Project member:

Bin Zhang (5660329599)
Hanzhi Zhang (4395561906)

Project description

This is a project related to the Multiplayer Online Battle Arenas assistant tool. Here we choose League of Legends (LOL) as the carrier to launch our project. The main purpose of the project is to crawl tens of thousands of matches of LOL, to analyze the match's detail information, make recommendations for champions (characters) selection, item selection and predict the win rate of each team. Here are the details of the project:

1. When two teams are starting a new match, the first process is champions selection, where each team should ban five ones and pick five ones in turn. Our system would recommend champion in each turn for teams based on preference of user, opponent bans, and opponent picks (This is implemented by clustering and networking).
2. During the matches, our system would recommend which item the user should purchase based on his golds, opponents' equipment and historical data.
3. Our system will also stimulate the winning rate prediction based on the current match data. Even though we don't have a real-time match information, but we can implement this module by historical data and deploy it when the new data comes in.

Innovation

1. There are some applications doing similar jobs. However, none of them take full advantages of the match details. Traditional recommendation systems are mostly used the experience of pros players. Some systems such as op.gg might use the historical data to produce a high-win-rate build for a specific champion. However, every match is different. Thus, a rigid build cannot be suitable for all situations. In our project, we not only regress a high-win-rate build from historical data, but also make some modifications based on the current situations.
2. Traditional prediction of winning rate is based on gold numbers and kill numbers of each team, which doesn't fully take advantage of the match information. Here our project will dive into the match details and look through the timeline information of the match and find which team is dominant at a specific group battle. In this way, we could get a more accurate prediction rate.

Pose and motivate your hypotheses.

For this project, we aim at helping users to pick the right champions and equipment’s based on their preference and the lineup attributes. Besides, we want to trend the match, predicting which team will be the winner. Here are our hypotheses:

1. We assume that matches are independent and identically distributed, which means that we can put these match data into some machine learning algorithms.

2. We assume that win of loss of a match not only depends on players' proficiency, but also match itself objective attributes, like lineup strength, ban-pick state, time period, and even some latent attributes. The objective conditions are what we are interested in. We can extract worthy information from lots of match data and operate data mining. In this way, we can prove or disprove the influence of some match attributes.

Literature review

Recently, Multiplayer Online Battle Arenas (MOBAs) is popular over the world. As representatives, Dota2 and League of Legends (LOL) has risen a lot of witness. Some researchers started to analysis the match information generated by players' battle. There are interesting topics related to research like tracking match trend, predicting winning team and analysis champion's properties.

There are some applications analyzing match information, such as, Dota Picker, Winsight, which help counter overall strategies by suggesting “counter- picks” to the enemy team’s lineup in Dota2.

However, analysis for LOL is not too much. A few researchers apply Naive Bayes Classifier to do research on wining rate prediction in LOL. There are other research using neural network to build a model for LOL to make recommendation without giving structure of model. Some research focuses on data itself and try to use traditional methods like average data, standard deviation and correlation to explore the match information.

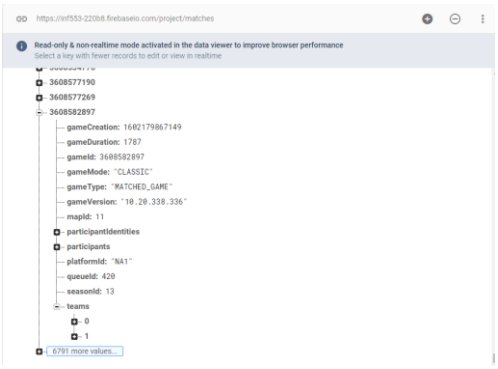
Deviation from the proposal

1. At the beginning, we plan to use PyQt to implement the visualization module. Now we decided to build a frontend html to stimulate the real-time operation;
2. We remove the mini map monitor module;
3. We add an equipment recommendation module;
4. We update our idea about how to predict the winning rate of each team from gold number statistics to deep analysis;
5. After doing some analysis on the dataset provided by Kaggle, we find the data is outdated and don't have too much valuable information. So, we decide to collect all data by ourselves via Riot API and web scraper;

The reason why we make some changes is that after further communication, both of us think that we should focus more on data mining instead of content displaying which may spend too much time.

