Predicting NFL Outcomes Using Machine Learning

Nicholas Romano, Robert Kelley

Bellarmine University Data Science Program

Abstract

In recent years, there has been a rise in the use of sports analytics alongside an increase in volume of available data in the NFL. This project's goal is to develop a predictive analytics model that predicts the results of upcoming NFL games by training a K-Nearest Neighbor Model on team statistics of previous games. The Kaggle dataset that was used for this project consisted of game data that was pulled from ESPN and includes a wide variety of team statistics for both the home and away team for each NFL game since 2002. Statistics that were included range from third down and fourth down efficiencies to passing yards gained, rushing yards gained, and offensive time of possession. For this project, Python was used for the data manipulation, visualization and modeling stages. The model for this project provided accuracy scores of 62% and 60%.

Introduction/Objectives

This project focused on incorporating offensive and defensive team statistics over previous games played and integrate them into a model that will predict who would win an upcoming NFL game.

The decision for considering the performance of a team over a span of a few games was made because it was thought that taking into account past performances might help take into consideration teams are either going through somewhat of a slump and teams that have been playing on the top of their game.

This project focused on the future research directions propositions by Matt Clifford and Tuncay Bayrak in their *Decision Analytics Journal*:

- Limiting the data to a shorter window of seasons to train the model on
- Trying to primarily focus and train a model on previous game data.

Objectives

- The goal of this project was to develop a predictive analytics model that can be used to predict the outcomes of NFL games using team's previous game statistics.

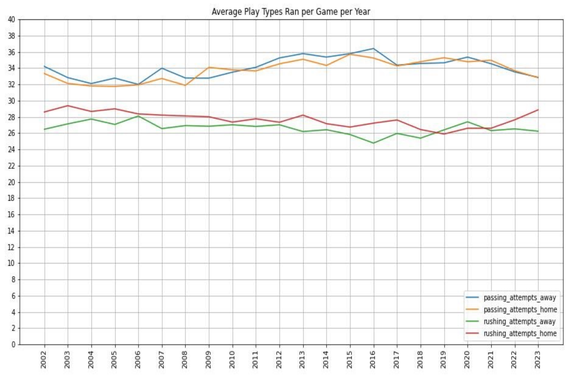

Figure 1: Since the 2012, the NFL has seen an upward trend on the average number of passing plays and a downturn on the average number of rushing plays ran per game. This observation helped to pinpoint the cutoff of the data used for the model.

Materials/Methods

Data Collection:

The data for this project was from a dataset from Kaggle posted by user CVIAXMIWNPTR. At the time of retrieval, the dataset contained game data dating back to the 2002-2003 to the 2022-2023 NFL seasons and included both regular and postseason games. The user obtained the data by scraping the data from ESPN.

Data Cleaning

- Dealing With Games Resulting in a Tie
  - Out of the total number of games played, only 14 games resulted in a tie (0.2482%). These 14 games were eventually dropped from the dataset. This change enabled the target variable, game_result, to be binary (either the home team or away team wins).
- Dealing With Postseason Games
  - With the dataset including postseason games, the 32 NFL teams ended up playing different amounts of games played. This was resolved by dropping the postseason games, causing each team to have roughly the same number of games played.
- Obtaining Team and Average Team Statistics
  - 32 separate data fames were created, one for each team. For each of these data frames, the statistics were modified so that each games consisted of the previous 4 game averages for the team for both offensive and defensive statistics. Then each team data frame was combined back into one data frame.
- Condensing Number of Seasons
  - During the EDA of the dataset, it was found that beginning in 2012 that there was an uptick in the number of passing plays ran per game alongside a slight downturn in the number of rushing plays ran per game. This insight provided the season to trim the dataset down to.

Model Selection

The model selected was the K-Nearest Neighbor (KNN) Classifier Model. In the case of this project, the KNN Classifier model will be looking at games where the home and away teams had similar statistics entering a game compared to the game that the model is currently trying to predict.

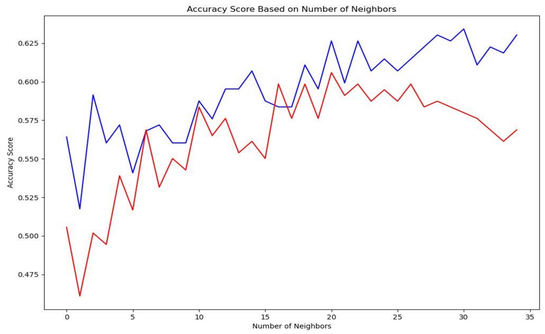

Figure 2: A for loop was created to iterate through the train, test, and validation steps of the model with different numbers of neighbors being considered. The lines represent the accuracy scores of the testing data (blue) and validation data (red).

Results

Model Selection (Cont.)

Train-Test-Validate Split of Data

- For the model, the validation data that was selected was the most recent year in the dataset (the 2022-203 NFL Season).
- The remaining games were used to create the training and test data. To match the roughly 270 games in the 2022-2023 NFL season, 10% of the remaining data was used to create the testing data. (2309-257-269).

Optimal Number of Neighbors

- The optimal number of neighbors to evaluate the future games was determined by using a for loop that changed the number of neighbors for the model.
- The optimal number of neighbors that returned the highest accuracy score for both the testing and validation data was 21 neighbors.
- **Testing Data Accuracy:** 62.64591439688716%
- **Validation Data Accuracy:** 60.59479553903345%

Model Results

Running the model with 21 neighbors, the following classification reports and confusion matrices were reported.

Testing Data Accuracy: 0.6264591439688716

	precision	recall	f1-score	support
0	0.57	0.49	0.52	109
1	0.66	0.73	0.69	148
accuracy			0.63	257
macro avg	0.61	0.61	0.61	257
weighted avg	0.62	0.63	0.62	257

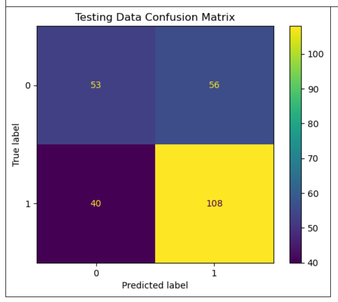

Figure 3 (top), Figure 4 (left): The classification report (top) helped provide the numerical breakdown of the accuracy of the predictions made by the model on the testing data. The confusion matrix (left) help to provide a visual breakdown of the predictions made by the model on the testing data.

Validation Data Accuracy: 0.6059479553903345

	precision	recall	f1-score	support
0	0.56	0.47	0.51	118
1	0.63	0.72	0.67	151
accuracy			0.61	269
macro avg	0.60	0.59	0.59	269
weighted avg	0.60	0.61	0.60	269

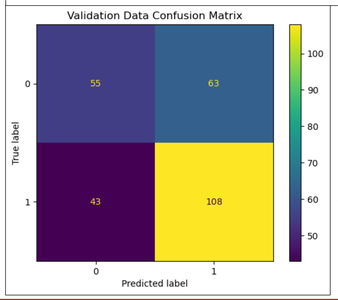

Figure 5 (top), Figure 6 (left): The classification report (top) helped provide the numerical breakdown of the accuracy of the predictions made by the model on the validation data. The confusion matrix (left) help to provide a visual breakdown of the predictions made by the model on the validation data.

Conclusions

Conclusions

With the KNN Classifier Model predicting the outcome of future games produces accuracy scores in the mid-to-upper fifty percent and lower sixty percent range. A plausible reason for the lower accuracy scores is that the model has difficulties when it comes to accurately predicting away team victories. When trying to predict both the testing and validation data, the model recorded a large number of false negatives relative to the number of away team victories. This could be due to there being more games in which the home team wins.

Future Work

Future work for this project could inslude trying different ranges for the number of games that are included in the rolling averages of the team statistics, similar to that of the optimal number of neighbors. Perhaps other models need to be taken into consideration like Support Vector Machines.

References

Tools

- Python v. 3.11.5
- Pandas v. 2.1.4
- NumPy v. 1.26.3
- Matplotlib v. 3.8.0
- Sci-kit learn v. 1.2/2

References

Kaggle Dataset

- cviaxmiwnptr. NFL Team Stats 2002 - Feb. 2023 (ESPN). *Kaggle*. Retrieved 18. Jan 2024 from https://www.kaggle.com/datasets/cviaxmiwnptr/nfl-team-stats-20022019-espn.

Decision Analystics Jorunal NFL Article

- Clifford, Matt, and Tuncay Bayrak. "A predictive analytics model for forecasting outcomes in the National Football League games using decision tree and logistic regression." *Decision Analytics Journal,* vol. 8, 2023. *Science Direct,* https://doi.org/10.1016/j.dajour.2023.100296. Accessed 15 Jan. 2024

ESPN Schedule Webpage

- "NFL Schedule." ESPN. https://www.espn.com/nfl/schedule/_/week/5/year/2022/seasontype/3. Accessed 22 Feb. 2024.

Contact

Nicholas Romano

School Email: nromano@bellarmine.edu