Predicting Win Rates in Professional e-Sports via Supervised Learning

Advanced Topics in Data Science[†]

Karim Carroum

April 16, 2020

Abstract

This paper has two main purposes. First, I evaluate how the prediction accuracy and the amount of variables selected changes when it comes to predict winrates of League of Legends' professional players (using per-game season averages of ingame features) via LASSO (CV or chosen via the BIC criterion), Bayesian Model Averaging (BMA) and Random Forests in comparison to MLE. Random Forests and LASSO-BIC perform considerably better than the other procedures, and the latter selects a significantly lower amount of regressors than the rest. BMA slightly improves MLE's accuracy, even though its performance, together with that of LASSO-CV are far from decent. Second, I analyze which variables are more relevant in the prediction using a feature of Random Forests that evaluates the increase in MSE caused by dropping a certain variable from the regression. For both goals, the analysis is performed both at a pooled (i.e. set of all players) and position-wise (subsets of players by positions) level, and a temporal dimension is added to capture differences in those features between one season to another (namely Seasons 8 & 9). The more relevant variables for prediction purposes change between seasons for some positions, while for the others there are no large changes.

[†]Master of Research in Economics, Universitat Pompeu Fabra.

1 Introduction

E-Sports are a raising trend in the current world of entertainment. They gather hundreds of millions of users in terms of playerbase of the videogames and fanbase of their professional corresponding competitions. While the former have been highly popular for the past two decades, the latter was almost nonexistent, at least in relative terms to what it has become nowadays, up until the last decade. Such sharp increase in the popularity of e-Sports competitions has not been matched yet in terms of literature predicting the outcomes of those games (even though that does not mean that there has not been any literature dedicated to it at all). That probably may be due to the fact that, in contrast with already established traditional sports, the games featured in e-Sports are brand new and therefore their level of popularity tends to be highly volatile, so any research in a specific game has a high risk of becoming useless if that game does not succeed in maintaining a large playerbase.

Nevertheless, some of the games and their corresponding e-Sports league/s have been able to maintain a large (and in some cases still increasing) fanbase, and that makes research on them more appealing. The most clear case is that of the game with the both the highest fanbase and active playerbase in the world, also known as League of Legends (LoL). LoL is a game inside of the MOBA genre, which stands for "Multiplayer Online Battle Arena". In a few words, two teams formed by five players (each with their corresponding specific tasks, as in traditional sports) control avatars that are used to take down the base of the opponent team. This oversimplification hides the fact that, during the game, a lot of small tasks are targeted by both teams in order to increase their chances at winning the game. In fact, the richness and joy of the game lie on those tasks, and the end goal can be considered as just an immediate consequence of the rest. According to that, the per-game individual statistics gathered by eSports analysts focus those tasks.

As for traditional sports, for the case of LoL, predicting how well teams' performances given the abilities of their players is also of high value. This information is of high relevance for fans, players, coaches/analysts, team owners and sponsors, among other interested parties. Additionally, in a game like LoL, where its developers make modifications to its structure in a yearly[1] basis, it is also of high interest to see if the predictive ability (in terms of winrate prediction) of players' skills changes between seasons.

In line with the aforementioned reasons, this paper aims at predicting the winrates of LoL players given their abilities, as well as at analysing how their relevance in such prediction has changed in the last two seasons. Additionally, an inter-position analysis is carried out to capture the differences between seasons in the predictive relevance of the variables depending on the position of the player.

The methods used to achieve those goals are in line with the characterstics of the targeted data. First of all, I use MLE to obtain a benchmark providing the results of the more traditional approach. Then I employ more sophisticated methods that have advantages when the amount of regressors is considerably high (and even higher than the amount of observations in some cases), and that provide extra insights about the data, such as for instance the predictive relevance of the variables. Those methods are LASSO, Bayesian Model Averaging (BMA) and Random Forests. The advantages of those procedures is explained in the

[1] A patch is introduced every two weeks, but those patches are usually dedicated to balance changes of the avatars, leaving structural bigger changes to their yearly end-of-season patches.

"Methodology" section.

The paper is divided in the following sections. First, I provide a short Literature Review of previous work on predicting winrates or outcomes of games in eSports, mainly for the game "DOTA 2", which is quite similar in structure to League of Legends (in fact, the company developing them is the same). The following section explains the most relevant features and limitations of the Data. Then, I proceed to give some background regarding the Methodology of the paper, namely the theoretical base of LASSO, Bayesian Model Averaging and Random Forests while explaining why they are suitable for this paper's analysis. The next section displays and describes the main features of the predictive Results of those procedures, both at a pool and position-wise level. Lastly, I summarize the findings of the paper and discuss their relevance.

2 Prediction in eSports

Several predictive analysis regarding eSports have been performed. Hodge et al. (2017) predicts the winrate in the MOBA called "DOTA 2" employing in-game data and hero selection via the use of Random Forests and Logistic Regression, and evaluates the usage of non-professional training data to better train the machine learning algorithms since the availability of professional data is sometimes scarce. Wang (2018) also employs Logistic Regression and Decision Trees to identify different types of features affecting the results of LoL games. Yang et. al (2016) performs a two stage model to predict the winner of DOTA 2 games, where the first stage would gather information corresponding to pre-game factors, and the second stage would add real-time ingame data to evaluate the evolution of the winning odds during the matches.

Deolalikar and Peng (2015) employ unsupervised learning procedures to distinguish LoL players' clusters using different ingame statistics, which is further implemented in the prediction of the outcome of games. The authors argue that, due to the fact that because of a simultaneity bias regarding some of those statistics (i.e. once the first stages of the game have been played, the statistics in the following minutes are highly determined by those results), the analysis would be improved by adjusting those variables for their time-dependency. This simultaneity bias is analyzed for the case of DOTA 2 by Harboell et al. (2019) via the usage of instrumental variables and the application of a control function approach. Note though, that this is only relevant for the proper identification of causal-effects of ingame covariates on the winrates of the players, and so is not of high importance for the purpose of this paper, which is prediction accuracy and predictive relevance of variables.

