Regression Analysis on Seoul Bike Sharing Demand Prediction

– Sanjay Verma
Data science trainee at
Almabetter

ABSTRACT-

Bike rental predictions forecasts the demand for bikes rentals in dependency of weather conditions like the temperature and calendric information e.g. holidays. To make predictions machine learning is used. To predict the bike sharing demand, regression techniques are employed in this study to predict the demand on rental bikes in Seoul Bike sharing system. The prediction is carried out with the Seoul Bike data. The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.

PROBLEM STATEMENT-

Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

INTRODUCTION-

The bike sharing systems are very innovative ways of renting bicycles for use without having a burden of responsibility of ownership. This bike sharing model works in two modes -first user can get a membership for cheaper rates, second- take and pay for the bicycle on an hourly basis. The user of this system can pick up a bicycle from a Kiosk in one location and sign out to the kiosk in any designated location of the city.

In recent times this bike sharing system has gained a lot of attention around the world and also feasibility studies are being taken up all over the world,places like China ,Australia ,Sao Paulo to understand the infrastructural requirements as well as the benefits and impact of the same on the citizens. With more than 500 bike renting schemes across the globe. popular bike renting programs functional in London (Boris bikes),Washington (capital bike share) and New York (City bikes) which are used by millions of citizens every month.this scheme provides a very wide variety of data set for analysis purpose, this promoted the team to take up the interesting problem and the challenges faced for inventory management. In a bike sharing system which can be formulated as a 'bike sharing demand' problem wherein given a supervised set of data you have to create a model to predict the number of bikes that will be rented at the given hour in the future.

Objective-

The main objective of the project is to build a model to predict the demand of rental bikes and the bike count required at each hour for the stable supply of rental bikes.

Dataset Prepping-

The dataset has 8,760 observations, with each observation representing one hour of one day. The target variable is 'Rented Bike Count' and there were 14 attributes to work with. The baseline rental count for any given day is 735 bikes.

1. There are no NaN values in the dataset.
2. Changed the format of the Date.
3. Added some columns which are extracted from the Date column.

Data Description-

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.

Attribute Information:

- Date : year-month-day
- Rented Bike count - Count of bikes rented at each hour
- Hour - Hour of he day
- Temperature-Temperature in Celsius. (%)
- Humidity - a quantity representing the amount of water vapor in the atmosphere or in a gas.
- Wind speed - The rate at which air is moving in a particular area
- Visibility - a measure of the distance at which an object or light can be clearly discerned (10m)
- Dew point temperature - The temperature below which the water vapor in a volume of air at a constant pressure will condense into liquid water
- Solar radiation - The electromagnetic radiation emitted by the sun (MJ/m2)
- Rainfall - the quantity of rain falling within a given area in a given time (mm)
- Snowfall - the quantity of snow falling within a given area in a given time (cm)
- Seasons - Winter, Spring, Summer, Autumn
- Holiday - Holiday/No holiday
- Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours

CHALLENGES FACED-

To make the data tenable for understanding and further analysis , the data set was analyzed for identifiable statistical trends and patterns. After preliminary analysis, the following steps were undertaken to transform the data into a systematically workable dataset:

The following are the challenges faced in the data analysis:

- ➢ Changing date-time into timestamps
- ➢ Data engineering and adding new columns.
- ➢ Changing date-time into timestamps
- ➢ Splitting timestamps into days, months, years and days of the week.
- ➢ Converting season, holiday, working day and weather into categorical variables or factors.
- ➢ Model Implementation

APPROACH-

We performed the Outliers treatment and normalized the features for better results.Also,validate the data using VIF factor to check the multicollinearity in the dataset. One hot encoding technique is applied to create a new dataframe for model fitting.We use supervised learning regression analysis Linear Regression, Lasso Regression, Random Forest Regression,Decision Tree regressors and Gradient Boosting Regressors for the purpose of training the dataset to predict future supply and demand of bikes.
Hyperparameter tuning plays an important role to predict the best model among the above regression models.

TOOLS USED-

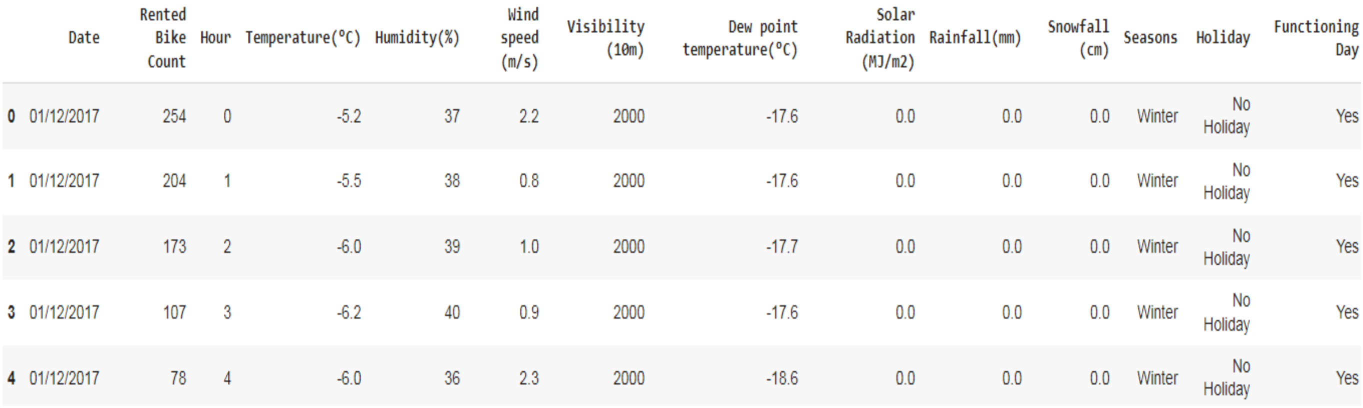
The whole project was done using python, in google colaboratory. Following libraries were used for analyzing the data and visualizing it and to build the model to predict the bike count required at each hour for the stable supply of rental bikes.

