Role Impact on Team Success

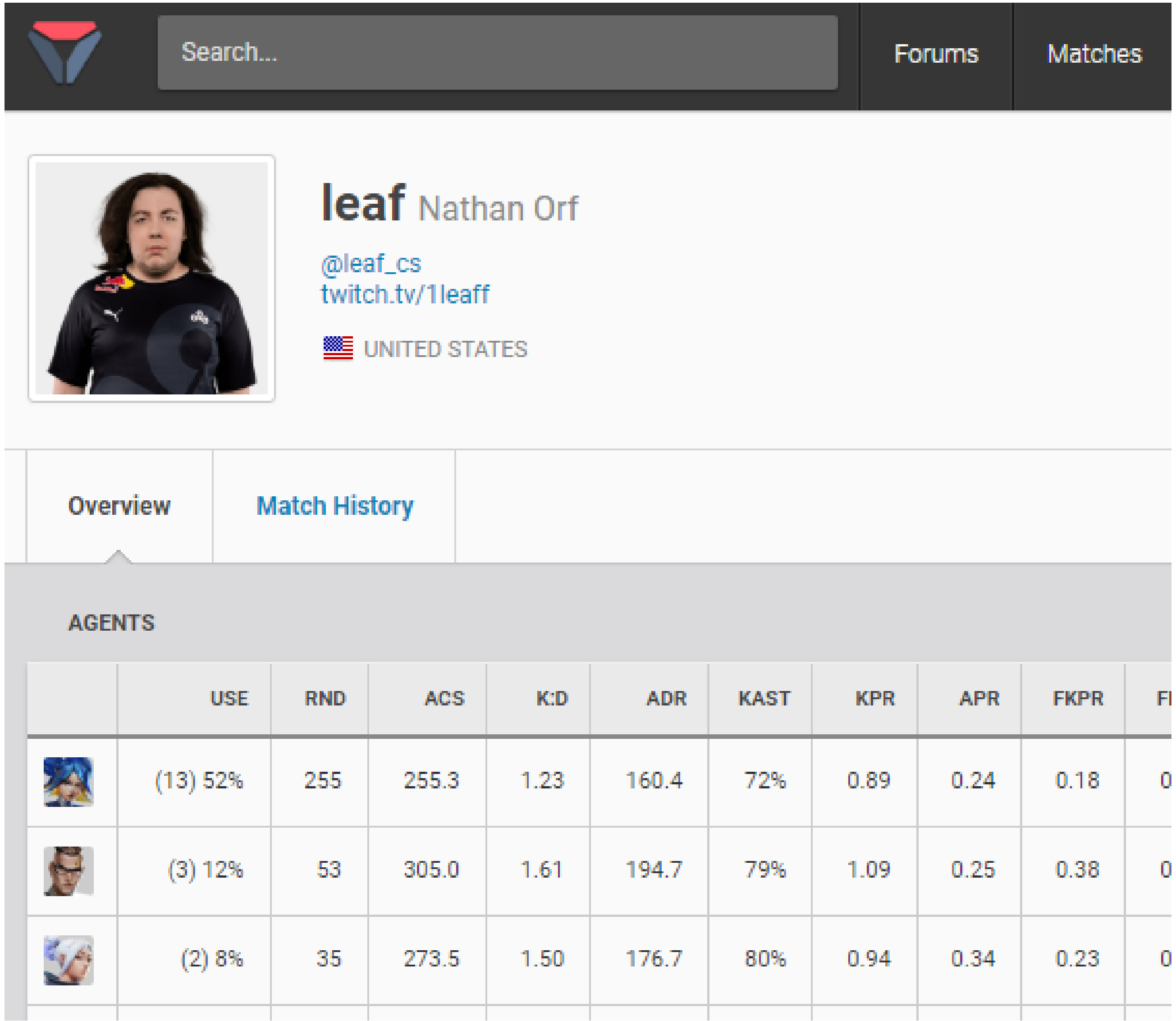
Angus Leung

Design

- Client: Valorant has an emerging e-sports scene with large and small organizations constantly scouting for the next superstar player. Potential clients could be e-sport organizations looking to replace current members of their team
- Objective: Explore whether a player’s role in a team and the stats that they put up in game can be modeled to predict a team’s success
- Goal: Produce a regression model that helps identify key stat points to look for when looking to fill a role in a team.

