Predicting Plays in Regular Season NFL Games with Bayesian Logistic Regression

Thomas Benacci
EN.553.632 Bayesian Statistics Fall 2024
Johns Hopkins University
tbenacc1@jh.edu

Abstract

In NFL games, a number of entities stand benefit if they can predict if the next play will be a pass or a run. Previous literature has explored the predictability of plays using frequentist implementations of neural networks and random forests. This paper describes the use of publicly available data and a Bayesian logistic regression methodology to deliver an interpretable model. Using play-by-play and EA Madden Ratings data between 2013-2023 seasons, a Bayesian logit model achieved an accuracy of 72.3%, an improvement over the league-wide probability of a pass, 61.2%. Overall, the Bayesian logit approach delivers an interpretable model while remaining competitive with black-box model performances.

1 Introduction

In American NFL games, defending teams have an advantage if they can correctly predict whether the team with possession will pass or run the ball. Much of the judgment necessary for this task is ingrained in the minds of professionals and avid fans. Even without this judgment, the proliferation of accessible data and extensive literature on statistical learning methods enables spectators off the field to make informed predictions by modeling the league-wide play system in the NFL as a classification problem. There are papers that have modeled this problem and delivered valuable insights using frequentist machine learning methods, most notably including neural networks and random forests. The consensus among these experiments is that neural networks deliver the highest out-of-sample accuracies ranging between 75.3% and 80.0% (Fernandes, 2020), (Goyal, 2019). Given the difficulty to interpret the high performing models, an alternative approach in the form of a Bayesian logistic regression is pursued. The focus of this paper will be on developing a league-wide Bayesian method to predict whether the next play will be a pass or not. This will be done using a set of variables that track in-game data, team performance statistics, and player ratings over a period of 10 years.

Prediction research in sports analytics treads the line between accuracy and interpretation. To supplement their machine learning methods, Fernandes, et al. (2020) show that a simple decision tree, one that could fit on the iPad screen of a coach, is able to capture 86% of the prediction accuracy of their neural network. This paper is tailored to fans who are remotely watching the game and have access to a computer with which they can track and update variables live while monitoring one or more games. To do so, data is collected and processed in a method as similar as possible to that described by Fernandes, et al. (2020). After observing some characteristics that describe the NFL pass process during regular season games, it is demonstrated that a Bayesian logistic regression performance characteristics are competitive with those of models presented in previous literature while being easy to interpret. That being said, these performance characteristics were attained on a sub-sample of the available data, with the under-sampling done to expedite the MCMC sampling process.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Using a random draw of 50,000 plays from a training set of 356,000 plays with 44 predictors collected between seasons 2013-2022, it is shown that a simple implementation of Bayesian logistic regression using weakly informative priors and the Metropolis sampling algorithm achieves an accuracy of 72.3%.

2 Literature Review

In the study "Predicting plays in the National Football League," Fernandes et al. developed machine learning models to predict whether an NFL play would be a pass or a rush, using data from 1,034 games and 130,344 plays between the 2013–2014 and 2016–2017 seasons. Their neural network model achieved a prediction accuracy of 75.3%, while a simpler decision tree model captured 86% of this accuracy, offering a more interpretable solution for real-time application by coaches and players. Additionally, team-specific decision tendencies further improved predictive capabilities, demonstrating the importance of contextual data in sports analytics.

In his 2020 thesis, "Leveraging Machine Learning to Predict Playcalling Tendencies in the NFL," Udgam Goyal applied four machine learning models to NFL play-by-play data from 2009 to 2018 to predict whether a team would execute a run or pass play. The most accurate league-wide model achieved a test accuracy of 80%, while the best team-specific model reached 86%, indicating that team-specific data enhances predictive performance. The study also found that teams become more predictable as games progress and that less predictable teams tend to have greater offensive success.

3 Data

In an attempt to align with data used in similar works, the same sources used in Fernandes, et al. (2020) were used by the author. Likewise, similar variables and calculations were replicated to the best of the author's understanding of earlier works [1][3]. Using the two sources of data described below, 23 binary and 21 continuous predictor variables that describe possession team pass (run) ability, opponent pass (run) block ability, in-game statistics and events, and field position.

