Applying Machine Learning Techniques to eSports

by John Chmielowiec and Ritwik Katiyar

Introduction to eSports

- Money
  - sponsorships of teams/players/tournaments
  - advertising on content streams
  - prize pools for players
- Entertainment
  - Spectating
  - Playing

Introduction to eSports

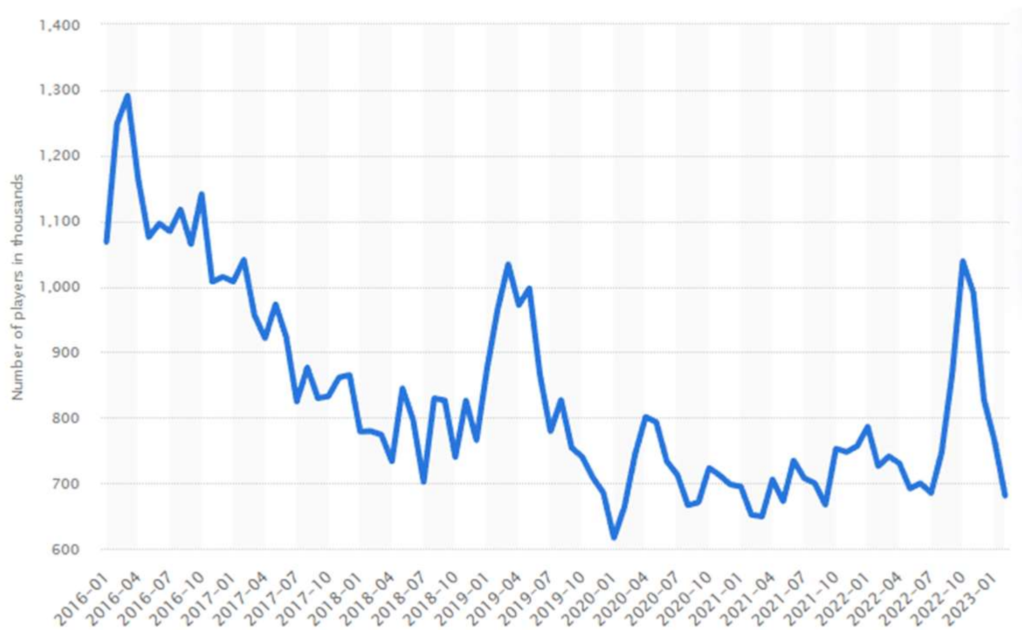
#Rank	Tournament	Year	Game	Prize Pool Money in USD
1	The International	2019	Dota 2	$34,330,069.00
2	The International	2018	Dota 2	$25,532,177.00
3	The International	2017	Dota 2	$24,687,919.00
4	The International	2016	Dota 2	$20,770,460.00
5	The International	2015	Dota 2	$18,429,613.05
6	Fortnite World Cup Finals - Solo	2019	Fortnite	$15,287,500.00
7	Fortnite World Cup Finals - Duo	2019	Fortnite	$15,100,000.00
8	The International	2014	Dota 2	$10,931,103.00
9	PUBG Global Invitational.S	2021	PUBG	$7,068,071.00
10	LoL World Championship	2018	LoL	$6,450,000.00
11	LoL World Championship	2016	LoL	$5,070,000.00
12	LoL World Championship	2017	LoL	$4,946,969.00

Largest prize pools as of August 16, 2021

Introduction to eSports

- Number of viewers
- Time watched
- Hours of content

Introduction to eSports



KEY INSIGHTS

Number of registered Dota 2 accounts	**81.2m**
Monthly peak concurrent players of Dota 2 on Steam	**680.75k**
Average number of Dota 2 players on Steam per hour in 2022	**466.5k**

Monthly number of peak concurrent players of Dota 2 on Steam worldwide as of February 2023

Extremely high numbers of active players

Introduction to League of Legends

- Two teams
  - 5 players on each team
  - Each has a base. The objective is to destroy the enemy base.
  - Base endless creates an army to help attack and defend.
- Champions
  - Each player chooses from over 150 champions with unique abilities.
  - During the selection process, players also ban specific champion from play.
- General Gameplay
  - Assault enemy base, killing enemies to earn experience and gold.
  - Experience will increase the power of a champion's abilities.
  - Gold allows items to be bought to further customize a champion.

Introduction to League of Legends

- Map
  - Lanes
  - Vision (fog of war)
  - Wards

Introduction to League of Legends

- Objective
  - Baron: Highly contested, spawns after 20 minutes, can trigger endgame
  - Dragon: similar to baron, but available more frequently, with smaller rewards
- Roles
  - Top, Bottom, Middle
  - Support
  - Jungle

Data Cleaning and Exploration

- eSports match data for 2023
  - 123 features
  - 72,552 observations
- Cleaning Process
  - Separate Players from Teams
  - Separate Complete Player information from Partial Player Information
    - Some Leagues only report a subset of the total features
  - Removed features not applicable to the analysis
  - Removed features that had too much missing data to be of use
  - Check for invalid data on the categorical features
  - Examine remaining data programmatically and via graph visualizations.

Data Cleaning and Exploration

	datacompleteness	league	side	position	playername	teamname	champion	result	kills	dragons
0	complete	LFL2	Blue	top	Wylenz	Klanik Esport	Jax	1	4	NaN
1	complete	LFL2	Blue	jng	Julbu	Klanik Esport	Poppy	1	2	NaN
2	complete	LFL2	Blue	mid	Sintax	Klanik Esport	Taliyah	1	2	NaN
3	complete	LFL2	Blue	bot	Axelent	Klanik Esport	Ezreal	1	5	NaN
4	complete	LFL2	Blue	sup	Wixo	Klanik Esport	Karma	1	0	NaN
5	complete	LFL2	Red	top	Anathar	MS Company	Sejuani	0	0	NaN
6	complete	LFL2	Red	jng	nicolaiy	MS Company	Viego	0	0	NaN
7	complete	LFL2	Red	mid	Kuroneel	MS Company	Syndra	0	3	NaN
8	complete	LFL2	Red	bot	Scripter	MS Company	Zeri	0	3	NaN
9	complete	LFL2	Red	sup	Zimba	MS Company	Yuumi	0	1	NaN
10	complete	LFL2	Blue	team	NaN	Klanik Esport	NaN	1	13	4.0
11	complete	LFL2	Red	team	NaN	MS Company	NaN	0	7	3.0

