

Predicting NFL Winners and Attendance

By Noah Yasbin, Harvey Barnes



What We are Exploring?

- Link team performance trends to probability of winning the next game
- Analyze how team strength metrics relate to home attendance
- Build rolling features to effectively prevent data leakage and show true predictive power



The Data

Source:

<https://github.com/rfordatascience/tidytuesday/blob/main/data/2020/2020-02-04/readme.md>

Data is read in with:

```
10  attendance <- readr::read_csv(  
11    "https://raw.githubusercontent.com/rfordatascience/tidytuesday/main/data/2020/2020-02-04/attendance.csv"  
12  )  
13  
14  standings <- readr::read_csv(  
15    "https://raw.githubusercontent.com/rfordatascience/tidytuesday/main/data/2020/2020-02-04/standings.csv"  
16  )  
17  
18  games <- readr::read_csv(  
19    "https://raw.githubusercontent.com/rfordatascience/tidytuesday/main/data/2020/2020-02-04/games.csv"  
20  )
```



Feature Creation

- Rolling point Differential
- Rolling cumulative Defensive and Offensive yards
- Rolling cumulative wins and losses
- Win percentage
- TurnOver differential
- These features are more powerful than the raw features from the data set, example points for.



Why rolling features are important?

Rolling features are important because they prevent data leakage. For example if your trying to predict the winner of a week 10 game and your model has the points differences of that game and future games, your model will have an unfair advantage over a real world situation.

