BIKE SHARING DEMAND PREDICTION

Project Type – Regression

Name - Abhay Kumar

Contents

1. INTRODUCTION

Urban mobility is a critical concern in today's fast-paced, densely populated cities, and bike-sharing programs have emerged as a sustainable and convenient solution to alleviate some of the transportation challenges. In densely populated areas like Seoul, these programs offer a practical means of improving public mobility and reducing traffic congestion. However, a significant hurdle in ensuring the success of these bike-sharing systems is the consistent supply of bicycles for rental, particularly during peak demand periods. This project focuses on harnessing the power of historical data to predict and optimize bike-sharing services in Seoul, with a particular emphasis on data-driven insights to address this challenge.

With a dataset comprising approximately 8,760 records and 14 attributes, this project embarks on a journey to transform raw data into actionable knowledge. The initial steps involve importing the dataset and the necessary libraries, followed by conducting exploratory data analysis (EDA) to gain a deeper understanding of the data. During this stage, the project identifies and addresses outliers and missing values, ensuring that the data is clean and ready for analysis. Moreover, data transformations are applied to ensure compatibility with machine learning models.

Recognizing the issue of target class imbalance, this project implements square root normalization to level the playing field. The subsequent stage delves into model building, where 11 different machine learning models are trained using the cleaned and scaled data. To evaluate model performance, a battery of metrics is employed, providing a comprehensive view of their capabilities. Particular emphasis is placed on two key metrics - the R2 score (coefficient of determination) and RMSE score (Root Mean Square Error). The R2 score, being scale-independent, allows for a direct comparison of model performance across different target variables and units of measurement.

Ultimately, this project endeavours to address the practical challenge of predicting bike-sharing demand in urban environments, enhancing public access to sustainable transportation options. Through the application of machine learning and comprehensive model evaluation, it aims to provide invaluable insights and solutions that can benefit both bike-sharing service providers and the public, creating a win-win situation for all stakeholders.

Key highlight:

Urban mobility, Bike-sharing programs, Sustainable transportation, Data-driven insights, Machine learning models, R2 score, RMSE score, Predicting bike-sharing demand, Service providers, Public access

2. PROJECT SUMMARY

The objective of this project is to improve public mobility and convenience through bike-sharing programs in urban areas. A key challenge in these programs is ensuring a consistent supply of bikes for rental. Bike-sharing systems offer automated rental and return services, allowing individuals to access and return bicycles at various locations. This project focuses on leveraging historical data, including factors such as temperature and time, to predict the demand for bike-sharing services in Seoul.

Data Preparation:

- The dataset consisted of approximately 8,760 records and 14 attributes.
- Initial steps involved importing the dataset and necessary libraries, followed by conducting exploratory data analysis (EDA).
- Outliers and missing values were identified and addressed, and data transformations were applied to ensure compatibility with machine learning models.
- To handle target class imbalance, square root normalization was applied to the data.

Model Building and Evaluation:

- The cleaned and scaled data was used to train 11 different machine learning models.
- Multiple metrics were used to evaluate model performance, providing a comprehensive view of their capabilities.
- Particular emphasis was placed on the R2 score (coefficient of determination) and RMSE score (Root Mean Square Error).
- The R2 score is a scale-independent metric, allowing for direct model performance comparison across different target variables and units of measurement.

This project seeks to address the practical challenge of predicting bike-sharing demand in urban environments, enhancing public access to sustainable transportation options. Through the application of machine learning and comprehensive model evaluation, it aims to provide insights and solutions that can benefit both service providers and the public.

3. PROBLEM STATEMENT

In metropolitan areas, bike-sharing programs have become a key feature, enhancing public mobility and convenience. However, maintaining a consistent supply of rental bikes is a critical challenge. These bike-sharing systems, automated and efficient, allow people to rent and return bikes at multiple locations. This project is centered on leveraging historical data, encompassing factors like temperature and time, to forecast the demand for Seoul's bike-sharing program.

The offering of bike rentals in many metropolitan areas is designed to improve the accessibility and convenience of transportation. It is vital to ensure timely access to rental bikes, reducing waiting times for the public. The consistent supply of rental bikes is a primary concern, with the expected hourly bicycle count being of paramount importance.

These bike-sharing systems bring automation to membership, rentals, and bike returns, all facilitated through an extensive network of locations. Individuals have the flexibility to rent bikes from one location and return them to either the same spot or another, providing a versatile and efficient mode of transportation. The entire process is overseen by an interconnected network of automated stores, ensuring seamless operations.

The dataset's purpose is to predict the demand for Seoul's Bike Sharing Program, leveraging historical patterns, including factors like temperature, time, and various other data points. This predictive model aims to optimize bike supply and meet the dynamic needs of the city's residents.

4. BUSINESS OBJECTIVE

Predicting the optimal number of bikes needed at any specific moment and day is a crucial and intricate business challenge. Striking the right balance is of paramount importance: too few bikes can lead to resource inefficiencies, including maintenance costs and the need for parking and security infrastructure, while an excess of bikes can result in financial losses. These losses encompass immediate revenue reduction due to a limited customer base and potential long-term impacts, such as the erosion of customer trust and loyalty.

Hence, it is imperative for bike rental enterprises to possess a robust demand estimation mechanism. This mechanism equips them to operate efficiently, maximize resource utilization, and ensure a seamless experience for their customers. Accurate demand forecasts are essential for informed decision-making and the overall success of bike-sharing businesses.

5. DATA OVERVIEW

The data set is obtained from AlmaBetter with the following column labels

1. **Date**: Date of recording
2. **Bike Count:** Number of rented bikes
3. **Hour:** Hour of the day
4. **Temp (°C):** Temperature in Celsius
5. **Humidity (%):** Relative humidity
6. **Wind Speed (m/s):** Wind speed
7. **Visibility (10m):** Visibility in meters
8. **Dew Point Temp (°C):** Dew point temperature
9. **Solar Radiation (MJ/m2):** Solar radiation
10. **Rainfall (mm):** Rainfall
11. **Snowfall (cm):** Snowfall
12. **Seasons:** Season (1 = spring, 2 = summer, 3 = fall, 4 = winter)
13. **Holiday:** Holiday indicator (0 or 1)
14. **Functioning Day:** Working day indicator (0 or 1)

In this project, we leverage a comprehensive dataset with 14 essential columns to build our predictive model. The columns we focus on include: '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', and 'Holiday'. These features form our feature set, and our ultimate goal is to predict the 'count' of rented bikes.

It's important to note that the dataset comprises 8760 observations, and it contains a wealth of information that we can use to understand and predict bike rental patterns. Additionally, the two other columns, 'casual' and 'registered', represent the components that make up the target variable 'count'. The 'datetime' column, although not included in our feature set, has been utilized as an index for the data, ensuring efficient organization and access to the information we need for our analysis.

6. EDA & VIZUALIZATION

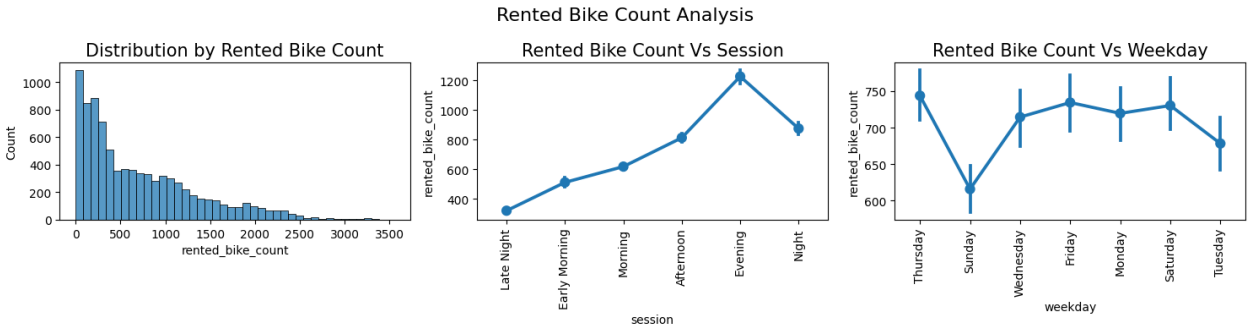
Before we dive into the modelling phase, it's essential to gain insights into how the number of bike rentals is influenced by the array of features at our disposal. To accomplish this, we'll meticulously examine each feature individually. This process involves assessing how variables such as temperature, humidity, hour of the day, weather conditions, and other factors impact bike rental counts. By scrutinizing these relationships one by one, we can uncover critical patterns and dependencies. This preliminary analysis serves as the foundation for our subsequent modelling efforts, providing us with a clearer understanding of the factors that drive bike rentals and enabling us to build more accurate predictive models.

6.1 Creating Some New Features

The hours of the day display a distinct sequential order, where 9 am lies closer to 10 am than to 8 am, and is more distant from 6 pm. This feature is best classified as a discrete ordinal variable. To investigate potential influences, we'll consider hours both categorically and numerically, aiming to discern any resulting variations in the outcomes.

- Daytime hours form a clear sequence.
- 9 am is closer to 10 am than 8 am, and farther from 6 pm.
- It's a discrete ordinal variable.
- We'll experiment by treating hours categorically and numerically.
- The goal is to evaluate the impact of this transformation on the results.

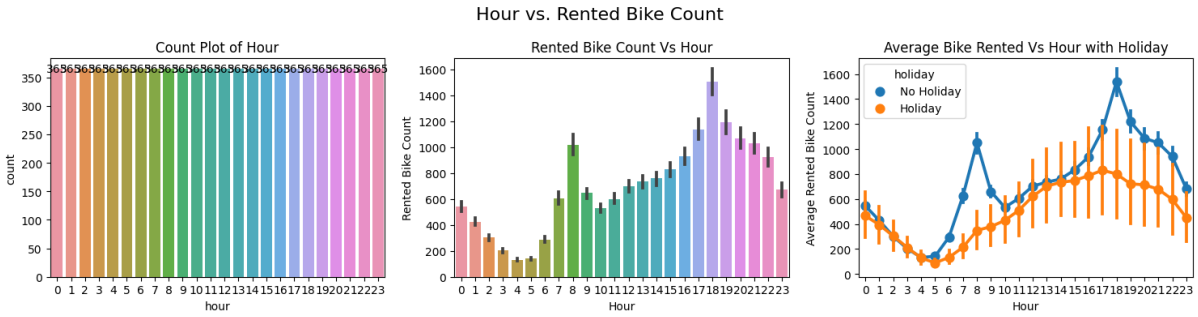
6.2 Column: rented_bike_countt



Key Insights:

- The distribution of Rented Bike Count is positively skewed, featuring numerous outliers that require attention in subsequent steps.
- Rental bike demand experiences a notable upswing during the evening hours, indicating a peak period.
- Sundays consistently record the lowest demand for rented bikes, signaling a day of reduced activity.

