AI maBetter

Capstone Project

BIKE SHARING DEMAND PREDICTION

By – Lavanya Shinde

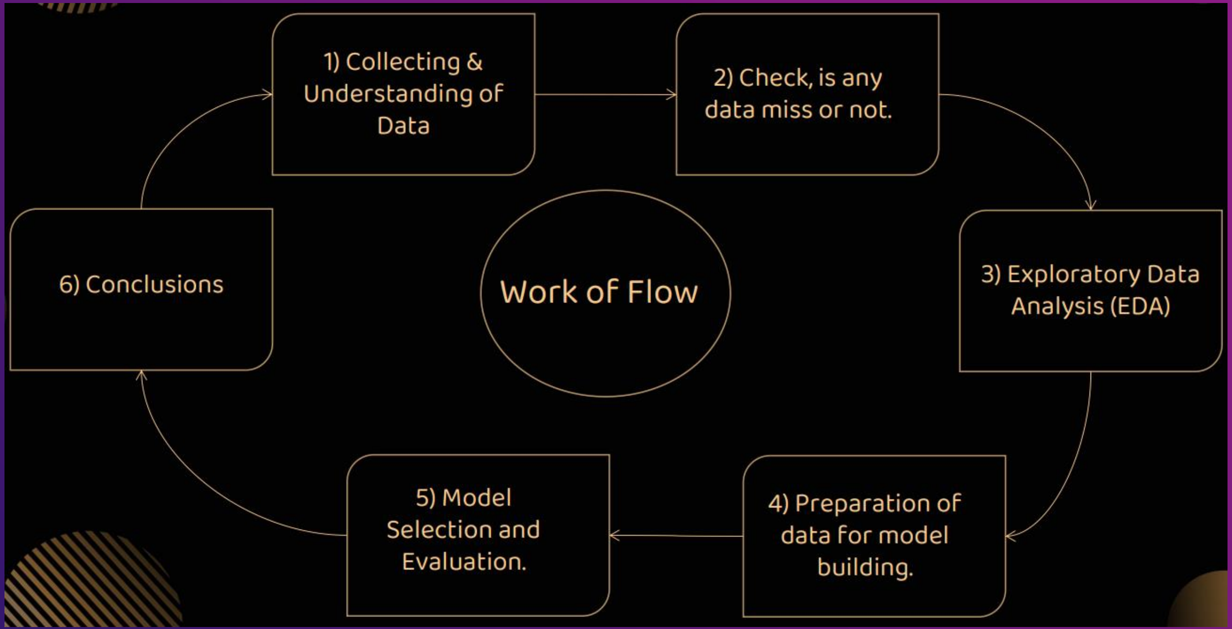
CONTENTS

01 PROBLEM STATEMENT

- 1 Bike sharing systems is a process of obtaining membership, rental bike and return it throughout the city. Using these systems, people are able rent a bike from a one location and return it to a different place as there need .
- 2. Most people having no personal vehicles and they avoid some congested public transport and that's why they want to use rental bikes .
- 3. That's why, this business going to make good profit and it has to be always ready to supply no. of bikes at different locations , to fulfil the demand .
- 4. In Analyzing the data we work with Seoul city Bike rental data, in this dataset include the information such as Date , Rented Bike Count , Hour , Temperature , Humidity & other information .

02 WORKFLOW

- Here is the Simple Work of Flow we use for our Project :-



03 DATA REVIEW

Let’s understand every columns which is contain in dataset :--

1. Date :- Contain Data in form of Year-Month-Day .
2. Rented Bike Count :- Number of bikes rented at each hour .
3. Hour :- Total Hour of The day .
4. Temperature (°C) :- Temperature data in Celsius .
5. Humidity (%) :- Humidity Data in % .
6. Wind speed (m/s) :- Wind Speed in m/s.
7. Visibility (10m) :- Shows the data of Visibility by 10m .
8. Dew point temperature (°C) :- Shows the data of Dew point temperature in Celsius .
9. Solar Radiation (MJ/m2) :- Shows the data of solar Radiation in Mj/m2.
10. Rainfall (mm) : - Rainfall Data in mm .

11. Snowfall (cm) :- Snowfall data in cm .
12. Seasons :- Seasons data such as winter, spring, summer, autumn .
13. Holiday :- Contain categorical data such as Holiday or No Holiday .
14. Functioning Day :- NoFunc (Non Functional Hours), Fun(Functional hours).

Let's Check Missing Value in Dataset. If some value is null in dataset , then we target every missing value to fill & make data complete .

```
## Find the missing value, show the total null values for each column and sort it in descending order
bike_data.isnull().sum().sort_values(ascending=False)[:10]
```

```
Date                          0
Rented Bike Count             0
Hour                          0
Temperature(°C)               0
Humidity(%)                   0
Wind speed (m/s)              0
Visibility (10m)              0
Dew point temperature(°C)     0
Solar Radiation (MJ/m2)       0
Rainfall(mm)                  0
dtype: int64
```

But, there are no null value in our dataset. So, data is perfect for start the project .

- There are No Missing Values present in Dataset
- There are No Duplicate values present in Dataset
- There are No null values.
- We change the name of some features for our convenience , they are as below 'Rented_Bike_Count', 'Hour', 'Temperature', 'Humidity', 'Wind_speed', 'Visibility', ‘Dew_point_temperature', 'Solar_Radiation', 'Rainfall', 'Snowfall', 'Seasons', 'Holiday', 'Functioning_Day', 'month','weekdays_weekend' .
- Also we Formating the date column .

