IEOR E4523 Data Analytics Group Project
Topic: NFL Offensive Play Prediction
Group Name: NaN
Group Members: Kin Wai Wong, Jiaqi Liu, Zhongyuan Gu, Yiwen Sun

1. Introduction

This project aims at predicting NFL offensive plays using data from a variety of different sources. To enhance the feasibility of team defensive strategies based on prediction outcomes, we divided the main objective into two parts: 1) predicting the National Football League offensive play types (rush/pass); 2) predicting yards of rushing plays.

In general, the project could provide insights from two perspectives:

a. For in-game decision-making: Optimize teams' in-game defensive/offensive play decisions and tactics by improving the prediction accuracy of offensive play types and their possible outcomes;

b. For team management at the club level: Redesign the training programs to assist players in increasing reaction speed and gaming-reading abilities.

To achieve our goals, we combined play by play data of 2016-2019 seasons and team stats to form a dataset for further analysis. Then, we explored data and implemented features engineering to develop, compare, and select optimal predictive models on both rush/pass prediction as well as yards prediction. In the end, we developed a recommendation system for team managers to apply the project findings to real-world in-game situations.

2. Data Collection

a. Play Type Prediction Data(Play by play data from season 2016-2019 from nflsavant.com)
After all the feature engineering, there are 82 columns and 117738 rows in this dataset. Columns includes 'Quarter', 'Minute', 'Second', 'Remaining', 'OffenseTeam', 'DefenseTeam', 'Down', 'ToGo', 'YardLine', 'SeriesFirstDown', 'SeasonYear', 'Yards', 'Formation', 'PlayType' etc.

b. Rushing Yards Prediction Data(Rush play data from Kaggle NFL big data bowl)
After all the feature engineering, there are 89 columns and 504812 rows in this dataset. Columns includes 'YardLine', 'Quarter', 'GameClock', 'PossessionTeam', 'Down', 'Distance', ‘'FieldPosition', 'HomeScoreBeforePlay', 'VisitorScoreBeforePlay', 'NflIdRusher', 'OffenseFormation', 'OffensePersonnel', 'DefendersInTheBox', 'DefensePersonnel', 'PlayDirection' etc. Further description on the features could be found at https://www.kaggle.com/c/nfl-big-data-bowl-2020/data,

c. Every single NFL game data from 2016-2019 from https://www.pro-football-reference.com/ by web scraping
This dataset will be used for computation and creation of new columns in the first two datasets.

d. Team statistics rankings data from 2016-2019 from https://www.pro-football-reference.com/
This dataset will be used for computation and creation of new columns in the first two datasets. The columns are explained in the Appendix 1.

e. Please refer to readme.txt for the procedures of running the code.

3. Data Cleaning & Preprocessing

a. Play Type Prediction Data
We first standardized the team names because the team names in this data conflicts with the data from pro-football-reference.com. Then, we removed all rows with NaN values. Since we are predicting pass or rush type, we filtered out plays that aren't pass, rush, or scramble(which we treated as pass). Next, by computing the time remaining for every row(every play) and using the game data, we got the scoreboard exactly before the play and add this as a column to the dataset. Then, by using the team ranking data, we computed the team ranks on multiple offense and defense categories(based on distribution) of the offensive team and defensive team of every single play. Last but the least, we generated derived features from the data, which contained cumulated game information like cumulated interceptions and cumulated pass incompletes, to help us better utilize the data.

b. Rushing Yards Prediction Data

We first standardized the team names to avoid conflicts with other data. Then, we fixed typos, extracted relevant info from categorical variables and standardized categories (e.g. We categorized weather conditions based on the frequency of words occurred in weather descriptions). Next, we constructed new features based on raw data (e.g. player BMI, age). Besides, we cleaned the personnel data, correcting typos, filling out missing positions and transforming them into a standard way. Finally, we compute team ranks of the offensive team and defensive team of every single play.