3 Data

The data[2] contains 982 and 1092 observations corresponding to players in the main LoL regional and international leagues and tournaments of the competitive scene for seasons eight and nine. It contains information regarding a total of eighteen (the dependent variable and seventeen independent) individual statistics at a player level, which correspond to season averages of per-game aspects (e.g. how many avatars did a player

[2]The data has been retrieved from the website "Games of Legends eSports."

kill during a game, in average, in a specific season). [3] Those statistics cover a wide range of aspects of the game, which vary significantly and so the magnitude of their contribution to the prediction of the winrate of a player can be considerably different.[4] At a glance, before the inclusion of interactions and nonlinear terms (which increases the range of regressors to an extent that makes any attempt of the following classification unfeasible), the independent variables of the analysis can be divided by tasks into the following groups:

- KDA, Average Kills, Solo Kills, Average Assists and Average Deaths. Degree of implication in kills in the game (whether they are kills on the opossing team's avatars or on the player's avatar). Killing or assisting in killing an opponent avatar helps in winning games and thus dying is detrimental.
- GD15, CSD15, XPD15, First Blood% and First Blood Victim. Relative advantage/disadvantage of the player compared to the enemy team's player in the same position in the first stages of the game. Positive and/or higher values in the first three variables are related with advantages and its magnitudes (and the converse for negative values), while the last variable gathers a measure regarding deaths and so is detrimental.
- Damage% and Average Damage per minute. Implication of the player in terms of damage output in the skirmishes against the opponents. Intuitively enough, damage helps in killing (between other tasks) avatars and so increases the chances of winning games.
- CSM and GPM. Gold (i.e. individual resources) generation by the player. Gold is used to improve the avatars stats and thus enhances its winrate.
- Average WCPM and Average VWPM. Degree of implication of the player in giving vision (beneficial for strategic purposes) or destroying vision from the enemy team.
- Pentakills. Very rare feature in which a single player annihilates all the enemy team's avatars.

In addition, the database provides the position (”Top”, ”Jungle”, ”Mid”, ”Adc” or ”Support”) of the players, which will allow to carry out the inter-position analysis. The subsets of players by position have near to 200 observations, and in some cases it is lower than the amount of variables employed in the regression.

In Figure 1 we can see that, to some extent, the relation between KDA, which may arguably be the most important predictor of winrate, and winrate itself is likely to be non-linear. Additionally, the lack of previous literature predicting winrates in LoL with the set of regressors that I am using makes it compelling to add nonlinear terms for the rest of variables, which I proceed to add up to the third order. Furthermore, to obtain a more complete set of regressors I also add interactions between the first order values of the regressors. Those additions account for a total of 176 variables, and so the total amount ascends to 193.

Since players in teams with extremely negative dynamics (i.e. 0% winrate or very close to zero) might be exposed to significantly different team environments than those of the average teams, I drop them out of the analysis. That yields datasets with 896 and 932 players for seasons eight and nine.

[3] Note that the database does not provide per game statistics but averages, which clearly shapes the approach of the analysis.

[4] E.g. $KDA = \frac{Kill+Assists}{Deaths}$ is arguably the most representative index of player performance, while the player's ability to achieve Penta Kills is a very marginal, almost never achieved feature, and so is not that likely to determine the players' winrate percentages.

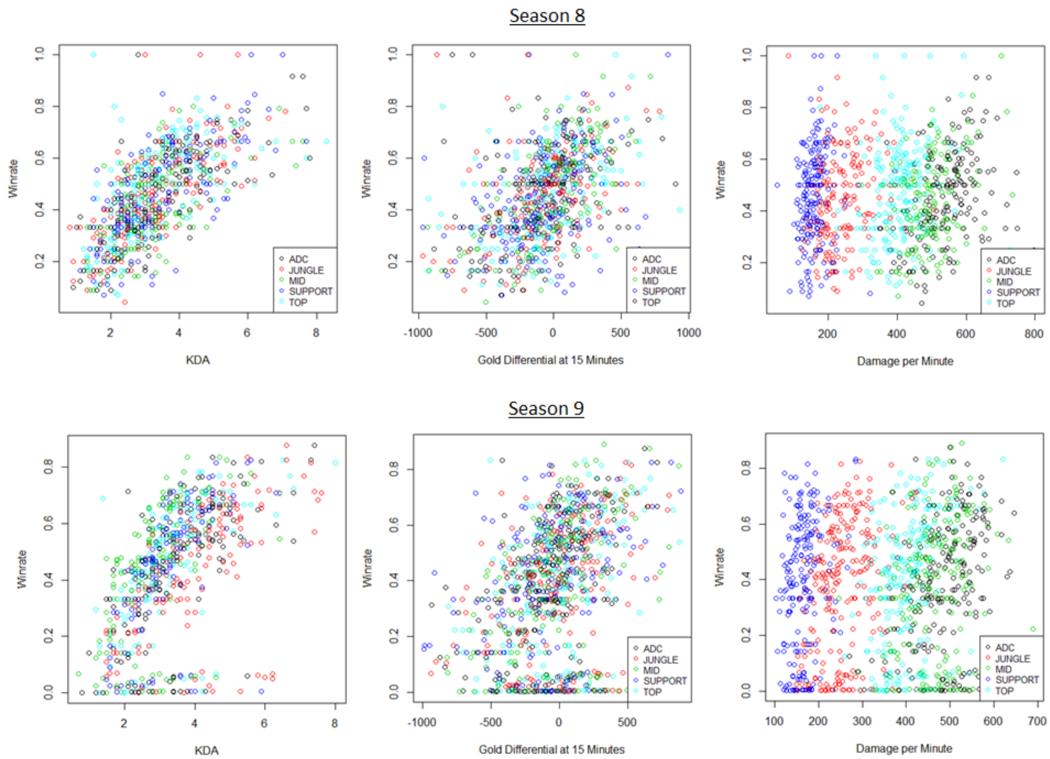

Figure 1: Winrate vs KDA, Gold Differential at 15 Minutes and Damage per Minute for Seasons 8 & 9 Season 9.

An important limitation faced by this analysis is that the database does not include the team of the player. It seems reasonable to think that the inclusion of that variable would potentially allow to extract, up to some extent, the difference in the winrate of a player (compared to similar players) caused by the abilities of his teammates rather than the player's.

Figures $1-2$ give a glimpse of the variables of the topic, concretely of the relation between some of the regressors and the winrate for the two analysed seasons, both at the pool and position-wise levels. The direction of the relations seem to be in line with that described above (i.e. with the one that I expect), while obviously some variables appear to be correlated with winrate in a higher degree than others. We can see that for some variables, such as damage and gold per minute, their magnitude (scale) is considerably different between positions, while for others there is no substantial disparity. Additionally, the same happens regarding the degree of correlation (slope) of some variables between positions, such as a higher slope of gold per minute vs winrate for "Adc" players than that for "Support" players, which may mean that for players of the latter position it is not as important (in terms of increasing their winrates) to generate resources than for the former. Another important highlight of the relation of the regressors with winrate, even though it is not clearly showcased in those figures, is that its magnitude and/or slope may vary from one season to the other for some variables.

