Predicting the Outcomes of Soccer Games

Jack Schooley

Abstract

The outcomes of soccer games can be massively important to soccer fans and bettors alike. To predict the outcomes of these games, we consider the English Premier League (EPL) dataset from Kaggle, which has data from the 2006/2007 through the 2017/2018 seasons. The main predictive model used was an ordinal logistic regression model, and its predictive ability was compared against simple logistic regression models for each of the three possible outcomes of soccer matches. Variable selection for all models was done using separate lasso fits, which selected the features for each model using shrinkage. Finally, the resulting probabilities generated from the ordinal logistic model were compared against the implicit probabilities of the published Bet365 odds for each game prior to kickoff. Afterwards, we then formulated our own betting strategy using our model. We chose optimal bets to make from a probabilistic perspective and created a custom wager function to determine the monetary size of these bets. After running several simulations on past data to optimize for a threshold parameter, we then applied our model and betting strategy in a real-time experiment using real money over the course of 4 months. Starting with an initial budget of $200, we were able to earn a slight profit with a final balance of $223.40 after making 79 bets. While this is not quite a large enough sample size to judge the significance of our results, we view them quite positively, especially in the context of the current state of literature regarding sports betting using statistical learning techniques.

1 Introduction

Soccer is the world's most popular sport, and there is enormous interest in the outcomes of games. In this paper, we try to make predictions about EPL soccer games using statistical learning methodology. There are many quality papers that look at predicting outcomes of several different sports, but one that is especially relevant to our interests is Baboota and Kaur (2019), which has an exhaustive review of literature and a thorough attempt of applying machine learning to soccer games. This paper concluded that features are the limiting factor in predicting games. Another interesting paper is Marek and Chovanec (2019), where the authors note a difficulty in predicting draws. The high probability of a draw means that soccer is unlike most sports when it comes to predicting outcomes, as standard two-case classification methods must be extended to accommodate three outcomes.

For all of the papers published along the lines of this subject, though, the vast majority stop at predicting the outcomes of games and do not venture into betting territory. However, there is some literature on the subject, and luckily Hubáček, Šourek, and Železny (2019) summarizes the advances in this field quite well. That is to say, there is quite little significant work on the topic. The only paper that we are aware of that creates a profitable strategy for long-term betting is Kaunitz, Zhong, and Kreiner (2017); these people essentially pit the sportsbooks against each other by taking advantage of odds that were better at one book as compared to the whole. While this paper shows that profiting is possible, they did not use a statistical model for the outcomes of games to choose their bets, which is what we will do in this paper.

There are a couple main reasons for the lack of known profitable models, foremost among them being the fact that profiting in the long-term is quite difficult. It's relatively common wisdom in the sports betting community that at least 99% of bettors are not long-term winners. That being said, the existence of professional sports bettors proves that it is possible to profit long-term. This brings us to what we believe is the second main reason why there is no literature showing sustained profits, and that is because there is a clear economic incentive to keep successful methods to yourself. If someone comes up with a successful method, then they can either use it to make money for themselves, or they can share it with the world and potentially lose their competitive advantage in the market.

The model of choice for our purposes was an ordinal logistic regression model. This method will generate probabilities for each of the three outcomes of soccer games. This model turns out to be relatively effective; to show this, we compare model efficacy against simple logistic regression models for each outcome using ROC curves. For example, a model classifying home win against all other outcomes (away win and draw) will be compared to the home win probability generated by the ordinal model, and so on for the other two outcomes. Next, we can also compare the probabilities generated by the ordinal logistic regression model to the implied probabilities of betting odds. Betting odds are perhaps the most accurate predictors of the results of games, so they can serve as an effective way of checking the accuracy of the model. Assuming that betting odds approach the true probabilities of each outcome, then the closer our model is to them, the better.

After our classification model is finished, we can consider how we can use our model to make strategic and effective bets. To do this, we will run simulations on past games to develop a betting strategy, including the development of functions to determine which outcome to bet on for each game (if any) and the amount wagered. When doing this, we will examine the Kelly criterion (Kelly 2011) and its limitations (Chu, Wu, and Swartz 2018). Finally, after running several simulations on past results, we will run an experiment with our model on current games as they happen; this is the only true test of how well our model performs in a betting context.

2 Data

The input data used to train the model comes from a Kaggle dataset, which itself is data scraped from the official website of the Premier League (Nalla 2018). The data consists of stats from the 2006/2007 through the 2017/2018 season, and it is pretty wide-ranging in terms of what is tracked, from basic things like goals and shots to pretty in-depth things like total passes and touches. However, this data is only available on a per-team, per-season basis. The data is not on a per-game basis, which was not a concern at first, but later we will discuss why this causes a lot of bias when running betting simulations on past results. This

Kaggle dataset also contains the results of every game from the aforementioned seasons.

Our other data source is Football-Data, which we use mainly for historical betting odds data (Football-Data 2020). This dataset has historical odds from a variety of different sportsbooks, but we will only consider the odds from Bet365; this choice was made largely based on completeness of the data across all of the seasons considered. This dataset also contains a few basic statistics for each match, such as goals, fouls, and yellow cards; while it is much more limited in scope than our Kaggle dataset, we will nonetheless be making use of these statistics in our final simulation.

3 The Model

3.1 Feature Construction and Selection

We did some transformation to the original Kaggle dataset before training the model. Specifically, we transformed the main statistical data to a input vector for each game; each game has two main "groups" of features that comprise the total feature set. Firstly, the "level" features (denoted with an l subscript) are just the home team's stats (denoted with an h subscript) for the season in which the game was played. Thus, for any feature x in the feature set,

$$x_l = x_h.$$

The level features are mainly used to contrast with the "difference" features (denoted with a d subscript), which are constructed by subtracting the away team's stats (denoted with an a subscript) from the home team's stats for the season in which the game was played. Thus, for any feature x in the feature set,

$$x_d = x_h - x_a.$$

For every response y_i, which is the outcome of game i, our input vector for this game x_i is comprised of the level and difference features together. To give a concrete example, consider a match where Manchester United hosted Liverpool on September 12, 2015, which United won 3-1. In the 2015/2016 season, Manchester United scored 49 goals, whereas Liverpool scored 63; thus, the level feature for goals would have a value of 49, whereas the difference feature for goals would have a value of -14.

