How to Survive in PlayerUnkown's Battlegrounds (PUBG)

Chen Yiran (Becky)

12/06/2019

Summary

This project aims to analyze user post-game statistics of a popular shooting game PlayerUnknown's Battlegrounds (PUBG) in order to see what factors and to what extent they would affect whether or not the team gets into the top ten percentile (defined as winning) in a game for squad team players. From that, some winning strategies for actions are proposed. By using logistic regression, it is found out that the cooperation among teammates is essential, giving one or more assists or revives of teammates greatly increases the odds of winning. On top of that, having a more aggressive strategy to eliminate the enemies is better than "playing it safe" and hiding. Compared to average, having more number of kills or enemies knocked down and higher headshot accuracy than average would increase the odds of winning. Using more boosting items than average is advantageous which speeds up player's movement. Therefore it is good idea to keep full booster meter both for attack and defend whenever possible.

Introduction

In each match of PUBG, there are around 100 players, i.e. 25 squad groups, parachute onto an island and scavenge for weapons to kill while avoid being killed for each match. The available moving area of the game shrinks in size periodically, forcing players to face each other over time and eliminating those who are falling too far out. The players are free to use different weapons to cause damages, ride vehicles, run or swim, shoot, and revive down-but-not-out (knocked) teammates. The winning placement percentile for every group member remains the same. As long as there is one member standing, all group members win the battle as the same first ranking.

The project aims to analyze user post-game statistics across different matches to reveal better gaming strategies to increase the chance of winning as the last one standing group using logistic regressions. Targeted specifically at players in squad match types, the average game statistics for players in a group are examined. In order to have an accurate reflection of the average player performance, several new features apart from given covariates are created and investigated to identify cheaters which are to be deleted. The factors which greatly affects the chance of winning will be identified and quantified, from which better game strategies will be suggested.

Data

The data is obtained from Kaggle competition where each row each row contains one player's post-game statistics for each group he or she belongs to in each match (https://www.kaggle.com/c/pubg-finish-placement-prediction/data).

The raw dataset used contains 4,446,966 rows and 25 attributes (Appendix 1). This is then filtered by squad match type. By removing misleading or deprecated game metrics, the original predictors include: *assists*, *boosts*, *damageDealt*, *DBNOs*, *headshotKills*, *heals*, *kills*, *killStreaks*, *numGroups*, *revives*, *rideDistance*, *roadKills*, *swimDistance*, *teamKills*, *vehicleDestroys*, *walkDistance*, *weaponsAcquired*.

After checking with no missing values in the dataset, new features are created in order to identify possible cheaters and provide more insights, including sum of number of kills and number of knocked-down enemies (*killsandknocked*) as a refelction of the aggressiveness of the player, total distance traveled (*totalDistance*), the accuracy rate of giving headshot (*headshot_rate*, where players with 0 kills are counted as 0).

Some possible cheating behaviors were spotted and such player records along with that team records were then discarded: There is one player who achieved more than 30 kills in a single match, 3 had kills without any moving distances, and 2 obtained more than 80 weapons. Players with 100% headshot accuracy might seem suspicious as cheaters, however, they could be simply really good players and therefore were not removed. There are also some players who have three or four teammate kills which seem dubious but were kept for some eccentric playing style.

After removal of potential cheater records, the interested variables are aggregated by each user's respective group and respective match by taking the mean value of members in that group for that match. This is because the fact that each group is able to attend multiple matches, i.e. one group can win in one match but lose in another. Therefore it would be most reasonable to separate both situations as data input by grouping by groupId and matchId at the same time. There are 177,079 rows left after aggregation. It is then randomly sampled to have 10,000 records as the dataset in the following analysis. The outcome variable is created as 1 when winning placement percentile equals 1, and 0 otherwise.

Transformation

Most data of *assists*, *revives*, *teamKills*, *roadKills*, and *vehicleDestroys* are clustered around small values with limited range. These predictors are made into categorical variables of having 0 values or not. All the continuous variables are all mean centered in order to give results compared to an average player, denoted with a letter c at the end of variable name.

EDA

For categorical variables, chi-squared tests were conducted, and it shows that whether or not providing assists to teammates, whether or not helping to revive teammates, whether or not the player chooses to kill any teammates, and whether there kills in a vehicle, are all significant for the outcome. On the other hand, whether or not the player chooses to destroy any vehicles is not significant.

