ECEN 360

FINAL PROJECT REPORT

BETWEEN THE LINES: A DEEPER LOOK INTO NFL BETTING

ILAN SELA: ECEN, isela1@tamu.edu, ML Specialist

ALEX MANDANIS: ECEN, alexmandanis@tamu.edu, Football Specialist

1. Executive Summary

This project explores the building of a successful model for the NFL over/under bet. Historical NFL play-by-play data is processed to train and test these models. This data first undergoes feature selection based on predictive and cross correlation. Utilizing the optimal features, four models were implemented: MLP, random forest, logistic regression, and SVM. The highest performing models have a validation accuracy of 53%, surpassing the initial lower bound set at 52.4% at the beginning of the project.

The implementation and code is available here: Shared NFL.ipynb

2. Introduction

Since a landmark Supreme Court Case in 2018, sports gambling has taken a seat in the spotlight of everyday Americans [1]. Sports talk TV and radio shows, as well as podcasts and even sports broadcasts themselves are now filled with ads for the various firms. And these sports betting firms have raked in tens of billions of dollars and are growing each year [2]. However, these bets are priced very optimally, and it often costs the average American a lot in lost bets or subscriptions to gain an edge in the betting market.

This project aims to look into this growing industry by searching for a model that can take historical data that any sports fan will have access to, to gain an edge. Specifically, it will look at one of the most popular bets, game over/under, in the most popular league, the NFL, and aim to find a model that can generate >52.4% accuracy (very successful in the sports betting world).

The datasets used is from NFLverse, and it contains popular statistical information that is publicly available [3]. Specifically, this project will take use of their play-by-play dataset that tracks every play in every NFL game all the way back through the 1999 season.

The game over/under bet is a bet in which a sports betting firm will put out a total line for a certain game and bettors will bet if the combined score between the two teams will go over or under this line. For example, Super Bowl LIX's line was 48.5 [3]. The final score was 40-22 [3], and thus the over bet hit.

The NFL is short for the National Football League, and it is the preeminent professional American football league in the world.

3. Methodology

3.1 Data Processing

First, the scope of the project was narrowed down to the last 15 seasons, from 2010 to 2024. This was done for two reasons. First, the NFL has changed playstyle drastically over its history, and more recent seasons reflect more recent trends in how the game is played. Second, Google Colab had limited compute space and could not process full play-by-play data from all 26 seasons.

Second, a list of 36 statistics were selected for analysis. This list is composed of common statistics available to the average viewer, and it is not personal to specific players. The stats were: Total Yards Gained, Total Touchdowns, Pass Attempts, Pass Completions, Passing Touchdowns, Interceptions, QB Dropbacks, QB Scrambles, Sacks, Rush Attempts, Rushing Yards, Rushing Touchdowns, Third Downs Failed, Third Downs Converted, Fourth Downs Failed, Fourth Downs Converted, Penalty Yards, Return Yards, Punt Attempts, Field Goal Attempts, Two Point Attempts, Safeties, Tackles for Loss, Fumbles Lost, Aborted Plays, Out of Bounds Plays, Plays in Shotgun Formation, No Huddle Plays, Total Yards to Go, Total Score for Each Team, Whether it was a divisional game or not, Pass Yards Per Pass Attempt, Passing Completion Percentage, Rush Yards Per Rush Attempt, Third Down Conversion Percentage.

For every game, Week 5 onwards, these stats were calculated on a four-game rolling average for each team. Our processed data contained every game played from Week 5 with these stats for each team, as well as the over/under line and the final over/under result.

3.2 Feature Selection

Next, for a stronger model, the top 20 non-highly-cross-correlated features (<0.7) were selected to select great data for the model. In this way, optimal features were chosen while avoiding redundancy.

First, the 20 features with the highest absolute correlation with the label (over/under) were selected. Next, the pairwise correlation between each of these features was measured. If the correlation between any two features was > 0.7, the feature in this pair less correlated with the label was removed. This removed feature was replaced with the next highest correlated feature to the label. This was repeated until 20 features remained.

3.2 Predictive Model

The algorithm was tested on four very popular predictive models: MLP Classifier, Random Forest, Logistic Regression, and SVM. These models are implemented from the sklearn package. For the MLP, two hidden layers of sizes 64 and 32 were used. The random forest implementation utilized 100 decision trees. Lastly, the SVM is an RBF kernel with regularization parameter C as 0.5.

There was a need to differentiate between learning and luck, especially since the aim is for an accuracy only slightly above 50%. To accomplish this, custom PyTorch MLP was implemented and the loss per epoch was tracked. In this way, confirmation of the loss function being minimized could be established.

Regarding the data used, it was split into 80% training and 20% testing, and the stratify was specified to ensure the original class proportions are maintained when splitting the dataset. Additionally, features were standardized to ensure a common scale.

