Bike-Sharing Analysis and Predictive Modeling

Optimizing Operations and Enhancing User Experience

By,

Shafiq Abubacker

Introduction

- Cities worldwide are embracing bike sharing for sustainable transportation. Key to smooth operations is predicting bike rental demand.
- We present a machine learning model that accurately forecasts the number of bike rentals for any given time period
- This prediction can assist bike sharing service providers in optimizing bike availability, improving user experience, and managing resources efficiency.

Workflow

Data Exploration and Preprocessing

Exploratory Data Analysis

Hypothesis Testing

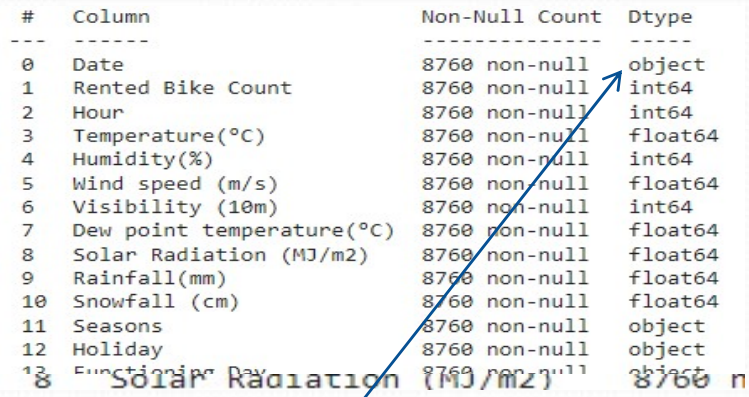
Machine Learning Model Implementation

Evaluation Metrics

Conclusion and Business Insights

Data Overview

- **Data Types:**
  - **Float64:** Temperature, Wind speed, Visibility, Dew point temperature, Solar Radiation, Rainfall, Snowfall
  - **Int64:** Rented Bike Count, Hour, Humidity
  - **Object:** Date, Seasons, Holiday, Functioning Day
- **Non-Null Counts:** All columns have 8760 non-null entries, indicating that there are no missing values in the dataset.



#	Column	Non-Null Count	Dtype
0	Date	8760 non-null	object
1	Rented Bike Count	8760 non-null	int64
2	Hour	8760 non-null	int64
3	Temperature(°C)	8760 non-null	float64
4	Humidity(%)	8760 non-null	int64
5	Wind speed (m/s)	8760 non-null	float64
6	Visibility (10m)	8760 non-null	int64
7	Dew point temperature(°C)	8760 non-null	float64
8	Solar Radiation (MJ/m2)	8760 non-null	float64
9	Rainfall(mm)	8760 non-null	float64
10	Snowfall (cm)	8760 non-null	float64
11	Seasons	8760 non-null	object
12	Holiday	8760 non-null	object

To ensure seamless temporal analysis and accurate date-based operations, we'll convert the "Date" feature from object to date time format.

Data Overview

	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day	month	year	week
count	8760.000000	8760.0	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760	8760	8760	8760	8760.0	8760
unique	NaN	24.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4	2	2	12	2.0	2
top	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Spring	No Holiday	Yes	December	2018.0	Weekdays
freq	NaN	365.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2208	8328	8465	744	8016.0	6264
mean	704.602055	NaN	12.882922	58.226256	1.724909	1436.825799	0.569111	0.148687	0.075068	NaN	NaN	NaN	NaN	NaN	NaN
std	644.997468	NaN	11.944825	20.362413	1.036300	608.298712	0.868746	1.128193	0.436746	NaN	NaN	NaN	NaN	NaN	NaN
min	0.000000	NaN	-17.800000	0.000000	0.000000	27.000000	0.000000	0.000000	0.000000	NaN	NaN	NaN	NaN	NaN	NaN
25%	191.000000	NaN	3.500000	42.000000	0.900000	940.000000	0.000000	0.000000	0.000000	NaN	NaN	NaN	NaN	NaN	NaN
50%	504.500000	NaN	13.700000	57.000000	1.500000	1698.000000	0.010000	0.000000	0.000000	NaN	NaN	NaN	NaN	NaN	NaN
75%	1065.250000	NaN	22.500000	74.000000	2.300000	2000.000000	0.930000	0.000000	0.000000	NaN	NaN	NaN	NaN	NaN	NaN
min	0.000000	NaN	-17.800000	0.000000	0.000000	27.000000	0.000000	0.000000	0.000000	NaN	NaN	NaN	NaN	NaN	NaN

- **Count:** Indicates the number of non-null entries for each column. All columns have 8760 non-null entries, suggesting that there are no missing values.
- **Unique:** Displays the number of unique values in each column. For instance, “Hour" has 24 unique values, suggesting Hourly observations .
- **Top:** Represents the most frequently occurring value in each column. For example, the most common season is "Spring," and the most common holiday status is "No Holiday.“
- **Frequency (freq):** Specifies the frequency of the top value in each column. For instance, "01/12/2017" (presumably in the format MM/DD/YYYY) occurs 24 times in the "Date" column.

Mean: Represents the mean (average) of each numerical column. For instance, the mean of the "Rented Bike Count" is approximately 704.60.

Standard Deviation (std): Indicates the standard deviation, a measure of the amount of variation or dispersion, for each numerical column.

Min: Represents the minimum value observed in each numerical column.

25% (Q1): Represents the 25th percentile, also known as the first quartile. 25% of the data falls below this value.

50% (Q2 or Median): Represents the median or the 50th percentile. It is the middle value of the dataset.

75% (Q3): Represents the 75th percentile, also known as the third quartile. 75% of the data falls below this value.

Max: Represents the maximum value observed in each numerical column.

Data Overview

- The dataset comprises 8,760 observations, and each observation corresponds to a unique hour. This structure covers the entire duration of 365 days in a year, reflecting a daily and hourly granularity.
- There are 14 features in the dataset, encompassing various aspects such as date, time, weather conditions, and bike rental counts. These features are likely to be relevant for analyzing the factors influencing bike rentals.
- There are no missing values (null values) in any of the columns, ensuring completeness and making it easier to proceed with analysis and modeling without the need for imputation.
- All entries in the dataset are distinct, with no duplicate rows. This helps maintain the integrity of the data and prevents biases that duplicates might introduce.

```
# Dataset Duplicate Value Count
df.duplicated().value_counts()
```

```
False    8760
dtype: int64
```

