Insights of Dota2 Hero and Match Data

Yuan Gao, Rui Fang,
Shujian Wen, Jiahuan Yu

What is Dota 2

- Dota 2 is a multiplayer online video game.
- Five heroes on each side.
- Only one side wins.
- Each hero is different!
- Ban/pick, choice of heroes, etc. are important.

Outline - What Have We Done?

1. Clustering Model of Hero Types (K-Means Clustering)
2. Match outcome influencing factors
    a. Match outcome prediction (Decision Tree, Logistic Regression and k-Nearest Neighbors)
    b. First blood & win/lose relationship (Regression)
3. Pick rate & win rate relationship for heroes (Pearson's r)

Data tables

1. **matches**: one match is one record

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	match_id	match_s	radiant_wi	start_time	duration	tower_stati	tower_stati	barracks_s	barracks_s	cluster	first_bloo	human_play	leagueid
2	7.29E+08	7E+08	FALSE	1403182330	3224	1536	1846	0	63	225	65	10	1284
3	1.98E+09	2E+09	TRUE	1449396864	1476	1983	1796	63	51	224	126	10	3877
4	1.89E+09	2E+09	TRUE	1445649594	3063	1972	0	63	0	121	251	10	3781

2. **player_matches**: one player/hero in one match is one record

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	match_id	account_id	hero_id	item_0	item_1	item_2	item_3	item_4	item_5	backpack_0	backpack_1	backpack_2	kills
2	3521803363	138885864	108	231	226	90	36	127	46	0	0	0	5
3	956361773	131237305	43	100	119	116	50	46	114	0	0	0	6
4	1106986271	93944475	40	48	90	1	108	0	1	0	0	0	9

3. **heroes**: one hero is one record & **picks**: each match/hero is one record

	A	B	C	D	E	F	G
1	id	name	localized_na	primary_attr	attack_type	roles	legs
2	1	npc_dota_h	Anti-Mage	agi	Melee	Carry,Escape	2
3	2	npc_dota_h	Axe	str	Melee	Initiator,Dur	2
4	3	npc_dota_h	Bane	int	Ranged	Support,Dis	4

	A	B	C	D	E
1	match_id	is_pick	hero_id	team	ord
2	3877565311	TRUE	1	0	16
3	1057683331	FALSE	1	1	17
4	3564047563	TRUE	1	1	17
5	2717363456	FALSE	1	1	17

Data Cleansing

- Remove unnecessary columns.
- Join **matches** and **player_matches** to get each player's performance in each match and the match outcome. -> **detailed_matches**

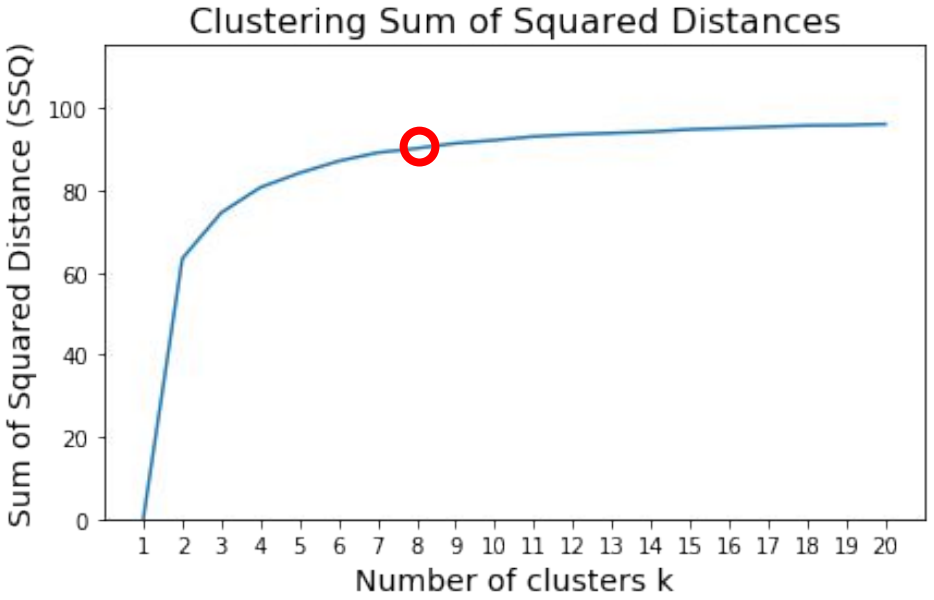
Task 1: Clustering Model of Hero Types

- Select interested columns from **detailed_matches**, define each hero as a vector
  - [hero_id, kills, deaths, assists, gold, last_hits, denies, gold_per_min, xp_per_min, gold_spent, hero_damage, tower_damage, hero_healing, level]
- Aggregate heroes by hero_id, calculate mean of each attribute

