Capstone Project

SEOUL BIKE SHARING DEMAND PREDICTION

PROBLEM DESCRIPTION:

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.

❑ CONTENT

❑ BUSINESS UNDERSTANDING

❑ DATA SUMMARY

❑ FEATURE ANALYSIS

❑ EXPLORATORY DATA ANALYSIS

❑ DATA PREPROCESSING

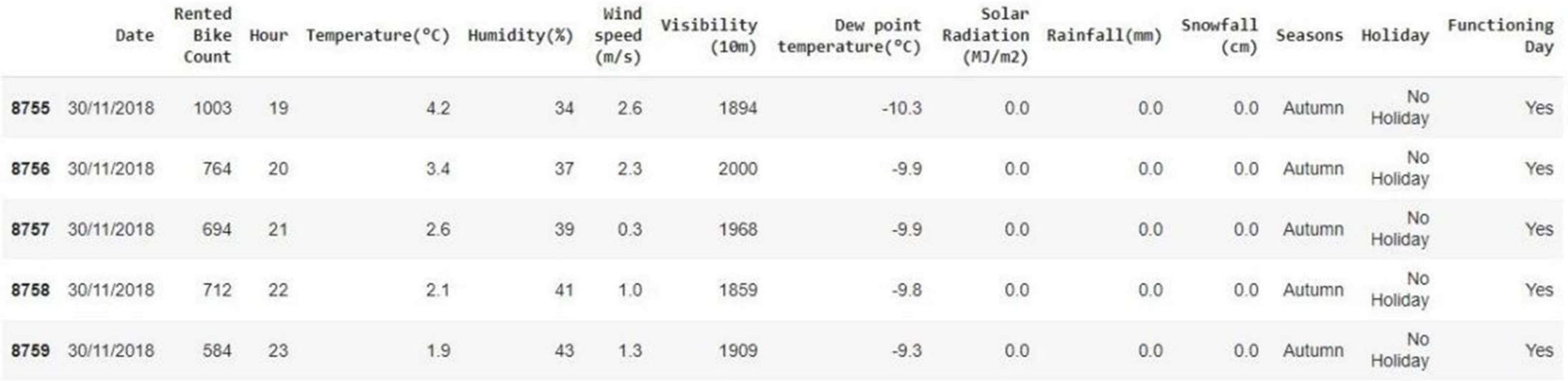
❑ IMPLEMENTING ALGORITHMS

❑CONCLUSION

❑ BUSINESS UNDERSTANDING

- Bike rentals have became a popular service in recent years and it seems people are using it more often. With relatively cheaper rates and ease of pick up and drop at own convenience is what making this business thrive.
- Mostly used by people having no personal vehicles and also to avoid congested public transport which that’s why they prefer rental bikes.
- Therefore, the business to strive and profit more, it has to be always ready and supply no. of bikes at different locations, to fulfil the demand.
- Our project goal is a pre planned set of bike count values that can be a handy solution to meet all demands.

❑ DATA SUMMARY



	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
8755	30/11/2018	1003	19	4.2	34	2.6	1894	-10.3	0.0	0.0	0.0	Autumn	No Holiday	Yes
8756	30/11/2018	764	20	3.4	37	2.3	2000	-9.9	0.0	0.0	0.0	Autumn	No Holiday	Yes
8757	30/11/2018	694	21	2.6	39	0.3	1968	-9.9	0.0	0.0	0.0	Autumn	No Holiday	Yes
8758	30/11/2018	712	22	2.1	41	1.0	1859	-9.8	0.0	0.0	0.0	Autumn	No Holiday	Yes
8759	30/11/2018	584	23	1.9	43	1.3	1909	-9.3	0.0	0.0	0.0	Autumn	No Holiday	Yes

- This Dataset contains 8760 record and 14 columns.
- Three categorical features ‘Seasons’, ‘Holiday’, & ‘Functioning Day’.
- One Datetime features ‘Date’.
- We have some numerical type variables such as temperature, humidity, wind, visibility, dew point temp, solar radiation, rainfall, snowfall which tells the environment conditions at that particular hour of the day.

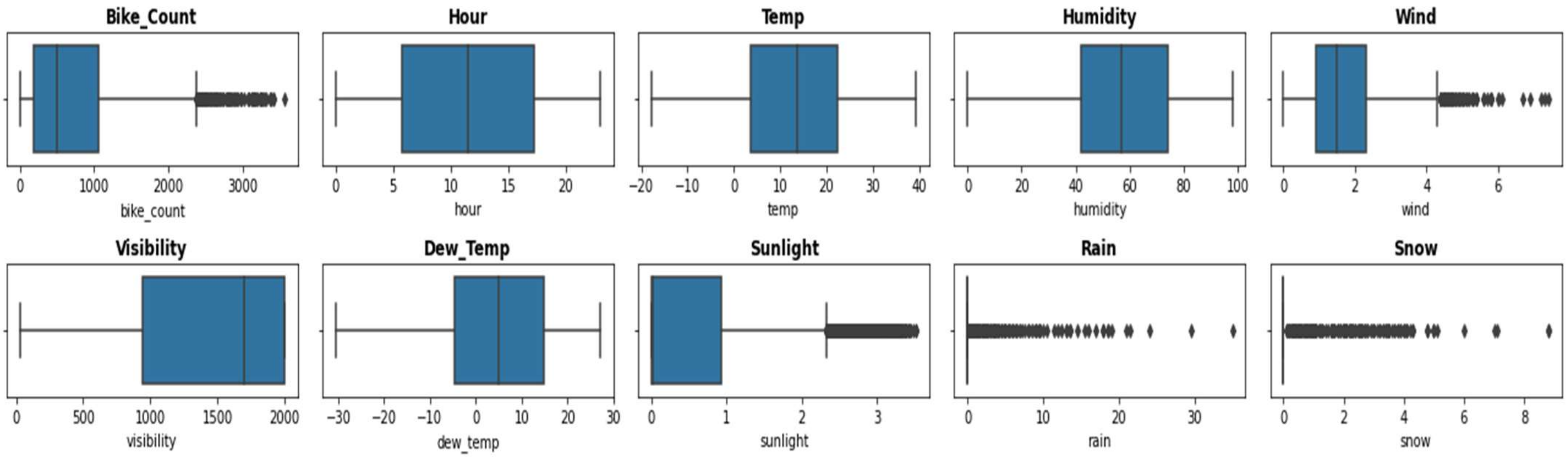
❑ FEATURE SUMMARY

- Date : Year-Month-Day
- Rented Bike Count - Count of bikes rented at each hour
- Hour - Hour of the day
- Temperature - Temperature in Celsius
- Humidity - %
- Wind Speed - m/s
- Visibility - 10m
- Dew point temperature -Celsius
- Solar radiation -MJ/m2
- Rainfall -mm
- Snowfall –cm
- Seasons -Winter, Spring, Summer, Autumn
- Holiday -Holiday/No Holiday
- Functional Day - NoFunc(Non Functional Hrs),Fun(Functional Hrs)

