Leveraging a Change in Odds to Hedge Sports Bets Live

Simon Lam, Nico Manzonelli, Jeffery Mayolo, Joseph Zuccarelli

December 13, 2021

1 Introduction

1.1 Motivation

In the past, sports gamblers had to place bets at the casino or with a local bookie operating illegally. Now, with recent relaxations in gambling laws across the United States and the emergence of online sports books, sports betting is more accessible than ever. Gamblers can easily place bets online (in states where sports gambling is legal) before or during sporting events. The materialization of live sports betting opens up a completely new branch of opportunities to make money through gambling. Unlike ever before sports gamblers are able to observe a change in odds on their opening bets and hedge when the opportunity is right to maximize profits or minimize losses.

1.2 To Hedge or Not to Hedge?

Given that live betting is so accessible, sports gamblers have the opportunity to “hedge” almost every bet they make. One can “hedge” by placing an additional competing bet to maximize profits or minimize losses based on how the odds have changed since they placed the initial bet. These wagers are typically only relevant when betting the money-line (i.e. picking the winner of the competition outright).

After a bettor hedges their bet there are two outcomes: 1.) the initial bets wins or 2.) the competing bet (i.e. "the hedge") wins. When hedging, you want your entire payout to be equal no matter which team wins. The equations to calculate your payout are as follows:

$$P_1 = b_i * m_i - b_h \tag{1}$$

$$P_2 = b_h * m_h - b_i \tag{2}$$

where P_1 and P_2 are your payouts for outcome one and outcome two, respectively. Variables b_i and b_h represent the dollar amount of money wagered in the initial and hedge bet, respectively. Finally, m_i and m_h represent the money multiplier for the return on dollar (based on the odds) from the initial and hedged bet.

In order to ensure that one wins or loses the same amount of money no matter which team wins, they should hedge with a bet where $P_1 = P_2$. Therefore, by solving for the amount required to hedge (as a function of initial odds, initial bet and half-time odds), we get the following:

$$b_h = \frac{b_i m_i + b_i}{m_h + 1} \tag{3}$$

Whether the bet will guarantee profits or minimize losses is dependent on the direction that the line moved. The conditional $m_i > m_h$ guarantees profits, while the conditional $m_i < m_h$ minimizes losses.

1.3 Study Objectives

This study aims to address the following two main objectives. First, we analyze the association of the change in money multiplier with various game-related factors such as difference in score, number of interceptions, number of penalties, etc. Second, we build a predictive model to determine the outcome of any game at halftime given the first half statistics in order to propose a hedging strategy that could potentially optimize net gain.

2 Exploratory Data Analysis

The population under observation in this study is NFL and NCAAF games during the 2021 season. The data set used in our analysis is composed of 66 complete NFL games as well as 46 NCAAF games. Each game has a complete record with the following variables: a unique GameID for each game, the home team, the away team, date of the game, ESPN's win percentage prediction for the home team at half time, and various betting and football-related statistics for both teams–the initial money-line odds, the money line odds halftime, the halftime score, total yards, first downs, fumbles (lost), defensive sacks, defensive interceptions, and penalties against during the first half. Using these provided statistics, we also mutated a few other useful predictors: the point differential at halftime (home - away), a binary indicator for whether or not the observation is an NFL or NCAAF game, a binary indicator for whether or not the home team is considered an underdog, and the change in the home team's money multiplier from pregame to halftime. The predictor variables are listed and defined in the codebook in the appendix as well.

The response variable in our study is the change in the home team's money multiplier, which represents the difference between the home team's money multiplier at the start of the game and the home team's money multiplier at halftime. The money multiplier can be derived from the odds using the following piece-wise function:

$$m = \begin{cases} \frac{100}{|odds|} & odds < 0 \\ \frac{odds}{100} & odds > 0 \end{cases}$$

Note that for each game we collected odds from several different bookmakers; therefore, we narrowed down our response variable to a single value for each observation using the median odds across the bookmakers to calculate the money multiplier. We chose to use the median odds because not every bookmaker provided pregame and halftime odds for each game, leading to incomplete and often skewed odds. Additionally, the non-linear nature of odds would potentially cause problems with using the mean odds if a game had an even spread and a team was given favorite odds (less than -100) by one bookmaker, and underdog odds (greater than 100) by another bookmaker.

Table 1 included below provides a numerical summary of a few noteworthy variables included in the data set. All variables represent measurements that were recorded at halftime of each game. Notice that both the home and away team appear to perform similarly on average across the data set. Also notice that each variable appears to follow a relatively symmetric distribution.

Table 1: Numerical Summary (H - Home, A - Away)

	Yards (H)	**Yards (A)**	**1st Downs (H)**	**1st Downs (A)**	**ESPN Win Pred. (H)**	**Score Diff (H-A)**	**Money Mul. Change (H)**
Mean	185.90	183.80	13.41	13.21	0.53	0.96	-1.12
Median	179.00	180.50	13.00	13.00	0.54	1.00	0.00
SD	70.85	64.57	4.74	3.95	0.32	12.32	3.05
Min	44.00	48.00	4.00	3.00	0.01	-25.00	-13.31
Max	382.00	387.00	27.00	24.00	0.99	32.00	6.30

Next, we refine our exploration to the response variable–the change in the home team's money multiplier at halftime. Figure 1 included below displays the distribution of the change in money multiplier from pregame to halftime for the home team. Notice that there appears to be some outlying games in which the home team's change in money multiplier at halftime was exceedingly large in magnitude (i.e. over five). These outlying games cause the distribution to be left-skewed, as the mean home team change in money multiplier at halftime is -1.12 and the median home team change in money multiplier at halftime is 0.00. However, note that if we ignore these outlying games the distribution appears to be unimodal, fairly symmetric, and centered around zero.

