BIKE RIDE DEMAND

Forecasting bike ride demand with a dataset from Seoul.

25.10.2024.

AGENDA

01 Introduction

02 Data Overview

03 Data Preprocessing

04 Feature Selection

05 Modeling Approaches

06 Results & Conclusion

INTRODUCTION
BIKE RIDE DEMAND

- Objective: To develop a predictive model for bike rental demand.
- Significance: Improve operational efficiency for bike-sharing systems.
- Goal: Compare the effectiveness of different machine learning algorithms.

DATA OVERVIEW

Dataset Source: Public dataset

Description:

Historical bike rental counts
Weather conditions
Temporal data (hour, day, month, season)

Feature Name	Description	Data Type
`Date`	Date and timestamp of when the ride occurred	Categoric
`Rented Bike Count`	Count of bikes rented at each hour	Numeric
`Hour`	Hour of the day (0 to 23) when the ride occurred	Categoric
`Temperature(°C)`	Temperature in Celsius during the ride	Numeric
`Humidity(%)`	Humidity percentage during the ride	Numeric
`Wind speed (m/s)`	Wind speed in meters per second during the ride	Numeric
`Visibility (10m)`	Visibility measured in meters (the distance at which objects can be clearly seen)	Numeric
`Dew point temperature(°C)`	Dew point temperature in Celsius during the ride	Numeric
`Solar Radiation (MJ/m2)`	Solar radiation measured in megajoules per square meter during the ride	Numeric
`Rainfall(mm)`	Precipitation in millimeters during the ride	Numeric
`Snowfall (cm)`	Snow accumulation measured in centimeters during the ride	Numeric
`Seasons`	Season during which the ride occurred (e.g., Winter, Spring)	Categoric
`Holiday`	Indicates whether the day was a holiday	Categoric
`Functioning Day`	Day when the bike-sharing system is operational	Categoric

DATA
OVERVIEW

Dataset Source: Public dataset

Description:
Historical bike rental counts
Weather conditions
Temporal data (hour, day, month, season)

DATA PREPROCESSING STEPS

01

Handling Missing and Duplicate Values: Identifying and managing any missing or duplicate data points in the dataset.

02

Data Type Conversion: Converting data types to appropriate formats for analysis. (Date to datetime64[ns])

03

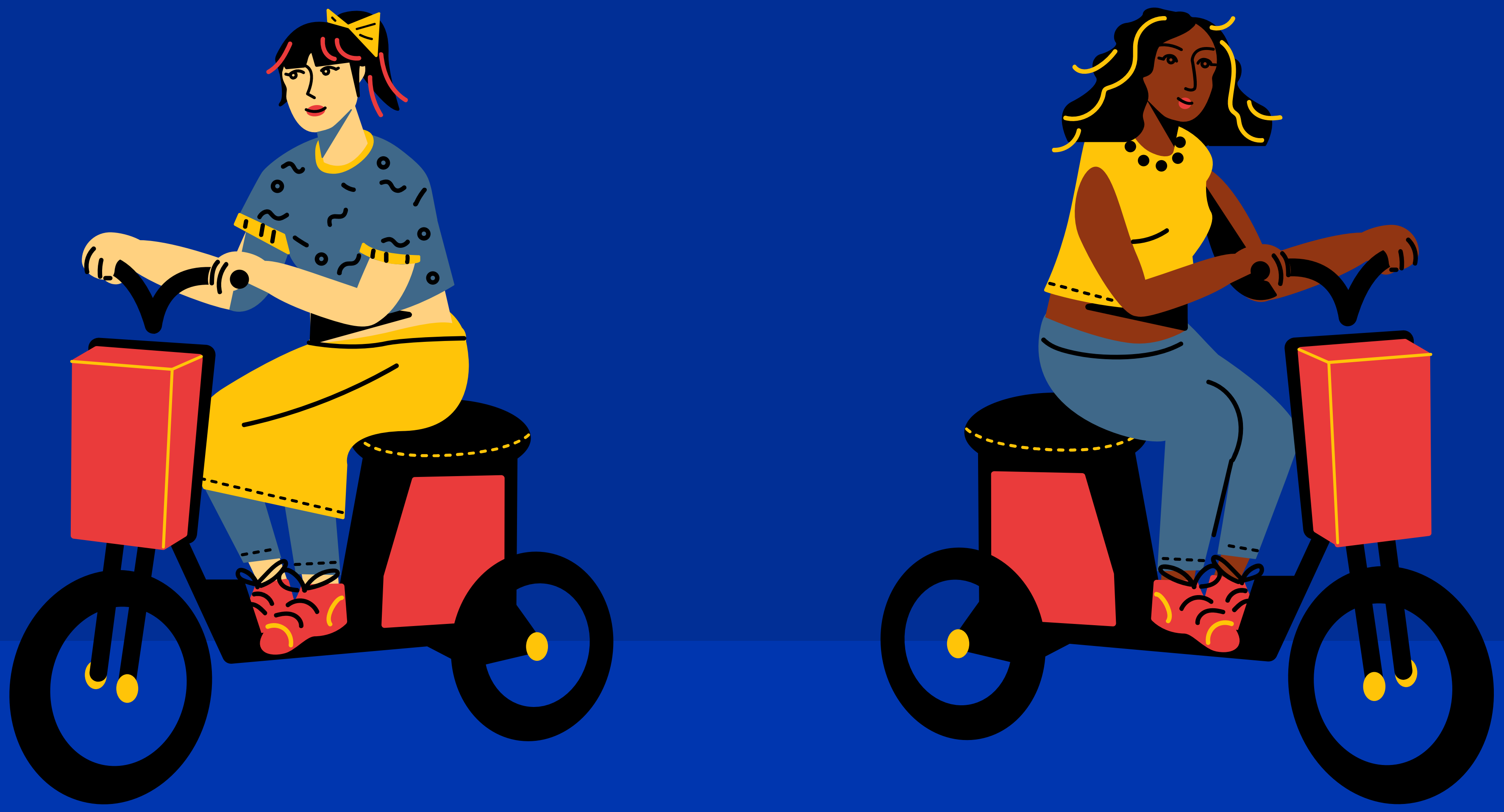
Data Splitting: Splitting data where needed to enhance further handling. (Namely Date to Day , Month and Year)

04

Data Profiling: Getting a comprehensive report on the data with y-data profiling ProfileReport.

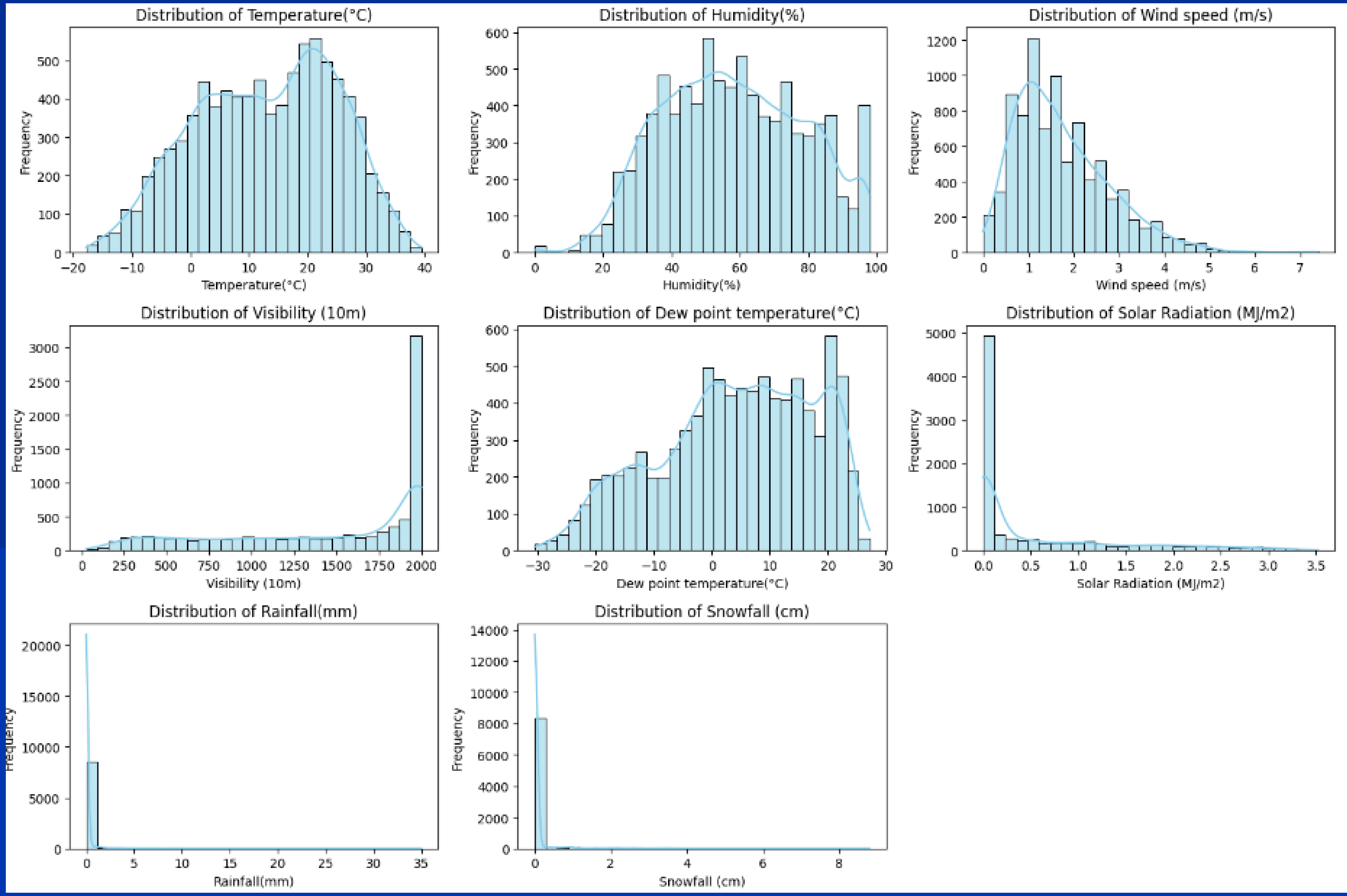
04

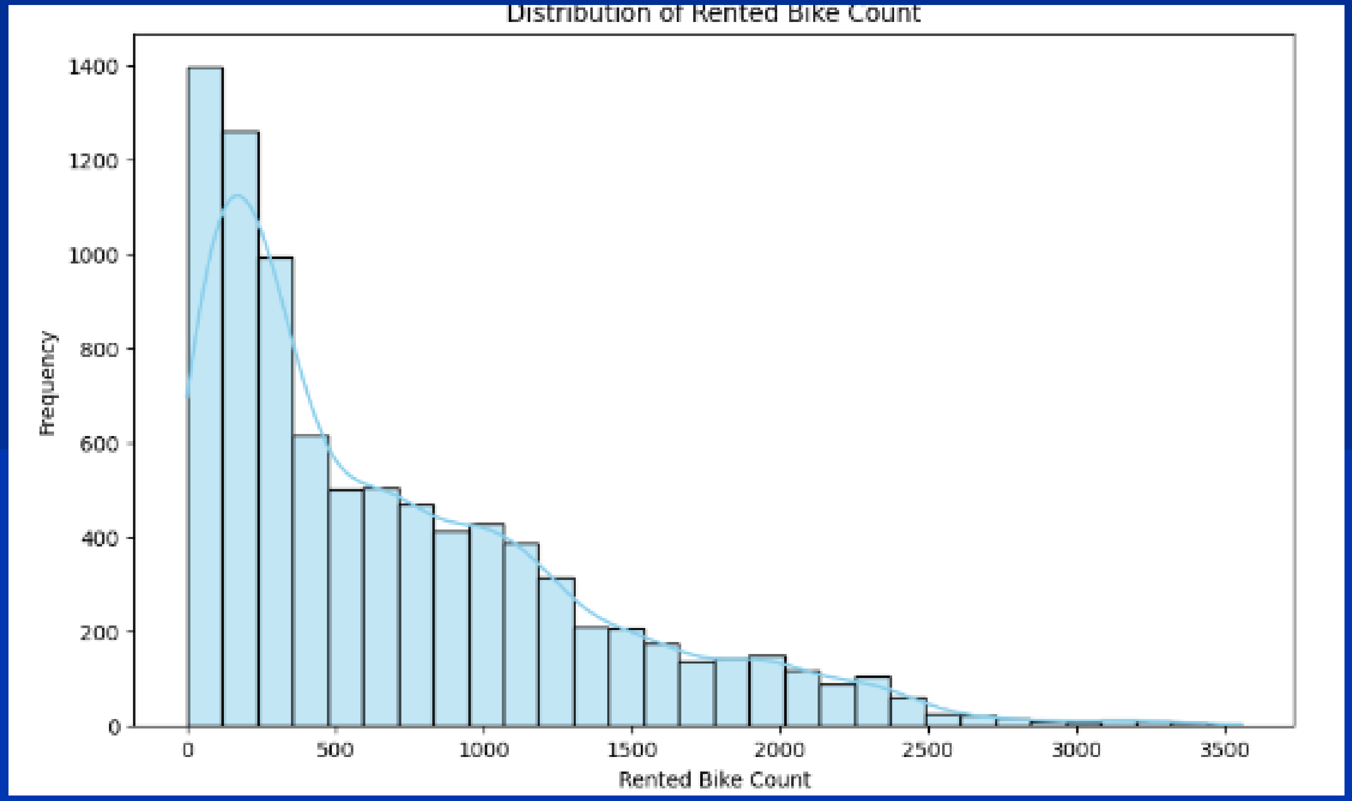
Outlier Detection: Identifying outliers that may skew the analysis. Those were mainly Dew Point and Snow. Also, the dependent variable, which had to be normalized.



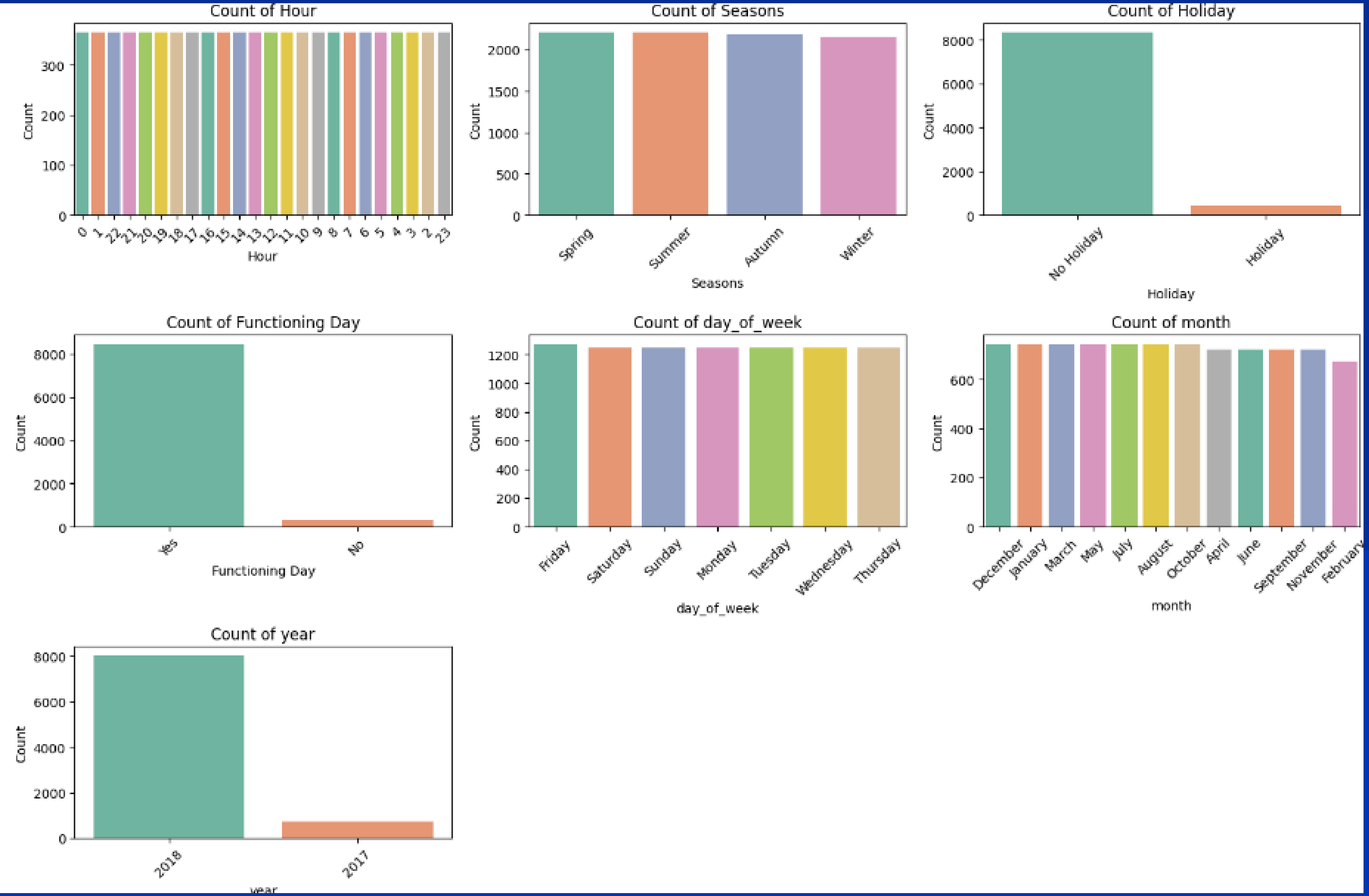
DATA EXPLORATION

Data Distribution:
Temporal Analysis
Influencing Factors

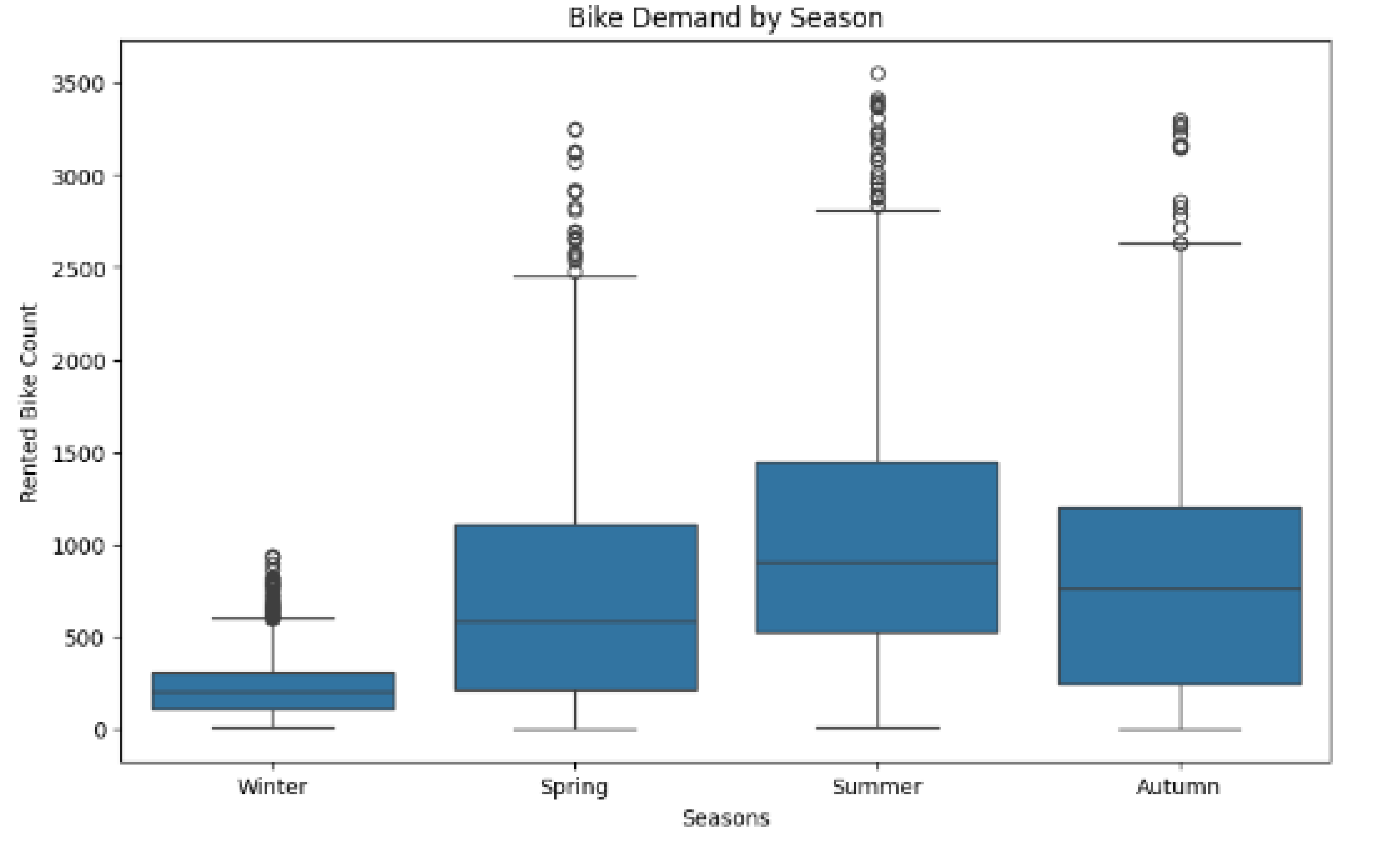




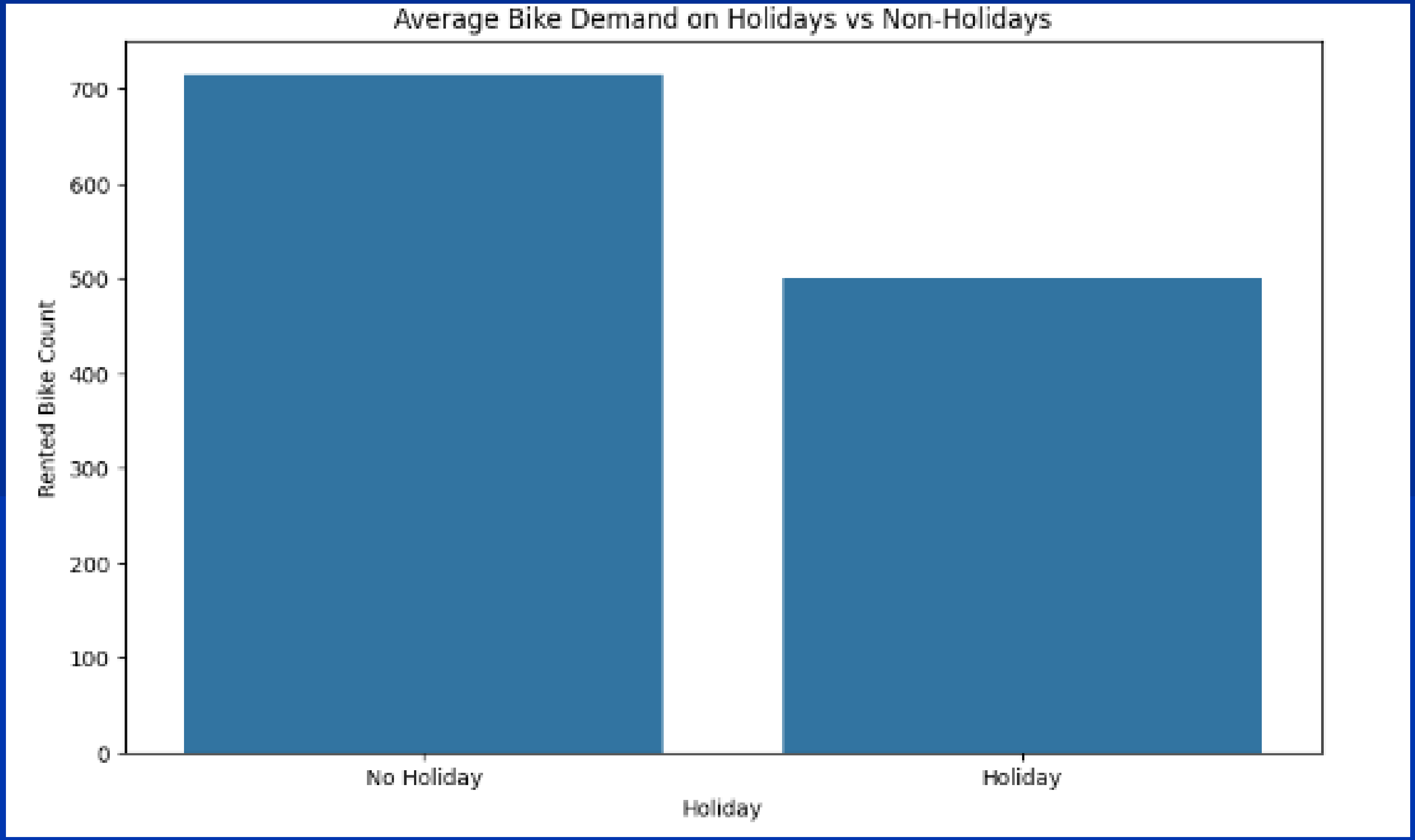
VARIABLE DISTRIBUTION



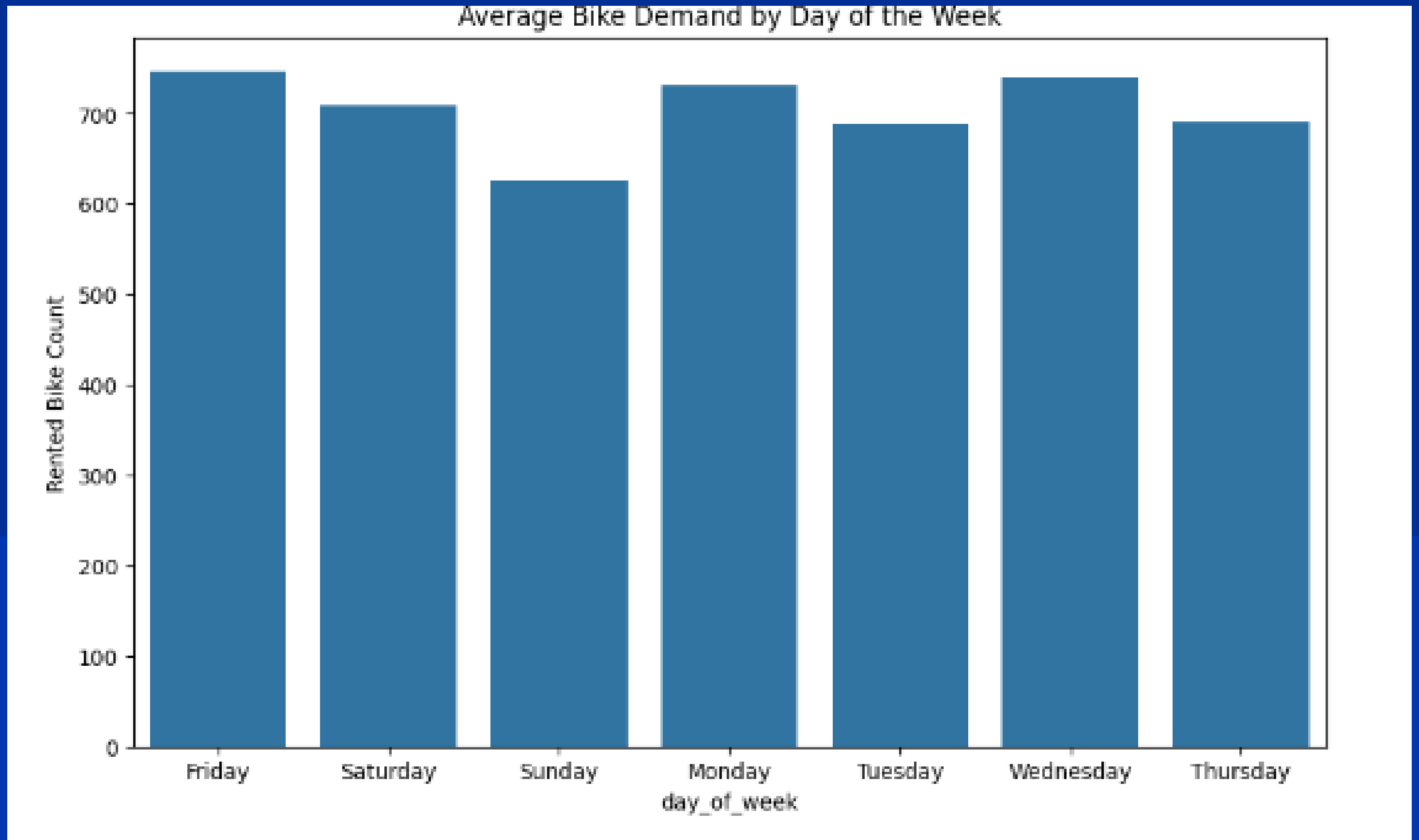
VARIABLE COUNT

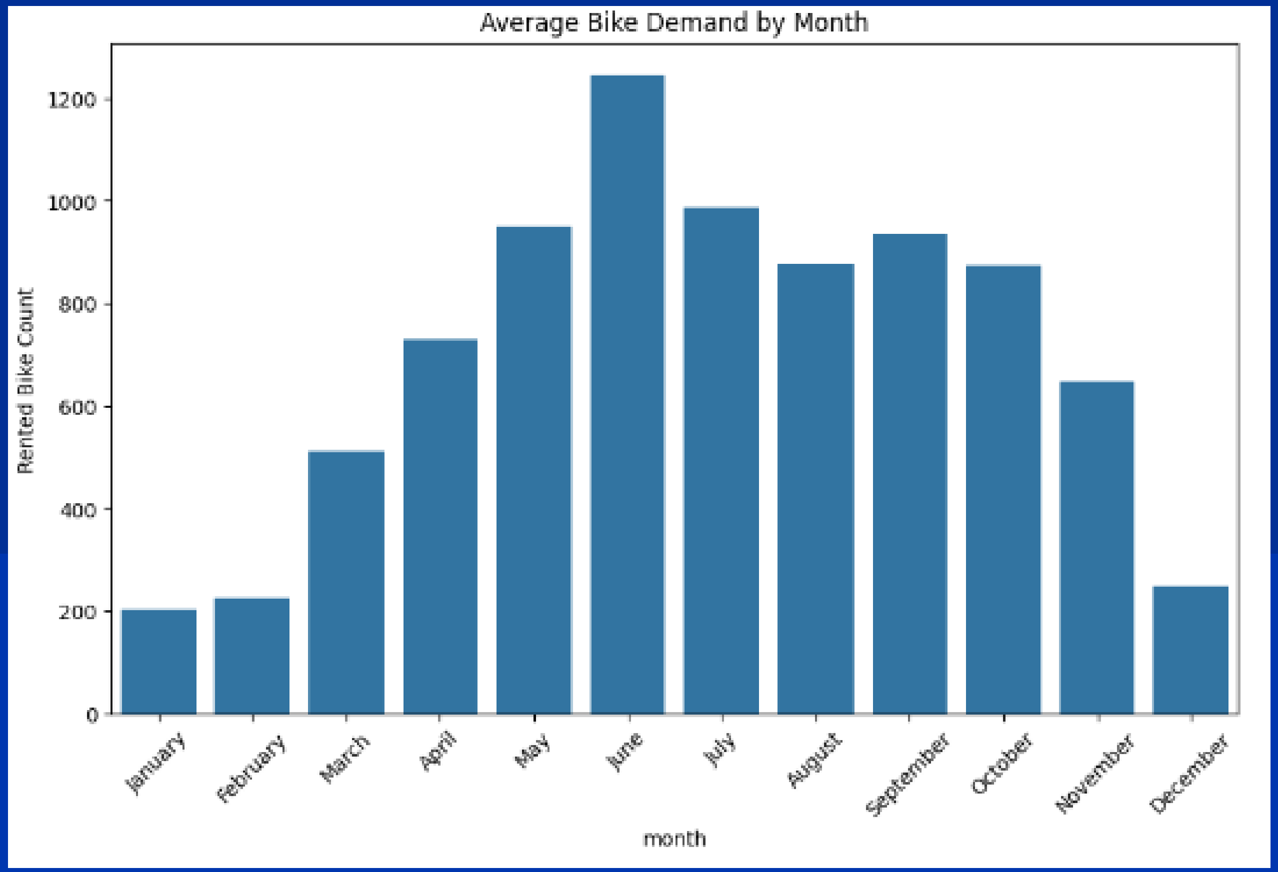


SEASONAL DEMAND

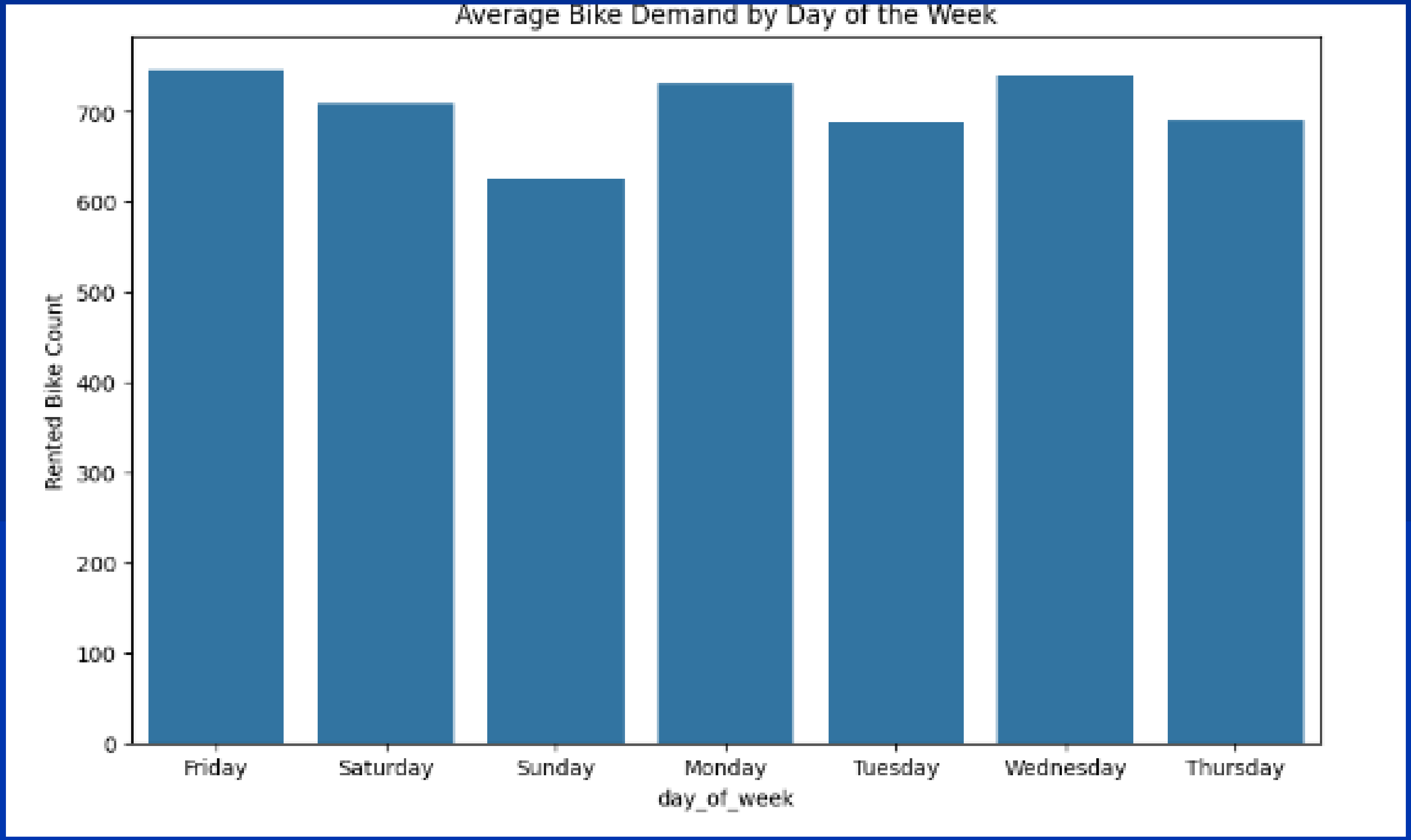


HOLIDAY DEMAND

