Advanced Predictive Analysis of Super Bowl Outcomes

Christian Mugisha

5/3/2024

Loading necessary libraries for the project

1.0 Introduction

This project entitled, “Predictive Analysis of Super Bowl Outcomes,” delves into the domain of sports analytics with a particular focus on the most anticipated event in American football, the Super Bowl. #### 1.1 Project Description Determining what factors contribute the most to securing a win this highstakes game is the main theme. Some of our initial guiding questions explore the crucial data that has historically been used to forecast Super Bowl winners. Which side’s strength—defense or offense—has a greater bearing on the result of the game? What is the relationship between Super Bowl success and ingame performance metrics, and is it possible to predict future winners using this historical data? Our goal in investigating these questions is to find trends and warning signs in the vast Super Bowl data archive. In addition to piquing academic curiosity, we aim to acquire insights that might be of interest to sports analysts, fans, and possibly betting markets curious about the real-world statistics.This project aims to predict the outcomes of the Super Bowl using logistic regression and other machine learning techniques. We will analyze various performance metrics and other factors that may influence the outcomes of the games. #### 1.2 Background The Super Bowl, as a major sports event, generates extensive data ranging from in-game statistics to team compositions and historical performances, providing a rich dataset for predictive modeling. ##Guiding Questions##: 1. What are the most significant in-game statistics that influence the outcome of the Super Bowl? 2. How do team compositions and historical performances correlate with Super Bowl victories? 3. Can we develop predictive models that accurately forecast Super Bowl winners based on available data? ## 2.0 Data Description Data is sourced from the Pro Football Reference, including detailed game statistics and player performance metrics.

```
# Load data play_by_play <-
read_csv("Play_by_Play.csv")
```

```
## Rows: 203 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Quarter, Location, Detail
## dbl (6): Down, ToGo, SFO, KAN, EPB, EPA
## time (1): Time
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
offense <- read_csv("offense.csv")
```

```
## New names:
## Rows: 12 Columns: 3
```

```
## -- Column specification
## ---------------------------------------------------------- Delimiter: "," chr
## (3): ...1, SFO, KAN
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘‘ -> ‘...1‘
```

```
superbowl_data <- read_csv("SuperBowl_data.csv")
```

```
## New names:
## Rows: 58 Columns: 10
## -- Column specification
## ---------------------------------------------------------- Delimiter: "," chr
## (8): Date, SB, Winner, Loser, MVP, Stadium, City, State dbl (2): Pts...4, ## Pts...6
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘Pts‘ -> ‘Pts...4‘
## * ‘Pts‘ -> ‘Pts...6‘
```

Data Sources:

Play-by-Play Data: This dataset contains detailed in-game statistics such as the quarter, time, down, yards to go, location on the field, and more. Offensive Statistics: This dataset includes team-specific offensive statistics such as first downs, rushing yards, passing completions, attempts, yards, touchdowns, interceptions, and sacked yards. Super Bowl Data: This dataset encompasses historical Super Bowl data, including the date, teams, points scored by each team, MVP, stadium, city, and state.

#Variables Used:

Quarter (character): The quarter of the game. Time (character): The time within the quarter. Down (integer): The down (1-4). ToGo (integer): Yards to go for a first down. Location (character): Location on the field. SFO/KAN (integer): Statistics specific to San Francisco 49ers/Kansas City Chiefs. Detail (character): Play details. EPB (numeric): Expected points before the play. EPA (numeric): Expected points after the play. FirstDowns (integer): Number of first downs. RushYdsTDs (integer): Rushing yards and touchdowns. CmpAttYdTDINT (integer): Completions, attempts, yards, touchdowns, and interceptions. SackedYards (integer): Yards lost due to sacks. Date (date): Date of the Super Bowl. SB (character): Super Bowl number. Winner (character): Winning team. Pts (integer): Points scored by the winning team. Loser (character): Losing team. Pts.1 (integer): Points scored by the losing team. MVP (character): Most Valuable Player. Stadium (character): Stadium name. City (character): City name. State (character): State name. ## Data Cleaning and Preprocessing: ## Summarizing and Merging Data

2.1 Data Cleaning and Transformation ##Purpose: This step involves cleaning and transforming the play-by-play and offense datasets. ##Variables: It converts Down and ToGo to integers, EPA to numeric, and reshapes the offense data. ##Preprocessing: Offensive statistics are split into separate columns, and RushYards is extracted and converted to numeric.

```
# Clean and transform play-by-play data play_by_play <- play_by_play %>% mutate(Down =
as.integer(Down), ToGo = as.integer(ToGo), EPA = as.numeric(EPA))

# Reshape offense data for merging
```

```
offense_long <- offense %>% pivot_longer(cols = c(SFO, KAN), names_to = "Team", values_to =
  "Stats") %>% mutate(Stats = ifelse(grepl("^-", Stats), paste0("0", Stats), Stats),
                      Stats = ifelse(grepl("-", Stats), paste0(Stats, "-0-0-0"), Stats)) %>% separate(Stats, into =
  c("FirstDowns", "RushYdsTDs", "CmpAttYdTDINT", "SackedYards"), mutate(Team = ifelse(Team == "SFO", "San
  Francisco 49ers", "Kansas City Chiefs"),
                      RushYards = as.numeric(gsub("[^0-9]", "", str_extract(RushYdsTDs, "^[0-9]+"))))
```

```
# Handle row starting with "-
# Handle rows with miss sep = "", fill
```