Initial Dataset Example

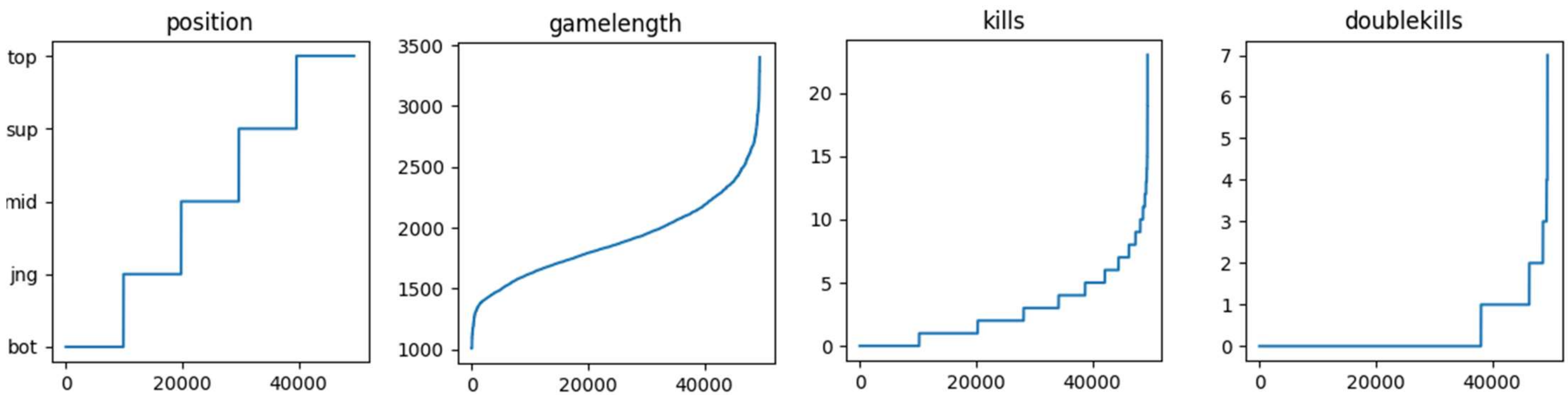
Data Cleaning and Exploration

```
df=df_players, len=0, cols with nunique of 0=[]
df=df_players, len=1, cols with nunique of 1=['year']
df=df_players
Total Missing Values: 2488564
  col=2=>'url', nans=59760
  col=5=>'split', nans=13128
  col=9=>'patch', nans=120
  col=13=>'playername', nans=12105
  col=14=>'playerid', nans=12375
  col=16=>'teamid', nans=84
  col=17=>'champion', nans=12092
  col=18=>'ban1', nans=7026
  col=19=>'ban2', nans=7002
  col=20=>'ban3', nans=7182
  col=21=>'ban4', nans=7146
  col=22=>'ban5', nans=7308
  col=30=>'doublekills', nans=12792
  col=31=>'triplekills', nans=12792
  col=32=>'quadrakills', nans=12792
  col=33=>'pentakills', nans=12792
  col=34=>'firstblood', nans=10660
  col=35=>'firstbloodkill', nans=12092
  col=36=>'firstbloodassist', nans=22752
  col=37=>'firstbloodvictim', nans=22752
```

```
df=df_players
  col=0=>'gameid', nunique=6046
  col=1=>'datacompleteness', nunique=2
  col=2=>'url', nunique=449
  col=3=>'league', nunique=41
  col=4=>'year', nunique=1
  col=5=>'split', nunique=10
  col=6=>'playoffs', nunique=2
  col=7=>'date', nunique=6035
  col=8=>'game', nunique=5
  col=9=>'patch', nunique=10
  col=10=>'participantid', nunique=12
  col=11=>'side', nunique=2
  col=12=>'position', nunique=6
  col=13=>'playername', nunique=2568
  col=14=>'playerid', nunique=2583
  col=15=>'teamname', nunique=418
  col=16=>'teamid', nunique=417
  col=17=>'champion', nunique=160
  col=18=>'ban1', nunique=127
  col=19=>'ban2', nunique=124
  col=20=>'ban3', nunique=131
  col=21=>'ban4', nunique=151
  col=22=>'ban5', nunique=148
```

