**ALY6015.71629.202515 – Team Epsilon – Final Project Report**

**Massachusetts Bay Transportation Authority (MBTA) On-Time Prediction Accuracy Analysis​ Report**

**Date:** December 11, 2024

**Team Members:**

Hari Priya Ramamoorthy ([ramamoorthy.h@northeastern.edu](mailto:ramamoorthy.h@northeastern.edu))

Isaac Nyinaku ([nyinaku.i@northeastern.edu](mailto:nyinaku.i@northeastern.edu))

Shilin Wang ([wang.shili@northeastern.edu](mailto:wang.shili@northeastern.edu))

# **Introduction**

The Massachusetts Bay Transportation Authority (MBTA) strives to provide accurate on-time predictions to improve ridership and operational efficiency. Accurate prediction models are critical for optimizing resources, improving customer experience, and ensuring operational efficiency. The data set is derived from the Massachusetts Bay Transport Authority (MBTA) openData portal which has **"Rapid Transit and Bus Prediction Accuracy Data"** datasetabout accuracy of predictions for the transport modes available in Boston with routes, date, and services alerts. The key columns in this dataset include, date/period, **route\_id,delay bin, arrival\_departure, num\_predictions, num\_accurate\_predictions.** This data was combined with “**MBTA Monthly Ridership by Mode**” to fetch the **ridership volume** data for each transit mode on month level for comprehensive analysis on impact of ridership on MBTA on-time prediction model.

# This study examines the factors influencing the accuracy of MBTA’s model predictions, focusing on ridership volume, route characteristics, and month as potential predictors. Additionally, the study compares different modeling techniques, including linear regression, stepwise regression, and lasso regression, to identify the most significant features affecting prediction accuracy.

# **Methodology**

The analysis included several steps:

1. **Exploratory Data Analysis (EDA)**: This phase involved handling data integration potential data issues, along with plotting visualization graphs to derive insights and business questions for the relationships between the features and the target variable.
2. **Hypothesis Testing**: An ANOVA, Correlation and Chi-square Independence test was conducted to explore the statistical differences in prediction accuracy across transportation modes.
3. **Linear Modeling**: A linear regression model was fit to predict the accuracy of MBTA’s model, considering route, ridership, and month as predictors.
4. **Feature Selection**: Both stepwise regression and lasso regression were employed to assess feature importance and select the most relevant predictors.

**Statistical Techniques**

* **Linear Regression**: This model was used to examine the relationship between the predictors and MBTA On-time prediction accuracy.
* **Stepwise Regression**: This method automatically selected the most significant features based on AIC criteria.
* **Lasso Regression**: Lasso regularization was applied to shrink coefficients of less important features, simplifying the model.

Let’s investigate details on the findings from the analysis.

# **Exploratory Analysis & Insights**

# Initial analysis focused on understanding the general distribution of ridership across different modes, prediction accuracy trends over time, and variations across different routes.

**Insight 1:** Ridership volume is high with Bus than Other Routes (Figure 1)

**Insight 2:** **Orange, Red and Green lines have high variation** in prediction Accuracy across delay bins. **Blue and bus line has constant prediction accuracy**. (Figure 2)

**Insight 3:** Ridership follows the similar trend as Prediction Accuracy Trend over time (Figure 3)

A graph of different colored squares

Description automatically generated A graph of a number of colored boxes

Description automatically generated with medium confidence A graph of blue and red lines

Description automatically generated

**Figure 2: “Bus” has less variations in prediction Accuracy across delay intervals than any other MBTA Routes**

**Figure 3: Ridership and Prediction Accuracy Trend**

**Figure 1: Bus has More Ridership than any other Route**

Based on our EDA, we can see that there is relationship between route’s prediction level and ridership count. To confirm this, hypothesis test was performed.

# **Statistical Tests**

1. Hypothesis Test- 1 : Accuracy Rate Vs Ridership

To test the hypothesis on the impact of ridership on accuracy rate, **Pearson’s Correlation test** was performed to check the relationship between the continuous variables.

**Hypotheses:**

* **Null Hypothesis (H₀):** There is no correlation between accuracy rate and ridership volume. (Correlation=0)
* **Alternative Hypothesis (H₁):** There is a correlation between accuracy rate and ridership volume. (Correlation is not equal to 0)

**Figure 4: Pearson’s Correlation Test Results: Negative correlation between Accuracy rate and Ridership volume**

A white background with black text

Description automatically generated

**Test Decision:**

Based on results, as the p-value is less than 0.05, we **reject the null hypothesis**. The sample estimate of the correlation coefficient is approximately **-0.298**. Hence, we can conclude that there is a **significant negative correlation between** **prediction accuracy** and **total ridership**. As ridership increases, prediction accuracy appears to decrease (moderately).

