62	0.41	0.52
✓	✓		✓							0.67	0.41	0.45
	✓	✓								0.85	0.77	0.82
	✓	✓	✓							0.80	0.69	0.76
✓		✓								0.78	0.60	0.70
✓		✓	✓							0.72	0.53	0.64
✓	✓	✓								0.82	0.62	0.73
✓	✓	✓	✓							0.82	0.62	0.73
				✓	✓					0.57	0.37	0.45
					✓	✓	✓			1.00	0.50	0.61
				✓		✓	✓			0.88	0.48	0.58
				✓	✓	✓	✓			0.88	0.48	0.58
✓				✓	✓	✓	✓			0.86	0.58	0.70
	✓			✓	✓	✓	✓			0.86	0.58	0.70
		✓		✓	✓	✓	✓			0.79	0.53	0.64
								✓	✓	0.75	0.53	0.64

✓								✓	✓	0.75	0.53	0.64
	✓							✓	✓	0.73	0.50	0.61
		✓						✓	✓	0.75	0.62	0.70
			✓	✓	✓	✓	✓	✓	✓	0.86	0.58	0.70
		✓	✓	✓	✓	✓	✓	✓	✓	0.74	0.57	0.67
	✓		✓	✓	✓	✓	✓	✓	✓	0.86	0.58	0.70
✓			✓	✓	✓	✓	✓	✓	✓	0.65	0.44	0.55
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.78	0.60	0.70
	✓	✓				✓	✓	✓	✓	0.94	0.80	0.88

Analysis

Initial analysis

The accuracy of the comparisons were low but were within the range of expectations as there were many variables not used(Constraints) that grant flexbility in creating different sets.

Highest and Lowest accuracy

The highest accuracy achieved was 0.88 where the variables used were **['kill assists', 'deaths', 'game duration', 'turret kills', 'inhibitor kills','dragon kills']**. This means that for the subject being analyzed, these specific variables contribute the most winning the game and therefore should focus on these objectives more. The lowest accuracy achieved was 0.45 where the variables used were **['gold earned','first blood']** which brings up the question "why do these variables weigh so little as compared to the variables used to acheive the highest accuracy?"

Weight of variables

The reason why certain variables weigh more than others is becuase they may have correlation with one another. For example, you can earn gold from killing the enemy team therefore, gold earned is directly relates to getting kill assists and getting killed by the enemy team. Looking at the overall situation, everything except for the game duration is related to the gold earned as those variables all grant gold once performed and so gold earned is a variable dependent on things that things that give gold whereas **['kill assists', 'deaths', 'game duration', 'turret kills', 'inhibitor kills','dragon kills']** are independent variables (do note that dependency classification is strictly for this project as if the constraints mentioned above were to be applied, there would be different dependency classifications). But something interesting to point out is why Kills weigh less than Kill assist? Getting kill nets a player more gold than kill assists so why? Thats because for this person, kill assists were obtained more frequently than kills so this means that although there were less kills, there was more kill participations. Kill assists are given when an enemy is killed while another player helps therefore it is harder to measure kills but easier to measure kill assists as kill assists were more frequent than kills. Furthermore, the way total gold earned can be visualized is points on a graph that are seemingly random. The same thing could be applied to first blood as even though this person may not have achieved first blood, it didn't affect the outcomes of win/loss outcome therefore, the visualization of first blood could also be seen as random. Lastly, first blood is also related to gold as you earn more gold from getting first blood, so gold is dependent on first blood and first blood is dependent on getting the fastest kill.

Some variables VS all variables

Understanding that some variables are more random and do not contribute to outcome as much, its easier to understand why using all variables for comparison yields a lower accuracy than using a certain selection of them. Looking at the table **['Kills', 'damage dealt', 'gold earned', 'first blood']** were not used because there is greater disparity in these variables. This means that these 4 variables when applied were too scattered therefore the accuracy is lowered. The goal is so that accuracy would be as highest as possible therefore the need for numbers less scattered were required to increase accuracy.

Unknown variables

We have known variables, but some unknown variables may also interfere with predicting the outcome. One variable that is important is the skill level. It is one thing to achieve a certain rank that may or may not represent your skill level, but its another thing considers that not everyone may be performing at their best therefore this is an unknown variable. A person may end up losing a game perhaps because they are not in the mood or are not playing something, they are comfortable with. Furthermore, another unknown variable would be luck. There is such thing as critical strikes which have % chances of landing so if a person is lucky enough, they could turn entire fights around. There are many more unknown variables such as bugs however, these unknown variables are much more prevalent in the game therefore, they are considered in the increase or decrease in accuracy.

Conclusion

What was found in this research was that certain variables are more dependent on each other however; more variables are required to achieve greater accuracy. Furthermore, there are also some unknown variables that may end up dictating whether a game is won or not so sometimes, accuracy may be scattered all over the place. However, the findings in this research were still remarkable because as a person who plays this game, it can be confirmed that these are variables that player a bigger role in contributing to a win.

Game Defintions

Champion

There are 148 different champions to pick as the character you control for the match, each having their own unique abilities

Lane

There are 3 lanes, the Top lane, Mid lane and Bottom lane. Each lane performs a different role

Roles

1. **Mid Laner** - Person who competes with the enemy in the Mid lane. A very flexible role that allows the player to have many more options of champions to pick from
2. **ADC(Attack Damage Carry)** - Person who specializes in doing physical damage to the enemy team and is generally a person who has ranged attacks
3. **Support** - Person who makes sure the ADC does not die or makes sure the ADC gets kills
4. **Top laner** - Person who competes with the enemy in the top lane and is often the tank(person who can take in damage)
5. **Jungler** - Person who has no lane, starts in a jungle and gets his resources there. The jungler is in charge attacking enemy lanes whenever they want to, think of this as a psuedo support. They are also in charge of getting monster objectives such as dragons

Buff

A buff is a term used to describe an increase in power, temporary or not

Dragon

A monster objective, comes in 5 different varieties and each having different buffs that are extremely beneficial for the entire team

First Blood

The first kill in the match, grants extra gold upon completion

Critical Strike

Critical strikes are regular attacks that deal 200% of normal damage and starts with 0% rate of occuring but may be increased with items bought in the shop