❑ INSIGHTS FORM THE DATASETS

- There are No Missing Values present
- There are No Duplicate values present
- There are No null values.
- And finally we have dependent feature 'rented bike count' variable which we need to predict for new observations .
- The dataset shows hourly rental data for one year (1 December 2017 to 31 November(2018)(365 days).we consider this as a single year data

❑ COLUMNS WISE ANALYSIS



- 1.From above graphs we detect ,Outliers are present in the bike_count, wind,sunlight,rain,snow.
- 2.Dependent variable i.e bike_count having a moderate right skewed, to apply linear regression dependent feature have to follow the normal distribution .Therefore we use IQR and log transformation.
- 3.Features like wind , sunlight,rain,snow are to be treated by IQR method, where we capped the record after 99 percentile with the median Value of that column .

MANIPULATING THE DATASETS

1. Added new feature called **weekend** which takes Saturday and Sunday as one else zero .
2. Added one more new feature called day_or_night which is followed by two segment named as Day, Night.
3. Dropping the date column after taking out month_name, and day_name.
4. Defining a label encoder of three different columns Holiday , functioning day and day_or_night .

a.In holiday column it takes **holiday** as **one** and no holiday as **zero** .

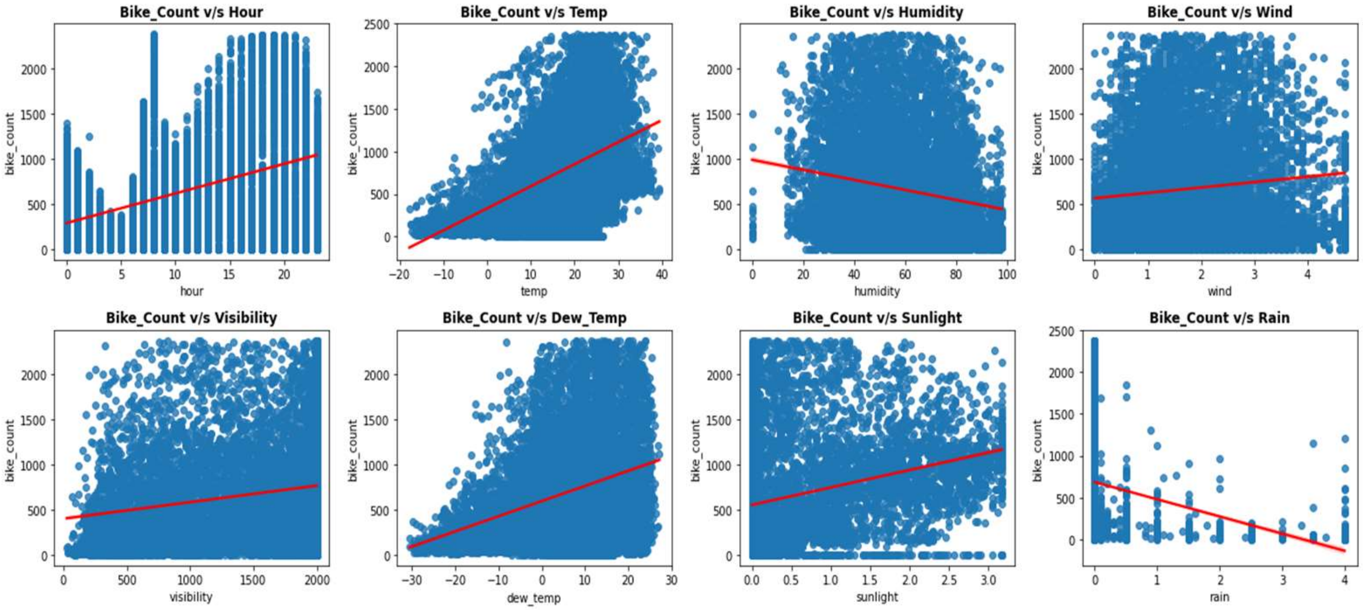
b.From functioning day column takes **one** as **functioning** day and **zero** as no functioning happen.

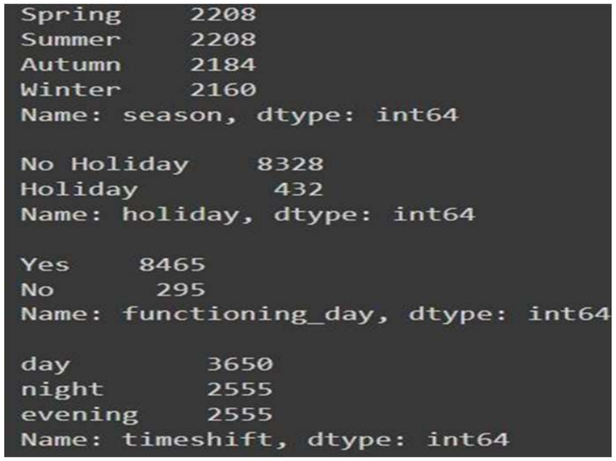
c.Lastly from the day_or_night column which takes night equal to **0**, day equal to **1** .