For Better Analysing we change the column name , Because variable having units with name

```
[22] bike_data.rename(columns={'Rented Bike Count':'Rented_Bike_Count','Temperature(°C)':'Temperature','Humidity(%)':'Humidity','Wind speed (m/s)':'Wind_speed',
                         'Visibility (10m)':'Visibility','Dew point temperature(°C)':'Dew_point_temperature', 'Solar Radiation (MJ/m2)':'Solar_Radiation',
                          'Rainfall(mm)':'Rainfall','Snowfall (cm)':'Snowfall','Functioning Day':'Functioning_Day'},inplace=True)
     new_column_name  = pd.DataFrame(bike_data.columns).T
     new_column_name
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	Date	Rented_Bike_Count	Hour	Temperature	Humidity	Wind_speed	Visibility	Dew_point_temperature	Solar_Radiation	Rainfall	Snowfall	Seasons	Holiday	Functioning_Day

```
[23] # Formating the date column
     bike_data['Date'] = pd.to_datetime(bike_data['Date'], format = '%d/%m/%Y').dt.date
     pd.DataFrame(bike_data['Date'][:5])
```

	Date
0	2017-12-01
1	2017-12-01
2	2017-12-01
3	2017-12-01
4	2017-12-01

- Count the value from seasons, holidays, Functioning day, Month & Weekdays_or_weekend

Count the value from seasons, holidays, Functioning day, Month & Weekdays_or_weekend

```
pd.DataFrame(bike_data['Seasons'].value_counts()).T
```

	Spring	Summer	Autumn	Winter
Seasons	2208	2208	2184	2160

```
[ ] pd.DataFrame(bike_data['Holiday'].value_counts()).T
```

	No Holiday	Holiday
Holiday	8328	432

```
[ ] pd.DataFrame(bike_data['Functioning_Day'].value_counts()).T
```

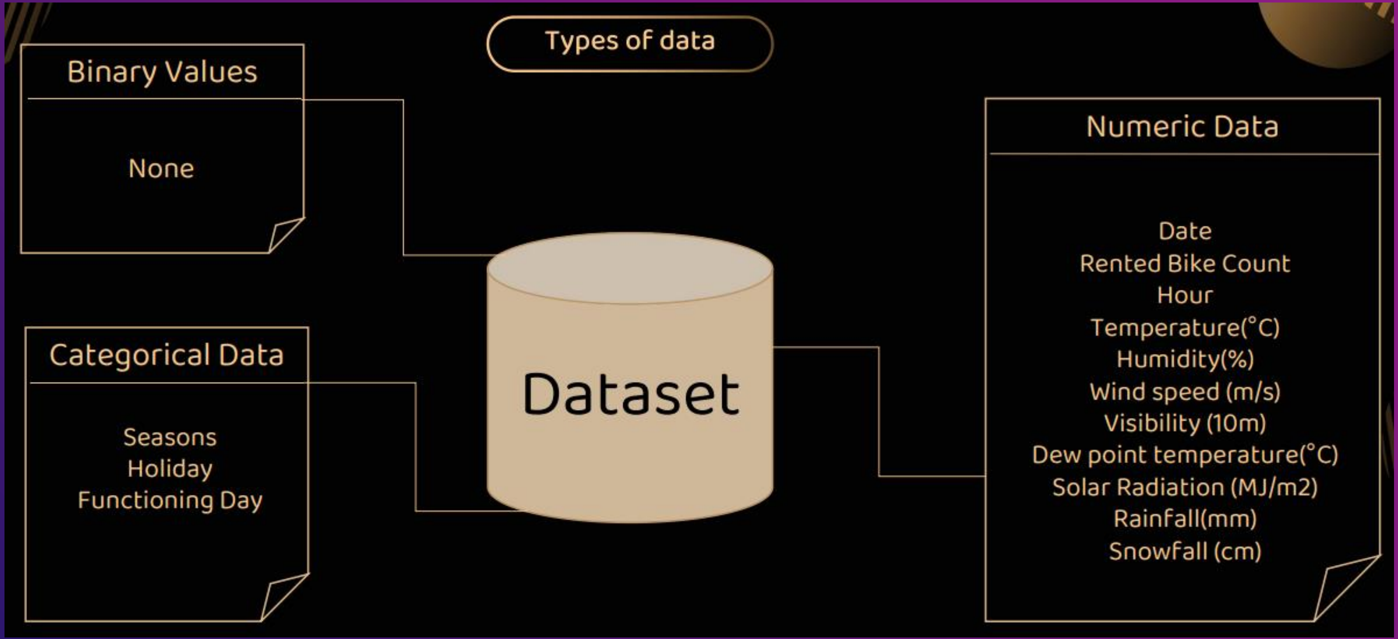
	Yes	No
Functioning_Day	8465	295

```
[ ] pd.DataFrame(bike_data['Month'].value_counts()).T
```

	12	1	3	5	7	8	10	4	6	9	11	2
Month	744	744	744	744	744	744	744	720	720	720	720	672

04

TYPES OF DATA IN DATASET


Types of data
Binary Values
None
Categorical Data
Seasons
Holiday
Functioning Day
Dataset
Numeric Data
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)

05

EXPLORATORY DATA ANALYSIS (EDA)

1. We Import all required library in code, so we take advantages of library to solve out our problem. If in future, we need more library, so we import in this Collab. Currently we add numpy, pandas , matplotlib , seaborn, pycountry etc.
2. Then we add our Data-Setfile i.e. excel file in it . Our Data-Setfile is in google drive, so we import google drive to link with that file& we import google drive then we give location of our file then call file with pandas library with the function of pd.read_csv() .This function read file excel file.

Importing Packages

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn

import warnings
warnings.filterwarnings('ignore')
```

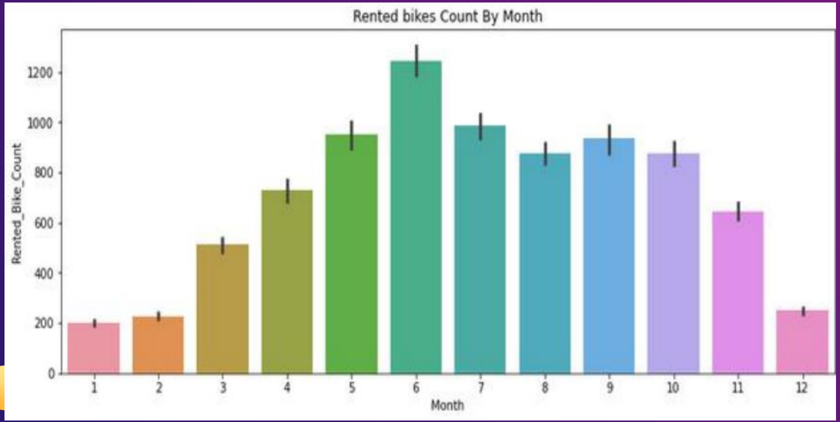
Importing Data From Google Drive

```
[ ] ## Importing Data From Google Drive
from google.colab import drive
drive.mount('/content/drive')

data_loc = '/content/drive/MyDrive/Almabetter/Capston Project/Bike Sharing Demand Prediction/SeoulBikeData.csv'
bike_data = pd.read_csv(data_loc,encoding='latin')
```

```
Mounted at /content/drive
```

