“The Bounce” in NFL Football: Final Report



**Table of Contents**

[Problem Statement 2](#_Toc64920140)

[Data Collection 2](#_Toc64920141)

[Data Wrangling 2](#_Toc64920142)

[Data Quality 2](#_Toc64920143)

[Useful Data 2](#_Toc64920144)

[Feature Engineering 3](#_Toc64920145)

[Exploratory Data Analysis 5](#_Toc64920146)

[Machine Learning Analysis 5](#_Toc64920147)

[Data Preparation 5](#_Toc64920148)

[Logistic Regression 6](#_Toc64920149)

[Random Forest 6](#_Toc64920153)

[XGBoost Classification 7](#_Toc64920154)

[Conclusion 8](#_Toc64920155)

[Looking Forward 9](#_Toc64920156)

# Problem Statement

Sports betting lore includes a concept called “The Bounce” or “The Bounce-Back Effect” whereby a team which is favored in a game and loses outright comes back in the following game with a vengeance and is more likely to win.

We are going to investigate this in American NFL Football and see whether it is truly predictive and whether it is sufficiently so to provide an edge in betting. For a useful edge in betting, it must predict whether the team will “cover the spread” – or beat the point differential expected for the game.

While we are at it, we will run the rest of the available features through a series of algorithms to test for predictive power elsewhere in the data.

# Data Collection

The data used for this analysis is the freely available Kaggle NFL Game and Betting Data. It includes all NFL games with results and betting spreads, complete since 1979.

We will be using regular season games from then to the 2019 season and determine initially whether the bounce effect exists.  This will require some feature engineering to identify which games are bounce candidates.

We will also use the available data to look for any other edge that may present itself.

# Data Wrangling

## Data Quality

As it was a Kaggle supplied dataset, the data was fairly complete and had few unexpected values. Game data for games prior to 1979 did not reliably include point spread information so the range of seasons from 1979 to 2019 was used.

## Useful Data

Pre-season games, playoff games and championship games were not used as they have different stakes and would deserve individual treatment.

As each game had a home and away team in the data, we repeated and adapted the data to provide a game profile and result from the perspective of each team with home or away specified.

Data was also merged with the Teams data included with the dataset to import team identification numbers as opposed to names which are prone to change.

# Feature Engineering

Primary in feature engineering for our “Bounce” investigation was to create the *bounce\_candidate* feature. This is simply a calculation of whether a team was favored in the prior game but lost outright. In the process, we can establish a categorical favored variable based on the point spread for each game.

*Covered\_by* and *covered\_over\_by* are both calculations of how each team did relative to their anticipated point differential. If a team was favored by two points and won by four points, then *covered\_by* would be +2. *Win\_by* works the same way, without the point spread, and push indicates the differential was equal to the point spread.

Pushes (point differential equals the point spread) were removed as well as “Picks” (point spread of zero) as they are not easily faithfully categorized.

Other performance features such as *total\_wins, total\_losses, win\_loss\_ratio, net\_wins*, *pfa* (points for an against) *ratio/net/average* are also calculated as of that point in each season.

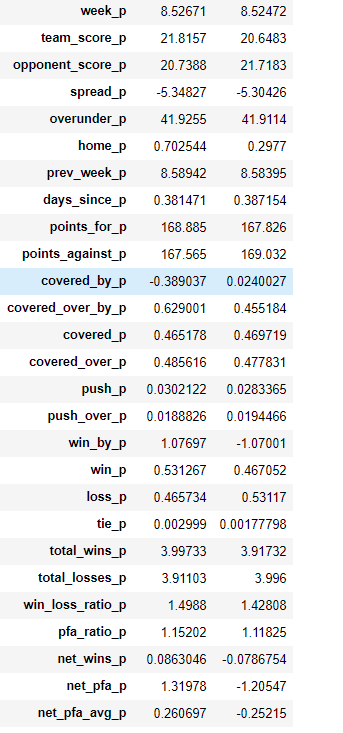
Using the dates in the data we can add features for the number of days since the last game played – *days\_since*, and we can create features identifying the day of the week of the game in question.

The day of the week can be pushed into a categorical feature like *day\_of\_week\_Thursday* via pd.get\_dummies. The *days\_since* will also correlate with the *bye* feature which indicates whether the team had a week off the prior week.

Many values are not available at game time. We use them to populate the features referring to the prior week. Once we have these current game related features, we need to join the data to itself in order to populate this information for the next game. Since these features are not available until the game is played, they are only useful values referring to the prior week. These feature names are followed with a “\_p” such as *covered\_by\_p*. After the prior week values for these features are populated, the unavailable current week values can be dropped as they would not be available in time to predict an outcome.

