DOTA 2

Predicting Outcomes from Drafts

Samuel Campbell

What is DOTA?

- Multiplayer online game played 5 versus 5 (Radiant / Dire)
- Each team selects 5 heroes from a pool of 115
- Players play individual heroes drafted
- Objective: Destroy opponent’s Ancient
- Game has been described as much more complex than Chess and GO
- In 2017, Elon Musk, got a team to create an AI to play world’s best mid player in a 1 versus 1 scenario.

Answering what Tesla didn't

Is it possible to predict the victor by the end of the drafting phase and before the game begins?

- Answer to this question enables us to know if a Min/Max algorithm is feasible for drafting heroes.
- Give us better insight on team strategies and matchups.
- Guide newcomers during game play as they are overwhelmed with choices.
- Similar research peek around 61% prediction accuracy

Data collection

Key Features: Player Stats, Overall Hero stats, Hero roles, Team Composition, & Team synergy

- Online datasets on Kaggle are outdated → Game is patched every few months with game changing elements
- Match & Hero data collected via dota2api
  - Max of 10 000 calls a day and 50 000 calls a month
- 10 000 games collected from API
- Player stats scrapped from DOTABUFF with GET requests
  - Around 50 000 points of player information
- Additional hero information obtained from Liquipedia

Player Stats & Overall Hero Stats

- Player stats are recorded on DOTABUFF
    - Kill / death / assist
    - Win rate per hero
    - Number of times played per hero
- No external API → Must scrape HTML content
- Problem: Some players enable a feature to hide their stats
- Solution: Scrape overall monthly performance of heroes
    - Replace Nan values of player stats with this information

Hero Roles

- 7.45e+13 ways of drafting
  - Classify heroes to reduce drafting complexity
- Used to enrich Hero DataFrame with their roles
  - Data not well formatted
  - Names in the file do not match with rest of the dataset → Entity Matching required
  - Need regex to extract hero entities

```
{| class="wikitable"
!colspan=8| ■■■ Carry
|-
|{{HeroIcon|am}}{{HeroIcon|arc}}{{HeroIcon|ck}}{{HeroIcon|gyro}}{{HeroIcon|medusa}}{{HeroIcon|morph}}{{HeroIcon|naga}}
{{HeroIcon|pa}}<br>{{HeroIcon|sniper}}{{HeroIcon|spectre}}{{HeroIcon|tb}}{{HeroIcon|tiny}}{{HeroIcon|troll}}
|-
!colspan=8| ■■ Carry
```

Entity Matching

- Every Hero possesses a Name & Title
- Most heroes have abbreviations and acronyms created by the DOTA community
- Need an Entity Matching dataset
    - Standardizing Pipeline: Title | Name | Acronym | Nickname → Title → Lower case

Title: Storm Spirit
Name: Raijin Thunderkeg
Acronym: SS
Nickname: Storm

Hero Roles Cont.

Durable	Jungler	Pusher	Support	Escape	Disabler	Initiator	Nuker	Carry
0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1

- Heroes can fulfill more than 1 role
- Roles represented by a 1 hot vector
- Roles can also be represented as integers
    - Nominal data to Discrete numerical data

Vector:

