Executive Summary

Project Proposal

American football is one of the most popular sports in the United States, and it is also a very complex sport. There are many different factors that can affect the outcome of a game, including the players, the coaches, the plays, and the weather.

Our project will focus on using data to analyze American football. We will use data from past games to identify trends and patterns. We will also be using team statistics and team data for past seasons and use the game and team data to predict the outcome of future games and for this current NFL season.

We are interested in this project because we are big fanatics of the NFL and football in general. Also, we believe that this data could be used to improve the understanding of the sports and its results.

Our data sources for the project include:

- Game statistics
- Team statistics
- Weather data

We will use this data to answer the following research questions:

- What are the factors that most affect the outcome of an American football game?
- How can data be used to predict the outcome of future games (scores and winners)?

We believe that this project has the potential to make a significant contribution to the field of American football. We are excited to see what we can learn from the data.

Description of Data Sources

NFL Box Scores

- Dataset includes box scores for every NFL game from 1960 to 2017
- Box scores include statistics for each home and away team for each game
- Another csv file includes the weather data for each game but we will not be using this data because it is not complete

NFL Game Results and Teams

- There are 2 main datasets that we will be using from this source
- The first dataset includes game results for every NFL game from 1966 to 2023
  - It also includes descriptive info such as if it is a playoff game, if it is played at a neutral site, and some weather information if available

  - It also includes betting data for each game such as the spread, over/under, and money line for the games
- The second dataset includes information regarding the teams in the NFL
  - It includes the team name, the team abbreviation, the conference, and the division

NFL Team Stats

- Dataset includes team statistics for each team for each season from 2010 to 2021
- The dataset includes basic information such as win-loss percentage and which season
- The dataset also includes more in depth information regarding offensive and defensive statistics such as points scored, yards gained, and turnovers
- It also includes playoff data for each team for each season

Description of Observational Units

NFL Box Scores

- Observational unit in the box scores data set is each game played in the NFL from 1960 to 2017
- There is an emphasis on the statistics between each home and away team
- Data set was built by an individual contributor in 2017, and no additional information is provided in Kaggle about the data set
- The observational unit in the weather data set is each game played in the NFL from 1960 to 2013
  - Data after 2013 is not included in the data set

NFL Game Results and Teams

- The observational unit in game results dataset is each game played in the NFL from 1966 to 2023
- The observational unit in the team dataset is each team in the NFL (including teams that are no longer in the NFL or have changed their names, i.e. the San Diego Chargers or the Washington Football Team)
- Data set was built from publicly available NFL data, weather provided by the NOAA, and betting data from a variety of sources but cross referenced with Pro Football Reference.

NFL Team Stats

- The observational unit in the team stats dataset is each team in the NFL for each season from 2010 to 2021
  - So, the dataset includes data for each of the 32 teams for each of the seasons from 2010 to 2021

- This data was created using three different sources: pro-football-reference.com, covers.com, teamrankings.com
- The author used various CSV files and web scraping to combine 88 tables to create this data set

Description of Main Variables of Interest

NFL Box Scores

- In this dataset, the main variables of interest for us would be home and away teams' first downs, their net yards, the total plays, average gain, and time of possession
- We trimmed down to these variables because this data will be valuable when combining with the other datasets
- We deemed these variables important since they give a good summary of the game and how the statistics differed throughout the seasons when teams were at home or away

NFL Scores and Teams

- ScoresGame Resultset Game Results main variables of interest in this dataset are the home and away teams' scores, the home and away teams' yards, and the home and away teams' turnovers - The main variables of interest in this dataset are the home and away teams' scores, the betting favorite, the date of the game, and the weather
  - The score and date variables were taken from publicly available NFl data, the weather was taken from the NOAA, and the betting favorite was taken from a variety of sources but cross referenced with Pro Football Reference
- Team Dataset
  - The main variables of interest in this dataset are the team name, the team abbreviation, the conference, and the division
  - These variables are important because they will be used to merge the data with the other datasets
  - This data was taken from public NFL data

