Capstone Project

Seoul Bike Sharing Demand Prediction –
Supervised ML Regression

KISHOR SHIVAJI PATIL
Data Science Trainee,
AlmaBetter, Bangalore.

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.

PROJECT 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.

- Therefore, to strive the business With more Profit , it has to be always ready tosupply no. of bikes at different locations to fulfill the demand.

- My project goal is to predict bike count values that can be a handy solution to meet all demands based on given Dataset.

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 contain 8760 rows and 14 columns.
- Three categorical features ‘Seasons’, ‘Holiday’, & ‘Functioning Day’.
- One Datetime column ‘Date’.
- We have some numerical type variables such as temperature, humidity, wind, visibility, dew point temp, solar radiation, rainfall, snowfall which shows the environmental conditions for that particular hour of the day.

DATA SUMMARY

- There are No Missing Values present
- There are No Duplicate values present
- There are No null values.
- The dependent variable is 'bike count' which we need to make predictions on.
- The dataset shows hourly rental data for one year (1 December 2017 to 31 November 2018) (365 days).
- We changed the name of some features for our convenience, they are as follows - ‘date', 'Bike_Count', 'Hour’, ‘temp’, ‘humidity', ‘wind', ‘visibility’, ‘dew_temp’, ‘sunlight’, rain', ‘snow', ‘seasons', ‘holiday', ‘functioning_day’.

FEATURE TYPES

FEATURES

NUMERIC

1.Hour
2.temp
3. humidity
4.wind
5.dew_temp
5.sunlight
6.rain
7.snow

CATEGORICAL

1.season
2.holiday
3.Functioning day
4.timeshift

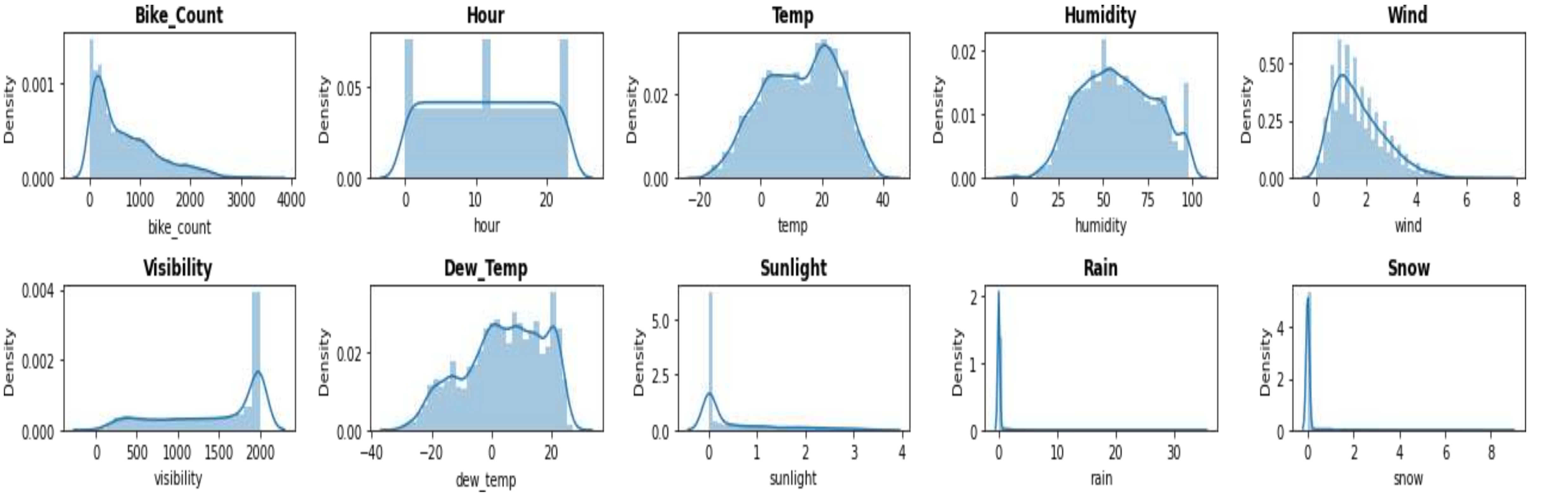
TARGET VARIABLE

BIKE COUNT

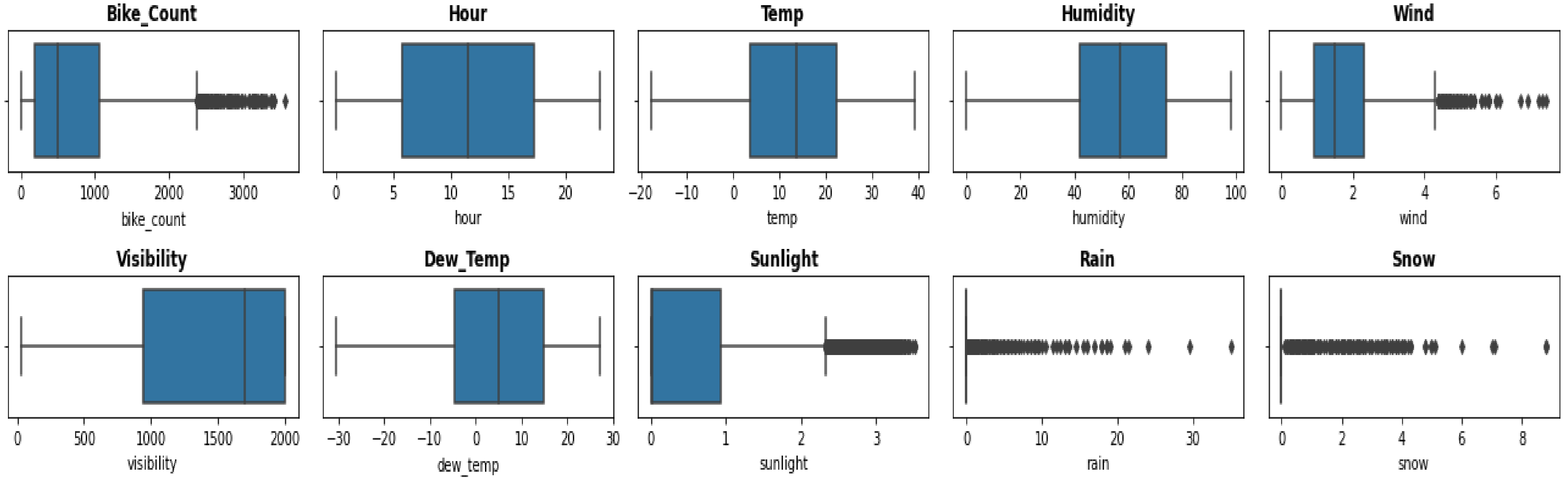
FEATURE SUMMARY

- 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)

