Predicting the Outcome of a Soccer Match

And the Codifying the Impact of Match Events

Introduction

Soccer players are known for being great actors. Their facial expressions and body language don't equal the severity of their pain, but a first-time viewer can't tell the difference. Even experienced fans and referees can be fooled. In the flurry of a scramble, with arms flailing and legs twisting, it is difficult to see what happens. The player slides to a dramatic fall, grimaces, and clutches his hamstring in sheer agony. One minute later they are back in action. In the end, soccer players get a bad rep for it.

Why do summer league kids act tough during a game, while grown men, professional athletes, strive for the opposite?

This analysis doesn't strive to define the acting abilities of professional soccer players. Instead, it attempts in-part to codify a phenomenon that soccer players seem to know intuitively – a single card may affect the outcome of a match. More broadly, this analysis attempts to (a) gauge the impact of match events on a match outcome and (b) find a model that predicts match outcome with greater accuracy than random guessing or simple statistics.

Industry Impacts

Accurately gauging these events can affect several industries. Most notably, it can affect a team's draft picks and player contracts by monetizing which skills are most valuable. In the gambling world, it can affect the odds that betting houses and gambling websites place on matches. In a further reaching sense, it might also affect post-game entertaining staffing. A win means more celebration at home after the game.

For the purposes of this study, we will focus on betting houses, as the impact of wins and other game events is most immediately felt by their business.

Introduction to the Problem

We have limited ability to accurately forecast the outcomes of soccer matches. Paradoxically, the unpredictability of soccer matches makes the business of betting possible. Gamblers try their luck at predicting outcomes, and betting houses set odds that are profitable.

Fortunately, detailed data is available for soccer matches, from goals and penalties per match to individual player attributes. Part of the problem is to determine which data are significant and which are not.

Data

The data originally comes as an SQLite database with eight tables, which are converted into eight separate data frames:

Player	**team***	**league***	country
Playerattributes	teamattributes	**match***	Sqlitesequence

Figure 1

Our analysis primarily focuses on the bolded table names. The other data frames contain useful information, such as which league ID's belong to which country. However, they are small and require little analysis. The focus data frames are larger and require restructuring.

The most important of these data frames is 'match'. It contains individual match information, such as number of goals scored, fouls committed, and length of ball possession.

Data Frame: 'match'

The variables in match can be divided into four categories – Base Data, Player Data, XML Data, and Betting House Odds. We used the **Base Data** to do a high-level exploration. We used **Base Data and XML Data** to do further analysis. Restructured and calculated data is asterisked* and in **bold**:

Base Data	Player Data	**XML Data**	Betting House Odds

country_id	league_id
Season	stage
Date	match_api_id
home_team_api_id	away_team_api_id
home_team_goal	away_team_goal
goalDiff*	**points***

goal	**goalDiff***	
shoton	**shotson***	**oppShotson***
shotoff	**shotsoff***	**oppShotsoff***
foulcommit	**fouls***	**oppFouls***
card	**Ycards***	**oppYcards***
	Rcards*	**oppRcards***
cross	**crosses***	**oppCrosses***
corner	**corners***	**oppCorners***
possession	**poss***	**oppPoss***

Figure 2

Base Data:

Our highest-level analysis comes from this data frame. The dataframe 'statsMB3' contains only these variables, plus two calculated variables - goalDiff and points. It is used to show that a higher goal differential correlates to more points accumulated in a season. Because a better goal differential results in a better season outcome, we continue by focusing on what causes a better goal differential, which is the difference between goals scored and goals permitted.

XML data: The XML data allows us to focus on what causes a better goal differential. These variables possess details of shots and gameplay. Because the number of shots and plays can vary from game to game, it is stored in nested data frames which are not easily read by R. We later explain the packages and techniques necessary to extract the desired data from here.

Player Data: This data shows the position on of first, second, and third string players. The variables are coordinates on the plane of a soccer field. This was not used for our analysis.

Betting house odds: These numbers show the odds placed on games by various betting houses. Because we want to find factors contributing to wins ourselves, we do not use these numbers in our analysis.

Limitations

This is a relatively involved data set. It has 198 variables between 8 data frames, not including calculated fields or data within nested XML data frames.

While any data set's limitations can be defined by the data it does not have, this data set is a case where the limitations are better ascribed to the analyst's creativity and by data that exist but aren't complete for all observations. For example, the match data frame has ~25,000 rows. However, only 14,217 of those rows contain XML data, and 8,124 of those rows contain complete XML nested data. Because certain seasons lack complete XML data, this might restrict seeing higher-level trends.

While this data set has the benefit of already being rich, it could be further improved by including referee and external event data. For example, it does not contain information on league rules, match referees, or external events, such as league policy changes or weather during the soccer matches. Finally, no documentation exists to explain the variable names.

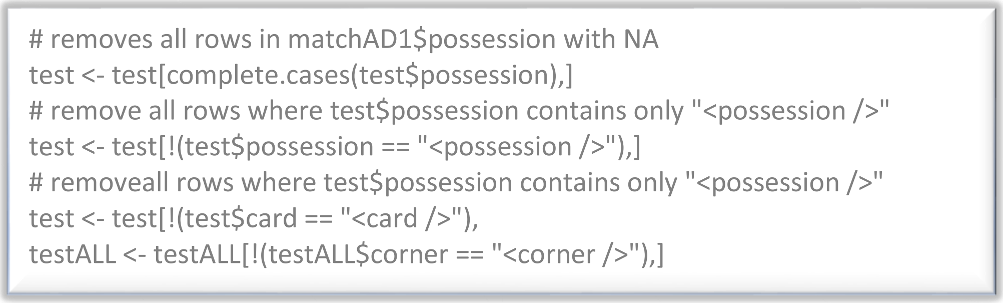
Cleaning and Wrangling

This data set is intended to thwart practitioners of R. It is stored in SQLite format, and many interesting variables are stored in nested XML data frames within individual observations. This requires a combination of R packages - RSQLite, DBI, XML, dplyr, and magrittr.

The initial transformation involves reading the SQLite data into R and creating a variable for each table. R plays nicer with csv files, so we wrote each table to a csv file that was stored in the project folder. We then created a variable for each csv file and read it back into R. This allowed us to move forward with base R and common R packages.

