Sports Performance Analytics, Assignment 1

Assignment 1: Win Probability

Reed Ballesteros

Northwestern University SPS, Fall 2022
MSDS-456-DL
Instructor: Prof. Bradley Smith
2023-02-05

1. Create a win probability model that will help us determine the impact of each play we run. For starters, include down, distance, yard line, time left in the game, and point differential.

Using guidance from the following article 'Building a Basic, In-Game Win Probability Model for the NFL' by Stephen Hill, Medium (https://medium.com/@technocat79/enhancing-our-basic-in-game-win-probability-model-for-the-nfl-random-forests-f9e8bb40583e), we will create a basic win probability model using game elements quarter, down, yards to go, yard line, time left, and point differential.

- Load play-by-play data for NFL seasons from 2010 to 2021 from nflfastR (already loaded into a local file), denoted as pbp:

```
# pbp <- nflfastR::load_pbp(2010:2021)
# saveRDS(pbp, "pbp_data_2010_2021.rds")
pbp <- readRDS("pbp_data_2010_2021.rds")
```

- Load play-by-play data for the 2022 NFL season (for testing) from nflfastR (already loaded into a local file), denoted as pbp_2022:

```
# pbp_2022 <- nflfastR::load_pbp(2022)
# saveRDS(pbp_2022, "pbp_data_2022.rds")
pbp_2022 <- readRDS("pbp_data_2022.rds")
```

We will perform the following data cleanup for pbp:

- Add column 'winner' to pbp to denote the winner of the football game for each play:

```
pbp <- pbp %>% mutate(winner = ifelse(home_score > away_score, home_team, away_team))
```

- Add column 'poswins' to pbp set to denote if the team on offense (the team with possession of the ball) won the game:

```
pbp <- pbp %>% mutate(poswins = ifelse(winner == posteam, "Yes","No"))
```

- Convert columns 'qtr', 'down', and 'poswins' (pbp only) to factors:

```
pbp$qtr = as.factor(pbp$qtr)
pbp$down = as.factor(pbp$down)
pbp$poswins = as.factor(pbp$poswins)
```

```
pbp_2022$qtr = as.factor(pbp_2022$qtr)
pbp_2022$down = as.factor(pbp_2022$down)
```

- Simplify the play-by-play dataset to a subset of columns relevant for our model (pbp_reduced and pbp_2022_reduced):

game_id, game_date, posteam, home_team, away_team, winner (training only), qtr, down, ydstogo, game_seconds_remaining, yardline_100, score_differential, poswins (training only)

```
# pbp_reduced = pbp %>% filter(play_type != "No Play"
#                                      & qtr != 5
#                                      & qtr != 6
#                                      & down != "NA"
#                                      & poswins != "NA") %>%
#                  select(game_id, game_date, posteam, home_team, away_team, winner, qtr,
#                         down, ydstogo, game_seconds_remaining, yardline_100, score_differential, posw
# saveRDS(pbp_reduced, "pbp_reduced_2010_2021.rds")
pbp_reduced <- readRDS("pbp_reduced_2010_2021.rds")
```

```
# pbp_2022_reduced = pbp_2022 %>% filter(play_type != "No Play"
#                                      & qtr != 5
#                                      & qtr != 6
#                                      & down != "NA") %>%
#                 select(game_id, game_date, posteam, home_team, away_team, qtr,
#                        down, ydstogo, game_seconds_remaining, yardline_100, score_differential, desc)
#  saveRDS(pbp_2022_reduced, "pbp_reduced_2022.rds")
pbp_2022_reduced <- readRDS("pbp_reduced_2022.rds")
```