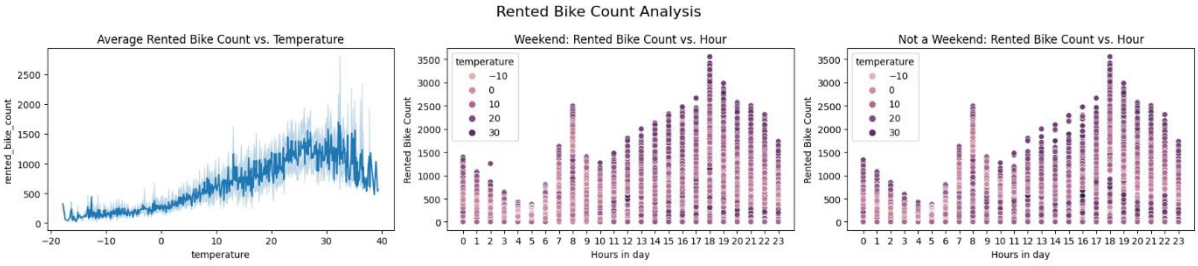
6.3 Column: 'hour'



Key Insights:

- The dataset shows an even distribution of bike counts for every hour.
- Peak demand for rented bikes occurs at 8 AM and 6 PM, indicating a surge during typical business hours.
- Rented Bike Count reveals two distinct patterns, one for holidays and the other for non-holidays.
    - Holidays: Peak rental times are around 8 AM and 6 PM, likely reflecting local commuters on workdays (Monday to Friday).
    - Non-Working Days: A different pattern emerges, with fairly consistent rentals throughout the day and a peak around noon, likely representing tourists exploring the city on weekends (Saturday and Sunday).

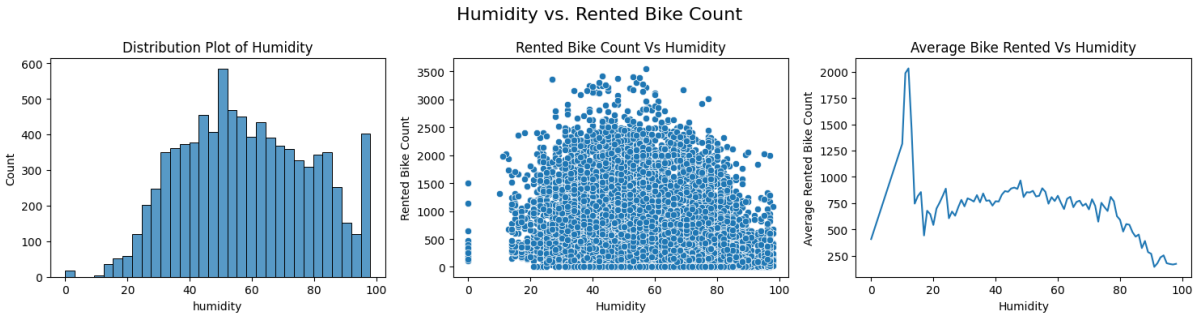
6.4 Column: 'temperature'



Key Insights:

- The line plot illustrates a steady increase in the average number of rented bikes as the temperature rises, suggesting a positive correlation.
- People tend to favor renting bikes in warmer conditions, indicating higher demand when the temperature is comfortably warm.
- Notably, there is a slight dip in rental counts during extremely hot temperatures, possibly due to discomfort associated with such conditions.

6.5 Column: 'humidity'

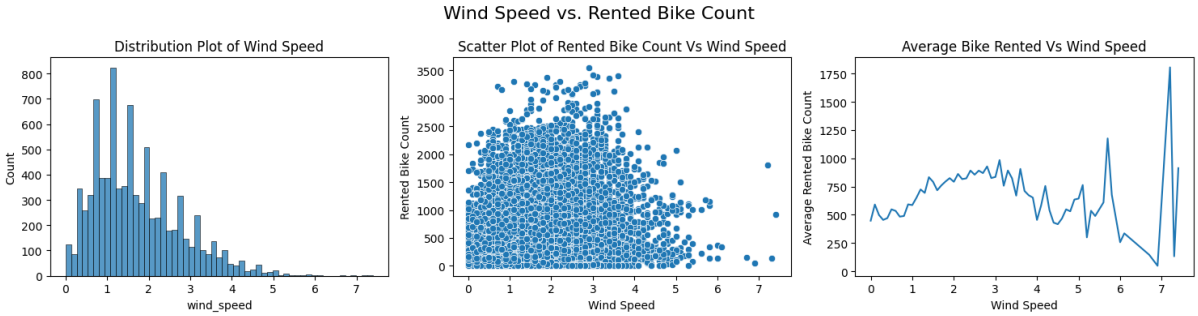


Key Insights:

The rental bike count exhibits significant fluctuations, with a peak around 50.

The most favourable humidity range for bike rentals falls between 20 and 90, indicating that customers prefer conditions within this range.

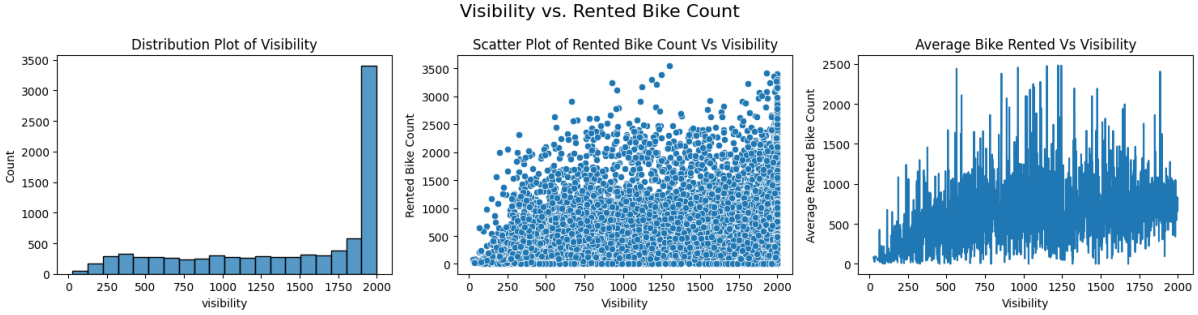
6.6 Column: 'wind_speed'



Key Insights:

- Wind speed follows a slightly positive skew in its distribution.
- The scatter plot visually represents the relationship between wind speed and the rented bike count.
- The line plot indicates that while moderate wind speeds are generally preferred, there is a notable increase in demand at a wind speed of 7.

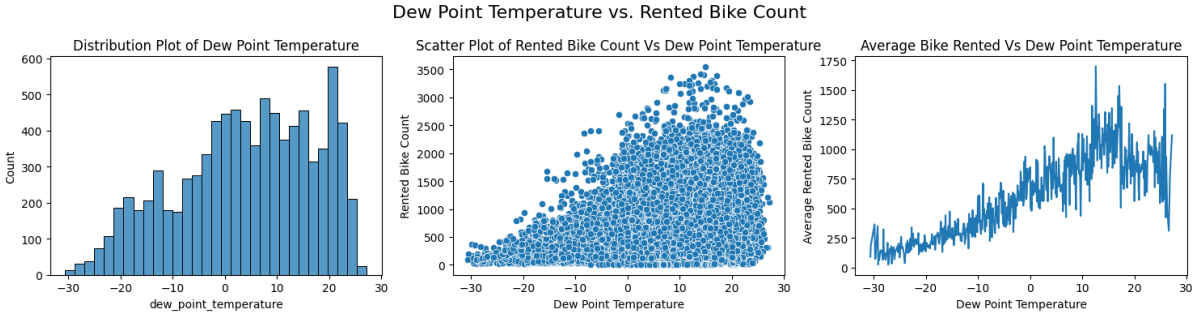
6.7 Column: 'visibility'



Key Insights:

- Visibility is left-skewed in its distribution.
- The plots demonstrate a positive relationship between visibility and the number of rented bikes – as visibility increases, so does the rented bike count.

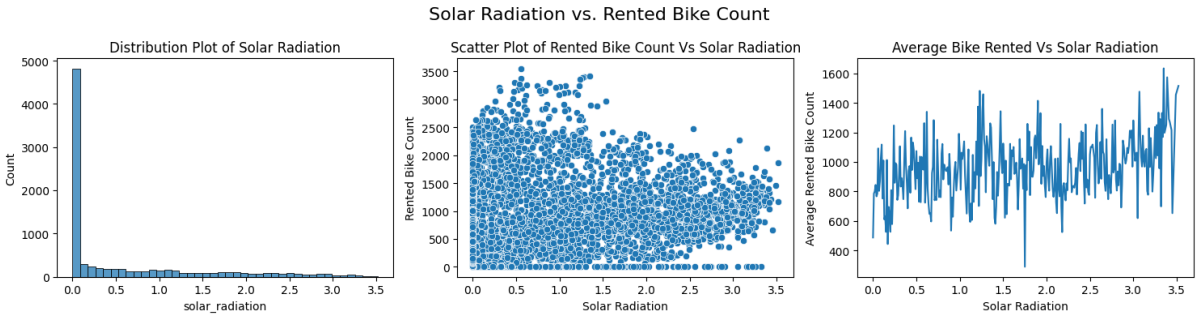
6.8 Column: 'dew_point_temperature'



Key Insights:

- Dew point temperature exhibits negative skewness and a positive correlation with the rented bike count.
- The average number of rented bikes generally increases as the dew point temperature rises, except for a slight decrease at the highest dew point temperatures.
- The line plot for dew point temperature follows a pattern similar to that of the temperature variable.