4. Results

The features maintained after undergoing the feature selection process are presented in Figure 1 and Figure 2. Figure 1 is the heatmap of feature correlation, which shows that the features have minimal correlation. On the other hand, Figure 2 shows the correlation between the selected features and label (over/under).

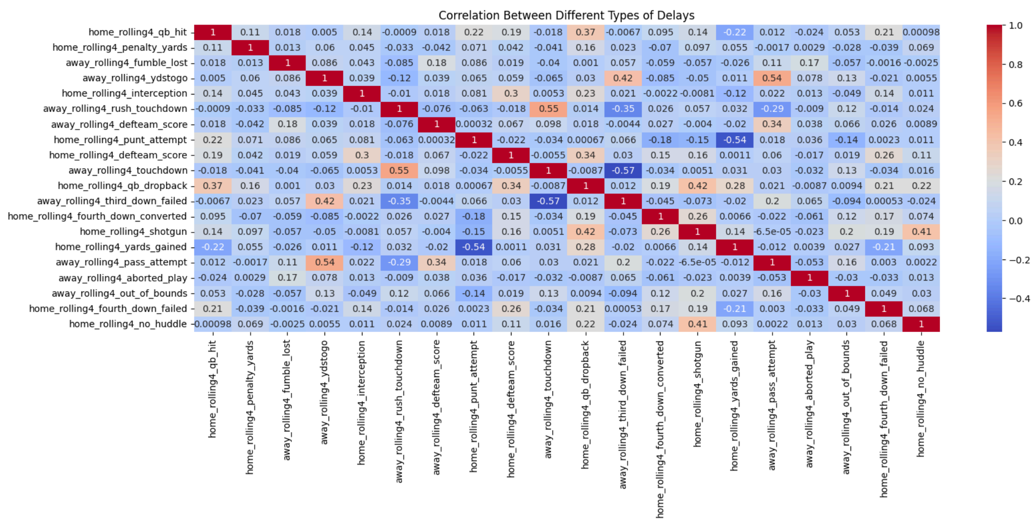

Figure 1: Correlation heatmap between features

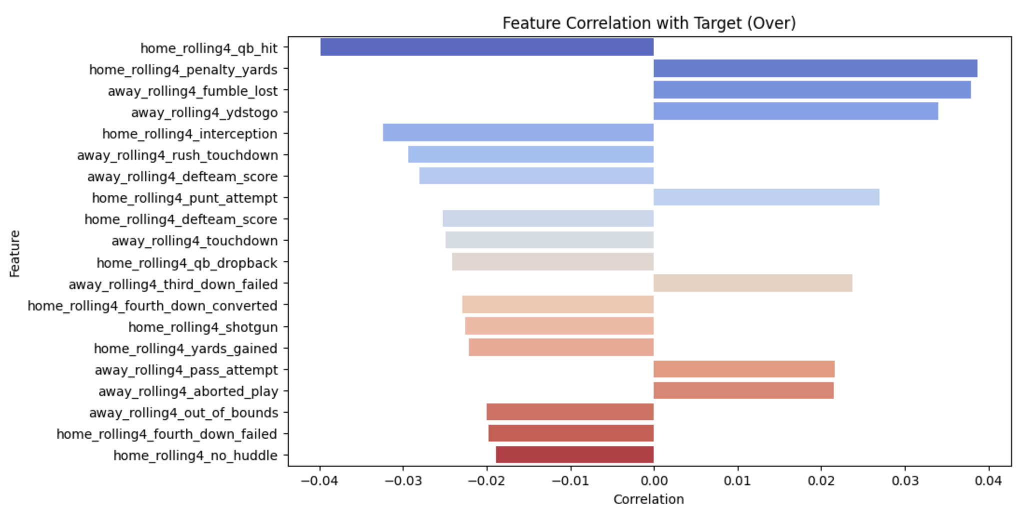

Figure 2: Feature correlation with target

As shown in these figures, the features avoid redundancy with the highest absolute correlation being 0.57. Figure 2 highlights the correlation with the target, the highest absolute correlation being around 0.04 and the lowest being around 0.02.

