SIADS 696 Milestone II Project Report
NFL Stats Prediction
Aarushi Srivastava (saarushi@umich.edu), Eldon Wilderman (eldonw@umich.edu)

Note: During this project while coding, deepnote was used with the AI autocomplete code suggestions turned on.

Introduction

The NFL stands for National Football League and is one of the most popular sports in the United States. The NFL was founded in 1920 and there are currently 32 teams in the NFL, and over the period we investigated. We investigated NFL seasons from 2012 through 2024. The NFL season runs from September to February and ends with the Super Bowl in February. NFL is known for its sportsmanship, team camaraderie, and huge fan base. Teams specialize in offensive and defensive play, fielding positions such as quarterback, running back, and defensive back, each with distinct roles and quantifiable performance metrics.

Our motivation for this project was based largely on the competitiveness of the NFL and our curiosity regarding the factors that contribute to success. The NFL presents a large number of possible strategies due to its complexity and evolving nature over time. We are trying to solve the problem of identifying which team performance characteristics and play styles most strongly correlate with winning outcomes in the NFL. Data can provide key insights that intuition, conventional wisdom, or observation may not yield by themselves. Given that this is such a complicated game where these things may be especially difficult to use to process strategies, the potential utility of data analysis and machine learning methods is even more promising.

For the unsupervised learning portion of the project, we used K-means clustering to group teams based on offensive and defensive performance metrics into three distinct performance categories (elite, average, and underperforming). We found the optimal amount of clusters for K-mean clustering through the elbow method. This clustering approach revealed meaningful patterns in team strategies and outcomes, which can help inform team management decisions for the coaches. Each cluster was analyzed to identify characteristic strengths and weaknesses, providing interpretable insights for coaches and analysts on how different play styles correlate with success metrics.

For the supervised learning portion of the project, we aimed to produce a model that could predict whether or not a team would win a given game, based on a few key performance statistics of the team. After testing some options, we found a logistic regression model that was the best predictor of win success. We found that Defensive Interceptions, Offensive interceptions and fumbles, Sacks, and QB hits were key predictors of whether a team would win, among other important factors. This has large implications for what teams ought to focus on and emphasize in order to win games. Compared to related projects, our novel contribution was combining supervised win prediction with unsupervised play style clustering to provide both predictive and descriptive insights into team success.

Related Work

1. Kaggle NFL Big Data Bowl (2020)
    a. Summary: This project primarily focused on classifying defensive coverage schemes using player tracking data and supervised learning models to predict defensive play types.
    b. Difference: Our project aims to combine both team-level offensive and defensive statistics to cluster teams and predict win outcomes. We are not just focused on defensive statistics only. Hence, providing more broader strategic insights.
2. Uncovering NFL Team Patterns by Clustering (2025)
    a. Summary: This project aims to use K-Means clustering to group NFL teams into offensive style clusters using offensive yard metrics.

    b. Difference: However, our project aims to combine both unsupervised clustering with supervised win prediction to not only categorize teams based on style and performance but also to quantify how these clusters will relate to win outcomes.
3. A Predictive Analytics Model for Forecasting Outcomes in the National Football League Games Using Decision Tree and Logistic Regression (2023)
    a. Summary: This project created a decision tree and logistic regression model to predict NFL wins based on some selected team statistics, then evaluates the best model
    b. We have selected different team statistics than this paper, which focused heavily on yards and record prior to game day. By contrast we focused on a broader range of more granular statistics such as sacks, solo tackles, and whether it was a home vs. away game. We included statistics related to yards and turnovers, which were included in this study, although we were more granular with the statistics. We also tested a Gradient Boosted Decision Tree model.

Data Source

The NFL data we are using can be downloaded from Kaggle. The data is provided in a CSV format. On the Kaggle site, there are 4 files that are found upon clicking download: weekly_player_stats_offense.csv, weekly_player_stats_defense.csv, weekly_team_stats_offense.csv, and weekly_team_stats_defense.csv. However, for the purposes of our machine learning models, we only used weekly_team_stats_offense.csv and weekly_team_stats_defense.csv. There are a lot of important variables in these datasets. In the offense dataset, the fields included total_off_yards, pass_attempts, complete_pass, incomplete_pass, passing_yards, rush_attempts, rushing_yards, pass_touchdown, rush_touchdown, interception, fumble, fumble_lost, and win_pct. In the defense dataset, the fields included solo_tackle, assist_tackle, sack (defensive sacks made), interception (defensive interceptions forced), def_touchdown (defensive touchdowns), qb_hit (quarterback hits), and win_pct. For the unsupervised analysis, we combined weekly_team_stats_defense.csv and weekly_team_stats_offense.csv into one dataframe called team_df_merged, merging on season, season_type, and team. The total amount of records for this dataframe was 582 for all 32 NFL teams. In order to perform the supervised learning analysis, we also combined the weekly team stats offense and defense files, doing so on week, team, and season, in order to get data for the NFL games we wanted to predict. From there we performed further manipulation to ready the data for the supervised analysis. The time period of all the records was from 1999 to 2024.

Feature Engineering

As mentioned, for the unsupervised learning algorithm, we combined weekly_team_stats_offensive.csv and weekly_team_stats_defensive.csv and aggregated weekly data to season-team level data by grouping on season, team, and season_type. For unsupervised learning, we selected key features using domain knowledge, focusing on offensive production, defensive effectiveness, and outcome metrics. To prepare the data for k-means clustering, we dropped any non-numeric columns. We also checked for any missing values across the columns and dropped any columns that contained all 0's as those columns have no variance and k-means clustering relies on distance calculations. We also applied VarianceThreshold to remove features with near zero variance that do not contribute to clustering. Prior to clustering, we applied StandardScaler to standardize all the numeric features to ensure that all features contribute equally to distance calculations and model weights. List of all the features used in k-means clustering are as follows: selected_features ['total_off_yards', 'pass_attempts', 'rushing_yards', 'pass_touchdown', 'rush_touchdown', 'sack', 'interception_def', 'solo_tackle', 'assist_tackle', 'def_touchdown', 'win_pct_off', 'win_pct_def'].