- Pandas: Extensively used to load and wrangle with the dataset.
- Matplotlib: Used for visualization.
- Seaborn: Used for visualization.
- Datetime: Used for analyzing the date variable.
- Warnings: For filtering and ignoring the warnings.
- Numpy: For some math operations in predictions.
- Sklearn: For the purpose of analysis and prediction.
- Datetime: For reading the date.
- Statsmodels: For outliers influence.

The below table shows the dataset in the form of Pandas DataFrame.

Date, Rented Bike count, Hour ,Temperature, Humidity ,Wind speed, Visibility, Dew point temperature, Solar radiation, Snowfall, etc.

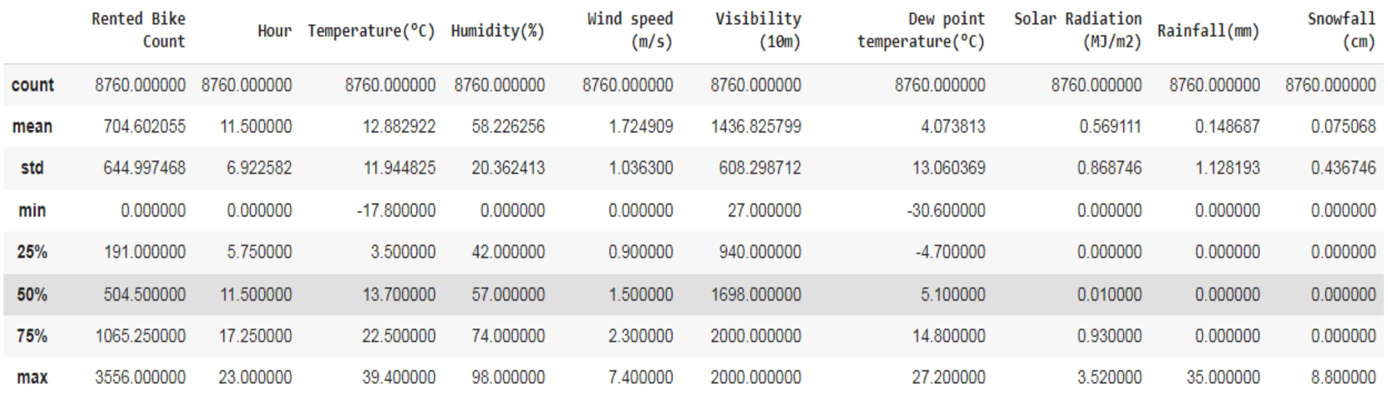
Pandas DataFrame



	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	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
2	01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	Yes
3	01/12/2017	107	3	-6.2	40	0.9	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
4	01/12/2017	78	4	-6.0	36	2.3	2000	-18.6	0.0	0.0	0.0	Winter	No Holiday	Yes

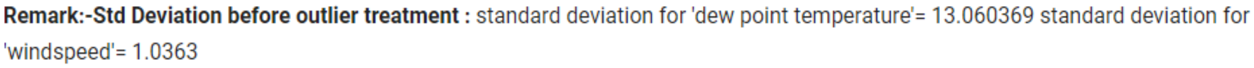
Description

The below table shows the mathematical calculations such as count, mean,standard deviation, minimum and percentiles of all features of the dataset.

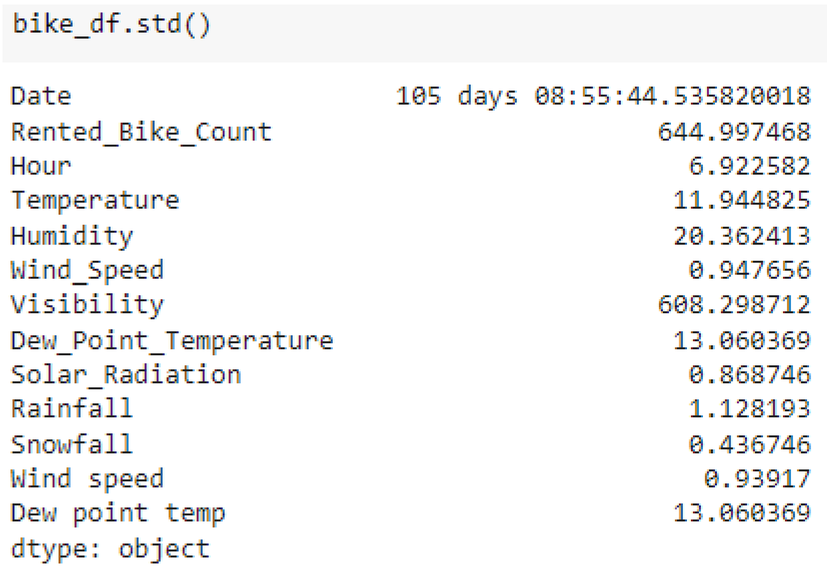


	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)
count	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000
mean	704.602055	11.500000	12.882922	58.226256	1.724909	1436.825799	4.073813	0.569111	0.148687	0.075068
std	644.997468	6.922582	11.944825	20.362413	1.036300	608.298712	13.060369	0.868746	1.128193	0.436746
min	0.000000	0.000000	-17.800000	0.000000	0.000000	27.000000	-30.600000	0.000000	0.000000	0.000000
25%	191.000000	5.750000	3.500000	42.000000	0.900000	940.000000	-4.700000	0.000000	0.000000	0.000000
50%	504.500000	11.500000	13.700000	57.000000	1.500000	1698.000000	5.100000	0.010000	0.000000	0.000000
75%	1065.250000	17.250000	22.500000	74.000000	2.300000	2000.000000	14.800000	0.930000	0.000000	0.000000
max	3556.000000	23.000000	39.400000	98.000000	7.400000	2000.000000	27.200000	3.520000	35.000000	8.800000

OUTLIERS ANALYSIS-

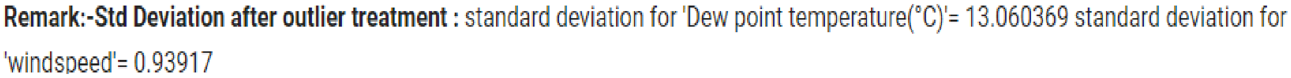


Remark:-Std Deviation before outlier treatment : standard deviation for 'dew point temperature'= 13.060369 standard deviation for 'windspeed'= 1.0363



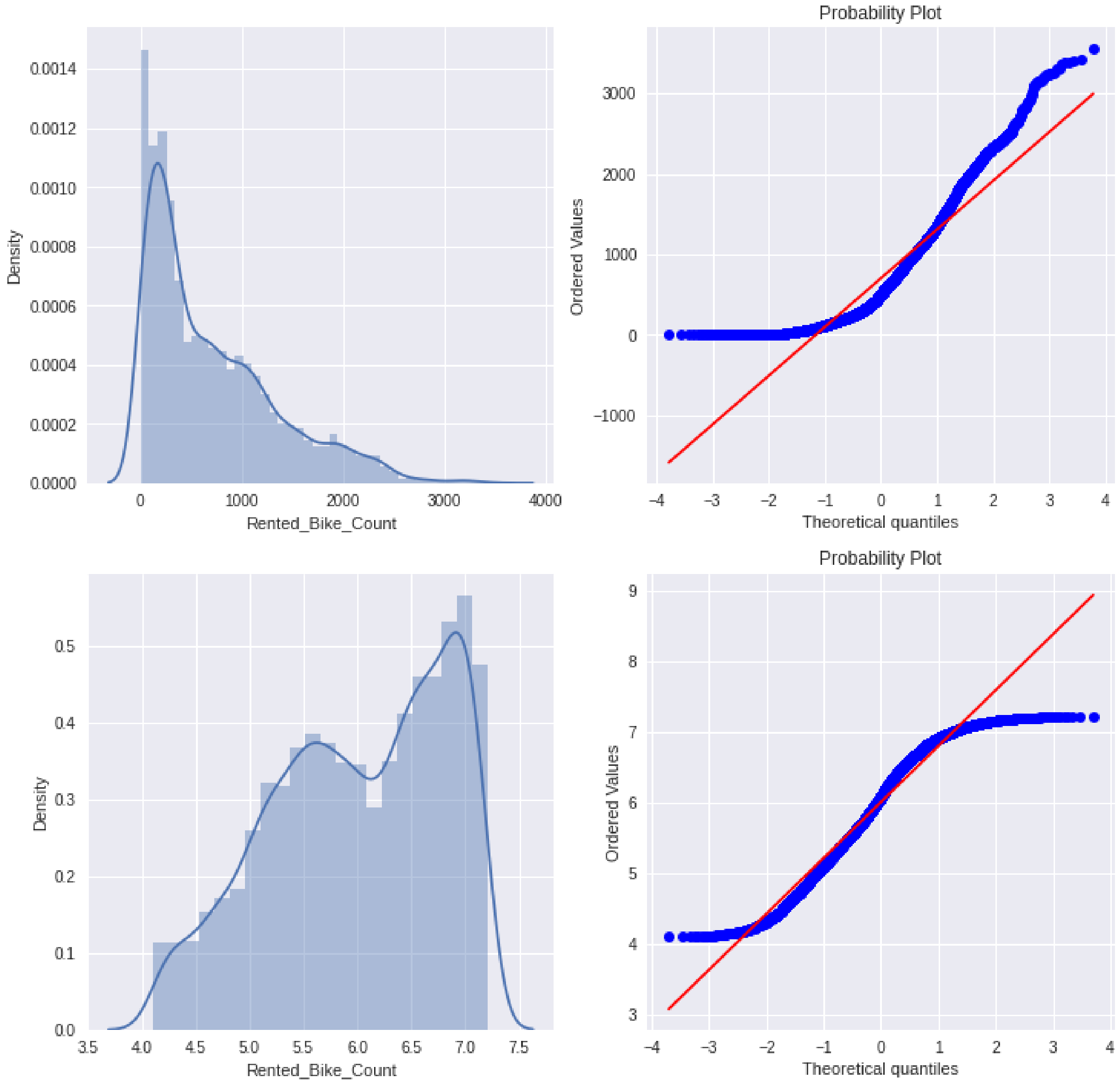
```
bike_df.std()
```

```
Date                     105 days 08:55:44.535820018
Rented_Bike_Count                         644.997468
Hour                                        6.922582
Temperature                                11.944825
Humidity                                   20.362413
Wind_Speed                                  0.947656
Visibility                                608.298712
Dew_Point_Temperature                      13.060369
Solar_Radiation                             0.868746
Rainfall                                    1.128193
Snowfall                                    0.436746
Wind speed                                   0.93917
Dew point temp                             13.060369
dtype: object
```