3.1 Play-by-Play Data

Play-by-play data were obtained using the *nflfastR* R library. This library is carefully maintained and hosts data that goes back to the 1999 season. Some of the raw features necessary for calculating predictors include season, seconds remaining (quarter, half, game), yards to go to first down, yards from endzone, formation, down, home team status, play type, score, yards gained, huddle status, and more. These raw features were used to calculate the majority of predictors described in Table 1.

3.2 EA Madden Ratings Data

EA Madden player rating data were obtained from maddenratings.weebly.com. Player ratings used in the EQ video game series use data from the previous season to compute player ratings for the current in-game season. For example, real data from the 2012-2013 season is used to calculate the ratings used in the Madden 2014 game. With this in mind, Madden 2014 is used as the source of player rating data for the real life 2013-2014 season. This is a valuable data source, but it is prone to inaccurately conveying player performance as the season progresses. As time goes on, player performance may change radically due to injuries or other outside factors.

3.3 Pre-Processing the Data

The first eight variables in Table 1 summarize the ability of the possession team to pass (run) the ball and the defensive team's ability to block a pass (run). QB rating was the only value calculated by taking a single maximum rating value between QB players on a team in a given season. Otherwise, ratings were calculated by taking the average of the top 3 players in a given position. The difference rating variable for offense quantifies the disparity between a team's ability to run and pass by subtracting RB rating from WR rating, with similar logic applied to the defense difference rating. Beyond ability ratings, down, quarter, and home team status are made binary variables. Regarding

the previous play, binary variables identify pass, run, kickoff, QB spike, extra point, punt, and field goal plays. Previous yards gained, in-game pass percentage, successful in-game pass percentage, in-game average pass yards per pass play, in-game average run yards per run play, in-game pass yard standard deviation, and in-game run yard standard deviation are calculated and updated as game data accumulates. Field goal range is a binary heuristic set within 50 yards of the defense team's end zone. Whether the team did a huddle before formation and whether that formation was a shotgun formation are recorded as binary variables. Yards to go until first down, yards to go to the defense team's end zone, and the score difference are recorded as continuous variables. Finally, the red zone is set within 20 yards of the defense team's end zone as a binary variable.

With the variable values calculated and split between train and test sets, continuous columns were scaled using the `scale()` function in R, which centers each column by subtracting the mean and scales by dividing by the standard deviation. The resulting values have a mean of 0 and a standard deviation of 1. This allows for even an interpretation of the linear influence coefficients have on the dependent variable.

Table 1: Predictors and their Data Types

Predictor	**Type**	**Predictor**	**Type**
ratingQB	Continuous	prev_qb_ks	Binary
ratingWR	Continuous	prev_ep	Binary
ratingRB	Continuous	prev_punt	Binary
ratingWRRB	Continuous	prev_field_goal	Binary
ratingLB	Continuous	prev_yds_gain	Continuous
ratingDefPass	Continuous	ig_pass_pct	Continuous
ratingDefRun	Continuous	ig_pass_success_pct	Continuous
ratingDefDiff	Continuous	ig_avg_pass_yds	Continuous
1d	Binary	ig_avg_run_yds	Continuous
2d	Binary	pass_sd	Continuous
3d	Binary	run_sd	Continuous
4d	Binary	fieldgoalrange	Binary
1qtr	Binary	lt_3min_half	Binary
2qtr	Binary	no_huddle	Binary
3qtr	Binary	shotgun	Binary
4qtr	Binary	qtr_sec_remain	Continuous
home	Binary	half_sec_remain	Continuous
prev_pass	Binary	game_sec_remain	Continuous
prev_run	Binary	ydstogo	Continuous
prev2_run	Binary	yardline_100	Continuous
prev2_pass	Binary	score_diff	Continuous
prev_kickoff	Binary	redzone	Binary

4 Proposed Methodology

In this section, the prior assumptions, likelihood model, posterior, and sampling process are described. For these descriptions, let $\boldsymbol{\beta}$ be the coefficient vector, $\mathbf{X}$ be the data matrix, $\mathbf{y}$ be the binary target variable vector, and π_i is the probability that $y_i = 1$. Using Bayes' theorem, the posterior is expressed as:

$$p(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}) \propto p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}) \cdot p(\boldsymbol{\beta}),$$

the likelihood as $p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta})$, and the prior as $p(\boldsymbol{\beta})$.