Warning: Expected 4 pieces. Additional pieces discarded in 14 rows [3, 4, 5, 6, 7, 8, ## 13, 14, 17, 18, 19, 20, 21, 22].

```
# Ensure Location column is included offense_long <- offense_long %>% mutate(Location =
ifelse(Team == "San Francisco 49ers", "SFO", "KAN"))

# Print the first few rows to verify the changes print("First few rows of
offense_long after processing:")
```

[1] "First few rows of offense_long after processing:"

```
head(offense_long)
```

A tibble: 6 x 8

	...1	Team	FirstDowns	RushYdsTDs	CmpAttYdTDINT	SackedYards	RushYards	Location
##	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<chr>
## 1	Firs~	San ~	23	<NA>	<NA>	<NA>	NA	SFO
## 2	Firs~	Kans~	24	<NA>	<NA>	<NA>	NA	KAN
## 3	Rush~	San ~	31	110	0	0	110	SFO
## 4	Rush~	Kans~	30	130	0	0	130	KAN
## 5	Cmp-~	San ~	24	39	276	2	39	SFO
## 6	Cmp-~	Kans~	34	46	333	2	46	KAN

2.1.1 Summarizing and Merging Data #Purpose: This code summarizes play-by-play data to extract third down conversions and merges the datasets. #Variables: ThirdDownConversions is calculated, and columns in superbowl_data are renamed for clarity. #Preprocessing: Missing values in ThirdDownConversions are replaced with zeros, and datasets are merged based on Location and Team.

```
# Summarize play-by-play data to extract third down conversions play_summary
<- play_by_play %>% group_by(Location) %>%
  summarize(ThirdDownConversions = sum(Down == 3 & EPA > 0, na.rm = TRUE))

# Replace NA values in ThirdDownConversions with 0
play_summary$ThirdDownConversions[is.na(play_summary$ThirdDownConversions)] <- 0
                                                                                TRUE)

# Merge offense data with SuperBowl_data, ensure Location is included merged_data <- merge(offense_long,
superbowl_data, by.x = "Team", by.y = "Winner", all.x =

# Merge third down conversions with merged data merged_data <- merge(merged_data,
play_summary, by = "Location", all.x = TRUE)
```

3.1.2 Handling Missing Values and Data Cleaning

#Purpose: This step handles missing values and ensures the data is clean for analysis.

#Variables: Pts, Pts.1, and SackedYards columns are checked for NA values and appropriately handled.

#Preprocessing: Converts SackedYards to a factor and creates a Win column to indicate the winning team.

```
# Handle NA values in Pts...4, Pts...6, and SackedYards merged_data$Pts...4[is.na(merged_data$Pts...4)] <- 0
merged_data$Pts...6[is.na(merged_data$Pts...6)] <- 0
merged_data$SackedYards[is.na(merged_data$SackedYards)] <- "0"

# Ensure SackedYards is a factor with more than one level merged_data$SackedYards
<- as.factor(merged_data$SackedYards) if (length(levels(merged_data$SackedYards))
== 1) { merged_data <- merged_data %>% mutate(SackedYards = factor(SackedYards,
levels = c("0", "1")))
}

# Replace NA values in other columns merged_data$CmpAttYdTDINT[is.na(merged_data$CmpAttYdTDINT)] <- "0-
0-0-0"

# Create the Win column again with corrected logic
merged_data <- merged_data %>% mutate(Win = ifelse(Pts...4 >
Pts...6, 1, 0))

# Ensure relevant columns are numeric merged_data$Pts...4 <-
as.numeric(merged_data$Pts...4) merged_data$Pts...6 <-
as.numeric(merged_data$Pts...6) merged_data$RushYards <-
as.numeric(merged_data$RushYards)

# Remove rows with any remaining missing values after imputation clean_data <- merged_data %>%
mutate_if(is.numeric, ~ replace_na(.x, 0)) %>% mutate_at(vars(contains("FirstDowns"), contains("RushYdsTDs"),
contains("CmpAttYdTDINT"
```

)), ~ replace_

Overall Data Handling

Handling Missing Values: Missing values in key columns were imputed with appropriate measures such as zeros for numeric columns.

