Stats 101C Project Report

In Yong Lee, Matt Matsuo, Yixiang Zhang

12/8/2019

Data Exploration

The data for this project comes from the realm of basketball. Our main goal was to predict the winner in each individual game given the over two hundred variables. The data ranged from team stats such as points per game to a player's plus minus. In order to more fully take advantage of the data, we decided to create two variables: days since last game and shooting percentage. We theorized that if one of the two teams had more rest than their opponent, they would be healthier, more prepared, and would therefore have a better chance to win the game. In order to do this, we found parsed through the Home Team and Visiting Team columns to find each team's most recent game and found the number of days from that game to the team's current game. In our training models, this variable was significant, however, it was not very effective when predicting game outcomes on the test data and was excluded from our final model. Additionally, we converted the number of shots and attempts into percentages. This is because the number of actual attempts and makes can be misleading if not standardized.

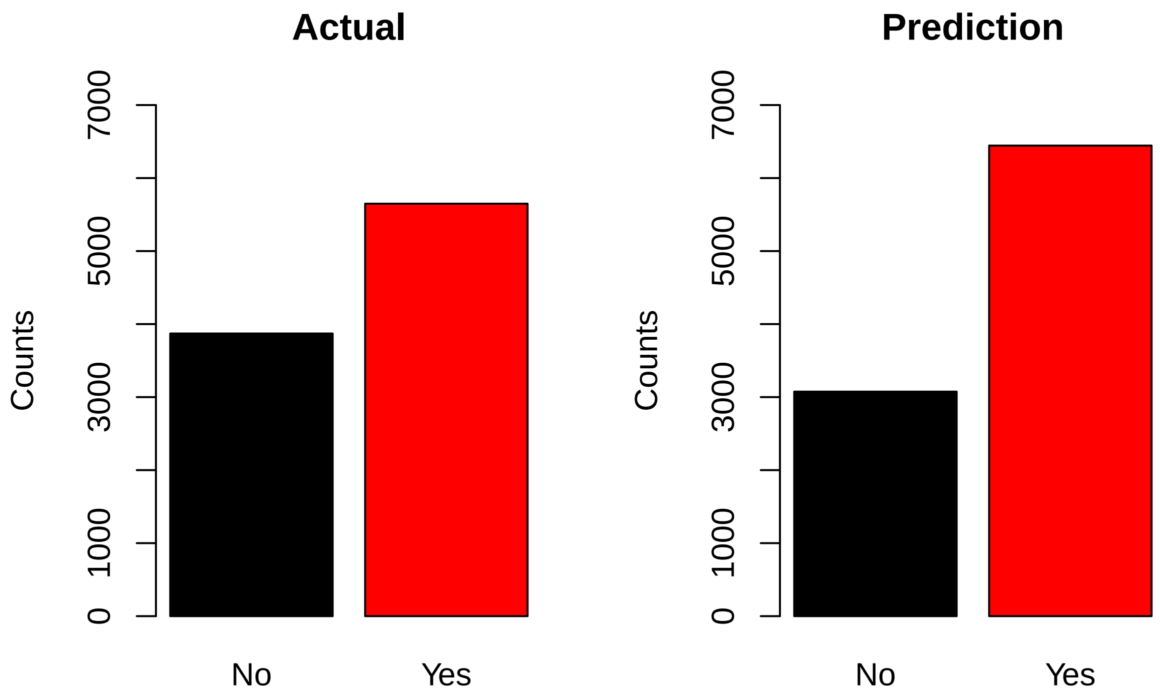
Model Description

We obtained our model by first running a backward stepwise selection from the full model using AIC as the cross-validated prediction error. (Note that the full model only consists of variables from the visiting team (VT) because variables from VT are identical with variables from HT. For example, HT.TA = VT.OTA and HT.TS = VT.OTS). Then, we removed variables that had a VIF greater than or equal to 5 in order to solve the issue of multicollinearity. From there, we decided to perform logistic regression by using generalized linear model with the remaining variables in order to classify whether the home team won each game. The final model consists of 47 predictors.