Illustration of current and prior (“\_p”) features below showing mean values for home and away.



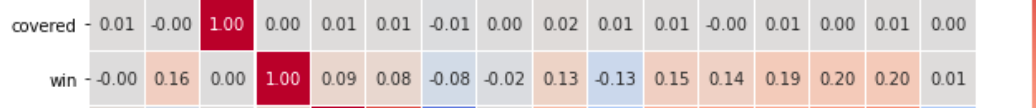
 

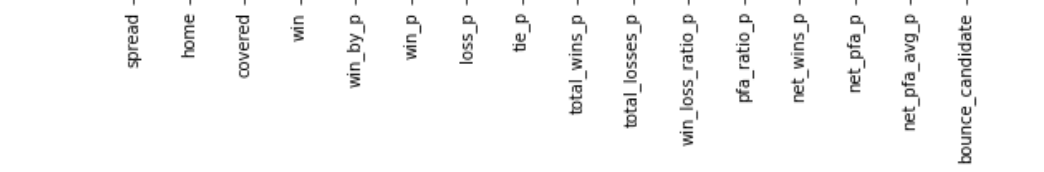
# Exploratory Data Analysis

Exploratory data analysis did not turn up any significant unexpected useful correlations. In fact it didn’t turn up any superstars acting on their own.

Running a correlation matrix for all of these variable results in a very large grid with only small correlations between the features and the win and cover variables.

Running the matrix on subsets of the features yields similar results. There are some weak correlations among the variables predicting the *win* and almost nothing individually predicting the *cover*





Since there are no individual standouts as predictors for the win or cover, we will run some multivariate analyses to search for combined influence.

# Machine Learning Analysis

## Data Preparation

The data is divided into a training sets and two testing sets. These are divided separately so we can run each of the models to predict both the *win* outcome and the *cover* outcome. The test sets use 33% of the data.

A Variance Inflation Factor (VIF) function was applied to each training dataset to remove features which were highly correlated with other features in order to clarify the dataset. This reduces the feature count from 38 to 22 and will prevent prediction errors related to inflation of impact of a feature that is essentially duplicated.

## Logistic Regression

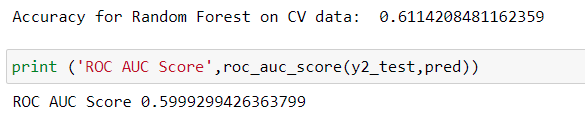
## Logistic Regression run on both datasets returned an accuracy of .71 for predicting *wins* and 0.54 for predicting *covers*. This is a baseline to evaluate the performance of more advanced models.

## 

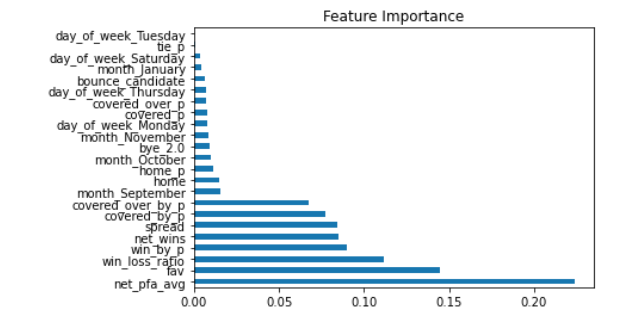
## As one might expect, the wins are relatively easy to predict but the covers are not.

## Random Forest

Random Forest returned a similar accuracy for the win predictor but did a better job predicting covers (61%). Adding a CVGridSearch to optimize the Random Forest parameters returned a model performance with a fair AUC Score (.59%). Indicating a good balance of predictions without any large bias toward a particular type of error.

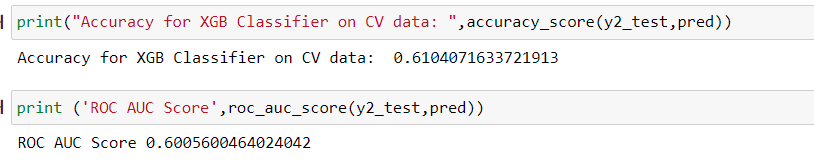


Random Forest also gave us some indication of the importance of the individual features.