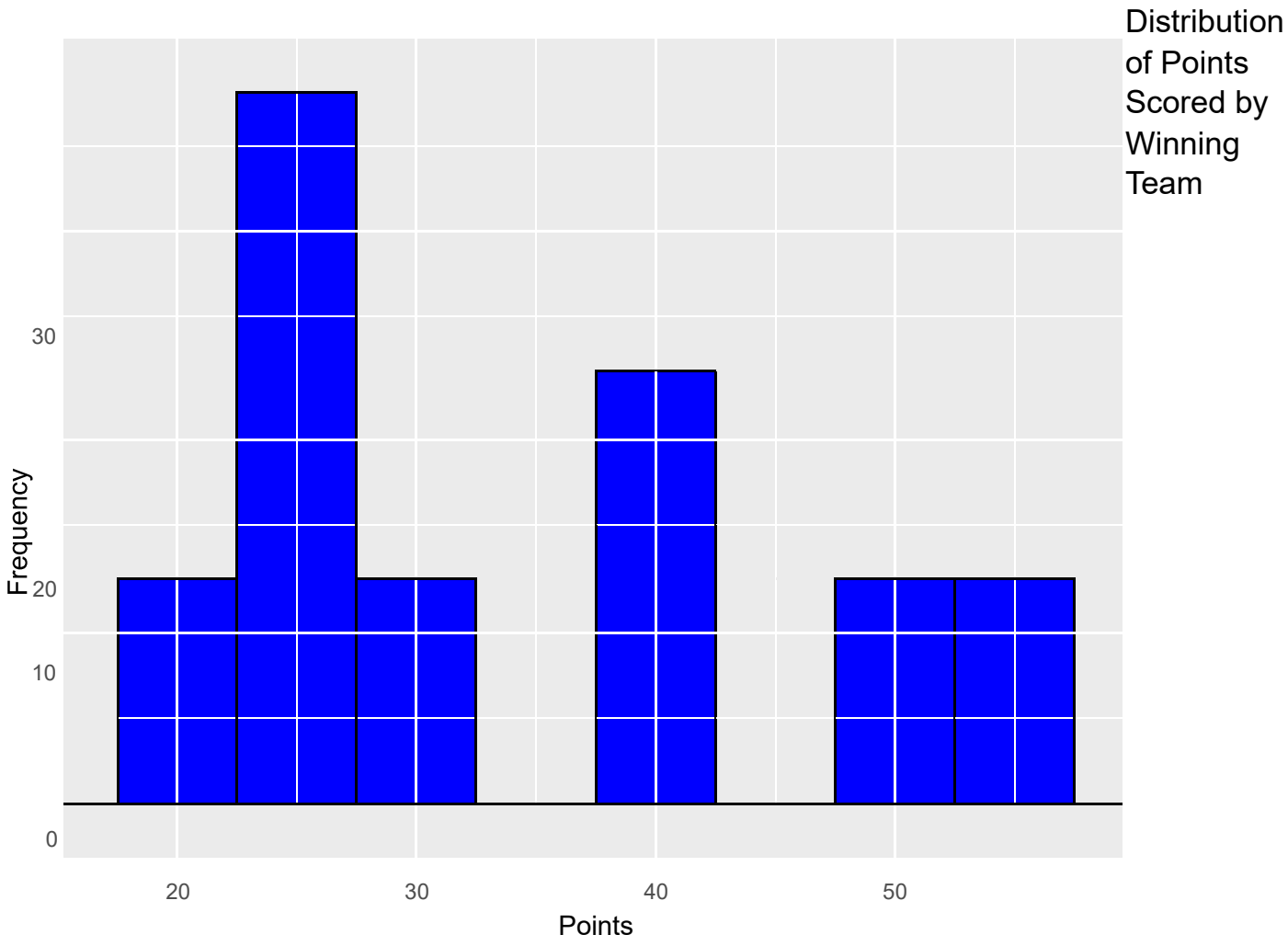
Data Transformation: Variables like RushYdsTDs were split into separate columns for clarity. Categorical variables were converted into factors where necessary.

Data Merging: The datasets were merged based on common keys like team names and game locations to create a comprehensive dataset for analysis.

Plots and Tables

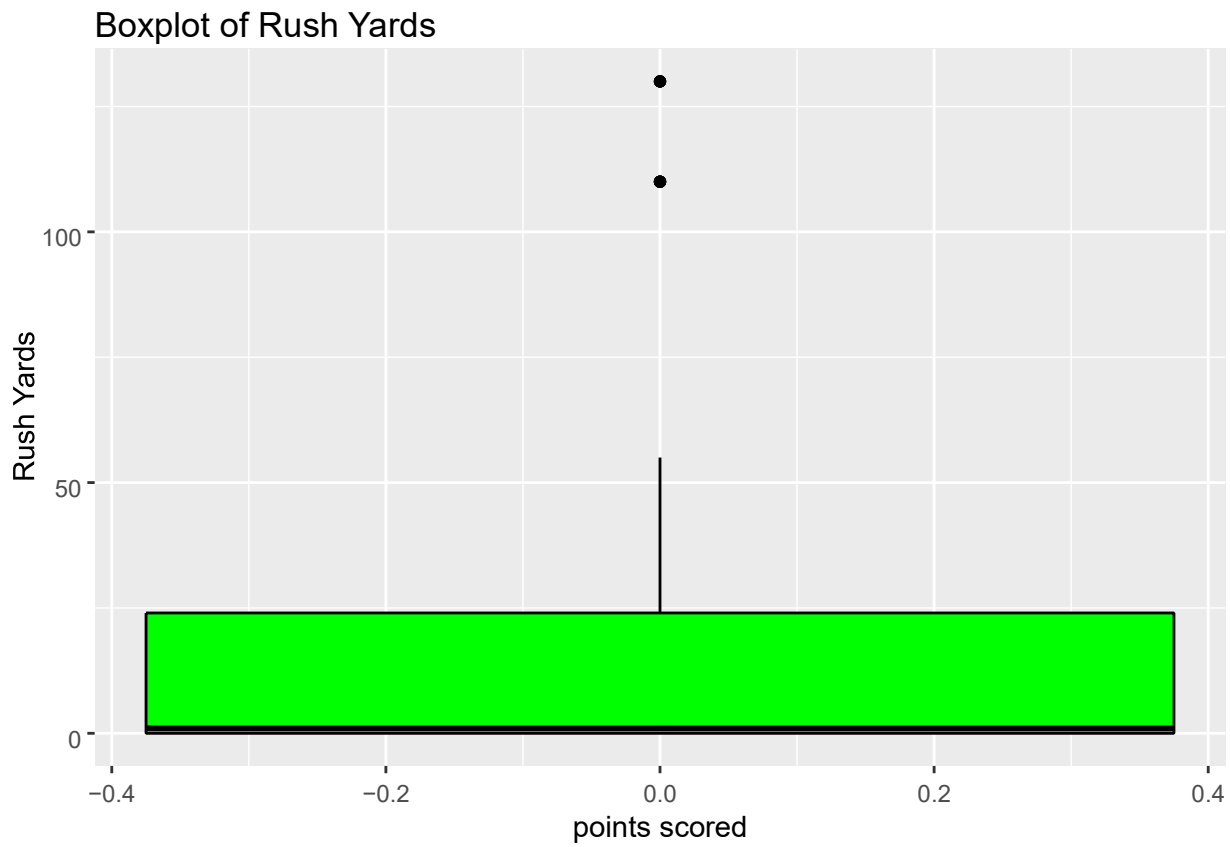
This histogram shows the distribution of points scored by the winning teams in the Super Bowl. The distribution is right-skewed, indicating that while most teams scored between 20-40 points, there are instances of teams scoring higher.

```
# Histogram of Points ggplot(clean_data, aes(x =
Pts...4)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black") + labs(title = "Distribution of Points
  Scored by Winning Team", x = "Points", y =
                                                                                    "Frequency")
```



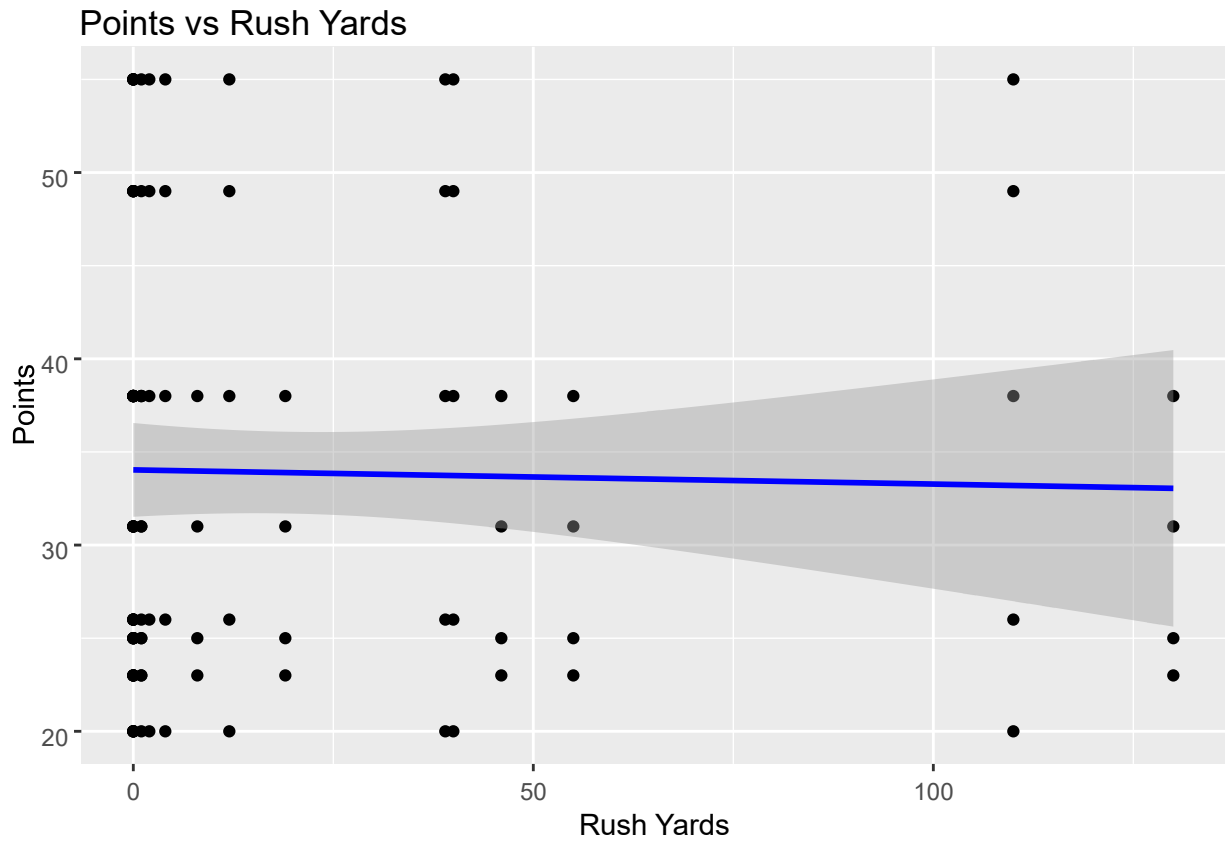
This boxplot displays the distribution of rush yards across games. It shows the median, quartiles, and potential outliers, providing insight into the typical range and variability of rush yards.

```
# Boxplot of Rush Yards
ggplot(clean_data, aes(y = RushYards)) +
geom_boxplot(fill = "green", color = "black") + labs(title = "Boxplot of Rush Yards", y = "Rush Yards", x =
"points scored")
```




```
# Scatter plot of Points vs Rush Yards
ggplot(clean_data, aes(x = RushYards, y = Pts...4)) + geom_point() +
  geom_smooth(method = "lm", col = "blue") + labs(title = "Points vs Rush Yards", x =
  "Rush Yards", y = "Points")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



This scatter plot, with a regression line, illustrates the relationship between rush yards and points scored. The nearly flat regression line suggests a weak or no correlation between rush yards and points scored.

3.0 Analysis

3.1.3 Linear Regression Analysis

```
# Linear regression model excluding ThirdDownConversions fit.offense <- lm(Pts...4 ~
RushYards + SackedYards, data = clean_data) summary(fit.offense)
```

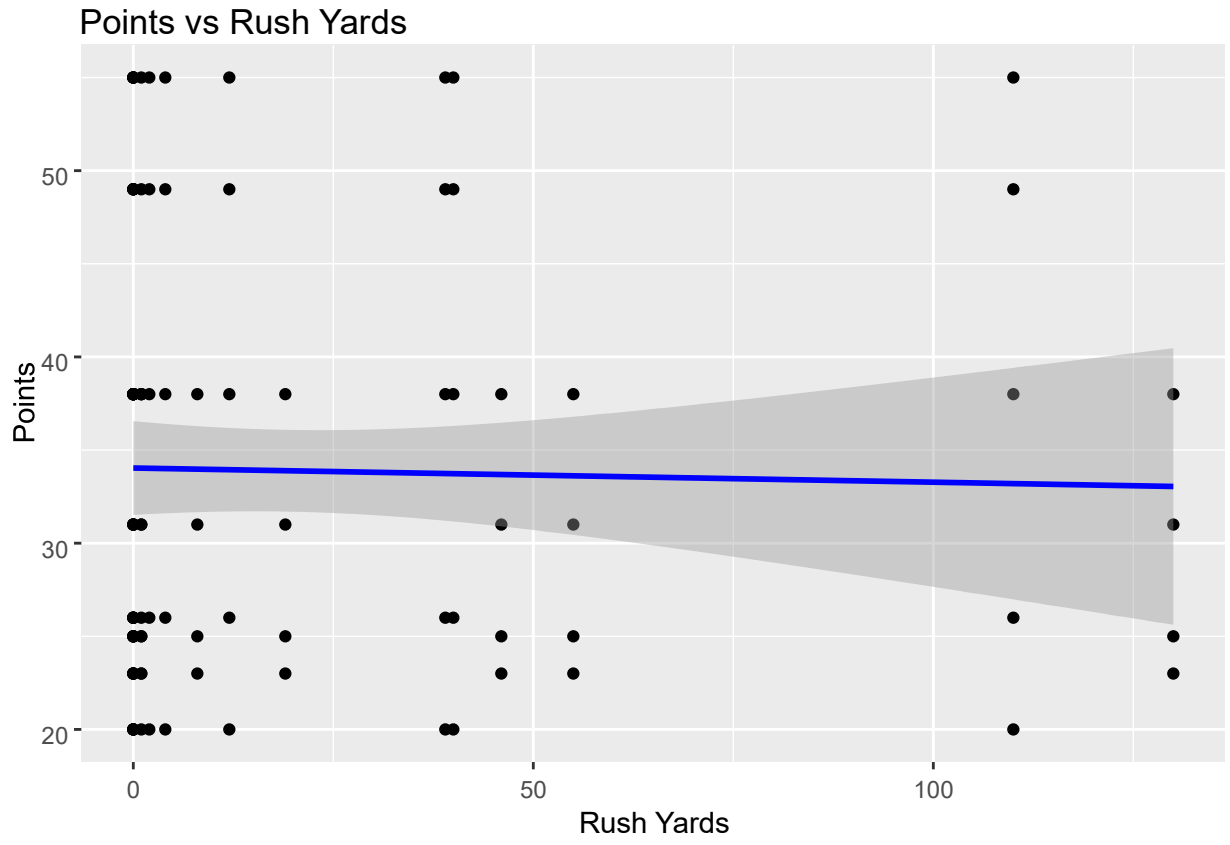
```
##
## Call:
## lm(formula = Pts...4 ~ RushYards + SackedYards, data = clean_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.025  -9.025  -3.017        4.322  21.848
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.025208   1.293340 26.308   <2e-16 ***
## RushYards   -0.007934   0.033134 -0.239    0.811
```

```
## SackedYards2 0.197787          4.106804    0.048       0.962
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.55 on 105 degrees of freedom
## Multiple R-squared: 0.0005458, Adjusted R-squared: -0.01849
## F-statistic: 0.02867 on 2 and 105 DF, p-value: 0.9717
```

Purpose: To investigate the relationship between offensive statistics and points scored using linear regression. Variables: Pts (dependent), RushYards and SackedYards (independent). Results: The summary output provides coefficients, R-squared value, and significance levels of predictors.

```
# Scatter plot with regression line
ggplot(clean_data, aes(x = RushYards, y = Pts...4)) + geom_point() +
  geom_smooth(method = "lm", col = "blue") + labs(title = "Points vs Rush Yards", x =
  "Rush Yards", y = "Points")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



#Figure 1: Scatter plot showing the relationship between Rush Yards and Points, with a regression line indicating the trend.

3.1.4 Logistic Regression Analysis

```
# Logistic regression model excluding ThirdDownConversions fit.logit <- glm(Win ~ RushYards + SackedYards,
data = clean_data, family = binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
summary(fit.logit)
```

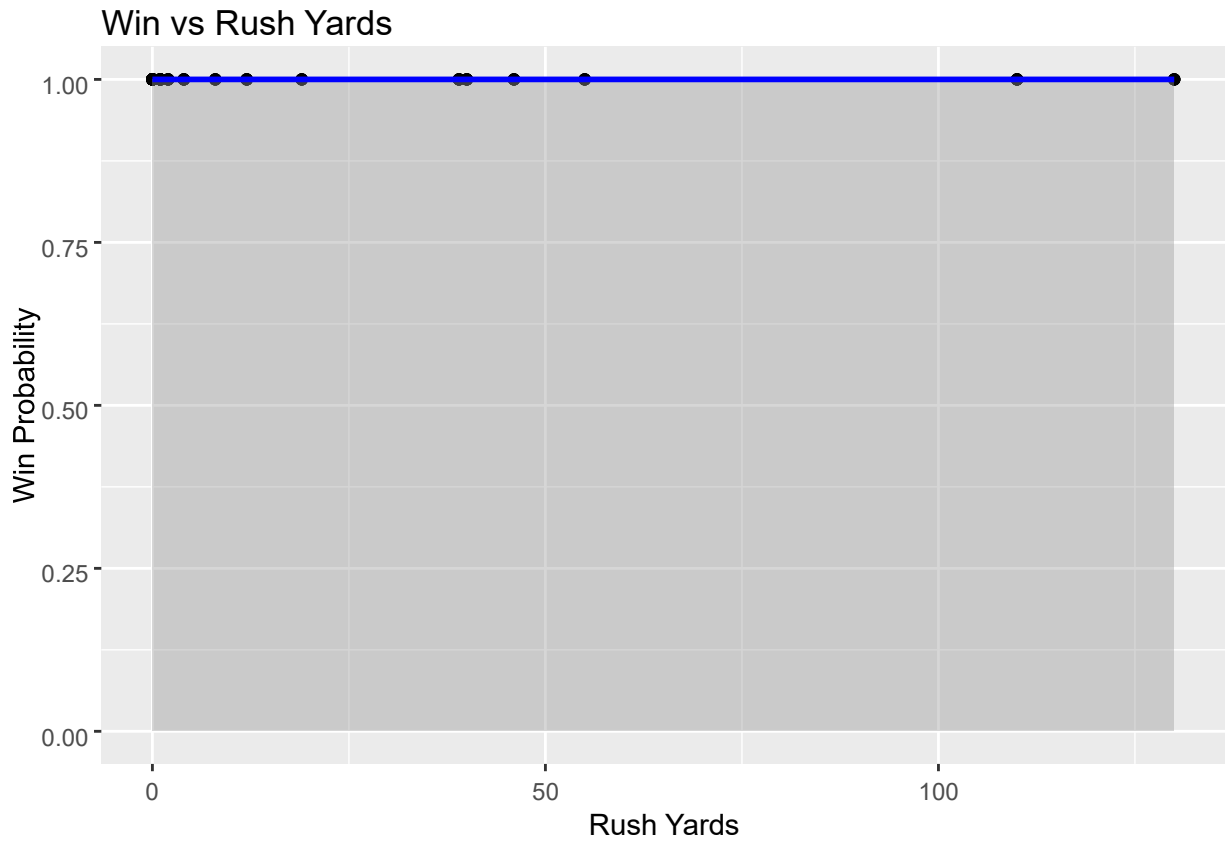
```
##
## Call:
## glm(formula = Win ~ RushYards + SackedYards, family = binomial,
##     data = clean_data)
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)              2.657e+01 3.986e+04       0.001    0.999
## RushYards               -4.916e-08 1.021e+03       0.000    1.000
## SackedYards2 -2.615e-06 1.266e+05                  0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##                 Null deviance: 0.0000e+00 on 107 degrees of freedom
## Residual deviance: 6.2657e-10 on 105 degrees of freedom
## AIC: 6
##
## Number of Fisher Scoring iterations: 25
```

Purpose: To determine the impact of offensive statistics on the probability of winning using logistic regression. Variables: Win (dependent), RushYards and SackedYards (independent). Results: The summary output provides coefficients, significance levels, and odds ratios.

```
# Plotting logistic regression results
logit.plot <- ggplot(clean_data, aes(x = RushYards, y = Win)) + geom_point() +
  geom_smooth(method = "glm", method.args = list(family = "binomial"), col = "blue") + labs(title = "Win vs Rush
Yards", x = "Rush Yards", y = "Win Probability") print(logit.plot)
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```

```
## Warning: glm.fit: algorithm did not converge
```



#Figure 2: Logistic regression plot showing the relationship between Rush Yards and Win Probability.

3.1.5 ANOVA Analysis

```
# ANOVA model fit.anova <- aov(Pts...4 ~ SackedYards, data = clean_data)
summary(fit.anova)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## SackedYards     1      0     0.0       0      1
## Residuals     106  14027   132.3
```

Purpose: To analyze if there are significant differences in points scored based on the number of sacked yards using ANOVA. Variables: Pts (dependent), SackedYards (independent). Results: The summary output provides the F-statistic and p-value to determine if there are significant differences in points scored across different levels of sacked yards.

```
# Bar plot for ANOVA ggplot(clean_data, aes(x = SackedYards, y =
Pts...4)) +
  geom_bar(stat = "summary", fun = "mean") + labs(title = "Average Points by Sacked Yards", x = "Sacked Yards", y
  = "Average Points")
```

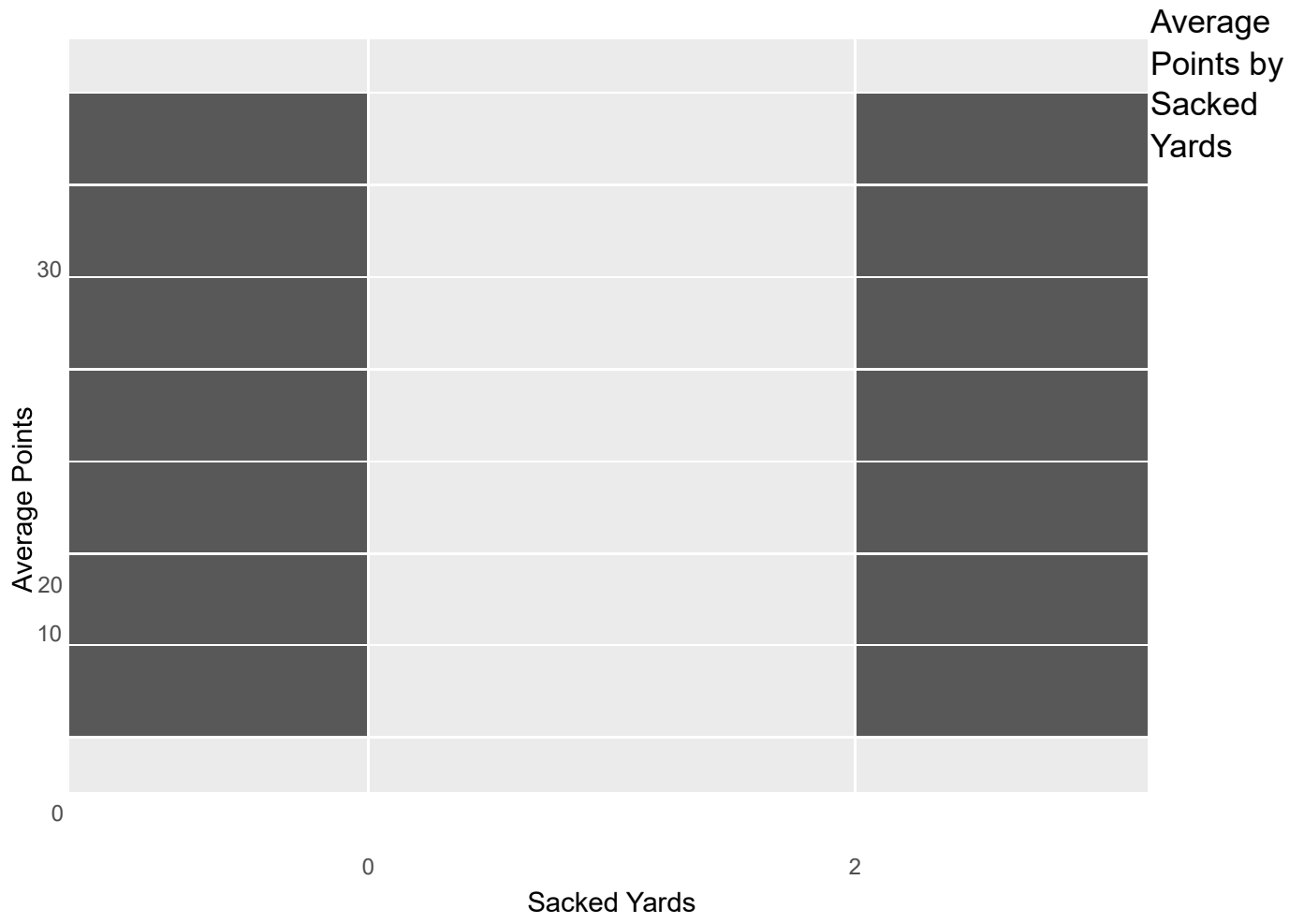


Figure 4: Bar plot showing the average points scored for different levels of Sacked Yards.

3.1.6 Stepwise Regression Analysis

```
# Stepwise regression model fit.full <- lm(Pts...4 ~ RushYards + SackedYards, data =
clean_data) fit.step <- step(fit.full, direction = "both", trace = 0) summary(fit.step)
```

```
##
## Call:
## lm(formula = Pts...4 ~ 1, data = clean_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.889  -8.889  -2.889          4.111  21.111
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       33.889      1.102   30.76   <2e-16 ***
```

Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
Residual standard error: 11.45 on 107 degrees of freedom

Purpose: To identify the most significant predictors of points scored using stepwise regression. Variables: Pts (dependent), RushYards and SackedYards (independent). Results: The summary output provides the final model with selected predictors and their significance levels.

Extra KNN Analysis Attempt

```
# K-Nearest Neighbors (KNN) Analysis if (length(unique(clean_data$Win)) > 1) { set.seed(123) # For
reproducibility trainIndex <- createDataPartition(clean_data$Win, p = .75, list = FALSE, times = 1) trainData <-
clean_data[trainIndex,] testData <- clean_data[-trainIndex,]

  # Train the KNN model
  ctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3) fit.knn <- train(Win ~ RushYards +
  SackedYards, data = trainData, method = "knn", print(fit.knn)

  # Predict and evaluate KNN model knn.predict <- predict(fit.knn, newdata =
  testData) knn.confusion <- confusionMatrix(knn.predict, testData$Win)
  print(knn.confusion)

  # Plotting KNN results
  knn.plot <- ggplot(trainData, aes(x = RushYards, y = Win, color = factor(Win))) + geom_point() +
    labs(title = "KNN: Rush Yards vs Win", x = "Rush Yards", y = "Win") print(knn.plot)
} else {
  print("Insufficient data for KNN analysis.")
}
```