VARIABLE DISTRIBUTIONS



CHECKING OUTLIERS



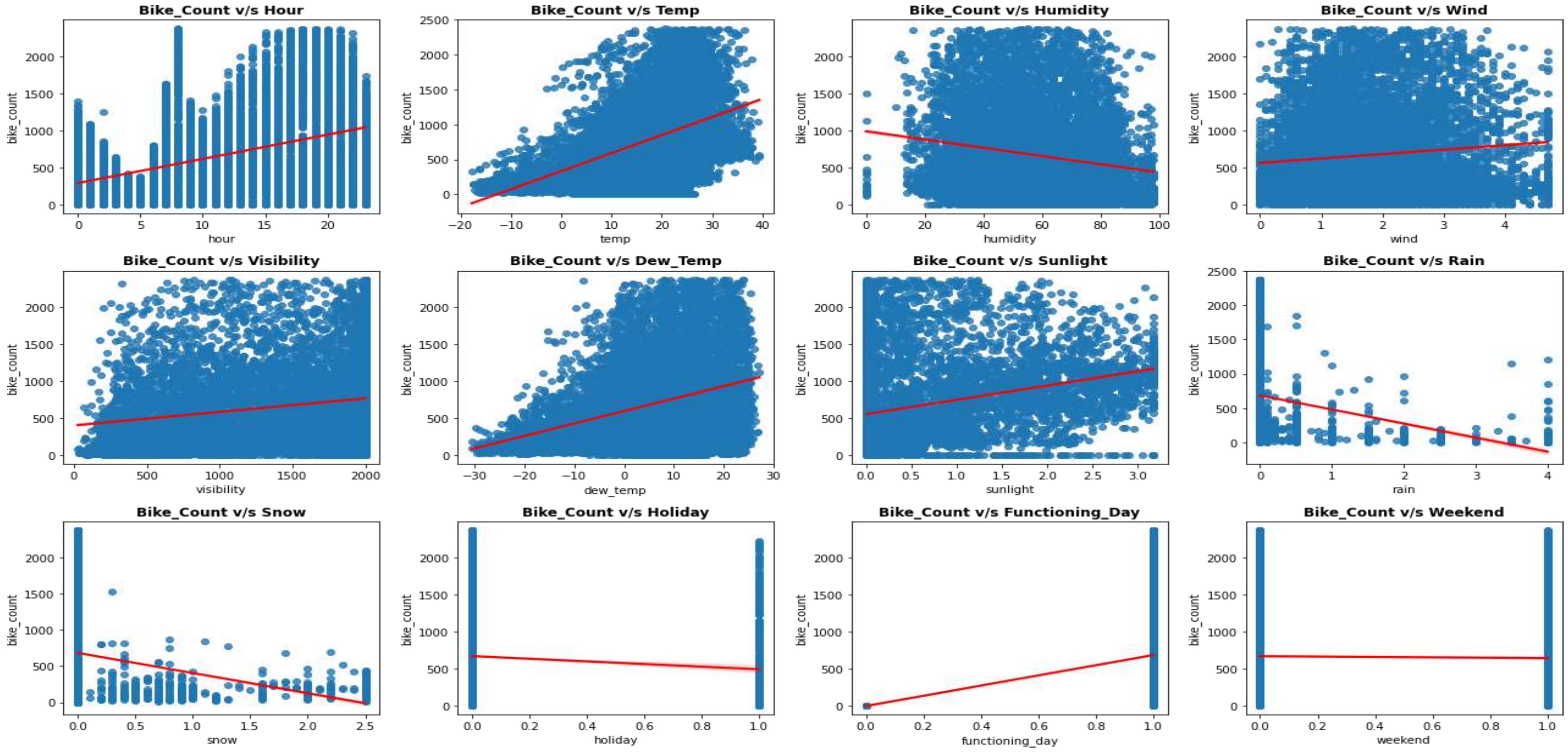
- We see outliers in some columns like Sunlight, Wind, Rain and Snow but lets not treat them because they may not be outliers as snowfall, rainfall etc. themselves are rare event in some countries.
- We treated the outliers in the target variable by capping with IQR limits.

MANIPULATION OF DATASET

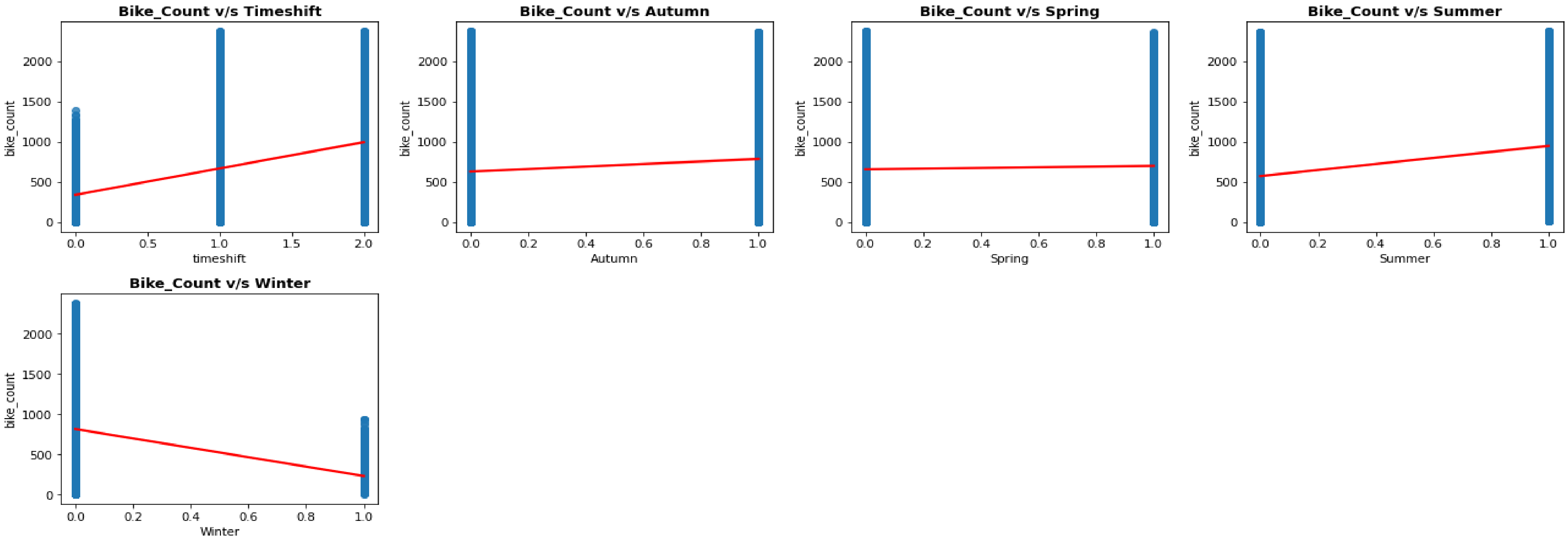
```
Spring    2208
Summer    2208
Autumn    2184
Winter    2160
Name: season, dtype: int64

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
```