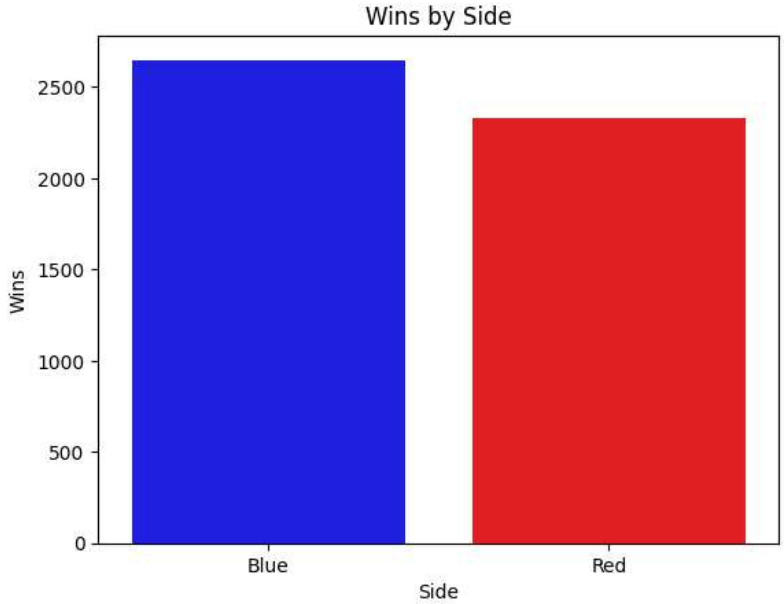
Debugging Function Output

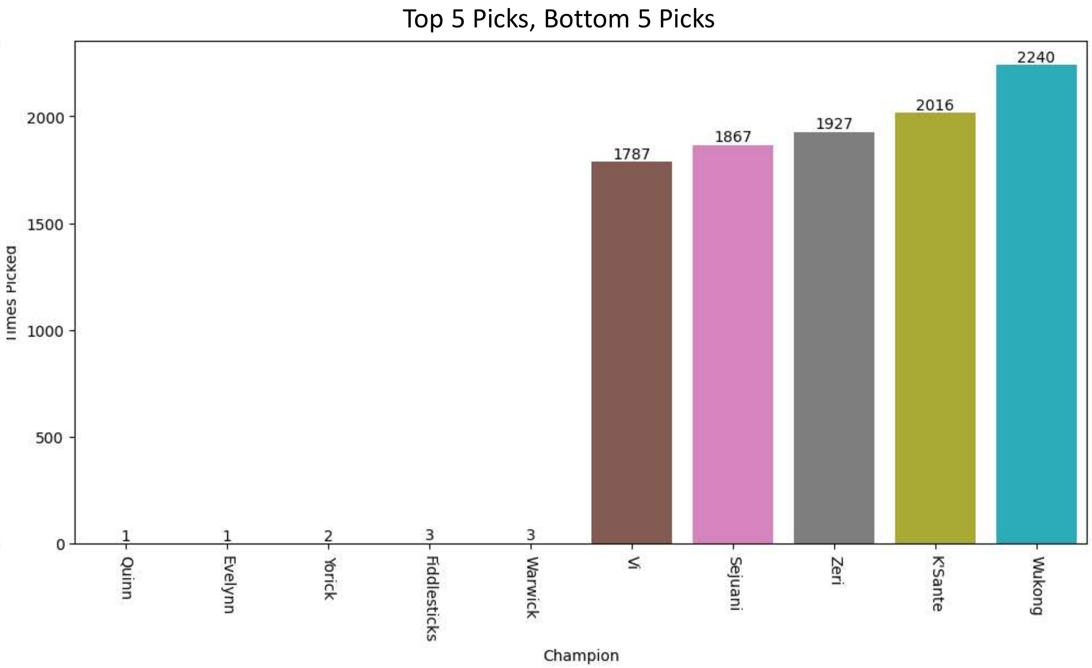
Data Cleaning and Exploration



Data Visualization (sorted features)

Data Cleaning and Exploration





Do the distributions of data points make sense?

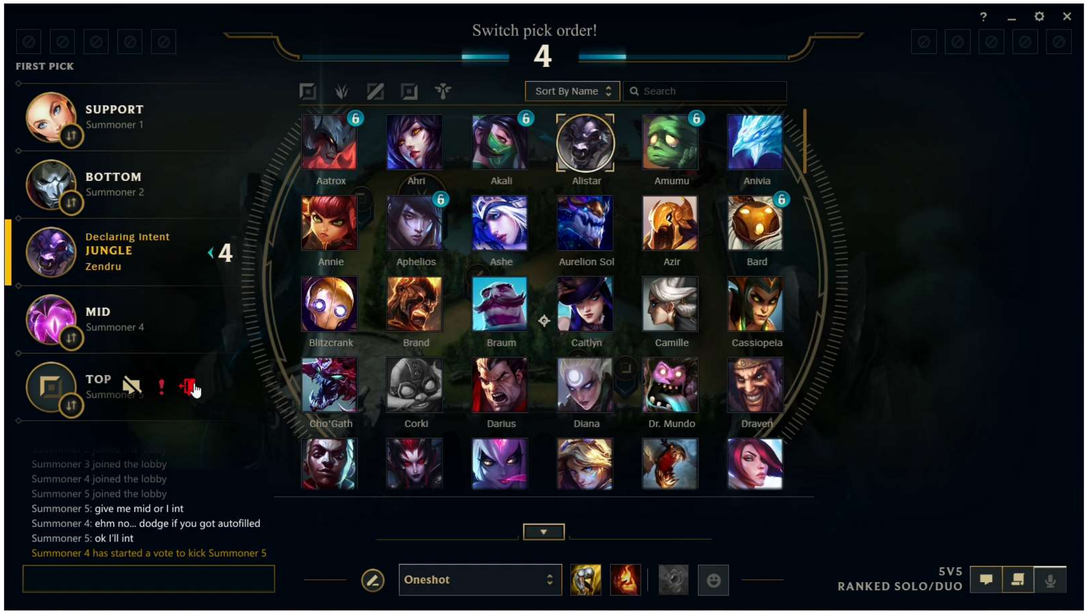
Does the data make sense given preexisting domain knowledge?

3 Questions

1. Frequent Champion Itemsets
   - What champions are frequently picked together?
2. Player Performance Analysis
   - What features are important to the commonly used KDA metric?
3. Game Predictions
   - Can the outcome of a new game be predicted using existing data?

Frequent Champion Itemsets

- Champion Select Phase
  - Similar to Draft Selection
  - Also deny selections to opposing team
  - Done before every game



Frequent Champion Itemsets

	antecedents	consequents	support	confidence	lift	conviction
1	(Nami)	(Lucian)	0.087952	0.942949	9.732410	15.830028
0	(Lucian)	(Nami)	0.087952	0.907772	9.732410	9.831365
3	(Lulu)	(Zeri)	0.065562	0.600736	3.887804	2.117601
5	(Rakan)	(Xayah)	0.052510	0.443220	3.678729	1.579652
4	(Xayah)	(Rakan)	0.052510	0.435833	3.678729	1.562528
2	(Zeri)	(Lulu)	0.065562	0.424301	3.887804	1.547448

Apriori results, Support Threshold = 0.05

Frequent Champion Itemsets

	antecedents	consequents	support	confidence	lift	conviction
7	(Nami)	(Lucian)	0.087952	0.942949	9.732410	15.830028
6	(Lucian)	(Nami)	0.087952	0.907772	9.732410	9.831365
9	(Lulu)	(Zeri)	0.065562	0.600736	3.887804	2.117601
1	(Thresh)	(Aphelios)	0.025703	0.504931	4.635127	1.799879
15	(Rakan)	(Xayah)	0.052510	0.443220	3.678729	1.579652
14	(Xayah)	(Rakan)	0.052510	0.435833	3.678729	1.562528
8	(Zeri)	(Lulu)	0.065562	0.424301	3.887804	1.547448
3	(Jayce)	(Maokai)	0.025904	0.271865	2.279273	1.209560
0	(Aphelios)	(Thresh)	0.025703	0.235945	4.635127	1.242183
13	(Rakan)	(Wukong)	0.026908	0.227119	1.257422	1.060159

