CSCE 5222: FEATURE ENGINEERING
Project Increment 1

NFL PredX 2022

By
Salim Jivani, Ali Tahririan, Aniv Chakravarty, Satish Bitra

Project 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 NFL 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.

Related Work

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 utilising 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 to combine, filter, augment, standardise 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 features and model architecture. Try a bunch of combinations to determine the best model.

Dataset (Salim)

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

Features

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.

Here's one of the datasets we've created from the raw data:

	GameId	OffenseTeam	Score	PassingYards	RunningYards	PenaltyYards	Penalties	Turnovers	Win_Lose
1	2014090400	SEA	34	191	210	55	7	1	Win
0	2014090400	GB	18	210	80	79	5	2	Lose
2	2014090700	ATL	37	448	123	47	6	1	Win
3	2014090700	NO	34	333	139	76	7	2	Lose
5	2014090701	CIN	23	301	81	34	4	0	Win
...	...	...	...	...	...	...	...	...	...
3807	2021110710	SF	17	333	39	55	5	3	Lose
3808	2021110711	LA	28	328	94	47	8	2	Win
3809	2021110711	TEN	22	149	70	87	9	1	Lose
3811	2021110800	PIT	29	205	113	94	10	1	Win
3810	2021110800	CHI	27	291	99	51	8	2	Lose

3812 rows × 9 columns

We'll continue to transform the data and put them into the model for evaluation.

Implementation

We are mainly using google colab and python to perform our analysis. Python provides various libraries in order to manipulate, store and visualize the data. We make use of Pandas for our initial data exploration and pre-processing followed by the use of Scikit learn to train models. We first run the models on all the features available on the dataset and obtain scores that would act as a baseline. We then proceed with analysis of various feature combinations and methods to determine the effects of certain features on the overall accuracy of the various models. In order to streamline the process the model training and predictions are put in methods for easy execution.

Preliminary Results and Analysis

For phase 1, we've been collecting, examining, and transforming the dataset into the way we need it. We were able to separate out stats by teams who have won and lost the game by different metrics, and done some preliminary discovery.

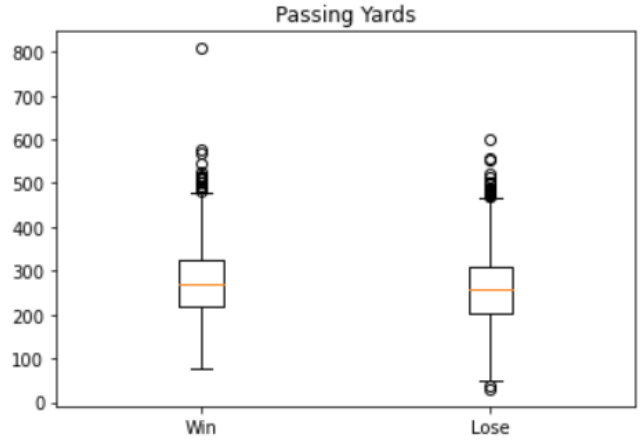
Losing Teams:

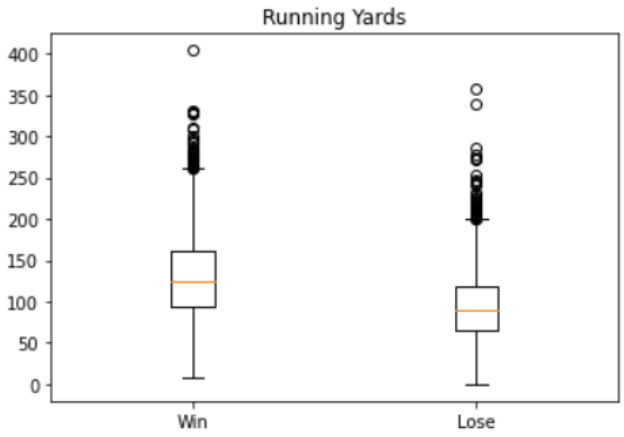
	GameId	Score	PassingYards	RunningYards	PenaltyYards	Penalties	Turnovers
count	1.906000e+03	1906.000000	1906.000000	1906.000000	1906.000000	1906.000000	1906.000000
mean	2.017394e+09	18.013641	259.719307	94.645331	53.144281	6.378279	2.486359
std	2.190359e+06	8.235562	80.251183	43.054798	26.818113	2.840304	1.552941
min	2.014090e+09	0.000000	30.000000	0.000000	0.000000	0.000000	0.000000
25%	2.015122e+09	13.000000	204.000000	65.000000	34.000000	4.000000	1.000000
50%	2.017120e+09	17.000000	256.000000	89.500000	50.000000	6.000000	2.000000
75%	2.019112e+09	23.000000	310.000000	119.000000	70.000000	8.000000	3.000000
max	2.021111e+09	48.000000	600.000000	358.000000	169.000000	19.000000	9.000000

