The Winning Formula: A Fencing Case Study

The Outliers: Alisha Li and Elaine Zhao

Abstract

This study investigates which fencing actions most significantly influence match outcomes for an épée fencer AL, when competing against right-handed pistol-grip opponents. Drawing from three years of match data, we modeled AL's net point outcomes based on the frequency of 15 offensive and defensive actions. We applied multiple statistical techniques—Multiple Linear Regression, Stepwise Regression, All-Subset Selection, and Regression Trees with ensemble methods—to identify the most impactful actions. Cross-validation revealed a six-variable linear regression model as the most predictive and parsimonious. Key negative contributors included being caught unprepared on the body or toe, while offensive lunges, especially to the body, were the strongest positive contributors. This case study not only offers actionable insights for AL's training but also demonstrates a replicable analytical framework for performance optimization in individual sports.

Background and Significance

Épée fencing is a combat sport where two opponents attempt to score points by hitting each other with their weapon. In Épée, fencers can hit each other on any part of the body and can attack/defend at any time with a variety of actions. Fencers hold their weapon in their dominant hand and can choose between one of two grips: french, which provides length, or pistol, which provides strength. Épée has been steadily increasing in popularity in the US and internationally, and competition has never been stronger. Many fencers are interested in gaining as much of an advantage as possible: whether through additional physical training, video analysis, or even meditation and other mental training. Our study thus began with a personal goal: One of the authors, who we'll refer to as AL, wanted to understand which actions were costing and earning her points in her matches. Since she had the most data on right-handed pistol grips, and because they are the most common opponent within Épée, we sought to answer the following question: What actions most positively and negatively contribute to AL's net points against right-handed pistol grips?

Data and Data Cleaning

AL has been collecting data, through video recordings, on all her competition matches from the past three years, from the national to collegiate level. For each match, she recorded the characteristics of the opponents such as hand and grip, as well as the results of all the actions she took against these opponents. This is an important note: each action in the dataset describes something AL did, not the opponent–since AL was interested in what *she* could control. Examples include p-riposte (when AL blocks the opponent's attack with her blade and then attempts to hit them back) and lunge (AL lunges at the opponent). In total we had 16 actions and 125 observations. Most actions have three columns: the first contains the number of times AL executed the action and gained a point, the second contains the number of times AL executed the action and both AL and the opponent received a point (in the case of simultaneous hits), and the third contains the number of times AL executed the action and lost the point. Some actions like CaughtB, which indicates AL being caught unprepared on the body, only have this third column and thus can only negatively contribute to the net outcome. The caught actions are unique in that they represent a lack of action. They were considered to signal moments of unpreparedness which may be something AL needs to work on if found significant.

To clean, the first thing we did was filter for right-handed pistol grips. Then we removed extraneous predictor variables including outdated and rarely used actions, and location of the match floor. We also renamed the columns to a more standard and readable format. The model structure we aimed to build was *netoutcome ~ number of action1 executions* + *number of action2 executions* + *. . .* + *error.* That is, given the number of times each action was executed in a match, we can predict the net outcome of the match. Thus we summed each action's three columns into one, labeled *actionname_exec*, to indicate the total number of times the specific action was executed against each opponent. We also created a *netoutcome* column which was the result of adding the first column and subtracting the third column. Note that the second column contributes net points of 0 since AL both gains and gives a point to the opponent, so it does not contribute to the net outcome. We also checked for multicollinearity. VIF screening revealed that *62Att_Exec* (the number of times AL executed action 62 Attack) exhibited severe multicollinearity (Fig A). After removing this, we are left with 15 predictors.

Methodology

To answer our research question, we applied a wide range of statistical techniques to identify the model that best characterizes the contributions of the actions to the response variable, *netoutcome*. The techniques we employed include Multiple Linear Regression (MLR), Stepwise Regression, All-Subset Selection, and Regression Tree and ensemble methods.

a. Model Fitting

We began by fitting the full MLR model. This yielded a model with 15 predictors, among which *CaughtB_Exec*, *CaughtT_Exec*, and *LungeB_Exec* contributed most significantly to *netoutcome*.