From binned plots of continuous variables, there seem to have positive trend with all the game statistics except number of groups participated in each match. Such results were also supported by boxplots where it can be seen that most non-winning players are mostly clustered below the average in all predictors while winning players tend to have higher values and means in both kills, distance traveled, healing and boosting items used, accuracy of giving headshots, and the number of weapons acquired.

Regarding interactions, it also seems that whether or not giving assists to teammates will affect how *killsandknocked* resulting in getting into top ten percentile of a game. Naturally by providing assists to teammates, the player helps his or her teammates to live longer, and therefore the chance of winning as a group is larger.

Variable Selection

It was found out from the correlation matrix that some noticeable variable pairs which are highly correlated: *kills* and *killStreaks*; *kills* and *damageDealt* and *DBNOs*. As the max number of enemy players killed in a short amount of time (*killStreaks*) is directly linked to the number of successful kills, it is more reasonable to focus on the total amount of kills in a match instead of targeting at a short period of time. It is very likely for a player who have battled, knocked or killed more enemies to receive more damages. The number of enemies getting knocked is directly associated with the number of enemies killed. From the three original predictors, *killsandknocked* is chosen to be included as a measure of player aggressiveness. On the other

hand, the total distance traveled (*totalDistance*) is not a suitable metric to be included in the model, as it is largely determined by external factor of the team location with regard to a randomly generated player zone which continually shrinks by time. Specifically, a player will continuously receive damages if caught outside the circle and therefore need to move more to get inside. If he or she luckily lands within the circle already then the moving distance will be significantly reduced. As the data is averaged by team, the effect of this external factor on distance will hinder the true relationship and not appropriate to include for the purpose of the project.

Model

Model Building Process

The methodology adopted in the model building was initialized by including all categorical variables, mean-centered numeric variables, and the interaction between *assists_bi* and *killsandknocked*. In this full scope model, *vehicleDestroys_bi* and *roadKills_bi* were found not significant, while the interaction term was found significant. These are also supported by ANOVA tests comparing models which add one term at a time. The full model is put through stepwise selection with AIC as the judgement criteria for model fit.

Table 1: Logistic Models

Model	AIC	BIC	Threshold	AUC	Accuracy	Sensitivity	Specificity
all indiv	4418.034	4504.558	0.3	0.9038087	0.9020	0.5807050	0.9408204
all indiv+interaction	4401.898	4495.633	0.3	0.9050059	0.9026	0.5825603	0.9412688
stepwise result	4397.794	4469.897	0.3	0.9049843	0.9025	0.5834879	0.9410446

Final Model

The final model can predict the true positives and true negatives proportionally with an overall accuracy of 89.4%. The sensitivity and specificity with threshold 0.3 suggest that the model is 55.0% correct at identifying true positives (i.e. winning teams that made into top 10 percentile), and 93.8% correct identifying true negatives who did not made it.

$$logit(Pr[outcome = 1]) = \beta_0 + \beta_1 assists\_bi_i + \beta_2 heals\_bi_i + \beta_3 boostsc_i + \beta_4 revives\_bi_i$$
$$+\beta_5 teamKills\_bi_i + \beta_6 weaponsAcquiredc_i + \beta_7 killsandknockedc_i + \beta_8 numGroupsc_i$$
$$+\beta_9 headshot_r atec_i + \beta_{10} assists\_bi_i : killsandknockedc_i + \epsilon_i$$
$$\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

Results

In the final logistic regression, the baseline is an player in a squad team who gives no assists, no revives to teammates, and also do not kill teammates, who have average number of all numeric game statistics who will have probability of 2.9% of getting into top ten percentile ("winning").

Holding all other variables constant, having one or more assists to teammates increases the winning odds by 104%. Holding all other variables constant, having revived knocked-down teammates increases the odds of winning by 90%. While on the other hand, if the player chooses to kill one or more his or her teammates,

Table 2: Final Model Summary

term	estimate	std.error	statistic	p.value
(Intercept)	-3.8281020	0.0971403	-39.407975	0.0000000
assists_bi	0.9922140	0.1056760	9.389212	0.0000000
healsc	-0.0947236	0.0167923	-5.640892	0.0000000
boostsc	0.6604001	0.0306786	21.526422	0.0000000
revives_bi	0.5372575	0.0915978	5.865396	0.0000000
teamKills_bi	-0.4206762	0.1552287	-2.710041	0.0067275
vehicleDestroys_bi	0.2524819	0.1531967	1.648089	0.0993344
weaponsAcquiredc	0.1599849	0.0252163	6.344500	0.0000000
killsandknockedc	0.1097858	0.0299178	3.669576	0.0002430
assists_bi:killsandknockedc	0.1518337	0.0362484	4.188696	0.0000281

then the odds of winning decreases by 44%. These suggest that taking the effort and risk to help teammates is worthwhile and team cooperation is essential for the group to success.