After we construct the total feature set, we must now perform some feature selection, as our input dataset has 62 potential features with some correlation between them. To do this, we used the lasso loss function

$$\sum_i (y_i - \alpha - \sum_j \beta_j x_{ij})^2 + \lambda \sum_j |\beta_j|$$

to yield a sparse set of coefficients $\hat{\boldsymbol{\beta}}$ by letting the quantity $\lambda \sum_j |\beta_j|$ set some of the coefficients to 0 (Tibshirani 1996). As λ increases, the coefficients will shrink closer to 0, and more coefficients will shrink to exactly 0, yielding a sparser model. Selection for an optimal λ regularization parameter was first done using 10-fold cross-validation, and the variables that were selected using this method are shown in Table 1.

Table 1: Ordinal Logistic Regression Variables

Goals, difference	Shots on target, difference
Free kick goals, difference	Clean sheets, difference
Goals conceded, difference	

Using the features that the lasso selected, we then randomly separated our data into a training set ($n = 1000$) and a test set. We then trained our ordinal model, which is a proportional odds logistic regression model, on the training set. This model is

$$\log(\frac{\gamma_j}{1-\gamma_j}) = \zeta_j - \boldsymbol{\beta}^T \mathbf{x},$$

where $\gamma_j = P(y \leq j)$, $\boldsymbol{\beta}$ is our coefficient vector, $\mathbf{x}$ is our input vector, and $j \in 1, 2$ (given that we have three outcomes) (Brant 1990). We obtain coefficients β_i for $i \in \{1, 2, ..., 5\}$ (as we now have $p = 5$ predictors) and two intercepts ζ_1 and ζ_2. Then, for each game in the test set ($n = 3560$), we use the estimated values of γ_j to compute the estimated probabilities of home wins, away wins, and draws, which are henceforth denoted as $\hat{p}_h$, $\hat{p}_a$, and $\hat{p}_d$, respectively.

3.2 Comparison to Two-Outcome Classification

Our first way of comparing model efficacy was to compare our ordinal logistic model against individual simple logistic models for each of the three outcomes, given as

$$\log(\frac{\pi}{1-\pi}) = \alpha + \boldsymbol{\beta}^T \mathbf{x},$$

where $\pi = P(y = j)$ (with j corresponding to the positive outcome in the one vs. all scenario), α is the intercept term, $\boldsymbol{\beta}$ is the coefficient vector, and $\mathbf{x}$ is the input vector (Peng, Lee, and Ingersoll 2002). The same total feature set (with level and difference variables) for the ordinal model was also used for the home win and away win simple logistic models. After the total feature set is constructed, we once again use separate lasso fits for feature selection. The selected variables for the home win model are in Table 2, and the selected variables for the away win model are in Table 3. Note that both level and difference features were selected in each of these models.

Table 2: Home Win Simple Logistic Variables

Goals, level	Goals, difference
Free kick goals, difference	Inside box goals, difference
Offsides, difference	Goals conceded, difference
Touches, difference	

Table 3: Away Win Simple Logistic Variables

Header goals, level	Penalty goals, level
Outside box goals, level	Counterattack goals, level
Offsides, level	Tackles, level
Own goals, level	Long balls, level
Corners, level	High claims, level
Goals, difference	Shots on target, difference
Header goals, difference	Free kick goals, difference
Outside box goals, difference	Clean sheets, difference
Goals conceded, difference	Tackles, difference
Touches, difference	Goal line clearances, difference

There is a key difference in the feature set used for the draw simple logistic model in that the absolute value of the difference variable is used in the total feature set, whereas this is not the case for the other models. So, for the draw model only, for any feature x in the feature set,

$$x_d = |x_h - x_a|,$$

where x_d denotes the difference feature, and x_h and x_a denote the raw statistics for that season for the home and away team, respectively. The reason for this is that when we are looking to predict draws against all other outcomes, we are only concerned *if* one team is better than another in a given metric, not *which* team is better. Once again, we used a lasso model to select the variables shown in Table 4.

Table 4: Draw Simple Logistic Variables

Inside box goals, level	Counterattack goals, level
Interceptions, level	Tackles, level
Goals, difference	Penalty goals, difference
Counterattack goals, difference	Clean sheets, difference
Goals conceded, difference	

Now that we have simple logistic models to compare to our ordinal logistic model for each outcome, we can construct ROC curves to measure the effectiveness of our ordinal model, as shown in Figure 1. We construct these ROC curves by taking the estimated probabilities of each outcome and comparing them to the probability estimated by their respective simple logistic model. Then, by varying the threshold at which we classify outcomes in a binary sense (one vs. all) based on their probabilities, we can generate curves.