- Added new feature named **weekend** that shows whether it's a weekend or not**.** Here Saturday and Sunday means 1 else 0.
- Added one more new feature named **timeshift** based on time intervals. It has three values Night, Day and Evening.
- Dropped the date column because we already extracted some useful features from that column.
- Defined a label encoder to replace the string values in the columns with some numeric values.
  - Replaced **holiday** with **1** and **No holiday** with **0.**
  - Replaced **functioning_day** column **Yes** with **1** and **No** with **0**
  - In the **timeshift** column we replaced **night** with **0**, **day** with **1** and **evening** with **2**.
- Created dummy features from the season column named **summer**, **autumn**, **spring** and **winter** with one hot encoding.

CHECKING LINEARITY IN DATA

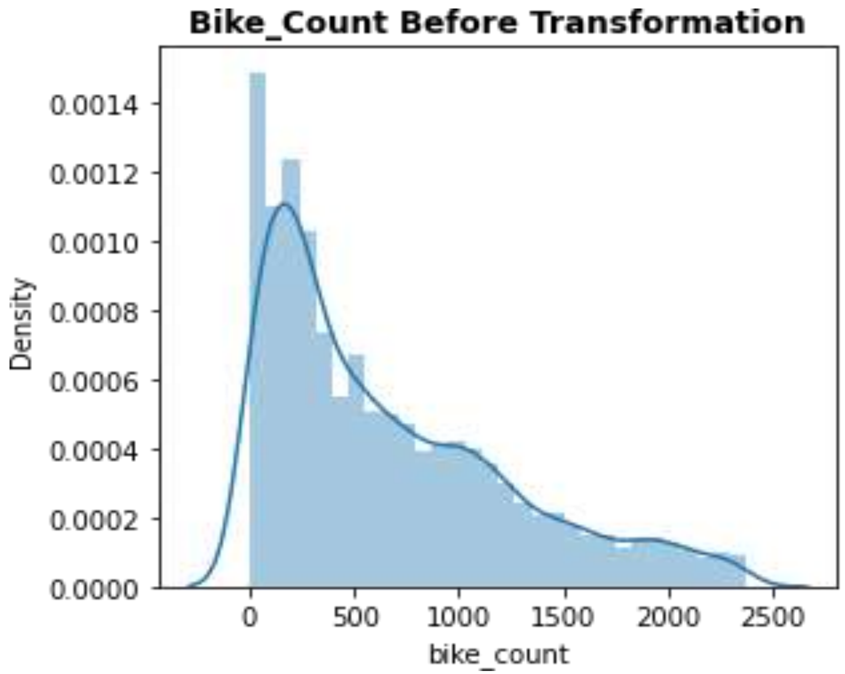


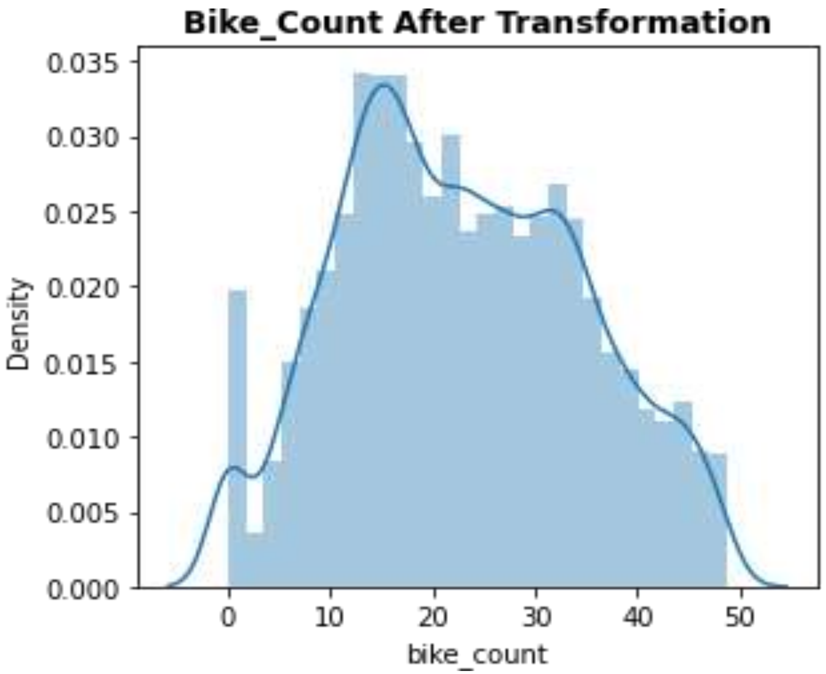
CHECKING LINEARITY IN DATA



- From the visualizations we observed that hour, temp, sunlight , dew_temp is positively correlated with the bike_count.
- Humidity , rain, snow, winter features are having a negative correlation with the bike_count.
- Some features are also showing close to zero correlation with the target variable as the regression line is not inclined.

DEPENDENT VARIABLE





- Earlier the distribution of the target variable was positively skewed with a skewness value of 0.983. I tried to make this distribution somewhat close to normal distribution.
- First I applied log transformation, but it did not give the desired results, Then I finally applied square root transformation. We got the favourable results, the skewness value was dropped to 0.153, which is comparatively closer to the normal distribution.

MULTICOLLINEARITY ANALYSIS

	bike_count	hour	temp	humidity	wind	visibility	dew_temp	sunlight	rain	snow	holiday	functioning_day	weekend	timeshift	Autumn	Spring	Summer	Winter
bike_count	1	0.39	0.53	0.19	0.1	0.19	0.37	0.29	0.18	0.16	0.066	0.21	0.02	0.43	0.12	0.032	0.28	0.43
hour	0.39	1	0.12	0.24	0.29	0.099	0.0031	0.15	0.0016	0.023	1.4e-16	0.0054	2.3e-17	0.94	2e-15	1.2e-15	8.6e-16	1.7e-15
temp	0.53	0.12	1	0.16	0.036	0.035	0.91	0.35	0.061	0.25	0.056	0.05	0.013	0.11	0.06	0.008	0.67	0.74
humidity	0.19	0.24	0.16	1	0.34	0.54	0.54	0.46	0.33	0.095	0.05	0.021	0.037	0.21	0.028	0.016	0.19	0.24
wind	0.1	0.29	0.036	0.34	1	0.17	0.18	0.34	0.038	0.0024	0.023	0.0046	0.021	0.26	0.13	0.083	0.064	0.11
visibility	0.19	0.099	0.035	0.54	0.17	1	0.18	0.15	0.24	0.12	0.032	0.026	0.031	0.091	0.12	0.19	0.062	0.0086
dew_temp	0.37	0.0031	0.91	0.54	0.18	0.18	1	0.094	0.17	0.18	0.067	0.053	0.029	0.0042	0.063	0.0021	0.65	0.72
sunlight	0.29	0.15	0.35	0.46	0.34	0.15	0.094	1	0.1	0.079	0.0048	0.0077	0.0082	0.084	0.031	0.08	0.13	0.18
rain	0.18	0.0016	0.061	0.33	0.038	0.24	0.17	0.1	1	0.00061	0.017	0.0092	0.02	0.0025	0.019	0.041	0.059	0.082
snow	0.16	0.023	0.25	0.095	0.0024	0.12	0.18	0.079	0.00061	1	0.0091	0.036	0.038	0.018	0.044	0.11	0.11	0.27
holiday	0.066	1.4e-16	0.056	0.05	0.023	0.032	0.067	0.0048	0.017	0.0091	1	0.028	0.0063	1.3e-16	0.015	0.045	0.074	0.1
functioning_day	0.21	0.0054	0.05	0.021	0.0046	0.026	0.053	0.0077	0.0092	0.036	0.028	1	0.024	0.0058	0.25	0.038	0.11	0.11
weekend	0.02	2.3e-17	0.013	0.037	0.021	0.031	0.029	0.0082	0.02	0.038	0.0063	0.024	1	2.1e-17	0.008	0.01	0.01	0.012
timeshift	0.43	0.94	0.11	0.21	0.26	0.091	0.0042	0.084	0.0025	0.018	1.3e-16	0.0058	2.1e-17	1	8.5e-16	1.3e-16	6.2e-16	9.9e-17
Autumn	0.12	2e-15	0.06	0.028	0.13	0.12	0.063	0.031	0.019	0.044	0.015	0.25	0.008	8.5e-16	1	0.33	0.33	0.33
Spring	0.032	1.2e-15	0.008	0.016	0.083	0.19	0.0021	0.08	0.041	0.11	0.045	0.038	0.01	1.3e-16	0.33	1	0.34	0.33
Summer	0.28	8.6e-16	0.67	0.19	0.064	0.062	0.65	0.13	0.059	0.11	0.074	0.11	0.01	6.2e-16	0.33	0.34	1	0.33
Winter	0.43	1.7e-15	0.74	0.24	0.11	0.0086	0.72	0.18	0.082	0.27	0.1	0.11	0.012	9.9e-17	0.33	0.33	0.33	1

HANDLING MULTICOLLINEARITY

	variables	VIF
0	dew_temp	119.298136
1	Summer	116.141121
2	Spring	112.673201
3	Autumn	110.725563
4	Winter	107.844468
5	temp	90.833188
6	humidity	21.238433
7	hour	8.781649
8	timeshift	8.555039
9	sunlight	2.078721
10	visibility	1.691780
11	wind	1.313277
12	rain	1.179250
13	snow	1.147787
14	functioning_day	1.081776
15	holiday	1.023520
16	weekend	1.007038