Apriori results, Support Threshold = 0.025

Frequent Champion Itemsets

	antecedents	consequents	support	confidence	lift	conviction
258	(Gnar, Nami)	(Lucian)	0.012048	0.960000	9.908394	22.577811
252	(Nami, Azir)	(Lucian)	0.010843	0.955752	9.864551	20.410341
251	(Lucian, Azir)	(Nami)	0.010843	0.947368	10.156932	17.227811
281	(Nami, Sejuani)	(Lucian)	0.013855	0.945205	9.755696	16.481802
274	(Lucian, Maokai)	(Nami)	0.011747	0.943548	10.115976	16.062020
163	(Nami)	(Lucian)	0.087952	0.942949	9.732410	15.830028
286	(Nami, Vi)	(Lucian)	0.016265	0.941860	9.721171	15.533534
276	(Maokai, Nami)	(Lucian)	0.011747	0.936000	9.660684	14.111132
291	(Nami, Wukong)	(Lucian)	0.015763	0.934524	9.645448	13.792990
280	(Lucian, Sejuani)	(Nami)	0.013855	0.932432	9.996800	13.419558

Apriori results, Support Threshold = 0.01

Frequent Champion Itemsets

	antecedents	consequents	support	confidence	lift	conviction
302	(Lulu, Vi)	(Zeri)	0.012450	0.673913	4.361387	2.592811
296	(Lulu, Sejuani)	(Zeri)	0.012651	0.656250	4.247076	2.459584
308	(Wukong, Lulu)	(Zeri)	0.012952	0.629268	4.072458	2.280576
181	(Lulu)	(Zeri)	0.065562	0.600736	3.887804	2.117601
270	(K'Sante, Lulu)	(Zeri)	0.011747	0.590909	3.824207	2.066734
39	(Thresh)	(Aphelios)	0.025703	0.504931	4.635127	1.799879
249	(Yuumi)	(Zeri)	0.018976	0.517808	3.351118	1.753414
295	(Zeri, Sejuani)	(Lulu)	0.012651	0.488372	4.474872	1.741233
312	(Wukong, Xayah)	(Rakan)	0.012149	0.487903	4.118234	1.721405
62	(Caitlyn)	(Lux)	0.022088	0.424710	15.725338	1.691308
			...			
108	(Heimerdinger)	(Varus)	0.016566	0.421995	3.097324	1.494373

Apriori results, Support Threshold = 0.01, sorted by Conviction

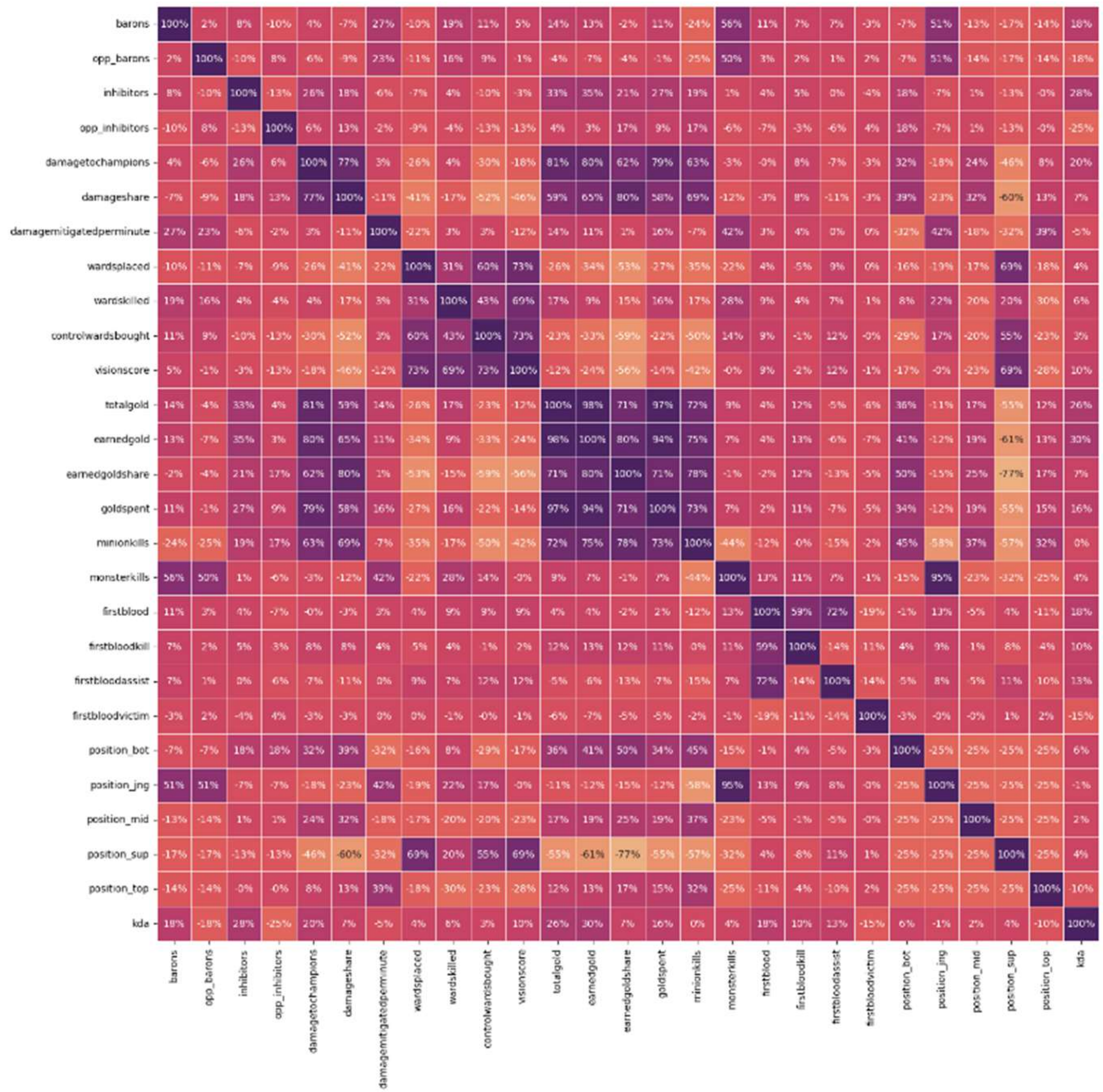
Player Performance Analysis

- KDA
  - Kills, Deaths, Assists
  - Kill Death Ratio: (K+A)/D
- Valuable and easy to understand metric
  - Kills is straightforward measure of offensive capability
  - Assists demonstrate teamwork
  - Deaths show how much a player is overcommitting themselves
  - All three measure player skill, situational awareness, and in-game power
- What are the specific factors that contribute to a player’s KDA?