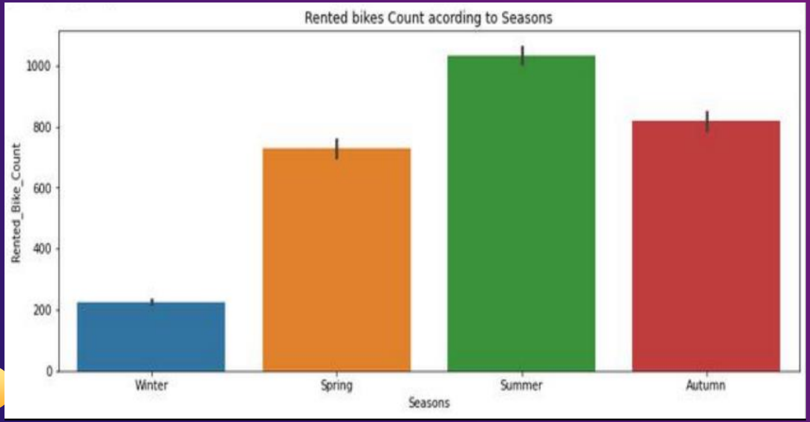
1. Rented bikes Count By Month



Key Insights:

According to visualization, we can say that from the monthof5(may) to 10 (oct) the demand of the rented bike is high with the compare to other months and June was the highest month for Rented Bikes Count.

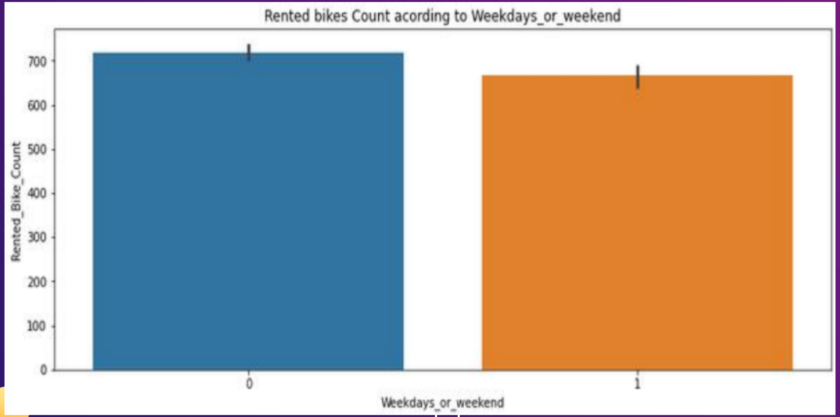
2. Rented bikes Count acording to Seasons



Key Insights:

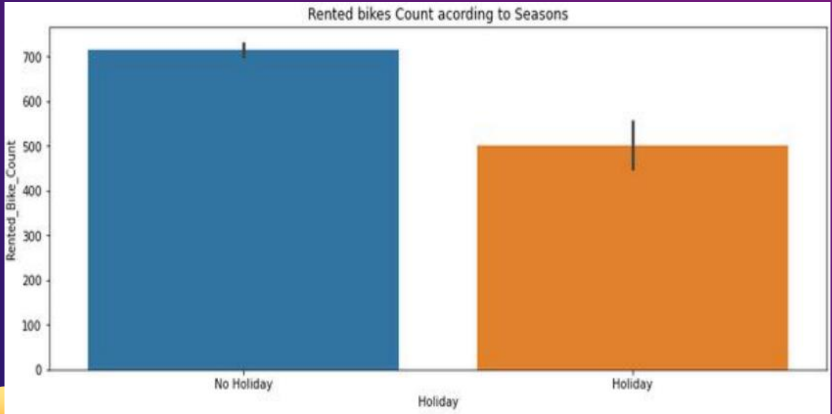
Chart shows, Summer season had the highest Bike Rent Count. So, People are love to rented bikes in summer season and winter season is very less compared to other season.

3. Rented bikes Count acording to weekdays_or_weekend



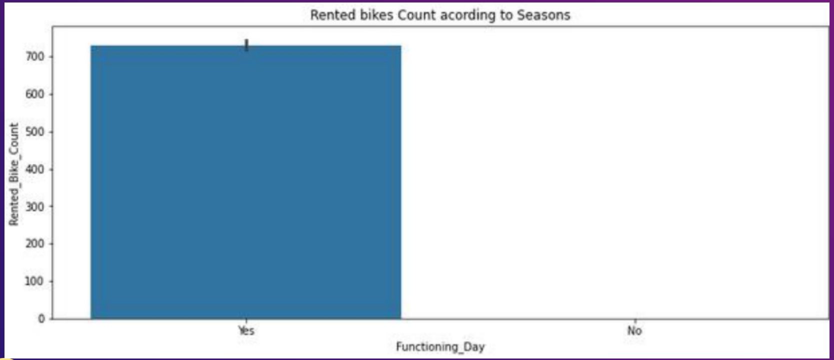
Key Insights:
According to visualization, More than 700 bikes were rented on weekdays. On weekend, almost 650 bikes were rented. So, weekdays rented more bikes with the comparison o fweekend.

4. Rented bikes Count acording to Holidays



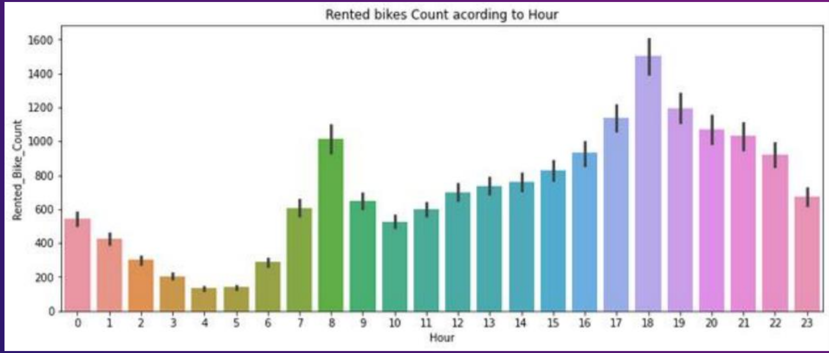
Key Insights:
Chart shows almost 700 bikes rented on No Holiday and nearly 500 bikes rented on holidays

5. Rented bikes Count acording to Functioning Day



Key Insights:
In this chart we can clearly see that, zero bikes were no functioning day and nearly 700 bikes rented on functioning day.

