**Performance Assessment: D214- Capstone Data Analysis**

**A. Research Question**

**1.** For this capstone assessment, the research question will be as follows. To what extent do the following predictor variables affect the total number of rushing touchdowns: age, games started, rushing attempts, rushing yards, first down runs, longest rush, yards per attempt, yards per game, and number of fumbles (Carlan 2024)?

**2.** The hypothesis for this research question will be as follows: these variables (age, games started, rushing attempts, rushing yards, first down runs, longest rush, yards per attempt, yards per game, and number of fumbles) statistically significantly affect the number of rushing touchdowns (Carlan 2024).

**3.** In order to win games in the NFL, it is crucial that teams know how to improve their offense. Teams need to have complex, creative, and innovative ways to score points, both in the passing attack as well as the ground game (rushing). This project will focus on rushing statistics to see if we can predict rushing touchdowns based on the various predictor variables (rushing statistics). If we can use this analysis to discover which variables are statistically significant to determining rushing touchdowns, then NFL teams can focus on improving their metrics in those key variables in a way that produces more points and in turn wins more games. This summary provides the context of the research question (Carlan 2024).

**4.** Random Forest Analysis will be the technique used to analyze this data source and to address the aforementioned research question. This analysis, along with all necessary data cleaning, will be completed using R Studio.Random Forest is a statistical algorithm that can be used to determine which variables in a data set are statistically significant in determining the target variable. In this case, it can be used to determine which of the aforementioned predictor variables have the strongest and weakest statistical significance in determining the total number of rushing touchdowns. The algorithm itself can capture non-linear relationships and is very robust in handling multicollinearity which is especially important in sport statistics where variables may have high levels of correlation, such as the number of rushing attempts and total rushing yards. This justifies the use of the random forest algorithm to answer this research question (Carlan 2024).

**B. Data Collection**

**1.** The data that will need to be collected will be a series of sport statistics data that will include the variables mentioned as well as the players name, the year, and some other variables that will not be used for the analysis**.** The data will be gathered from the CSV file published on Kaggle (see source). After which, the data will be imported into R for cleaning and random forest analysis (Carlan 2024).

**2.** One advantage of this data-gathering method is that the data is published on Kaggle.com which is a free and opened source platform. That means that anyone is free to use the published data sets without paying for any various fees or getting permission from the original owners, etc., so long as the author/publisher of the data is appropriately cited. One disadvantage of this method is that anyone can publish data sets about anything. The author’s profession and credibility are not something that can be verified without doing extensive research into the author and the data set itself. This means that we cannot simply assume that the data sets are presenting legitimate and accurate data. Since this project is for academic purposes only, however, the legitimacy of the data is not relevant to our needs but is something that would be extremely important in a real-world business scenario.

**3.** No other challenges were encountered during the data collection process.

**C. Data Extraction and Preparation**

**1.** The data extraction and preparation steps for this analysis are relatively simple. Extracting the data is as simple as downloading the CSV file from Kaggle (the link is listed in the sources). The screenshot below showcases this step for extracting the data:

A screenshot of a web page

Description automatically generated

After the data has been extracted, the data was then moved to a location on my personal computer (for security purposes, I will not include screenshots of the file directory on my personal computer for this step). After moving the file to an appropriate location, it was renamed to rushing\_stats so as not to get confused with the new, cleaned file name that would be created once the data set has been prepared for the analysis.

**2.** Before we prepare the data for analysis, the first step is to load necessary packages into our RScript programming environment. The following packages were used to prepare the data for analysis:

* Dplyr: to create a subset of the dataset
* Ggplot2: to create boxplots to check for outliers
* Caret: to create training and testing data splits

**3.** To begin preparing the data for the analysis, the data was imported into R, after which a subset of the data was created. This subset was created to include only the 10 variables of interest within the data and to remove all variables that would not be used for the analysis. After subsetting the data, I then used the colSums command in R to check to see if there were any missing values within our data that we may have needed to replace. After completing this function in R, it was determined that there was no missing data within our data set. The following screenshot showcases these steps in R:

A computer error message

Description automatically generated

**4.** The next step in the data cleaning process was to use boxplots to check for outliers within the data. Using R, a boxplot was made of each variable individually in order to check for outliers within the data and to determine the appropriate action as to whether to keep the outliers or to remove them. After observing all of the boxplots, I made the decision to keep any that were present. Every field except for one had at least one outlier present. The decision was made to keep them for several reasons. One of those reasons being that the random forest algorithm is actually designed to handle outliers fairly well, but the most important reason being that outliers are actually fairly common in sport statistics, and unless there are outliers that are extremely egregious, it makes more sense to keep them than to remove them. Football is a sport that has a lot of players on one team and a varying degree of participation within the roster. Every team has a depth chart or backups and backups to the backups for any particular position. Statistically, it makes sense that a player who is a starter (GS variable) would have more yards, touchdowns, etc., than a player at the bottom of the depth chart who doesn’t get as much playing time. Also, it can be expected that superstar athletes (of which there are only a handful in the entire league) would produce outliers when compared to less productive players on a team. The following screenshots go through each variable and discusses why none of the outliers are too egregious to be considered for removal.

A graph with a number of objects

Description automatically generated with medium confidence

This boxplot of age showcases a few outliers in the positive direction with the highest being age 45, and the mean being around 26-27. NFL players have varying lengths in their career, but it is uncommon for players to play into their 40s, however, it is not unheard of. Most famously Tom Brady played until he was 45, which fits these outliers, and other players have as well. There is nothing too egregious here to suggest human error, such as a player under 20 or over 50.

A graph with a line and numbers

Description automatically generated with medium confidence

This boxplot of fumbles showcases a few outliers. Some players have better hands when it comes to ball protection and some players have been known to fumble a lot. It would be impossible to fumble less than 0 times and it is not unheard of (although rare) to fumble into the 20s. Also, considering that we are trying to determine the statistical relevance of fumbling to the number of rushing touchdowns, I am choosing to keep all of these outliers for the context of the analysis.

A graph with a bar

Description automatically generated with medium confidence

The Games Started variable has no outliers, which is a good sign. You cannot start less than 0 games and you cannot start more than 16 (now 17) games in one season.

A graph of a bar graph

Description automatically generated with medium confidence

This boxplot is for the number of rushes that led to a first down. Like the previous examples, you cannot rush for a negative first down and it is expected that players would have various degrees of success running for first downs, which can be explained by the outliers; and, in order to best determine if this field is statistically significant in the number of rushing touchdowns, all data points should be included.

A graph of a graph showing a bar

Description automatically generated with medium confidence

This boxplot is for the number of rushing attempts. It is impossible to rush for a negative attempt, and it can be expected that players in run-heavy offenses would have more attempts than others, and that players at bottom of depth charts don’t get as many touches. For the context of the analysis, it would be important to keep them all in order to determine how statistically significant this variable is.

A graph with a line and a line

Description automatically generated with medium confidence

Here we have the boxplot of the longest rush. The outlier peaks at 99, which makes sense. If a team is backed up at the 1-yard line, 99 yards is the furthest rush possible. There are also some negatives here, which is possible. A team could run a trick play, where a receiver or some other position who rarely ever rushes the ball gets the ball within the backfield and is then tackled behind the line of scrimmage for negative rushing yardage. Like with the previous fields, it is important to keep them all for the analysis as players who get a very long rush could contribute to more touchdowns over the course of a season.

A graph with a line and numbers

Description automatically generated with medium confidence

This is the boxplot for our target variable, rushing touchdowns. As with previous fields, you cannot have a negative rushing touchdown. Since this is the target variable, it is definitely more critical to keep the outliers, as doing so can help find stronger statistical significance in the predictor variables. None of the outliers are also too egregious. It makes sense that teams with more dominant backs and rushing offenses would have more rushing touchdowns for their featured back. Ladanian Tomlison famously broke the record for most rushing touchdowns in a season at 28, breaking the previous record of 27. You can see both of those years represented here. If there was anything above that number, then it would be human error.

A graph with lines and numbers

Description automatically generated

This is the boxplot for rushing yards per attempt. As with the other examples, neither the outliers on the positive or negative ends stand out as being too egregious and the data is relevant for our analysis. Players could have had a one-time rush or trick play resulting in a single negative or positive yard per attempt. If you only attempt once (trick play) then those outliers do make sense.

A graph of a game

Description automatically generated with medium confidence

This boxplot of rushing yards per game is correlated to the previous boxplot in a way. If a player were to only appear in one game, perform one rush, then that explains a possible negative yardage per game, as their usage would be low over the course of the season. Also, dominant running backs can, and have, averaged over a hundred yards a game over the course of a season. These outliers are most likely from the first-string star running backs who get the ball multiple times over the course of a game over the course of the season. It is also why it is crucial to keep these outliers in our data set for the analysis. We need to know just how statistically significant this variable is as whole.

A graph of a number of numbers

Description automatically generated with medium confidence

Our last boxplot is for the total number of rushing yards. This field is important for the context of the regression analysis, so like the others, it is important to keep these data points. It is definitely possible for star running backs to reach over 2000 yards in a season and has happened several times before. We can also assume that players who play more over the course of the season will finish the season with more rushing yards, which can explain many, if not all, of these outliers.

In summary, while most of the fields contain the presence of outliers, the random forest regression algorithm is equipped to handle their presence, and none of the outliers appear to be too egregious and they can all easily be explained. It is also important to keep them because, in the context of sport statistics, their presence is crucial in determining which of our predictor variables are the most statistically significant in determining the number of rushing touchdowns, our target variable.

**5.** The last step to prepare the data for analysis is to partition the data in order to create our training and testing data. Using R, I set the seed for reproducibility and partitioned the data on an 80-20 split. This means that our training set will have 80% of the data and then our testing set will contain 20% of the data. The trains the algorithm in order to better make predictions using the testing data, which is why the training set contains more data. After creating both of these data sets, each one was written as a new CSV file and saved onto my personal desktop.

**6.** The code used for the data cleaning steps, as well as all four of the CSV files, has been attached alongside this written assessment. The following screenshot shows the full code used for data preparation:

A screenshot of a computer code

Description automatically generated

**7.** The justification for using R for data preparation is that R has various packages that make data cleaning a seamless process. It is also very useful for any visualizations (such as boxplots) that make it easy to detect the presence of outliers, as well as performing exploratory data analysis. It is a free and open-source platform that is available for anyone to download and use. These are just a few of the advantages of using these techniques in R for data cleaning. One disadvantage of using R is R stores everything in your system’s memory which means that bigger data sets can slow down your computer and take a long time to load and perform certain tasks.

**D. Analysis**

**1.** Now that the data has been extracted and prepared for analysis, we begin, as before, by loading the packages that we will be using in R. For this part, the necessary packages will be the following:

* randomForest: as the name suggests, this package is for the random forest regression algorithm
* caret: this package will be used to for cross-validation in order to determine the appropriate value for mtry in the random forest algorithm

After loading these packages into R, the next step is to import the data. This includes the original data as well as the training and testing data sets that were created in the data preparation steps.

**2.** Cross-validation is a technique that is used to determine the optimal number of a specific variable in a handful of algorithms. In the case of the random forest regression algorithm, it can be used the determine the beset value for “mtry”. It is important to determine this value as the number that is used for mtry can greatly affect how accurate the algorithm is at predicting our target variable. For the random forest algorithm, this variable is used to determine how many variables are being used at every split within the decision trees that are inside the “forest”; and for this analysis, the number of “trees” in that forest will 500. That is the default value that is used by R, as it is large enough to generate enough trees in the forest to make predictions and to determine the most statistically significant predictor variables. The following screenshot shows the code that was used for cross-validation as well as the output where it was determined that the best value for mtry is 5:

A computer code on a white background

Description automatically generated

**3.** Now that the appropriate value for mtry has been determined, the next step would be to train the random forest model so that it can later be used to make predictions against the testing data set. To do this, we run the random forest algorithm in R, setting the rTD variable as the target variable and then set data equal to the training data, or df\_train, as I called it. The default value of 500 for ntree was used and then mtry equal to 5 as discussed in the previous step. This creates our model that will be used to make predictions. The output of this model is shown in the screenshot below:

A computer code with numbers and letters

Description automatically generated

By printing the summary of the initial model, it was determined that the mean of squared residuals (or MSE) is 1.578251 and the percentage of the variance is 77.72%. The MSE value is used to determine how far off the models predictions were from the actual value. In this case, the model’s predictions were off by 1.578251 touchdowns from the actual number of rushing touchdowns. Now you obviously cannot have a percentage of a rushing touchdown but since this is an average that is why the number comes out as a percent value. The percentage of variance signifies that 77.72% of the variance of the target variable (rTD) is explained by this model. In other words, a little over three-quarters of the variability in the number of rushing touchdowns can be explained by this model. This is a relatively high amount and signifies that the model is a good fit.

**4.** Now that the initial model has been made, the testing (df\_test) data can be used for our predictions. By creating the model with the training set, we can then use it against the testing set and use the algorithm to predict the number of rushing touchdowns. After which, there are various ways to measure its accuracy. The following screenshot showcases this part in R, including a command that shows the head of our predictions comparison (mean the actual value vs the predicted value):

A computer screen with text

Description automatically generated

The head command showcases just the first few rows of data. This printed command is showing the actual number of rushing touchdowns versus what the model predicted. For example, in the first row the actual number of rushing touchdowns was 7 but our model overpredicted a value of 8.609700. In row two, the actual number was 9 but our model underpredicted a value of 8.138367. As stated before, since these are averages, that explains why these numbers are returned as a decimal when you cannot have a fraction of a touchdown in a real game. In the next step, we will determine the accuracy of the model by looking at its mean squared error, root mean squared error, and the mean absolute error, as well as the value of R-squared.

**5.** The following screenshot shows the code in R used to calculate the mean squared error (MSE), the root mean squared error (RMSE), the mean absolute error (MAE), and the value for R-squared:

A computer code with numbers and letters

Description automatically generated

* MSE: this is the mean, or average, of the differences squared between our predicted values and the actual values within the testing data. For this model our MSE came out to 1.592295. This means that the average difference in between the actual number of touchdowns to the predicted value is approximately 1.59 touchdowns.
* RMSE: This is the square root of the MSE, which for our model comes out to 1.261862. This is a more interpretable version of the MSE, as it removes the square from the differences. In this case, the RMSE suggests that the average difference between the actual number of touchdowns to the predicted is approximately 1.26 touchdowns.
* MAE: This is the absolute value of the differences, which is less sensitive to outliers as the positive and negative directions are removed. The value here is 0.6596553. This suggests that the difference between the actual number of touchdowns to the predicted is, on average, 0.66 touchdowns.
* R-Squared: Lastly, the value for r-squared is calculated at 0.768542. This metric, when written as a percentage, is telling us the percent variance of the data. In this case, approximately 76.85% of the variability of rushing touchdowns is explained by this model, which is a high value.

**6.** Next, I made a plot of the data comparing the actual number of rushing touchdowns to the predicted amount, as shown by the screenshot below:

A graph of a running touchdown

Description automatically generated with medium confidence

The red line is a line at 45 degrees added in for reference. This graph is helpful to determine just how close the predictions are to the actual value. Most of the data on this plot cluster around the added line. This visually showcases how the model was fairly accurate in making predictions. The model either overestimated or underestimated the amount by a small enough amount to where the points cluster around the line. If the predictions were perfect then they would all be on that line, but you can see where they cluster around. No model is perfect, but the fairly consistent clustering is a good sign that the model was a good fit. There are some outliers as well which can be expected when making predictions.

**7.** The final step to complete our analysis is to determine which of the predictor variables are the most statistically significant in predicting the number of rushing touchdowns. This was determined by taking the importance of our initial model and observing the increase in node purity. The following code showcases those steps:

A computer screen shot of a computer code

Description automatically generated

Variables with a higher purity are more statistically significant. This can be better seen on the following plot:

A graph with numbers and lines

Description automatically generated with medium confidence

Variables that have a point further away on the horizontal axis have a higher purity which means those variables are the most significant when predicting rushing touchdowns. Variables closer to the vertical axis are less significant. In this plot, you can see that the number of first downs is the most significant while the number of fumbles is the least significant.

**8.** This concludes the random forest regression analysis. The code to perform this analysis has been attached alongside this written assessment.

**9.** The justification for using the random forest regression algorithm to analyze this data set is that the algorithm itself can capture non-linear relationships and is very robust in handling multicollinearity which is especially important in sport statistics where variables may have high levels of correlation, such as the number of rushing attempts and total rushing yards (Carlan 2024). The algorithm is also well equipped to determine which variables of a particular data set are statistically important, which was the ultimate goal of this analysis, which is a great advantage for choosing this particular method. Some disadvantages would be that sometimes it is difficult to interpret some of the calculations, especially in sport statistics where you cannot have fractions of a touchdown. It is also intense for your computer and can be time consuming due to your computational memory.

**E. Data Summary and Implications**

**1.** In summary, using a random forest regression algorithm, the goal of this analysis was to determine which of the aforementioned predictor variables within our data are the most statistically significant in determining the number of rushing touchdowns. By running this algorithm, we calculated a MSE of approximately 1.59, which means that, on average, the models predictions were within 1.59 touchdowns of the actual value. This combined with an R-squared value of 76.85% suggests that the model was a relatively good fit for making predictions. After which, the importance of the initial model was derived to determine exactly which of the predictor variables were the most statistically significant. It was determined that the number of rushing first downs (r1D) was the most statistically significant, followed by rushing yards (rYds) and rushing attempts (rAtt) with the number of fumbles being the least statistically significant. In the context of professional football, this makes sense as a running backs ability to sustain drives and keep the chains moving would suggest an ability to score more points thus leading to more rushing touchdowns.

**2.** One limitation of this analysis is that the model is outputting continuous variables (or decimals). In a real football game, you cannot score fractions of a touchdown. Additional rounding is something that may be required in future analyses, depending on the ultimate goal.

**3.** Since the ability to get first downs is the strongest predictor variable in scoring rushing touchdowns, it is recommended that teams focus on their ability to sustain their drives. This can be done with focusing on the play calling as well as offensive line blocking to give your running back the best change of obtaining first downs in non-passing situations (3rd and inches, for example).

**4.** One suggestion for a future study would be to try a different method of analysis for more interpretable results. Negative binomial regression and poisson regregssion are examples of other algorithms that are better suited at providing predictions when the target variable is an integer or whole number. Another recommendation is to incorporate more features into the analysis, such as the weather in outdoor games or the opponents defensive schemes to provide additional insights into how to best score rushing touchdowns.

**F. Sources**

Jadhav, Rishab. “NFL Rushing Statistics (2001-2023).” Kaggle, 4 Mar. 2024, [www.kaggle.com/datasets/rishabjadhav/nfl-rushing-statistics-2001-2023](http://www.kaggle.com/datasets/rishabjadhav/nfl-rushing-statistics-2001-2023).

Carlan, Christian. "Predicting NFL Rushing Touchdowns with Random Forests." *MSDA Capstone Topic Approval Form*, 4 Apr. 2024.