Player Performance Analysis

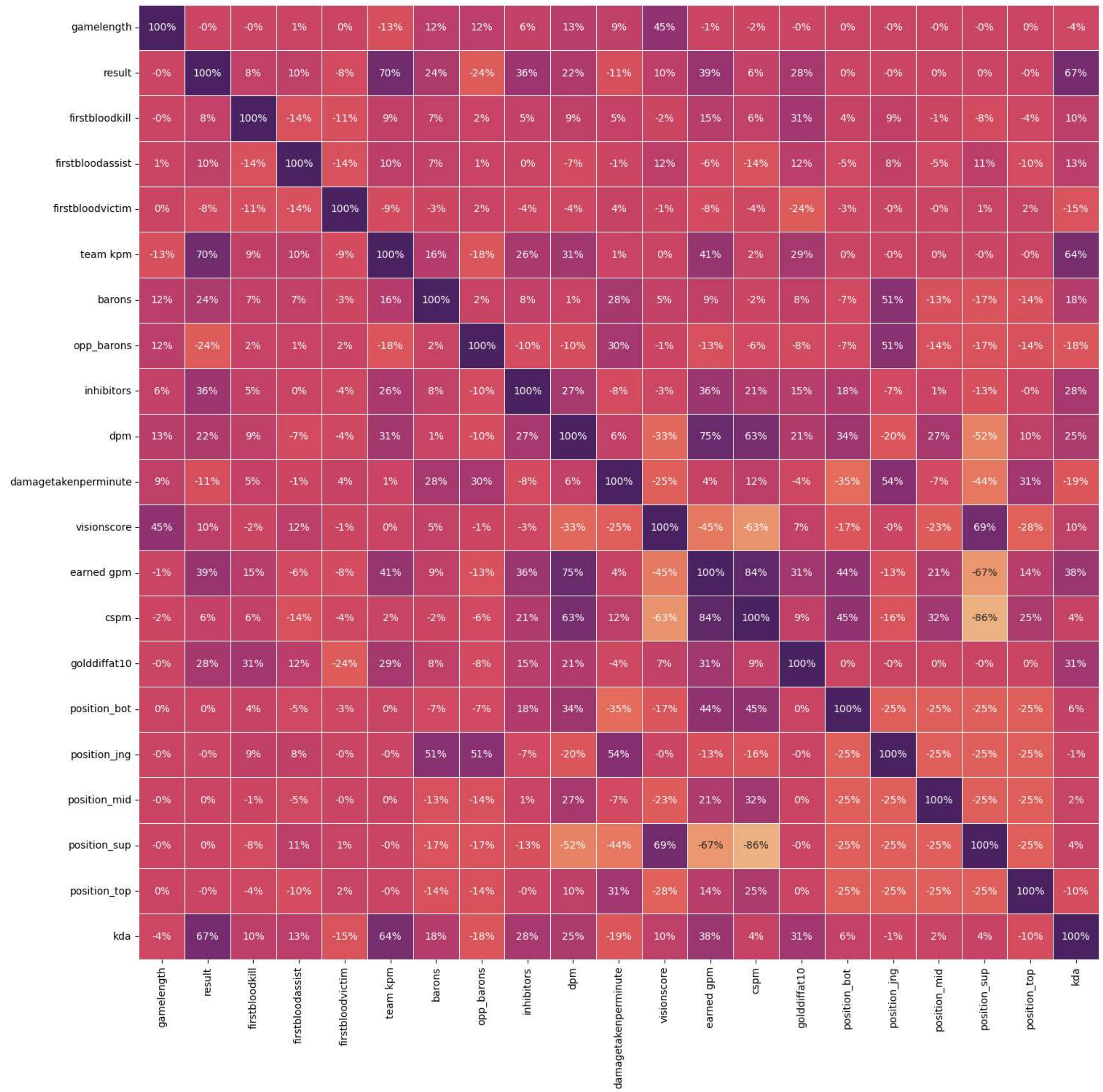
Feature Correlation Matrix
- Too many redundant features
- Standardized the data and performed feature selection

Configuring the Correlation Matrix
- Manual inspection
- Programmatically removing values over a specified threshold.



Player Performance Analysis

- Removed redundant features
    - Wards kills
    - Wards placed
    - Vision score
- Manual feature selection before algorithmic feature selection



Player Performance Analysis

Model	Alpha	MSE	RMSE
Elastic Net	0.006613574935661336	10.62266328957894	3.2592427478754846
Lasso (1st Model)	0.003306787467830671	10.462391247702186	3.234561986993322
Lasso (2nd Model)	0.003306787467830671	9.618834866112751	3.2592427478754846

	Elastic Net	Lasso (1st Model)	Lasso (2nd Model)
gamelength	0.016363	0.019969	0.016469
result	1.530820	1.528595	1.514470
firstbloodkill	0.033550	0.031021	0.028964
firstbloodassist	0.168931	0.168186	0.184970
firstbloodvictim	-0.324197	-0.325953	-0.329507
team kpm	1.253303	1.240388	1.233766
barons	0.072567	0.068835	0.082316
opp_barons	-0.087912	-0.088543	-0.075856
inhibitors	0.008753	0.003970	0.012062
dpm	0.160635	0.149988	0.156015
damagetakenperminute	-1.034454	-1.042921	-1.037783
visionscore	0.082562	0.076096	0.062009
earned gpm	1.688466	1.755590	1.795120
cspm	-0.716054	-0.781936	-0.830665
golddiffat10	0.089676	0.083239	0.077308
position_bot	-0.342338	-0.347551	-0.346279
position_jng	0.616786	0.616505	0.604199
position_mid	-0.000000	-0.000000	-0.000000
position_sup	0.310935	0.293839	0.300962
position_top	-0.068102	-0.066003	-0.068717

