Capstone Project

Bike Sharing Demand Prediction

Team Members:

Vidit Ghelani

Mahesh Lakum

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.

“Prediction of bike count required per hour.”

How to Approach:

Approach the problem in three simple steps:

1. Pre- Processing
    - 1.1 View the data
    - 1.2 Inspect the data
    - 1.3 Clean the data
2. Performing exploratory data analysis (EDA)
    - 2.1 Summary of Observations
    - 2.2 Normalizing the data
    - 2.3 Checking correlation
3. Training the Models
4. Identify the best model and conclude

Pre-Processing

In just few simple steps:

1. View the data

2. Inspect the data

3. Clean the data

   3.1 Remove missing values

   3.2 Remove duplicate values

   3.3 Change column names (if needed)

   3.4 Add derived features (date □ day, month, year & day □ weekend)

   3.5 Drop the redundant features

   3.5 Change data types

View the 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.

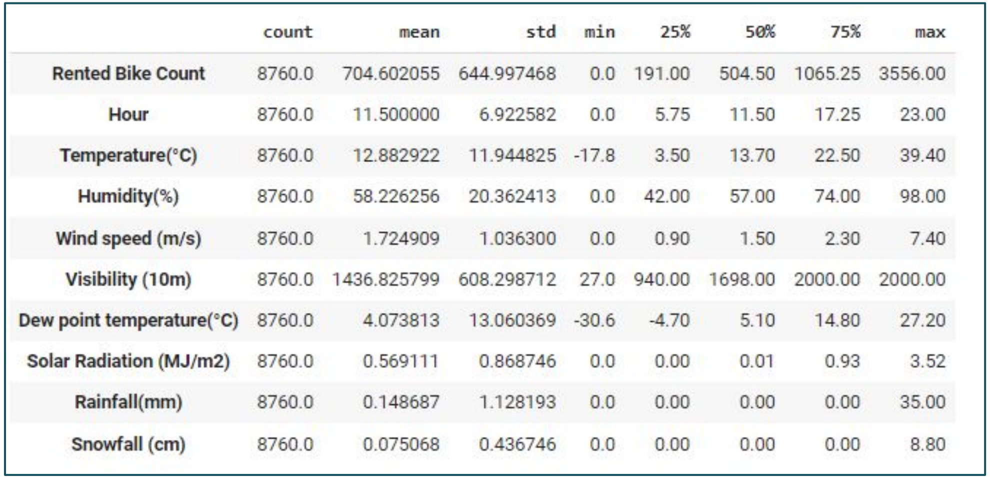
Dependent Variable

Features:

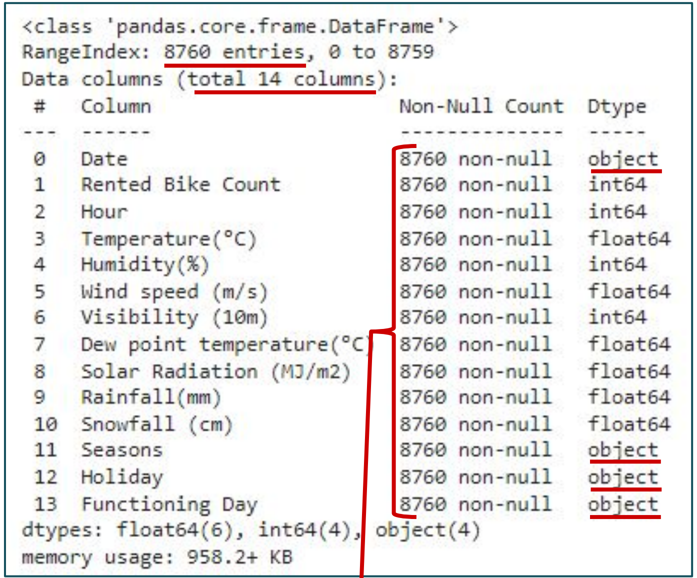
1. Date: year-month-day
2. Rented Bike count: Count of bikes rented at each hour
3. Hour: Hour of the day
4. Temperature: Temperature in Celsius
5. Humidity: %
6. Windspeed: m/s
7. Visibility:10m
8. Dew point temperature: Celsius
9. Solar radiation: MJ/m2
10. Rainfall: mm
11. Snowfall: cm
12. Seasons: Winter, Spring, Summer, Autumn
13. Holiday: Holiday/No holiday
14. Functional Day: NoFunc(Non Functional Hours), Fun(Functional hours)

View the data (Cont.):

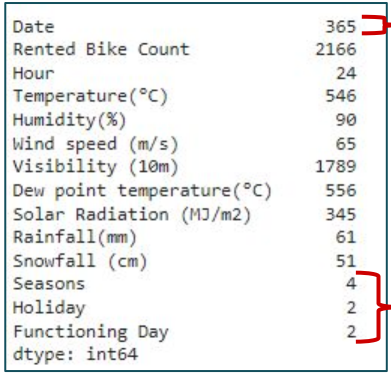
Using the basic commands like info, describe, unique, isnull, etc. we try to learn basic stats of the data:



	count	mean	std	min	25%	50%	75%	max
Rented Bike Count	8760.0	704.602055	644.997468	0.0	191.00	504.50	1065.25	3556.00
Hour	8760.0	11.500000	6.922582	0.0	5.75	11.50	17.25	23.00
Temperature(°C)	8760.0	12.882922	11.944825	-17.8	3.50	13.70	22.50	39.40
Humidity(%)	8760.0	58.226256	20.362413	0.0	42.00	57.00	74.00	98.00
Wind speed (m/s)	8760.0	1.724909	1.036300	0.0	0.90	1.50	2.30	7.40
Visibility (10m)	8760.0	1436.825799	608.298712	27.0	940.00	1698.00	2000.00	2000.00
Dew point temperature(°C)	8760.0	4.073813	13.060369	-30.6	-4.70	5.10	14.80	27.20
Solar Radiation (MJ/m2)	8760.0	0.569111	0.868746	0.0	0.00	0.01	0.93	3.52
Rainfall(mm)	8760.0	0.148687	1.128193	0.0	0.00	0.00	0.00	35.00
Snowfall (cm)	8760.0	0.075068	0.436746	0.0	0.00	0.00	0.00	8.80



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8760 entries, 0 to 8759
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Date                       8760 non-null   object
 1   Rented Bike Count          8760 non-null   int64
 2   Hour                       8760 non-null   int64
 3   Temperature(°C)            8760 non-null   float64
 4   Humidity(%)                8760 non-null   int64
 5   Wind speed (m/s)           8760 non-null   float64
 6   Visibility (10m)           8760 non-null   int64
 7   Dew point temperature(°C)  8760 non-null   float64
 8   Solar Radiation (MJ/m2)    8760 non-null   float64
 9   Rainfall(mm)               8760 non-null   float64
 10  Snowfall (cm)              8760 non-null   float64
 11  Seasons                    8760 non-null   object
 12  Holiday                    8760 non-null   object
 13  Functioning Day            8760 non-null   object
dtypes: float64(6), int64(4), object(4)
memory usage: 958.2+ KB
```



