Members: Charlie Morris (just me)

Introduction

Having always been peripherally interested in sports analytics (by way of sports betting), I attended Michael Lopez's lecture on September 18th, 2023. He is the Senior Director of Football Data and Analytics in the NFL, and the event was hosted by the Reese T. Prosser Mathematics Lecture Series.

The title of the talk was "Analyzing NFL Data is challenging, but player tracking data is here to help." Mr. Lopez first spoke of his involvement working for the NFL and emphasized how sports analytics is truly hard since all the data is observational (or in other words, it is very difficult to estimate the effects of treatments due to the constant presence of confounding variables).

He then discussed the analytics surrounding NFL teams going for it on 4th down. The point he made was that the theory of "always going for it on 4th down" lacked the nuance of exactly how close teams are to a first down. Play-by-play NFL stats typically do not differentiate 4th and inches vs. 4th and 1.5 yards (both of them are labeled as 4th and 1). Only by virtue of player tracking data can this difference be truly seen.

All in all, I loved the presentation, and it inspired me to make football analytics my final project. After Mr. Lopez's talk, I asked him what would be some good football data sets to analyze, and he told me about the Kaggle competitions that the NFL hosts. These competitions provide datasets from AWS's Next Gen Football Stats.

Soon after the talk, I formed a team including me and 2 other Dartmouth students (Parker Seegmiller and Zachary Gottesman) to participate in the 2024 NFL Kaggle competition (`https://www.kaggle.com/competitions/nfl-big-data-bowl-2024`), where the goal is to use deep learning to evaluate tackling tactics and strategy. Only a little bit of work has gone into it so far, but I will include our initial code as an attachment to the project.

Getting back to this project though, I've created something that will have hopefully given me the skills to better contribute to my Databowl competition. In my code, I apply what we learned in Cosc 89 to 4 smaller sections that I outline. I practice data visualization, modeling, and scraping. Personally, as someone without a ton of Pandas experience before the Fall term, I'm quite proud of the diversity of what I produce.

Part 1: Betting + API

Having never used an API before on my own, I first set out to learn how to extract football related data using an API. As an avid sports better, I chose to specifically look at NFL sports betting odds.

I found a cool API that helped me do this, made a free account, and then successfully learned to call a GET request. My request looked only at winning odds for upcoming NFL games.

By way of some manipulation in JSON, I successfully print out moneyline betting odds for the upcoming football games. Here's a little snippet:

```
                New York Jets: +112
Game: Buffalo Bills vs. Denver Broncos
        Fanduel odds (Last update: 2023-11-12 19:40:50 EST
                Buffalo Bills: -333
                Denver Broncos: +270
        DraftKings odds (Last update: 2023-11-12 19:41:21 EST)
                Buffalo Bills: -323
                Denver Broncos: +260
Game: Baltimore Ravens vs. Cincinnati Bengals
        Fanduel odds (Last update: 2023-11-12 19:40:50 EST
                Baltimore Ravens: -189
                Cincinnati Bengals: +158
        DraftKings odds (Last update: 2023-11-12 19:41:21 EST)
                Baltimore Ravens: -179
                Cincinnati Bengals: +150
Game: Green Bay Packers vs. Los Angeles Chargers
        Fanduel odds (Last update: 2023-11-12 19:40:50 EST
                Green Bay Packers: +144
                Los Angeles Chargers: -172
```

I then make a dataframe of the moneyline odds just for FanDuel's betting app. Below is a snippet, where the columns are sorted by the biggest betting underdogs:

	gameID	team	Last Update	moneyline
14	8	Carolina Panthers	2023-11-12 19:40:50-05:00	410
25	13	Tampa Bay Buccaneers	2023-11-12 19:40:50-05:00	385
12	7	Las Vegas Raiders	2023-11-12 19:40:50-05:00	380
20	11	New York Giants	2023-11-12 19:40:50-05:00	360
22	12	Chicago Bears	2023-11-12 19:40:50-05:00	330
1	1	Washington Commanders	2023-11-12 19:36:28-05:00	320
5	3	Denver Broncos	2023-11-12 19:40:50-05:00	270
27	14	New York Jets	2023-11-12 19:40:50-05:00	250
11	6	Tennessee Titans	2023-11-12 19:40:50-05:00	220
16	9	Arizona Cardinals	2023-11-12 19:40:50-05:00	200
19	10	Pittsburgh Steelers	2023-11-12 19:40:50-05:00	180
7	4	Cincinnati Bengals	2023-11-12 19:40:50-05:00	158
8	5	Green Bay Packers	2023-11-12 19:40:50-05:00	144
28	15	Los Angeles Rams	2023-11-12 19:40:50-05:00	122
33	17	Philadelphia Eagles	2023-11-12 19:40:50-05:00	114
31	16	Minnesota Vikings	2023-11-12 19:40:50-05:00	102
2	2	Las Vegas Raiders	2023-11-12 19:40:50-05:00	-106
3	2	New York Jets	2023-11-12 19:40:50-05:00	-110
30	16	Denver Broncos	2023-11-12 19:40:50-05:00	-120
32	17	Kansas City Chiefs	2023-11-12 19:40:50-05:00	-133
29	15	Seattle Seahawks	2023-11-12 19:40:50-05:00	-145
9	5	Los Angeles Chargers	2023-11-12 19:40:50-05:00	-172

This project because I can actually use it each week to quickly see if I have better betting odds on either DraftKings of Fanduel.

