MIST 5635 Group Project Deliverable 4 (<2,000 words)

By: Ryan Cullen, Alec Zerona, Warren Paintsil, and Will Hensley

DATA TRANSFORMATION AND FEATURE ENGINEERING:

In any data analysis project, especially leveraging machine learning technologies, it is important to make sure that the data is structured to offer an interpretable answer to your research question. Given that the U.S. Men's Soccer team has performed at such a poor level, we must transform raw data to give us the desired key performance indicators that are indicative of team success.

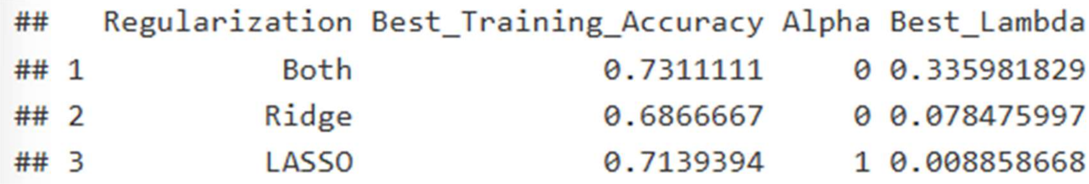
We focused on three feature categories: offensive opportunities, defensive opportunities, and tactical movement & passing dynamics. For offensive opportunities, we included metrics like shot accuracy, total shot attempts, crosses, corners, and free kicks, as well as a new feature, team1_win, which indicates whether Team 1 won the match or not. For defensive opportunities, we looked at features like turnovers, pressure, fouls, and defensive line break efficiency, and also created relative features by subtracting Team 1's values from Team 2's - another method of feature engineering. In the tactical movement & passing dynamics section, we explored features such as possession, passing strategies, and preferred field positioning, also calculating relative values between the two teams. Finally, we performed standard data cleaning such as removing any duplicates, null values, or infinite values to ensure our dataset was accurate and ready for analysis.

Overall, these methods of data transformation and feature engineering not only organize our methodologies into contextual aspects of soccer, but it sets the foundation for a great analysis project. We are significantly reducing the number of features by slowly turning the dataset from wide into long. This will allow us to narrow down to the true most important key performance indicators that are credited for team success in the FIFA World Cup.

LOGISTIC MODEL RESULTS:

For the first model that we fit - the logistic regression model - there are 3 different variations of regularization (Elastic Net, Ridge, and LASSO) fitted to 5-fold cross validation, where ultimately we optimally tune each model to show performance on training data. Once these options have been completely fine tuned, we will look at the coefficients that are more pronounced for each variation to explain what features are important in team success. Lastly, we will see how well each model generalizes to unseen data given that these models are being trained under partitions of the full dataset.

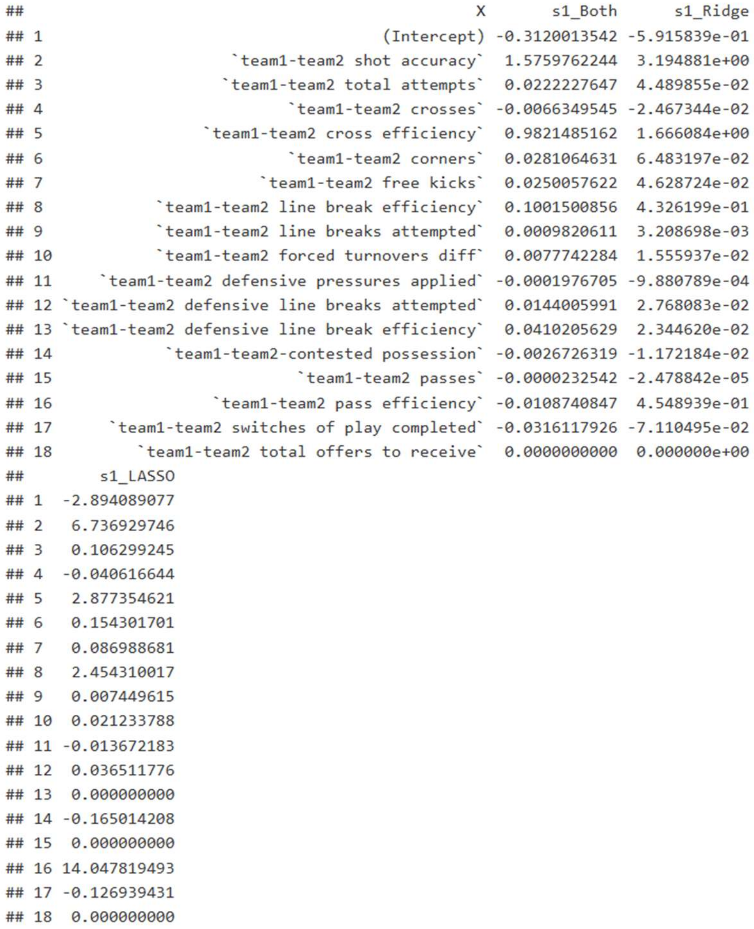
In fitting all 3 models, we wanted to display the best training accuracy given the optimal lambda tune.



```
##   Regularization Best_Training_Accuracy Alpha Best_Lambda
## 1           Both              0.7311111     0 0.335981829
## 2          Ridge              0.6866667     0 0.078475997
## 3          LASSO              0.7139394     1 0.008858668
```