Data

- Player stats is scrapped from [vlr.gg](vlr.gg), a website that tracks pro-player stats.
- Each row represents the stats of a player on one agent they played in their last 60 days of games.
- A total of 2019 player stats were scrapped. Most having multiple agents played. Players that were on teams that have not been ranked were dropped (22 teams without rankings.
- Tools Used: BeautifulSoup, Numpy, Pandas, Scikit-learn, Statsmodels, Matplotlib, Seaborn



What is ELO?

- Rating system originally developed for chess to calculate relative skill levels of players.
- Higher ELO means the player/team should be better than a lower ELO player/team.
- “A player whose rating is 100 points greater than their opponent's is expected to score 64%; if the difference is 200 points, then the expected score for the stronger player is 76%.” – Elo rating system
- “If the higher-rated player wins, then only a few rating points will be taken from the lower-rated player. However, if the lower-rated player scores an upset win, many rating points will be transferred.” – Elo rating system

Feature Engineering

- Dummy variables to represent the role_on_team
  - Adds ‘is_duelist’,’is_controller’,’is_initiator’,’is_sentinel’ columns
- Adding negative weight/connotation for deaths and first deaths by making values negative.
  - This leads to inability to scale data in order to keep this negative weighing.
- Adding Interaction Terms
  - Interaction of kills and death and interaction of first kill and first death

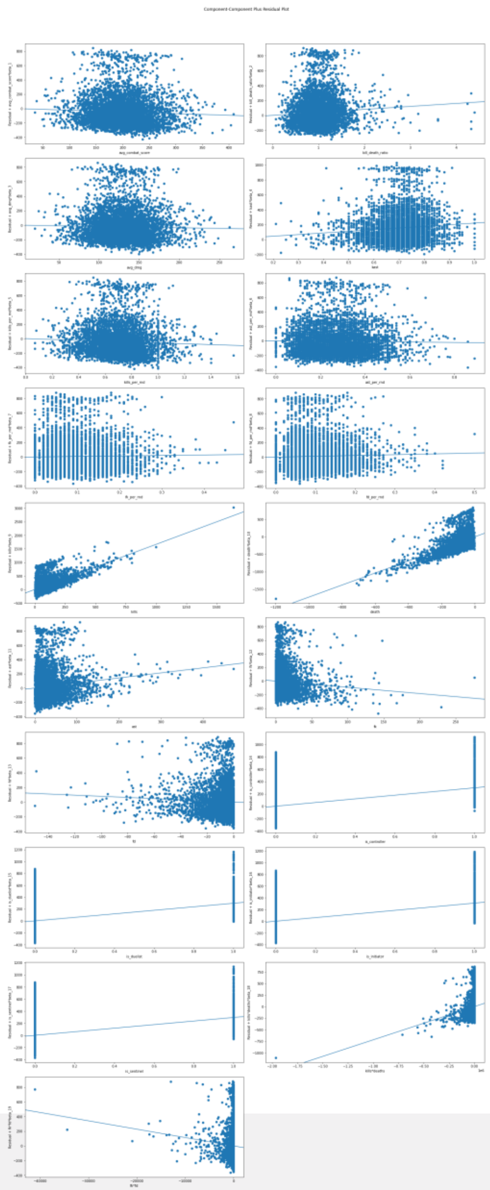
Spoilers Results

Model Name	R2 Score	MAE	Test R2	Test MAE
unfiltered initial model	0.033	154.35	-0.0002	146.40
filtered by rank	0.033	121.14	0.016	118.21
duelist only	0.043	121.07	-0.006	122.78
controller only	0.027	121.17	0.043	119.21
initiator only	0.027	121.40	0.024	115.76
sentinel only	0.038	120.66	-0.057	120.52
log y	0.034	0.076	0.017	0.075
log duelist	0.043	0.076	-0.005	0.078
log controller	0.028	0.076	0.045	0.075
log initiator	0.028	0.076	0.025	0.073
log sentinel	0.038	0.076	-0.054	0.076
poly y	0.048	119.57	0.092	111.89
poly duelist	0.047	119.52	0.140	111.72
poly controller	0.040	119.18	0.120	110.44
poly initiator	0.041	120.20	0.104	108.90
poly sentinel	0.074	117.59	0.109	108.57

First Models

A few too many variables to read easily. The most robust effects seem to come from kills, deaths, assists, first kill and first death.

