**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:**

1. Hari Priya Ramamoorthy (ramamoorthy.h@northeastern.edu)

2. Isaac Nyinaku (nyinaku.i@northeastern.edu)

3. Shilin Wang (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. 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.

## Data Analysis

The analysis included several steps:

1. **Exploratory Data Analysis (EDA)**: This phase involved inspecting the relationships between the features and the target variable, along with detecting any potential data issues.

2. **Hypothesis Testing**: An ANOVA test was conducted to explore the 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 model 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 has 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)

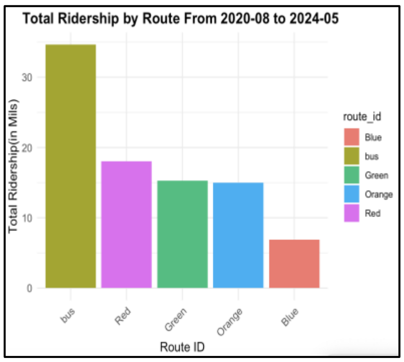



*Figure 1: Bus has More Ridership than any other Route*

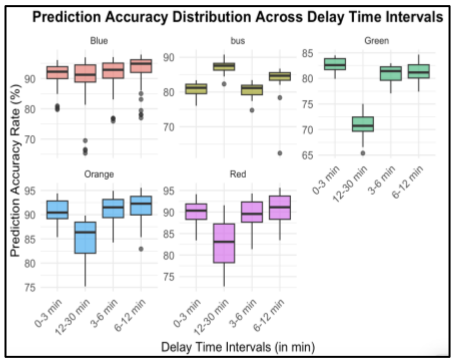



*Figure 2: "Bus" has less variations in prediction Accuracy across delay*

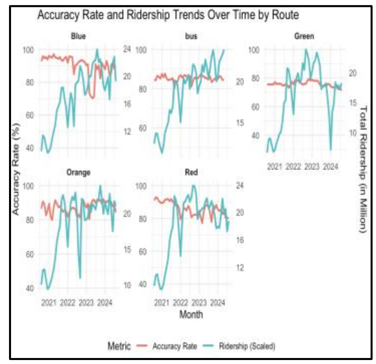



*Figure 3: Ridership and Prediction Accuracy Trend*

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 ($H_0$):** There is no correlation between accuracy rate and ridership volume. (Correlation=0)
- **Alternative Hypothesis ($H_1$):** 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*

```
> correlation_result

	Pearson's product-moment correlation

data:  final_data_merged_ridership_grouped$accuracy_rate and final_data_merged_ridership_grouped$total_ridership
t = -9.692, df = 966, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3540648 -0.2391729
sample estimates:
       cor
-0.2976964
```

**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.

## 2. 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 ($H_0$):** There is no significant difference in prediction accuracy across different transportation modes.
- **Alternative Hypothesis ($H_1$):** 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.

```
> summary(anova_result_modes)
             Df Sum Sq Mean Sq F value Pr(>F)
route_id      4  19848    4962   220.6 <2e-16 ***
Residuals   963  21664      22
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
> print(tukey_result)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = accuracy_rate ~ route_id, data = final_data_merged_ridership_grouped)

$route_id
                  diff        lwr          upr     p adj
bus-Blue      -7.899998  -9.230570  -6.56942618 0.0000000
Green-Blue   -12.185227 -13.494621 -10.87583267 0.0000000
Orange-Blue   -1.541363  -2.850757  -0.23196850 0.0116684
Red-Blue      -2.786627  -4.096021  -1.47723249 0.0000001
Green-bus     -4.285229  -5.615801  -2.95465714 0.0000000
Orange-bus     6.358635   5.028063   7.68920704 0.0000000
Red-bus        5.113371   3.782799   6.44394305 0.0000000
Orange-Green  10.643864   9.334470  11.95325841 0.0000000
Red-Green      9.398600   8.089206  10.70799442 0.0000000
Red-Orange    -1.245264  -2.554658   0.06413024 0.0712908

>
```

*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.

## 3. Hypothesis Test - 3: Accuracy Rate Seasonality

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

**Hypotheses:**

- **Null Hypothesis ($H_0$):** Prediction accuracy does not vary significantly across time periods.
- **Alternative Hypothesis ($H_1$):** Prediction Accuracy is varies across Time.