**Interpretation:**

This suggests that the prediction models are less accurate for routes with higher ridership, possibly due to the increased complexity and variability associated with busier routes.

1. Hypothesis Test - 2: Accuracy Rate Vs Routes

To test the hypothesis about **Operational Efficiency** in relation to **Prediction Accuracy across Different Transportation Modes**, performed an **ANOVA (Analysis of Variance)** test.

**Hypotheses:**

* **Null Hypothesis (H₀):** There is no significant difference in prediction accuracy across different transportation modes.
* **Alternative Hypothesis (H₁):** Prediction accuracy differs significantly across transportation modes.

Figure 5 shows ANOVA results on relationship across different route accuracy. Figure 6 shows the post-hoc Tukey test results.

A close-up of numbers

Description automatically generated A screenshot of a computer

Description automatically generated

**Figure 5: ANOVA Test Results : Prediction Accuracy Varies significantly across Different Transportation Modes**

**Figure 6: Tukey Test Results : Prediction Accuracy across Different Transportation Modes**

**Test Decision:**

Based on results, as the p-value is less than 0.05, we **reject the null hypothesis**. The **ANOVA** test showed that there is a **significant difference** in prediction accuracy across different modes (p < 0.05). Post-hoc **Tukey HSD** analysis revealed significant differences in prediction accuracy between routes, particularly between the **Bus** and **Blue**, **Green**, **Orange**, and **Red** routes. The **Orange vs. Red** comparison had a marginal p-value of 0.071, indicating a near-significant difference.

**Interpretation:**

There are notable differences in prediction accuracy across routes, with **Green** and **Blue** showing

the most significant differences. This variation can inform future model improvements to tailor

predictions based on route characteristics.

1. Hypothesis Test - 3: Accuracy Rate Seasonality

To Assess whether **seasonal factors (e.g., specific months)** have a statistically significant effect on on-time predictionaccuracy, a Chi-Squared test was conducted.

**Hypotheses:**

* **Null Hypothesis (H₀):** Prediction accuracy does not vary significantly across time periods.
* **Alternative Hypothesis (H₁):** Prediction Accuracy is varies across Time.

**Figure 7: Chi-Square Test Results : P >0.05 rejects null**

A black background with white text

Description automatically generated

# **Test Decision:**

Based on results (in Figure 7), the P-Value of 0.5975 > 0.05. Hence, we **cannot** **reject the null hypothesis**, concluding that there is no significant difference in the accuracy of predictions identified in the data.

# **Interpretation:**

Prediction accuracy does not significantly vary across time periods, implying no seasonal factors affecting accuracy.

**Key Insights**

# The analysis revealed a negative correlation between prediction accuracy and ridership volume for MBTA routes, suggesting that higher ridership results in decreased prediction accuracy. The increased complexity of busier routes may contribute to this phenomenon.

# ANOVA tests confirmed significant variability in prediction accuracy across transportation modes, with the Green and Blue routes exhibiting more substantial deviations compared to other modes. The Orange Line showed higher accuracy than the other routes.

# Chi-square Test of independence confirmed that Prediction accuracy does not significantly vary across time periods, implying that while accuracy fluctuates, there is no systematic improvement or decline over time.

# **Linear Model to Understand Influencers of MBTA’s Model Accuracy**

A linear regression model was fit to predict the accuracy of MBTA’s model, considering route, ridership, and month as predictors. Figure 8 shows the residual plot for the model after feature engineering. The plots indicate that linearity, no multicollinearity, homoscedasticity, and outliers have been appropriately addressed, confirming the assumptions of a well-fitting linear regression model.

Figure 8 : Linear Model Summary Figure 9 : Residual Analysis of Linear Model

A screenshot of a computer

Description automatically generated A group of graphs showing different values

Description automatically generated

The linear regression model provided baseline results for predicting the accuracy of MBTA's predictions. The **Training RMSE** was 3.861065, and the **Test RMSE** was 5.561925. These results indicate potential overfitting, as there was a significant difference between the training and test RMSE values.