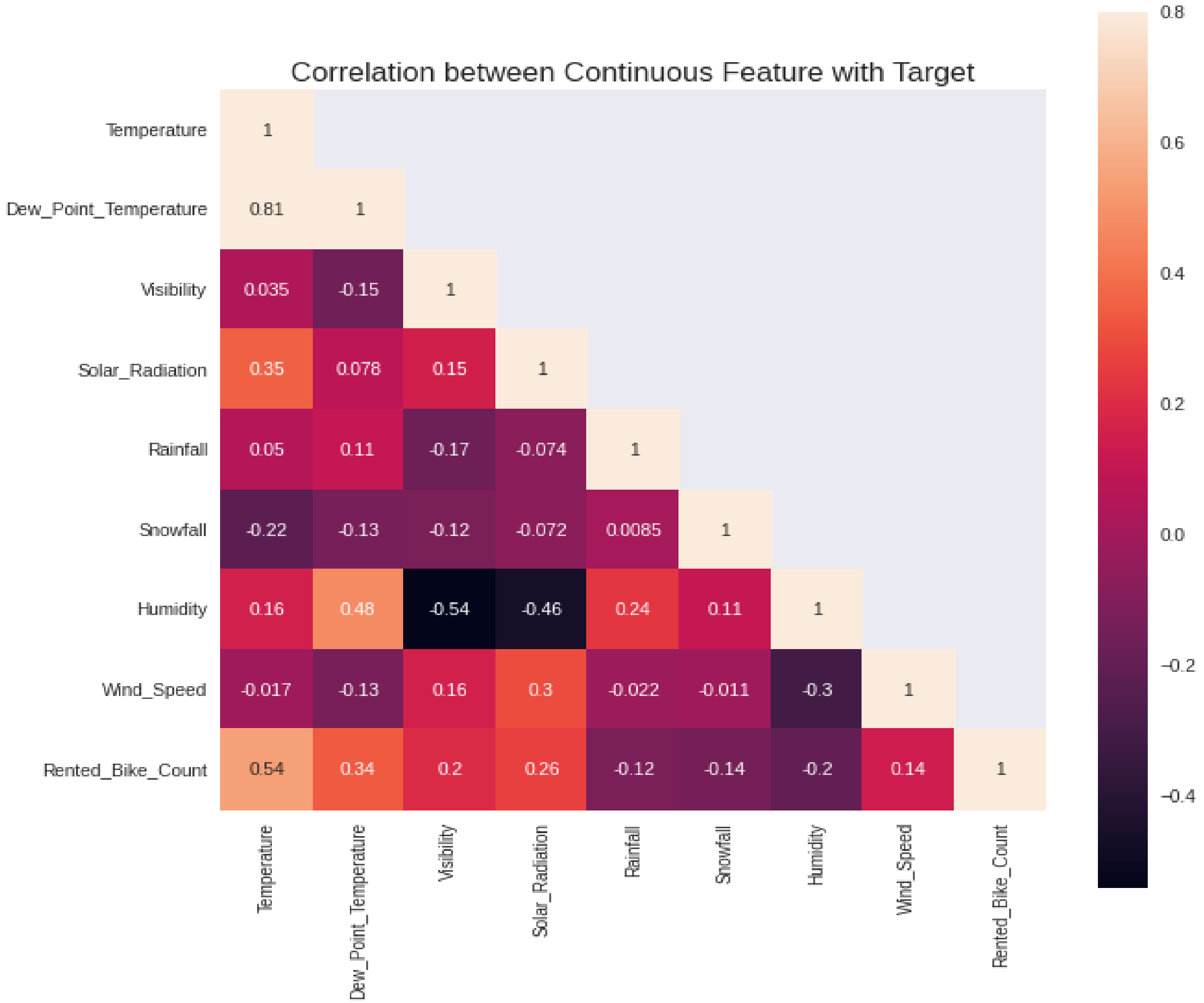
Remark:-Std Deviation after outlier treatment : standard deviation for 'Dew point temperature(°C)'= 13.060369 standard deviation for 'windspeed'= 0.93917

Target Distribution With and Without Outliers

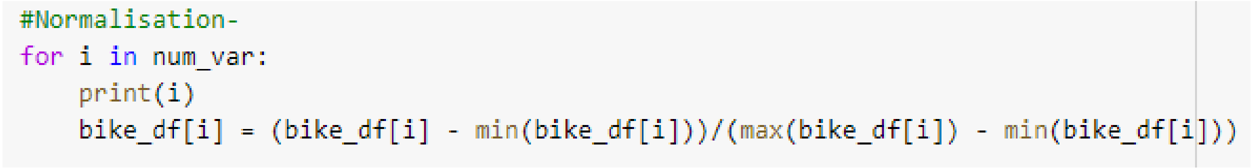


CORRELATION HEATMAP-

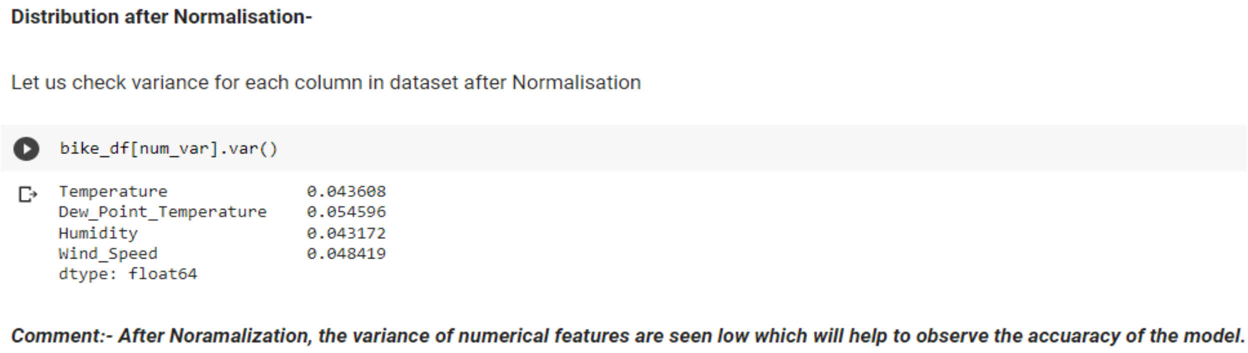
To refine the data further, a correlation matrix was created amongst all the feature variables to analyze interaction effects.



NORMALIZATION OF NUMERICAL FEATURES-



```
#Normalisation-
for i in num_var:
    print(i)
    bike_df[i] = (bike_df[i] - min(bike_df[i]))/(max(bike_df[i]) - min(bike_df[i]))
```



Distribution after Normalisation-

Let us check variance for each column in dataset after Normalisation

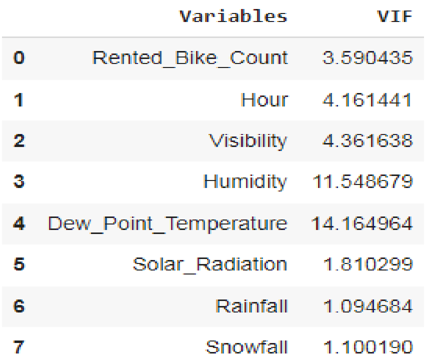
```
bike_df[num_var].var()
```

```
Temperature             0.043608
Dew_Point_Temperature   0.054596
Humidity                0.043172
Wind_Speed              0.048419
dtype: float64
```

Comment:- After Noramalization, the variance of numerical features are seen low which will help to observe the accuarcy of the model.