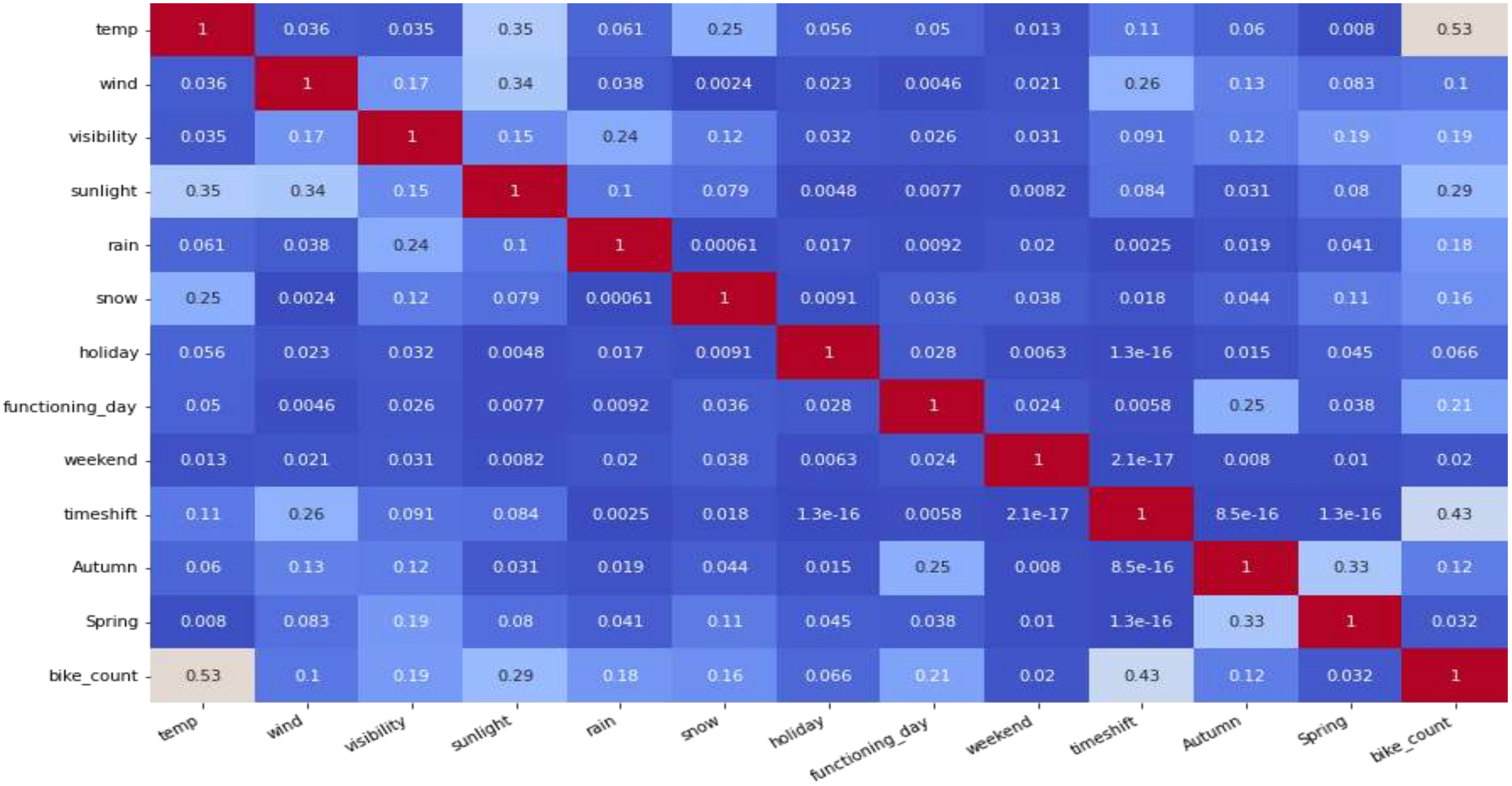
- Multicollinearity allows us to look at correlations (that is, how one variable changes with respect to another).
- Dew_temp and temp are highly correlated. Hour and timeshift are also highly corelated.
- We can see some highly correlated features. Lets treat them by excluding them from dataset and checking the variable inflation factors(VIF).
- 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.

HANDLING MULTICOLLINEARITY

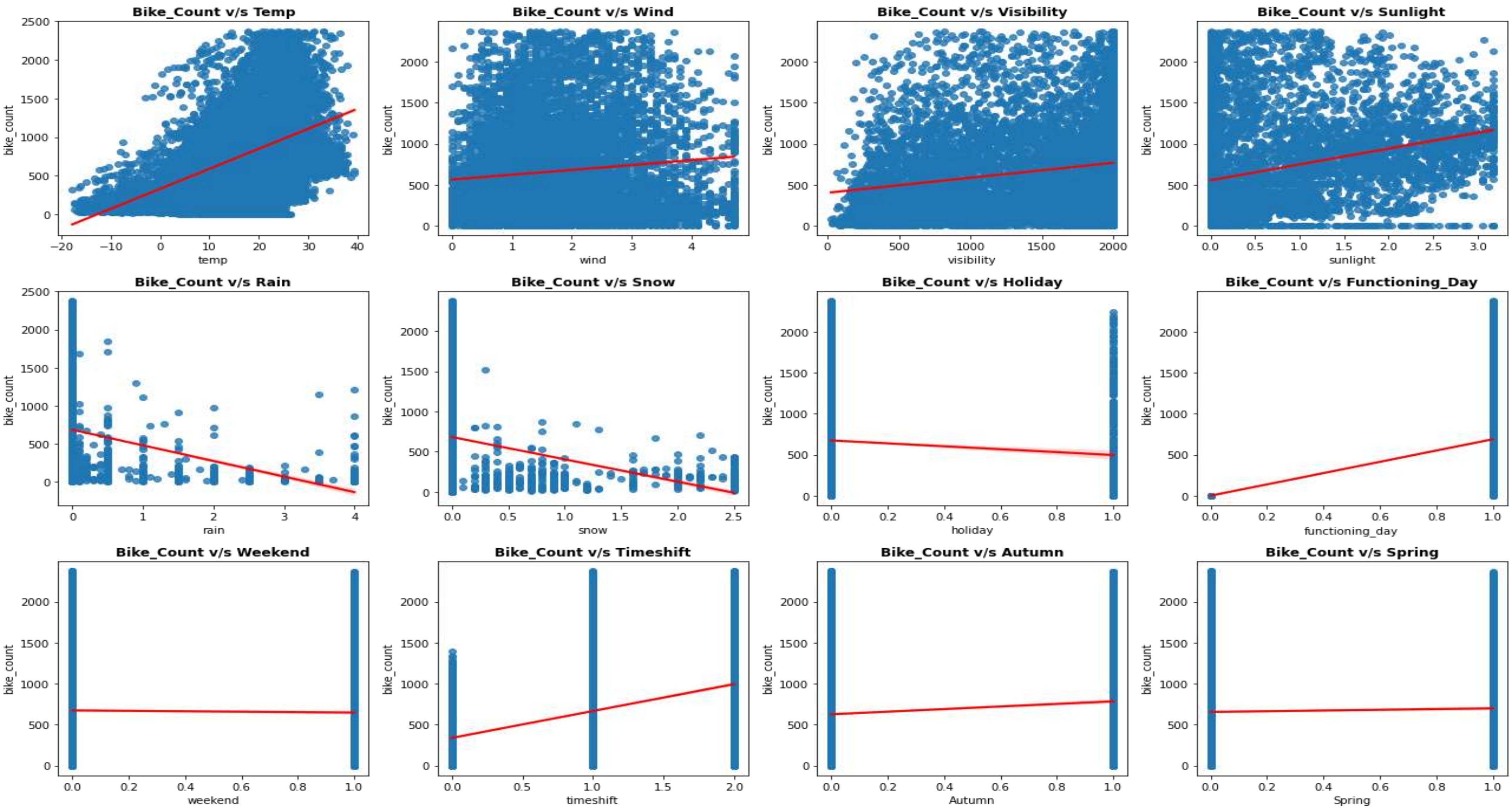
- Since Summer and Winter can also be classified on the basis of temperature and we already have that feature present. Even if we drop these features the useful information will not be lost. So we dropped them.
- We continued to exclude the features with VIF > 10 and finally we obtained the following results.