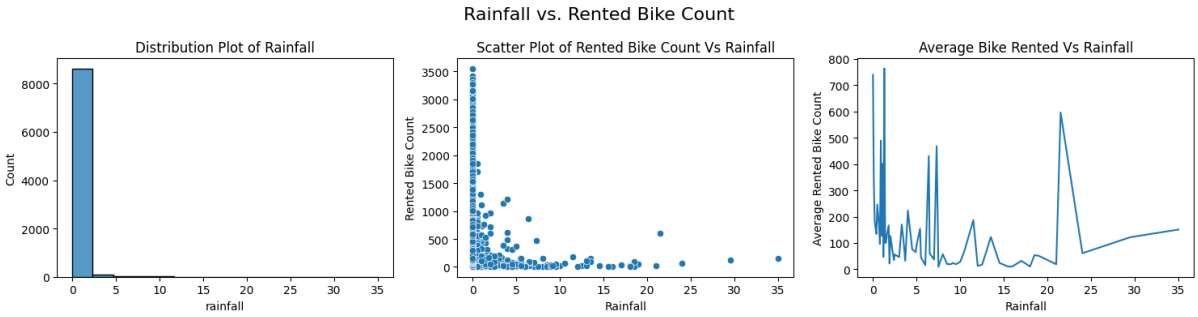
6.9 Column: 'solar_radiation'



Key Insights:

- Solar radiation shows a strong positive skew in its distribution.
- The average number of rented bikes increases as solar radiation levels rise.
- There is a direct relationship between solar radiation and temperature.

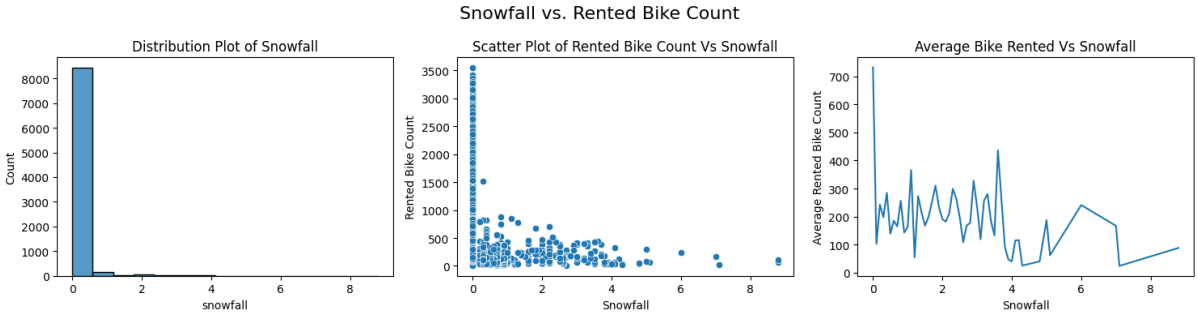
6.10 Column: 'rainfall'



Key Insights:

- The distribution of rainfall is highly positively skewed.
- Users show a clear preference for days with little to no rainfall, indicating reduced bike rentals on rainy days.

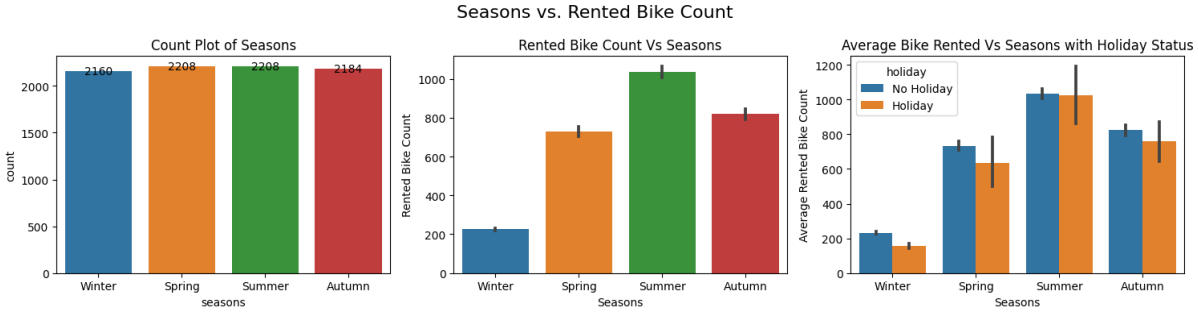
6.11 Column: 'snowfall'



Key Insights:

- The distribution of snowfall is highly positively skewed.
- Users exhibit a clear preference for days with minimal or no snowfall, suggesting reduced bike rentals during snowy conditions.

6.12 Column: 'seasons'

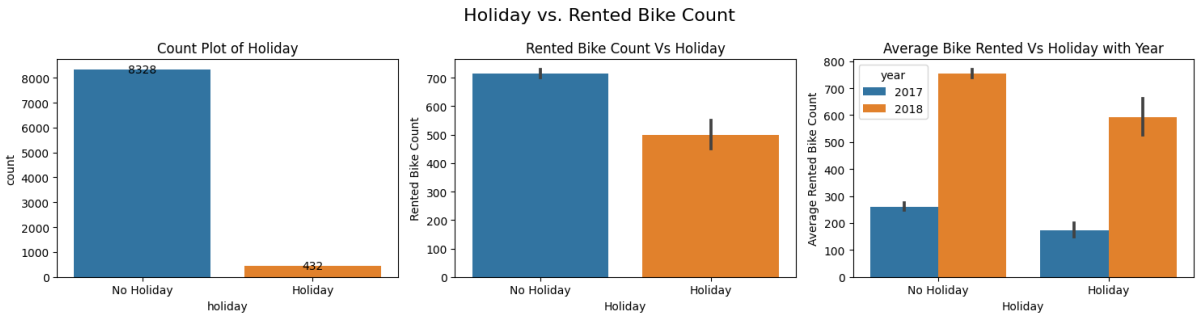


Key Insights:

- The dataset includes data for all four seasons, with each season having over 2,000 counts.

- Summer stands out as the most popular season for bike rentals, while winter is the least favored, indicating a strong preference for renting bikes in warmer weather.
- Across all seasons, non-holiday days consistently have higher bike rental counts compared to holiday days, suggesting that people are more likely to rent bikes for daily commuting or leisure on non-holidays.

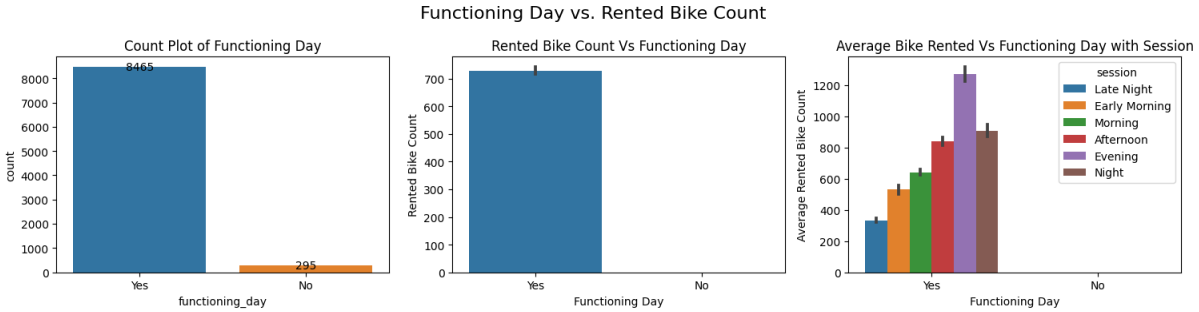
6.13 Column: 'holiday'



Key Insights:

- The dataset contains a larger number of records for non-holiday days compared to holiday days. This is expected, as most days are working days.
- On non-holiday days, there is a higher demand for bike sharing, suggesting that these rentals are primarily related to work or daily commuting.
- The dataset also includes more records from the year 2018 than from 2017, indicating an increase in the usage of the bike-sharing service over time.

6.14 Column: 'functioning_day'



Key Insights:

- The dataset predominantly consists of records for functioning days, which aligns with the fact that most days are working days.
- Despite having some records for non-functioning days, there are no bike rentals during such days, indicating that the bike-sharing service is primarily utilized on functioning days.
- On functioning days, the evening hours experience the highest demand for rented bikes, likely due to commuters using bikes for their daily work commutes.

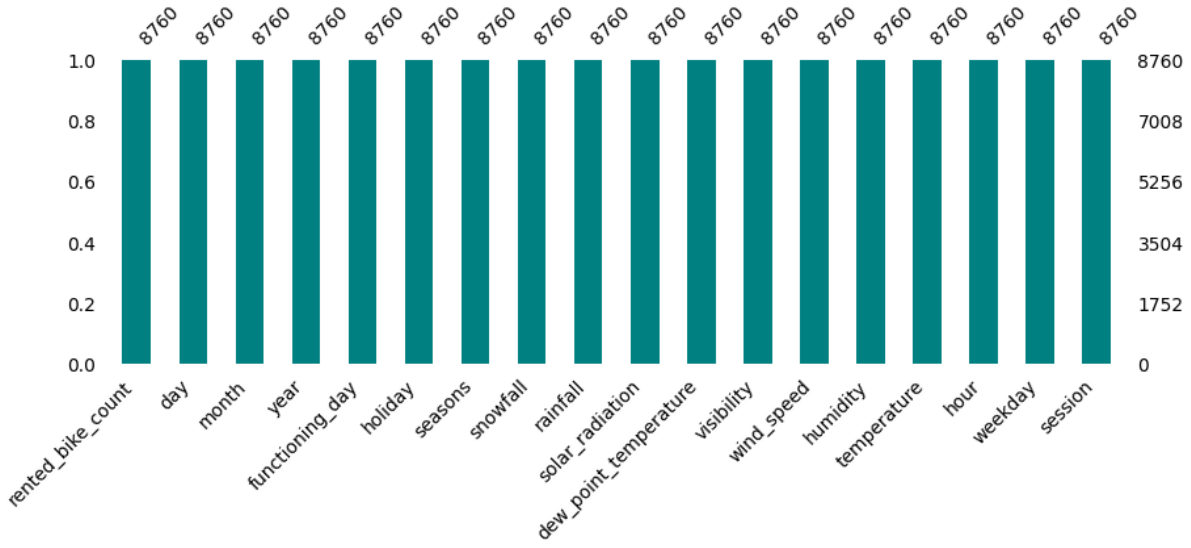
7. DATA CLEANING

Data cleaning is a critical and indispensable step in the data preparation process. Its primary role is to identify and correct inaccuracies, inconsistencies, and missing data within a dataset. Situated at the core of data quality assurance, this phase is responsible for ensuring that the data is not only accurate but also complete, making it suitable for in-depth analysis. Data cleaning activities encompass tasks such as removing duplicate records, filling in missing values, rectifying errors, and standardizing data formats. The ultimate objective of data cleaning is to elevate data quality, making it ready for rigorous analytical exploration and modelling, thus ensuring the reliability and validity of any subsequent insights or predictions.

7.1 Duplicate Values

The code `bike_df.duplicated().sum()` is used to count the number of duplicated rows in a DataFrame named `bike_df`. The result, with an output of 0, indicates that there are no duplicate rows in the DataFrame, meaning that all rows in the DataFrame are unique with no identical values across all columns.

7.2 Missing Value

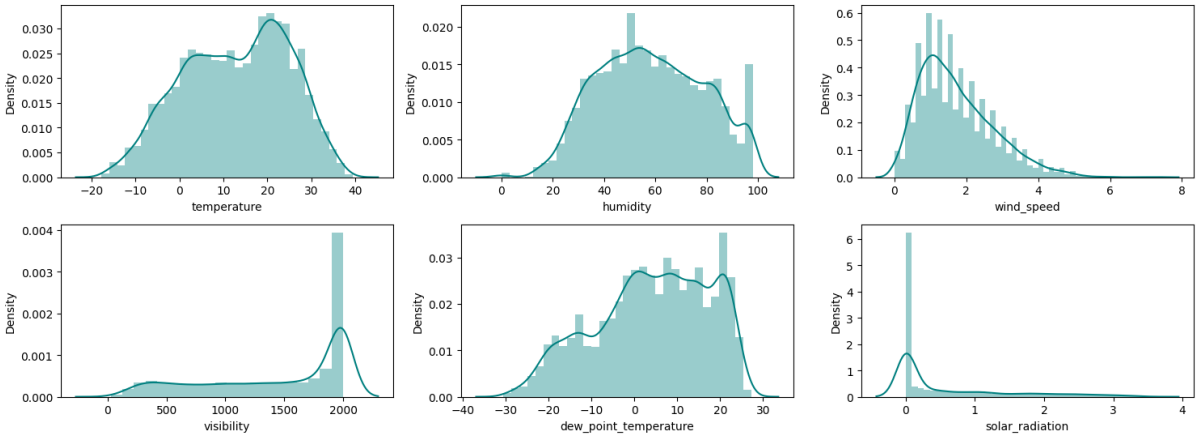


Fortunately, the dataset is free from any missing values, ensuring that all data points are complete and available for analysis without the need for imputation or data recovery procedures. This data completeness simplifies the analytical process and enhances the dataset's reliability.

7.3 Skewness

Upon inspecting the statistical summary of the numerical attributes, a conspicuous divergence between the 75th percentile and the maximum value becomes evident. This divergence serves as an indicator of skewness and the presence of outliers within the dataset, warranting attention.

Data Distribution of Numerical Features



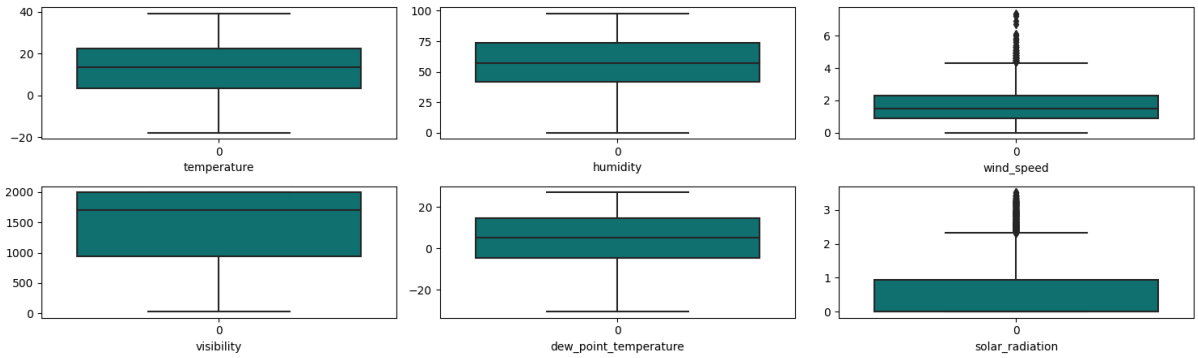
Key Insights:

When examining the numerical attributes, it becomes evident that the prevailing distributions are skewed to the right. Particularly, the distributions of rainfall, snowfall, and solar radiation exhibit pronounced right-skewness, indicating the presence of numerous outliers. Conversely, some variables display negative skewness.

Furthermore, it is worth noting that the removal of outliers from certain columns can potentially render their distributions more normal. Consequently, it appears imperative to address outliers prior to considering transformation techniques. The established approach involves the initial removal of outliers, followed by an assessment of the necessity for further transformation.

7.4 Treating Outliers

Outlier Analysis of Numerical Features

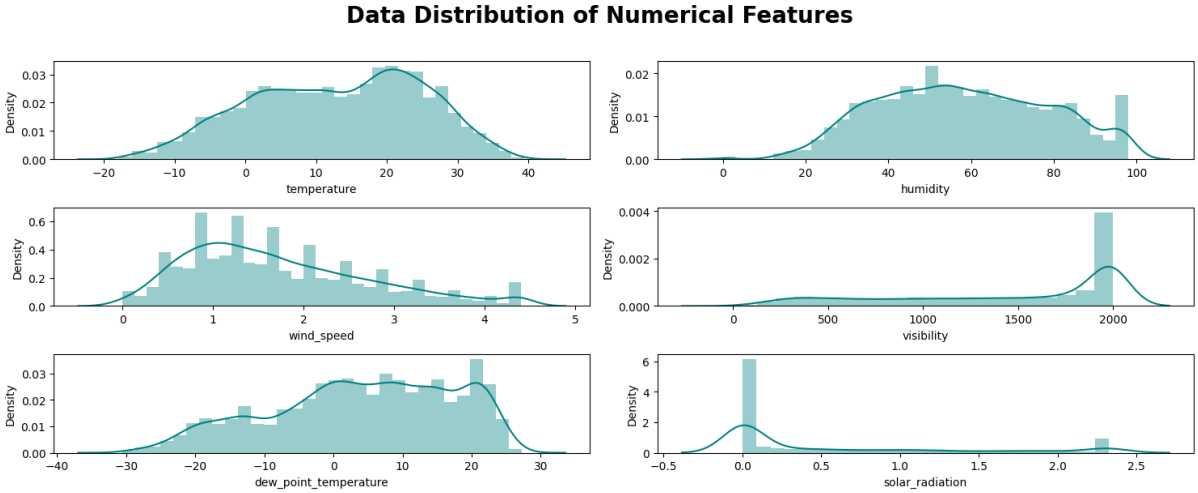


Key Insights:

- The presence of outliers is noticeable in the 'wind_speed' and 'solar_radiation' columns.
- Due to the relatively limited dataset size, a cautious approach is taken regarding outlier handling. Instead of outright removal, the clipping method is employed.

Clipping Method:

The application of the clipping method involves establishing upper and lower thresholds to identify outliers. When a value exceeds these thresholds, it is considered an outlier and is consequently replaced with either the maximum or minimum value within the defined range. This method allows for a more controlled approach to handling outliers without losing valuable data.



Key Insights:

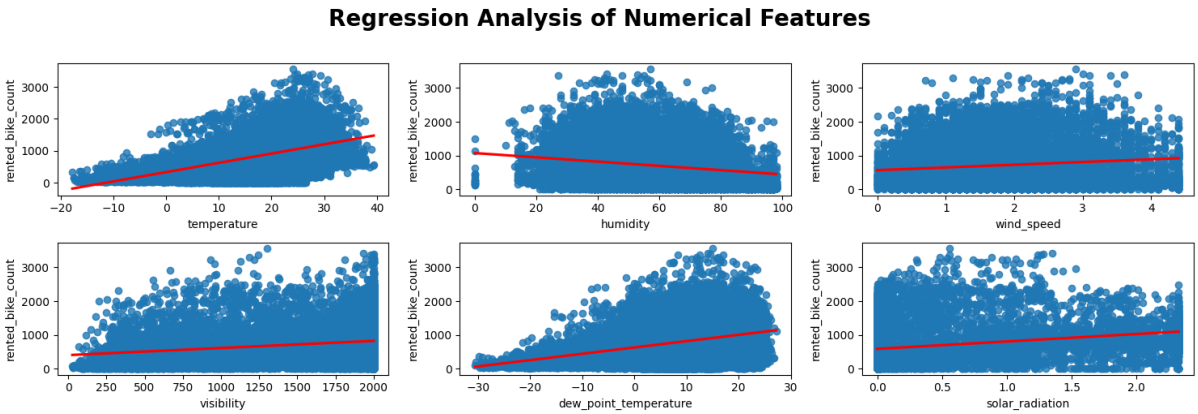
- The treatment of outliers has induced noticeable shifts in the data distribution.
- Several features exhibited significant skewness before the outlier treatment, but post-treatment, they align more closely with a normal distribution.
- Consequently, there is no current need for employing numerical feature transformation techniques, as the data is now more amenable to analysis and modelling.