- Create train and test game datasets from pbp_reduced, 80%/20% split:

```
set.seed(123)

# get list of unique game_ids
game_ids <- unique(pbp_reduced$game_id)

# split the list of game ids 80/20
split <- sample.split(game_ids, SplitRatio = 0.8)

# create lookup table of game_id split
split_games <- data.frame(game_id = game_ids, split = split)

# get list of game_ids that are in training only
train_game_ids <- split_games[split_games$split==TRUE,"game_id"]

# make train column in pbp dataset to flag games that are in training set
pbp_reduced <-mutate(pbp_reduced, train = pbp_reduced$game_id %in% train_game_ids)

# get training games susbset of pbp
train <- pbp_reduced[pbp_reduced$train==TRUE,]

# get test games susbset of pbp
test <- pbp_reduced[pbp_reduced$train==FALSE,]

# cleanup
pbp_reduced <- subset(pbp_reduced, select = -c(train))
train <- subset(train, select = -c(train))
test <- subset(test, select = -c(train,winner,poswins)) # test data shouldn't have winner/poswins colum
rm(split_games)
rm(train_game_ids)
rm(split)
rm(game_ids)
```

With the datasets prepared we can create a win probability model (model_pbp) using the train data:

```
model_pbp <- glm(poswins ~ qtr + down + ydstogo + game_seconds_remaining
             + yardline_100 + score_differential, train, family = "binomial")
```

We have the following summary and coefficients for model_pbp:

```
summary(model_pbp)
```

```
##
## Call:
## glm(formula = poswins ~ qtr + down + ydstogo + game_seconds_remaining +
##     yardline_100 + score_differential, family = "binomial", data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.00230  -0.83244   0.08525   0.86318   3.10300
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              1.037e+00  4.857e-02  21.350  < 2e-16 ***
## qtr2                    -1.715e-02  1.750e-02  -0.980 0.327208
## qtr3                    -4.159e-02  2.902e-02  -1.433 0.151874
## qtr4                    -1.286e-01  4.213e-02  -3.053 0.002269 **
## down2                   -7.784e-02  9.784e-03  -7.956 1.78e-15 ***
## down3                   -1.840e-01  1.137e-02 -16.188  < 2e-16 ***
## down4                   -3.722e-01  1.413e-02 -26.347  < 2e-16 ***
## ydstogo                 -9.270e-03  1.045e-03  -8.870  < 2e-16 ***
## game_seconds_remaining -5.401e-05  1.470e-05  -3.674 0.000239 ***
## yardline_100           -8.789e-03  1.682e-04 -52.247  < 2e-16 ***
## score_differential      1.747e-01  6.286e-04 277.858  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 531567  on 383474  degrees of freedom
## Residual deviance: 379618  on 383464  degrees of freedom
## AIC: 379640
##
## Number of Fisher Scoring iterations: 5
```

The factored property 'qtr4' seems to be the only statistically significant 'qtr' in the model, which makes sense as it is the last quarter in the game. We might consider simplifying the model to remove 'qtr' altogether since it could be viewed as a redundant property, as we have 'game_seconds_remaining' which represents a much more granular time element.

To help simplify processes we will create a wrapper function predict_wp() around model_pbp to predict away/home win probabilities and add it to any play-by-play dataset input:

```
predict_wp <- function(model, dataset) {
  pred <- predict(model, dataset, type = "response")
  results <- cbind(dataset,pred)
```

```
  results <- mutate(results, pred_away = ifelse(posteam == away_team, pred, 1-pred))
  results <- mutate(results, pred_home = 1-pred_away)
  results <- subset (results, select = -pred)
  return(results)
}
```

- Predict win probability with training data games using predict_wp():

```
train_results <- predict_wp(model_pbp,train)
```

- Predict win probability with test data games using predict_wp():

