An Analysis of Competitive Matches

Jared Martin

2024-05-19

Intro:

In this report I will be looking into the effect of different factors on the outcomes of competitive matches of the game Valorant. The game is played on a round by round format where the first team to win 13 rounds wins the game. However, if the score is 12-12 then you must play until one team wins by 2 rounds or the players vote to draw. There are two main cases I want to analyze. One is to find a model that can predict the outcome of the match using only pre game factors. This includes things like map, team compositions, time of day, how many games have been played already etc. The second is finding a model that predicts the outcome of the match once 4 rounds have been played, this will be done by adding varibles represented by the outcomes of rounds 1-4. Each of these two models serves a different purpose, and will have different expectations for predictive ability. For the pre game factor model I want to have a strong sensitivity. The model should be able to give a prediction on the outcome of the match with a strong true positive rate to give confidence to the player to trust the process. For the model including round data the main goal is to maximize the specificity, or true negative rate. The competitive environment of the game is not just a test of skill, but a test of mentality. So if the model can give strong predictions in relation to losing, it could help settle frustration and allow the player to focus on more technical aspects of their game while letting the current match run its course.

Data Set:

```
##  [1] "idx"         "Day"         "Date"        "Temp"        "Cloud"
##  [6] "Time"        "Steps"       "SleepT"      "SleepQ"      "SleepV"
## [11] "TOD"         "DOW"         "Num"         "PW"          "X2W"
## [16] "X3W"         "X4W"         "FFR"         "G."          "Team.Score"
## [21] "Opp.Score"   "Rounds"      "W."          "RndDiff"     "RD.."
## [26] "TRS"         "Perf"        "POS"         "PerPos"      "stst"
## [31] "W"           "W.L"         "W.L.."       "RR"          "RR.."
## [36] "Opp.Coms"    "Team.Comp"   "X"           "Team.Rank"   "Opp.Comp"
## [41] "X.1"         "Opp.Rank"    "Rank.Diff"   "TAGRO"       "OAGRO"
## [46] "TD"          "TI"          "TC"          "TS"          "OD"
## [51] "OI"          "OC"          "OS"          "DDIF"        "IDIF"
## [56] "CDIF"        "SDIF"        "Map"         "StartSide"   "Agent"
## [61] "Role"        "DD"          "HS."         "K"           "D"
## [66] "A"           "K.D"         "KA.D"        "X..."        "KPR"
## [71] "DPR"         "APR"         "Fights"      "FPR"         "F.win."
## [76] "X.2"
```

The 3 main categories of data I will be looking at are, Environmental factors, Pre Match Factors, and Round Factors. The Environmental Factors include the date, temperature, cloud cover, time of day, how active I was that day, sleep quality the night before, day of the week, number of games played consecutively, and if my friend was on my team or not. These are all

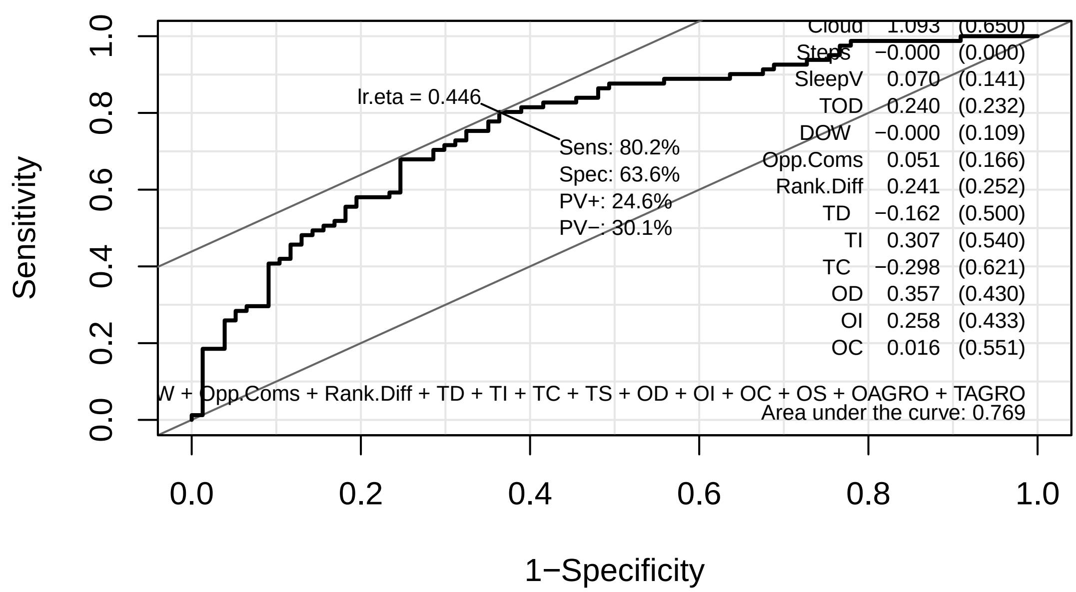
factors that are known when the match starts but are independent of the game. The Pre Match Factors include the map, if my team starts on attack or defense, the role I was playing, the number of people queued together on the other team, both my team and the opponents teams average ranks, as well as both teams role compositions. I also created some metrics which were added to this data. For example Rank.Diff is the difference in average ranks between the teams. Also TD, TI, TC, and TS as well as the OD, OI, OC, and OS variables are just the number of each role seen on either my team (T) or the opponents team (O). These basically break down the team role compositions into each specific role. Finally I created a metric called TAGRO for my team as well as OAGRO for the opponents. This is a measure of aggression based on chosen roles. With roles D and I typycally played by more aggressive players and have more of a fighting role, where roles C and S tend to be more passive positions on the team. This metric just ranges from 10 (More aggressive) to -10 (less aggressive) by summing the values assigned to each role for each team. (D:2, I:1, C:-1, S:-2). Finally are the Round Factors. This set only contains 4 variables, each one is a binary win or loss for each of the first 4 rounds of the match. PW being the pistol round, or round 1, X2W is round 2, X3W is round 3, and X4W is round 4. This is an overview of the data I will be working with.