4. **Analysis**
   a. Play Type Prediction Data analysis

      Our analysis of play-by-play data have two main purposes. One is to better understand the NFL game process by going through each features and doing exploratory data visualisation. Another one is to visualize the relationship between play types (pass/rush) with different features which may influence pass type decisions. Here we list our 10 analysis perspectives. Graphs are shown in our Jupyter notebook 'Play Type Prediction Data Analysis.ipynb'.

      i. Investigation into seemingly strange values
      ii. A game process for example
      iii. Yards distribution analysis
      iv. Formation & Play Type analysis
      v. Formation & Yards analysis
      vi. Down & Yards & Play Type analysis
      vii. Down & Formation analysis
      viii. Trend of play type in different seasons
      ix. Comparison of play type of different teams
      x. ToGo (Yards to go) & Yards analysis

   b. Rushing Yards Prediction Data analysis

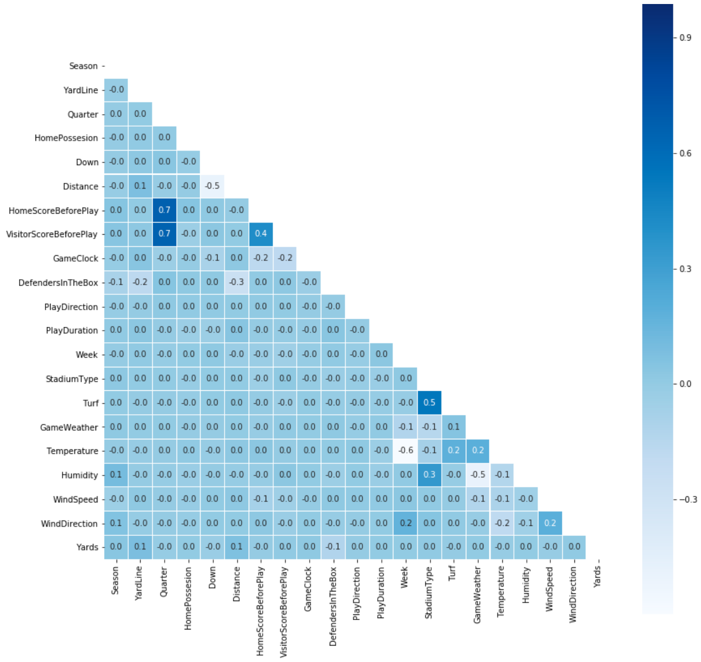
      In general, analysis towards the Yards Prediction Data consist of 4 parts: distribution of the dependent variables, play-by-play data analysis, athlete data analysis and team ranking data analysis. They all based on our assumption of factors that might influence the yards gain. Most of the graphs are shown in our Jupyter notebook 'Rushing Yards Prediction Data Analysis.ipynb'.

      i. Distribution of the dependent variable

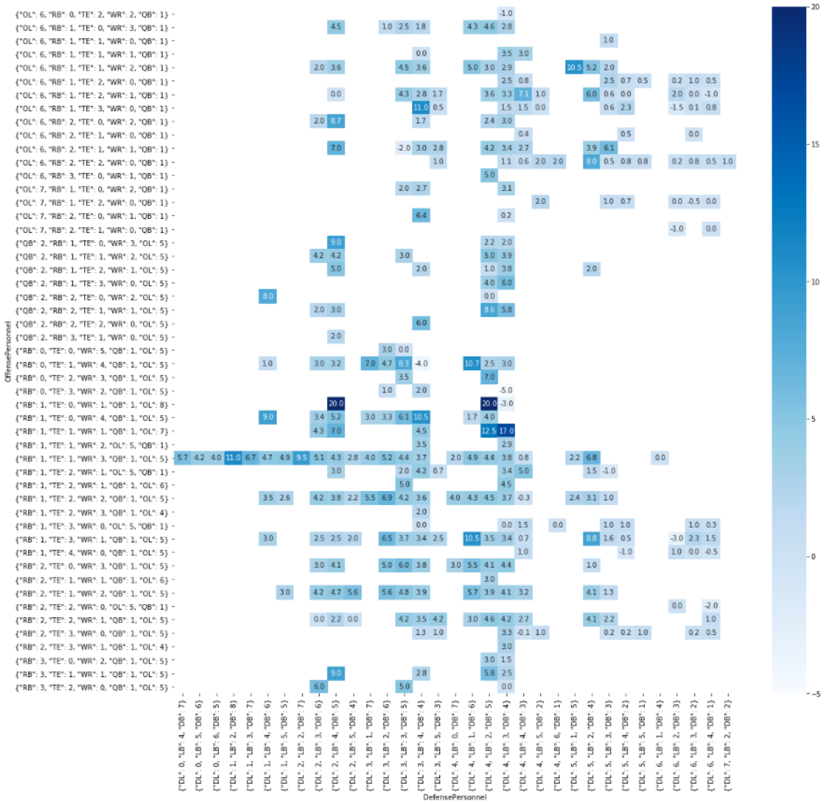
         As an initial step, we drew the distribution of the dependent variable in order to have a basic idea about our target value. We found out that the distribution was highly skewed. Thus, we did a further analysis of what might influence the yards by grouping the data by outliers and normal values.

