Predicting the Over/Under Result of NFL Games

Vincent Goyette, Blaise von Ohlen, Tyler Horwitz, Marcelo Castellanos
University of Notre Dame

ABSTRACT

This is a proposal for a project in Data Science - CSE 40647 at the University of Notre Dame. The goal of this project is to produce a methodology that will successfully predict the over/under of a given game in the National Football League (NFL).

ACM Reference Format:
Vincent Goyette, Blaise von Ohlen, Tyler Horwitz. 2021. Predicting the Over/Under Result of NFL games. In *Proceedings of the ACM Conference (Conference '21)*. ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/1234567890

1 PROBLEM DEFINITION

The National Football League is one of the largest and most successful sporting organizations in the world. As more states around the U.S. legalize sports gambling, the amount of money placed on NFL games in betting markets is bound to increase. One of the easiest and most popular ways to bet on a game is to bet on the over/under of a game, a prediction by bookmakers of what the total score of a particular game will be.

The goal of this project will be to produce a machine learning model that is able to successfully predict the over/under result of a particular NFL game with high accuracy. Features in our data will include each team's rolling average stats coming into our particular game; labels for each game will either be "over" or "under". Our plan is to develop a model based on existing NFL games that have been played so that we can accurately predict whether a given NFL game will hit the over or under.

2 RELATED WORK

The problem of predicting the outcome of NFL games is not a new concept in the field. In fact, the Over/Under predictions that are set by sportsbooks and casinos are themselves set by machine learning models (Silverio). Different models have surfaced over the years using different information in the attempt to get the best results. One of the first surfaced in 1996. Created by M.C. Purucker, the model used a neural network to achieve an accuracy of 61% in predicting NFL results. This model was then expanded by Khan in 2003 to reach an accuracy of 75%. Skipping ahead, in 2015 Tax and Joustra used data from past Dutch Football competitions to predict future ones, and the interesting part is that they used the betting odds as variables as well to achieve an accuracy of 54.7%. It is clear that there are many ways to approach the kind of problem we chose and that the best model can be found through a large amount of experimentation.

978-1-4503-0000-0/18/06...$15.00
https://doi.org/10.1145/1234567890

3 PROBLEM DEFINITION

Given a certain matchup between two NFL teams, will the total score be over or under the set over/under line? How does our model compare with the results of real NFL games?

4 METHODOLOGY

4.1 Data Collection

The first step in creating our machine learning model was finding a good dataset, and then ensuring that the dataset had good feature values. We started out with a dataset called spreadspoke_scores.csv, which contained every NFL game since 1966, along with team, score, weather, and stadium information. To augment this dataset, we scraped pro-football-reference.com to find the team statistics for each game (e.g. total yards, turnovers, passing yards, etc.).

4.2 Data Preprocessing

The new dataset that included statistics for each game gave a good place to start with data preprocessing. One of the first things we did to preprocess the data was to find all of the games with missing over/under results in the dataset. The over/under line was often unavailable for games prior to 1980, with the exception of each season's Super Bowl. As a result, our data was reduced from about 12,000 total games to 10,000 NFL games. Luckily our data had good

class balance, as there was a relatively even split between games that hit the over and games that hit the under.

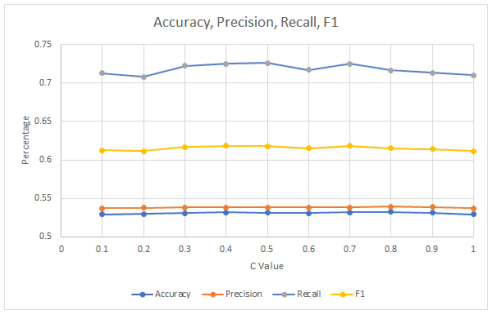
The next data preprocessing technique we implemented was feature extraction. One example was the calculation of each team's rolling average stats. The team stats associated with each game were their stats for that particular game, which is useless for predicting a future game since we obviously don't know their stats for a game ahead of time. As a result, we created datasets with the stats for each particular team, and associated their game stats with a game ID. This allowed us to easily take their rolling average stats, and then reassociate these average stats with a game in the spreadspoke_scores dataset. This means that each team's stats for a particular game would be the average of their *previous* 16 games, rather than previous season averages or their stats for that game. We figured that this would give a better indication of each team's recent form and thus would give our model better performance.

5 MODELING

At this stage, with our data fully processed, we were ready to continue on to modeling the data. For each of our models, the features were home yards per game, home turnovers per game, away yards per game, and away turnovers per game. For each model, the data was split such that 70% of the games in the dataset were used for training, and 30% was used for testing. This would give us about 7,000 training instances, and about 3,000 training instances. Each of the following models was implemented using the sklearn Python library. The results of each model will be discussed further in the evaluation section.

