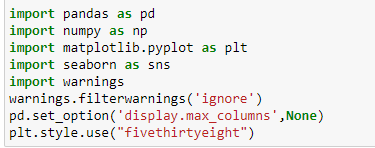
**Bike Sharing Demand Analysis & Model**

# Created by Kaushik Kar

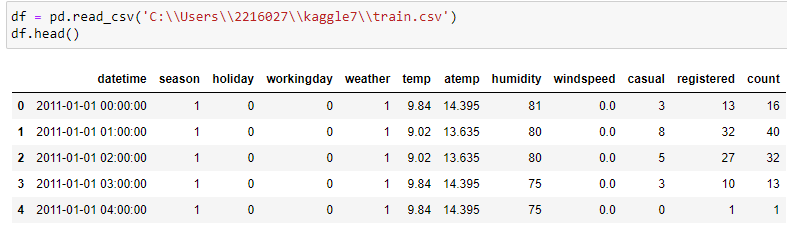
# Employment id- 2216027

# Importing Libraries:

First, importing the important external Python packages using the pip package manager.



1. NumPy is used for mathematical operations like addition, subtraction, multiplication, division, etc. on arrays and matrices.
2. Pandas provides data structures for efficiently storing and manipulating large datasets, and tools for reading and writing data to and from various file formats, including CSV, Excel, and SQL databases
3. Seaborn is a data visualization library based on Matplotlib which is a plotting library used for creating static, interactive, and animated visualizations in Python.
4. Imports the warnings module uses its filterwarnings() function to ignore all warnings that may be raised in the code.
5. Style in matplotlib to "fivethirtyeight", which is a pre-defined style that mimics the visual style of the website FiveThirtyEight. This style is known for its bold colors, thick lines, and high contrast, which make it particularly suitable for data visualizations that need to be eye-catching and easy to read.
6. **Upload the data and display:**



With the help of pandas library, we can read and upload the data in csv form. we can display the first five rows of the data and also by applying **df.shape** we can also find out the number of rows and columns the data has.



* **Null Values:** isnull() is the process by which we can check whether there is any null value present in the data or not i.e. isnull() is part of the data cleaning stage. Here in the data, we found that there is no missing values as shown in the below image.



* **Info()** and **Describe()** is also a part of the data analysis where information can show you whether the column contains integer, object or float and describe can show you the statistical information of the columns which van also be considered as 5 stage method.

1. **Data Preprocessing:**

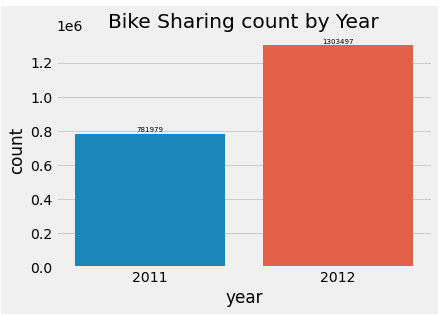
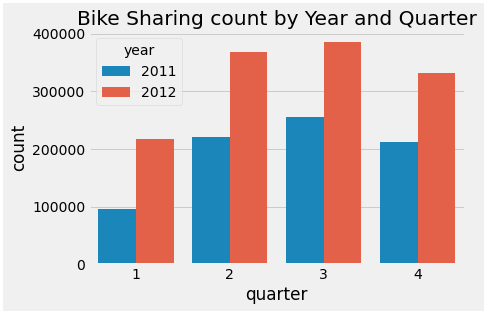
****

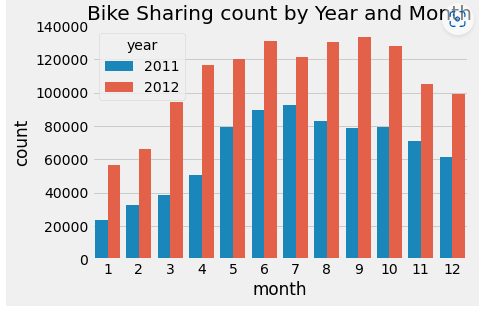
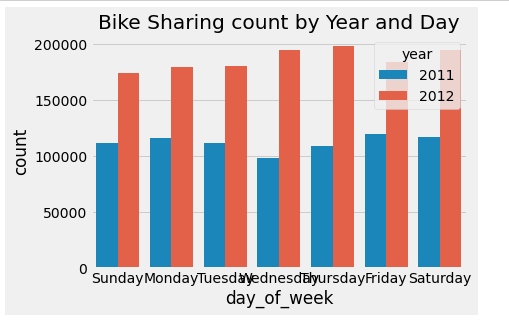
This code creating new columns in the dataframe based on the datetime column. The pd.to\_datetime() function is used to convert the datetime column to a datetime data type, which allows for easy manipulation of dates and times. The dt attribute is used to access various properties of the datetime column, such as year, quarter, month, day of the week, and hour. These properties are then assigned to new columns in the dataframe.

