ISyE 6740 – Summer 2021
Project Proposal

Team Member Names: Stone Hayden (903567377)

Project Title: Predicting 2015 Carolina Panthers’ Next Play Call

August 3rd, 2021

1. Background

The Carolina Panthers had the most successful season in the history of their franchise in the 2015-2016 season: going 15-1 enroute to a Super Bowl berth leading to a disappointing and surprising loss, at the hands of Peyton Manning and the Denver Broncos. In addition to being my favorite team, the Panthers were also one of the most unpredictable teams of the 2015 season, varying typical passing and running plays with inventive RPO's (Run-Pass Options) and designed runs for the QB (a player who typically would only pass).

2. Problem Statement

While defensive coordinators may not have been able to accurately predict the Carolina Panthers' next offensive play, we will attempt to apply some machine learning to get a likely prediction of what the next play of the Carolina Panthers will be – either a run or pass play, and if a pass play, what kind of pass. The pass plays are separated into 2 categories that separate the field: Short & Deep passes. Additionally, we will inspect the run/pass splits of other NFL teams against the Carolina Panthers. We will compare the accuracy of our model for each other NFL Team against the Panthers and expect the accuracy of the model would be lowest (or among the lowest) for the Panthers, as they were the most successful and unpredictable team of the 2015 season. We will also explore which aspects of the raw data most successfully predict whether the upcoming play-call is either a run or pass.

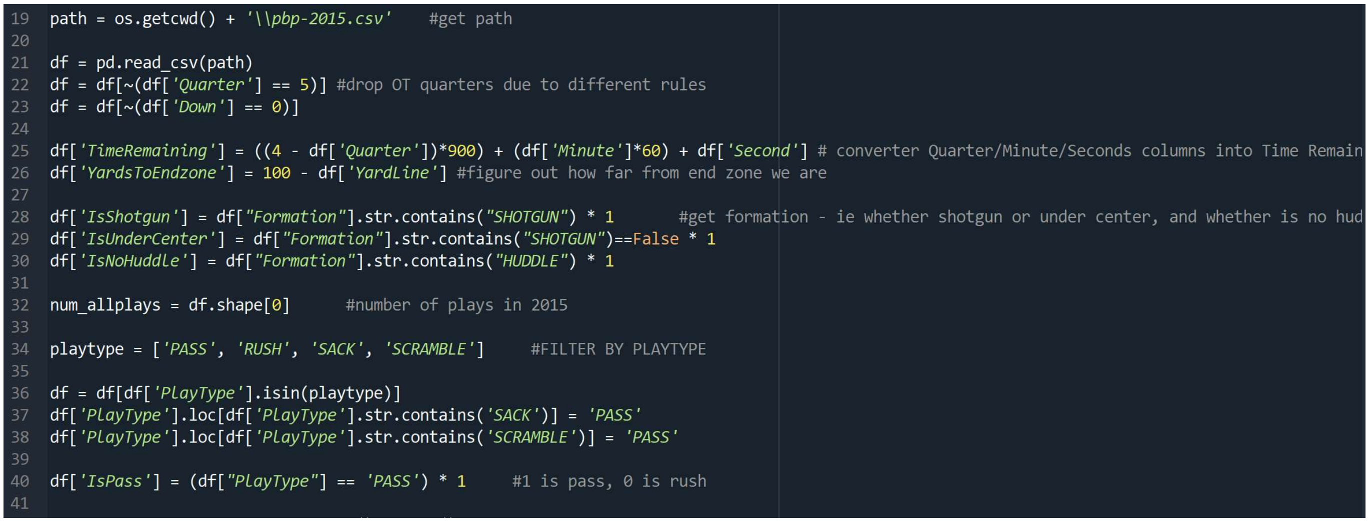
3. Data Source

We will be basing this model off the data provided by NFLSavant.com. NFLSavant.com provides play by play data for each NFL season dating back to 2013, as well as additional datasets such as combine measurements, historical gameday weather, and individual NFL player stats. We will specifically be using the 2015 Play-by-Play dataset. Source for this data can be found here: http://nflsavant.com/about.php

The resulting .csv file (pbp-2015.csv) has 45 columns and 46277 rows of data. The columns correspond to features of the dataset such as type of play, time in the game, teams playing, and many other useful (and not useful) features. The rows of data correspond to individual plays and their conditions and outcomes.

4. Methodology

To begin this project, we will have to start by cleaning the dataset. We will only be analyzing pass vs run plays for prediction, which means we will only be needing offensive plays. The dataset contains the column "PlayType" which describes the type of play performed. Beyond pass & rush, there are other descriptions in the series such as "QB Kneel", "Field Goal", "Punt", "No Play", "Extra Point", "Sack", "Scramble", "Two Point Conversion", and potentially other labels that will be discovered upon further examination of the dataset. We will need to remove and relabel some of these plays. For example, special teams plays such as "Field Goal", "Punt", and "Extra Point" will need to be removed. Additionally, plays blown dead ("No Play") can be removed as we don't know their outcome and plays occurring in the "5th Quarter" (AKA Overtime) can be dropped as this quarter has different rules than the others. Additionally, some labels such as "Sack" and “Scramble” will have to be relabeled as pass plays – as that was the play’s original design and intention. The code we used to clean up the data frame is shown below:



```
path = os.getcwd() + '\\pbp-2015.csv'    #get path

df = pd.read_csv(path)
df = df[~(df['Quarter'] == 5)] #drop OT quarters due to different rules
df = df[~(df['Down'] == 0)]

df['TimeRemaining'] = ((4 - df['Quarter'])*900) + (df['Minute']*60) + df['Second'] # converter Quarter/Minute/Seconds columns into Time Remain
df['YardsToEndzone'] = 100 - df['YardLine'] #figure out how far from end zone we are

df['IsShotgun'] = df["Formation"].str.contains("SHOTGUN") * 1       #get formation - ie whether shotgun or under center, and whether is no hud
df['IsUnderCenter'] = df["Formation"].str.contains("SHOTGUN")==False * 1
df['IsNoHuddle'] = df["Formation"].str.contains("HUDDLE") * 1

num_allplays = df.shape[0]      #number of plays in 2015

playtype = ['PASS', 'RUSH', 'SACK', 'SCRAMBLE']     #FILTER BY PLAYTYPE

df = df[df['PlayType'].isin(playtype)]
df['PlayType'].loc[df['PlayType'].str.contains('SACK')] = 'PASS'
df['PlayType'].loc[df['PlayType'].str.contains('SCRAMBLE')] = 'PASS'

df['IsPass'] = (df["PlayType"] == 'PASS') * 1     #1 is pass, 0 is rush