	variables	VIF
0	functioning_day	8.973136
1	visibility	6.903425
2	wind	4.784533
3	timeshift	2.956516
4	temp	2.685255
5	sunlight	1.944365
6	Spring	1.528702
7	Autumn	1.468795
8	weekend	1.396051
9	snow	1.131983
10	rain	1.110783
11	holiday	1.056152

UPDATED HEATMAP



UPDATED DATASET



MODEL BUILDING PREREQUISITES

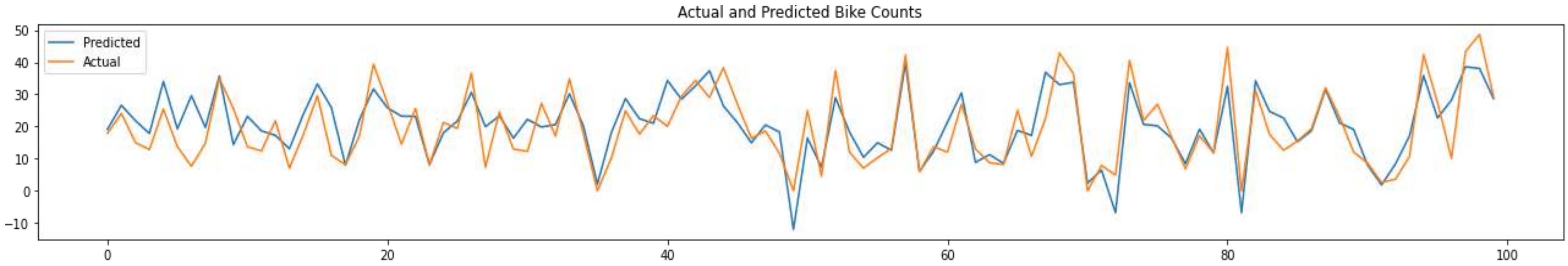
- Feature Scaling or Standardization: It is a step of Data Pre Processing which is applied to independent variables or features of data. It basically **helps to normalise the data within a particular range**. Sometimes, it also helps in speeding up the calculations in an algorithm.

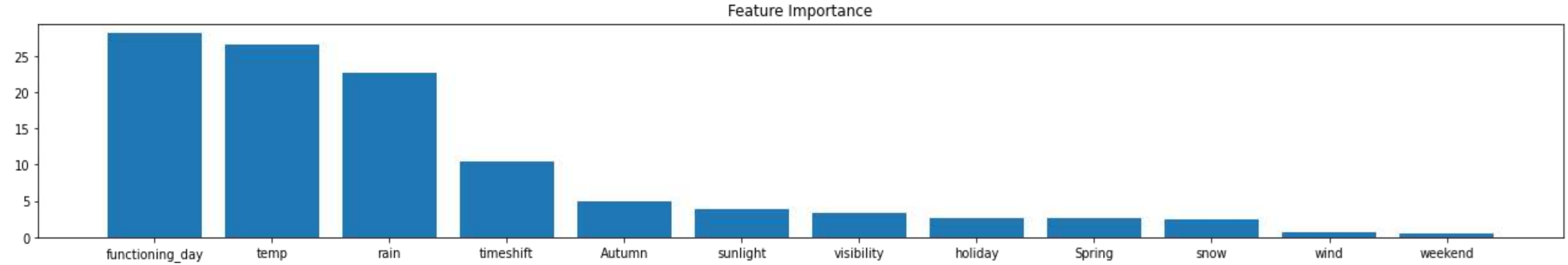
- Here we used MinMax scaler :***Normalisation*** scales our features to a predefined range (normally the 0–1 range), independently of the statistical distribution they follow. It does this **using the minimum and maximum values** of each feature in the dataset.

MODEL BUILDING PREREQUISITES

- Defining a new function called **analyse_model** which takes **model, X_train, X_test, y_train, y_test** and prints evaluation matrix like MSE, RMSE, MAE, TRAIN R2, TEST R2 , ADJUSTED R2. Also plots the feature importance based on the algorithm used.
- We also defined some range of values for hyperparameters such as:
  - Number of trees: n_estimators=[50,100,150]
  - Maximum depth of trees: [6,8,10]
  - Minimum number of samples required to split a node: [50,100,150]
  - Minimum number of samples required at each leaf node: [40,50]
  - learning rate : Eta=[0.05, 0.08, 0.1]