At this point, it is easy to select the non-XML data with dplyr to create simple plots and regressions. Extracting the desired data from the XML data is more involved. It requires converting the XML data into a data frame and extracting the desired data with the following process:

1. Use dplyr to create a sub-data frame with the desired columns
2. Remove all rows with incomplete data:



```
# removes all rows in matchAD1$possession with NA
test <- test[complete.cases(test$possession),]
# remove all rows where test$possession contains only "<possession />"
test <- test[!(test$possession == "<possession />"),]
# removeall rows where test$possession contains only "<possession />"
test <- test[!(test$card == "<card />"),
testALL <- testALL[!(testALL$corner == "<corner />"),]
```

Figure 3

3. Load library(XML)
4. Create for-loops that extract the necessary data from XML variables and place them into new variables. All for-loops contain elements of the dplyr package, the XML package, and base R conditional coding.
    a. Some XML columns do not convert into clean data frames. Many data values are placed in the wrong column. This may or may not be a result of the parser. However, we mitigate this within the for-loop for all variables (except "possession") by using "|". Given additional time, it would be prudent to find a more efficient parser.
    b. The "possession" variable also produces incorrect variables, and the output of the for-loop contains excess characters that need to be removed. We ameliorate both of these issues outside of the for loop using substr() and magritrr().
    c. Six for-loops require a conditional statement, because some observations have a missing column.

Preliminary exploration

Before unpacking the XML data, we create various plots to see if teams with more points or goals proceed to further stages.

```
ggplot(data = statsMB3, aes(spoints, statID, col = factor(maxStage))) + geom_point() +
  labs(title = "Points and Maximum Stage") + theme(plot.title = element_text(hjust = 0.5)) +
  labs(x = "Points", y = "Teams")
```

Figure 4

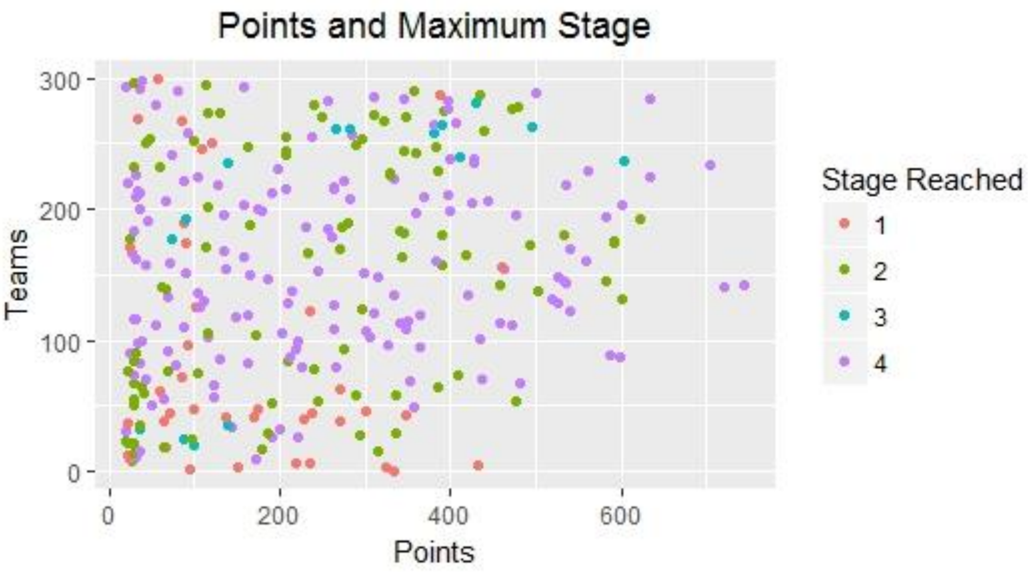


Figure 5

We include this chart here because we originally intended to analyze factors that contributed to stage progression, as described in the milestone report. On further analysis, we found that the “stage” variable represented the match number in the season. For example, the teams in “Stage 3” were playing their third match of the season. The seasons varied in length from 30 matches to 38 matches (half at home and half away) for each league. Thus, there was no “stage progression.”

While this nullifies any stage-progression style analysis, it still allows us to compare leagues with shorter seasons vs leagues with longer seasons. The following chart shows that teams with longer seasons tend to score more points (as expected), but it also shows that the dispersion of points accumulated by teams across leagues appears similar:

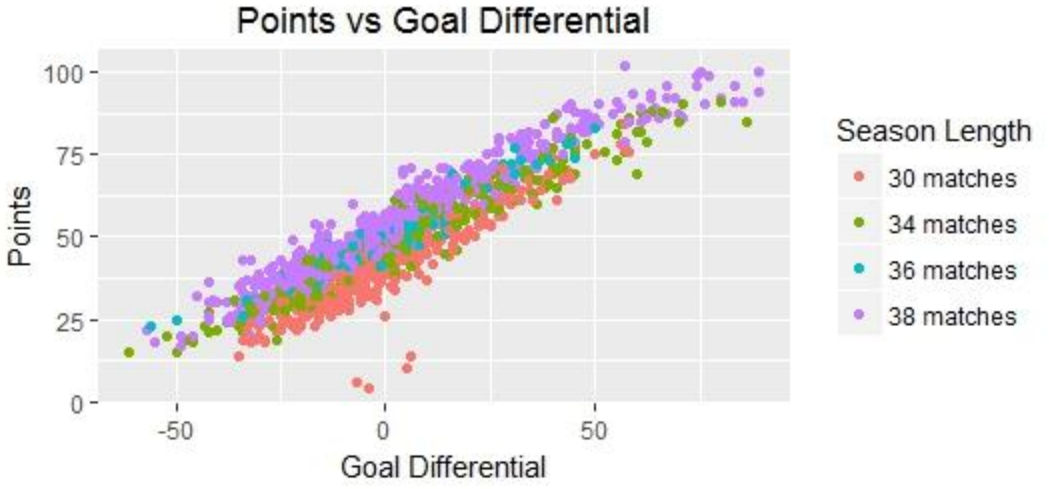


Figure 6

This chart shows expected data – more goals scored (and fewer goals permitted) lead to more wins, which directly leads to more points acquired.

The interesting part begins here. Why do certain teams score more goals, win more games, and acquire more points? This is where the nested XML data enters the analysis.

Starting with summary statistics, we can see how many games are played and how events are distributed between home and away teams. The following figure shows the distribution of these events.