Part 2: Webscraping + NextGenStats

Since webscraping was such an integral part of the first half of the class, I chose to webscrape player information from the NextGenStats website (`https://nextgenstats.nfl.com/stats/top-plays/fastest-ball-carriers`). Doing this was a little more complicated because I needed to learn how to use request headers and couldn't just read the data off of the html code. However, I was successful and I built dataframes for stats across the past 6 seasons. Below are snippets for the 4 dataframes I made: teams, receiving, rushing, and passing.

	abbr	cityState	conferenceAbbr	fullName	nick	season	stadiumName	teamId	teamSiteTicketUrl	
0	AFC	AFC Pro Bowl	AFC	AFC Pro Bowl Team	AFC Pro Bowl	2023	Allegiant Stadium	8600	None	
1	ARI	Arizona	NFC	Arizona Cardinals	Cardinals	2023	State Farm Stadium	3800	http://www.azcardinals.com/tickets/	http://www
2	ATL	Atlanta	NFC	Atlanta Falcons	Falcons	2023	Mercedes-Benz Stadium	0200	http://www.atlantafalcons.com/tickets/	http://www.at
3	BAL	Baltimore	AFC	Baltimore Ravens	Ravens	2023	M&T Bank Stadium	0325	http://www.baltimoreravens.com/tickets/	http://www.balt
4	BUF	Buffalo	AFC	Buffalo Bills	Bills	2023	Highmark Stadium	0610	http://www.buffalobills.com/tickets/	http://www
5	CAR	Carolina	NFC	Carolina Panthers	Panthers	2023	Bank of America Stadium	0750	http://www.panthers.com/tickets/	http://w
6	CHI	Chicago	NFC	Chicago Bears	Bears	2023	Soldier Field	0810	http://www.chicagobears.com/tickets/	http://www.c
7	CIN	Cincinnati	AFC	Cincinnati Bengals	Bengals	2023	Paycor Stadium	0920	http://www.bengals.com/tickets/	http://w
8	CLE	Cleveland	AFC	Cleveland Browns	Browns	2023	Cleveland Browns Stadium	1050	http://www.clevelandbrowns.com/tickets/	http://www.cleve
9	DAL	Dallas	NFC	Dallas Cowboys	Cowboys	2023	AT&T Stadium	1200	http://www.dallascowboys.com/tickets/index.html	http://www.da
10	DEN	Denver	AFC	Denver Broncos	Broncos	2023	Empower Field at Mile High	1400	http://www.denverbroncos.com/ticketOffice	http://www.de
11	DET	Detroit	NFC	Detroit Lions	Lions	2023	Ford Field	1540	http://www.detroitlions.com/tickets/index.html	http://www

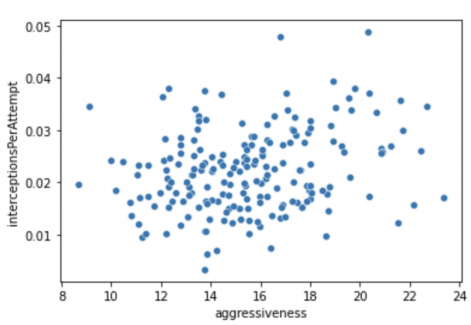
aboveExpectation	catchPercentage	percentShareOfIntendedAirYards	recTouchdowns	receptions	targets	yards	playerName	position	teamId	playerID	season
0.523604	55.769231	13.654206	5	29	52	402	Josh Reynolds	WR	2510	44930	2018
0.254860	55.405405	21.423041	4	41	74	774	DeSean Jackson	WR	4900	33130	2018
0.458069	66.071429	23.001960	1	37	56	466	Taywan Taylor	WR	2100	44884	2018
-0.269350	72.043011	23.005470	2	67	93	688	Taylor Gabriel	WR	0810	42016	2018
0.385912	60.000000	19.152271	5	39	65	494	Curtis Samuel	WR	0750	44852	2018
...	...	...	...	...	...	...	...	...	...	...	...
0.896506	57.142857	16.408709	2	20	35	258	Odell Beckham	WR	0325	41238	2023
1.531928	73.469388	14.292806	1	36	49	418	Trey McBride	TE	3800	54520	2023
-0.052894	60.000000	10.122767	0	15	25	158	DeVante Parker	WR	3200	42357	2023
0.266766	53.571429	8.429188	1	15	28	102	Dawson Knox	TE	0610	47879	2023
0.960697	68.571429	20.560462	3	24	35	412	Josh Reynolds	WR	1540	44930	2023

