

# Project D4: Predicting LoL winner

Team members - Veronika Kukk, Robin Piir, Otto Kase

Github - [https://github.com/VeronikaKukk/DS\\_project](https://github.com/VeronikaKukk/DS_project)

## Glossary

Term	Meaning
Riot Games	Creator of League of Legends
League of Legends	Multiplayer online battle arena (MOBA) genre video game
EUNE	League of Legends game server (Europe Nordic & East)
EUW	League of Legends game server (Europe West)
Rank	Way of comparing the skill level of players. Similar to Elo in chess.
API	Application programming interface - essentially a way for two or more computer programs to communicate with each other.
API key	A way to identify the user that is trying to use an API.
Kill	Reducing an enemies health to 0
Assist	Helping with a kill
Level	The level of the player. Each new level gives a bunch of strength. Start at 1. Max is 18.
Experience	Used to level up
Warding totem/Ward	An item that a player can put on the map to reveal the nearby area
Minions	NPCs that belong to both teams. They give gold to the player that kills them.
Jungle monsters	Neutral NPCs that give gold and experience to the player that kills them.
Towers	Structures you have to destroy to reach the enemy base in order to destroy it.
Elite monsters (dragons/herald)	Monsters with high durability that give big bonuses to the team that slays them.

## **Identifying our business goals:**

### **Background.**

We decided to do a project about the videogame League of Legends because it is a game that we like and we want to find out more about the game. The main idea of our project is to predict game winners based on the first ten minutes of the match.

### **Business goals**

Our first goal is to try to predict the winning team of a game based on the first 10 minutes of data from a ranked game in Diamond I elo. We are planning on using machine learning algorithms on the dataset to predict the winner. In addition to predicting the winning team based on the first 10 minutes, we will be comparing Europe Nordic & East server (EUNE) and Europe West server (EUW). The reason for that is that there has always been a debate about which European server has better players. Since we are comparing players in the same rank (Diamond I), it should mean that players that have reached this rank are on the same level in terms of being good at the game. We want to investigate whether this hypothesis is true or not.

### **Business success criteria**

In our mind the project will be successful if our trained model consistently predicts better than a random chance.

## **Assessing your situation:**

### **Inventory of resources.**

At first we had an idea of using the data from a Kaggle dataset (<https://www.kaggle.com/datasets/bobbyscience/league-of-legends-diamond-ranked-games-1-0-min>), but we have opted for using data that we have collected by ourselves from Riot API. The decision was made, because the Kaggle dataset has data that may not be relevant anymore since it is from at least 3 years ago (we do not know for sure, because there was no indication of when the data was collected). In addition the Kaggle data is from the North American server that is not interesting to use for the reason that we are playing League of Legends in the European servers (EUNE and EUW). The data resources for this project are collected from Riot API (<https://developer.riotgames.com/>) by making API requests. As for expert knowledge we will contact our practicals teacher if any help is needed. Other resources include our own computers and Python libraries (we are not yet sure which ones exactly, because it is a learning process).

### **Requirements, assumptions, and constraints.**

Since the deadline is the 12th of December, we are planning on finishing the project on the 10th of December, to leave some additional time to review that everything that is needed is completed. The legal obligations are to not monetize our project, since we used the development API key for data collecting. Our project is not meant for public consumption.

### **Risks and contingencies.**

The only problem that we can foresee at this moment is that we may need some assistance from the practice session teacher with making the winning team prediction more accurate. We think that most of the labs and homeworks have already prepared us for the project and these materials will be also used if any problems occur.

### **Terminology.**

See the Glossary.

### **Costs and benefits.**

There are no monetary costs or benefits for the project, because we will be using materials that are public and for free. No monetary benefits are allowed, because we are using data that is requested by using the development API key that does not allow us to monetize the project. Each team member will contribute at least 30 hours of work to the project. The benefit of the project is for us to summarize what we have learned during this course and to fortify our knowledge about data science, to gain insight on making a project come to life from our own idea, to satisfy our curiosity regarding the EUNE and EUW comparisons and predicting the winning team of a game.

### **Data mining goals:**

1. Develop a model for predicting the winning team.
2. Determine the most accurate learning algorithms for our problem.
3. Find correlations between specific game data attributes and the match result.
4. Compare EUNE and EUW servers

Data mining success criterium: the model predicts the outcome of a match better than a roll of a dice would. There are no right or wrong results for the 4th goal, we count our goal to be a success if we get some insight on whether the servers have similarities or not.

### **Gathering data:**

#### **Outline data requirements.**

All of the data that we have collected is numerical because as we have seen almost all of the information that Riot collects from the games is numerical and there is no need for us to

change that. We have the same columns of data as the Kaggle dataset because we agree that these characteristics describe the game in the best way possible. There are probably more columns that can be used to help us but at this moment we do not see any more that are needed.