      ii. Play-by-play data analysis

         In this part, we first did correlation analysis among all the play-based features and showed the result by heatmap. In the meantime, to make the results clearer, we applied mask to the heatmap which resulted as a map showing the correlation between a pair of features only once.



We also assumed that teams would choose different offensive and defensive personnel based on the current situation(measured by yards left, down and others). Meanwhile, we believed that there might be a counter scheme exists in the defensive and offensive personnel, meaning that when a certain defensive personnel faces a certain offensive personnel, they would more likely to lose more yards and vice versa. We confirmed our assumptions by drawing a heatmap computing the average yards of each pair of offensive and defensive personnel combination (Shown below), a heatmap counting the occurrence of personnel under different situations and other box plots. (Shown in our analysis jupyter notebook)

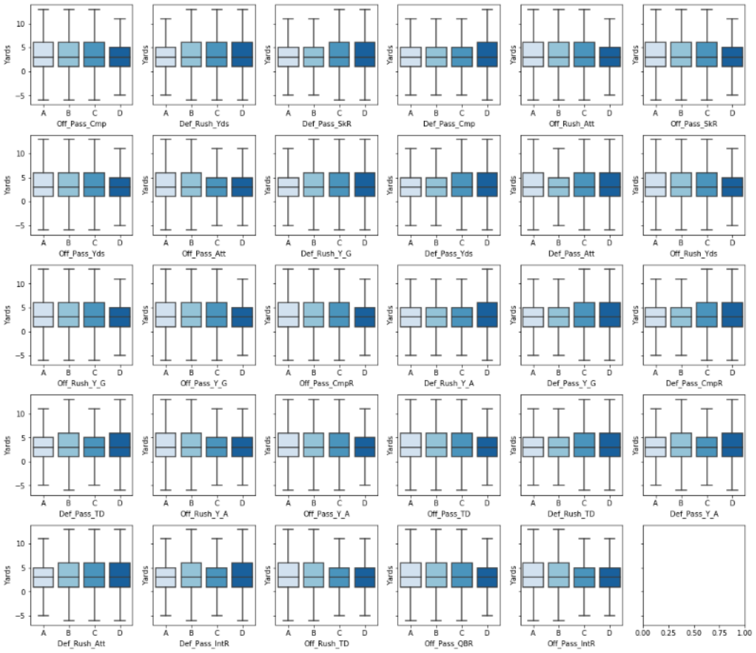


iii. Athlete data analysis

We explored the relationship between yards and player data, including their BMI and age.

iv. Team ranking data analysis

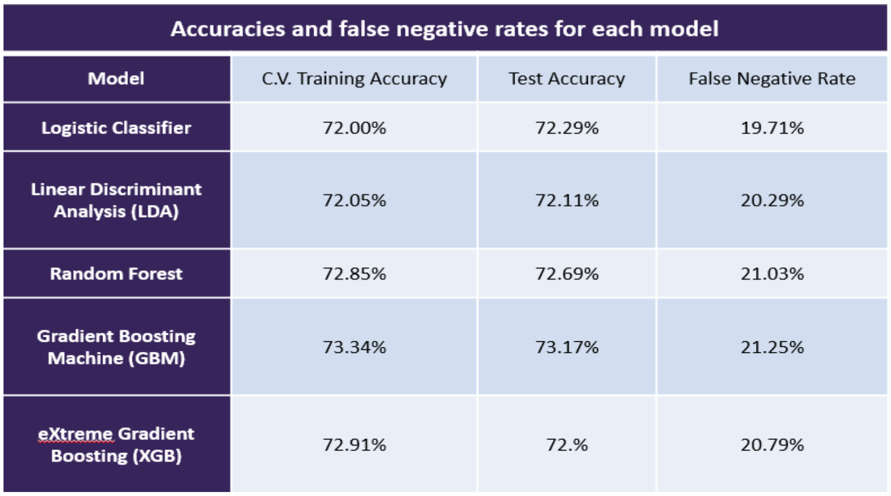
We collected a lot of information that evaluate teams performance from different aspects and converted them into rankings. The assumption behind this was that the relationship among teams also existed such a counter scheme, meaning that some characteristics of a certain team could make it more likely to win or gain more yards on a rush play against teams with some other characteristics. Under this assumption, we plotted the yards distribution among all the ranking criterion and figured out for each criteria, teams with relatively better ranks always had better results. This proved our assumption right. Thus we put these ranking data into our model.