Feature Creation Code

```
# Team-game level data -----

games_team <- games %>%
  mutate(game_id = row_number()) %>% # unique id per game

# Make two rows per game: one for home_team, one for away_team
pivot_longer(
  cols      = c(home_team, away_team),
  names_to  = "home_away",
  values_to = "team"
) %>%

# Stats from the *team's* point of view
mutate(
  # yards / points for & against
  yds_for      = if_else(team == winner, yds_win,  yds_loss),
  yds_against  = if_else(team == winner, yds_loss, yds_win),
  pts_for      = if_else(team == winner, pts_win,  pts_loss),
  pts_against  = if_else(team == winner, pts_loss, pts_win),

  # turnovers for & against -----
  turnovers_for      = if_else(team == winner, turnovers_win,  turnovers_loss),
  turnovers_against  = if_else(team == winner, turnovers_loss, turnovers_win),

  # result flags -----
  win      = as.integer(team == winner & is.na(tie)),
  loss     = as.integer(team != winner & is.na(tie)),
  tie_flag = as.integer(!is.na(tie))
) %>%
  arrange(team, year, week, date, time)
```

```
# Rolling stats -----

games_rolling <- games_team %>%
  group_by(team, year) %>%
  arrange(week, date, time, .by_group = TRUE) %>%
  mutate(
    games_played_prior = row_number() - 1L,

    # cumulative totals BEFORE this game -----
    cum_yds_for      = lag(cumsum(yds_for),      default = 0),
    cum_yds_against  = lag(cumsum(yds_against),   default = 0),