```

Figure 1: Code cleaning df

Now that we have the dataset in a clean manner, we can parse it down into the information we want. So, we will filter the dataset by “OffenseTeam” == ‘CAR’ to get only plays where the Carolina Panthers have the ball. We can take this data frame and compare some attributes of it to the NFL 2015 plays data frame as a whole. Below are some graphs and insights on the data frames:

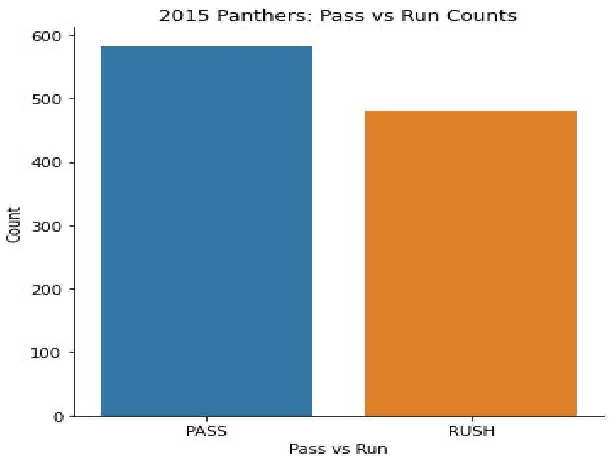


Figure 2: 2015 Panthers Pass/Run Split

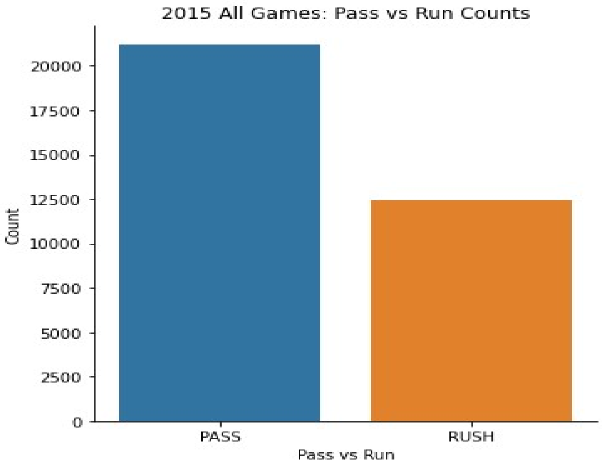


Figure 3: 2015 All NFL Pass/Run Split

As seen in the above, the 2015 Panthers had a much lower Pass vs Run percentage when compared to the rest of the NFL. This could be explained for 1 of 2 reasons: the Panthers had a QB capable of running, adding an extra running threat to the field per play, and teams that are winning are more likely to run the ball to keep the clock moving and end the game. Additionally, let’s get a more detailed picture by comparing Pass/Run splits per down:

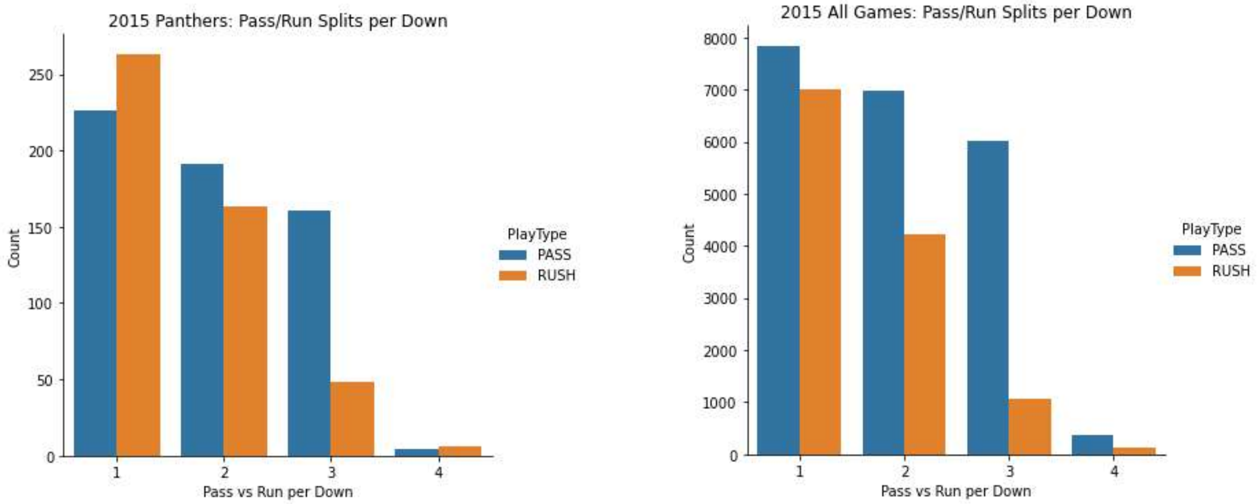


Figure 4: Comparing 2015 Panthers & All NFL Pass/Run Splits per Down

Interestingly, the 2015 Panthers called more runs than passes on first down, which is the only visualized down with a higher run percentage. For every down, the Panthers’ pass/run split was tighter than the rest of the league, which lends to the thought that winning teams will run more.

To validate the above, we plotted the time remaining in the game versus the pass likelihood. Below is the 2015 Panthers and All NFL Time Remaining vs Pass Likelihood:

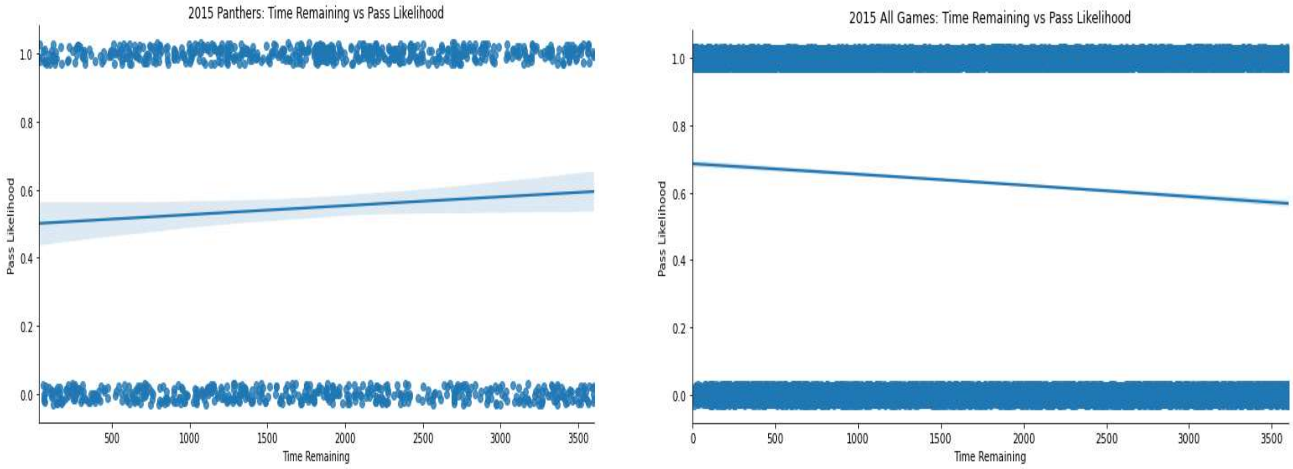


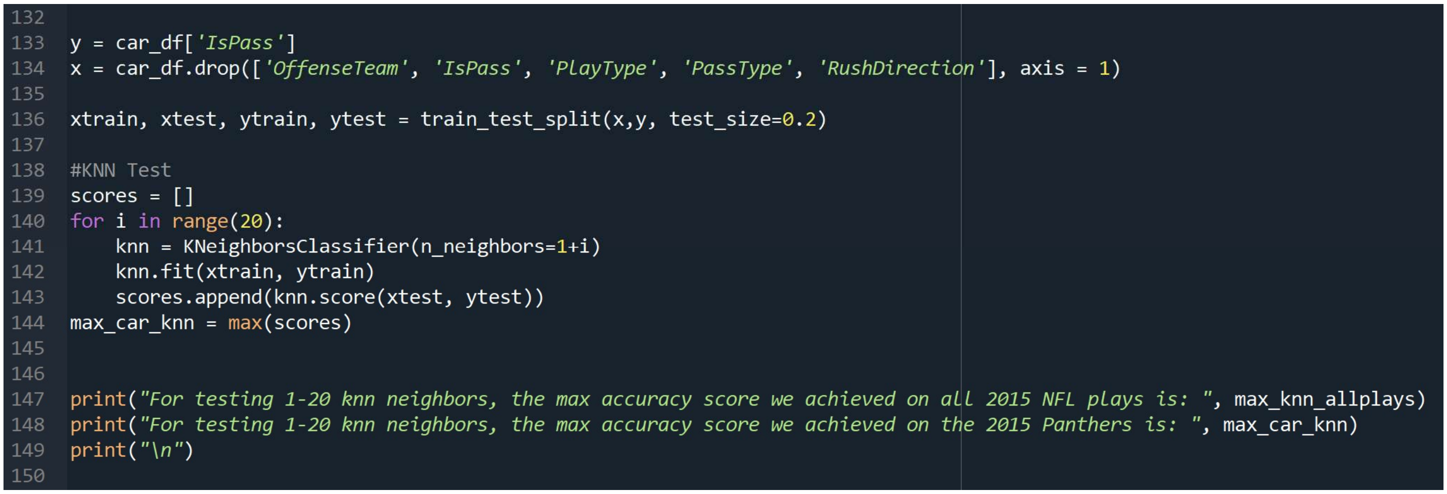
Figure 5: Comparing 2015 Panthers & All NFL Time Remaining vs Pass Likelihood

The chart above is similar to charting Pass/Run splits per quarter. As time remaining increases, the Panther’s are more likely to run; signaling they are winning and trying to run the clock down. The rest of the NFL is more likely to pass as the game progresses;

signaling that they are trying to pass and advance the ball quickly, most likely to catch up in the game.

Once we have the data frame we want to work with, we will create a few different machine learning models to apply to the data and see which model most accurately predicts the correct play call. Machine learning models to be created/considered are: KNN Clustering, SVM, and Random Forest Classification.
With these models, we will essentially be predicting two different outcomes: run or pass, and if pass, what type of pass. So, each model will have an accuracy score for predicting run vs pass and a separate accuracy score for if pass was predicted, did it accurately predict the type of pass? Now we want to compare the model accuracy scores for each individual team, and figure which team was most predictable. So, to accomplish this, we won't be taking the dataset as a whole for a prediction model. Rather, we will loop through each individual team and filter the original dataset with each of the other 31 NFL teams in the "OffenseTeam". So, we will train the model 32 separate times, but with the same parameters we originally found for the Carolina Panthers.

- KNN Model:



```

y = car_df['IsPass']
x = car_df.drop(['OffenseTeam', 'IsPass', 'PlayType', 'PassType', 'RushDirection'], axis = 1)

xtrain, xtest, ytrain, ytest = train_test_split(x,y, test_size=0.2)

#KNN Test
scores = []
for i in range(20):
    knn = KNeighborsClassifier(n_neighbors=1+i)
    knn.fit(xtrain, ytrain)
    scores.append(knn.score(xtest, ytest))
max_car_knn = max(scores)


print("For testing 1-20 knn neighbors, the max accuracy score we achieved on all 2015 NFL plays is: ", max_knn_allplays)
print("For testing 1-20 knn neighbors, the max accuracy score we achieved on the 2015 Panthers is: ", max_car_knn)
print("\n")

```

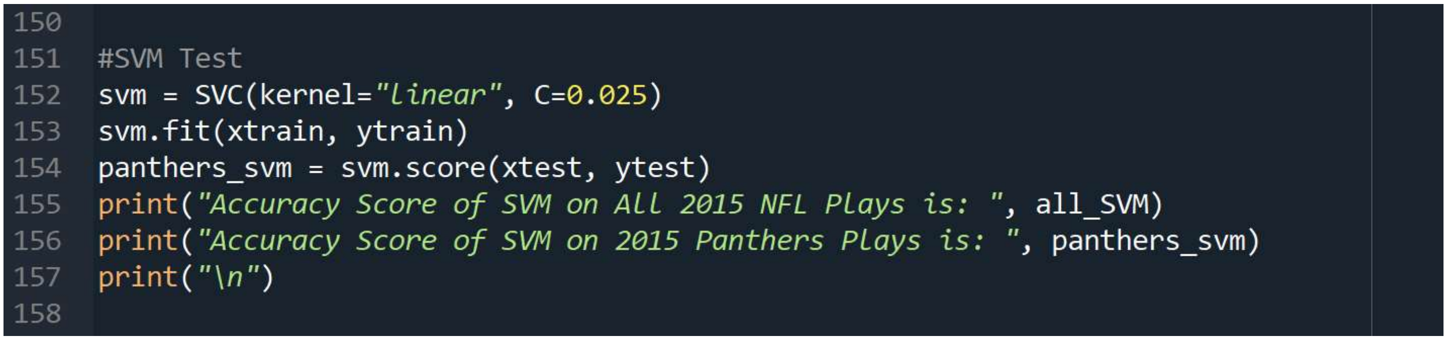
Figure 6: KNN Model Code

The code above parses our df down into features ("x") and responses("y"), splits the data into 80% training and 20% testing datasets, and will loop through the values of 1-20 for the number of neighbors to consider in the KNN model for the Panthers' specific data set. We also created a list to save the model accuracy score for each number of neighbors, we will then get the highest accuracy score and how many neighbors used to find this. The above code outputted:

```
For testing 1-20 knn neighbors, the max accuracy score we achieved on all 2015 NFL
plays is:  0.6118381915526472
For testing 1-20 knn neighbors, the max accuracy score we achieved on the 2015
Panthers is:  0.5727699530516432
```

And the optimum number of neighbors was found to be 10. As predicted, the Panthers were more difficult to predict than the rest of the NFL; however, the model is not super accurate on either dataset. We will use this KNN model with 10 neighbors to test all 32 teams individually and find the Panthers rank among all teams. Results to be discussed in the next section.