Exploring Features for the Pre Game Model:

First I will output the summary and ROC curve for the fullmodel, which will use all Pre Match and Environmental Factors. I will also test it against the null model using the Likelihood Ratio Test to make sure that the model is significant.

```
##
## Call:
## glm(formula = W ~ Map + StartSide + Role + G. + Num + Day + Temp +
##     Cloud + Steps + SleepV + TOD + DOW + Opp.Coms + Rank.Diff +
##     TD + TI + TC + TS + OD + OI + OC + OS + OAGRO + TAGRO, family = binomial,
##     data = data)
##
## Coefficients: (4 not defined because of singularities)
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.233e+00  3.521e+00  -0.918  0.35850
## MapBind          -2.038e-01  6.858e-01  -0.297  0.76631
## MapBreeze         7.627e-01  6.526e-01   1.169  0.24249
## MapIcebox         2.281e-01  6.672e-01   0.342  0.73240
## MapLotus         -6.521e-01  7.735e-01  -0.843  0.39919
## MapSplit         -4.380e-01  6.535e-01  -0.670  0.50269
## MapSunset         2.172e+00  7.512e-01   2.891  0.00384 **
## StartSideDefense  2.647e-01  3.849e-01   0.688  0.49171
## RoleD             7.593e-01  8.181e-01   0.928  0.35335
## RoleI             1.117e+00  9.768e-01   1.143  0.25296
## RoleS             1.449e+00  9.434e-01   1.536  0.12462
## G.               -3.570e-02  4.148e-01  -0.086  0.93142
## Num              -2.155e-01  1.348e-01  -1.598  0.10995
## Day              -2.196e-02  1.893e-02  -1.160  0.24607
## Temp              1.547e-02  3.127e-02   0.495  0.62067
## Cloud             1.093e+00  6.496e-01   1.682  0.09259 .
## Steps            -2.443e-06  6.350e-05  -0.038  0.96931
## SleepV            6.967e-02  1.413e-01   0.493  0.62206
## TOD               2.404e-01  2.318e-01   1.037  0.29955
## DOW              -2.403e-04  1.088e-01  -0.002  0.99824
## Opp.Coms          5.077e-02  1.658e-01   0.306  0.75939
## Rank.Diff         2.408e-01  2.522e-01   0.954  0.33984
## TD               -1.619e-01  5.005e-01  -0.324  0.74626
```

```
## TI              3.071e-01  5.404e-01   0.568  0.56979
## TC             -2.983e-01  6.206e-01  -0.481  0.63076
## TS                    NA         NA      NA       NA
## OD              3.567e-01  4.299e-01   0.830  0.40666
## OI              2.576e-01  4.325e-01   0.596  0.55146
## OC              1.626e-02  5.511e-01   0.030  0.97646
## OS                    NA         NA      NA       NA
## OAGRO                 NA         NA      NA       NA
## TAGRO                 NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 218.93  on 157  degrees of freedom
## Residual deviance: 182.92  on 130  degrees of freedom
## AIC: 238.92
##
## Number of Fisher Scoring iterations: 4
```




```
## Analysis of Deviance Table
##
## Model 1: W ~ Map + StartSide + Role + G. + Num + Day + Temp + Cloud +
##     Steps + SleepV + TOD + DOW + Opp.Coms + Rank.Diff + TD +
##     TI + TC + TS + OD + OI + OC + OS + OAGRO + TAGRO
## Model 2: W ~ 1
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1       130     182.92
## 2       157     218.93 -27  -36.013    0.115
```

