

# 369 Final Project

TeamFight Tactics Dataset Analysis

David Z. - Hannah H. - Tony L. - Zhisong L.,





# **What is TFT or TeamFight Tactics?**



## Background on TFT

# TEAMFIGHT TACTICS™

TFT is a 8 player online game that involves a wide roster of units called champions.

Each champion has a corresponding cost, rarity, and tier.

Beyond this, special modifications known as Augments, Items, and Traits add further complexity to the game.

It's a free-for-all game, meaning players can get any placement between 1st and 8th place

Each round every player plays another random player



**Where is the data from?**



# Riot Developer API

The Riot Dev API allows for access to data from games such as League of Legends, Valorant, and TeamFight Tactics.

Information such as player info, player match history, match metadata can all be obtained.



**How was the data used?**



# What does the original data look like?

```
{
  "tier": "GRANDMASTER",
  "leagueId": "3512903d-d261-3891-b12a-169e134d8fbf",
  "queue": "RANKED_TFT",
  "name": "Shaco's Scions",
  "entries": [
    {
      "summonerId": "oM9tAMwCG-oYtPFCHarMh_mGFQIGiJKctSBosdCZIXFewfs",
      "summonerName": "Jordn",
      "leaguePoints": 570,
      "rank": "I",
      "wins": 174,
      "losses": 140,
      "veteran": false,
      "inactive": false,
      "freshBlood": false,
      "hotStreak": false
    },
    {
      "summonerId": "0brV7Ata6vPFa2QPjZ1-PCKwQURA-uyhcHuMLTJ-b7UtwIY",
      "summonerName": "CAMILE BOT PV",
      "leaguePoints": 519,
      "rank": "I",
      "wins": 146,
      "losses": 123,
      "veteran": false,
      "inactive": false,
      "freshBlood": false,
      "hotStreak": false
    }
  ],
}
```

[illegible]





# Parsing Methodology



# Step 1: Maps, Maps, Maps

The JSON match data that we have has many fields, most of which aren't important to our unit prediction algorithm

We split and mapped the JSON to only grab the “participants” field, which contains info about where each player in the match placed, along with the units they used to obtain that placement

This list of units was also parsed here, in order to remove the irrelevant info

Example: data post parse

```
"NA1_4504812291", 4, ("TFT7_Sylas",2,2), ("TFT7_Seraphine",2,2), ("TFT7_DragonPurple",5,2), ("TFT7_Zoe",6,2), ("TFT7_Bard",6,2), ("TFT7_AurelionSol",7,2), ("TFT7_Soraka",6,1)
```

match\_id, placement, units\_list



## Step 2: Remove Duplicates

Players tend to match with those of similar rating points to themselves, which means that repeated match information may occur if two or more players in the top 250 play against one another.

Since our data became formatted in a [(match\_id, placement), (unit info)] style, any duplicate entry for a given (match\_id, placement) was removed.



## Step 3: Unit Aggregation

Our data was now in a form of match\_id, placement, unit\_list

We needed to remove units from the list, in order to be able to get data on each individual unit

This was done by mapping each unit in the list the placement associated with it

Example:

```
"NA1_4504812291", 4, ("TFT7_Sylas",2,2), ("TFT7_Seraphine",2,2), ("TFT7_DragonPurple",5,2),
```

Became

```
("TFT7_Sylas", 2, 2, 4), ("TFT7_Seraphine", 2, 2, 4)...
```



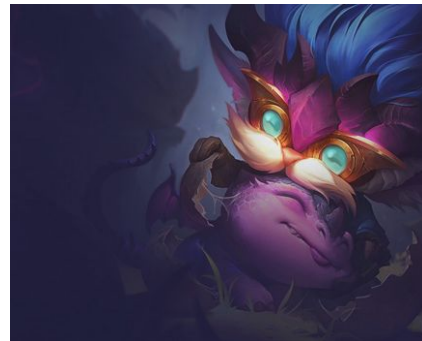
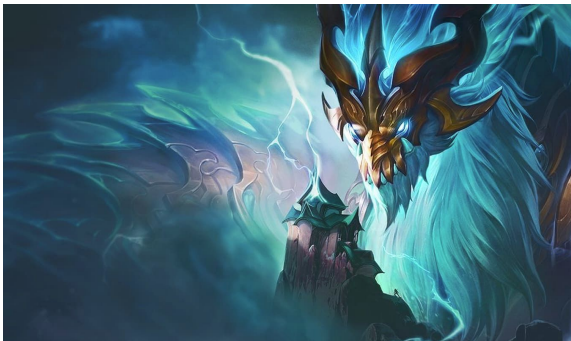
## Step 4: Placement Generation