5. Doing one hot encoding of feature seasons and month_name

6. We come up with 28 features. By applying RFE we select top 13 feature contributing towards prediction

Continued.. LINEAR COLLINEARITY





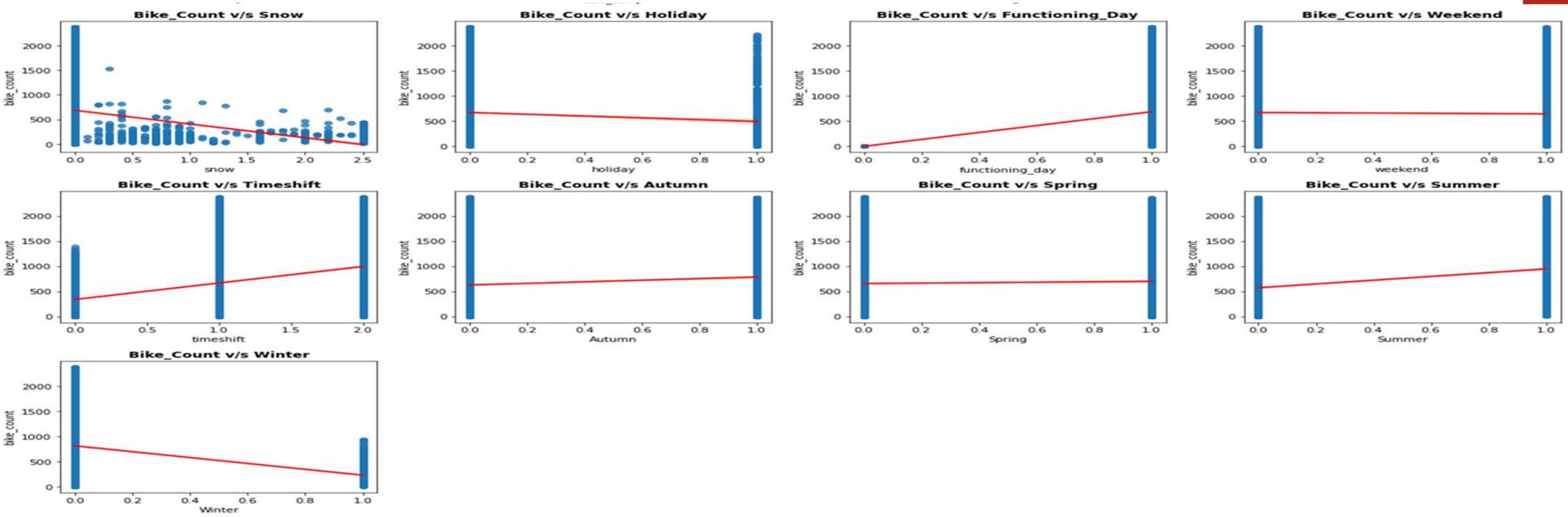
```
Spring     2208
Summer     2208
Autumn     2184
Winter     2160
Name: season, dtype: int64

No Holiday    8328
Holiday        432
Name: holiday, dtype: int64

Yes     8465
No       295
Name: functioning_day, dtype: int64

day          3650
night        2555
evening      2555
Name: timeshift, dtype: int64
```

5.Using one hot encoding in the season column to create the dummies feature of column segment like **summer** , **autumn**, **spring** and **winter**.

6.Adding a new variable called **independent**_variable contains all column except bike_**count**.

Adding some value counts of the categorical column : **Autumn** has max count in the **season** col, followed by **no holiday** , **yes** in the **functioning** day , and **day** segment in the **timeshift** column.

Linear collinearity with dependent column

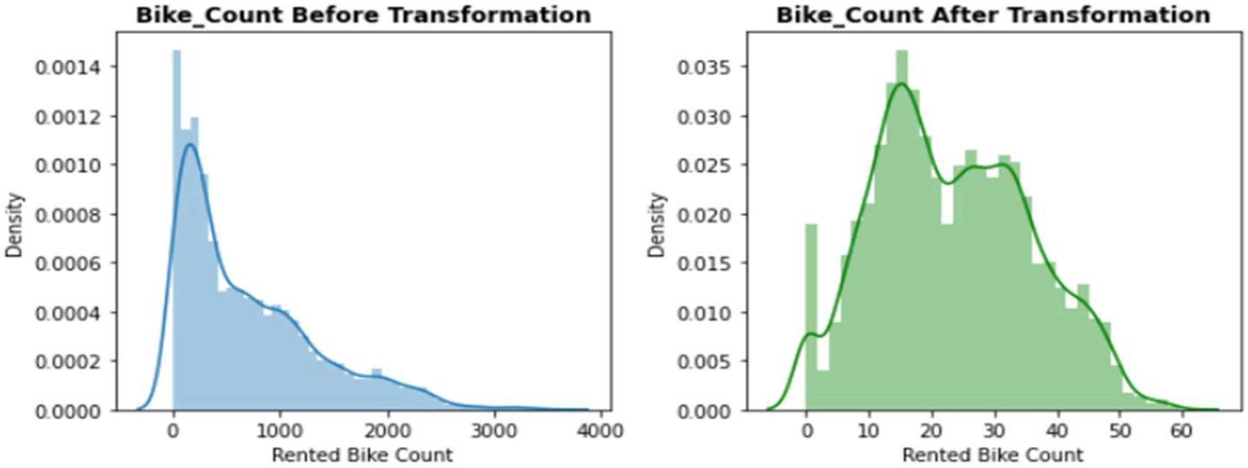


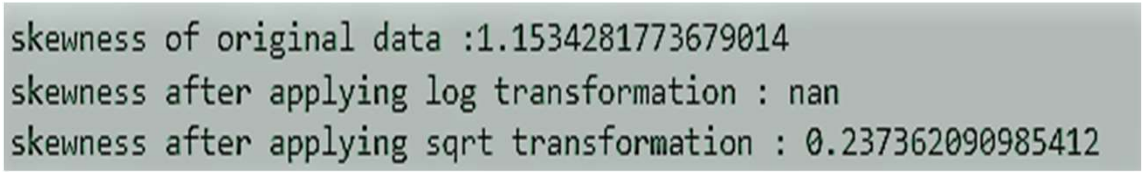
7. From the previous slide graphs hour column is positively correlated with the bike_count , sudden increase in the dataset after 5.

8.temp, sunlight , dew_temp are also positively co-related with the bike_count , increasing in the count as raise in the x –axis observation.

9.Humidity , rain, snow, winter column having a negative correaltion between the bike_count , rest of it having moderate match up b/w the count.