Figure 1: Table of training data performance for 3 variations of regularized logistic regression models.

As seen in Figure 1, the Elastic Net regularized model performed the best on the training data with an accuracy of 0.73, indicating strong model fit. We will check later to see if this aligns with performance on the testing data.

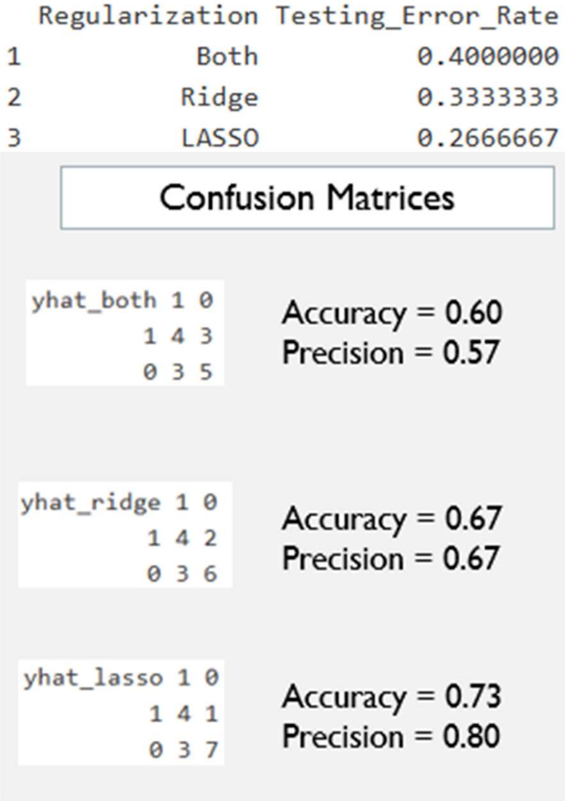


```
##                                             X     s1_Both     s1_Ridge
## 1                                 (Intercept) -0.3120013542 -5.915839e-01
## 2                `team1-team2 shot accuracy`  1.5759762244  3.194881e+00
## 3               `team1-team2 total attempts`  0.0222227647  4.489855e-02
## 4                      `team1-team2 crosses` -0.0066349545 -2.467344e-02
## 5             `team1-team2 cross efficiency`  0.9821485162  1.666084e+00
## 6                      `team1-team2 corners`  0.0281064631  6.483197e-02
## 7                   `team1-team2 free kicks`  0.0250057622  4.628724e-02
## 8        `team1-team2 line break efficiency`  0.1001500856  4.326199e-01
## 9        `team1-team2 line breaks attempted`  0.0009820611  3.208698e-03
## 10       `team1-team2 forced turnovers diff`  0.0077742284  1.555937e-02
## 11   `team1-team2 defensive pressures applied` -0.0001976705 -9.880789e-04
## 12 `team1-team2 defensive line breaks attempted`  0.0144005991  2.768083e-02
## 13 `team1-team2 defensive line break efficiency`  0.0410205629  2.344620e-02
## 14        `team1-team2-contested possession` -0.0026726319 -1.172184e-02
## 15                      `team1-team2 passes` -0.0000232542 -2.478842e-05
## 16              `team1-team2 pass efficiency` -0.0108740847  4.548939e-01
## 17    `team1-team2 switches of play completed` -0.0316117926 -7.110495e-02
## 18       `team1-team2 total offers to receive`  0.0000000000  0.000000e+00
##        s1_LASSO
## 1  -2.894089077
## 2   6.736929746
## 3   0.106299245
## 4  -0.040616644
## 5   2.877354621
## 6   0.154301701
## 7   0.086988681
## 8   2.454310017
## 9   0.007449615
## 10  0.021233788
## 11 -0.013672183
## 12  0.036511776
## 13  0.000000000
## 14 -0.165014208
## 15  0.000000000
## 16 14.047819493
## 17 -0.126939431
## 18  0.000000000
```