```
Date                          365
Rented Bike Count            2166
Hour                           24
Temperature(°C)               546
Humidity(%)                    90
Wind speed (m/s)               65
Visibility (10m)             1789
Dew point temperature(°C)     556
Solar Radiation (MJ/m2)       345
Rainfall(mm)                   61
Snowfall (cm)                  51
Seasons                         4
Holiday                         2
Functioning Day                 2
dtype: int64
```

Not present above.
Hence,
Categorical Data.

No Null Values

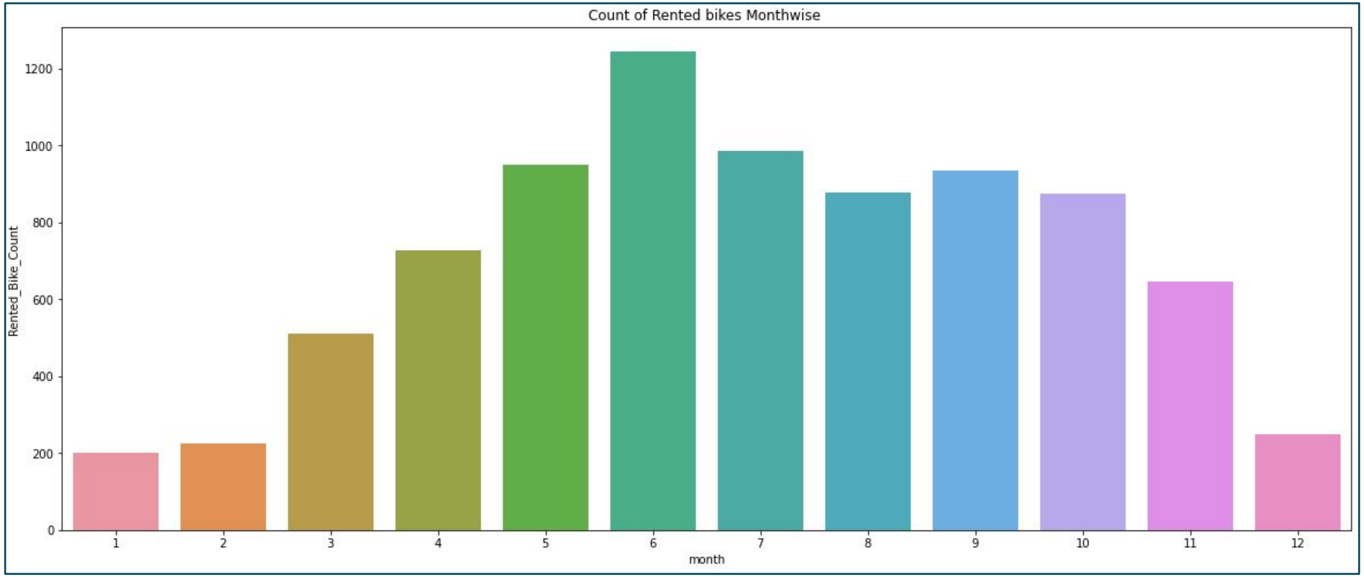
View the data (Cont.):

Summary about the data:

- Shape □ (8760,14) □ 14 features and 8760 data
- Dtypes □ object, float and integer. We might have to alter a few data types.
- Missing values □ None
- Duplicate values □ None
- Null Values □ None
- Date, Seasons, Holiday and Functioning Day are categorical variables.
- Date, Hours, Seasons □ 365, 24, 4 □ Data of 24 hours of day is available for a period of 1 year. □ 24*365 = 8760. Hence, each entry corresponds to data of each hour.