DEPENDENT VARIABLE (bike_count)





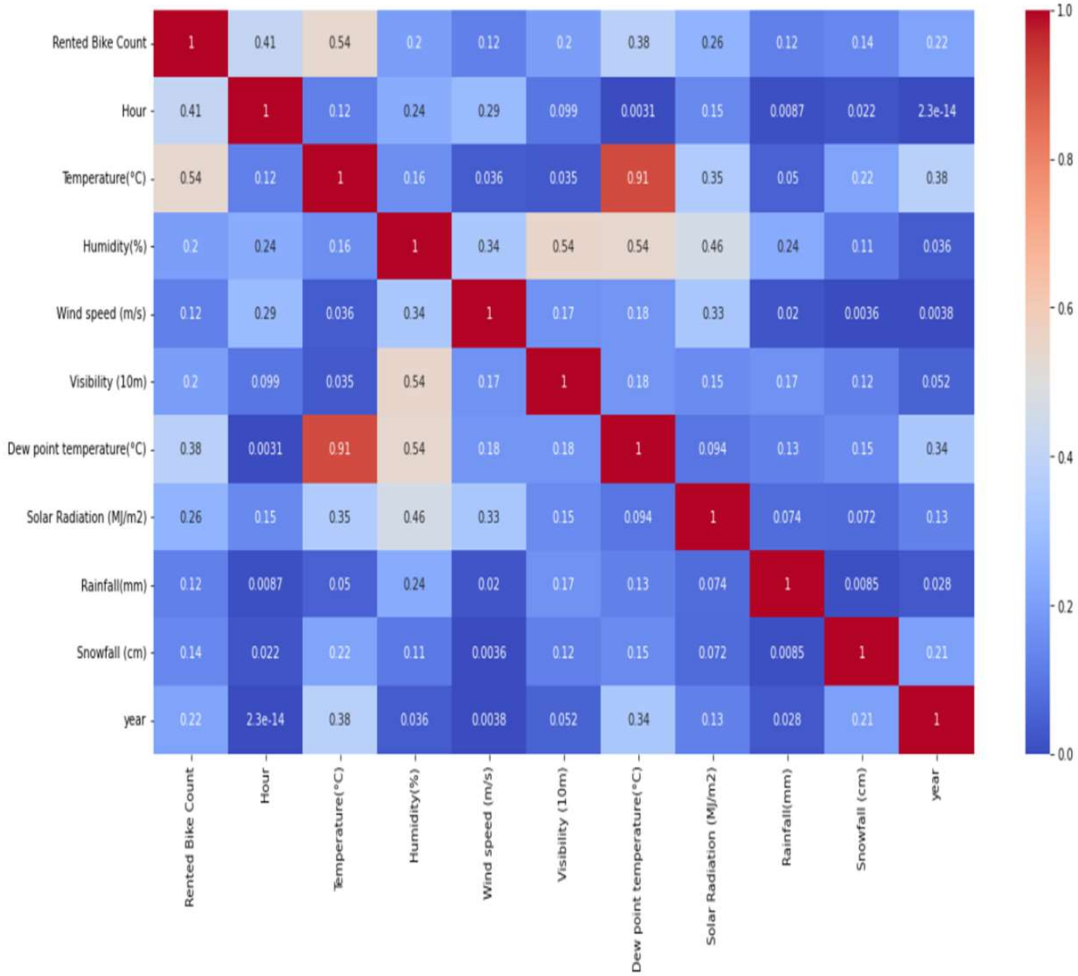
```
skewness of original data :1.1534281773679014
skewness after applying log transformation : nan
skewness after applying sqrt transformation : 0.237362090985412
```

From the original records of the bike_count having high skew value 0.98 which is approaching towards the right , to apply model ,we need to transform our dependent feature to look like a normal distribution.

After applying square root transformation skewness value = 0.15, there is no outlier present in the records , distribution looks like a normal one .

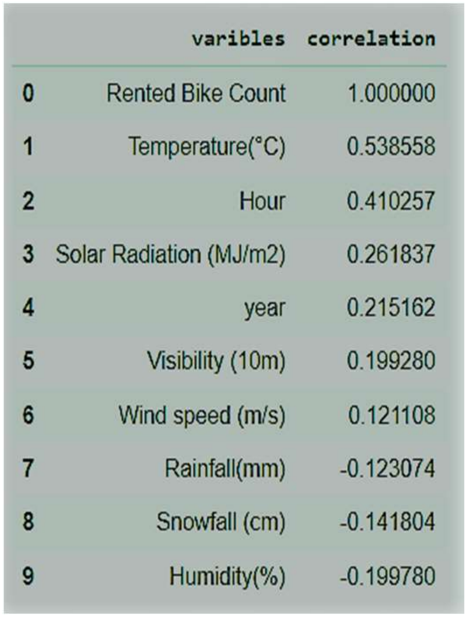
MULTICOLLINEARITY

- ❖ Multicollinearity allows us to look at correlations (that is, how one variable changes with respect to another).
- ❖ In words, the statistical technique that examines the relationship and explains whether, and how strongly, pairs of variables are related to one another is known as correlation.
- ❖ Dew_temp and temp are highly correlated .
- ❖ humidity and visibility are also having much relation between them.
- ❖ We can see one highly correlated feature. Lets treat it by excluding it from dataset and checking the variance inflation factors.

