Feature Engineering Project

Author Names: Salim Jivani, Ali Tahririan, Aniv Chakravarty, Satish Bitra
Project proposal submission date: 10/18/2022

Project Title:

NFL PredX 2022

Idea Description:

The National Football League (NFL) is the most popular sports league in the United States. American football's governing body, which was established in 1920, created the blueprint for a prosperous modern sports league. Our team has decided to use indicators to predict which team will win a game in the NFL league. These indicators are yards, penalty, or running back yards.

Goals and Objectives:

Despite the fact that current algorithms concentrate on event recognition, player style, or team analysis, predicting the outcomes of a match is still a difficult task. The goal is to combine different features to determine what defines a victory. Data is included from 2014 to 2021.

Motivation:

The National Football League is one the most followed professional sports leagues across the country. The motivation for this project is the interest in watching football and linking sports statistics to the results. The practice of using Machine learning model and sports analytics mainly based on the same mathematics – statistics. After going through various sports predicting websites the curiosity increased to create a similar model that can be built up with other features such as player-specific data.

Significance:

Over the past few decades, to make any informed decisions the team owners, team officials, coaches, and players have come to depend more and more on sports analytics. The Houston Astros also have used similar analytics to their defensive techniques that helped them to win their first World Series title in their franchise history. In other words, statistics make a difference. If statistics can be used internally by the teams to increase the probability of winning, there can be absolutely no reason why external viewers or observers cannot utilize the same statistics to predict which team has a higher probability of winning any event.

Literature Survey:

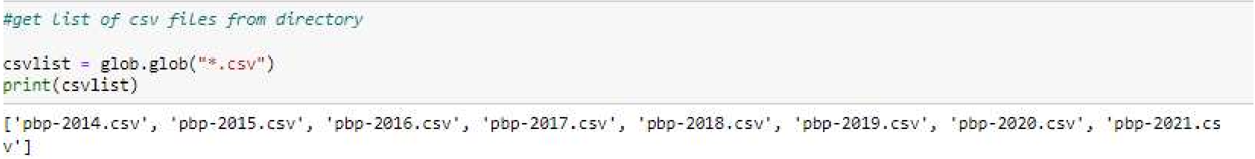
There are a large number of predictive models based on various NFL datasets. Some use Twitter data with tags and statistics from NFLdata.com based on weekly, pre game and post game tweets utilizing bag of words [1] while others make use of more traditional numeric features such as player performance, bookmaker spreads and moving averages before being passed to a neural network [2]. We decide to make use of a simple prediction model on our obtained dataset consisting of textual and numeric data with various feature extraction methods to determine their overall effectiveness on the model prediction. When it comes to feature selection on sports data a robust process is required and for the NFL in particular. The CART methodology appears to perform the best on a 28 feature dataset making use of decision trees and ANN by comparing win home/away against numeric data [3]. Further critical analysis of the various machine learning methods [4] showed that Naive Bayes to be the most effective.

Objectives:

1. First we would like to run the data as is through various logistic regression models. Get the accuracy levels
2. Then we would like combine, filter, augment, standarized the features and run them through the models. Get the accuracy levels.
3. From then we would like to determine the next best path. Maybe a combination of changing feature and model architecture. Try a bunch of combinations to determine the best model.

Features:

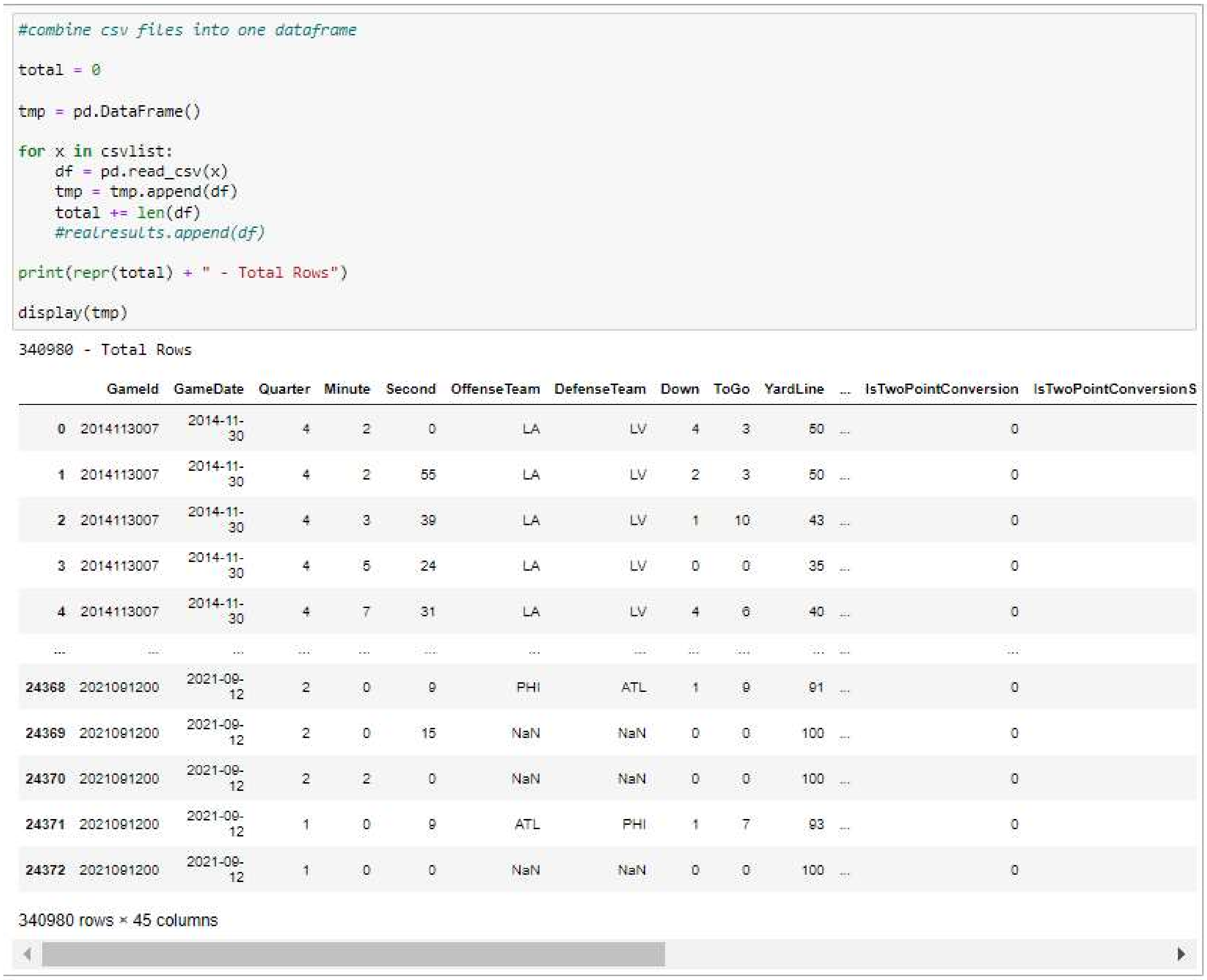
We have got data from 2014-2021 on NFL games played, play-by-play. Each year is comprised of its own CSV file. Below is the python code reading the files in the directory:



```
#get list of csv files from directory

csvlist = glob.glob("*.csv")
print(csvlist)
```

```
['pbp-2014.csv', 'pbp-2015.csv', 'pbp-2016.csv', 'pbp-2017.csv', 'pbp-2018.csv', 'pbp-2019.csv', 'pbp-2020.csv', 'pbp-2021.cs
v']
```

We’ve attached a sample of the data so we can view it using python:



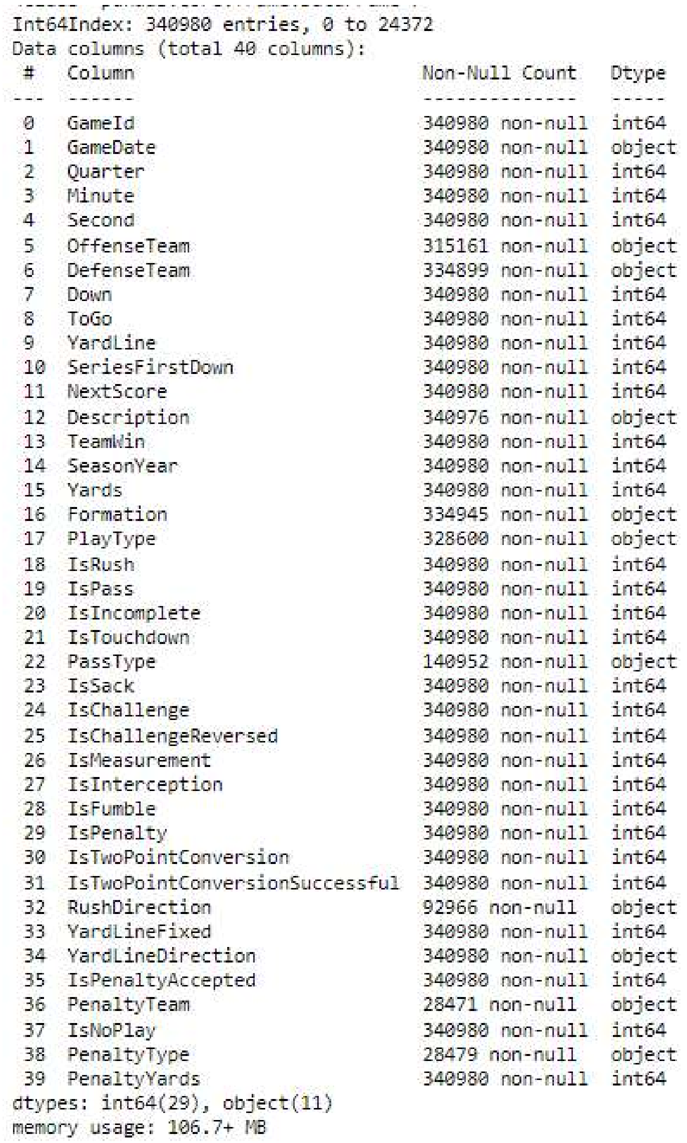
```
#combine csv files into one dataframe

total = 0

tmp = pd.DataFrame()

for x in csvlist:
    df = pd.read_csv(x)
    tmp = tmp.append(df)
    total += len(df)
    #realresults.append(df)

print(repr(total) + " - Total Rows")

display(tmp)
```

```
340980 - Total Rows
```

	GameId	GameDate	Quarter	Minute	Second	OffenseTeam	DefenseTeam	Down	ToGo	YardLine	...	IsTwoPointConversion	IsTwoPointConversionS
0	2014113007	2014-11-30	4	2	0	LA	LV	4	3	50	...	0	
1	2014113007	2014-11-30	4	2	55	LA	LV	2	3	50	...	0	
2	2014113007	2014-11-30	4	3	39	LA	LV	1	10	43	...	0	
3	2014113007	2014-11-30	4	5	24	LA	LV	0	0	35	...	0	
4	2014113007	2014-11-30	4	7	31	LA	LV	4	6	40	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
24368	2021091200	2021-09-12	2	0	9	PHI	ATL	1	9	91	...	0	
24369	2021091200	2021-09-12	2	0	15	NaN	NaN	0	0	100	...	0	
24370	2021091200	2021-09-12	2	2	0	NaN	NaN	0	0	100	...	0	
24371	2021091200	2021-09-12	1	0	9	ATL	PHI	1	7	93	...	0	
24372	2021091200	2021-09-12	1	0	0	NaN	NaN	0	0	100	...	0	

340980 rows × 45 columns

The data for each file contains 45 different columns. Below we use Python .info to gain more insight about the data.