Figure 2: Feature coefficients of regularized regression models

Above here in Figure 2, we see the data frame for these fully optimized models given their best tune. There are 4 features that stand out the most from each model, some of which are in agreement with the others, others not so much. The most pronounced features based on their coefficients are relative team shot accuracy, relative team cross efficiency, relative team line break efficiency, and relative team pass efficiency. Given that these regularized models will promote important features and silence less-important ones, we can assume that these model variations are suggesting these 4 features as the most important in explaining team success at the FIFA World Cup.

For more model validation context, we want to see how each model will generalize to unseen data, or in other words, perform on the testing data.



```
  Regularization Testing_Error_Rate
1           Both          0.4000000
2          Ridge          0.3333333
3          LASSO          0.2666667
```

Confusion Matrices

```
yhat_both 1 0
        1 4 3
        0 3 5
```

Accuracy = 0.60
Precision = 0.57

```
yhat_ridge 1 0
         1 4 2
         0 3 6
```

Accuracy = 0.67
Precision = 0.67

```
yhat_lasso 1 0
         1 4 1
         0 3 7
```

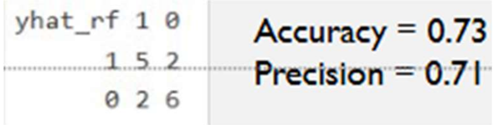
Accuracy = 0.73
Precision = 0.80

Figure 3: Testing data performance results for logistic regression model

We see that although the elastic net model performed the best on the training data, it actually performed the worst on the testing data with the LASSO model outperformed the others. The confusion matrices also support this claim with the LASSO model having the best accuracy and precision compared to the others. It appears that there is a need for distinguishing important features from the non-important ones with this dataset. A Ridge model would preserve all feature coefficients while only changing their degree of influence, where the LASSO model prioritizes and emphasizes only the important features, therefore making it the more reliable model.

RANDOM FOREST MODEL RESULTS:

Random forests incorporate extra randomness during the process of growing the trees. Random forest should improve the decision tree by decorrelating the numerous trees to reduce the variance when we average the trees. Bootstrap sampling (with replacement) drew new datasets, helping with the dataset’s downside pertaining to there being a small number of observations. We also focus on variable importance to help explain the model, displaying what features contributed and did not contribute to accurate predictions. The downside is trading interpretability with the random forest black box for accuracy, but the feature importance will still provide some insights into explaining significant contributing features to the model.



```
yhat_rf 1 0
      1 5 2
      0 2 6
```

Accuracy = 0.73
Precision = 0.71

Figure 4: Testing data performance results for the random forest model

The training accuracy for the random forest is about 0.63, relatively low compared to the best tunes of the logistic regression and boosted models. However, the positives of the random forest model include the testing accuracy and the low number of false positives (predict a win but actually lose). Figure 4 above shows the testing accuracy of 0.73 (1 - test error rate of 0.27) and the low number of false positives which is 2 false positives.