### **Verify data availability.**

We got the idea of gathering the data from Riot API because of the discussion under the Kaggle dataset. The Kaggle dataset was old and not relevant to us, so we have gathered new data for us to use. Since the mentioned data was collected from the Riot API, we decided to do the same. The development API keys are available for anyone that has an Riot account and this key has to be updated every 24 hours. The gathering was time consuming because of the request rate limits, but we got the amount of data that we decided is enough for this project. Narrowing the scope of the project means that we are only using data from EUNE and EUW servers (there are more servers, ex. North America).

### **Describing data.**

Model for goal 1- using the first 10 minutes of game data to predict the outcome of the match

The goal of this model is to predict the winning side (red or blue) of a match based on data gathered from the first 10 minutes of said match. The data necessary for this is obtainable through the Riot API, therefore it is available to us. For this, we wrote a program in Python to collect the necessary information into CSV files. To achieve the model's mission, we need to train it with attributes that directly correlate with victory. League of Legends is a very complicated game, so there is a lot of data being gathered from each game and a lot of small differences in the teams' performances can accumulate to a big one. Therefore it is quite natural that the list of attributes used for training this model will be a long one. The exact list of columns that we will use will be determined later once we investigate correlations.

Comparisons for goal 4- comparing game data from EUNE and EUW servers to find which one is stronger

The goal of this is to objectively determine the stronger of the two regions within Europe since it has been a highly disputed topic in the community for a very long time. Realistically speaking, this mission is a very hard one to complete since there is much more nuance to it than simply game data. The best case scenario for this would be to have data that dwelves deeper - for example something that compares the rank of the same player on two different

servers. However, this kind of data is not available and impossible to gather so we will make comparisons on the data that is available to us and see if we can reach a conclusion based on that. The overall data for this goal will be the same as for the previous one, again, the exact columns we use will be decided later.

### **Data description report**

Since the Riot API has rate limits for each user, the time it took for our program to collect all the data was quite high, limiting the amount that we could gather for use. However, we believe we have plenty to achieve our goals. In total, we gathered 4 dataframes from each server, where each file holds about 800 KB of data. This might not seem like a lot, but it is actually around 16000 games per region, giving us a plethora of data to train the models with. There are 39 attributes in the dataframe, the datatype being numeric for all of them.

### **Exploring data**

The 39 total attributes consists of 18 per team, as well as the outcome of the match described by a "1" or a "0" based on whether the blue team won or not. The 18 columns of each team consist mostly of stats that should be helpful in training a model to determine the winner of the match. The specific attributes for each team are the following: Wards placed, wards destroyed, first blood (first kill of the game - 1 or 0), kills, deaths, assists, dragons, heralds, elite monsters (in total), towers, total gold, average level, total experience, total minions killed, total jungle monsters killed, gold difference, experience difference, minions/monsters killed per minute, gold per minute.

Immediately it is clear to see that some of the data is redundant, since one attribute directly determines the other. For example, the amount of kills on the blue team is, for the most part, equal to the amount of deaths on the red team. The gold/experience difference of one team is equal to the gold/experience difference of the other multiplied by -1. The amount of elite monsters killed is simply dragons + heralds.

### **Verifying data quality**

We think that the data that we have collected is good enough to support our goals. While requesting the data we have already filtered out games that are not useful to us. This includes games that are less than 10 minutes or where the data was corrupted (request resulted in an empty game, request timed out, game data missing). The amount of games that we wanted were available for us so there were no problems with that. For example when first requesting data we had an error in our code that read the winning team wrong from the request. This

mistake resulted in all of the games being marked as the blue team had lost. This mistake has been corrected and we are sure that the quality of data is good.

### Planning your project

We are planning on using Github to upload and share the code and data. The code will be written in the Jupyter Notebook. We want to limit the project to be only written in Python and only use the resources that Python provides for coding. As for the algorithms we want to try out the ones that we have learned in the course and also research other machine learning algorithms that could be useful for our project. We have not yet done the research because there are many steps that need to be done before choosing learning algorithms and it is a part of our plan to test out different ones.

Task	Description	Time per person (h)	Comments
Homework 10	Write the reports for the homework 10	3	
Gather data	Write code that requests the data from the Riot API. Create CSV files with EUNE and EUW server games data	2	
Combine data	Combine each of the servers data into one dataframe	1	
Clean data	Remove any games that have corrupted data. Balance the data if it is needed	1	Most of this should be already done in gathering data task
Divide the data into training, test and validation	Divide the data and choose the sizes of these sets.	1	
Create a model	Build a model for predicting the winning team using machine learning algorithms.	3	

Optimize the model	Choose the most accurate algorithm on the test set and optimize the hyperparameters	3	
Validate model	Validate the best performing algorithms on the validation set to find the accuracy of models.	2	
Summarize the work progress of building a model	Make conclusions based on the best performing algorithms and summarize the work that has been made.	2.5	
Compare the EUNE and EUW servers	Compare the game attributes from both servers. Find whether there are similarities.	4	This includes graphs and written comparisons.
Summarize the work progress of comparing servers	Make conclusions based on the gathered information about the servers and summarize the work that has been made.	2.5	
Clean and format the results	Make the conclusions clear and easy to read. Make the code readable and add any additional comments that are needed.	3	
Create the poster	Create the poster that summarizes the project	3	