Next, we fitted stepwise regression models using both AIC and BIC criteria. Both approaches resulted in the same model, containing 6 predictors with identical coefficients. We then performed All-Subset Selection using Adjusted R^2, Mallow's Cp, and BIC. The Adjusted R^2 criterion produced a model with 8 variables, while both Mallow's Cp and BIC selected the same 6-variable model as the stepwise regression, with the same predictors and coefficients.

To identify potential interaction terms for our optimal first-order model (the 6-variable model), we performed a two-time stepwise regression. This model was chosen based on a balance between cross-validation scores and the number of predictors, which we discuss in detail later. The second regression produced the same 6 variables, with slight changes in coefficients. Thus, suspicions about significant interaction terms were dismissed, confirming the first-order model as the best so far.

We then explored the regression tree and ensemble methods. The regression tree produced an unpruned tree with 7 terminal nodes, placing *CaughtB_Exec* at the top as the most influential predictor (Fig B). However, the size vs. deviance plot (Fig C) indicated that the optimal tree size was 1 node, suggesting poor fit. We pruned the tree to the second best deviance, with two terminal nodes (Fig D) but this offered limited insight beyond confirming the importance of *CaughtB_Exec*. We then turned to tree ensemble methods, including bagging and random forest, which aggregate unpruned trees to yield more robust results. Comparing the variable importance plots from both methods (Figs E and F), we identified *CaughtB_Exec*, *LungeB_Exec*, *JumpLunge_Exec*, *AdvLunge_Exec*, and *RiposteDef_Exec* as the most important predictors of *netoutcome*.

b. Model Comparison

After trying all of the approaches, we performed cross validation to determine the best model to predict *netoutcome*. We performed both leave-one-out cross validation (LOOCV) and 5-fold cross validation, which yield similar results regarding model comparison (Table 1). Given our dataset of 125 observations, 5-fold cross-validation is more computationally efficient than LOOCV, especially for tree ensemble methods, as it requires building and fitting the model only 5 times instead of 125. Therefore, we took the results of 5-fold cross validation as the calibration of our model comparison. Comparing the CV scores, we found that the 6-variable model yielded by both AIC and BIC criterion in stepwise regression as well as both Mallow's CP and BIC criterions in all-subset selection had the equally lowest CV score, suggesting a best performance among all of our models.

AIC (step) 5-fold score: 3.890077 BIC (step) 5-fold score: 3.890077 BIC 5-fold score: 3.890077 AdjR 5-fold score: 4.116329 CP 5-fold score: 3.890077 BIC (2-time step) 5-fold score: 3.890077 Full 5-fold score: 5.164047 Bagging 5-fold score: 5.19934 Random Forest 5-fold score: 5.031806 Single Tree 5-fold score: 6.178653	AIC (step) LOOCV score: 3.962502 BIC (step) LOOCV score: 3.962502 BIC LOOCV score: 3.962502 AdjR2 LOOCV score: 4.106143 CP LOOCV score: 3.962502 BIC (2-time step) LOOCV score: 3.962502 Full LOOCV score: 4.59915

Table 1

c. Model Diagnostics

After settling on our optimal model (the model shared by AIC step, BIC step, Mallow's CP all subset, and BIC all subset), we performed some diagnostics. Our QQ plot (Fig G), which shows a mostly linear distribution, indicates that the normality assumption is met. Rows in the dataset represent distinct bouts, so independence is met as well. Assumptions of linearity, constant variance, and a mean of zero were evaluated using the residuals vs. fitted values plot (Fig H), which shows a random scatter evenly distributed around zero. Thus all assumptions are satisfied. After inspecting the outliers, we decided to retain them in the analysis because they represent valid and meaningful matches that we wanted to consider. Since each of the observations has been carefully recorded and reviewed, the outliers will also reflect real-world variability in AL's performances.