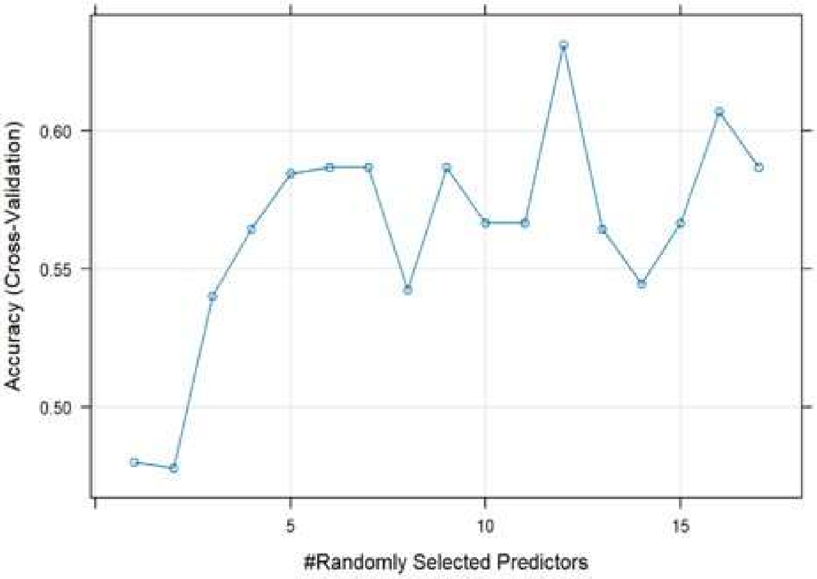


Figure 5: Cross-validated accuracy vs. randomly selected predictors for random forest model

The random forest was fitted with K-fold cross validation where K = 5 folds. The cross validation procedure includes the tuning of the number of features for each tree to ensure optimal performance. The best tune for the random forest model has an optimal number of features (parameter named mtry) equal to 12 as shown in figure 5 above.

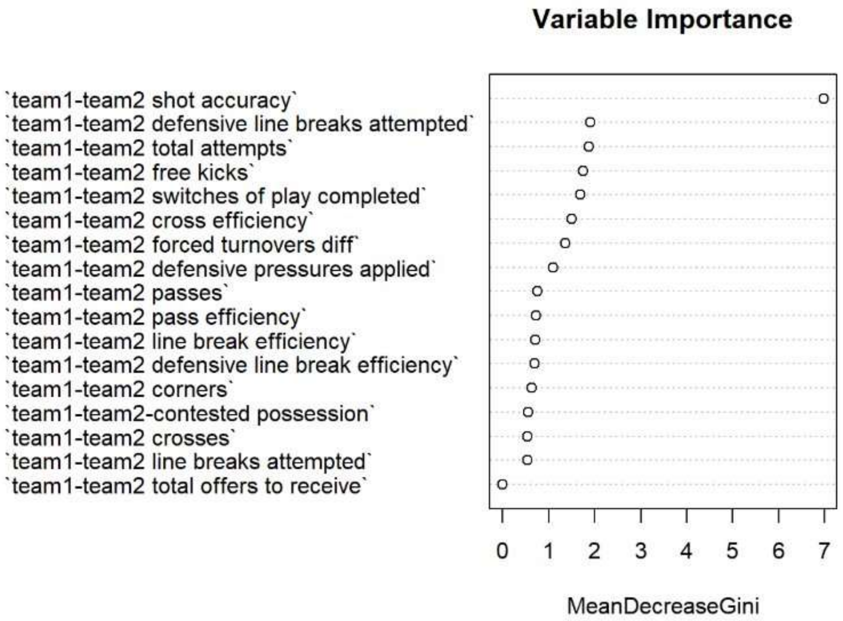


Figure 6: Feature importances for the random forest model.

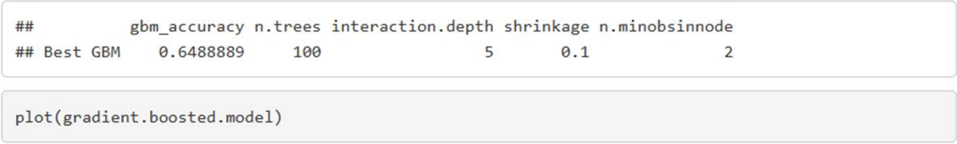
The most important feature according to figure 6 above is the net shot accuracy, suggesting focusing on higher quality shots is more important than shooting continuously. Several features beneath net shot accuracy are close in significance to each other, but net shot accuracy is much more significant in the random forest model than other features.

BOOSTING MODEL RESULTS:

For our boosting models, we utilized both a Gradient Boosted Model (GBM) and an Extreme Gradient Boosting (XGBoost) model to evaluate their performance on the training data after fitting them using 5-fold cross-validation. Both models were optimized using hyperparameter tuning to achieve the best results on the training dataset, while also identifying the most important features contributing to predicting team1 win. In addition to this process, we determined the testing error rate for each model and developed a confusion matrix for each model.