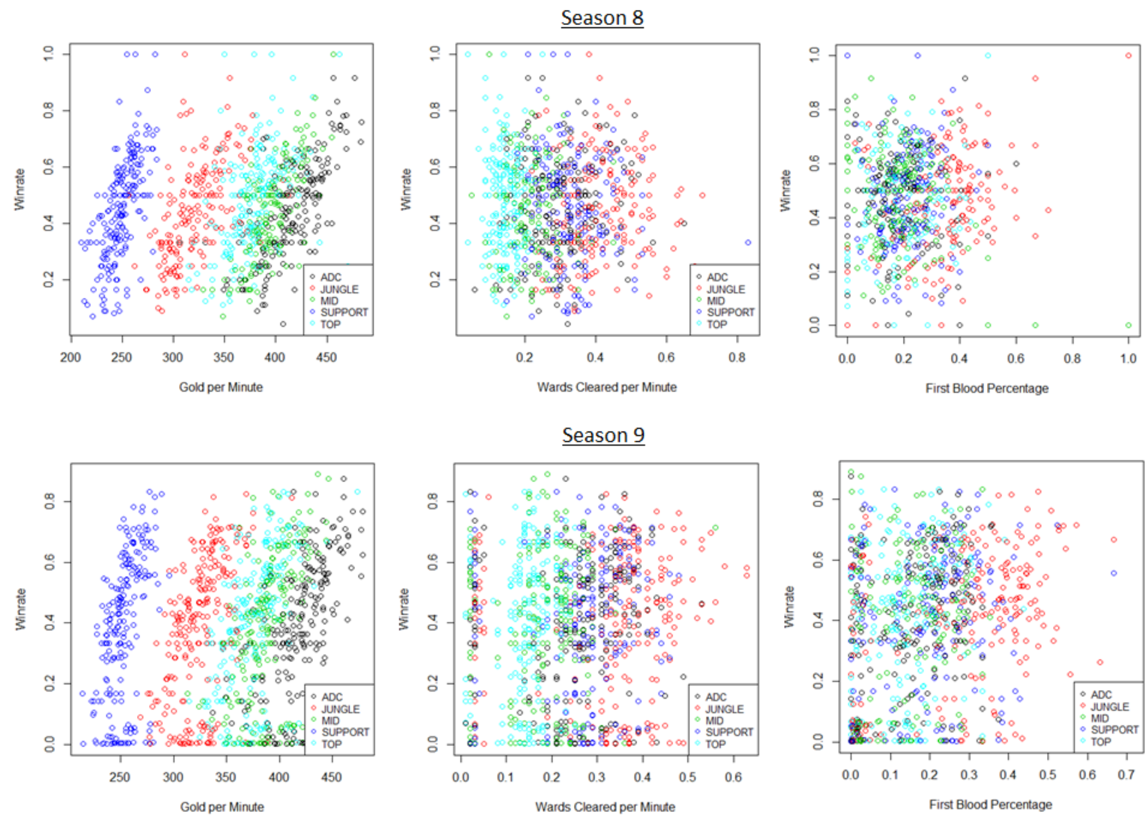

Figure 2: Winrate vs Gold per Minute, Wards Cleared per Minute and First Blood Percentage for Seasons 8 & 9.

4 Methodology

The following section summarizes the procedures used in the predictive analysis. Note that, as a benchmark, MLE regressions are performed (via 10-fold cross-validations), to then see what are the contributions of the proposed methods. Nevertheless, I do not explain how MLE works since its relevant aspects for this paper are already mentioned in the other procedures, which in turn are of higher relevance for this paper. For details regarding the code used to implement such methods (in R) check section 1 of the appendix.

4.1 LASSO

The motivation behind LASSO is improving the predictability and interpretability of regression models. Additionally, it allows dealing with regressions where $p >> n$, where the traditional method of MLE breaks down due to calculation issues. This is done by adding a penalization (a.k.a. L1 penalty) in the process of estimation of the parameters that shrinks towards zero the less relevant variables when it comes to prediction purposes. Thus, the regression is run over a subset of covariates rather than over the whole set, which is specially useful in situations with a considerably high amount of regressors. I hence employ LASSO to deal with $p >> n$ (which happens with some subsets of player's positions) as well as to improve the predictability of the regression.[5]

Formally, LASSO's estimates are given by the following optimization:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} ||y - X\beta||_2^2 + \lambda ||\beta||_1 \right\}$$

where λ is the parameter adjusting for the L1 penalty. Namely, if we decrease λ then we decrease the amount of bias (which is zero if we set lambda to zero because then we obtain the MLE optimization problem) since we incur in a higher degree of shrinkage, but then we may lose the aforementioned benefits of such procedure. Therefore, choosing λ must be thought meticulously. In my analysis this is done via the so called K folds cross-validation (or leave-one-out validation), where K is the amount of folds in which we split the data before training it in $K-1$ of those folds and validating it in the remaining fold, and this is done K times. Then, we choose λ such that the prediction error is minimized. Alternatively, I also perform model selection via the BIC criterion, namely choosing the model that displays the highest BIC, to see if the results considerably change from those of the base LASSO scenario.

4.2 Bayesian Model Averaging

Through the use of Bayes factors, where priors $P(M_j)$ over the models are incorporated, this procedure quantifies the likelihood of different possible models. Then, the effect of a covariate is quantified through the weighted average of its estimations for each model, weighted by the posterior probability of each model $P(M_j|y)$. Formally, from Bayes' Theorem we have that $P(M_j|y) \propto L(M_j)P(M_j)$ where $L(M_j)$ is the marginal likelihood of M_j. The latter results from the likelihood integration with a prior over the parameters of M_j denoted by $p(\theta_j|M_j)$, and so

$$L_y(M_j) = \int p(y|\theta_j, M_j)p(\theta_j, M_j)d\theta_j$$

I employ beta priors $(1,1)$ for the models, which correspond to complete model uncertainty, and the standard Zellner prior on coefficients $\tau = 0.348$, which in turn means that it is assumed that the variance-covariance structure of the coefficients is "broadly in line with that of the data X_γ" (Zeugner, 2011)[6].

One of the advantages of BMA is that, instead of recurring to a "reject, do not reject" decision it reaches a much more flexible result. Additionally, BMA is more robust to model misspecification than other standard regression procedures (Hinne et al., 2019). This is specially useful for this paper's predictive purposes since p is considerably high, and also because there are a large amount of interactions (which are by definition more complex and thus its analysis also is). The fact that there does not exist a vast literature predicting winrates in eSports makes BMA more useful sine it accentuates the issues in which we can incur regarding the selection of the correct model selection. Another advantage worth pointing out is that BMA results in optimal predictions when we cannot identify consistently the model (Hoeting et al. 1999).