- SVM Model:



```

#SVM Test
svm = SVC(kernel="linear", C=0.025)
svm.fit(xtrain, ytrain)
panthers_svm = svm.score(xtest, ytest)
print("Accuracy Score of SVM on All 2015 NFL Plays is: ", all_SVM)
print("Accuracy Score of SVM on 2015 Panthers Plays is: ", panthers_svm)
print("\n")

```

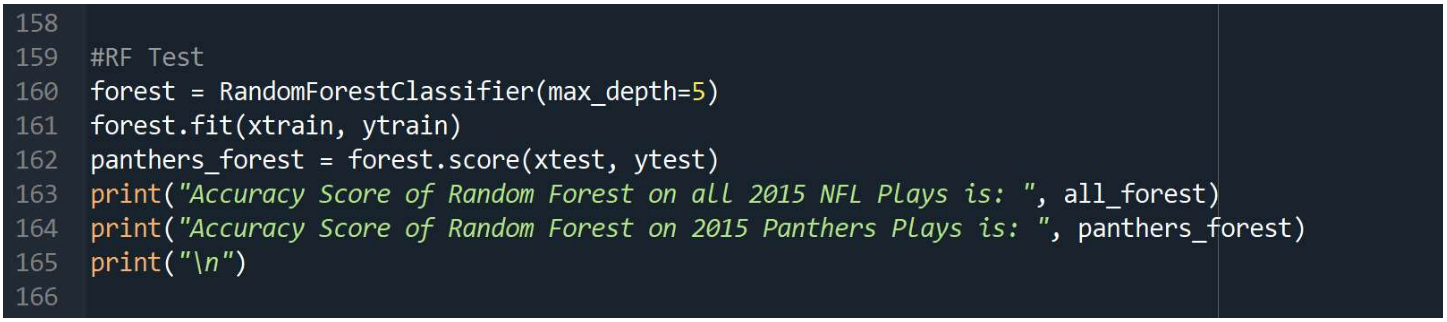
Figure 7: SVM Model Code

The code above creates a linear SVM model to classify plays as either pass or run. The model was fit on the same split of training/test data as the KNN model. We also created a variable to save the accuracy score. Applying this model to both the Panthers' data set and entire NFL data set returned the following output:

```
Accuracy Score of SVM on All 2015 NFL Plays is:  0.7462819750148721
Accuracy Score of SVM on 2015 Panthers Plays is:  0.6901408450704225
```

Again, the model found the Panthers' to be more difficult to predict than the NFL as a whole. This model also appears significantly more accurate than KNN at making play predictions. We will apply this model to all 32 NFL teams individually and discuss in the next section.

- Random Forest Model:



```

#RF Test
forest = RandomForestClassifier(max_depth=5)
forest.fit(xtrain, ytrain)
panthers_forest = forest.score(xtest, ytest)
print("Accuracy Score of Random Forest on all 2015 NFL Plays is: ", all_forest)
print("Accuracy Score of Random Forest on 2015 Panthers Plays is: ", panthers_forest)
print("\n")

```

Figure 8: Random Forest Model Code

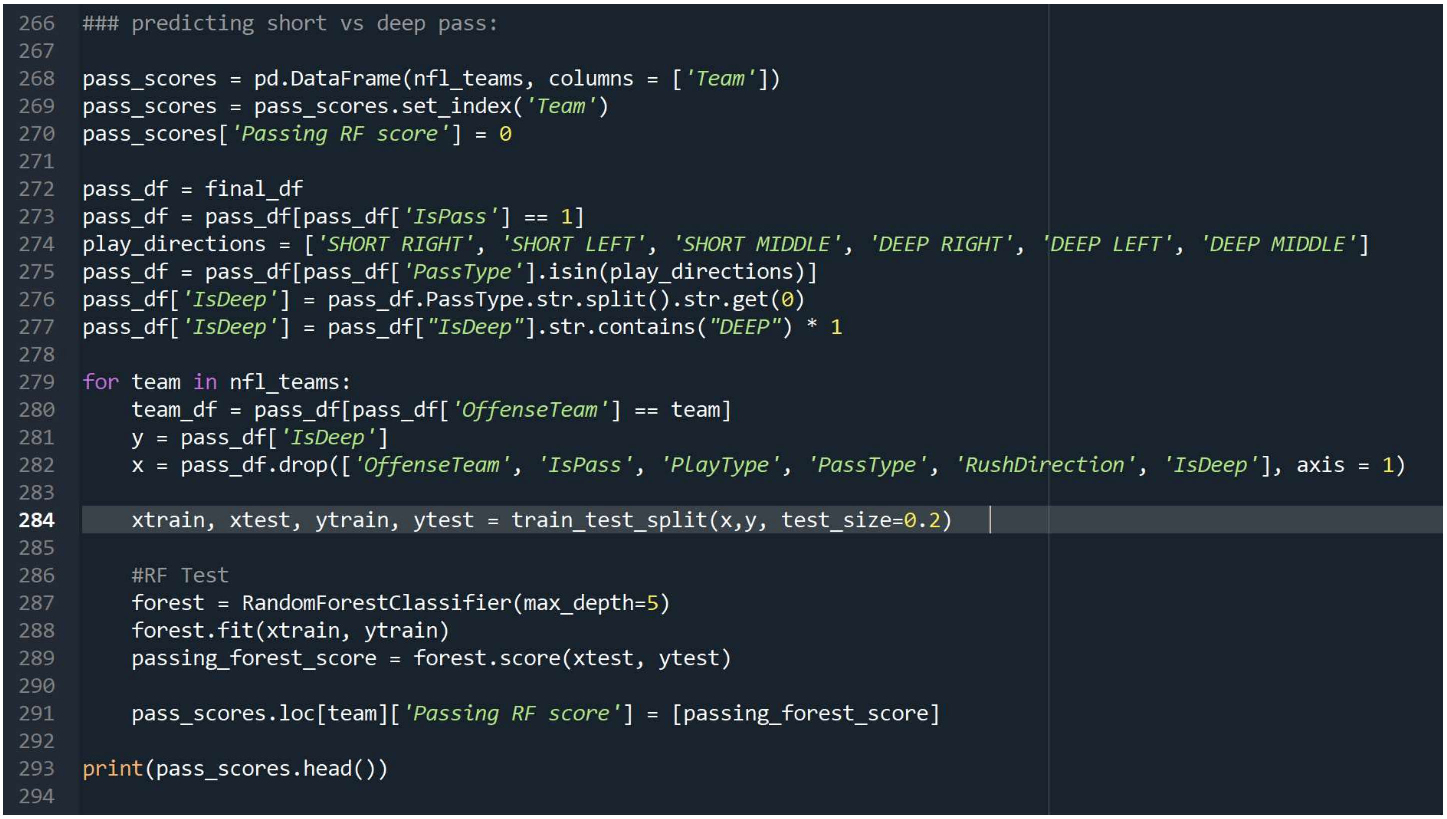
The code above creates a Random Forest model to classify plays as either pass or run. The model was fit on the same split of training/test data as the KNN & SVM model. We also created a variable to save the accuracy score. Applying this model to both the Panthers' data set and entire NFL data set returned the following output:

Accuracy Score of Random Forest on all 2015 NFL Plays is: 0.7562462819750149
Accuracy Score of Random Forest on 2015 Panthers Plays is: 0.7136150234741784

Again, the Random Forest model found the Panthers' to be more difficult to predict than the rest of the NFL. Additionally, the Random Forest model appears to be the most accurate of all 3 models in play prediction. We will apply this model to all 32 NFL teams individually and discuss in the next section.

- Pass Direction Prediction:

On top of predicting Pass/Run plays, we will also attempt to predict whether a pass was thrown short or deep. The below shows the code for predicting pass direction:



```
### predicting short vs deep pass:

pass_scores = pd.DataFrame(nfl_teams, columns = ['Team'])
pass_scores = pass_scores.set_index('Team')
pass_scores['Passing RF score'] = 0

pass_df = final_df
pass_df = pass_df[pass_df['IsPass'] == 1]
play_directions = ['SHORT RIGHT', 'SHORT LEFT', 'SHORT MIDDLE', 'DEEP RIGHT', 'DEEP LEFT', 'DEEP MIDDLE']
pass_df = pass_df[pass_df['PassType'].isin(play_directions)]
pass_df['IsDeep'] = pass_df.PassType.str.split().str.get(0)
pass_df['IsDeep'] = pass_df["IsDeep"].str.contains("DEEP") * 1

for team in nfl_teams:
    team_df = pass_df[pass_df['OffenseTeam'] == team]
    y = pass_df['IsDeep']
    x = pass_df.drop(['OffenseTeam', 'IsPass', 'PlayType', 'PassType', 'RushDirection', 'IsDeep'], axis = 1)

    xtrain, xtest, ytrain, ytest = train_test_split(x,y, test_size=0.2)

    #RF Test
    forest = RandomForestClassifier(max_depth=5)
    forest.fit(xtrain, ytrain)
    passing_forest_score = forest.score(xtest, ytest)

    pass_scores.loc[team]['Passing RF score'] = [passing_forest_score]

print(pass_scores.head())
```

Figure 9: Pass Depth Prediction Code

The code above shows that we will create a new data frame to store model accuracy scores for passing depth predictions and that we will loop through each individual team for the RF model. We will approach only with the Random Forest model as we found it to be the most accurate model previously. The code above separates the PassType column and separates it into deep and short passes. As a response column, IsDeep takes deep passes to be 1 and short passes to be 0. We will discuss the passing depth prediction data frame in the next section.

- PCA Top 5 Components:

Finally, we will attempt to re-fit the models using only the top 5 components found via PCA. While accuracy will almost definitely go down, it will be intriguing to see how much (if any) the accuracy drops.

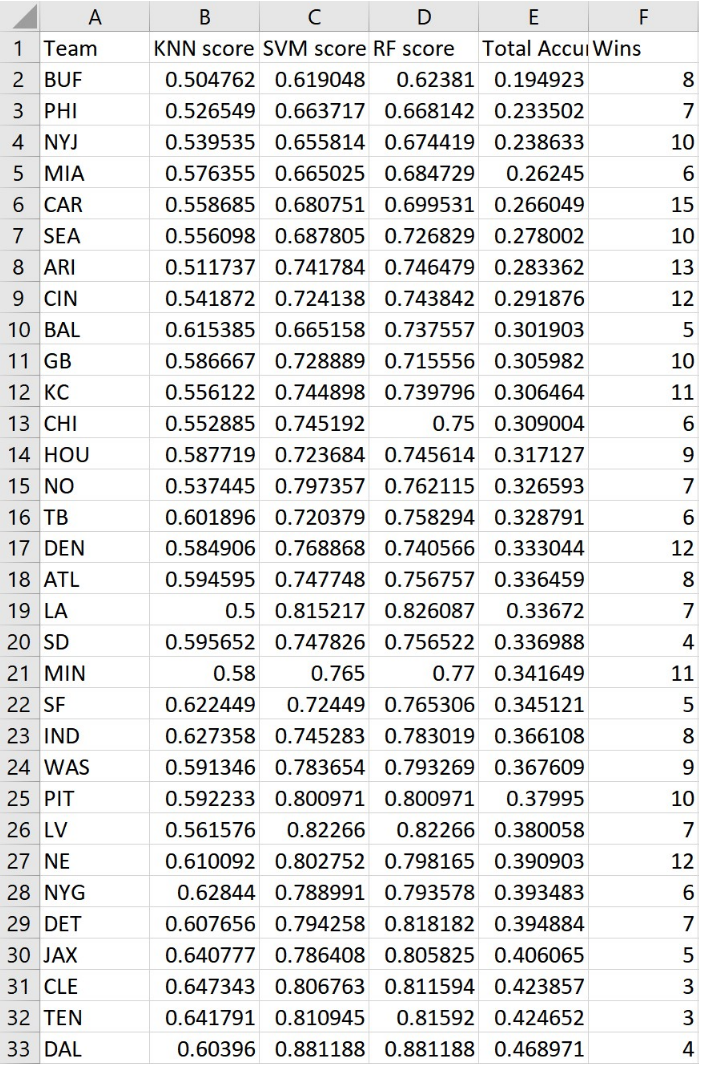
5. Hypothesis

We will find the best machine learning model to use on the Carolina Panthers dataset and use the given accuracy scores as a baseline to compare against the other NFL teams. We will run each created model on the dataset several times to find the true best model to predict the next play of the Carolina Panthers. Once run, we will have 64 best accuracy scores: a run vs pass prediction accuracy and a pass direction prediction accuracy for each NFL team. Finally, we will re-run the experiment with PCA on the top 5 features and compare these accuracy scores with the originals.