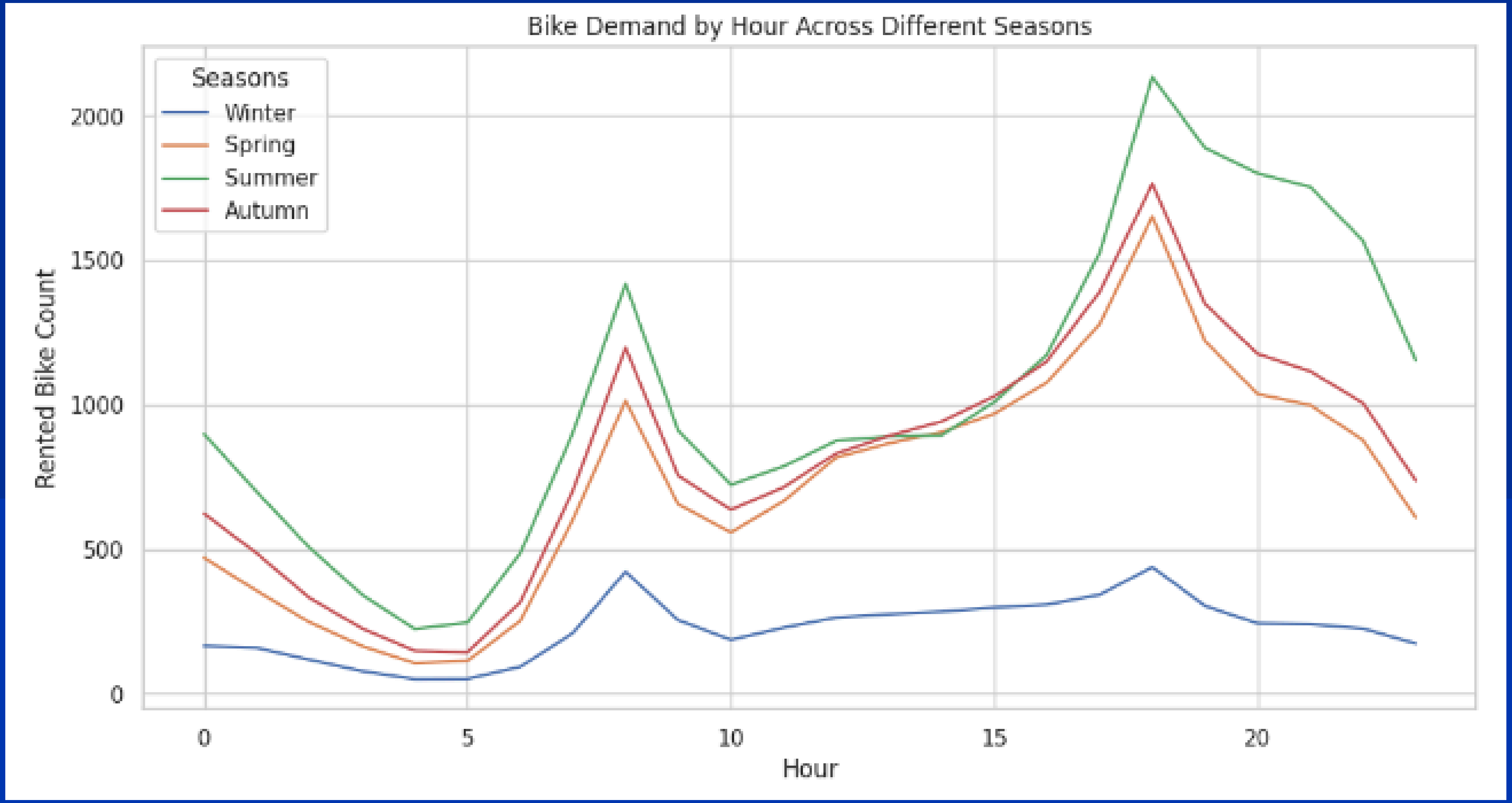




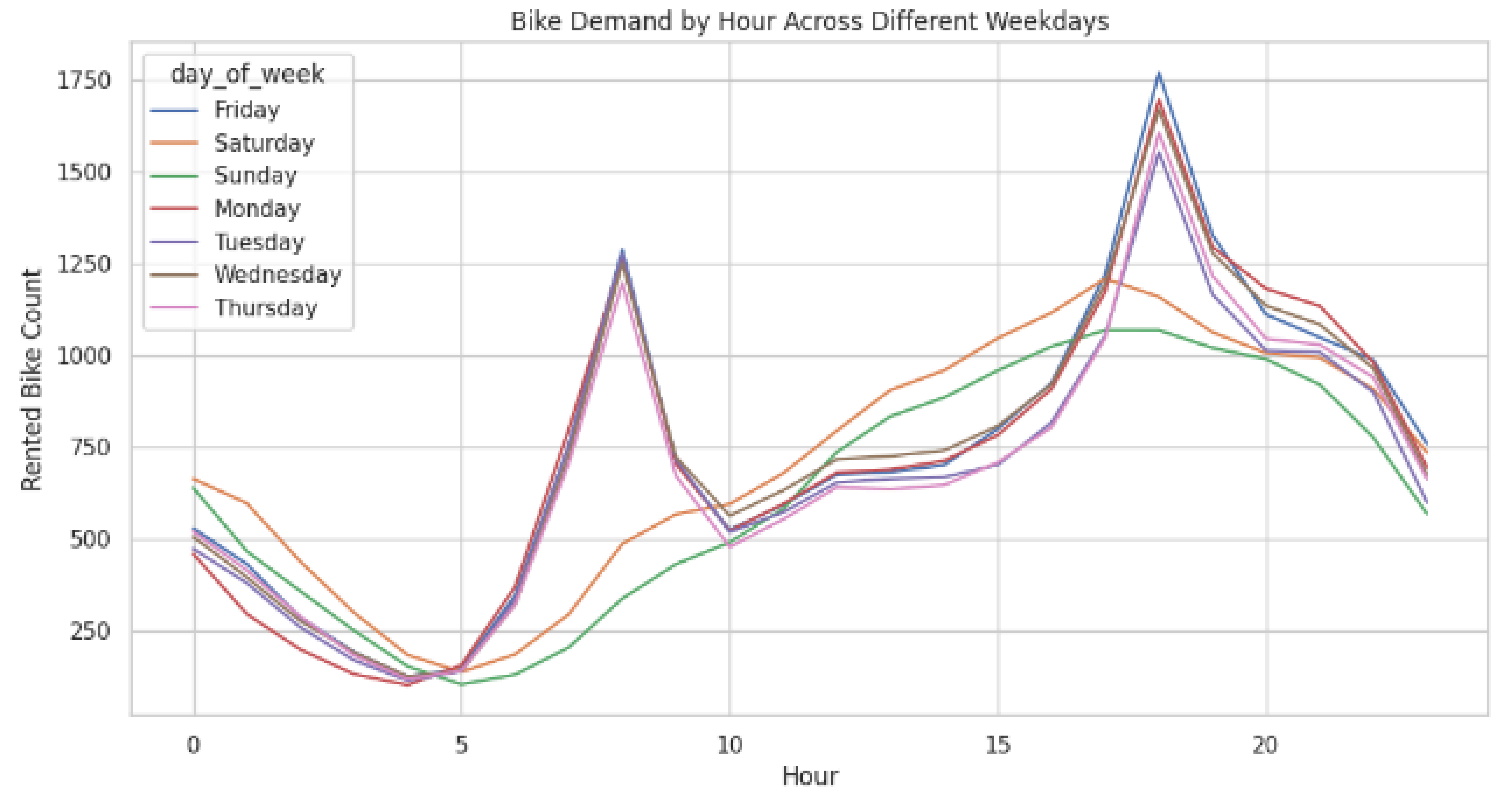
TEMPORAL DEMAND



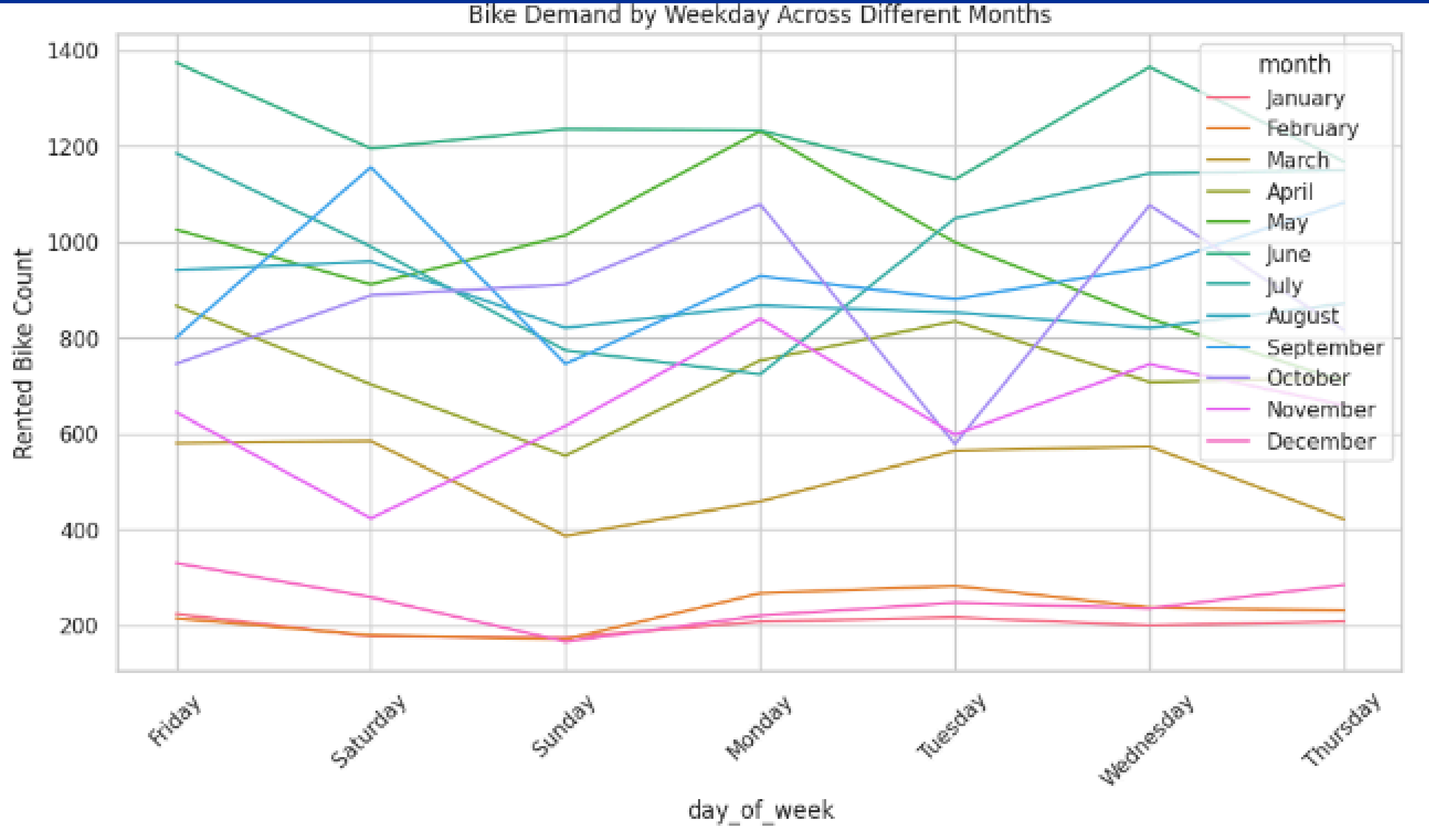
WEEKDAY DEMAND



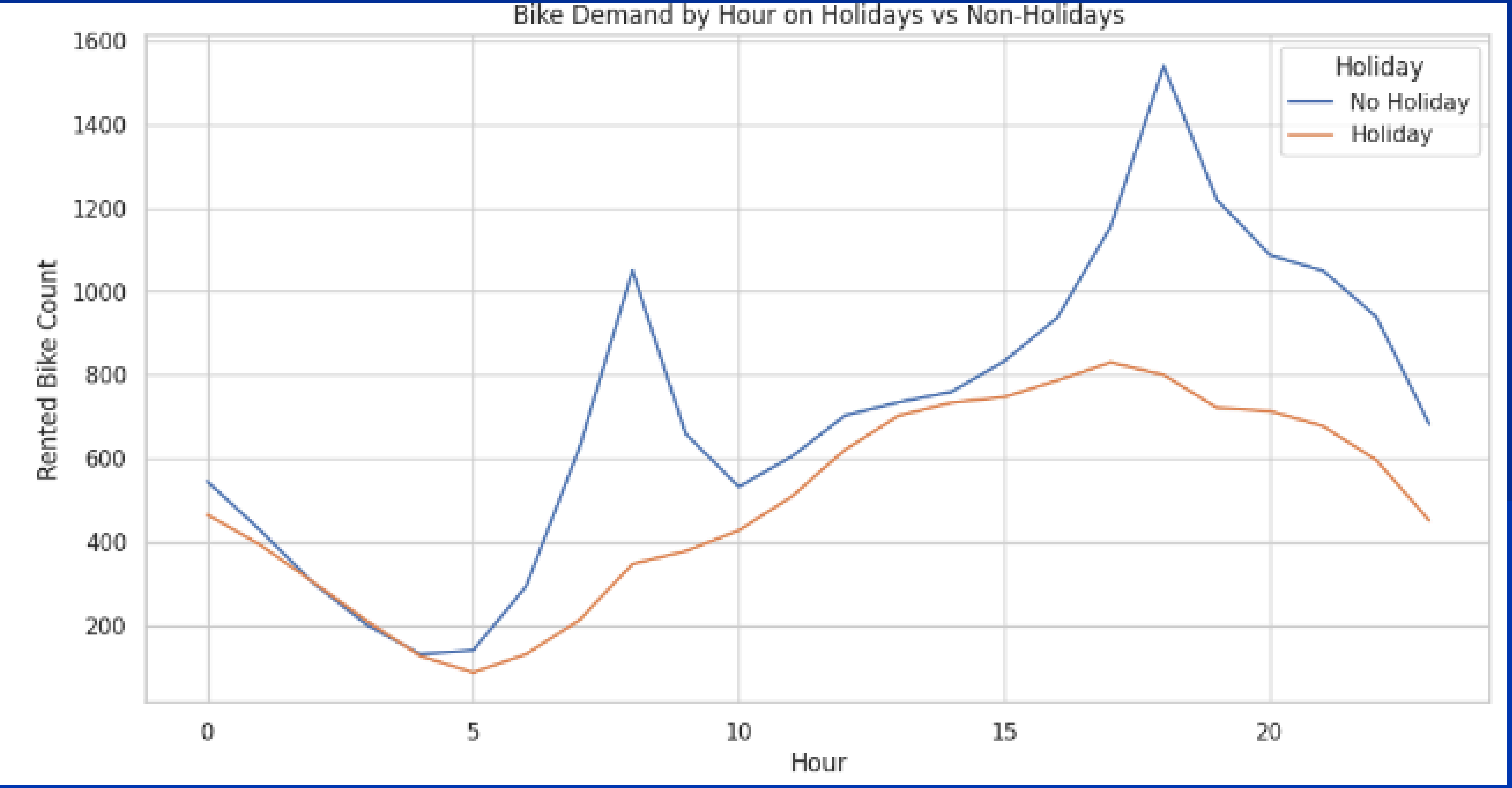