[5] LASSO also improves the interpretation.

[6] For more about how to choose the prior on coefficients, see Zeugner (2011) - Section 1.2

4.3 Random Forests

For continuous regressions, decision trees try to form nodes that contain a large proportion of samples from a specific interval by finding ranges of values in the features that partition the data in such intervals. This paper's analysis uses an R package[7] based on Breiman (2001) Random Forest algorithm.

As formalized in Segal (2004), "a Random Forest (RF) is a collection of tree predictors $h(x; \theta_k), k = 1, ..., K$ where x is the observed input vector of length p and θ_k are independent and identically distributed random vectors. The RF prediction is the unweighted average over the collection $h(x) = \frac{1}{K}\sum_{k=1}^{K} h(x; \theta_k)$." The predictive power of RFs depends on the correlation between any pair of trees and the strength (i.e. how accurate the "classifiers" are) of each tree. In broad terms, the interplay of those two aspects yields the following paradigm: a RF will increase in predictive power when the ratio of the correlation over the square of the strength is minimized.[8] Residual correlation is minimized by a randomization using a bootsrap sample from the training data to grow each tree, and specifing $m << p$ covariates at each node of every tree to then select the best partition of that node according on them.

As a default, we set $m = \frac{p}{3}$, which is an accepted result in the literature,[9] and is the approach that I follow in this paper. If the amount of variables m used per tree is reduced, both correlation and strength also decrease. When it comes to the amount of trees of the forest, in most of the cases it does not drastically change the prediction power of the model. Nevertheless, Segal (2004) finds that, for some cases in which the chosen size of trees causes an overfit, gains can be obtained by limiting the number and/or the size of nodes for which splitting is allowed.

RFs' most notable feature that makes it interesting for this paper's prediction is that they provide useful measures (specially for regression purposes) such as variable importance, which quantifies how much the accuracy of the predictions decreases when a certain variable is excluded. That is of special relevance for prediction in cases with large p since it eases the predictability analysis of individual variables providing a subset of the most relevant ones. Lastly, another advantage of RFs is that they are relatively robust to extreme observations.

5 Results

The results of this paper are divided into two blocks. The first one provides information regarding the predictive ability of the regressions and the amount of regressors selected by each method[10]. The second section analyses variables' relevance for the RF case. The analysis is done both at a pooled and per-position level, and over seasons.

[7] randomForest. Available at https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[8] Too see more about such interplay, see Breiman (2001) - Section 2.

[9] E.g. "The Elements of Statistical Learning" - Friedman et al 2001, chap. 15. Further explorations can be done running loops with different values m in order to see which is the optimal value in terms of prediction purposes.

[10] Note that, due to the nature of the Random Forest's methodology, no amount of selected regressors is provided for it.

5.1 Predictive Ability and Amount of Selected Regressors

Regarding prediction accuracy, it is important to point out first that the more intuitive indicator out of MSE (Mean Squared Error) and MAE (Mean Absolute Error) is the latter since the dependent variable is a percentage, and thus when the error is squared its value is reduced, while this does not happen to MAE. Due to this, I only use the MAE to analyse predictive ability. Additionally, I use the out of sample MAE in such analysis since it gives a more realistic expectation of how well the prediction would work on new data. As a technical remark, note that for the subsets of players in the position "Adc" and "Support" we have that $p > n$ and thus MLE breaks down; in line with this, the in-sample predictions obtained by such regression yield zero prediction errors, and so in order to gather information regarding this behavior of the MLE, I do not erase that from the tables. Results for the pooled sets are displayed in Table 1 and those for the subsets of positions in Tables $2-6$.

The procedure that yields the best out of sample predictions is the Random Forest, with an average error of around 8% and 14% in S8 and S9 for the pooled cases. Since, in average, the winrate is around 50% for the population[11], the mean error rate for the pooled cases in terms of the average of the independent variable is of around $16\% - 28\%$. Position-wise RF has a greater predictive ability too, with prediction errors of around $8\% - 12\%$ (and thus of around 16% and 24% in terms of the independent variable). In general, LASSO-BIC's out of sample predictions are just slightly less accurate. BMA provides worse predictions than the latter approaches but still outperforms MLE. LASSO-CV turns out to perform worse than MLE in general, even though it still has the advantage that it can yield regressions properly in the cases where MLE cannot due to $p > n$.

	Coef.Sel.S8	Coef.Sel.S9	MAE (In Sample) S8	MAE (In Sample) S9	MAE (Out of Sample) S8	MAE(Out of Sample) S9
MLE-CV	193	193	0.0614	0.1197	0.1953	0.2489
LASSO-CV	53	36	0.0772	0.1475	0.1646	0.2782
LASSO-BIC	20	17	0.0757	0.1498	0.0734	0.1538
Bayesian MA	72	74	0.0729	0.1377	0.1907	0.2377
Random Forest	-	-	0.0333	0.0542	0.0792	0.1415

Table 1: General features of the pool of players in seasons 8 and 9 for each approach.

It is interesting to note that winrates are predicted considerably more poorly (out-of-sample wise) for season 9 than for season 8 for the pool of players by each of the procedures. This suggests that the regressors became less relevant to predict winrates from S8 to S9. A possible explanation for this might be that after some changes introduced in LoL in S9, the output of the games was decided more largely by factors not gathered by the regressors. This would showcase a potential need for a relatively frequent update in the selection of regressors used to predict winrates in LoL, which can also be a general feature in other eSports since most of them change the structure of the game in a very frequent fashion.

Nevertheless, position-wise we can see that S8's winrates are predicted worse than S9's by some of the procedures while for the other procedures the opposite happens. Specifically, for all the positions but for "Supports", RF and LASSO-BIC yield better intra-position predictions for S8 while BMA and LASSO-CV perform better, in general, for S9. Overall, the results do not showcase that the degree of predictability of winrates via the employed set of regressors changed significantly for those positions from one season to the

[11] Not exactly 50% since, as explained in the "Data" section, some observations are excluded from the original set of players.

other. In spite of that, for S9 the predictability of the winrate of Supports is considerably worse, which suggests that, as argued above the set of predictive regressors could have been more relevant in S8 than in S9 for Supports, or that more specifically the role of those players in S9 became less relevant from one season to the other and so their performance did not affect that much their winrates.[12]

When it comes to the amount of regressors selected, the criteria is that the estimate is not exactly zero, without taking into account confidence intervals to perform inference, since that is not of high relevance for the purposes of this paper. The results display that LASSO-CV and LASSO-BIC select a considerably lower amount of regressors than MLE (which selects p) and BMA. Additionally, LASSO-BIC selects less regressors than LASSO-CV for almost all the sets (pools) and subsets (positions) analyzed. Additionally, note that for the subsets of positions the amount of selected regressors is considerably lower than for the pool of players when it comes to LASSO-CV and LASSO-BIC, while it is higher, in general, for BMA. Out of those three methods, the LASSO-BIC outperforms the other two when it comes to predict winrates of LoL players since it achieves better predictions while employing a significantly lower amount of regressors. Nevertheless, recall that LASSO-BIC's predictive ability is still lower than that of RF.