	kills	deaths	assists	last_hits	denies	gold_per_min	xp_per_min	gold_spent	hero_damage	tower_damage	level
	mean	mean	mean	mean	mean	mean	mean	mean	mean	mean	mean
hero_id											
1	6.823423	2.962883	5.501261	374.985225	16.980541	652.436757	657.051892	21718.535135	13676.197117	4531.254054	20.953514
2	6.532593	6.262192	8.362144	144.345727	2.922743	393.459440	436.087880	11826.640512	12353.809029	329.294302	17.274988
3	2.949980	6.332933	11.152461	25.117047	3.522209	247.899760	308.888956	7775.392157	6757.598639	228.969988	14.617647
4	8.385020	5.354008	9.347572	236.993563	16.308953	506.483909	529.891750	16953.282621	19034.640140	2047.499122	19.747221
5	2.803379	6.296534	12.744538	59.079231	1.074862	291.036411	333.853772	9306.788523	8957.451209	208.286630	15.554908

Task 1: Clustering Model of Hero Types

- Apply K-Means Clustering
  - Generate and plot the SSQ statistics
  - k = 1 ~ 20

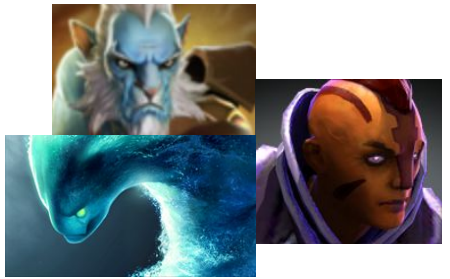


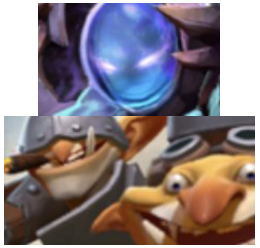
Task 1: Clustering Model of Hero Types

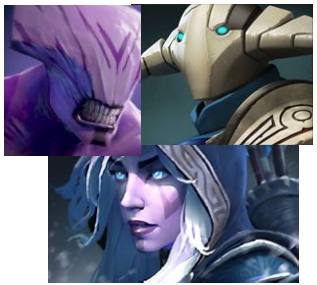
- Apply K-Means Clustering
  - Generate and plot the SSQ statistics
  - k = 1 ~ 20
- Pick n_clusters = 8 and fit the data to get labels for different heroes
  - kmeans = KMeans(n_clusters=8).fit(hero_data)
  - labels = kmeans.labels_

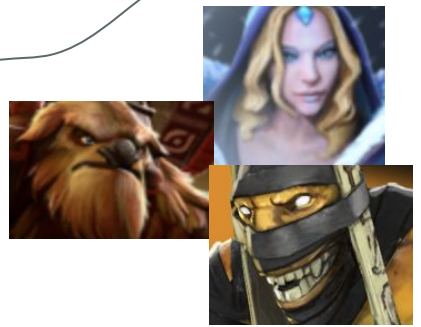
cluster	hero_id
0	[3, 20, 26, 28, 30, 50, 57, 66, 71, 83, 84, 85...
1	[4, 9, 13, 15, 17, 19, 25, 36, 39, 43, 44, 47,...
2	[2, 14, 16, 23, 51, 55, 58, 65, 69, 78, 92, 96...
3	[34, 113]
4	[6, 18, 21, 33, 41, 42, 49, 53, 61, 77, 81, 89...
5	[1, 8, 10, 11, 12, 46, 48, 73, 80, 82, 94, 95,...
6	[5, 7, 27, 29, 31, 32, 37, 38, 60, 62, 64, 68,...
7	[22, 35, 40, 45, 67]

Task 1: Clustering Model of Hero Types









Task 2: Match outcome influencing factors

- Part 1: Match outcome prediction
  - Use game data (kills, deaths, assists, etc) to predict match outcome
  - Group the table by team, aggregate the feature columns by sum of each team member
  - Prediction model: Decision Tree, Logistic Regression and k-Nearest Neighbors
- Result

Model	Accuracy	Precision	Recall	Roc
Decision Tree	0.960	0.955	0.965	0.99
Logistic Regression	0.970	0.969	0.970	0.99
K-Nearest Neighbor	0.931	0.930	0.932	0.96

Task 2: Match outcome influencing factors

Part 2: First blood & outcome relationship

- Use columns('**firstblood claimed**' and '**first blood time**') from jointed table, then calculate the corresponding match result, and aggregate by 'match id' and 'is radiant'.
- Split training and testing dataset, and train the linear regression model
- Compute metrics including Accuracy, Precision, Recall, Confusion matrix and plot ROC curve.