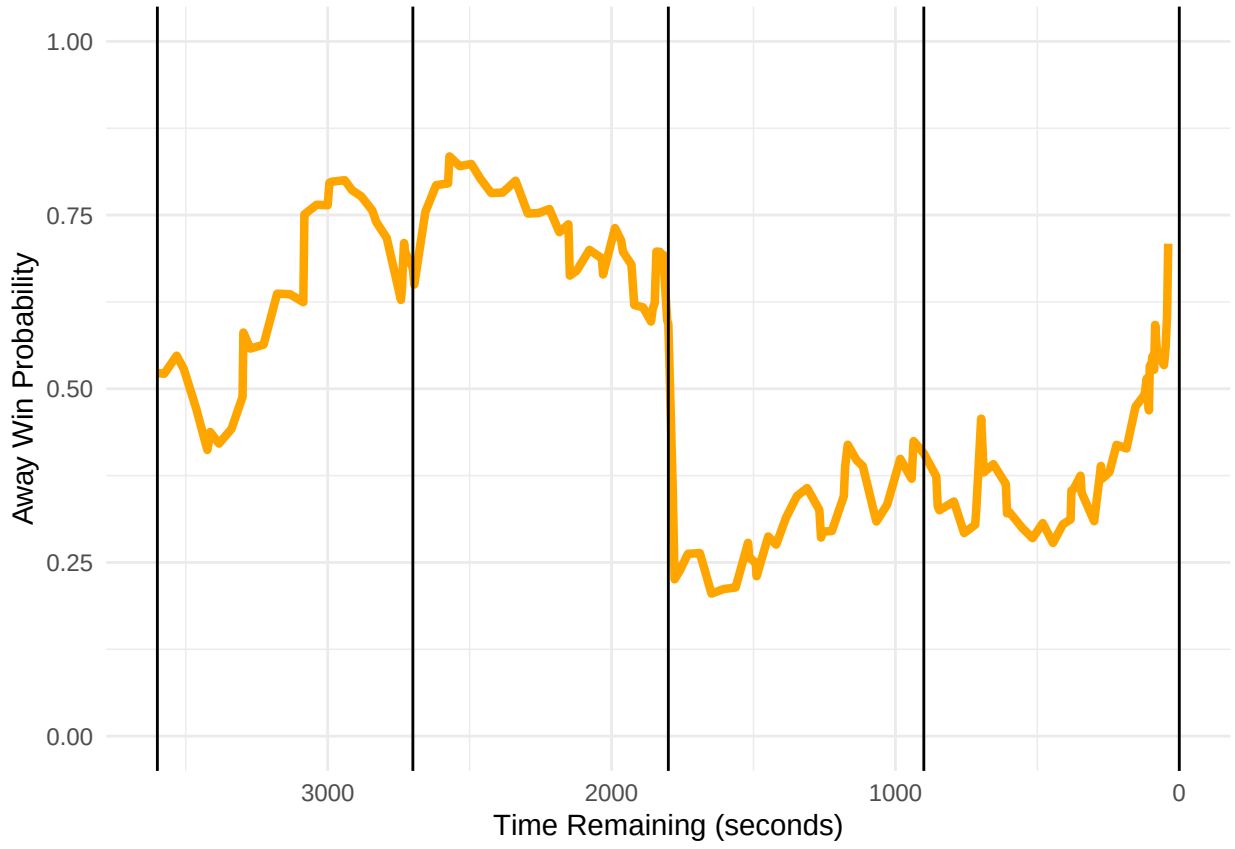
```
test_results <- predict_wp(model_pbp,test)
```

- Predict win probability with 2022 play-by-play data using predict_wp():

```
pred_2022_results <- predict_wp(model_pbp,pbp_2022_reduced)
```

2. Use the model you created to describe one game from the previous season. Include a visual display that has the time left in the game referenced on the x-axis and the win probability referenced on the y-axis. Describe and explain large shifts in win probability from the game, focusing on specific plays and/or drives that best explain these shifts.