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

```
> print(chi_square_test)

        Pearson's Chi-squared test

data:  contingency_table
X-squared = 88.041, df = 92, p-value = 0.5975

>
```

**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

1. 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.

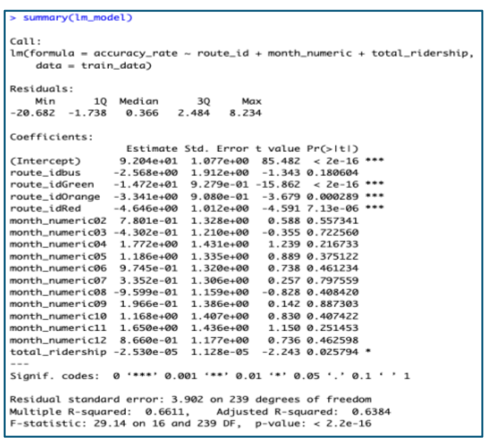
2. ANOVA tests confirmed significant variability in prediction accuracy across transportation modes, with the Green and Blue routes exhibiting more substantial deviations compared to other modes. The Orange Line showed higher accuracy than the other routes.

3. 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

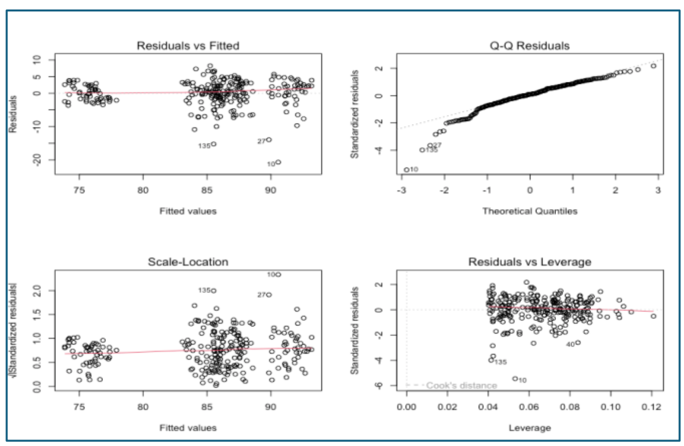
A linear regression model was fit to predict the accuracy of MBTA's model, considering route, ridership, and month as predictors. Figure 8 shows the residual plot for the model after feature engineering. The plots indicate that linearity, no multicollinearity, homoscedasticity, and outliers have been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

*Figure 8 : Linear Model Summary*



```
> summary(lm_model)

Call:
lm(formula = accuracy_rate ~ route_id + month_numeric + total_ridership,
    data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-20.682  -1.738   0.366   2.484   8.234

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      9.204e+01  1.077e+00  85.482  < 2e-16 ***
route_idbus     -2.568e+00  1.912e+00  -1.343 0.180604
route_idGreen   -1.472e+01  9.279e-01 -15.862  < 2e-16 ***
route_idOrange  -3.341e+00  9.080e-01  -3.679 0.000289 ***
route_idRed     -4.646e+00  1.012e+00  -4.591 7.13e-06 ***
month_numeric02  7.801e-01  1.328e+00   0.588 0.557341
month_numeric03 -4.302e-01  1.210e+00  -0.355 0.722560
month_numeric04  1.772e+00  1.431e+00   1.239 0.216733
month_numeric05  1.186e+00  1.335e+00   0.889 0.375122
month_numeric06  9.745e-01  1.320e+00   0.738 0.461234
month_numeric07  3.352e-01  1.306e+00   0.257 0.797559
month_numeric08 -9.599e-01  1.159e+00  -0.828 0.408420
month_numeric09  1.966e-01  1.386e+00   0.142 0.887303
month_numeric10  1.168e+00  1.407e+00   0.830 0.407422
month_numeric11  1.650e+00  1.436e+00   1.150 0.251453
month_numeric12  8.660e-01  1.177e+00   0.736 0.462598
total_ridership -2.530e-05  1.128e-05  -2.243 0.025794 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.902 on 239 degrees of freedom
Multiple R-squared:  0.6611,    Adjusted R-squared:  0.6384
F-statistic: 29.14 on 16 and 239 DF,  p-value: < 2.2e-16
```