	Games	Wins	Ties	Loss	Goals	Poss%	ShOn	ShOff	Crosses	CKicks	Fouls	YCards	RCards
Home	8124	3741	2054	2329	12697	48.77	50028	50739	149923	47374	99654	14371	442
% T	50%	61.6%	50%	39%	57.3%	48.7%	55.6%	55.5%	56.1%	56.3%	48.9%	45.5%	46.8%
Away	8124	2329	2054	3641	9471	51.23	39995	40662	117108	36819	104221	17220	503
% T	50%	38.4%	50%	61%	42.7%	51.2%	44.4%	44.5%	43.9%	43.7%	51.1%	54.5%	53.2%

Figure 7

This figure highlights what is commonly known as "home team advantage." 61.6% of wins are acquired by the home team. We see further that the home team typically scores more goals, and gets more shorts, crosses, and corner kicks than the away team. We see a few additional interesting numbers:

1) The home team has less ball possession than the away team
2) The away team gets 20% more yellow cards than the home team → (17,220-14371)/14371
3) The away team gets 14% more red cards than the home team → (503-442)/442

Figure 8 shows the same data, but split up amongst winning, losing, drawing teams:

	#Matches	Home	Away	Ycards	Rcards	Fouls	Goals	Crosses	c-kicks	ShOn	ShOff
Wins	6070	3741	2329	10706	185	73779	14610	96495	32558	35851	35483
% of total	74.7%	46.0%	28.7%	33.9%	19.6%	36.2%	65.9%	36.1%	38.7%	39.8%	38.8%
Draws	2054	2054	2054	8317	202	52918	4104	69993	21657	22909	23442
% of total	25.3%	25.3%	25.3%	26.3%	21.4%	26.0%	18.5%	26.2%	25.7%	25.4%	25.6%
Losses	6070	2329	3741	12568	558	77178	3454	100543	29978	31263	32476
% of total	74.7%	28.7%	46.0%	39.8%	59.0%	37.9%	15.6%	37.7%	35.6%	34.7%	35.5%
Totals	8124	8124	8124	31591	945	203875	22168	267031	84193	90023	91401

Figure 8

While the home teams win fewer than half of the matches (46%), the away teams win only 29% of the matches. We see some additional expected figures. Winning teams outscore and outplay losing teams, as shown by goals, crosses, corner kicks, shots on goal, and shots off goal. Winning teams receive 34% of total yellow cards, while losing teams receive 40% of total yellow cards. Even more stark is red cards – winning teams receive 20% of total red cards, while losing teams receive 59% of total red cards.

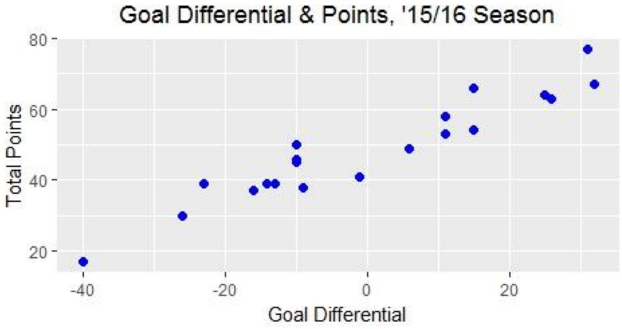
While stark, the numbers make sense. When a team receives a red card, they continue the match with a player down. Also, if the foul happens within the penalty box by the defending team, the attacking team is awarded a penalty shot. It makes sense that a team disadvantaged by a red card may have reduced chances of winning. However, further analysis would explore whether any bias exists in carding. Is it a result of aggressive play, or it a result of something else, such as referee favoritism or home field advantage?

Another interesting point can be gleaned from ratios here. Out of 8,124 matches total, 3,026 of them (or 37%) were decided by 1 goal. Out of 945 total red cards, 358 of them (38%) were awarded in a match that was determined by one goal. This seems to show that regardless of whether a match is a close victory or not, red cards appear to be evenly awarded (for this metric).

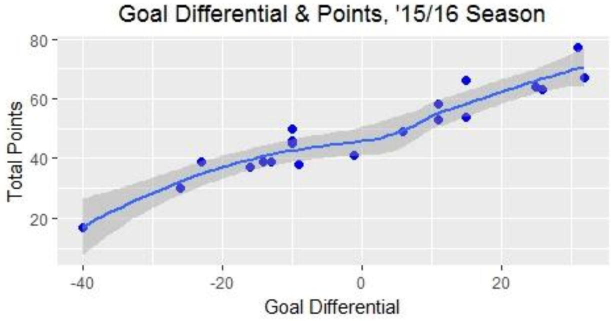
However, in matches determined by 1 goal that also have a red card, the red card is given to the losing team 68% of the time (243 matches). This compares to the overall total, where in 59% of matches with a red card, it is given to the losing team.

Several visuals can help show whether this is a trend between leagues and seasons.

To begin, we use the English Premier league for the 2015/2016 season as an example. Figures (9) and (10) show the relationship between goal differential and points acquired. Each point represents a team. The scatterplots naturally place the teams in rough order from left to right by those which won the fewest matches to those which won the most matches.

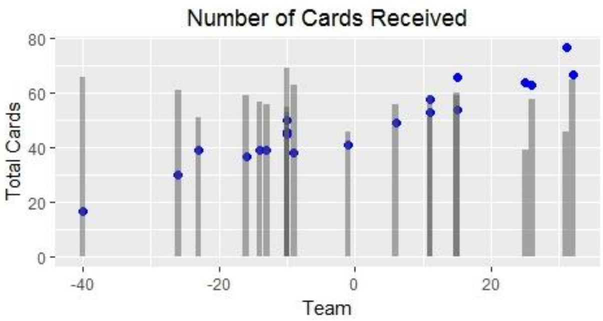


(9) Goal Differential vs Points Scored

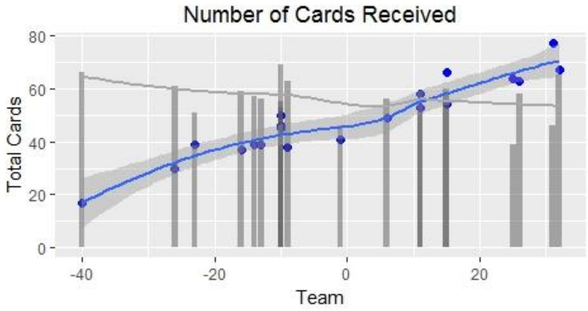


(10) With trendline

Figures (11) and (12) superimpose number of cards awarded to each team, where each bar is placed in line with its corresponding team's point:



(11) Superimposed bar chart of cards awarded to each team



(12) With trendlines

These plots show us that higher winning teams receive fewer yellow and red cards than lower winning teams in the English Premier League.

The power of data allows us to see if the trends displayed in this league and season are an isolated occurrence, or if this happens on a broader scale. The following figure shows the 2015/2016 season again. Instead of just the English Premier League, it shows all leagues with available data.

Again, these plots naturally arrange the teams roughly from least winning (left) to most winning (right). The dark blue trendline shows the relationship between goal differential and wins. The light blue trendline shows the carding trend of teams from least to most winning.