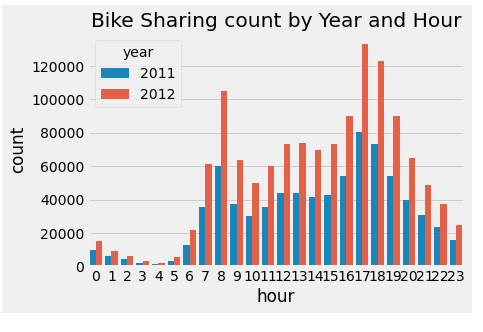
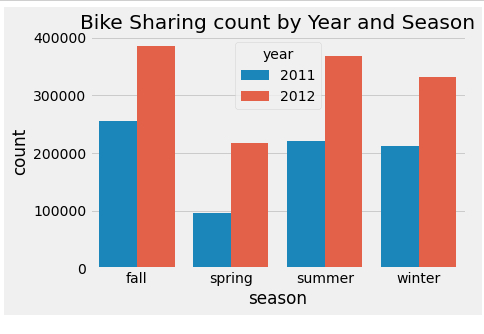
Mapping numerical values in the season, holiday, workingday, and weather columns tocorresponding text values. This is useful for making the data more readable and easier to understand.

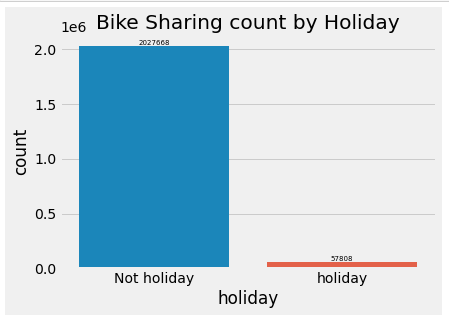
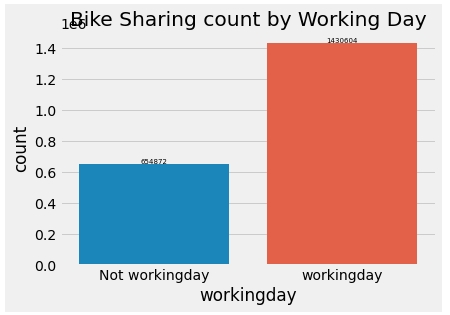
Overall, this code is cleaning and preprocessing the data to prepare it for further analysis.

