Predicting Outcomes of NFL Games

The Wiggles [cli94, ngarg11, pdam, jmanuel2]

Goal

Sports betting is now a major industry in the U.S. Consequently, we are interested in developing a model which can aid in making bets on pro-football NFL games. The use case for this project is to help bettors make informed bets on the winners of NFL games. Thus, the objective we are most interested in is as follows: given historical regular season game data, predict the winner of NFL games for a full season (~240 games).

Data

Data was sourced from SportRadar API, an officially-partnered distributor of NFL-related statistics. They provided over 500 unique features for each game. We used games from the 2012 to 2019 seasons and dropped features such as those with null values (indicating the feature was not present in earlier seasons) and those that would allow the model to too easily predict winners, such as team IDs (NE, ATL, etc.), points, and touchdowns scored. This gave us 1912 games, each with 232 features. We extracted the points scored by the home team and away team for each game to create the labels: 1 indicating the home team won or tied, 0 otherwise. Our test set consisted of half of the 2019 season (about 120 games), and the other half was set aside for validation. The 2012-2018 seasons were used to train our models. This ensures no future games were used to predict current games.

Model and Evaluation Setup

We decided to implement three models: a feed forward neural network, a logistic regression model, and an SVM model.

Our feed forward network utilized two dense layers, with relu activation for the first layer and the sigmoid activation for the second. Our logistic regression model utilizes l2 normalization. Our models were primarily evaluated in two ways. First, we checked if our models could successfully predict the outcomes of games which already happened using statistics from those games. Then we moved on to the main prediction objective which was to predict the results of games using statistics from past games. In our case, we used 2019 as our test set, and trained our models on the average statistics from the past five games, the average statistics resulting from averaging statistics for both the home and away team in a game.

Results and Analysis

Claim #1: Our classifier trained using all features outperforms a baseline model without training by a significant margin, and is similar to other models.

Support for Claim #1: Table 1 shows the accuracy of our models compared to baseline models for a full season of football games.[1]

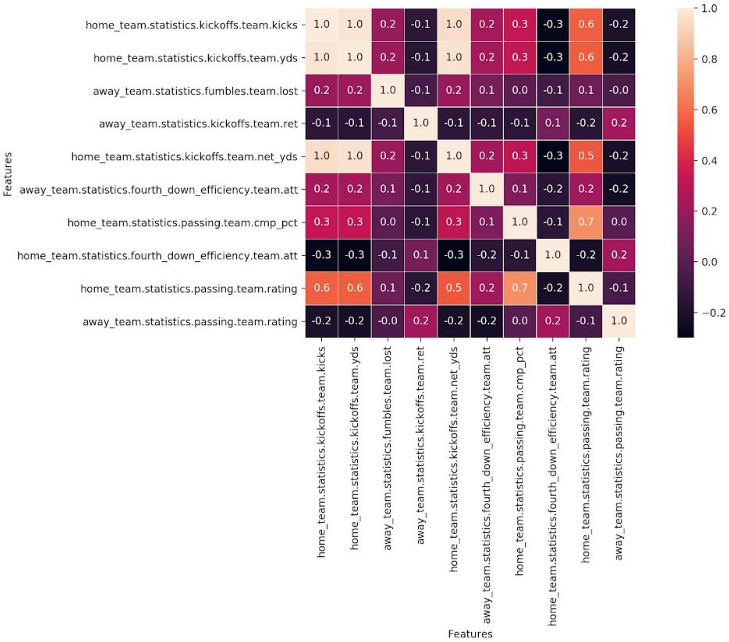
Model	Testing accuracy on predicting who wins game
Our Logistic Regression Model	63%
Our Logistic Regression Model**	63%
Our Feed Forward Network	60%
Our Feed Forward Network**	63%
Our SVM Model	52%
FiveThirtyEight's ELO Model*	63%
Microsoft's Cortana*	67%
Baseline: Without Training	50%

* Over the first 15 weeks of the 2017 NFL regular season.