3.3 Comparison to Betting Odds

Finally, we can test our model efficacy by comparing it to betting odds. To do this, we must simply graph the probabilities $\hat{p}_h$, $\hat{p}_a$, and $\hat{p}_d$ generated by our model for each game against

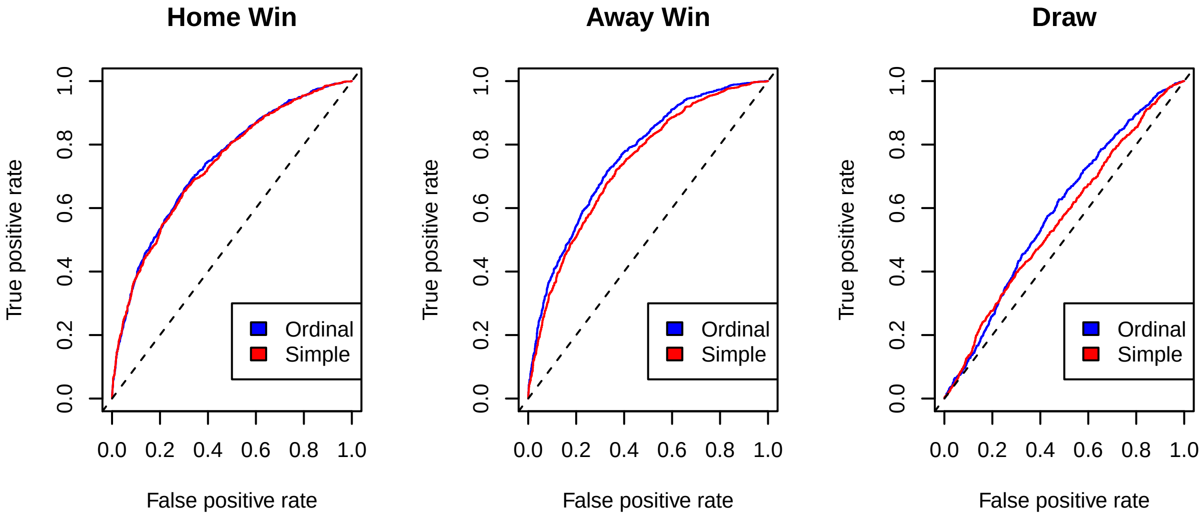

Figure 1: ROC Curve Comparisons

the implied probabilities of betting odds, henceforth denoted as p_h^*, p_a^*, and p_d^* for home wins, away wins, and draws, respectively. This is shown in Figure 2. We use the closing odds published by Bet365 just prior to kick off to represent the implied probabilities, and the red lines are simply identity lines.

As you can see, the identity lines look to be pretty close to being legitimate best-fit lines. The probabilities generated by our ordinal model match up rather well to the implied probabilities of betting odds. However, there is a caveat to this in that bookmakers shade their lines to make events appear more likely than they actually are; that is, the implied probabilities will add up to slightly greater than 1. This is the bookmaker's cut (also known as "juice" or "vig"), and it is the main reason why bookmakers make money. As far as our results go, though, this fact actually makes them look even more favorable, as this would move the true implied probabilities (not counting the bookmaker's cut) down on the graph, which would appear to move them even closer to our generated probabilities.

Now, while our model appears to approximate betting odds very well, we will shift our concern to where our generated probabilities significantly differ from the implied probabilities of betting odds. If we were to bet on these games, in general we would want to bet on the outcomes which are to the right of the red line on the graphs. That is, we want to bet on outcomes where the probability generated by our model is greater than the implied probability of the betting odds. In the long-term, this means that we will generate profit by getting rewarded at a rate which is higher than one would expect for a given payout. This idea will form the basis of our betting strategy, although there are many other things to consider.

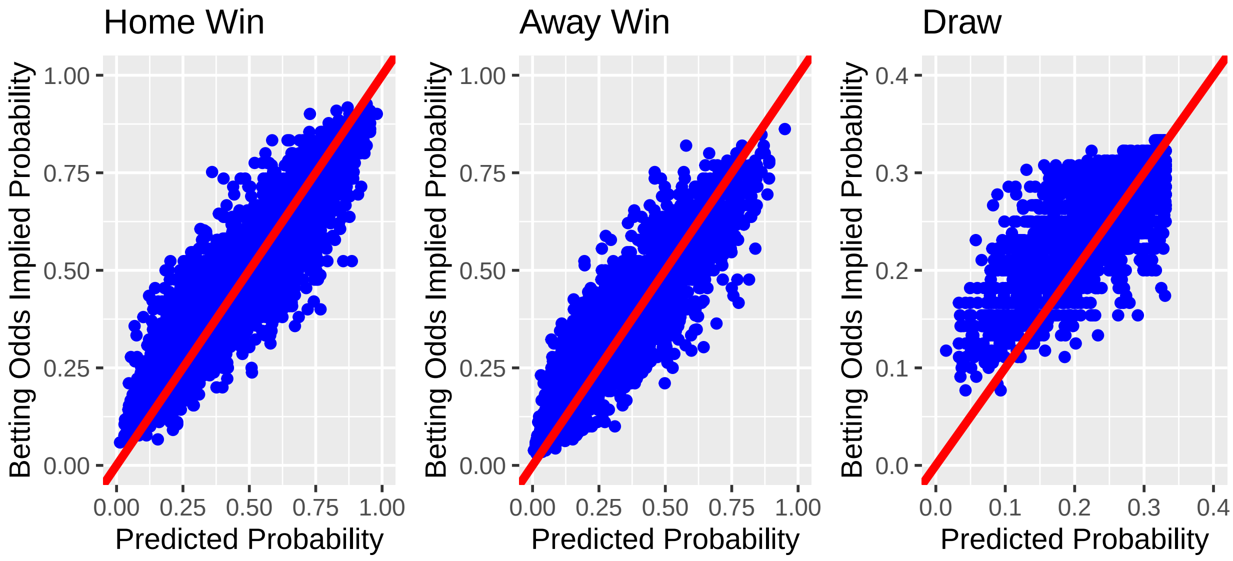

Figure 2: Betting Odds Probability Comparison

4 Betting

4.1 Selecting Optimal Outcomes for Betting

The gist of our betting strategy is that we want to bet on events that our model thinks are more likely than their odds in betting markets imply. To start with, we use our model to compute the probability of each event, which are denoted as $\hat{p}_h$, $\hat{p}_a$, and $\hat{p}_d$ for home wins, away wins, and draws, respectively. Then we compare these probabilities to the implied probabilities of betting odds, which are similarly denoted as p_h^*, p_a^*, and p_d^*. Thus, for any outcome $j \in \{h, a, d\}$, we want to bet on outcomes such that $\hat{p}_j > p_j^*$. These bets have a positive expected return, and we will show this using decimal odds.