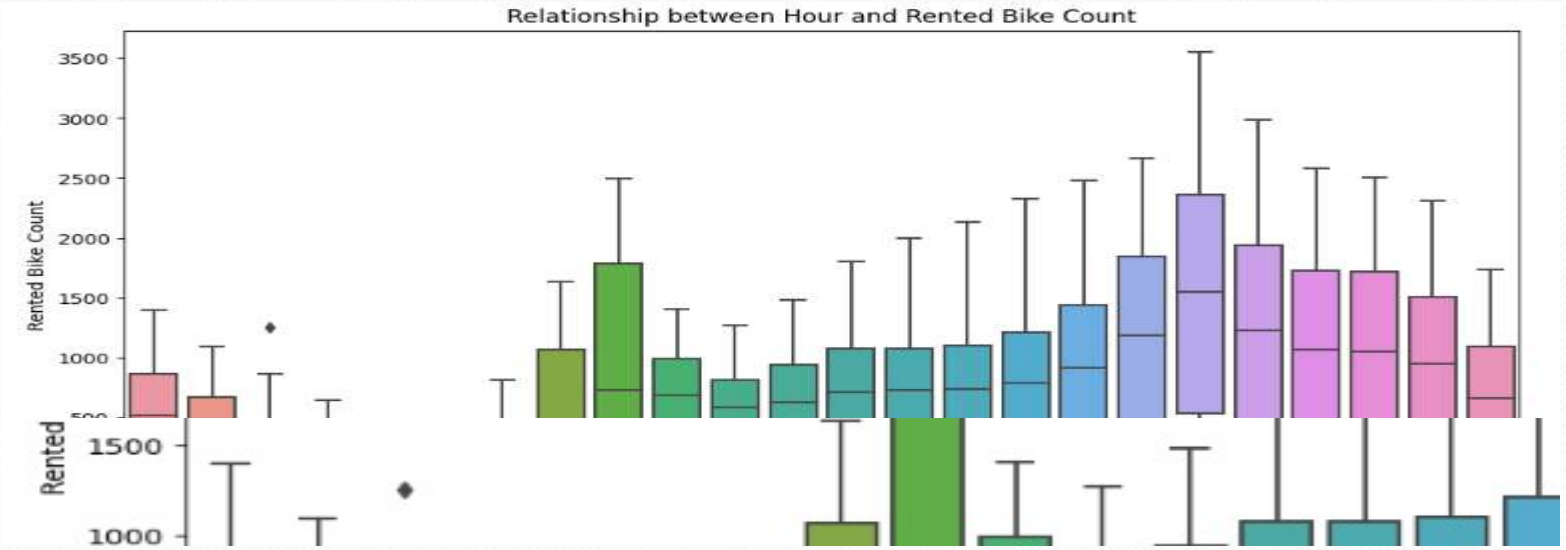
```
# Missing Values/Null Values Count
df.isnull().sum().sort_values(asc
```

```
Date                          0
Rented Bike Count             0
Hour                          0
Temperature(°C)               0
Humidity(%)                   0
Wind speed (m/s)              0
Visibility (10m)              0
Dew point temperature(°C)     0
Solar Radiation (MJ/m2)       0
```

Data Wrangling

- **Conversion of 'Date' Column:**
  - Converted the 'Date' column to the datetime data type using pd.to_datetime().
  - This step ensures that the 'Date' column is treated as date objects, allowing for easier manipulation and analysis.
- **Feature Engineering from 'Date':**
  - Split the 'Date' into day of the week, month, and year, creating three new columns.
  - This feature engineering provides additional dimensions to the dataset, allowing for analysis based on temporal patterns.
- **Handling 'Hour' Column:**
  - Converted the 'Hour' column to the categorical data type.
  - This conversion is done to treat hours as categories rather than numerical values, which might be more suitable for certain analyses.
- **Separation of Features:**
  - Separated the dataset into numeric and categorical features.
  - This step distinguishes between features that contain numerical values and those that contain categorical values. It sets the stage for specific analyses based on feature types.
- **Creation of 'Week' Column:**
  - Created a new column 'week' to distinguish between weekdays and weekends.
  - This additional column facilitates the exploration of variations in bike rental patterns between weekdays and weekends.
- **Dropping 'day_of_week' Column:**
  - Dropped the 'day_of_week' column from both the main dataframe and the categorical features dataframe.
  - This decision may have been made for various reasons, such as redundancy or to streamline the dataset. It's essential to understand why specific columns are dropped.

Hourly Patterns



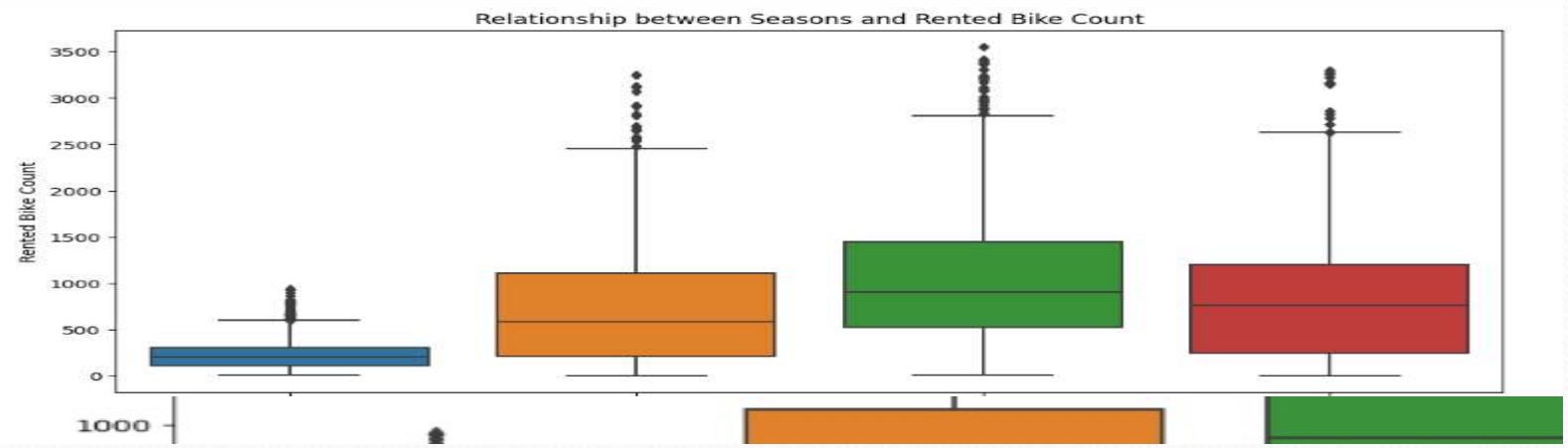
Peak Hours:

- **Observation:** Bike rentals are highest during morning rush hours (around 7-9 AM) and evening rush hours (around 5-7 PM), indicating common commuting patterns.
- **Strategy:** Optimize bike distribution during peak hours by moving bikes from low-demand stations to high-demand stations. This helps in reducing shortages and improving customer satisfaction during the busiest times.

Low Usage Hours:

- **Observation:** Rentals are lowest during late-night and early morning hours (1-5 AM), suggesting reduced demand during those times.
- **Strategy:** Implement dynamic pricing strategies. Adjust rates based on demand, charging higher prices during peak hours and offering discounts during low-demand hours. This can increase revenue during peak times while encouraging usage during off-peak hours..

