Capstone Project

Seoul Bike Sharing Demand Prediction

By

Sourav Chowdhury

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 ofrental bikes becomes a major concern.

- To come up with a Machine Learning Model which best predicts the Bike Count Required ateach hour for stable supply of rental bikes.

Problem Faced

- Dataset contains correlated variables .

- Applying Machine Learning Models, it was very much tedious to get the best algorithm which can perfectly gives the best prediction.

-Since Hyper Parameter Tuning was done on Algorithm , Thus it took more computational time to execute.

Python Modules/Packages/Libraries

```
import numpy as np                                          # Numerical Computing
import pandas as pd                                         # Data Processing
from numpy import math                                      # For Mathematical Formulation
from sklearn.model_selection import train_test_split        # Splitting Data into train and test dataset
from sklearn.linear_model import LinearRegression           # Linear Regression implementation
from sklearn.metrics import r2_score                        # To get R- Squared Score
from sklearn.metrics import mean_squared_error              # To get Mean Squared Error in Evaluation Metrics
import seaborn as sns                                       # To use Modern Visualisation
import matplotlib.pyplot as plt                             # Visualisation
import plotly.express as px                                 # To create High level graph from entire data
from sklearn.model_selection import GridSearchCV            # Implementing gridsearchCV
from sklearn.linear_model import Lasso                      # Lasso Implementation
from sklearn.linear_model import Ridge                      # Ridge Implementation
from sklearn.tree import DecisionTreeRegressor              # Use of Decision Tree Algorithm
from sklearn.ensemble import RandomForestRegressor          # Using RandomForest Algorithm
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import AdaBoostRegressor
import xgboost as xgb
```

Reading Input Dataset

To load the dataset in the colab notebook, first we mounted the notebook with google drive.And then read the dataset using pandas built in function read_csv.

```
from google.colab import drive
drive.mount('/content/drive')
dataset =
pd.read_csv("/content/drive/MyDrive/Supervised ML Regression( Bike Sharing Demand Prediction)/Seou
BikeData.csv", encoding = "unicode_escape")
```

Data Pipeline

- **Data Processing - 1:** In this part we have done EDA in which we have removed some columns and extracted some information from Univariate and Bivariate Analysis.
- **Data Processing - 2:** In this part we have Checked for Outliers and Correlation, merged the columns having mutual correlation.
- **Data Preparation :** Done Feature engineering on the data by assigning dummy values to categorical features.
- **Creating Model :** Finally, in this we have created and trained the models with iterative process so as to make ideal model with better Evaluation metrics.

Data Summary

- **Dependent Variable-**

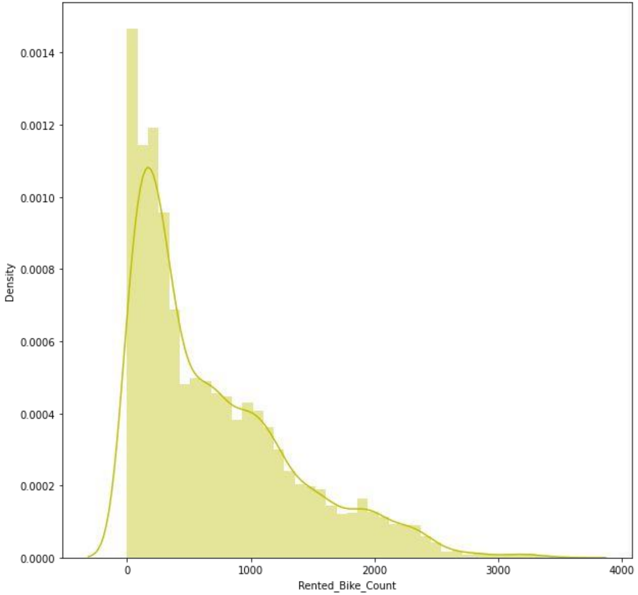
1. **Rented Bike count :** Count of Bikes rented each hour

- **Independent Variables -**

1. **Date :** Date of the bike rented in (yy- mm- dd).
2. **Hour :** Hour of the day
3. **Temperature :** Temperature at the time bike was rented.
4. **Humidity :** Moisture in the atmosphere at the time bike rented out, given in percentage.
5. **Windspeed :** Speed of the wind.
6. **Visibility :** visibility within 10 meters.
7. **Dew point temperature :** Temperature in Degree Celsius
8. **Solar radiation :** Radiation directly coming from sun in MJ/m2
9. **Rainfall :** Amount of rainfall in MilliMetres
10. **Snowfall :** Amount of snowfall in Centimeter
11. **Seasons :** Type of Season Winter, Spring, Summer, Autumn
12. **Holiday :** Holiday/No Holiday
13. **Functional Day :** NoFunc(Non Functional Hours), Fun(Functional hours)