- We have renamed the columns for ease of access.
- We extract the day, month and year from ‘date’ column and save it as separate columns. From this we identify the weekends and then drop the ‘year’, ‘day’ and ‘date’ column as they are irrelevant to us now.

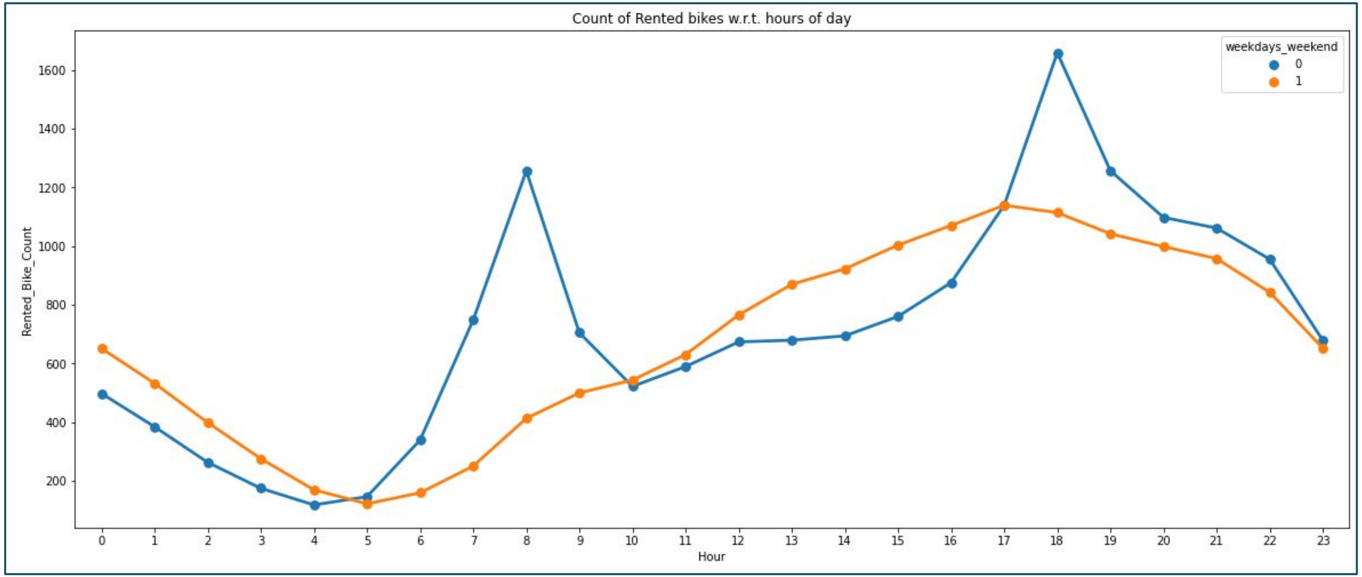
Performing the EDA

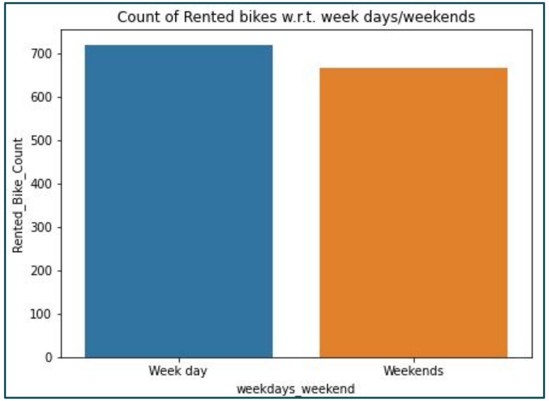
We performed EDA on the data for various features like Month, Weekend, Hour, functioning day, season and holiday and based on it we noted down our observations.



Month □ From the month 5 to 10 the demand of the rented bike is high as compare to other months. These months corresponds to the summer season.

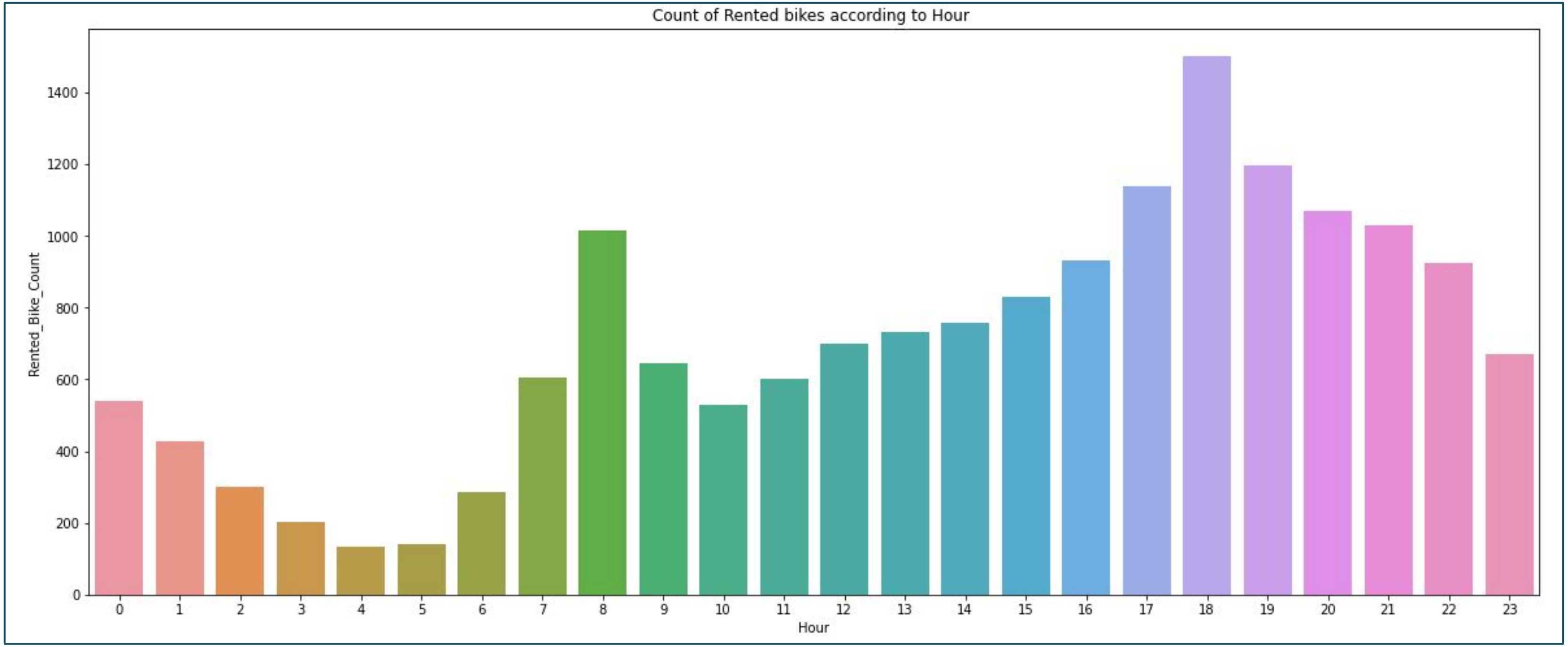
EDA (Cont.)





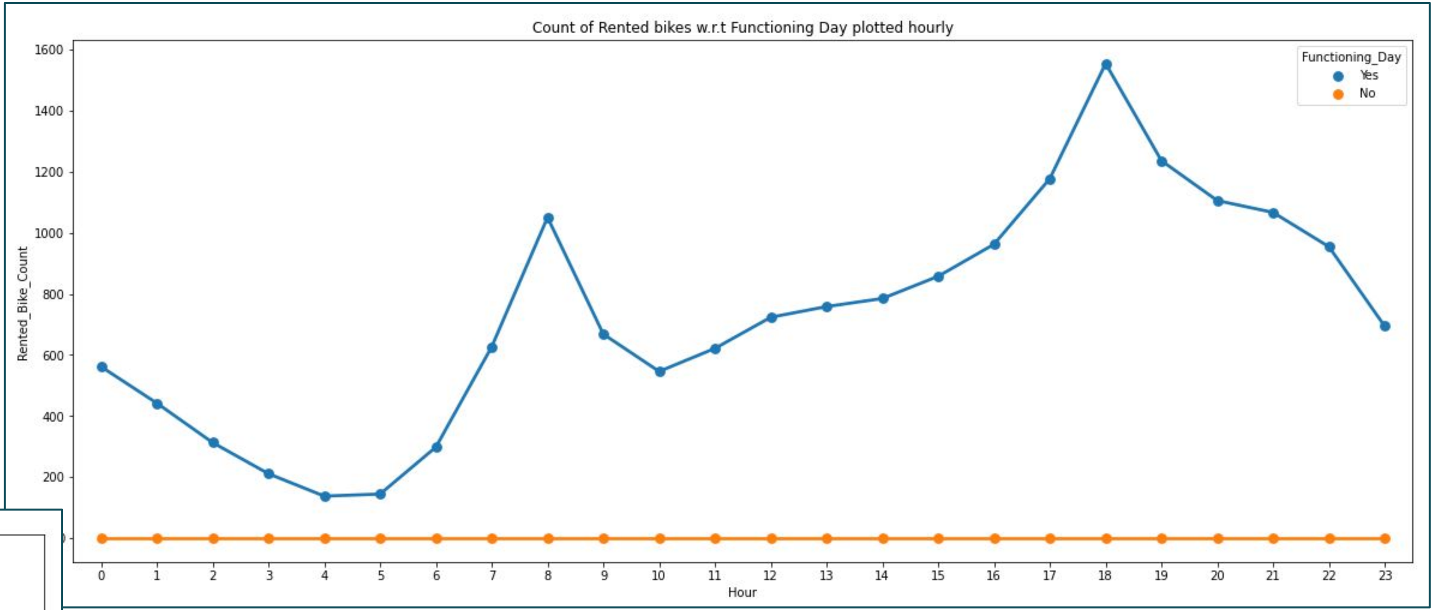
Weekend □ The week days which represent in blue color show that the demand of the bike higher because of the office. Peak Time are 7 am to 9 am and 5 pm to 7 pm. The orange color represent the weekend days, and it show that the demand of rented bikes are very low especially in the morning hour but when the evening start from 4 pm to 8 pm the demand slightly increases.

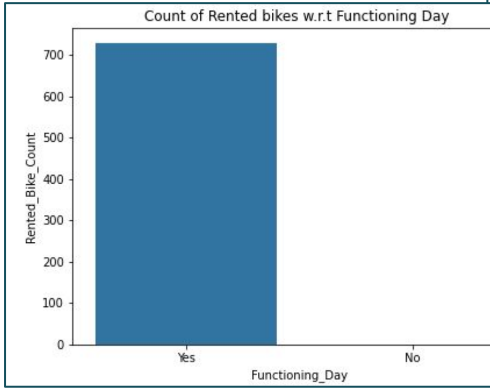
EDA (Cont.)



Hour □ Generally people use rented bikes during their working hour from 7am to 9am and 5pm to 7pm.

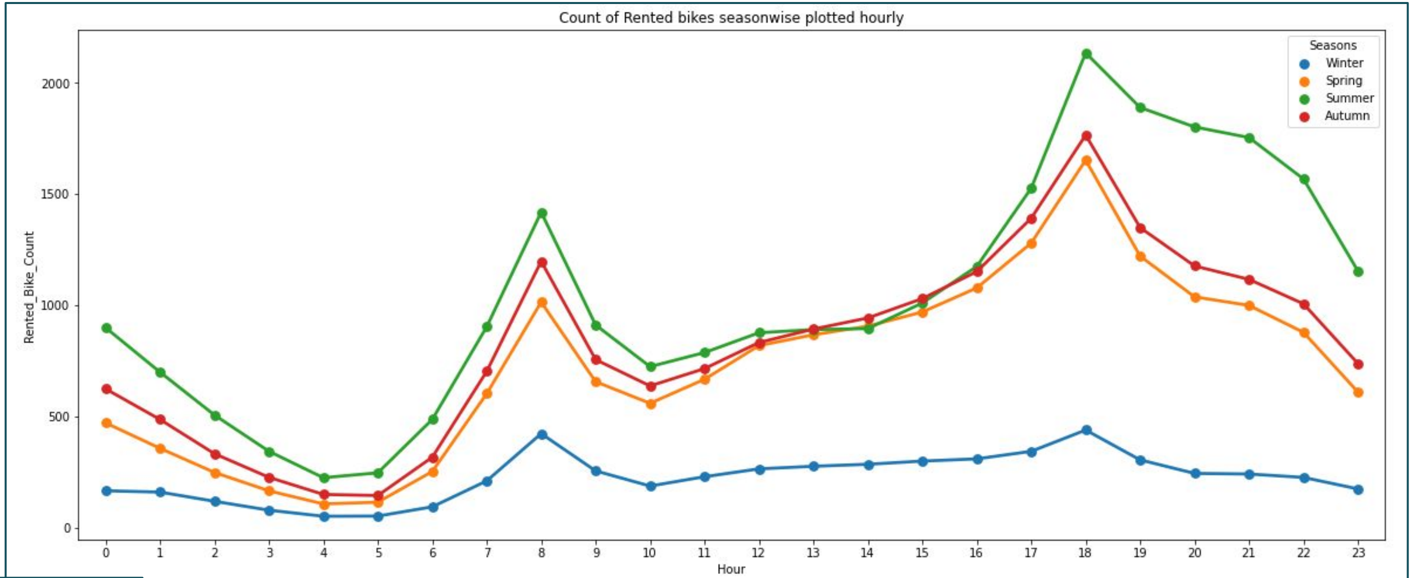
EDA (Cont.)

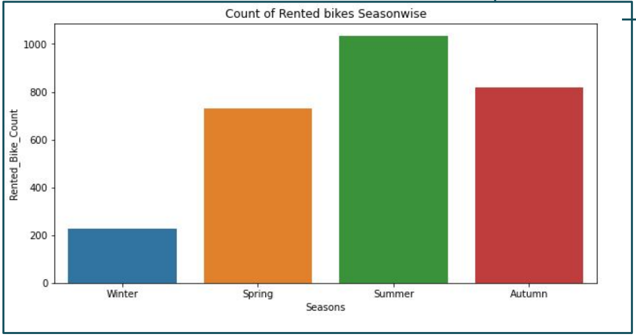




Functioning Day □ The use of rented bike in functioning days or not, and it clearly shows that peoples don't use rented bikes in no functioning day.

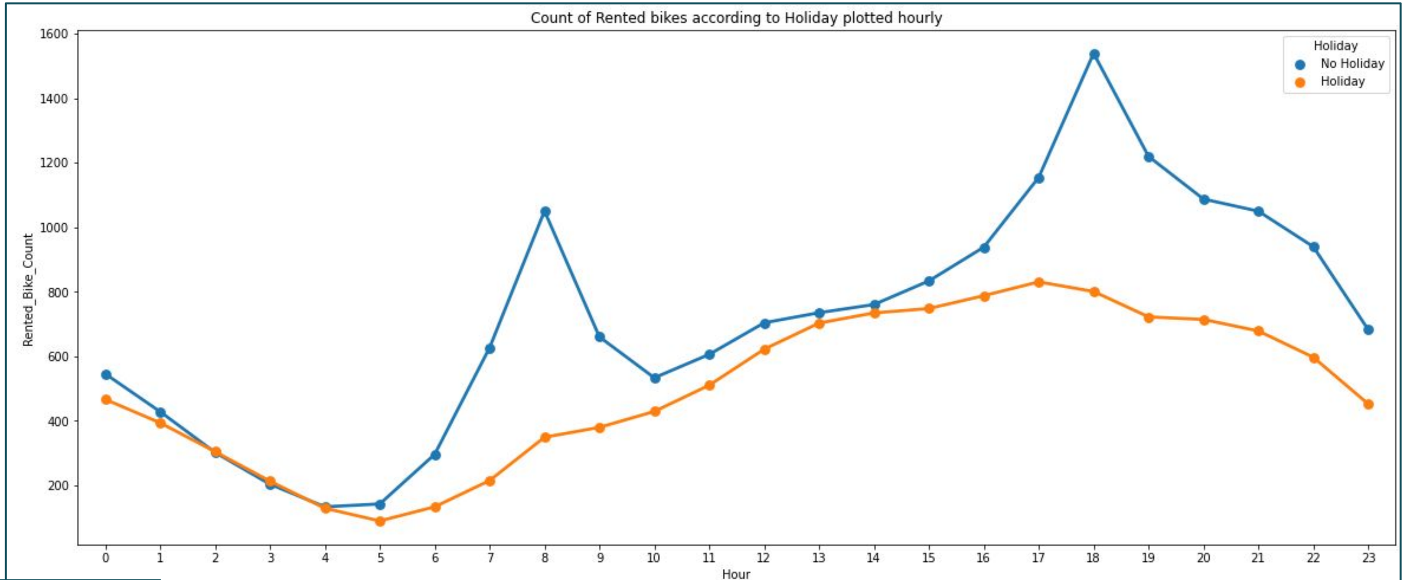
EDA (Cont.)

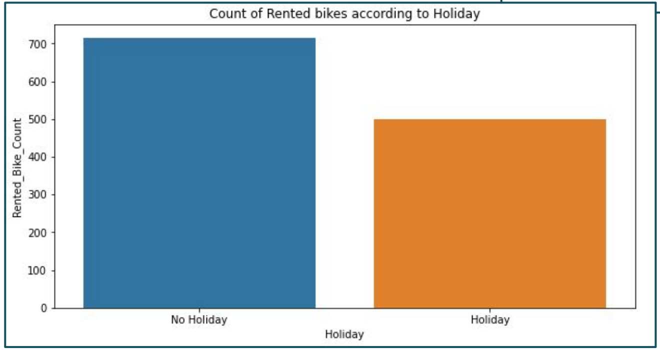




Seasons □ In summer season the use of rented bike is high and peak time is 7am-9am and 7pm-5pm and in winter it could be low because of snowfall.

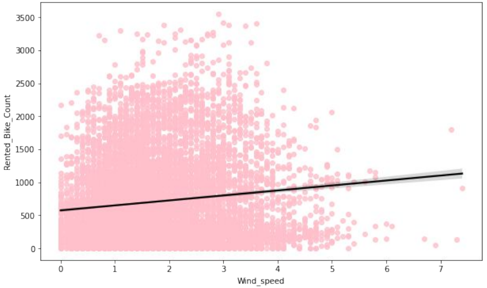
EDA (Cont.)

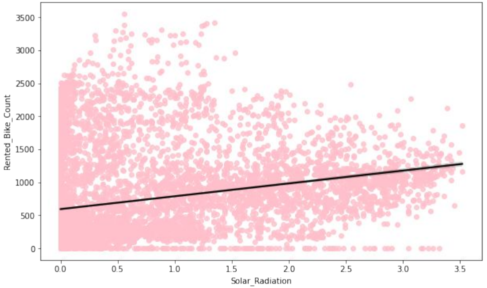

Count of Rented bikes according to Holiday plotted hourly

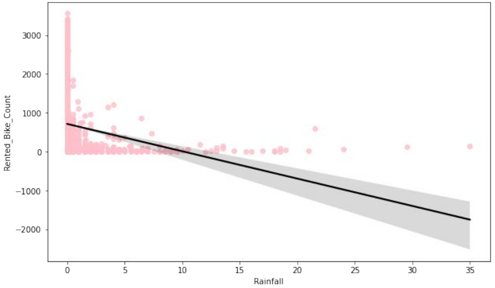

Count of Rented bikes according to Holiday