For the supervised learning analysis we began initially with the dataframe mentioned previously in the Data Source section of the report, which had initially 177 columns (many were not columns of interest) and 7,088 rows of team game stats. We first created several binary variables that were going to be

important for our analysis. These were “regular_season,” “home_game,” and “won.” regular_season was equal to one if the game was a regular season game. Home_game was one if the team was the home team, and “won” was one if the team won the game. Once this was completed, we filtered out ties. One remaining potential issue at this point was that of data leakage resulting from stats from the same game appearing in both the training and test sets. In other words, for example defensive interception stats for one team are equivalent to offensive interceptions for the team’s opponent, but the opponent may appear in a different set than the team. This could lead to the model predicting a team’s win or loss based on knowledge of how it performed leaked into the model. To remedy this and make sure the same game did not appear twice, we filtered out the home team for half the games and the away team for the other half. This way the balance of home and away games would remain consistent, every game would still appear, and we would not have one, not two rows per game. This cut our row count in half to 3,544. The final task was to select our columns of interest for each team, which were: ["shotgun", "no_huddle", "qb_dropback", "qb_scramble", "pass_attempts", "complete_pass", "incomplete_pass", "air_yards", "yards_after_catch", "rush_attempts", "rushing_yards", "tackled_for_loss", "first_down_pass", "first_down_rush", "third_down_converted", "third_down_failed", "fourth_down_converted", "fourth_down_failed", "fumble_forced_off", "fumble_not_forced_off", "interception_off", "interception_def", "fumble_forced_def", "fumble_not_forced_def", "solo_tackle", "assist_tackle", "sack", "qb_hit", "home_game", "regular_season", "won", "game_id_off"] (note that game_id_off was not used during model training or testng and was simply kept to create a reference index so that we could find specific games a datapoint corresponds to during the failure analysis). At this point, data was ready to be divided into features and labels and then training and test sets.

Part A: Supervised Learning

The model was split into train and test sets with “won” as the labels and the features described in the previous section as the features (except “game_id_off”). Each feature was in the format of the value of the statistic the team recorded for the game, except the “home_game” and “regular_season” which was either one or zero. For logistic regression model testing, the standard scaler was used to normalize the data. A random_state parameter of 42 was used consistently throughout the modeling process. The model types selected for testing were Random Forest, Gradient Boosted Decision Tree, and Logistic Regression (for all the logistic regression max_iter was set to 1000 so that all tested models would converge before max_iter was reached). This gives us a lot of variety, as we have tree based models and a non-tree based model (Logistic Regression), as well as the gradient boosting aspect of the Gradient Boosted Decision tree model. These models were also chosen because we needed classifiers to predict the categorical win variable, with about a few thousand datapoints. Hyperparameter tuning was performed with grid searches using five fold cross validation using a variety of hyperparameters depending on the model. The parameter grids for the models were as follows: Random Forest: {"n_estimators": [10, 100, 1000, 2000], "max_depth": [2, 5, 10, 20]}, Gradient Boosted Decision Tree: {"max_depth": [2, 5, 10, 20], "learning_rate": [0.01, 0.1, 1]}, Logistic Regression: {"C": [0.01, 0.1, 1, 10, 100], "solver": ["newton-cg", "lbfgs", "liblinear", "sag", "saga"]}.

In terms of the evaluation metric used for model comparison, it is not so relevant to us whether the model favors minimizing false negatives versus false positives, so as a result we compared accuracy between models. However, we recorded precision and recall during our comparison of models in order to be aware of these dynamics of weighting more toward minimizing false negatives versus false positives. We also include F1 scores. Below in figure 1 is a table representing five fold cross validation metric results compared for each of our selected best models after hyperparameter tuning. As can be seen in figure 1, the best performing model was the logistic regression model. This model was chosen as our final model, and it had the default hyperparameter values, where C was 1.0 and with the lbfgs solver (except max_iter was 1000). It had a slightly higher recall score versus precision score, indicating that there were a bit fewer false negatives versus false positives predicted by the model.

Figure 1.

Model	Accuracy std	Accuracy mean	Recall std	Recall mean	Precision std	Precision mean	F1 Score std	F1 Score mean
Random Forest	0.01265468496	0.8419771179	0.02135225133	0.8495391174	0.01008173899	0.8343289356	0.01373694742	0.8417635955
GBDT	0.01814213294	0.8476218796	0.02588472621	0.8541162576	0.02116409765	0.8410780477	0.01833995852	0.847297517
Logistic Regression	0.01127143196	0.865682568	0.02454306897	0.8700656758	0.01581044197	0.8606938934	0.01206944888	0.8650406393

Below in figure 2 is the size of the coefficients in the final logistic regression model, showing the significance of each of the features. As can bee seen the five most important predictors that contribute positively to the chance of a win were rush attempts, defensive interceptions, completed passes, and sacks, and QB hits. The five most important predictors that contributed negatively to the chance of a win were QB dropbacks, failed fourth downs, offensive interceptions, offensive forced fumbles, and incomplete passes. Some notes on some of these predictors, it's important to note that failed fourth downs and rush attempts are likely powerful predictors in part because once a team is winning later in the game, it becomes more advantageous to pursue conservative strategies in order to run clock and lower risk. It is unsurprising that teams that win tend to log more rush attempts, because later in the game they are likely playcalling more conservatively, and likewise teams that get behind and then lose are more likely to have failed fourth down conversions just from going for it (and failing) on fourth down more. QB dropbacks negatively impacting performance may also reflect this dynamic of riskier strategies becoming more optimal for teams that are losing, The other key insight we can gain from glancing at figure 2 is that statistics measuring turnovers for and against a team are very important, as are statistics indicating pressure on the quarterback (sacks and QB hits). Completing and not completing passes both also seem to predict the success of teams greatly.

Figure 2.