Win probability from 2021 NFL Season (training set): 2/13/2022 - Los Angeles Rams (LA, Away) vs Cincinnati Bengals (CIN, Home) vs - Super Bowl LVI

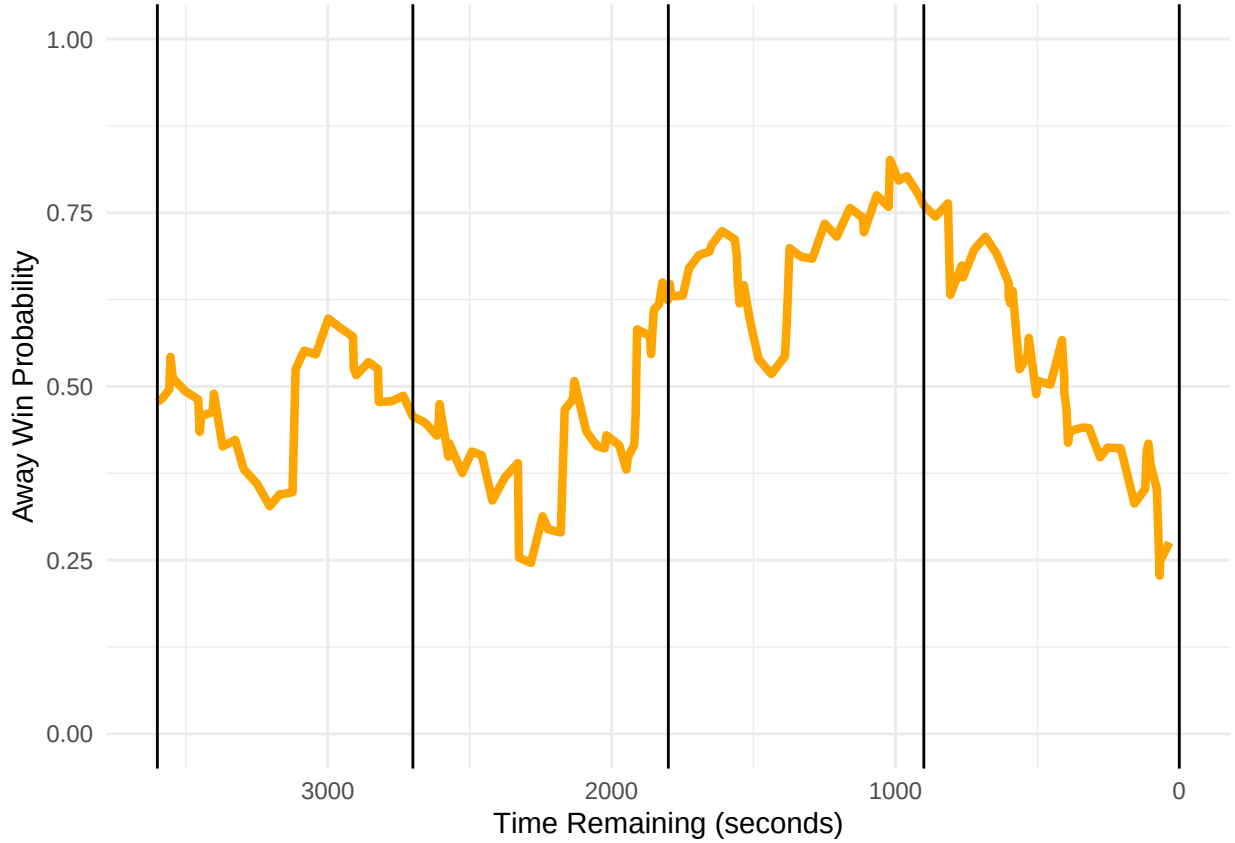


The largest swing from the chart above was at the first play at the start of the 3rd quarter (1800 seconds left) where CIN QB Joe Burrow throws a 75-yard pass to Tee Higgins for a touchdown, swinging the win probability from 59.3% LA to 65.1% CIN.

From 373 to 85 seconds left (6:13-1:25 4th) we see the LA's 79-yard offensive drive to score a touchdown and regaining the lead 20-23, swinging the probability from 64.5% CIN to 59.2% LA.