    cum_yds_diff     = cum_yds_for - cum_yds_against,

    cum_wins         = lag(cumsum(win),           default = 0),
    cum_losses       = lag(cumsum(loss),          default = 0),
    cum_ties         = lag(cumsum(tie_flag),       default = 0),

    cum_pts_for      = lag(cumsum(pts_for),        default = 0),
    cum_pts_against  = lag(cumsum(pts_against),    default = 0),
    cum_pts_diff     = cum_pts_for - cum_pts_against,

    # cumulative turnovers BEFORE this game -----
    cum_to_for      = lag(cumsum(turnovers_for),   default = 0),
    cum_to_against  = lag(cumsum(turnovers_against), default = 0),
    cum_to_diff     = cum_to_for - cum_to_against,

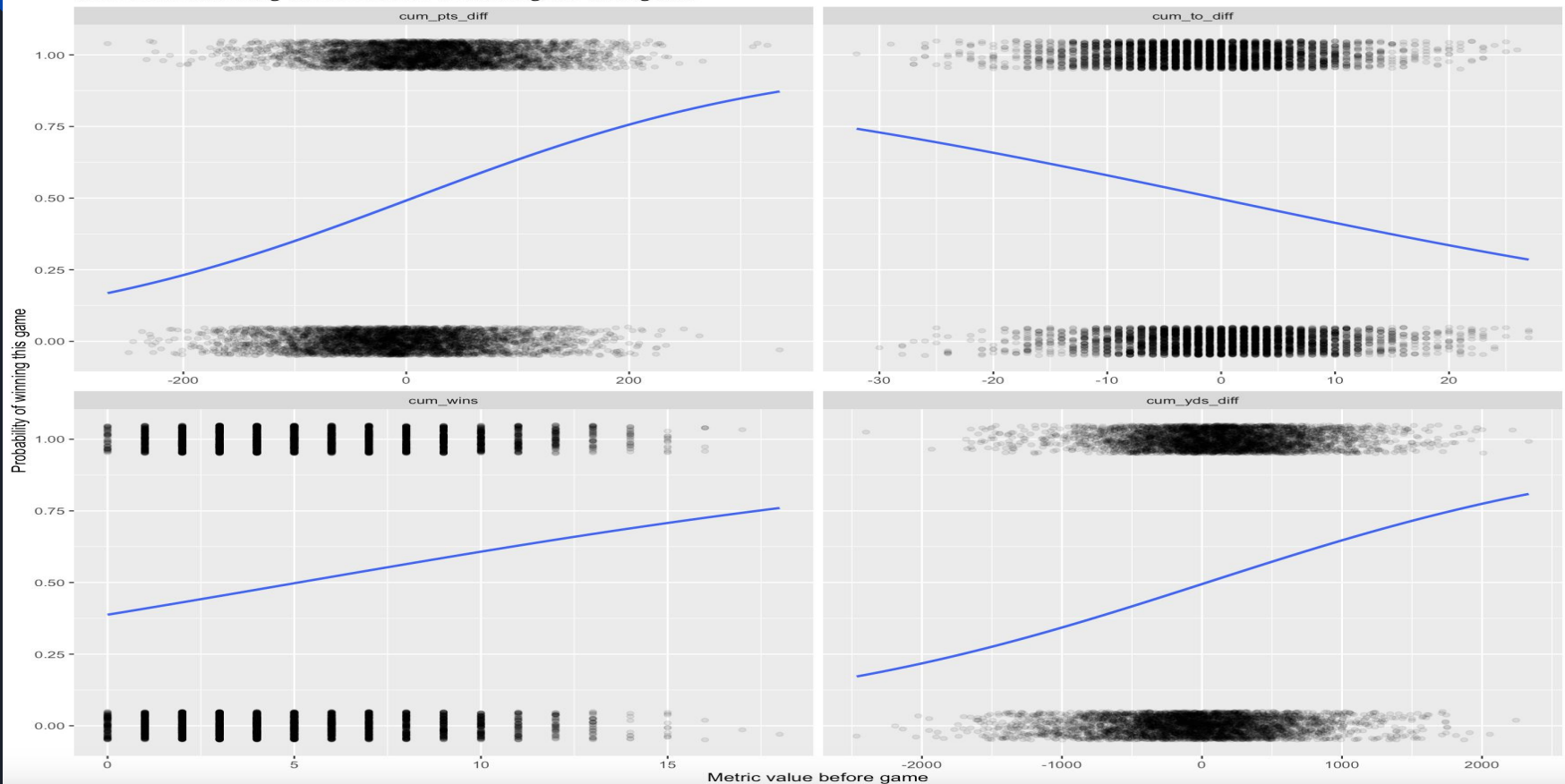
    # optional averages & win% -----
    avg_yds_for      = if_else(
      games_played_prior > 0,
      cum_yds_for / games_played_prior,
      NA_real_
    ),
    avg_yds_against  = if_else(
      games_played_prior > 0,
      cum_yds_against / games_played_prior,
      NA_real_
    ),
    avg_to_for       = if_else(
      games_played_prior > 0,
      cum_to_for / games_played_prior,
      NA_real_
    ),
    avg_to_against   = if_else(
      games_played_prior > 0,
      cum_to_against / games_played_prior,
      NA_real_
    ),
    win_pct          = if_else(
      games_played_prior > 0,
      cum_wins / games_played_prior,
      NA_real_
    )
  ) %>%
  ungroup()
```