NFL Team Stats

- The main variables of interest in this dataset are the team name, the season, the win-loss percentage, the point differential, the points scored per game, the points allowed per game, the yards gained per game, the yards allowed per game, and the possession time per game
- This data was taken from 3 sources: pro-football-reference.com, covers.com, teamrankings.come

Description of Data Cleaning Process

NFL Box Scores

- We decided to only take into account seasons from 2010 and onwards since this would be more relevant to the current NFL season
- First we got the columns information and decided which variables were important to us
- Next, we dropped the columns that we did not need and converted the date column to a datetime object
- We also converted all the numeric columns to numeric data types
- We discarded the weather csv file in this data set

NFL Game Results and Teams

- Game Result Dataset
  - We decided to only take into account seasons from 2010 and onwards since this would be more relevant to the current NFL season (especially so the players, coaches, and teams would be more relevant and similar to the current NFL season)
  - We first decided which columns were important to us and dropped the columns that we did not need
  - There were some rows missing data for scores, so we dropped those rows
  - We also converted the date column to a datetime object
  - There were some rows missing data for the weather, but we decided to keep them as the scores for those games were still present and we could still use that data for our analysis
- Team Dataset
  - We decided which columns were important to us and dropped the columns that we did not need such as the team conference and division before 2002 (especially because we are only considering seasons from 2010 and onwards)
  - There were some teams that did not have a current division, so we dropped those rows (these were teams that existed before 2002)
  - We realized that the conference value for the Jets was incorrect, so we manually changed that to the correct value
  - We sorted the teams by their division and conference to make it easier to read
  - We decided to keep the teams that are no longer in the NFL because their team name still exists in the game results and scores datasets as the teams changed names or moved cities (i.e. the San Diego Chargers to the Los Angeles Chargers or the Washington Football Team to the Washington Commanders)
    - This provides a way to connect the past team data with more recent team data as the names and cities of the teams have changed over time

NFL Team Stats

- This data set only contains data from 2010 to 2021, so we did not need to filter out any data
- We decided which columns were important to us and dropped the columns that we did not need such as perc_punt_20, Turnover_perc, etc.
- There were no missing values in this data set, so we did not need to drop any rows

Setting Up Data

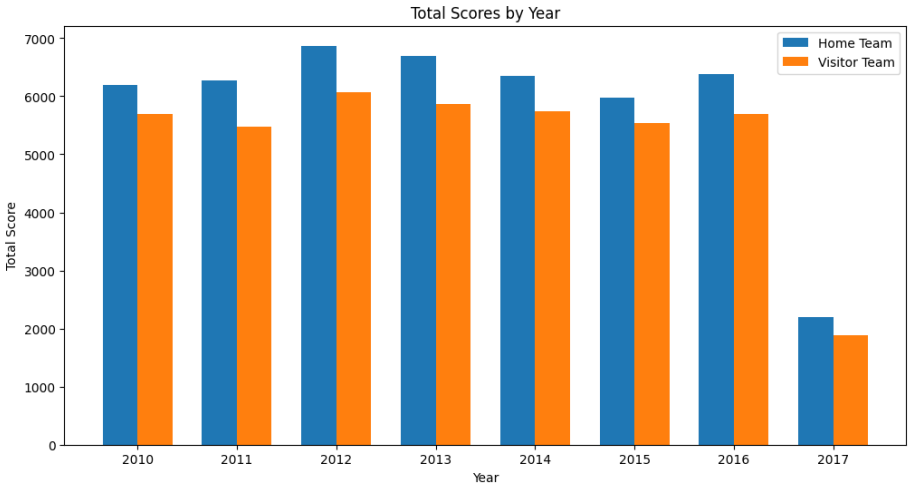
- We realized that it would be best if we joined the box scores dataset with the game results dataset, so that we get more information and features for each game that we could build a model off of
- Before merging, we still had a little cleaning we had to do in order to make sure that each dataset and the same features (in the same format) to merge on
- We inspected all of our data and decided which teams names were missing/incorrect and fixed/added them in a dataframe
- Most importantly, we created a function called `get_team_id` which pulled the team id from the team dataframe applied it to the box scores and game data dataframes to create new columns for the home id and the away id
  - This ensured that we had a common key to join the dataframes on