At 39 seconds left after the Bengals were not able to convert a 4th down and the Rams have possession the probability jumps about other 10% in favor of LA, from 60.6% to 70.9%, and sealing the win.

Win probability from 2021 NFL Season (test set): 1/30/2022 - San Francisco 49ers (SF, Away) at Los Angeles Rams (LA, Home) - 2022 NFC Championship Game

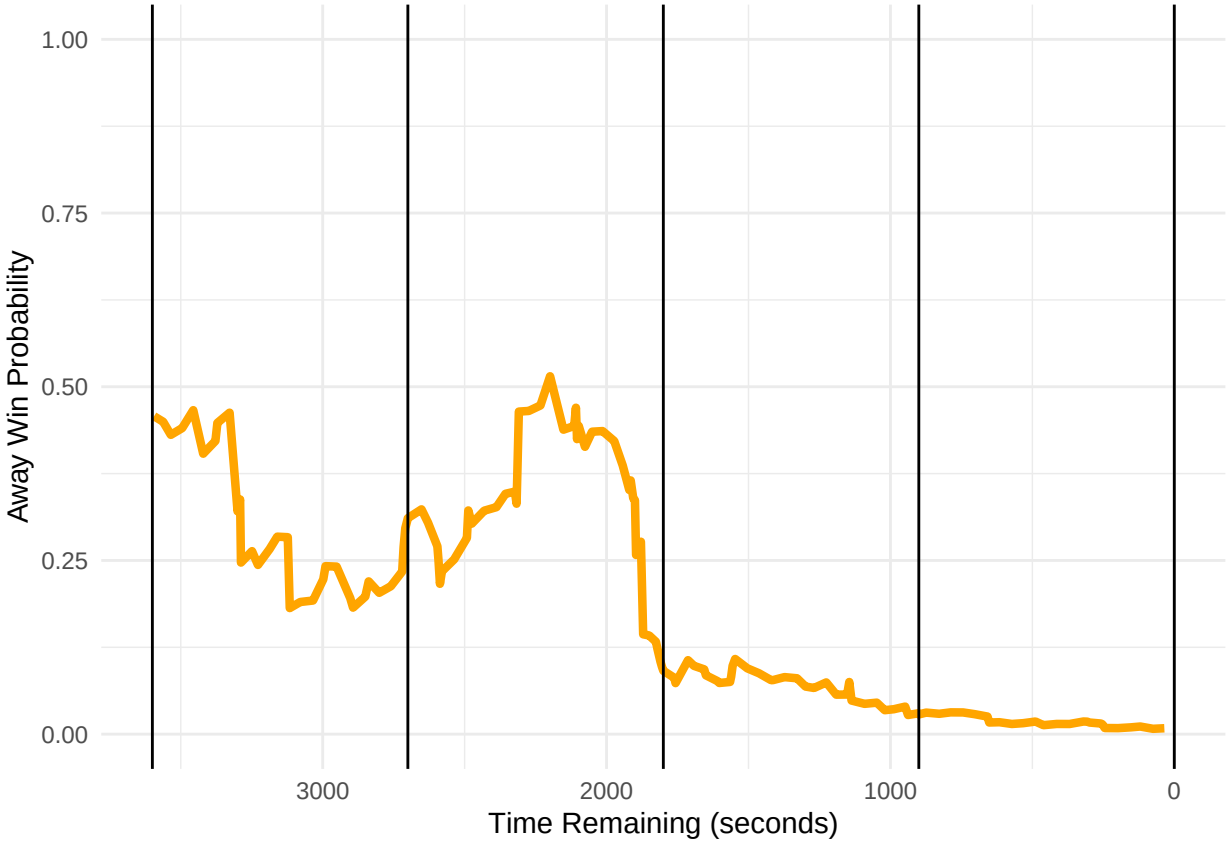


At 2179 seconds left (6:19 2nd) SF QB Jimmy Garoppolo throws a pass to Deebo Samuel for a 44-yard touchdown and tying the game, reducing the LA win probability from 71% to 53.3%.