MULTICOLLINEARITY-

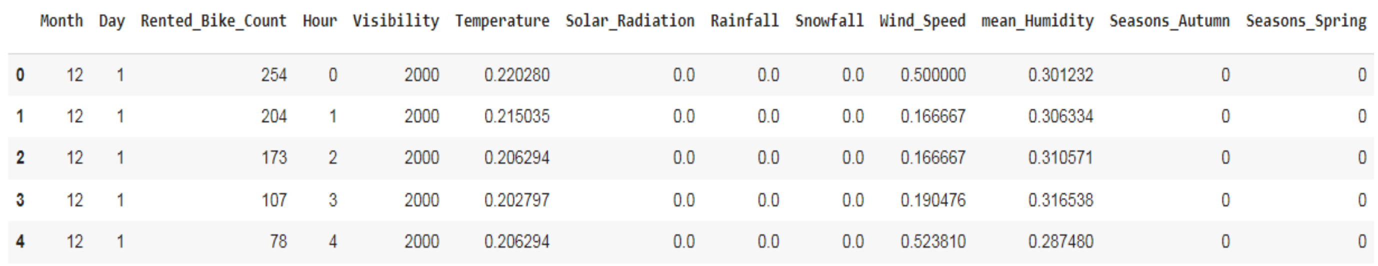
- Multicollinearity is a statistical concept where several independent variables in a model are correlated.
- Two variables are considered to be perfectly collinear if their correlation coefficient is +/- 1.0.
- Multicollinearity among independent variables will result in less reliable statistical inferences.
- It is better to use independent variables that are not correlated or repetitive when building multiple regression models that use two or more variables.
- The existence of multicollinearity in a data set can lead to less reliable results due to larger standard errors.



	Variables	VIF
0	Rented_Bike_Count	3.590435
1	Hour	4.161441
2	Visibility	4.361638
3	Humidity	11.548679
4	Dew_Point_Temperature	14.164964
5	Solar_Radiation	1.810299
6	Rainfall	1.094684
7	Snowfall	1.100190

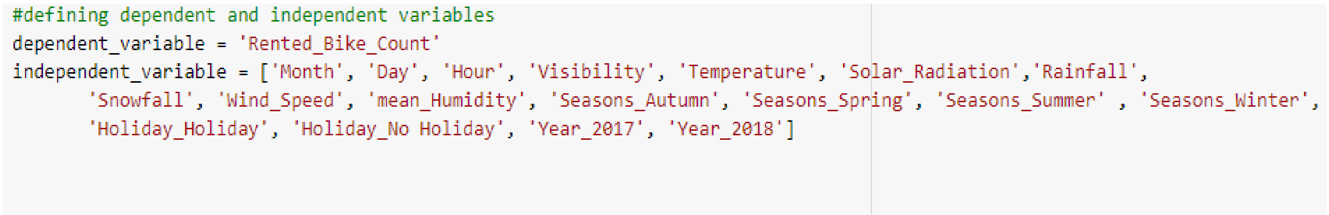
From above Dataframe, we see that there is Multicollinearity in our Data for- Dew point temperature(°C) and Humidity(%) has highest VIF value

One-Hot Encoding-

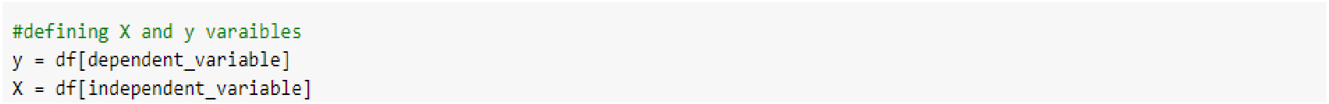


	Month	Day	Rented_Bike_Count	Hour	Visibility	Temperature	Solar_Radiation	Rainfall	Snowfall	Wind_Speed	mean_Humidity	Seasons_Autumn	Seasons_Spring
0	12	1	254	0	2000	0.220280	0.0	0.0	0.0	0.500000	0.301232	0	0
1	12	1	204	1	2000	0.215035	0.0	0.0	0.0	0.166667	0.306334	0	0
2	12	1	173	2	2000	0.206294	0.0	0.0	0.0	0.166667	0.310571	0	0
3	12	1	107	3	2000	0.202797	0.0	0.0	0.0	0.190476	0.316538	0	0
4	12	1	78	4	2000	0.206294	0.0	0.0	0.0	0.523810	0.287480	0	0

DATA PREPROCESSING-



```
#defining dependent and independent variables
dependent_variable = 'Rented_Bike_Count'
independent_variable = ['Month', 'Day', 'Hour', 'Visibility', 'Temperature', 'Solar_Radiation','Rainfall',
       'Snowfall', 'Wind_Speed', 'mean_Humidity', 'Seasons_Autumn', 'Seasons_Spring', 'Seasons_Summer' , 'Seasons_Winter',
       'Holiday_Holiday', 'Holiday_No Holiday', 'Year_2017', 'Year_2018']
```



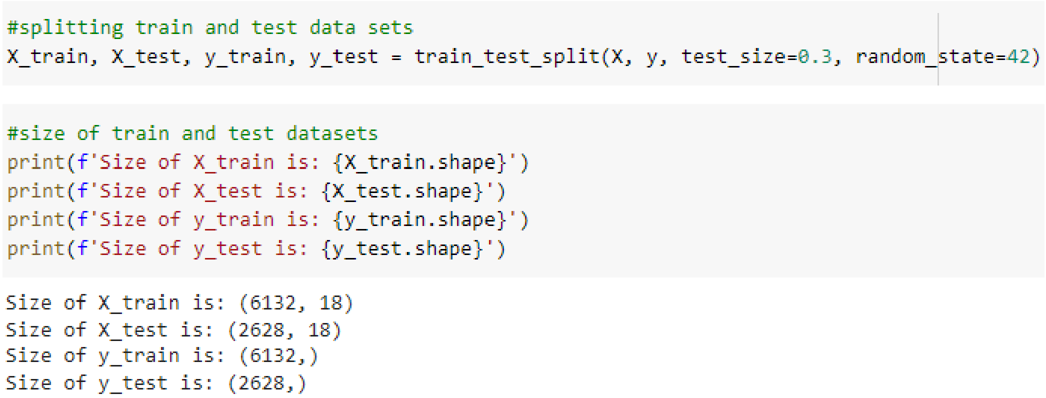
```
#defining X and y varaibles
y = df[dependent_variable]
X = df[independent_variable]
```

DATA MODELING-

After the data preparation was completed it is ready for the purpose of building the model to predict the sales. Only numerical valued features are taken into consideration. The data was combined and labeled as X and y as independent and dependent variables respectively. The sales feature is taken as dependent variable (y) and remaining all are considered as independent variables (X).

SPLITTING OF DATASET-

The train_test_split was imported from the sklearn.model_selection. The data is now divided into 70% and 30% as train and test splits respectively.70% of the data is taken for training the model and 30% is for test and the random state was taken as 42.



```
#splitting train and test data sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
#size of train and test datasets
print(f'Size of X_train is: {X_train.shape}')
print(f'Size of X_test is: {X_test.shape}')
print(f'Size of y_train is: {y_train.shape}')
print(f'Size of y_test is: {y_test.shape}')
```

```
Size of X_train is: (6132, 18)
Size of X_test is: (2628, 18)
Size of y_train is: (6132,)
Size of y_test is: (2628,)
```

METRICS USED-

The metrics are tools used for evaluating the performance of a regression model. The following are the metrics that have been used in the analysis of the data.

- **Mean Squared Error**

The mean squared error (MSE) is a commonly used metric for estimating errors in regression models. It provides a positive value as the error gets closer to zero. It is simply the average of the squared difference between the target value and the value predicted by the regression model.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

- **Root Mean Squared Error**

The root-mean-Square error or RMSE is a frequently used measure to evaluate the differences between the values that a model has predicted and the values that were observed. It is computed by taking the second sample moment and dividing it by the quadratic mean of the differences.

RMSD is a non-zero measure that shows a perfect fit to the data, and it is generally better than a higher one. It is not used to evaluate the relationships among different types of data.

RMSD is the sum of the average of all errors. It is sensitive to outliers.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - \hat{x}_i\right)^2}{N}}$$

- **R2 score**

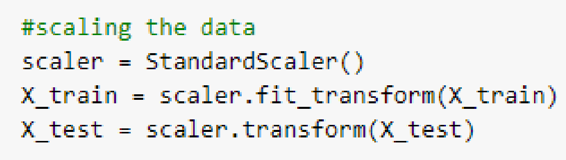
The goodness-of-fit evaluation should not be performed on the R2 linear regression because it quantifies the degree of linear correlation between the two values. Instead, the linear correlation should only be taken into account when evaluating the Ypred-Yobs relationship.

The metric helps us to compare our current model with a constant baseline and tells us how much our model is better. The constant baseline is chosen by taking the mean of the data and drawing a line at the mean. R^2 is a scale-free score that implies it doesn't matter whether the values are too large or too small, the R^2 will always be less than or equal to 1.

$$R^2 = 1 - \frac{RSS}{TSS}$$

SCALAR TRANSFORMATION OF DATASET-

To normalize the data, a standardscaler was used from sklearn preprocessing. It scales the data in the form of standard deviation of the feature divided from the difference of variable and its mean of the feature. At first the training data was made fit into the scaling function and test data is transformed now.



```
#scaling the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

MODELS-

➔ Linear Regression -

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Mathematically, we can represent a linear regression as:

$$y= a_0+a_1x+ \varepsilon$$

Here,

Y= Dependent Variable (Target Variable)

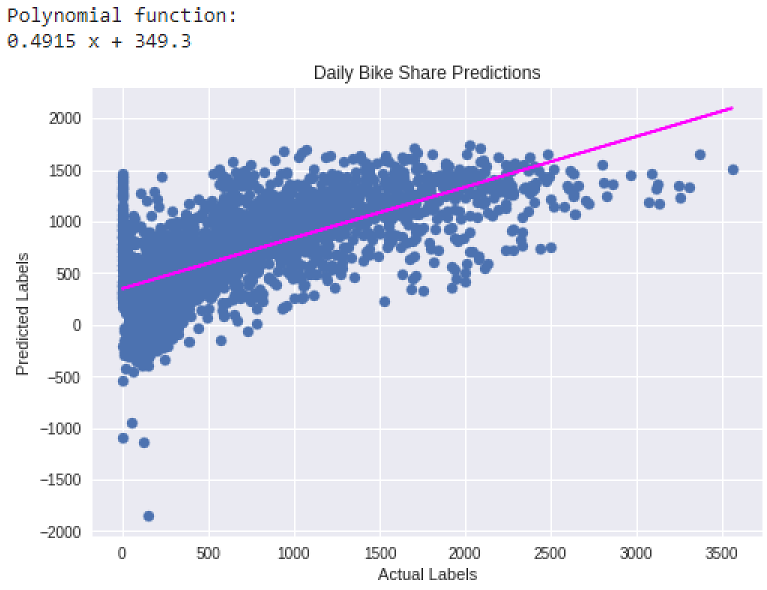
X= Independent Variable (predictor Variable)

a0= intercept of the line (Gives an additional degree of freedom)

a1 = Linear regression coefficient (scale factor to each input value).

ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.



