

Objective:

We want to examine the accuracy of match outcome prediction solely by hero selection.

For this purpose we build 6 models as follows:

1. Logistic regression based on the sum of hero roles per team.
2. Logistic regression based on the sum of hero roles of both teams (dire roles are considered negative).
3. ANN based on the sum of hero roles of both teams (dire roles are considered negative).
4. Logistic regression based on the selected heros (radiant hero is 1, dire hero is -1, and not picked is 0).
5. Convolutional NN based on the selected heros (radiant hero is 1, dire hero is -1, and not picked is 0).
6. And KNN clustering the selected heros (radiant hero is 1, dire hero is -1, and not picked is 0).

Results:

Model 1:

Coefficients:					
	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	0.067378295	0.149474706	0.45077	0.652157	
melee	0.094488271	0.013951263	6.77274	1.26E-11	***
support	0.037934935	0.007080245	5.35786	8.42E-08	***
initiator	-0.02624818	0.007088644	-3.70285	0.000213	***
disabler	0.001430504	0.007711289	0.18551	0.852831	
nuker	-0.037869959	0.006093104	-6.21522	5.13E-10	***
carry	-0.020143747	0.007579819	-2.65755	0.007871	**
escape	-0.044017004	0.005686282	-7.74091	9.87E-15	***
jungler	-0.072631401	0.010966884	-6.62279	3.52E-11	***
durable	0.021044013	0.006687291	3.14687	0.00165	**
pusher	-0.018792197	0.007271618	-2.58432	0.009757	**
Dire_melee	0.084764218	0.013889425	6.10279	1.04E-09	***
Dire_support	0.040149243	0.007094968	5.65883	1.52E-08	***
Dire_initiator	-0.023755136	0.007106979	-3.34251	0.00083	***
Dire_disabler	-0.004685346	0.007712094	-0.60753	0.543498	
Dire_nuker	-0.042000662	0.006091745	-6.89468	5.4E-12	***
Dire_carry	-0.009938153	0.007621261	-1.304	0.192232	
Dire_escape	-0.046173258	0.005686083	-8.1204	4.65E-16	***
Dire_jungler	-0.051553957	0.010906003	-4.72712	2.28E-06	***
Dire_durable	0.021830063	0.006697947	3.25922	0.001117	**
Dire_pusher	-0.015599856	0.007263232	-2.14778	0.031731	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					

The LR model was trained on 80% of the data with the accuracy of 55.72%. The model predicted the outcome for test matches with accuracy of 56.32%.

Model 2:

Coefficients:					
	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	0.065666104	0.010109469	6.4955	8.27554E-11	***
melee	0.089521251	0.009662092	9.2652	< 0.000000000000000222	***
support	0.039070631	0.004897428	7.97779	1.48981E-15	***
initiator	-0.024934082	0.004843903	-5.14752	2.63955E-07	***
disabler	-0.001744975	0.005247548	-0.33253	0.73948797	
nuker	-0.039944899	0.004136528	-9.65663	< 0.000000000000000222	***
carry	-0.01506198	0.005318128	-2.8322	0.00462295	**
escape	-0.045093191	0.003877763	-11.62866	< 0.000000000000000222	***
jungler	-0.062063024	0.007611889	-8.15343	3.5374E-16	***
durable	0.02147374	0.00460554	4.66259	3.12257E-06	***
pusher	-0.017194827	0.005032455	-3.41679	0.00063365	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					

The LR model was trained on 80% of the data with the accuracy of 55.71%. The model predicted the outcome for test matches with accuracy of 56.30%.

Model 3:

ANN with 1 layer and 5 nodes. The model was trained on 80% of the data until 99.99% fitness was gained. The model predicted the outcome for test matches with accuracy of 56.30%.

Model 4:

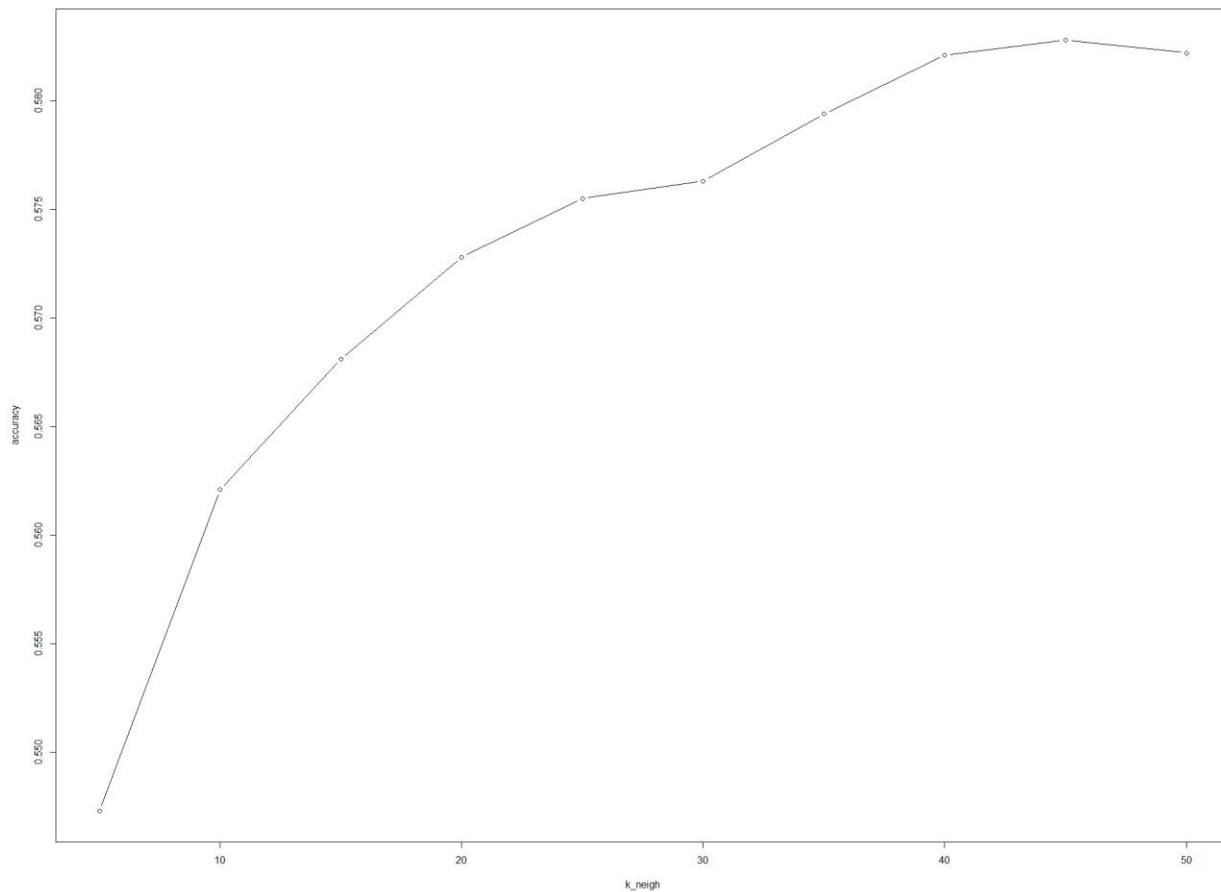
The LR model was trained on 80% of the data with the accuracy of 60.52%. The model predicted the outcome for test matches with accuracy of 61.04%.

Model 5:

Two convolutional NN with 3 layers of (50,20,10) and (100,50,100) nodes were trained. The models was trained on 60% of the data and validated on 20%. The model predicted the outcome for validation matches (20% of the data) with accuracy of 59.11% and 59.37%, respectively. The model predicted the outcome for test matches (20% of the data) with accuracy of 60.33% and 61.16%, respectively.

Model 6:

The KNN model was trained on 80% of the data and tested on the remaining 20% of the data. The results are shown in the following figure. The highest accuracy gained is 58.5%.

