NFL Win Projections: Bigger and Better

December 4th, 2025

Justin Choi, Tengis Kelley, Heidi Tamm

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ---------------------------------------- tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ---------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(lme4)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
train = readRDS("training_data.rds")
test = readRDS("test_data.rds")
```

1. Introduction

In the first project of the semester, we attempted to predict NFL win totals for each team using historical indicators of strength, such as previous point differential and strength of schedule. This approach yielded reasonable results, but there was much room for improvement. Notably, we failed to incorporate the actions teams take during a typical offseason to bolster their roster. In this updated version of that project, we focus on (1) **free agent signings** and (2) **amateur draft picks**. Both methods of acquiring players can dramatically elevate the outlook of teams. Marquee free agents are potentially worth multiple wins on their own, while amateur players, particularly those selected in the first round of the NFL draft, are superstars in the making.

We first discuss how we collected necessary data. We then show that knowing which free agents and amateur players a team signs during an offseason provides more accurate predictions about their next-season win total. We conclude with a summary of our findings and avenues for future research.

2. Data Collection

3. Modeling

3.1 Baseline Model

$$
\begin{aligned}
W_i &\sim \text{Binomial}(G_i, p_i)\\
\text{logit}(p_i) &= \beta_0 + \beta_1 \cdot \text{Past PD}_i + u_{Tm[i]}\\
u_{\text{Tm}[i]} &\sim \mathcal{N}(0, \sigma^2)
\end{aligned}
$$

3.2 Adding New Information

$$
\begin{aligned}
W_i &\sim \text{Binomial}(G_i, p_i)\\
\text{logit}(p_i) &= \beta_0 + \beta_1 \cdot \text{Past PD}_i + \beta_2 \cdot \text{Max APY}_i\\
&\quad + \beta_3 \cdot \text{pick}_i + \sum_{j=1}^{J} \beta_j \cdot \text{Position}_{ij} + u_{Tm[i]}\\
u_{\text{Tm}[i]} &\sim \mathcal{N}(0, \sigma^2)
\end{aligned}
$$

3.3 Bayesian Beta-Binomial

$$
\begin{aligned}
W_i &\sim \text{Beta-Binomial}(G_i, p_i, \phi)\\
\text{logit}(p_i) &= \beta_0 + \beta_1 \cdot \text{Past PD}_i + \beta_2 \cdot \text{Max APY}_i + \beta_3 \cdot \text{pick}_i + \sum_{j=1}^{J} \beta_j \cdot \text{Position}_{ij} + u_{Tm[i]}\\
u_{\text{Tm}[i]} &\sim \mathcal{N}(0, \sigma^2)\\
\phi &\sim \text{Exponential}(0.1)\\
\beta_0 &\sim \mathcal{N}(0, 0.25)\\
\beta_1, ..., \beta_j &\sim \mathcal{N}(0, 4)\\
\sigma &\sim \text{Exponential}(1)
\end{aligned}
$$

3.4 Gaussian Process Regression

$$
\begin{aligned}
W_i &\sim \text{Binomial}(G_i, p_i),\\
\text{logit}(p_i) &= \beta_0 + g_i(\text{Year}_t) + u_{\text{Tm}[i]},\\
\beta_0 &\sim \mathcal{N}(0, 0.25),\\
g_i(\cdot) &\sim \text{GP}(0,\, k(\cdot, \cdot))\,,\\
u_{\text{Tm}[i]} &\sim \mathcal{N}(0, \sigma^2),\\
\sigma &\sim \text{Exponential}(1)
\end{aligned}
$$

Now it's time to put our new data to the test. We will train models on NFL regular-season data from 2016-23, in an attempt to predict 2024 NFL regular-seasons wins for each team.

We wanted to start off my establishing a **baseline** model, a starting point from which we can improve on. We'll use the model that returned the best predictions from our previous report: a simple binomial model with three terms: an intercept, the previous season’s point differential, and a team random effect. The model was specified using the lme4 R package:

A summary of the model is provided below:

```
summary(baseline)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(W, L) ~ prior_PD + (1 | Tm)
##    Data: train
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##    1289.6    1300.2    -641.8    1283.6       253
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.91929 -0.96938 -0.00664  0.89125  2.92332
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Tm     (Intercept) 0.0713   0.267
## Number of obs: 256, groups:  Tm, 32
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.0002108  0.0568173   0.004    0.997
## prior_PD    0.2392434  0.0389607   6.141 8.22e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##          (Intr)
## prior_PD -0.001
```

The intercept term is 0, which makes sense - on a logit scale, this corresponds to a team with a .500 winning percentage. When a team allows as many points as it gains, we should expect it to be roughly average. Past year’s point differential has a significant effect on a team’s winning percentage: If a team had a large point differential in the past, it’s more likely to be successful in the future as well (and vice-versa).

But how well does the model do? It’s time to predict on unseen 2024 data and establish a baseline. For our error metric, we’ll use both Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

```
base_prob = predict(baseline, test, type = "response")
base_wins = 17 * base_prob
mean(abs(test$W - base_wins)) # 2.83
```

```
## [1] 2.829519
```

```
sqrt(mean((test$W - base_wins)^2)) # 3.4
```

```
## [1] 3.396982
```

The results are pretty decent - on average, we're around 3 wins away from the true 2024 win totals. The RMSE is slightly higher than the MAE, in part because the 2024 NFL season seemed to feature a good number of unexpected outcomes (for example, the Washington Commanders going from a 4-13 record in 2023 to a 12-5 record in 2024).

Next, we'll keep the same model structure, but add our free agent and draft-related covariates:

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: cbind(W, L) ~ max_apy + prior_PD + pick + category + (1 | Tm)
##    Data: train
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##    1278.6    1321.1    -627.3    1254.6       244
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.83912 -0.97637 -0.02712  0.93520  2.85782
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Tm     (Intercept) 0.07728  0.278
## Number of obs: 256, groups:  Tm, 32
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.007624   0.089603  -0.085  0.93220
## max_apy      0.060916   0.036473   1.670  0.09489 .
## prior_PD     0.211733   0.045665   4.637 3.54e-06 ***
## pick         0.026479   0.040094   0.660  0.50898
## categoryDL  -0.007004   0.105903  -0.066  0.94727
## categoryLB   0.185652   0.129768   1.431  0.15253
## categoryOL   0.040779   0.109117   0.374  0.70861
## categoryQB  -0.194678   0.131178  -1.484  0.13779
## categoryRB   0.562509   0.173091   3.250  0.00115 **
## categoryTE  -0.246516   0.177511  -1.389  0.16491
## categoryWR  -0.069844   0.114115  -0.612  0.54050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) max_py prr_PD pick   ctgrDL ctgrLB ctgrOL ctgrQB ctgrRB
## max_apy     0.035
## prior_PD   -0.029  0.160
## pick       -0.048 -0.212 -0.445
## categoryDL -0.605 -0.084 -0.007  0.071
## categoryLB -0.493 -0.126 -0.039  0.020  0.431
## categoryOL -0.573  0.064  0.076  0.001  0.476  0.384
## categoryQB -0.491 -0.024  0.132  0.148  0.419  0.333  0.395
## categoryRB -0.365  0.016  0.009  0.007  0.304  0.255  0.305  0.245
## categoryTE -0.352  0.112 -0.004 -0.061  0.290  0.247  0.296  0.227  0.197
## categoryWR -0.539  0.017  0.035 -0.025  0.451  0.356  0.454  0.356  0.278
##            ctgrTE
## max_apy
## prior_PD
## pick
## categoryDL
## categoryLB
## categoryOL
## categoryQB
## categoryRB
## categoryTE
## categoryWR  0.277
```

As expected, big free agent contracts. corresponds to a greater win probability the following season. Prior point differential largely maintains its influence, while surprisingly, the draft pick number is almost completely uninformative. Also surprising is first-round running backs (categoryRB) giving their new teams a noticeable boost to their win totals, but other positions not contributing much. Before running the model, we expected the quarterback position to have a similar impact. Perhaps there are far more rookie quarterbacks who flame out than dominate the league a là Tom Brady or Patrick Mahomes.

4. Results

4.1 Model Summaries and Diagnostics

4.2 Predictive Capability

5. Conclusion / Limitations