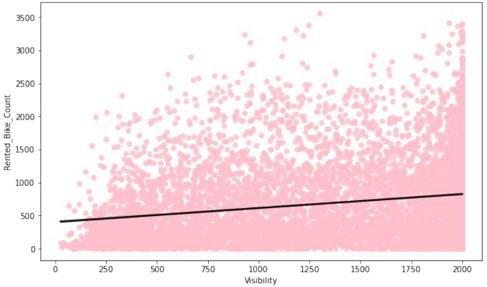
Holiday □ The use of rented bike in a holiday, and it clearly shows that in holiday people uses the rented bike from 2pm-8pm

EDA (Cont.)

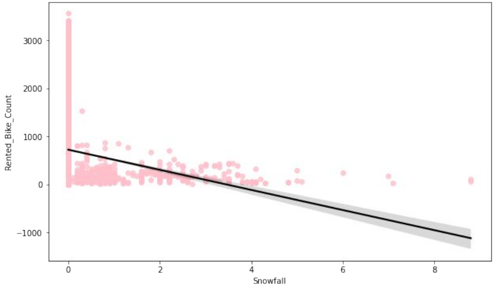


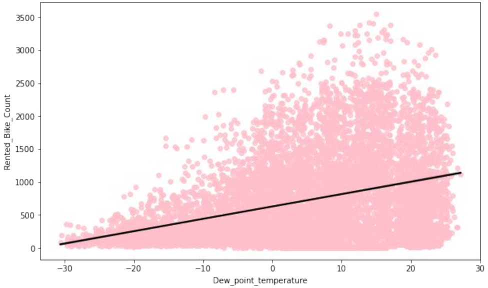


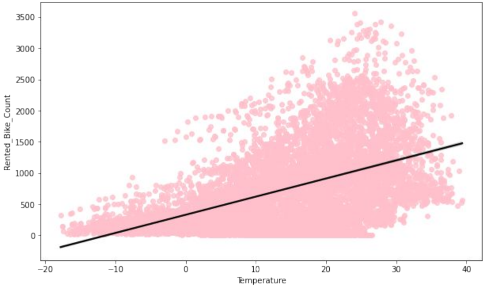


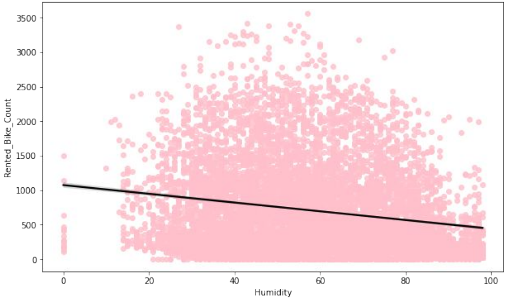


Regression Plots





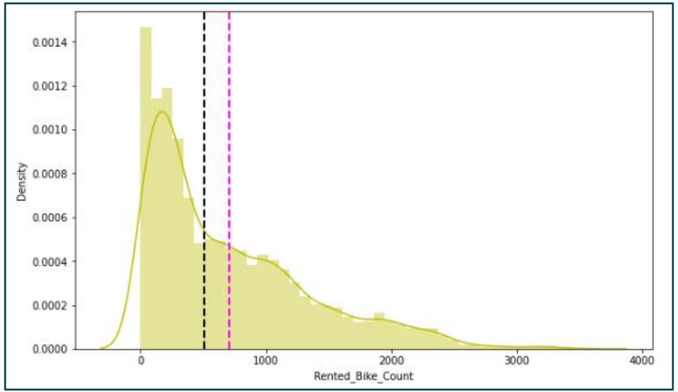


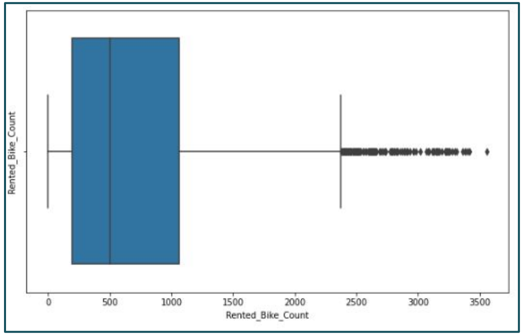


EDA (Cont.)

Normalizing the data:

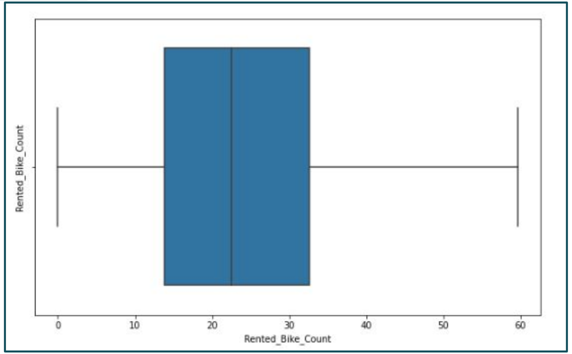
We now check the regression plot for each feature with respect to the dependent variable to know the relation between them. We find that the '**rented_bike_count**' feature is left skewed. Hence we need to Normalize this feature. We use **square root transformation** for the same.

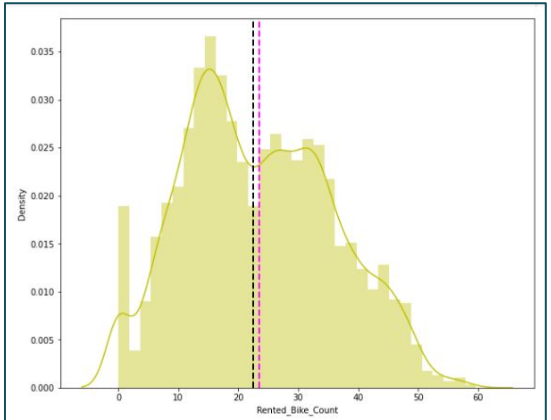




Before

After Normalizing

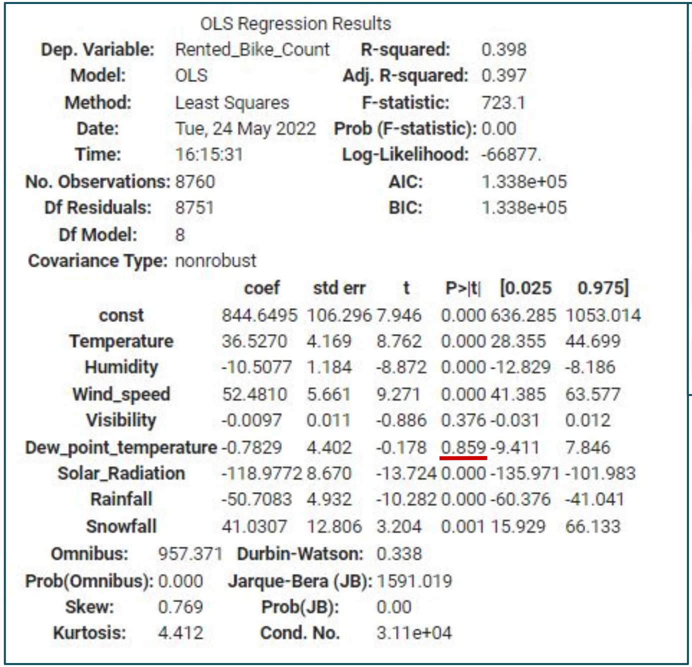




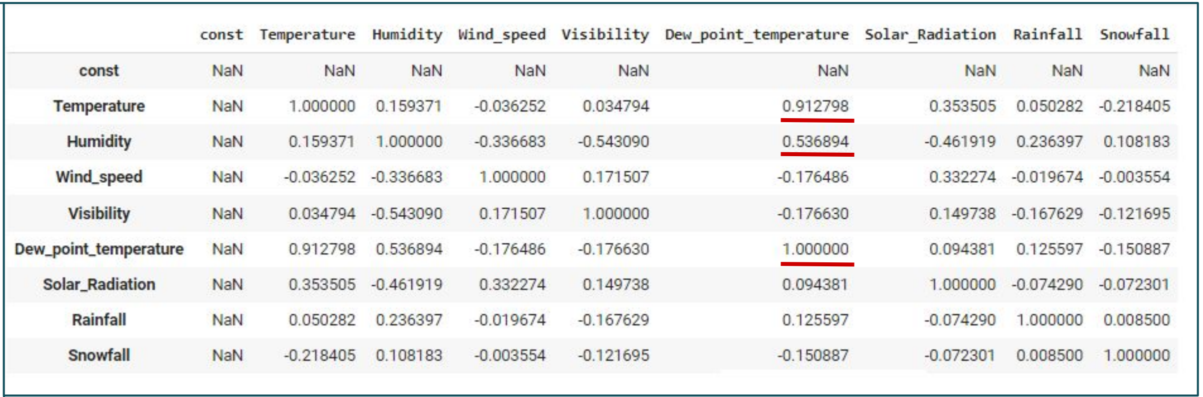
EDA (Cont.)

Checking Correlation between models:

OLS Model



```
                      OLS Regression Results
    Dep. Variable:   Rented_Bike_Count    R-squared:          0.398
            Model:   OLS               Adj. R-squared:        0.397
           Method:   Least Squares        F-statistic:        723.1
             Date:   Tue, 24 May 2022  Prob (F-statistic):    0.00
             Time:   16:15:31           Log-Likelihood:       -66877.
 No. Observations:   8760                       AIC:          1.338e+05
     Df Residuals:   8751                       BIC:          1.338e+05
         Df Model:   8
  Covariance Type:   nonrobust
                           coef     std err   t       P>|t|  [0.025    0.975]
            const          844.6495 106.296  7.946   0.000  636.285   1053.014
      Temperature          36.5270  4.169    8.762   0.000  28.355    44.699
         Humidity          -10.5077 1.184    -8.872  0.000  -12.829   -8.186
       Wind_speed          52.4810  5.661    9.271   0.000  41.385    63.577
       Visibility          -0.0097  0.011    -0.886  0.376  -0.031    0.012
Dew_point_temperature      -0.7829  4.402    -0.178  0.859  -9.411    7.846
   Solar_Radiation         -118.9772 8.670   -13.724 0.000  -135.971  -101.983
         Rainfall          -50.7083 4.932    -10.282 0.000  -60.376   -41.041
         Snowfall          41.0307  12.806   3.204   0.001  15.929    66.133
      Omnibus:     957.371  Durbin-Watson:    0.338
Prob(Omnibus):     0.000    Jarque-Bera (JB): 1591.019
         Skew:     0.769       Prob(JB):      0.00
     Kurtosis:     4.412       Cond. No.      3.11e+04
```

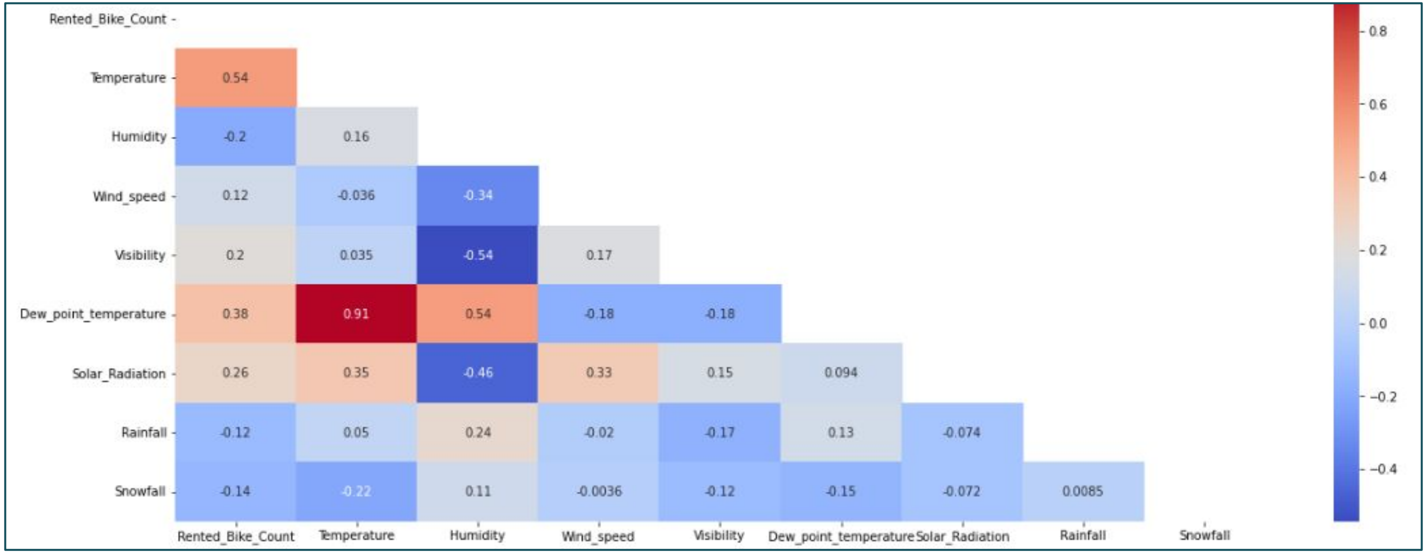


	const	Temperature	Humidity	Wind_speed	Visibility	Dew_point_temperature	Solar_Radiation	Rainfall	Snowfall
const	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Temperature	NaN	1.000000	0.159371	-0.036252	0.034794	0.912798	0.353505	0.050282	-0.218405
Humidity	NaN	0.159371	1.000000	-0.336683	-0.543090	0.536894	-0.461919	0.236397	0.108183
Wind_speed	NaN	-0.036252	-0.336683	1.000000	0.171507	-0.176486	0.332274	-0.019674	-0.003554
Visibility	NaN	0.034794	-0.543090	0.171507	1.000000	-0.176630	0.149738	-0.167629	-0.121695
Dew_point_temperature	NaN	0.912798	0.536894	-0.176486	-0.176630	1.000000	0.094381	0.125597	-0.150887
