BUSINESS CASE : YULU- HYPOTHESIS TESTING

PROBLEM STATEMENT: The problem statement provided revolves around Yulu, India's leading micro-mobility service provider, which offers shared electric cycles for daily commutes. Yulu has recently experienced a decline in revenues and seeks to understand the factors influencing the demand for their shared electric cycles in the Indian market. Specifically, they aim to identify significant variables affecting demand and assess how well these variables describe the electric cycle demand.

Understanding the factors affecting bike rental demand is crucial for Yulu's business strategy for several reasons:

1. **Revenue Optimization:** By identifying the key drivers of bike rental demand, Yulu can optimize their revenue generation strategies. This includes pricing adjustments, targeted marketing campaigns, and resource allocation to high-demand areas.
2. **Customer Experience Enhancement:** Understanding demand factors allows Yulu to tailor its services to meet customer needs effectively. This may involve improving fleet availability, enhancing user experience through the mobile app, and providing additional services based on demand patterns.
3. **Operational Efficiency:** By accurately predicting demand, Yulu can optimize its operational processes, such as bike distribution, maintenance scheduling, and staffing, leading to cost savings and improved service reliability.
4. **Sustainability Impact:** Yulu's mission to eliminate traffic congestion and promote sustainable commuting relies on effectively meeting the demand for shared electric cycles. Understanding demand factors enables Yulu to expand its services strategically, contributing to reducing carbon emissions and promoting eco-friendly transportation alternatives.

In summary, gaining insights into the factors driving bike rental demand empowers Yulu to make informed decisions that not only enhance its financial performance but also contribute to its mission of providing safe, affordable, and sustainable commuting solutions in India.

ANALYZING BASIC METRICS: Analyzing basic metrics involves examining fundamental characteristics and summary statistics of the dataset to gain an initial understanding of its structure and content. This step is crucial in exploratory data analysis (EDA) as it provides insights into the data's distribution, variability, and potential issues that may need to be addressed before proceeding with further analysis. Here's how we can define and conduct basic metrics analysis:

Analyzing basic metrics involves:

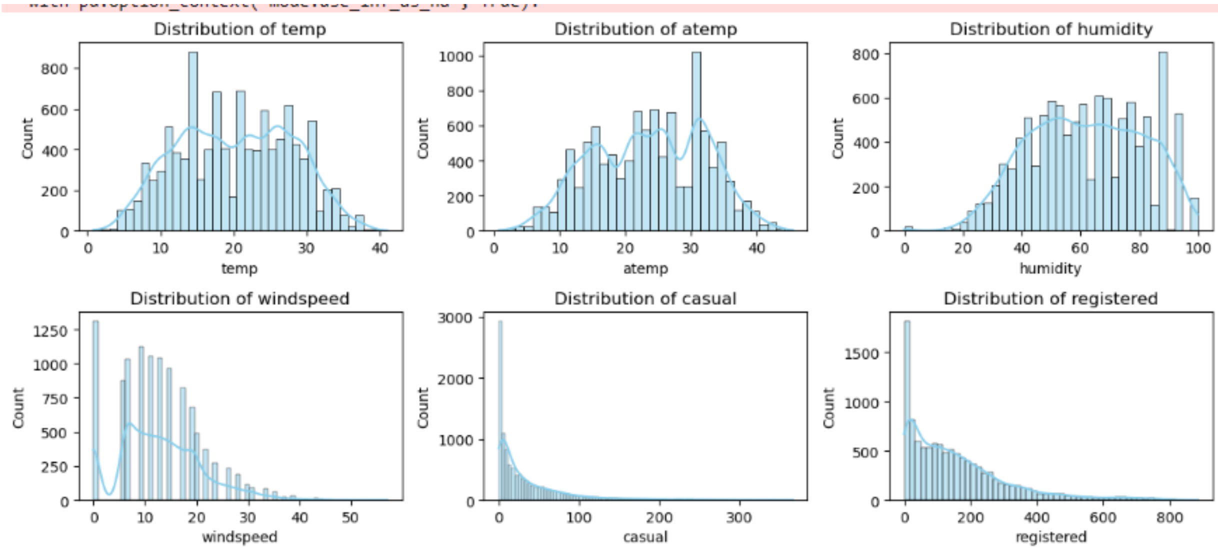
1. **Descriptive Statistics:** Calculating summary statistics such as mean, median, mode, standard deviation, minimum, maximum, and quartiles for numerical variables.

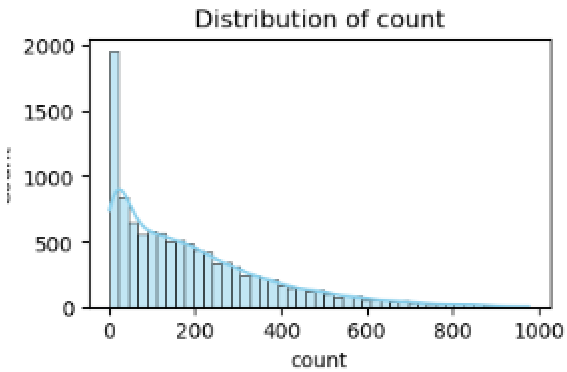
```
# 1. Descriptive Statistics
numerical_variables = ['temp', 'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count']
print(data[numerical_variables].describe())
```

```
             temp         atemp      humidity     windspeed        casual  \
count  10886.00000  10886.000000  10886.000000  10886.000000  10886.000000
mean      20.23086     23.655084     61.886460     12.799395     36.021955
std        7.79159      8.474601     19.245033      8.164537     49.960477
min        0.82000      0.760000      0.000000      0.000000      0.000000
25%       13.94000     16.665000     47.000000      7.001500      4.000000
50%       20.50000     24.240000     62.000000     12.998000     17.000000
75%       26.24000     31.060000     77.000000     16.997900     49.000000
max       41.00000     45.455000    100.000000     56.996900    367.000000

         registered         count
count  10886.000000  10886.000000
mean     155.552177    191.574132
std      151.039033    181.144454
min        0.000000      1.000000
25%       36.000000     42.000000
50%      118.000000    145.000000
75%      222.000000    284.000000
max      886.000000    977.000000
```

2. **Data Distribution:** Examining the distribution of numerical variables through histograms, kernel density estimation (KDE) plots, or box plots.

```
# 2. Data Distribution
plt.figure(figsize=(12, 8))
for i, col in enumerate(numerical_variables):
    plt.subplot(3, 3, i+1)
    sns.histplot(data[col], kde=True, color='skyblue')
    plt.title(f'Distribution of {col}')
plt.tight_layout()
plt.show()
```





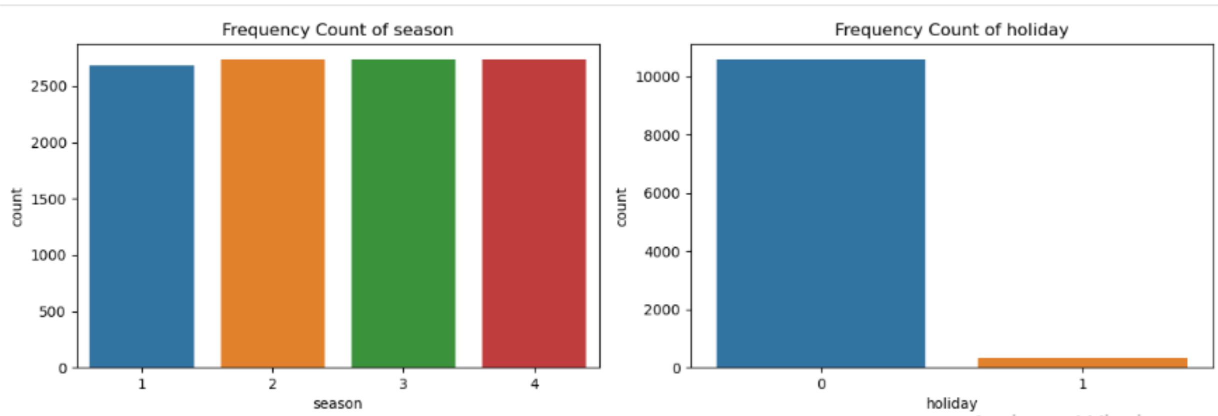
Insights: Upon analyzing the data distribution, we found that temperature (temp) and feeling temperature (atemp) exhibit distributions that closely resemble a normal distribution, suggesting a relatively balanced spread of data around the mean.

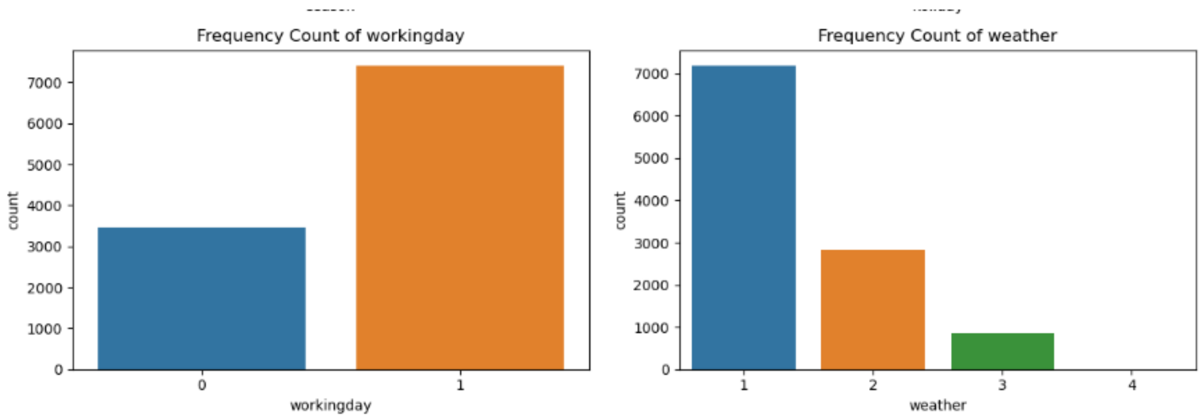
In contrast, humidity's distribution is slightly left-skewed, indicating a tendency towards higher humidity levels in the dataset.

Furthermore, windspeed, casual, registered, and total rental counts (count) all display left-skewed distributions. This skewness implies that lower values are more prevalent in these features, with a few instances having notably higher counts.

3. **Frequency Counts:** Counting the frequency of categorical variables and visualizing them through bar plots or count plots.

```
# 3. Frequency Counts
categorical_variables = ['season', 'holiday', 'workingday', 'weather']
plt.figure(figsize=(12, 8))
for i, col in enumerate(categorical_variables):
    plt.subplot(2, 2, i+1)
    sns.countplot(data=data, x=col)
    plt.title(f'Frequency Count of {col}')
plt.tight_layout()
plt.show()
```





Insights: Based on the visual analysis, it appears that the season variable does not significantly influence the rental counts, as the distribution of counts across different seasons appears relatively consistent.

There is a minimal presence of holidays compared to non-holidays in the dataset.

Regarding working days, it's notable that the majority of observations occur on workdays, with approximately 7,000 instances recorded.

Weather condition 1 appears to be the most favorable, as it is associated with the highest number of users, with approximately 7,000 observations.

OBSERVATIONS OF DATASET: In this step, we will begin by importing the dataset and conducting a series of operations to gain fundamental insights into its structure and contents. Through these operations, we aim to extract basic details about the dataset, such as its size, the types of data it contains, and summary statistics that provide a snapshot of the dataset's characteristics.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#IMPORTING DATASET
data=pd.read_csv('G:/dsml-scaler/probability and stats/casestudy/bike_sharing.txt')
```

```
data.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	0	1	1

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   datetime    10886 non-null  object 
 1   season      10886 non-null  int64  
 2   holiday     10886 non-null  int64  
 3   workingday  10886 non-null  int64  
 4   weather     10886 non-null  int64  
 5   temp        10886 non-null  float64
 6   atemp       10886 non-null  float64
 7   humidity    10886 non-null  int64  
 8   windspeed   10886 non-null  float64
 9   casual      10886 non-null  int64  
 10  registered  10886 non-null  int64  
 11  count       10886 non-null  int64  
dtypes: float64(3), int64(8), object(1)
memory usage: 1020.7+ KB
```

```python
print("The shape of dataset is:",data.shape)
```

```
The shape of dataset is: (10886, 12)
```

The dataset consists of 10886 rows and 12 columns, encapsulating a comprehensive range of data points for analysis.

```python
print("The datatype of attributes are:")
print(data.dtypes)
```

```
The datatype of attributes are:
datetime       object
season          int64
holiday         int64
workingday      int64
weather         int64
temp          float64
atemp         float64
humidity        int64
windspeed     float64
casual          int64
registered      int64
count           int64
dtype: object
```

The dataset comprises numerical values across most columns, with 'datetime' in object format.

```python
# Convert 'datetime' column to datetime format
data['datetime'] = pd.to_datetime(data['datetime'])

# Verify the data type conversion
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   datetime    10886 non-null  datetime64[ns]
 1   season      10886 non-null  int64
 2   holiday     10886 non-null  int64
 3   workingday  10886 non-null  int64
 4   weather     10886 non-null  int64
 5   temp        10886 non-null  float64
 6   atemp       10886 non-null  float64
 7   humidity    10886 non-null  int64
 8   windspeed   10886 non-null  float64
 9   casual      10886 non-null  int64
 10  registered  10886 non-null  int64
 11  count       10886 non-null  int64
dtypes: datetime64[ns](1), float64(3), int64(8)
memory usage: 1020.7 KB
None
```

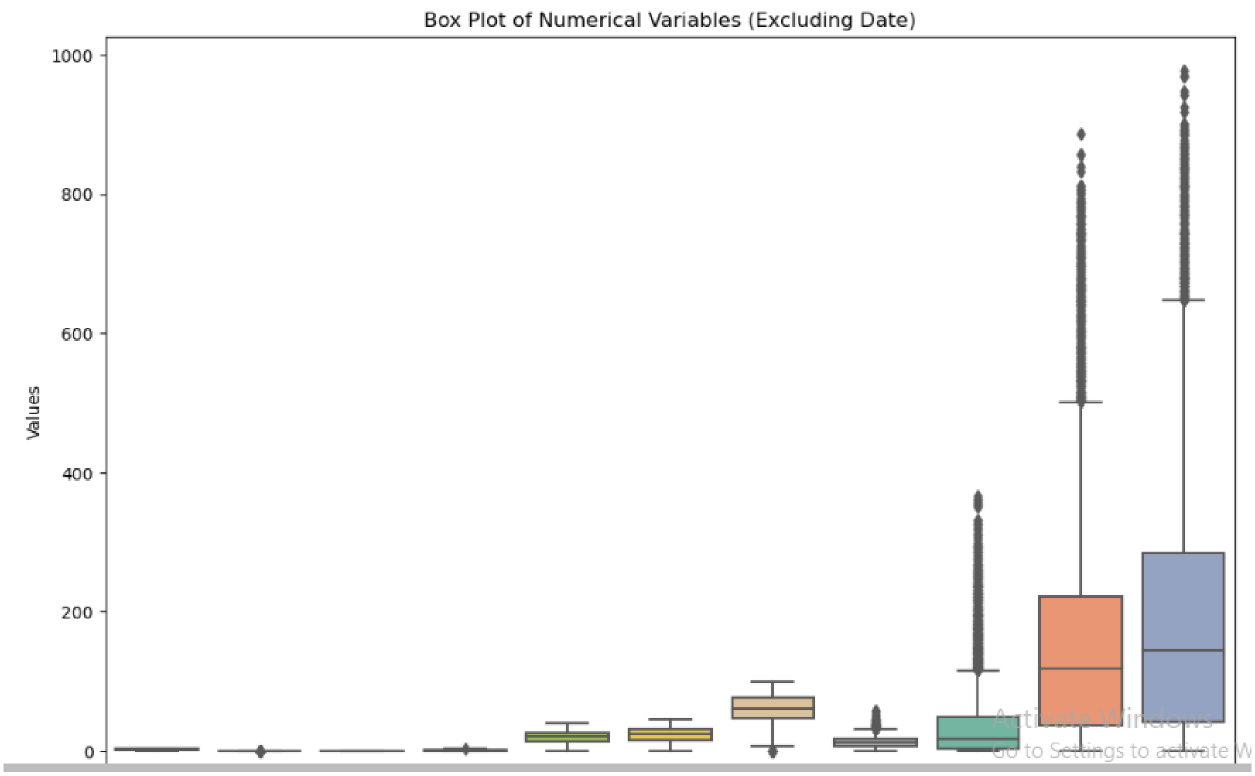
```python
# checking the null values
data.isna().sum()
```

```
datetime      0
season        0
holiday       0
workingday    0
weather       0
temp          0
atemp         0
humidity      0
windspeed     0
casual        0
registered    0
count         0
dtype: int64
```

There are no null values. The data is clean.

Handling Outliers: Since there are no missing values in the dataset, our attention will now shift towards detecting and managing outliers, if any. This process will involve scrutinizing the data for any unusual or extreme observations that may impact the robustness of our analysis.

```
# checking for outliers
# Select numerical variables excluding the 'datetime' column
numerical_variables = ['season', 'holiday', 'workingday', 'weather', 'temp', 'atemp', 'humidity', 'windspeed', 'casual',

# Plot box plots for selected numerical variables
plt.figure(figsize=(12, 8))
sns.boxplot(data=data[numerical_variables], orient='v', palette='Set2')
plt.title('Box Plot of Numerical Variables (Excluding Date)')
plt.ylabel('Values')
plt.xticks(rotation=45)
plt.show()
```



```python
q1= data.windspeed.quantile(0.25)
q3 = data.windspeed.quantile(0.75)
iqr = q3-q1
outliers = data[((data.windspeed < (q1-1.5*iqr)) | ((data.windspeed > (q3+1.5*iqr))))]
print("number of outliers",len(outliers))
print("percentage of outliers", len(outliers)/data.shape[0])
```

```
number of outliers 227
percentage of outliers 0.02085247106375161
```

```python
q1= data.casual.quantile(0.25)
q3 = data.casual.quantile(0.75)
iqr = q3-q1
outliers = data[((data.casual < (q1-1.5*iqr)) | ((data.casual > (q3+1.5*iqr))))]
print("number of outliers",len(outliers))
print("percentage of outliers", len(outliers)/data.shape[0])
```

```
number of outliers 749
percentage of outliers 0.06880396839977954
```

```python
q1= data.registered.quantile(0.25)
q3 = data.registered.quantile(0.75)
iqr = q3-q1
outliers = data[((data.registered < (q1-1.5*iqr)) | ((data.registered > (q3+1.5*iqr))))]
print("number of outliers",len(outliers))
print("percentage of outliers", len(outliers)/data.shape[0])
```

```
number of outliers 423
percentage of outliers 0.03885724784126401
```

```python
# Define the column name
column_name = 'count'

# Calculate quartiles and IQR
q1 = data[column_name].quantile(0.25)
q3 = data[column_name].quantile(0.75)
iqr = q3 - q1

# Determine outliers
outliers = data[(data[column_name] < (q1 - 1.5 * iqr)) | (data[column_name] > (q3 + 1.5 * iqr))]

# Calculate the number of outliers
num_outliers = len(outliers)

# Calculate the percentage of outliers
percentage_outliers = (num_outliers / data.shape[0]) * 100

# Print the results
print("Number of outliers:", num_outliers)
print("Percentage of outliers:", percentage_outliers)
```

```
Number of outliers: 300
Percentage of outliers: 2.75583318023149
```

```python
# Handling outliers using imputation method
# Define the columns with outliers
columns_with_outliers = ['windspeed', 'casual', 'registered', 'count']

# Define a function to impute outliers
def impute_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Replace outliers with median value
    df[column] = np.where((df[column] < lower_bound) | (df[column] > upper_bound),
                          df[column].median(), df[column])
    return df

# Impute outliers for each column
for column in columns_with_outliers:
    df = impute_outliers(data, column)

# Verify that outliers have been imputed
print("Number of outliers after imputation:")
for column in columns_with_outliers:
    num_outliers = ((df[column] < df[column].quantile(0.25) - 1.5 * (df[column].quantile(0.75) - df[column].quantile(0.25
                    (df[column] > df[column].quantile(0.75) + 1.5 * (df[column].quantile(0.75) - df[column].quantile(0.25
    print(f"{column}: {num_outliers}")
```

```
Number of outliers after imputation:
windspeed: 0
casual: 557
registered: 161
count: 114
```

```python
# Define the columns with remaining outliers
columns_with_remaining_outliers = ['casual', 'registered', 'count']

# Remove remaining outliers for each column
for column in columns_with_remaining_outliers:
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Filter out rows with outliers
    df = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

# Print the shape of the dataset after outlier removal
print("Shape of the dataset after outlier removal:", df.shape)
```

```
Shape of the dataset after outlier removal: (9866, 12)
```

Outlier Removal Process:

In our analysis of the dataset provided by Yulu, we encountered outliers in several key variables, including 'windspeed', 'casual', 'registered', and 'count'. Outliers are data points that significantly deviate from the rest of the dataset and may skew our analysis results if left unaddressed. Therefore, it was essential to implement an outlier removal process to ensure the accuracy and reliability of our analysis.

We adopted a two-step approach to address outliers:

Step 1: Imputation of Outliers: We initially employed the Interquartile Range (IQR) method, a widely used technique for outlier detection and removal. This method calculates the lower and upper bounds based on the first quartile (Q1) and third quartile (Q3) of the data distribution. Any data points falling outside these bounds are considered outliers and were subsequently replaced with the median value of the respective variable.

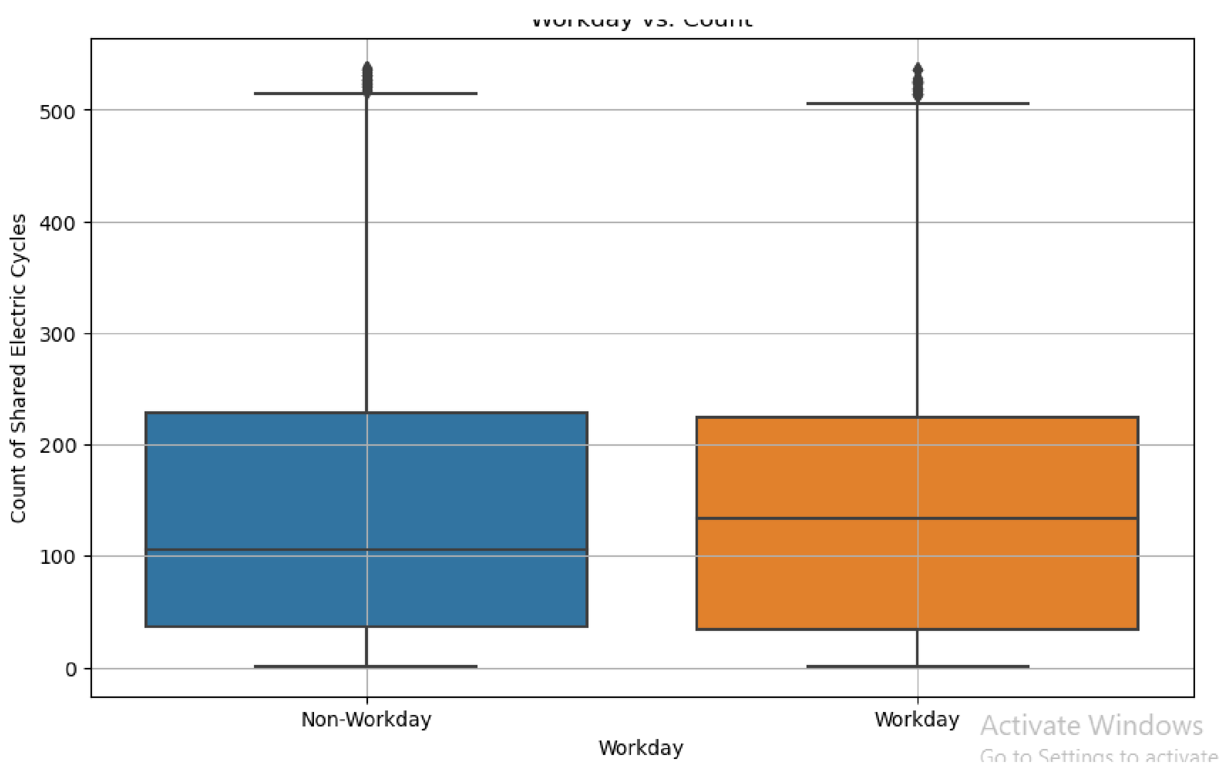
Step 2: Further Outlier Removal: Following the imputation process, we observed that a subset of outliers persisted in the 'casual', 'registered', and 'count' columns. To address these remaining outliers, we opted for their removal from the dataset. This decision was made after careful consideration of the impact of the outliers on the analysis results and the desire to ensure the robustness of our findings.

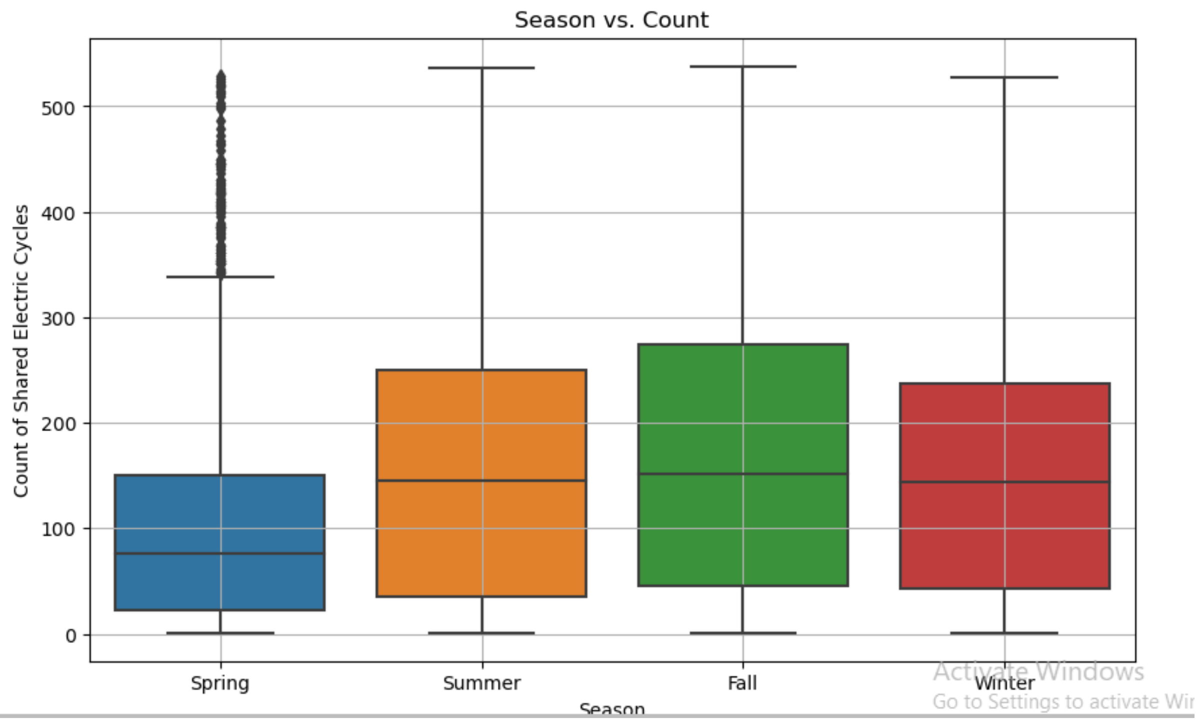
Result: As a result of the outlier removal process, we obtained a cleaned dataset that is more representative of the underlying data distribution and less susceptible to the influence of extreme values. The size of the dataset was reduced from its original dimensions, but the removal of outliers was deemed necessary to mitigate potential biases and ensure the validity of our analysis.

Conclusion: Overall, the outlier removal process was instrumental in enhancing the quality and reliability of our analysis. By systematically identifying and addressing outliers, we were able to derive more accurate insights into the factors affecting the demand for shared electric cycles in the Indian market, thus empowering Yulu to make informed decisions and develop effective strategies to address revenue dips.

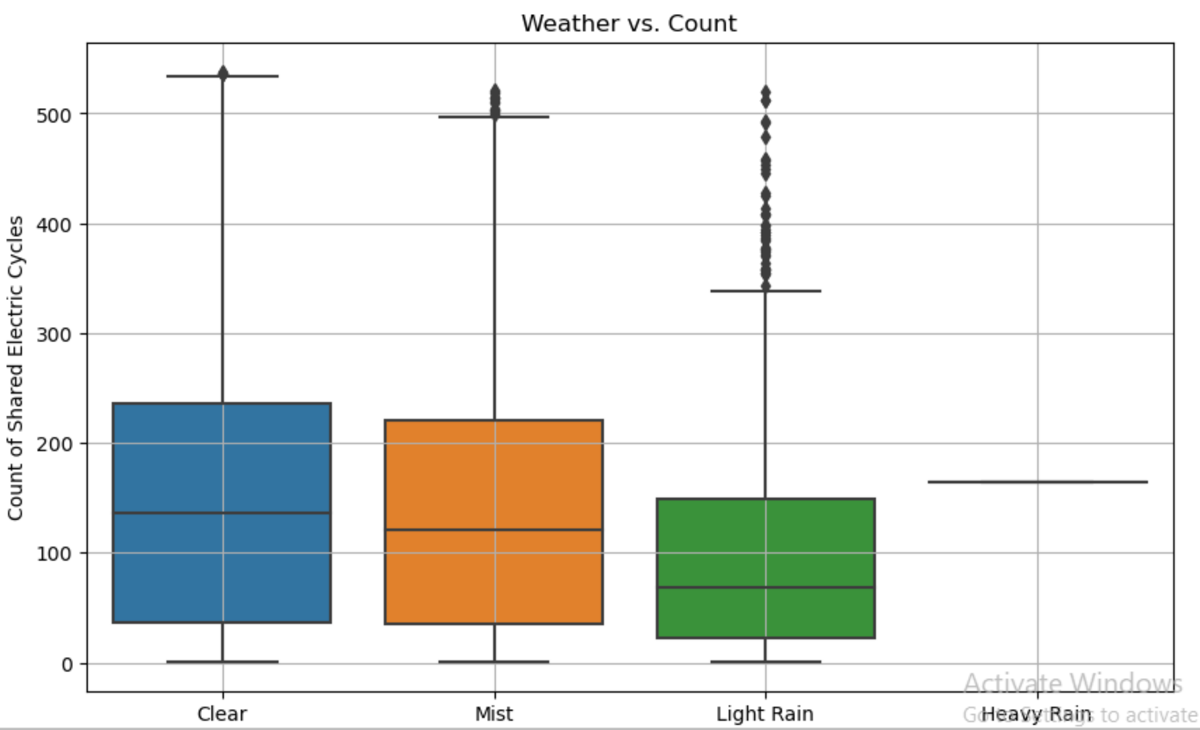
Bivariate Analysis: For each pair of variables, we'll examine how they relate to the 'Count' variable, which represents the demand for shared electric cycles. We'll use various visualizations such as scatter plots, bar plots, or box plots to illustrate the relationships between these variables and the 'Count' variable. Through this analysis, we aim to gain insights into the factors influencing the demand for shared electric cycles and their potential impact on Yulu's business.