From 1611 to 1373 seconds left (11:51-7:53 3rd) we see a dip in SF's win probability where it drops from 72.3% to as low as 51.8% then back to 70% due to an LA possession in which LA could not convert a 4th down play and SF is able to maintain a 3-point lead.

From 960 seconds left (1:00 3rd) to the end of the game we see SF's win probability at 80.2% and a 10-point lead drop to 27.5% and losing by 3 points from a comeback win by LA, which included a touchdown and two field goals in the 4th quarter.

Win probability from 2022 NFL Season (pbp 2022 set): 1/26/2023 - San Francisco 49ers (SF, Away) at Philadelphia Eagles (PHI, Home) - 2023 NFC Championship Game



With the game tied with 2199 seconds left (6:39 2nd) we see SF's win probability fall from 51.5% to 14.4% during PHI's 7-minute offensive drive where they were able convert a 4th and 1 situation, eventually score a touchdown, and SF not able to tie it before halftime. PHI continued extend their lead in the 2nd half. SF was already at a disadvantage losing starting QB Brock Purdy early in the game due to an elbow injury.

3. In addition, provide a discussion on what aspects of the model may be limiting and how you would advance this model in the future. Draw from course readings and cite sources where appropriate. What other applications are there for this model? Could we determine when to punt or go for it based on this model?

The limitation of the win probability model does not take into account a coaching staff's play calling tendencies and the personnel on the field, two areas that might require complex modeling in itself. Also factoring in events such as injuries or even penalties would be difficult to model as well. That said, if it was theoretically possible to include those factors, it would make for a much more granular and accurate model.

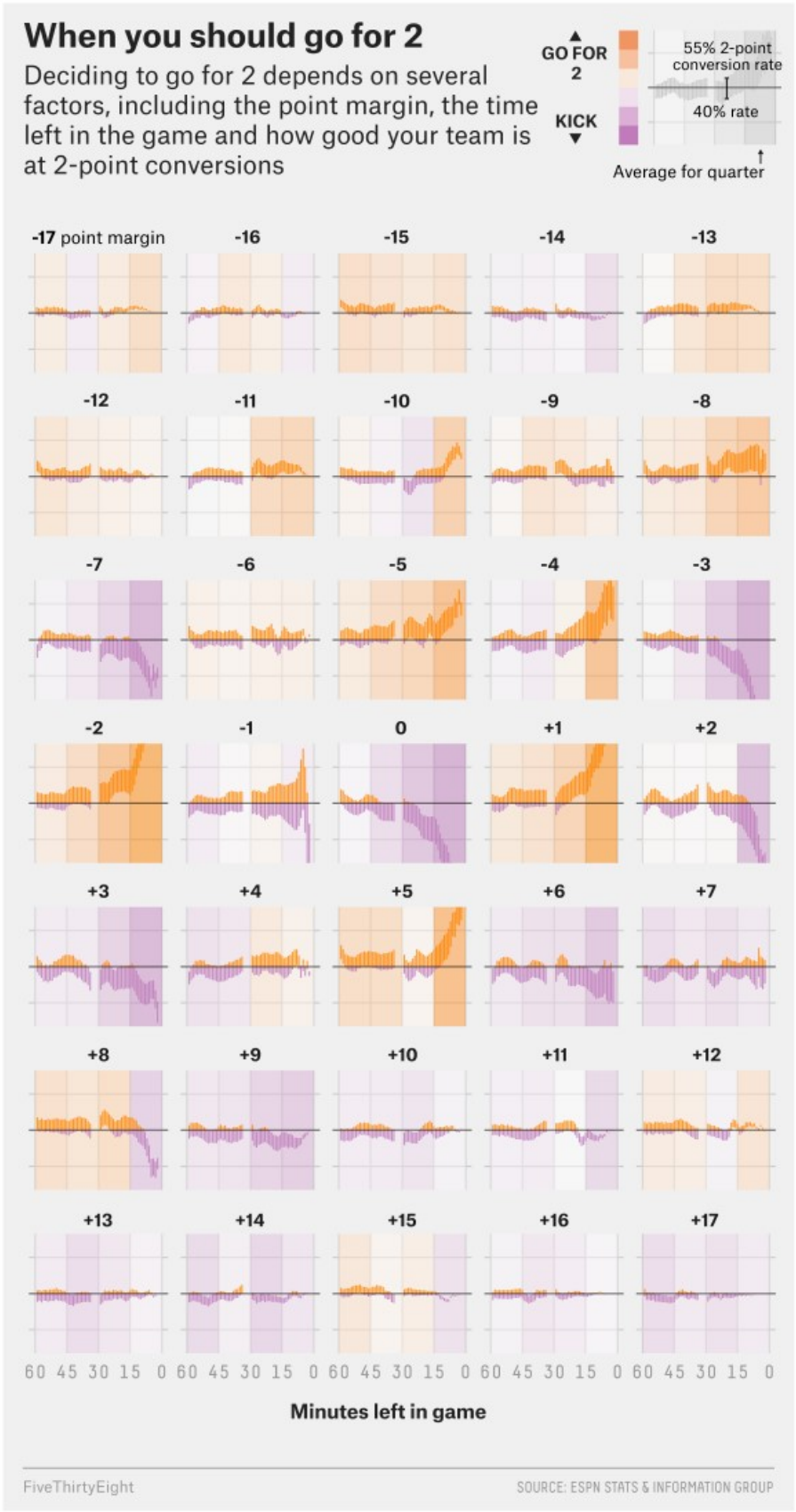
Given that basic win probability model includes the current score differential, time remaining, field position, down, yards to do to first down, quarter, and historical data, which this model uses, teams still find it helpful for 4th down situations. In a current 4th down situation, if there is a high probability the team could lose if they don't convert that 4th down, the best move might be to go for it. If there is still a high probability they can lose despite the 4th down conversion, it might be better to punt and hope to find a better opportunity later in the game.