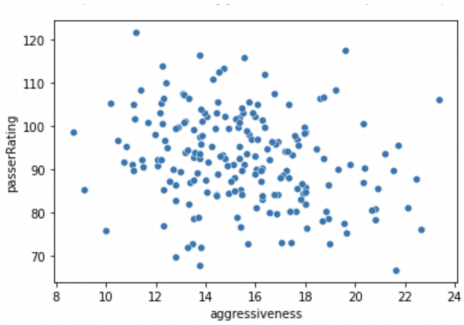
hYardsOverExpected	rushYardsOverExpectedPerAtt	teamId	efficiency	percentAttemptsGteEightDefenders	avgRushYards	playerID	playerName	position	season
37.316278	0.270408	3000	3.756298	27.857143	4.128571	40129	Latavius Murray	RB	2018
-64.078076	-0.413407	2100	4.576228	25.806452	3.335484	37224	Dion Lewis	RB	2018
-53.326871	-0.467780	0610	4.611351	20.869565	3.347826	36052	Chris Ivory	RB	2018
-41.525111	-0.477300	3700	3.872555	6.896552	4.183908	43442	Wendell Smallwood	RB	2018
192.420589	1.028987	1050	3.478876	33.854167	5.187500	46104	Nick Chubb	RB	2018
...	...	...	...	...	...	...	...	...	...
17.243789	0.183445	0750	3.572536	17.021277	3.734043	53555	Chuba Hubbard	RB	2023
-18.105746	-0.306877	0325	4.137752	5.000000	4.300000	47896	Justice Hill	RB	2023
-7.686715	-0.048344	2250	3.808204	24.375000	3.862500	53454	Travis Etienne	RB	2023
95.922344	0.786249	0200	4.017974	25.600000	4.896000	55872	Bijan Robinson	RB	2023
53.976263	0.399824	3410	3.680669	17.985612	4.086331	46071	Saquon Barkley	RB	2023

:ionPercentageAboveExpectation	...	maxCompletedAirDistance	passTouchdowns	passYards	passerRating	playerName	season	seasonType	position	teamId	p
0.529207	...	52.863115	34	5130	96.466049	Ben Roethlisberger	2018	REG	QB	3900	
4.202118	...	53.715015	17	2366	100.423442	Ryan Fitzpatrick	2018	REG	QB	4900	
-1.041831	...	61.410319	25	4442	97.574679	Aaron Rodgers	2018	REG	QB	1800	
-1.562972	...	49.899188	29	4355	97.660819	Tom Brady	2018	REG	QB	3200	
2.276724	...	48.220415	22	3885	96.863118	Dak Prescott	2018	REG	QB	1200	
...	...	...	...	...	...	...	...	...	...	...	
3.197938	...	52.736638	10	2177	98.052536	Lamar Jackson	2023	REG	QB	0325	
1.733906	...	57.006754	14	2507	99.054192	Jared Goff	2023	REG	QB	1540	
-5.100817		52.550940	14	2009	80.541667	Jordan Love	2023	REG	QB	1800	

From these data frames, I then did some basic analysis about passing statistics. I looked first at quarterback aggressiveness. From the data I pulled, NextGenStats gives an aggressive percentage for each quarterback, detailing how aggressive their throws were. The official definition of is aggressiveness tracks the percentage of passing attempts a quarterback makes that are into tight coverage, where there is a defender within 1 yard or less of the receiver at the time of completion or incompletion.

I thought the statistic was interesting because aggressiveness could imply that a QB makes some spectacular plays in tight windows or that they take unnecessary risks. It turns out though that the latter is more true by virtue of some correlations I run and the below scatterplots. Interception rates are correlated with high aggressive and passer ratings and inverstly correlated with aggressiveness.



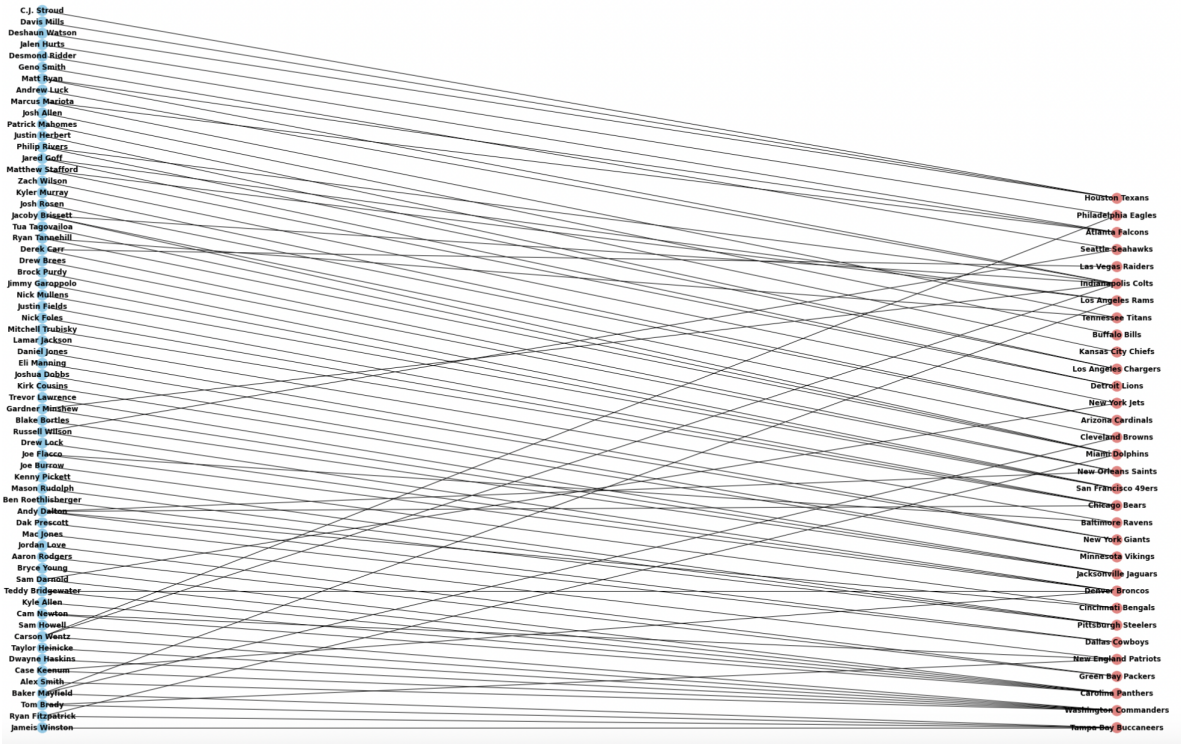


I then use passer ratings to find some of the best and worst regular season quarterback play. Below are snippets of the result. Not surprisingly, New York Jets QBs make up 4 of the worst 14 quarterback season stats (among qbs with a minimum of 200 passing attempts.