	Coef.Sel.S8	Coef.Sel.S9	MAE (In Sample) S8	MAE (In Sample) S9	MAE (Out of Sample) S8	MAE(Out of Sample) S9
MLE-CV	193	193	0.0162	0.0175	0.1747	0.1980
LASSO-CV	28	13	0.8206	0.1556	0.3810	0.1086
LASSO-BIC	16	10	0.0675	0.1104	0.0847	0.1114
Bayesian MA	109	103	0.0659	0.0990	0.1748	0.1710
Random Forest	-	-	0.0325	0.0477	0.0853	0.1037

Table 2: General features of players in the position "Top" in seasons 8 and 9 for each approach.

	Coef.Sel.S8	Coef.Sel.S9	MAE (In Sample) S8	MAE (In Sample) S9	MAE (Out of Sample) S8	MAE(Out of Sample) S9
MLE-CV	193	193	0.0072	0.0223	0.2091	0.1857
LASSO-CV	23	15	0.5600	0.4751	0.3528	0.5552
LASSO-BIC	14	17	0.0715	0.0861	0.0899	0.1068
Bayesian MA	79	104	0.0708	0.0861	0.1963	0.1774
Random Forest	-	-	0.0362	0.0376	0.0843	0.1043

Table 3: General features of players in the position "Jungle" in seasons 8 and 9 for each approach.

	Coef.Sel.S8	Coef.Sel.S9	MAE (In Sample) S8	MAE (In Sample) S9	MAE (Out of Sample) S8	MAE(Out of Sample) S9
MLE-CV	193	193	0.0000	0.0000	0.1923	0.2022
LASSO-CV	29	22	0.3129	0.3278	0.7054	0.3540
LASSO-BIC	16	5	0.0061	0.1302	0.0711	0.1244
Bayesian MA	63	84	0.0622	0.1130	0.1762	0.1829
Random Forest	-	-	0.0304	0.0490	0.0704	0.1232

Table 4: General features of players in the position "Mid" in seasons 8 and 9 for each approach.

[12] I.e. some of the changes introduced in the game in S9 may have had decreased the relevance of Support players, just like if in basketball the 3-point line was removed then the importance of the specialized long-range shooters would decrease.

	Coef.Sel.S8	Coef.Sel.S9	MAE (In Sample) S8	MAE (In Sample) S9	MAE (Out of Sample) S8	MAE(Out of Sample) S9
MLE-CV	193	193	0.0000	0.0000	0.2032	0.1879
LASSO-CV	12	13	0.1061	0.1038	0.1258	0.1170
LASSO-BIC	9	13	0.0645	0.0985	0.0791	0.1019
Bayesian MA	91	84	0.0592	0.0890	0.1907	0.1697
Random Forest	-	-	0.0275	0.0438	0.0794	0.1009

Table 5: General features of players in the position "Adc" in seasons 8 and 9 for each approach.

	Coef.Sel.S8	Coef.Sel.S9	MAE (In Sample) S8	MAE (In Sample) S9	MAE (Out of Sample) S8	MAE(Out of Sample) S9
MLE-CV	193	193	0.0194	0.0163	0.2267	0.1911
LASSO-CV	27	11	0.6079	0.7606	0.1006	0.7431
LASSO-BIC	14	7	0.0805	0.1060	0.0874	0.1106
Bayesian MA	92	88	0.2014	0.4490	0.2852	0.4792
Random Forest	-	-	0.0352	0.0399	0.0953	0.1108

Table 6: General features of players in the position "Support" in seasons 8 and 9 for each approach.

5.2 Variable Importance

Variable importance describes by how much the predictive ability of the model decreases when we drop a certain variable. Concretely, the measure provided by the R package "RandomForest" calculates the percentage of increase of the MSE if a such variable is excluded.

Pool of players

The results showcase that the groups of variables that have more relevance in predicting winrates are those regarding the degree of implication in kills, GPM (a measure on the generation of resources), and some variables corresponding to the relative advantages and disadvantages in the early stages of the game (in spite of CSD15 and XPD15). On the other hand, variables regarding either damage contribution or the degree of implication in giving and/or destroying vision are not that relevant for predicting winrates. The more relevant variables (for the pooled case) for each season are displayed in Figure 3.

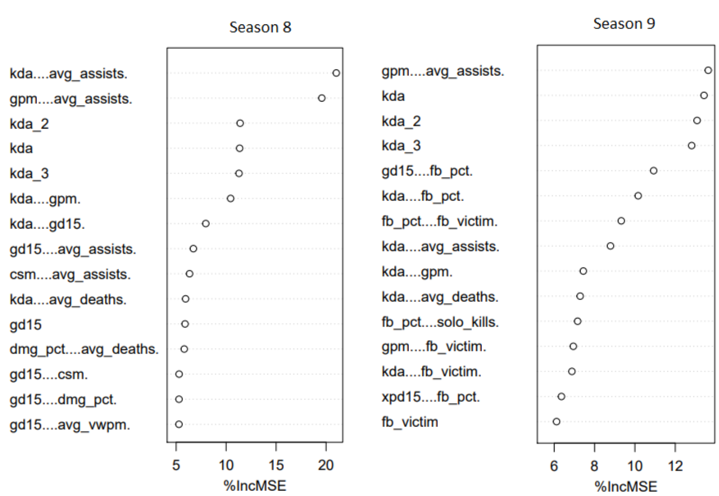

Figure 3: Top regressors in terms of variable relevance (Pooled case) for Season 8 and Season 9.

Some of the most relevant (in terms of predictability) variables changed from S8 to S9. Concretely, while the interaction of KDA with Assists was the most relevant variable in S8 with an increase in MSE when excluded (from now on "variable score") of around 20%, it became the eigth more important with a variable score of just 9% in S9. The interactions of KDA with GPM and KDA with GD15 also displayed substantial decreases of their importance. Minor decreases in relevance were suffered by a set of interactions of GD15 with a few other variables. Several variables though gained a significant amount of relevance. In broad terms, interactions regarding either FB Percentage or FB Victim gained importance, with the most notable cases being those of the interactions of FB Percentage with GD15, KDA and FB Victim.