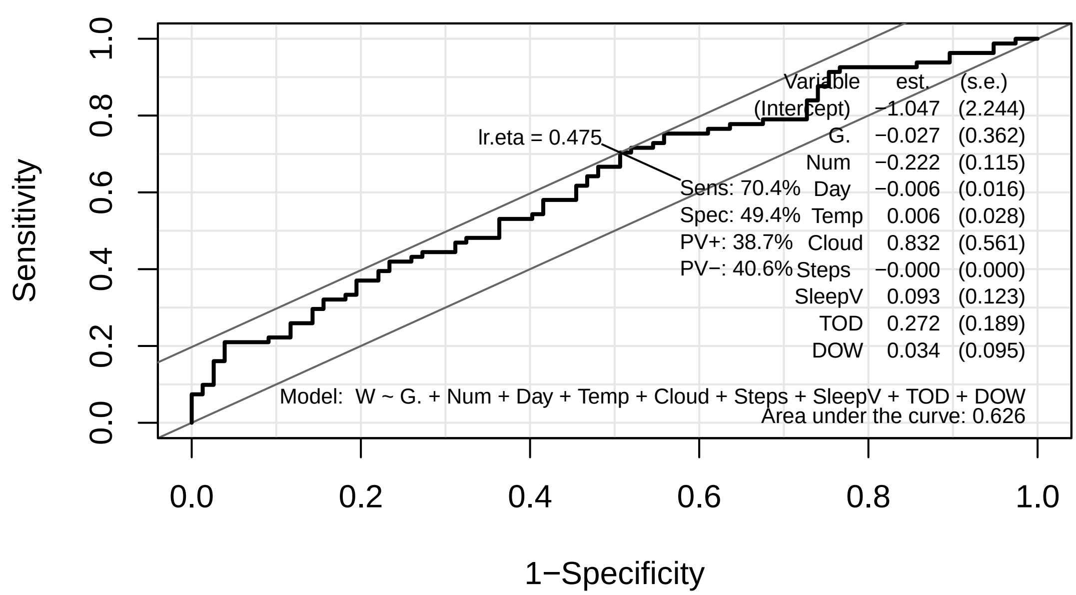
This model has an AIC of 238.92, an AUC of 0.769, sensitivity of 0.802, and a specificity of 0.636 at the cuttoff point of 0.446. There are some issues however with multiple NA values. This is due to multicollinearity between certain variables such as TAGRO and OAGRO being directly correlated with the numbers of each role on each team. Also TS and OS are both NA because the number of people in the S role depends on the number of people in the other 3 roles. Due to this I can remove TS, OS, TAGRO, and OAGRO, since they all can be represented by the other values. The LRT shows that the fullmodel is not significantly different from the nullmodel, so we are going to have to find th ebest features in order to optimize it.

First I will split the fullmodel into an environmental model and a pre match model, basically seperating the 2 aspect of the data to see which factors are stronger.

Environmetal Model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.04662	2.24369	-0.46647	0.64088
G.	-0.02722	0.36181	-0.07523	0.94004
Num	-0.22249	0.11486	-1.93699	0.05275
Day	-0.00613	0.01608	-0.38143	0.70288
Temp	0.00637	0.02752	0.23144	0.81698
Cloud	0.83216	0.56123	1.48275	0.13814
Steps	-0.00004	0.00005	-0.67903	0.49712
SleepV	0.09325	0.12260	0.76062	0.44688
TOD	0.27153	0.18914	1.43558	0.15112
DOW	0.03379	0.09520	0.35493	0.72264

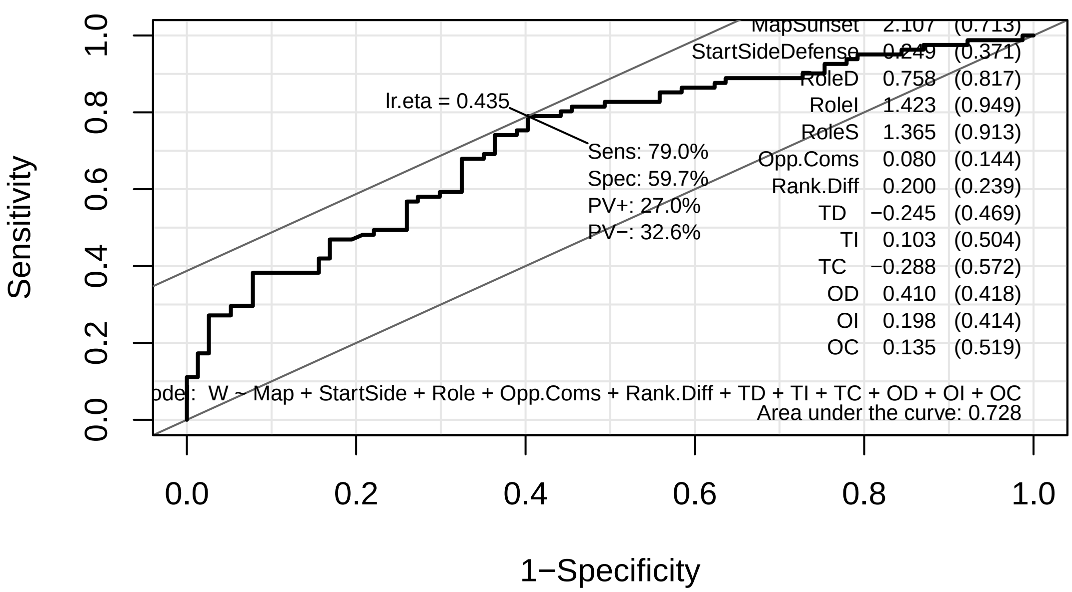
```
## Environmental Model AIC: 230.2532
```



Pre Match Model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.91420	2.42132	-0.79056	0.42920
MapBind	-0.23994	0.65908	-0.36405	0.71582
MapBreeze	0.68340	0.62173	1.09919	0.27169
MapIcebox	0.43728	0.60978	0.71711	0.47331
MapLotus	-0.57262	0.73331	-0.78087	0.43488
MapSplit	-0.50538	0.61900	-0.81644	0.41425
MapSunset	2.10705	0.71300	2.95521	0.00312
StartSideDefense	0.24890	0.37096	0.67097	0.50224
RoleD	0.75779	0.81741	0.92706	0.35390
RoleI	1.42272	0.94910	1.49901	0.13387
RoleS	1.36464	0.91337	1.49406	0.13516
Opp.Coms	0.07973	0.14354	0.55542	0.57861
Rank.Diff	0.19996	0.23908	0.83637	0.40295
TD	-0.24541	0.46896	-0.52329	0.60077
TI	0.10270	0.50447	0.20359	0.83868
TC	-0.28797	0.57186	-0.50356	0.61457
OD	0.41016	0.41812	0.98097	0.32661
OI	0.19761	0.41438	0.47689	0.63344
OC	0.13548	0.51893	0.26107	0.79404

```
## Pre Match Factor Model AIC: 228.719
```



The Environmental Factor model has a Higher AIC as well as a lower sensitivity, specificity, and AUC. With this information we can see that the pre match factors do play a better role in predicting match outcome. Now we can test this model against the full model using a Likelihood Ratio Test.