5. **Feature Engineering**
   a. Constructing New Features
   We created a column for duration between TimeSnap and TimeHandoff, a column for Yardleft(Yards left to touchdown) based on the current Yardline and FieldPosition(Home or Away team side), a column for DefendersInTheBox Vs Distance, and computed the offensive and defensive teams' score before the play as two columns.
   b. Incorporating Team Ranks
   We computed the offensive team's and defensive team's play tendency and statistical rankings on multiple categories (e.g. Offensive team's rushing attempt, rushing yards per game, Defensive team's opponent rushing yards per attempt, opponent number of rushing TDs etc.) All the ranking columns name and their descriptions are given in the appendix 1.
   c. Normalizing Features
   We converted GameClock into how many seconds left in the game, DateOfBirth into age, normalized the unit of PlayHeight to inches, converted invalid and rare Offense and Defense Personnels into regular ones, as well as reducing complexity on StadiumType, Turf, Weather, WindDirection and WindSpeed.
   d. Standardizing Features
   Since our data includes many categorical columns and some of them have multiple values, we had to do categorical data encoding. Encoding them by one hot encoder might lead to improper weight of these features in our model. In order to address this problem, we iterate through all the encoder methods, compared their f-score and chose the one that gives the best score(BaseNEncoder) to encode our data. Moreover, we standardized all the features using StandardScaler to ensure variables are having the same weights while building models.
   e. Reselecting Features
   After running each model, we modified features to be included based on the Feature Importance plot, and we repeated these steps until the best set of features was found.
6. **Modeling and Results**
   a. Play Type Prediction

We took three steps to build play type prediction models. First, select features based on causality in games (whether a feature is before a play or is the result of a play), and correlation with the result. Then, we try six machine learning models and carried out hyperparameters tuning to get more accurate models. Finally, we use cross validations to obtain more reliable prediction results of each model. Here are some of our interesting findings:

i. Model results comparison

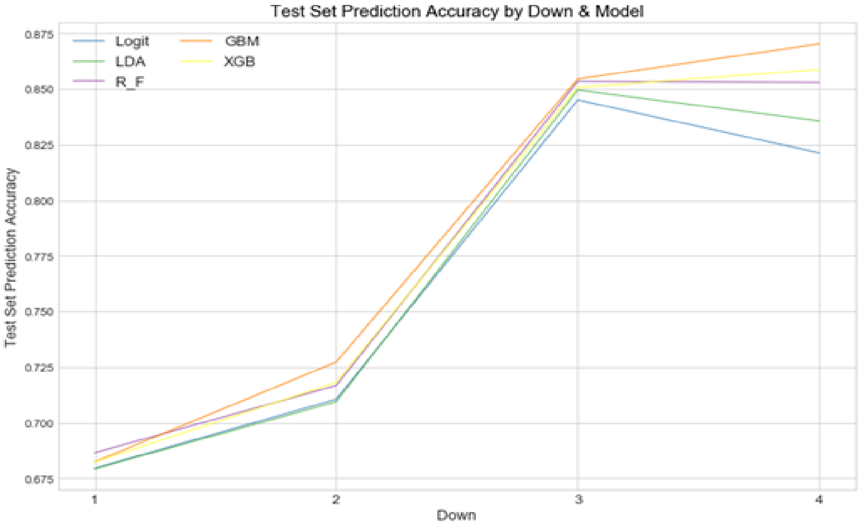


Accuracies and false negative rates for each model

Model	C.V. Training Accuracy	Test Accuracy	False Negative Rate
Logistic Classifier	72.00%	72.29%	19.71%
Linear Discriminant Analysis (LDA)	72.05%	72.11%	20.29%
Random Forest	72.85%	72.69%	21.03%
Gradient Boosting Machine (GBM)	73.34%	73.17%	21.25%
eXtreme Gradient Boosting (XGB)	72.91%	72.%	20.79%