Plot code- Predicting win

```
1   # one column for feature name, one for value
2   feature_long <- games_model %>%
3     select(win_home, delta_win_pct, delta_pts_diff, delta_yds_diff, delta_to_diff) %>%
4     pivot_longer(
5       cols = c(delta_win_pct, delta_pts_diff, delta_yds_diff, delta_to_diff),
6       names_to = "feature",
7       values_to = "value"
8     )
9
10  ggplot(feature_long, aes(x = value, y = win_home)) +
11    geom_jitter(height = 0.05, width = 0, alpha = 0.1) +
12    geom_smooth(
13      method = "glm",
14      method.args = list(family = "binomial"),
15      se = FALSE
16    ) +
17    facet_wrap(~ feature, scales = "free_x") +
18    labs(
19      x = "Feature value (home - away)",
20      y = "Probability home team wins",
21      title = "How each feature relates to home win probability"
22    )
```

Conclusion/results

How different rolling metrics relate to winning the next game





Predictive Power analysis: Game Winner

Insight #1: Rolling metric Strongly Predict next-game outcomes:

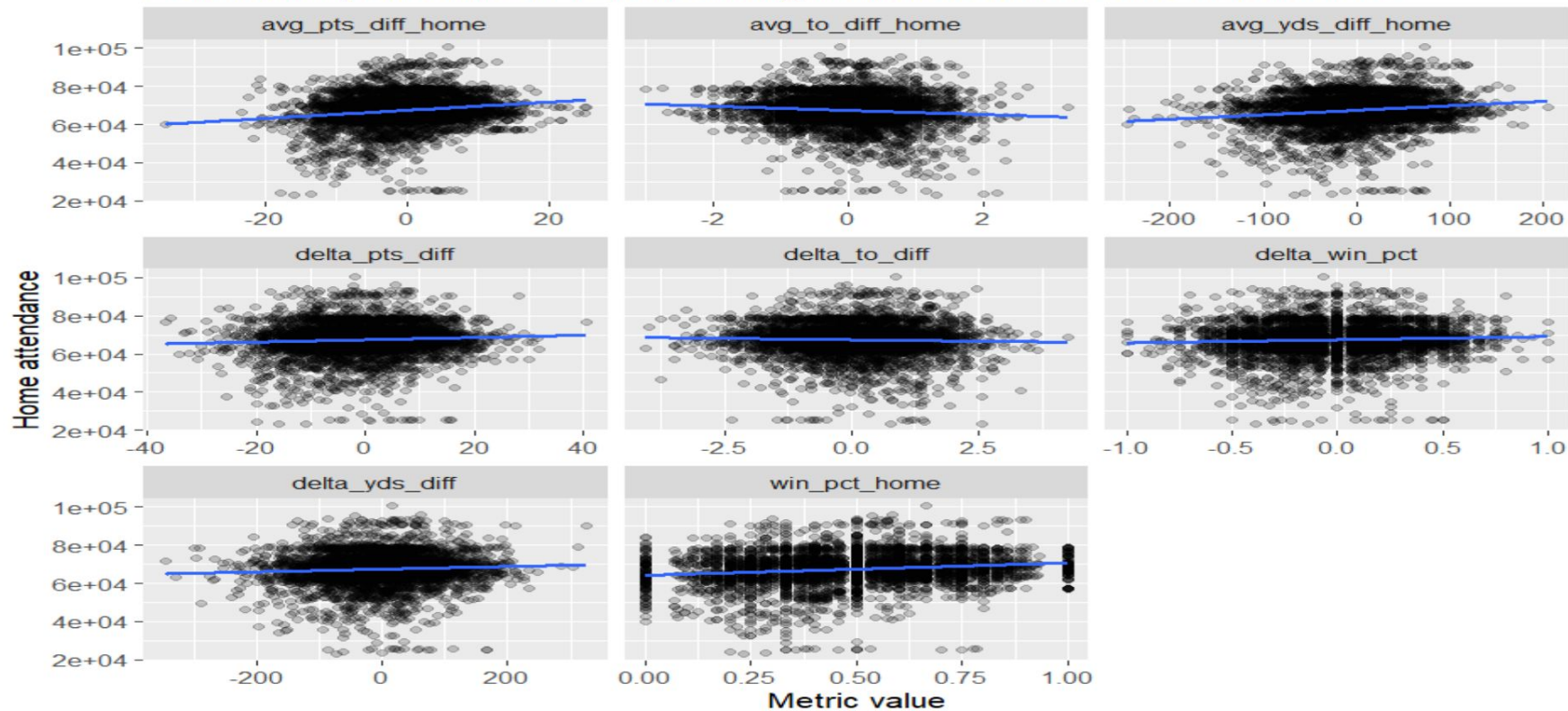
- Higher point differential -> higher win Probability
- Higher Yards differential -> higher win probability
- Turnover differential shows a meaningful negative trend
- Win % difference between teams is also predictive