Player Performance Analysis

- Lasso (3rd Model)
  - Remove Jungle and Support players

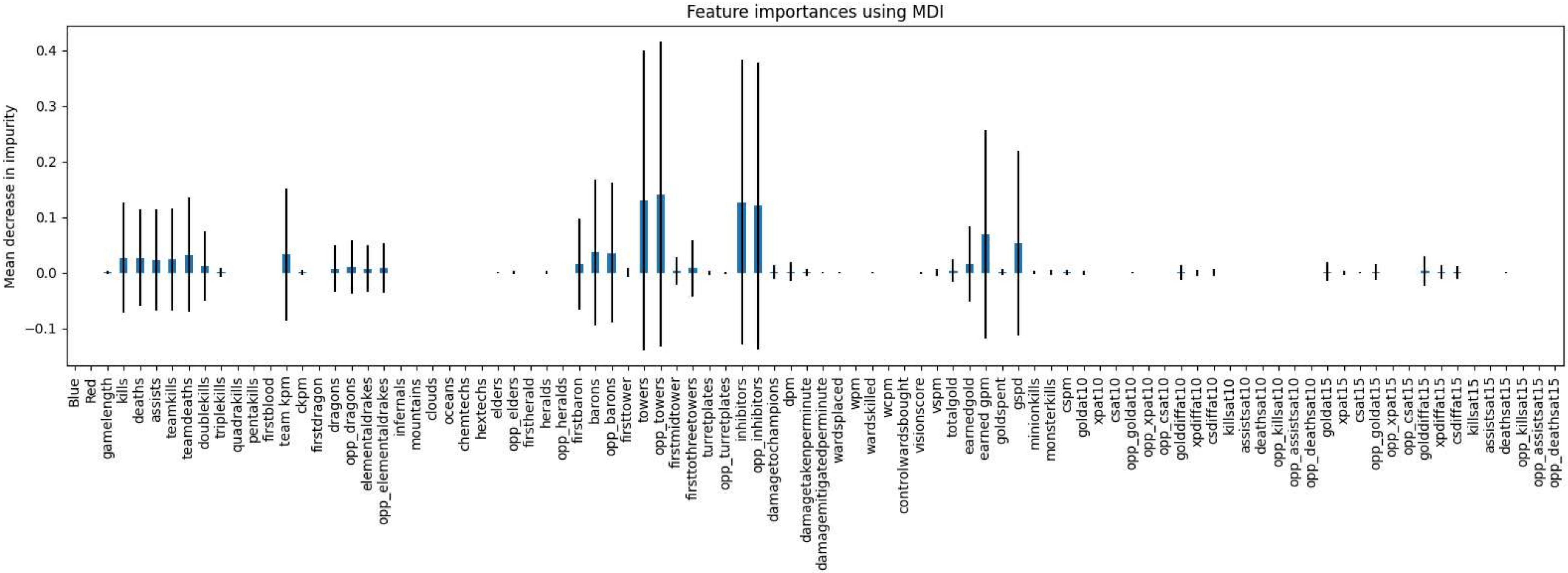
Lasso (3rd Model)
Alpha: 0.003306787467830671
MSE: 9.618834866112751
RMSE: 3.2592427478754846

gamelength	-0.084609
result	1.515701
firstbloodkill	0.042931
firstbloodassist	0.171845
firstbloodvictim	-0.320791
team kpm	1.173139
barons	0.083743
opp_barons	-0.034667
inhibitors	0.016006
dpm	0.256964
damagetakenperminute	-0.695696
visionscore	0.196611
earned gpm	1.136409
cspm	-0.181922
golddiffat10	0.029694
position_bot	-0.221354
position_mid	0.131389
position_top	-0.000000

Predicting Results of Matches

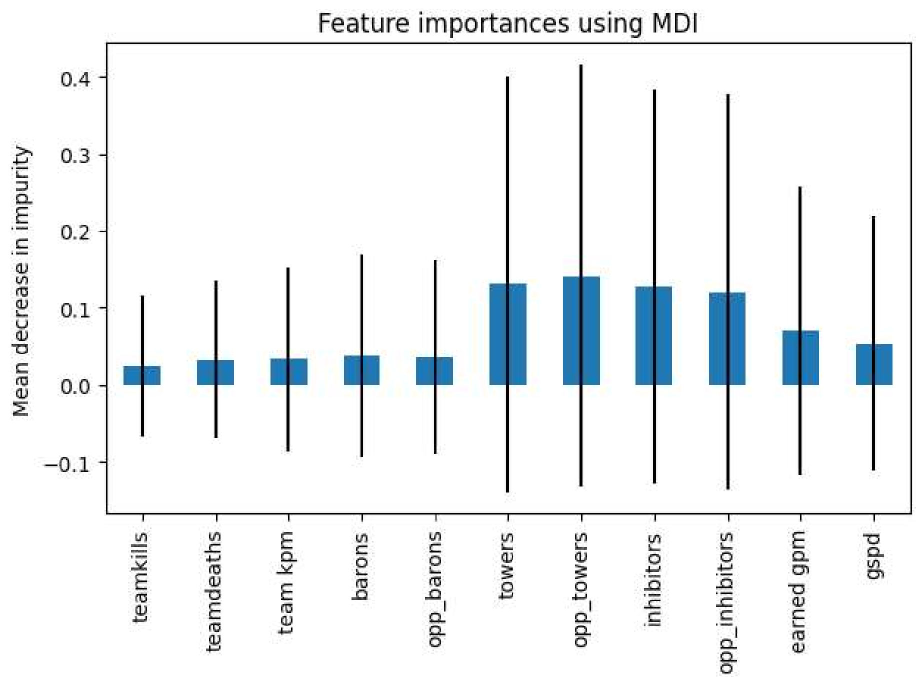
- Most important aspects of sports analytics: predicting winners.
- First approach: White box models
  - Important to understand how a model makes decisions
  - Logistic Regression
- Second approach: Black box models
  - Random Forest
  - Neural Networks