Not only are decimal odds the standard odds used in Europe (as opposed to fractional or American odds), they also are very convenient for calculation purposes. The return for any successful bet made using d dollars at decimal odds o_j is $d(o_j - 1)$, and the implied probability of decimal odds is simply their inverse (that is, $o_j = \frac{1}{p_j^*}$) Thus, for any bet made using d dollars on an event with true probability p_j, our expected return r_j is

$$E[r_j] = d(\frac{p_j}{p_j^*} - 1).$$

We know that $d > 0$, therefore the sign of the expectation depends strictly on p_j and p_j^*. If $p_j > p_j^*$, then $E[r_j] > 0$, and if $p_j < p_j^*$, then $E[r_j] < 0$. If $p_j = p_j^*$, then $E[r_j] = 0$ and these odds are fair. However, fairness is not really relevant to our concerns, as bookmakers make money by offering explicitly unfair odds such that $\sum_j p_j^* > 1$.

We will only consider a maximum of one bet on each game; while it is theoretically possible for two of the three outcomes to be undervalued at the expense of a highly overvalued third outcome, we assume competence on the part of the bookmaker to the degree that it is impossible to make multiple positive expected return bets on mutually exclusive outcomes. We believe this assumption lines up well with reality. In addition, we made the decision to not bet on draws altogether, as it is clear that our model is relatively poor at selecting for them in comparison with home or away wins. Thus, we only consider home win and away win outcomes when choosing bets. We do not show the results used to justify this action here, as the simulations used to inform our decision were biased (which we will address later). However, these simulations showed that not only did draws produce almost no net profit, they also took profitability away from bets made on the other two outcomes.

Another thing that we are forced to reconcile with is that our model probabilities are only estimates. Thus, not all values $\hat{p}_j$ such that $\hat{p}_j > p_j^*$ will actually lead to positive expected returns due to the error involved in estimation. Thus, we introduce a threshold parameter, denoted as δ, which is the minimum difference between $\hat{p}_j$ and p_j^* that we will accept in order to make a bet. Formalized, we will only make a bet if

$$\max(\hat{p}_h - p_h^*, \hat{p}_a - p_a^*) > \delta.$$

Given this constraint, we can then select a bet on outcome j such that

$$\hat{p}_j - p_j^* = \max(\hat{p}_h - p_h^*, \hat{p}_a - p_a^*).$$

As δ increases, we become more selective about our bets. Selectivity with bets is one of the core ideas that make it possible to profit in sports betting, as bookmakers have to create lines for every outcome of every game, while bettors have the ability to pick and choose their bets in such a way that exploits the bookmakers' inefficiencies. We will select an optimal δ after choosing a wager function.

4.2 Selecting Optimal Wager Amounts

There are two major parts to betting; while we have described our methodology for choosing which outcomes to bet on, there is still the equally important task of determining how much money to wager for each bet. Luckily, the optimal wager amount for any bet is a question that has already been answered by academics. The Kelly Criterion, developed by John L. Kelly, Jr., is a mathematical formulation for the optimal wager amount (Kelly 2011). The Kelly criterion $k(p_j)$ is a fraction of the bankroll that a bettor should wager on an event with decimal odds o_j but with true probability p_j such that

$$k(p_j) = \frac{p_j o_j - 1}{o_j - 1}.$$

This formula is subject to our earlier constraint that $p_j > p_j^*$, as otherwise the optimal wager amount would be 0. As stated earlier, $k(p_j)$ will return a value between 0 and 1, and this is merely the fraction of your bankroll that you should wager.

While this criterion is a great starting point, there are some assumptions that it makes that do not apply to our methods. The most common argument against the Kelly criterion is that it favors risking too much money, as many sports betting enthusiasts recommend using a "half-Kelly" or some other fractional amount of the full criterion. A more rigorous line of reasoning for this idea is that the criterion assumes that the true probability of the event is known, and this is not the case in sports (Chu, Wu, and Swartz 2018). As previously mentioned, our model can only provide estimates of the probabilities of events, so this would lead us to be more conservative in our wager amounts as compared to the Kelly criterion.

Another problem with applying the Kelly criterion to sports betting is that the criterion assumes that only one bet occurs at a time, and we can always readjust our budget using the previous result when making a new bet. In sports, this does not usually apply, as there are multiple games occurring simultaneously more often than not. This would also cause the Kelly criterion to select wagers that might be riskier than we would like, as multiple bets in progress at the same time will distort our budget for the criterion's purposes. In these situations, we will always err on the side of caution, as sports betting is a long-term game, and we want to avoid ruin at all costs.

Thus, using the Kelly criterion as a base, we made adjustments to develop our own custom wager function. In our case, we wager a fraction of our bankroll $w(\hat{p_j})$ at decimal odds o_j such that

$$w(\hat{p_j}) = \min(0.1, k\frac{\min(0.1, \hat{p_j} - p_j^*)o_j}{o_j - 1}).$$

We know that p_j^* is the implied probability of o_j, and thus $p_j^* = \frac{1}{o_j}$. However, it is more convenient to think of the numerator as a function of the difference between estimated and implied probabilities, thus we denote it as such. This change will produce a numerator value that is less than the p_j value that the Kelly criterion uses, so we multiply our quantity by o_j as opposed to $o_j - 1$ to compensate for this. In addition, our difference quantity is capped at 0.1, as we believe that any difference $p_j - p_j^* > 0.1$ is more likely to be poor estimation on our part as opposed to the bookmakers'. It is circumstances like these where betting odds most likely have information encoded in them that is hard to account for using a model, like an injury to a key player or a manager change. k is a parameter that we will set to be between 0.1 and 1; $k = 1$ approximates the full Kelly amount $k(p_j)$, so we will most likely want to use something lower. Finally, we also cap the entire quantity $w(\hat{p_j})$ at 0.1 so that we do not bet more than 10% of our bankroll on a single outcome. This is actually a pretty high limit by sports betting standards, as conventional community wisdom suggests a cap of around 5%.