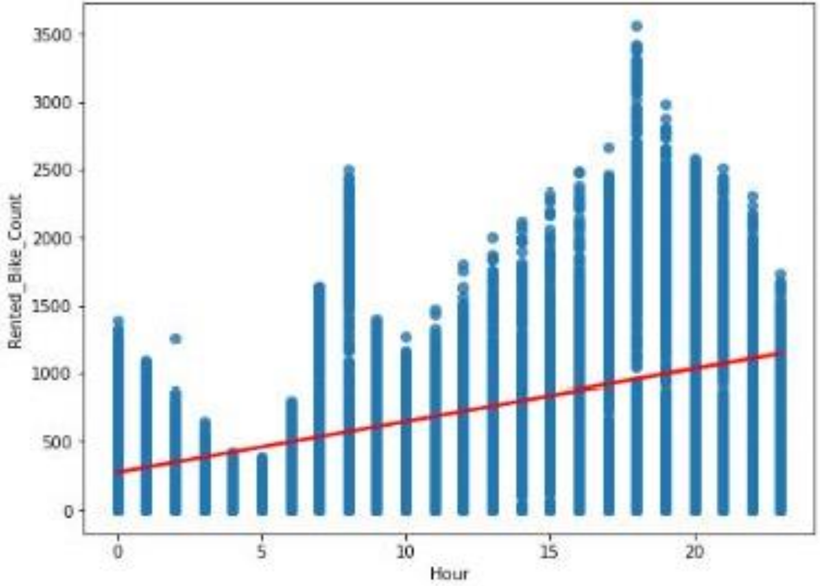
Define Dependent Variable

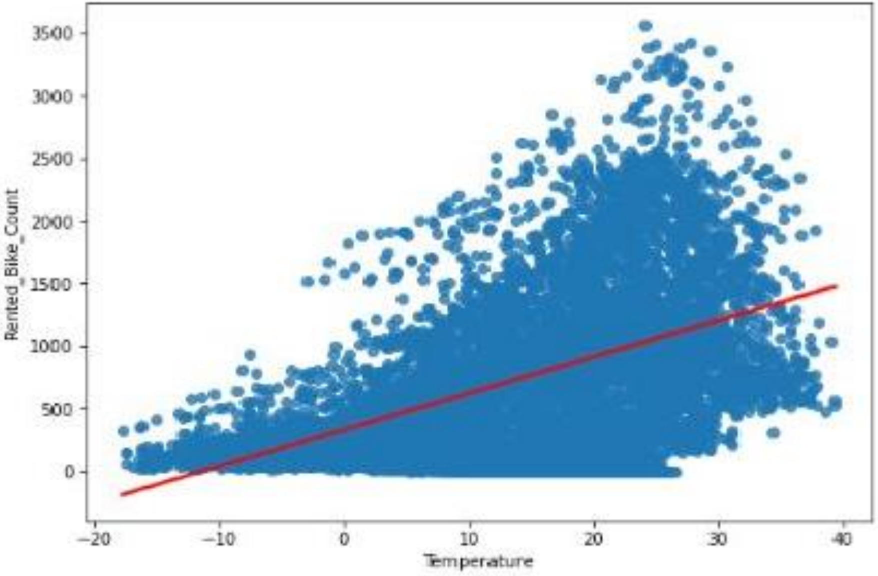
The Rented Bike Count Plot is Positively Skewed. And total no of bikes rented in 2017 and 2018 are depicted using the bar graph. Less in 2017, Mostly rented in 2018.



EDA

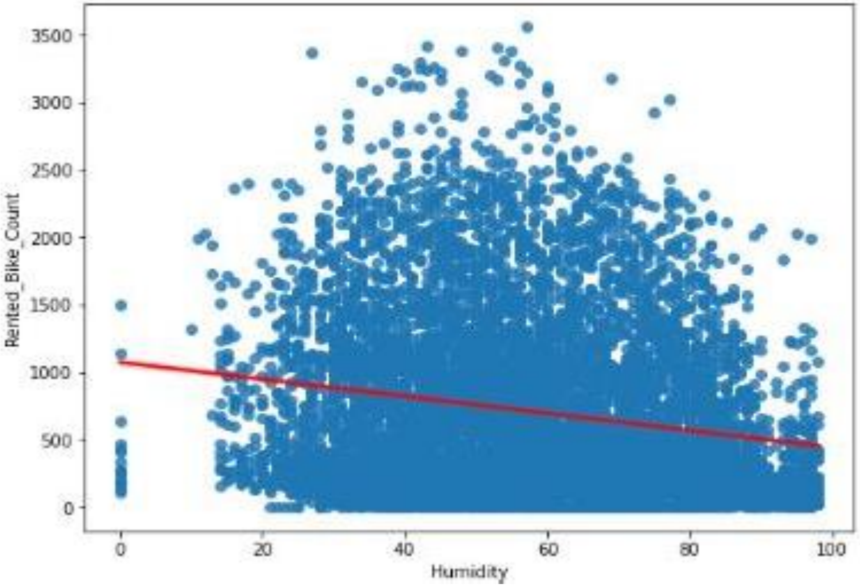
- Demand for bikes is mostly in the evening between 3 to 8 pm, also the least demand is at morning 5pm.
- People prefer to rent bikes at normal temperature of 20°C. to 30°C. Hence it is positively related to Rented Bike.

