Final Report - Paul, Jackson, Mitch

I. Introduction

In recent years, the betting landscape in America has been flipped upside down with sports betting becoming a societal norm. With the legalization of this new type of betting in over 38 states, the market for this industry has grown exponentially. In 2023 alone, Americans wagered over $119.84 billion dollars on numerous sports bets, but only walked away with a little over $10 billion in revenues.[1]

With the legalization of sports books and apps that are easily accessible to the everyday consumer, many Americans find themselves placing nonstrategic and unresearched bets on their team. Moreover, Americans find that sports betting allows them to have more stake in the game, and a once boring Thursday Night Football game can give one the same rush as the SuperBowl.

As alluded to in the previous paragraph, unless one is an expert on sports betting, that person is more likely to lose than to win these bets. Gambling companies are raking in their fortune at the expense of the average sports fan's dollars. To combat this, we wrote a complex algorithm and created models that will predict a wide receiver or a running back's yards per game against a specific opponent. The model takes in account the specific player and their past performances, along with the specific opponent and their defensive performances. Our motivation was to give Americans the ability of being able to bet on their favorite games and players, and receive the joy by winning their bets and taking the money from these greedy gambling companies and putting it back into their wallets.

II. Explanatory Data Analysis

To access this data we used an R-package named *nflfastR*, which contains NFL play-by-play data and statistics all the way back to the 1999 season. For the scope of this project and for recent necessity, we only loaded the data from the 2023 season and the few games played during the 2024 season. We then combined these years into one big dataset to easily access and

[1] Where is sports betting legal - all 50 states covered 2023, accessed October 4, 2024, https://www.legalsportsreport.com/sportsbetting-bill-tracker/

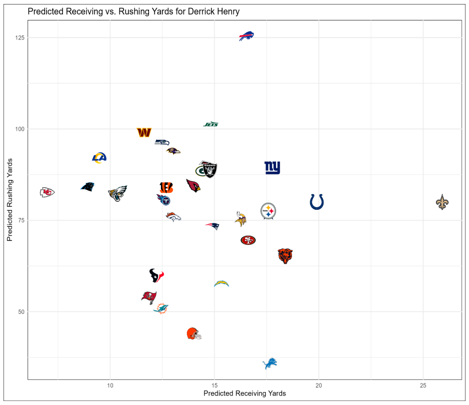
fluctuate between the seasons. The data itself contained over 300 variables and 60000 observations, but much of this was insignificant for our project. The variables in which we played close attention to were: player name, play type, rushing and passing yards, week, and opponent.

Once we combined the dataset, we had to filter it to get the necessary information for our project. We filtered the new dataset and grouped all player data by the specific player's name. Upon doing this we had to also filter the data to include only completed run and pass plays, and get rid of plays such as field goals, kickoffs, and turnovers. After receiving our filtered data, we had to divide this data and group the plays by both the offensive team and the defensive opponent.

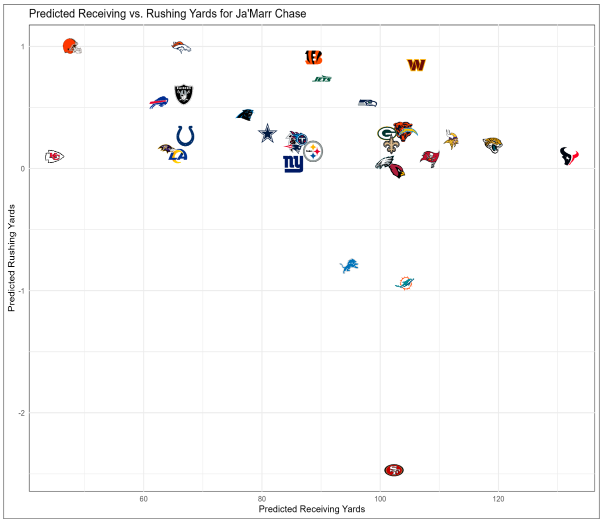
Because the package we used did not have a variable for average passing and rushing yards for a specific player against an opponent, we had to manually create these variables. When doing this we ran into the problem of finding NA's in our new data for players that did not have an efficient impact on the field or did not play at all. We simply removed these values from our dataset because gambling sites would not even list these players for available prop bets.