We also see greater residuals at the highest levels of play. Perhaps at higher levels there are more intangible factors that lead to team success.



OLS Regression Results

Dep. Variable:	team_rank	R-squared:	0.033
Model:	OLS	Adj. R-squared:	0.029
Method:	Least Squares	F-statistic:	9.547
Date:	Fri, 05 Aug 2022	Prob (F-statistic):	1.27e-26
Time:	23:04:29	Log-Likelihood:	-34247.
No. Observations:	5065	AIC:	6.853e+04
Df Residuals:	5046	BIC:	6.866e+04
Df Model:	18		
Covariance Type:	nonrobust		

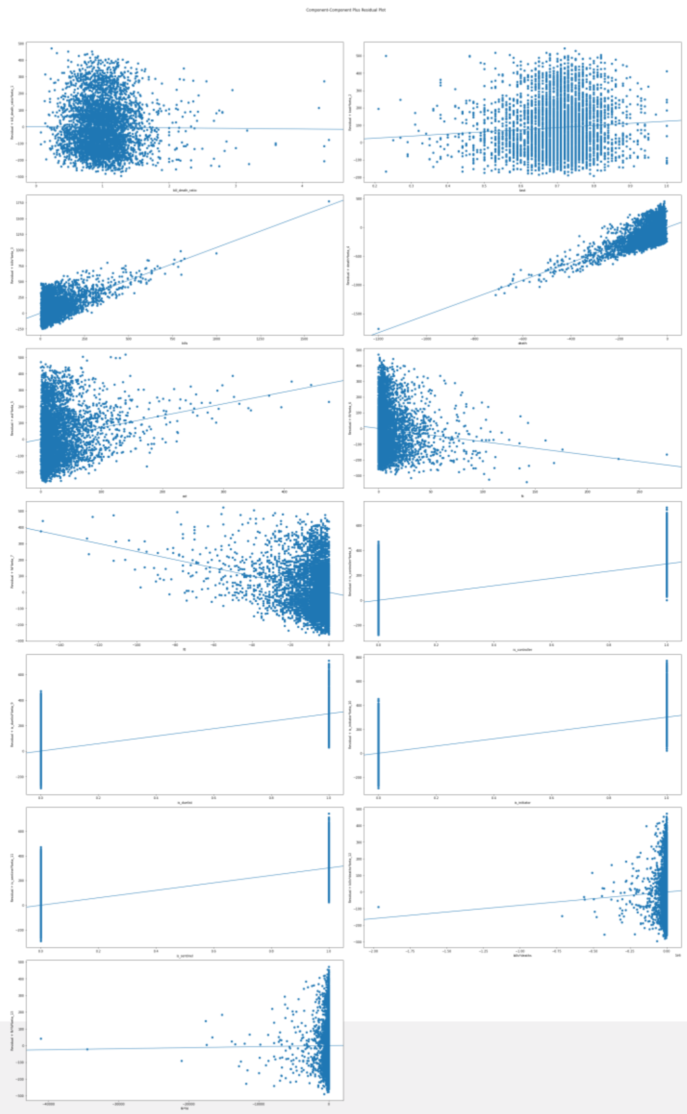
	coef	std err	t	P>|t|	[0.025	0.975]
const	1199.6496	26.202	45.785	0.000	1148.283	1251.016
avg_combat_score	-0.2333	0.374	-0.624	0.532	-0.966	0.499
kill_death_ratio	39.9248	19.473	2.050	0.040	1.749	78.101
avg_dmg	-0.1794	0.404	-0.444	0.657	-0.972	0.613
kast	220.2823	51.285	4.295	0.000	119.742	320.823
kills_per_rnd	-59.2719	82.265	-0.721	0.471	-220.546	102.003
ast_per_rnd	-26.7032	38.788	-0.688	0.491	-102.744	49.337
fk_per_rnd	66.5966	86.882	0.767	0.443	-103.731	236.924
fd_per_rnd	109.5679	77.484	1.414	0.157	-42.334	261.470
kills	1.6635	0.275	6.047	0.000	1.124	2.203
death	1.7271	0.288	5.990	0.000	1.162	2.292
ast	0.7076	0.264	2.681	0.007	0.190	1.225
fk	-0.9173	1.096	-0.837	0.403	-3.066	1.232
fd	-0.7845	1.013	-0.774	0.439	-2.771	1.202
is_controller	297.7196	8.359	35.615	0.000	281.332	314.108
is_duelist	297.4054	11.088	26.823	0.000	275.668	319.142
is_initiator	309.5937	8.211	37.703	0.000	293.496	325.691
is_sentinel	294.9310	8.441	34.941	0.000	278.383	311.479
kills*deaths	0.0007	0.000	3.109	0.002	0.000	0.001
fk*fd	-0.0114	0.009	-1.229	0.219	-0.030	0.007