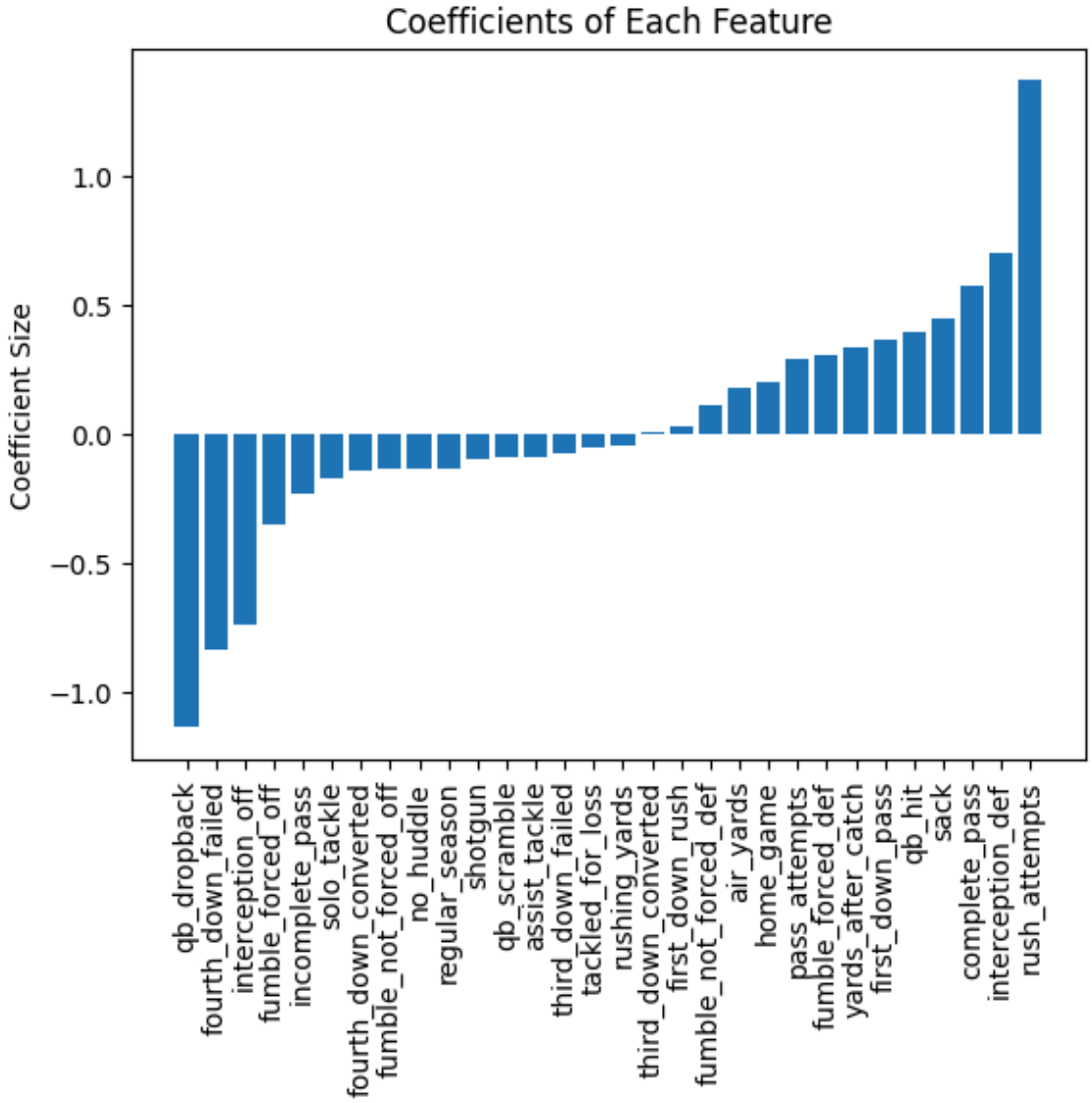

Coefficients of Each Feature
Coefficient Size
1.0
0.5
0.0
−0.5
−1.0
qb_dropback
fourth_down_failed
interception_off
fumble_forced_off
incomplete_pass
solo_tackle
fourth_down_converted
fumble_not_forced_off
no_huddle
regular_season
shotgun
qb_scramble
assist_tackle
third_down_failed
tackled_for_loss
rushing_yards
third_down_converted
first_down_rush
fumble_not_forced_def
air_yards
home_game
pass_attempts
fumble_forced_def
yards_after_catch
first_down_pass
qb_hit
sack
complete_pass
interception_def
rush_attempts

Figure 3.