We then created two distinct models. One for rushing plays, and the other for passing plays. We tested an XGboost model, Decision Tree model, and a Random Forest model, and found that the Random Forest model gave us the strongest results. The train/test split was even in both, with 80% of our data going to the train for their respective model, and 20% going to the test. When running the MAE on each of these models we found very appealing results with the error on the rushing model being about 8 yards, and the error on the passing model being about 16 yards.

Because players are highly different from each other in the NFL, sample statistics would not be beneficial for this project when looking at player data. For example Cedric Wilson's average yards will not be nearly the same as Ja'Marr Chase's average yards. There is such a high variance between players. For the next portion of this analysis we are going to analyze some of our results from specific NFL players.



The chart above represents the model's prediction for Baltimore Ravens RB Derrick Henry against all opponents. The model predicts that he would have his best games against teams such as the Buffalo Bills, the New York Jets, and the New Orleans Saints, where he would capture over 100 all purpose yards in each of these games.



This chart represents the model's prediction for Cincinnati Bengals WR Ja'Marr Chase against all opponents. Because Chase is a WR and will not be rushing the ball much we will be looking at the x-axis to make our prediction. He plays extremely well against many teams in the league, but if someone were to bet the over on his passing yards, they should do so against the teams above the line which has been set.

III. Learning Algorithm Training and Testing

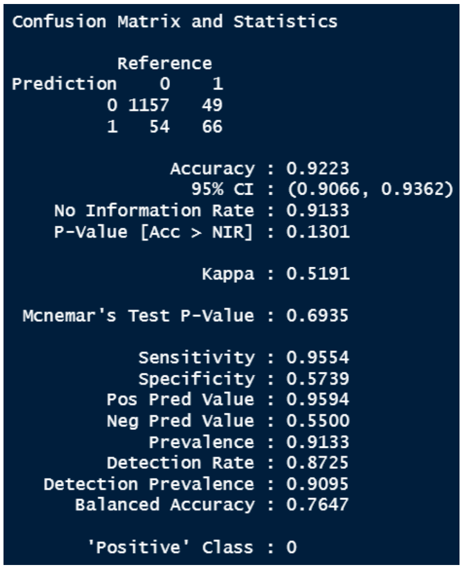
To utilize the XGBoost model, we employed the "xgboost" package in R to predict total rushing and receiving yards. The process began with preparing the data, where we loaded the training data for rushing and receiving and selected relevant features such as average rushing yards per game, average receiving yards per game, and opponent. Training was performed using the train function from the caret package with the procedure set to "xgbTree", and a cross-validation method with 5 folds was specified. We created a set of hyperparameters to fine-tune, including variables like “nrounds, max depth, eta, gamma, colsample_bytree, min kid weight, and subsample”. To evaluate the system, We used these parameters to predict the rushing and receiving yards on the test data and calculated the mean absolute error (MAE) for each prediction. The XGBoost algorithm was used to determine the most significant factors, with a focus on reducing system complexity and improving efficiency. The results showed graphs comparing estimated and actual data for both rushing and receiving yards, and confusion matrices were generated to evaluate the success of binary classification using set thresholds.

To predict total rushing and receiving yards using the Random Forest system, we utilized the randomForest package in R. Similar to the XGBoost algorithm, the process involved preparing the data by loading the training data and selecting relevant features. We then trained the machine using the "rf" procedure from the caret package and specified a cross-validation process with 5 folds. To assess the performance of the machine, we made predictions on the test data and calculated the MAE for both predictions. The caret package's varImp function was used to identify the most important factors, with a focus on reducing system complexity and improving efficiency. This resulted in plots comparing predicted and actual data for rushing and receiving yards, and confusion matrices were generated to evaluate the binary classification accuracy based on set thresholds.