A possible interpretation of this is that, due to changes in the structure of the game from one season to the other, killing or assisting in killing enemy avatars while dying the least possible of times (gathered in KDA) became less important to predict winrates, while being directly involved in a positive manner (i.e. being the player killing or assisting in killing the killed avatar instead of controlling the avatar getting killed) in the first kill of the game or being the first player to die was more decisive.

Players by position

When it comes to the position-wise analysis, over the two seasons, the general result is that some interactions regarding either KDA or GPM (or the interaction between the two) were the more relevant variables. When it comes to the variable relevance's scores, they tended to decrease for the majority of variables from S8 to S9. Some of those results are in contrast to those regarding the pool of players, since for the latter GPM was not as important as for the subsets of positions, and also FB (First Blood) Percentage and FB Victim became more important in S9 for the pool while they did not for the subsets of positions (with the exception of FB Percentage's increase in relevance in S9 for "Top" players). The more relevant variables (position-wise) for each season are displayed in Figures $4-8$ in the Appendix (subsection 2).

Specifically, for "Top" players, KDA's interactions were more important than those of GPM for both seasons and their relative orders did not vary significantly. Additionally, Average Assists were also important for prediction in S8 but lost relevance in S9, while the converse happened for FB Percentage. Regarding "Jungle" players, the relative order (in terms of relevance) of the variables remained similar between seasons but their scores decreased, specially that of FB Percentage. The more relevant variables were also KDA's and GPM's interactions in both seasons. When it comes to "Mid" players, KDA's and GPM's interactions were the most relevant in S8 but for S9 both lost importance while Damage Percentage's interactions with some variables became the better predictors, specially those with Average Deaths and GD15. Variable relevance's scores were lower for almost all variables overall in S9. For "Adc" players, KDA's and GPM's interactions were the most relevant for both seasons, while the interaction of Average DPM with KDA considerably gained relevance in S9 and the opposite happened with that of GPM with Average Assists. Variable relevance's scores were also lower in S9. Lastly, regarding "Support" players, KDA's interactions were the most relevant for S8 while those of GPM were for S9. Variable relevance's scores did not vary significantly for the rest of variables between seasons.

This suggests that the changes introduced in the structure of the game between seasons altered the relative importance of some aspects of the game for predicting winrates. The manner in which this happened varied position-wise, with some positions keeping the same aspects as the most relevant ones (for Jungle and Adc players), while others displayed notables changes such as the irruption of FB Percentage (for Top players) and Damage Percentage (for Mid players) as important predictors, or significant changes in the order of the main predictors while keeping the elements of such set almost unchanged (for Support players).

An adventurous interpretation of this would be that, while the roles of Jungle and Adc players did not vary considerably from S8 to S9, it was not the same case for the other three positions, which were altered towards aspects of the game concerning the aforementioned variables that gained relevance in predicting winrates for such positions.

6 Conclusion and Discussion

LASSO-BIC, Bayesian Model Averaging and Random Forest procedures perform better (based on out of sample accuracy) than MLE when it comes to the prediction of winrates for LoL players during seasons 8 and 9. This holds for the pool of players and by positions. LASSO-CV's relative performance to MLE is worse for the pool and some positions. Out of those approaches, Random Forest is the one providing the most accurate predictions, followed closely by LASSO-BIC, which provides considerably better predictions than those of BMA and LASSO-CV (which behave poorly) while using a considerably lower amount of regressors (reduced from $p = 193$ from MLE to not more than 20 for LASSO-BIC). The mean absolute errors achieved by the best regression (Random Forest) are of the order of 8% and 14% for the pool while around $8\% - 12\%$ for the different positions. Similar (though slightly worse) results are displayed by LASSO-BIC. Considering that the average winrate is of around 50%, this suggests that Random Forests and LASSO-BIC perform adequately for the prediction of such variable. It is interesting to highlight that both methods achieve better predictions for S8 than for S9, for both the pool of players and by positions.

Variable relevance in prediction is deeply analyzed through the variable importance statistic provided by Random Forests. The behavior of the changes in the relative importance of the variables for prediction suggests an interesting narrative. It suggests that, potentially, due to changes in the structure of the game introduced between seasons 8 and 9, the rol of the players changed for some positions, or in other words there was a change in the set of aspects of the game that had more impact on how those players, in turn, contributed on winning games. This did not occur equally between positions: for some, the aspects that helped more in predicting their winrates did not change much from one season to another, while for others the opposite happened. Such phenomenon highlights a need of a relatively frequent updating of the set of predictors in LoL according to the magnitude of the changes to the structure of the game during the analysed period. It would be interesting to check if this is the case for other eSports too, since the usual strategy followed by videogame developers is changing them in a frequent basis.

The fact that some variables gathering information regarding the first stages of the game are considerably relevant is of special importance for research in LoL (and to some extent in eSports) focusing on the causal channels of player performance and their winrate. This is because they may suffer less of simultaneity bias, which is a concern in regressions regarding outcomes in eSports, since it has been shown that earlier ingame advantages in another videogame (very similar to LoL) have effects on some regressors that are relevant in determining the outcome of the games (Harboell et. al, 2019), and thus the variables that represent just those advantages should not suffer that much from this issue. Nevertheless, the other relevant groups of regressors may still considerably suffer from this bias and so further research on causal effects should still put a special focus on this phenomena.

Lastly, the absence in the database of the teams of the players constitutes a clear path of improvement for the analysis, since it would allow to isolate, up to some extent, the effect of the teammates' performances on the winrate of a given player, which arguably is a concern for any type of team sport or eSport.

References

[1] Breiman, L. (2001). ”Random Forests”. Machine Learning, 45(1), 5-32.

[2] Harboell, A. et al. (2019). ”Resolving Simultaneity Bias: Using Features to Estimate Causal Effects in Competitive Games”. Retrieved December, 2019 from http://ieee-cog.org/2019/papers/paper$_1$29.*pdf*

[3] Hastie, T., Tibshirani, R., Friedman, J. (2001). *The Elements of Statistical Learning.* Springer.