LINEAR REGRESSION

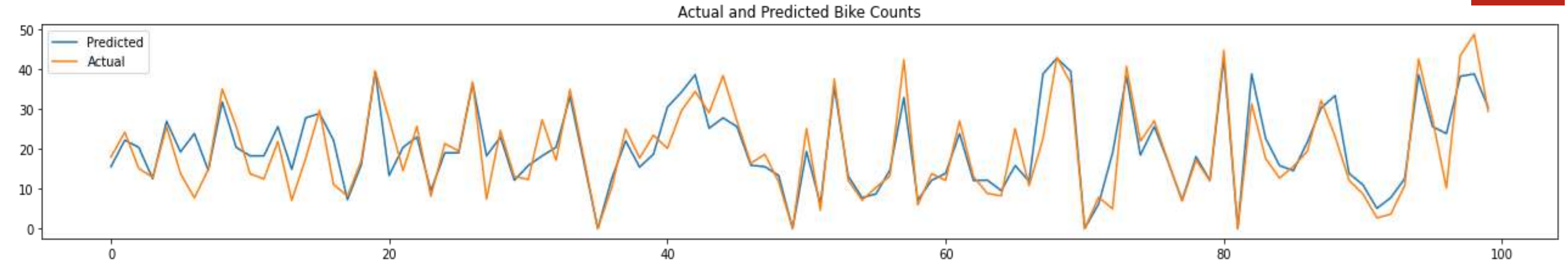


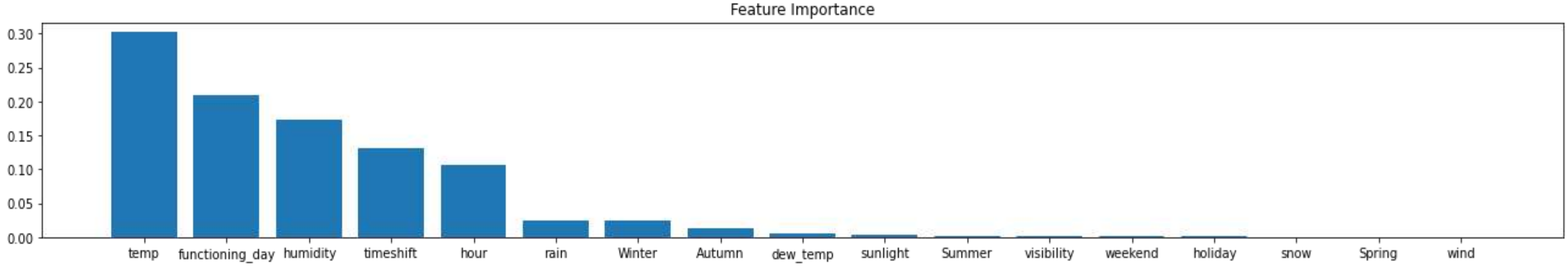


- We plotted the absolute values of the beta coefficients which can be seen parallel to the feature importance of tree based algorithms.
- Since the performance of simple linear model is not so good. We experimented with some complex models.

```
MSE : 137241.3084686744
RMSE : 370.46094054390454
MAE : 254.74045552944642
Train R2 : 0.5837621350247335
Test R2 : 0.5924062591863408
Adjusted R2 :  0.5895936514291448
```

DECISION TREE

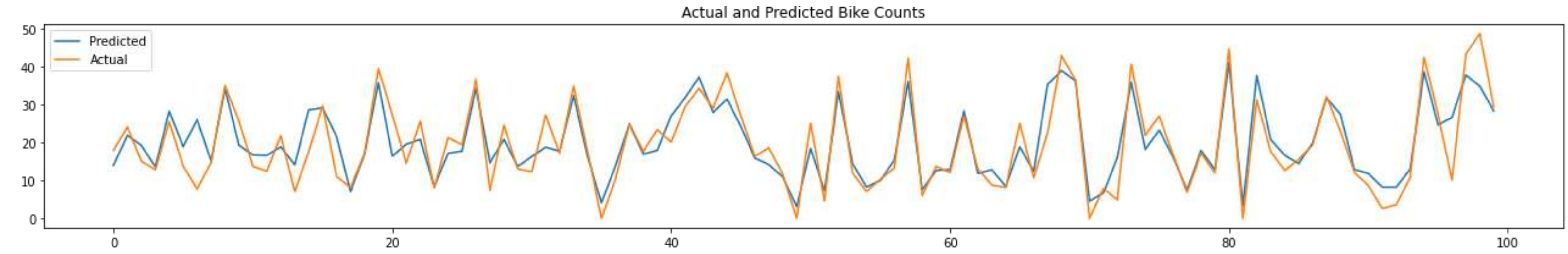


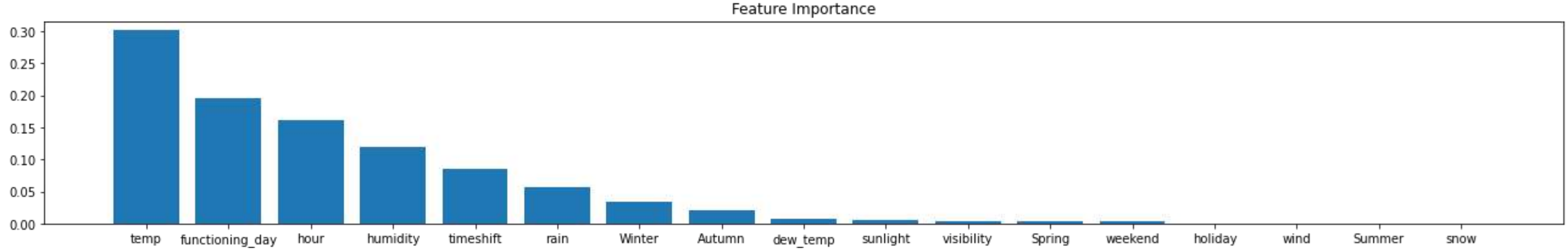


- DecisionTreeRegressor(max_depth=10, min_samples_leaf=40, min_samples_split=50, random_state=1)
- Decision tree performs well better than the linear reg with a test r2 score more than 70%.

```
MSE : 91524.53332018365
RMSE : 302.53021885455286
MAE : 188.5071046099557
Train R2 : 0.7598960015979025
Test R2 : 0.7281807691252598
Adjusted R2 :  0.7255158747049192
```