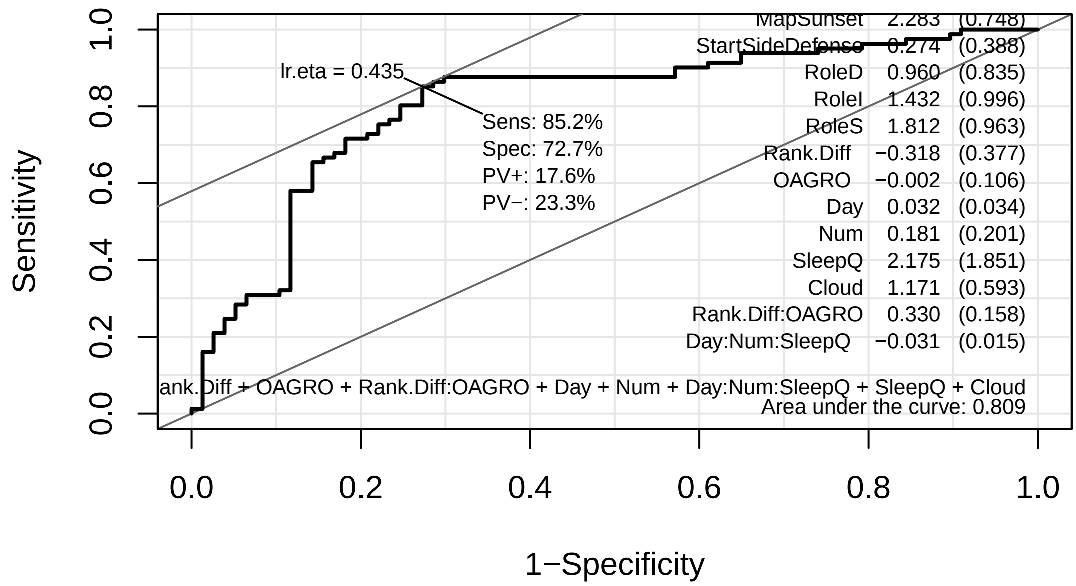
```
## Analysis of Deviance Table
##
## Model 1: W ~ Map + StartSide + Role + G. + Num + Day + Temp + Cloud +
##     Steps + SleepV + TOD + DOW + Opp.Coms + Rank.Diff + TD +
##     TI + TC + TS + OD + OI + OC + OS + OAGRO + TAGRO
## Model 2: W ~ Map + StartSide + Role + Opp.Coms + Rank.Diff + TD + TI +
##     TC + OD + OI + OC
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       130     182.92
## 2       139     190.72 -9   -7.799   0.5545
```

The Likelihood Ratio Test does not show that the models are significantly different. To try and make the model more accurate I will try and use domain knowlegde to test interactions between variables, as well as remove insignificant variables, to try and optimize the model.

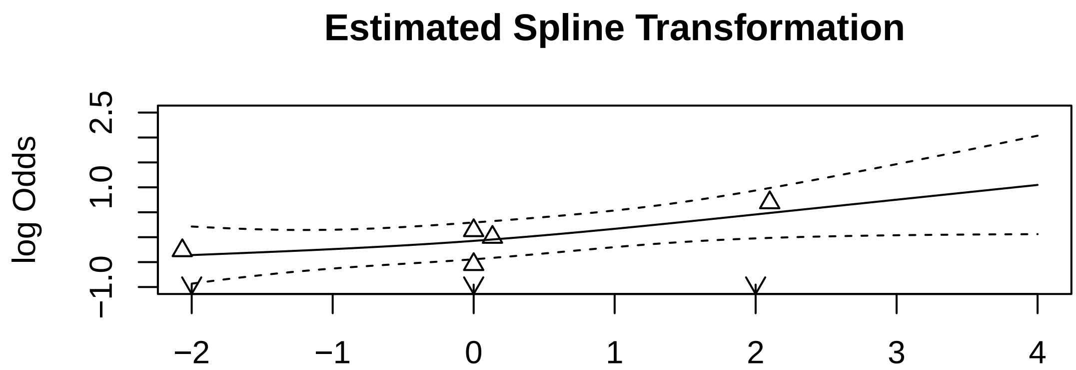
Looking at the 3 models I gathered the more significant variables and created a new modelcalled test.model. This model includes, map, startside, role, rank difference, OAGRO, day, the number of games that day, sleep quality, and cloud cover.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-3.34404	2.15015	-1.55526	0.11988
MapBind	-0.07343	0.67566	-0.10868	0.91346
MapBreeze	0.88184	0.63629	1.38590	0.16578
MapIcebox	0.23106	0.63877	0.36172	0.71756
MapLotus	-0.72990	0.76855	-0.94970	0.34226
MapSplit	-0.70866	0.64384	-1.10069	0.27103
MapSunset	2.28289	0.74797	3.05211	0.00227
StartSideDefense	0.27370	0.38820	0.70506	0.48077
RoleD	0.95986	0.83478	1.14985	0.25021
RoleI	1.43246	0.99593	1.43832	0.15034
RoleS	1.81186	0.96295	1.88158	0.05989
Rank.Diff	-0.31816	0.37676	-0.84446	0.39841
OAGRO	-0.00207	0.10614	-0.01948	0.98446
Day	0.03164	0.03377	0.93697	0.34878
Num	0.18146	0.20145	0.90075	0.36772
SleepQ	2.17546	1.85129	1.17510	0.23995
Cloud	1.17068	0.59259	1.97551	0.04821
Rank.Diff:OAGRO	0.33048	0.15750	2.09827	0.03588
Day:Num:SleepQ	-0.03081	0.01526	-2.01935	0.04345

```
## Test Model AIC: 213.2335
```