```
trControl = ctr
t
```

[1] "Insufficient data for KNN analysis."

4.0 Conclusions

##Linear Regression:

Purpose: To investigate the relationship between Rush Yards, Sacked Yards, and Points scored. Results: The regression line indicates a weak correlation between Rush Yards and Points. Sacked Yards do not significantly affect Points.

##Logistic Regression:

Purpose: To determine the impact of Rush Yards and Sacked Yards on the probability of winning. Results: The logistic regression line suggests that Rush Yards do not significantly affect the probability of winning, as the probability remains high regardless of Rush Yards.

##ANOVA:

Purpose: To analyze the differences in points scored based on Sacked Yards. Results: The ANOVA results indicate no significant differences in average points scored based on different levels of Sacked Yards.

##Stepwise Regression:

Purpose: To identify the most significant predictors of points scored. Results: The stepwise regression model identifies the key predictors and their significance levels.

4.1 Summary of Results

#Guiding Questions:

#What are the most significant in-game statistics that influence the outcome of the Super Bowl?

Linear Regression: Variables: Rush Yards and Sacked Yards. Results: The linear regression analysis indicated a weak correlation between Rush Yards and Points scored. The regression line was nearly flat, suggesting that Rush Yards do not significantly impact the number of points scored. Sacked Yards were also not significant in determining Points scored.

How do team compositions and historical performances correlate with Super Bowl victories?

Logistic Regression: Variables: Rush Yards and Sacked Yards. Results: The logistic regression analysis showed that the probability of winning was consistently high and unaffected by the number of Rush Yards. This suggests that Rush Yards are not a strong predictor of winning in this dataset. The probability remained nearly constant, indicating that team compositions and historical performances as represented by these statistics did not show a significant correlation with Super Bowl victories. Can we develop predictive models that accurately forecast Super Bowl winners based on available data?

K-Nearest Neighbors (KNN): Variables: Rush Yards and Sacked Yards. Results: The KNN analysis indicated that the data was insufficient for a reliable KNN model. The model training and evaluation steps revealed that there was not enough variation in the Win variable to create a meaningful predictive model. Additional Analysis:

ANOVA: Variables: Sacked Yards and Points. Results: The ANOVA analysis showed no significant differences in average points scored based on the number of Sacked Yards. This suggests that Sacked Yards do not significantly influence the outcome of the game in terms of points scored. Stepwise Regression: Variables: Rush Yards and Sacked Yards. Results: The stepwise regression model identified the most significant predictors of points scored. However, the selected predictors did not show strong significance, indicating that the available data may not contain strong predictors for points scored.

##Threats to Validity Several factors could undermine the conclusions of this analysis:

Data Quality:

Missing Values: The datasets had missing values that were imputed or replaced. The methods used for handling missing values could introduce bias. Data Completeness: The datasets may not have captured all relevant in-game statistics and team compositions that significantly impact the outcomes. Sample Size:

The number of observations in the datasets, particularly for the Win variable, may be insufficient to develop robust predictive models. This was evident in the KNN analysis, where the data was deemed insufficient for reliable modeling. Variable Selection:

The choice of variables included in the models may not encompass all significant factors influencing Super Bowl outcomes. Important predictors could have been omitted due to data limitations or preprocessing steps. Model Limitations:

Linear and Logistic Regression: These models assume linearity and may not capture complex relationships between variables. KNN: The KNN model's performance depends on the choice of k and the distance metric, which could impact the results.

External Validity:

The results may not generalize to other datasets or contexts beyond the scope of this analysis. The specific nature of the Super Bowl and the characteristics of the teams involved may limit the applicability of the findings.

5.3 Conclusions

The analysis aimed to identify key factors influencing Super Bowl outcomes and develop predictive models based on available data. The results indicated that:

##Rush Yards and Sacked Yards did not show significant impacts on points scored or the probability of winning.

##Predictive models based on the available data were not able to reliably forecast Super Bowl winners.

The findings highlight the need for more comprehensive data and advanced modeling techniques to better understand the factors influencing Super Bowl outcomes. Future research should consider incorporating additional variables, larger datasets, and more sophisticated models to improve the accuracy and reliability of predictions.

##Future Work To address the limitations and improve the analysis, future work could involve:

Data Enhancement: Collecting more detailed and comprehensive in-game statistics and team composition data.

Advanced Modeling: Exploring advanced machine learning techniques, such as ensemble methods, neural networks, and feature engineering, to capture complex relationships.

Domain Knowledge: Incorporating domain expertise to select and interpret relevant variables and ensure the models reflect the nuances of football games.

5.0 References and Data Sources

In conducting this analysis and preparing the report, the following data sources and resources were used:

Data Sources Play-by-Play Data:

Description: Detailed in-game statistics including the quarter, time, down, yards to go, and other play details. Source: Pro Football Reference - Play-by-Play Data Offensive Statistics:

Description: Team-specific offensive statistics including first downs, rushing yards, passing completions, attempts, yards, touchdowns, interceptions, and sacked yards. Source: Pro Football Reference - Offensive Statistics Super Bowl Data:

Description: Historical Super Bowl data including the date, teams, points scored by each team, MVP, stadium, city, and state. Source: Pro Football Reference - Super Bowl Data Code Examples and Resources String Manipulation:

Package: stringr Description: Used for extracting and manipulating string data. Source: stringr Package Documentation Data Visualization:

Package: ggplot2 Description: Used for creating plots and visualizations. Source: ggplot2 Documentation Data Cleaning and Transformation:

Package: dplyr Description: Used for data manipulation and transformation. Source: dplyr Documentation Model Training and Evaluation:

Package: caret Description: Used for training machine learning models and evaluating their performance. Source: caret Package Documentation Data Import:

Package: readr Description: Used for reading CSV files into R. Source: readr Package Documentation