Results

Our final model is the 6-variable model, which has both the lowest CV score and the lowest number of predictors, satisfying both parsimony and goodness-of-fit. The model is below:

$$\widehat{\text{netoutcome}} = 0.67 - 0.35\,\text{RiposteDef\_Exec} - 1.86\,\text{CaughtB\_Exec} - 3.41\,\text{CaughtT\_Exec} + 1.21\,\text{LungeB\_Exec} + 0.73\,\text{JumpLunge\_Exec} + 0.39\,\text{AdvLunge\_Exec}.$$

The result shows that *CaughtT* most negatively contributes to netoutcome, while *LungeB* most positively contributes to netoutcome. Our final model produced a p-value of 1.131e-11 (Fig I), much smaller than 0.05, leading us to conclude that this model is adequate for our data. The Adjusted R-squared of 0.3748 indicates that 37.48% of variation in our data is explained by this model. The f-statistic indicated that overall, the model is statistically significant, meaning the predictors have a collective effect on the net outcome.

Conclusions and Other Considerations

The main purpose of this case study is to evaluate which actions most positively or negatively contributed to the net points AL earned in her past competitions. Given our best model, we can conclude that *RiposteDef, CaughtB, and CaughtT* negatively contributed to the *netoutcome,* while *LungeB, JumpLunge,* and *AdvLunge* positively contributed.

Indeed, across both regression models (Adj. R^2, and the rest) *CaughtB, CaughtT,* and *LungeB* were the most significant. These actions were also some of the highest in our ensemble variable importance plots as well. *CaughtB* and *CaughtT* represented getting caught off guard on the body and on the toe/foot respectively. Thus, we would advise AL to focus on always being ready for the opponent's attack. *LungeB* (which represents a lunge to the body), being positive, means we'd also advise AL to continue using this simple but effective action.

It is also interesting to note that all the negative contributions of our optimal model came from defensive actions and all the positive contributions came from offensive actions. This was generally the case for the outputs of our Full MLR, Adj. R^2, and tree models as well. This might indicate that when it comes to competition time, AL should try to be more aggressive and use these offensive actions since at the moment they produce the most positive results. But at practice, AL should definitely work on improving her defensive actions to lessen their negative effect on her net outcome come competition time, and which might help her in not getting caught off guard as much. We'd like to stipulate, however, that since the model only explains 37.48% of the data and has a residual standard error of 1.87 points, AL should be careful in letting it dictate all of her training. Other factors like coaching, mental and physical states during competitions, and timing of actions likely also have an influential effect on her performance.

In the future, AL can apply this same process to other groups of opponents like right-handed french, left-handed french, and left-handed pistols, once she gets enough data on those opponents. She may also be interested in creating a program that automatically displays these models on a running window of time. For instance, if she goes to more competitions and has more data points, she could train on only the last year's data to get a more relevant model.

Appendix

Figure A: VIF Screening

```
> vifstep(clean[, -17], th=5)
1 variables from the 16 input variables have collinearity problem: 

62Att_Exec 

After excluding the collinear variables, the linear correlation coefficients ranges between: 
min correlation ( LungeB_Exec ~ LungeT_Exec ):  -0.001032925 
max correlation ( LungeB_Exec ~ CaughtT_Exec ):  0.2540481 

---------- VIFs of the remained variables -------- 
        Variables      VIF
1    CounterA_Exec 1.208533
2  RiposteDef_Exec 1.140937
3  RiposteAtt_Exec 1.130921
4     CaughtH_Exec 1.150833
5     CaughtB_Exec 1.105136
6     CaughtT_Exec 1.134497
7       62Def_Exec 1.086469
8      LungeT_Exec 1.146990
9      LungeL_Exec 1.130484
10     LungeB_Exec 1.158701
11  JumpLunge_Exec 1.114313
12   AdvLunge_Exec 1.065461
13     Fleche_Exec 1.127257
14  PrepGuard_Exec 1.175412
15      Squat_Exec 1.141173
> vifstep(clean[, -17], th=10)
1 variables from the 16 input variables have collinearity problem: 
```