The first main interaction I wanted to test was OAGRO and rank difference. With OAGRO being a metric of how aggressive a teams role selection is, I was curious to see that if a team with a higher rank than us chose aggressive roles, they would have a negative effect on the log odds of winning. The interaction is the product of Rank.Diff and OAGRO, and we can see its relationship to the log odds of winning in this spline plot here:



This shows that it was true that higher ranked teams playing more aggressive comps decreased the odds of my team winning. It also showed that low ranked teams playing passive comps decreased the odds of winning. Typically lower AGRO scores for teams tend to consist of a more even distribution of roles on the team. This allows for more of the startegical advantages of the more passive roles to shine and help team play overall. More aggressive teams tend to be less about team play and often lose, unless the other team is much better so they can rely on aggressive gameplay to win. This is shown here that playing aggressive decreases my odds of winning when they are the higher rank, and playing a more passive and strategic game decreases my odds of winning when my team is the higher rank.

Another interesting interaction I found to be significant was between day, num, and SleepQ. Day represents the integer number of the day the match topok place since I started tracking the data. Num represents the game as the nth game of the day with n starting at 1 for the first game of the day all the way to the last game of the day. SleepQ is a measure of sleep quality based on statistics from a sleep tracking app I use on a nightly basis. The interaction I was looking for here was a relationship between sleep quality and fatigue from playing many games in a single day. Then seeing how this interacted with overall fatigue of playing over time. I did some testing with opponent rank to see that if using day was just capturing the effects of oponnents getting harder as I ranked up, but I found no significance. So the model did show that the interaction between day, num, and sleepQ was had a significant negative effect on my odds of winning. This is most likely due to low sleep leading to my later games being much worse as I get tired much faster.

One variable I found interesting was cloud cover. I feel like my head is more clear, im more focused, and generally in a better mood on overcast days. So it was interesting for me to see that the level of cloud cover during my matches has a significant positive effect on the odds of winning.

Finally for the map variable I was able to see how I performed on the different maps in the game. This I could already tell from experience was going to be very significant. For the variables I used in this test model, MapSunset was the most significant variable. The map used in the intercept was Ascent, which is a map that my teams never perform well on. This is a large reason why the intercept is negative. Maps like Split and Lotus while, not significant, had the largest negative effects on the odds per unit change of any other variable. This makes lots of sense because I am terrible on these maps. This shows that not having a variety of maps I am good on hurts my overall win rate.

Conclusions on Pre Game Model

With an AIC of 213, the test.model outperformed the null model, full model, evironmental model, and the pre match model.