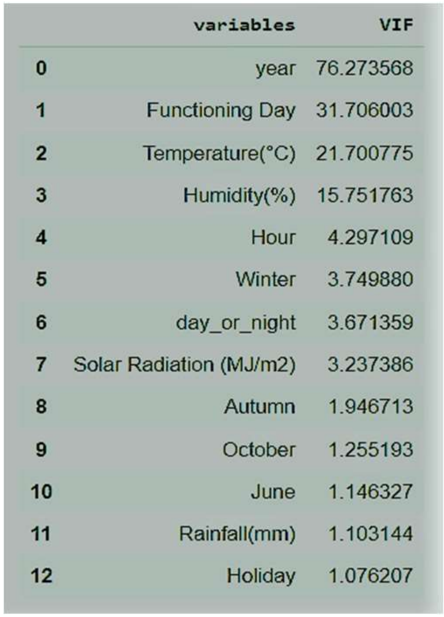


	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	year
Rented Bike Count	1	0.41	0.54	0.2	0.12	0.2	0.38	0.26	0.12	0.14	0.22
Hour	0.41	1	0.12	0.24	0.29	0.099	0.0031	0.15	0.0087	0.022	2.3e-14
Temperature(°C)	0.54	0.12	1	0.16	0.036	0.035	0.91	0.35	0.05	0.22	0.38
Humidity(%)	0.2	0.24	0.16	1	0.34	0.54	0.54	0.46	0.24	0.11	0.036
Wind speed (m/s)	0.12	0.29	0.036	0.34	1	0.17	0.18	0.33	0.02	0.0036	0.0038
Visibility (10m)	0.2	0.099	0.035	0.54	0.17	1	0.18	0.15	0.17	0.12	0.052
Dew point temperature(°C)	0.38	0.0031	0.91	0.54	0.18	0.18	1	0.094	0.13	0.15	0.34
Solar Radiation (MJ/m2)	0.26	0.15	0.35	0.46	0.33	0.15	0.094	1	0.074	0.072	0.13
Rainfall(mm)	0.12	0.0087	0.05	0.24	0.02	0.17	0.13	0.074	1	0.0085	0.028
Snowfall (cm)	0.14	0.022	0.22	0.11	0.0036	0.12	0.15	0.072	0.0085	1	0.21
year	0.22	2.3e-14	0.38	0.036	0.0038	0.052	0.34	0.13	0.028	0.21	1

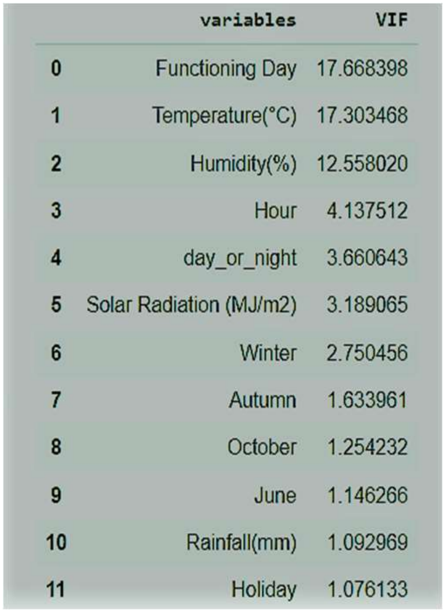
REMOVING MULTICOLLINEARITY



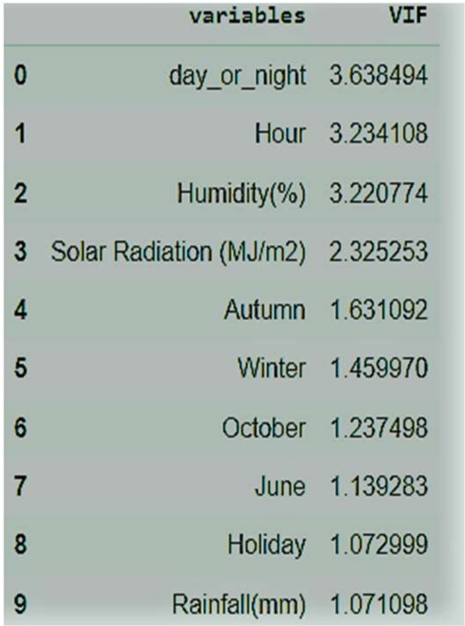
	varibles	correlation
0	Rented Bike Count	1.000000
1	Temperature(°C)	0.538558
2	Hour	0.410257
3	Solar Radiation (MJ/m2)	0.261837
4	year	0.215162
5	Visibility (10m)	0.199280
6	Wind speed (m/s)	0.121108
7	Rainfall(mm)	-0.123074
8	Snowfall (cm)	-0.141804
9	Humidity(%)	-0.199780



	variables	VIF
0	year	76.273568
1	Functioning Day	31.706003
2	Temperature(°C)	21.700775
3	Humidity(%)	15.751763
4	Hour	4.297109
5	Winter	3.749880
6	day_or_night	3.671359
7	Solar Radiation (MJ/m2)	3.237386
8	Autumn	1.946713
9	October	1.255193
10	June	1.146327
11	Rainfall(mm)	1.103144
12	Holiday	1.076207



	variables	VIF
0	Functioning Day	17.668398
1	Temperature(°C)	17.303468
2	Humidity(%)	12.558020
3	Hour	4.137512
4	day_or_night	3.660643
5	Solar Radiation (MJ/m2)	3.189065
6	Winter	2.750456
7	Autumn	1.633961
8	October	1.254232
9	June	1.146266
10	Rainfall(mm)	1.092969
11	Holiday	1.076133



	variables	VIF
0	day_or_night	3.638494
1	Hour	3.234108
2	Humidity(%)	3.220774
3	Solar Radiation (MJ/m2)	2.325253
4	Autumn	1.631092
5	Winter	1.459970
6	October	1.237498
7	June	1.139283
8	Holiday	1.072999
9	Rainfall(mm)	1.071098

- VIF determines the strength of the correlation between the independent variables. It is predicted by taking a variable and regressing it against every other variable. VIF score of an independent variable represents how well the variable is explained by other independent variables.
- Since Summer and Winter can also be classified on the basis of temperature and we already have that feature present. Even if we drop these features the useful information will not be lost. So lets drop them.
- Taking less than VIF of nine for the best input we can give to the model.

❑ MODEL BUILDING

Prerequisites

- Feature Scaling or Standardization: It is a step of Data Pre Processing which is applied to independent variables or features of data. It basically **helps to normalise the data within a particular range**. Sometimes, it also helps in speeding up the calculations in an algorithm.
- Here we used MinMax scaler :***Normalisation*** scales our features to a predefined range (normally the 0–1 range), independently of the statistical distribution they follow. It does this **using the minimum and maximum values** of each feature in our data set, which makes it a bit sensitive to outliers.