Joining The Data After the initial set up of the data was done, our next step was to join the box score dataset with the game data dataset.

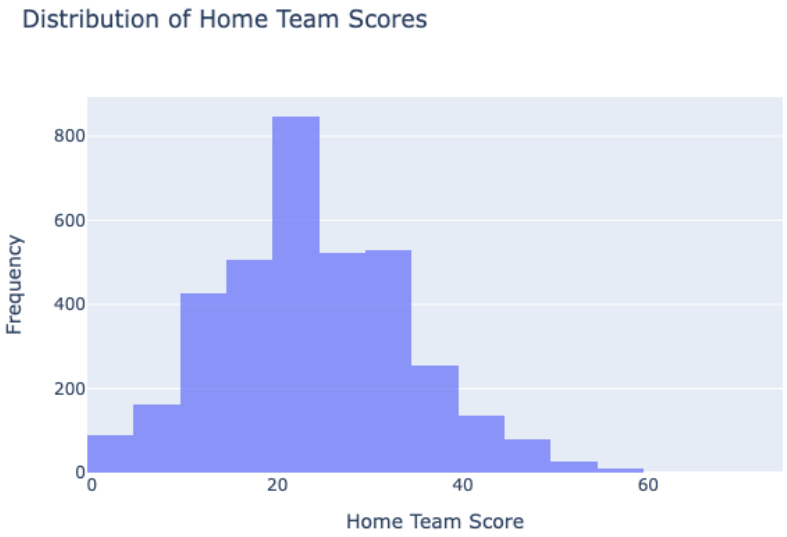
- we used inner join, so we only got the games that were present in both datasets
- we joined, so that we had more variables for each game that we could train our model on
- we then dropped repeating columns and columns we knew we were not going to use to train (i.e. duplicate score columns, duplicate date columns, and duplicate team names)
- we then added a columns for the winning team (0 if the home team won, 1 if the away team won, and 2 if the game resulted in a tie)
- we also converted all the numeric / quantitative features that we will be using to integers and floats (to build the model)

Visualizations

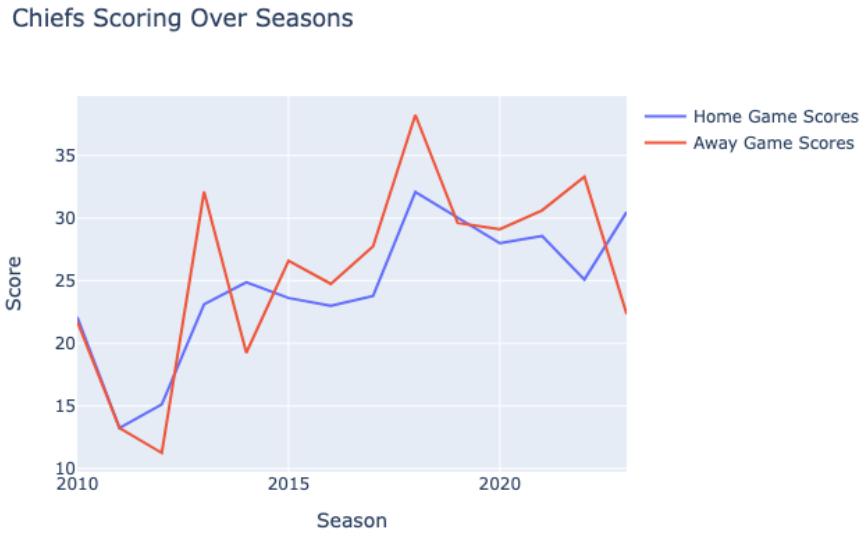
- Total Scores by Year: created this to see how the total scores have changed throughout the years



- Distribution of Scores: shows the distribution of scores for home teams



- Chiefs scoring over the Seasons for home and away games



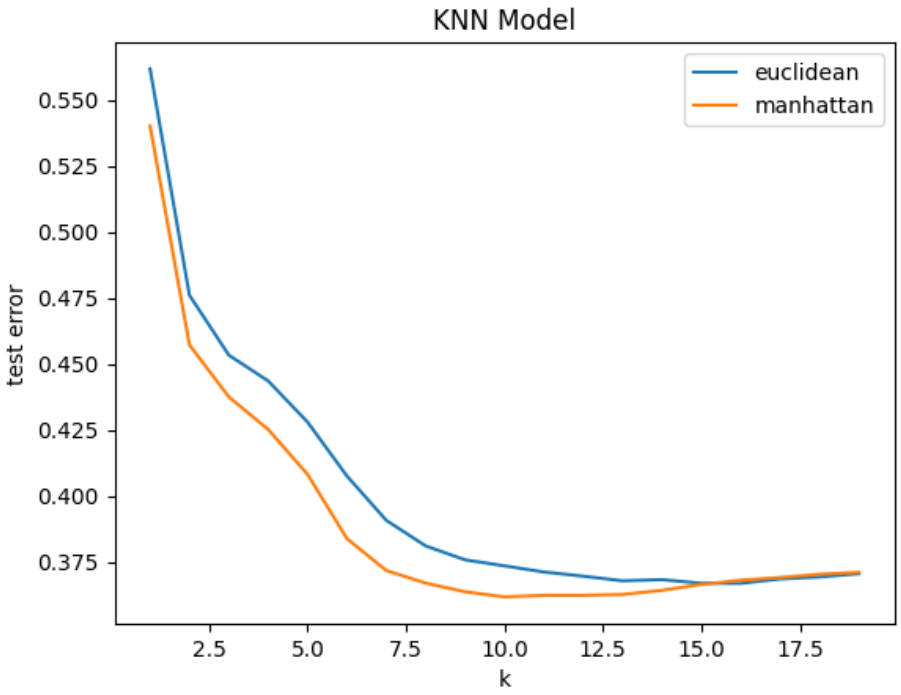
- We used these plots above to determine which variables we wanted to use to train our model.
- We found out that we should include various models to see which one would be the best to use.
- We came to this conclusion because we saw that there is an impact on the score when a team is home or away.

Machine Learning

Explain Model

The next step was to train a model. The model that we chose to go with was a KNN Regressor, since we wanted to predict who would win a game given net yards, time of possession, and scores for both home and away teams.

- First we transformed the ids of the teams to become one hot encoded since they are categorical variables.
- Next we made a pipeline with a standard scaler and a KNN Regressor with 5 neighbors.
- To get the best k value, we used a grid search with cross validation.
  - For the scoring, we chose the negative mean squared error, since we wanted to minimize the error.
  - We used 10 folds for the cross validation, but we intend to model folds vs error to see if we can get a better k value.



- After getting the best k value of 10 with the manhattan distance metric as the best metric, we trained the model on the training data.
- We then used the model to predict the outcome of the test data.
- The predictions were float values, so we rounded them to the nearest integer.
  - From the predictions, the model predicted 340 results correctly out of 417 games.

We then realized that our model was more of an "explaining" model rather than a predicting model as we used in-game statistics to predict who won that game itself.