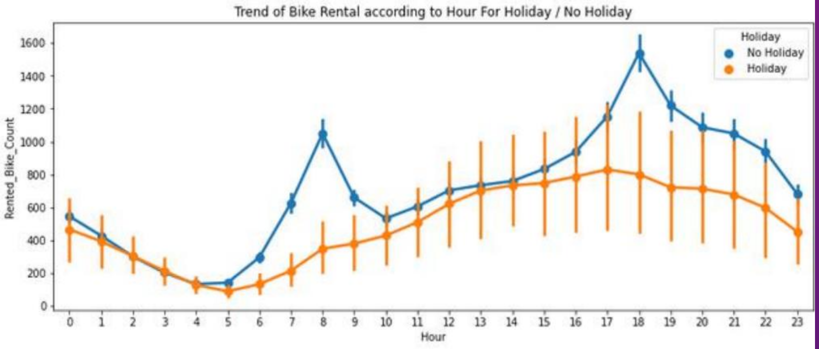
6. Rented bikes Count according to Hour



Key Insights:

In this chart we can clearly see that the use of rented bike according to the hours and the data are from all over the year. So, basically people use rented bikes during their working hour from 7am to 9am and 5pm to 7pm.

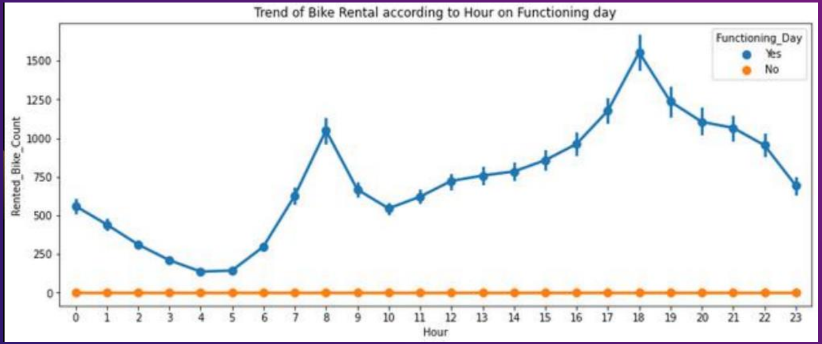
7. Trend of Bike Rental according to Hour For Holiday / No Holiday



Key Insights:

Now in this chart we observe there is a peak between 6AM to 10AM on no Holiday. Basically, people use this time for office & colleges. Another peak is between 4PM to 7PM & people use this time for college/office leaving.

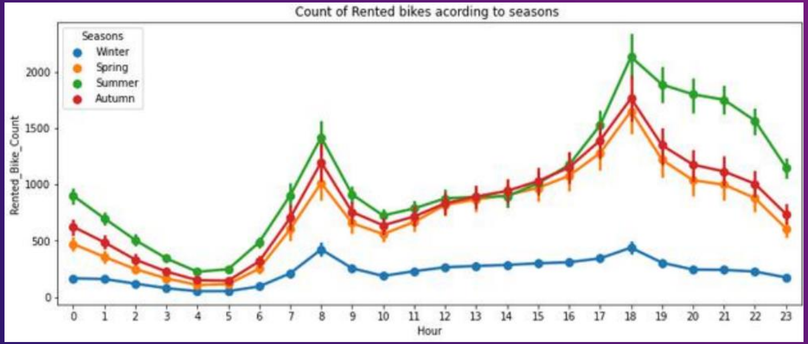
8. Trend of Bike Rental according to Hour on Functioning day



Key Insights:

Now in this chart we clearly observe people use rented bikes only on Functioning Day. Nobody use rented bikes on non-functioning Day.

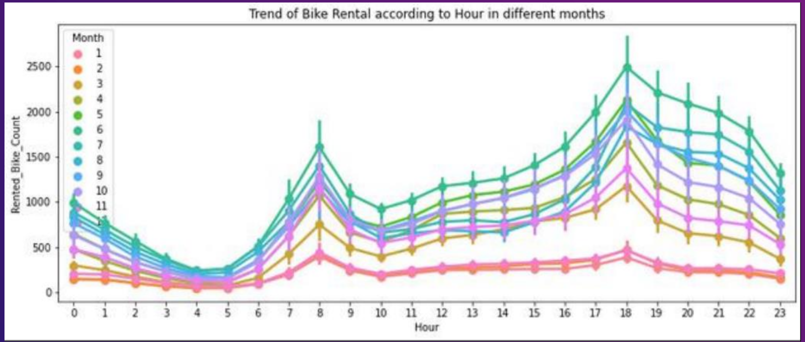
9. Count of Rented bikes according to seasons



Key Insights:

Now, this chart shows use of rented bikes in four different seasons. In this chart we observe summer season is very good for bike rental. As well as autumn is on 2nd highest season. Remaining season Spring & Winter are use low for bike rental maybe due to snowfall.

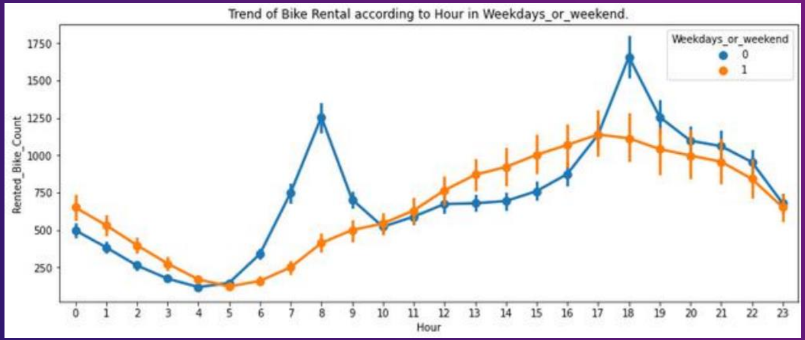
10. Trend of Bike Rental according to Hour in different months ?



Key Insights:

In this chart we observe that from Jun to Sept month mostly use month for bike rental and in this month most people use bike from 7am to 9am & 5pm to11pm.