Predicting Results of Matches



Random Forest results

Predicting Results of Matches



Random Forest results, importance ≥ 0.02

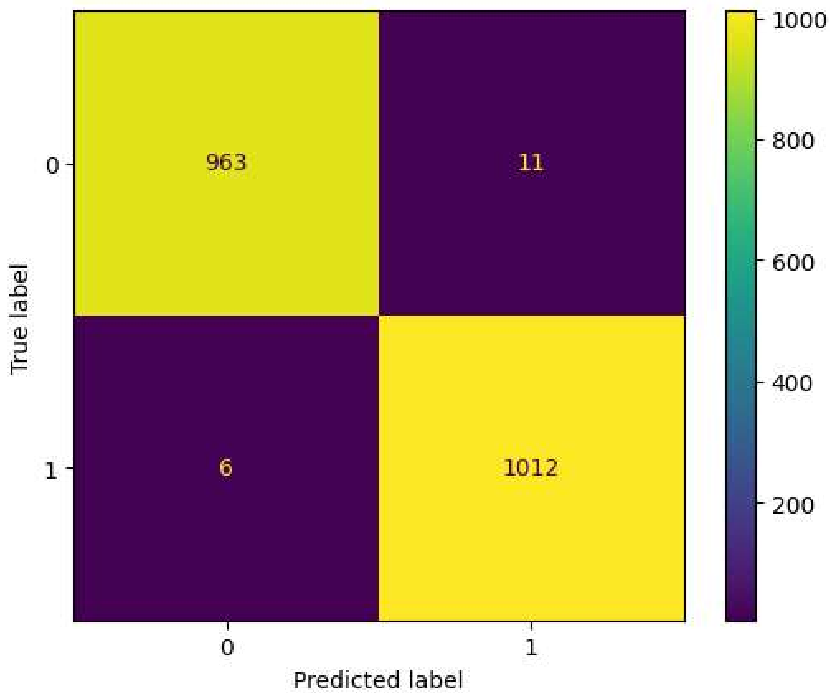
Predicting Results of Matches

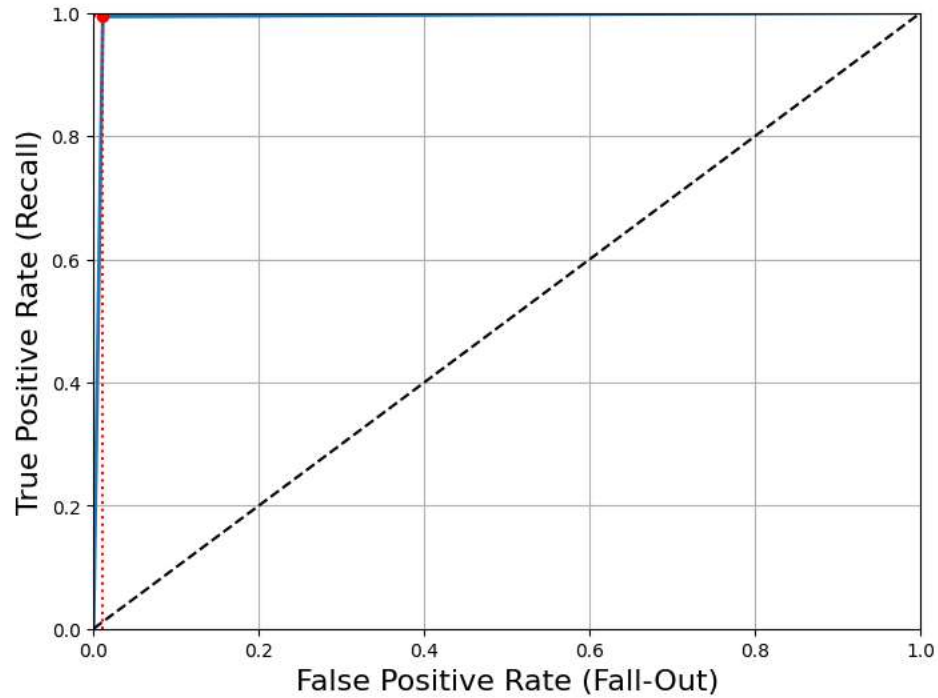
Generalized Linear Model Regression Results			
Dep. Variable:	results	No. Observations:	9956
Model:	GLM	Df Residuals:	9944
Model Family:	Binomial	Df Model:	11
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-176.82
Date:	Thu, 22 Jun 2023	Deviance:	353.64
Time:	17:52:57	Pearson chi2:	963.
No. Iterations:	11	Pseudo R-squ. (CS):	0.7410

Logistic Regression results

coef	std err	z	P>|z|	[0.025	0.975]	
Intercept	-19.3729	2.644	-7.326	0.000	-24.556	-14.190
teamkills	0.5195	0.083	6.286	0.000	0.357	0.681
teamdeaths	-0.4173	0.036	-11.509	0.000	-0.488	-0.346
team_kpm	-11.2241	3.159	-3.553	0.000	-17.416	-5.032
barons	-0.2478	0.290	-0.854	0.393	-0.817	0.321
opp_barons	-0.2202	0.287	-0.766	0.444	-0.783	0.343
towers	1.2213	0.136	8.970	0.000	0.954	1.488
opp_towers	-1.6059	0.136	-11.771	0.000	-1.873	-1.339
inhibitors	-0.5160	0.173	-2.991	0.003	-0.854	-0.178
opp_inhibitors	0.5090	0.166	3.068	0.002	0.184	0.834
earned_gpm	0.0224	0.003	8.233	0.000	0.017	0.028
gspd	-40.4736	3.514	-11.518	0.000	-47.361	-33.587