*Figure 9 : Residual Analysis of Linear Model*




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*

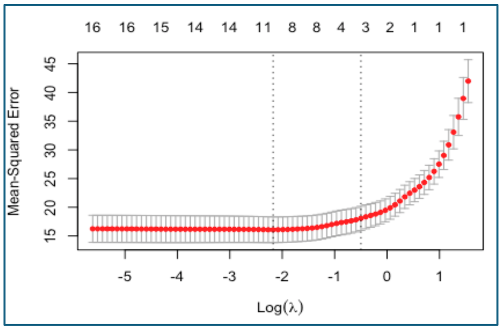



*Figure 11 : Lasso Regression Coefficient Plot*

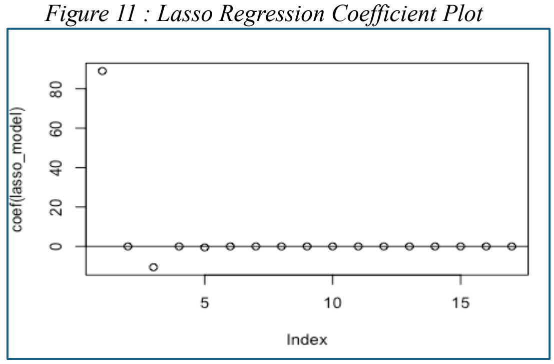
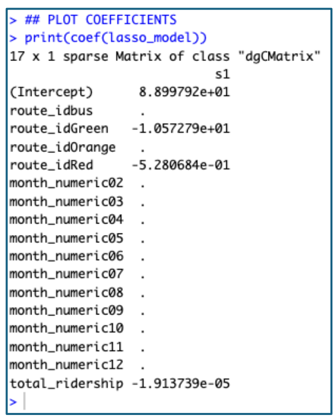



Figure-12 : Lasso Coefficients Shrinked to 0.



```
> ## PLOT COEFFICIENTS
> print(coef(lasso_model))
17 x 1 sparse Matrix of class "dgCMatrix"
                        s1
(Intercept)      8.899792e+01
route_idbus      .
route_idGreen   -1.057279e+01
route_idOrange   .
route_idRed     -5.280684e-01
month_numeric02  .
month_numeric03  .
month_numeric04  .
month_numeric05  .
month_numeric06  .
month_numeric07  .
month_numeric08  .
month_numeric09  .
month_numeric10  .
month_numeric11  .
month_numeric12  .
total_ridership -1.913739e-05
>
```

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*

Table: Stepwise Regression: Steps, AIC, and Variables Added/Removed

| Step | AIC | Variable Added/Removed |
|---:|---:|:---|
| 1 | 713.4887 | |
| 2 | 703.6829 | - month_numeric |

```
> summary(step_model)

Call:
lm(formula = accuracy_rate ~ route_id + total_ridership, data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-21.4566  -1.4096   0.4395   2.6867   6.8982 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      9.229e+01  6.704e-01 137.672  < 2e-16 ***
route_idbus     -2.629e+00  1.822e+00  -1.442 0.150417    
route_idGreen   -1.475e+01  9.085e-01 -16.238  < 2e-16 ***
route_idOrange  -3.403e+00  8.929e-01  -3.811 0.000174 ***
route_idRed     -4.628e+00  9.843e-01  -4.702 4.26e-06 ***
total_ridership -2.295e-05  1.056e-05  -2.174 0.030626 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.907 on 250 degrees of freedom
Multiple R-squared:  0.6446,	Adjusted R-squared:  0.6375 
F-statistic: 90.68 on 5 and 250 DF,  p-value: < 2.2e-16
```

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*

Table: Comparison of Models: Adjusted R-squared and RMSE

| Model | Adj..R.squared | Train.RMSE | Test.RMSE |
|:---|:---|---:|---:|
| Linear Regression | 0.638431932912848 | 3.770193 | 5.842114 |
| Stepwise Regression | 0.637477500404615 | 3.861065 | 5.561925 |
| Lasso Regression | NA | 3.819269 | 5.546328 |

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','g
lmnet','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$accu
racy_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)),l
ty=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
```

```r
# 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/

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-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