4.1 Likelihood Function

The logistic regression model assumes the response $y_i \in \{0, 1\}$ follows a Bernoulli distribution with success probability:

$$\pi_i = \frac{1}{1 + e^{-\mathbf{X}_i^\top \boldsymbol{\beta}}}$$

The likelihood function for N observations is:

$$p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}) = \prod_{i=1}^{N} \pi_i^{y_i} (1 - \pi_i)^{1 - y_i}$$

Taking the natural logarithm (log-likelihood):

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} [y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i)]$$

4.2 Prior Distribution

The author specifies a Cauchy prior for $\boldsymbol{\beta}$ with location 0 and scale 2.5:

$$p(\boldsymbol{\beta}) = \prod_{j=1}^{K} \text{Cauchy}(\beta_j \mid 0, 2.5)$$

The log-prior is:

$$\log p(\boldsymbol{\beta}) = \sum_{j=1}^{K} \log \left[\frac{1}{\pi(1 + (\beta_j / 2.5)^2)} \right]$$

This choice of priors was made at the recommendation of Gelman et al. (2008) in the case of routine data analysis.

4.3 Posterior Distribution

The log-posterior is the sum of the log-likelihood and the log-prior:

$$\log p(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}) = \ell(\boldsymbol{\beta}) + \log p(\boldsymbol{\beta})$$

4.4 Metropolis Sampling

The Metropolis algorithm is used to sample from the posterior distribution of $\boldsymbol{\beta}$:

1. **Initialization:** Start with an initial value $\boldsymbol{\beta}^{(0)}$.
2. **Proposal:** Generate a proposed value $\boldsymbol{\beta}^*$ from a Gaussian distribution:
$$\boldsymbol{\beta}^* = \boldsymbol{\beta} + \mathcal{N}(0, \sigma^2 I)$$
3. **Acceptance Rule:** Compute the acceptance probability:
$$\alpha = \min\left(1, \exp\left[\log p(\boldsymbol{\beta}^* \mid \mathbf{y}, \mathbf{X}) - \log p(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X})\right]\right)$$
Accept $\boldsymbol{\beta}^*$ with probability α; otherwise, retain $\boldsymbol{\beta}$.

4.5 Multiple Chains

To improve convergence diagnostics, multiple independent chains are run. Each chain starts with a different initialization, $\boldsymbol{\beta}^{(0)} \sim \mathcal{N}(0, 1)$, and follows the Metropolis algorithm. In the author's run, 5 independent chains with 10,000 iterations each were used, resulting in 25,000 posterior samples to estimate the model coefficients after dropping the first half of each chain.

4.6 Final Model

After discarding the burn-in period, posterior samples are used to compute summaries such as posterior means:

$$\hat{\boldsymbol{\beta}} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\beta}^{(m)}$$

where M is the number of retained samples. These posterior means serve as the coefficients in the resulting function: Using the estimated posterior mean $\hat{\boldsymbol{\beta}}$, the resulting logit model for the probability π_i that $y_i = 1$ is:

$$\pi_i = \frac{1}{1 + e^{-\mathbf{X}_i^\top \hat{\boldsymbol{\beta}}}}.$$

5 Diagnostics and Performance

5.1 MCMC Convergence Diagnostics

The posterior sample results in Table 2 strongly suggest that the simple Metropolis algorithm struggled to converge upon stationary posterior distributions for the regression coefficients.