0	0	1	0	0	0	0	1	1

Integer: **67**

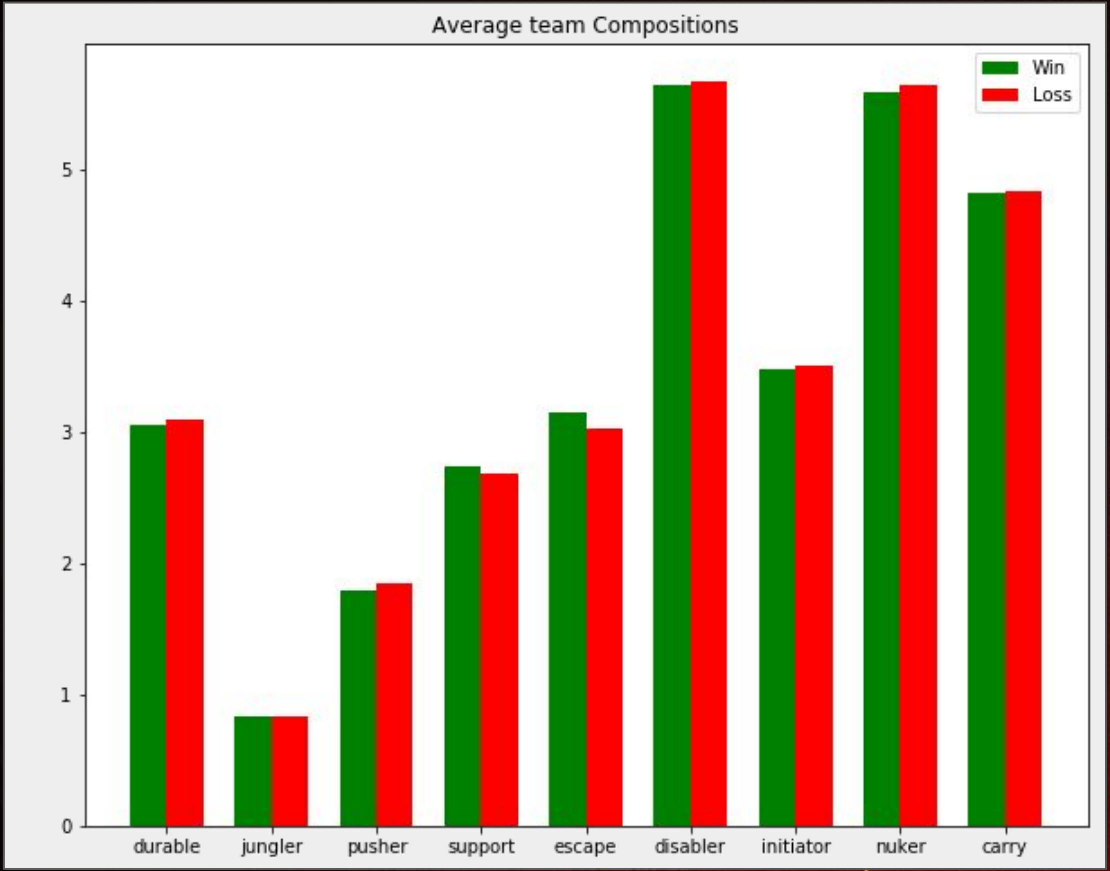
Team Compositions

Team compositions can be represented by the sum of the columns of every hero's 1X9 vector on a team.

	0	1	0	1	1	0	0	0	0
	0	1	1	1	0	0	0	0	0
	0	0	0	0	0	1	0	1	1
	0	1	1	0	0	1	0	0	1
+	0	1	0	0	0	1	0	0	0
=	**0**	**3**	**2**	**2**	**1**	**3**	**0**	**1**	**2**

9^6 ~= 10^6 Compositions!

Team Synergy: In Search of Hidden Links



- Team compositions too similar between winners and losers
- Possible explanation: Players understand good team synergy but do not understand different matchups

Team Synergy: Correlating Roles with Impact

Attempted finding correlations between the hero roles and their impacts on the game. Objective is to find a team classification heuristic.

- Game Duration
- Tower Status

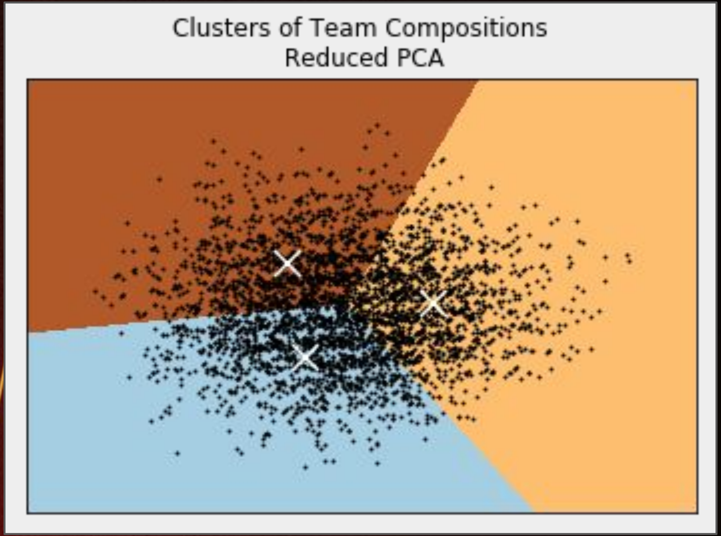
	carry	disabler	durable	escape	initiator	jungler	nuker	pusher	support
Duration	-0.002249	0.055559	0.001986	0.019569	0.048365	-0.00803	0.024151	-0.103928	-0.002974
Tower Score	0.054621	-0.006784	0.006066	-0.014536	-0.053806	-0.02382	0.037321	0.007540	0.000699

No meaningful correlations. Most of them are around 0. Quit this approach after 2 attempts.