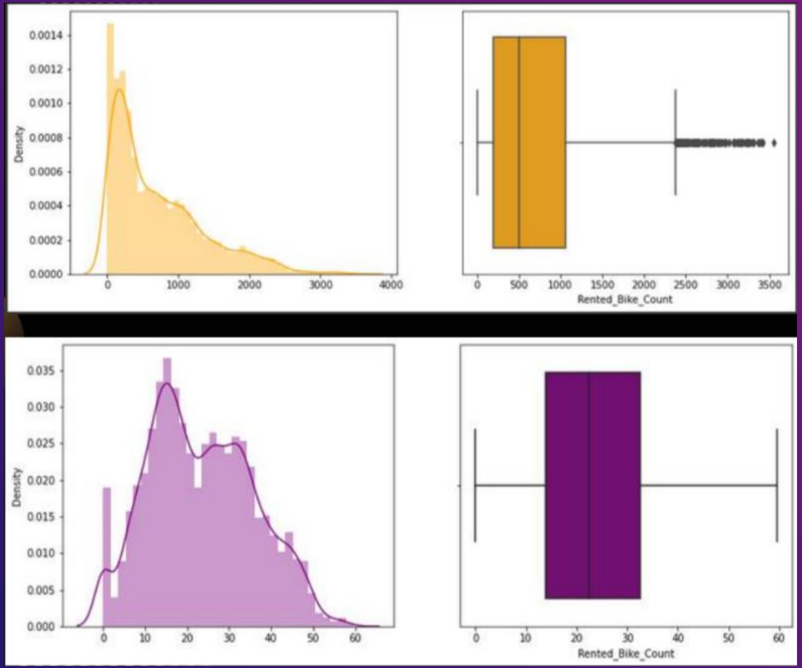
11. Trend of Bike Rental according to Hour in Weekdays_or_weekend



Key Insights:

In this chart, we observe that trend of bike rental according to Hour in Weekdays is on High Trend on near 7am to 9am & 5pm to 8pm for weekdays.

12) Distribution of target variables - "BikeRentedCount"

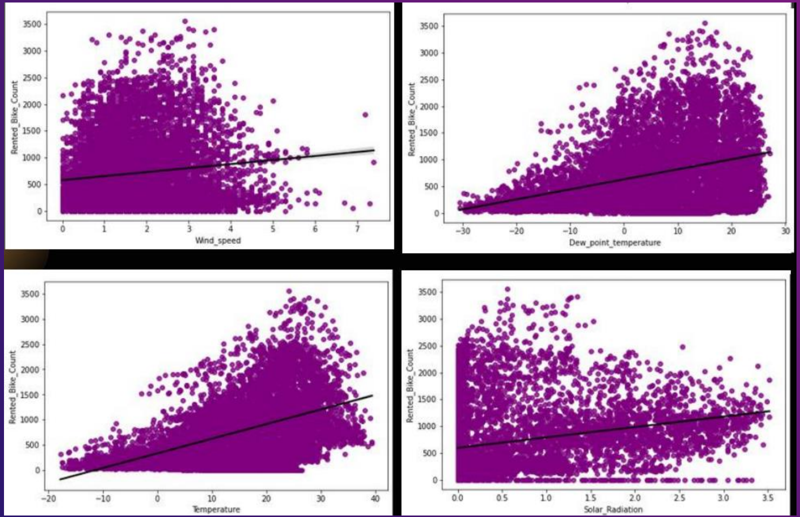


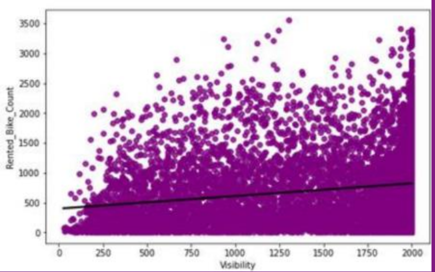
Key Insights:

1. In 1st (Yellow Chart), we observe Distribution is rightly skewed and some outliers are observed.

2. In 2nd (Purple Chart) by Using Square Root Method we Normalize our target variable. There are no outliers were found afte rnormalization .

13) Distribution of target variables - "Bike Rented Count"

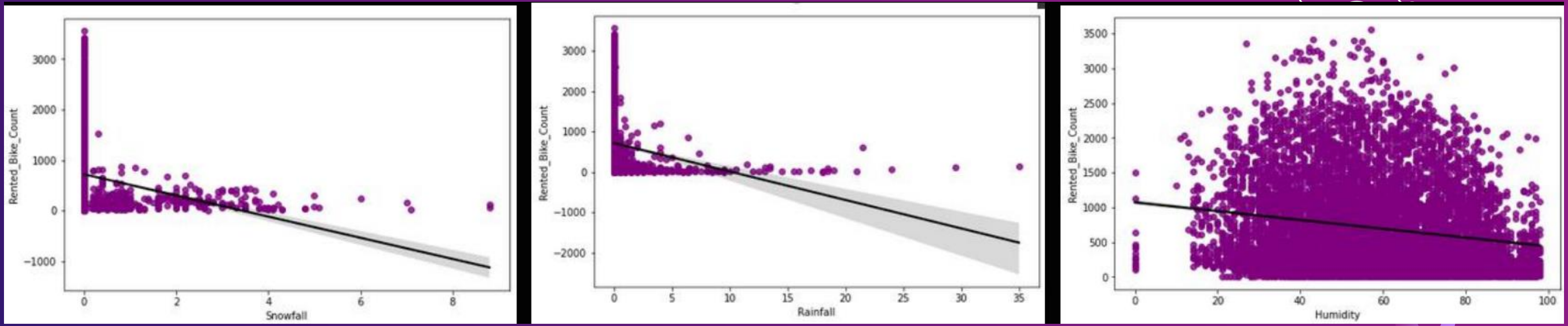




Key Insights:

1. In 1st (Yellow Chart), we observe Distribution is rightly skewed and some outliers are observed.
2. In 2nd (Purple Chart) by Using Square Root Method we Normalize our target variable. There are no outliers were found after normalization .

14) Distribution of target variables - "Bike Rented Count"

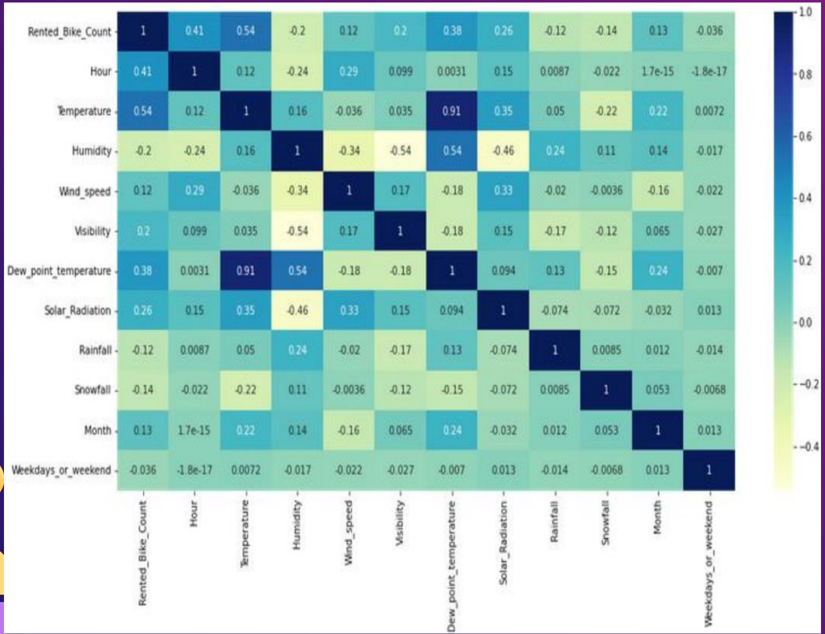


Key Insights:

1. In 1st (Yellow Chart), we observe Distribution is rightly skewed and some outliers are observed.

2. In 2nd (Purple Chart) by Using Square Root Method we Normalize our target variable. There are no outliers were found after normalization .

Correlation Matrix between dependent and independent variable.



Key Insights:

Variables like Temperature & Dew Point Temperatures are highly Correlated nearly 91%. So, we dropped the Dew point temperature because it has very low correlation with our target variable as compared to temperature.

Now we Start Model Building For :-

1) LINEAR REGRESSION
2) LASSO REGRESSION
3) RIDGE REGRESSION
4) ELASTIC NET REGRESSION
5) DECISION TREES REGRESSOR
6) RANDOM FOREST REGRESSOR
7) GRADIENT BOOSTED REGRESSOR
8) GRADIENT BOOSTING REGRESSOR WITH GRIDSEARCHCV

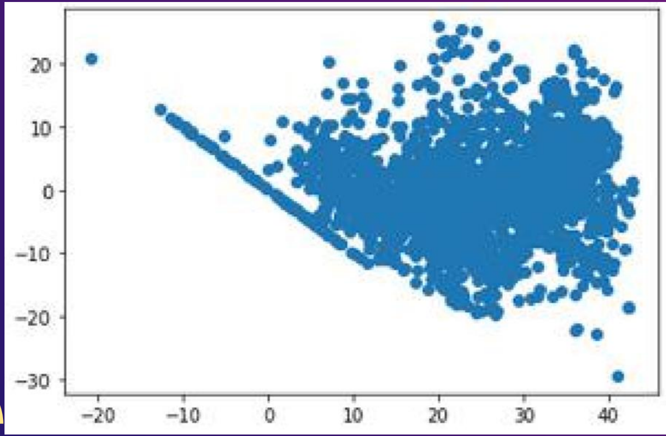
1) Linear Regression

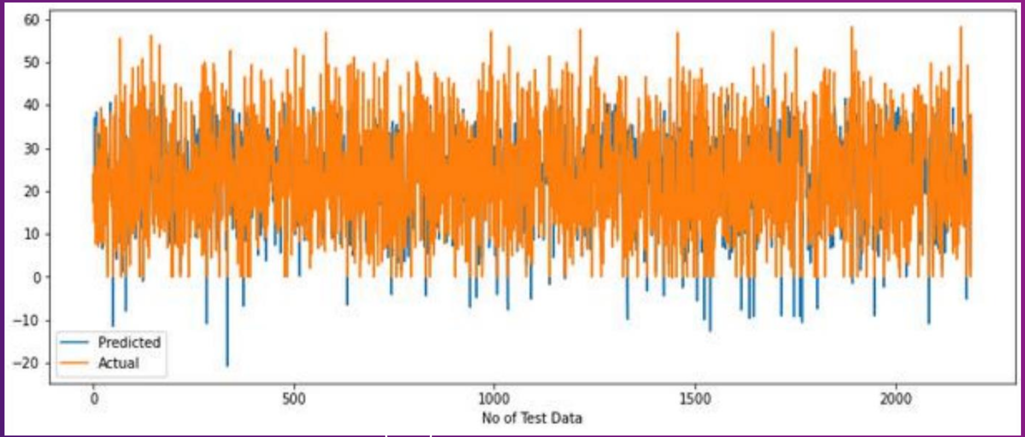
Result on Train Test

MSE : 53.080960809327934
RMSE :- 7.28566817864552
MAE :- 5.586424669493191
R2 :- 0.6552975724025564
Adjusted R2 :- 0.6527594965435785

Result on Test Set

MSE : 52.84573767539748
RMSE : 7.269507388771091
MAE : 5.608326408788622
R2 : 0.6654621707125412
Adjusted R2 : 0.6629989377311334