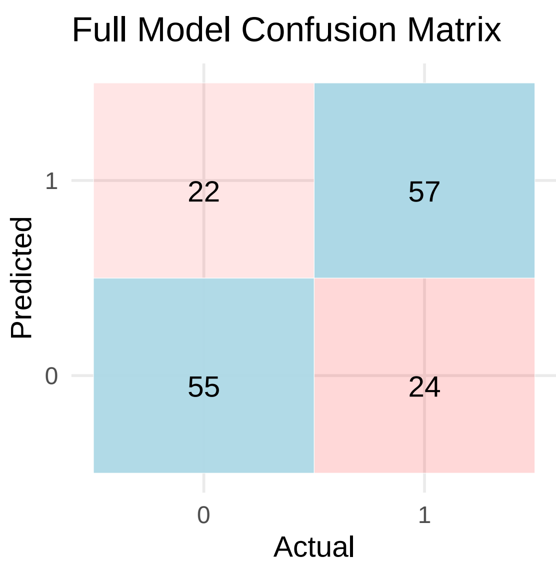
```
## Null Model AIC: 220.9332
```

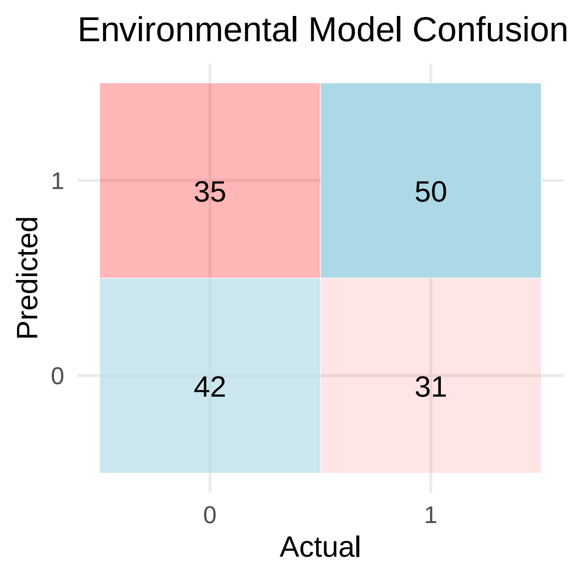
```
## Full Model AIC: 238.92
```

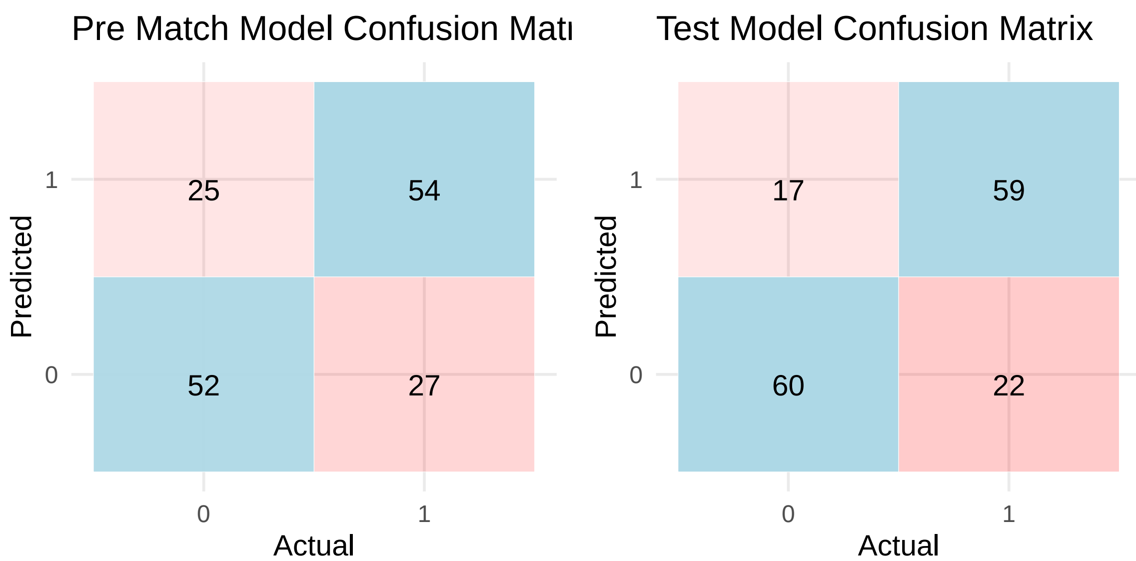
```
## Environmental Model AIC: 230.2532
```

```
## Pre Match Model AIC: 228.719
```

```
## Test Model AIC: 213.2335
```







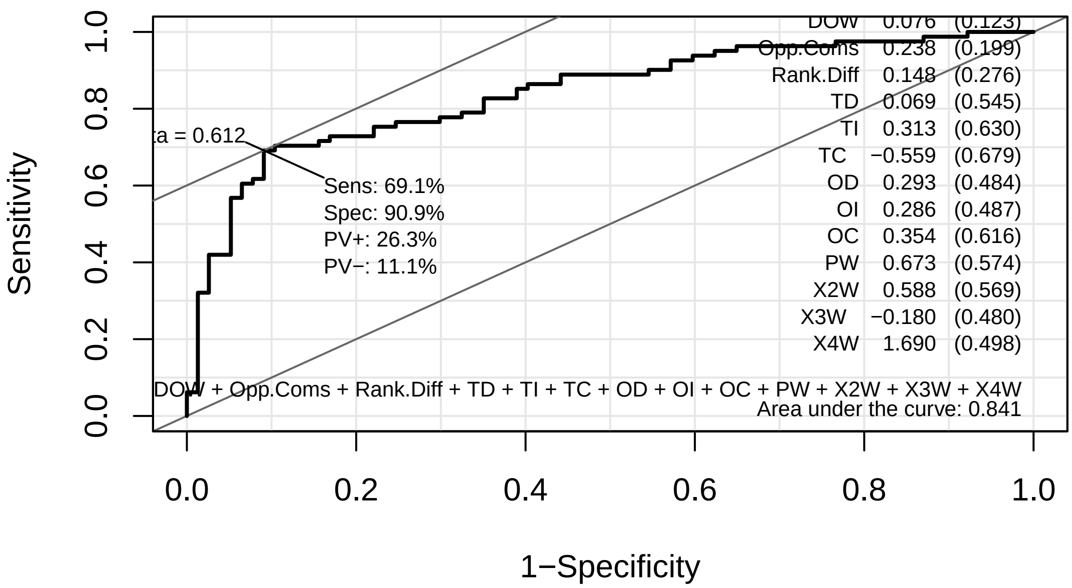
We can see in the confusion matrices that the test model outperforms the other models in every quadrant. This is also shown by its high sensitivity at 85.2% and good specificity at 72.7%. Also have an AUC of 0.809, while not of the other models were able to hit the 0.8 mark. With high performance, as well as a strength leaning towards sensitivity, which is what I was looking for in this model, I decided to use the test model as my Pre Game Factor Model.