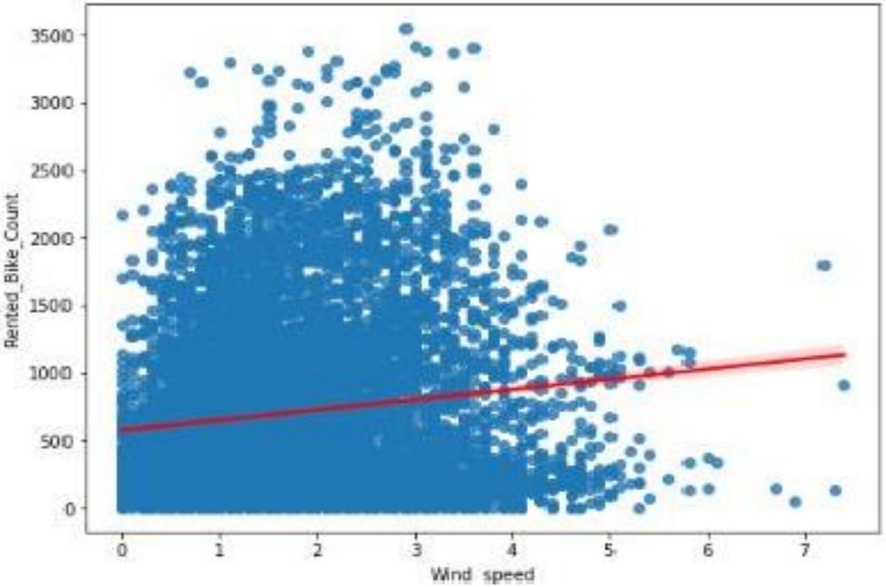




EDA

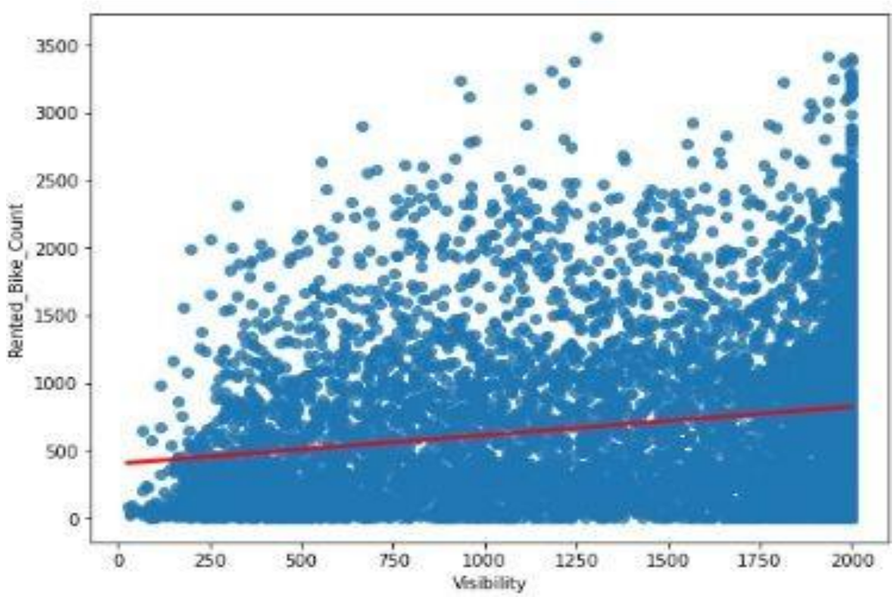
- Humidity is negatively correlated , as people prefer to rent a bike less if there is more moisture in the air.
- Wind Speed doesn't affect much for renting a bike but is slightly positively correlated.