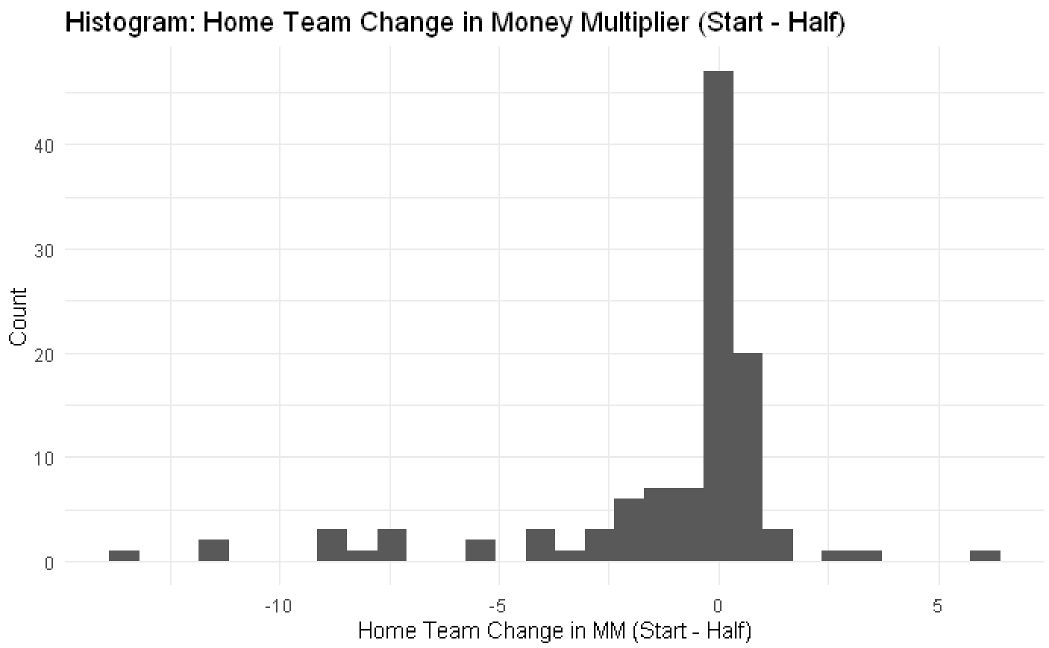


Figure 1: Distribution of Home Team Change in Money Multiplier (Start - Half)

Finally, we explore the relationship between the response variable, the home team's change in money multiplier at halftime, and one of our main predictors, the score differential at halftime (home score - away score). Figure 2 included below displays this relationship colored by type of game (NFL vs. NCAAF). Notice that there appears to be a positive association between the home team's change in money-multiplier at halftime and the score differential at halftime. This association is slightly different for NFL vs. NCAAF games, as the NCAAF games appear to experience more drastic shifts in the money multiplier.

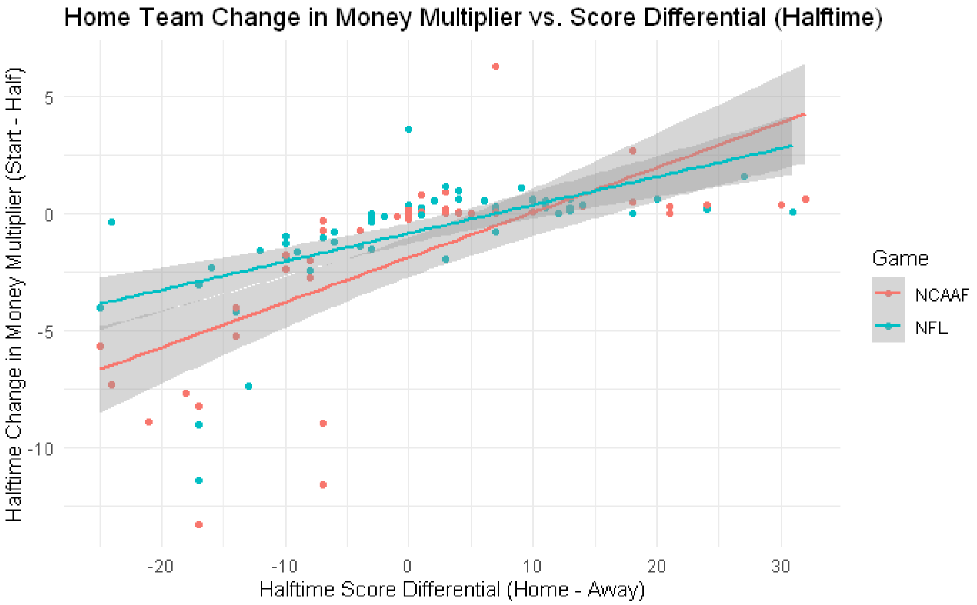


Figure 2: Home Team Change in Money Mul. vs. Score Differential (Halftime)

3 Methodology

3.1 Interpretive Modeling

In order to address the first main study objective (i.e. analyze the association of change in money multiplier with various game-related factors such as difference in score, number of turnovers, number of penalties, etc), we use regression modeling. First, we build a multi-linear regression model to quantify the relationship between our response variable, the home team's change in money multiplier at halftime, and our main explanatory variables–the score differential at halftime (home - away), a binary indicator of whether or not the home team is an underdog, a binary indicator variable of NFL or NCAAF game, and all interactions with score differential. This model is structured as follows:

$$\begin{aligned} MoneyM\hat{u}l.Change = \beta_0 + \beta_1 ScoreDifferential + \beta_2 Underdog + \beta_3 NFL \\ \beta_4 ScoreDifferential * Underdog + \beta_5 ScoreDifferential * NFL. \end{aligned} \quad (4)$$

Next, due to suspicions concerning the linearity of the relationship between the home team's change in money multiplier and the score differential, we build a polynomial regression model that includes score differential as a quadratic term and adjusts for whether the game is NFL or NCAAF. Although a bit of interpretability is lost by including score differential as a quadratic term, it provides more insight into the true relationship between the home team's change in money multiplier and score differential. This model is structured as follows:

$$MoneyM\hat{u}l.Change = \beta_0 + \beta_1 ScoreDifferential + \beta_2 ScoreDifferential^2 + \beta_3 NFL. \quad (5)$$