My hypothesis is that the Carolina Panthers should be one of the more difficult teams for the model to accurately predict, if not the most difficult. As a fan of the Panthers, I have the background knowledge of knowing the Panthers got a lot of production by running plays out of formations that would typically have the opposite play-call. For example, a 3 WR, 1 TE, 1 RB, shotgun formation is typically a passing play but the Panthers would often call a rushing play with the QB as they had a very athletic QB. On the flipside, a heavy formation with less WR's and more TE's and RB's would typically by a running play, but the Panthers would often pass out of this formation due to the skillset of their receiving TE. By parsing down the data into a team-by-team training set, I would expect a higher accuracy score than taking the entire dataset as a league and fitting it to the machine learning models. This is because by training the data by each individual team, there are most likely hidden trends that are specific to each team that the model would catch. If the model were to be trained on the entire league, these trends would most likely be smoothed by other teams not behaving in the same manner. As for the accuracy scores (or as we are judging here, each team's unpredictability score) it will be interesting to see whether there were other teams that were actually more unpredictable than the Carolina Panthers, and what those teams' records for the 2015 season were. This could validate whether the Panthers were the most successful team based on varied play-calling and being unpredictable, or whether they simply had the best players of the season who could execute better than others. However, if another team has a lower accuracy score, but also presents a good season record, it most likely proves that a combination of unpredictability and raw player talent lead to the 15-1 record. Additionally, it will be interesting to see if it was difficult to predict run vs pass, was it easy or difficult to predict pass location if pass was successfully predicted. Ultimately, football is a team game and there are a plethora of factors that could lead to a better record that aren't present in this dataset. Some examples of outside factors that could present noise in the data could be: total travel distance, gameday weather, fan noise, coaching changes, player injuries, or other random variances in performances.

6. Evaluation and Final Results

The final accuracy scores for each NFL team were compiled into a data frame named "nfl_scores" and output into the below image:

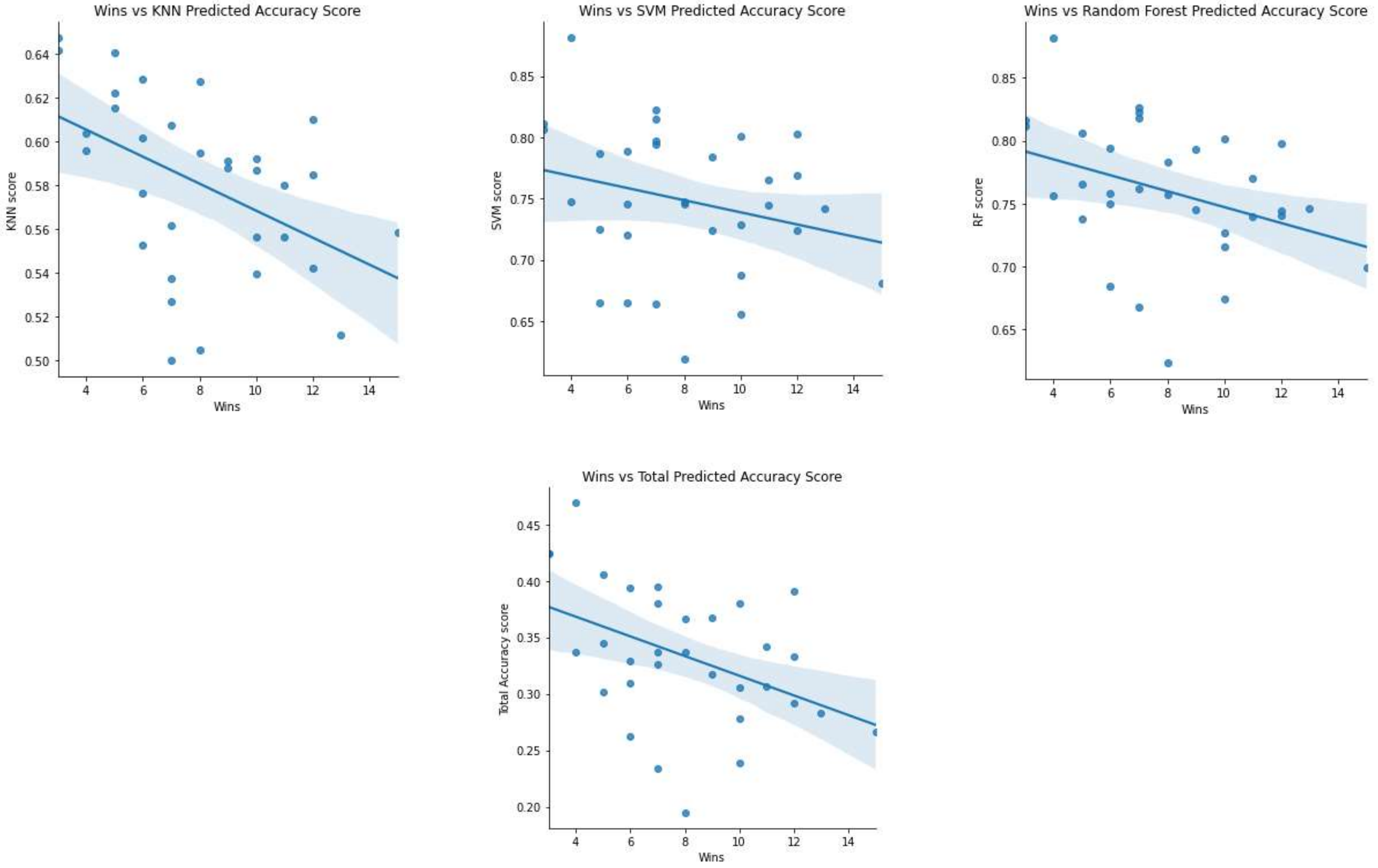


	A	B	C	D	E	F
1	Team	KNN score	SVM score	RF score	Total Accui	Wins
2	BUF	0.504762	0.619048	0.62381	0.194923	8
3	PHI	0.526549	0.663717	0.668142	0.233502	7
4	NYJ	0.539535	0.655814	0.674419	0.238633	10
5	MIA	0.576355	0.665025	0.684729	0.26245	6
6	CAR	0.558685	0.680751	0.699531	0.266049	15
7	SEA	0.556098	0.687805	0.726829	0.278002	10
8	ARI	0.511737	0.741784	0.746479	0.283362	13
9	CIN	0.541872	0.724138	0.743842	0.291876	12
10	BAL	0.615385	0.665158	0.737557	0.301903	5
11	GB	0.586667	0.728889	0.715556	0.305982	10
12	KC	0.556122	0.744898	0.739796	0.306464	11
13	CHI	0.552885	0.745192	0.75	0.309004	6
14	HOU	0.587719	0.723684	0.745614	0.317127	9
15	NO	0.537445	0.797357	0.762115	0.326593	7
16	TB	0.601896	0.720379	0.758294	0.328791	6
17	DEN	0.584906	0.768868	0.740566	0.333044	12
18	ATL	0.594595	0.747748	0.756757	0.336459	8
19	LA	0.5	0.815217	0.826087	0.33672	7
20	SD	0.595652	0.747826	0.756522	0.336988	4
21	MIN	0.58	0.765	0.77	0.341649	11
22	SF	0.622449	0.72449	0.765306	0.345121	5
23	IND	0.627358	0.745283	0.783019	0.366108	8
24	WAS	0.591346	0.783654	0.793269	0.367609	9
25	PIT	0.592233	0.800971	0.800971	0.37995	10
26	LV	0.561576	0.82266	0.82266	0.380058	7
27	NE	0.610092	0.802752	0.798165	0.390903	12
28	NYG	0.62844	0.788991	0.793578	0.393483	6
29	DET	0.607656	0.794258	0.818182	0.394884	7
30	JAX	0.640777	0.786408	0.805825	0.406065	5
31	CLE	0.647343	0.806763	0.811594	0.423857	3
32	TEN	0.641791	0.810945	0.81592	0.424652	3
33	DAL	0.60396	0.881188	0.881188	0.468971	4

Figure 10: nfl_scores Data Frame Accuracy Scores per Model

As seen above, the Panthers are 5th lowest in Total Accuracy percentage (found by multiplying all 3 models' accuracy scores against each other) and are among the lowest for each model. The below charts show the trend line in accuracy percentage and wins.

The first line of charts shows each individual model's accuracy score vs wins.
The second line of charts shows the total accuracy score of all models vs wins.


Wins vs KNN Predicted Accuracy Score

Wins vs SVM Predicted Accuracy Score

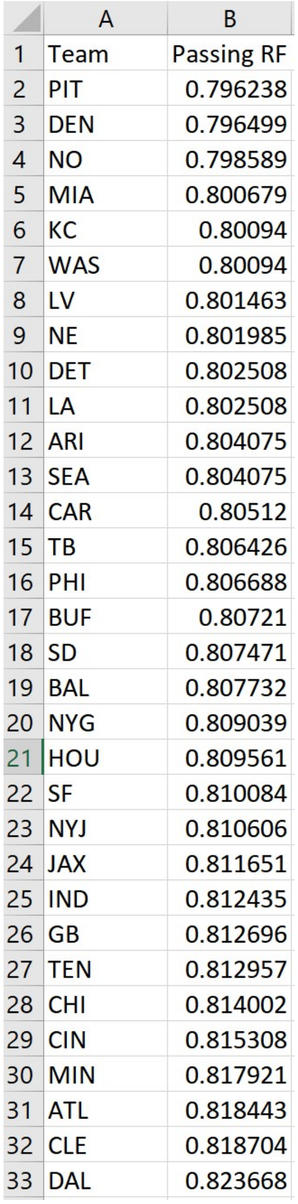
Wins vs Random Forest Predicted Accuracy Score

Wins vs Total Predicted Accuracy Score

Figure 11: Model Accuracy Scores vs Win Totals

All 3 models show a direct relationship between win totals and accuracy score. Confirming the hypothesis that the more successful teams are more difficult to predict. It is interesting that the KNN model has the lowest total accuracy scores, but also provides the steepest relationship between accuracy score and wins.

The below data frame “pass_scores” shows the random forest model accuracy for predicting whether a pass will be short or deep. Only the random forest model was used because it was found to be the most accurate model and due to necessary computing power of running all 3 models on each individual NFL team.



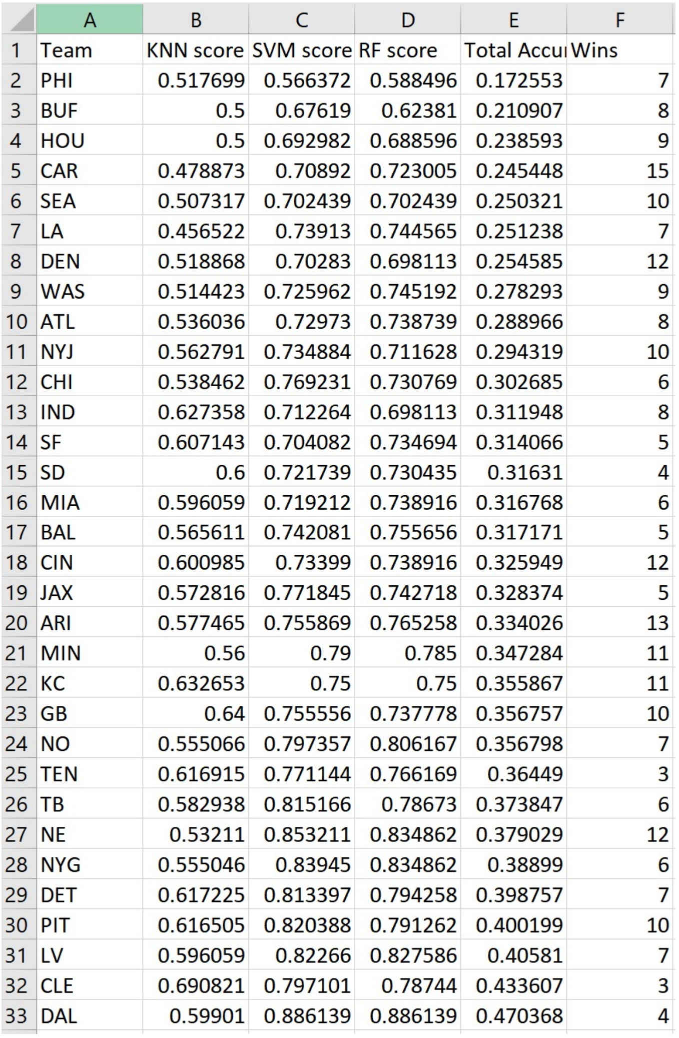
	A	B
1	Team	Passing RF
2	PIT	0.796238
3	DEN	0.796499
4	NO	0.798589
5	MIA	0.800679
6	KC	0.80094
7	WAS	0.80094
8	LV	0.801463
9	NE	0.801985
10	DET	0.802508
11	LA	0.802508
12	ARI	0.804075
13	SEA	0.804075
14	CAR	0.80512
15	TB	0.806426
16	PHI	0.806688
17	BUF	0.80721
18	SD	0.807471
19	BAL	0.807732
20	NYG	0.809039
21	HOU	0.809561
22	SF	0.810084
23	NYJ	0.810606
24	JAX	0.811651
25	IND	0.812435
26	GB	0.812696
27	TEN	0.812957
28	CHI	0.814002
29	CIN	0.815308
30	MIN	0.817921
31	ATL	0.818443
32	CLE	0.818704
33	DAL	0.823668

Figure 12: Passing Depth Prediction Random Forest Accuracies

The passing depth prediction dataframe shows the Panthers were the 13th worst team in prediction accuracy – pretty middle of the pack. However, it appears the random forest model is pretty accurate for every NFL team in predicting whether a team will be throwing short or deep – hovering around 80% accuracy. This is most likely due to deep passing situations being obvious when teams are down big, late in games, or certain formations.