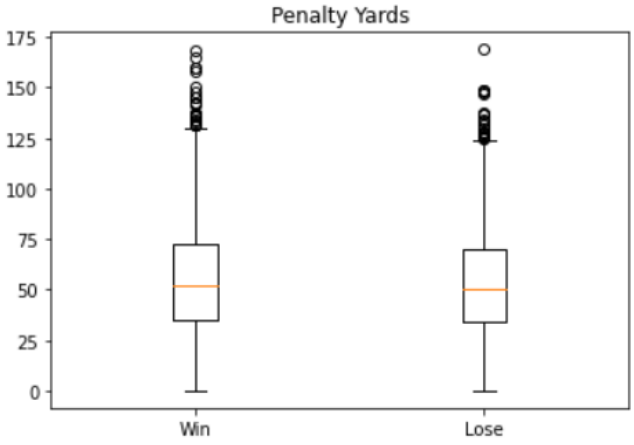
Winning Teams:

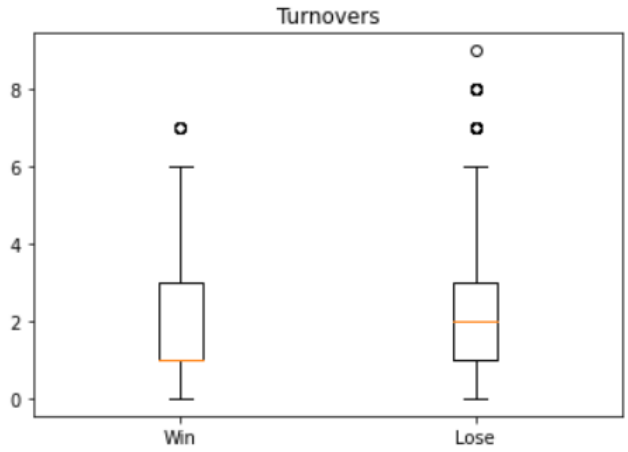
	GameId	Score	PassingYards	RunningYards	PenaltyYards	Penalties	Turnovers
count	1.906000e+03	1906.000000	1906.000000	1906.000000	1906.000000	1906.000000	1906.000000
mean	2.017394e+09	29.061910	272.967996	130.395068	55.615425	6.697272	1.734523
std	2.190359e+06	8.527451	78.194679	52.563250	27.429079	2.839016	1.353845
min	2.014090e+09	6.000000	75.000000	7.000000	0.000000	0.000000	0.000000
25%	2.015122e+09	23.000000	218.250000	94.000000	35.000000	5.000000	1.000000
50%	2.017120e+09	28.000000	269.000000	124.000000	52.000000	6.000000	1.000000
75%	2.019112e+09	35.000000	323.000000	161.000000	73.000000	9.000000	3.000000
max	2.021111e+09	63.000000	808.000000	404.000000	168.000000	18.000000	7.000000

Box Plots with the data comparing Win vs Lose









Project Management

The goal of increment 1 is to have the models prepared and the dataset preprocessed in order to obtain the preliminary baseline results. We then proceed on increment 2 of further analysis of the various features.

The team has weekly meetings on discord which is a platform that allows for text and video chat as well as file sharing.

We also have a github repository where all our code, documentation and results are updated and maintained.

Roles and Responsibilities:

- Salim : Team lead, data preparation/ exploration, documentation and presentation
- Ali: Model development of K-nearest neighbour and Random Forest, documentation and presentation
- Aniv: Model development of Naive Bayes and Decision Trees, documentation and presentation
- Satish: Model development of Linear Regression, Logistic Regression and Support Vector Machines, documentation and presentation

Implementation Status:

- Data pre-processing and Exploration (Salim): Preliminary data has been transformed and ready for first pass into the models
- Naive Bayes and Decision Tree models (Aniv): Models are completed and ready for execution by the main method in order to obtain results
- KNN and Random Forest (Ali):
- Linear Regression, Logistic Regression and SVM (Satish): Models are completed and ready for execution

Report: Overall we meet our initial goals for increment 1

Work Completed:

- Data pre processing
- Machine Learning models

Contributions:

- Salim : Collecting Dataset and Preliminary visualisations.
- Ali: KNN and random forest are my part to see the predictions.
- Aniv: Did preliminary literature survey and went over the papers to see viability with regards to the project. Built the models for Naive Bayes and Decision Trees, worked on creating editing and formatting increment 1 report.
- Satish: Implemented models for SVM and linear regression and logistic regression along with obtaining various output results for the models and verified the models by testing with different datasets.

Work to be Completed:

- Analysis of models on data(Salim)
- Feature analysis(Satish)
- New feature combination and techniques(Ali)
- Results and analysis(Aniv)

Issues and Concerns:

So far we have built the models and main concerns would be related to the number of features to consider in order to obtain decent results.

References/Bibliography

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