Omnibus:	1553.942	Durbin-Watson:	1.981
Prob(Omnibus):	0.000	Jarque-Bera (JB):	4444.533
Skew:	1.616	Prob(JB):	0.00
Kurtosis:	6.258	Cond. No.	2.54e+19

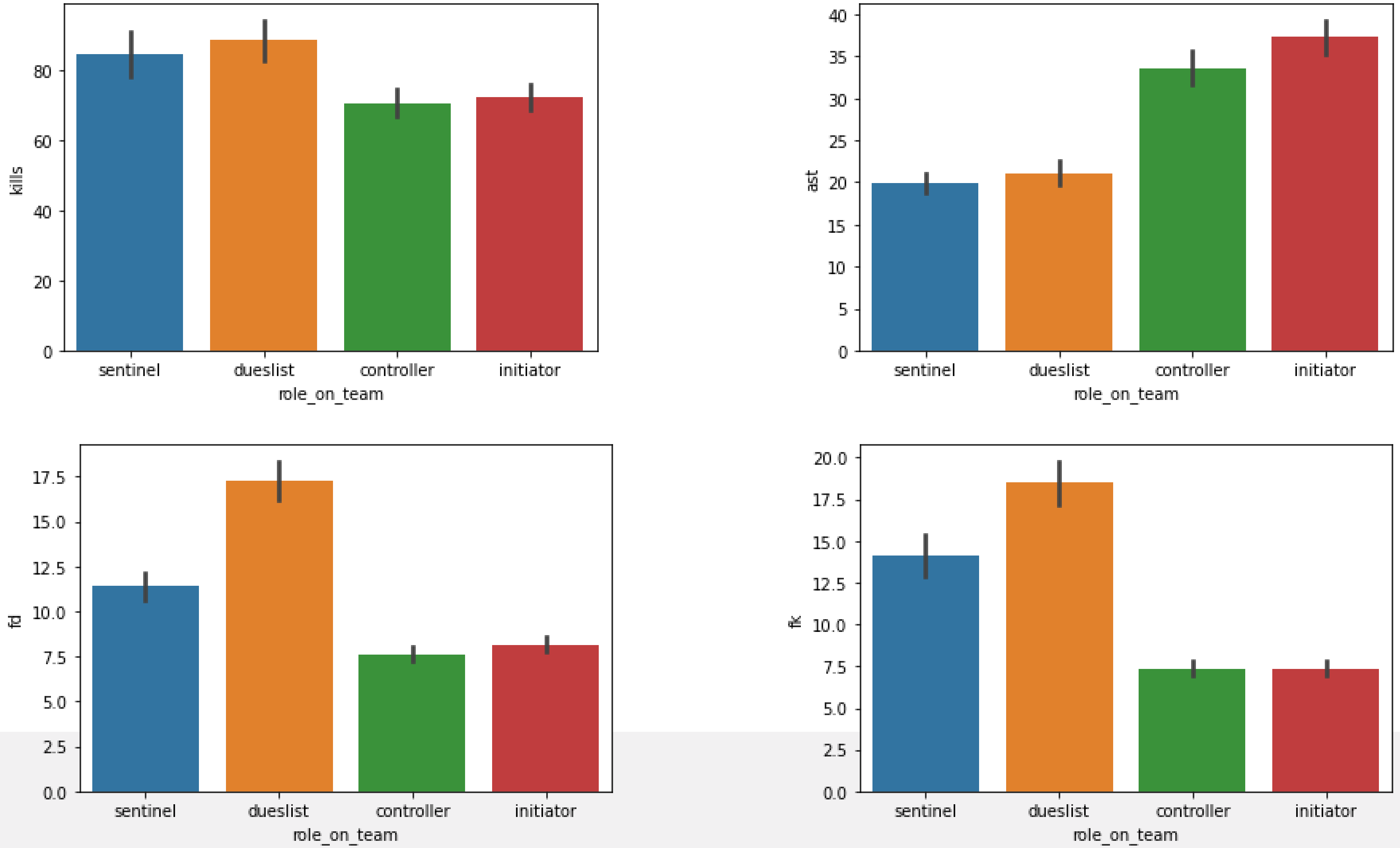
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.01e-26. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.