	playerName	season	team	passerRating	passTouchdowns	interceptions
14	Aaron Rodgers	2020	Green Bay Packers	121.530418	48	5
137	Ryan Tannehill	2019	Tennessee Titans	117.497086	22	6
92	Drew Brees	2019	New Orleans Saints	116.269841	27	4
91	Drew Brees	2018	New Orleans Saints	115.682515	32	5
141	Patrick Mahomes	2018	Kansas City Chiefs	113.843391	50	12
81	Lamar Jackson	2019	Baltimore Ravens	113.336451	36	6
183	Deshaun Watson	2020	Houston Texans	112.400429	33	7
188	Aaron Rodgers	2021	NFC Pro Bowl Team	111.899718	37	4
164	Russell Wilson	2018	Seattle Seahawks	110.865535	35	7
90	Brock Purdy	2023	San Francisco 49ers	109.900000	15	5
47	Joe Burrow	2021	Cincinnati Bengals	108.261218	34	14
143	Patrick Mahomes	2020	Kansas City Chiefs	108.234127	38	6

	playerName	season	team	passerRating	passTouchdowns	interceptions
104	Josh Rosen	2018	Arizona Cardinals	66.735581	11	14
113	Josh Allen	2018	Buffalo Bills	67.890625	10	12
122	Zach Wilson	2021	New York Jets	69.685596	9	11
72	Trevor Lawrence	2021	Jacksonville Jaguars	71.857697	12	17
42	Sam Darnold	2021	Carolina Panthers	71.941708	9	13
121	Sam Darnold	2020	New York Jets	72.687729	9	11
123	Zach Wilson	2022	New York Jets	72.813361	6	7
177	Carson Wentz	2020	Philadelphia Eagles	72.830854	16	15
33	Dwayne Haskins	2020	Washington Commanders	72.951245	5	7
66	Justin Fields	2021	Chicago Bears	73.225309	7	10
58	Drew Lock	2020	Denver Broncos	75.380926	16	15
43	Bryce Young	2023	Carolina Panthers	75.933908	8	7
32	Dwayne Haskins	2019	Washington Commanders	76.077586	7	7
124	Zach Wilson	2023	New York Jets	76.331227	5	5
4	Kenny Pickett	2022	Pittsburgh Steelers	76.676307	7	9

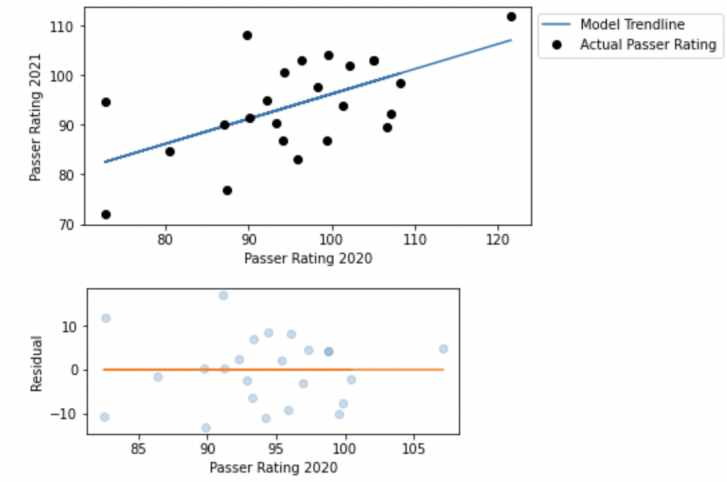
Because we touched on networks at the end of the course, I then created a bipartite graph (undirected) to represent player and team relationships of quarterbacks.



The network also allows me to see that the Washington Commanders and Indianapolis Colts are tied for the most number of distinct starting quarterbacks in the past 6 seasons (where qbs are required to have a minimum number of passing attempts). Each node has a degree of 6.

Finally, I use linear regression to explore if past quarterback success implies future success. By an arbitrary decision, I look at individual quarterback passer ratings in the 2020 season vs. the 2021 season. I find there is a positive correlation seen by an R squared value of 0.293. I provide a graph showing the line of best fit over the actual passer ratings of

individual players. Confirming the reliability of this autoregressive model, the residual plot below strongly demonstrates the normality of errors in the predictiveness of the model.