8. FEATURE ENGINEERING & DATA PRE-PROCESSING

Feature engineering is a pivotal phase in the development of machine learning models. It revolves around the strategic manipulation of data to create new features or to carefully select and reformat existing ones. The objective is to empower models with the ability to discern patterns, relationships, and meaningful information within the data. By crafting new features or refining existing ones, the raw data is transformed into a more informative and interpretable format, aligning it with the model's requirements.

The success of a machine learning model is profoundly impacted by the quality and relevance of the features it uses as input. Effective feature engineering not only optimizes a model's performance but also accelerates the learning process. It can uncover latent patterns, reduce noise, and enhance the model's ability to make accurate predictions or classifications. In essence, feature engineering is akin to providing the model with the right set of tools to extract meaningful insights from the data, ultimately serving as a cornerstone for successful machine learning endeavours.

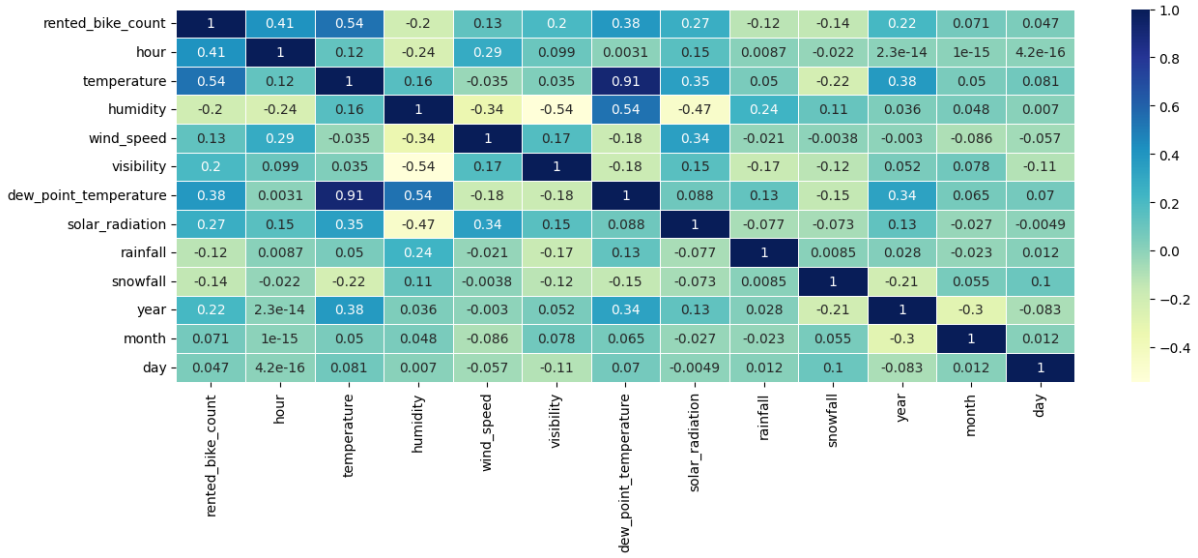
8.1 Regression plot



The majority of numerical features exhibit a positive correlation with the target variable, indicating that as these features increase or decrease, the target variable tends to follow a corresponding trend, highlighting their potential significance in influencing the target's behaviour.

8.2 Correlation Coefficient and Heatmap

The correlation coefficient is a statistical measure that gauges the intensity and direction of a linear connection between two variables. Ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 denoting no correlation, it plays a pivotal role in identifying associations between variables. Correlation analysis facilitates effective feature selection by detecting and eliminating highly correlated features, mitigating overfitting, and enhancing model performance by retaining the most informative and independent variables for analysis and prediction.



Decision Explanation:

- Due to the high correlation coefficient (0.91) between dew_point_temperature and temperature, we have chosen to drop dew_point_temperature.
- Dew_point_temperature shows less correlation with our target variable, making it a less relevant feature for our predictive model.
- This decision simplifies the model and removes potential multicollinearity, which can improve model interpretability and performance.

8.3 VIF (Variance Inflation Factor) Analysis Explanation:

- VIF analysis is a statistical technique used to detect multicollinearity within a set of predictor variables in a regression model.
- Multicollinearity occurs when two or more predictors are highly correlated, leading to redundant information in the model.
- This analysis is crucial for building reliable and interpretable regression models by identifying and addressing multicollinearity.
- VIF analysis helps to ensure that each predictor variable contributes unique information to the model, improving its overall quality and accuracy.

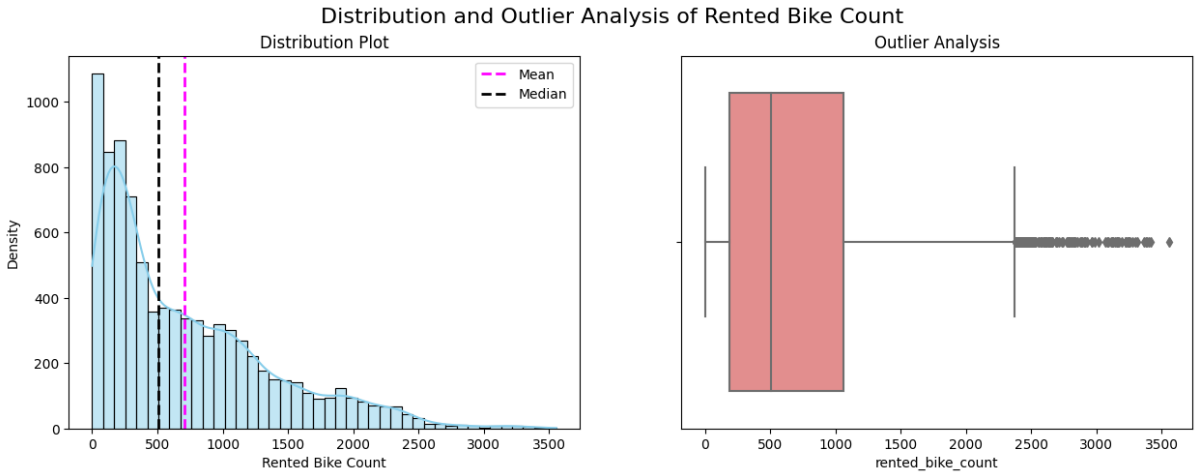
8.4 Encoding in Feature Engineering:

- Encoding is a fundamental technique in feature engineering that transforms categorical variables into numerical values suitable for machine learning algorithms.
- Various encoding methods are available, each with its unique approach:
  1. One-hot encoding: Generates binary columns for each category, indicating its presence or absence.
  2. Label encoding: Assigns a unique integer to each category.
  3. Ordinal encoding: Assigns ordered integers based on the natural order of categories.
  4. Count encoding: Replaces categories with their occurrence frequency in the dataset.

- The choice of encoding method depends on the nature of the categorical data and the requirements of the machine learning model.

8.5 Normalization of Target Variable

- Data normalization, a vital aspect of data mining, involves the transformation of source data into a format suitable for efficient processing.
- The core objective of data normalization is to minimize or remove duplicate data, streamlining it for further analysis and modelling.
- By standardizing data, data normalization enhances the quality and consistency of information, making it more valuable for decision-making.



Key Insights:

- The distribution of Rented Bike Count exhibits a moderate right skew, which doesn't align with the normal distribution assumption of linear regression. To address this, we should take steps to normalize the distribution.
- A boxplot analysis reveals the presence of outliers in the Rented Bike Count column, which may impact the model's performance.

9. ML Model Implementation

The "train-test split" is a fundamental practice in machine learning. It involves dividing a dataset into two subsets: the "training set" used to train a machine learning model and the "test set" used to evaluate its performance. This split allows for assessing how well the model generalizes to unseen data. Typically, a large portion of the data is allocated to training (e.g., 70-80%), while the remainder is reserved for testing. By evaluating the model on data it hasn't seen during training, we can estimate its ability to make accurate predictions on new, real-world data.

9.1 Train Test Split

The separation of the target variable and the feature matrix is performed. The dataset `bike_df` is split into two components:

1. `X` is the feature matrix, which contains all the input features for your machine learning model, excluding the target variable 'rented_bike_count'. It's essential for building and training the model, and it's denoted as `X`.

2. `y` represents the transformed target variable. In this case, the square root of 'rented_bike_count' is used as the transformed target variable. This transformation can be useful to make the target variable more suitable for modeling, especially when its distribution is skewed or has extreme values.

With `X` containing the feature matrix and `y` containing the transformed target variable, you're now equipped to proceed with your machine learning tasks, such as model training and evaluation.

9.2 Scaling Data

- Mean Squared Error (MSE): This metric quantifies how close a regression model's predictions are to the actual data points. It measures the average of the squared differences between predicted and actual values.
- Root Mean Square Error (RMSE): RMSE is a variation of MSE, but it provides the standard deviation of prediction errors, offering a more interpretable measure of model performance.
- Mean Absolute Error (MAE): MAE calculates the average of the absolute differences between predicted and actual values, providing insight into the model's accuracy in predicting numerical values.
- R-squared (R2): R2 is a statistical metric that signifies the proportion of variance in the dependent variable explained by the independent variables in a regression model. It ranges from 0 to 1, with higher values indicating better model fit.Adjusted R-squared: Adjusted R2 is a modification of R2, accounting for the number of predictors in the model. It helps prevent overfitting by adjusting the goodness-of-fit for the model's complexity.

These evaluation metrics are essential for assessing the performance and accuracy of regression models, enabling data scientists and analysts to make informed decisions about model selection and refinement.