Figure B: Unpruned Tree

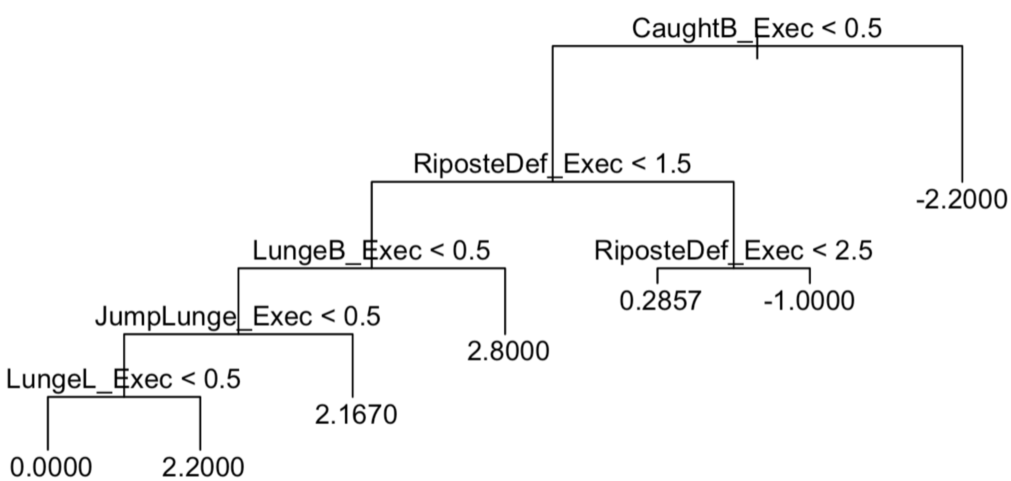

Figure C: Size vs. Deviance Plot

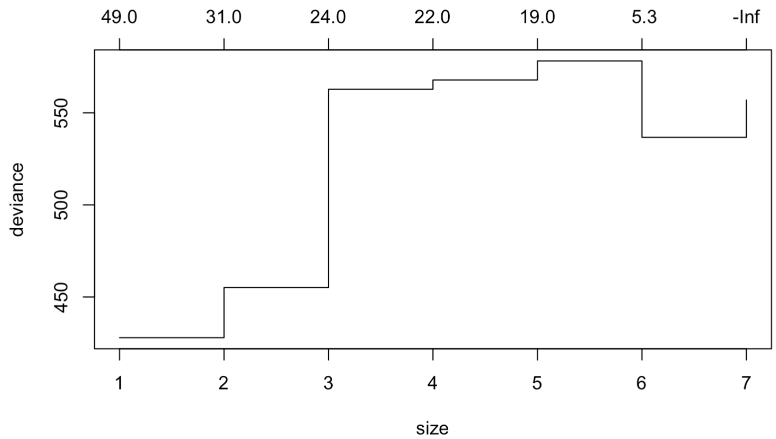

Figure D: Pruned Tree (2 nodes)

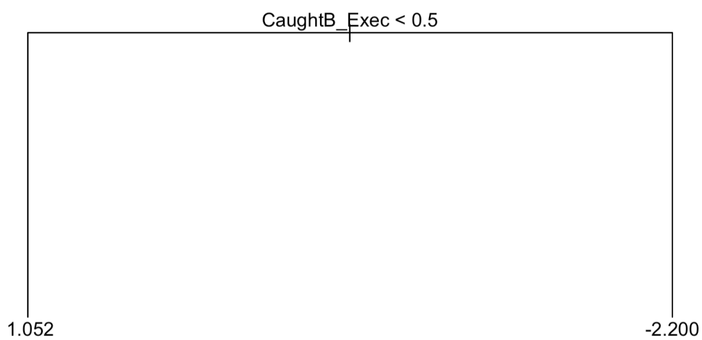

Figure E: Variable Importance Plot (Random Forest)