**Feature Importance: Lasso Vs Step-Wise Regularization**

**Lasso Regression**:

Lasso regression performed feature selection by shrinking coefficients for less important predictors. The optimal lambda values identified through cross-validation were **λ-min = 0.113751** and **λ.1se = 0.607055**, with the model utilizing between 3 to 10 predictors.

*Figure 10 : Lasso Regression Optimal Lambda Plot Figure 11 : Lasso Regression Coefficient Plot*A graph with numbers and a line

Description automatically generated A graph with numbers and lines

Description automatically generated

Figure-12 : Lasso Coefficients Shrinked to 0.

A screenshot of a computer

Description automatically generated

The lasso model emphasized the importance of features like route\_id,total\_ridership by eliminating less significant “month”, improving model interpretability.

**Stepwise Regression**:

The stepwise regression model identified **route\_id** and **ridership volume** as the most significant predictors of model accuracy, while the **month** feature was excluded from the final model. The adjusted R-squared value for the stepwise model was **63.74%**, indicating that the model explained a substantial portion of the variance in accuracy.

*Figure 13 : Step-wise Regression Variables Figure 14 : Lasso Regression Coefficient Plot*

A close-up of a white background

Description automatically generated A screenshot of a computer

Description automatically generated

This suggests that, despite potential seasonality or time-based patterns, **month** might not be as critical as other features in predicting the target variable.

**Key Findings**

The findings suggest that prediction accuracy is negatively affected by higher ridership volumes, likely due to the increased variability and complexity of busier routes. The ANOVA test confirmed that prediction accuracy significantly varies across transportation modes, with the Green and Blue lines showing greater deviation compared to the other routes. The Orange Line, in particular, demonstrated a higher level of prediction accuracy, potentially due to less variability in ridership patterns.

The exclusion of the **month** feature in both the lasso and stepwise regression models implies that it may not be as critical as other features in predicting model accuracy. Despite potential seasonal patterns, the month did not significantly contribute to the model’s performance. This result could suggest that the month’s influence is captured by other features, such as ridership volume or route-specific characteristics.

*Figure 15 : Model Comparison*

A screenshot of a computer

Description automatically generated

The comparison of different regression models (linear, stepwise, and lasso) demonstrated that while the linear model provided baseline insights, feature selection techniques like lasso and stepwise regression enhanced model performance and interpretability. The stepwise model identified **route\_id** and **ridership volume** as the key drivers of model accuracy, which aligns with the insights from the linear model.

**Conclusion**

This analysis highlights the importance of ridership volume and route characteristics in predicting the accuracy of MBTA's ridership predictions. While the **month** feature was excluded from both stepwise and lasso regression models, suggesting its weak relationship with prediction accuracy, the models identified **route\_id** and **ridership volume** as the primary predictors. The study recommends that MBTA focus on refining its models by addressing the operational inefficiencies identified in the analysis.

Future efforts should aim to improve model accuracy, particularly for high-traffic routes, through more sophisticated modeling techniques and better feature engineering.

# Appendix : R Code

#install.packages('dplyr','tidyr','ggplot2','lubridate','zoo','glmnet','caret','car','MASS','knitr')  
library(dplyr)  
library(tidyr)  
library(ggplot2)  
library(lubridate)  
library(zoo)  
library(glmnet)  
library(caret)  
library(MASS)  
library(knitr)  
library(car)  
  
################################## Load Inputs ##################################  
##'rapid\_transit\_and\_bus\_prediction\_accuracy\_data.csv'  
mbta\_data <- read.table(file.choose(), sep=",",header=TRUE, stringsAsFactors = FALSE)  
##'MBTA\_Ratings\_%26\_Seasons.csv'  
season\_data<- read.table(file.choose(), sep=",",header=TRUE, stringsAsFactors = FALSE)  
# Load ridership data  
ridership <- read.table(file.choose(), sep=",",header=TRUE, stringsAsFactors = FALSE)  
################################## Load Inputs ##################################  
  
################################## Merge Seasons and Ridership ##################################  
# Convert date fields to Date type  
mbta\_data$weekly <- **as**.Date(mbta\_data$weekly)  
season\_data$date\_start <- **as**.Date(season\_data$date\_start)  
season\_data$date\_end <- **as**.Date(season\_data$date\_end)  
  