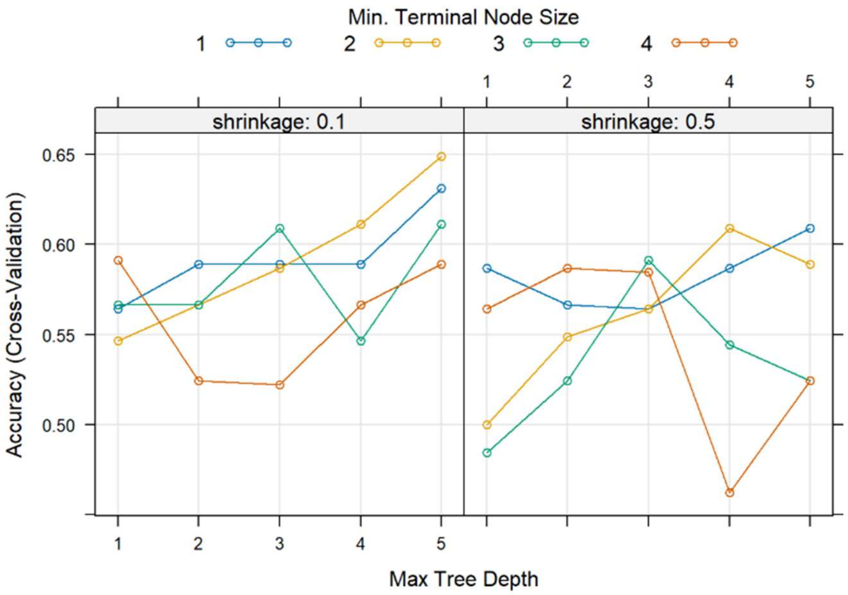
The GBM was trained by varying hyperparameters such as the number of trees, interaction depth, shrinkage, and the minimum number of observations in terminal nodes. We then plot the accuracy of the terminal nodes identified from the grid search in the figure below.

Figure 7: Gradient Boosted Model Best Tune on Training Data



```
##          gbm_accuracy n.trees interaction.depth shrinkage n.minobsinnode
## Best GBM    0.6488889     100                 5       0.1              2
```

```
plot(gradient.boosted.model)
```



As shown in Figure 7, the gradient-boosted model with the best tune had an accuracy of 64.89% on training data when considering parameters such as n.trees, interaction.depth, shrinkage, n.minobsinnode.

We did a similar procedure with the XGBoost Model, allowing us to effectively compare the models' training accuracy. The results can be found in the figure below.

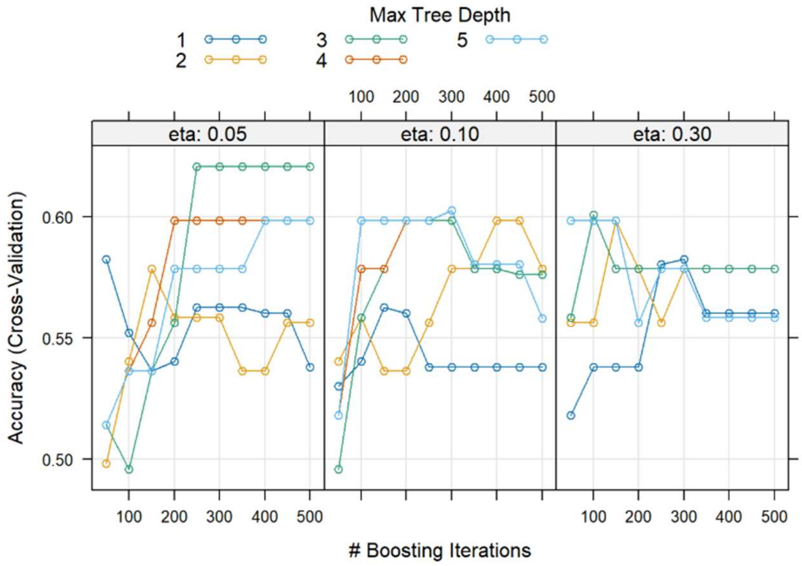
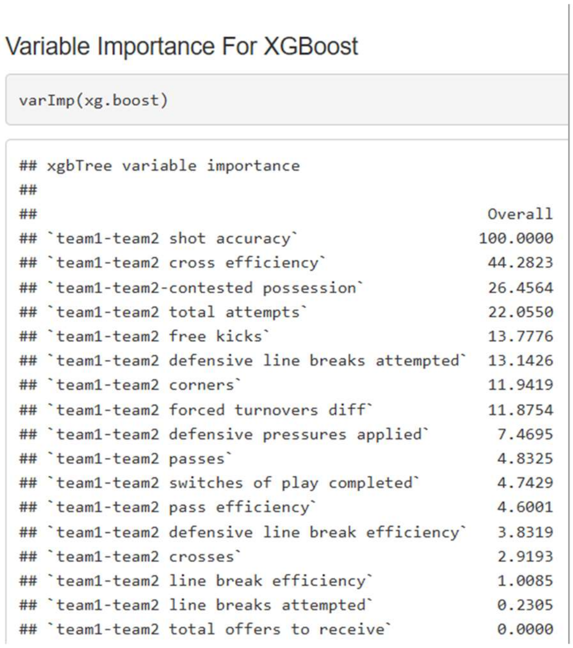