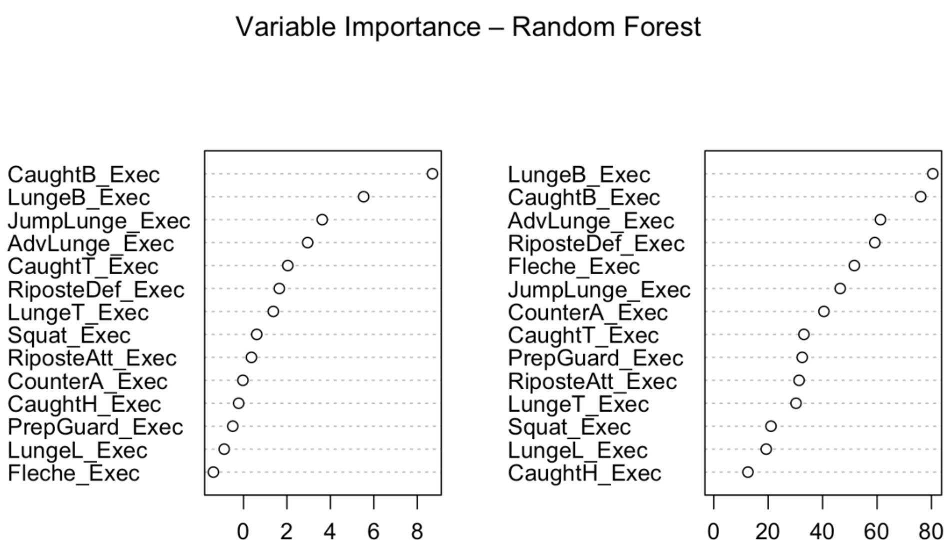

Figure F: Variable Importance Plot (Bagging)