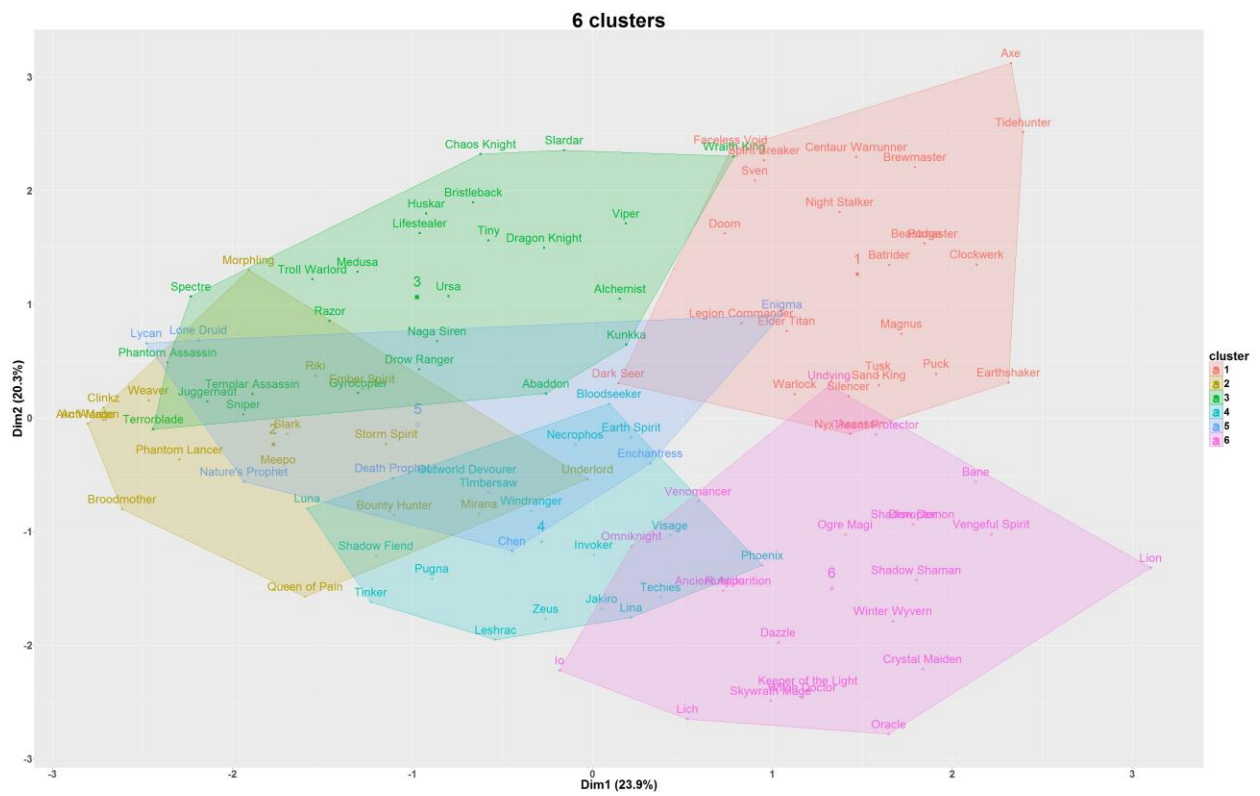
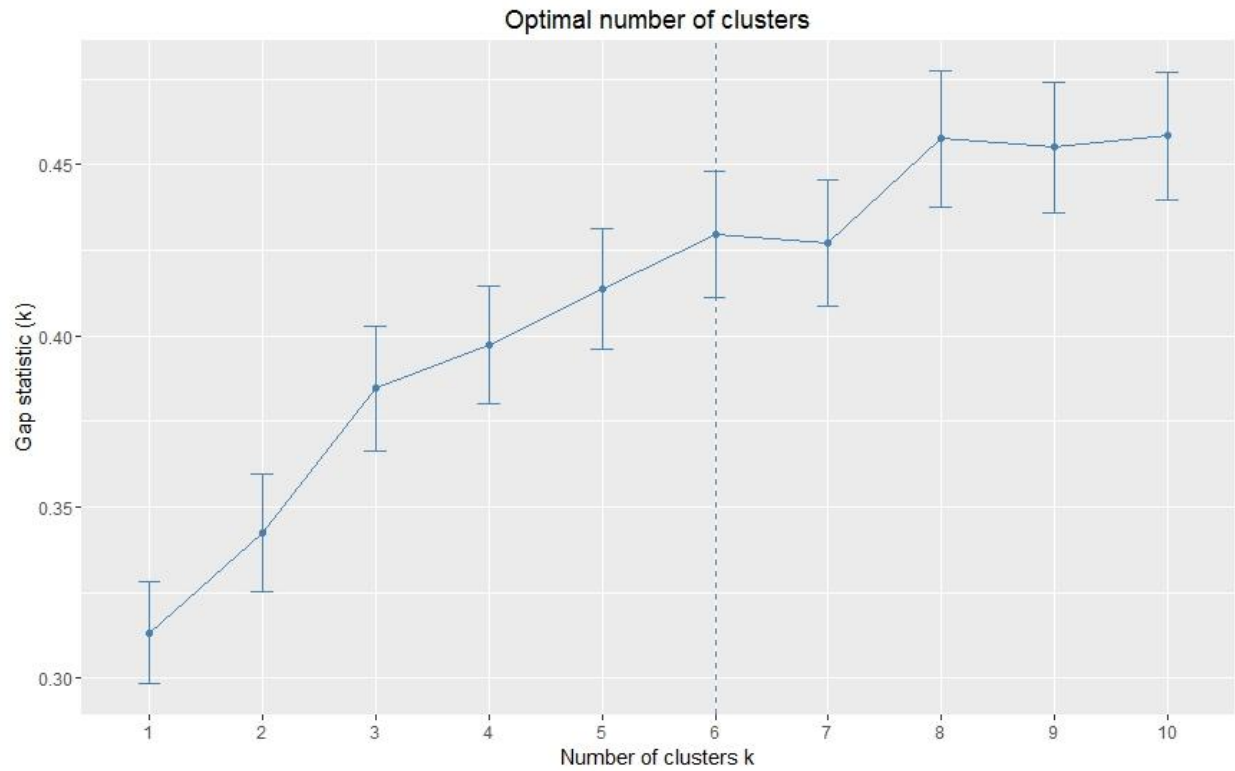


Discussion:

Using 6 different models, I have achieved almost the same accuracy (about 60%) in predicting the match outcome solely based on the selected hero. This result indicate achieving higher accuracy solely based on hero selection is due to over-fitting and not accurate.

PS: can we achieve 5 main roles by role clustering provided by steam?

The results of the optimal number of clusters (k means clustering) is shown in the following. Also, I have displayed the 6 and 5 number of clusters results. Based on my experience the 5 cluster results almost achieve the professional players' hero selection method (1 from each cluster).



5 clusters

