Brandon Mather
6/13/2023
Predictive Analytics
DSC630-T302 (2237-1)

Project Milestone 2

Problem Statement:

In the NFL finding any advantage is welcomed and needed. Every team wants to be the best and will always try to find an edge. That is the goal of my project, to use machine learning to determine which teams are most likely to win in a head-to-head matchup and to find which statistics are the most important to the predicted outcome. This could help NFL coaches prepare for games to know which areas of the game are the most important to win and why the other team is favored. Not only will this be helpful for NFL teams themselves, but this could help with sports gambling becoming legal in more states. This model could be used to help with odds of games to give companies better chances at making money.

Description of Data:

The data I'll be using will all be from Pro Football Reference (Pro Football Reference, 2023). This is all data that is collected straight from the NFL and will involve team statistics over individual statistics since I will be looking at head-to-head matchups. I will be using the last ten years' worth of data both from the offensive side of the ball and the defensive side of the ball. The actual statistics selected will be the most influential and I feel like make the biggest difference in the outcome of games. I feel like ten years will give a good amount of data that is accurate to the current way the game is played. Those statistics would be first down conversions, total yards gained, total passing yard gained, total rushing yard, turnovers, offensive penalties, scoring percentage, first downs allowed, total yards allowed, total passing yards allowed, total rushing yards allowed, turnovers made, defensive penalties, and scoring percentage allowed.

Brandon Mather
6/13/2023
Predictive Analytics
DSC630-T302 (2237-1)

Evaluation Method/Plan:

The type of model that I plan on using is a logistic regression model which is a type of classification model. This model will be the best to use for probability to predict the outcome of a head-to-head matchup. I plan to evaluate the results of the model by finding the accuracy which measure the ratio of correct predictions from all predicted results, precision which measures what proportion of the positive predictions is actually correct, recall which represents the model's ability to correctly predict the positives out of actual positives (higher is better), and F1-score which combines the precision and recall of the model and can be used to find out if the data is imbalanced.

My biggest hope is that I learn not only what outcome is more likely in a head-to-head matchup, but also which statistics are the most important in that outcome. This I believe would be the biggest benefit to a team, to know which part of their game they need to focus on to win. The actual ability to find the chances of a win would be more beneficial to companies that are creating the odds for companies since that's the area most bets are most placed.

Unfortunately, with gambling there can be many ethical concerns. The biggest thing that I need to ensure is that I choose data that can't be manipulated. Many companies that collect data for the NFL will create their own statistics to try and figure out what makes a team the best. I want to avoid these since they aren't concrete stats that are collected. As an example, the company I'm getting my data from, Pro Football Reference, comes up with their own stat which is called a simple rating system or a SRS which is a team quality relative to average which is margin of victory plus strength of schedule. This could be considered valuable if you are trying

to create a point spread, but I would make sure to take that out for my model. The other thing is to just be transparent and make it known where I'm getting my data, how it was collected, and why the data I'm using is important.

If my original plan doesn't work out, I think I would first change the type of data I'm working with. If I find that most of my statistics aren't contributing to the model, then I would want to make sure I change that so it's more influential to the outcome. I also may look at new models. Maybe a logistic regression model isn't the best fit for my data, and I can get stronger outcomes with a different type of model.

Project Milestone 3

Data and Expectations:

I have collected 10 years' worth of NFL statistics, 2013-2022. I had to collect three separate types, yearly offensive statistics, yearly defensive statistics, and overall record for each year. With this data I will have all the statistics I mentioned before and will be able to answer my questions with the target being wins. I will need to clean my data, there are a lot that of stats I don't need for my model, but the actual numbers won't be changed. The biggest thing that I've learned from the past is to make sure none of the stats are repetitive or tell me the same things. I will add them all together into one dataframe before modeling, but that will be the only modification that will be made. A logistic regression model should still be the best model to use at this time, all the data should go in without issues. The best visualizations that will be useful will be scatter plots, bar graphs, and histograms. These will help visualize the relationships

between the data. My original expectations of being able to predict head-to-head matches are still achievable.

Project Milestone 4

First Attempt/Changing Data:

This was my first attempt at creating a logistic regression model. After cleaning the data, I started the model, but I was getting a 0% accuracy. After trying to troubleshoot, I decided to test the model and realized my model was trying to predict all 10 years' worth of wins, which makes sense with the data I collected. I decided to try new data, but still have the same goal.

```
#Building Logistic Regression Model
Y = overall_stats.Wins
X = overall_stats.drop('Wins', axis = 1)
```

```
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=1)
```

```
pipe = make_pipeline(StandardScaler(), LogisticRegression())
```

```
pipe.fit(x_train, y_train)
```

```
Pipeline(steps=[('minmaxscaler', MinMaxScaler()),
                ('logisticregression', LogisticRegression())])
```

```
#Accuracy Score
pipe.score(x_test, y_test)
```

```
0.0
```

```
pd.DataFrame({'Offensive Yards': x_test['Offensive Yards'],
              'Yards Allowed': x_test['Yards Allowed'],
              'Wins': y_test,
              'Prediction': pipe.predict(x_test)})[:10]
```

	Offensive Yards	Yards Allowed	Wins	Prediction
27	56187	54005	79	80
3	56047	52286	92	80
22	61432	55690	97	100
18	54522	55198	84	80
23	53533	59148	61	78
17	60137	55725	79	100
21	59029	54562	111	100
28	57533	54032	103	100

Brandon Mather
6/13/2023
Predictive Analytics
DSC630-T302 (2237-1)

Import and Cleaning:

My initial idea for my model was to use data from all NFL teams in the last 10 years. I realized that wouldn't get the results I was looking for, so my new data for each game for the last 5 years for the Green Bay Packers specific. This includes both offensive and defensive stats. To clean my data, I needed to do a couple of things. The first was to combine all the data together. It was easiest to initially combine all the offensive data and the defensive data together separate and clean it that way since all the columns were easy to match. I dropped all the statistics that weren't needed for the model either because they wouldn't make a large enough difference in a win, or they were repetitive to the data that was kept. Then I renamed the columns so they would be understood by anyone that looked at the statistics and so they weren't just abbreviations. After those were complete, I combined both groups of stats together into one dataframe.

Interpretation or Results/Conclusion:

I decided to do two models instead of just one. My initial thought was to just do a logistic regression model, but I thought it would be useful to do a Random Forest Model as well to see if one or the other was better. After shifting my data for the models from finding all possible teams win possibility to just one team per model the accuracy went up a lot. My initial logistic regression model wasn't trying to predict a single game outcome, but instead the amount of wins a team would have in an entire 10-year period. This led to a 0% accuracy, and the model was not working properly. I decided it would be better to have just one team, the models would easily be able to be copied and to other teams. I also shrank the data size to just 5 years since this would be more accurate to any team's current playing style. The goal would ultimately stay the same. My logistic regression model produced a 88.2% accuracy and my Random Forest produced a 94.1% accuracy. The precision,

recall, and F1 score were higher on Random Forest as well. Both Models seem to predict Wins better than a loss. I also did a feature importance analysis for both models. For the regression model the biggest stats that impacted the results in either a positive or negative way were points scored, points scored against, run yards against, interceptions thrown, and interceptions by the defense. For the Random Forest it was Point scored, points scored against, run yards, run yards against, and interceptions made by the defense. So, Random Forest puts more emphasis on Running ability then the Logistic Regression.

Overall the model was a success and could be replicated for each team to figure out outcomes of NFL games and for coaches to find out which area of their game is the most important to focus on.

References:

NFL Standings & Team stats. Pro Football Reference. (2023). https://www.pro-football-reference.com/