9.3 Model Training

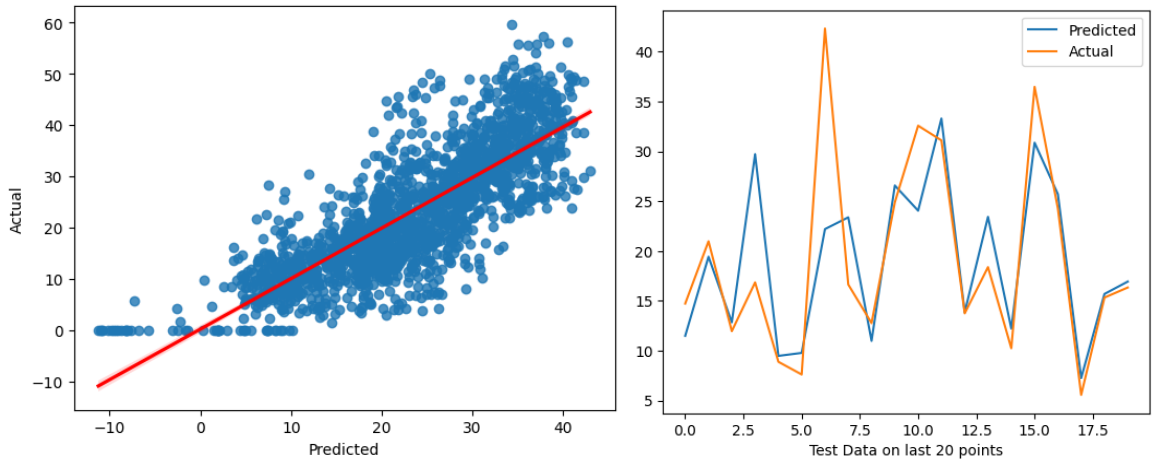
This code defines a function named `predict` that assesses the performance of a machine learning model. It takes the machine learning model and its name as inputs. The function trains the model, makes predictions on both the training and testing datasets, and calculates various evaluation metrics. It also reverses a square root transformation on the predictions, plots actual vs. predicted values, and stores key metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R2), and more in the `model_result` list. This function aids in comparing and assessing different machine learning models. It provides insights into the model's effectiveness and stores the results for analysis.

10. MODEL IMPLEMENTATION

10.1 Linear Regression

Linear regression is a powerful statistical technique employed to establish a connection between a target variable and one or more predictor variables. It achieves this by determining the optimal linear equation that can best forecast the target variable's value, considering the predictor variables. Here are the key points:

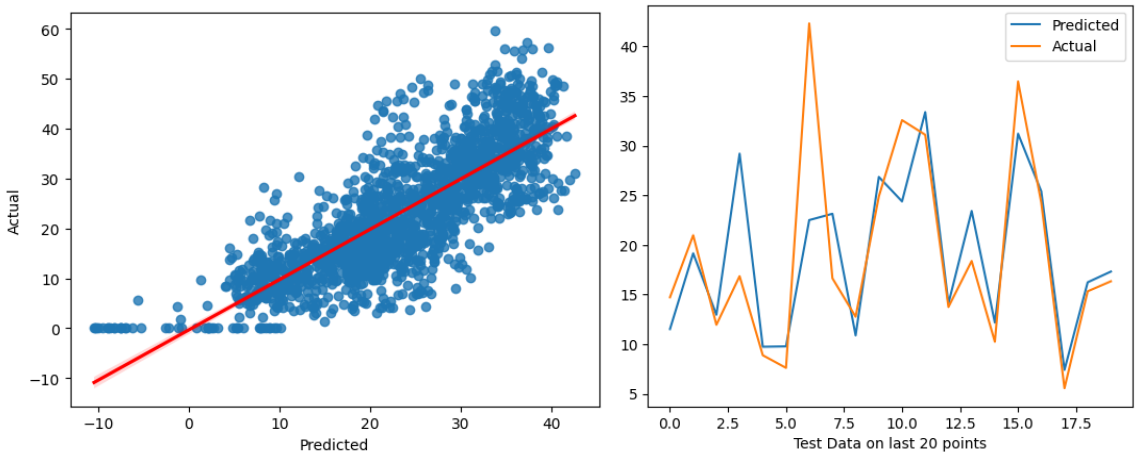
- Linear regression seeks to identify the most suitable line that captures the relationship between the dependent variable and one or more independent variables.
- It's a fundamental tool for predicting outcomes, making it valuable in various fields, from finance to science, and it provides a foundation for more advanced modeling techniques.



10.2 Lasso

Lasso, which stands for Least Absolute Shrinkage and Selection Operator, is a regularization method applied in linear regression models. Its primary purpose is to simplify the model and enhance its ability to generalize by penalizing the size of feature coefficients.

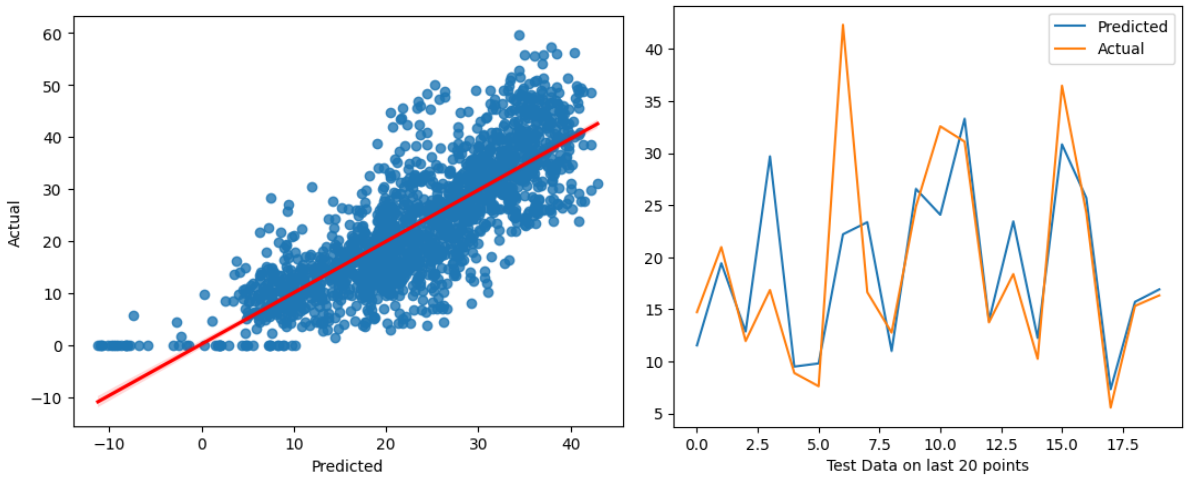
- Lasso introduces a penalty term in the optimization loss function that's related to the absolute values of the coefficients. This addition helps control the magnitude of feature coefficients.
- Unlike ridge regression, lasso has the unique property of driving some feature coefficients to exactly zero. In other words, it not only shrinks coefficients but can exclude certain features altogether.
- Lasso is a valuable tool for feature selection, making it beneficial in scenarios where you want to identify the most important predictors while reducing model complexity.



10.3 Ridge

Ridge Regression, a form of regularized linear regression, addresses multicollinearity and overfitting by introducing a penalty term to the loss function. This penalty term is the L2 regularization, also called the weight decay term, which imposes a penalty based on the square of the coefficient magnitudes.

- Ridge Regression is a valuable technique in linear regression to combat multicollinearity, where predictor variables are highly correlated, and overfitting, where the model captures noise in the data.
- The L2 regularization term in Ridge Regression encourages the model to have smaller but non-zero coefficients for all features, unlike Lasso, which can drive some coefficients to exactly zero.
- By applying Ridge Regression, one can strike a balance between the fit of the model and its complexity, resulting in more stable and reliable predictions.

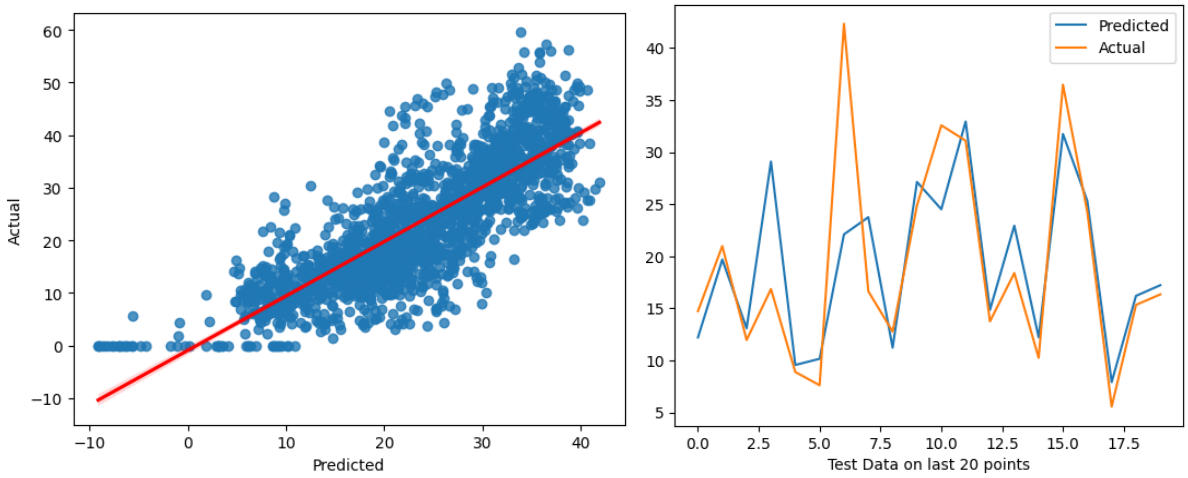


10.4 Elastic Net

ElasticNet, a linear regression method, harnesses the strengths of both L1 (Lasso) and L2 (Ridge) regularization techniques. These regularization methods prevent overfitting by introducing penalty terms into the loss function the model seeks to minimize. Lasso

penalizes coefficients based on their absolute values, while Ridge penalizes coefficients based on their squares.

- ElasticNet offers a balanced approach by simultaneously applying L1 and L2 regularization. This helps in feature selection and reduces multicollinearity, enhancing the model's generalization capability.
- Lasso tends to force some coefficients to zero, effectively performing feature selection, while Ridge keeps all features but with reduced influence, allowing ElasticNet to inherit these characteristics.
- ElasticNet is a versatile tool for tackling complex datasets where both feature selection and regularization are necessary.



10.5 K-Nearest Neighbors

K-nearest neighbor (KNN) is a supervised machine learning algorithm employed for both classification and regression tasks. KNN is a non-parametric method, which means it doesn't rely on underlying data assumptions. It categorizes an input dataset by finding the most similar data points from its neighbors. The measure of similarity is determined by various distance metrics, including Euclidean Distance, Manhattan Distance, Minkowski, and Cosine Similarity Measure.