```python
# Plotting Workday vs. Count
plt.figure(figsize=(10, 6))
sns.boxplot(x='workingday', y='count', data=df)
plt.title('Workday vs. Count')
plt.xlabel('Workday')
plt.ylabel('Count of Shared Electric Cycles')
plt.xticks([0, 1], ['Non-Workday', 'Workday'])
plt.grid(True)
plt.show()

# Plotting Season vs. Count
plt.figure(figsize=(10, 6))
sns.boxplot(x='season', y='count', data=df)
plt.title('Season vs. Count')
plt.xlabel('Season')
plt.ylabel('Count of Shared Electric Cycles')
plt.xticks([0, 1, 2, 3], ['Spring', 'Summer', 'Fall', 'Winter'])
plt.grid(True)
plt.show()

# Plotting Weather vs. Count
plt.figure(figsize=(10, 6))
sns.boxplot(x='weather', y='count', data=df)
plt.title('Weather vs. Count')
plt.xlabel('Weather')
plt.ylabel('Count of Shared Electric Cycles')
plt.xticks([0, 1, 2, 3], ['Clear', 'Mist', 'Light Rain', 'Heavy Rain'])
plt.grid(True)
plt.show()
```





Season vs. Count
Count of Shared Electric Cycles
0
100
200
300
400
500
Spring
Summer
Fall
Winter
Season
Activate Windows
Go to Settings to activate Wir



Weather vs. Count
Count of Shared Electric Cycles
0
100
200
300
400
500
Clear
Mist
Light Rain
Heavy Rain
Activate Windows

```python
# Grouping by workingday and calculating the count
count_by_workday = df.groupby('workingday')['count'].sum()
print("Count of shared electric cycles by workingday:\n", count_by_workday)

# Grouping by season and calculating the count
count_by_season = df.groupby('season')['count'].sum()
print("\nCount of shared electric cycles by season:\n", count_by_season)

# Grouping by weather and calculating the count
count_by_weather = df.groupby('weather')['count'].sum()
print("Count of shared electric cycles by weather:\n", count_by_weather)
```

```
Count of shared electric cycles by workingday:
 workingday
0    472387.0
1    995166.0
Name: count, dtype: float64

Count of shared electric cycles by season:
 season
1    269414.0
2    388265.0
3    410093.0
4    399781.0
Name: count, dtype: float64
Count of shared electric cycles by weather:
 weather
1    1002070.0
2     379216.0
3      86103.0
4        164.0
Name: count, dtype: float64
```

Insights: Based on the counts and visuals of shared electric cycles grouped by different variables, we can derive several insights:

Working Day vs. Count:

- On non-working days (workingday = 0), the total count of shared electric cycles is 472,387.
- On working days (workingday = 1), the total count significantly increases to 995,166.
- This suggests that there is a higher demand for shared electric cycles on working days compared to non-working days, possibly due to commuting to work or running errands.

Season vs. Count: The distribution of shared electric cycles across seasons is as follows:

- Spring (season = 1): 269,414 cycles
- Summer (season = 2): 388,265 cycles
- Fall (season = 3): 410,093 cycles
- Winter (season = 4): 399,781 cycles

The demand for shared electric cycles appears to be relatively consistent across different seasons, with a slight increase in fall.

Weather vs. Count: The distribution of shared electric cycles across weather conditions is as follows:

- Clear weather (weather = 1): 1,002,070 cycles
- Misty weather (weather = 2): 379,216 cycles
- Light rain (weather = 3): 86,103 cycles
- Heavy rain (weather = 4): 164 cycles

The majority of shared electric cycles are rented during clear weather conditions, indicating a strong positive correlation between favorable weather and increased demand for electric cycles. Conversely, the demand decreases significantly during adverse weather conditions such as heavy rain.

These insights provide valuable information for Yulu to optimize its operations and marketing strategies based on factors such as weekdays, seasons, and weather conditions. Yulu can leverage these insights to allocate resources effectively, tailor promotions, and enhance user experience to meet the varying demands of customers across different conditions.

HYPOTHESIS TESTING: We are going to perform several hypothesis testing on the data to get the results.

1) 2-Sample T-Test:

i) **Hypothesis:**
   - **(a)** Null Hypothesis (H0): There is no significant difference in the number of electric cycles rented between working days and non-working days.
   - **(b)** Alternative Hypothesis (H1): There is a significant difference in the number of electric cycles rented between working days and non-working days.

ii) **Test Selection:** 2-sample T-test is appropriate for comparing the means of two independent groups (working day vs. non-working day).

iii) **Assumptions Checking:**
   - (a) Normality: We'll check if the number of electric cycles rented follows a normal distribution within each group using statistical tests or visual inspections.
   - (b) Equal Variance: We'll check if the variances of the number of electric cycles rented are approximately equal between the two groups.

iv) **P-value and Conclusion:** After conducting the T-test, we'll find the p-value. If the p-value is less than the chosen significance level (alpha), we reject the null hypothesis and conclude that there is a significant difference in the number of electric cycles rented between working days and non-working days.

2-SAMPLE T-TEST

```
from scipy.stats import ttest_ind

# Subset data for working day and non-working day
working_day_data = df[df['workingday'] == 1]['count']
non_working_day_data = df[df['workingday'] == 0]['count']