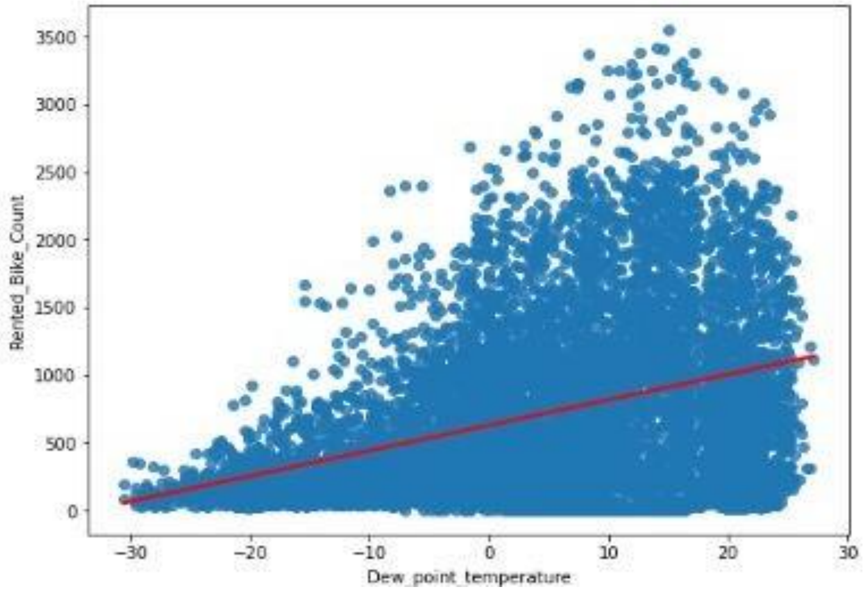




EDA

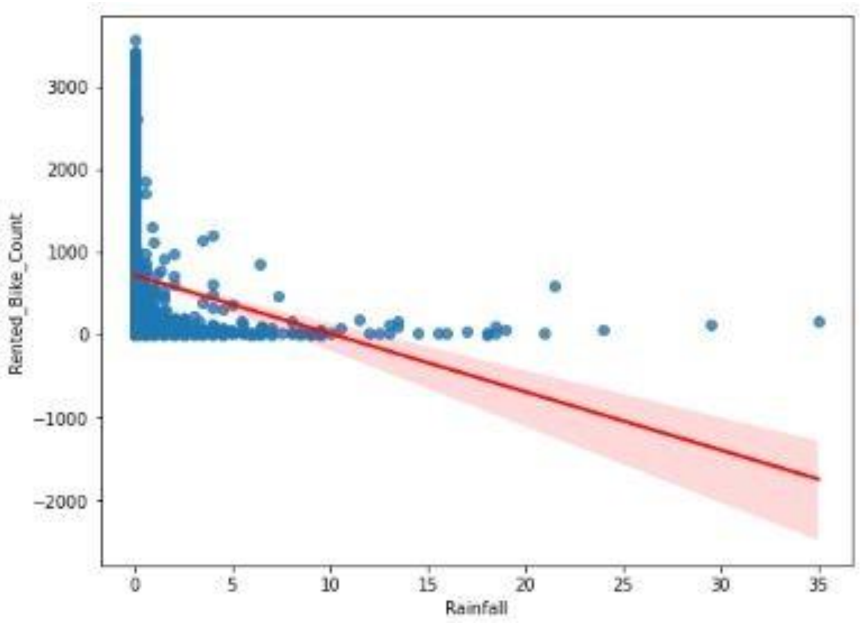
- Visibility does not affect that much, similar to wind speed, but there seems slightly positive correlation.
- The dew point is the temperature the air needs to be cooled to (at constant pressure) in order to achieve a relativehumidity. It is positively correlated with data.

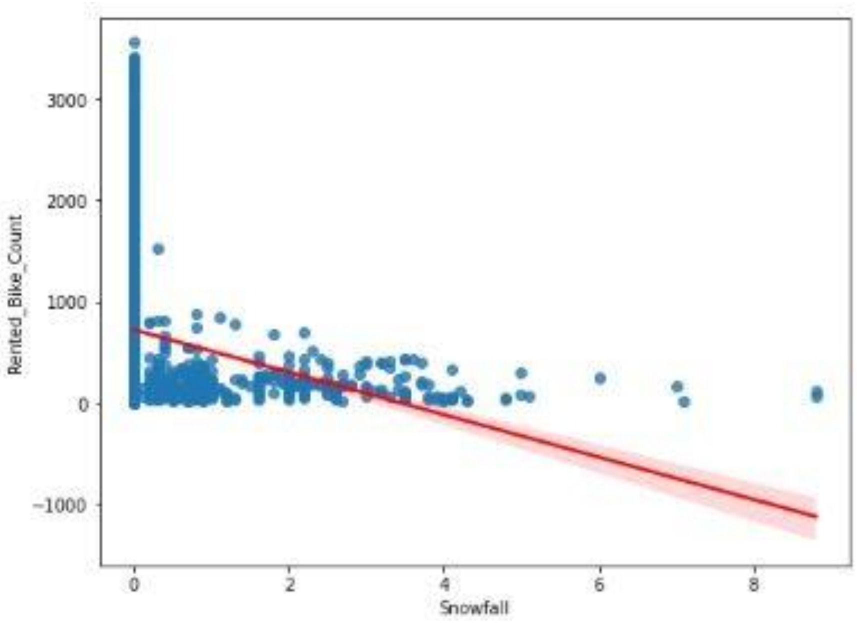




EDA

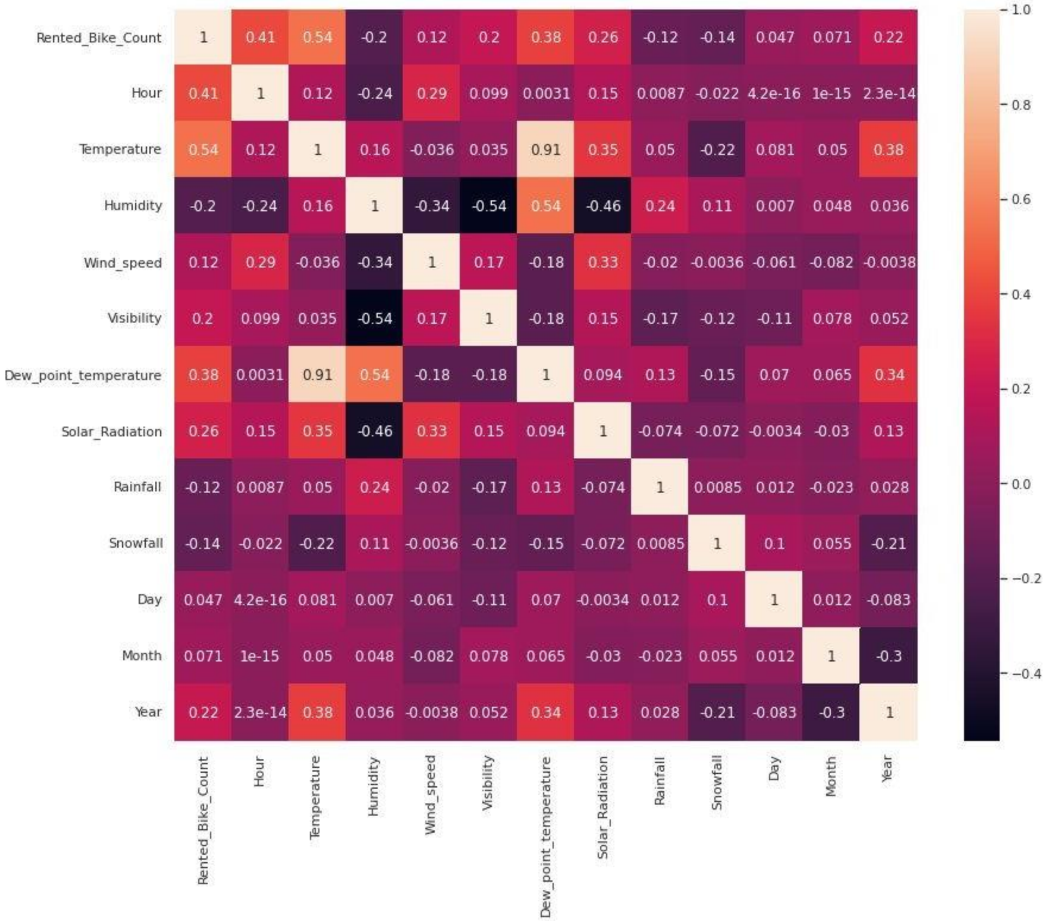
- People don't prefer to rent a bike, when there is rainfall or snowfall. Hence Both are highly Negatively related withRented Bike Count.





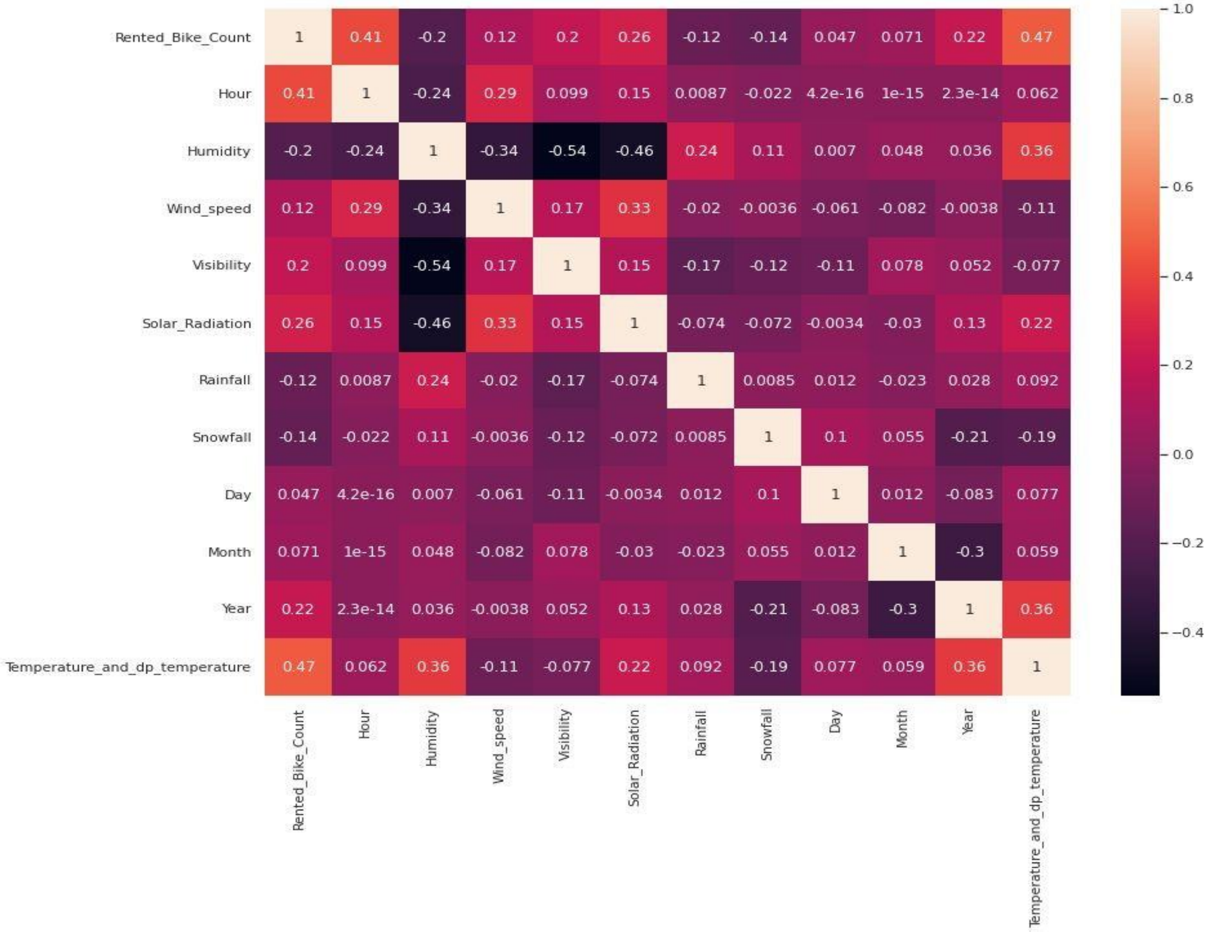
EDA (Correlation)