The predictive model accuracies are plotted in Figure 3.

	precision	recall	f1-score	support
0	0.51	0.50	0.50	411
1	0.49	0.50	0.49	392
accuracy			0.50	803
macro avg	0.50	0.50	0.50	803
weighted avg	0.50	0.50	0.50	803

(a) MLP Classifier with layers (64, 32)

	precision	recall	f1-score	support
0	0.54	0.61	0.57	411
1	0.52	0.45	0.49	392
accuracy			0.53	803
macro avg	0.53	0.53	0.53	803
weighted avg	0.53	0.53	0.53	803

(b) Random Forest

	precision	recall	f1-score	support
0	0.54	0.58	0.56	411
1	0.52	0.47	0.49	392
accuracy			0.53	803
macro avg	0.53	0.53	0.53	803
weighted avg	0.53	0.53	0.53	803

(c) Logistic Regression

	precision	recall	f1-score	support
0	0.52	0.64	0.57	411
1	0.51	0.39	0.44	392
accuracy			0.52	803
macro avg	0.51	0.51	0.51	803
weighted avg	0.51	0.52	0.51	803

(d) SVM (RBF Kernel) with C = 0.5

Figure 3: Model Accuracies

As shown in the figure above, the highest performing models are random forest and logistic regression with accuracies of 53%. Additionally, the loss per epoch of the custom PyTorch MLP model is plotted in Figure 4 to demonstrate the model's capability to learn with the provided features and data.

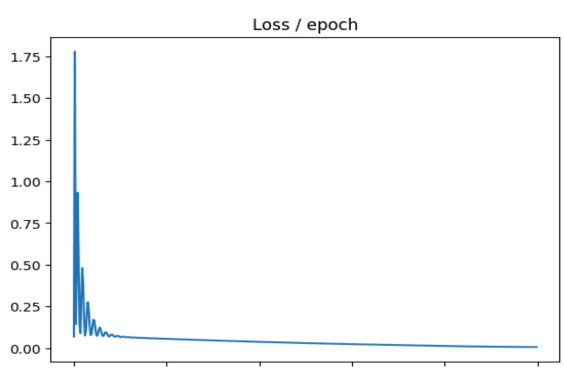

Figure 4: Loss per epoch of custom MLP

5. Discussion

Our results demonstrate that our algorithm for feature selection successfully achieved features that correlate highly with the target and avoids redundancy between features. However, we observe that none of the features in the dataset correlate highly with the target, as the highest correlation is only 0.04. This is probably due to two reasons.

First, this is due to how well-priced these over/under lines typically are. The lines reflect both the extensive research and data collection of the sports betting firms that price the lines, as well as the money bet by the public at large. Additionally, the statistics used individually may not contribute to the final over/under in all games, and they may contribute to an over/under result more collectively than individually.

The project's predictive models, specifically random forest and logistic regression, demonstrate successful predictive capabilities, exceeding the initial lower bound of 52.4%.

6. Conclusion

This project successfully created an NFL over/under predictive model with a validation accuracy of 53%. This was done through the processing of statistics available for free to the everyday NFL viewer.

This means two things. First, there is now a model that, with the successful input of the required statistics, should output a bet with a success rate to make money in the long run. There is a caveat - the user will need to preprocess the data in a way (i.e., four-game rolling average) to successfully input into the model. But, this is not much additional work and only requires a few lines of added code.

Second, in the bigger picture, it means that it is possible for NFL viewers with coding knowledge to generate models that make money in the long run while sports betting. It opens the door to creating models that give everyday people an edge in the unfair world of sports betting.

7. Future work

There are a few directions for future work with this project. The first would be expanding the scope of the bets used. There are an assortment of bets that can be placed on games.

Another one that is very popular is the spread, which presents a spread of points that one team must win by, and bettors can bet whether or not a team will win by this much. For example, in Super Bowl LIX, the spread was Kansas City to win by 1.5 points. Philadelphia ended up winning by 18, so bettors betting PHI +1.5 would have won the bet. It would not require much more work to add this bet into the analysis.

Another bet is Moneyline. A bettor bets which team will win the game, but the amount that a person can win, or the moneyline, can change depending on what the sports betting firms think are the favorites. Weighting based on this moneyline would have to be factored into the analysis.

Finally, as discussed earlier, the project was limited to the usage of play-by-play data from 2010 onwards. The full data of the dataset could be incorporated into the analysis to see if this improves or decreases the effectiveness of the models.

References:

The references used for this project were either background for the introduction and the dataset itself. Much of the football knowledge was already known by Alex Mandanis.

ChatGPT was also used to help design code, with the permission of the professor.

[1] J. Rich, "*Legalizing Sports Betting is the Right Call.*" Reason Foundation. www.reason.org/commentary/legalizing-sports-betting-is-the-right-call/#:~:text=In%202018%2C%20the%20Supreme%20Court,30%20states%20and%20Washington%2C%20DC

[2] F. Richter, "*America's Sports Betting Boom.*" Statista. www.statista.com/chart/29332/gross-gaming-revenue-from-sports-betting-in-the-us/#:~:text=According%20to%20the%20American%20Gaming,record%20set%20the%20year%20before

[3] A. Ostrander, guga31bb, mrcaseb, and T. Ho, 2025, "pbp," nflverse. [Online]. github.com/nflverse/nflverse-data/releases/tag/pbp

Signature Page:

Ilan Sela:

Alex Mandanis: