Kendall Andrews
Tanjanay Hardy

April 25, 2025

# Final Project Report

Kendall Andrews – Data Preprocessing, Feature Selection, Visualizations, XG Boost, and PowerPoint Presentation

Tanjanay Hardy – Data, LASSO, Linear Regression, Model Discussion, and PowerPoint Presentation

## a. Problem Definition

The primary objective of this project,  is to predict game outcomes by determining the winner between the home team (X) and the away team (Y). The model will leverage historical game data and player statistics to enhance prediction accuracy.

## b. Analysis, Solution and Results

## Analysis

Our dataset consists of two key sources:

1. *Game-by-game,* game by game weekly player statistics
2. *TeamScores,* seasonal weekly schedules, including team outcomes (win/loss)

These datasets were merged using team abbreviation and week number to form a comprehensive dataset covering all 18 weeks of the season.

- Data Types: The dataset includes categorical (object), integer (int64), and float (float64) variables.
- Handling Missing Data: Identified and addressed missing values where necessary.
- Duplicate Removal: Checked for and eliminated duplicate entries.
- Feature Selection & Removal: Removed irrelevant features to optimize model performance.
- Encoding: Applied label encoding to the outcome variable, converting wins/losses into binary values.

- Additional Calculations:
- Mean and Z-score calculations for numerical features.
- Outlier detection using Z-scores to identify extreme values.

### Feature selection

**Passing Metrics:**

1) Completions
2) Attempts
3) Passing yards
4) Passing air yards
5) Passing yards after catch
6) Passing first downs
7) Passing touchdowns
8) Interceptions

**Sack & Fumble Metrics:**

9) Sacks
10) Sack yards
11) Sack fumbles
12) Sack fumbles lost

**Rushing Metrics:**

13) Carries
14) Rushing yards
15) Rushing first downs
16) Rushing fumbles
17) Rushing fumbles lost

**Receiving Metrics:**

18) Receptions
19) Receiving yards
20) Receiving touchdowns
21) Receiving fumbles
22) Receiving fumbles lost
23) Receiving air yards
24) Receiving yards after catch
25) Receiving first downs
26) Receiving expected points added (EPA)

**Advanced Metrics:**

27) Target share
28) Air yards share
29) Weighted Opportunity Rating (WOPR)

