APPLICATION OF MACHINE LEARNING TO GAME PREDICTION,

Andrew, Jiang, Allen Sha, Janum Shah, and Devon Ulrich

NFL GAME PREDICTION OVERVIEW

The National Football League has become widely popular across the USA in terms of household entertainment, as well as economic interests of home team cities. The NFL consists of 32 teams which are divided equally between the American Football Conference and the National Football Conference. They play first in the regular season, which lasts for 17 weeks (early September to late December), followed by the playoff season, which continues until early February. Each team plays 16 games in the 17-week season, resulting in a total of 256 games in a regular season. The fast-paced season is a thrill for many football fans and even a hobby for those who analyze tirelessly to predict winning teams, demonstrating football's status as an extremely popular sport across the nation. Moreover, sports betting has become a popular practice around the country.

The objective of this project was to predict the outcomes of NFL games based on data from previous games. The results of each game were collected and analyzed in terms of point spread, which is a way to predict the outcome of a game that decides which team should theoretically win. The point spread is the expected margin of victory (in percentage) based on the two teams' relative abilities. Point spread is widely used in sports betting and gambling, and the point spread for a game is designed to make it equally likely for either team to perform better than the other in relation to expected performance (4). Accurately predicting football team performance is incredibly difficult; in fact, very few professional analysts can correctly predict games over 50% of the time (5). The goal of this project was to optimize the data category combinations to predict NFL team performance at least as well as professional analysts can. In order to do so, data consisting of 10 categories (such as yards per play, passing yards, and points per game) were obtained for 900 games over multiple years. There were four subsets per category: Home Team Offense (HTO), Home Team Defense (HTD), Away Team Offense (ATO), and Away Team Defense (ATD).

NFL GAME PREDICTION DATA

All of the NFL team data were obtained from *TeamRankings* (6). The dataset was split into two groups: a set with 779 games from the 2014, 2015, and 2016 seasons that was used for training, and another set with 121 games from the second half of the 2017 season that was used for testing. The training data was then split again to have data from the 1st through 8th weeks of each season and then the 9th through 16th weeks. A total of 40 categories of data were obtained for each game, 10 for each of the four subsets mentioned above. The results from each game were obtained as well and stored in binary format: a game's result was recorded as 1 if the favored team performed as well or better than the point spread predicted, and 0 if the favored team performed worse than predicted.

The data were then reformatted and adjusted to improve the accuracy of the machine learning algorithms. As shown in Figure 2, there were three main steps taken to format and

process the data: feature scaling (normalization), feature sorting, and balancing. Feature scaling, specifically min-max feature scaling, was used to adjust the range of the data of each feature to fit in the range [0, 1]. Certain machine learning algorithms such as logistic regression and SVM require input values to be normalized, so this scaling was necessary to ensure that these algorithms could be used on the data (7). The scaled features then had to be trimmed down to a smaller set of features, since many of the features had little correlation to the game results and acted as noise that could have lowered the algorithms' accuracies. To remove these extraneous features, a Least-Squares Regression Line was created between each feature and the point spread results, and the coefficient of correlation (r^2) was calculated for each line. Since r^2 values help explain how strongly correlated two variables are, all of the features were sorted from greatest r^2 to least in order to facilitate feature selection later in the training process. Figure 3 contains two scatterplots with regression lines: each shows the relationship between one specific feature and the binary results column. The plot on the left in Figure 3 shows a much more prominent association between the feature and results than the plot on the right does, which explains their relative r^2 values. After all 40 features were sorted based on their r^2 values, the nine features with the greatest r^2 values were chosen as the final features. All of the training data were then balanced: there were originally 411 games labeled as "1" as opposed to 368 labeled as "0" in the training dataset, so each match was duplicated a set number of times to balance out the number of 1's and 0's in the training data. This ensured that the algorithm was not biased towards a particular result, and considered 1's and 0's equally while being trained (8).