Part 3: Draft Profiles + Text Analysis

Because we've used text so much in this class, I wanted to figure out how to incorporate text analysis into my final project. The idea struck me then look at NFL Draft Profiles. I found on the Internet a great GitHub source that had 2 cool related dataframes. Here's the link: `https://github.com/nflverse/open-source-football/tree/master/_posts/2023-07-14-linguistic-analysis-of-nfl-prospect-reports`

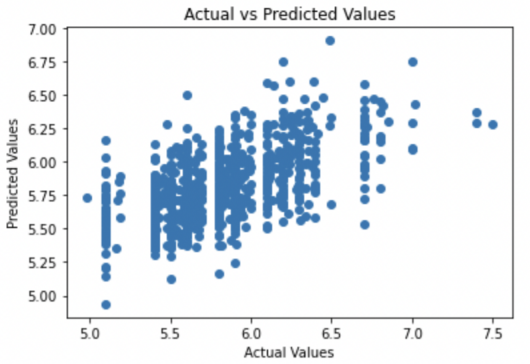
My first task was to see how well pre-draft short text profiles of NFL players would reflect their given player grades. My expectation going in though was that it would not be a great predictor though because I assumed that combine and college stats were more relevant information.

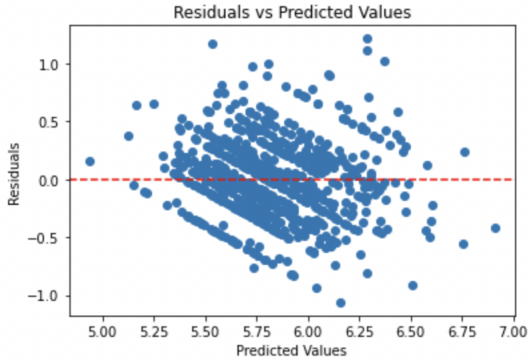
Nonetheless, I took a crack at the problem. I initially created a feature that measured the log word length of a draft profile (b/c longer profiles are correlated with higher grades) and performed sentiment analysis over each text.

I then used TF-IDF for generating features from the text profiles. Before doing this though, I tokenized text, remove stop words, and lemmatize text. To test my models, I split the data in training and testing data frames when doing this.

I used a Ridge regression initially to make predictions on player grades. Below are the results:

```
Optimal Alpha from Cross-Validation: 0.1
```






```
Root Mean Squared Error: 0.3264245182580533
R^2 score: 0.334398769820372
```

In comparison, my using the median value for prediction gives a RMSE of 0.4. Clearly then, my model was a success in some regards then.

I didn't stop there though. Afterwards I tried predicting draft pick rounds of players based just on their short text profiles. I had less data to do this though because my dataset did not include draft profiles and eventual draft selections for all its players. Nonetheless, using a logistic regression that handles multi-class classifications (rounds 1 through 7), I try to predict what rounds players are drafted. I also use logistic regression to try and predict if a player was drafted in the first 3 rounds or not. Below are my results.

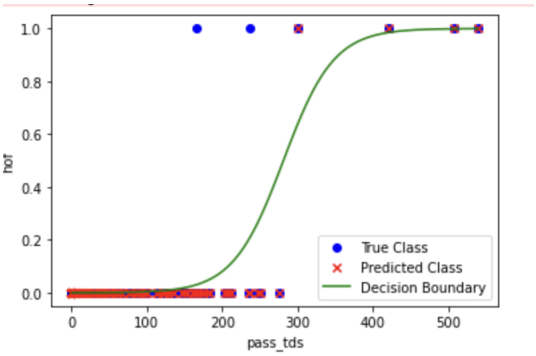
```
Confusion Matrix:
[[31  7 10  5  3  5  1]
 [28  2 13  1 10  5  4]
 [ 4  8 26 10 11  6  4]
 [ 9  1 21 11 14 13  4]
 [ 6  2 16  7  9  7 13]
 [ 7  3 16 11 15 12  8]
 [ 7  1 16  8 11 16  3]]

Macro F1 Score: 0.18195249281585815

Accuracy: 0.2039045553145336
```

```
Confusion Matrix:
[[230  37]
 [114  80]]

Macro F1 Score: 0.6336668052478409

Accuracy: 0.6724511930585684
```

Comparatively to the associated dummy classifiers, my models are actually not horrible. Both the Macro F1 and accuracy scores are higher in my models than either the "most frequent" or "stratified" dummy classifiers.

Lastly, I was curious about how hall of fame status for quarterbacks relates to total passing touchdowns. While looking at only quarterbacks drafted before 2000 (and are in my limited

data set), that way they are no longer eligible for the HOF, I modeled a logistic regression to predict Hall of Fame status. Among my data points, there were 6 Hall of Famers, and 273 non Hall of Famers.



There were clearly 2 outliers with the extra far left one representing Troy Aikman. Below are the stats for the HOF quarterbacks comparatively:

	season	pfr_player_name	pass_ints	pass_yards	pass_tds	allpro	probowls	seasons_started
3007	1989	Troy Aikman	141.0	32942.0	165.0	0	6	12
1012	1983	Jim Kelly	175.0	35467.0	237.0	1	5	11
999	1983	John Elway	226.0	51475.0	300.0	0	9	16
1025	1983	Dan Marino	252.0	61361.0	420.0	3	9	16
3705	1991	Brett Favre	336.0	71838.0	508.0	3	11	19
5532	1998	Peyton Manning	251.0	71940.0	539.0	7	14	17

Not surprisingly, it seems like Troy Aikman should not be in the Hall of Fame, and many fans agree. Yes, Troy Aikman won three Super Bowls, winning MVP in one of those games, and this is a driving factor for why he's in the Canton, OH. However, one has to take into account that he also had Emmitt Smith, Michael Irvin, and other stars on the team with him, and that he never lead the league in a major statistical category (except completion percentage once).