Figure 3 shows a graph of final balance as a function of k. We computed these final balances by running simulations on past data, in which for each season, we trained a model using data from all other available seasons, and then we made hypothetical bets using the closing odds for each game. The starting balance for the hypothetical bets is $200, and the balance is adjusted after each bet until the season ends. The final balances for each of the 11 seasons are then averaged and plotted as a function of k, which is shown as follows. As you can see, the final balance is highest when $k = 1$, but this simulation did not match real conditions with regards to multiple games going on at a time. Thus, tweaking k is more of a product of intuition as opposed to optimization. We chose to use $k = 0.3$ for future simulations and experiments because on average there are around 3 games occurring simultaneously.

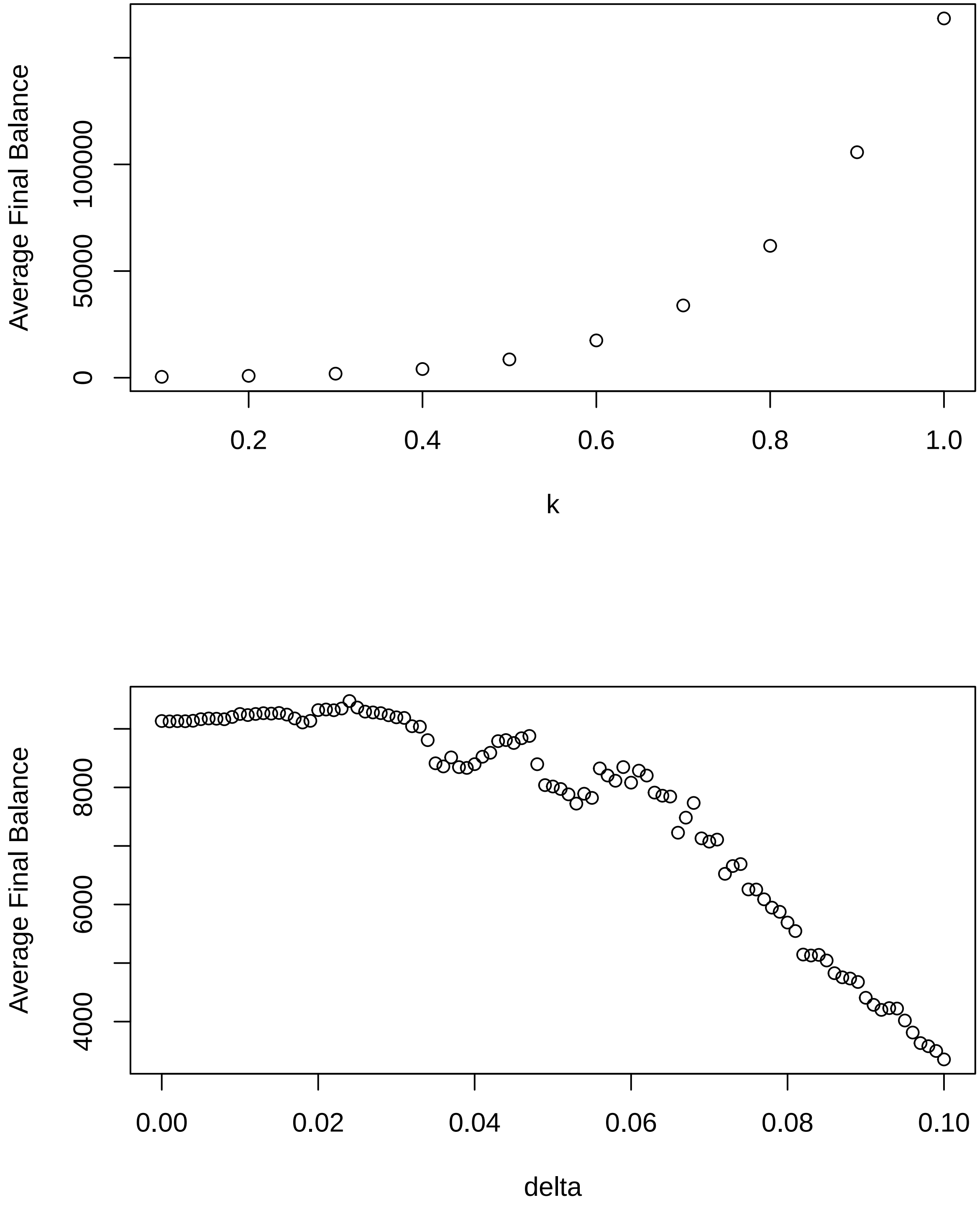

Figure 3: Selecting for k (top) and delta (bottom)

This value also allows us to be pretty conservative with our bankroll, which is desirable in a long-term game like sports betting.

Now that we have a wager function, though, we can optimize for the threshold parameter δ. Using a k value of 0.3, we constructed a simulation in the same manner as the previous one, plotting final balances as a function of δ.

As also seen in Figure 3, it appears that the optimal δ is at about 0.025. As you go from left to right, you are betting on fewer games, and so there is more variance involved. You might think that as we increase our threshold, we are only removing the bets that the model is less confident in, thus we should have higher expected returns. This would be true if not for the fact that we are dealing with probabilistic events, and so a small number of bets leaves us vulnerable to a high amount of variance. Thus, it makes more sense that the optimal point is at a moderate value that is nonzero yet not especially high.