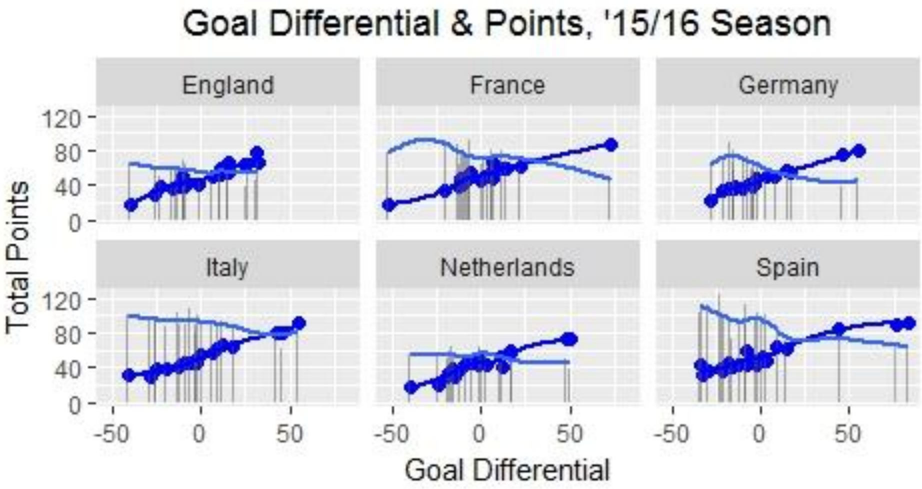


Figure 13

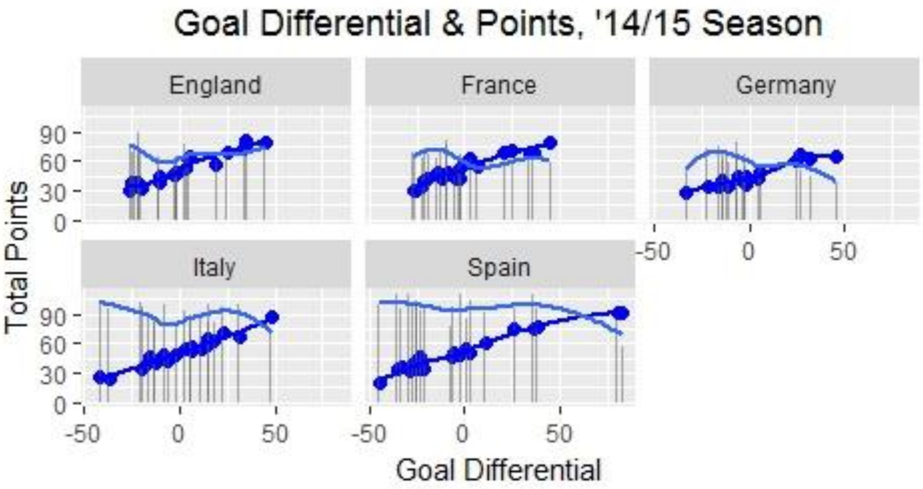


Figure 14

It seems to be a clear relationship. We see this same relationship in other seasons too. Figure 14 shows this relationship for available data in the 2014/2015 season.

We cannot say whether winning teams have less aggressive players or referees who call in their favor, but it warrants further investigation and special attention in the later analysis.

Do other factors show a similar relationship? We've included charts for remaining variables. In this case, we've ordered the teams along the x-axis from least to most winningest, and plotted the corresponding point on the y-axis.

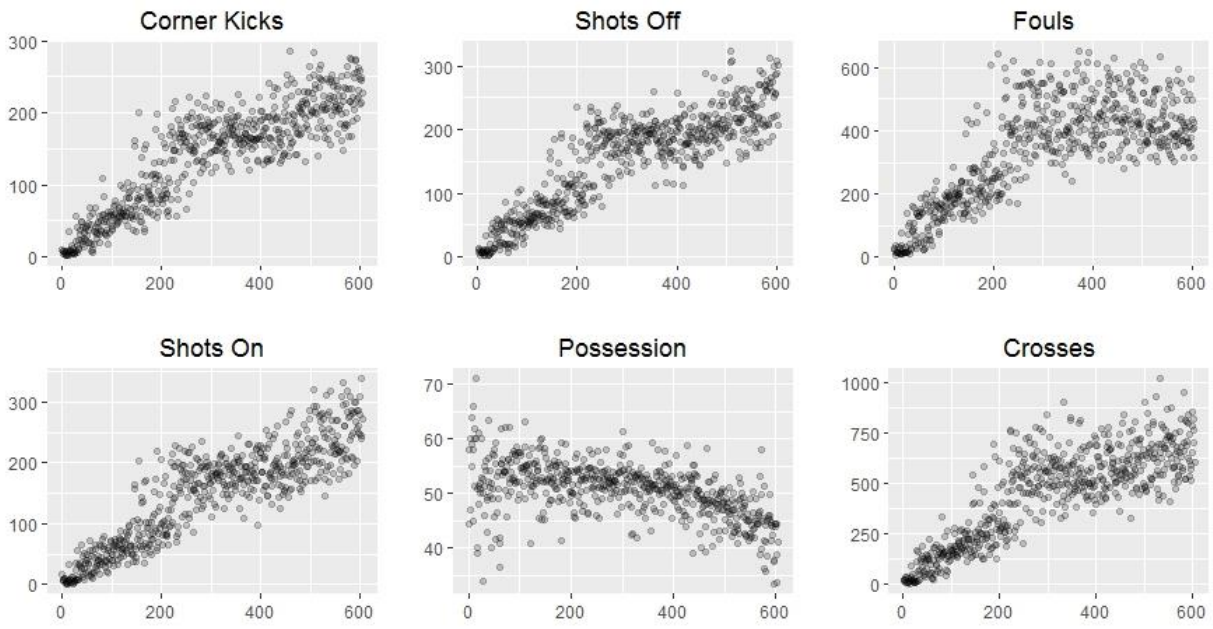


Figure 15

Each point represents a unique season-team. Some of these plots are expected – teams that produce more chances for goals tend to win more games. Interestingly, this does not seem to be the case with ball possession. It shows that teams who win more have less ball possession time.

By how much do these variables affect match outcome? What is the effect of yellow and red cards, and is ball possession significant? This is where the benefit regression analysis appears. It can show which variables are significant, and by how much they might affect a match.

Because match outcome is measured in the number of goals a team scores minus the number of goals they permit, we run a linear regression where the independent variables are measured by their effect on goal differential. An added benefit of our data is that it is panel or longitudinal in style. This means that we can use earlier seasons as our training data and a subsequent season as our test data.