Exploring the Round 4 Prediction model

This model will use the same factors as the full model from before, but now I will be including the round outcomes of the first 4 rounds. To start this again I will create a new full model for the round prediction.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-6.84140	4.10043	-1.66846	0.09522
MapBind	-0.15777	0.77640	-0.20321	0.83897
MapBreeze	0.65131	0.70443	0.92458	0.35518
MapIcebox	0.18325	0.72344	0.25330	0.80004
MapLotus	-0.58339	0.86235	-0.67651	0.49871
MapSplit	-0.19080	0.73418	-0.25988	0.79496
MapSunset	2.31559	0.88366	2.62044	0.00878
StartSideDefense	0.45408	0.46382	0.97900	0.32758
RoleD	0.63156	0.90296	0.69943	0.48429
RoleI	0.80891	1.06550	0.75918	0.44774
RoleS	1.16642	1.04035	1.12118	0.26221
G.	0.08215	0.46395	0.17707	0.85945
Num	-0.28067	0.15259	-1.83940	0.06586
Day	-0.02002	0.02110	-0.94894	0.34265
Temp	0.03917	0.03589	1.09131	0.27514
Cloud	1.16108	0.68836	1.68674	0.09165
Steps	-0.00003	0.00007	-0.46563	0.64148
SleepV	0.04018	0.15608	0.25747	0.79682
TOD	0.26298	0.26524	0.99147	0.32146
DOW	0.07588	0.12255	0.61919	0.53579
Opp.Coms	0.23835	0.19851	1.20070	0.22987

	Estimate	Std. Error	z value	Pr(>|z|)
Rank.Diff	0.14806	0.27646	0.53555	0.59227
TD	0.06905	0.54516	0.12666	0.89921
TI	0.31322	0.63047	0.49679	0.61933
TC	-0.55883	0.67883	-0.82323	0.41038
OD	0.29272	0.48375	0.60511	0.54511
OI	0.28620	0.48666	0.58809	0.55647
OC	0.35418	0.61593	0.57503	0.56527
PW	0.67324	0.57377	1.17336	0.24065
X2W	0.58836	0.56865	1.03467	0.30082
X3W	-0.17964	0.47976	-0.37444	0.70808
X4W	1.68984	0.49784	3.39433	0.00069



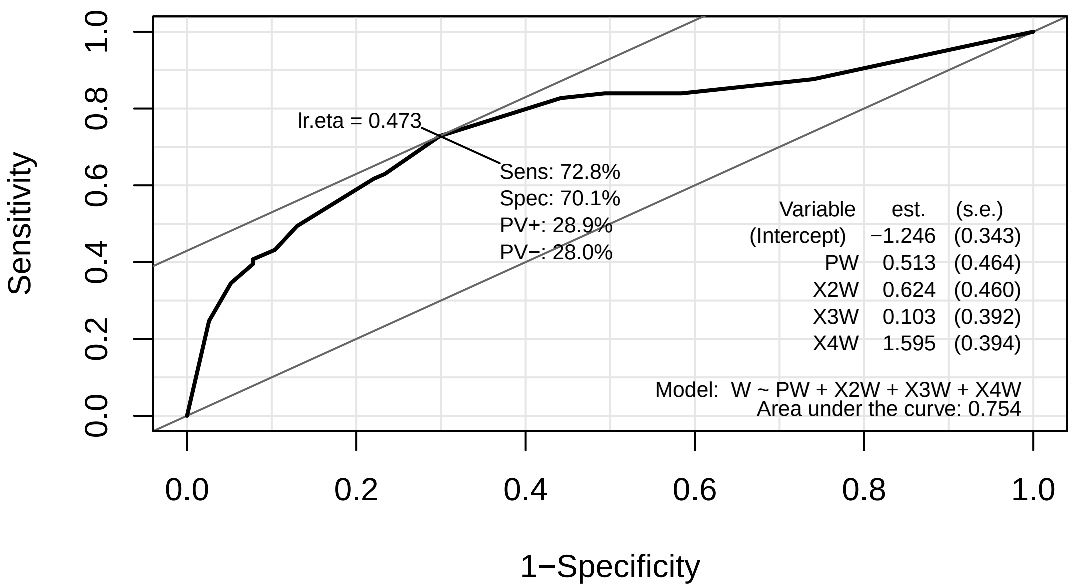
The full model shows very high specificity at 90.9% and good sensitivity at 69.1. To make sure that this model is not overfitting I used k-fold cross validation.

```
## [1] 0.2756048 0.2692169
```

The residual deviance per observation for the full model was 156.75/158 which is 0.99, however the k-fold cross validation gave us an error of 0.27. Since the cross validation error is much lower than the residual error, the model is not overfitted.

Next I wanted to see how well a model that only used the round outcomes as varaibles would perform, so I created the round.model, this is its summary and ROC curve,

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.24634	0.34263	-3.63754	0.00028
PW	0.51274	0.46449	1.10387	0.26965
X2W	0.62440	0.46046	1.35603	0.17509
X3W	0.10321	0.39249	0.26297	0.79257
X4W	1.59474	0.39396	4.04798	0.00005



The summary of the round.model shows that the intercept and round 4 are the most significant. This makes sense since the intercept is when all variables are set to 0, meaning in this case the intercept means my team lost the first 4 rounds of the game and therefore the odds of my team wining drop a significant amount. It was interesting though to see that round 4 was much more significant than the other rounds of the game, with the first round being next, then round 2, and round 3 being the least significant. I found this very interesting because it actually unveils some of the underlying strategy and framework built into the game.

The game has an economy system which adds more complex dynamics to the game. Things like when your team should save and when your team should go all in. When a team wins round 1 they go into round 2 with an advantage. If that team then wins round 2 given the advantage they had, they purposefully will go into round 3 with a disadvantage. This may seem weird, but due to the economics of the game, they will gain a larger advantage by taking a gamble on round 3. Round 4 is then when you see if that gamble payed off or not. These relationships can be seen in the correlation matrix here,

```
##              PW      X2W         X3W       X4W
## PW   1.00000000 0.6202532 -0.05066946 0.2169349
## X2W  0.62025316 1.0000000  0.12667365 0.1148479
## X3W -0.05066946 0.1266736  1.00000000 0.3368716
## X4W  0.21693487 0.1148479  0.33687163 1.0000000
```

Pistol round, or round 1 and round 2 have the highest correlation with eachother, which makes sense given the advantage. Then Pistol round and round 3 have a slightly negative correlation. While it is not enough to be significant it still shows that underlying gamble that teams often take. Given this information about the games flow we can see why round 3 was the least significant in predicting wins.