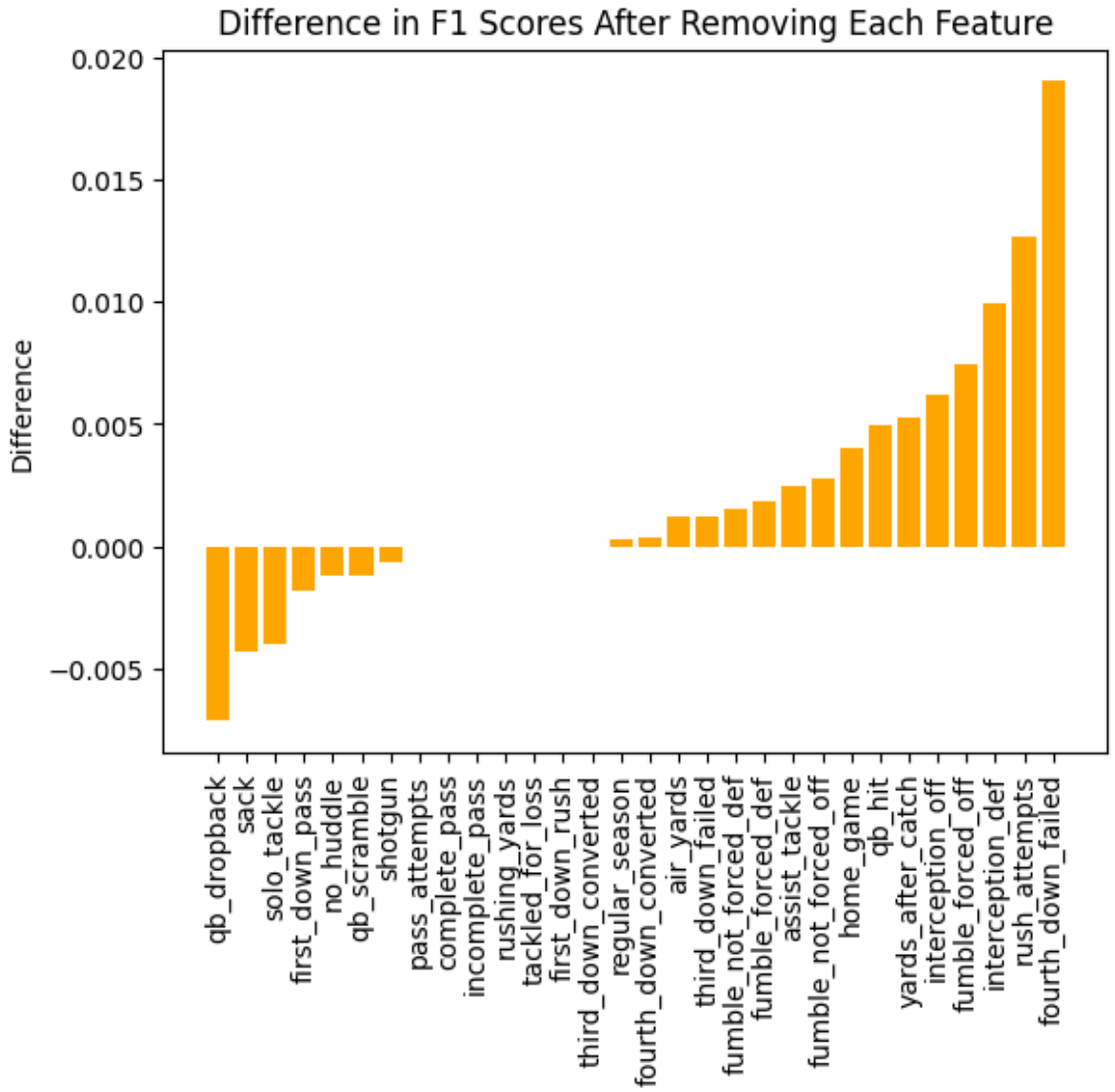
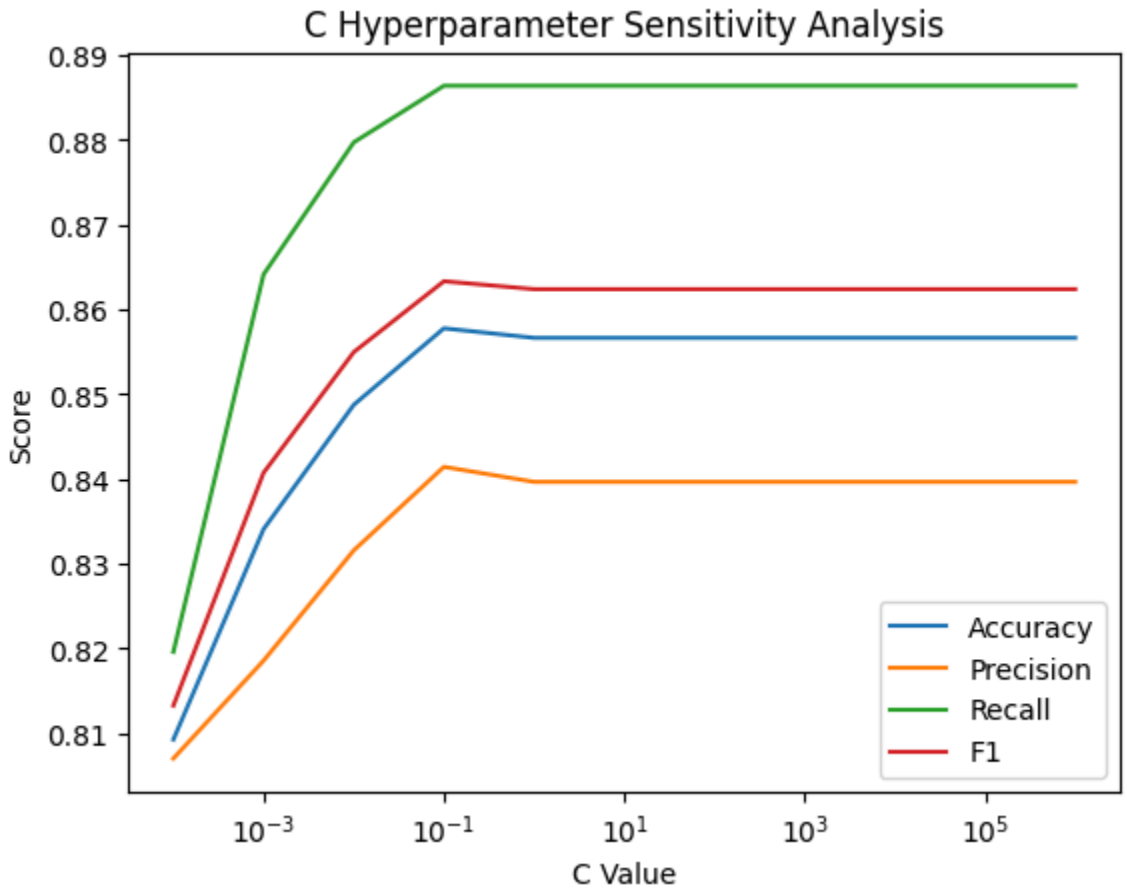


Figure 3 above shows the results of some testing of the effects of removing various features from the model. The x axis shows the feature removed, and the y axis shows the difference in F1 score between the model with and without the feature. As can be seen and should be noted from the scale of the y axis, very little difference was made in terms of model prediction F1 score by removing most features, with some slightly more significant effects from defensive_interceptions, rush_attempts, and fourth_down_failed, though these were still very small and caused reduced F1 score. Therefore, we concluded that it was important to maintain all of the variables in our model without eliminating them, because removing them did not notably improve the model and even could make it less effective.

Figure 4.



We found during hyperparameter tuning that the choice of solver did not substantially impact accuracy, however we did do some sensitivity analysis of the C parameter, shown by figure 4 above. We tested multiple values of C, scaling by a factor of 10, and we tested Accuracy, Precision, Recall, and F1 scores. Scores increase rapidly until about C = 0.1, at which point they plateau. This indicates that increasing regularization is not very consequential until beyond about C = 0.1, at which point it is bad for model predictive effectiveness. The effect is most pronounced for Recall, then F1 score, Accuracy, and Precision.

For our failure analysis, we identified three examples of failure, each believed to likely be of a different type. These are that: The model falsely predicted (False positive) that the Green Bay Packers were the victors of the 2014 NFC championship game against the Seattle Seahawks, it predicted that falsely that the 2023 Packers lost to the Saints in their home matchup (false negative), and it predicted wrongly that the Miami Dolphins beat the Bills in their 2024 road matchup (false positive). The data from these games are respectively represented by figures 5, 6, and 7, where the axis is standard deviations from the training set average, and the y axis is features.

Figure 5.