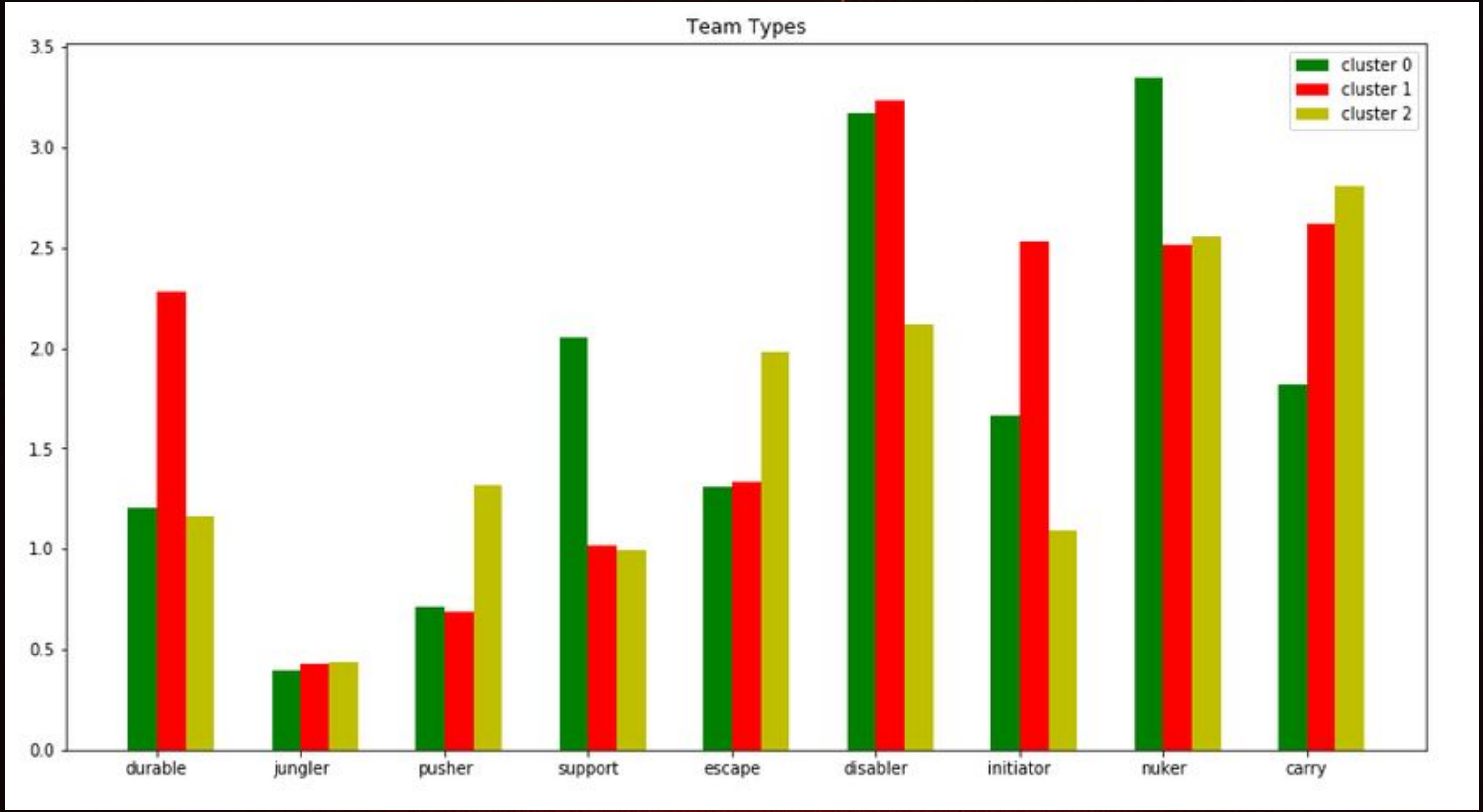
Team Synergy: Unsupervised Learning

Desperate Attempt

- According to community players, the strategies are:
    - Pushing
    - Turtling
    - Ganking
- Use K-Means clustering with n = 3
- Dimension Reduction with Principal component Analysis → Project clusters
- 'X' are centroids