Solar_Radiation	NaN	0.353505	-0.461919	0.332274	0.149738	0.094381	1.000000	-0.074290	-0.072301
Rainfall	NaN	0.050282	0.236397	-0.019674	-0.167629	0.125597	-0.074290	1.000000	0.008500
Snowfall	NaN	-0.218405	0.108183	-0.003554	-0.121695	-0.150887	-0.072301	0.008500	1.000000

- **From this OLS model we observe that the 'Temperature' and 'Dew_point_temperature' are highly correlated so we need to drop one of them.**
- **For dropping that we check the (P>|t|) value from above table and we can see that the 'Dew_point_temperature' value is higher so we need to drop ’Dew_point_temperature’ column.**

EDA (Cont.)

Checking Correlation between models: Heatmap



We can observe on the heatmap that on the target variable line the most positively correlated variables to the rented bike count are : Temperature, dew point temperature and solar radiation

And most negatively correlated variables are:

Humidity, Rainfall and Snowfall

The positive correlation between columns 'Temperature' and 'Dew point temperature' is 0.91. So even if we drop this column it does not affect the outcome of our analysis. Also they have the same variations, so we can drop the column 'Dew point temperature(°C)'.

Training the Model

Split the data for Training and Testing: We split the data into 75 : 25 ratio.
Now we train the model and then test the same for various models. Based on that we will select the best suited model for our prediction.

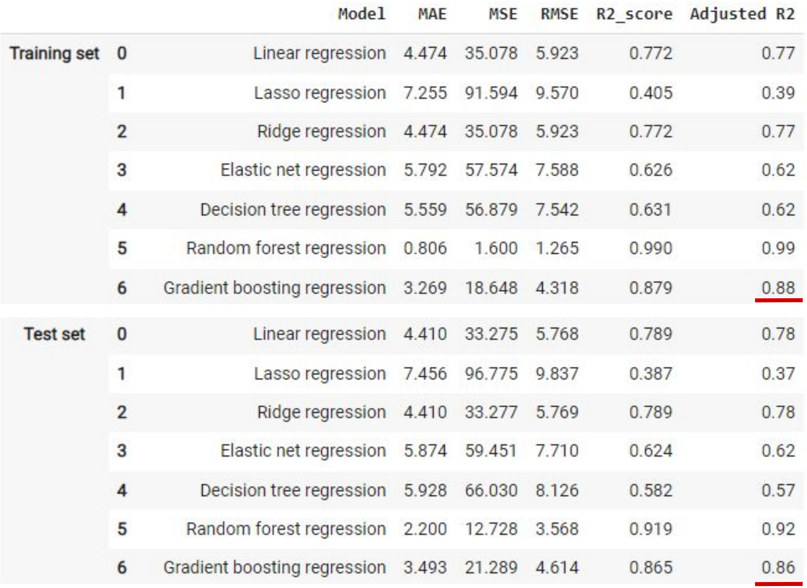
The models used are:

- **Linear regression**
- **Lasso regression**
- **Ridge regression**
- **Elastic net regression**
- **Decision Tree regression**
- **Random forest regression**
- **Gradient Boosting regression**

Training the Model (Cont.)

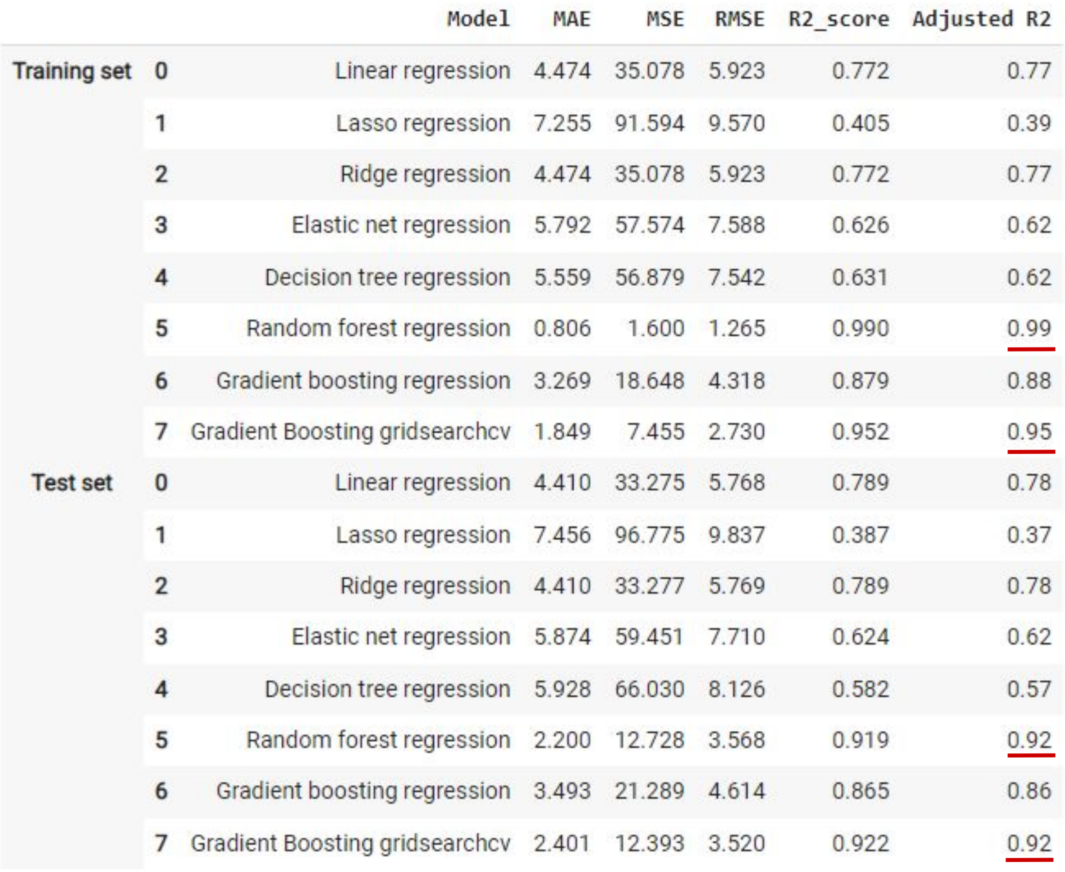
Hyper parameter Tuning:

Since the gradient boost regression algorithm is the second best in performance we tried hyper parameter tuning on it with GridSearchCV. Now, we will see its performance.



		Model	MAE	MSE	RMSE	R2_score	Adjusted R2
Training set	0	Linear regression	4.474	35.078	5.923	0.772	0.77
	1	Lasso regression	7.255	91.594	9.570	0.405	0.39
	2	Ridge regression	4.474	35.078	5.923	0.772	0.77
	3	Elastic net regression	5.792	57.574	7.588	0.626	0.62
	4	Decision tree regression	5.559	56.879	7.542	0.631	0.62
	5	Random forest regression	0.806	1.600	1.265	0.990	0.99