While both models described above provide insight as to how the home team's money multiplier changes based on a few explanatory variables, they do not account for several potential sources of confounding. Therefore, our final multi-linear regression model keeps score differential as a quadratic term and also incorporates several in-game statistics for both teams at halftime. More specifically, this model includes the difference in yards, first downs, fumbles, sacks, interceptions, and penalties (Home - Away) at halftime to predict the home team's change in money multiplier at halftime. This model is structured as follows:

$$\begin{aligned} MoneyM\hat{u}l.Change = \beta_0 + \beta_1 ScoreDifferential + \beta_2 ScoreDifferential^2 \\ + \beta_3 NFL + \beta_4 YardsDiff. + \beta_5 FirstDownDiff. + \beta_6 FumblesDiff. \\ + \beta_7 SacksDiff. + \beta_8 IntsDiff. + \beta_9 PenaltyDiff. \end{aligned} \quad (6)$$

3.2 Predictive Modeling

In order to address the second main study objective (i.e. build a predictive model to determine the game outcome), we use decision-tree based methods. With the end goal of informing a potential hedge at half-time, we propose a predictive model to determine the end results of the game based on the first half statistics. To predict the game outcome, we consider a single decision tree and a random forest model trained on 88 randomly sampled observations and tested on the remaining 24 observations. Both models are trained on all predictive variables at halftime: point differential, total yards for each team, total first downs for each team, total fumbles for each team, total sacks for each team, total interceptions for each team, total penalties for each team, the ESPN win prediction percentage for the home team, a binary indicator to identify if the home team is an underdog, and a binary indicator to identify whether the competition is an NCAAF or NFL game.

Due to the simple interpretations associated with splits on a single decision tree, the tree returns a set of rules that a potential sports gambler can consider when deciding whether or not to hedge their bet. Trained without setting maximum depth, the decision tree only reaches depth 3, which allows for ease of interpretation. Figure 3 displays the single decision tree structure and train set classification purity. For example, 21 games in the train set are perfectly classified as a loss for the home team if the ESPN win prediction was less than or equal to 56.9% and the home team was losing by 8 or more points at half time.

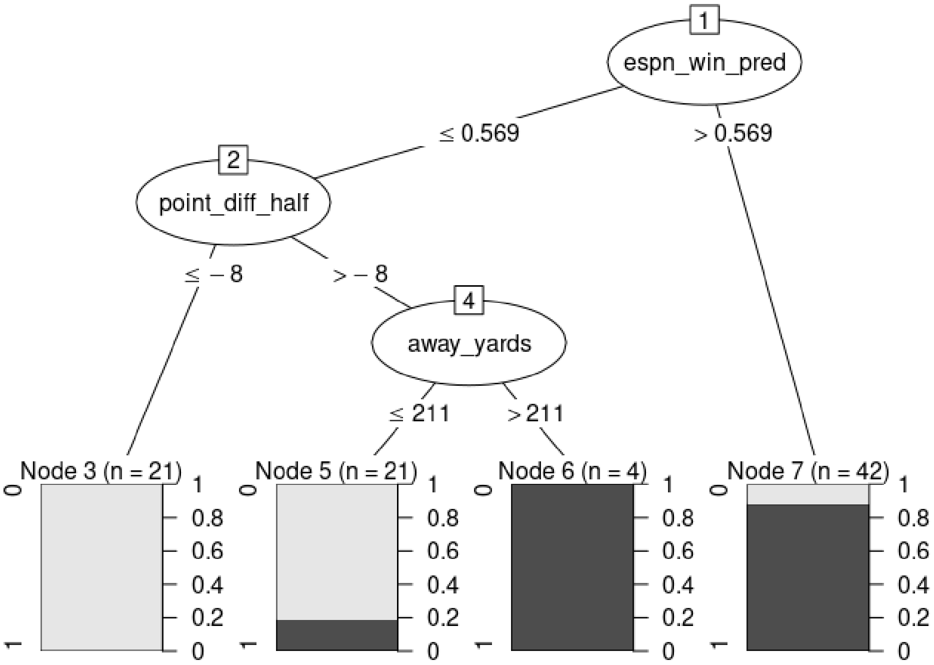


Figure 3: Simple Decision Tree Structure

Although a single-decision tree is easy to interpret, random forests are more effective predictive models that significantly decrease the variance of fitting a single decision tree. Through out-of-bag cross-validation, we tune and select hyperparameters based on highest validation accuracy. Finally, we train a random forest model with the optimal parameters selected through cross-validation: max of 75 nodes and 6 randomly selected variables at each split.

4 Results

4.1 Regression Models

First, we fit our baseline regression model on the entire data set. Table 2 provides the summary output of this model. As indicated by the model coefficients, for a college game where the home team is the favorite, every point the home team scores more than the away team is associated with a 0.17 unit increase in the money-multiplier. When the home team is an underdog the intercept is shifted upwards by 0.32 units and when it is an NFL game the intercept is shifted upwards by 0.98 units. The interaction term between score differential and home underdog shows that the slope of the score differential term increases by 0.13 when the home team is an underdog and decreases by 0.09 when the competition is an NFL game. All terms in this model are significant except for the home underdog main effect. However, the model residual plots (see Appendix B) suggest that this model violates a few of the assumptions of a linear model. More specifically, the curvature of the residuals vs. fitted values plot perhaps indicates that the relationship between the change in money multiplier and score differential is quadratic rather than linear.

Next, we fit the polynomial regression model on the entire data set using first and second order terms for score differential as well as a binary indicator for the type of game (NFL vs. NCAAF). As shown in Table 3, the intercept term of -0.97 for college games (when NFL is 0) and -0.27 for NFL games is much closer to what we would expect. Additionally, the first and second order coefficients for score differential are both

Table 2: Baseline Regression Model Summary Output

Coefficients	**Estimate**	**Std. Error**	t **value**	p **value**
$Intercept$	-1.75	0.37	-4.77	6.01×10^{-6}
$ScoreDifferential$	0.17	0.03	6.39	4.46×10^{-9}
$HomeUnderdog$	0.32	0.45	0.71	4.81×10^{-1}
NFL	0.98	0.43	2.29	2.40×10^{-2}
$ScoreDifferential * HomeUnderdog$	0.13	0.04	3.27	1.45×10^{-3}
$ScoreDifferential * NFL$	-0.09	0.03	-2.64	9.51×10^{-3}

incredibly significant, confirming our belief that the relationship between the change in money multiplier and score differential is more accurately modeled by a quadratic relationship. The model residual plots (see Appendix B) are much less concerning than those of the baseline regression model discussed previously.