Round 4 Prediction model Conclusion

After checking through all the models that excluded the insignificant variables in the full model, I was not able to find any models that could outperform the full model in specificity. It seemed to cap out at 90.9%, as I found 2 other models with the same specificity. So looking at these models I will just pick one based on the AIC,

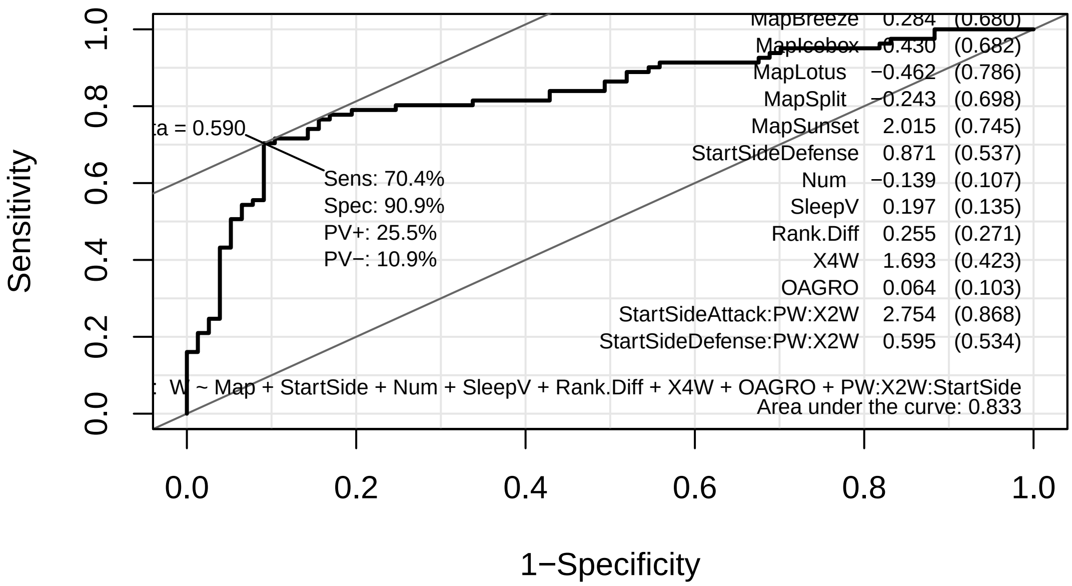
```
## Round 4 FUll Model AIC: 220.751
```

```
## First New Model AIC: 187.4373
```

```
## Second New Model AIC: 190.8338
```

Based off of the AIC for the model it looks like the first model had the best trade off of specificity and simplicity. Here is the model,

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.69273	1.00543	-2.67820	0.00740
MapBind	-0.01906	0.69442	-0.02744	0.97811
MapBreeze	0.28430	0.68001	0.41809	0.67588
MapIcebox	0.43038	0.68207	0.63099	0.52805
MapLotus	-0.46229	0.78624	-0.58797	0.55655
MapSplit	-0.24295	0.69820	-0.34796	0.72787
MapSunset	2.01541	0.74522	2.70445	0.00684
StartSideDefense	0.87111	0.53746	1.62078	0.10507
Num	-0.13875	0.10680	-1.29921	0.19387
SleepV	0.19666	0.13540	1.45249	0.14637
Rank.Diff	0.25468	0.27089	0.94014	0.34715
X4W	1.69279	0.42295	4.00232	0.00006
OAGRO	0.06434	0.10317	0.62366	0.53285
StartSideAttack:PW:X2W	2.75359	0.86813	3.17186	0.00151
StartSideDefense:PW:X2W	0.59475	0.53372	1.11435	0.26513



I think the most interesting part about this model is the significance of the interaction between winning round 1, winning round 2, and which side you start on. Winning rounds 1 and 2 is significant to the odds of winning when my team starts on attack. In the game defending tends to be much easier, and attacking is often what makes or breaks a team. Its interesting seeing this dynamic in the predictors. As its seen that starting strong on attack has a significant impact on the odds of winning.

Final Thoughts

In the end these are the 2 final models,

Pre Game Model:

glm(W~ Map+StartSide+Role+Rank.Diff+OAGRO+Rank.Diff:OAGRO+Day+Num+Day:Num:SleepQ+Slee data=data, family=binomial)

Round 4 Model:

glm(W~ Map+StartSide+Num+SleepV+Rank.Diff+X4W+OAGRO+PW:X2W:StartSide, data=data, family=binomial)

In this project I found 2 different models to predict the outcomes of Valorant competitive matches. One model was meant for pre game data that could be used to find the chances of winning before the game started. For this model I looked for low AIC as well as High Sensitivity for a string true positive rate. In the second model I wanted to find the chances of winning once 4 rounds of the game have been played, and combine this with the pre game variables. In this model I also looked for low AIC, however I also wanted to optimize the specificity for a high true negative rate. I ended up finding models with 85.5% sensitivity for the first, and 90.9% specificity for the second. I also was able to find many interesting insights

and interactions in the predictors and how they effected the odds of winning. It was fascinating to see the underlying strategies of the game show up in the models.