Seasonal Trends



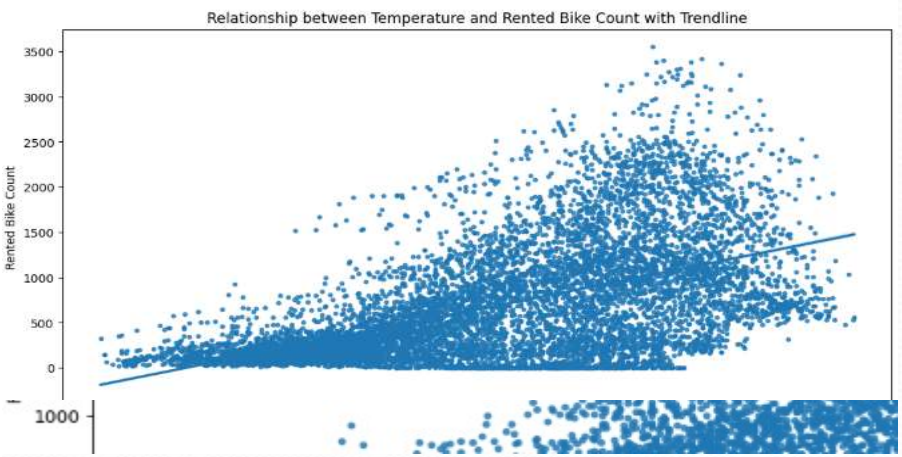
Highest Demand:

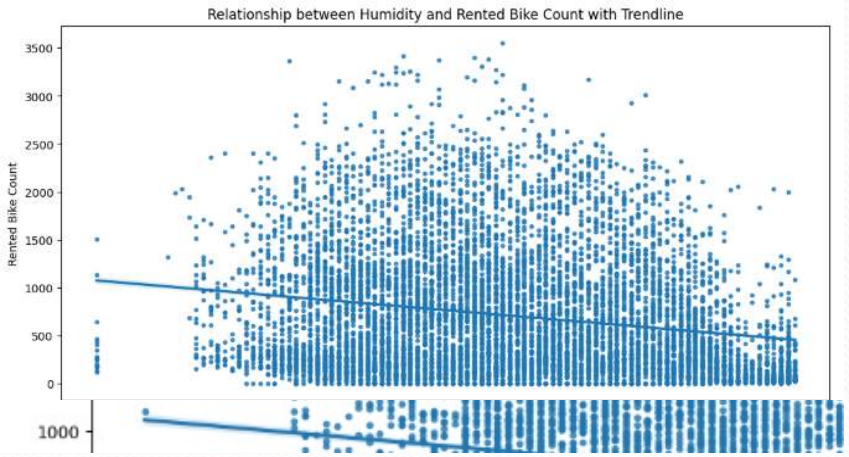
- **Observation:** Autumn (fall) and summer has the highest median rented bike count.
- **Interpretation:** Favorable weather conditions and outdoor activities during these seasons likely drive increased bike usage.

Lowest Demand:

- **Observation:** Winter has the lowest median count
- **Interpretation:** Decrease in demand during winter, potentially due to colder temperatures, shorter days, and less conducive weather for cycling.

Weather Factors

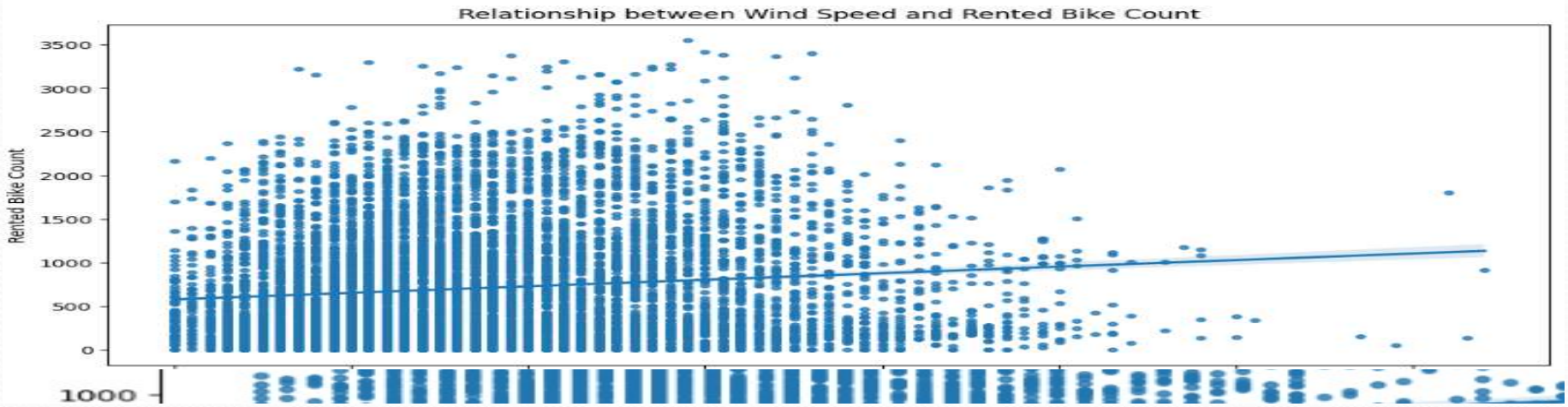




- **Positive Correlation:**
  - The positive slope of the trendline indicates that, on average, higher temperatures are associated with a higher number of rented bikes.
- **Business Implications**
  - The bike-sharing service can optimize operations during warmer days when demand for bike rentals is higher.
  - Consider increasing the number of available bikes or adjusting staffing levels to meet the expected demand during periods of favorable weather.

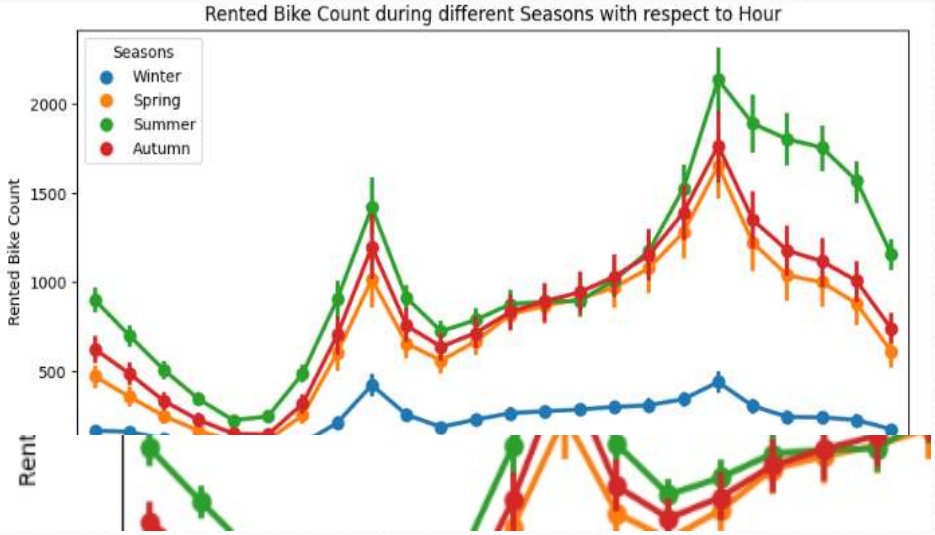
- **Negative Correlation:**
  - The scatter plot indicates a general negative correlation between humidity and rented bike count where humidity increases, the number of rented bikes tends to decrease.
- **Business Implications**
  - Enhance resource allocation and reduce shortages by incorporating humidity data into forecasting models.
  - Offer incentives during low-humidity periods to encourage bike rentals.

Wind Speed Relationship



- The positive trendline shows that as wind speed increases, the number of rented bikes tends to increase.
- Wind speed may influence different customers in various ways.
- Identify customer segments that may be particularly interested in cycling during windy conditions
- Implement targeted marketing campaigns during seasons or periods when wind speed is typically higher.

Hourly Trends



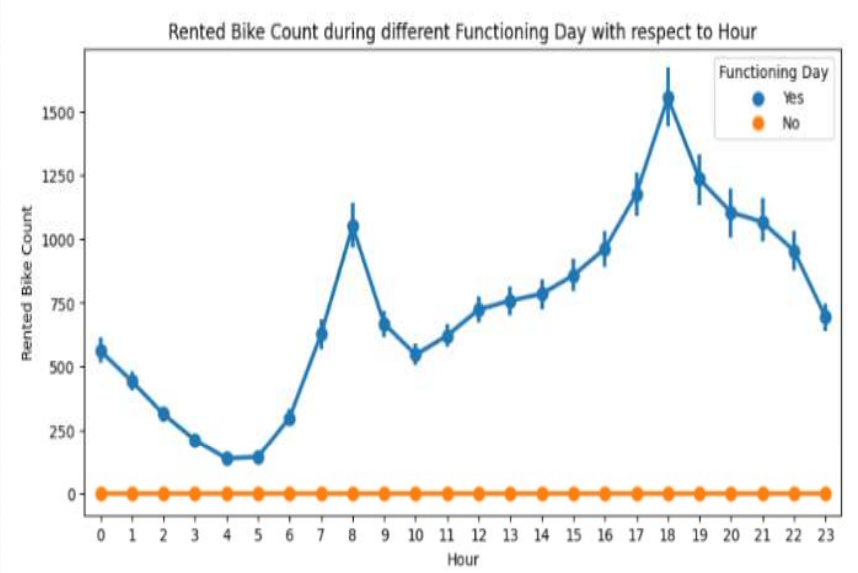
Season:

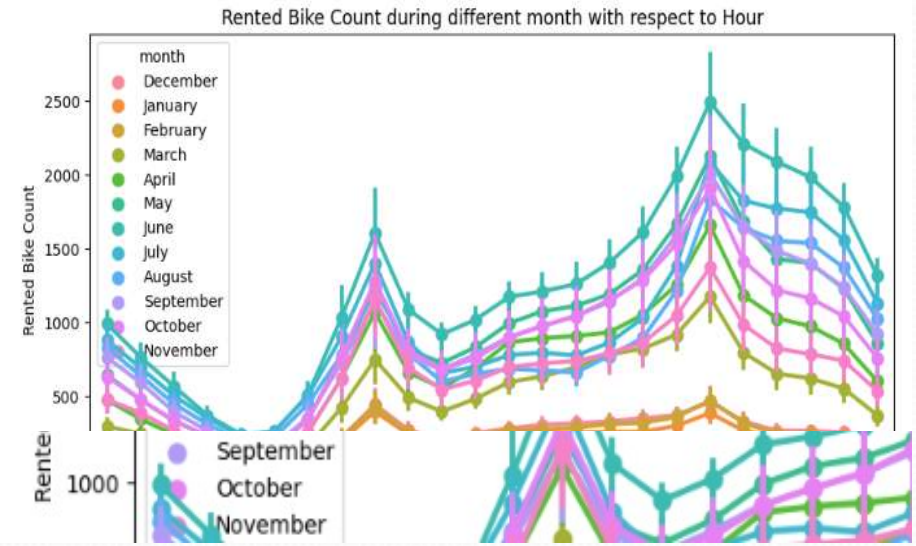
- **Observation:** Lower demand during winter compared to other seasons.
- **Implication:** Challenge to incentivize bike usage during colder months.

Holiday:

- **Observation:** Lower demand during holidays; higher demand on non-holidays
- **Implication:** Potential for targeted marketing during non-holidays to boost revenue.

Hourly Trends





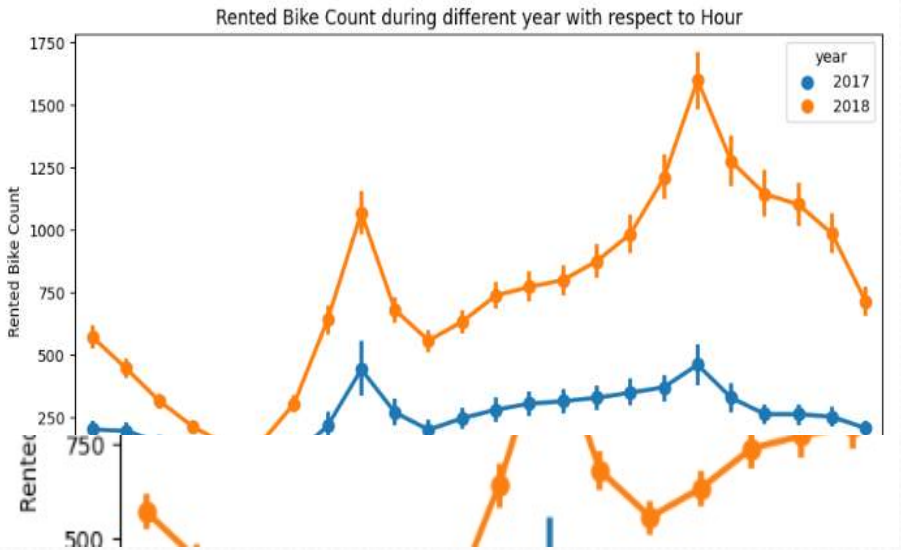
Functioning Day:

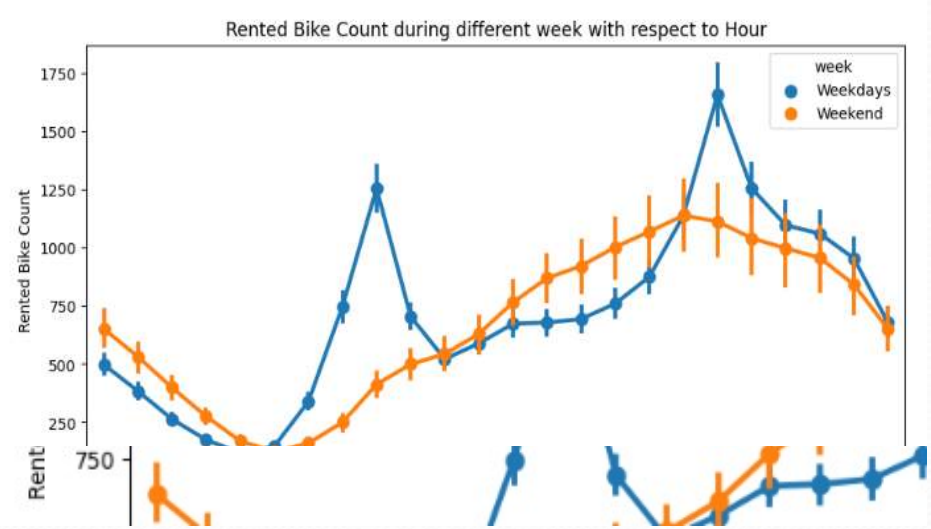
- **Observation:** Absence of Functioning Day corresponds to lack of demand.
- **Implication:** Consistent service on functioning days crucial for user satisfaction.