Table 3: Polynomial Regression Model Summary Output

Coefficients	**Estimate**	**Std. Error**	t **value**	p **value**
$Intercept$	-0.52	0.25	-2.12	3.64×10^{-2}
$ScoreDifferential$	0.18	0.02	10.49	2.00×10^{-16}
$ScoreDifferential^2$	-0.005	0.001	-5.185	1.000×10^{-6}
NFL	0.71	0.41	1.72	8.92×10^{-2}

Finally, we fit another polynomial regression model with score differential as a quadratic term; however, we also included yard differential, first down differential, fumble differential, sack differential, interception differential, and penalty differential as additional predictors. The effect of these additional predictors appear to be minimal. The intercept term along with the first and second order coefficients for score differential remained as significant predictors; however, none of the additional predictors proved to be significant. The results suggest that even after adjusting for several in-game factors, score differential remains a significant predictor of the change in the money multiplier. See Appendix B for full summary table output for each model.

4.2 Predictive Models

Table 4 contains the train and test set accuracies for the single tree and the random forest model. As expected, the random forest model yields higher predictive accuracy for both train and test sets.

Table 4: Predictive Modeling Classification Accuracy

Model	**Train Accuracy**	**Test Accuracy**
Single Tree	89.7%	87.5%
Random Forest	100%	91.7%

The results of both prediction models allow us to conclude that the random forest model outperforms a single tree when predicting on unseen data, but only marginally. A 4.2% difference in test set accuracy is equivalent to mis-classifying the winner in only one more game. While these results may be skewed by the sample of games considered in the test set, the single tree provides a few simple decision criteria to determine the game outcome at halftime. Therefore, both models are reasonably valid provided that a bettor may not always have the technology required to predict the results of a new game with the random forest model readily accessible.

To determine if hedging based on our models' predictions can win a bettor more money than not hedging, we run a simulation. The strategy is simple: if the model predicts that the original bet placed will lose at halftime, the bettor can simply hedge to optimize their gain. Given 1000 completely uninformed gamblers who bet $100 initially on a random team with equal probability, we compare the profits under the model-based hedging strategy to not hedging on the test set games. Figure 4 displays the results of our simulation.

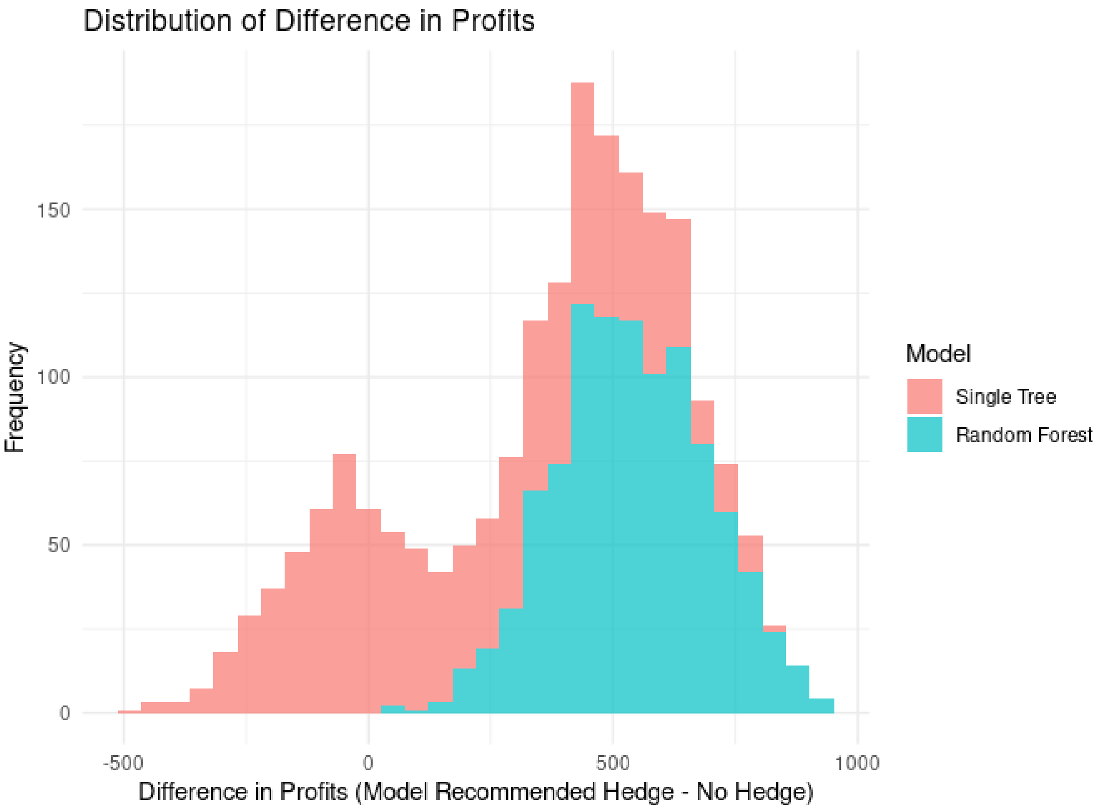


Figure 4: Profit Distribution

The shape and location of the difference in profit distributions tell us that most bettors stand to profit more when hedging based on our model's predictions. The mean of the difference in profits when using the single decision tree is approximately 200. This tells us that the average bettor who selects their picks randomly stands to profit $200 dollars more by hedging based on the single tree's predictions. As expected, the average bettor who hedges based on the random forest's predictions stands to make significantly more (Mean of $537). It's important to note that there is a population of bettors, about 31%, who stand to make less money when hedging based on the single-tree's predictions. This population is representative of those who randomly place a winning bet initially, but at halftime they hedge on an incorrect prediction from the single-tree. While some bettors lose money hedging on the single tree's predictions, no bettor stands to lose money when hedging on the random forest's predictions.