```
Polynomial function:
0.4915 x + 349.3
```

➔ **Lasso regression-**

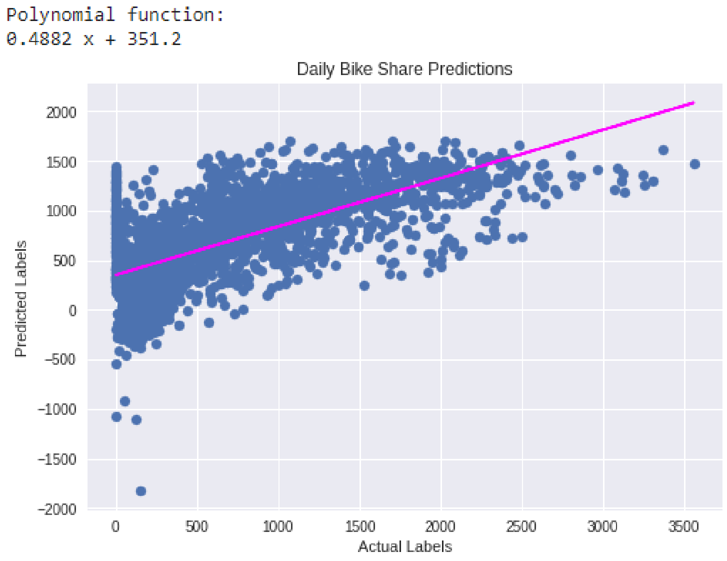
In statistics and machine learning, lasso (least absolute shrinkage and selection operator; also Lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model. Mathematical equation of Lasso Regression as-

Residual Sum of Squares + λ * (Sum of the absolute value of the magnitude of coefficients)

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Where,

- λ denotes the amount of shrinkage.
- λ = 0 implies all features are considered and it is equivalent to the linear regression where only the residual sum of squares is considered to build a predictive model
- λ = ∞ implies no feature is considered i.e, as λ closes to infinity it eliminates more and more features
- The bias increases with increase in λ
- variance increases with decrease in λ



```
Polynomial function:
0.4882 x + 351.2
```

➔ **Decision Tree regressors-**

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

Decision Tree Terminologies

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

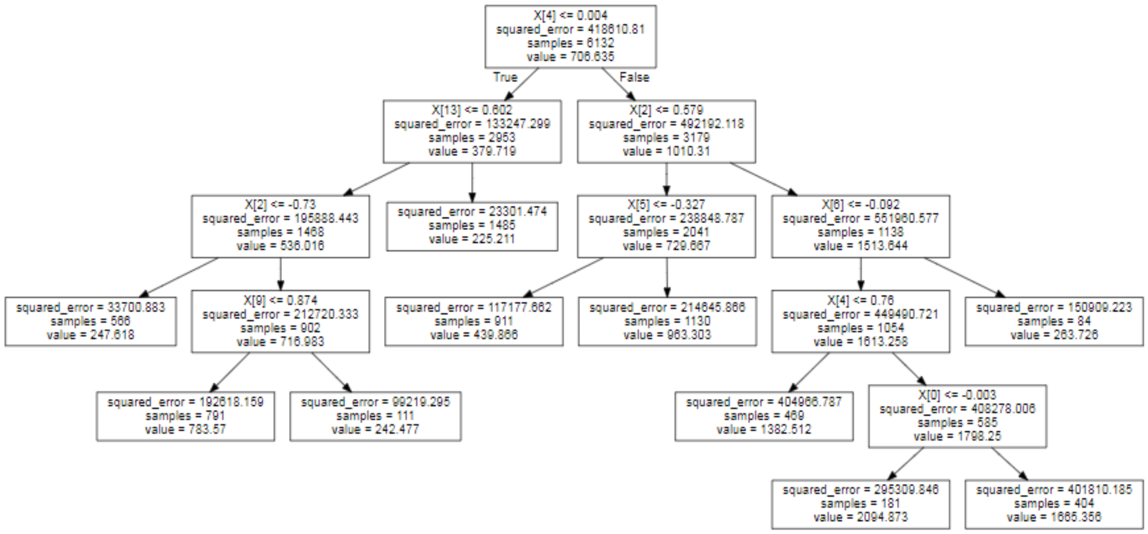
Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Subtree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches from the tree.

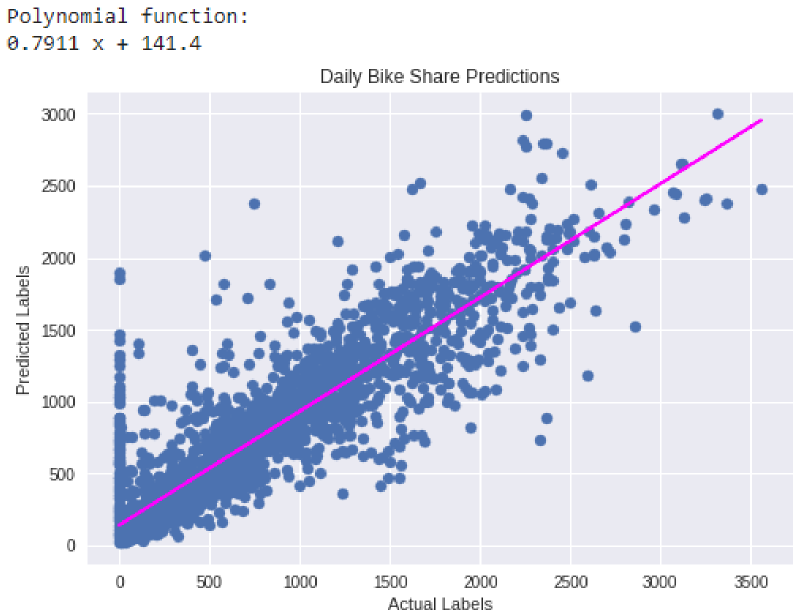
Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.



➔ **Random Forest Regressors-**

Regression is the other task performed by a random forest algorithm. A random forest regression follows the concept of simple regression. Values of dependent (features) and independent variables are passed in the random forest model.In a random forest regression, each tree produces a specific prediction. The mean prediction of the individual trees is the output of the regression. This is contrary to random forest classification, whose output is determined by the mode of the decision trees' class.

- The random forest algorithm provides a higher level of accuracy in predicting outcomes over the decision tree algorithm.

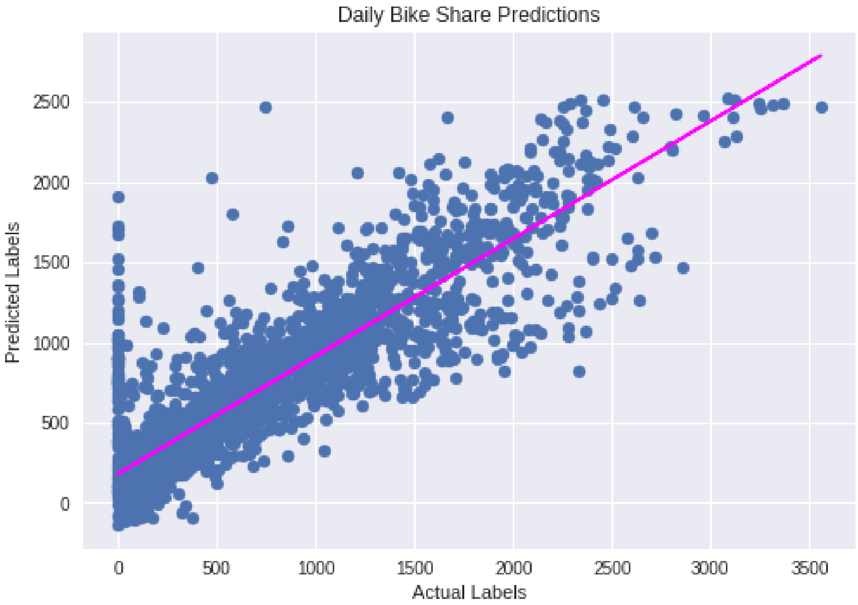