Temperature and Dew Point temperature are highly correlated. We can add them to make one single column
Thus removing them and then analyzing the data.



EDA (Correlation)

Highest correlation is shown by Temperatureand Dp Temperature.



Data Preparation

One hot encoding

Creating dummies column for the given feature

dataset_copy=pd.get_dummies(dataset,drop_first=True)

Train Set `(7008, 16)`

Test Set `(1752, 16)`

	Rented_Bike_Count	Hour	Humidity	Wind_speed	Visibility	Solar_Radiation	Rainfall	Snowfall	Day	Month	Year
0	254	0	37	2.2	2000	0.0	0.0	0.0	12	1	2017
1	204	1	38	0.8	2000	0.0	0.0	0.0	12	1	2017
2	173	2	39	1.0	2000	0.0	0.0	0.0	12	1	2017
3	107	3	40	0.9	2000	0.0	0.0	0.0	12	1	2017
4	78	4	36	2.3	2000	0.0	0.0	0.0	12	1	2017

Temperature_and_dp_temperature	Seasons_Spring	Seasons_Summer	Seasons_Winter	Holiday_No Holiday	Functioning_Day_Yes
-22.8	0	0	1	1	1
-23.1	0	0	1	1	1
-23.7	0	0	1	1	1
-23.8	0	0	1	1	1
-24.6	0	0	1	1	1

Model Building

For Linear Regression to be implemented we have to take certain assumptions.

1. **Linear relationship** - There should be a linear relationship between feature variable and dependent variable.
2. **Little or no-multicollinearity** - There should not be multicollinearity among variables.
3. **Little or no auto-correlation** - Another assumption is that there is little or no autocorrelation in the data. Autocorrelation occurs when the residual errors are not independent from each other.
4. **Homoscedasticity** - Variance should be the same, i.e. error term should be same across all values of the independent variable.

Evaluation Metrics

1. **Mean Squared Error (MSE)** is the mean of the squared errors.
2. **Root Mean Squared Error (RMSE)** is the square root of the mean of the squared errors.
3. **R-Squared**
4. **Adjusted R-Squared**

For Train Dataset(Linear Regression)

MSE : 190710.62548259995

RMSE : 436.70427692272483

R2: 0.5488899632663061

Adjusted R2 : 0.5478575271931064

For Test Dataset(Linear Regression)

MSE : 169871.70934024057

RMSE : 412.15495792267325

R2: 0.5624656403341544

Adjusted R2 : 0.5584307413401177

Model Summary

For Train Dataset :

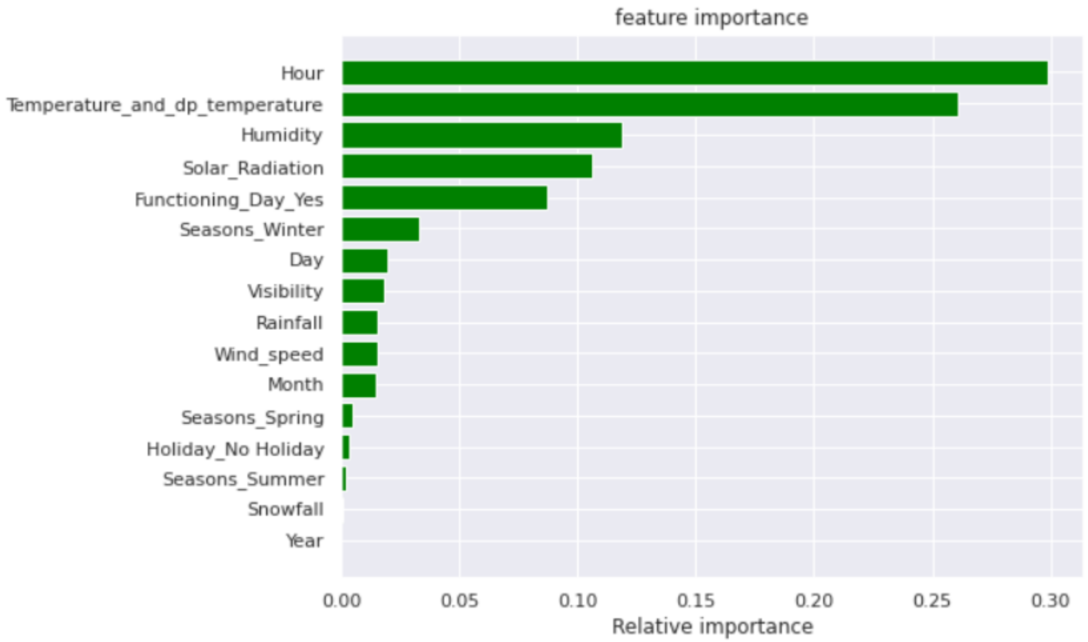
SL NO	MODEL_NAME	Train MSE	Train RMSE	Train R^2	Train Adjusted R^2
1	Linear Regression	190710.62548259995	436.70427692272483	0.5488899632663061	0.5478575271931064
2	Lasso Regression	190710.6254850217	436.7042769254976	0.5488899632605777	0.5447298707027501
3	Ridge Regression	190710.63414152752	436.7042868366734	0.5488899059814547	0.5447298128954048
4	ElasticNet Regression	203220.4874674924	450.7998308201683	0.5192989308565615	0.5148659526973137
5	DecisionTree Regressor	76328.2706429576	276.27571489900737	0.8194518586133761	0.8177868613441046
6	RandomForest Regressor	7014.866722873859	83.75480119296958	0.983406919373109	0.9832538996094028
7	Gradient Boost	61001.9124998751	246.9856524170485	0.8557050771607113	0.8543744035206948
8	Xg Boost	61295.43648699404	247.57915196355697	0.855010770714616	0.8536736942485836

For Test Dataset :

SL NO	MODEL_NAME	Test MSE	Test RMSE	Test R^2	Test Adjusted R^2
1	Linear Regression	169871.70934024057	412.15495792267325	0.5624656403341544	0.5584307413401177
2	Lasso Regression	169871.72597325977	412.1549781007865	0.5624655974928983	0.5584306981037839
3	Ridge Regression	169871.77465071727	412.1550371531534	0.5624654721155751	0.5584305715702432
4	ElasticNet Regression	183719.51037262668	428.6251396880807	0.5267982017652659	0.522434381147539
5	DecisionTree Regressor	91626.47632392391	302.6986559664973	0.7639999514779179	0.7618235821543713
6	RandomForest Regressor	52363.04539891553	228.8297301464902	0.8651297992599828	0.863886039483706
7	Gradient Boost	67959.97028150507	260.69133142761973	0.8249571856578451	0.823342957975151
8	Xg Boost	68250.19807949613	261.2473886558412	0.8242096530978654	0.8225885317431483

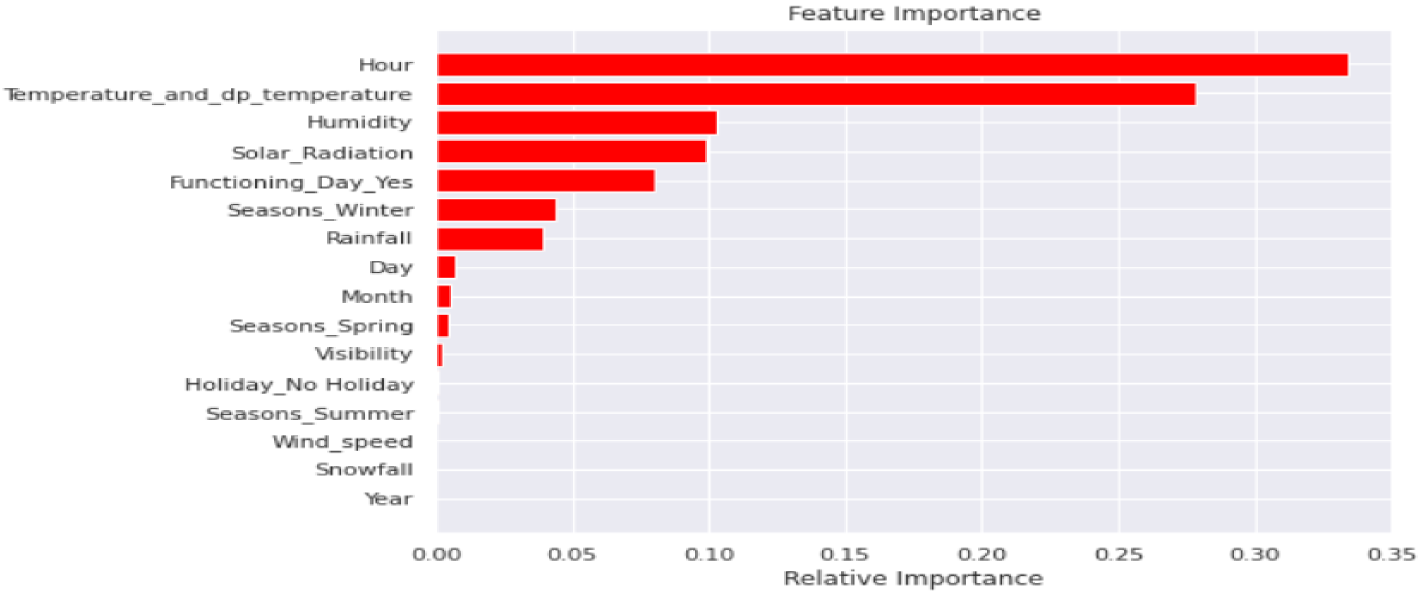
Model Validation & Selection (Feature Importance)