Regarding medicines, with one unit more healing item used compared to average, the winning chances decreases by 6%. As healing items are only useful when receiving damages causing HP lower than 75%. Using more healing items suggests more damage experienced for the player and that other players are out winning. On the other hand, boosting items are able to recover HP and at the same time provide movement speedup within a short period of time. With each additional boosting item taken compared to average, the chance of winning increases by 85%.

With each additional weapon obtained compared to average, the winning odds increases by 16%. With every extra enemy killed or knocked down compared to average, the winning odds increases by 8%. Apart from trying to gain advantages by possessing more weapons and being aggressive to have more kills, with each unit increase in accuracy of headshots, the winning odds increases by 75%. Regarding the interaction term, given the player provides assists to teammates, with each extra kill or knocked down with respect to average number, the odds of winning increases by 20% compared to baseline. Naturally if the player provides help teammates, they are all able to survive longer, and eventually contribute to team victory.

Assessment

By checking with variance inflation factors, the covariates in the final model do not have multicolinearity issue. The confusion matrix is defined for the model. With threshold being 0.3, the model accuracy is 89.4% while sensitivity is 55.0% and specificity is 93.8%. The reason behind adjusting to a lower threshold is that as the data itself is imbalanced with around 11% records making into top ten percentile in different matches, the model is not able to predict the true positives very well but can highly accurate predicting true negatives. The model has a rate of 89.6% (AUC) of successful classification. Finally, k-fold cross validation with k=10 was conducted, which validated model's overal predictive ability with an AUC value of 89.5%.

From binned residual plots of fitted values, there seems to have a downward trend suggesting the model is somewhat limited at predicting the true positives and there might exist some other relationships which were not captured. In addition, there are some outliers falling outside the error bounds for *boostsc*. Transformations on this predictor were attempted such as square, square root, cubic, logarithm, etc. However, the residual plot did not seem to improve with persisting outliers lying at low and high-end values. This is included as part of the model limitations.

Conclusion

In conclusion, in squad team plays, team cooperation is vitally important. Giving one or more assists or revives knocked-down teammates increases the odds of winning a lot given all other variables constant. Naturally if the player chooses to kill teammates instead the odds of the group winning decreases. Given the player provides assists to teammates, they are able to survive longer and giving full strength, and the effect of each extra kills or knocked-down enemies on the winning odds is increased.

In addition, it might be better to have a more aggressive playing style compared to the "safe" way of hiding and avoiding conflicts, as a greater number of kills and knocked-down enemies compared to average increases the odds of winning. Practicing shooting accuracy in general will be a good idea as higher headshot accuracy compared to average player increases the odds of winning as well. In PUBG, the players expose their locations when shooting, therefore, it is good strategy to make shots count and worthwhile by having higher shooting accuracy. Also, as every additional weapon possessed compared to average increases the odds of winning, it is good game plan to gain as much advantages as possible by scavenging for a greater number of weapons.

When it comes to the use of medic items, the more boosting items taken compared to average increases the odds of winning. Compared to healing items which are only useful to recover health once suffered from damage and health below 75%, boosting items are the only way to regenerate health above 75% and able to provide movement speedups at the same time within certain time limit. In practice, taking boosts until a full boost bar is useful to get fully prepared particularly when it comes to the last few people.

Limitation

Regarding the data used for analysis, there is imbalanced outcome data variables given the nature of the survival game, which means most of the population is non-winning and only a few gets to the top. This results in limited ability for the model to predict the true positives (relatively low sensitivity) but high accuracy in predicting true negatives (high specificity). In addition, the binned residual plots show that there are some outliers the model fails to capture (e.g. *boostsc*). As attempted transformation on the predictors did not show improvements, there might exist some other relationships than logistics which is hard to infer from given information.

There was also every uneven available data for each category for some of the binary predictors, for example, there are mostly 0s in *teamKills* as most players tend not to kill teammates. This could make it hard to draw co-relations such as whether there was influence from one binary predictor on the odds of winning or simply such relationship was wrongly deducted from the biased data available.