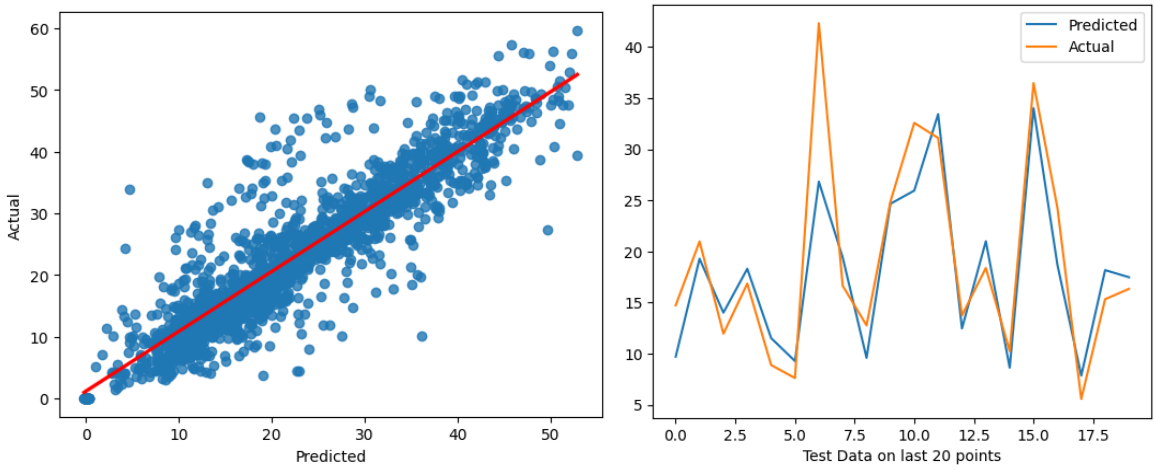
- KNN makes predictions based on the majority class of its K-nearest neighbors in classification and the average of their values in regression.
- It's a versatile algorithm suitable for a wide range of data types and can adapt to various distance metrics.
- The choice of K (the number of neighbors to consider) and the distance metric significantly impact the algorithm's performance and should be carefully selected based on the dataset and problem.

10.6 Support Vector Machine

The Support Vector Machine (SVM) is a robust machine learning algorithm used for both classification and regression tasks. Its fundamental idea involves identifying the optimal

hyperplane that effectively separates classes in classification problems or predicts target values in regression.

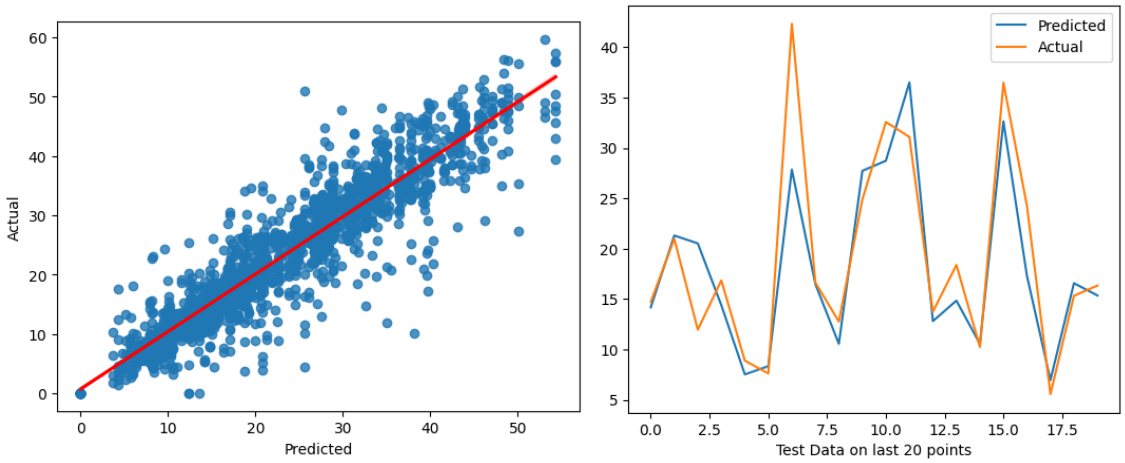
- SVM is known for its effectiveness in handling high-dimensional data, as it can find complex decision boundaries.
- The choice of the kernel function (e.g., 'rbf' for Radial Basis Function) significantly impacts SVM's performance.
- The regularization parameter (C) influences the trade-off between maximizing the margin and minimizing classification errors in SVM.
- SVM's ability to handle both linear and non-linear data makes it a versatile choice for various machine learning applications.



10.7 Decision Tree

A decision tree is a fundamental machine learning model used for prediction and decision-making. It operates by recursively applying rules and conditions to input features, breaking down complex data into smaller, more manageable segments.

- Each node in the tree represents a feature test, while branches depict the outcomes of these tests.
- The leaves, located at the end of the branches, contain the final class predictions or decisions.
- Decision tree construction involves selecting the best features to split data, optimizing criteria like information gain or impurity reduction.
- In the given example, the DecisionTreeRegressor is customized with parameters like minimum samples per leaf, minimum samples for splitting, maximum depth, and random state, allowing users to fine-tune the model for specific needs.



10.8 Random Forest

Random Forest, an ensemble learning algorithm, stands out by constructing numerous decision trees and aggregating their outputs to arrive at a final prediction. It's a departure from single decision trees, as it mitigates the risk of overfitting by blending the outcomes of multiple trees, each trained on a distinct subset of the data.

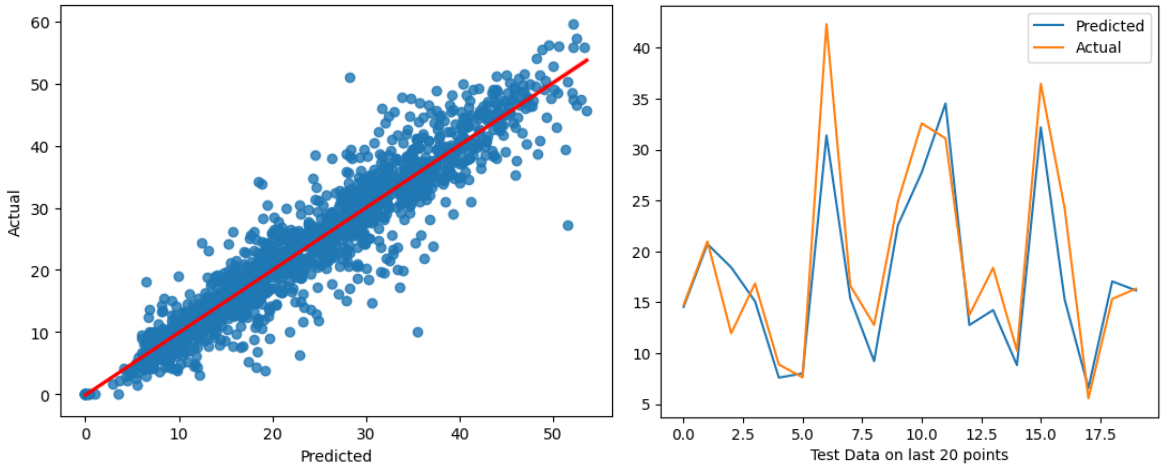
- Ensemble approach: Random Forest leverages the wisdom of many trees, drawing on their collective insights.
- Overfitting prevention: By using diverse data subsets and combining results, it reduces the likelihood of overfitting, where a model fits the training data too closely and performs poorly on new data.
- Robust and versatile: This method is widely appreciated for its robustness, ability to handle high-dimensional data, and suitability for both classification and regression tasks.

Hyperparameter Tunning using GridSearchCV

In this code snippet, a hyperparameter tuning process is performed using scikit-learn's `GridSearchCV` with a Random Forest Regressor model. The code defines a parameter grid, which includes various hyperparameters such as the number of estimators (trees) in the ensemble, maximum tree depth, minimum samples required to split a node, and minimum samples in a leaf node.

A Random Forest Regressor model is created, and the `GridSearchCV` object is initialized with 5-fold cross-validation (cv=5) and R-squared (R2) as the scoring metric. The goal is to identify the optimal combination of hyperparameters that maximizes the model's performance in predicting the target variable.

The `grid_search` object is then fitted to the provided data (X, y), conducting an exhaustive search over the hyperparameter space to determine the best hyperparameter settings for the Random Forest Regressor.

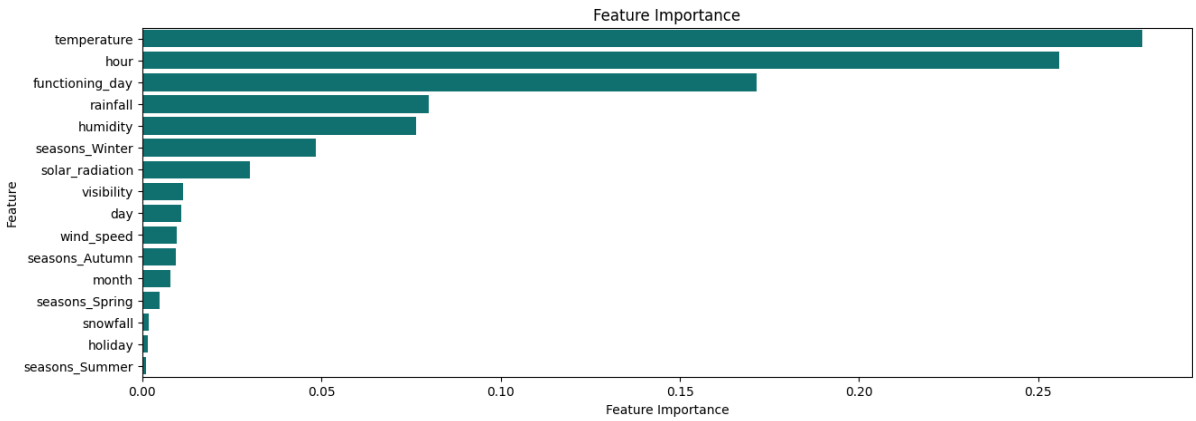


In the Random Forest model, the top 5 influential features are temperature, hour, functioning_day, rainfall, and humidity. These variables play a significant role in determining the predictions made by the model and are considered the most important factors in the analysis.