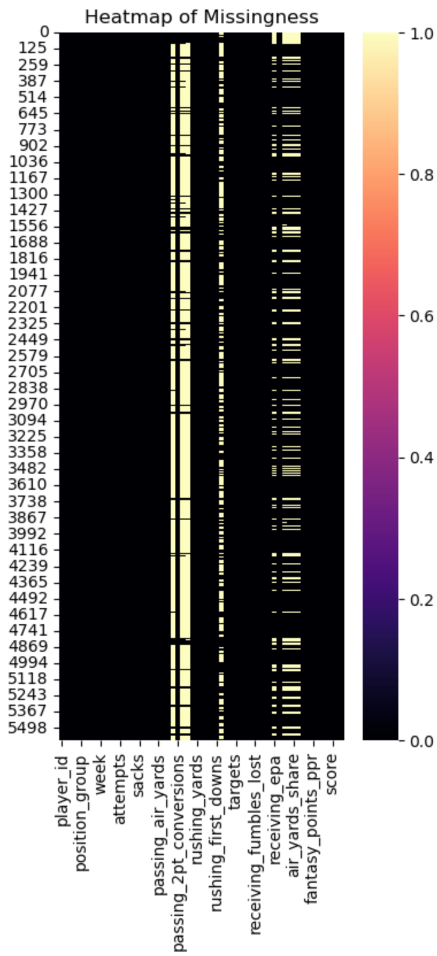
**Outcome Variable:**

30) Score

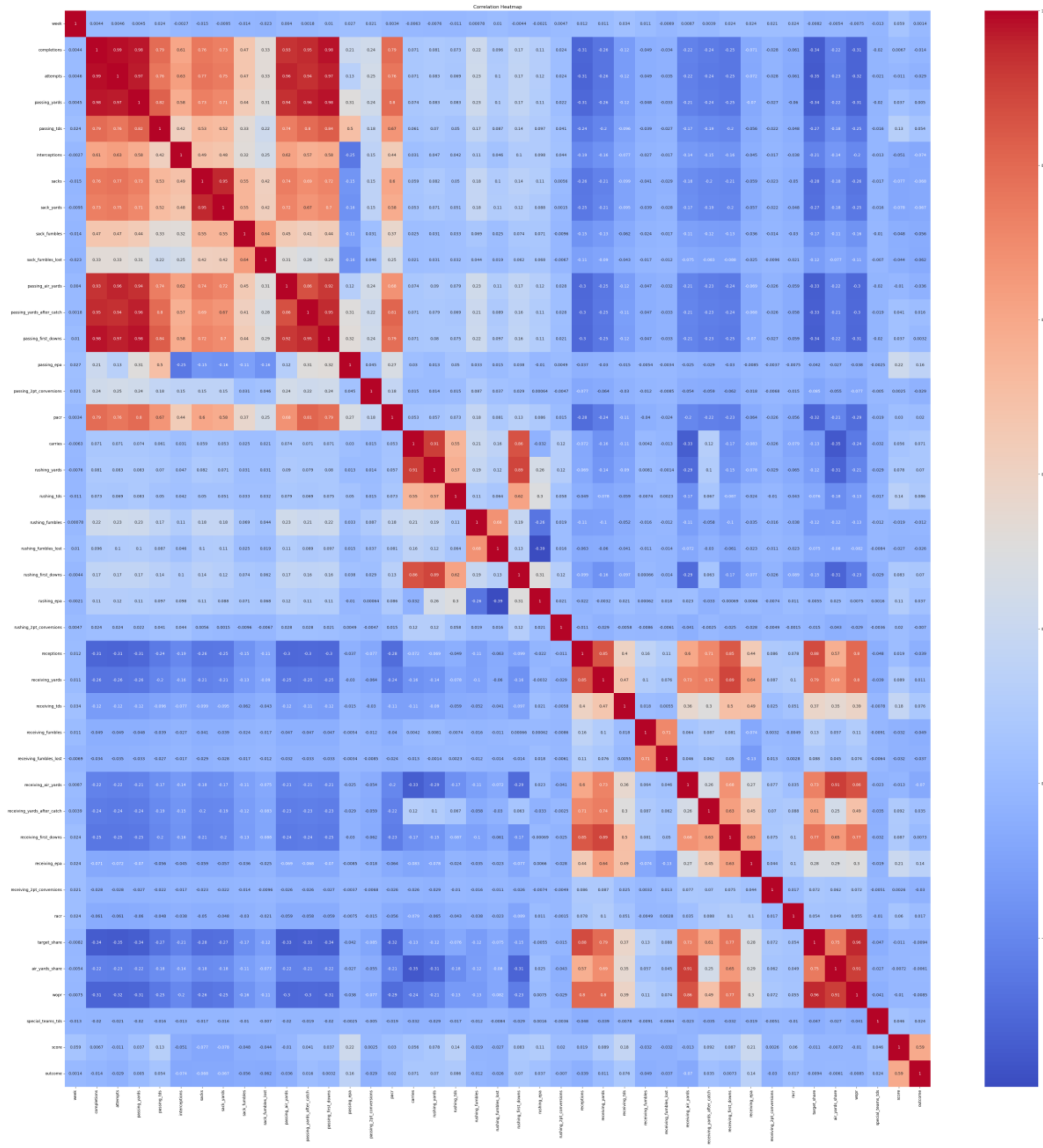
31) Game outcome (binary: win/loss)

## Visualizations

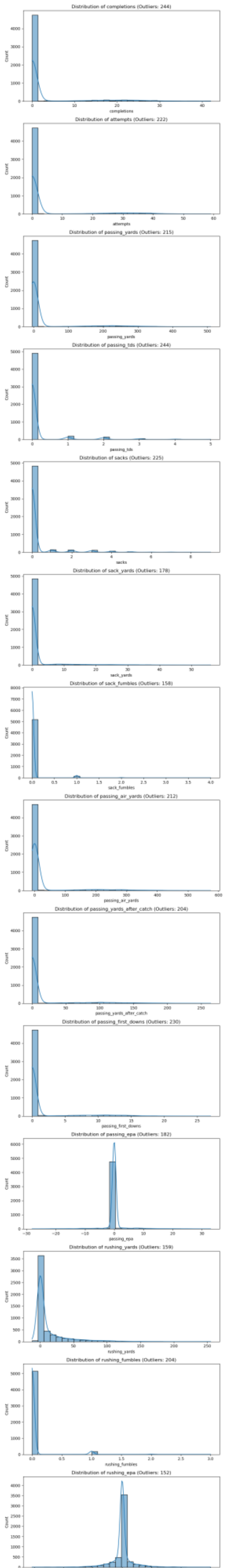
**Missingness Heatmap –** Identified and visualized features that have missing values



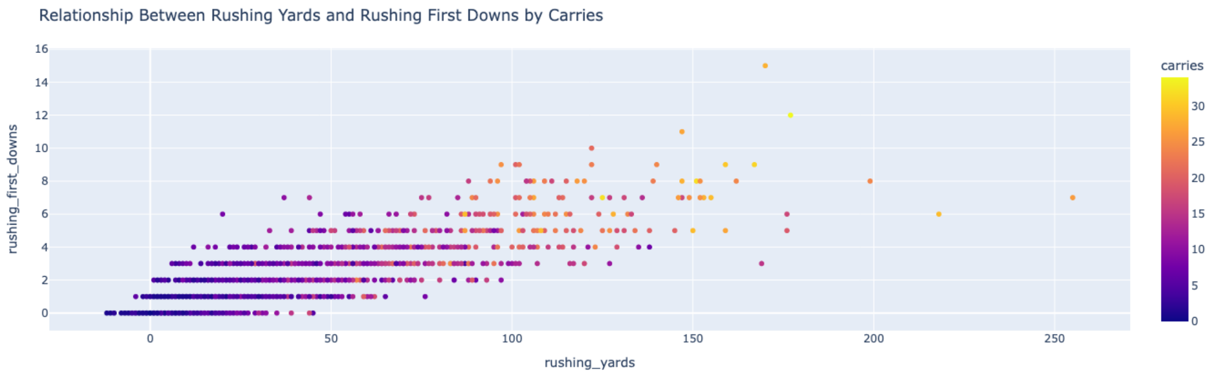

**Correlation Heatmap –** Identified **25-30 features** with strong correlations to game outcomes.

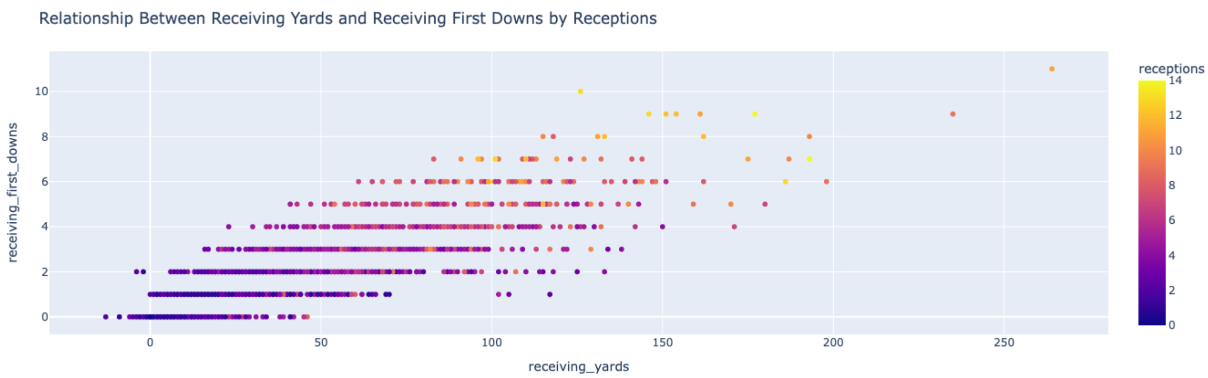


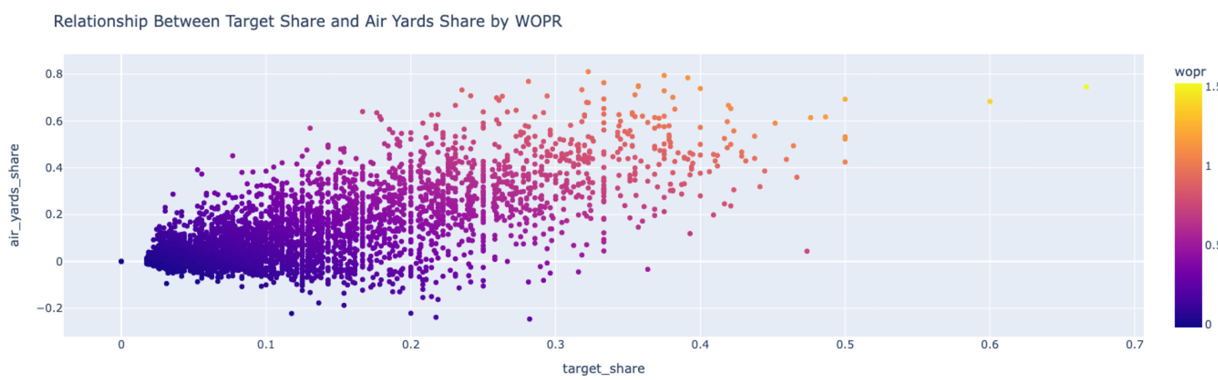
**Bar Plots –** Highlighted features with **over 150 outliers**, indicating potential data skews or key influential variables.


Distribution of completions (Outliers: 244)
Count
completions
Distribution of attempts (Outliers: 222)
Count
attempts
Distribution of passing_yards (Outliers: 215)
Count
passing_yards
Distribution of passing_tds (Outliers: 244)
Count
passing_tds
Distribution of sacks (Outliers: 225)
Count
sacks
Distribution of sack_yards (Outliers: 178)
Count
sack_yards
Distribution of sack_fumbles (Outliers: 158)
Count
sack_fumbles
Distribution of passing_air_yards (Outliers: 212)
Count
passing_air_yards
Distribution of passing_yards_after_catch (Outliers: 204)
Count
passing_yards_after_catch
Distribution of passing_first_downs (Outliers: 230)
Count
passing_first_downs
Distribution of passing_epa (Outliers: 182)
Count
passing_epa
Distribution of rushing_yards (Outliers: 159)
Count
rushing_yards
Distribution of rushing_fumbles (Outliers: 204)
Count
rushing_fumbles
Distribution of rushing_epa (Outliers: 152)
Count

**Scatterplots –** Analyzed relationships among **three highly correlated features**, providing insights into their interactions.




## Solution

1. LASSO Regression – A linear model that applies regularization by shrinking less important feature coefficients to zero, effectively selecting only the most meaningful predictors. It is suitable for predicting continuous outcomes, such as score.
2. Logistic Regression – A linear classification algorithm that estimates the probability of a binary outcome, making it well-suited for predicting win/loss scenarios.
3. XG Boost – A powerful non-linear gradient boosting algorithm that builds decision trees using a loss function to evaluate model performance. It is effective for both classification and regression tasks, particularly when the data has complex patterns.

**Results**

- LASSO
    a. Performance Stats

    ```
    LASSO Training MSE: 82.5279
    LASSO Test MSE: 84.5517
    R² score on test set: 0.1545
    ```

    b. Feature of Importance

```
Significant Features in LASSO:
 receiving_epa                   1.646931
passing_tds                     1.606100
passing_epa                     1.445790
rushing_tds                     1.302056
receiving_tds                   1.207005
rushing_epa                     0.706202
receiving_yards                 0.655407
special_teams_tds               0.552045
receiving_yards_after_catch     0.435670
week                            0.427243
racr                            0.312292
interceptions                   0.184405
carries                         0.030683
rushing_2pt_conversions         0.023545
passing_yards_after_catch       0.010972
rushing_fumbles                -0.023909
air_yards_share                -0.028696
receiving_2pt_conversions      -0.053811
receiving_fumbles              -0.074460
pacr                           -0.101113
receiving_air_yards            -0.355589
sacks                          -0.436256
passing_air_yards              -0.641032
completions                    -0.789505
...
```

- Logistic Regression
    c. Performance Stats

```
 Accuracy: 0.6161
 F1 Score: 0.6359
 Recall: 0.6549
 Precision: 0.6180
```

d. Feature of Importance

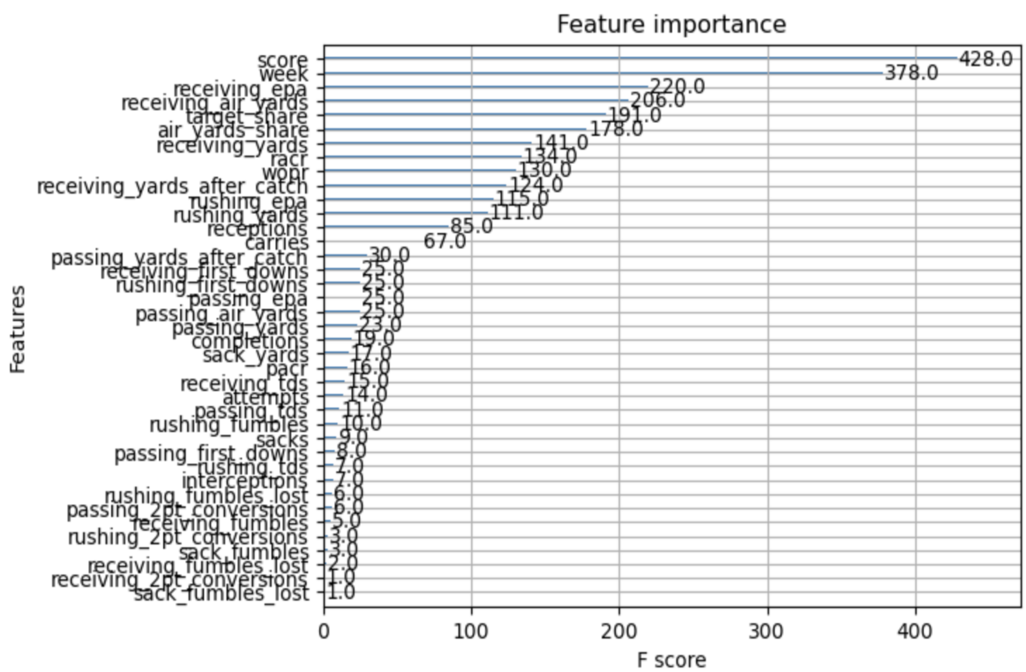
```
Logistic Regression Coefficients:
                          Feature  Coefficient
32                  receiving_epa     0.530657
35                   target_share     0.486793
37                           wopr     0.429338
3                   passing_yards     0.400287
11    passing_yards_after_catch      0.333370
13                    passing_epa     0.287991
36                 air_yards_share    0.284714
25                receiving_yards     0.183341
16                        carries     0.169336
7                      sack_yards     0.167744
4                     passing_tds     0.161356
18                    rushing_tds     0.148777
28        receiving_fumbles_lost     0.118343
22                    rushing_epa     0.087375
26                  receiving_tds     0.063483
38              special_teams_tds     0.057221
34                           racr     0.030902
0                            week     0.009733
21            rushing_first_downs     0.002501
5                   interceptions    -0.012235
20            rushing_fumbles_lost   -0.021072
19                rushing_fumbles    -0.032875
```

- XG Boost

e. Performance Stats

```
XGBoost Model Evaluation Metrics:
✅ Accuracy:  0.81
📌 F1 Score:  0.81
🔁 Recall:    0.81
🎯 Precision: 0.82
```

f. Feature of Importance




**c. Further discussion for the result and lessons you learned**

**Discussion:**

Our XGBoost model (a non-linear algorithm) performed the best, achieving 81% accuracy. The training error was 0.075 and the test error was 0.1891, indicating reliable model performance with both errors below 20%. However, a ~11% gap between training and test error suggests some minor overfitting.

In contrast, the LASSO regression and Logistic regression models underperformed. LASSO regression yielded training and test mean squared errors (MSE) above 80 and an $R^2$ of just 0.15, indicating severe underfitting. The Logistic regression model achieved 62% accuracy, with a training error of 0.3587 and test error of 0.3839, also suggesting underfitting—but with consistent performance between training and test data.

As expected, **‘**score**’** emerged as the most important feature in the XGBoost model, which makes sense given that it directly reflects game outcomes. However, in the LASSO and Logistic regression models, **‘**score**’** was used as a target, introducing bias and likely contributing to poor performance. This suggests that the dataset may be better suited for non-linear algorithms.

Interestingly, all three models identified ‘receiving_epa**’** as a top 3 feature, highlighting the importance of receiver performance in determining game outcomes. For example, a big reception or touchdown can drastically increase the chances of winning a game.

**Lessons Learned:**

- XGBoost was effective, but we plan to explore other non-linear models such as Support Vector Machines (SVM)and K-Nearest Neighbors (KNN).
- We want to remove ‘score’ as a predictor and reassess model performance to reduce bias.
- Future improvements include hyperparameter tuning using grid search to optimize model performance.