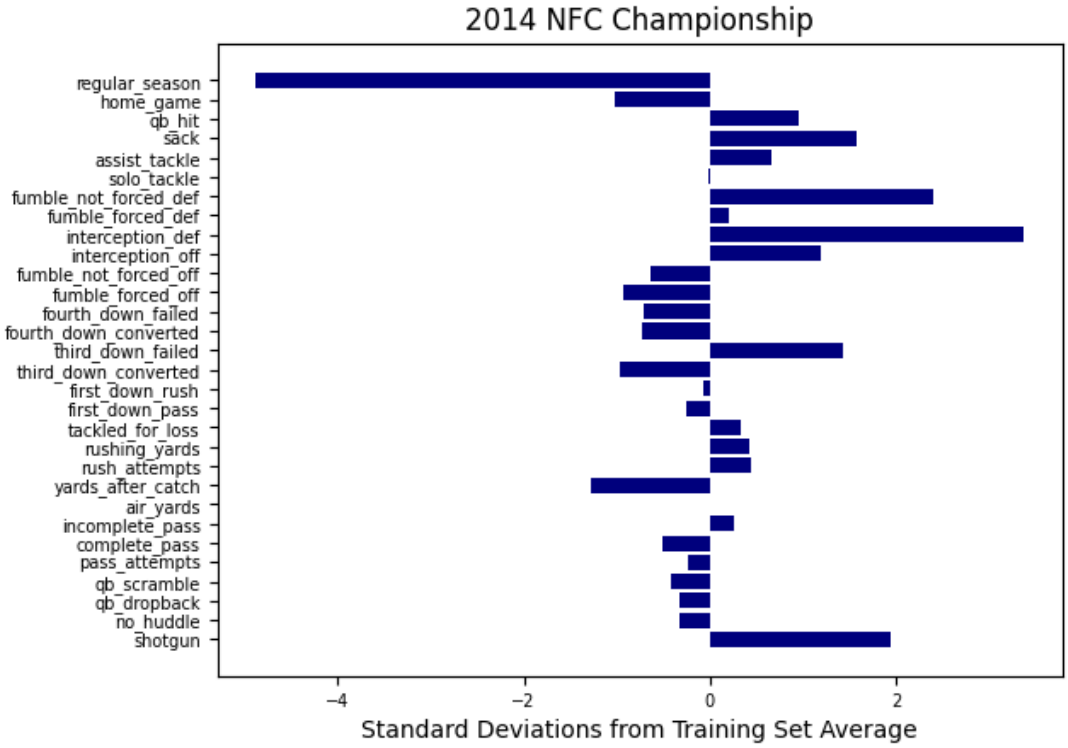


Figure 6.

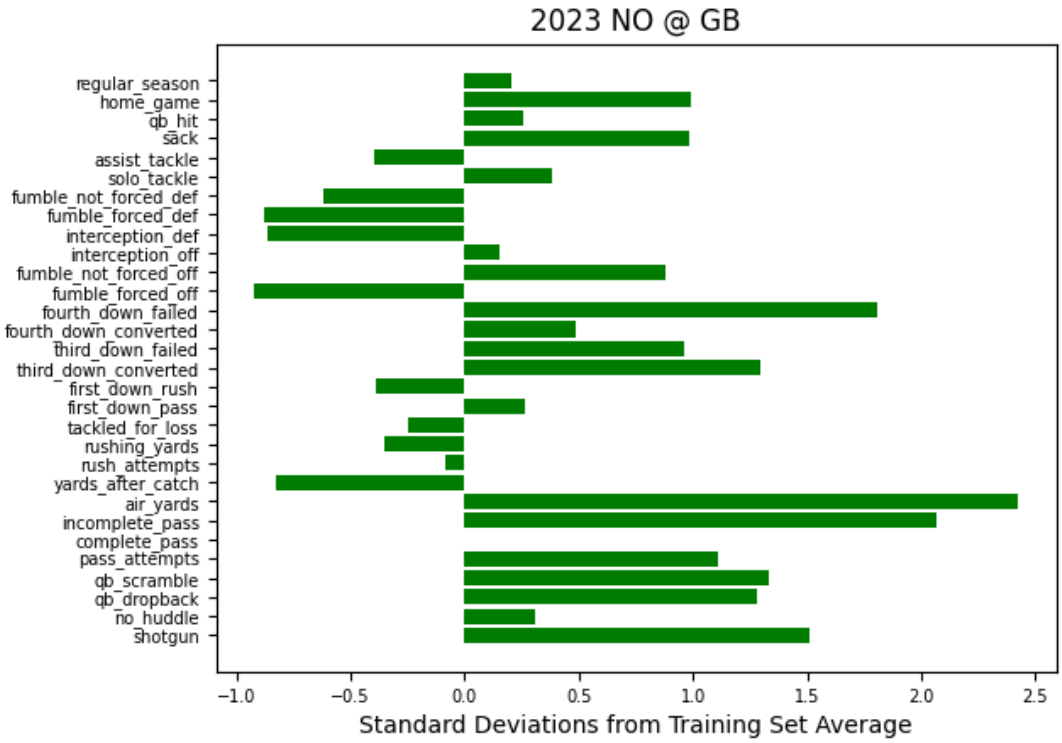
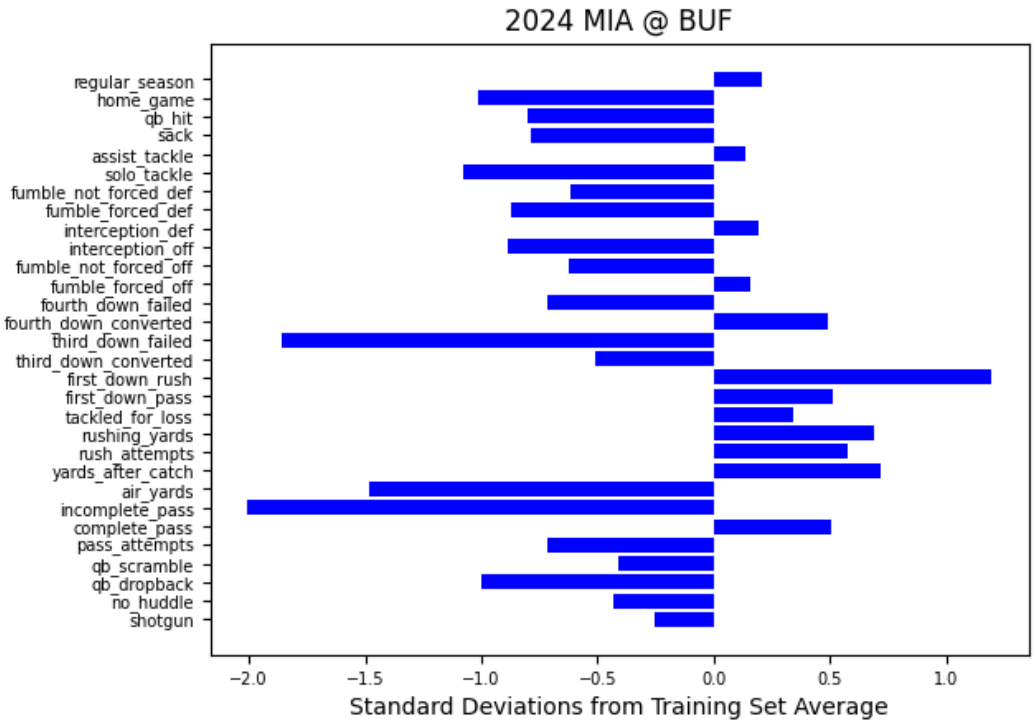


Figure 7.



For the 2014 NFC championship (figure 5), it seems likely to us that this false positive is likely caused by the relatively high number of defensive interceptions and defensive fumbles not forced. We think this example could represent a systematic error. The model was highly confident in this prediction, yet got it wrong. Many of the other crucial and important stats to our model read as being fairly mediocre and modest, not strongly indicating a win. One idea to help remedy this is more regularization could help bring down the weight of this in this case and make a correct response more likely.