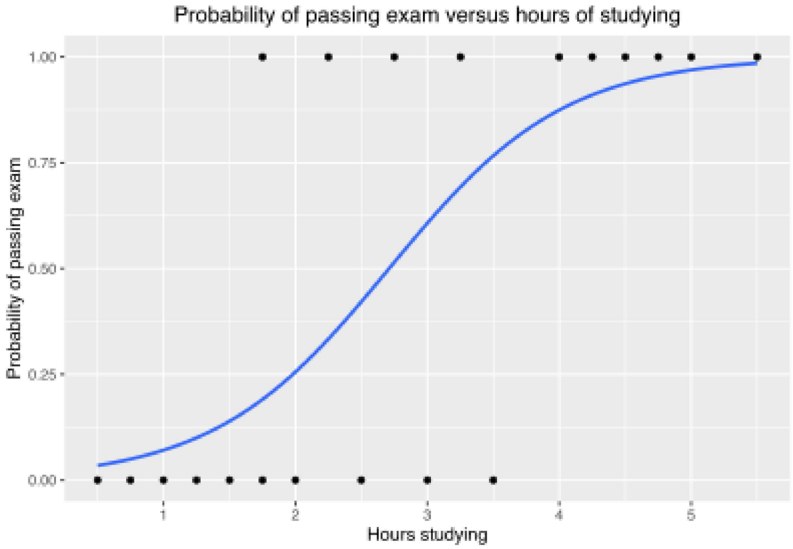


```
Int64Index: 340980 entries, 0 to 24372
Data columns (total 40 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   GameId                          340980 non-null  int64
 1   GameDate                        340980 non-null  object
 2   Quarter                         340980 non-null  int64
 3   Minute                          340980 non-null  int64
 4   Second                          340980 non-null  int64
 5   OffenseTeam                     315161 non-null  object
 6   DefenseTeam                     334899 non-null  object
 7   Down                            340980 non-null  int64
 8   ToGo                            340980 non-null  int64
 9   YardLine                        340980 non-null  int64
 10  SeriesFirstDown                 340980 non-null  int64
 11  NextScore                       340980 non-null  int64
 12  Description                     340976 non-null  object
 13  TeamWin                         340980 non-null  int64
 14  SeasonYear                      340980 non-null  int64
 15  Yards                           340980 non-null  int64
 16  Formation                       334945 non-null  object
 17  PlayType                        328600 non-null  object
 18  IsRush                          340980 non-null  int64
 19  IsPass                          340980 non-null  int64
 20  IsIncomplete                    340980 non-null  int64
 21  IsTouchdown                     340980 non-null  int64
 22  PassType                        140952 non-null  object
 23  IsSack                          340980 non-null  int64
 24  IsChallenge                     340980 non-null  int64
 25  IsChallengeReversed             340980 non-null  int64
 26  IsMeasurement                   340980 non-null  int64
 27  IsInterception                  340980 non-null  int64
 28  IsFumble                        340980 non-null  int64
 29  IsPenalty                       340980 non-null  int64
 30  IsTwoPointConversion            340980 non-null  int64
 31  IsTwoPointConversionSuccessful  340980 non-null  int64
 32  RushDirection                   92966 non-null   object
 33  YardLineFixed                   340980 non-null  int64
 34  YardLineDirection               340980 non-null  object
 35  IsPenaltyAccepted               340980 non-null  int64
 36  PenaltyTeam                     28471 non-null   object
 37  IsNoPlay                        340980 non-null  int64
 38  PenaltyType                     28479 non-null   object
 39  PenaltyYards                    340980 non-null  int64
dtypes: int64(29), object(11)
memory usage: 106.7+ MB
```

For our feature set we will use a combination of these columns to get to our expected outcome. We'll also be combining different columns to and training the model with the newly created feature. For example: we can combine PenaltyYards and subtract out the Yards column to get to an overall Net_Yards_Per_Game and determine if that gets better results. Also, we can take the average yards per throw and combine that with interceptions to determine if that increases the possibility for a team to win a game. As we go through the project we will do a bunch of data augmentation, combination, and filtering running through the models to determine which gives us the best results.

Expected Outcomes:

Our expected outcomes will be a logistic regression problem. We want to use varying features in different logistic regression models such as decision tree, random forest, SVM, neural networks, Bayesian Inferencing, and others as we go through the project. We want to go game by game and find the winners of the game and what stat is most important in winning games. So we can determine in the future, if a team achieves a certain set of stats they are more likely to win the game. The graph below is a simplified version of what we expect in our outcome.



References:

1 https://arxiv.org/pdf/1310.6998.pdf

2 https://www.academia.edu/43912415/IJERT_An_Applicationhttps:/www.jair.org/index.php/jair/article/view/13509/26786_of_Linear_Regression_and_Artificial_Neural_Network_Model_in_the_NFL_Result_Prediction?auto=citations&from=cover_page

3 https://www.jair.org/index.php/jair/article/view/13509/26786

4 https://eprints.soton.ac.uk/446078/1/NFL_ML_IJCSS.pdf