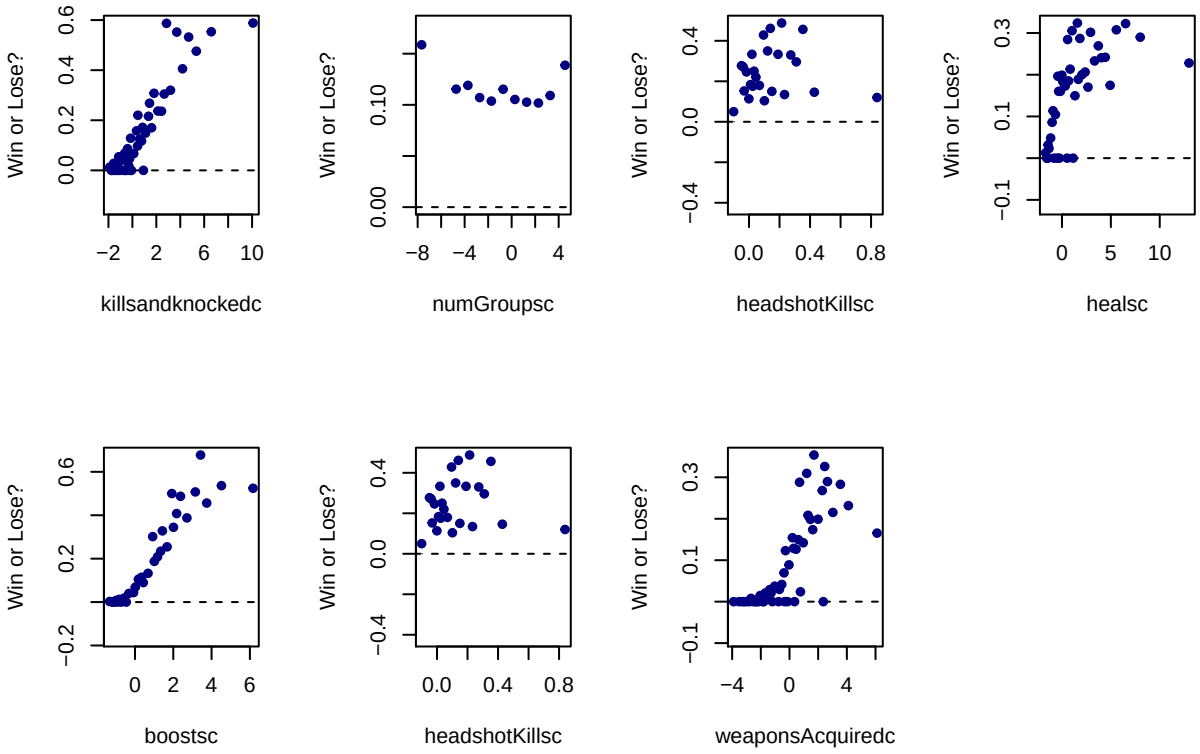
During data processing, by remapping continuous variables given into binary variables, information of the effect of each increase level was lost, hence the final model is unable to interpret the effect of having each additional unit of assists, revives, etc. on the odds of winning. Further, the dataset used was obtained by aggregation of each team by each match, which may not be fully representative statistics for an individual player in a squad team. Also, cheaters commonly exist in games and there might be other ways to cheat which are not accounted for. For example, players with 100% headshot kills were not removed in the dataset given the benefit of doubt that they are simply extremely good players. However, if they are actually cheaters, by aggregating based on each group of each match and taking the average statistics, cheaters' outstanding suspicious data will distort the end analysis result.