Season	2008/09	2009/10	2010/11	2011/12	2012/13	2013/14	2014/15	2015/16
# of Obs	1502	1714	1982	1904	716	724	3572	4134

Figure 16

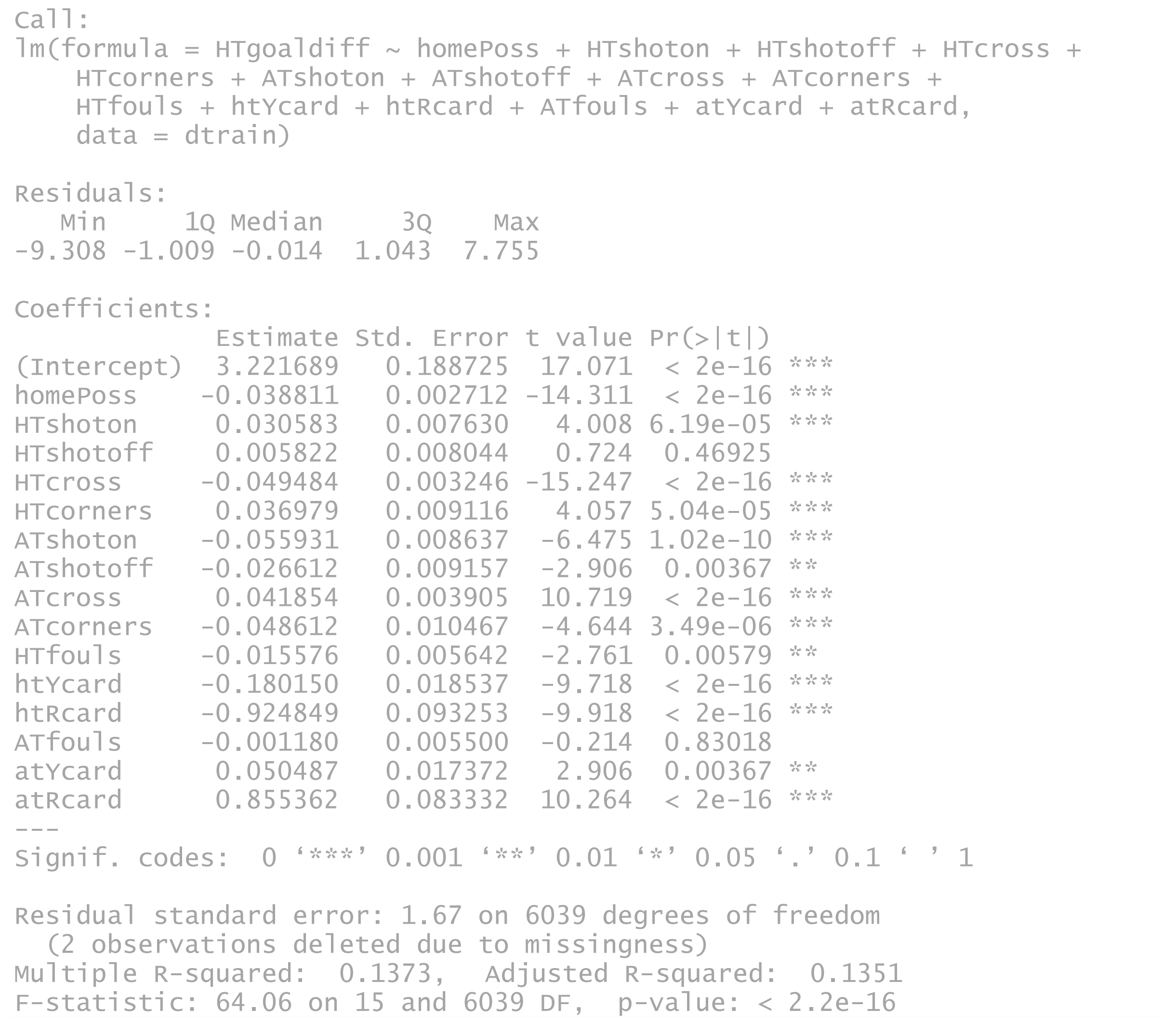
We first check for correlation and potential multicollinearity by checking if any variables correlated with each other using the following code:

```
dragon_numeric <- dragon[, sapply(dragon, is.numeric)]
cor(dragon_numeric)
```

Figure 17

We find that the only highly correlated variables are "goals," "points," and "goal differential." This makes sense, since they are directly derived from each other. We chose goal differential as the dependent variable. There were other variables that had some correlation, but nothing approaching concern. For example, home team shots on goal correlated with corner kicks (43%) and crosses (36%). This is expected, as the intention of both plays are intended to set up a shot on goal.