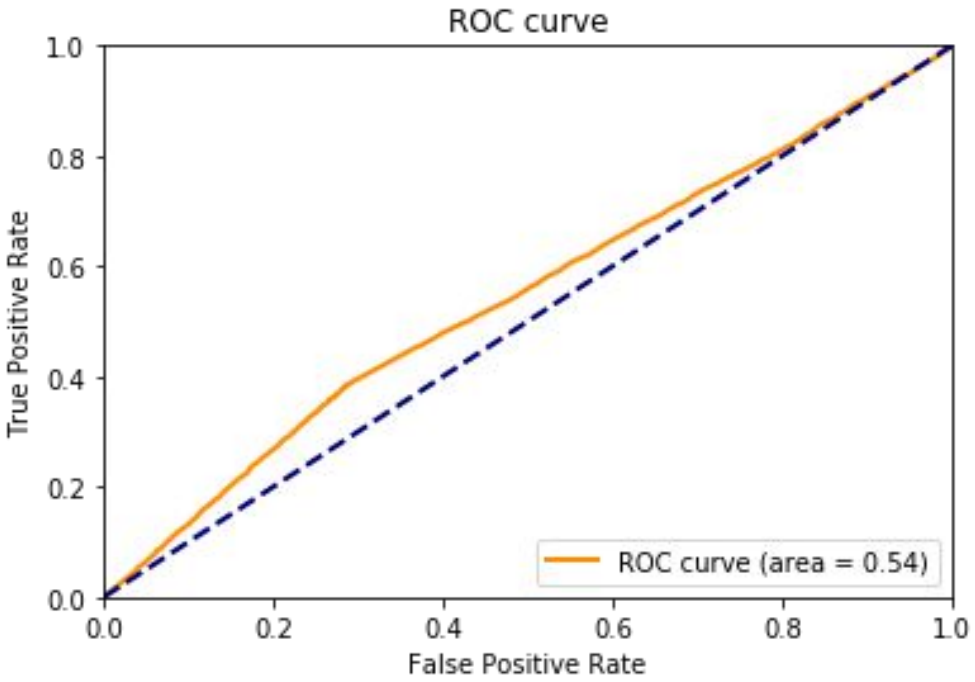
Task 2: Result

Accuracy: 0.5468

Precision: 0.579

Recall: 0.385

Confusion Matrix:
[[5617 2272]
 [4970 3121]]



Task 3: Pick rate & win rate relationship

Methods

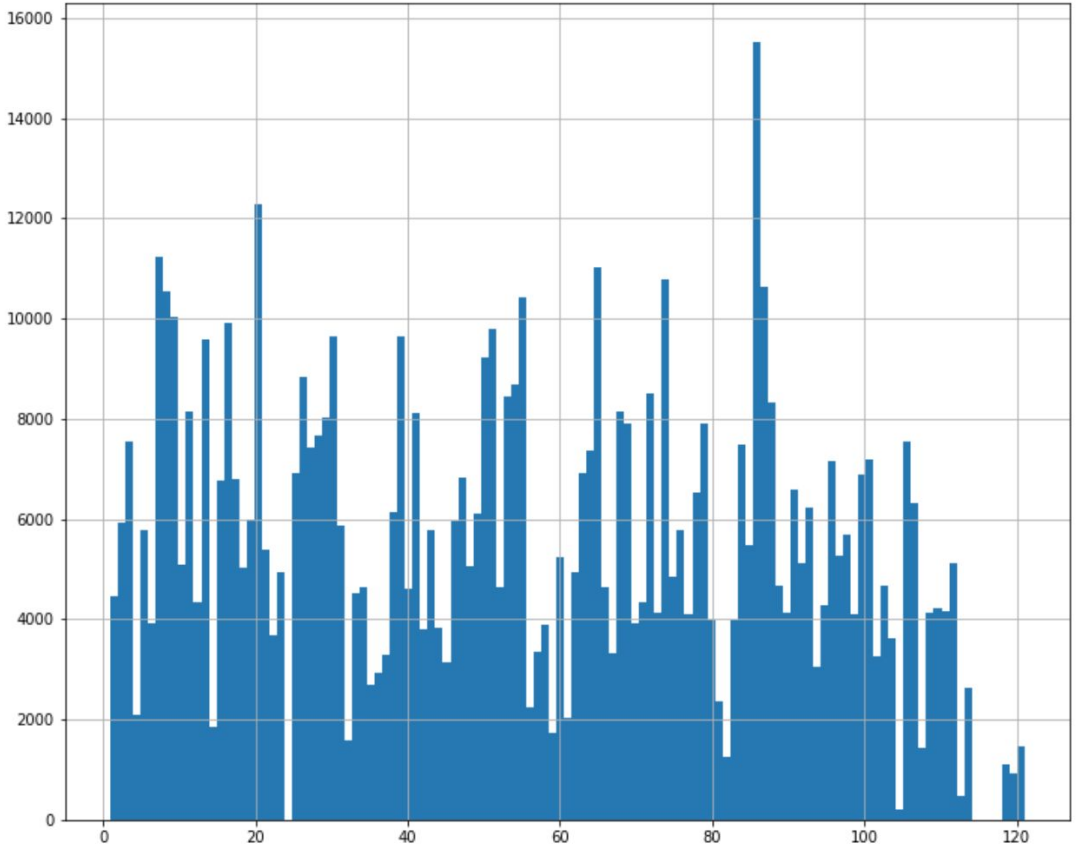
- Hero selection statistics
- Generate table with hero pick rate and win rate for individual heroes
    - Pick rate = (number of matches with hero existing) / (total number of matches)
    - Win rate = (number of winning matches with hero existing) / (total number of matches with hero existing)
- Relationship analysis: Pearson's r