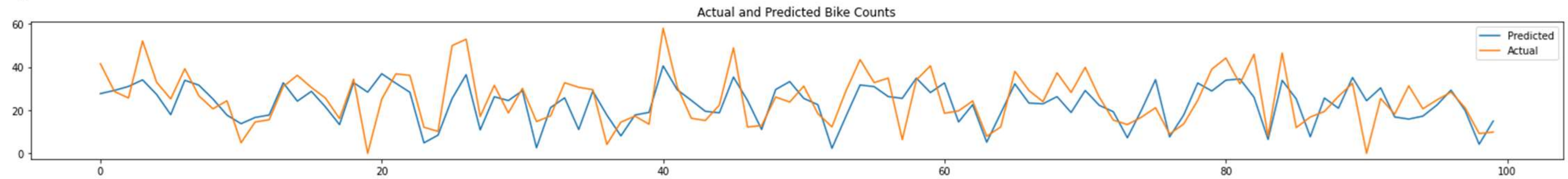
$$Xnormalised = \frac{X - Xmin}{Xmax - Xmin}$$

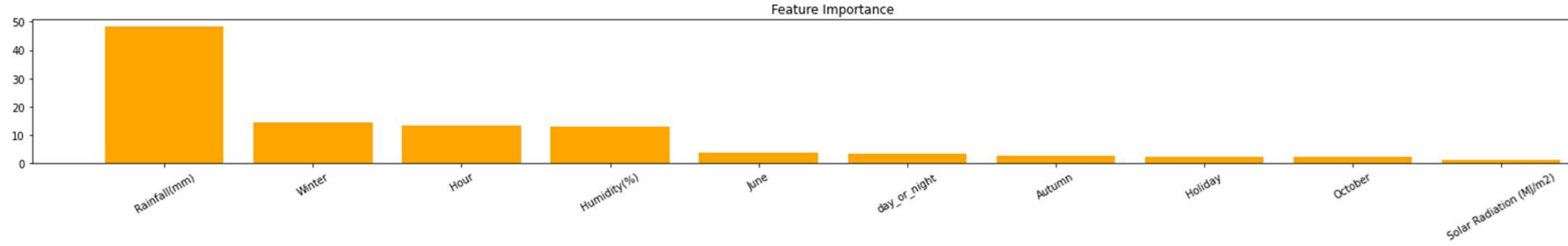
❑ MODEL BUILDING

Prerequisites

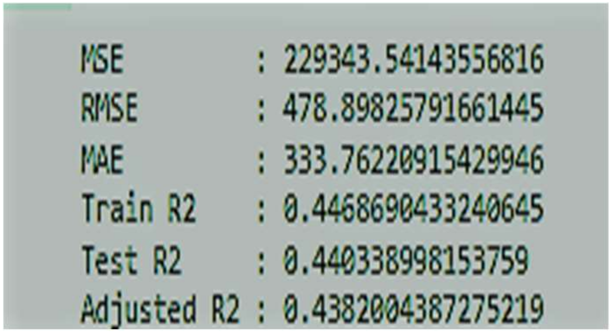
- Defining a new function called **analyse_model** which takes **model, X_train, X_test, y_train, y_test** tends to give the value of the evaluation matrix like MSE,RMSE,MAE,TARIN R2 ,TEST R2 , ADJUSTED R2.Continuing to that it also be liable to print the feature importance of the algorithm used.
- In linear regression we used to print the feature importance with the help of beta coefficient which be prone ot give the exact feature importance used case like other algorithm.
- range of values for hyperparameters such as Number of trees: n_estimators=[100,120]
- Maximum depth of trees: [1-20]
- Minimum number of samples required to split a node:min_samples_split=[50,100,150]
- Minimum number of samples required at each leaf node:min_sample_leaf=[40,50]
- learning rate : Eta=[0.3,0.2,0.1]

LINEAR REGRESSION


Actual and Predicted Bike Counts

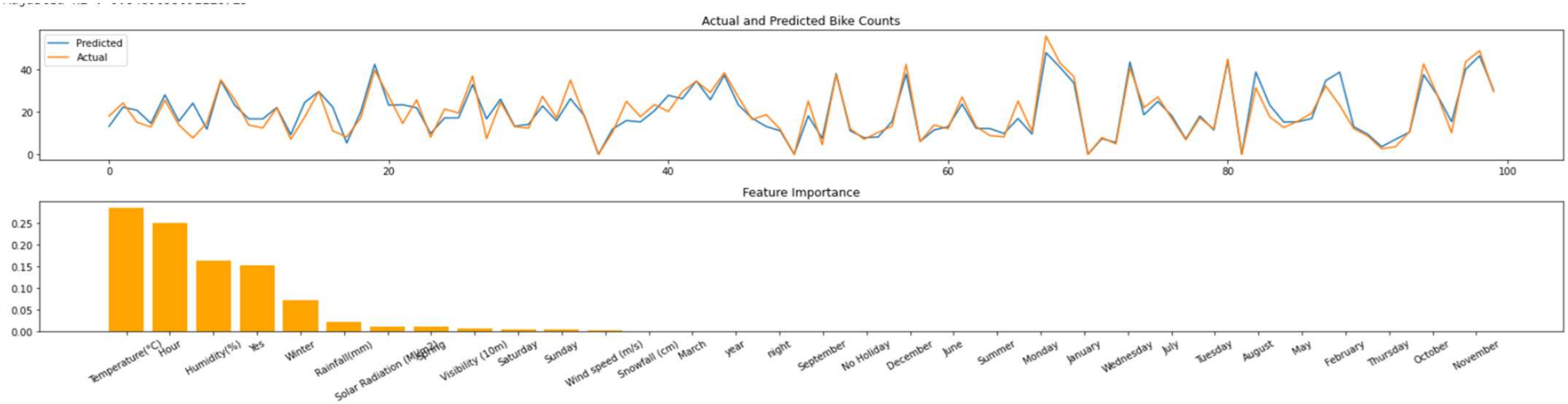

Feature Importance

- Linear Regression Supervised machine learning models are those where we use the training data to build the model and then test the accuracy of the model using the loss function.
- Here we can accommodate linear regression model is unable to capture the evaluation matrix.
- rainfall is best coefficient which is mostly responsible for the model performance.