Figure 8: Extreme Gradient Boosting Model Best Tune on Training Data

According to Figure 8, the extreme gradient boosting model with the best tune had an accuracy of 62.06% on training data when considering the optimal parameters for nrounds, max_detph, and eta.. When comparing the two different models, the Gradient Boosted Model had the most accurate performance on the training data.

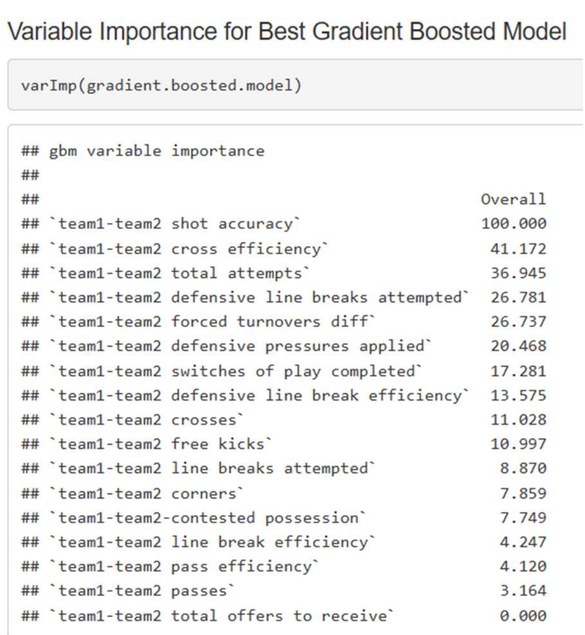
We next considered the important variables for both the GBM and the XGBoost.



Variable Importance For XGBoost

```
varImp(xg.boost)
```

```
## xgbTree variable importance
##
##                                                 Overall
## `team1-team2 shot accuracy`                    100.0000
## `team1-team2 cross efficiency`                  44.2823
## `team1-team2-contested possession`              26.4564
## `team1-team2 total attempts`                    22.0550
## `team1-team2 free kicks`                        13.7776
## `team1-team2 defensive line breaks attempted`   13.1426
## `team1-team2 corners`                           11.9419
## `team1-team2 forced turnovers diff`             11.8754
## `team1-team2 defensive pressures applied`        7.4695
## `team1-team2 passes`                             4.8325
## `team1-team2 switches of play completed`         4.7429
## `team1-team2 pass efficiency`                    4.6001
## `team1-team2 defensive line break efficiency`    3.8319
## `team1-team2 crosses`                            2.9193
## `team1-team2 line break efficiency`              1.0085
## `team1-team2 line breaks attempted`              0.2305
## `team1-team2 total offers to receive`            0.0000
```



Variable Importance for Best Gradient Boosted Model

```
varImp(gradient.boosted.model)
```