HOURLY DEMAND ACROSS SEASONS



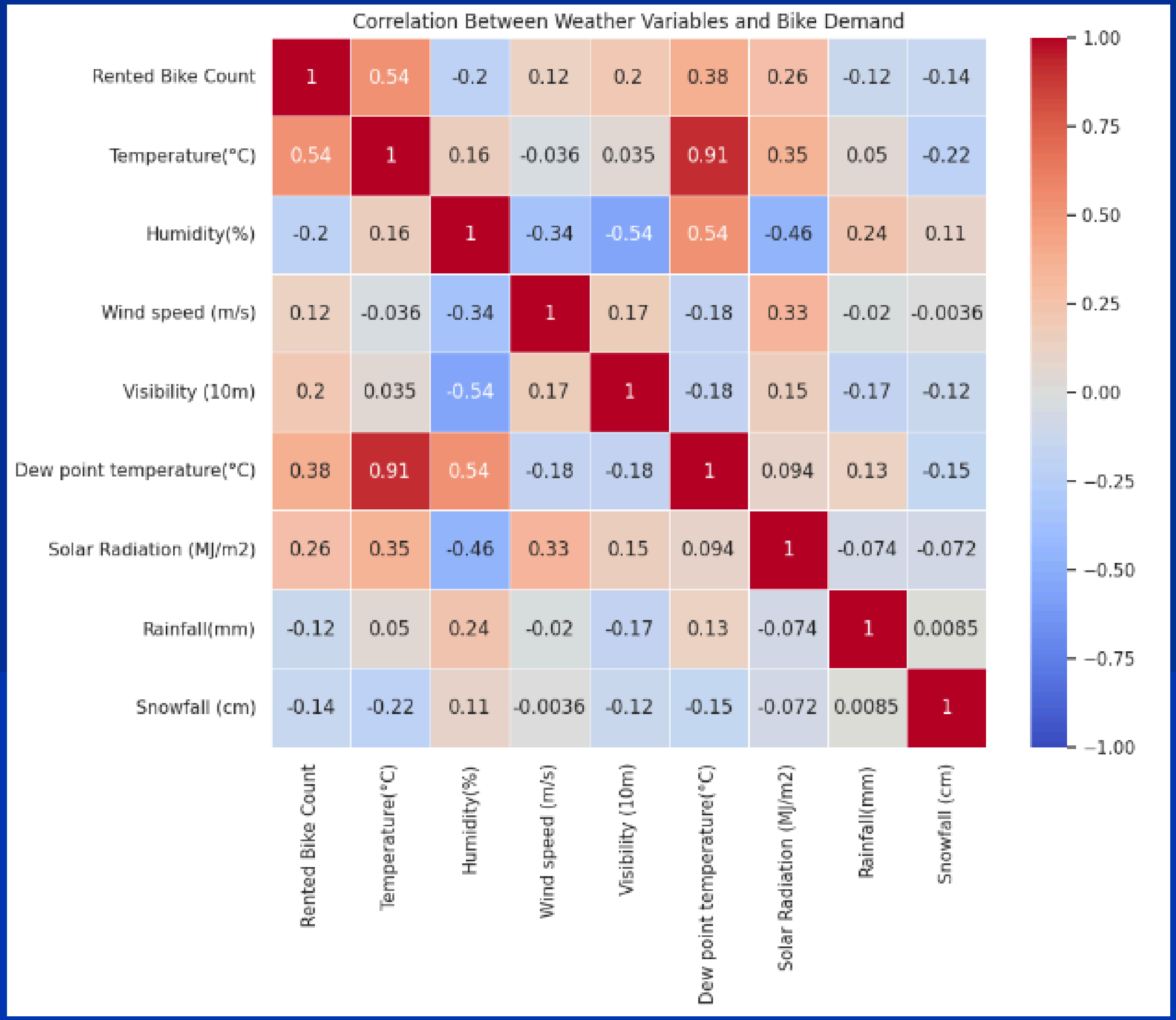
HOURLY DEMAND ACROSS WEEKDAYS



DAILY DEMAND ACROSS MONTHS



HOLIDAYS VS NON HOLIDAYS HOURLY



CORRELATION MATRIX

- Temperature and hour of the day show the strongest positive correlations, implying these factors significantly impact bike rentals.
- Rainfall, and snowfall have weak negative correlations.