Part 4: Past Data Bowl

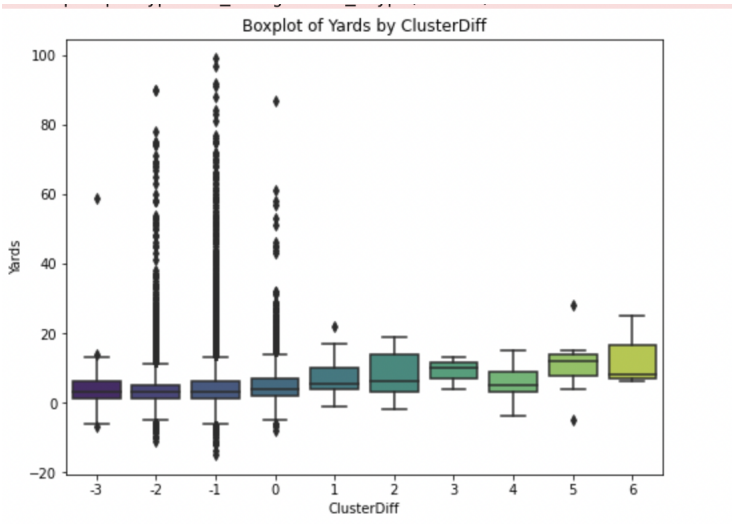
In looking at past NFL Kaggle Data Bowls, I was really intrigued by the one about predicting yardage on running plays. The dataset gives you information about the formations and player positions at the time of handoff. Each play has 22 rows associated with it (one for each player on the field).

I chose to tackle the problem by creating a binary classification model predicting whether a runner will gain 4 or more yards. My first steps after broadly exploring the data included data cleaning and filling in Null values. One of the cooler things I did here was transform the string representing how many line backers, defensive lineman, and defensive backs are on the field, into separate numeric columns.

I then try to create a relevant feature from all the X, Y coordinate positions of the players. My strategy was to use DBSCAN (an unsupervised clustering algorithm) to find how many

more defensive players were in a nearby cluster with the running than offensive players. I did this for both at the time of handoff and 0.5 seconds later (using the orientation, acceleration, and velocity of players to predict future positions). As expected, the clusters with higher number of defenders near the running back (ClusterDiff = supporting offensive players - defensive players), often entail fewer rushing yards. Below is some data related to this (where ClusterDiff is calculated at the time of handoff):

	ClusterDiff	Mean	Median	StandardDeviation	Count
0	-3	4.541667	3.0	7.589722	72
1	-2	3.487128	3.0	5.604272	10488
2	-1	4.532323	3.0	6.820292	18609
3	0	5.265006	4.0	6.644855	1766
7	4	5.666667	5.0	6.470446	6
4	1	6.477273	5.5	4.872746	44
5	2	8.000000	6.0	8.455767	5
9	6	13.000000	8.0	10.440307	3
6	3	9.000000	10.0	4.582576	3
8	5	11.000000	12.0	8.173127	11



I found though that not that much new information is gained from the clustering of players 0.5 seconds later, so I don't use the results.

The other features I use are as follows (note that for string categories I use one-hot encoding to make them numeric): running back speed, running back acceleration, down, number of defensive lineman, number of linebackers, number of defenders in the box, offensive formation, and distance to first down.

Next after splitting up the data frame into training and testing data, I get the following results from a logistic regression predicting 4+ yard run or not:

```
Confusion Matrix:
[[3303 1019]
 [2054 1376]]

AUC (Area Under the ROC Curve): 0.6444721763895616

Macro F1 Score: 0.5774774377695706

Accuracy: 0.6035861713106295
```

Comparatively, by using a stratified dummy classifier, we would get these results:

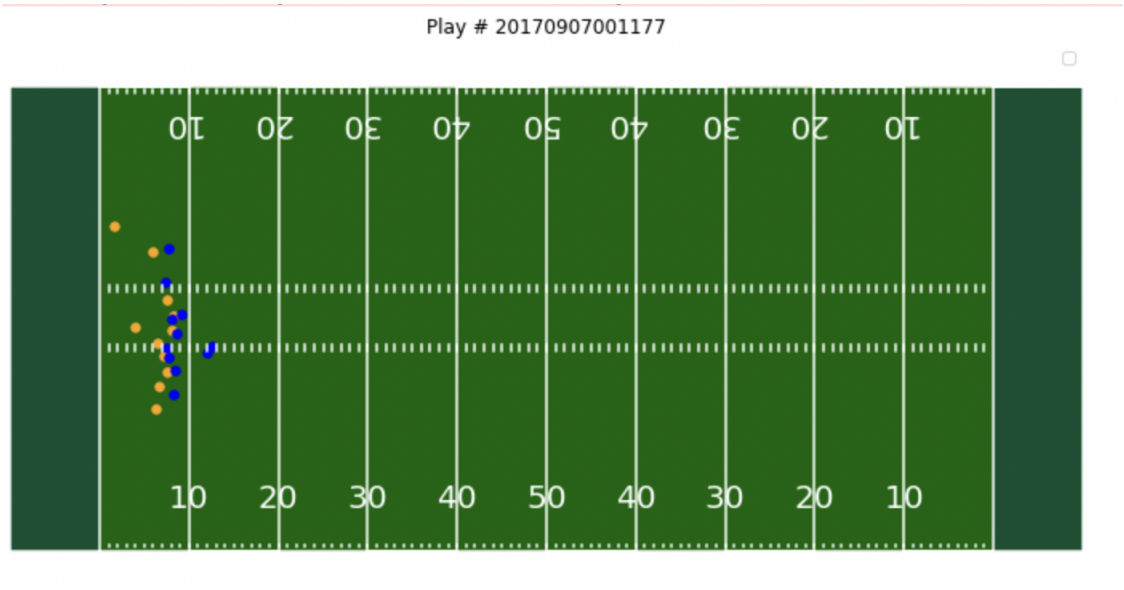
```
Confusion Matrix:
[[2535 1953]
 [1804 1460]]

AUC (Area Under the ROC Curve): 0.49628175133689845

Macro F1 Score: 0.5058481150975929

Accuracy: 0.5153508771929824
```