$$p(HTWins) = \frac{e^{\begin{array}{l} VT.TS.fgm+VT.TS.tpa+VT.TS.fta+VT.TS.oreb+VT.TS.stl+VT.TS.pts+VT.TA.tpm \\ +VT.TA.fta+VT.TA.oreb+VT.TA.dreb+VT.TA.ast+VT.TA.pts+VT.OTS.fgm \\ +VT.OTS.fga+VT.OTS.tpa+VT.OTS.stl+VT.OTS.pts+VT.OTA.tpa+VT.OTA.oreb \\ +VT.OTA.ast+VT.OTA.blk+VT.S1.plmin+VT.S1.pts+VT.S1.ast+VT.S2.pts+VT.S2.ast \\ +VT.S3.pts+VT.S3.min+VT.S3.stl+VT.S3.ast+VT.S4.pts+VT.S4.min+VT.S4.ast \\ +VT.S5.pts+VT.S5.min+VT.S5.stl+VT.OS1.plmin+VT.OS1.dreb+VT.OS2.plmin+VT.OS2.dreb \\ +VT.OS2.fgm+VT.OS3.plmin+VT.OS3.dreb+VT.OS3.to+VT.OS4.dreb+VT.OS5.to+VT.OS5.oreb0 \end{array}}}{1+e^{\begin{array}{l} VT.TS.fgm+VT.TS.tpa+VT.TS.fta+VT.TS.oreb+VT.TS.stl+VT.TS.pts+VT.TA.tpm \\ +VT.TA.fta+VT.TA.oreb+VT.TA.dreb+VT.TA.ast+VT.TA.pts+VT.OTS.fgm \\ +VT.OTS.fga+VT.OTS.tpa+VT.OTS.stl+VT.OTS.pts+VT.OTA.tpa+VT.OTA.oreb \\ +VT.OTA.ast+VT.OTA.blk+VT.S1.plmin+VT.S1.pts+VT.S1.ast+VT.S2.pts+VT.S2.ast \\ +VT.S3.pts+VT.S3.min+VT.S3.stl+VT.S3.ast+VT.S4.pts+VT.S4.min+VT.S4.ast \\ +VT.S5.pts+VT.S5.min+VT.S5.stl+VT.OS1.plmin+VT.OS1.dreb+VT.OS2.plmin+VT.OS2.dreb \\ +VT.OS2.fgm+VT.OS3.plmin+VT.OS3.dreb+VT.OS3.to+VT.OS4.dreb+VT.OS5.to+VT.OS5.oreb0 \end{array}}}$$

Figure 1: Final Model

Classification Rate



```
##       Predicted
## Actual   No  Yes
##    No  1975 1896
##    Yes 1100 4549
```

```
## [1] 0.6852941
```

The table above shows the confusion matrix that compares the model predictions to the true HTWins value from the training dataset. The diagonal elements represent the values whose HTWins were correctly predicted whereas the off-diagonal elements represent vales that were misclassified. From the matrix, our accuracy on the training data was 68.52941%. On the testing data set, our classification rate was 67.111% on the public leaderboard and 67.475% on the private leaderboard (final test score).

Why the Model works well & Areas of Improvement

This model somewhat worked well in predicting the home team wins since the stepwise selection helped remove variables that lead to unnecessary complexity in the full model (i.e. improve model interpretability and prediction accuracy). In addition, by removing variables with high VIF's, we were able to remove predictors that can be linearly predicted from other predictors with a high degree of accuracy (i.e. solve multicollinearity). However, the model's classification rate was only around 67%, mainly because the extensive number of predictors (even after backward selection) caused the model to overfit the data. If we had more submissions, we were going to remove insignificant variables (i.e. p-value was greater than 0.05 in the summary table), which slightly increased the classification rate from the training data and may have improved the score in the testing data.