Month:

- **Observation:** Lower demand in December, January, and February (winter months).
- **Implication:** Potential challenges during winter; strategies needed to incentivize usage.

Hourly Trends





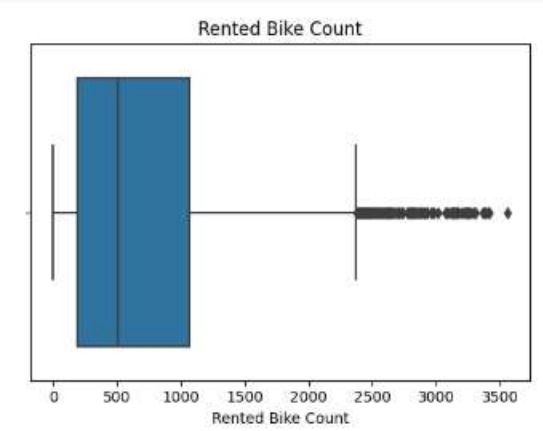
Year:

- **Observation:** Increase in demand from 2017 to 2018.
- **Implication:** Positive growth; focus on expansion, infrastructure improvement, and user experience.

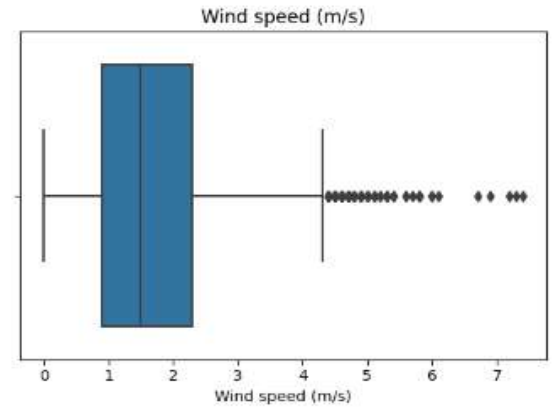
Week:

- **Observation:** Weekdays exhibit higher demand during office hours; weekends see increased demand in the afternoon.
- **Implication:** Tailoring marketing efforts based on weekdays can enhance user engagement.

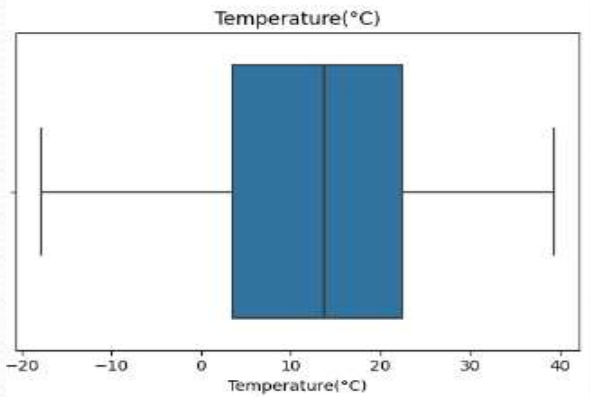
Outliers and Data Distribution



- The distribution is skewed to the right, indicating more data points on the high end of the range.
- There are instances of high bike rental counts, potentially representing peak demand periods.



- The distribution has two peaks, indicating two groups of data points that are more common than others.
- Different wind speed conditions may lead to varied patterns of bike rentals. This could be related to weather conditions affecting user preferences.



- The temperature range is from -20°C to 40°C.
- The boxplot does not show many outliers. The temperature range suggests diverse weather conditions, and the absence of extreme outliers may indicate data consistency.

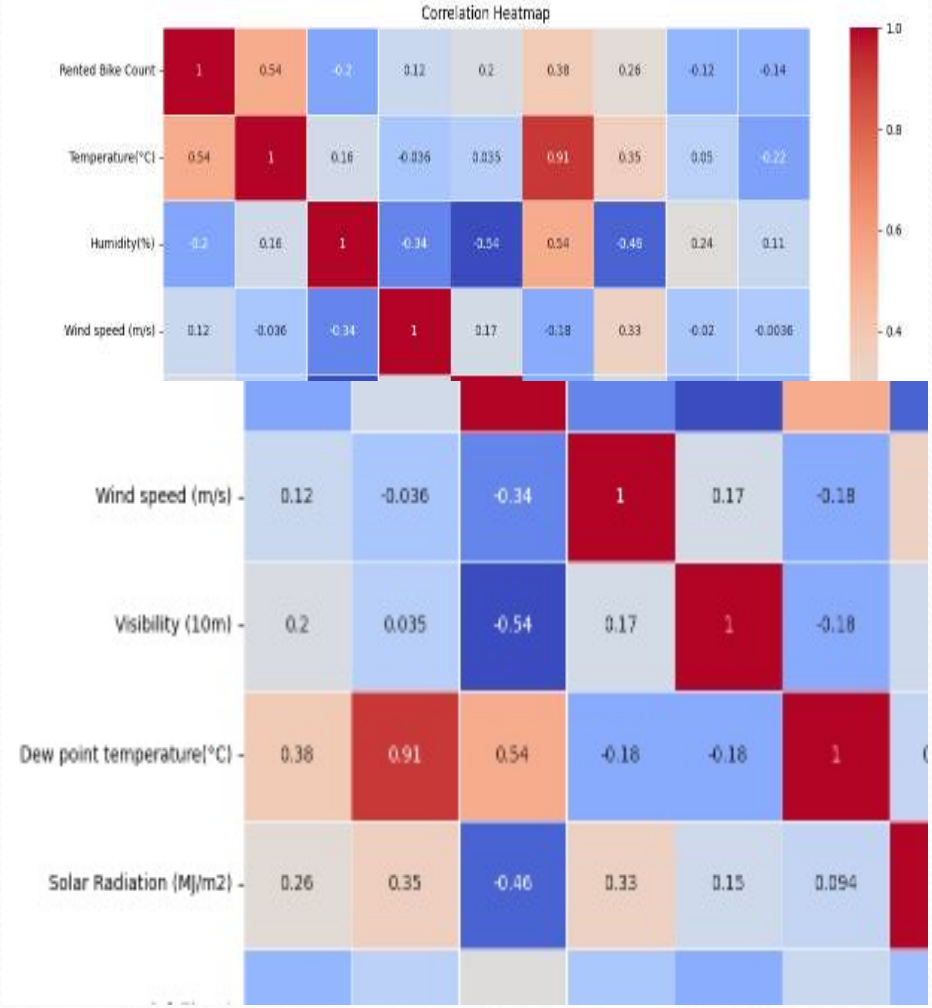
Correlation

Positive Correlations:

- **Rented Bike Count and Temperature(°C):** There is a strong positive correlation (0.54), indicating that as the temperature increases, the number of rented bikes tends to increase.
- **Rented Bike Count and Dew point temperature(°C):** Another strong positive correlation (0.38), suggesting that higher dew point temperatures are associated with more bike rentals.
- **Rented Bike Count and Solar Radiation (MJ/m2):** A moderate positive correlation (0.26), implying that higher solar radiation is associated with increased bike rentals.