Filter down to teams with ELO<2000

- Roughly the same performance
- Reduced variance after filtering.



How to Improve?



Separate Models for Roles

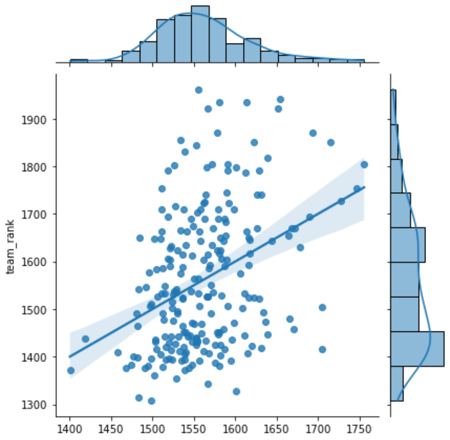
Model Name	R2 Score	MAE	Test R2	Test MAE
duelist only	0.043	121.07	-0.006	122.78
controller only	0.027	121.17	0.043	119.21
initiator only	0.027	121.40	0.024	115.76
sentinel only	0.038	120.66	-0.057	120.52

- The separation helped to increase accuracy for duelist and sentinels
- We see MAE with differences of around 121 points.

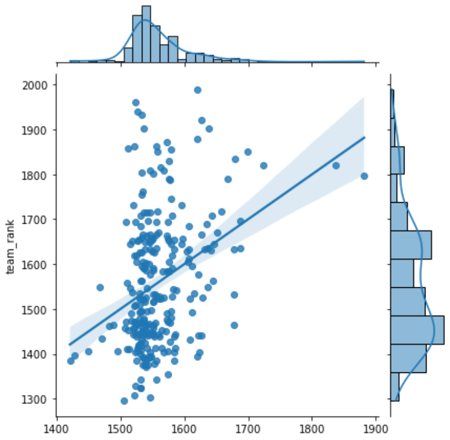
Conclusion

- Best performing model was the polynomial model
- Stats may be an indication of how good a single player is but there may be several intangibles at play that determine overall success

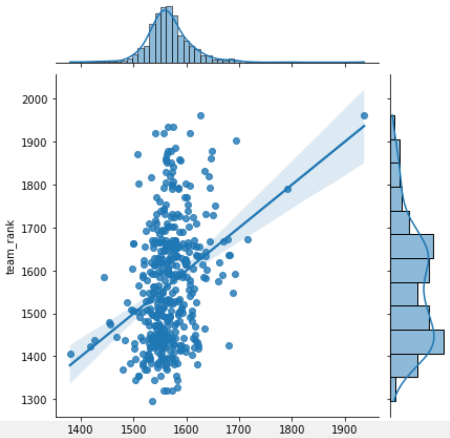
Duelist Model



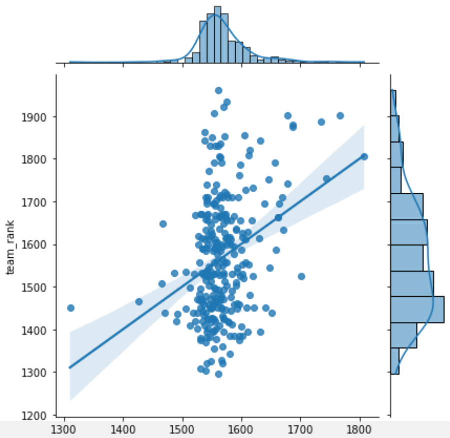
Controller Model



Initiator Model



Sentinel Model



Future Work

- Looking at an overall team’s composition (multiples of a role in a team)
- Refining dependent variable
- Looking at per match stats instead of overall history of stats

- Looking at more complex modeling methods