# Merge datasets based on weekly date falling within season date ranges  
merged\_data <- mbta\_data %>%   
 mutate(route\_id = gsub("^\\s\*$", "bus", route\_id), # Replace empty spaces **with** 'bus'  
 route\_id = ifelse(grepl("Green", route\_id), "Green", route\_id),  
 route\_id = ifelse(grepl("Orange", route\_id), "Orange", route\_id),  
 route\_id = ifelse(grepl("Blue", route\_id), "Blue", route\_id),  
 route\_id = ifelse(grepl("Red", route\_id), "Red", route\_id)) %>%   
 inner\_join(season\_data, by = character()) %>%  
 filter(weekly >= date\_start & weekly <= date\_end)  
  
  
  
# Add a month column (convert weekly dates to YYYY-MM format)  
merged\_data <- merged\_data %>%  
 mutate(month = format(weekly, "%Y-%m")) # Converts to "YYYY-MM" format  
  
# Create a mapping table for ridership routes to prediction accuracy routes  
route\_mapping <- tibble::tibble(  
 ridership\_routes = c("Bus", "Commuter Rail", "Green Line", "Orange Line", "Red Line",   
 "Silver Line", "The RIDE", "Blue Line", "Boat-F1", "Boat-F3",   
 "Boat-F4", "Ferry"),  
 prediction\_routes = c("bus", NA, "Green", "Orange", "Red",   
 NA, NA, "Blue", "Ferry", "Ferry",   
 "Ferry", "Ferry") # Map to comparable names or NA for no equivalent  
)  
  
# Join ridership **with** route\_mapping  
standardized\_ridership <- ridership %>%  
 inner\_join(route\_mapping, by = c("route\_or\_line" = "ridership\_routes"))  
  
# Ridership By Month and Routes  
ridership\_group <- standardized\_ridership %>%  
 mutate(  
 service\_date = **as**.Date(service\_date), # Convert to Date format **if** not already  
 yyyy\_month = format(service\_date, "%Y-%m") # Extract year and month **in** "YYYY-MM" format  
 ) %>%  
 group\_by(yyyy\_month, prediction\_routes) %>%  
 summarize(  
 total\_ridership = sum(average\_monthly\_ridership, na.rm = TRUE)  
 )  
  
# Step 3: Merge ridership\_group **with** merged\_data  
final\_data\_merged\_ridership <- merged\_data %>%  
 inner\_join(ridership\_group, by = c("month" = "yyyy\_month","route\_id"="prediction\_routes"))  
  
final\_data\_merged\_ridership\_grouped\_new <- final\_data\_merged\_ridership %>%  
 # Take out Month alone  
 mutate(month\_numeric = substr(month, nchar(month) - 1, nchar(month))) %>%  
 group\_by(route\_id,month\_numeric,season\_name) %>%  
 summarize(  
 total\_predictions = sum(num\_predictions, na.rm = TRUE),  
 total\_accurate = sum(num\_accurate\_predictions, na.rm = TRUE),  
 total\_ridership= min(total\_ridership),  
 bin=min(bin)  
 ) %>%  
 mutate(accuracy\_rate = (total\_accurate / total\_predictions) \* 100)  
  
# Check for missing values  
colSums(is.na(final\_data\_merged\_ridership\_grouped\_new))  
  
final\_data\_merged\_ridership\_grouped <- final\_data\_merged\_ridership %>%  
 group\_by(route\_id,month,bin) %>%  
 summarize(  
 total\_predictions = sum(num\_predictions, na.rm = TRUE),  
 total\_accurate = sum(num\_accurate\_predictions, na.rm = TRUE),  
 total\_ridership= min(total\_ridership),  
 bin=min(bin)  
 ) %>%  
 mutate(accuracy\_rate = (total\_accurate / total\_predictions) \* 100)  
  
# Check for missing values  
colSums(is.na(final\_data\_merged\_ridership\_grouped))  
  
  
################################## Merge Seasons and Ridership ##################################  
  
################################## EDA ##################################  
# Bar Plot for `total\_ridership` by `route\_id`  
ggplot(final\_data\_merged\_ridership\_grouped, aes(x = reorder(route\_id,desc(total\_ridership)), y = total\_ridership/1000000, fill = route\_id)) +   
 geom\_bar(stat = "identity") +  
 labs(title = "Total Ridership by Route From 2020-08 to 2024-05 ", x = "Route ID", y = "Total Ridership(in Mils)") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))+  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, size = 14, face = "bold"),  
 axis.title.x = element\_text(size = 12),  
 axis.title.y = element\_text(size = 12),  
 axis.text.x = element\_text(size = 10, angle = 45, hjust = 1),  
 axis.text.y = element\_text(size = 10)  
 )  
################################## EDA ##################################  
  
################################## EDA ##################################  
  
# Rescaling ridership for consistent scaling  
final\_data\_merged\_ridership\_grouped <- final\_data\_merged\_ridership\_grouped %>%  
 mutate(scaled\_ridership = total\_ridership / max(total\_ridership) \* 1000)  
  
  
# Plot : Prediction and Ridership follows almost similar trends  
ggplot(final\_data\_merged\_ridership\_grouped, aes(x = **as**.Date(paste0(month, "-01")))) +  
 geom\_line(aes(y = accuracy\_rate, color = "Accuracy Rate"), size = 1) +  
 geom\_line(aes(y = scaled\_ridership, color = "Ridership Volume"), size = 1)+ #linetype = "dashed") +  
 scale\_y\_continuous(  
 name = "Accuracy Rate (%)",  
 sec.axis = sec\_axis(~ . \* max(final\_data\_merged\_ridership\_grouped$total\_ridership) / 1000000, name = "Total Ridership (in Million)")  
 ) +  
 labs(  
 title = "Accuracy Rate and Ridership Trends Over Time by Route",  
 x = "Month",  
 color = "Metric"  
 ) +  
 facet\_wrap(~ route\_id, scales = "free\_y") +  
 theme\_minimal() +  
 theme(  
 legend.position = "bottom",  
 strip.text = element\_text(face = "bold")  
 )  
  
################################## EDA ##################################  
  
################################## EDA ##################################   
# Box plot **of** prediction accuracy by route and bin  
ggplot(final\_data\_merged\_ridership\_grouped, aes(x = bin, y = accuracy\_rate, fill = route\_id)) +  
 geom\_boxplot(alpha = 0.7) +  
 labs(title = "Prediction Accuracy Distribution Across Delay Time Intervals",  
 x = "Delay Time Intervals (in min)",  
 y = "Prediction Accuracy Rate (%)") +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5, size = 14, face = "bold"),  
 axis.title.x = element\_text(size = 12),  
 axis.title.y = element\_text(size = 12),  
 axis.text.x = element\_text(size = 10, angle = 45, hjust = 1),  
 axis.text.y = element\_text(size = 10)  
 ) +  
 facet\_wrap(~ route\_id, scales = "free\_y") # Add route **as** a facet  
################################## EDA ##################################   
  
################################## Hypothesis -1 : Prediction Accuracy Vs Ridership ################################################  
# Perform Pearson correlation test  
correlation\_result <- cor.test(final\_data\_merged\_ridership\_grouped$accuracy\_rate, final\_data\_merged\_ridership\_grouped$total\_ridership)  
correlation\_result  
################################## Hypothesis -1 : Prediction Accuracy Vs Ridership Volume ################################################  
  
################################## Hypothesis -2 : Prediction Accuracy Vs Modes ################################################  
# Perform one-way ANOVA to test the difference **in** prediction accuracy across modes  
anova\_result\_modes <- aov(accuracy\_rate ~ route\_id, data = final\_data\_merged\_ridership\_grouped)  
  
# Summary **of** the ANOVA result  
summary(anova\_result\_modes)  
  
# Perform Tukey's HSD post-hoc test to compare each pair of modes  
tukey\_result <- TukeyHSD(anova\_result\_modes)  
  
# Summary of Tukey's results  
print(tukey\_result)  
################################## Hypothesis -2 : Prediction Accuracy Vs Modes ################################################  
  
################################## Hypothesis -3 : Prediction Accuracy Vs Time ################################################  
# Data Cleaning  
data\_cleaned <- mbta\_data %>%  
 filter(!is.na(route\_id)) %>%  
 mutate(  
 prediction\_accuracy = num\_accurate\_predictions / num\_predictions,  
 weekly = **as**.Date(weekly)  
 )  
  
# Filter data for the last 6 months  
latest\_date <- max(data\_cleaned$weekly, na.rm = TRUE)  
six\_months\_ago <- latest\_date - months(6)  
  
data\_last\_6\_months <- data\_cleaned %>%  
 filter(weekly >= six\_months\_ago)  
  
# Categorize prediction accuracy into bins  
data\_last\_6\_months <- data\_last\_6\_months %>%  
 mutate(  
 accuracy\_category = cut(  
 prediction\_accuracy,  
 breaks = c(0, 0.7, 0.9, 1),  
 labels = c("Low", "Medium", "High"),  
 include.lowest = TRUE  
 )  
 )  
  
# Calculate weekly mean prediction accuracy  
accuracy\_trends <- data\_last\_6\_months %>%  
 group\_by(weekly) %>%  
 summarize(mean\_accuracy = mean(prediction\_accuracy, na.rm = TRUE))  
  
summary(accuracy\_trends)  
  
# Plot the trends  
ggplot(accuracy\_trends, aes(x = weekly, y = mean\_accuracy)) +  
 geom\_line(color = "blue") +  
 geom\_point(color = "red") +  
 labs(  
 title = "Prediction Accuracy Trends Over the Past 6 Months",  
 x = "Week",  
 y = "Mean Prediction Accuracy"  
 ) +  
 theme\_minimal()  
  
# Create a contingency table  
contingency\_table <- table(data\_last\_6\_months$weekly, data\_last\_6\_months$accuracy\_category)  
  
# Perform Chi-Square Test  
chi\_square\_test <- chisq.test(contingency\_table)  
print(chi\_square\_test)  
  
################################## Hypothesis -3 : Prediction Accuracy Vs Time ################################################  
  
# Set seed for reproducibility  
set.seed(123)  
  
# Partition the data: 80% training and 20% testing  
trainIndex <- createDataPartition(final\_data\_merged\_ridership\_grouped\_new$accuracy\_rate, p = 0.8, list = FALSE)  
train\_data <- final\_data\_merged\_ridership\_grouped\_new[trainIndex, ]  
test\_data <- final\_data\_merged\_ridership\_grouped\_new[-trainIndex, ]  
  
# Model 1: General Linear Regression  
lm\_model <- lm(accuracy\_rate ~ route\_id + month\_numeric + total\_ridership, data = train\_data)  
lm\_pred\_train <- predict(lm\_model, train\_data)  
lm\_pred\_test <- predict(lm\_model, test\_data)  
summary(lm\_model)  
  
### Residual Plot  
par(mfrow=c(2,2))  
plot(lm\_model)  
  
## Component +Residual Plot for each predictor  
### Residual Plot  
par(mfrow=c(1,1))  
crPlots(lm\_model)  
  
# Calculate RMSE for Linear Regression  
lm\_rmse\_train <- sqrt(mean((lm\_pred\_train - train\_data$accuracy\_rate)^2))  
lm\_rmse\_test <- sqrt(mean((lm\_pred\_test - test\_data$accuracy\_rate)^2))  
cat(lm\_rmse\_train,lm\_rmse\_test)  
  
# Model 2: Lasso Regression  
x\_train <- model.matrix(accuracy\_rate ~ route\_id + month\_numeric + total\_ridership, data = train\_data)[, -1]  
y\_train <- train\_data$accuracy\_rate  
lasso\_model <- cv.glmnet(x\_train, y\_train, alpha = 1)  
lasso\_pred\_train <- predict(lasso\_model, x\_train, s = "lambda.min")  
lasso\_pred\_test <- predict(lasso\_model, model.matrix(accuracy\_rate ~ route\_id + month\_numeric + total\_ridership, data = test\_data)[, -1], s = "lambda.min")  
  
# Plots and Results  
cat("Lasso Regression: lambda.min =", lasso\_model$lambda.min, "lambda.1se =", lasso\_model$lambda.1se, "\n")  
abline(v=log(c(lasso\_model$lambda.min,lasso\_model$lambda.1se)),lty=2)  
## PLOT COEFFICIENTS  
print(coef(lasso\_model))  
plot(coef(lasso\_model))  
abline(h=0)  
  
# Calculate RMSE for Lasso Regression  
lasso\_rmse\_train <- sqrt(mean((lasso\_pred\_train - y\_train)^2))  
lasso\_rmse\_test <- sqrt(mean((lasso\_pred\_test - test\_data$accuracy\_rate)^2))  
cat(lasso\_rmse\_train,lasso\_rmse\_test)  
  
# Model 3: Stepwise Regression  
step\_model <- stepAIC(lm\_model, direction = "both", trace = FALSE)  
step\_pred\_train <- predict(step\_model, train\_data)  
step\_pred\_test <- predict(step\_model, test\_data)  
  
# Calculate RMSE for Stepwise Regression  
step\_rmse\_train <- sqrt(mean((step\_pred\_train - train\_data$accuracy\_rate)^2))  
step\_rmse\_test <- sqrt(mean((step\_pred\_test - test\_data$accuracy\_rate)^2))  
cat(step\_rmse\_train,step\_rmse\_test)  
  
# Collect results for comparison  
coefficients\_lm <- coef(lm\_model)  
coefficients\_lasso <- coef(lasso\_model, s = "lambda.min")  
coefficients\_step <- coef(step\_model)  
summary(step\_model)  
  
# Extract the anova table **from** the stepwise model, which contains the step information.  
step\_info <- step\_model$anova  
  
# Create a data frame to store step information for plotting  
stepwise\_df <- data.frame(  
 Step = 1:nrow(step\_info),  
 AIC = step\_info$AIC,  
 Variable = step\_info$Step  
)  
  
# Use kable to display the table **in** a report-friendly format  
kable(stepwise\_df,   
 caption = "Stepwise Regression: Steps, AIC, and Variables Added/Removed",  
 col.names = c("Step", "AIC", "Variable Added/Removed"),  
 format = "markdown")   
  
# Print the table for reporting  
print(stepwise\_df)  
  
# Load necessary libraries  
library(knitr)  
  
# Linear Regression - Adjusted R-squared  
adj\_r2\_linear <- summary(lm\_model)$adj.r.squared  
  
# Stepwise Regression - Adjusted R-squared  
adj\_r2\_step <- summary(step\_model)$adj.r.squared  
  
# Create a data frame for comparison  
model\_comparison <- data.frame(  
 Model = c("Linear Regression", "Stepwise Regression", "Lasso Regression"),  
 `Adj. R-squared` = c(adj\_r2\_linear, adj\_r2\_step, 'NA'),  
 `Train RMSE` = c(lm\_rmse\_train, step\_rmse\_train, lasso\_rmse\_train),  
 `Test RMSE` = c(lm\_rmse\_test, step\_rmse\_test, lasso\_rmse\_test)  
)  
  
kable(model\_comparison, caption = "Comparison of Models: Adjusted R-squared and RMSE")  
  
# Create a data frame for RMSE comparison  
rmse\_comparison <- data.frame(  
 Model = c("Linear Regression (Train)", "Stepwise Regression (Train)", "Lasso Regression (Train)",  
 "Linear Regression (Test)", "Stepwise Regression (Test)", "Lasso Regression (Test)"),  
 RMSE = c(step\_rmse\_train, step\_rmse\_train, lasso\_rmse\_train,  
 step\_rmse\_test, step\_rmse\_test, lasso\_rmse\_test)  
)  
  
# Load the knitr library for kable  
library(knitr)  
  
# Create a table using kable to display RMSE comparison  
kable(rmse\_comparison, caption = "RMSE Comparison Across Models", digits = 4)

# References

Bluman, A. G. (2014). Elementary statistics: A step by step approach (9th ed.). McGraw-Hill.

MBTA. (n.d.). MBTA Blue Book Open Data Portal. Retrieved from [https://mbtahttps://mbta-massdot.opendata.arcgis.com/massdot.opendata.arcgis.com/](https://mbta-massdot.opendata.arcgis.com/)

Massachusetts Bay Transportation Authority (MBTA). (n.d.). MBTA Rapid Transit and Bus Prediction Accuracy Data. Retrieved from <https://mbta-massdot.opendata.arcgis.com/>

Massachusetts Bay Transportation Authority (MBTA). (n.d.). MBTA Seasonality Data. Retrieved from [https://mbta-](https://mbta-massdot.opendata.arcgis.com/datasets/a2d15ddd86b34867a31cd4b8e0a83932_0/explore)

[massdot.opendata.arcgis.com/datasets/a2d15ddd86b34867a31cd4b8e0a83932\_0/explore](https://mbta-massdot.opendata.arcgis.com/datasets/a2d15ddd86b34867a31cd4b8e0a83932_0/explore)

Chen, M., Liu, X., Xia, J., & Chien, S. (2004). Predicting bus arrival time on the basis of global positioning system data. *Journal of Transportation Research Board*, 1885(1), 98-106. Retrieved from <https://www.researchgate.net/publication/245562763_Predicting_Bus_Arrival_Time_on_the_Basis_of_Global_Positioning_System_Data>