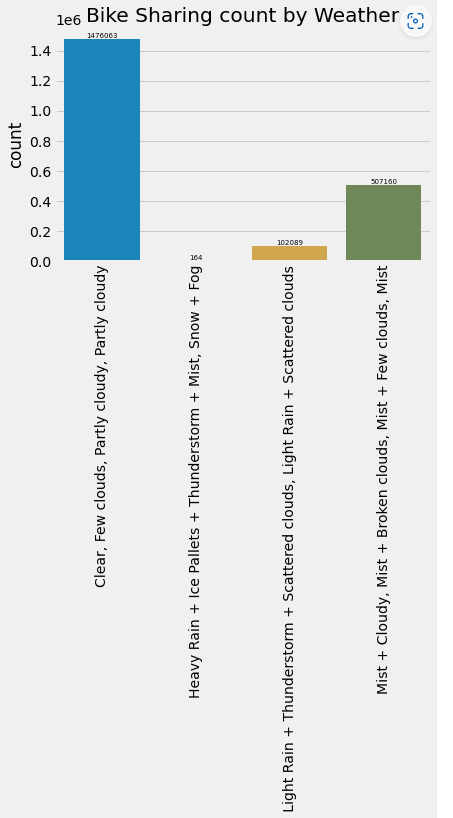
1. **Visualisation:**

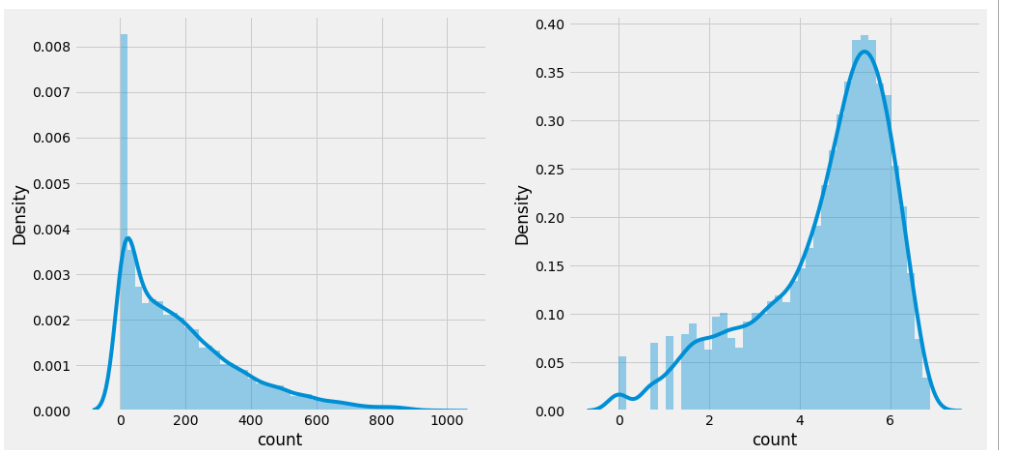
 

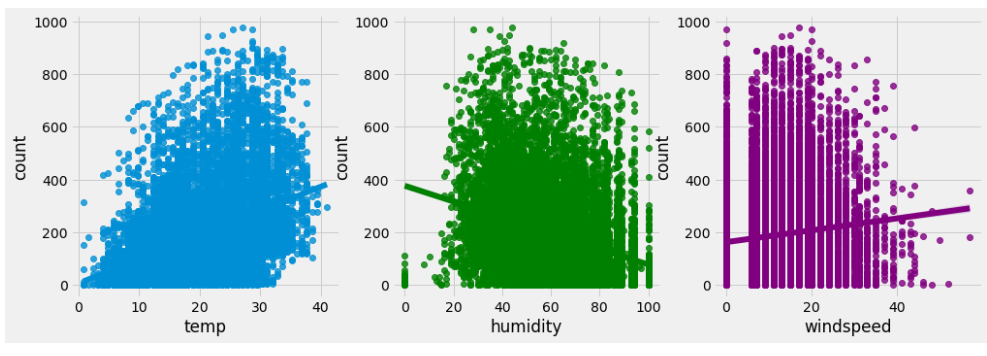
 



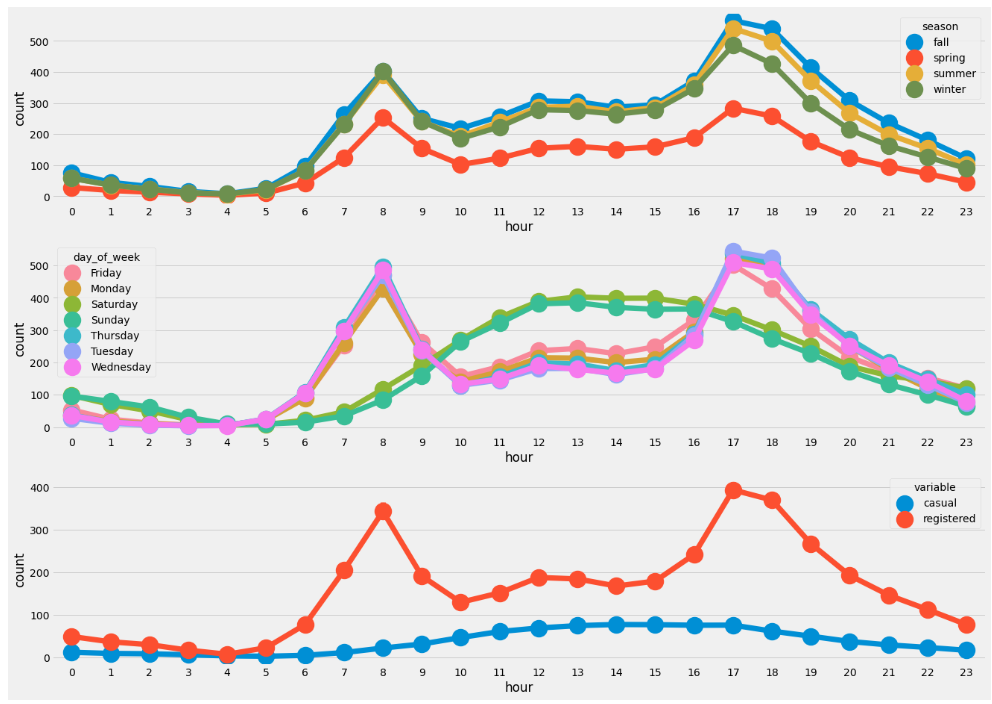
* **Distribution of count**

****

* **Visualization of temp, humidity and windspeed**

****

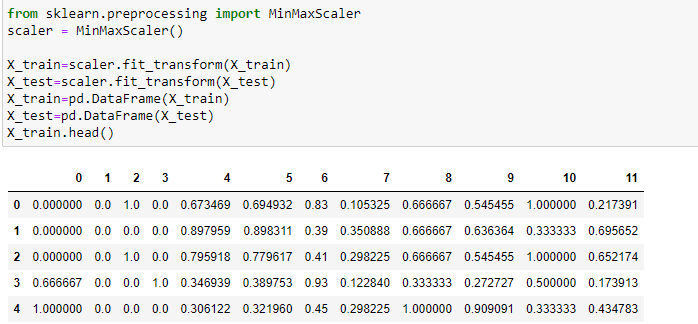
* **Visualization of Hour & Count Vs (season, day, registered)**

****

In the data preprocessing label encoding is such a technique where we can convert the categorival data into the numerical one and it has an advantage that we can apply the machine learning model here such as decision tree and regression model.

GridSearchCV and RandomizedSearchCV are two popular techniques for hyperparameter tuning in machine learning models. GridSearchCV takes RandomizedSearchCV, on the other hand, is a randomized search over a defined hyperparameter space. RandomizedSearchCV takes a similar input as GridSearchCV, but instead of performing an exhaustive search, it randomly samples a predefined number of combinations of hyperparameters from the hyperparameter space.

1. **Feature Scaling:**

****

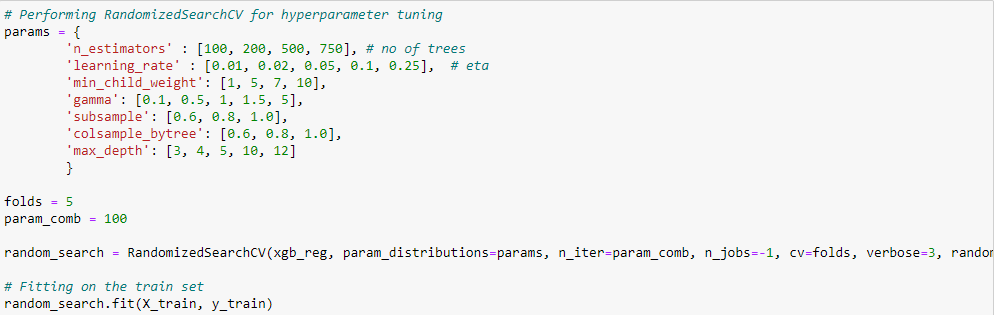
The code is performing feature scaling using MinMaxScaler, which scales the data so that all features are in the range of [0, 1]. This is a common preprocessing step in machine learning to ensure that all features are on the same scale and can be treated equally by the model. By fitting the scaler on the training data and then applying it to both training and testing data, the code ensures that the scaling is learned only from the training data and applied consistently to both the training and testing data.

**Model Building:**

The XGBRegressor class is a popular implementation of gradient boosting that is often used for regression problems. It can handle missing values and is relatively fast and scalable. Once the class is imported, you can create an instance of the XGBRegressor class, set its hyperparameters and fit it to the training data using the fit method.