Task 3: Results

Figure showing hero selection calculation among all game matches.

Top 5 of the most popular heroes among all game are Rubick, Vengeful Spirit, Earthshaker, Batrider, Invoker.

Lease popular are Meepo, Dark Willow, Pangolier, Arc Warden, Techies.



Task 3: Results

hero_id	pick_rate <lambda>	win_rate <lambda>
1	0.067370	0.498313
2	0.089609	0.473035
3	0.114378	0.528609
4	0.031708	0.489250
5	0.087730	0.528579

<matplotlib.collections.PathCollection at 0x7f714a655390>

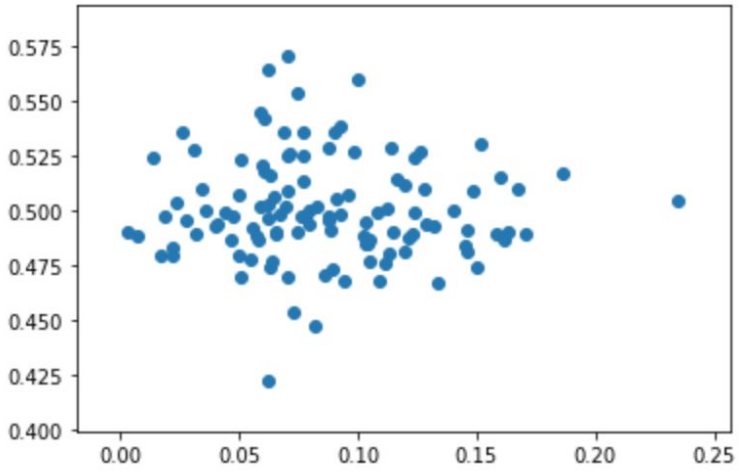
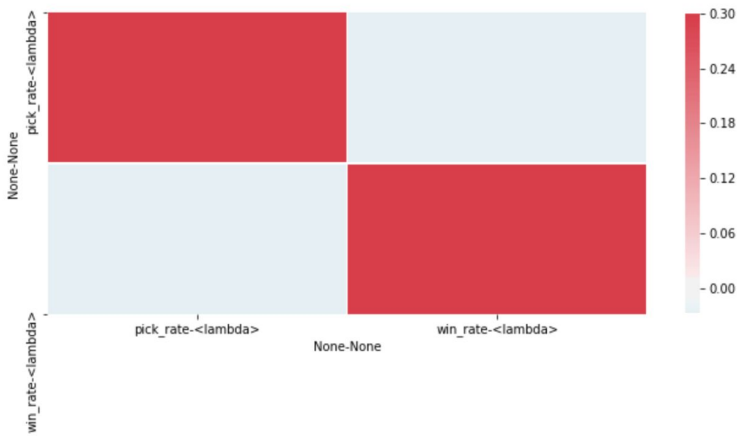


Table example and scatter plot showing pick rate and win rate.

Task 3: Results

		pick_rate <lambda>	win_rate <lambda>
pick_rate	<lambda>	1.000000	-0.026916
win_rate	<lambda>	-0.026916	1.000000



Pearson's r and heat map showing there is no strong relationship between pick rate and win rate.

Future work

- Find team balancing using hero clustering
- Analyze the game factors by dividing the matches into different player levels
- Hero recommendation

Conclusion

- Clustering model is successfully built into 8 types.
- Three prediction models are also well applied on data set while logistic regression performs the best with recall score above 96.9%.
- First blood information has weak correlation with the match outcome.
- No relationship between pick rate and win rate.

Thank you!