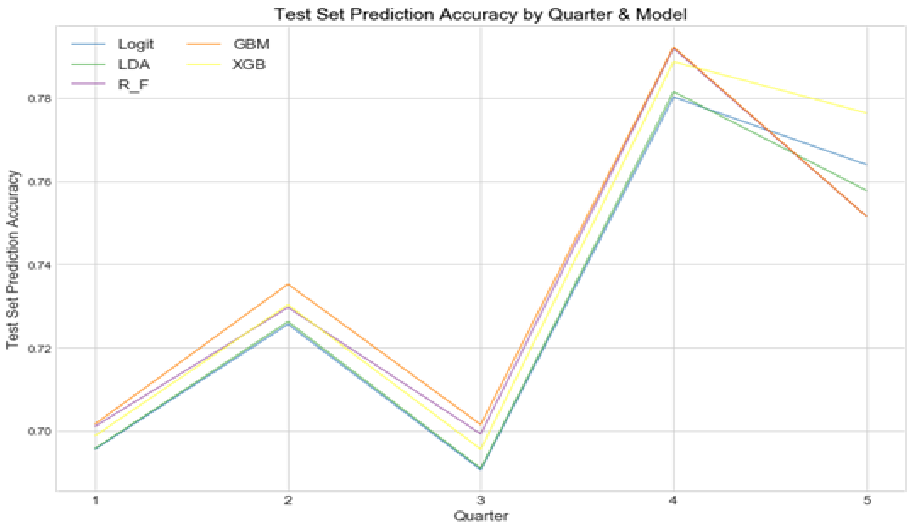
We find GBM is the most accurate in terms of both training accuracy and test accuracy (73.17% is also close to the highest accuracy in related literature, which is around 75%). Also, the test accuracies of all the five models are quite close to the training accuracies, meaning that our models are likely to avoid the overfitting problem. In addition, we're more concerned with the false negative rate and list the results. This is because a false negative sample means predicting a pass (positive) to be a rush (negative). In NFL games, a formation to defend rush is also less ready to defend a pass. So we don't want the false negative rate to be high. For the models we find there's no so much difference.

ii. Prediction accuracy by down



We also observe that our accuracy increases on later downs, which may relate to the fact that teams are more concerned about gaining larger yards in the later downs so they can have bigger probability to win the first down. Therefore, the available choices of the play are limited. We also see very little deviation in the performance of our algorithms relative to each other, suggesting that our best models tended to outperform the others across different inputs. The conclusion is consistent with previous literature.

iii. Prediction accuracy by quarter

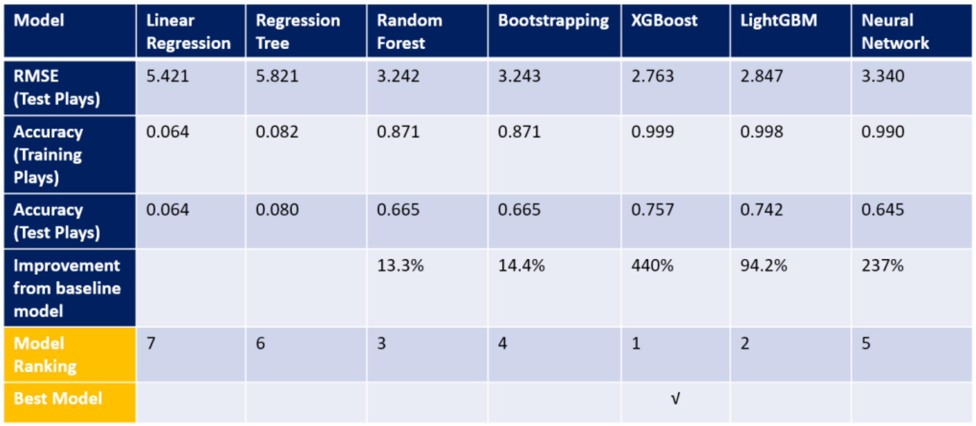


We still see very little deviation in the performance of our algorithms relative to each other, except for GBM in the 4th and 5th quarter. GBM is still the best if we mainly focus on regular game time. It's also interesting to notice that the 2nd and 4th downs are more likely to predict, because they are closer to the end of half-time and the end of the game, which largely affect the final result of the game. Specifically, the 4th downs are the most likely to predict.

b. Rushing Yards Prediction

After selecting features based on our prior analysis, we fitted our data into several models, including Linear Regression, Regression Tree, Random Forest Regressor, Bootstrapping Regressor, XGBoost Regressor, LightGBM and Neural Networks.