	ATD yards/play	ATD yards/game	...	Result
1	4.1	261.1		0
2	4.8	272.4		1
3	4.3	272.5		1
4	4.3	277.3		0

1. Original Data

	ATD yards/play	ATD yards/game	...	Result
1	0.0000	0.0000		0
2	0.2546	0.0663		1
3	0.0727	0.0669		1
4	0.0727	0.0950		0

2. Scaled Data

	ATD yards/game	ATD yards/play	...	Result
1	0.0000	0.0000		0
2	0.0663	0.2546		1
3	0.0669	0.0727		1
4	0.0950	0.0727		0

3. Sorted Data

	ATD yards/game	ATD yards/play	...	Result
1	0.0000	0.0000		0
1	0.0000	0.0000		0
...				
2	0.0663	0.2546		1

4. Balanced Data

Figure 2: The steps taken to preprocess the data. Table #1 is adapted from the original dataset. Table #2 shows the same data, but after it was scaled to the range [0, 1]. Table #3 shows the table after the two features were sorted based on their correlation with the "result" column, specifically where ATD yards / game switched places with ATD yards / play. Finally, Table #4 shows some of the table after it was balanced, since each row was duplicated a certain number of times to equalize the number of 0s and 1s in the "result" column.

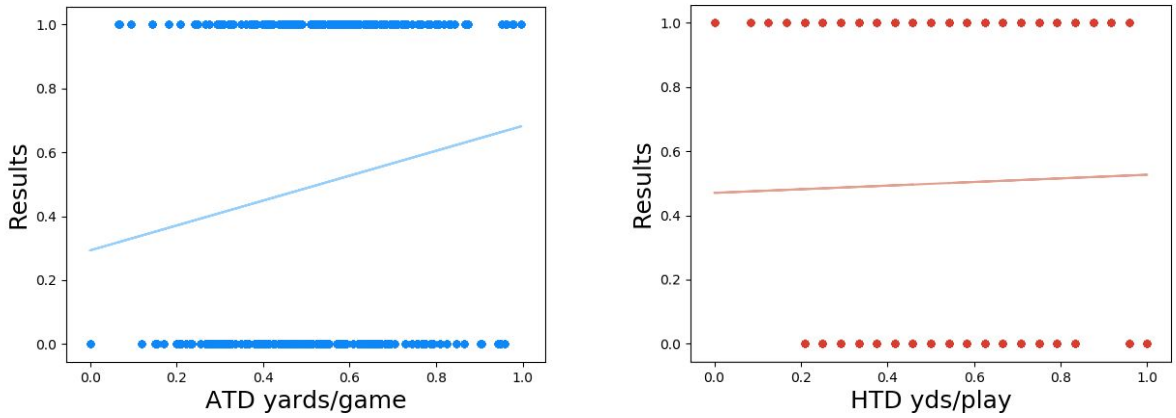

Figure 3: Example plots for the correlation calculation in Step 3 of Data Preprocessing. This figure visualizes the linear regression lines that were generated for two of the 40 features while sorting the features by correlation. $r^2_{ATD} = 0.5966$, $r^2_{HTD} = 0.0479$

NFL GAME PREDICTION METHODS

Logistic Regression

The classification algorithm was a logistic regression, described by

$$\pi(x) = \frac{1}{1 + \exp(w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n)} \quad (1)$$

where x is the input vector, w_i are the logistic regression weights for $1 \leq i \leq n$, and w_0 is the bias parameter. The complement of the probability was used, $1 - \pi(x)$. If the probability is greater than 0.5, then it is evaluated to be a 1; if it is less, then a 0.

Support Vector Machine (SVM)

Support Vector Machine, or SVM, algorithms have become widely known in the machine learning community as one of the most convenient algorithms to implement (9). In this project, a SVM algorithm was used for binary classification. Each game, represented as a point in high dimensional (initially 40, then 9 after normalization and feature selection) space, was assigned a label based on the result (favored team covering or not covering the spread). The SVM algorithm attempted to draw a hyperplane that split the data such that all the 1-labeled points were grouped together and all the 0-labeled were grouped together. The hyperplane always has one fewer dimension than the data it is trying to split. In this case, the games were represented by 9 vectors, and the hyperplane was an 8-dimensional space. If it is impossible to draw a hyperplane that splits the data perfectly, SVM generates a hyperplane that splits the data as best as possible. The predicted label generated by SVM is based on which side of the hyperplane the point is on.

While both logistic regression and the SVM algorithm were used for predicting the 2017 test data, the SVM algorithm was chosen because it was slightly more accurate both in the training accuracy and the cross-validated accuracy. Additionally, training the algorithm on the

9th through 16th weeks was determined to be the most effective because the stats of NFL teams are extremely volatile during the first half of the season and don't tend to reflect their true capabilities.