4. Pick a chapter (18–27) from Mathletics and provide a brief description of the topic discussed. Explain how this will help our organization in the future and include analysis and data from last year (or several years).

Chapter 23 of Mathletics discussed whether to go for a one-point or two-point conversion, and references a FiveThirtyEight.com article to figure out which one to use, particularly in the second half of the game:

'When To Go For 2, For Real' by Benjamin Morris of FiveThrityEight.com https://fivethirtyeight.com/features/when-to-go-for-2-for-real/

The article displays the following chart:


When you should go for 2
Deciding to go for 2 depends on several factors, including the point margin, the time left in the game and how good your team is at 2-point conversions
GO FOR 2
KICK
55% 2-point conversion rate
40% rate
Average for quarter
-17 point margin
-16
-15
-14
-13
-12
-11
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
+1
+2
+3
+4
+5
+6
+7
+8
+9
+10
+11
+12
+13
+14
+15
+16
+17
60 45 30 15 0 60 45 30 15 0 60 45 30 15 0 60 45 30 15 0 60 45 30 15 0
Minutes left in game
FiveThirtyEight
SOURCE: ESPN STATS & INFORMATION GROUP

While the chart suggests to go for the two-point conversion when behind 8 or 4 points in the second half, FiveThirtyEight.com reports that coaches rarely did so during the 2015 and 2016 seasons. The reason why it is strongly suggested to go for the two-point conversions in these situations is because the calculated difference in win probability is greater with the 2-point conversion compared to the 1-point conversion.

We should conduct an updated report to see if win probabilities have changed since then, and if coaching habits have changed when it comes to extra-point decisions in terms of data-driven influence. We also want to see how accurate the results match up to the calculated win probabilities as well. The league has become more data-driven the past six years in many aspects of the game and business and we should understand how much it has actually impacted the sideline when it comes to extra-point situations.