For each model, we found a way to find the best hyperparameters. To be specific, we iterated through all possible combinations of the features for the Linear Regression model and chose the one that gave the best RMSE. For other machine learning models, we implemented GridSearchCV many times to find the best set of hyperparameters for each of them. After tuning each model to its best, we chose the model giving best RMSE on testing set and used it as the algorithm for our recommendation system. The following graph shows the final results:

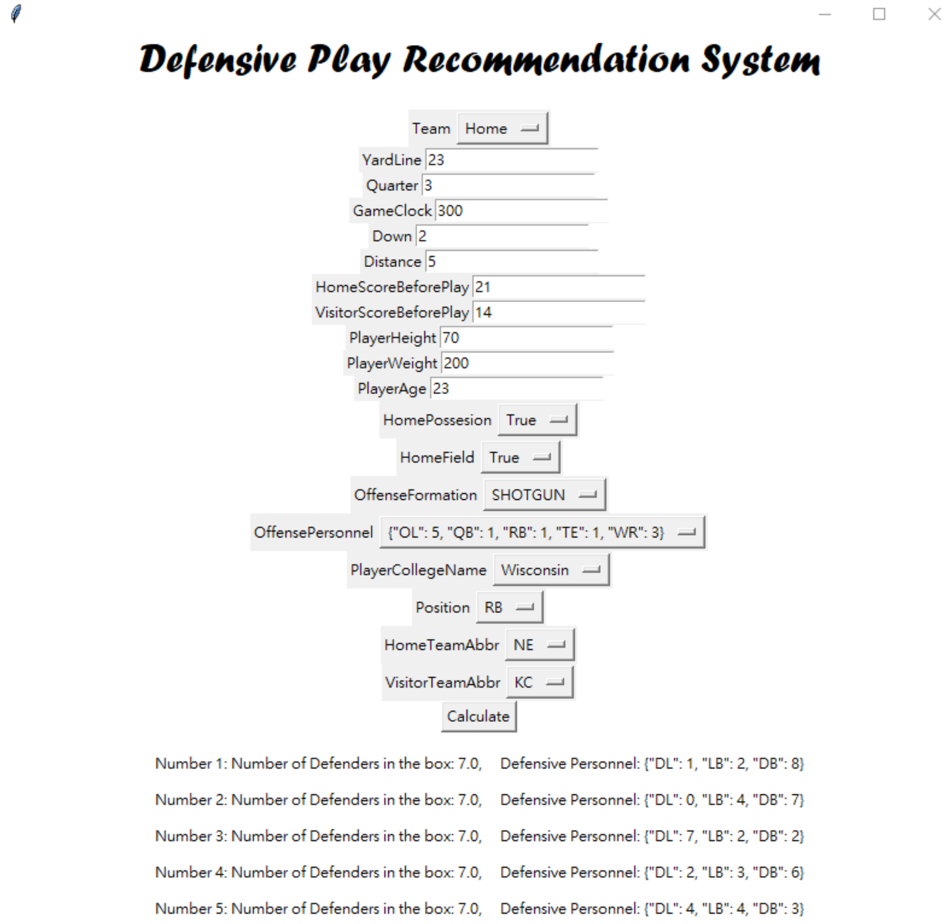


Model	Linear Regression	Regression Tree	Random Forest	Bootstrapping	XGBoost	LightGBM	Neural Network
RMSE (Test Plays)	5.421	5.821	3.242	3.243	2.763	2.847	3.340
Accuracy (Training Plays)	0.064	0.082	0.871	0.871	0.999	0.998	0.990
Accuracy (Test Plays)	0.064	0.080	0.665	0.665	0.757	0.742	0.645
Improvement from baseline model			13.3%	14.4%	440%	94.2%	237%
Model Ranking	7	6	3	4	1	2	5
Best Model					√		

As we can see from the graph, our XGBoost Model is the best model with an accuracy of 75.7% on the testing set, thus it would be our backend algorithm for our recommendation system.

