MDA9159A- Seoul Bike Sharing Demand

| Garima Gambhir | Ritika Pandey | Sumedha Galgali |

Literature Review

Research on bike-sharing systems highlights the influence of environmental, temporal, and contextual factors on demand. Faghih-Imani et al. (2017) found temperature, precipitation, and proximity to transit significantly impact bike usage, while Gebhart and Noland (2014) emphasized time of day and user demographics. Predictive models, such as regression and machine learning methods, have demonstrated their utility. Li et al. (2015) validated regression models for meteorological impacts, and Wang et al. (2018) showed Random Forest and GBM excel at capturing complex patterns, a strength reflected in our use of XGBoost.

Studies like Kim et al. (2020) focus on Seoul, identifying parallels with global trends, such as higher demand during favorable weather and peak hours. These insights, combined with predictive analytics, aid in system optimization by improving resource allocation and enhancing user satisfaction.

Abstract

The Seoul Bike Sharing system plays a key role in promoting sustainable urban mobility. This project analyzes the factors influencing bike-sharing demand and develops predictive models for demand forecasting. Using a comprehensive dataset, we explored how variables such as weather, temporal patterns, and holidays affect bike rental counts. Data preprocessing involved feature engineering for time-based attributes and dummy variable creation for categorical factors. We applied regression techniques, including multiple linear regression and regularized models (LASSO, Ridge), along with machine learning approaches like Random Forest and XGBoost. Exploratory Data Analysis revealed strong correlations between demand and temperature, rush hour usage, and seasonal effects. The best-performing model achieved a significant reduction in forecasting errors (15%) compared to baseline linear models, demonstrating the effectiveness of data-driven approaches in urban mobility planning.

Introduction

The rapid urbanization and growing environmental concerns have spurred a global shift toward sustainable transportation solutions. In cities like Seoul, where traffic congestion and pollution are pressing issues, bike-sharing systems have emerged as an eco-friendly and efficient mode of transport, offering flexibility, reducing reliance on private vehicles, and contributing to environmental sustainability. One such system, the Seoul Bike Sharing program, commonly known as "Ddareungi," plays a critical role in enhancing urban mobility and fostering a greener urban environment. However, for Ddareungi to be successful, it must effectively meet the demand for bikes in various urban settings and conditions.

Understanding the factors that influence bike demand is crucial for optimizing operations, improving the user experience, and reducing inefficiencies. These factors include a variety of

temporal, weather-related, and contextual variables that can impact how and when people use bikes. For example, temperature and humidity can significantly affect people's willingness to bike, while time of day and seasonality may dictate commuter vs. leisure use. Furthermore, special events, holidays, and even wind speed can create fluctuations in demand. By accurately understanding these factors, we can not only predict demand more effectively but also adjust the bike-sharing system's operations to ensure bikes are available where and when they are most needed.

This project investigates the key drivers of bike-sharing demand in Seoul, utilizing a comprehensive dataset that includes various temporal, weather, and contextual factors. The primary goal is to analyze these variables, identify patterns, and establish relationships that allow for accurate demand forecasting. This analysis aims to support real-time operational decisions, such as redistributing bikes to high-demand areas, optimizing fleet size, and scheduling maintenance to minimize downtime.

The dataset includes key variables such as temperature, humidity, wind speed, time-related attributes like the hour of the day and season, and indicators for holidays and special events. Understanding how these factors affect usage patterns is essential. For instance, commuter behavior during weekdays may differ significantly from leisure-oriented usage on weekends or holidays. This understanding will help ensure that resources are deployed effectively to meet demand of the masses.

In this study, we employ regression techniques and advanced machine learning models to analyze and predict demand. The regression framework allows for interpretable analysis of the relationships between variables, while machine learning models offer enhanced predictive performance by capturing non-linear interactions and complex patterns in the data. Additionally, extensive exploratory data analysis (EDA) is conducted to visualize trends, identify outliers, and ensure data quality. By accurately forecasting bike demand, we aim to optimize resource allocation, reduce inefficiencies, and ultimately improve the overall user experience of Ddareungi. This research has the potential to guide operational decisions that enhance service reliability, reduce wait times, and better match bike availability to actual demand.

Dataset Overview

The dataset used in this project is a comprehensive record of bike-sharing demand in Seoul, specifically from the "Seoul Bike Sharing System" (Ddareungi), a widely utilized public bike-sharing program in the city. This dataset captures various factors that influence bike rental demand, offering valuable insights into urban mobility patterns. The data spans multiple months and includes both temporal and environmental variables, providing a rich resource for identifying trends and understanding the underlying factors that govern bike-sharing usage.

The dataset consists of the following key variables:

Dependent Variable:

- **Rented Bike Count**: The number of bikes rented during a given time period.

Independent Variables:

- **Date (yyyy-mm-dd)**: The specific date of the observation.
- **Hour**: The hour of the day when the rental occurred.

- **Temperature (°C)**: The ambient temperature in degrees Celsius.
- **Humidity (%)**: The percentage of humidity in the air.
- **Wind Speed (m/s)**: The wind speed in meters per second.
- **Visibility (10m)**: The visibility range in 10-meter units.
- **Dew Point Temperature (°C)**: The temperature at which air becomes saturated with moisture.
- **Solar Radiation (MJ/m²)**: The amount of solar radiation in megajoules per square meter.
- **Rainfall (mm)**: The amount of rainfall in millimeters.
- **Snowfall (cm)**: The amount of snowfall in centimeters.
- **Seasons**: A categorical variable indicating the season (Winter, Spring, Summer, Autumn).
- **Holiday**: A binary indicator specifying whether the day was a holiday (Holiday/No Holiday).
- **Functioning Day**: A binary variable indicating whether the day was a functioning day for the bike-sharing system (working day/holiday).

This dataset serves as a foundation for exploring the relationship between various environmental and temporal factors and bike-sharing demand. By analyzing these variables, we can gain insights into the patterns of bike usage in relation to weather conditions, time of day, and other contextual factors.

	Date	Rented.Bike.Count	Hour	Temperature..C.	Humidity...	Wind.speed..m.s.	Visibility..10m.	Dew.point.temperature..C.	Solar.Radiation..MJ.m2.	Rainfall.mm.	Snowfall..cm.	Seasons	Holiday	Functioning.Day
1	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0	0	0	Winter	No Holiday	Yes
2	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0	0	0	Winter	No Holiday	Yes

Exploratory Data Analysis (EDA)

In the Exploratory Data Analysis (EDA) phase, we conducted a thorough examination of the dataset to understand its structure, identify key patterns, and prepare it for modeling. Below is a summary of the steps taken during the EDA process:

A. Data Inspection

The initial data inspection focused on assessing the structure and integrity of the dataset. It contained 8,760 observations, representing hourly bike rental data for an entire year, and included 14 variables. These variables consisted of both numerical and categorical data. The numerical variables, such as Rented Bike Count (RBC), Temperature, Humidity, and Wind Speed, were appropriately identified for quantitative analysis. The categorical variables, including Seasons, Holiday, and Functioning Day, represented distinct attributes of the bike-sharing system and were correctly classified as categorical data.

During the inspection, no missing values were found in the dataset, confirming its completeness and suitability for analysis. All variables were appropriately classified by their data type, ensuring they were ready for further processing. To enhance the clarity and accessibility of the dataset, column names were updated to more descriptive labels. For example, Rented.Bike.Count

was renamed to RBC, and Temperature..C. was changed to Temp. These modifications improved the dataset's interpretability and made it easier to work with in subsequent stages of the analysis.

B. Descriptive Statistics

The dataset contains 8,760 hourly observations, with Rented Bike Count (RBC) ranging from 0 to 3,556. The mean RBC is 704.6, with a median of 504.5, indicating that most bike rental counts are moderate, though some hours experience significantly higher demand. The Temperature variable varies from -17.8°C to 39.4°C, with a mean of 12.88°C, reflecting seasonal temperature fluctuations.

The Humidity ranges from 0% to 98%, with an average of 58.23%, and the Wind Speed spans from 0 m/s to 7.4 m/s, indicating moderate wind conditions on average. Visibility is mostly around 1,698 meters, suggesting typical urban visibility levels.

The Solar Radiation, Rainfall, and Snowfall variables show relatively low values, with solar radiation peaking at 3.52 MJ/m², and both rainfall and snowfall having numerous zero values, suggesting that the dataset predominantly covers non-precipitative periods.

By reviewing the summary statistics, we were able to identify potential outliers, skewness, and unusual patterns in several variables, which could impact the modeling process. These insights are crucial for refining the dataset and ensuring that the subsequent analysis produces accurate and reliable results.

Additional time-based features were derived from the Date column, including Day, Month, Year, and Weekday, providing insights into temporal patterns in bike rental activity.

Overall, the dataset provides a broad view of bike rental demand, with significant variability across weather conditions and time-related factors, while also highlighting areas that may require further attention before modeling.

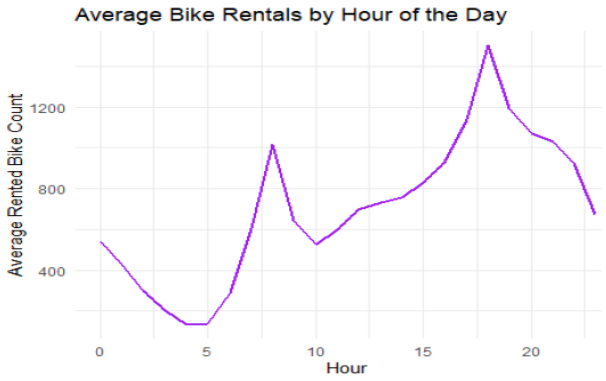
C. Bivariate Analysis

We performed bivariate analysis to explore the relationships between the dependent variable (Rented Bike Count) and key independent variables. Visualizations were created to examine how weather conditions, time-related features (e.g., hour of the day, day of the week), and holiday indicators influence bike rental demand.

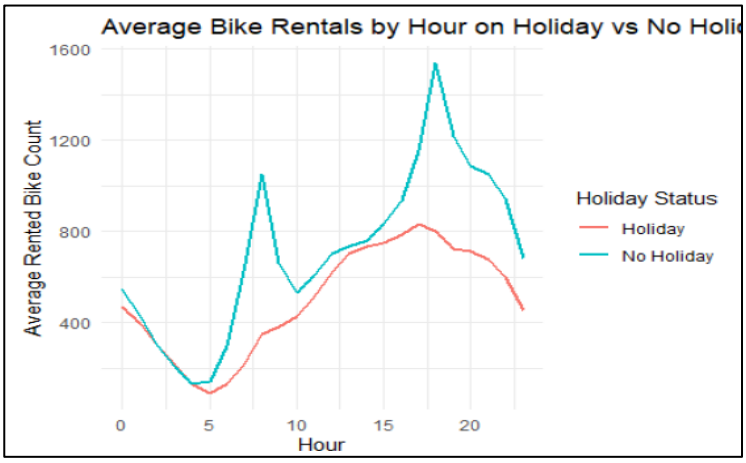
1. **Average Rented Bike Counts (RBC) for Each Day of the Week and RBC vs Weekends/Weekdays:** The graph illustrates the average number of bike rentals throughout the day, comparing weekdays and weekends. On weekdays, there is a clear bimodal pattern, with two distinct peaks. The first peak occurs during the morning hours (around 7-9 AM), likely corresponding to commuter activity. The second, more pronounced peak appears in the evening (around 6-8

PM), indicating another surge in rentals, possibly linked to the evening commute. In contrast, weekends exhibit a relatively flatter trend, with bike rentals steadily increasing from the morning and peaking in the early afternoon (around 2-3 PM). This suggests that weekend rentals are more associated with leisure or recreational activities. Late-night rentals (after 10 PM) are minimal and remain similar across both weekdays and weekends. Overall, this bivariate analysis highlights how the type of day significantly influences hourly bike rental patterns, offering insights into user behavior that could inform operational planning and marketing strategies.

2. **Average Rented Bike Counts (RBC) vs Hour:** The average number of bike rentals varies significantly by hour of the day, showing a distinct bimodal pattern. Rentals increase sharply during the morning hours (around 7-9 AM), likely driven by commuter activity. After this peak, there is a noticeable decline during midday, followed by a gradual rise in the afternoon. The highest number of rentals occurs in the evening (around 6-8 PM), corresponding to the evening commute. After this second peak, rentals steadily decline into the late-night hours. This pattern highlights the influence of time of day on bike usage, with clear peaks during typical commuting hours, indicating a strong association with work-related travel.

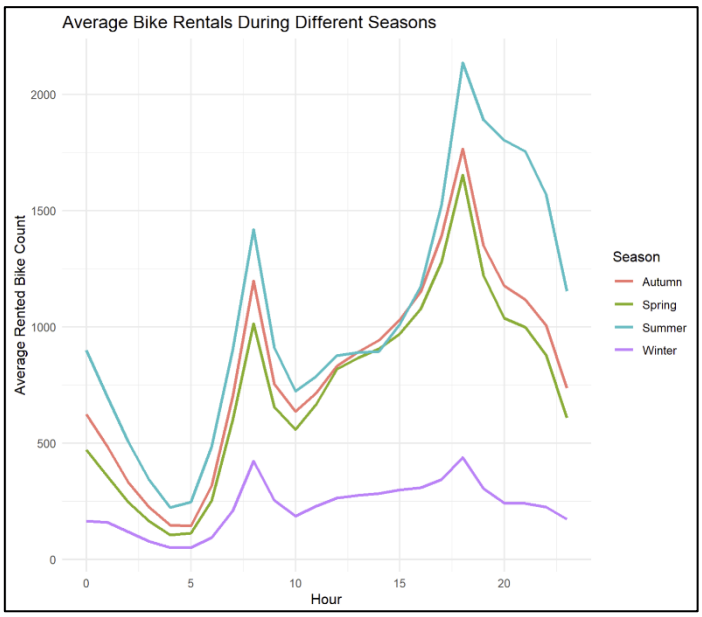


3. **RBC vs Holiday/No Holiday:** The average number of bike rentals differs significantly between holidays and non-holidays. On non-holidays, there is a clear bimodal trend with two distinct peaks: the first in the morning (around 7-9 AM), likely reflecting commuter activity, and the second, more pronounced peak in the evening (around 6-8 PM), associated with the evening commute. On holidays, the trend is relatively smoother, with rentals steadily increasing throughout the morning and peaking in the early afternoon (around 2-3 PM) before gradually declining into the evening. Overall, non-

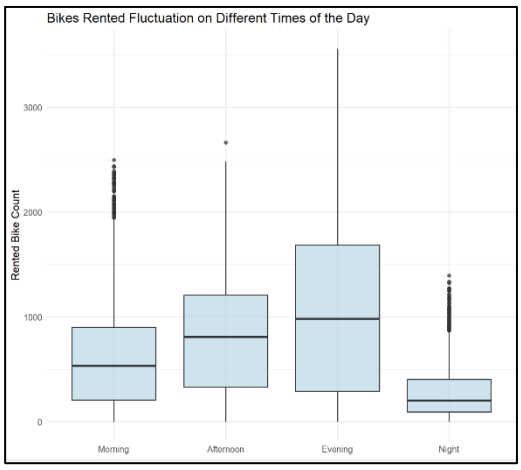


holidays show higher rental activity during commuting hours, while holidays exhibit a flatter pattern with moderate usage throughout the day, reflecting a shift toward recreational or leisure usage.

4. **RBC vs Seasons:** The analysis highlights the variation in average bike rentals throughout the day across different seasons. Two distinct peaks are observed during the morning (around 8 AM) and evening (around 5–6 PM), which align with typical commuting hours. Summer and autumn show the highest rental activity, particularly during the evening peak, while spring follows a similar pattern with slightly lower rental counts. In contrast, winter exhibits consistently lower rentals throughout the day, with minimal peaks, likely due to less favorable weather conditions. Overall, the findings indicate a strong relationship between bike rental trends, time of day, and seasonal changes.



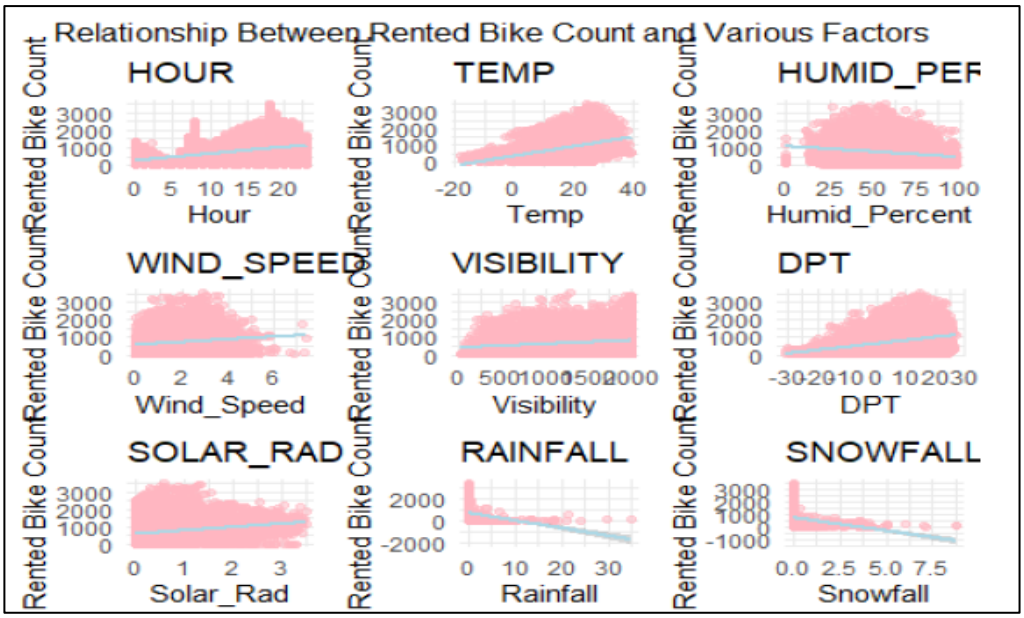
5. **RBC during Different Times of the Day:**



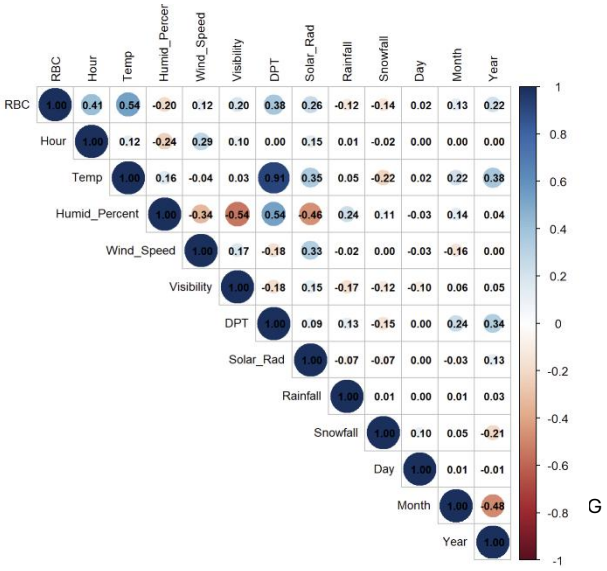
The analysis illustrates the distribution of bike rentals across different times of the day: morning, afternoon, evening, and night. Bike rentals are most variable in the evening, with a wider range and higher median compared to other periods, indicating it is the busiest time for rentals. Afternoon rentals display moderate variability and a slightly lower median. Morning rentals are relatively consistent, with fewer extreme values, while nighttime exhibits the lowest rental counts and minimal variability. This trend suggests that bike rental activity peaks in the evening, likely due to commuting or leisure activities, and decreases significantly during the night.

D. Multivariate Analysis:

1. Pairwise Scatter Plots- Relationship Between Rented Bike Count and Various Factors: This grid of scatter plots shows the relationship between RBC and several numerical variables. Rented bike count (RBC) shows a positive relationship with temperature, solar radiation, and dew point temperature, indicating that favorable weather conditions tend to increase bike rentals. Humidity, wind speed, rainfall, and snowfall, on the other hand, demonstrate negative correlations with RBC, suggesting that unfavorable weather conditions reduce bike rentals. Visibility has a weak positive impact, showing minimal influence on the number of rentals. The hourly trend exhibits a non-linear pattern, with prominent peaks during commuting hours, reflecting increased demand at specific times of the day. These patterns underscore the importance of weather and time in shaping rental behavior.



2. Correlation Matrix: This visualization displays the pairwise correlation coefficients between numerical variables in the dataset. The correlation matrix presents the pairwise relationships between numerical variables in the dataset, with correlation coefficients ranging from -1 to 1. A coefficient of 1 indicates a perfect positive correlation, -1 represents a perfect negative correlation, and values closer to 0 imply no significant linear relationship between the variables.



Notably, the correlation between RBC and Hour is moderate and positive (0.64), suggesting a trend where higher RBC values are associated with later hours. A perfect negative correlation of -1.00 is observed between Humid_Percent and Temp, indicating an inverse relationship where increases in temperature correspond to decreases in humidity. The correlation between Temp and DPT is very high at 0.91, which indicates that both variables are

likely capturing similar information. This could lead to multicollinearity if both are included in a model. Additionally, there is a high correlation (0.54) between Humid_Percent and DPT, reflecting the common relationship between humidity and dew point temperature in meteorological contexts.

A moderate positive correlation of 0.35 is observed between Solar_Rad and Temp. While not as strong as some of the other correlations, this value might still raise concerns about collinearity depending on the specific context of the analysis.

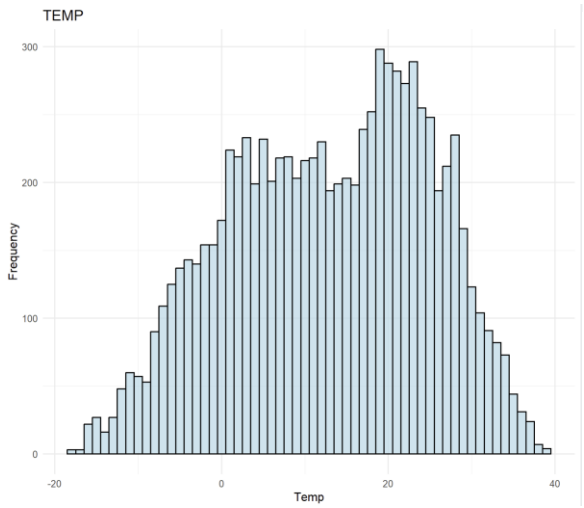
To address potential multicollinearity, we will examine the Variance Inflation Factor (VIF) for each variable to assess the extent of multicollinearity. Based on the results, we may consider removing one of the highly correlated variables through appropriate testing to ensure that the model remains stable and interpretable.

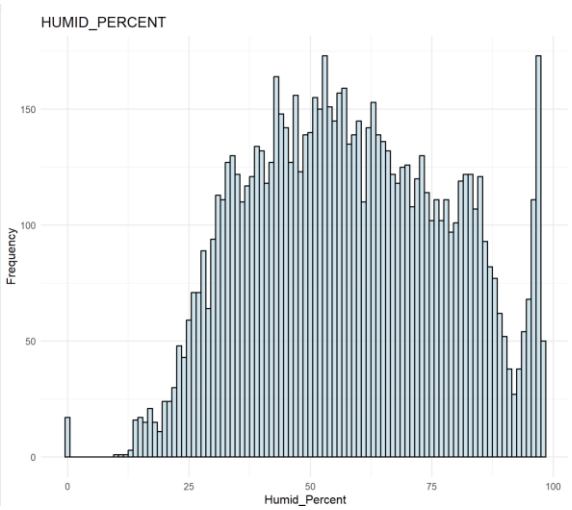
Other significant correlations include a strong positive correlation of 0.91 between DPT and Solar_Rad, indicating that higher dew points are linked with higher solar radiation. Additionally, a moderate negative correlation of -0.54 between Wind_Speed and Humid_Percent suggests that higher wind speeds tend to coincide with lower humidity. A moderate positive correlation of 0.53 between Rainfall and Solar_Rad implies that higher rainfall is associated with increased solar radiation.