Initial results

1. Write spider to crawl the match information via Riot API and have got 6k+ matches' information, store them into Google Firebase real-time database



2. Preprocess the match data by picking some important attributes and export it into csv file. Here are all champion features extracting from the matches

	gameId	gameDuration	championId	assists	kills	damageDealtToObjectives	damageDealtToTurrets	damageSelfMitigated	deaths	goldEarned	totalE
0	3017019005	1807	92	1	2	15514	4156	16343	-3	10173	
1	3017019005	1807	268	6	7	10747	6700	8507	-5	12013	
2	3017019005	1807	143	16	2	3867	1249	4562	-8	8896	
3	3017019005	1807	59	16	10	31144	2368	22353	-2	14153	
4	3017019005	1807	15	14	8	19962	9972	7533	-4	12849	
...	...	...	...	...	...	...	...	...	...	...	...
66256	3608223245	1036	202	18	13	716	716	8361	-7	12262	
66257	3608223245	1036	145	20	12	1309	1309	9788	-4	11747	
66258	3608223245	1036	37	32	8	2010	2010	13650	-5	11635	
66259	3608223245	1036	115	21	8	2227	2227	4071	-5	11412	
66260	3608223245	1036	114	11	4	1725	1725	25004	-8	9933	

66261 rows × 15 columns

3. Based on champions information, build a clustering system to get similar champions, we use two different methods:

a. Using champion's basic information to do clustering, here when I input champion's name like 'Yasuo', system will return top10 similar champions as recommendations

```
similar_champions('Yasuo')
```

```
Out[350]: [("Cho'Gath", 0.9883182366699904),
           ('Amumu', 0.9840738144214867),
           ('Zac', 0.9815528682064433),
           ('Shen', 0.9804517803728238),
           ('Urgot', 0.9786000667733094),
           ('Warwick', 0.9784939194028227),
           ('Trundle', 0.9756721409949738),
           ('Thresh', 0.9748297218889925),
           ('Garen', 0.9743201780485502),
           ('Volibear', 0.9740191121963685)]
```

b. Using word embedding to extract champion's skill description and convert it into vectors and do topic modeling, the following image shows related champions in a topic

```
**1**:['Aatrox', 'Ashe', 'Camille', 'Darius', 'Fiora', 'Garen', 'Graves', 'Illaoi', 'Kindred', 'Master Yi', 'Miss For
tune', 'Wukong', 'Nocturne', 'Olaf', 'Pantheon', 'Quinn', "Rek'Sai", 'Renekton', 'Sett', 'Shyvana', 'Sivir', 'Skarne
r', 'Tristana', 'Trundle', 'Tryndamere', 'Varus', 'Vi', 'Warwick', 'Xayah', 'Xin Zhao']
==========================================================================================================================
```

Roadmap

obstacle

1. Due to the Riot API's access limit (50 request per minutes), we cannot get large amount data in a short time. We update a python file on AWS, and crawl data for 24 hours.
2. Current, more 3 Gbit data from Riot API is collected. The problem, how to process the large amount data with a limited memory size (8G)? Spark or Hadoop might be used in the future work.
3. How can we evaluate our final system? An evaluation function should be built to measure the accuracy of our model. For example, if our system tells us that team 1 have 70% winning rate in this situation, we want to know the accuracy of prediction. In our historical data, only win or lose is showed.

anything left to do

1. Apply clustering algorithms like KMeans, GMM to do clustering job for champions and items and use neural network to do winning rate prediction for each team.
2. Deal with TimeLine Dataset, extract useful information and incorporate with processed data to do deep data mining.
4. Develop a Web UI to let user input and stimulate the real match and display the results generated from our system.
5. Improve our codes and try to make it more efficient and accurate.