Negative Correlations:

- **Rented Bike Count and Humidity(%):** A moderate negative correlation (-0.20), indicating that higher humidity is associated with slightly fewer bike rentals.
- **Rented Bike Count and Rainfall(mm):** A weak negative correlation (-0.12), suggesting that more rainfall is associated with slightly fewer bike rentals.



Multicollinearity
Variance Inflation Factor

- **Temperature and Dew Point Temperature:**
  - A strong positive correlation (0.91) indicates a close relationship between these two variables, likely due to their shared connection to humidity and atmospheric conditions.

  - Dropping the "Dew point temperature(°C)" variable could help alleviate multicollinearity concerns, especially given its strong correlation with other variables. This step may improve the stability and interpretability of regression models.

Hypothesis Testing

Hypothesis 1: Correlation between Temperature and Bike Count

Statistical Test: Pearson Correlation Coefficient (pearsonr)

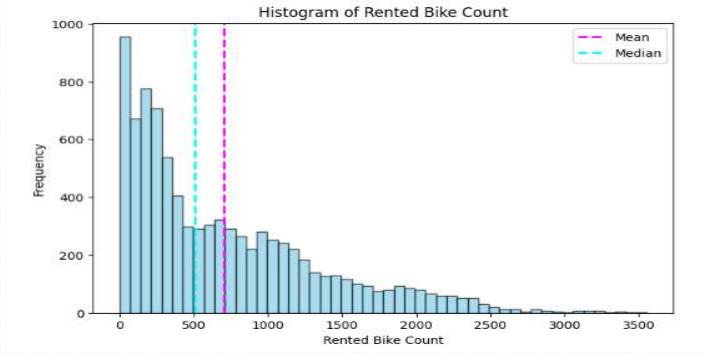
- Here we are exploring the correlation between temperature and the number of rented bikes.
- The Pearson Correlation Coefficient is an appropriate choice when assessing linear relationships between two continuous variables.
- As the p-value obtained from the test is less than the chosen significance level which is 0.05, we reject the null hypothesis. This implies that there is sufficient evidence to support the presence of a significant correlation.

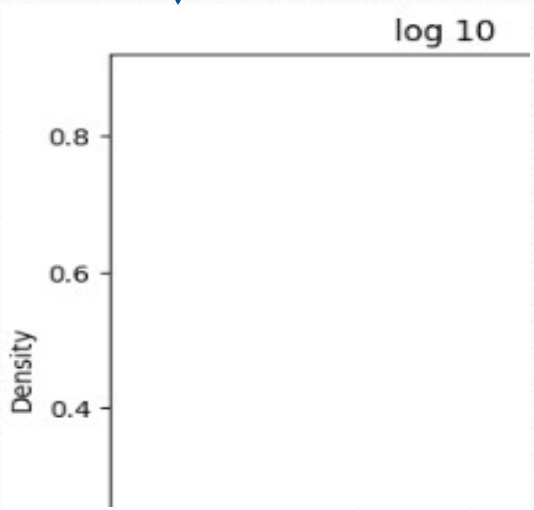
Hypothesis 2: Difference in Bike Counts between Holidays and Non-Holidays

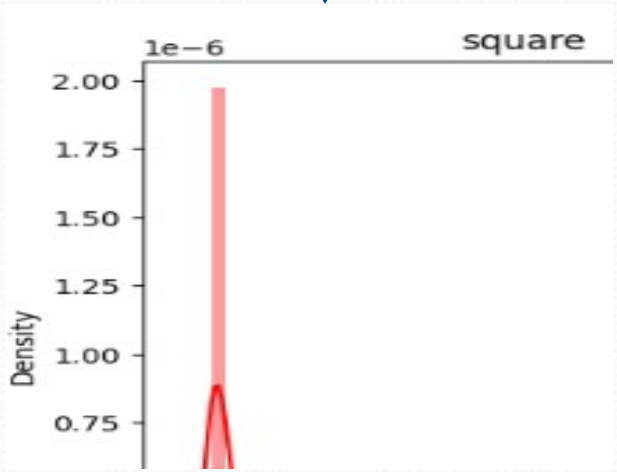
Statistical Test: Independent Samples T-Test (ttest_ind)

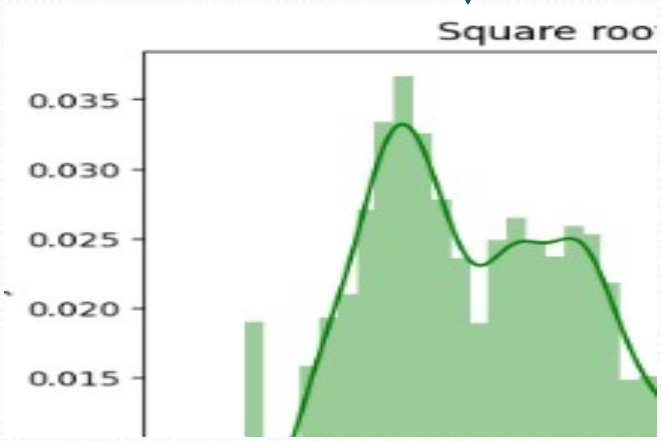
- Here we are comparing the mean number of rented bikes between two independent groups: holidays and non-holidays.
- The Independent Samples T-Test is suitable for comparing means of two independent groups.
- This test helps us determine whether there is a significant difference in the average bike counts on holidays compared to non-holidays.
- As the p-value obtained from the test is less than the chosen significance level (alpha), we reject the null hypothesis. This suggests that there is evidence of a significant difference in mean bike counts between holidays and non-holidays.