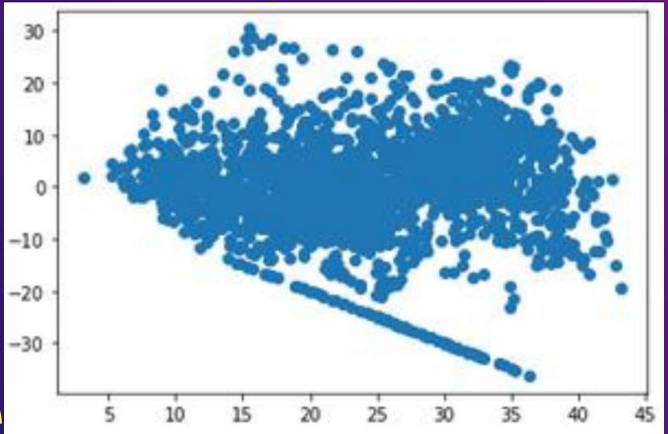
2) Lasso Regression

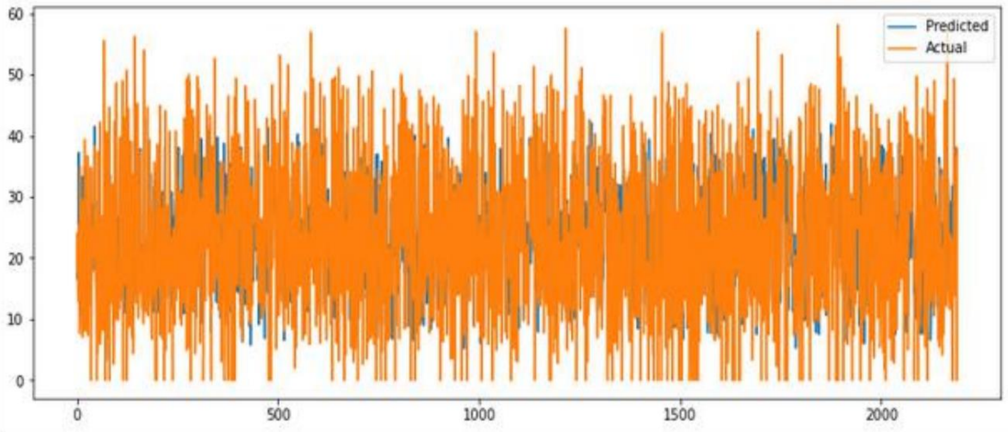
Result on Train Test

MSE : 80.53531396270661
RMSE : 8.974146976883464
MAE : 6.659731166521835
R^2 : 0.47701176077074947
Adjusted R^2 : 0.4731609499894941

Result on Test Set

MSE : 86.43678576363727
RMSE : 9.297138579349953
MAE : 6.8652938771568115
R^2 : 0.45281538394695453
Adjusted R^2 : 0.44878641300500854





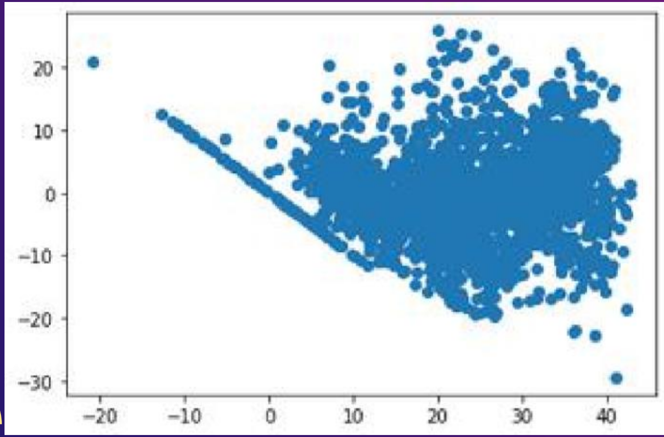
3) Ridge Regression

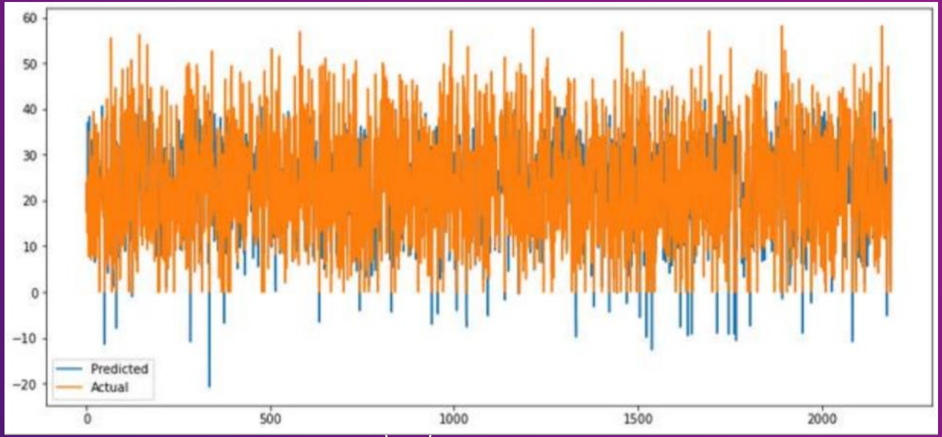
Result on Train Test

MSE : 53.08096841160499
RMSE : 7.2856687003737
MAE : 5.586440416080089
R2 : 0.6552975230341327
Adjusted R2 0.6527594468116504

Result on Test Set

MSE : 52.84593221813509
RMSE : 7.269520769496094
MAE : 5.608416221410825
R2 : 0.6654609391675197
Adjusted R2 : 0.662997697118132





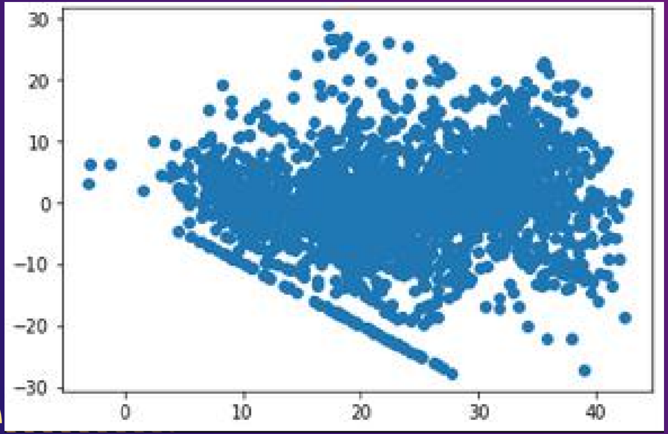
4) Elastic Net Regression

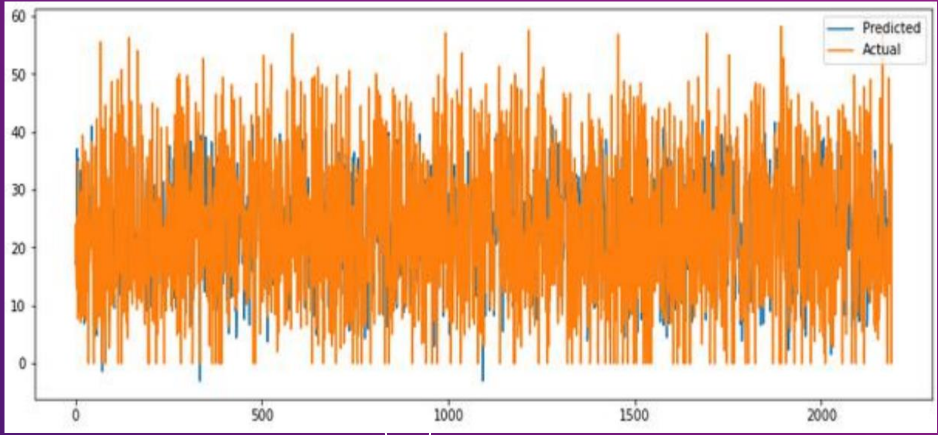
Result on Train Test

MSE : 64.13060361800518
RMSE : 8.008158565987888
MAE : 6.071434726026881
R2 : 0.5835423019221027
Adjusted R2 : 0.5804758853692972