The varying data were then normalized and analyzed based on regression and correlation factors to feed into the algorithm for the final training and testing datasets. The data were fed into various algorithms that finally predicted the expected performance of each matchup.

NFL GAME PREDICTION RESULTS

The model was tested on 121 games in weeks 9−16 of the 2017 NFL regular season. Of these 121 games, 68 were correctly predicted (Fig. 3).

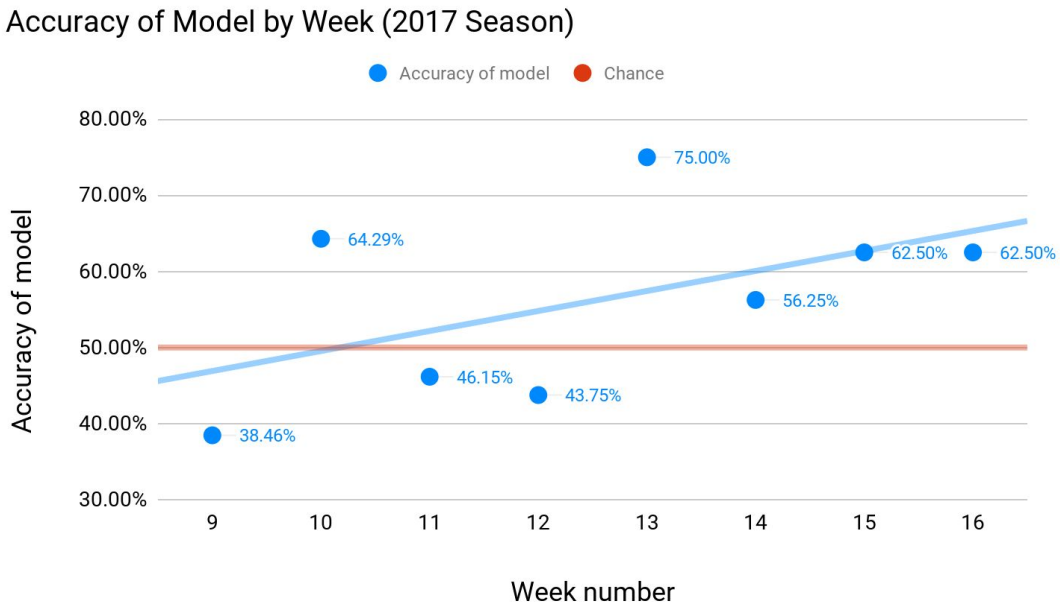

Figure 3: Percentage of games predicted correctly each week. The blue line represents the overall accuracy of the model. The red line represents the theoretical accuracy of random chance.

Figure 3 represents the percentage of games accurately predicted by the model for each week in 2017. The model was used to predict all games in Weeks 9-16. Weeks 1-8 from 2017 were included in the training data; although we have plenty of data from previous seasons, data from the same season were especially helpful because they offset any major changes in coaching or player trades that may have occurred during the offseason. These changes would likely not have drastically changed the ability of individual teams in each season, but the data from earlier in the same season were still useful in increasing the accuracy of the model.

Overall, there is an increasing trend in accuracy as the weeks progress. This may be because the model retrained on previous weeks' data which included the correct outcome of the previously predicted games. For example, after Week 10 and before Week 11, the model then had data from Weeks 9 and 10 before predicting the Week 11 games. The performance of the

model based on solely 2014-2016 data was slightly less accurate than the model that included data from the current season. As previously mentioned, the SVM model predicted the outcome of games with 56.2% accuracy overall. Although this may seems low at first, after considering that most top analysts fail to predict with even 50% accuracy (theoretically equivalent to guessing), 56.2% is a respectable percentage. Also indicative of the model's applicability to all situations is the fact that for the training data (2014-2016) the model tended to overpredict 1's and was also able to predict 1's with greater accuracy. However, when tested on the 2017 data, which had a higher percentage of 0's than 1's, the model adjusted accordingly and predicted more 0's than 1's, a stark reversal from before, but one that was intended to happen.