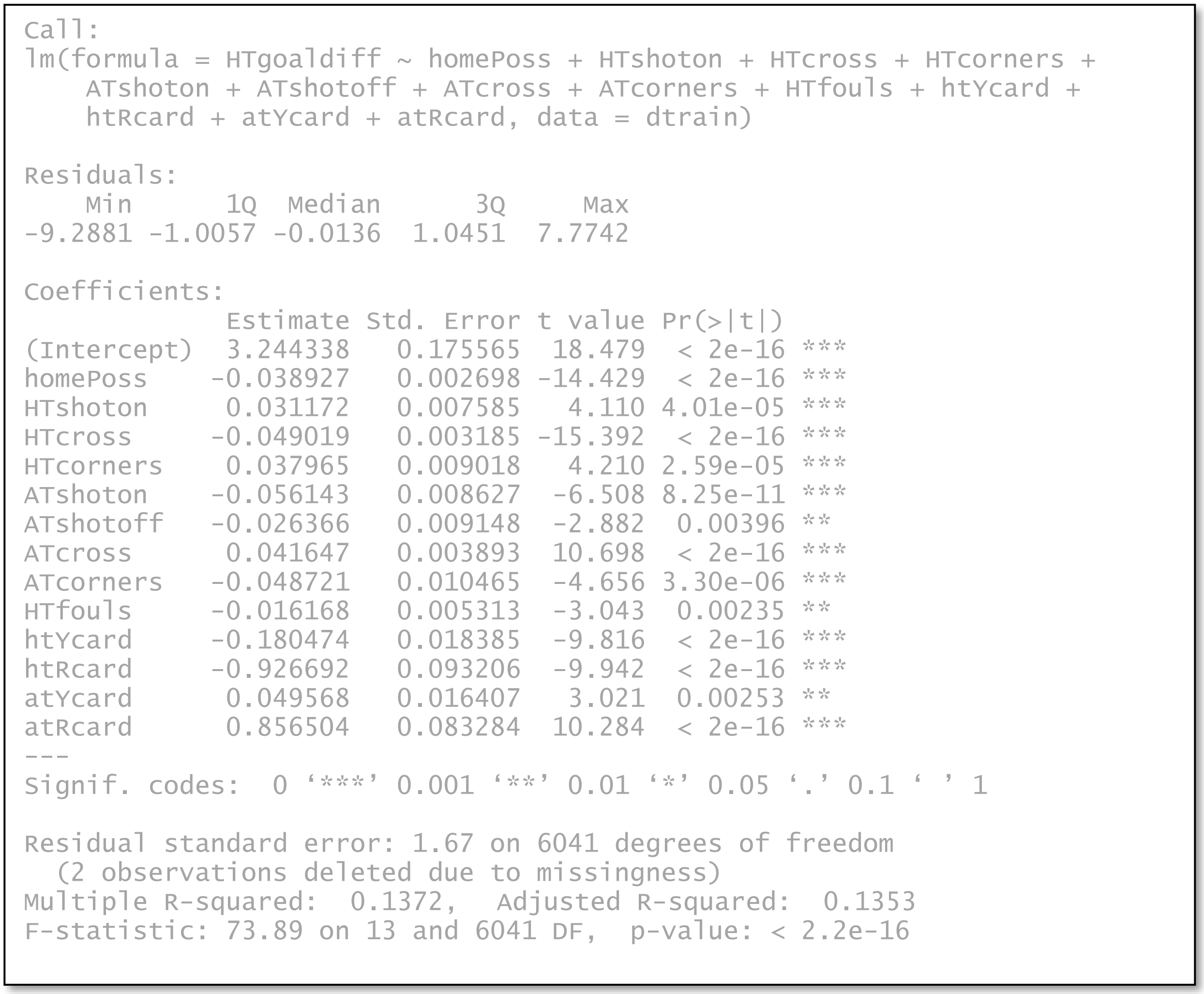
We take seasons 2008/09 to 2014/15 as our training set. We set our dependent variable as goal differential and regress it on the available variables. The benefit of regressing against goal differential is that this outcome variable is a result of both home and away team goals.



```
Call:
lm(formula = HTgoaldiff ~ homePoss + HTshoton + HTshotoff + HTcross +
    HTcorners + ATshoton + ATshotoff + ATcross + ATcorners +
    HTfouls + htYcard + htRcard + ATfouls + atYcard + atRcard,
    data = dtrain)

Residuals:
   Min     1Q Median     3Q    Max 
-9.308 -1.009 -0.014  1.043  7.755 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.221689   0.188725  17.071  < 2e-16 ***
homePoss    -0.038811   0.002712 -14.311  < 2e-16 ***
HTshoton     0.030583   0.007630   4.008 6.19e-05 ***
HTshotoff    0.005822   0.008044   0.724  0.46925    
HTcross     -0.049484   0.003246 -15.247  < 2e-16 ***
HTcorners    0.036979   0.009116   4.057 5.04e-05 ***
ATshoton    -0.055931   0.008637  -6.475 1.02e-10 ***
ATshotoff   -0.026612   0.009157  -2.906  0.00367 ** 
ATcross      0.041854   0.003905  10.719  < 2e-16 ***
ATcorners   -0.048612   0.010467  -4.644 3.49e-06 ***
HTfouls     -0.015576   0.005642  -2.761  0.00579 ** 
htYcard     -0.180150   0.018537  -9.718  < 2e-16 ***
htRcard     -0.924849   0.093253  -9.918  < 2e-16 ***
ATfouls     -0.001180   0.005500  -0.214  0.83018    
atYcard      0.050487   0.017372   2.906  0.00367 ** 
atRcard      0.855362   0.083332  10.264  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.67 on 6039 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.1373,	Adjusted R-squared:  0.1351 
F-statistic: 64.06 on 15 and 6039 DF,  p-value: < 2.2e-16
```

Model 1

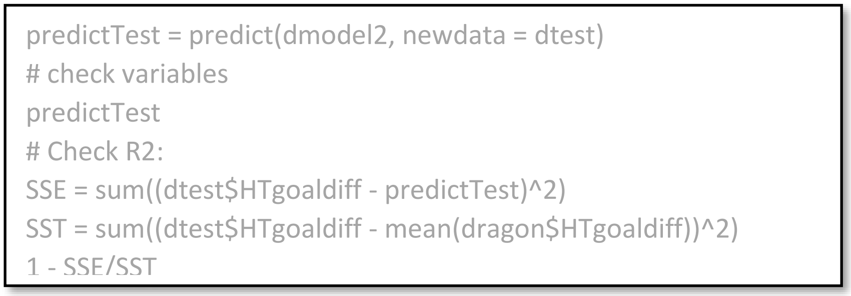
Because home team shots off goal and away team fouls appear to have no significance on the model, we remove these variables and try again.



```
Call:
lm(formula = HTgoaldiff ~ homePoss + HTshoton + HTcross + HTcorners + 
    ATshoton + ATshotoff + ATcross + ATcorners + HTfouls + htYcard + 
    htRcard + atYcard + atRcard, data = dtrain)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.2881 -1.0057 -0.0136  1.0451  7.7742 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.244338   0.175565  18.479  < 2e-16 ***
homePoss    -0.038927   0.002698 -14.429  < 2e-16 ***
HTshoton     0.031172   0.007585   4.110 4.01e-05 ***
HTcross     -0.049019   0.003185 -15.392  < 2e-16 ***
HTcorners    0.037965   0.009018   4.210 2.59e-05 ***
ATshoton    -0.056143   0.008627  -6.508 8.25e-11 ***
ATshotoff   -0.026366   0.009148  -2.882  0.00396 ** 
ATcross      0.041647   0.003893  10.698  < 2e-16 ***
ATcorners   -0.048721   0.010465  -4.656 3.30e-06 ***
HTfouls     -0.016168   0.005313  -3.043  0.00235 ** 
htYcard     -0.180474   0.018385  -9.816  < 2e-16 ***
htRcard     -0.926692   0.093206  -9.942  < 2e-16 ***
atYcard      0.049568   0.016407   3.021  0.00253 ** 
atRcard      0.856504   0.083284  10.284  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.67 on 6041 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.1372,	Adjusted R-squared:  0.1353 
F-statistic: 73.89 on 13 and 6041 DF,  p-value: < 2.2e-16
```

Model 2

With this model, we have fewer variables and a slightly higher Adjusted R-squared. We can test this model accuracy for out of sample data. We do this by bringing in the data from the testing set (2015/2016 season) to see if the model maintains integrity:



```
predictTest = predict(dmodel2, newdata = dtest)
# check variables
predictTest
# Check R2:
SSE = sum((dtest$HTgoaldiff - predictTest)^2)
SST = sum((dtest$HTgoaldiff - mean(dragon$HTgoaldiff))^2)
1 - SSE/SST
```

Figure 18

The model from our training set has an Adjusted R-squared of 0.1353. When we apply this model to the testing set, the R^2 is 0.1009. Given that this data attempts to measure human behavior, it is not surprising that any model we attempt has a low R^2, even if the variables in the model are statistically significant. Perhaps a better measure of model reliability in this case is the Sum of Squared Errors (SSE) or the Mean Squared Error (MSE). Figure 19 shows several models that we ran, the SSE, the MSE, the Adjusted R^2, and the Adjusted R^2 for the testing set. The MSE is also a more interpretable value, as it has the same unit of measurement as the dependent variable (goals).

Model	Variables	Adj-R^2	Test Adj-R^2	SSE	MSE
1	homePoss, HTshoton, HTshotoff, HTcross, HTcorners, ATshoton, ATshotoff, Atcross, ATcorners, HTfouls, htYcard, htRcard, ATfouls, atYcard, atRcard	.1351	.1016596	6194	1.731
2	homePoss, HTshoton, HTcross, HTcorners, ATshoton, ATshotoff, Atcross, ATcorners, HTfouls, htYcard, htRcard, atYcard, atRcard	.1353	.1009683	6198	1.732
3	homePoss, HTshoton, HTcross, HTcorners, ATshoton, ATshotoff, Atcross, ATcorners, HTfouls, htYcard, htRcard, atYcard, atRcard, (HTshoton*HTcorners), (ATshoton*ATcorners)	.1363	0.1015945	6195	1.731
4	homePoss, HTshoton, HTcross, ATshoton, ATshotoff, ATcross, HTfouls, htYcard, htRcard, atYcard, atRcard	.1303	.1016003	6193	1.730
5	HTshoton, ATshoton, HTcard, ATcard	.05449	.040111	6619	1.789
6	HTshoton, ATshoton, htYcard, htRcard, atYcard, atRcard	.07194	.058187	6494	1.772

Figure 19

> Note: We included a model that combines yellow and red cards for each team to see its impact on SSE, MSE, and Adjusted R^2. In addition to being slightly less accurate in all areas, the model loses the impact that red and yellow cards individually can have on a match.

The models above have similar Adjusted R^2 values, all which tend to carry over well to the test set, and they all have similar MSE. In such a case, we prefer to take a simpler model. Model 4 had the lowest MSE and is one of the simplest. This leads us to prefer Model 4, which has the lowest MSE, or Model 6, which controls for yellow and red cards separately and is simpler yet. The formulas for these models are:

Model 4

$$a_4 = 3.339 + -.040(x_1) + .039(x_2) + -.041(x_3) + -.066(x_4) + -.031(x_5) + .031(x_6)+ -.016(x_7) + -.18(x_8) + -.908(x_9) + .050(x_{10}) + .844(x_{11})$$

Model 6

$$a_6 = 0.817 + .029(x_1) + -.066(x_2) + -.224(x_3) + -.894(x_4) + .064(x_5) + .833(x_6)$$

It is not surprising that these variables all have some level of contribution to a match. As much as soccer may be a game of skill, it is also a game of odds. The more shots on goal a team has, the more opportunities they create to score. The more cards a team receives, the higher the chance they lose a player and play at a disadvantage.

Next, we run several logistic regressions. We use win = 1 / not win = 0 and draw = 1 / not draw = 0 as the dependent variables. Before conducting any regressions, we establish several baseline models.

The filtered data set we use has 16,248 observations:

Wins	Draws	Losses	Total
6,070	4,108	6,070	16,248
.3736	.2528	.3736	1

Baseline Model 1

That means that if we always guess win for a team, we will be right 37.36% of the time. If we always guess draw, we will be right 25.28% of the time. We can improve the accuracy of our baseline model by controlling for whether the team is playing at home or away.

Home Wins	Away Wins	Draws	Home Loss	Away Loss	Draws	Total
3741	2329	2054	2329	3741	2054	8,124
.4605	.2867	.2528	.2867	.4605	.2528	1

Baseline Model 2

In this case, if we always guess “win” for the home team (or loss for the away team), we will be right 46.05% of the time. If we always guess draw for the home (or away) team, we will be right 28.67% of the time. Both figures are slightly higher than the previous baseline model.

In creating the logistic regression model, our goal is to predict a win more than 46% of the time for a home team and draw more than 28.67% of the time. The logistic regression model allows for a binary outcome – 1 or 0. However, a team has three potential match outcomes – win, loss, or draw. We mitigate this by creating the logistic regression models in pairs – a model for win and a model for draw. A model that regresses on “win” will give us the probability of a win. A model that regresses on “draw” will give us the probability of a “draw.” We can roughly calculate the probability of a loss by subtracting the sum of any pair of models from 1.

We created eight initial regressions models – or four model pairs. Odd numbered models have “win” as the dependent variable, and even numbered models have “draw” as the dependent variable.

Model	Dep. Var	Variables	AIC	In-Sample Accuracy	Out-of-Sample
1	win	shotson, shotsoff, crosses, corners, oppShotson, oppShotsoff, oppCrosses, oppCorners, fouls, Ycards, Rcards, oppYcards, oppRcards	14810	. 6600	.6584
2	draw	shotson, shotsoff, crosses, corners, oppShotson, oppShotsoff, oppCrosses, oppCorners, fouls, Ycards, Rcards, oppYcards, oppRcards	13553	.7497	.7397
3	win	crosses, oppCrosses, fouls, Ycards, Rcards, oppYcards, oppRcards	15448	.6389	.6471
4	draw	crosses, oppCrosses, fouls, Ycards, Rcards, oppYcards, oppRcards	13548	.7497	.7397
5	win	shotson, shotsoff, crosses, oppShotson, oppShotsoff, oppCrosses, Ycards, Rcards, oppYcards, oppRcards, home_or_away	14683	.6691	.6684
6	draw	shotson, shotsoff, crosses, oppShotson, oppShotsoff, oppCrosses, Ycards, Rcards, oppYcards, oppRcards, home_or_away	13573	.7497	.7397
7	win	crosses, oppCrosses, Ycards, Rcards, oppYcards, oppRcards, home_or_away	15011	.6559	.6572
8	draw	crosses, oppCrosses, Ycards, Rcards, oppYcards, oppRcards, home_or_away	13569	.7497	.7397

Figure 20

The out-of-sample predictive capabilities for these models seem very high, especially for draws. However, this makes perfect sense when comparing against the baseline models. In Baseline Model 1, we can either always predict "win" or always predict "not win." (we can do the same for draw or loss). That means that if we always predict "win," we will be right 37.36% of the time, and if we always predict "not win," we will be right 62.64% of the time. The win/not win models in Figure 20 (1, 3, 5, and 7), are all slightly higher. Model 5 has the highest overall in and out-of-sample accuracy, at 66.91% and 66.84, respectively. This makes it stronger than both baseline model 1 and baseline model 2 for overall accuracy.

The capabilities of these models to predict a draw are almost identical to the baseline models. Because the of this, if the win/not win model predicts "not win," we will always predict "loss," regardless of which sister model we select. (If model 5 predicts "not win," then we would default to model 6. Having practically no different from the baseline model, we do best to predict "loss.")

Because of this, we can in treat the dependent variable in model 5 essentially as a "win/loss" variable.

We create several additional models using data where the outcome variable was "home team win / not home team win" and "home team draw / not home team draw" to see if we could improve on our overall accuracy. The following eight models (four model pairs) show the variables for each model, along with AIC, in-sample overall accuracy, and out-of-sample overall accuracy:

Model	Dep Var	Variables	AIC	In-sample Accuracy	Out-of-Sample
9	HTwin	HTshoton, HTshotoff, HTcross, HTcorners, ATshoton, ATshotoff, ATcross, ATcorners , HTfouls, htYcard, htRcard, ATfouls, atYcard, atRcard	7810.3	.6280	.6309
10	HTdraw	HTshoton, HTshotoff, HTcross, HTcorners, ATshoton, ATshotoff, ATcross, ATcorners , HTfouls, htYcard, htRcard, ATfouls, atYcard, atRcard	6773.1	.7495	.7397
11	HTwin	HTcross, ATcross, HTfouls, ATfouls, htYcard, htRcard, atYcard, atRcard	8011.5	.5993	.6222
12	HTdraw	HTcross, ATcross, HTfouls, ATfouls, htYcard, htRcard, atYcard, atRcard	6770	.7497	.7397
13	HTwin	HTcross, ATcross, htYcard, htRcard, atYcard, atRcard	8029.4	.5947	.6270
14	HTdraw	HTcross, ATcross, htYcard, htRcard, atYcard, atRcard	6782.5	.7497	.7397
15	HTwin	htYcard, htRcard, atYcard, atRcard	8151.8	.5769	.5955
16	HTdraw	htYcard, htRcard, atYcard, atRcard	6805.8	.7497	.7397

Figure 21

Note: Baseline Model 2 predicts “win” with accuracy of .4605 and “draw” with accuracy of .2528.

While this set of logistic regressions outperforms the baseline models, they are slightly weaker than the previous set of regression models. The predictive capabilities for the draw/not draw remain equal to almost all the first set of draw/not draw logistic regressions models, and the predictive capabilities for win are slightly weaker. This makes sense, given that the first set of logistic regression models attempt to include both home and away game events in the independent variables.

Our strongest of all the above models is the Model 5/6 pair from Figure 20. We have included its out-of sample confusion matrix below, along with the confusion matrix for its sister model (Model 6). This illustrates that the independent variable for Model 5 can essentially behave as a win/not win variable.

Outcome	Predicted	
Actual	FALSE	TRUE
0	2,220	385
1	986	543

Figure 22 (Model 5: win/not win)

Outcome	Predicted	
Actual	FALSE	TRUE
0	9,082	0
1	3,032	0

Figure 23 (Model 5: draw/not draw)

In Model 5, we correctly predict “win” 543 times. 385 times we predict a win, but the outcome is a draw or loss. We predict “not win” 3,206 times, and we are correct 2,220 of those times.

This gives us an out-of-sample sensitivity of .3551. More importantly, it gives us overall accuracy of (2,220+543)/(2,220+385+986+543) = 66.84%.

Recommendation

Because our goal is to predict outcome of a soccer match at a profitable rate, we need to do it beyond random guessing, and we need to do it beyond simple statistics. To both those ends, we have succeeded. For games of chance that involve betting on the simple match outcome (win/loss/draw), these models will perform more strongly than the average gambler.

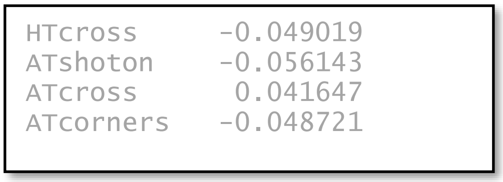
In an academic setting, we might point out how little the AIC and mean outcome predictions change for the model pairs and thus argue for a simpler model. However, because every little improvement in the model means additional revenue for the firm, we opt in favor of a more complex model pair if necessary. In Figure 20, Models 5 and 6 have the highest prediction accuracy.

For games of chance that involve betting on the finer details of a game, such win or loss goal differential, we turn to the linear regressions shown earlier.

Of all variables in the preferred linear regressions (Linear Models 4 and 6), yellow and red cards have the largest coefficients. While the other variables are all statistically significant, it is worth paying close attention to cards. They occur less commonly than the other variables, but the introduction of a card to the game may signal a significant shift in the game outcome. For example, a home team red card is equal to 0.93 of an away team goal (0.18 for a yellow card), and an away team red card is roughly equal to 0.86 of a home team goal (0.05 for a yellow card). For this reason, we recommend gathering additional data on the players who receive more cards and the referees who award more cards.

Unfortunately for betting houses, yellow cards are not extremely common, and red cards even less so. For this reason, it is worth viewing other variables.

The variables with the next four largest coefficients are:



```
HTcross     -0.049019
ATshoton    -0.056143
ATcross      0.041647
ATcorners   -0.048721
```

Figure 19

Interestingly, we see that home team crosses are negatively correlated with home team goal differential, and away team crosses are positively correlated with home team goal differential. The other two variables have the expected sign. We re-inspected the data to ensure the signs were correct, and they appear to be so. Figure 15 shows that higher winning teams tend to have more crosses. It is likely that these variables are compensating for another in the model. However, the models appear to lose strength when these variables are removed. We recommend looking further into the relationship between crosses and goals scored.

Both the linear and logistic approaches give a good start to predicting match outcomes. To improve accuracy, we need to inspect the component parts of these variables. “Component” parts include individual players, referees, or playing conditions that may contribute to corner kicks, crosses, and shots on goal. We also need to see if variations exist between various groups. Various “groups” include leagues, seasons, or time of year. Not only will this help improve the accuracy of our model, but it may also help better understand why the original model gives an unexpected sign for home team crosses (HTcross) and away team crosses (ATcross).

To do this, we recommend creating a more efficient XML parser. The XML cells contain much of the “component” data we seek, such as players responsible for taking shots on goal or receiving a red card. A more efficient parser will allow us to quickly pull this data and model it. Second, because of the impact of yellow and red cards on a match, we suggest investing resources in collecting this data that we do not have, specifically on the referees and their carding tendencies. Knowing the probability of a match’s yellow and red card outcome may prove to be almost as valuable as knowing the match outcome itself.

Finally, gathering these finer details may also help the logistic regression models. Being able to correctly identify a “draw” will help improve the logistic regression model pairs.