RANDOM FOREST REGRESSOR

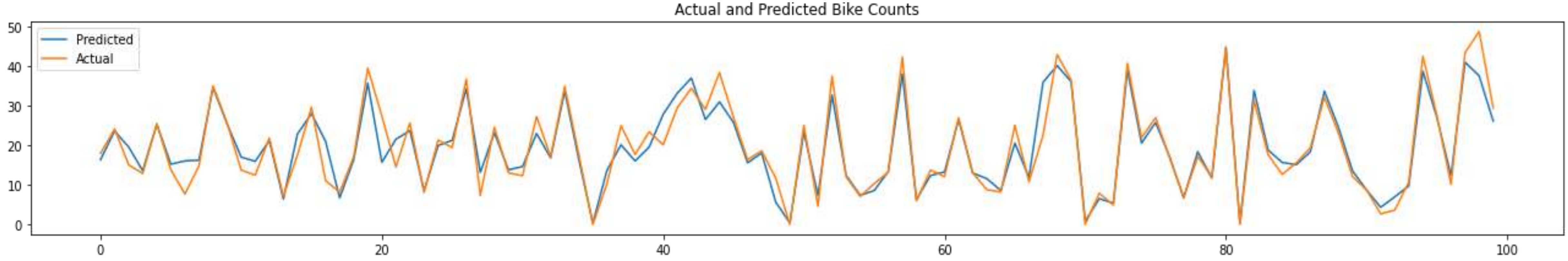


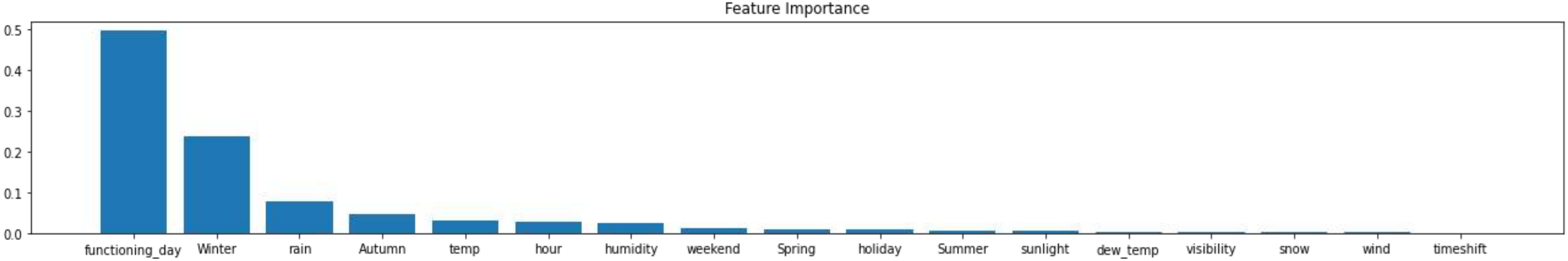


- RandomForestRegressor(max_depth=10, min_samples_leaf=40, min_samples_split=50, random_state=2)
- Random forest also performs well in both test and train data with a r2 score 77% on train data and around 75% on the test data.

```
MSE : 84111.62102061682
RMSE : 290.0200355503337
MAE : 178.3082494922640З
Train R2 : 0.7738012599759755
Test R2 : 0.750196419429182
Adjusted R2 :  0.74774736471774
```

XGBOOST REGRESSOR

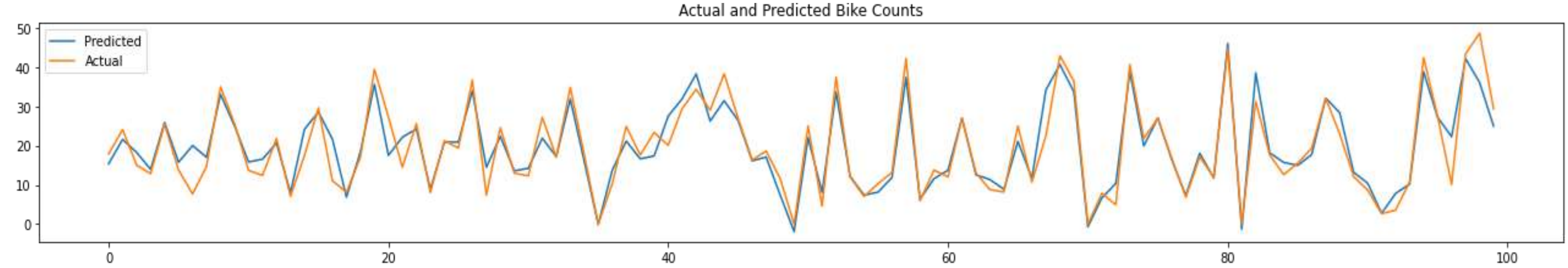


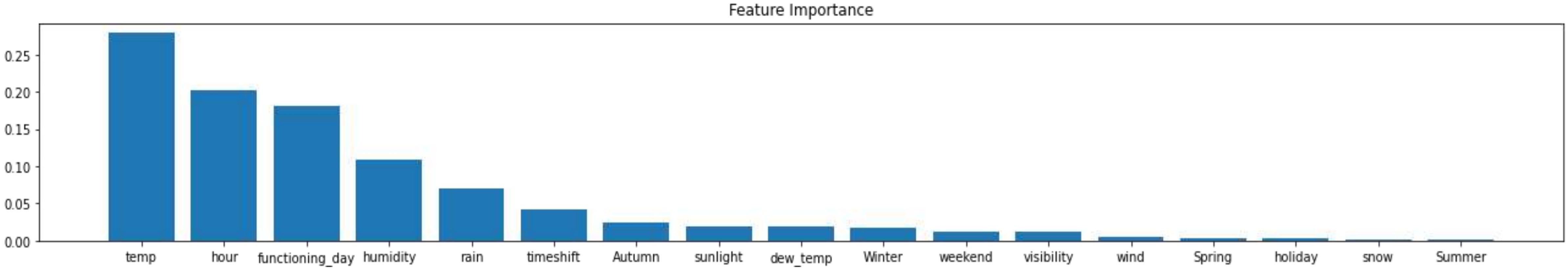


- XGBRegressor(eta=0.05, max_depth=8, min_samples_leaf=40, min_samples_split=50, n_estimators=150, random_state=3, silent=True)
- XGBoost regressor emerges as the best model according to the evaluation matrix score both in the train and test.

```
MSE : 58913.99867290589
RMSE : 242.72206054025227
MAE : 134.1361227702089
Train R2 : 0.961794302333021
Test R2 : 0.82503098102666
Adjusted R2 :  0.8233155984877119
```

GRADIENT BOOSTING REGRESSOR





- GradientBoostingRegressor(max_depth=10, min_samples_leaf=50, min_samples_split=50, n_estimators=150, random_state=4)
- We experimented this boosting algorithm in order to enhance the performance but we found out that its performance is closely equal to the XGBoost model only.

```
MSE : 61590.84383506893
RMSE : 248.17502661442174
MAE : 141.1977249022155
Train R2 : 0.9078337007467008
Test R2 : 0.8170810033894738
Adjusted R2 :  0.8152876798932921
```

CONCLUSION

- The independent variables in data does not have a good linear relation with the target variable so the simple linear model was not performing good on this data.
- Tree based Algorithms seem to perform well in this case.
- Functioning day is the most influencing feature and temperature is at the second place for LinearRegressor.
- Temperature is the most important feature for DecisionTree, RandomForest and Gradient Boosting Regressor.
- Functioning day is the most important feature and Winter is the second most for XGBoost Regressor.
- The feature temperature is on the top list for all the regressors except XGBoost.
- XGBoost is acting different from all the regressors as it is considering whether it is winter or not. And is it a Functioning day or not. Though winter is also a function of temperature only but it seems this trick of XGBoost is giving better results.
- XGBoost Regressor has the Least Root Mean Squared Error(RMSE) compare to other models. So It can be considered as the best model for the given Dataset.