

Mini Project 2

TECHNICAL Report

Apache Spark



Khalid Abdelnabi Alabyad, 201700500

Moaaz Elsaid, 201600819

INTRODUCTION

This project uses Apache Spark to analyse the data derived from Riot Api.

In this project, we faced many and many problems and were able to solve every piece of problem perfectly, thanks to God.

The project data and codes can be found in Drive with the following link:

[BigData MiniProject 2](#)

MATERIALS

1. Data from Riot Api
2. coding

PROCEDURE

1. Derived the Data from Riot Api using Riot Watcher api on python
2. Then we cleaned the data and organized them in sectors
3. Then used these Data to analyse and get the results needed using the pyspark library on google colab.
4. Then the output was saved and organized in csv files to be used again for visualization and other staff

DATA

The data collecting stage was tiring and took so much time. Although we used many riot accounts to request data faster.

The collected data are about 180,000 matches

The codes used for collecting data are in file: ../Data/Codes

As there are several codes to collect data from several servers and every server for highelo and lowelo ranks.

The codes are straight and commented.

The output data can be found in the drive folder :

../BigDataMiniProject2/CollectingData/Data

The format of the output is a .json files

Every .json file have matches for the specified server and rank

Analysis

We used pyspark library on google colab to do the analysis required for the project

The code for the analysis is found in : ../Analysis/SparkAnalysis.ipynb

AparkAnalysis description:

- It has several functions for every requirement
- All the functions has one argument which is a json string for one match
- functions are
 - `get_champion_win` >> this function output every champion for the winning team for every match
 - `get_champion_pick` >>> This function outputs every champion for both teams for every match
 - `get_championsID_bans` >>> this function outputs every champion id that is banned from both teams
 - `get_itemsID_pick` >>> this function outputs all the items Ids picked from both teams for every match
 - `get_items_win` >> this function outputs the items selected from the winning team
 - `get_champions_synergies_win` >>> this function outputs a combination of the champions for the winning team for every match
 - `get_items_champions_synergies` >>> this function outputs a combination for every item selected by one player and the champion selected by the same player for both teams for every match
 - `get_items_picks_for_champion` >>>> this function is used for the purpose of items suggestion to use it and the other function for win
 - `get_items_wins_for_champion` >>>> this function is used for the purpose of items suggestion, it is used to get the win rate and scores for every item have been selected by a specific champion

- `Get_champions_vs_times >>>>` This function is supposed to print all champions that have faced each others
- `Get_champions_vs_won >>>>` This function is supposed to print all champions that have faced each others it results a list of tuples and every tuple have two champions the first champion is from the winning team and the second from the losing team

- Then the above functions are used by the pyspark library to get the analysis and the output is saved in a text file.
- Another step is used to map all the ids(items, champions) from the results to their original name with the use of Datadragon File

Results

The results are obtained from the output files from the analysis stage.

The output from the analysis stage is formatted into a csv files to help visualize the output and the results

The code for this stage is found in `../Output/making_csvFiles.ipynb`

`making_csvFiles.ipynb` Description:

- First it loads the wins and picks files of champions then both are saved in a DataFrame then another column is added which is the win rate based on the win column and the pick column
- Another column is merged into the data frame which is banns for the same

champions.

- The DataFrame is saved into a csv file for every rank(highelo, and lowelo)
- The same code is used for items and synergies

The output files are very good for visualizing and for other purposes

And here are some snapshots from it:

C	D	E	F	G	H	I	J
Champion	pick	win	win_rate	bans	Best Enemy, winRate	Worst Enemy, winRate	
Jhin	16366	8280	50.592692166687	1707	('Nocturne', 58.844765342960294)	('AurelionSol', 37.735849056603776)	
Graves	12332	6248	50.6649367499189	18409	('Illaoi', 75.0)	('Skarner', 37.096774193548384)	
Jinx	11619	6006	51.6911954557191	2600	('Azir', 64.86486486486487)	('AurelionSol', 40.816326530612244)	
LeeSin	10924	5295	48.4712559502014		('Malzahar', 59.61538461538461)	('Shyvana', 35.36585365853659)	
Yone	10335	4979	48.1761006289308	12662	('Illaoi', 67.3469387755102)	('Rammus', 32.6530612244898)	
Ezreal	10285	4983	48.4491978609626	2121	('Yorick', 60.86956521739131)	('DrMundo', 34.355828220858896)	
Lucian	10104	4912	48.6144101346002	10683	('Azir', 59.4059405940594)	('Neeko', 38.333333333333336)	
Lulu	9779	5032	51.4572042131097	12825	('Neeko', 61.76470588235294)	('AurelionSol', 39.02439024390244)	
Viego	9631	4745	49.2679887862112	7087	('Varus', 61.25000000000001)	('Yorick', 27.27272727272727)	
Caitlyn	9567	4732	49.4616912302707	3951	('Illaoi', 60.0)	('Akali', 39.76608187134503)	
Lux	9427	4688	49.729500371274	2889	('Rammus', 60.97560975609756)	('AurelionSol', 34.0)	
Camille	9119	4713	51.6832986073034	17699	('Nasus', 62.5)	('Quinn', 36.84210526315789)	
Thresh	8806	4393	49.8864410629116	6888	('Garen', 59.154929577464785)	('AurelionSol', 37.5)	
Talon	8686	4441	51.1282523601197	24670	('Neeko', 62.16216216216216)	('Reli', 40.8)	
Yasuo	8424	4170	49.5014245014245	8519	('Olaf', 63.74999999999999)	('Yorick', 33.33333333333333)	
Nami	8194	4141	50.5369782767879	5987	('Illaoi', 64.86486486486487)	('KogMaw', 40.909090909090914)	
Vayne	8193	4329	52.8377883559136	7055	('Neeko', 72.91666666666666)	('Yorick', 36.17021276595745)	
Viktor	7763	4135	53.2654901455623	7973	('Malzahar', 70.37037037037037)	('Rammus', 38.297872340425535)	
Jayce	7575	3713	49.016501650165	2129	('Illaoi', 63.63636363636363)	('Reli', 35.59322033898305)	
Irelia	7472	3626	48.5278372591006	15442	('Varus', 59.45945945945946)	('AurelionSol', 38.63636363636363)	
Ekkko	7314	3632	49.658189773038	4846	('Udyr', 63.07692307692307)	('Rammus', 22.857142857142858)	
Zed	7254	3612	49.793217535153	18489	('Neeko', 67.5)	('Nasus', 37.333333333333336)	

A	B	C	D	E	F
	champion	pick	win	win_rate	bans
0	Ezreal	19404	9406	48.47454133	7757
1	Lux	18489	9318	50.39753367	9233
2	Jhin	17552	8956	51.02552416	5697
3	Yasuo	16513	8311	50.330043	23975
4	Caitlyn	15404	7497	48.66917684	8969
5	MissFortune	14899	7548	50.6611182	
6	Vayne	14587	7588	52.01892096	15297
7	LeeSin	13573	6505	47.92602962	
8	Yone	13568	6789	50.03685142	19277
9	Jinx	13216	6656	50.36319613	3037

	Item	pick	win	win_rate
0	Ionian Boots of Lucid	131201	68252	52.02094496
1	Plated Steelcaps	127811	63952	50.03638185
2	Control Ward	114535	52412	45.76068451
3	Sorcerer's Shoes	107369	54876	51.10972441
4	Berserker's Greaves	96593	48955	50.68172642
5	Refillable Potion	71717	31687	44.18338748
6	Zhonya's Hourglass	67874	37262	54.89878304
7	Immortal Shieldbow	65640	33959	51.73522243

	champions	wins
0	Jhin**Lux'	1090
1	Caitlyn**Lux'	1041
2	Ezreal**Lux'	981
3	Lucian**Nami'	940
4	Caitlyn**Morgana'	870

	Champion**Item	picks
0	<u>Jhin**Galeforce</u>	15328
1	<u>Jhin**Boots of Swiftess</u>	12574
2	<u>Jinx**Berserker's Greaves</u>	10946
3	<u>Yone**Berserker's Greaves</u>	9923
4	<u>Yone**Immortal Shieldbow</u>	9474
5	<u>LeeSin**Goredrinker</u>	9241
6	<u>Graves**Immortal Shieldbow</u>	8963

And also there are graphs done on tableau software for the previous results :

Graphs can be found in : ../Output/graphs

So here are all the project requirements that can be brought from the previous results :

- i. Champion win, pick, and ban rates (for highelo and lowelo)
- ii. Champion duos (for highelo and lowelo)
- iii. Item win, pick rates (for highelo and lowelo)
- iv. Item Synergies (item with champion) (for highelo and lowelo)
- v. Item suggestion (for Jhin and Lulu)
- vi. More requirements : champions vs each others top picks, champions vs each others win rates, best enemy and worst enemy for every champion based on win rate

Challenges and solutions

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Sed diam nonummy nibh euismod