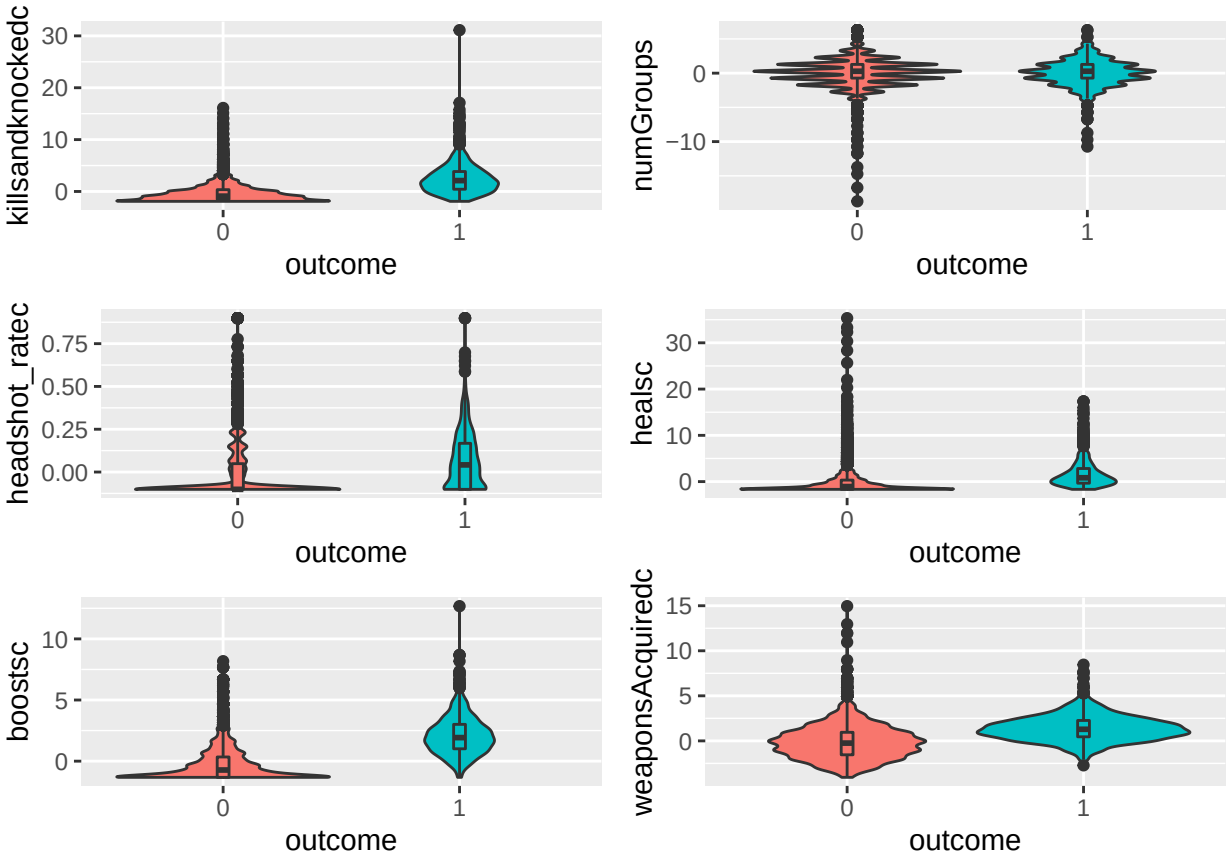
Finally, there are many nuances and external factors for each predictor present in the model in practice. For medical items, there are different types of healing and boosting items with each having different effects depending on both health bar and boost bar. Apart from number of kills and enemies knocked down, it would be interesting to look at different ways of giving damage, for example, the type of weapons (gun type, equipped or not, etc.).