The y-axis for both this plot and the previous one is reflective of average final balance after a full season when starting with an initial budget of $200. You might think that these values, which offer enormous returns, are way too good to be true, and you would be correct. Up until this point, every point of analysis and simulation has been done with biased data. Because the data that we have is only on a season-by-season basis and not a game-by-game basis, there is already implicit information about the results of games in the input data. There are 38 games in a season, so 1 game out of 38 might contribute only a few percentage points to the whole, but this is still essentially using the future to predict the past. The betting odds are reflective of their time, and we know more about a team by virtue of its full-season performance than we could have if we were actually betting on these games in the past. Thus, analysis of this past data would not translate well at all to betting on current games.

4.3 Revised Simulation

At this point, it is then important that we should construct a simulation that more closely resembles a method that we can use to bet on current games. There are a number of key changes that need to be made, the most important of which is our data source. The betting odds data that we use also happens to have basic box score stats for each game like goals and shots on target, and it just so happens that our model largely uses only these variables, so we can proceed with this data provided by Football-Data. The only difference in the features used for our ordinal logistic model is that free kick goals will be dropped from the model, as our new dataset did not have this at all. Table 5 shows the remaining variables that we will use from now on.

Table 5: Final Ordinal Logistic Variables

Goals, difference	Shots on target, difference
Clean sheets, difference	Goals conceded, difference

Another question that has to be answered is how we should construct our test set. We chose to use an equally weighted combination of the previous season's data as well as the current season's data up the point of when the game was played. We felt that only one previous year was sufficient to judge team quality, as the English Premier League is relatively stagnant in terms of each team's relative performance. That is to say, the good teams and bad teams are pretty constant year in and year out. The training set was just all the other seasons that we had data for prior to the start of data used for the test set. Mathematically, for any feature x in the feature set corresponding to season i,

$$x = \frac{1}{2}x_i + \frac{1}{2}x_{i-1}, \tag{1}$$

where x_i is an accumulated season statistic that increases as the season progresses, whereas x_{i-1} is static. Each of these are weighted equally, so as the current season's data accumulates as more games occur, the actual weight placed on the current season increases relative to the prior season's. These team-level statistics x were then used in difference calculations to calculate x_d for each of the difference features.

The construction of the test set raises yet another complication, and that has to do with newly promoted teams. Each year, the bottom 3 teams in the Premier League are relegated to the second division, called the Championship. Likewise, the top 3 teams from the Championship (more specifically, the top 2 teams and the promotion playoff winner) are promoted to the Premier League. This means that the newly promoted teams for the season that we are trying to simulate will have no data to use for the previous season, creating another issue. We chose to rectify this by assigning each of the three teams the same full-season stats for this missing year, where each individual stat is assigned as the league average for that stat (denoted as x_{i-1}^{-}) minus half a standard deviation (denoted as $\sigma_{x_{i-1}}$), rounded down in all cases except goals conceded, which was rounded up. These newly promoted teams are nearly always below average, and we feel that the way that we assigned these stats is fair. Thus, for newly promoted teams that do not have defined stats for season $i-1$, all their previous season stats x_{i-1} are calculated as

$$x_{i-1} = x_{i-1}^{-} - \frac{\sigma_{x_{i-1}}}{2}. \tag{2}$$

These values are then used in Equation (1) to calculate any feature x for newly promoted teams.

Finally, we made a change in the calculation of subsequent budgets throughout the season, in that the budget is only recalculated once per day at most. This aligns more with a realistic scenario in which you make all your bets for a given day at one time and only reevaluate your budget once they are all complete. This experimental design is now easily in line with what we can do to bet on current games, and we can once again do a full-season simulation. Figure 4 shows the results with a plot of final balances as a function of δ once again.

Two plots are included in Figure 4. The top plot is reflective of our wager function with $k = 1$, and the bottom plot is reflective of our wager function with $k = 0.3$. This is evidence