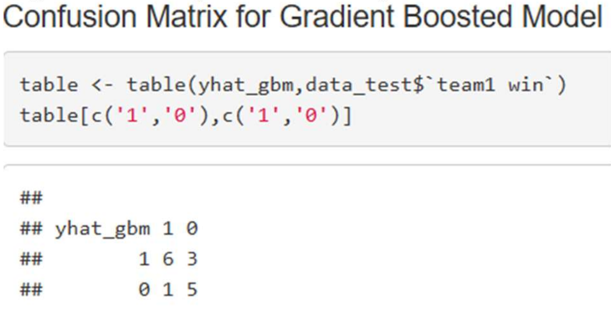
```
## gbm variable importance
##
##                                                Overall
## `team1-team2 shot accuracy`                    100.000
## `team1-team2 cross efficiency`                  41.172
## `team1-team2 total attempts`                    36.945
## `team1-team2 defensive line breaks attempted`   26.781
## `team1-team2 forced turnovers diff`             26.737
## `team1-team2 defensive pressures applied`       20.468
## `team1-team2 switches of play completed`        17.281
## `team1-team2 defensive line break efficiency`   13.575
## `team1-team2 crosses`                           11.028
## `team1-team2 free kicks`                        10.997
## `team1-team2 line breaks attempted`              8.870
## `team1-team2 corners`                            7.859
## `team1-team2-contested possession`               7.749
## `team1-team2 line break efficiency`              4.247
## `team1-team2 pass efficiency`                    4.120
## `team1-team2 passes`                             3.164
## `team1-team2 total offers to receive`            0.000
```

Figure 9: Variable Importance for Gradient Boosted Model and Extreme Gradient Boosting Model

According to the visuals the most significant predictors for the Gradient Boosted Model were shot accuracy, cross efficiency, total attempts, defensive line break attempts, and forced turnovers diff. For the XGBoost, the most significant predictors were short accuracy, cross efficiency, contested possession, and total attempts. Both models highlight the importance of shot accuracy and cross efficiency. This indicates that they both have a critical role in predicting the match outcomes.

Finally, we evaluated the testing error rate for each model. The results can be viewed in the figure below.

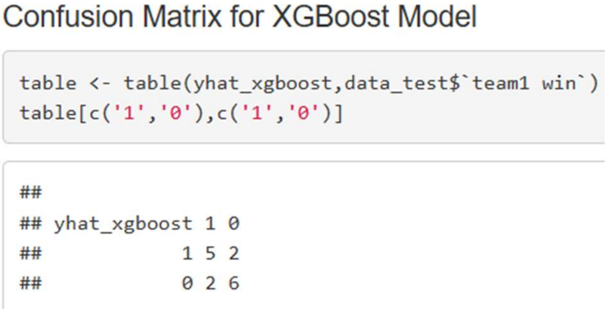
Figure 10: Confusion Matrix for Gradient Boosted Model and Extreme Gradient Boosting Model



Confusion Matrix for Gradient Boosted Model

```
table <- table(yhat_gbm,data_test$`team1 win`)
table[c('1','0'),c('1','0')]
```

```
##
## yhat_gbm 1 0
##        1 6 3
##        0 1 5
```



Confusion Matrix for XGBoost Model

```
table <- table(yhat_xgboost,data_test$`team1 win`)
table[c('1','0'),c('1','0')]
```

```
##
## yhat_xgboost 1 0
##            1 5 2
##            0 2 6
```

Based on the results of the models' performance on the testing data, both models seemed to have similar performance. Both models achieved a testing error rate of 26.67%, meaning 11 observations for both models were correct.

CONCLUSION AND WHAT'S NEXT?:

Given our results, we want to look forward to where the project can be extended to different scenarios. First, we want to prioritize further model optimization. As we continue to fine-tune these models, we can refine the existing models to generate an even higher accuracy and predictive power (or explaining power in this scenario). This will come from testing additional hyperparameters and considering more machine learning technique alternatives that can be applied to this scenario. Next, we want to expand upon the existing work of feature engineering. Creating the relative team statistics and encoding victory was a great starting point, however incorporating more features like player-level statistics and other aspects of match dynamics will truly allow us to improve predictions moving forward. Another idea is to implement the data analysis to create action, otherwise known as actionable insights. If these findings can be translated into interpretable recommendations for soccer practice, the quality of the game will improve as a whole (hopefully mainly for the US Men's National Team). We can use the important features to focus on creating offensive and defensive strategies tailored directly from the analysis. Lastly, having a stage for future testing and validation is crucial for next steps. Finding new data and continuously validating the models will act as a continuous feedback loop to ensure robustness and adaptability over time. This may be the longest step because it relies on attaining more match data especially for a rare event such as the FIFA World Cup, but it is very important in gaining more information to get a greater understanding of what features are truly important in deciding victory.