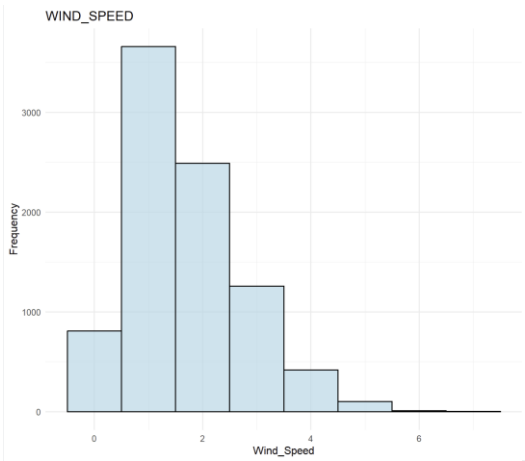
Many other variables, including Year and several weather-related metrics, show weak or no significant correlation, reflecting the absence of linear relationships in those pairs. This matrix provides valuable insights into the linear dependencies between variables, which will guide the next steps in our analysis, particularly regarding model formulation and multicollinearity mitigation.

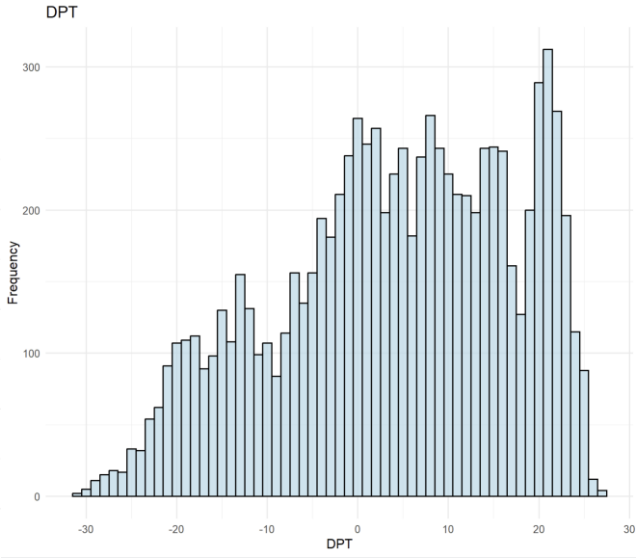
E. Univariate Analysis:

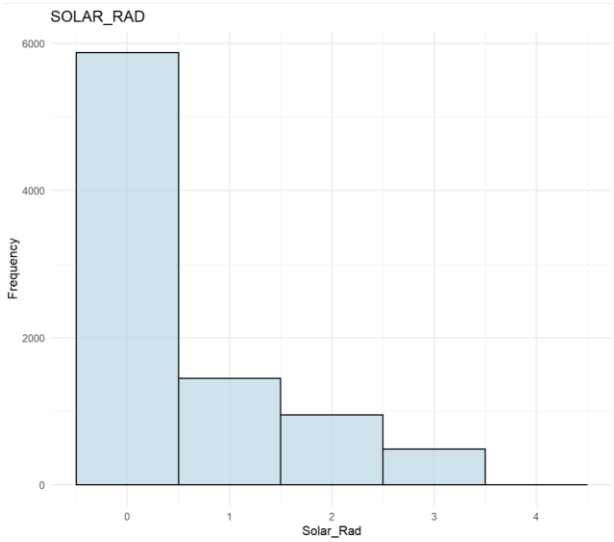
The distribution of bike rental counts (RBC), our dependent variable, is positively skewed, with a skewness value of 1.153. This indicates that the bike-sharing system is underutilized during many hours, with higher demand observed during specific periods, such as rush hours or on days with favorable weather conditions. The histogram of bike rental counts reveals a heavy right tail, highlighting a few high-demand periods (e.g., rush hours, pleasant weather days) amidst a majority of low-demand hours. To address this positive skewness, we may apply a transformation, such as taking the logarithm of the bike rental count, to normalize the distribution before modeling.

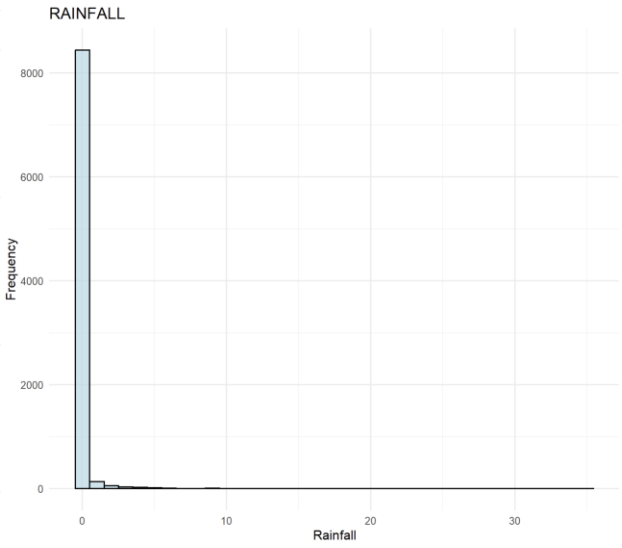

TEMP
Frequency
Temp

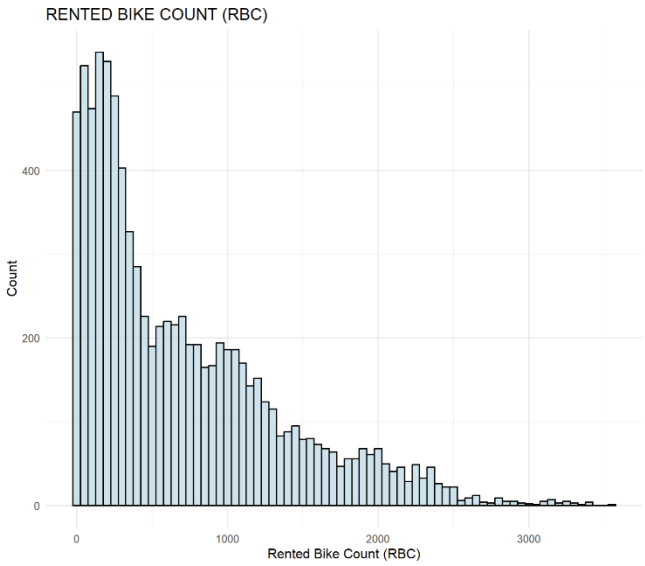

HUMID_PERCENT
Frequency
Humid_Percent


WIND_SPEED
Frequency
Wind_Speed


DPT
Frequency
DPT


SOLAR_RAD
Frequency
Solar_Rad


RAINFALL
Frequency
Rainfall



The skewness in RBC could potentially violate model assumptions, particularly in regression models that assume normally distributed error terms. While this may present an issue during model evaluation, we will revisit this problem in later stages if it results in model assumption violations and apply appropriate remedies as necessary.

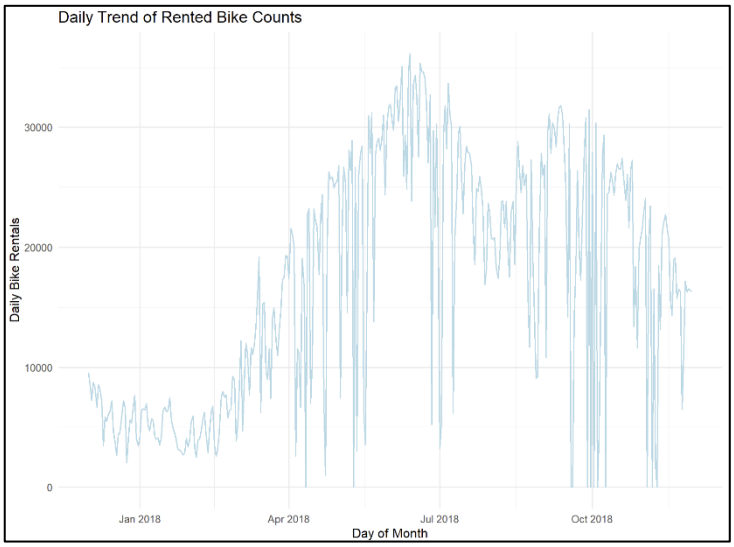
Additionally, we generate individual boxplots for numerical variables such as Temperature, Humidity, and Wind Speed to visualize their distributions. The distribution plots (histograms) for each numerical column provide valuable insights into how data is spread across different features.