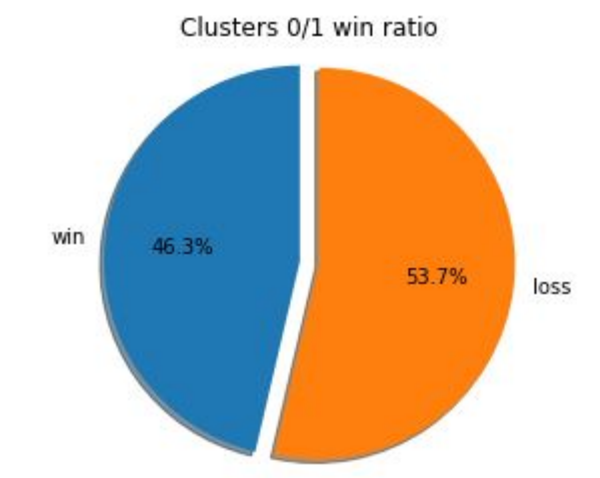


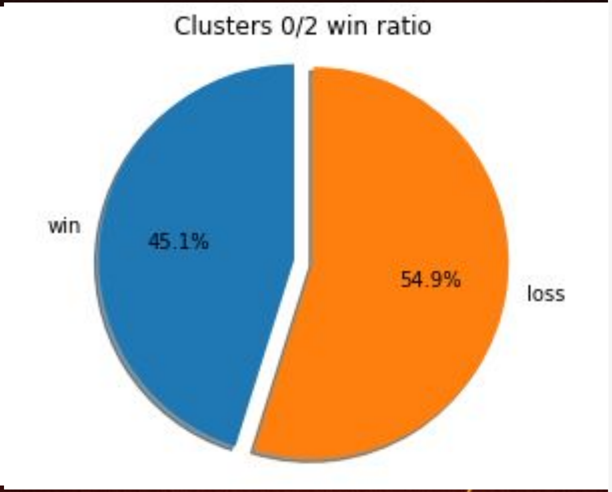
Team Synergy: Clusters

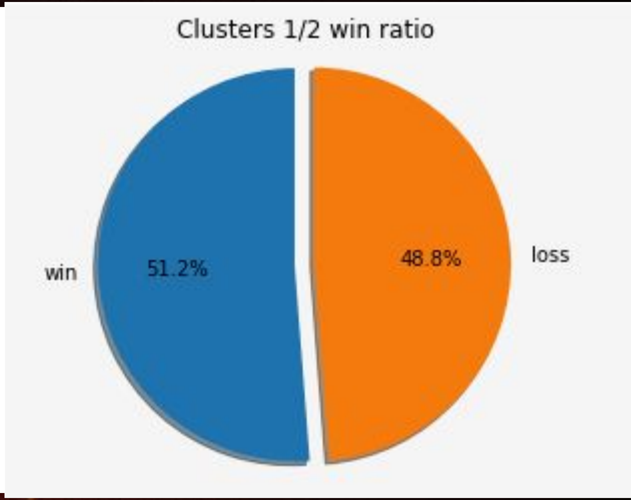


Team Synergy: Comparing Matchups

If the clusters are good, then we should see biases in the team matchups win ratios since this is considered to be rock/paper/scissors.







Training Features

First Configuration:

1) Players win ratio their hero (float) [10 columns]
2) Hero community win ratio (float) [10 columns]
3) Hero role (integer) [10 columns]
4) Team Synergy (integer -- index to cluster) [2 columns]

Second Configuration:

1) Hero community win ratio (float) [10 columns]
2) Hero role (integer) [10 columns]
3) Team Synergy on both sides (integer -- index to cluster) [2 columns]

Outcome Prediction

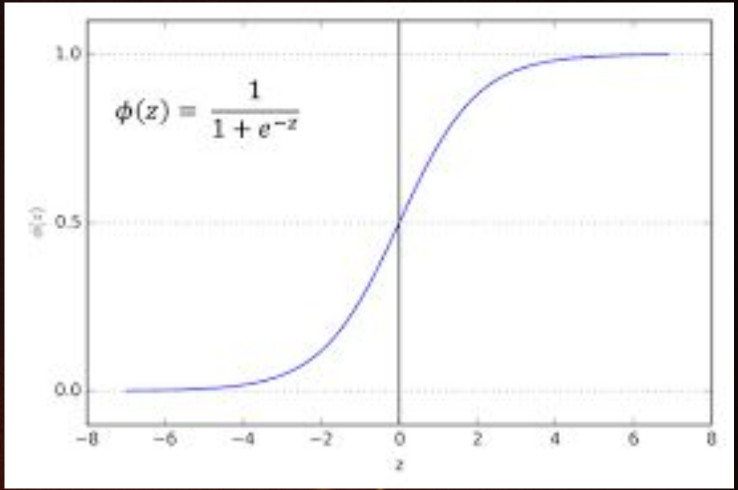
Machine Learning Technique: Logistic Regression

First Configuration:

- Accuracy: 0.61
- Precision: [0.58, 0.63]
- Recall: [0.56, 0.64]
- F1: [0.57, 0.63]

Second configuration:

- Accuracy: 0.63%
- Precision: [0.59, 0.65]
- Recall: [0.60, 0.65]
- F1: [0.60, 0.66]



Strange!! The results show that the player's win ratio with a hero isn't a good metric..

Conclusion

1) Able to predict with 63% accuracy which team will win before gameplay.
2) 63% accuracy achieved from a dataset only 4% of the size used to create the model of 61% accuracy. (better features found)
3) Team Synergy, Hero selection, and Hero overall complexity + strength (represented by win ratio) are good features to predict outcome.
4) It was possible to accurately represent a complex drafting process consisting of 7.45e+13 combinations as a 1X9 vector. → This is the defining feature of this research.. (increased accuracy by 3-4 %)