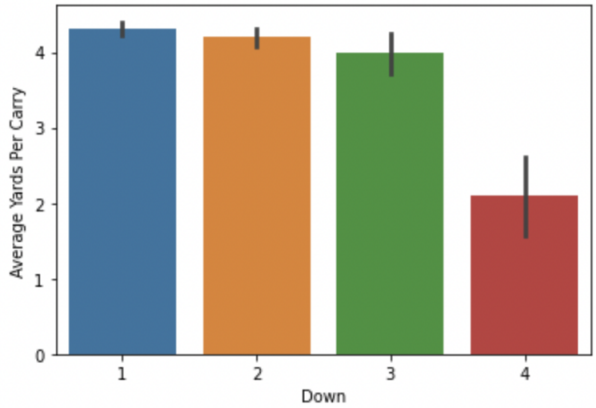
Clearly, my model (even though it is basic and simplifies a lot), can do much better then the dummy baseline.

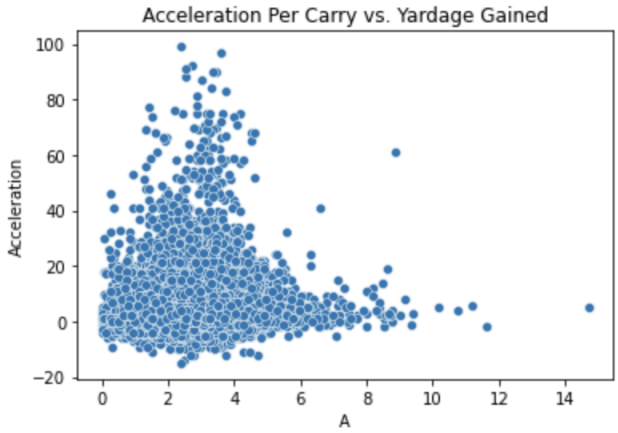
Using this model, I also test for who are the best running backs. I do this by finding which players have the greatest count of carries gaining 4+ yards even when my model predicts otherwise. As expected, the top players in the below list also happen to be the running back all pros and pro bowlers for the 2020 season.

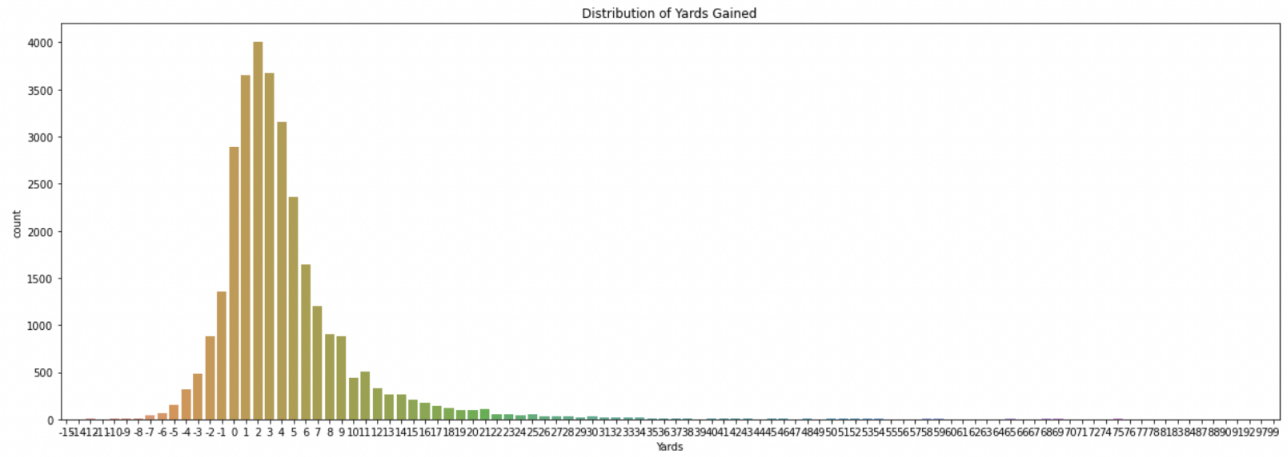
```
DisplayName          PlayerWeight  PlayerHeight
Ezekiel Elliott      228           6-0             229
Todd Gurley          224           6-1             185
Derrick Henry        247           6-3             172
Leonard Fournette    228           6-0             168
Mark Ingram          215           5-9             165
Carlos Hyde          229           6-0             163
Le'Veon Bell         225           6-1             152
Jordan Howard        224           6-0             151
Christian McCaffrey  205           5-11            151
Frank Gore           212           5-9             150
Adrian Peterson      220           6-1             145
Marlon Mack          210           6-0             142
Peyton Barber        225           5-11            139
LeSean McCoy         210           5-11            139
Melvin Gordon        215           6-1             133
Chris Carson         222           5-11            129
Latavius Murray      230           6-3             123
Joe Mixon            220           6-1             118
Lamar Miller         221           5-10            116
Kareem Hunt          216           5-11            115
dtype: int64
```

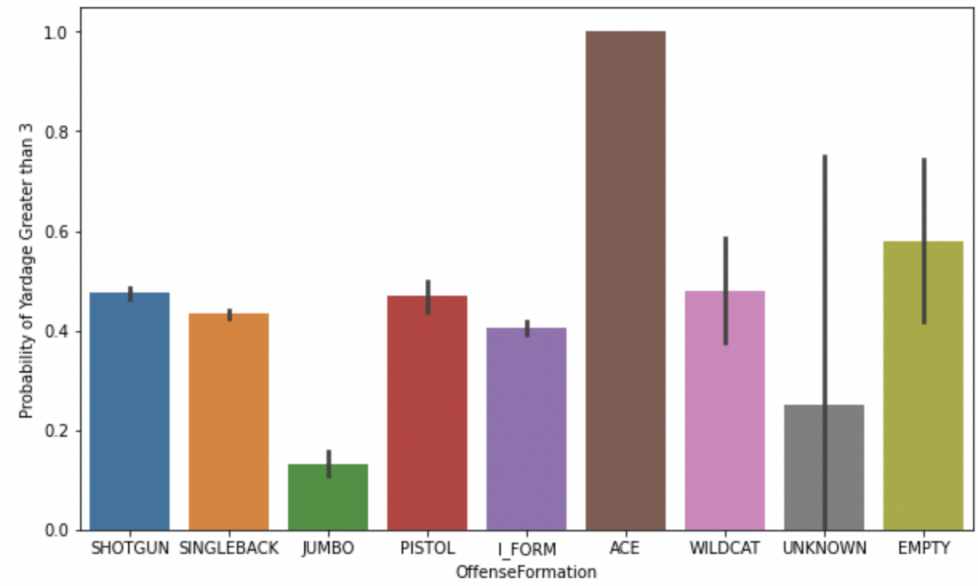
In addition, I create some cool visualizations as shown below.


Play # 20170907001177
10 20 30 40 50 40 30 20 10
10 20 30 40 50 40 30 20 10

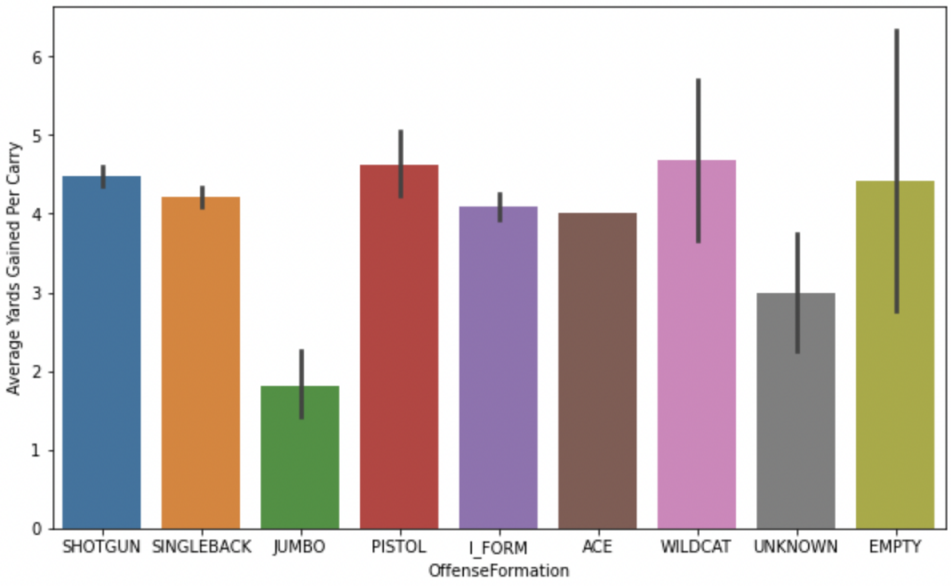

Average Yards Per Carry
0
1
2
3
4
Down
1
2
3
4


Acceleration Per Carry vs. Yardage Gained
Acceleration
100
80
60
40
20
0
−20
0
2
4
6
8
10
12
14
A





OffenseFormation	
SINGLEBACK	13624
SHOTGUN	9389
I_FORM	6225
PISTOL	979
JUMBO	677
WILDCAT	77
EMPTY	31
UNKNOWN	4
ACE	1



Part 5: Upcoming Data Bowl Project

As mentioned before, I've joined up with 2 students (outside of this class) to work on this year's Data Bowl challenge. While we've only done preliminary work, I'll just share the code as well in the zip file (I simply downloaded it from our shared google colab). My goal from this side project is that I learn more a bit about deep learning and how to actually apply it.

Conclusion

This project was a ton of fun and makes me want to get more into sports analytics. I feel like I got a taste into a lot of different things related to sports, betting, and data collection. Inspired by this class, I just bought my ticket to attend the 2024 Sloan Sports Analytics Conference at MIT in early March. I've also just recently started a messaging chat with the head of analytics at the New York Giants (my home team!) and plan to send him my final project.

Overall, I've really enjoyed the opportunity to take some time to think about data and its relation sports/sports betting!