1. **Hour**: The distribution is nearly uniform, with equal data points across all hours of the day, indicating consistent bike rental activity throughout the day.
2. **Temperature**: The temperature data spans a reasonable range, with higher frequencies observed in moderate temperature ranges, suggesting that bike rentals are more frequent under moderate weather conditions.
3. **Humidity**: This variable shows a slight skew, with most data points concentrated at higher humidity levels, indicating that bike rentals tend to occur more frequently when humidity is higher.
4. **Wind Speed**: The distribution is right-skewed, with most data points clustered at lower wind speeds and fewer occurrences of high wind speeds, suggesting that higher wind speeds are less common in the dataset.
5. **Rainfall and Snowfall**: Both features exhibit heavily right-skewed distributions, reflecting the relatively rare occurrence of these weather conditions and their limited impact on bike rentals.
6. **Solar Radiation**: Solar radiation data shows varying distributions, with values concentrated at lower levels, indicating that lower solar radiation is more common during the observed periods.

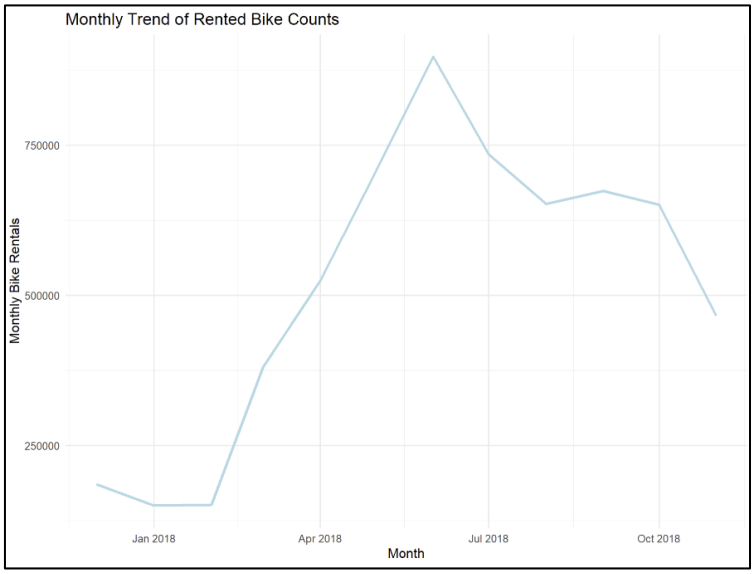
These distribution insights help in understanding the underlying patterns of the data and will inform the modeling process, ensuring that any data preprocessing or transformation steps are appropriately implemented.

F. Time Based Analysis

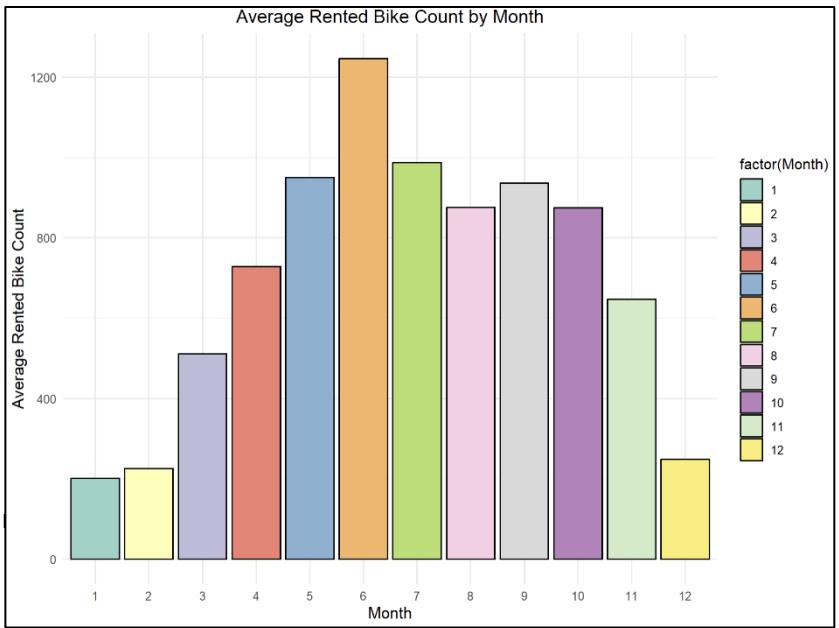
Time-based analysis is crucial for exploring how the target variable, Rented Bike Counts (RBC), varies across time-related features such as days and months. This analysis helps uncover patterns tied to specific periods, including seasonal trends and daily cycles, which are essential for understanding rental behavior.



The graph illustrating daily trends shows significant day-to-day fluctuations in rental counts, with noticeable spikes and dips that might be influenced by factors such as weather, events, or holidays. A seasonal trend is evident: bike rentals increase steadily from late winter to summer, peaking around mid-year. After the peak in summer, rentals start to decline gradually during the fall and drop further towards the end of the year. This trend highlights that daily bike rentals are strongly influenced by weather and seasonality. Warmer months encourage outdoor activities and increase bike rentals, while colder months see reduced usage.



The graph summarizing monthly trends provides a broader overview of the data. Rentals are at their lowest in January and February, likely due to cold weather reducing outdoor activity. Starting in March, rentals increase sharply, peaking around June or July, aligning with warmer weather and potential vacation periods. Rentals decline after the summer peak, with a steady decrease from August through December. This monthly trend underscores a strong seasonal pattern, with higher rentals during warmer months and lower rentals during colder months. This broad trend is consistent with the daily analysis, emphasizing the role of weather and seasonality in shaping rental behavior.



The combination of daily and monthly trends offers valuable insights into user behavior. The rental patterns clearly align with seasonal changes, which is important for optimizing operations, such as resource allocation and bike maintenance schedules. Rental businesses

can focus their marketing campaigns during peak months to attract more users and implement promotions in off-peak months to boost rentals. Staffing, inventory, and maintenance schedules can be adjusted based on expected demand during different months and seasons. These findings provide a foundation for understanding rental behavior and planning accordingly to maximize efficiency and user satisfaction.

Feature Engineering

In the Feature Engineering and Feature Selection process, several critical steps were undertaken to ensure that the data was clean, relevant, and effective for making accurate predictions. Variance Inflation Factor (VIF) analysis was conducted to identify and address multicollinearity among predictor variables. The initial VIF results revealed that variables like Temp, Humid_Percent, and DPT exhibited extremely high VIF scores (e.g., Temp: 87.73, DPT: 115.69), indicating strong multicollinearity. To mitigate this, the DPT variable was removed, and VIF scores were recalculated. This adjustment significantly reduced multicollinearity, resulting in a more robust and reliable dataset for modeling.

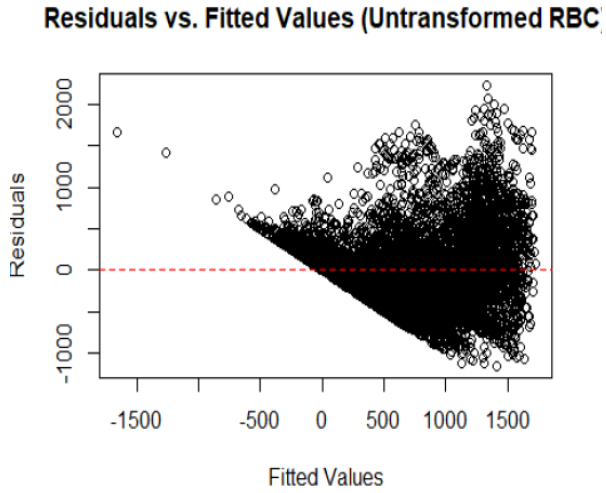
```
print(vif_result)

##          Hour          Temp Humid_Percent    Wind_Speed    Visibility
##      1.188483     87.735664     20.448556      1.303892      1.673881
##           DPT     Solar_Rad      Rainfall      Snowfall           Day
##    115.697293      2.034245      1.084795      1.130244      1.044487
##         Month          Year
##      1.880074      2.013793
```

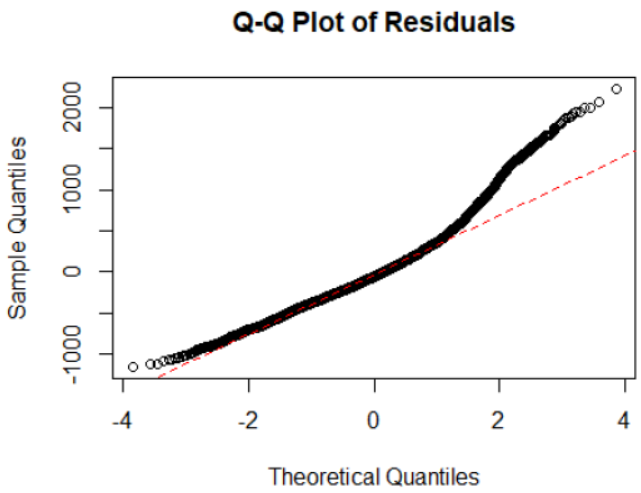
```
print(vif_result_1)

##          Hour          Temp Humid_Percent    Wind_Speed    Visibility
##      1.186224      2.173417      2.627656      1.301828      1.663552
##     Solar_Rad      Rainfall      Snowfall           Day         Month
##      1.937487      1.070908      1.125286      1.044456      1.880066
##          Year
##      2.013747
```

Categorical variables were transformed using One-Hot Encoding to ensure compatibility with machine learning models. Binary variables such as Holiday and Functioning_Day were encoded into 0 and 1, representing “No Holiday/No” and “Holiday/Yes,” respectively. For multi-class variables like Seasons and Weekday, One-Hot Encoding was applied, generating binary columns for each category while dropping the first dummy column to avoid introducing multicollinearity.