```
Polynomial function:
0.7911 x + 141.4
```

➔ **Gradient Boosting Regressors-**

Gradient boosting algorithm is one of the most powerful algorithms in the field of machine learning. As we know that the errors in machine learning algorithms are broadly classified into two categories i.e. Bias Error and Variance Error. As gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

```
Polynomial function:
0.7332 x + 180.1
```

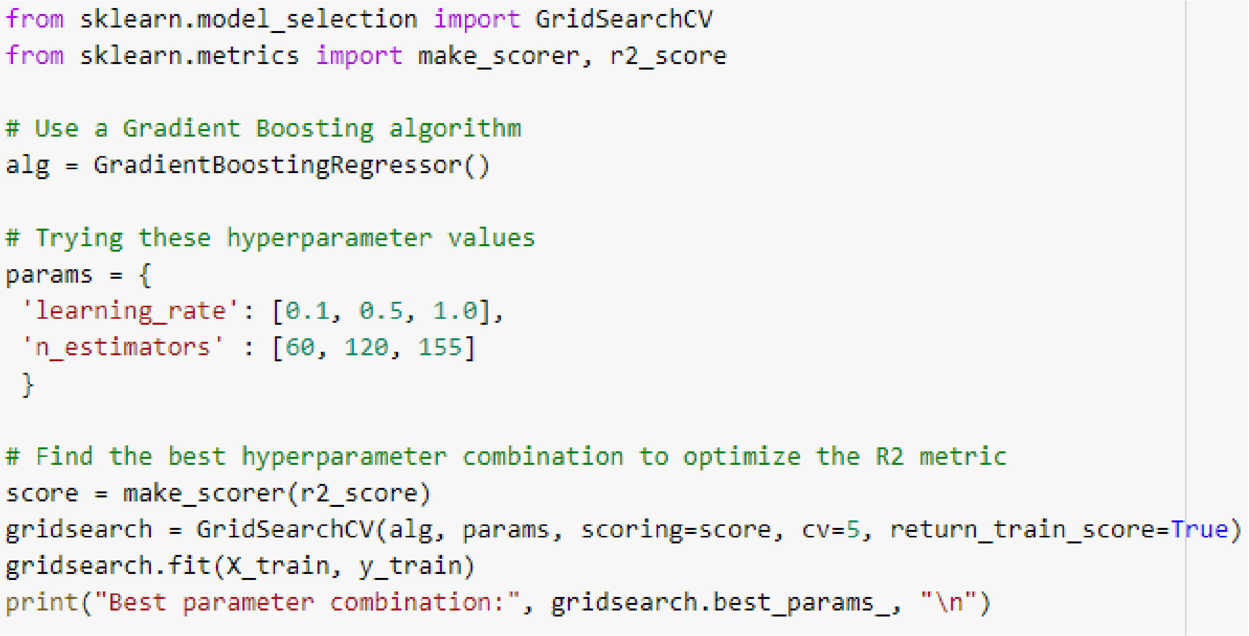


CROSS VALIDATION (Gradient Boosting Regressors)-

Cross-validation is a resampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.Cross-validation gives a more accurate measure of model quality, which is especially important if you are making a lot of modeling decisions.

HYPERPARAMETER TUNING-

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.



```python
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import make_scorer, r2_score

# Use a Gradient Boosting algorithm
alg = GradientBoostingRegressor()

# Trying these hyperparameter values
params = {
 'learning_rate': [0.1, 0.5, 1.0],
 'n_estimators' : [60, 120, 155]
 }

# Find the best hyperparameter combination to optimize the R2 metric
score = make_scorer(r2_score)
gridsearch = GridSearchCV(alg, params, scoring=score, cv=5, return_train_score=True)
gridsearch.fit(X_train, y_train)
print("Best parameter combination:", gridsearch.best_params_, "\n")
```

Conclusions-

- ❑ Overall we observe that,Linear Regression model and Lasso Regression model are worst fitted models as their accuracy is less than 50% whereas, Random Forest Regressor and Gradient Boosting Regressor are the best fitted model for the train and test data set.
- ❑ Random Forest Regressor has the accuracy rate of train data set 98% and test data set 81%. Also,MSE is 463.08 for the train data set and 280.61 for the test data set. After,hyperparameter tuning the accuracy rate gives the similar result for the train and test data set.
- ❑ Gradient Boosting Regressor has the accuracy rate of train data set 79% and test data set 77%. Also,MSE is 463.08 for the train data set and 309.65 for the test data set. With hyperparameter tuning the accuracy of the model increases and RMSE decreases which implies that the model fitted is the best model for higher accuracy rate of regression models with the predictions.

With Hyperparameter tuning-

1. Accuracy of the model of train data set is 90%
2. Accuracy of the model of test data set is 80%
3. RMSE of the model of train data set is 463.08
4. RMSE of the model of test data set is 285.46

- **Among all the above models we conclude that Gradient Boosting Regressor(With hyperparameter tuning) is the best fitted model for Seoul Bike Rental Prediction data set.**

THANKYOU !!...