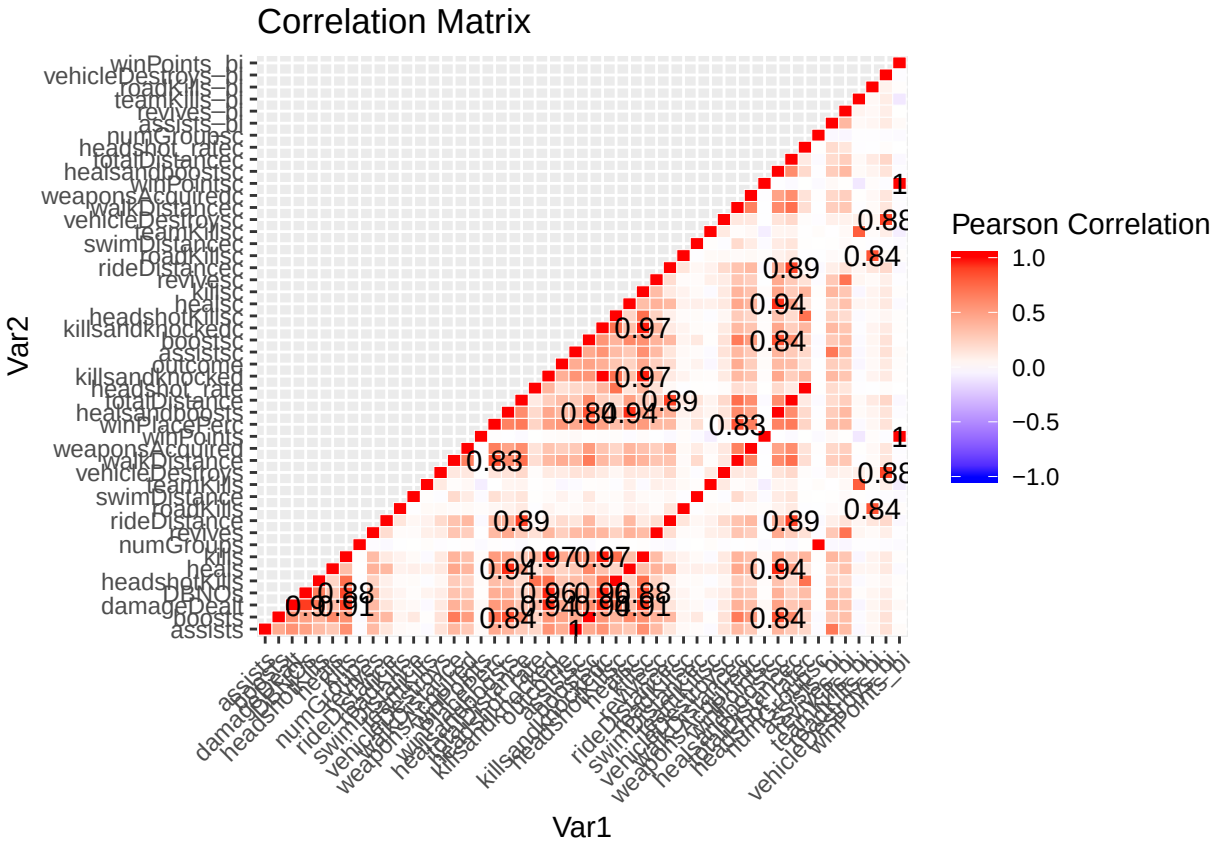
Appendix

link to github repository:

https://github.com/becky-yc/pubg

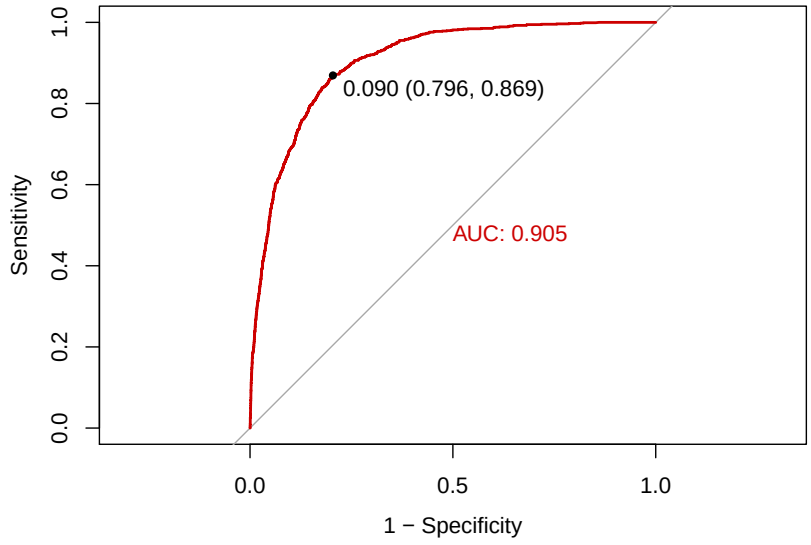



killsandknockedc
30
20
10
0
0
1
outcome
numGroups
0
−10
0
1
outcome
headshot_ratec
0.75
0.50
0.25
0.00
0
1
outcome
healsc
30
20
10
0
0
1
outcome
boostsc
10
5
0
0
1
outcome
weaponsAcquiredc
15
10
5
0
0
1
outcome




```
Accuracy
  0.9025

Sensitivity Specificity
  0.5834879   0.9410446
```



```
Call:
roc.default(response = df$outcome, predictor = predict(reg2,     type = "response", newdata = df), plot

Data: predict(reg2, type = "response", newdata = df) in 8922 controls (df$outcome 0) < 1078 cases (df$ou
Area under the curve: 0.905
```