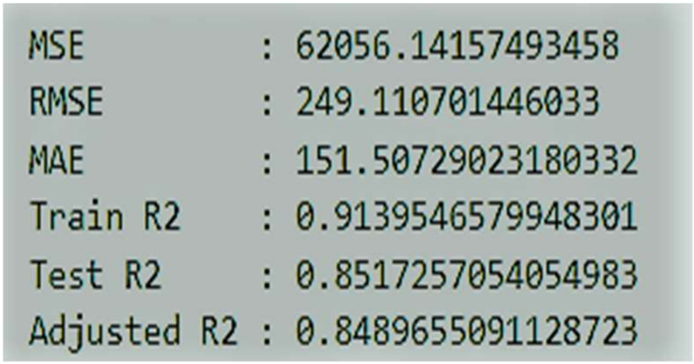


```
MSE          : 229343.54143556816
RMSE         : 478.8982579161445
MAE          : 333.76220915429946
Train R2     : 0.4468690433240645
Test R2      : 0.440338998153759
Adjusted R2  : 0.4382004387275219
```

DECISION TREE

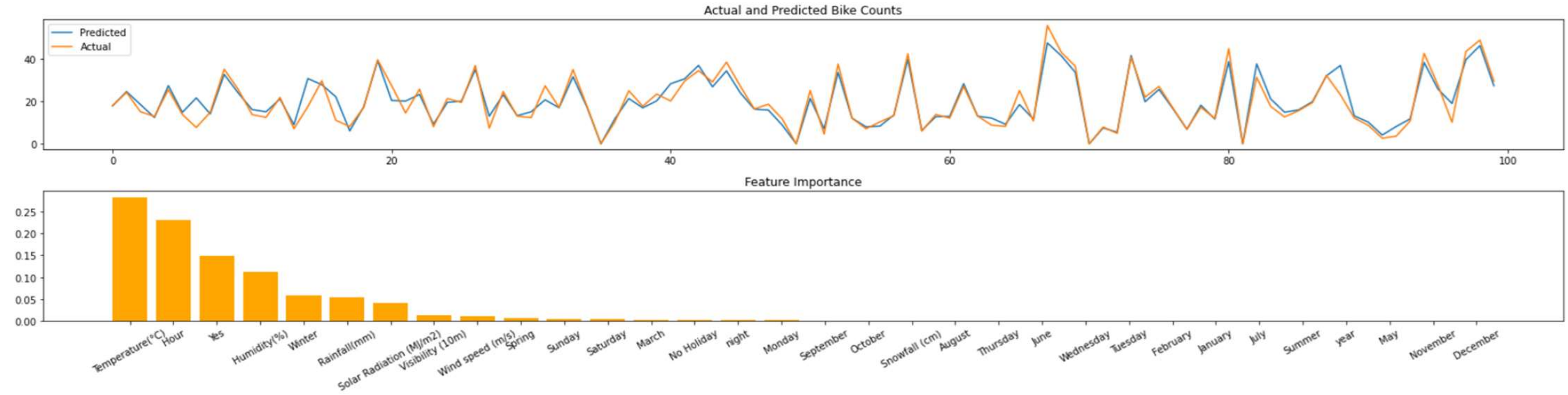


- Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance.
- Decision tree performs well better than the linear reg with a test r2 score more than 75%.
- Temp attended to be the best feature_importance for the model performance .
- DecisionTreeRegressor(max_depth=14, min_samples_leaf=8, random_state=0)

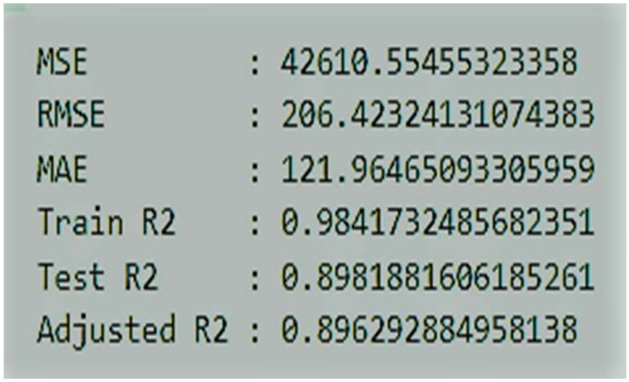


```
MSE         : 62056.14157493458
RMSE        : 249.110701446033
MAE         : 151.50729023180332
Train R2    : 0.9139546579948301
Test R2     : 0.8517257054054983
Adjusted R2 : 0.8489655091128723
```

RANDOM FOREST REGRESSOR

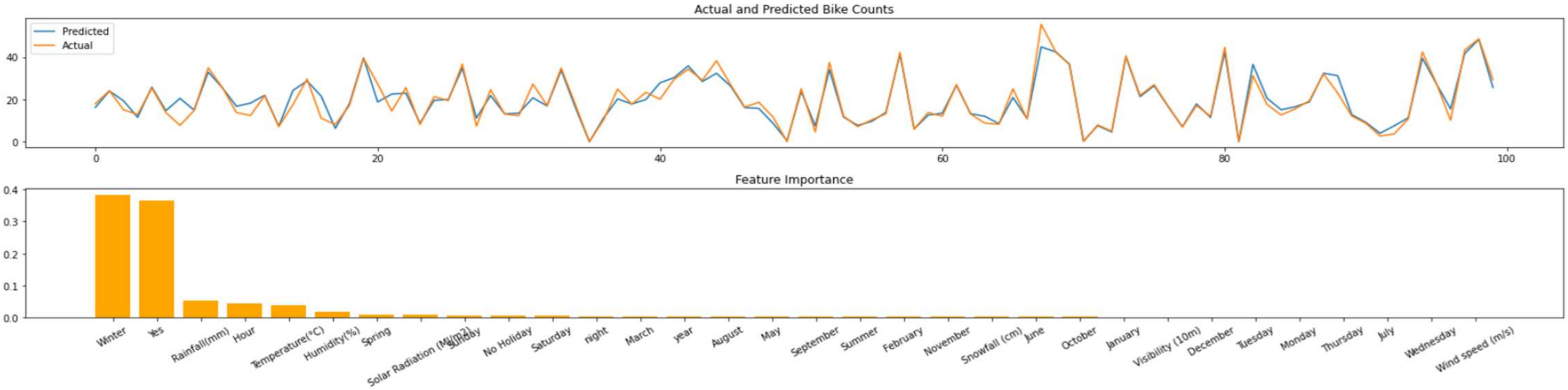


- Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the training set.
- Random forest also performs well in both test and train data with a r2 score of train 98% and in test r2 score with 89% .
- Talking in terms of the feature importance , temp emerges as the contributor and is also became the highly important for the model performance.
- RandomForestRegressor(max_depth=19, n_estimators =100)