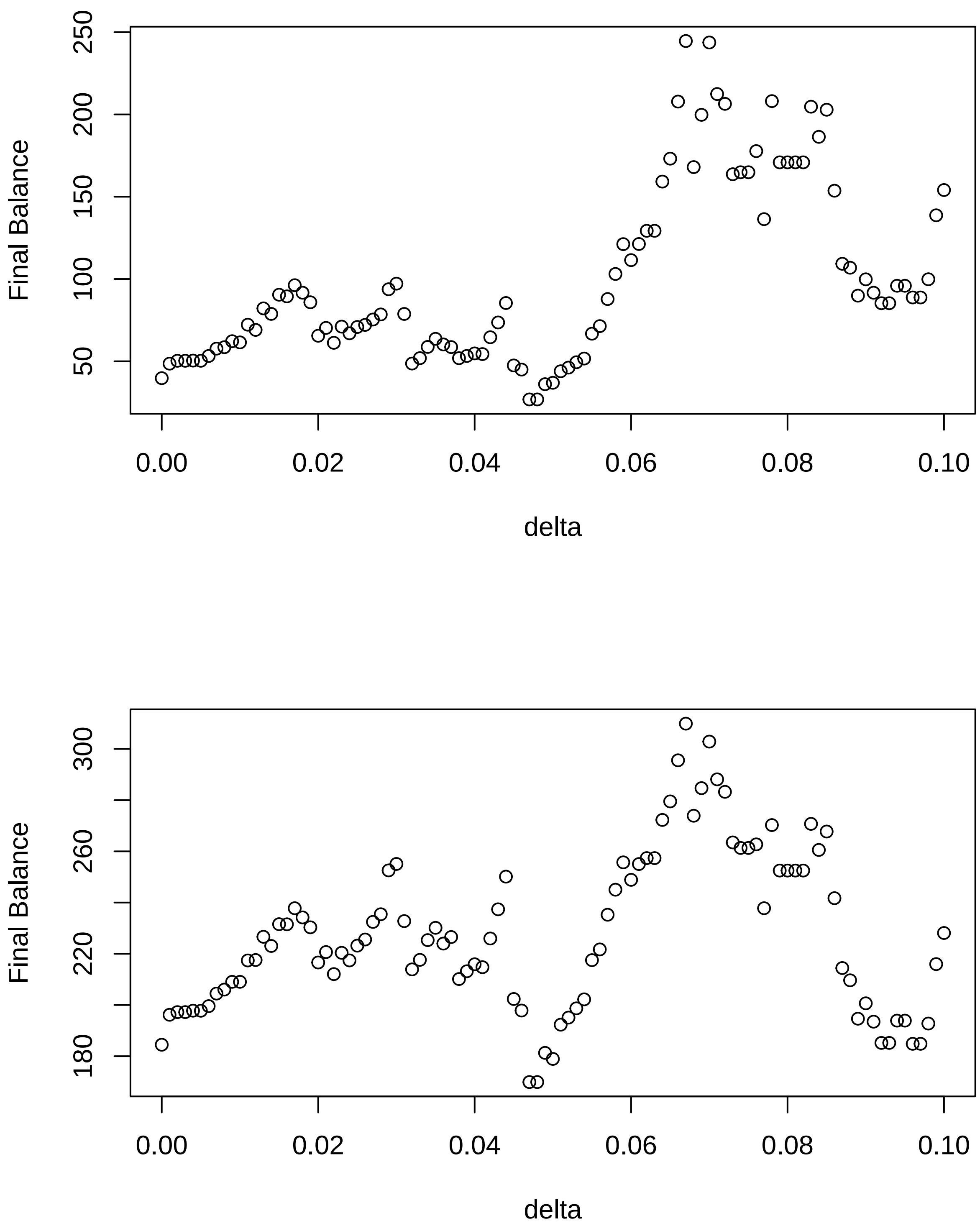

Figure 4: Unbiased Simulation Results with k = 1 (top) and k = 0.3 (bottom)

that a fraction of the Kelly criterion is actually more optimal in sports betting than the full Kelly, as a k value of 1 is when our wager function approximates the true Kelly criterion. We can clearly see that a k value of 0.3 outperforms the higher value, justifying our use of a custom wager function.

The starting balance was again \$200, and so this is much more in line with what we could expect with this model. There appears to be a very slight upward trend, but there is also a lot more variance as we progress further on the x axis. There are a lot of δ values with small or even decent profits with $k = 0.3$, but there are also some values with money losses as well. As far as an optimal δ goes, it's hard to say exactly from only these plots. We decided to stick with our previously obtained value of 0.025 due to our previous results. The previous simulation might have been biased, but 0.025 still seems to be as good a choice as any other.

4.4 Real-Time Betting Experiment

Finally, after all of this tinkering, we were able to apply our model to current games with a true betting experiment that lasted around 4 months, which is about half a season. We use the same experimental design as the previous simulation with respect to our model, training set, test set, and accounting for newly promoted teams. We took betting odds from the betting website BetOnline.ag and fed them into the model for each day there were games, and made real bets on the website with a starting balance of \$200. Each week, we collected the most updated statistics from the official website of the Premier League, and these statistics served as the x_i from Equation (2). The results are shown in Figure 5.

In the end, we were able to finish with a final balance of \$223.40 for a slight profit after making 79 bets. There were definitely ups and downs over the course of the experiment, and this should provide evidence that sports betting requires large sample sizes to accurately evaluate performance. In fact, a sample size of 79 is still pretty small by sports betting standards, and ideally, we would have at least a full season-long experiment to accurately judge our model. It might seem that our model performed better as time went on, and this could make sense in theory as our model shifted more weight to the current season as more games were played, but again the sample sizes involved here are too small to say anything meaningful regarding this uptake in performance.

5 Conclusion

Starting from the efficacy of our model in a vacuum, we showed that our ordinal logistic regression model was at least as good at prediction as three separate simple logistic regression models for each of the three outcomes of soccer games. These classification models are relatively good at predicting wins for either the home or away team, but they are relatively poor at predicting draws. This result is consistent with previously established literature that states that draws are the most difficult outcome to predict.

We compared the probabilities generated by our model to the implied probabilities of betting odds, and the results were quite excellent. Betting odds are known to be highly accurate