- Since we now had a RDD of units with their placements for 25 matches each containing 8 players for a subtotal of 200 rows of units + placements.
- These rows were then split by (unitName, tier) to be individual rows
  - (this process was also run on the other 249 players' 25 match history).
  - This would result in a generated file containing all units grouped by (name, tier) containing their expected placement.

## Step 5. Adding Weights to our Values

While we had the average placement of each unit, we needed to add weights, as some units contribute more to the end placement of a whole board than others

Ex. Having a dragon like Ao Shin which costs 8 gold (rarity of 7) has a greater effect on your board, than a unit that costs 1 gold (rarity of 0).





# How our data can be used





# Match Prediction

The primary use of our parsed data is to predict how well a board of units will place. Since we have how well each unit does individually, our added weights can try to show how much a unit will affect the board

Often in game, a player would want to know how strong their current board of units is, and how much stronger it could get if a unit was swapped for another, or how strong a potential opponent's board is


















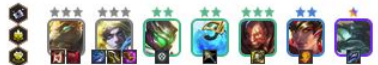










# Demo

# Example Match for Comparison

Date 12-05-2022 09:21 (v12.22)

Rank	Summoner	Round	Alive	Traits	Units	Left Gold
1	 RC1	6-6	38:07			2
2	 KurumHeroAugment	6-6	38:07			4
3	 Hezii	6-5	37:11			53
4	 xyk000	6-3	35:52			1
5	 cannon minion 33	5-6	31:47			63
6	 magic sail	5-3	29:07			2
7	 Xujhan	5-2	27:25			8
8	 kSpRaymond	5-1	26:27			20



# 4th Place

Result: 4.285

Program Output:

```
JOINED DATA:  
DragonSwain, 2, 5, 4.02172, 1.86200  
Qiyana, 1, 1, 4.38353, 1.33300  
Rell, 3, 1, 3.68537, 1.33300  
Aphelios, 2, 1, 4.33263, 1.33300  
Graves, 2, 4, 4.13037, 1.77400  
Rengar, 2, 2, 4.66242, 1.52800  
Rengar, 2, 2, 4.66242, 1.52800  
(53.28940684302459,12.4355995000000002)  
4.285230224970221
```

4



GM xyk000

6-3

35:52



1



# 7th Place

Result: 4.522

Program Output:

```
Braum, 2, 1, 4.61564, 1.33300  
Lillia, 2, 1, 4.57766, 1.33300  
Wukong, 2, 0, 4.59915, 1.00000  
Sylas, 2, 2, 4.40348, 1.52800  
Diana, 2, 2, 4.60852, 1.52800  
Yone, 3, 1, 3.91291, 1.33300  
Olaf, 2, 2, 4.83972, 1.52800  
4.521559619047801
```





## 2nd Place

Result: 4.579

Program Output:

```
JOINED DATA:  
Zoe, 1, 6, 4.73114, 1.93600  
Lillia, 2, 1, 4.57766, 1.33300  
Taliyah, 2, 0, 4.33833, 1.00000  
Sylas, 2, 2, 4.40348, 1.52800  
Vladimir, 3, 0, 4.41362, 1.00000  
Lux, 1, 1, 4.87234, 1.33300  
(44.59449802844655, 9.739689)  
4.578636754053086
```





# What We Learned





# Learnings

- Using the Riot Dev API to collect data
- JSON Filtering in scala/spark
- Filtering and mapping large amounts of data to only get the important information



# Major Obstacles







# Obstacles

- Initial grabbing of JSON from Riot API
- Deciding how to weight placement values in order to assess how a certain unit affects the eventual weight
- In a game of TFT, there are many factors other than the units that affect how well a player performs



# Future Work





# Improvements upon the process

There are many factors that contribute to the expected placement of a given player beyond just the champions chosen.

- Augments, Traits, Opponents, Items

These could be somehow factored in to get a more accurate estimate

We could also potentially look at other regions unit placements, as different regions have different metas and play different variations of boards