Finally, we have performed Principal Component Analysis (PCA) on each individual team's dataset to get the top 5 components per team and then re-fitting the original 3 models of KNN, SVM, and

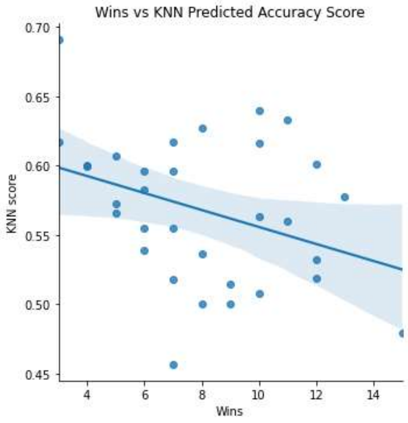
Random Forest to predict either pass or run for each play. The following data frame is the output of the PCA'd model accuracies.

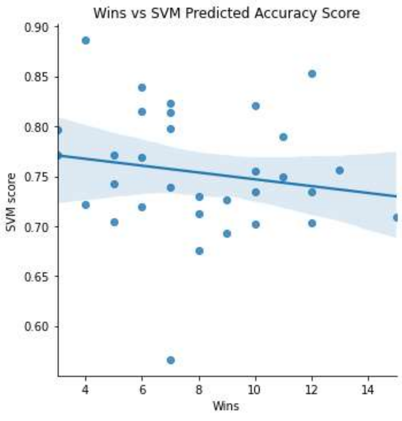


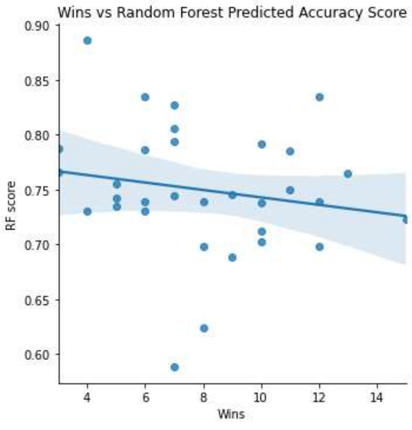
	A	B	C	D	E	F
1	Team	KNN score	SVM score	RF score	Total Accu	Wins
2	PHI	0.517699	0.566372	0.588496	0.172553	7
3	BUF	0.5	0.67619	0.62381	0.210907	8
4	HOU	0.5	0.692982	0.688596	0.238593	9
5	CAR	0.478873	0.70892	0.723005	0.245448	15
6	SEA	0.507317	0.702439	0.702439	0.250321	10
7	LA	0.456522	0.73913	0.744565	0.251238	7
8	DEN	0.518868	0.70283	0.698113	0.254585	12
9	WAS	0.514423	0.725962	0.745192	0.278293	9
10	ATL	0.536036	0.72973	0.738739	0.288966	8
11	NYJ	0.562791	0.734884	0.711628	0.294319	10
12	CHI	0.538462	0.769231	0.730769	0.302685	6
13	IND	0.627358	0.712264	0.698113	0.311948	8
14	SF	0.607143	0.704082	0.734694	0.314066	5
15	SD	0.6	0.721739	0.730435	0.31631	4
16	MIA	0.596059	0.719212	0.738916	0.316768	6
17	BAL	0.565611	0.742081	0.755656	0.317171	5
18	CIN	0.600985	0.73399	0.738916	0.325949	12
19	JAX	0.572816	0.771845	0.742718	0.328374	5
20	ARI	0.577465	0.755869	0.765258	0.334026	13
21	MIN	0.56	0.79	0.785	0.347284	11
22	KC	0.632653	0.75	0.75	0.355867	11
23	GB	0.64	0.755556	0.737778	0.356757	10
24	NO	0.555066	0.797357	0.806167	0.356798	7
25	TEN	0.616915	0.771144	0.766169	0.36449	3
26	TB	0.582938	0.815166	0.78673	0.373847	6
27	NE	0.53211	0.853211	0.834862	0.379029	12
28	NYG	0.555046	0.83945	0.834862	0.38899	6
29	DET	0.617225	0.813397	0.794258	0.398757	7
30	PIT	0.616505	0.820388	0.791262	0.400199	10
31	LV	0.596059	0.82266	0.827586	0.40581	7
32	CLE	0.690821	0.797101	0.78744	0.433607	3
33	DAL	0.59901	0.886139	0.886139	0.470368	4

Figure 13: PCA'd Models' Accuracy Scores

The above dataset shows a slightly less accurate, but largely similar outcome to the un-PCA'd dataset. Let's also examine the same trend lines we displayed above.







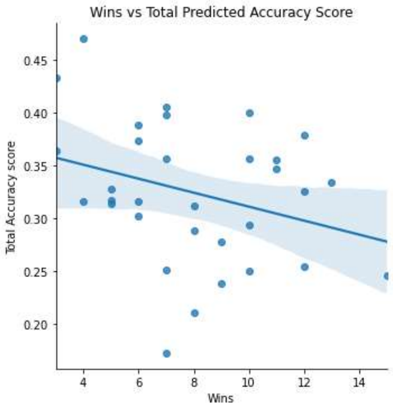


Figure 14: PCA'd Models Win Totals vs Predicted Accuracy Scores

The above graphs show similar outcomes to the non-PCA’d data, showing that KNN has the steepest trend in misclassification and win totals. However, as a whole the PCA’d data does not have as steep of a trend vs wins as the un-PCA’d data has. This shows that features that were dropped have an effect on lowering the prediction accuracy of the models for higher winning teams.

Overall non-PCA’d data shows a correlation between winning teams being more unpredictable. While the 2015 Panthers were not the most unpredictable team, they were in the top ranks in misclassification for all models, showing that they were largely unpredictable and coupling this with higher end talent, lead to the best record in the NFL in 2015. This has been an interesting statistical deep dive into one of my personal favorite seasons in NFL history and gave a lot of insight to the kind of unpredictability and randomness that sports (or any human interaction – especially in a team environment) can show.