[4] Hinne, M. et al. (2019). ”A Conceptual Introduction to Bayesian Model Averaging”. Retrieved December, 2019 from https://psyarxiv.com/wgb64/

[5] Hodge, J. et al. (2017). ”Win Prediction in Esports : Mixed-Rank Match Prediction in Multi-player Online Battle Arena Games. Working Paper”. *White Rose Research Online.* Retrieved December, 2019 from http://eprints.whiterose.ac.uk/124333/1/ArXiv$_1$711.06498.*pdf*

[6] Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. (1999). ”Bayesian model averaging: A tutorial”. Statistical Science, 14(4), 382–401.

[7] Ong, H. Y., Deolalikar, S., & Peng, M. (2015). ”Player Behavior and Optimal Team Composition for Online Multiplayer Games”. Retrieved December, 2019 from https://arxiv.org/abs/1503.02230

[8] Segal, M. (2004). ”Machine Learning Benchmarks and Random Forest Regression”. Technical Report, Center for Bioinformatics & Molecular Biostatistics, University of California, San Francisco.

[9] Wang, T. (2018). ”Predictive Analysis on eSports Games - A case Study of Legends (LOL) eSports Tournaments”. Retrieved December, 2019 from https://cdr.lib.unc.edu/downloads/7h149t62f

[10] Yang, Y., Qin, T., & Lei, Y. H. (2016). ”Real-time eSports Match Result Prediction.” Retrieved December, 2019 from https://arxiv.org/abs/1701.03162

Apendix

1. R code

The following code has been used individually for all the different subsets analyzed (i.e. for the pooled set of players for seasons eight and nine and for the subsets of players by position and by season). In broad terms, I have used the following code to implement the regressions, to obtain measures of out of sample prediction errors, and to obtain measures regarding variable relevance. This code does not include some results that I have coded in R but omitted in the paper such as MSE (both in and out of sample) and the coefficient estimates. The reason for this is that the former is pretty much irrelevant for this paper given that MAE is more representative (as I already have argued), while the inclusion of the latter is basically unfeasible due to the size of p and the amount of different sets and subsets being analysed, even though its analysis could bring relevant insights. If the reader is interested in those results, they can be found in the complete R code of this paper.

Additionally, the cleaning process of the data, together with the gathering of the results, plus the creation of plots are also omitted in the following code, but can be found in the complete R code.

LASSO-CV

```
#Mylars function computes the cross-validation-optimal regression
#coefficients for lasso by minimazing the cv error
fit.lasso.8= mylars(data_xint, data_yint, k=10, use.Gram=TRUE,
normalize=TRUE)
sel.lasso.8= fit.lasso.8$coef!=0
predicted_lasso_8 = t(as.matrix(fit.lasso.8$coefficients))
%*%t(as.matrix(data_xint))
prediction_error_lasso_8 = predicted_lasso_8-data_yint
mse_lasso_mylars_8 = sum(prediction_error_lasso_8^2)/length(data_yint)
mean_error_lasso_mylars_8 = mean(sqrt(prediction_error_lasso_8^2))

#Partition of the dataset into a training set and a validation set
#to obtain out of sample MSE and MAE
train <- sample(nrow(data_int), 0.7*nrow(data_int), replace = FALSE)
TrainSet_xint <- data_int[train,4:196]
TrainSet_yint <- data_yint[train]
ValidSet_xint <- data_int[-train,4:196]
ValidSet_yint <- data_yint[-train]

fit.lasso.mylars.test.8 = mylars(TrainSet_xint,TrainSet_yint,k=10,
use.Gram=TRUE, normalize=TRUE)
predicted_lasso_validset_8 = t(as.matrix(fit.lasso.mylars.test.8$coefficients))
%*%t(as.matrix(ValidSet_xint))
```

```
mse_lasso_mylars_out_8 = mean((predicted_lasso_validset_8-
ValidSet_yint))^2
mean_error_lasso_mylars_out_8 = mean(sqrt((predicted_lasso_validset_8-
ValidSet_yint)^2))
```

LASSO-BIC

The lasso.bic function uses the function a loop that involves the function glmnet, which yields a regularization path that is computed for the lasso penalty at a grid of values for the regularization parameter lambda, and then chooses the model with the best BIC.

```
fit.lassobic.8= lasso.bic(y=data_yint,x=data_xint,intercept=TRUE)
sel.lassobic.8 = (fit.lassobic.8$coef != 0)
predicted_lassobic_8 = fit.lassobic.8$ypred
prediction_error_lassobic_8 = predicted_lassobic_8-data_yint
mse_lassobic_8 = sum(prediction_error_lassobic_8^2)/length(data_yint)
mean_error_lassobic_8 = mean(sqrt(prediction_error_lassobic_8^2))
fit.lassobic.test.8 = lasso.bic(y=TrainSet_yint,x=TrainSet_xint,
intercept=TRUE)
intercept_lassobic_test_8 = fit.lassobic.test.8$coef[1]
predicted_lassobic_validset_8 = t(as.matrix(fit.lassobic.test.8$coef))[,2:194]
%*%t(as.matrix(ValidSet_xint))+intercept_lassobic_test_8
mse_lassobic_out_8 = mean((predicted_lassobic_validset_8-ValidSet_yint))^2
mean_error_lassobic_out_8 = mean(sqrt((predicted_lassobic_validset_8-
ValidSet_yint)^2))
```

Bayesian Model Selection

```
pc= momprior(tau = 0.348)  # Prior on coefficients: NLP MOM
pm2 = modelbbprior(1,1)
ms.8= modelSelection(data_yint, data_xint, priorCoef = pc, priorDelta = pm2)
head(postProb(ms.8))
coef(ms.8)
pred_ms = predict(ms.8)[,1]
pred_ms = t(as.matrix(coef(ms.8)[2:194,1]))%*%t(as.matrix(data_xint))
+coef(ms.8)[1,1]
mse_ms_8 = mean((pred_ms-data_yint)^2)
mean_error_ms_8 = mean(sqrt((pred_ms-data_yint)^2))
predValid_ms = t(as.matrix(coef(ms.8)[2:194,1]))%*%t(as.matrix(TrainSet_xint))
+coef(ms.8)[1,1]
mse_ms_out_8 = mean((predValid_ms-ValidSet_yint)^2)
mean_error_ms_out_8 = mean(sqrt((predValid_ms-ValidSet_yint)^2))
```

Random Forests

```
fit.rf.8 <- randomForest(TrainSet_yint ~ ., data = TrainSet_xint, mtry = 64,
importance = TRUE)
predTrain <- predict(fit.rf.8, TrainSet_xint, type = "class")
mse_rf_8 = mean((predTrain-TrainSet_yint)^2)
mean_error_rf_8 = mean(sqrt((predTrain-TrainSet_yint)^2))
predValid <- predict(fit.rf.8, ValidSet_xint, type = "class")
plot(predValid,ValidSet_yint)
abline(0,1)
mse_rf_out_8 = mean((predValid-ValidSet_yint)^2)
mean_error_rf_out_8 = mean(sqrt((predValid-ValidSet_yint)^2))
importance(fit.rf.8)
varImpPlot(fit.rf.8)
```