Normalization









Data Splitting

```
# Split your data to train and test. Choose Splitting ratio wisely.

# Define features (X) and target variable (y)
X = df_encoded.drop('Rented Bike Count', axis=1)  # Features excluding the target variable
y = np.sqrt(df_encoded['Rented Bike Count'])  # Target variable

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Display the shape of the resulting sets
print("Train set shape:", X_train.shape, y_train.shape)
print("Test set shape:", X_test.shape, y_test.shape)
```

```
# Display the shape of the resulting sets
print("Train set shape:", X_train.shape, y_train.shape)
```

- Allocating 80% of the dataset to training (X_train, y_train) ensures a substantial amount of data for building and optimizing the machine learning model.
- The remaining 20% is dedicated to testing (X_test, y_test), allowing for a robust evaluation of the model's performance on unseen data.
- It ensures that the model is trained on a diverse set of examples while still having a substantial amount of data for robust testing, helping to avoid overfitting or underfitting issues.

Model Evaluation Metrics

- **Mean Absolute Error (MAE):**
  - Measures the average absolute difference between actual and predicted values.
  - Easy to interpret, gives an average magnitude of the errors.
- **Mean Squared Error (MSE):**
  - Measures the average of squared differences between actual and predicted values.
  - Emphasizes larger errors due to squaring, which may penalize outliers more.
- **Root Mean Squared Error (RMSE):**
  - The square root of MSE, providing a more interpretable scale.
  - Reflects the standard deviation of the residuals.
- **R-squared Score:**
  - Indicates the proportion of the variance in the dependent variable explained by the independent variables.
  - Ranges from 0 to 1, with 1 indicating a perfect fit.

Power transformation can help mitigate the impact of skewed distributions and improve the assumptions of linear regression models, enhancing their performance.

```
# Function to Calculate Score Metrics
from sklearn.preprocessing import PowerTransformer

def score_metrics(actual, predicted):
    # Calculate Mean Absolute Error
    mae = mean_absolute_error(actual**2, predicted**2)
    print('Mean Absolute Error:', mae)

    # Calculate Mean Squared Error
      print('Mean Absolute Error:', mae)

      # Calculate Mean Squared Error
      mse = mean_squared_error(actual**2
      print('Mean Squared Error:', mse)

      # Calculate Root Mean Squared Erro
      rmse = np.sqrt(mse)
      print('Root Mean Squared Error:',

      # Calculate R-squared Score
```

ML Model 1 – Linear Regression

Mean Absolute Error (MAE): The MAE is a measure of the average absolute errors between the predicted and actual values. Here, the MAE is approximately 4.11, which means, on average, my model's predictions are off by around 4.11 units.

Mean Squared Error (MSE): The MSE is a measure of the average squared errors between the predicted and actual values. In this case, the MSE is approximately 28.89.

Root Mean Squared Error (RMSE): The RMSE is the square root of MSE and provides an interpretable measure of the average errors in the same units as the target variable. In this case, the RMSE is approximately 5.37.

R-squared Score: The R-squared score measures the proportion of the variance in the dependent variable that is predictable from the independent variables. Here, the R-squared score is approximately 0.81, indicating that my model explains about 81% of the variance in the target variable.

Adjusted R-squared Score: The Adjusted R-squared score adjusts the R-squared value based on the number of predictors in the model. It penalizes the inclusion of irrelevant predictors. Here, the Adjusted R-squared score is approximately 0.81, which is very close to the R-squared score.

```
# ML Model - 1 Implementation

#Linear Regression

# Initialize
reg= LinearRegression()


# Fit the Algorithm"

reg.fit(X_train, y_train)
reg.score(X_train, y_train)

#Predict one te model
reg_pred = reg.predict(X_test)
```

```
reg.score(X_train, y_train)

#Predict one te model
reg_pred = reg.predict(X_test)


# Calculate Mean Absolute Error
mae = mean_absolute_error(y_test
```

```
# Calculate R-squared Score
r2 = r2_score(y_test, reg_pred)
print('R-squared Score:', r2)

# Calculate Adjusted R-squared Score (for
n = X_test.shape[0]  # Number of observati
p = X_test.shape[1]  # Number of predictor
adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n
print('Adjusted R-squared Score:', adjuste
```

ML Model 2 – Lasso Regression

Mean Absolute Error (MAE): Approximately 4.114. This indicates that, on average, the Lasso model's predictions are off by around 4.114 units.

Mean Squared Error (MSE): Approximately 28.891. MSE measures the average squared difference between predicted and actual values, giving more weight to larger errors.

Root Mean Squared Error (RMSE): Approximately 5.375. RMSE is the square root of MSE and provides an interpretable metric in the same units as the target variable. It is useful for understanding the magnitude of errors.

R-squared Score: Approximately 0.812. The R-squared score measures the proportion of the variance in the dependent variable that is predictable from the independent variables. An R-squared of 0.812 indicates that the Lasso model explains about 81.2% of the variance in the target variable.

Adjusted R-squared Score: Approximately 0.806. The Adjusted R-squared score accounts for the number of predictors in the model, penalizing for unnecessary variables. It is slightly lower than the R-squared, suggesting that adding more predictors may not significantly improve the model.

```
L1 = Lasso(alpha=0.001, max_iter=1000)

L1.fit(X_train, y_train)
L1_pred = L1.predict(X_test)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, L1_pred)
print('Mean Absolute Error (MAE):', mae)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, L1_pred)
print('Mean Squared Error (MSE):', mse)
```

```
# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, L1_pred)
print('Mean Squared Error (MSE):', mse)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)
print('Root Mean Squared Error (RMSE):', r

# Calculate R-squared Score
r2 = r2_score(y_test, L1_pred)
print('R-squared Score:', r2)
```

ML Model 3 – Ridge Regression

Mean Absolute Error (MAE): Approximately 4.11. This indicates that, on average, the Ridge model's predictions are off by around 4.11 units.

Mean Squared Error (MSE): Approximately 28.89. MSE measures the average squared difference between predicted and actual values, giving more weight to larger errors.

Root Mean Squared Error (RMSE): Approximately 5.37. RMSE is the square root of MSE and provides an interpretable metric in the same units as the target variable. It is useful for understanding the magnitude of errors.

R-squared Score: Approximately 0.81. The R-squared score measures the proportion of the variance in the dependent variable that is predictable from the independent variables. An R-squared of 0.81 indicates that the Ridge model explains about 81% of the variance in the target variable.