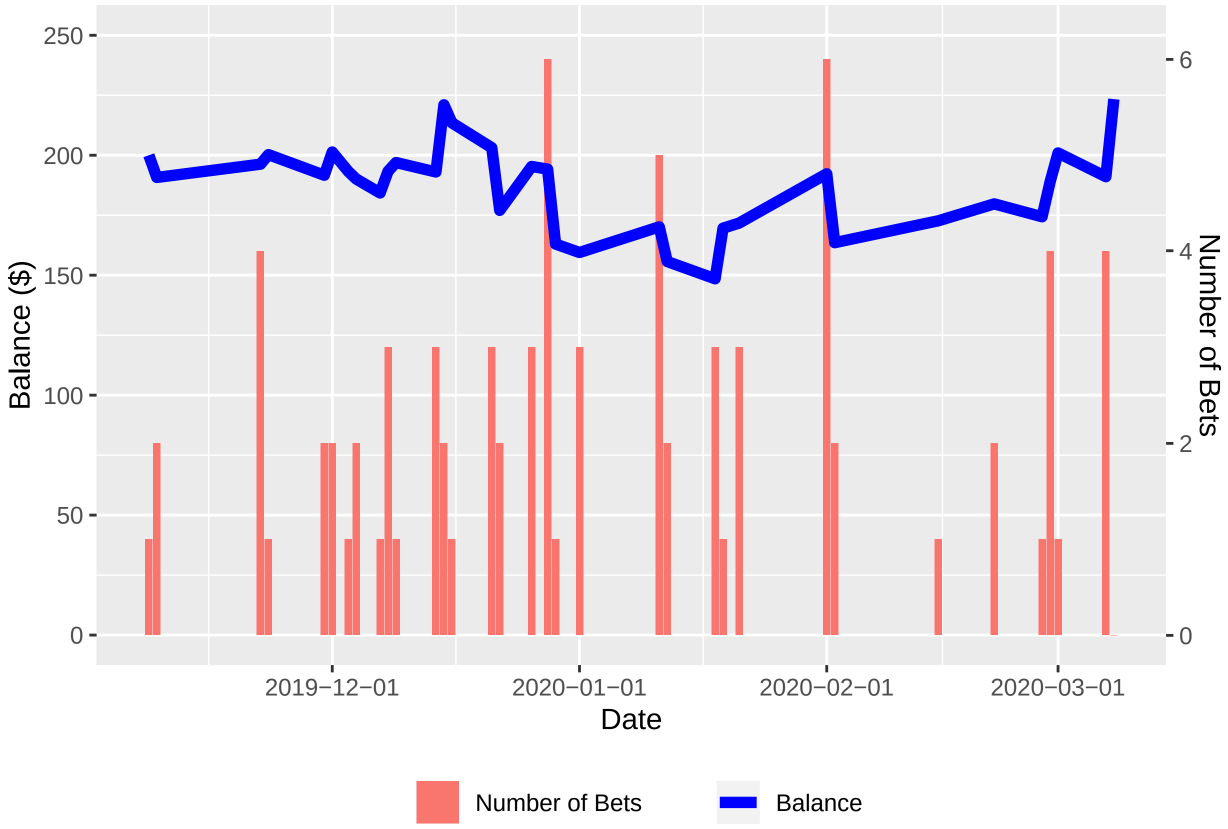

Figure 5: Betting Experiment Results

in modeling the probabilities of sporting outcomes, meaning that our own generated probabilities are also pretty accurate representations of the true probabilities. Our probabilities tended to be slightly less than the implied probabilities on the whole, but this is actually a good thing. The implied probabilities used for comparison did not have the bookmaker's cut taken out, so these probabilities were shaded to be slightly higher than what the bookmakers believe the actual probabilities to be. After all, these implied probabilities add up to slightly greater than 1, so this reflects well upon our model.

However, going from "fitting the betting odds well on a graph" to "making money consistently" with our model was quite a large leap. After running several simulations, we developed a betting strategy with optimal bet choices as well as a custom wager function for each bet. It was at this point that we learned that our statistics left a lot to be desired in terms of timing detail, and thus many of our simulations produced unrealistic results because our inputs were biased. Finally, we were able to run a 4-month real betting experiment for which we were able to produce a slight profit after 79 bets, in which our bankroll increased from \$200 to \$223.40. While our sample size was still not large enough to accurately conclude the efficacy of our betting strategy, we were able to produce a slight profit in the medium-term, which might at least suggest that our model was better than random guessing.

While this is not a terribly amazing result, our results are not too bad given the state of the literature regarding statistical learning methods specifically in a sports betting context. Considering the fact that the final model largely only uses derivatives of the scores of games as features, we think that there is a lot of promise in the general methodology described in this paper. The derivations and theory from the sports betting domain is very useful in constructing a model like this, and if we could improve the feature set, then we think that it could greatly improve the results of our model in a prediction as well as a betting context. Once again, the main statistics used in the model were quite basic, and we would argue that access to more detailed and granular statistics is one of the main hurdles for this effort. The two main improvements that we could make to the feature set are to include more game-by-game statistics as well as more advanced metrics that rely on location-level data.

Finally, even with a relatively accurate model for predicting results, there are many aspects to sports betting that are nonscientific in nature. This means that a good model is not the end-all be-all. It is extremely difficult to quantify the effects that a manager change, an injury to a star player, or other non-statistical factors might have on the result of a game. In addition, sports in general have a huge amount of variance relative to other domains. Thus, it would appear that both a good statistical model as well as knowledge about betting strategy and sports in general are necessary to succeed in this field. All in all, we believe that if nothing else, our results, particularly in context with the current state of the literature, point to the difficulty of profiting long-term in sports betting even with advanced statistical methodology.

References

Baboota, Rahul, and Harleen Kaur. 2019. “Predictive Analysis and Modelling Football Results Using Machine Learning Approach for English Premier League.” *International Journal of Forecasting* 35 (2). Elsevier: 741–55.

Brant, Rollin. 1990. “Assessing Proportionality in the Proportional Odds Model for Ordinal Logistic Regression.” *Biometrics*. JSTOR, 1171–8.

Chu, Dani, Yifan Wu, and Tim B Swartz. 2018. “Modified Kelly Criteria.” *Journal of Quantitative Analysis in Sports* 14 (1). De Gruyter: 1–11.

Football-Data. 2020. “Data Files:England.” Dataset. https://www.football-data.co.uk/englandm.php.

Hubáček, Ondřej, Gustav Šourek, and Filip Železny. 2019. “Exploiting Sports-Betting Market Using Machine Learning.” *International Journal of Forecasting* 35 (2). Elsevier: 783–96.

Kaunitz, Lisandro, Shenjun Zhong, and Javier Kreiner. 2017. “Beating the Bookies with Their Own Numbers-and How the Online Sports Betting Market Is Rigged.” *arXiv Preprint arXiv:1710.02824*.

Kelly, John L, Jr. 2011. “A New Interpretation of Information Rate.” In *The Kelly Capital Growth Investment Criterion: Theory and Practice*, 25–34. World Scientific.

Marek, RU, and Martin Chovanec. 2019. “The Application of the Machine Learning Principles in the Sports Betting Systems.” *Acta Electrotechnica et Informatica* 19 (3): 16–20.

Nalla, Zaeem. 2018. “Premier League.” Dataset. https://www.kaggle.com/zaeemnalla/premier-league/version/1.

Peng, Chao-Ying Joanne, Kuk Lida Lee, and Gary M Ingersoll. 2002. “An Introduction to Logistic Regression Analysis and Reporting.” *The Journal of Educational Research* 96 (1). Taylor & Francis: 3–14.

Tibshirani, Robert. 1996. “Regression Shrinkage and Selection via the Lasso.” *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1). Wiley Online Library: 267–88.