The first model we created was a logistic regression. This model was chosen because of its relative simplicity and relative effectiveness in predicting binary outcomes. We left the default sklearn configuration for this model to give us a simple baseline, but there is scope for us to further experiment with this particular type of model.

The second model created was an SVM. We experimented with this model using both linear and polynomial kernels, and we also experimented with different C values ranging from 0.1 to 1. Our best results came from a polynomial kernel with a C value of 0.4. The following table presents the metrics attained by each C value for the linear kernel.



We also implemented a Decision Tree Classifier for over/under prediction. The performance from this model wasn't that great, garnering 52% accuracy, which is ~1% worse than our other models. It is abundantly clear that more data (features) are required to create an optimal algorithm, although this comes at the price of potentially overfitting. We chose a max depth of 4 to prevent an overly-complex model while also ensuring each feature was considered. The library does not include an option to actually put the labels on the leaf nodes, but rather the value array which contains the class labels predicted for each testing data object. The figure is too large to insert in place right here, so it has been attached to the bottom section.

6 EVALUATION

To verify our model's performance, we ran each one 10 times. We then took the mean of each model's metrics to produce the following results.

Model	Accuracy	Precision	Recall	F1
SVM (linear kernel)	.53	.54	.73	.62
SVM (polynomial kernel)	.53	.53	.84	.65
Logistic Reg.	.53	.54	.67	.60
Decision Tree	.52	.54	.55	.55

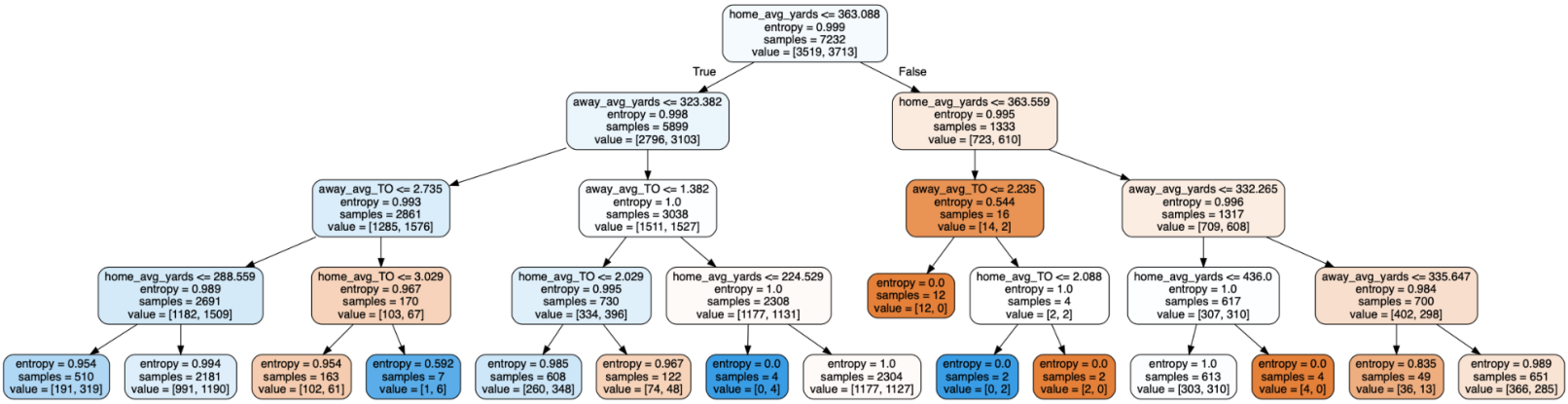
If we take the accuracy of a random guess to be 0.5, we can see that each of our models outperformed a random guess, but just barely. Each of our models had an accuracy of about 53%, representing a small 3% increase in accuracy. One interesting note is that the recall, in each case, was higher than all other metrics. This indicates that we generally predicted the *over* at a better rate than we did the *under*. Initial analysis failed to spotlight any obvious reasons for this, but it is something that we'll have to keep in mind as we move forward. Each of the models

performed similarly in other metrics; there were no obvious reasons for this similarity, but it is another piece of data that will further inform our future models.

7 TIMELINE

Oct. 25 - Different feature exploration w/ existing models
Nov 2 - Decision on features
Nov. 15 - Implementation of Models with New Features
Nov. 20 - New Model Implementation (Random Forest)
Nov. 27 - Comparison of the new models/iteration on them
Dec. 3 - Begin writing report / presentation
Dec. 10 - FINAL REPORT DUE

8 DECISION TREE



9 REFERENCES

[1] Silverio, Manuel. “My Findings on Using Machine Learning for Sports Betting: Do Bookmakers Always Win?” *Medium*, Towards Data Science, 26 July 2021, https://towardsdatascience.com/my-findings-on-using-machine-learning-for-sports-betting-do-bookmakers-always-win-6bc8684baa8c.