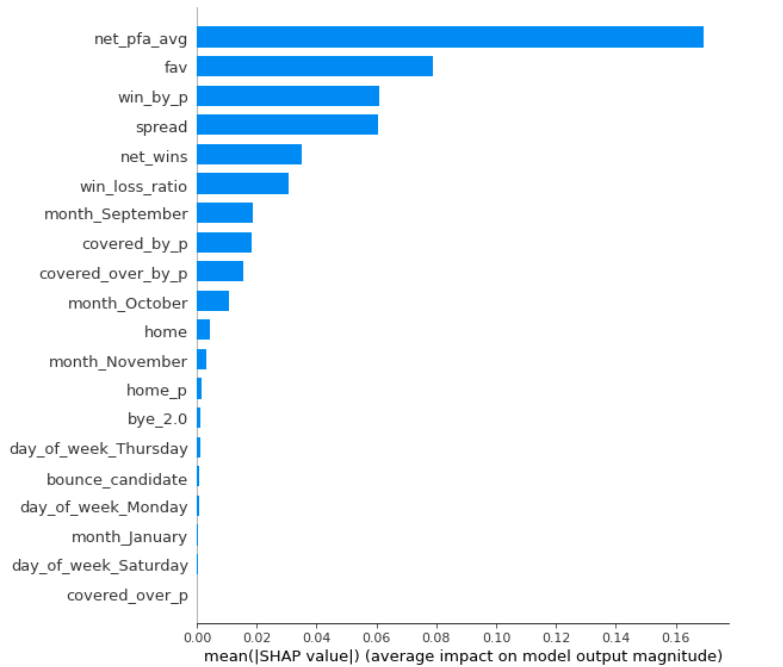


## XGBoost Classification

XGBoost Classification (Extreme Gradient Boosting) used in conjunction with a grid search methodology to optimize the model parameters returned the best results.



A plot of SHAP values below illustrating the relative impact of data features shows that our original *bounce\_candidate* is relatively inconsequential. There are however a fair number of readily available features that do figure prominently.



The optimized XGB Classified provides the best ROC\_AOC Score and has the highest accuracy of the models tested. The SHAP summary plot illustrates the relative impact of each of the features on the prediction. "Bounce Candidate", (the original feature under investigation) survived the VIF multicollinearity test but ranked very low on the scale of impact or direction of impact.

Whether a team wins or loses is predicted fairly easily in any of these models, but overcoming the spreads assigned by professionals predicting that outcome is shown to be more difficult as expected. As the "Vigorish," or percentage charged on winning wagers is usually around 10%, it is usually assumed that the bettor can make a profit with a winning percentage over 55%. As this model shows an accuracy of 60%, it is possible that a disciplined bettor could come out ahead betting with this model

Of the features shown to be impactful for the SHAP plot, there are no big surprises. The biggest

impact comes from the engineered feature net\_pfa\_avg. This is the average net point differential for the team. If on average, they win by two points, that would be the net\_pfa\_avg. This tends to impact the prediction on the positive side and interestingly sometimes does so with low values, indicating perhaps that emotional predictors do not expect teams that don't win by large margins to cover spreads - which may influence the setting of the spreads downward.

Being favored to win (fav) is the next most impactful feature and tends to impact the prediction

negatively. This could also speak to the emotional appeal of picking a favorite.

Win\_by\_p is the net point differential from the prior week and was somewhat predictive positively and negatively with very low values impacting negatively

Finally Spread is the most clearly positively or negatively impacting based on the size of the

spread. Negative spreads are negatively impacting. Since they are inverse to their meaning; a team with a -10 spread is expected to win by 10 points which is a big spread but appears negative to the model. As you might expect, a lot more fate can intervene to prevent a big spread from occurring so it makes sense that they would be less likely to cover.

There are some impactful features that can be used to predict whether a team might cover

the spread. The bounce does not appear to be among them.

# Conclusion

“The Bounce” did not appear to be a real phenomenon either regarding wins or covering the point spread. It has great logical and emotional appeal but has not been demonstrated to be a predictor of results for NFL Football games.

Other readily available information may be useful though in predicting both wins and covers. Using XGBoost Classification we were able to successfully predict whether a team would cover the point spread with accuracy exceeding 61% and an AUC Score over .60. This represents a usable model and given the typical 10% fee charged by sportsbooks, could theoretically be used to win bets by predicting whether or not a team will cover.

Sports betting requires substantial personal discipline in money management and bet selections, so of course there are no guarantees and this is no golden ticket. It does however give us some interesting insights into what factors are important and the potential of this type of analysis to generate a useful prediction model.

## Looking Forward

There are any number of available statistics that can be employed to improve this analysis. Some simple ones that might be possible with a similar process could be the following:

### Include Divisions

The NFL is divided into 8 divisions. Those divisions play multiple games within the division in order to top their division to make the playoffs. Divisional vs. Inter-divisional games involve rivalries and different play for familiar teams. Including this feature may be useful.

### Passing vs Rushing

Some scoring is done by passing and some by rushing (running). This history combined with the history of allowing a certain number or each type of scoring for an opponent could be a useful way of predicting how teams with different types of play might match up.