5 Discussion

We “tackled” the first study objective (i.e. understanding the nuanced relationship between the change in odds and in-game activity within the fist half) using interpretative regression models to predict the change in money multiplier based on point differential at halftime. In general, we concluded that when a team is out-scoring their opponent at half-time, they tend to be more favored to win than they were at the start of the game. Significant interaction effects showed that odds change differently depending on the type of competition (NFL or NCAAF) and the initial status of the team (underdog or favorite). Intuitively, this tells us that we expect an underdog who may be leading marginally to be less favored at halftime than a team who is initially favored to win and is leading marginally at halftime. Additionally, we concluded that the relationship between the change in money multiplier and point differential at half is better represented through a quadratic relationship because we expect that the change in money multiplier is more extreme for greater score differentials. Using the interpretations from our regression models, we can provide some insight when hedging a bet. Using our models, a bettor can assess the actual change in odds at half time. If a bettor initially bets on one team and the line corrects more towards the other team than estimated by our model, that could indicate that the line is over adjusted. Therefore, the hedge odds are better than they should be, and the bettor can hedge their bet. Likewise, if the line corrects more towards the initial bet than expected, the line is likely under adjusted and the bettor may benefit from sticking with the initial bet. However, these tips should not be taken as an absolute rule because other in-game factors (such as injuries) that our model does not account for could be influencing the lines.

In order to address the second study objective (i.e. create a best predictive model to determine the game outcome), we developed two non-parametric models: a single decision tree and a random forest model. As expected, the random forest ensemble method outperformed the single tree in prediction accuracy, but the single tree yielded easily interpretable decision criteria. Out of the variables included in both models, ESPN's predicted win percentage is definitely the most important for a bettor to consider. This predictor is the first split in our single tree, and has highest variable importance in the random forest model. ESPN likely uses a very complex model to determine win likelihood, which explains why this predictor dominates any tree based model. However, we include this predictor in our predictive models because it is readily available information that can inform a potential hedge. In the end, both predictive models are relevant because a bettor may not always have access to the random forest model when deciding to hedge their bet or not. Additionally, through simulation we showed that a bettor is likely to optimize their net gains when hedging according to either models predictions. However, like stated before, we do not recommend blindly betting according to our model's predictions because our model is quite limited.

Although our results provide interesting insights into hedging at halftime, the models developed in this study do contain a few limitations. Due to the manual nature of our data collection process and the limited project time frame, our data set is relatively small in terms of the total number of observations. Our ideal data set would contain multiple seasons worth of NFL and NCAAF games. Second, our data set likely does not contain all potential sources of confounding. Perhaps some other significant features that would be useful to incorporate into our models are weather conditions and player injuries. Third, the models developed in this study do not generalize well, as they were trained only on NFL and NCAAF regular season games during the 2021 season. Ideally, these models should be trained on both regular season and post-season games across multiple seasons.

6 Future Work

Given the limitations outlined in the previous section as well as the time constraints imposed on our study, there are several opportunities for future research. First and foremost, future studies should be carried out using a data set with both more game observations and more predictors. Although the modeling approach outlined in this study is sound, each predictive model would benefit from more training data. This would enable us to construct a better model across different seasons as well. Second, future researchers should explore using causal inference methods when drawing inferences from the data. Although multi-linear regression models can be an effective way to adjust for confounding, perhaps propensity score methods such as matching or weighting would be more effective. Third and finally, future studies should explore more machine learning techniques for predictive modeling. Although random forest models are known to be effective, future researchers should attempt to build boosting and neural net models.

Appendix A - EDA

This appendix contains all R code used to perform EDA.

```r
# Import necessary libraries
library(dplyr)
library(ggplot2)
library(tidyr)
library(randomForest)


# Read in the data and do some preliminary cleaning to merge dfs
nfl <- read.csv('nfl_odds.csv')
ncaa <- read.csv('ncaa_odds.csv')

ncaa <- ncaa %>%
  select(-home_plays, -away_plays) %>%
  na.omit()
ncaa <- ncaa[,2:length(ncaa)]

nfl['nfl'] <- 1
ncaa['nfl'] <- 0

all_games <- rbind(ncaa, select(nfl, -X))

# Extract the median odds for all bookmakers
# Make a column for underdog
# Make a column for home money multiplier at start and half
# Make column for change in money multiplier
# Impute NAs with mean
# Double check there are no more NAs
medians <- all_games %>%
  select(-bookmaker) %>%
  group_by(home_team, away_team, date) %>%
  summarise_all(median, na.rm = T) %>%
  mutate(home_underdog = ifelse(home_odds_start > away_odds_start, 1, 0)) %>%
  mutate(home_mul_start = ifelse(home_odds_start > 0,
        home_odds_start / 100, 100 / abs(home_odds_start))) %>%
  mutate(home_mul_half = ifelse(home_odds_half > 0,
        home_odds_half / 100, 100 / abs(home_odds_half))) %>%
  mutate(change_in_mul = home_mul_start - home_mul_half) %>%
  mutate(point_diff_half = home_score_half - away_score_half) %>%
  mutate(home_pen = ifelse(is.na(home_pen), mean(all_games$home_pen, na.rm = T), home_pen)) %>%
  mutate(away_pen = ifelse(is.na(away_pen), mean(all_games$away_pen, na.rm = T), away_pen)) %>%
  na.omit()
```

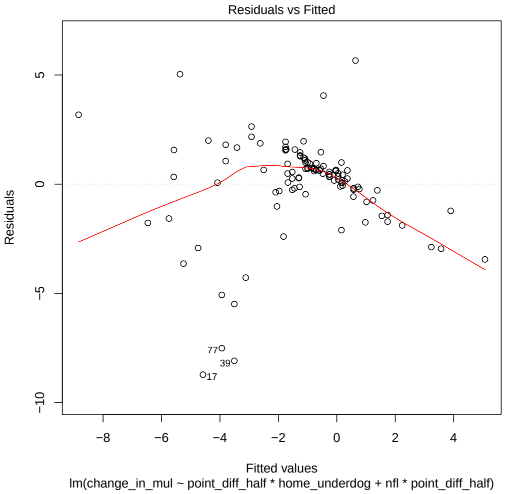
```r
# Histogram: Change in Home Team Odds (Initial - Halftime)
medians %>%
  ggplot(aes(change_in_mul)) +
  geom_histogram(bins = 30) +
  labs(x = "Home Team Change in MM (Start - Half)", y = "Count",
       title = "Histogram: Home Team Change in Money Multiplier (Start - Half)") +
  theme_minimal()
```

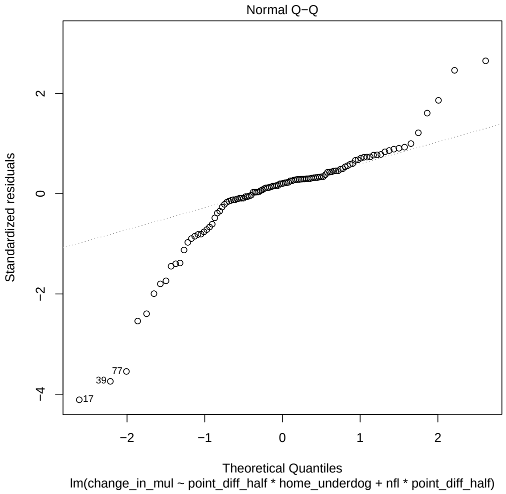
```
# Plot: Home Team Change in Odds (Start-Half) vs. Halftime Score Differential
medians %>%
  ggplot(aes(x=point_diff_half, y=change_in_mul, color = as.factor(nfl))) +
  geom_point() +
  geom_smooth(method = 'lm') +
  xlab("Halftime Score Differential (Home - Away)") +
  ylab("Halftime Change in Money Multiplier (Start - Half)") +
  ggtitle("Home Team Change in Money Multiplier vs. Score Differential (Halftime)") +
  labs(color = "Game") +
  scale_color_discrete(c("NCAAF", "NFL")) +
  theme_minimal()
```

Appendix B - Interpretive Modeling

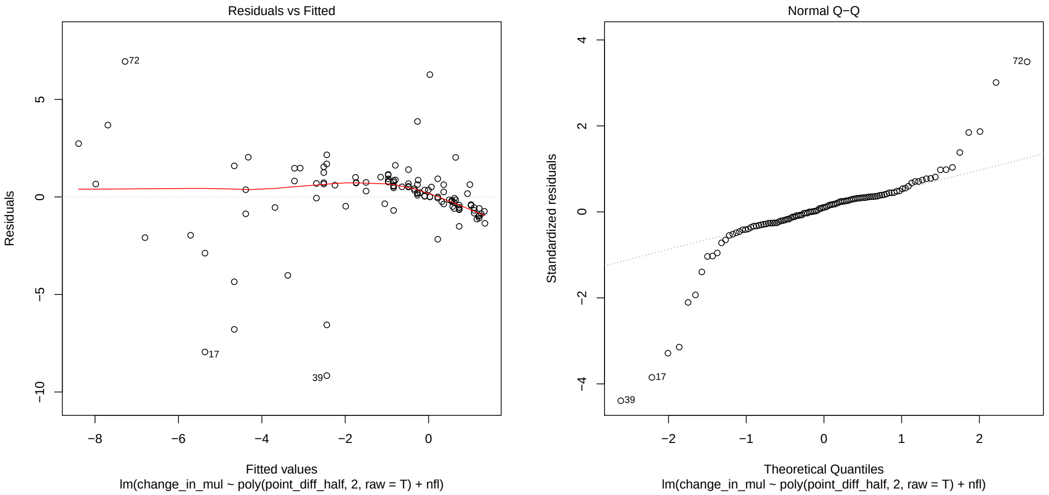
This appendix contains all R code used to perform interpretive modeling.

```
# Base Model (home_odds_change ~ score_differential_half)
baseModel <- lm(change_in_mul ~ point_diff_half * home_underdog + nfl*point_diff_half,
                data = medians)
summary(baseModel)

##
## Call:
## lm(formula = change_in_mul ~ point_diff_half * home_underdog +
##     nfl * point_diff_half, data = medians)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7263 -0.2926  0.4450  0.9868  5.6606
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.75451    0.36813  -4.766 6.01e-06 ***
## point_diff_half                0.16630    0.02601   6.393 4.46e-09 ***
## home_underdog                  0.32058    0.45340   0.707  0.48107
## nfl                            0.97789    0.42706   2.290  0.02401 *
## point_diff_half:home_underdog  0.12989    0.03973   3.270  0.00145 **
## point_diff_half:nfl           -0.09152    0.03465  -2.641  0.00951 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.213 on 106 degrees of freedom
## Multiple R-squared:  0.4969,Adjusted R-squared:  0.4732
## F-statistic: 20.94 on 5 and 106 DF,  p-value: 1.663e-14

# Check assumptions
plot(baseModel, which = c(1,2))
# Linearity and Constant Variance Threatened
```

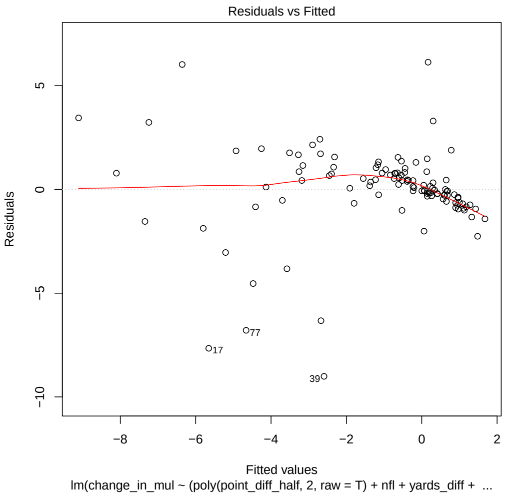


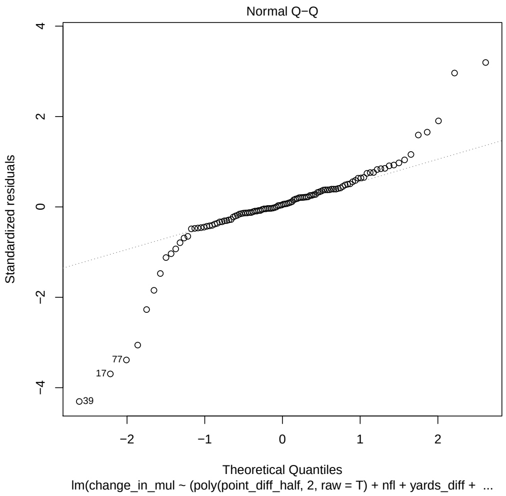


```r
# Fit a Quadratic Model
# Poly Model (home_odds_change ~ score_differential_half)
polyModel <- lm(change_in_mul ~ poly(point_diff_half, 2, raw = T) + nfl, data = medians)
summary(polyModel)

##
## Call:
## lm(formula = change_in_mul ~ poly(point_diff_half, 2, raw = T) +
##     nfl, data = medians)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1579 -0.5431  0.2198  0.7577  6.9444
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        -0.9701286  0.3573845  -2.715  0.00773 **
## poly(point_diff_half, 2, raw = T)1  0.1765674  0.0167751  10.526  < 2e-16 ***
## poly(point_diff_half, 2, raw = T)2 -0.0048164  0.0009695  -4.968 2.55e-06 ***
## nfl                                 0.7025686  0.4095742   1.715  0.08915 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.115 on 108 degrees of freedom
## Multiple R-squared:  0.5319,Adjusted R-squared:  0.5189
## F-statistic:  40.9 on 3 and 108 DF,  p-value: < 2.2e-16

plot(polyModel, which = c(1,2))
```



```r
# Make every game statistic a difference term for multi model
multi <- medians %>%
  ungroup() %>%
  mutate(yards_diff = home_yards - away_yards) %>%
  mutate(first_down_diff = home_first_downs - away_first_downs) %>%
  mutate(fumbles_diff = home_fumbles - away_fumbles) %>%
  mutate(sacks_diff = home_sacks - away_sacks) %>%
  mutate(ints_diff = home_ints - away_ints) %>%
  mutate(penalty_diff = home_pen - away_pen)

# Define multi model with poly point differential
multiModel <- lm(change_in_mul ~ (poly(point_diff_half, 2, raw = T) + nfl +  yards_diff
+ first_down_diff + fumbles_diff + sacks_diff + ints_diff + penalty_diff), data = multi)

summary(multiModel)

##
## Call:
## lm(formula = change_in_mul ~ (poly(point_diff_half, 2, raw = T) +
##     nfl + yards_diff + first_down_diff + fumbles_diff + sacks_diff +
##     ints_diff + penalty_diff), data = multi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0072 -0.5390  0.1025  0.8067  6.1318
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        -1.0229064  0.3713351  -2.755  0.00696 **
## poly(point_diff_half, 2, raw = T)1  0.2110184  0.0315261   6.693 1.20e-09 ***
## poly(point_diff_half, 2, raw = T)2 -0.0048209  0.0010013  -4.815 5.13e-06 ***
## nfl                                 0.7632521  0.4224747   1.807  0.07377 .
## yards_diff                         -0.0001103  0.0034734  -0.032  0.97473
## first_down_diff                    -0.0607863  0.0459599  -1.323  0.18893
## fumbles_diff                        0.1673947  0.2256219   0.742  0.45984
## sacks_diff                         -0.0123695  0.1419003  -0.087  0.93071
## ints_diff                          -0.2691821  0.2647874  -1.017  0.31175
## penalty_diff                       -0.0220363  0.0949279  -0.232  0.81690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.14 on 102 degrees of freedom
## Multiple R-squared:  0.5472,Adjusted R-squared:  0.5072
## F-statistic: 13.69 on 9 and 102 DF,  p-value: 3.44e-14

# Check model assumptions
plot(multiModel, which = c(1,2))
```


Residuals vs Fitted
Residuals
5
0
−5
−10
−8
−6
−4
−2
0
2
17
77
39
Fitted values
lm(change_in_mul ~ (poly(point_diff_half, 2, raw = T) + nfl + yards_diff + ...


Normal Q−Q
Standardized residuals
4
2
0
−2
−4
−2
−1
0
1
2
77
17
39
Theoretical Quantiles
lm(change_in_mul ~ (poly(point_diff_half, 2, raw = T) + nfl + yards_diff + ...

Appendix C - Predictive Modeling

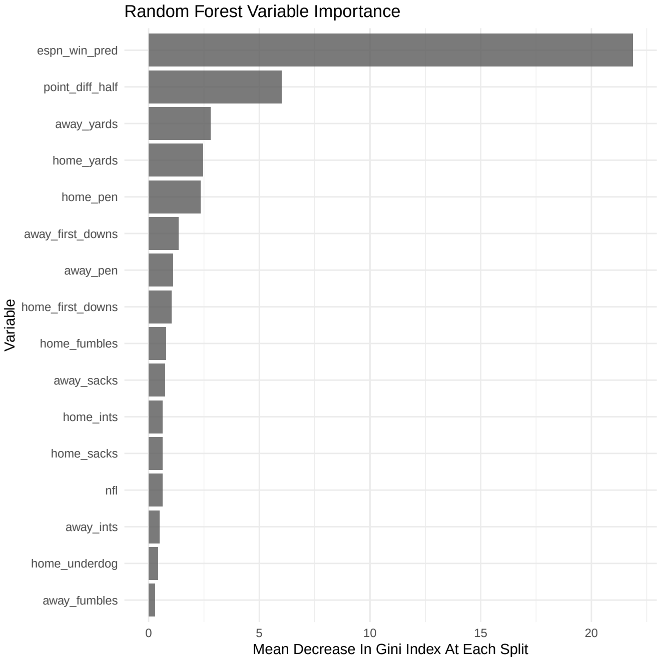
This appendix contains all R code used to perform predictive modeling.

```r
###SPLIT TEST AND TRAIN#########
set.seed(139)
train_idx <- sample(1:nrow(medians), 88)
train <- medians[train_idx,]
test <- medians[-train_idx,]

X_train <- train %>%
  ungroup() %>%
  select(point_diff_half , home_yards, away_yards,
         home_first_downs, away_first_downs, home_fumbles, away_fumbles,
         home_sacks, away_sacks, home_ints, away_ints, home_pen, away_pen,
         espn_win_pred, home_underdog, nfl)
y_train <- as.factor(train$home_win)

X_test <- test %>%
  ungroup() %>%
  select(point_diff_half , home_yards, away_yards,
         home_first_downs, away_first_downs, home_fumbles, away_fumbles,
         home_sacks, away_sacks, home_ints, away_ints, home_pen, away_pen,
         espn_win_pred, home_underdog, nfl)
y_test <- as.factor(test$home_win)
```

```r
######FIT A SINGLE DECISION TREE########
tree <- C5.0(x = X_train, y = y_train)

plot(tree)

#training accuracy
sum(predict(tree, X_train) == y_train) / nrow(X_train)
#testing accuracy
sum(predict(tree, X_test) == y_test) / nrow(X_test)
```

```r
####FIT A WELL TUNED RANDOM FOREST######
grid = expand.grid(maxnodes = c(6,12,25,50,75,100), mtrys = c(4, 6, 12,16))
val_scores <- rep(NA, nrow(grid))
X_train["home_win"] = y_train

# Cross validate
for (i in 1:nrow(grid)) {
  rf <- randomForest(home_win ~ ., data = X_train,
                     maxnodes= grid[i,1], mtry = grid[i,2], ntree = 300)
  val_scores[i] <- sum(rf$predicted == X_train$home_win, na.rm = T) / length(na.omit(rf$predicted))
}

## Warning in randomForest.default(m, y, ...):  maxnodes exceeds its max value.
## Warning in randomForest.default(m, y, ...):  maxnodes exceeds its max value.
## Warning in randomForest.default(m, y, ...):  maxnodes exceeds its max value.
## Warning in randomForest.default(m, y, ...):  maxnodes exceeds its max value.
```

```
# Extract best hyperparamters for tuning
best_grid <- which(val_scores == min(val_scores))

# Refit best model
rf <- randomForest(home_win ~., data = X_train,
                   maxnodes= grid[best_grid, 1], mtry = grid[best_grid, 2], ntree = 300)

## Warning in randomForest.default(m, y, ...):  maxnodes exceeds its max value.

#extract variable importance
variable_importance <- data.frame(MeanDecreaseGini = rf$importance[],
                                  Variable = names(rf$importance[,1]))

ggplot(variable_importance) +
  geom_bar(aes(x = reorder(Variable, MeanDecreaseGini), y = MeanDecreaseGini),
           stat = 'identity', alpha = .8) +
  coord_flip() +
  ylab("Mean Decrease In Gini Index At Each Split") +
  xlab("Variable") +
  ggtitle("Random Forest Variable Importance") +
  theme_minimal()
```



```
# Training accuracy
sum(predict(rf, X_train) == y_train) / nrow(X_train)

## [1] 1

# Testing accuracy
sum(predict(rf, X_test) == y_test) / nrow(X_test)

## [1] 0.9583333
```