For the 2023 Saints at Packers game (figure 6), we think this very confident yet false prediction of a loss could represent an edge case resulting from an impressive comeback (Eldon actually saw this game, and is a Packers fan). This game featured low opposing turnovers and many failed fourth downs, in particular, yet the ground was made up in air yards. This game represents a late comeback that could be an unusual circumstance edge case, in terms of why it was misclassified. In order to account for these in the future, we could make sure to include a sample weighted with a plethora of comeback games exhibiting the statistical patterns of this one.

Finally, For the 2024 Dolphins at Bills game, we believe this could be an example of a false prediction due to random noise. The model is very confident, yet there is not a clear statistical pattern here in terms of stats that are strong for the Dolphins. It seems like a large amount of stats spread out are contributing to this. One remedy that may help is to train and tune our models optimizing for precision, which can help reduce false positives.

Part B: Unsupervised Learning - Methods Description

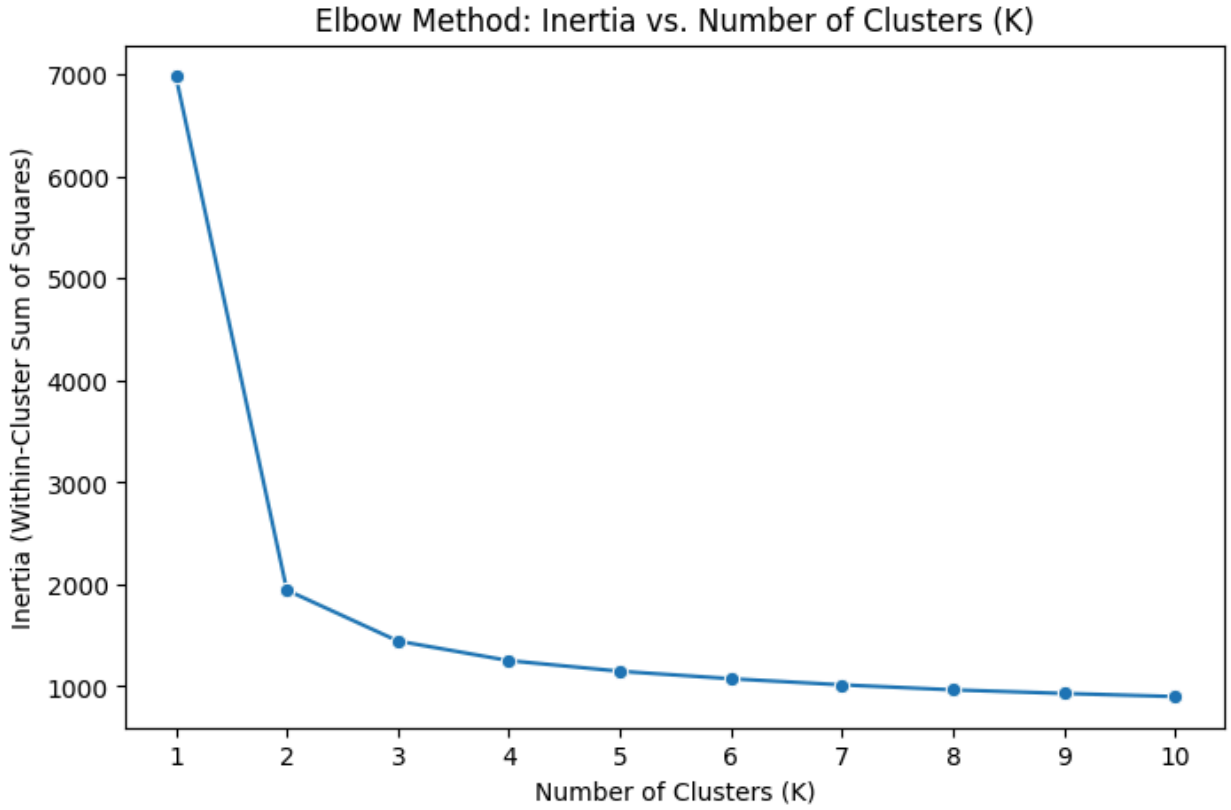
We applied unsupervised learning to group NFL teams (season and season type) into distinct performance profiles based on offensive and defensive statistics. Each row in our dataset represents a single team in a single season. This was aggregated from weekly_team_stats_offense.csv and

weekly_team_stats_defensive.csv. We received permission from our instructor, Saurabh Budholiya, to perform one unsupervised learning method due to the size of our team. The learning methods we used for unsupervised learning are K-Means Clustering. In terms of feature selection, we tried to select all the fields that capture both offensive and defensive characteristics using domain knowledge. For example, we picked ['total_off_yards', 'pass_attempts', 'rushing_yards', 'pass_touchdown', 'rush_touchdown', 'sack', 'interception_def', 'solo_tackle', 'assist_tackle', 'def_touchdown', 'win_pct_off', 'win_pct_def']**.** We picked these fields to ensure we accurately capture the team's strength and performance and meaningful to be able to cluster into various performance groups. As we prepared our model for unsupervised learning, we removed any non-numeric columns and any columns that might have low variance. We also applied StandardScaler to ensure that all the features are on the same scale. We used one unsupervised learning method, K-Means Clustering, because of its complementary strengths and fundamentally different mechanism as a non-probabilistic, distance-based algorithm. In this case non-probabilistic refers to the fact that this algorithm does not model data as coming from a probabilistic distribution. K-Means was selected for its computational efficiency, especially when applied to large, standardized numeric datasets like ours. In this context, non-probabilistic means that K-Means does not model the data as coming from a specific statistical distribution; instead, it deterministically assigns each data point to the nearest cluster centroid purely based on minimizing Euclidean distance. K-Means is purely a distance based algorithm. This makes K-Means particularly effective for our problem of identifying broad, distinct groupings of NFL team performance patterns, such as underperforming, average, or elite teams. Its distance-based nature ensures that teams with similar offensive and defensive statistics are grouped together in meaningful, compact clusters, allowing us to extract clear and actionable insights from high-dimensional team performance data.We determined the optimal number of clusters using the elbow method and validated the results with a Silhouette Score of 0.4061, indicating moderately strong cluster structure. For K-means clustering, we performed hyperparameter tuning by determining the optimal number of clusters (K) for our dataset. For the optimal number of clusters (K), we used the elbow method to derive the value. We plotted inertia values for K=1 to 10. We identified the elbow point (K=3) where adding more clusters yielded diminishing returns. After selecting K=3, we validated this by calculating the Silhouette Score to be 0.4061. This moderate silhouette score indicates reasonable cluster separation and cohesion, supporting the effectiveness of our chosen K value. Another example of hyperparameter tuning we did was in the KMeans function setting the function parameter, n_init to be equal to 10. The function parameter, n_init, specifies the number of times K-means runs with different initial centroid seeds. It is important to tune this parameter because higher n_init values reduce the risk of converging to a poor local minimum. Choosing the optimal value helps to improve clustering performance and stability.