The metrics imported from scikit-learn, r2\_score, mean\_absolute\_error, and mean\_squared\_error are commonly used metrics for evaluating the performance of regression models. The r2\_score is a measure of how well the model fits the data and ranges from 0 to 1, with 1 indicating a perfect fit. The mean\_absolute\_error and mean\_squared\_error are measures of how close the predicted values are to the actual values and are commonly used to compare the performance of different models.

**Random Search for XGboost Regression:**



Performing hyperparameter tuning using RandomizedSearchCV, which is a method for searching over a large hyperparameter space. The hyperparameters being tuned include the number of trees (n\_estimators), the learning rate (learning\_rate), the minimum weight of the child node (min\_child\_weight), the minimum loss reduction required to make a further partition on a leaf node (gamma), the fraction of observations to be randomly sampled for each tree (subsample), the fraction of columns to be randomly sampled for each tree (colsample\_bytree), and the maximum depth of a tree (max\_depth).

The 'folds' variable specifies the number of folds to be used in cross-validation, while 'param\_comb' specifies the number of combinations of hyperparameters to be sampled in the random search.

We can get score of : 0.8839963882867939.

Creating XGBRegression model with the select hyperparameters, Fitting the model on the train set then on the test set and also calculating the r2 score 87. Calculating Mean Sqaured Error & Root Mean Squared Error of the test set 60.18

In the prediction part and the model evaluation we can compare the actual value and the predicted value where the predicted R2 score on Test data : 89.0.