Outliers were systematically identified and handled to improve model performance and reduce distortion in predictions. Initial diagnostics using Residuals vs. Fitted Values plots and Q-Q plots revealed signs of non-normality, heteroscedasticity, and heavy-tailed residuals. Further statistical tests, including the Breusch-Pagan test for homoscedasticity and the Kolmogorov-Smirnov test for normality, were performed to confirm these findings. Influential points and outliers were detected using Cook’s Distance and Standardized Residuals, with thresholds of 4/n (where n is

the number of observations) and absolute residuals greater than 2, respectively. These flagged outliers were removed, resulting in significant improvements in model performance, including a reduction in residual standard error and an increase in the R-squared value from 55.84% to 61.86%. Post-removal diagnostics using Q-Q plots and Residuals vs. Fitted Values plots indicated better adherence to normality and linearity assumptions, although model assumption violations persisted.





Through these feature engineering and selection steps, the dataset was refined to maximize its predictive power while addressing critical issues such as multicollinearity, data inconsistency, and outlier influence. This systematic approach significantly improved the accuracy, stability, and interpretability of the final models.

Feature Selection

Feature selection is a vital step in improving model performance by removing irrelevant or redundant features, simplifying the model, and enhancing predictive accuracy. In this analysis, feature selection was performed using **Stepwise Selection** based on the **Akaike Information Criterion (AIC)**. AIC is a widely used metric that helps strike a balance between model fit and complexity by penalizing the inclusion of unnecessary predictors. Throughout the stepwise selection process, the AIC was evaluated at each step to minimize it, ensuring an optimal balance between goodness of fit and model simplicity. As a result, variables such as Weekday_Monday, Holiday, and Month were removed, and the AIC reached its minimum value of 99092.56. The final model retained the most significant predictors for further analysis and prediction. Although the adjusted R^2 showed a slight increase, it remained relatively unchanged, indicating that further transformations might be necessary.

```
Step:  AIC=99092.56
RBC ~ Hour + Temp + Humid_Percent + Wind_Speed + Solar_Rad +
    Rainfall + Snowfall + Holiday + Functioning_Day + Day + Month +
    Year + Seasons_Spring + Seasons_Summer + Seasons_Winter +
    Weekday_Monday + Weekday_Sunday
```

The key predictors retained after stepwise selection include: Hour, Temp, Humid_Percent, Visibility, DPT, Rainfall, Holiday, Functioning_Day, Month, Year, Seasons_Spring, Seasons_Summer, Seasons_Winter, Weekday_Monday, Weekday_Saturday, and Weekday_Sunday. These variables were identified as the most important for explaining the variance in bike rental counts.

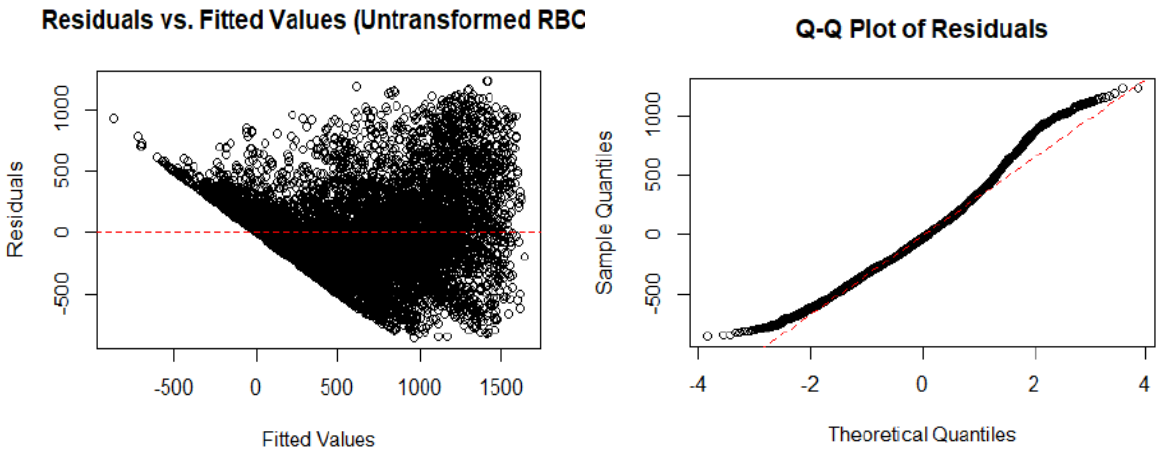
To address potential model assumption violations, particularly with skewness and heteroscedasticity, a **Box-Cox transformation** was applied to the dependent variable, RBC (Rented Bike Count). The goal was to normalize the distribution of RBC and stabilize its variance. However, after evaluating the model assumptions, the transformation did not resolve the violations and even worsened the R^2 value. As a result, we decided to drop the transformation and instead trained the model using the subset of variables identified through stepwise selection.

While the model assumptions (LINE: Linear relationship, Independence, Normality, and Equal variance) were not fully met in the final model, it performed better than the model with the transformed dependent variable. This decision was made based on the overall model performance and the trade-off between assumption violations and predictive accuracy. Thus, the subset of selected variables was used to train the remaining models in the project, ensuring a more stable and interpretable model despite some assumption violations.

Model Estimation

Number of influential points: 393
Number of outliers: 450
Number of overlapping points (outliers and influential): 321

In this analysis, several machine learning models were evaluated to predict the target variable, Rented Bike Count (RBC). The dataset was split into training and testing subsets to ensure unbiased evaluation of the models' performance. The models considered include Linear Regression, Ridge Regression, Lasso Regression, Random Forest, and XGBoost. Each model was trained on the training dataset and evaluated on the testing dataset. The performance metrics used were Mean Squared Error (MSE), which measures the average squared difference between predicted and actual values, and R-squared, which indicates the proportion of variance in the target variable explained by the model. These metrics provided insights into the models' predictive accuracy and goodness of fit, enabling a comparative analysis of their effectiveness.



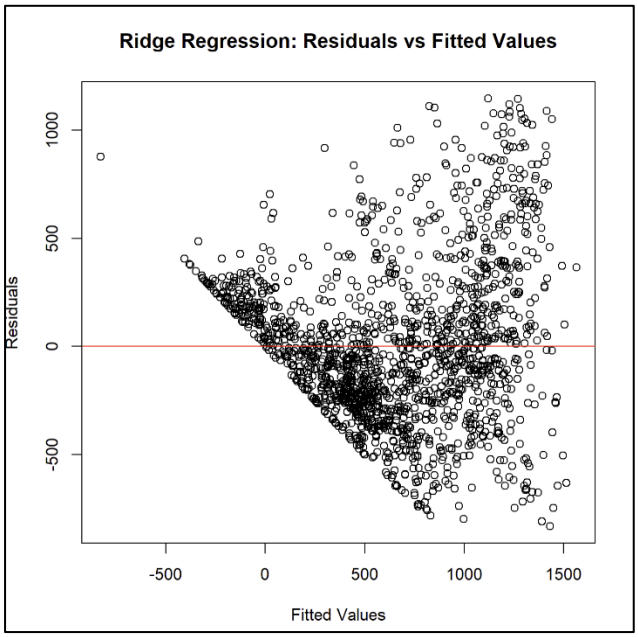
A. Data Preparation and Partitioning: To ensure the reproducibility of the results, the random number generator seed was set using the function set.seed(42). This guarantees that the random processes, such as data splitting and model initialization, produce consistent results across multiple runs of the analysis.

The dataset was divided into training and testing sets using the createDataPartition function from the caret package. This function ensures a stratified split, maintaining the distribution of the target variable in both subsets. Specifically, 80% of the data was allocated to the training set, while the remaining 20% was reserved for testing. This approach allows the model to be trained on a substantial portion of the data, while providing an independent test set for evaluating model performance on unseen data, thus helping to assess the generalizability of the model.

B. Ridge Regression (L2 Regularization):

To build the predictive model, a design matrix was first created for the training data. This was accomplished using the model.matrix function, which constructs a matrix of predictor variables while excluding the intercept term. The design matrix, referred to as X_train, serves as the input for the Ridge regression model and ensures that all predictors are appropriately represented for the model fitting process.

Next, to optimize the performance of the model and reduce the risk of overfitting, k-fold cross-validation was performed using the cv.glmnet function. This function facilitates the tuning of the regularization parameter, denoted as lambda (λ), which controls the strength of regularization applied in Ridge regression. The cross-validation procedure, using a default of 10 folds, splits the training data into ten subsets. Each subset is used iteratively as a validation set while the remaining nine subsets are used for training. This process is repeated for different values of lambda, and the optimal lambda is selected based on the value that minimizes the cross-validation error. This technique helps to identify the best balance between bias and variance, ensuring a robust model.



Once the optimal lambda value was determined, the Ridge regression model was fitted to the training data using this value. The fitting process involves solving for the coefficients of the predictors, which minimizes the residual sum of squares while applying the regularization penalty specified by the chosen lambda. This regularization prevents the model from overfitting by shrinking the coefficients of less important predictors towards zero, thereby simplifying the model.

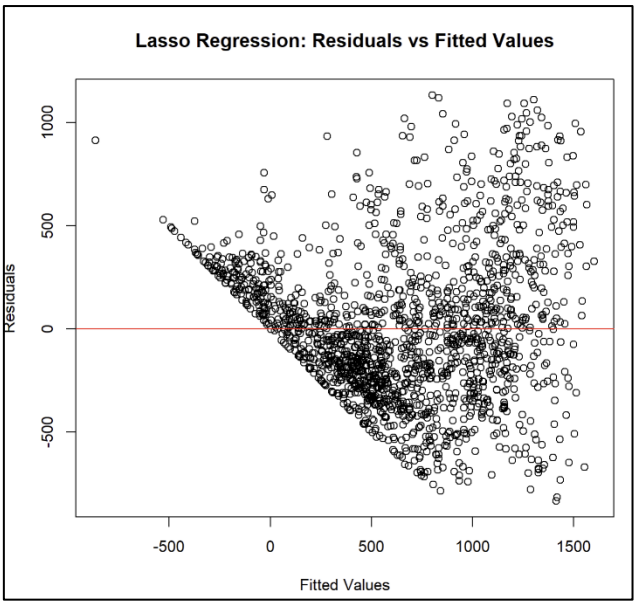
After the model was fitted, it was used to predict the target variable, which in this case was RBC, on the testing dataset. The performance of the model was then evaluated using two commonly used metrics: Mean

Squared Error (MSE) and R-squared. MSE quantifies the average squared difference between the observed actual outcomes and the predictions made by the model, providing a measure of the model's prediction accuracy. Meanwhile, R-squared indicates the proportion of variance in the target variable that is explained by the model, offering a sense of how well the model captures the underlying patterns in the data. Together, these metrics provide a comprehensive evaluation of the model's effectiveness and predictive power.

C. Lasso Regression (L1 Regularization)

The Lasso regression model, which incorporates L1 regularization, was evaluated as part of the model selection process. To identify the optimal regularization parameter, lambda (λ), we employed cross-validation using the cv.glmnet function from the glmnet package. This function performs k-fold cross-validation to determine the value of lambda that minimizes the cross-validation error. The optimal lambda value was selected based on the lowest mean cross-validation error, ensuring that the model would generalize well to unseen data without overfitting.

Once the optimal lambda was identified, the Lasso regression model was then fitted using this value. The fitting process involved training the model on the training dataset, where the L1 regularization enforced sparsity by driving less important coefficients toward zero, thereby performing both variable selection and regularization. This approach helps improve the model's interpretability and prevent overfitting by limiting the influence of irrelevant features.



Following the model fitting, predictions were generated on the test set to evaluate the performance of the Lasso regression model. The predicted values were compared against the actual test data to assess the accuracy of the model's predictions.

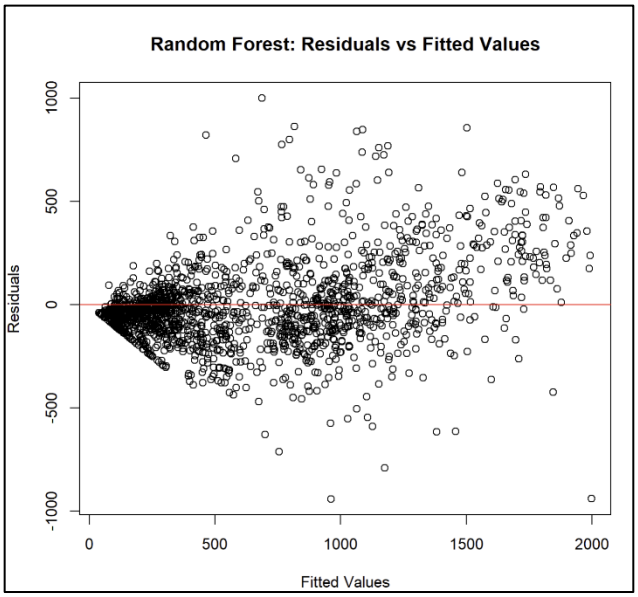
The performance of the model was quantified using two common evaluation metrics: Mean Squared Error (MSE) and R-squared (R^2). The MSE provided insight into the average squared difference between the predicted and actual values, with lower values indicating better predictive accuracy. The R-squared value, on the other hand, represented the proportion of variance in the target variable explained by the model, with values closer to 1 indicating a stronger fit. Both metrics were used to assess how well the Lasso regression model performed in predicting bike-sharing demand in the test data.

D. Random Forest:

To model the bike-sharing demand, the Random Forest algorithm was employed due to its robustness and ability to capture complex, non-linear relationships within the data. The model

was trained using the randomForest function from the randomForest package. In this process, 500 trees were constructed (as specified by the parameter ntree = 500), which ensures the model can effectively learn from a large number of decision trees, reducing overfitting and enhancing predictive performance. Additionally, a maximum of three variables (mtry = 3) were selected for each split, which helps control the complexity of the model and prevents it from becoming too sensitive to noise in the dataset. By setting this parameter, the model could focus on the most relevant variables at each decision point, improving generalization.

Once the model was trained, predictions were made on the test dataset, which consisted of 20% of the original data that had been reserved for this purpose. These predictions were compared to the actual values from the test set to evaluate the model's performance.

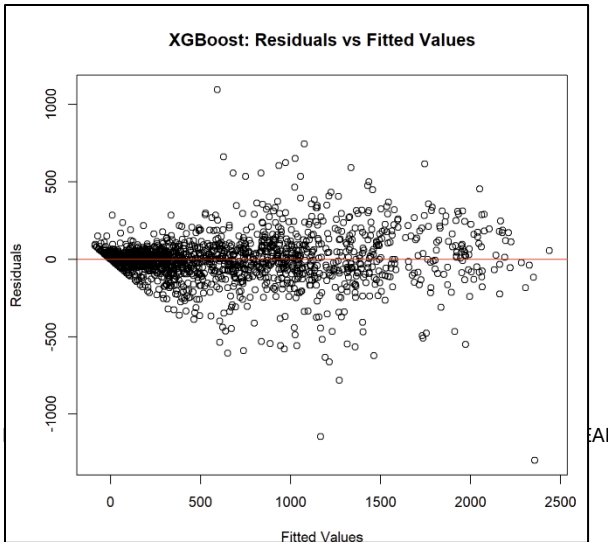


The performance of the Random Forest model was assessed using two key metrics: Mean Squared Error (MSE) and R-squared. The MSE provides a measure of the average squared difference between the predicted and actual values, quantifying the accuracy of the predictions—the lower the MSE, the better the model's performance. On the other hand, R-squared measures the proportion of variance in the target variable that is explained by the model. A higher R-squared value indicates a better fit of the model to the data.