Adjusted R-squared Score: Approximately 0.81. The Adjusted R-squared score accounts for the number of predictors in the model, penalizing for unnecessary variables. It is slightly lower than the R-squared, suggesting that adding more predictors may not significantly improve the model.

```
# ML Model - 3 Implementation

ridge_model = Ridge(alpha=1.0)

# Fit the Algorithm

ridge_model.fit(X_train, y_train)

# Predict on the model

y_pred = ridge_model.predict(X_test)
```

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

# Calculate Adjusted R-squared Score (for
n = X_test.shape[0]  # Number of observati
p = X_test.shape[1]  # Number of predictor
adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n

print(f'Mean Absolute Error (MAE): {round(
print(f'Mean Squared Error (MSE): {round(m
print(f'Root Mean Squared Error (RMSE): {r
print(f'R-squared Score: {round(r2, 2)}')
print(f'Adjusted R-squared Score: {round(a
```

ML Model 4 – Polynomial Regression

Mean Absolute Error (MAE): Approximately 2.30. This indicates that, on average, the Polynomial Regression model's predictions are off by around 2.30 units.

Mean Squared Error (MSE): Approximately 11.97. MSE measures the average squared difference between predicted and actual values, giving more weight to larger errors.

Root Mean Squared Error (RMSE): Approximately 3.46. RMSE is the square root of MSE and provides an interpretable metric in the same units as the target variable. It is useful for understanding the magnitude of errors.

R-squared Score: Approximately 0.92. The R-squared score measures the proportion of the variance in the dependent variable that is predictable from the independent variables. An R-squared of 0.92 indicates that the Polynomial Regression model explains about 92.2% of the variance in the target variable.

Adjusted R-squared Score: Approximately 0.92. The Adjusted R-squared score accounts for the number of predictors in the model, penalizing for unnecessary variables. It is slightly lower than the R-squared, suggesting that adding more predictors may not significantly improve the model.

```
# Define the degree of the polynomial features
degree = 2

# Create a pipeline with PolynomialFeatures and LinearRegression
model = make_pipeline(PolynomialFeatures(degree), LinearRegression())

# Fit the model
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
```

```
# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calculate Root Mean Squared Error (RMSE)
```

```
# Print the results
print(f'Mean Absolute Error (MAE): {mae}')
print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {r
print(f'R-squared Score: {r2}')
print(f'Adjusted R-squared Score: {adjuste
```

```
Mean Absolute Error (MAE): 2.3037009993562
```

ML Model Overall Metrics

Linear Regression:
Performance:
- Mean Absolute Error (MAE): 4.11
- Mean Squared Error (MSE): 28.89
- Root Mean Squared Error (RMSE): 5.37
- R-squared Score: 0.81

Considerations:
- Good performance, explaining about 81% of the variance.

Ridge Regression:
Performance:
- MAE: 4.11
- MSE: 28.89
- RMSE: 5.37
- R-squared Score: 0.81

Considerations:
- Similar performance to linear regression

Lasso Regression:
Performance:
- MAE: 4.11
- MSE: 28.89
- RMSE: 5.38
- R-squared Score: 0.81

Considerations:
- Similar performance to linear regression.

Polynomial Regression:
Performance:
- MAE: 2.30
- MSE: 11.97
- RMSE: 3.46
- R-squared Score: 0.92

Considerations:
- Improved performance with a higher R-squared score.

Final Choice: Polynomial Regression

The polynomial regression model shows the best overall performance based on the R-squared score and other evaluation metrics. It has a higher R-squared score, indicating that it explains a larger proportion of the variance in the target variable. The lower MAE, MSE, and RMSE also suggest improved accuracy compared to linear, lasso, and ridge regression.

Business Impact

- **Temperature Impact:**
  - There is a significant positive correlation between temperature and the number of rented bikes. Warmer temperatures are associated with increased bike rentals, suggesting a seasonal pattern.
- **Seasonal Trends:**
  - Bike rentals exhibit seasonal variations, with higher demand during autumn and summer. Understanding these trends allows for better resource allocation and marketing strategies tailored to specific seasons.
- **Hourly Patterns:**
  - Peak bike rentals occur during morning and evening rush hours, indicating a strong connection to commuting patterns. Low usage hours are observed during late-night and early morning hours.
- **Weather Factors:**
  - Humidity and wind speed influence bike rentals. High humidity is associated with decreased rentals, while wind speed shows a positive association with bike counts. Dynamic pricing and targeted marketing during specific weather conditions can be implemented.
- **Seasonal Marketing Strategies:**
  - Different seasons require tailored marketing and operational strategies. For instance, promotions and inventory adjustments during peak seasons can optimize business performance.
- **Operational Optimization:**
  - Optimal operational conditions involve adjusting bike availability and staff levels based on temperature, time of day, and seasonal demand. This ensures that resources are efficiently allocated during peak demand periods.
- **Holiday Impact:**
  - Bike demand is lower during holidays, suggesting potential shifts in user behavior. Marketing efforts or promotions may be employed to encourage bike rentals during holiday periods.
- **Predictive Modeling:**
  - Different machine learning models, including linear regression, Lasso, Ridge, and polynomial regression, were implemented to predict bike counts. The polynomial regression model demonstrated the best performance, achieving a high R-squared score.
- **Square Root Transformation:**
  - A square root transformation was applied to the target variable (bike count) for normalization, enhancing the model's predictive performance.
- **Evaluation Metrics:**
  - Evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared, and Adjusted R-squared were used to assess the performance of regression models.
- **Business Impact:**
  - Insights gained from the analysis can lead to positive business impacts, including optimized operations, targeted marketing, and seasonal strategies.
- **Consideration for Negative Impacts:**
  - While implementing strategies based on insights, it's crucial to consider potential negative impacts, such as user inconvenience or operational costs.

Conclusion

- **Understanding Bike-Sharing Patterns:**
  - The dataset offered valuable insights into the dynamics of bike-sharing, revealing distinct patterns influenced by weather conditions, seasons, and temporal factors.
  - Visualization played a pivotal role in uncovering trends, allowing us to identify peak demand periods and factors affecting bike rentals.
- **Influence of Weather Factors:**
  - Temperature and humidity emerged as crucial weather factors impacting bike rental counts. Positive correlations with temperature indicated increased usage during warmer periods, while negative correlations with humidity suggested a potential aversion to cycling in humid conditions.
- **Seasonal and Temporal Trends:**
  - Seasonal variations showcased higher demand during favorable weather conditions, with autumn and summer leading in bike rentals.
  - Temporal analysis highlighted the impact of holidays and functioning days, emphasizing the need for tailored strategies during these periods.
- **Hypothesis Testing and Statistical Insights:**
  - Hypothesis testing affirmed the significance of temperature in predicting bike rentals. Pearson correlation indicated a meaningful relationship between temperature and rented bike count.
  - Distinct differences in bike counts during holidays versus non-holidays were confirmed through an independent samples t-test, guiding strategic planning.
- **Machine Learning Model Evaluation:**
  - Multiple regression models, including Linear Regression, Lasso, Ridge, and Polynomial Regression, were implemented and evaluated.
  - The Polynomial Regression model emerged as the most promising, achieving a high R-squared score and demonstrating superior predictive capabilities.
- **Model Interpretation and Business Impact:**
  - The selected Polynomial Regression model provides a powerful tool for predicting bike rental counts based on weather and temporal features.
  - Insights gained from the analysis can inform operational decisions, marketing strategies, and resource allocation, ultimately contributing to the efficiency and profitability of the bike-sharing service.

In conclusion, this project not only uncovered valuable patterns in bike-sharing behavior but also equipped stakeholders with predictive models for informed decision-making. The intersection of data science, statistical analysis, and machine learning has provided a holistic approach to understanding and optimizing bike-sharing operations. The insights gained are poised to drive positive business impacts and enhance the overall user experience.