Part B: Unsupervised Learning - Unsupervised Evaluation

For evaluating our K-means clustering model, we used the silhouette score. The silhouette score measures how similar each data point is to its own cluster vs. other clusters. A higher silhouette score indicates well separated clusters with better cohesion and separation. In this case, our silhouette score was 0.4061. We used the elbow method to identify the optimal number of clusters. As you can see, the graph shows diminishing returns after k = 3.

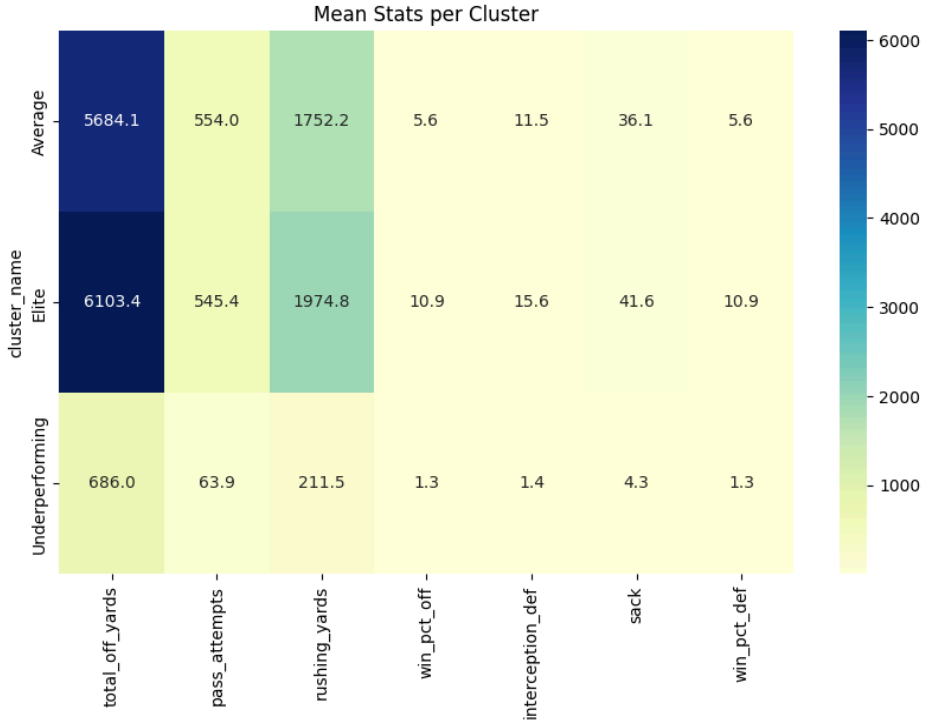
Figure 8 shows the optimal value of k using the elbow method



Model	Parameters	Silhouette Score	Key Findings
K-Means Clustering	K = 3, n_init = 10	0.4061	Identified three clusters representing underperforming, average, and elite based on offensive and defensive performance metrics.

We decided to do a further investigation of the three clusters created: underperforming, average, and elite. We created a heatmap showing the mean statistics per cluster for key performance metrics across NFL teams. For this heatmap the key performance metrics we selected were: total_off_yards, pass_attempts, rushing_yards, sack, interception_def, win_pct_off, and win_pct_def.

Figure 9



From this heatmap in figure 9, you can see the elite cluster has the highest total offensive yards (about 6103) and rushing yards(1975). It also has the highest win percentages offensively (10.9) and defensively(10.9) compared to the other clusters. Lastly, it has strong defensive performance with high sacks(41.6) and interceptions(15.6). The teams that belong to the elite cluster are in Cluster 2 mentioned in the code. Each tuple consists of the team, season year, and season type - regular or post season. From this group, we can identify that there are some top performing teams across multiple seasons, such as the Kansas City Chiefs, Buffalo Bills, Philadelphia Eagles, San Francisco 49ers, Green Bay Packers, and New England Patriots. Some of these teams appear in the cluster many times from 2012 to 2024, indicating consistent high performance over time. For example, KC was mentioned in (2013, 'KC', 'REG'), (2015, 'KC', 'REG'),(2016, 'KC', 'REG'),(2017, 'KC', 'REG'),(2018, 'KC', 'REG'),(2019, 'KC', 'REG'),(2020, 'KC', 'REG'),(2021, 'KC', 'REG'),(2022, 'KC', 'REG'),(2023, 'KC', 'REG'), (2024, 'KC', 'REG'). In fact, in 2024, KC (2024, KC, REG) won the superbowl and you can see the tuple being mentioned in the elite cluster group. Also, it was surprising to see teams such as Houston Texans in 2012 and 2014, suggesting that they had standout single season performances despite their struggles in other years. The Elite cluster is characterized by teams that not only excel offensively and defensively but also maintain high win percentages, validating the clustering model's effectiveness in identifying top-tier NFL teams each season. From the heatmap, you can see that the average cluster has the second highest total offensive yards (5684.1) and rushing yards (1752.2). It also has the second highest win percentages offensively (5.6) and defensively (5.6) compared to the other clusters. It has relatively good sacks (36.1) and interceptions (11.5).The teams that belong to the average cluster in Cluster 1 mentioned in the code. The average cluster consists of teams that consistently deliver competitive performances but do not dominate to the extent of elite teams. It includes a wide range of teams such as the Dallas Cowboys (DAL), Minnesota Vikings (MIN), Pittsburgh Steelers (PIT), Miami Dolphins (MIA), Detroit Lions (DET), and New York Giants (NYG) across multiple seasons. For example, some of the years where Pittsburgh Steelers had average performance was (2012, 'PIT', 'REG'), (2013, 'PIT', 'REG'),(2019, 'PIT', 'REG'),(2021, 'PIT', 'REG'),(2022, 'PIT', 'REG'). In addition the inclusion of strong teams as Green Bay