7. **Recommendation system**

Since our model now has a decent accuracy rate on predicting the yardage gain on a rushing play, we decided to create a recommendation system program to help defensive teams to minimize the yardage gain by the offensive teams. To achieve this goal, we created a program with Python Tkinter GUI. Users are asked to enter some attributes about the game and the play and the application would show the results in seconds. Here is the basic idea: After the program receives all the inputs from the user, it would call our best model, the XGBoost model, to iterate all possible combinations of DefendersInTheBox and DefensePersonnel. Then, the program would sort the results with ascending order and show the Top 5 combinations the defensive team could adopt to minimize the yardage gained by the opponent. Here is a sample output of the program:



Defensive Play Recommendation System

Team Home
YardLine 23
Quarter 3
GameClock 300
Down 2
Distance 5
HomeScoreBeforePlay 21
VisitorScoreBeforePlay 14
PlayerHeight 70
PlayerWeight 200
PlayerAge 23
HomePossesion True
HomeField True
OffenseFormation SHOTGUN
OffensePersonnel {"OL": 5, "QB": 1, "RB": 1, "TE": 1, "WR": 3}
PlayerCollegeName Wisconsin
Position RB
HomeTeamAbbr NE
VisitorTeamAbbr KC

```
Number 1: Number of Defenders in the box: 7.0,    Defensive Personnel: {"DL": 1, "LB": 2, "DB": 8}
Number 2: Number of Defenders in the box: 7.0,    Defensive Personnel: {"DL": 0, "LB": 4, "DB": 7}
Number 3: Number of Defenders in the box: 7.0,    Defensive Personnel: {"DL": 7, "LB": 2, "DB": 2}
Number 4: Number of Defenders in the box: 7.0,    Defensive Personnel: {"DL": 2, "LB": 3, "DB": 6}
Number 5: Number of Defenders in the box: 7.0,    Defensive Personnel: {"DL": 4, "LB": 4, "DB": 3}
```

8. Reference

Fernandes, C., Yakubov, R., Li, Y., Prasad, A., and Chan, T., Predicting plays in the National Football League

Lee, P., Chen, R., & Lakshman, V. Predicting Offensive Play Types in the National Football League.

Appendix 1: Team ranking columns description

Offensive ability metrics:

Off Pass Cmp: Offensive team’s total pass completions this year
Off Pass Att: Offensive team’s total passing attempt this year
Off Pass Cmp%: Offensive team’s pass completion rate this year
Off Pass TD: Offensive team’s total passing touchdowns this year
Off Pass Int%: Offensive team’s pass intercepted rate this year
Off Pass Sk%: Offensive team’s sacked rate this year
Off Pass Yds: Offensive team’s total passing yards this year
Off Pass Y/G: Offensive team’s passing yards per game this year
Off Pass Y/A: Offensive team’s passing yards per attempt this year
Off Pass QBR: Offensive team’s quarterback rating this year
Off Rush Att: Offensive team’s total rushing attempt this year
Off Rush Yds: Offensive team’s total rushing yards this year
Off Rush Y/G: Offensive team’s rushing yards per game this year
Off Rush Y/A: Offensive team’s rushing yards per attempt this year
Off Rush TD: Offensive team’s total rushing touchdowns this year

Defensive ability metrics:

Def Pass Cmp: Defensive team’s opponent total pass completions this year

Def Pass Att: Defensive team's opponent total passing attempt this year
Def Pass Cmp%: Defensive team's opponent total pass completion rate this year
Def Pass TD: Defensive team's opponent total passing touchdowns his year
Def Pass Int%: Defensive team's interception rate this year
Def Pass Sk%: Defensive team's sack rate this year
Def Pass Yds: Defensive team's opponent total passing yards this year
Def Pass Y/G: Defensive team's opponent passing yards per game this year
Def Pass Y/A: Defensive team's opponent passing yards per attempt this year
Def Rush Att: Defensive team's opponent total rushing attempt this year
Def Rush Yds: Defensive team's opponent total rushing yards this year
Def Rush Y/G: Defensive team's opponent rushing yards per game this year
Def Rush Y/A: Defensive team's opponent rushing yards per attempt this year
Def Rush TD: Defensive team's opponent total rushing touchdowns this year