** With feature selection

Claim #2: About 136 of the 232 features are sufficient for our models to get a higher-than-guessing accuracy.

Support for Claim #2: After applying recursive feature elimination to our model, we found that about 136 features were enough to get a comparable accuracy when using all 232 features. Please refer to the above table for these results. The heatmap below shows the correlation between the 10 features with the highest coefficient values.



[1] https://www.businessinsider.com/nfl-picks-microsoft-cortana-elo-week-16-2017-12

When these features were input as training data for the model, the model performed significantly worse than if it had all the features, achieving about a 52% accuracy on the test set. Initially, we thought that this poor accuracy could be attributed to pairs of features being highly correlated to each other, thus hampering the results of our logistic regression. However, even after dropping a feature from each pair which had correlations of 0.7 or higher, the accuracy failed to improve from the 52% mark. The accuracy also remained the same when features were selected with p-values less than 0.05.

Claim #3: Our models are learning a bias in favor of home team victories.
Support for Claim #3:

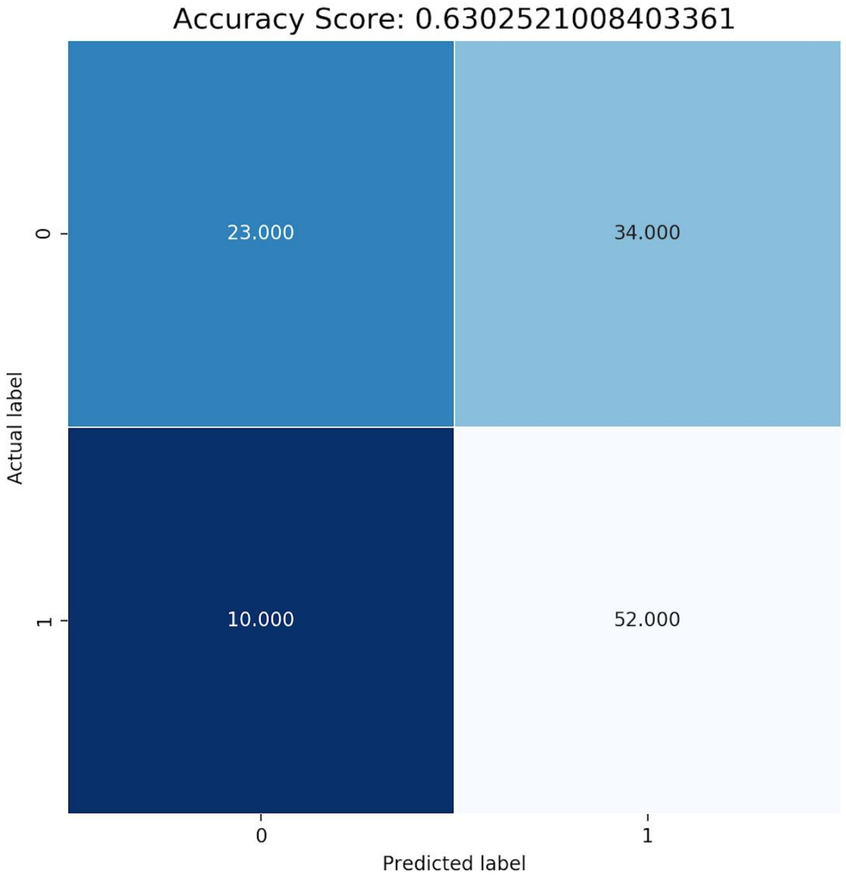
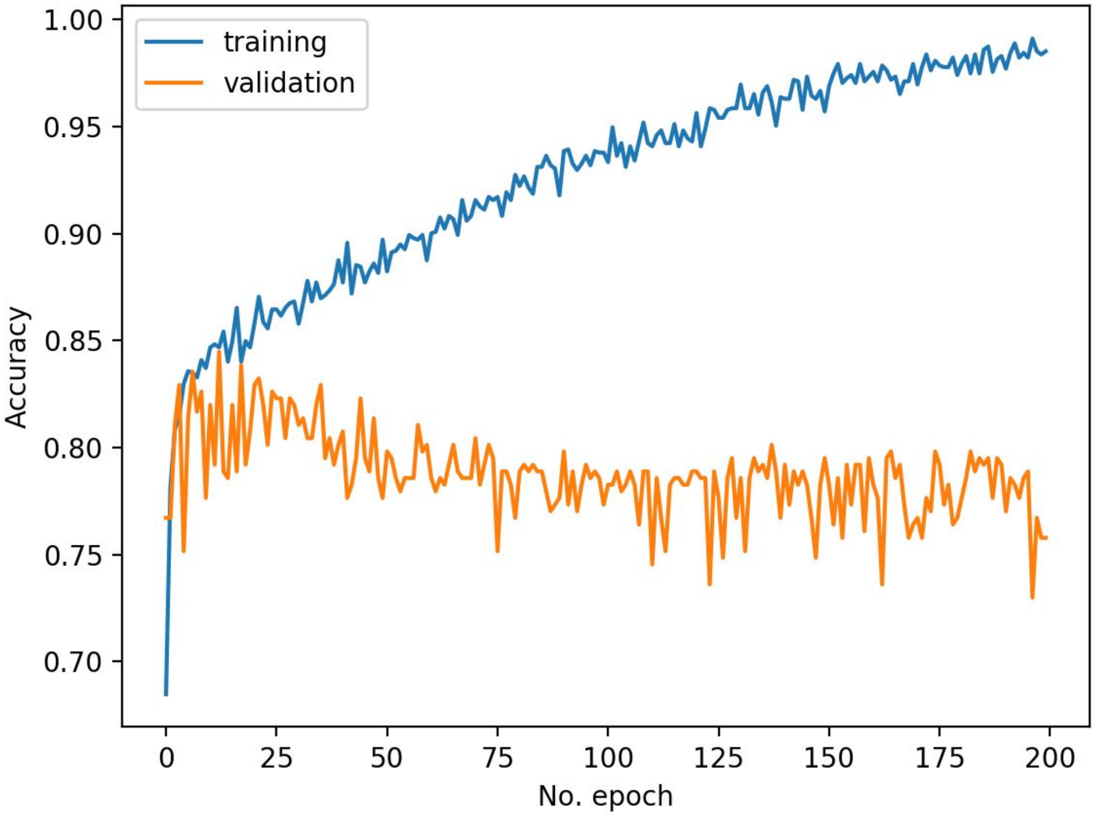


Figure 2: Confusion matrix for predicting the correct winner for the logistic regression model

The confusion matrix was produced by using the testing set, which consists of half of the 2019 season, and predictions from our trained logistic regression model after passing into it the averaged statistics from the past five games. As shown in the confusion matrix, in a significant number of tests where the model failed, it was in the case where the actual label was 0 (the away team won) but the predicted label was 1 (the home team won). A possible reason why the model is failing in this case is because it may have learned to overestimate home field advantage and thus in circumstances where the statistics of the home and away teams were fairly close it would defer to the home team. A possible resolution to this issue would be to get more data that quantifies how each individual team does at home versus away.

Claim #4: Our feed forward neural network is affected by overfitting.
Support for Claim #4: Graph 1 shows the training and validation accuracies of the network over 200 epochs.



Overfitting is illustrated through the decline of the validation accuracy after the first ~25 epochs. We attempted to address the overfitting bias through the following means: including a dropout layer between the dense layers, shuffling the training input, and reducing the number of epochs. Each measure implemented independently yielded approximately the same accuracies as the original model. Thus, we could not conclude that these preventive measures are effective in reducing overfitting in our model. We suspect that a lack of training data is the main contributor to overfitting, and this can be explored in future work.