Through these evaluation metrics, we were able to gauge the effectiveness of the Random Forest model in predicting bike-sharing demand in Seoul. These metrics offer insights into the model's predictive accuracy and its ability to generalize to new, unseen data, which is crucial for its potential application in real-world bike-sharing systems.

XGBoost

To prepare the dataset for use in the XGBoost model, the data was first transformed into a format suitable for the algorithm using the xgb.DMatrix function. This function efficiently handles both the feature matrix and the target variable, converting them into a format that XGBoost can process. The transformation also enables the model to work more efficiently, reducing the computational overhead typically associated with large datasets.



Next, the parameters for the XGBoost model were specified. The objective function chosen for this regression task was "reg:squarederror", which is commonly used for continuous prediction tasks. This objective function

minimizes the squared error between the predicted and actual values, a standard approach for regression problems. In addition, several important model hyperparameters were set. The max_depth parameter was set to 6, which controls the maximum depth of the individual trees in the boosting process. A deeper tree can model more complex relationships in the data but may also increase the risk of overfitting. The learning rate (eta) was set to 0.1, which determines the step size at each iteration while moving toward a minimum. A lower learning rate can help in improving the model's generalization by making smaller, more incremental updates. Additionally, subsample ratios were adjusted to prevent overfitting and to ensure that each boosting round uses a random subset of the data, further enhancing the model's robustness.

Once the parameters were defined, the XGBoost model was trained using the xgb.train function. This function initiates the model fitting process by performing a set number of boosting rounds, in this case, 500. Each boosting round involves the model adjusting its weights and improving its predictions based on the errors from the previous round. The number of rounds was chosen to ensure that the model had enough iterations to converge while avoiding excessive training time.

Finally, predictions were made on the test set using the trained XGBoost model. This evaluation step involved applying the learned model to the unseen data, allowing us to assess the model's predictive performance. The output of this process was a set of predicted values that were compared against the actual test set values to evaluate the accuracy and generalization ability of the model.

Model Evaluation and Comparison

To evaluate the performance of the models, two key metrics were calculated: Mean Squared Error (MSE) and R-squared (R^2). The Mean Squared Error (MSE) is a commonly used metric that quantifies the average squared difference between the actual and predicted values. It provides an indication of how well the model fits the data, with a lower MSE indicating better model performance. The R-squared (R^2), on the other hand, measures the proportion of variance in the target variable that is explained by the model. R^2 values range from 0 to 1, where a value closer to 1 suggests a model that explains a large portion of the variability in the data. These metrics were computed for each of the models under consideration and compared to determine which model provided the most accurate predictions.

In addition to these numerical metrics, residual plots were generated for each model. Residual plots display the residuals (the difference between the observed and predicted values) on the y-axis against the fitted values (the predicted values) on the x-axis. These plots are an essential diagnostic tool for assessing model fit. Ideally, the residuals should be randomly scattered around zero, without any discernible patterns. If the residuals exhibit patterns, such as a systematic curve or clustering, it suggests that the model may not have captured some underlying structure in the data, indicating potential improvements or model adjustments. A good model will exhibit residuals that appear random and are evenly distributed across the range of fitted values, implying that the model has adequately captured the data's relationships without overfitting or underfitting.

By analyzing both the MSE and R^2 values alongside the residual plots, we were able to assess the overall effectiveness of the models and identify any areas for improvement or model refinement.

Summary

After evaluating the models using MSE and R-squared, the following observations were made:

MODEL	MSE	R2	REMARKS
Linear Regression	125,235.4	61.8%	Linear regression provides a baseline model with moderate performance. It explains around 61.8% of the variance.
Ridge Regression	130,393	60.7%	Ridge regression slightly underperforms compared to linear regression, with a higher MSE and lower R-squared.
Lasso Regression	127,368	61.7%	Lasso regression performs similarly to Ridge, showing a slight improvement in MSE.
Random Forest	40,938.36	87.7%	Random Forest significantly improves predictive performance, showing much lower MSE and a higher R-squared.
XGBoost	24,433.74	92.6%	XGBoost outperforms all other models, with the lowest MSE and highest R-squared, demonstrating the best predictive accuracy.

Conclusion

The integration of Exploratory Data Analysis (EDA), feature engineering, and model development formed the backbone of our predictive framework for bike rental demand. By thoroughly examining the dataset during the EDA phase, we uncovered key patterns and trends in bike usage, such as the impact of weather conditions, temporal factors (e.g., time of day, seasonality), and contextual influences like holidays. These insights laid a strong foundation for selecting relevant features and preparing the data for model development.

Feature engineering played a pivotal role in enhancing the predictive power of our models. By creating additional features such as categorical encodings, interaction terms, and polynomial transformations of variables, we captured complex relationships that might otherwise go unnoticed. For instance, features like the interaction between temperature and humidity helped model the nuanced impact of weather on bike demand, while time-of-day variables captured cyclical trends in commuter and leisure activity.

The deployment of advanced machine learning techniques, particularly XGBoost, enabled us to achieve a highly accurate predictive model with an R-squared value of 92.6%. XGBoost's ability to model non-linear relationships and its inherent feature importance metrics provided not only exceptional predictive accuracy but also interpretability, allowing us to identify the most influential variables driving bike rental demand. This high level of performance underscores the robustness of our approach and its potential for real-world application.

These insights are highly actionable for city planners and bike-sharing operators. For instance, understanding peak demand periods and the factors driving those peaks can guide decisions on where and when to deploy bikes, ensuring optimal resource allocation. Additionally, the ability to predict demand accurately helps operators reduce operational inefficiencies, such as

overstocking bikes in low-demand areas or failing to meet demand in high-use zones. This optimization can enhance user satisfaction by reducing waiting times and ensuring bike availability where it's needed most.

Future Directions: While our model achieved impressive results, there are several avenues for further enhancement:

- **Incorporating Additional External Data Sources:** Integrating data such as traffic patterns, public event schedules, or nearby public transport availability could provide deeper insights into external factors influencing bike demand. For example, a major event in the city center might lead to a sudden spike in demand for nearby bike-sharing stations.
- **Real-Time Predictions:** Developing real-time prediction capabilities could significantly enhance the system's responsiveness. For instance, real-time weather updates or live traffic data could feed into the model to make dynamic adjustments to bike deployment strategies.
- **Expanding to Other Cities or Regions:** The current model is tailored to Seoul's bike-sharing system. Applying a similar framework to other cities or regions could validate its generalizability and highlight location-specific differences in bike rental behavior.
- **Advanced Modeling Techniques:** Exploring deep learning models or hybrid approaches (e.g., combining time-series forecasting with machine learning) could further boost predictive performance, especially in scenarios where complex temporal dependencies or non-linearities play a significant role.

In conclusion, this project demonstrates the power of integrating EDA, feature engineering, and advanced machine learning techniques to tackle real-world problems in urban mobility. The proposed framework not only delivers high predictive accuracy but also provides actionable insights for decision-makers, paving the way for smarter, more sustainable city planning.

References:

https://archive.ics.uci.edu/dataset/560/seoul+bike+sharing+demand (Dataset)

https://www.tandfonline.com/doi/full/10.1080/22797254.2020.1725789#d1e132

https://www.mathematicsgroup.com/articles/CMA-2-105.php

https://www.researchgate.net/publication/340978677_Seoul_Bike_Trip_Duration_Prediction_using_Data_Mining_Techniques

https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-021-01373-z

https://www.sciencedirect.com/science/article/abs/pii/S1366554517311560

https://ideas.repec.org/a/kap/transp/v41y2014i6p1205-1225.html

https://www.researchgate.net/publication/286379583_Effects_of_Built_Environment_and_Weather_on_Bike_Sharing_Demand_A_Station_Level_Analysis_of_Commercial_Bike_Sharing_in_Toronto

https://www.researchgate.net/publication/348974351_Machine_Learning_Approaches_to_Bike-Sharing_Systems_A_Systematic_Literature_Review

https://www.sciencedirect.com/science/article/pii/S2214140522000147