# Perform 2-sample T-test
t_statistic, p_value = ttest_ind(working_day_data, non_working_day_data)

# Print results
print("T-statistic:", t_statistic)
print("P-value:", p_value)

# Interpretation
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis. There is a significant difference in the number of electric cycles rented between 
else:
    print("Fail to reject the null hypothesis. There is no significant difference in the number of electric cycles rented
```

```
T-statistic: -2.124770091239619
P-value: 0.033630595775153704
Reject the null hypothesis. There is a significant difference in the number of electric cycles rented between working day
s and non-working days.
```

Insights based on p value: Based on the 2-sample T-test results, we find that the p-value is less than the significance level (alpha), indicating that we reject the null hypothesis. This implies that there is a significant difference in the number of electric cycles rented between working days and non-working days.

- **Higher Demand on Working Days:** The analysis suggests that there is a higher demand for shared electric cycles on working days compared to non-working days. This observation aligns with expectations, as individuals are more likely to use electric cycles for commuting to work or running errands during weekdays.
- **Business Implications:** Understanding the variation in demand between working days and non-working days is crucial for optimizing resource allocation and operational strategies. Yulu may consider deploying more electric cycles or adjusting pricing and promotional strategies to capitalize on the higher demand during weekdays. Additionally, the company could enhance user experience and availability of electric cycles during peak hours on working days to cater to commuter needs effectively.
- **Targeted Marketing:** Yulu could tailor its marketing efforts and promotions to target specific segments of users based on their usage patterns during different days of the week. For instance, promotional campaigns emphasizing convenience and time-saving benefits of using electric cycles for commuting could be targeted towards working professionals on weekdays.

Overall, recognizing the significant difference in demand between working days and non-working days enables Yulu to refine its business strategies and enhance customer satisfaction by providing tailored services and experiences based on varying usage patterns throughout the week.

2) **ANOVA:**

a) **Hypotheses:**

(1) Null Hypothesis (H0): There is no significant difference in the number of cycles rented across different weather conditions and seasons.

(2) Alternative Hypothesis (H1): There is a significant difference in the number of cycles rented across different weather conditions and seasons.

b) **Test Selection:** ANOVA (Analysis of Variance) is suitable for comparing means across multiple groups (weather conditions and seasons).

c) **Assumptions Checking:**

(1) Normality: We'll check if the number of cycles rented follows a normal distribution within each group for weather conditions and seasons.

(2) Equal Variance: We'll verify if the variances of the number of cycles rented are approximately equal across different groups.

d) **P-value and Conclusion:** Following the ANOVA test, we'll obtain the p-value. If the p-value is less than alpha, we reject the null hypothesis and conclude that there is a significant difference in the number of cycles rented across different weather conditions and seasons.

```
from scipy.stats import f_oneway

# Perform ANOVA test
weather_groups = [df[df['weather'] == i]['count'] for i in range(1, 5)]
f_statistic, p_value = f_oneway(*weather_groups)

# Print results
print("F-statistic:", f_statistic)
print("P-value:", p_value)

# Interpretation
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis. There is a significant difference in the number of cycles rented across different
else:
    print("Fail to reject the null hypothesis. There is no significant difference in the number of cycles rented across d
```

```
F-statistic: 43.97308431171039
P-value: 3.209921847226834e-28
Reject the null hypothesis. There is a significant difference in the number of cycles rented across different weather con
ditions.
```

Insights based on the p-value: Based on the ANOVA test results, we find that the p-value is significantly lower than the chosen significance level (alpha), indicating that we reject the null hypothesis. This suggests that there is a significant difference in the number of cycles rented across different weather conditions.

- **Impact of Weather on Demand:** The analysis reveals that weather conditions have a substantial impact on the demand for shared electric cycles. Specifically, there are significant variations in the number of cycles rented across different weather conditions.
- **Preference for Clear Weather:** The majority of shared electric cycles are rented during clear weather conditions compared to misty or rainy weather. This finding suggests that users may prefer to use electric cycles when weather conditions are favorable, such as during clear skies, mild temperatures, and low precipitation.

- **Operational Considerations:** Understanding the relationship between weather and demand is crucial for Yulu to optimize its operations and resource allocation. During periods of high demand, such as clear weather days, Yulu may need to ensure sufficient availability of electric cycles and optimize fleet management strategies to meet user demand effectively.
- **User Behavior Patterns:** These insights into weather-related demand patterns can inform Yulu's marketing and promotional strategies. For example, the company could launch targeted campaigns or incentives to encourage users to rent electric cycles during inclement weather conditions or during periods of low demand.
- **Weather Forecast Integration:** Integrating real-time weather forecasts into Yulu's platform could enable proactive planning and dynamic pricing strategies based on anticipated changes in weather conditions. By leveraging weather data, Yulu can enhance user experience and satisfaction by providing timely and relevant services tailored to weather-related preferences and needs.

Overall, recognizing the significant impact of weather conditions on demand for shared electric cycles empowers Yulu to adapt its operations and strategies accordingly, ensuring optimal service delivery and customer satisfaction across varying weather conditions.

3) Chi-square Test:

a) Hypotheses:

(1) Null Hypothesis (H0): Weather and season are independent of each other.

(2) Alternative Hypothesis (H1): Weather and season are dependent on each other.

b) **Test Selection:** Chi-square test is suitable for analyzing the association between two categorical variables (weather and season).

c) **Assumptions Checking:** There are no specific assumptions to check for the Chi-square test.

d) **P-value and Conclusion:** After conducting the Chi-square test, we'll obtain the p-value. If the p-value is less than alpha, we reject the null hypothesis and conclude that there is a significant association between weather and season.