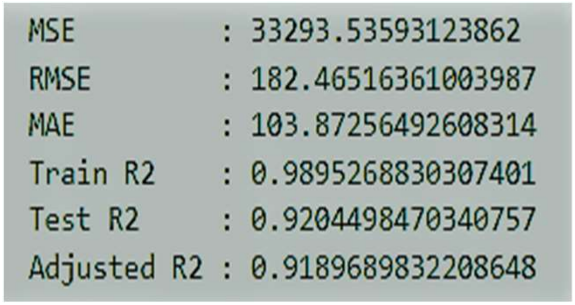


```
MSE         : 42610.55455323358
RMSE        : 206.4232413107438З
MAE         : 121.96465093305959
Train R2    : 0.9841732485682351
Test R2     : 0.8981881606185261
Adjusted R2 : 0.896292884958138
```

XGBOOST REGRESSOR

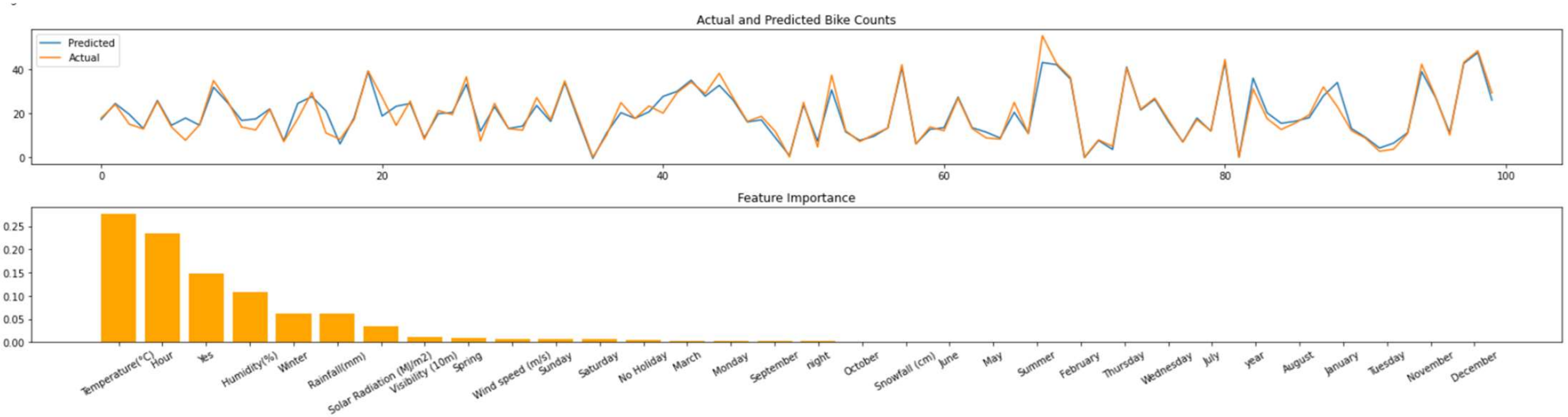


- The objective function contains loss function and a regularization term. It tells about the difference between actual values and predicted values, i.e how far the model results are from the real values.
- XGBoost regressor emerges as the best model according to the evaluation matrix score both in the train and test ,winter becomes as the best feature and is responsible for the model performance.
- XGBRegressor(eta=0.3, max_depth=9, n_estimators=120)

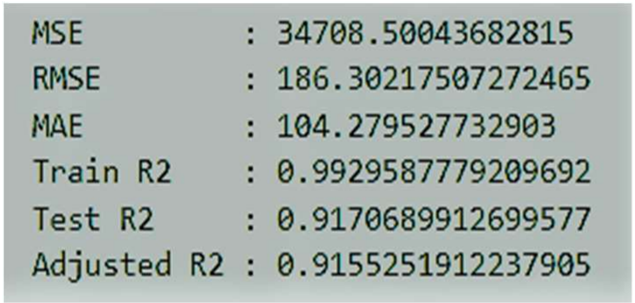


```
MSE          : 33293.53593123862
RMSE         : 182.46516361003987
MAE          : 103.87256492608314
Train R2     : 0.9895268830307401
Test R2      : 0.9204498470340757
Adjusted R2  : 0.9189689832208648
```

GRADIENT BOOSTING REGRESSOR



- Gradient boosting re-defines boosting as a numerical optimisation problem where the objective is to minimise the loss function of the model by adding weak learners using gradient descent.
- Here temperature became the best feature which responsible for the model performance.
- GradientBoostingRegressor(max_depth=9, n_estimators=120)



```
MSE         : 34708.50043682815
RMSE        : 186.30217507272465
MAE         : 104.279527732903
Train R2    : 0.9929587779209692
Test R2     : 0.9170689912699577
Adjusted R2 : 0.9155251912237905
```

CONCLUSION

1. Rainfall is the most influencing feature and winter is at the second place for Linear Regressor.

2. Temperature is the most important feature and Hour is at second place for Decision Tree, Random Forest and Gradient Boosting Regressor.

3. Winter is the most important feature and Functioning day[yes] is the second most for XGBoost Regressor.

4. RMSE Comparisons:

   A. Linear Regressor RMSE: 478.89

   B. Decision Tree Regressor RMSE: 249.11

   C. Random Forest Regressor RMSE: 207.06

   D. XGBoost Regressor RMSE: 184.27

   E. Gradient Boosting Regressor RMSE: 186.55

5. The feature temperature is on the top list for all the regressors except XGBoost and Linear Regessor.

6. XGBoost and Linear Regressor is acting different from all the regressors as it is considering whether it is winter or not. And is it a working day or not. Though winter is also a function of temperature only but it seems this trick of XGBoost is giving better results.

7. XGBoost Regressor has the Least Root Mean Squared Error. but also, GradientBoostingRegressor is very close to XGBoost, since both are the boosting algorithm. So, any one of them can be considered as the best model for given problem.