Predicting Results of Matches





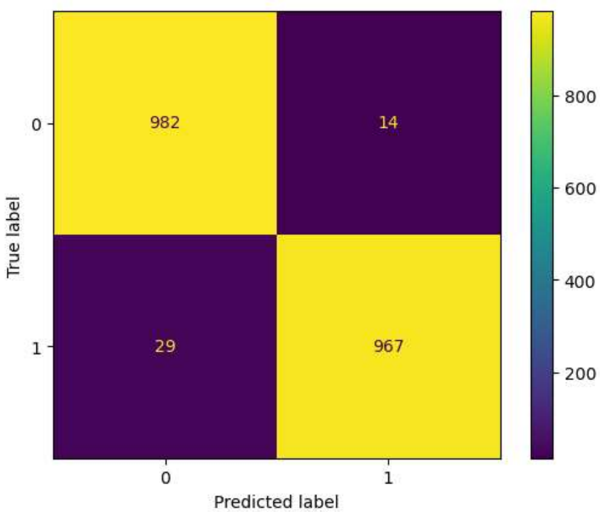
Logistic Regression results

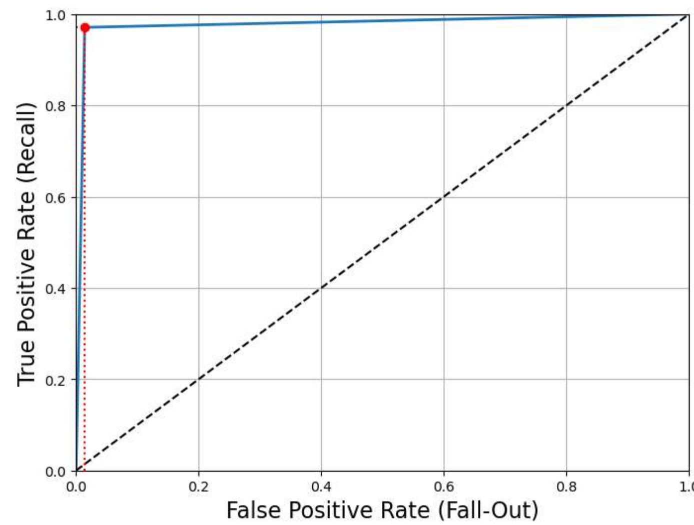
Predicting Results of Matches

- Neural Network 1
  - Used the same teams DataFrame we used for the rest of the analysis.
  - Problem: It only examines a single team at a time.
    - This resulted in the same that logistic regression had

Predicting Results of Matches

- Results were hard to interpret.
  - Probability of win for Team 1 (Gen. G): 0.9717535223444301
  - Probability of win for Team 2 (T1): 0.9981001104404654
  - roc score: 0.9784136...





Predicting Results of Matches

- Neural Network 2
  - Manipulated the teams DataFrame to better suit the problem.
  - Used the features from the random forest.

index	Side	Gold_Earned
1	Red	23,020
2	Blue	18,422
3	Red	30,991
4	Blue	27,634

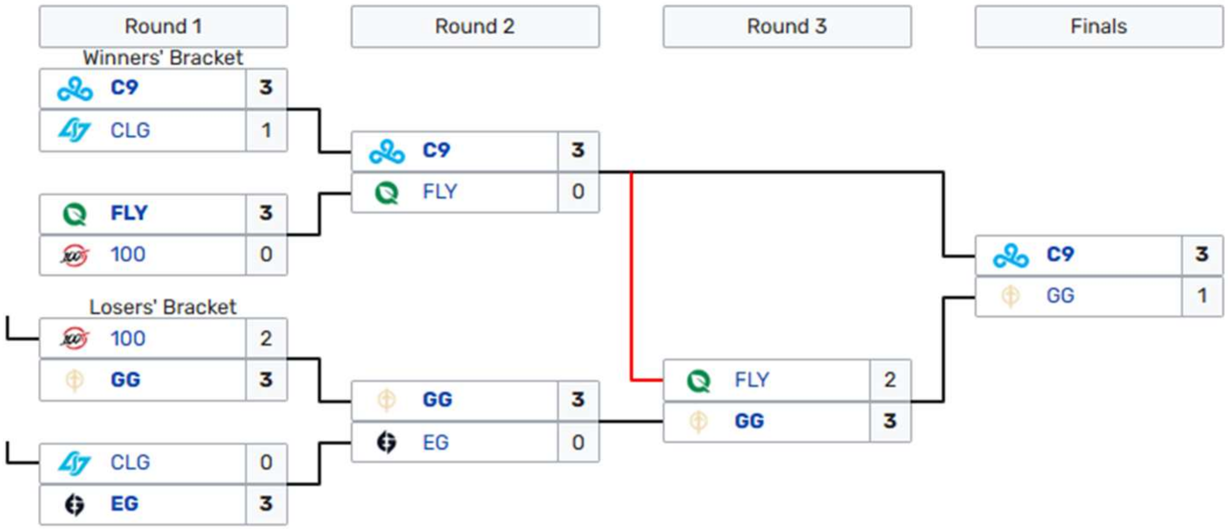
index	Side_red	Gold_Earned_red	Side_blue	Gold_Earned_blue
1	Red	23,020	Blue	18,422
2	Red	30,991	Blue	27,634

- Individual Teams
  - Combined every record for any give team by taking the average of desired features.
  - Combined the records for given two teams using the same method.

Predicting Results of Matches

- Neural Network performance
  - neural network train score=1.0
  - neural network test score=0.9909638554216867
  - both correct=987 (99.0964%)
  - one correct=1 (0.1004%)
  - neither correct=8 (0.8032%)
- Neural Network Tuning
  - Estimator: Grid Search Validation
  - Alpha = 0.001

Predicting Results of Matches



'Cloud9' defeated 'Counter Logic Gaming' in Upper Bracket Semifinals 1 [Correct Prediction]
'FlyQuest' defeated '100 Thieves' in Upper Bracket Semifinals 2 [Correct Prediction]
'GoldenGuardians' defeated '100 Thieves' in Lower Bracket Quarterfinals 1 [Correct Prediction]
'Evil Geniuses' defeated 'Counter Logic Gaming' in Lower Bracket Quarterfinals 2 [Correct Prediction]
'GoldenGuardians' defeated 'Evil Geniuses' in Lower Bracket Semifinals [Correct Prediction]
'Cloud9' defeated 'FlyQuest' in Upper Bracket Finals [Correct Prediction]
'GoldenGuardians' defeated 'FlyQuest' in Lower Bracket Finals [Correct Prediction]
'Cloud9' defeated 'GoldenGuardians' in Grand Final [Correct Prediction]
Correct Predictions: 8 (100.0%)

Analytical Conclusions

- We were able note interesting associations in champion groupings
- KDA analysis revealed that a few ways to have higher scores
  - have more battlefield awareness (e.g vision of the map)
  - selectively group with others for group conflicts
- Dataset may be too simple to consistently and accurately predict game results.
  - Our neural network showed 100% correct predictions on this run, but prediction accuracy ranged from 3/8 to 8/8.
  - League of Legends is a very complicated game due to the number of champions and items present.

Further Work

- Dataset focuses on game results
  - Does not contain data about moment to moment performance.
  - No information about the upgrade path a player chose (e.g. how they spent experience and gold.)
- This additional data may allow predictions based on teams of specific players instead of predefined groups.
- Combining more detailed datasets with the one used here might allow for greater understanding of the game.

Happy Data Sciencing!