LightGBM is a powerful gradient boosting framework known for its efficiency, especially with large datasets. Developed by Microsoft, it outperforms traditional gradient boosting algorithms. LightGBM employs a unique histogram-based method to split nodes in decision trees, enhancing its speed and performance.

Model Explainability

In the Random Forest model, the top 5 influential features are temperature, hour, functioning_day, rainfall, and humidity. These variables play a significant role in determining the predictions made by the model and are considered the most important factors in the analysis.



10.9 Light GBM

LightGBM is an open-source gradient boosting framework created by Microsoft. It stands out for its efficiency, making it an ideal choice for dealing with large datasets. This framework is particularly well-suited for tasks that require high-performance machine learning.

A key distinguishing feature of LightGBM is its innovative use of a histogram-based approach for splitting nodes in decision trees. This technique results in faster and more memory-efficient training of decision trees compared to traditional gradient boosting algorithms, such as those used in libraries like XGBoost.

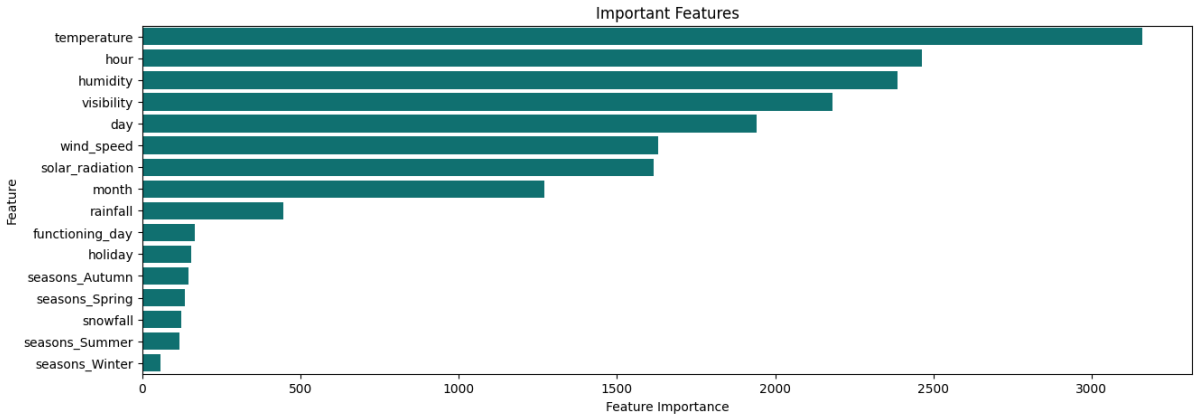
In summary, LightGBM offers a highly efficient and scalable solution for gradient boosting, thanks to its histogram-based node splitting and optimization techniques. This makes it a valuable tool for tackling large-scale machine learning problems.

HyperParameter Tunning using GridSearchCV

The provided code snippet demonstrates the process of hyperparameter tuning for an XGBoost (Extreme Gradient Boosting) Regressor model using scikit-learn's GridSearchCV. The hyperparameters under consideration include the number of trees in the ensemble (n_estimators), maximum tree depth (max_depth), minimum samples required to split a node (min_samples_split), and minimum samples in a leaf node (min_samples_leaf). The model is trained and evaluated using 5-fold cross-validation, and the evaluation metric is the R-squared (R2) score.

The best combination of hyperparameters to optimize the XGBoost Regressor's performance on a given dataset.

Model Explainability



In the XGBoost (Extreme Gradient Boosting) model, the top 5 important features, in order of importance, are:

1. Temperature: This feature has the highest importance in making predictions. It indicates that temperature plays a significant role in the model's decision-making process.
2. Humidity: Humidity is the second most important feature. It suggests that the level of humidity in the environment has a strong influence on the model's predictions.
3. Visibility: Visibility is the third most important feature. It implies that the extent to which objects are visible in the environment is a key factor in the model's decision-making.

4. Hour: The hour of the day is the fourth most important feature. This indicates that time of day plays a role in the model's predictions, which is common in time series or temporal data.
5. Day: The day of the week or month is the fifth most important feature. It suggests that the specific day can impact the model's predictions, which is relevant in scenarios where daily or monthly patterns are important.

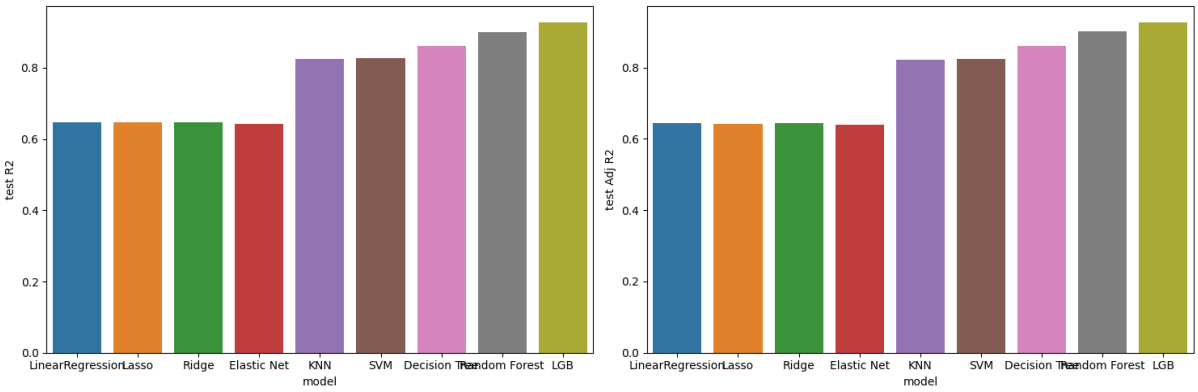
In summary, these features have been identified as the most influential in the XGBoost model, with temperature, humidity, visibility, hour, and day being the top factors contributing to its predictive accuracy.

10.10 Model Result

The model's goodness of fit can be assessed using the R-squared measure, which quantifies the relationship between dependent and independent variables. However, R-squared does not address overfitting concerns. In cases where regression models have numerous independent variables, they might perform well during training but poorly during testing due to excessive complexity. Adjusted R-squared is a valuable metric that not only considers the relationships between variables but also penalizes the addition of unnecessary independent variables, mitigating overfitting risks.

R-squared remains a reliable method for understanding the rented_bike_count's predictability, as it captures the link between this variable and the independent ones.

	model	train MSE	test MSE	train RMSE	test RMSE	train MAE	test MAE	train R2	test R2	train Adj R2	test Adj R2
8	LGB	0.000	12.645	0.000	3.556	0.002	2.340	1.000	0.916	1.000	0.915
7	Random Forest	8.112	15.330	2.848	3.915	1.921	2.691	0.948	0.898	0.948	0.897
6	Decision Tree	13.689	20.914	3.700	4.573	2.542	3.108	0.912	0.861	0.912	0.860
5	SVM	20.542	26.094	4.532	5.108	2.685	3.326	0.868	0.827	0.868	0.825
4	KNN	12.718	26.575	3.566	5.155	2.361	3.577	0.918	0.824	0.918	0.822
0	LinearRegression	54.120	53.132	7.357	7.289	5.639	5.622	0.653	0.648	0.652	0.645
2	Ridge	54.119	53.108	7.357	7.288	5.639	5.621	0.653	0.648	0.652	0.645
1	Lasso	54.273	53.316	7.367	7.302	5.648	5.636	0.652	0.647	0.651	0.643
3	Elastic Net	54.593	53.911	7.389	7.342	5.677	5.680	0.650	0.643	0.649	0.639



The analysis suggests that either XGB or LGB Regressor can serve as the final model. These models exhibit the lowest RMSE values and the highest R2 scores on the test data. Opting for

LGB is favored due to its reputation for excelling with large datasets and its ability to provide a more insightful interpretation of the features it considers.

11. CONCLUSION:

In this project, we addressed a critical regression problem, predicting bike sharing counts, which plays a pivotal role in enhancing urban mobility and public convenience. The increasing popularity of bike rentals emphasizes the need for timely and abundant access to rental bikes to minimize wait times, making the availability of bikes a key factor.

Our journey commenced with a comprehensive exploratory data analysis (EDA). We scrutinized the dependent variable, "Rental Bike Count," delving into both categorical and numerical variables. Through this analysis, we uncovered their correlations, distributions, and connections with the dependent variable. We also performed categorical variable one-hot encoding and judiciously removed numerical features with multicollinearity, which were primarily used for EDA purposes. Subsequently, we ventured into the realm of various models, ranging from elementary Linear Regression and Regularization Models like Ridge, Lasso, and Elastic Net to more sophisticated ensemble models, including Random Forest, Gradient Boosting, and Light GBM. Our journey culminated in model hyperparameter tuning to enhance predictive performance.

Notably, our analysis unveiled the patterns and insights crucial for efficient bike sharing demand management. It was evident that the majority of rentals are attributed to daily commutes to workplaces and educational institutions, with peak rental hours observed during the morning and evening rush, from 7–9 am and 5–6 pm. We discerned two distinct rental patterns based on working and non-working days, reflecting the peaks during office hours and a more consistent demand with a noon peak, respectively. The hour of the day emerged as a significant driver of bike rental counts, aligning with the daily routines of users.

Moreover, temperature was a vital factor influencing rental counts, with the highest demand observed between 32 to 36 degrees Celsius. Seasonal trends were evident, with Spring and Summer recording the highest rentals, while Winter witnessed lower counts. Unsurprisingly, weather conditions played a key role, with clear days attracting the highest number of rentals, while snowy or rainy days witnessed reduced demand. Lastly, we noted that increasing humidity correlated with decreased bike rental counts.

Ultimately, our model selection favored the Light GBM, prioritizing predictive accuracy. We rigorously tested and compared various models, including linear models, decision trees, Random Forests, and Gradient Boosting. The choice was informed by the performance metric R2. While our training R2 score indicated a remarkable 99%, the test R2 score, due to the dataset's limited size, reached 92.5%. We anticipate that with more data, our algorithm can be retrained to further improve its predictive prowess.