Plot code - Attendance

```
1 games_att_long <- games_att %>%
2   select(
3     home_attendance,
4     win_pct_home,
5     avg_pts_diff_home,
6     avg_yds_diff_home,
7     avg_to_diff_home,
8     delta_win_pct,
9     delta_pts_diff,
10    delta_yds_diff,
11    delta_to_diff
12  ) %>%
13  pivot_longer(
14    -home_attendance,
15    names_to = "metric",
16    values_to = "value"
17  )
18
19 ggplot(games_att_long, aes(value, home_attendance)) +
20   geom_point(alpha = 0.2) +
21   geom_smooth(method = "lm", se = FALSE) +
22   facet_wrap(~ metric, scales = "free_x") +
23   labs(
24     title = "Relationships Between Team Strength Metrics and Home Attendance",
25     x = "Metric value",
26     y = "Home attendance"
27   )
```

Relationships Between Team Strength Metrics and Home Attendance





Predictive Power analysis: Attendance

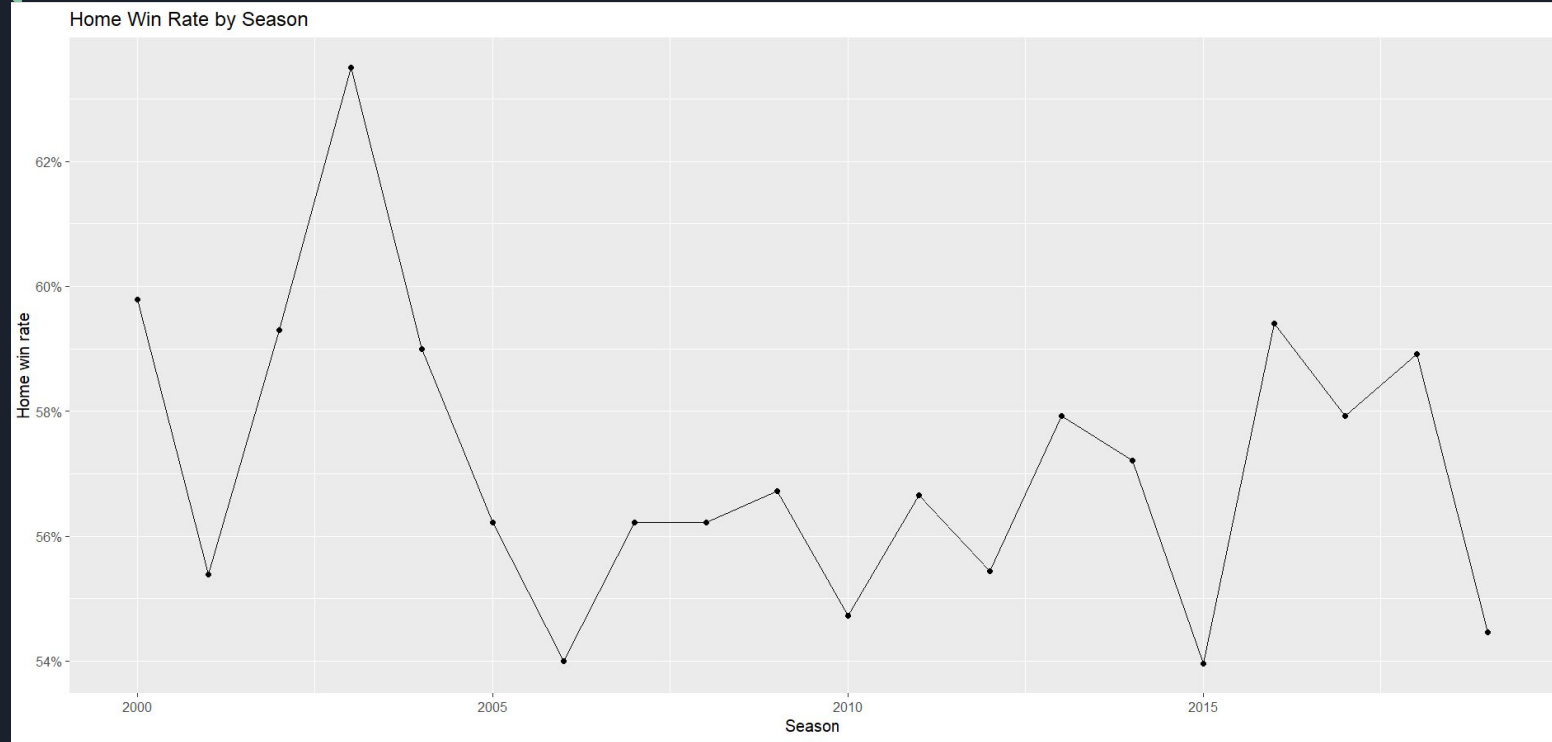
Insight #2: Better Performing teams draw bigger crowds:

- Attendance increases with strong offensive/defensive differentials
- Teams with better turnover differential also draw more fans
- Fans respond to momentum over season-long averages.

Code home effect on winning

```
1  # Overall home win rate
2  games_model %>%
3    summarise(
4      n_games      = n(),
5      home_win_rate = mean(win_home, na.rm = TRUE)
6    )
7  home_by_year <- games_model %>%
8    group_by(year) %>%
9    summarise(
10     n_games      = n(),
11     home_win_rate = mean(win_home, na.rm = TRUE)
12   )
13
14  home_by_year
15
16  ggplot(home_by_year, aes(x = year, y = home_win_rate)) +
17    geom_line() +
18    geom_point() +
19    scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
20    labs(
21      title = "Home Win Rate by Season",
22      x = "Season",
23      y = "Home win rate"
24    )
```

Home vs away effect on winning





Home effect on winning

Insight #3: more likely to win when playing at home:

- From the previous graph it was obvious to see that even after 15 seasons the avg home win % was greater than 50%
- Not only was it greater than 50%, it was greater than 54% for every season.
- This shows picking the home team, has a clear advantage over blindly picking a team



Challenges Faced

- Finding non-kaggle data
- Pivoting the games file created duplicate/overlapping columns
- Ensuring rolling stats only use past games and not current games to prevent data leakage
- Cleaning the data and dealing with missing values



Room For improvements:

- Add more predictive features
 - Injuries
 - Home-Field Advantage
 - Travel distance
 - Create a simple model ie decision tree or random forest to predict winners
 - Include advanced metrics like strength of schedule, quarterback rating etc
- Refine rolling metrics
- Better handling of missing or inconsistent data
- Improve feature alignment between home and away teams



Thank You