- this model would not work for predicting future games as we would need in-game statistics for those games (which we don't have as they are future games)
- so we worked on creating a new model using past games to predict a future game

We can draw some conclusions from the "explain" model:

- Since using a team's net yards and time of possession "explained" 340 results out of 417 games correctly, it seems as though there is a correlation between a teams net yards, their time of possession, and if they win or not
  - It seems as though more net yards correlate with winning games

- However, it makes sense for the other results to not be "explained" properly as net yards and time of possession is only a small part of the game
  - One team may get more yards than the other team, but it does not matter if they are not able to score as consistently as the other team

Predict Models

Training Models

- To effectively predict the winner of games and their scores, we decided to make a function that gets the last n games of each team and then uses those games to predict the outcome of the next game.
- `get_last_n_games()` as well as `create_last_n_games_stats_df()` were used to get the last n games of each team and then create a dataframe with the statistics of those games.
  - These functions were applied to our merged dataset to get the last n games of each team.
- Statistics that we deemed important in training out model were the scores, points allowed, time of possession, first downs, total plays, and average gain for each the home and away team.

KNN Regressor: Multi Output Model

- We decided to use a KNN Regressor as our model since we wanted to predict the scores of each team and then use those scores to predict the winner of the game.
- Our `last_n_games_stats_df` was our X.
  - These were split into training and testing sets.
- We used a multi output model to predict the scores of each team, so our model would be predicting 2 values.
- We used a column transformer to encode the categorical variables, which were the team ids.
  - Then we used a pipeline with a standard scaler and a KNN Regressor with 5 neighbors.
- To find the best k value, we used a grid search with cross validation.
  - For the scoring, we chose the negative mean squared error, since we wanted to minimize the error.
  - For the metrics, we used the manhattan distance and the euclidean distance.
  - We used 10 folds for the cross validation to get the best k value.
- Finally we predicted the scores of the test data and added them to the test data dataframe.
- We then used the predicted scores to predict the winner of the game.
  - At the end we calculated the amount of predictions that were correct.

KNN Regressor: Home and Away Model

- To better understand the impact of being home or away, we decided to make a model that predicts the scores of each team separately.
- We used the same `last_n_games_stats_df` but split the features into home and away features.
  - We also split the target into home and away targets.
- The same methods were used such as column transformer and pipeline.
- However, there were 2 models that were fit and predicted on.
  - Each model would only predict the scores of the home team or the away team.
  - So each yielded CV errors and predictions.
- At the end, we combined the model's predictions and concatenated them to the test data dataframe.
- We then used the predicted scores to predict the winner of the game.
  - At the end we calculated the amount of predictions that were correct.

Linear Regression Model

- After making the KNN model, we wanted to compare it with a Linear Regression model to see if Linear Regression would be a better way to predict scores for the teams and who would win.
- We used the same `last_n_games_stats_df` for our X train and the game scores for the y train
- We used the same method for the column transformer (in order to encode the team ids) and used that with a new pipeline with a Linear Regression model instead.
  - We still used the Standard Scaler in order to scale the quantitative values as the features units and ranges are different
- We fit the model with the X train and then predicted the game scores.
  - we then used those predicted values to find our predicted winner for the game and calculated the amount of predictions that were correct.

Voting Ensemble Model

- We decided to use a voting ensemble model to combine the predictions of the Linear Regression Model and the KNN Regressor: Multi Output Model to better predict the winner of the game.
- To do this, we used the `VotingRegressor` from sklearn, however we had to use `MultiOutputRegressor` since we are predicting 2 values.
- To get the best weight, we looped through a range of weights and then found the weight that yielded the best CV error.
- Then we used the best weight in a final model that would predict the scores of the test data.
- We then used the predicted scores to predict the winner of the game.
  - At the end we calculated the amount of predictions that were correct.