Packers and Kansas City Chiefs in some years reflects average performance due to injuries or roster changes.For example, Kansas City Chiefs is primarily in the elite cluster, however in (2012, 'KC', 'REG') and (2014, 'KC', 'REG') they were in the average cluster with the high possibility of internal team structure or sheer luck. Lastly, in the underperforming cluster, we have very low total offensive yards (686), rushing yards (211.5), and very low win percentages offensively (1.3) and defensively (1.3) compared to the other clusters. Poor sacks (4.3) and interceptions (1.4). The teams that belong to the underperforming cluster are in Cluster 0 mentioned in the code are typically the teams that do not perform well. There was a unique observation where Kansas City Chiefs in REG season are typically in the elite or average cluster. However, there was an instance where Kansas City Chiefs in the POST season was in the underperforming cluster (2015, 'KC', 'POST') and (2017, 'KC', 'POST'). This could be a relative within season performance finding and the clustering performed is data-driven and is driven by factors such as schedule, coaching, and situational success, which is not captured in numeric feature clustering. For our sensitivity analysis, we decided to test out the number of clusters (K) for various numbers. Even though K=3 was derived from doing the elbow method, we wanted to understand how changing the value of K to 2 or 4 would affect the clustering and the silhouette score. The results were pretty sensitive to varying the hyperparameters in this case K. Based on the elbow method, we selected K=3 as it provided a good balance between cluster separation and interpretability based on the elbow method. The silhouette score for K=3 was 0.4061, indicating moderate cluster cohesion and separation. When K=2, the silhouette score was higher at 0.6603, suggesting stronger separation and well defined clusters. However, utilizing only 2 clusters results in less detailed grouping. When K=4, the silhouette score decreased to 0.3919, indicating diminishing returns with additional clusters. This analysis highlights that our model is sensitive to the choice of K, and selecting an appropriate number of clusters is crucial to derive actionable insights in NFL team performance clustering.

Discussion

1. One thing that was quite challenging initially for the supervised section was the related to the data leakage issue described earlier. Initially, there was no filtering of the rows representing the same game, and so we were returning unusually high performance metrics for the initial training of the models. Once it was figured out what was going on, we had to figure out how to remedy the issue without compromising the underlying representative sample of the data. Luckily, a solution was able to be reached. We gained a lot of substantial insights from the unsupervised analysis, such as the importance of turnovers, completed passes, and putting pressure on the quarterback. Although it was expected that these would be important, the extent was quite surprising. Taken together, some of these trends all could imply the importance of big plays on performance. Just a few turnovers or sacks above normal can make a win more likely. It would have been interesting to investigate this hypothesis further through another analysis that put a more explicit emphasis on big plays, such as including number of long pass plays and run plays as a variable. It might also be interesting to look at things like number of long drives, consistent drives, or consecutive first downs etc. to measure the importance of consistency. It would be interesting to see if the importance of pass completion might be explained by this, or if it is something else such as simply that teams whose defenses get more stops get more pass completions, which is one hypothesis of why that is so crucial. These are all jumping off points that could be really interesting for further analyses with more time and/or data.
2. Through performing K-Means clustering on NFL team data, I learned how unsupervised learning can effectively uncover underlying patterns in complex sports datasets, grouping teams based on offensive and defensive characteristics into distinct clusters like elite, average, and underperforming. I was surprised to see that clustering labels reflect statistical groupings rather than performance rankings. A team's success can be influenced by qualitative factors not captured in numeric features and this is an important distinction to make when presenting the results to anyone. Another thing that was interesting to note is that when we were performing sensitivity analysis and set k=2, the silhouette score was very high.This was interesting because it indicated

clear grouping but lacked nuance. A major challenge was selecting meaningful features from over 160 columns without introducing noise. To combat this issue, I applied domain knowledge, variance filtering, and standardization using StandardScaler to ensure robust results. With more time and resources, I would extend this solution by incorporating additional unsupervised methods like hierarchical clustering for comparison and using dimensionality reduction (PCA). I would also explore time series analysis modeling to track how team clusters evolve across seasons, which could reveal patterns of growth, decline, or strategic shifts over time. This would provide actionable insights for team management and strategy decisions.

Ethical Considerations

1. In terms of ethical considerations for the supervised learning portion, it's important to be clear about how results should be interpreted and what they mean, so that teams and NFL stakeholders do not draw the wrong conclusions leading to poor decisions. Chiefly, these predictive models should be understood as explaining factors that predict team success in a game, not as unbeaten proof that orienting a team and its practices around maximizing a particular statistic is bound to lead to wins. Rather, these results should be taken as strong evidence that particular statistics or strategies indicate important aspects of how to win a game in the NFL. Further not all models are perfect and there may be key aspects that our model does not capture. Another important point is that the NFL football is a very physical and often dangerous game that can lead to serious injuries for players, and so when changing or emphasizing different strategies as a result of this analysis, it's critical to consider how player safety may be affected. Just because one perceives that as a result of this analysis it may be advantageous to pursue a particular strategy in order to win more games does not mean that it would be a good move in terms of players' health and safety. The best way to remedy these potential ethical issues is to make them clear and straightforward for those interpreting the results of our paper. We need to make it understood by readers what these ethical issues are, why they arise, and what they mean for those who might be interested in this paper in terms of utilizing it in an ethical way.
2. Using K-Means clustering for NFL team analysis can present several ethical considerations. For example, data biases may occur if certain teams or seasons are underrepresented. In addition, over reliance on historical data might ignore roster or team strategy changes, limiting relevance. Clusters might also be misinterpreted as absolute judgements, rather than relative groupings. This can lead to overconfident strategic decisions for some teams. Also, stakeholders may assume that all teams within a cluster are identical, ignoring individual nuances such as situational factors. Clusters may inadvertently reinforce perceptions of certain teams as inherently "bad" or "underperforming," affecting decisions like funding, fan engagement. As a result, it is important to mention that clustering affects current season team performance and not team worth or future potential.

Statement of Work

1. Aarushi worked on the unsupervised learning component of the project, which included feature engineering, selecting relevant features, choosing the appropriate clustering model (K-Means), performing model evaluation (e.g. elbow method and silhouette scores), interpreting cluster results, and writing sections of the final report
2. Eldon worked on initial data cleaning/exploration, writing sections of the final report, and the supervised learning component of the project, which included readying data for the unsupervised analysis (feature engineering, data manipulation), hyperparameter tuning, model evaluation tasks, and failure analysis.

Work Cited

Gifford, M., & Bayrak, T. (2023, August 7). *A predictive analytics model for forecasting outcomes in the National Football League games using decision tree and logistic regression*. Decision Analytics Journal. https://www.sciencedirect.com/science/article/pii/S2772662223001364#sec3

Hyde, P. (2025, May 22). *NFL stats 2012-2024*. Kaggle. https://www.kaggle.com/datasets/philiphyde1/nfl-stats-1999-2022/data

NFL Big Data Bowl. Kaggle. (n.d.). https://www.kaggle.com/competitions/nfl-big-data-bowl-2020

Thorward, R. (2025, May 5). *Uncovering NFL team patterns by clustering*. Medium. https://medium.com/inst414-data-science-tech/uncovering-nfl-team-patterns-by-clustering-8a34a0578908