	6	Gradient boosting regression	3.269	18.648	4.318	0.879	0.88
Test set	0	Linear regression	4.410	33.275	5.768	0.789	0.78
	1	Lasso regression	7.456	96.775	9.837	0.387	0.37
	2	Ridge regression	4.410	33.277	5.769	0.789	0.78
	3	Elastic net regression	5.874	59.451	7.710	0.624	0.62
	4	Decision tree regression	5.928	66.030	8.126	0.582	0.57
	5	Random forest regression	2.200	12.728	3.568	0.919	0.92
	6	Gradient boosting regression	3.493	21.289	4.614	0.865	0.86

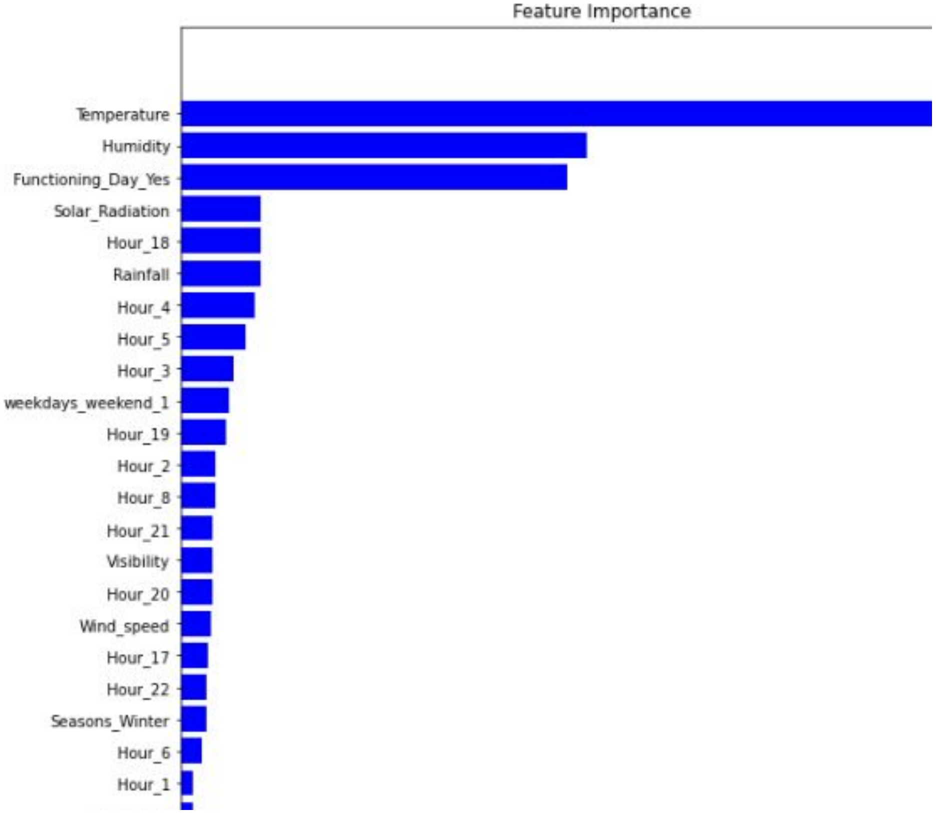
Identifying the best model



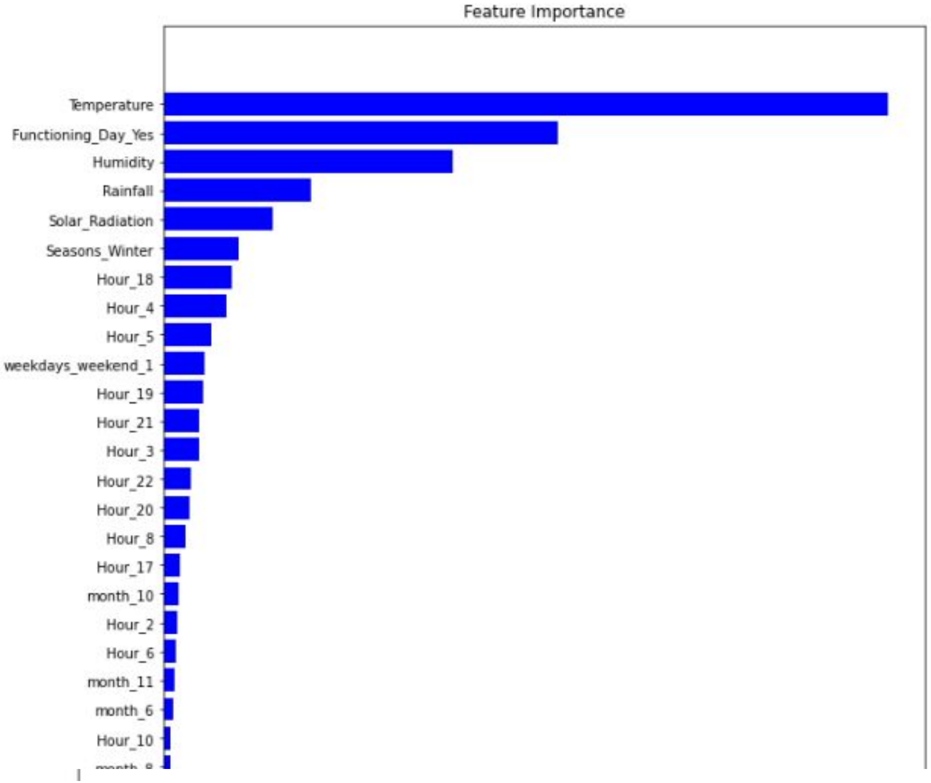
		Model	MAE	MSE	RMSE	R2_score	Adjusted R2
Training set	0	Linear regression	4.474	35.078	5.923	0.772	0.77
	1	Lasso regression	7.255	91.594	9.570	0.405	0.39
	2	Ridge regression	4.474	35.078	5.923	0.772	0.77
	3	Elastic net regression	5.792	57.574	7.588	0.626	0.62
	4	Decision tree regression	5.559	56.879	7.542	0.631	0.62
	5	Random forest regression	0.806	1.600	1.265	0.990	0.99
	6	Gradient boosting regression	3.269	18.648	4.318	0.879	0.88
	7	Gradient Boosting gridsearchcv	1.849	7.455	2.730	0.952	0.95
Test set	0	Linear regression	4.410	33.275	5.768	0.789	0.78
	1	Lasso regression	7.456	96.775	9.837	0.387	0.37
	2	Ridge regression	4.410	33.277	5.769	0.789	0.78
	3	Elastic net regression	5.874	59.451	7.710	0.624	0.62
	4	Decision tree regression	5.928	66.030	8.126	0.582	0.57
	5	Random forest regression	2.200	12.728	3.568	0.919	0.92
	6	Gradient boosting regression	3.493	21.289	4.614	0.865	0.86
	7	Gradient Boosting gridsearchcv	2.401	12.393	3.520	0.922	0.92

Feature Importance

As per Random Forest



As per Gradient Boosting with Grid Search CV



Conclusions:

From this exercise we can conclude that:

1. No overfitting is seen.
2. Random forest Regressor and Gradient Boosting Grid Search CV gives the highest R2 score of 99% and 95% respectively for Train Set and 92% for the Test set.
3. Upon applying GridSearchCV to Gradient Boosting, we observed that it resulted in a significant improvement in the performance of that model. However, not better than that of Random Forest model.
4. Feature Importance value for Random Forest and Gradient Boost are different. Hence, we can choose either of these models for our prediction.

Thank you

References of Images:

1. Vector image and graphics of the muse: shutterstock.com.