Metrics and Results

- To better understand the results of our models, we calculated the cross validation error for each model.
  - We used the negative mean squared error as our scoring metric.
- In addition, we calculated the amount of predictions that were correct for each model as the accuracy metric.
- However, we wanted to understand the impact of the last n games on the accuracy and CV errors, and to see if changing n would improve our models.
  - To achieve this we created a function for each model that take in a dataframe and a train/test split.
  - Each of these functions train and test the model as we described above.
  - The only difference is that these functions would return 4 values: the model itself, the CV error, the number of predictions that were correct, and the total number of predictions.
- Using these functions, we looped through a range of n values (from 1 to 10) and then added the outputs into a dataframe.
  - We then decided to go with the best model that yielded the best accuracy, which was the **KNN Regressor: Multi Output Model**.

Here are the the 5 best and 5 worst configurations for our models in the final dataframe:

	Last n Games	Model Name	CV RMSE	Correct Predictions	Total Predictions	Accuracy
16	5	KNN Multi Output	9.831835	195	299	65.217391
8	3	KNN Multi Output	9.882861	217	334	64.970060
35	9	Ensemble Voting	9.708033	154	238	64.705882
32	9	KNN Multi Output	9.748610	153	238	64.285714
0	1	KNN Multi Output	9.923542	234	366	63.934426
...	...	...	...	...	...	...
26	7	Linear Regression	9.848474	157	267	58.801498
30	8	Linear Regression	9.835840	147	252	58.333333
22	6	Linear Regression	9.884158	162	282	57.446809
6	2	Linear Regression	9.988611	200	350	57.142857
2	1	Linear Regression	10.007334	209	366	57.103825

Conclusions

From these results, we can draw some conclusions:

- The best model was the **KNN Regressor: Multi Output Model** with a CV error of 9.83 and an accuracy of 65.22
  - This model was trained on the last 10 games of each team.

- The worst model was the **Linear Regression Model** with a CV error of 10.007 and an accuracy of 57.10
  - This model was trained on the last 1 game of each team.
- NOTE: All models that we tested had an accuracy over 50% meaning that they predicting the winners correctly more than half of the time.
  - Therefore, features such as the scores, points allowed, time of possession, first downs, total plays, and average gain for each team have some correlation with the winner of the game but may not be the best features to use to predict the exact scores of the teams.
- The CV errors for most of these models ranged from 9-10.
  - These values could be improved by either choosing different features or reducing the number of features used.

Predicting Model: KNN Regressor Multi Output Model Once we found which model was the best (by testing out different n values and different regression models), we wanted to apply that to some current games to see if our model is still appropriate for the 2023 NFL Season.

So, we tested our model against 2 games that took place on Sunday, Dec. 10, 2023: the 49ers vs Seahawks game and the Cowboys vs Eagles game.

Since our best model uses averages from the previous 5 games for a team, we researched to find the stats we needed from each of the teams last 5 games, averaged the values out, and then added it to its own dataframe that we could use on model on.

49ers vs Seahawks Game Here is a table of our predictions for score compared to actual score:

Team	Predicted Score	Actual Score
SF	22.82	28
SEA	17.58	16

Here is a table of our predicted winner compared to the actual winner:

Predicted Winner	Actual Winner
SF	SF

Cowboys vs Eagles Game Here is a table of our predictions for score compared to actual score:

Team	Predicted Score	Actual Score
DAL	27.06	33
PHI	24.68	13

Here is a table of our predicted winner compared to the actual winner:

Predicted Winner	Actual Winner
DAL	DAL

Results It looks like our model correctly predicted who was going to win each of these two games, however, the points values were definitely off. This means that the features we used to build the model (i.e. last 5 games average net yards for both teams, last 5 games average first downs for both teams, last 5 games time of possession for both teams, last 5 games total plays for both teams, and last 5 games points scored/allowed for both teams) definitely helped to predict the winner at a somewhat high accuracy, but was not as useful to predict the exact scores for the games.

We would definitely need to consider more features and variables in order to predict the winner at a higher accuracy as well as the exact point totals such as weather, key player injuries, play types/schemes, etc. This is maybe something to consider for the future if we wanted to improve our model.