The Summary table shows that RandomForest and gradient boost are giving high R- Squared score. Let’s deep dive into Feature importance for both of these.



RandomForest Regressor

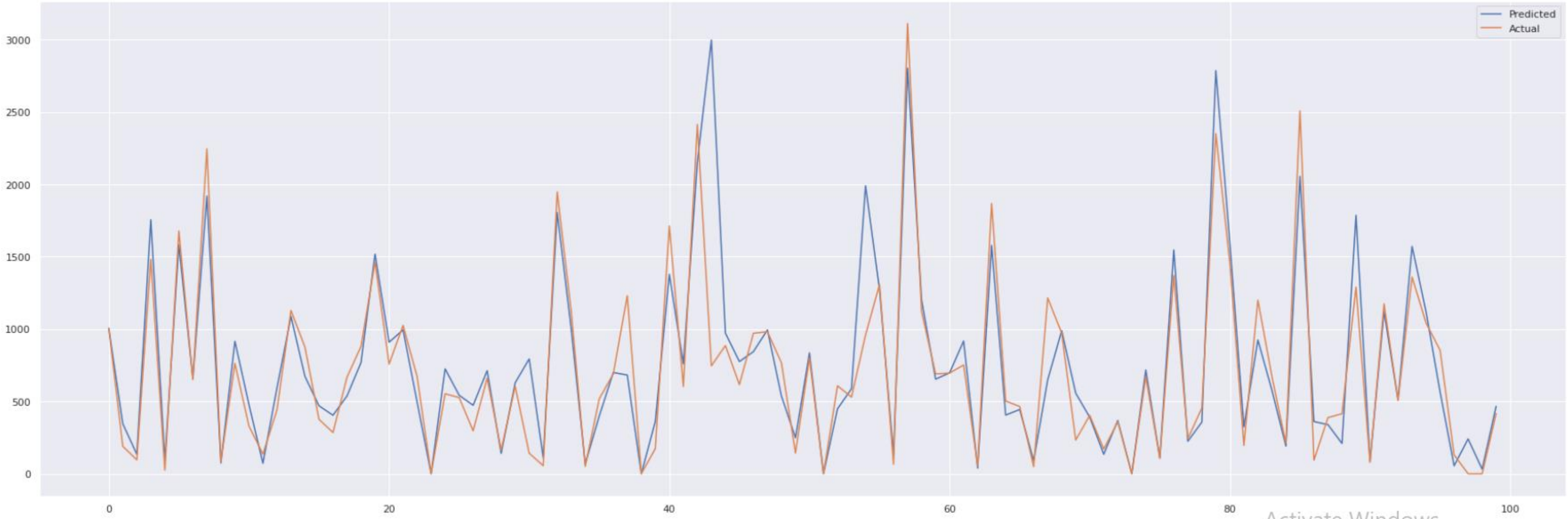
Model Validation & Selection (Feature Importance)



Gradient Boost Regressor

RandomForest Regressor(Graph)

Actual to predicted dependent variable line graph best shown by RandomForest, hence it is the best model to be implemented.



Conclusion

- As it was stated in the problem, rented bike count was low in 2017 untill november. After that rented bike count started increasing.
- There was sharp increase in demand from the end of 2017 that too in winter season of the year. The demand however decrease at the end of 2018.
- Bike count rent is highly correlated with 'Hour', which seems obvious. Demand for bike is mostly in morning (7 to 8) and in the evening (3 to 9).
- After doing exploratory data analysis, applying Linear Regression model didn't go quite well as it gave only 56% accuracy.
- Lasso and Ridge Regression helps to reduce model complexity and prevent over-fitting which may result from simple linear regression. with Lasso, ridge and ElasticNet regressor We got r squared value of 0.5624, 0.5624, 0.5267 respectively.
- With Decision Tree we are able to achieve the r2 score of 0.7639.
- Gradient Boost gave r squared value of 0.8249 on test data.
- XG Boost gave r squared value of 0.8242
- RandomForest Regressor gives higher value of R squared metric in train data 0.9834 and on test data it is 0.8651
- RandomForest Regressor came with best accuracy to approximate numbers of rented bikes demand. It gives amazing results of training r-square at 0.9834 and test r-square value at 0.8651 also with adjusted r-square with 0.8638.