2. Variable Relevance (position-wise)

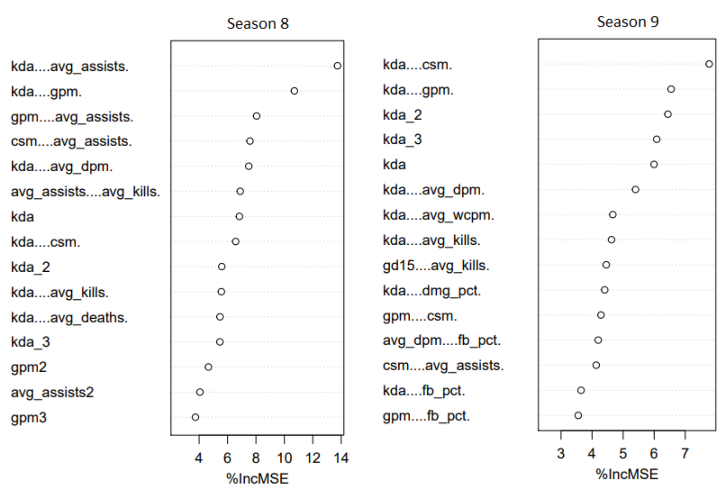

Figure 4: Top regressors in terms of variable relevance (Top players) for Season 8 and Season 9.

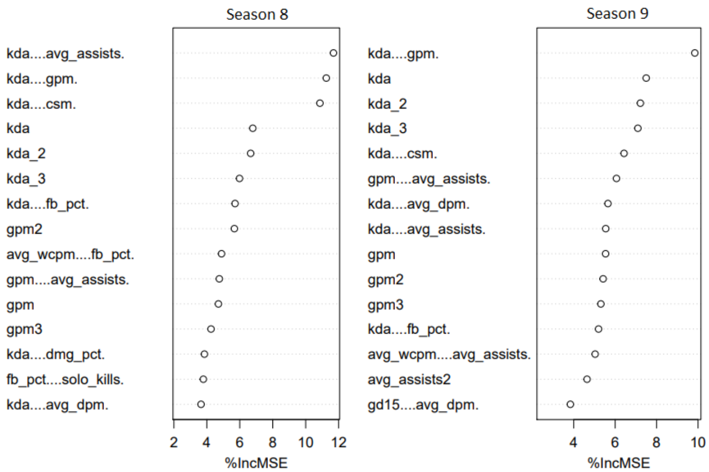

Figure 5: Top regressors in terms of variable relevance (Jungle players) for Season 8 and Season 9.

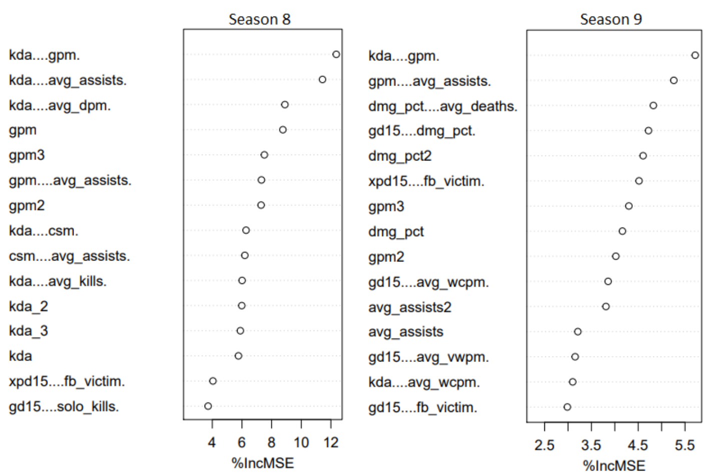

Figure 6: Top regressors in terms of variable relevance (Mid players) for Season 8 and Season 9.

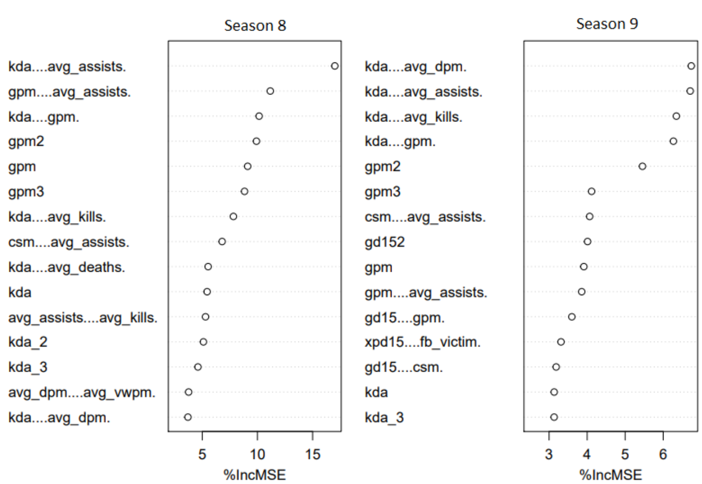

Figure 7: Top regressors in terms of variable relevance (Adc players) for Season 8 and Season 9.

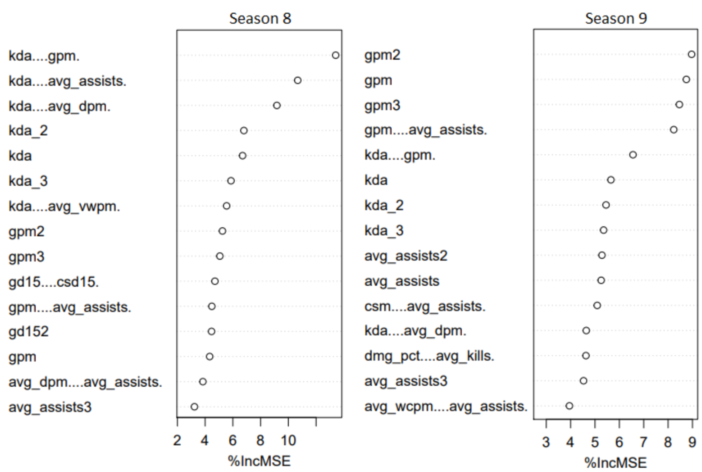

Figure 8: Top regressors in terms of variable relevance (Support players) for Season 8 and Season 9.