```
from scipy.stats import chi2_contingency

# Create contingency table
weather_season_table = pd.crosstab(df['weather'], df['season'])

# Perform Chi-square test
chi2_statistic, p_value, dof, expected = chi2_contingency(weather_season_table)

# Print results
print("Chi-square statistic:", chi2_statistic)
print("P-value:", p_value)

# Interpretation
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis. Weather and season are dependent on each other.")
else:
    print("Fail to reject the null hypothesis. Weather and season are independent of each other.")
```

```
Chi-square statistic: 43.69751021011294
P-value: 1.6046004443637115e-06
Reject the null hypothesis. Weather and season are dependent on each other.
```

Insights based on the p-value: Based on the Chi-square test results, we find that the p-value is significantly lower than the chosen significance level (alpha), indicating that we reject the null hypothesis. This suggests that there is a significant association between weather and season.

- **Seasonal Variation in Weather:** The analysis reveals a statistically significant relationship between weather conditions and seasons. This finding suggests that certain weather conditions are more prevalent during specific seasons, indicating a seasonal pattern in weather.
- **Weather Patterns Across Seasons:** Understanding the association between weather and season provides valuable insights into the weather patterns experienced during different times of the year. For example, certain weather conditions such as clear skies or heavy rain may be more common during particular seasons, influencing user behavior and demand for shared electric cycles.
- **Implications for Operations:** Recognizing the dependence of weather on season enables Yulu to anticipate and prepare for seasonal variations in weather-related demand. The company can adjust operational strategies, such as fleet management and maintenance schedules, to align with expected changes in weather patterns across seasons.
- **Targeted Marketing Strategies:** Yulu can leverage the insights gained from the association between weather and season to develop targeted marketing campaigns and promotions. For instance, the company could tailor its messaging and incentives to promote electric cycle rentals during seasons with favorable weather conditions or to encourage usage during periods of inclement weather.
- **Weather-Responsive Services:** Integrating weather data into Yulu's platform enables the provision of weather-responsive services and features. For example, the platform could offer real-time weather updates and route suggestions to users, helping them make informed decisions based on current weather conditions.

By recognizing the dependence of weather on season, Yulu can enhance its operational efficiency, optimize service delivery, and improve customer satisfaction by providing tailored services and experiences that align with seasonal weather patterns.

FINAL INSIGHTS: Based on the results of the hypothesis tests and analysis conducted, we can derive the following final insights:

A) **Significant Variables for Predicting Demand:**

- **Working Day:** The analysis revealed a significant difference in the number of electric cycles rented between working days and non-working days. On average, there was a higher demand for shared electric cycles on working days compared to non-working days (approximately 995,166 cycles rented on working days versus 472,387 cycles rented on non-working days).

- **Weather:** Weather conditions were found to have a significant impact on the demand for shared electric cycles. Clear weather was associated with the highest demand, followed by misty weather, light rain, and heavy rain. For instance, during clear weather, approximately 1,002,070 cycles were rented, while only 164 cycles were rented during heavy rain.
- **Season:** There was a statistically significant difference in the number of cycles rented across different seasons. Although the demand for shared electric cycles remained relatively consistent throughout the year, there was a slight increase in demand during fall compared to other seasons (410,093 cycles rented in fall).

B) **Description of Electric Cycle Demands:**

- **Working Day Impact:** Working days significantly influence the demand for shared electric cycles, with a higher demand observed on these days. This indicates that commuting and daily errands contribute significantly to the demand for electric cycles.
- **Weather Sensitivity:** Weather conditions play a crucial role in shaping the demand for shared electric cycles. Users tend to prefer renting cycles during clear weather, suggesting a preference for favorable weather conditions for outdoor activities and commuting.
- **Seasonal Variation:** While the demand for shared electric cycles remains relatively stable across seasons, there are subtle variations. Understanding seasonal patterns in demand allows for more targeted operational and marketing strategies tailored to seasonal preferences and needs.

C) Operational Implications:

- Yulu can optimize its operational strategies by adjusting fleet management, maintenance schedules, and resource allocation based on the observed variations in demand across different days, weather conditions, and seasons.
- Targeted marketing campaigns and promotions can be developed to capitalize on peak demand periods, such as working days with favorable weather conditions.
- Integration of weather forecasts into Yulu's platform can enhance user experience by providing real-time updates and recommendations based on anticipated weather conditions.

In summary, the significant variables for predicting demand for shared electric cycles in the Indian market include working day, weather, and season. These variables collectively provide valuable insights into the factors driving electric cycle demand and can guide Yulu in optimizing its operations and marketing strategies to meet the varying needs of users across different conditions.

RECOMMENDATIONS:

Based on the insights derived from the analysis, here are some recommendations for Yulu to enhance its operations and optimize its services in the Indian market:

A) Tailored Operational Strategies:

- Develop tailored operational strategies that account for variations in demand across different days, weather conditions, and seasons.
- Allocate resources such as fleet management and maintenance schedules based on anticipated fluctuations in demand.

B) Dynamic Pricing and Incentives:

- Implement dynamic pricing models and incentives to encourage usage during periods of high demand, such as working days with favorable weather conditions.
- Offer discounts or promotions during off-peak periods to incentivize users and balance demand.

C) Weather-Responsive Services:

- Integrate real-time weather forecasts into Yulu's platform to provide users with weather-responsive services and recommendations.
- Offer route suggestions and updates based on current weather conditions to enhance user experience and convenience.

D) Targeted Marketing Campaigns:

- Develop targeted marketing campaigns and promotions tailored to specific user segments based on their preferences and usage patterns.
- Highlight the benefits of using electric cycles for commuting and daily errands, particularly during peak demand periods.

E) Enhanced User Engagement:

- Foster user engagement through gamification, rewards programs, and community-building initiatives.
- Encourage users to share their experiences and recommendations, fostering a sense of belonging and loyalty within the Yulu community.

F) Strategic Partnerships:

- Explore strategic partnerships with local businesses, transportation hubs, and residential complexes to expand Yulu's reach and accessibility.
- Establish Yulu zones at key locations such as metro stations, bus stands, office spaces, and residential areas to enhance convenience and accessibility for users.

G) Continuous Data Analysis:

- Continuously analyze user data, market trends, and feedback to identify emerging patterns and opportunities for improvement.
- Leverage advanced analytics and machine learning algorithms to predict future demand and optimize operational efficiencies proactively.

H) Customer Support and Safety Measures:

- Prioritize customer support and safety measures to build trust and confidence among users. Implement robust safety protocols, user education initiatives, and responsive customer support channels to address user concerns and ensure a positive user experience.

By implementing these recommendations, Yulu can strengthen its position in the Indian micro-mobility market, enhance customer satisfaction, and drive sustainable growth. These strategies will enable Yulu to adapt to evolving user preferences and market dynamics while delivering innovative and convenient mobility solutions to users across India.