NFL GAME PREDICTION DISCUSSION AND CONCLUSION

The group limited the parameters used to the nine features that had the greatest correlation with the outcome. These features were: HTD rushing yards per game, HTO penalties per game, HTD rushing yards per attempt, ATO turnovers per game, HTD turnovers per game, HTO points per page, HTO yards per play, ATO yards per pass attempt, and ATD turnovers per game. Conversely, features that did not show significant correlation were not included in training the model because they were not very relevant to the outcome of the game. Some less relevant features include: ATD penalties per game, HTO rushing yards per game, ATD pass yards per game, and many more. If they had been included, the model predictions would possibly have been skewed because the algorithm may have attempted to create artificial correlations between irrelevant data and the outcome of their respective games.

To conclude, the NFL game prediction program was tested with the dataset from the 2017 NFL season and was able to predict the correct outcome with an accuracy of 56.2%. Top analysts are barely able to predict NFL performance expectations at above 50% (5). The computer has the ability to compile the statistics of the features that are most influential towards the outcome. For example, home team offense yards per play was found to be highly correlated with the outcome of various games, so the computer weighted this feature the most. Humans lack the ability to detect these minute and specific patterns and to weigh all of the different features simultaneously. Additionally, humans can have inaccurate biases favoring certain teams, players, coaches, or plays when predicting the outcome (10).

However, if a change occurs within a team, the computer will have to wait multiple games for the change to be observed in the statistics. Humans, on the other hand, are able to realize and sympathize with the detrimental effect of a star player injury, and quickly adjust their prediction of the team's future performances. This shows both the advantages and disadvantages of the algorithms of the program relative to human capability. Other underlying qualitative features that the program would not be able to compile include: the vitality of win or loss streaks, emotions of the team, and fan based support. Top analysts may not weigh the statistics like the program does but instead weigh in certain assets on teams such as a star rookie quarterback (10). As such, humans are better equipped to accommodate for sudden changes and holistic features, while a computer has a better overall prediction due to its ability to learn from a complex dataset.

Of course, there is always room for improvement. The algorithm uses data from previous seasons to predict the outcome of games for the next seasons and weeks. It does not take into

account trades or coaching changes, which often change the ability of teams as a whole. Training the model with data from individual players could not only increase the accuracy of predicted games, but could also offer insight about players, trades, and coaching staff. For example, a model trained with individual players' data could potentially suggest beneficial trades for owners or find fair contract values for players. Other features such as injuries during the week would help improve the program's accuracy because if a certain team begins to perform poorly suddenly as injuries increase, an improved algorithm would recognize this correlation and adjust its predictions.

REFERENCES

1. Kurama V. Introduction to Machine Learning. Towards Data Science. 2017 Jul 15 [accessed 2018 Jul 21]. https://towardsdatascience.com/introduction-to-machine-learning-db7c668822c4
2. Grus J. Data Science from Scratch. 1st ed. Beaugureau M, editor. Cambridge: O'Reilly Media, Inc.; 2015.
3. What is Python? Python. [accessed 2018 Jul 21]. https://www.python.org/doc/essays/blurb/
4. Moody A. Understanding How Point Spreads Work in Sports Betting. ThoughtCo. 2018 Mar 29 [accessed 2018 Jul 21]. https://www.thoughtco.com/sports-betting-understanding-point-spreads-3116853
5. Berkman F. Analysis of NFL Analysts: Which 'Experts' Get It Right? [Internet]. Mashable. 2012 Nov 18 [accessed 2018 July 20]. https://mashable.com/2012/11/18/nfl-analysts/#.Y_noaU0DqqO
6. TeamRankings. TeamRankings. [accessed 2018 Jul 21]. https://www.teamrankings.com/
7. Importance of Feature Scaling. Scikit Learn. [accessed 2018 Jul 22]. http://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html
8. Batista G, Prati R, Monard M. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explorations*. 2004.
9. Vishwanathan SVN, Murty MN. SSVM: A Simple SVM Approach. Proceedings of the 2002 International Joint Conference. 2002. https://ieeexplore.ieee.org/document/1007516/
10. McNerney S. Cognitive Biases in Sports: The Irrationality of Coaches, Commentators and Fans. *Scientific American*. 2011 Sep 22. Cognitive Biases in Sports: The Irrationality of Coaches, Commentators and Fans