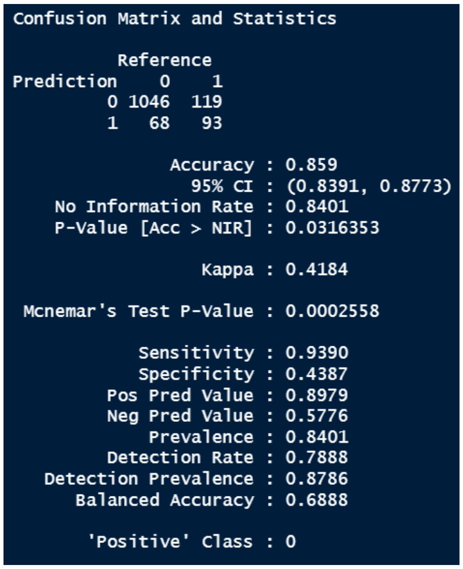
When assessing the results of the XGBoost and Random Forest models, it was found that the Random Forest performed better in terms of MAE for both rushing and receiving yards. This strong performance can be attributed to the Random Forest's ability to handle complex relationships among features more effectively in this situation. While the XGBoost model also produced decent results, it was slightly less accurate compared to the Random Forest system. Both models identified similar key features, showing consistency in feature selection.

Using the “rpart” package in R, we utilized a decision tree machine to perform the bonus procedure. This decision tree system was trained to predict both rushing and receiving yards. While it achieved higher accuracy and a lower mean absolute error (MAE) compared to the Random Forest system, we noticed that it tended to group the yardage into broader categories, providing less specific information. Although this classification method may be effective in certain situations, it was not as beneficial for my research compared to the more detailed Random Forest model. The efficiency of the decision tree demonstrates its abilities, however, in this particular case, the extensive predictions made by the Random Forest system were better suited to our objectives.

This thorough analysis of the XGBoost, Random Forest, and Decision Tree models outlines the processes involved, the methods used for selecting factors and reducing dimensions, and the relative effectiveness of each model. The results suggest that while all models are effective, the Random Forest system has a slight advantage in predictive accuracy for this particular dataset. Below the confusion matrix for rushing yards and receiving yards can be seen, with rushing on the left and receiving on the right.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1157   49
         1   54   66

               Accuracy : 0.9223
                 95% CI : (0.9066, 0.9362)
    No Information Rate : 0.9133
    P-Value [Acc > NIR] : 0.1301

                  Kappa : 0.5191

 Mcnemar's Test P-Value : 0.6935

            Sensitivity : 0.9554
            Specificity : 0.5739
         Pos Pred Value : 0.9594
         Neg Pred Value : 0.5500
             Prevalence : 0.9133
         Detection Rate : 0.8725
   Detection Prevalence : 0.9095
      Balanced Accuracy : 0.7647

       'Positive' Class : 0
```



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1046  119
         1   68   93

               Accuracy : 0.859
                 95% CI : (0.8391, 0.8773)
    No Information Rate : 0.8401
    P-Value [Acc > NIR] : 0.0316353

                  Kappa : 0.4184

 Mcnemar's Test P-Value : 0.0002558

            Sensitivity : 0.9390
            Specificity : 0.4387
         Pos Pred Value : 0.8979
         Neg Pred Value : 0.5776
             Prevalence : 0.8401
         Detection Rate : 0.7888
   Detection Prevalence : 0.8786
      Balanced Accuracy : 0.6888

       'Positive' Class : 0
```

IV. Discussion and Conclusion

When looking at our model and the data analysis, the rushing model is 92.2 % accurate where the receiving model is 85.9% accurate. You can find our summary of the models above. Both of the p values for the models are significant as well because of the value being less than 0.05. As a group, we feel that the model would do a pretty good job at predicting receiving yards and rushing yards for players against different teams. As seen in the graphs of the players, different teams are more susceptible to allowing passing yards and rushing yards. This can be a big use when it comes to the sports betting world for those who participate in that. Bettors could potentially look at our model and use that knowledge to help place their bets on certain lines that are listed. This could also be used in the fantasy football world. ESPN has their own projections of how many yards they think a certain player will get every week. Our model could provide a different and potentially more accurate number which could be insightful to managers of teams.

In conclusion, people could use this model in the real world in many different ways. As the sports betting world has come on the rise, this could help people predict the bets that they would want to place or certain players that they would choose. This could also be helpful for games such as fantasy football. When managers are looking at what players to put into their lineup, this model could help predict the rushing or receiving yards. In the end, we enjoyed looking at this data and the process that went along with it. I am excited to potentially offer this model to some people within the program or around campus and see how much they like it!

Bibliography

Where is sports betting legal - all 50 states covered 2023. Accessed October 4, 2024. https://www.legalsportsreport.com/sportsbetting-bill-tracker/