First, the posterior samples for most coefficients exhibit significant dispersion around their means, as evidenced by the large standard deviations and wide confidence intervals. Zero falls within 43% of the 90% confidence intervals, highlighting the uncertainty in the estimates. This suggests that the sampler failed to efficiently concentrate around meaningful posterior modes.

Second, the effective sample size values are alarmingly small compared to the total number of samples drawn. Such small ESS values indicate severe autocorrelation within the chains, reducing the effective number of independent samples. This implies that the chains are not exploring the posterior distribution effectively, likely getting "stuck" and failing to mix. Figure 1 provides evidence that increasing the number of chains reduces the impact of a single non-convergent chain, and thus is the best way to guarantee robust sample estimates.

Finally, the Gelman-Rubin statistic ($\hat{R}$) reveals significant between-chain variance. Many parameters exhibit high $\hat{R}$ values, with many greater than 10. These large values indicate that the chains are not converging to the same posterior distribution, with substantial variance persisting between chains.

In summary, the rudimentary nature of the Metropolis sampler is likely a key reason for these convergence issues. The previously mentioned shortcomings could be remedied via a more sophisticated sampling algorithm.

Table 2: Summary of Posterior Samples with 90% CI

Beta	**Mean**	**SD**	**CI Lower**	**CI Upper**	**ESS**	**R hat**
ydstogo	0.48	0.08	0.42	0.54	31.36	2.99
yardline_100	-0.08	0.31	-0.31	0.15	36.28	9.86
WRRB	0.37	0.92	-0.31	1.05	12.55	44.97
shotgun	1.52	0.19	1.38	1.66	22.51	9.01
score_diff	-0.26	0.08	-0.32	-0.20	32.30	2.83
run_sd	0.08	0.11	-0.01	0.16	29.49	10.37
redzone	-0.32	0.36	-0.58	-0.06	24.30	7.76
ratingWR	-0.25	0.74	-0.79	0.30	16.04	24.29
ratingRB	0.23	0.64	-0.24	0.70	20.95	27.28
ratingQB	0.05	0.09	-0.02	0.11	13.05	5.55
ratingLB	0.01	0.07	-0.04	0.07	18.15	3.26
ratingDefRun	-0.16	0.61	-0.62	0.29	24.63	14.41
ratingDefPass	0.14	0.55	-0.27	0.54	39.48	23.83
qrt_sec	-0.04	0.06	-0.08	0.01	29.49	2.93
prev2_run	-0.06	0.17	-0.19	0.07	18.06	4.75
prev2_pass	0.17	0.22	0.01	0.33	26.60	5.95
prev_yds_gain	-0.10	0.08	-0.16	-0.04	20.66	2.83
prev_run	-0.05	0.56	-0.46	0.36	37.80	21.03
prev_qb_ks	-1.01	1.04	-1.78	-0.24	16.25	56.43
prev_punt	-0.14	0.60	-0.58	0.30	12.42	16.51
prev_pass	-0.14	0.60	-0.58	0.30	22.99	23.36
prev_kickoff	-0.23	0.68	-0.74	0.27	27.90	16.90
prev_field_goal	-1.10	1.46	-2.17	-0.03	10.97	45.16
prev_ep	0.09	0.60	-0.35	0.53	25.74	14.12
pass_sd	-0.05	0.08	-0.11	0.01	23.28	2.73
no_huddle	0.45	0.25	0.27	0.64	21.09	9.79
lt_3min_half	0.11	0.14	0.01	0.21	27.42	4.39
Intercept	0.32	0.20	0.17	0.47	15.44	7.39
ig_pass_suc_pct	0.00	0.13	-0.10	0.09	20.32	9.31
ig_pass_pct	0.00	0.05	-0.04	0.04	24.93	1.48

5.2 Performance

Despite exhibiting endemic difficulties from a convergence perspective, the final model's predictive qualities not only mark a significant improvement over the baseline pass probability of 61.2%, but reasonably balance accuracy with model transparency. To make pass or run predictions using probability π_i provided by the model, $\pi_i \geq 0.612$ is counted at a pass.