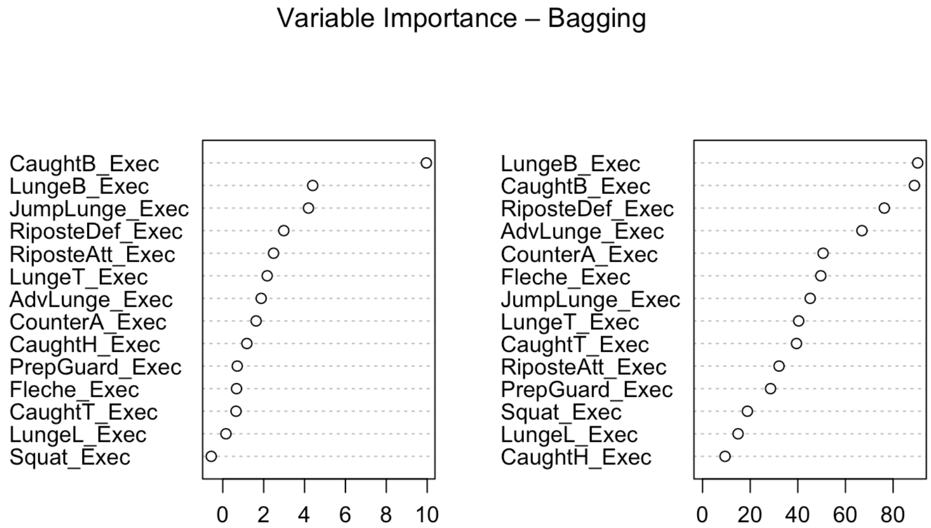

Figure G: QQ-Plot

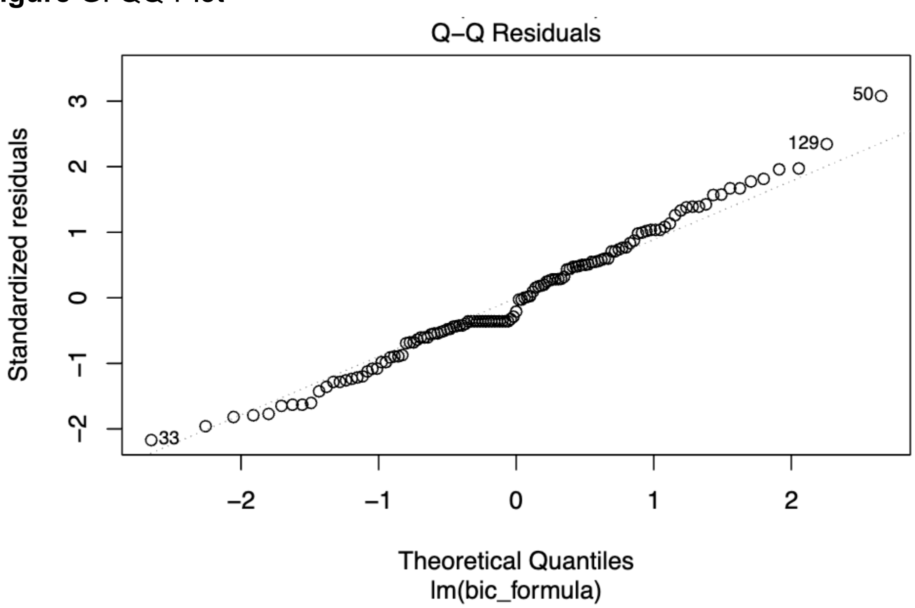

Figure H: Residual vs. Fitted Plot

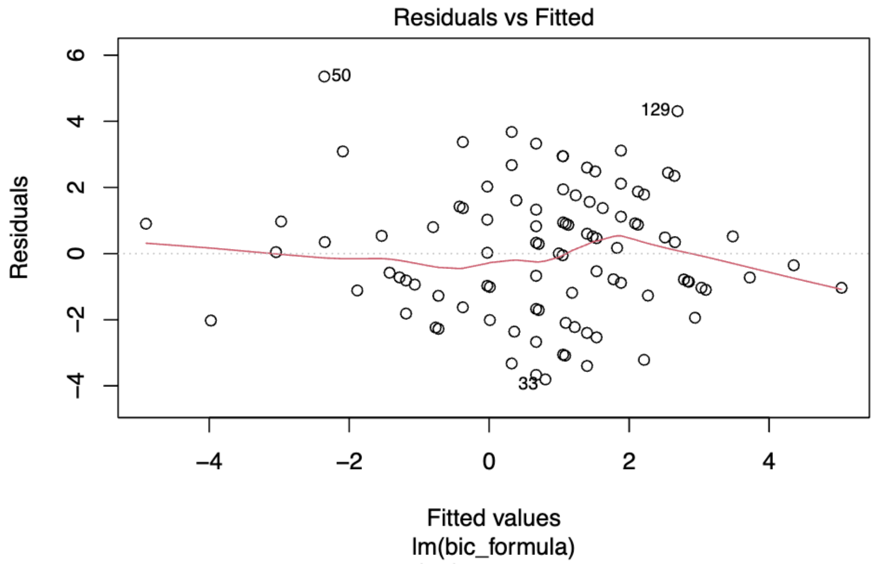

Figure I: Final Model

```
--- BIC Model ---

Call:
lm(formula = bic_formula, data = clean)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.8040 -1.0954 -0.3519  1.1167  5.3559 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)       0.6711     0.2538   2.644  0.00930 ** 
RiposteDef_Exec  -0.3488     0.1128  -3.092  0.00248 ** 
CaughtB_Exec     -1.8583     0.3676  -5.055 1.59e-06 ***
CaughtT_Exec     -3.4056     0.7076  -4.813 4.45e-06 ***
LungeB_Exec       1.2122     0.2708   4.477 1.76e-05 ***
JumpLunge_Exec    0.7274     0.2289   3.178  0.00190 ** 
AdvLunge_Exec     0.3856     0.1590   2.425  0.01684 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.892 on 118 degrees of freedom
Multiple R-squared:  0.4084,	Adjusted R-squared:  0.3784 
F-statistic: 13.58 on 6 and 118 DF,  p-value: 1.131e-11

BIC: 545.5279
```