Imagine 1000 different bettors betting on the same games. This simulation gets the difference in money made for each better if they followed our strategy or threw or decided not to hedge the runtime for this cell is quite long. We turned eval off, but the resulting plot is in the paper.

```r
simulate_profit_difference <- function(sim_test, model) {

  sim_test["model_prediction"] <- predict(model, sim_test)
  profit_diff <- c()

  for (i in 1:1000) {
  informed_stuff <- sim_test %>%
    mutate(underdog_win = ifelse(home_underdog == home_win, 1, 0)) %>%
    mutate(bet_underdog = sample(0:1, 1, prob = c(.5, .5))) %>%
    mutate(pred_underdog_win = ifelse(model_prediction == home_underdog, 1,0)) %>%
    mutate(hedge = ifelse(pred_underdog_win == bet_underdog, 0, 1)) %>%
    mutate(init_odds = ifelse(home_underdog == bet_underdog, home_odds_start, away_odds_start)) %>%
    mutate(init_mul = ifelse(init_odds > 0, init_odds / 100, 100 / abs(init_odds))) %>%
    mutate(init_win = ifelse(bet_underdog == underdog_win, 1, 0)) %>%
    mutate(init_wins_prof = ifelse(init_win == 1, 100*init_mul, -100)) %>%
    mutate(hedge_odds_half = ifelse(home_underdog == 1, away_odds_half, home_odds_half)) %>%
    mutate(hedge_mul = ifelse(hedge_odds_half > 0,
           hedge_odds_half / 100, 100 / abs(hedge_odds_half))) %>%
    mutate(hedge_amount = (100*init_mul + 100) / (hedge_mul + 1)) %>%
    mutate(if_hedge_wins_prof = hedge_amount * hedge_mul) %>%
    mutate(if_hedge_prof = ifelse(init_win == 1,
           init_wins_prof - hedge_amount, if_hedge_wins_prof - 100)) %>%
    mutate(total_profit = ifelse(hedge == 1, if_hedge_prof, init_wins_prof))

    profit_diff[i] <- sum(informed_stuff$total_profit) - sum(informed_stuff$init_wins_prof)
  }
  return(profit_diff)
}

sim_test <- test
tree_profit_diff <- simulate_profit_difference(sim_test, tree)
rf_profit_diff <- simulate_profit_difference(sim_test, rf)

profits <- rbind(data.frame(profit = tree_profit_diff, Model = "Single Tree"),
                 data.frame(profit = rf_profit_diff, Model = "Random Forest"))

ggplot(profits) +
  geom_histogram(aes(profit, fill = Model), alpha = .7) +
  ggtitle("Distribution of Difference in Profits") +
  xlab("Difference in Profits (Model Recommended Hedge - No Hedge)") +
  ylab("Frequency") +
  theme_minimal()


summary(tree_profit_diff)
summary(rf_profit_diff)
```

Codebook

- point_diff_half: point difference between teams at half time.
- home_yards: total yards acquired by home team in first half.
- home_yards: total yards acquired by home team in first.
- away_yards: total yards acquired by away team in first half.
- home_first_downs: total 1st & 10's acquired by home team in first half.
- away_first_downs: total 1st & 10's acquired by away team in first half.
- home_fumbles: total times home team offense fumbles ball in first half.
- away_fumbles: total times away team offense fumbles ball in first half.
- home_sacks: total sacks on away team quarterback by home team defense in first half.
- away_sacks: total sacks on home team quarterback by away team defense in first half.
- home_ints: total interceptions made by home team defense in first half.
- away_ints: total interceptions made by away team defense in first half.
- home_pen: total penalties committed by home team in first half.
- away_pen: total penalties committed by away team in first half.
- espn_win_pred: ESPN prediction of home team's winning probability after the first half.
- home_underdog: binary indicator to show if the home team is the underdog (1- Yes, 0- No).
- nfl: binary indicator to show if it is an NFL game or not (1- Yes, 0- No).