Using the 2023 NFL regular season as a test set with 33,386 plays, the Bayesian logit model achieved an accuracy of 72.3%, precision of 58.5%, sensitivity of 66.2%, and specificity of 75.4%. When compared to similar methodologies that use black-box models where it is impossible to draw linear relationships between coefficient value and the log odds, the Bayesian logit model delivers satisfactory results.

6 Conclusion

This study explored the use of a Bayesian logistic regression to predict whether the next play in an NFL game would be a pass or a run. Using publicly available play-by-play data and EA Madden player ratings spanning ten seasons, a predictive model was developed with 44 predictors representing team and player abilities, in-game statistics, and field position.

The proposed methodology utilized weakly informative Cauchy priors and the Metropolis algorithm for sampling. Diagnostics collected indicated significant convergence challenges, including low effective sample sizes and high Gelman-Rubin statistics. These limitations highlighted the inefficiency of the Metropolis sampler in exploring high-dimensional posterior spaces.

Nevertheless, the model achieved an accuracy of 72.3% on the 2023 regular season test set, improving upon the baseline pass probability of 61.2%. In addition to accuracy, the model delivered reasonable precision, sensitivity, and specificity while remaining interpretable, a key advantage over black-box methods such as neural networks. The Bayesian approach provided insights into the relationships between predictors and outcomes, offering a transparent alternative to machine learning models.

In summary, this paper demonstrates that Bayesian logistic regression can strike a balance between interpretation and predictive performance in sports analytics. Future work could address convergence issues by employing more advanced sampling techniques, such as Hamiltonian Monte Carlo, and expanding the dataset to include team-specific play tendencies for improved predictions.

References

[1] Fernandes, C.J., Yakubov, R., Li, Y., Prasad, A.K., & Chan, T.C.Y. (2020). Predicting plays in the National Football League. *Journal of Sports Analytics*, 6, 35–43. DOI: 10.3233/JSA-190348.

[2] Gelman, A., Jakulin, A., Pittau, M., Su, & Su, Y. (2008). A weakly informative default prior distribution for logistic and other regression models. *Annals of Applied Statistics*, 2(4), 1360-1383. DOI: 10.1214/08-AOAS191.

[3] Goyal, U. (2019). Leveraging Machine Learning to Predict Playcalling Tendencies in the NFL. Master's Thesis, Massachusetts Institute of Technology. Retrieved from `https://dspace.mit.edu/handle/1721.1/129909`.

Appendix

Figure 1: Sample Histograms for the Intercept Coefficient

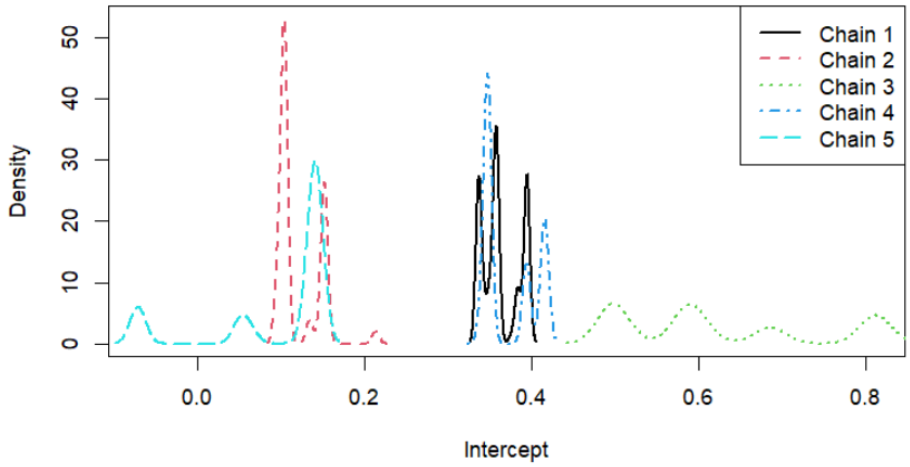

Figure 2: Sampled Coefficient Means with 90% CI