VIF: ADDRESSING MULTICOLLINEARITY

Temperature x Dew Point Temperature

```
[50] #In our Correlation Matrix, we also see that Temperature and Dew Point Temperature are multicollinear. We should address this before creatig a model.

     from statsmodels.stats.outliers_influence import variance_inflation_factor
     def calc_vif(X):
         # Calculating VIF
         vif = pd.DataFrame()
         vif["variables"] = X.columns
         vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
         return(vif)
```

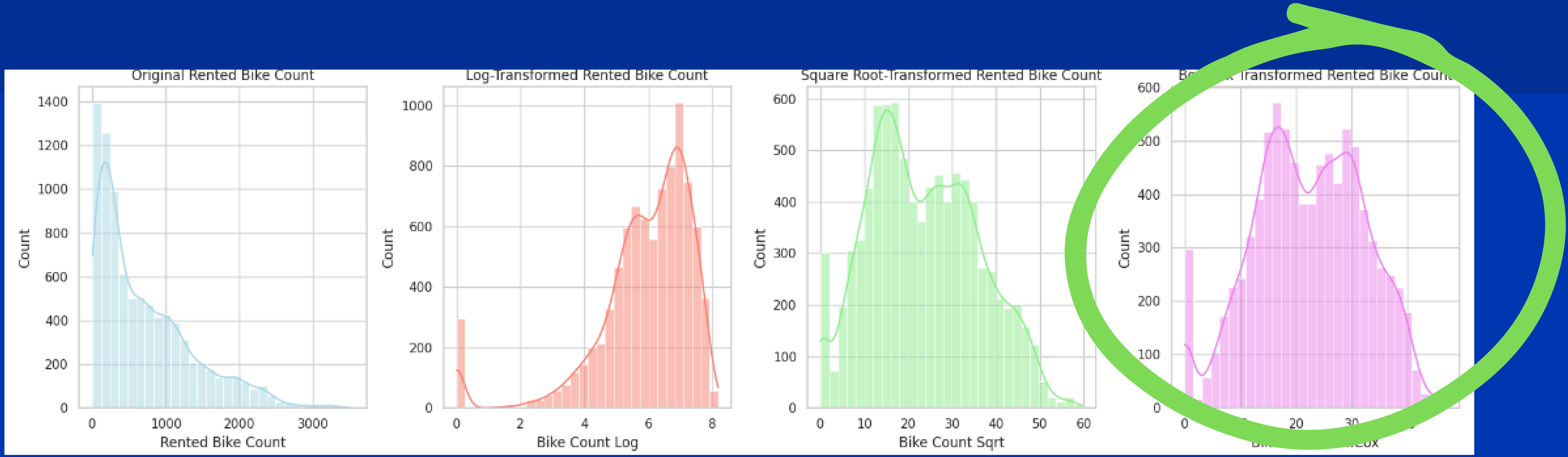
```
[51] numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns
     filtered_cols = [col for col in numeric_cols if col not in ['Rented Bike Count', 'Dew point temperature(°C)']]
```

```
[52] vif_result = calc_vif(df[filtered_cols])

     print("VIF Results:\n", vif_result)

     df.drop(columns=['Dew point temperature(°C)'], inplace=True)
```

```
VIF Results:
                  variables       VIF
0                     Hour  3.921832
1          Temperature(°C)  3.228318
2              Humidity(%)  4.868221
3         Wind speed (m/s)  4.608625
4         Visibility (10m)  4.710170
5  Solar Radiation (MJ/m2)  2.246791
6             Rainfall(mm)  1.079158
7            Snowfall (cm)  1.120579
```

NORMALISATION: BIKE RIDE COUNT

dependent variable - left-skewed



DATA MODELLING STEPS

01 **Selecting Features:** Identifying the most relevant features for predicting bike rentals.

02 **Splitting the Data:** Dividing the dataset into training and testing sets.

03 **Train Models:** Implement various machine learning algorithms to predict bike rentals.

04 **Evaluate Performance:** Assess the performance of each model using appropriate metrics.

FEATURE SELECTION

- Weather variables: Temperature, Humidity, Rainfall, Humidity, Solar Radiation, Visibility
- Temporal variables: Hour, Day of Week, Month, Season
- Special events: Holidays

METHODOLOGY: CORRELATION ANALYSIS TO IDENTIFY RELEVANT FEATURES.

MODELLING APPROACHES

01 Linear Regression

02 XGBoost Regressor

MODEL
TRAINING

80 %

TRAINING
SET

20 %

TEST
SET

SPLITTING

TRAINING

- Fit models using training data
- Hyperparameter definition

PERFORMANCE EVALUATION

LINEAR REGRESSION

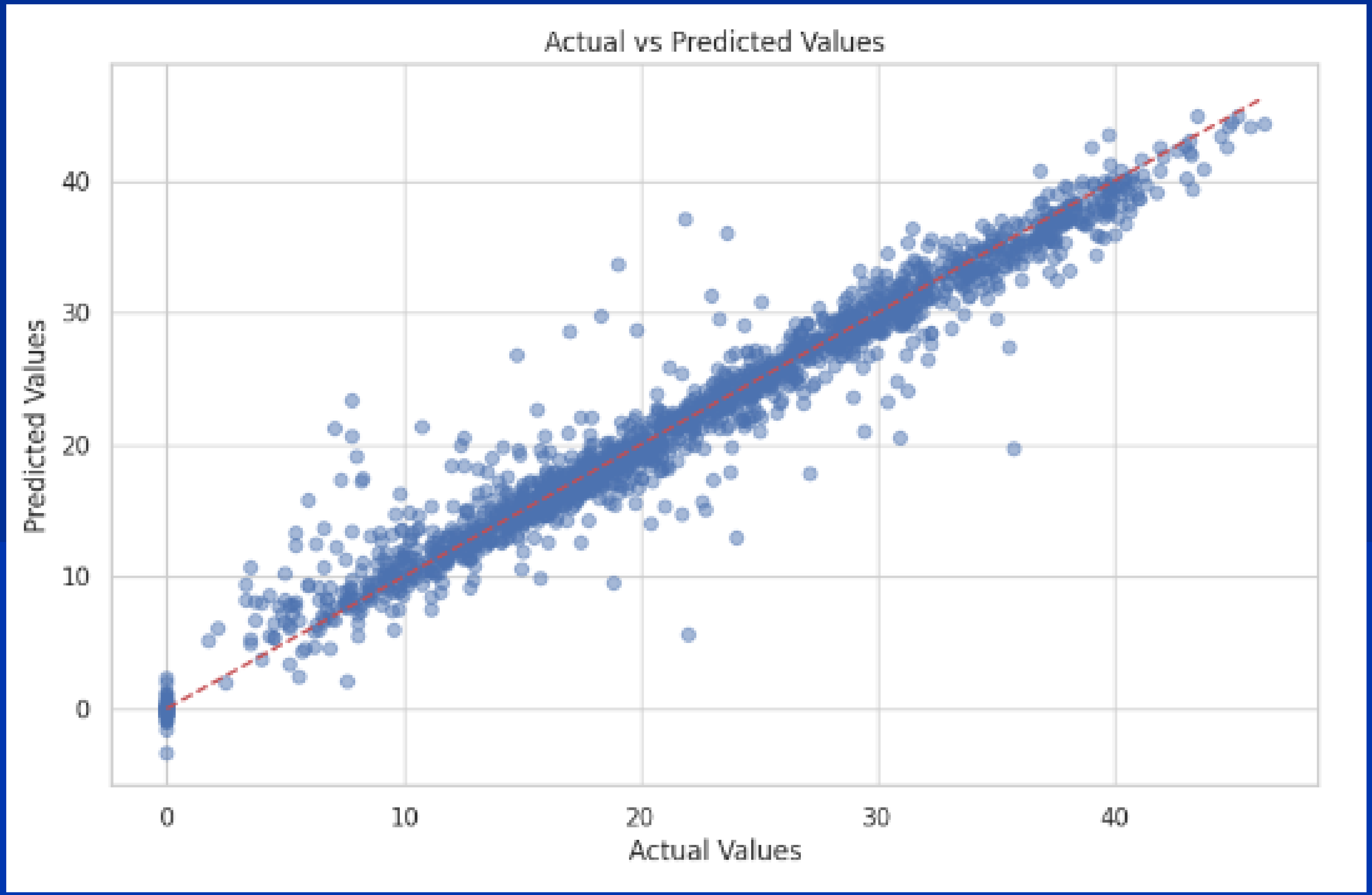
Mean Squared Error:
30.017399163802487

R^2 Score: 0.6927259377484193

XGBOOST

Mean Squared Error:
4.766630085882673

R^2 Score: 0.9537684553043096



PERFORMANCE

RESULTS

XGBOOST REGRESSION MODEL

-BECAUSE:
 The relationships in the dataset are complex.
- We needed a model that can handle large datasets and many features effectively.
- To improve prediction accuracy and robustness.

CONCLUSION

- XGBoost is the preferred model for bike rental demand prediction in this instance.
- Importance of feature selection and preprocessing in model performance.

THANK YOU