Result on Test Set

MSE : 66.72858042048135
RMSE : 8.168756357027755
MAE : 6.195878517871 55
R2 : 0.5775773898280898
Adjusted R2 : 0.5744670530757886

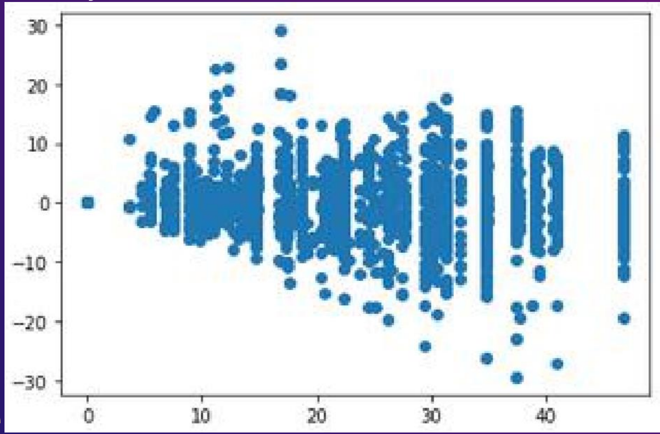




5) Decision Tree Regression

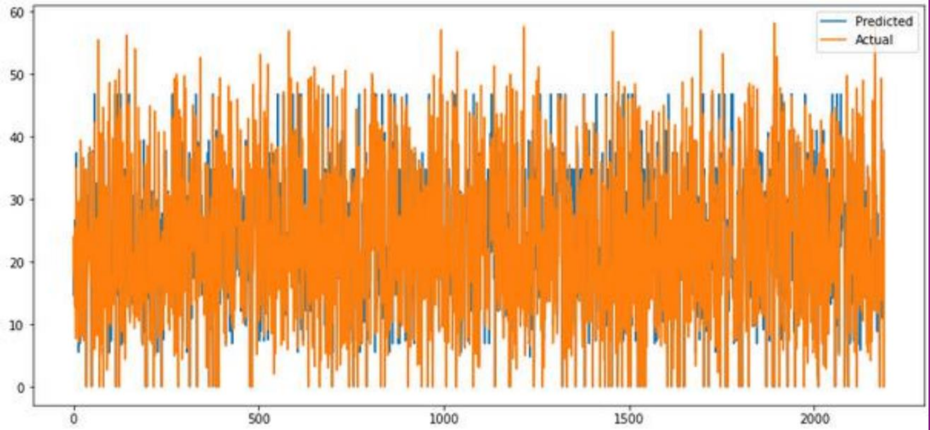
Result on Train Test

Model Score: 0.8393502705555953
MSE : 24.73856088598902
RMSE : 4.973787378446029
MAE : 3.60924255963705
R2 : 0.8393502705555953
Adjusted R2 : 0.838167391737781

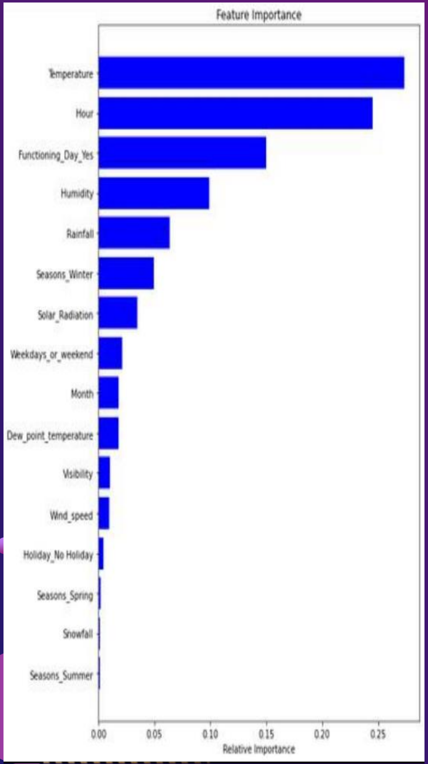


Result on Test Set

MSE : 28.895079669529146
RMSE : 5.375414371890705
MAE : 3.822269987246837
R2 : 0.8170808535381135
Adjusted R2 : 0.8157340029429041

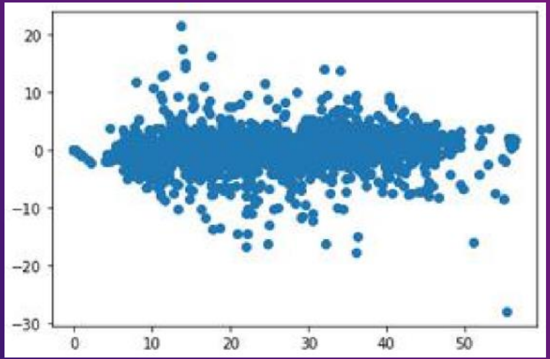


6) Random Forest Regression



Result on Train Test

Model Score: 0.9914237970816888
MSE : 1.320655308907066
RMSE : 1.149197680517615
MAE : 0.7196237222579589
R2 : 0.9914237970816888
Adjusted R2 : 0.9913606497063124

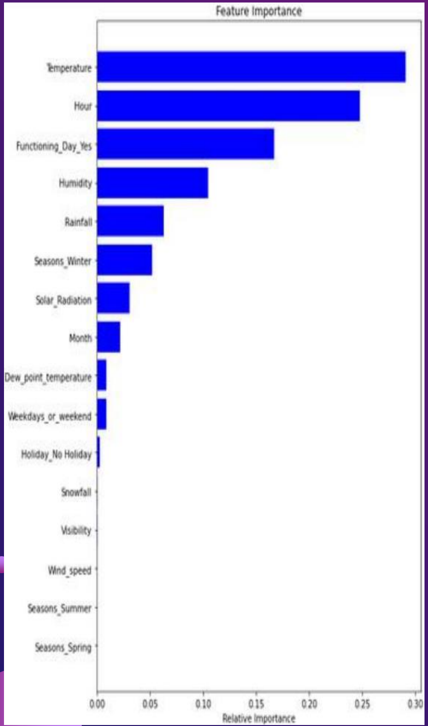


Result on Test Set

MSE : 9.814358154467069RMSE : 3.1327876012374456MAE : 1.9952888446617798R2 : 0.9378705981357959Adjusted R2 : 0.9374131336029715

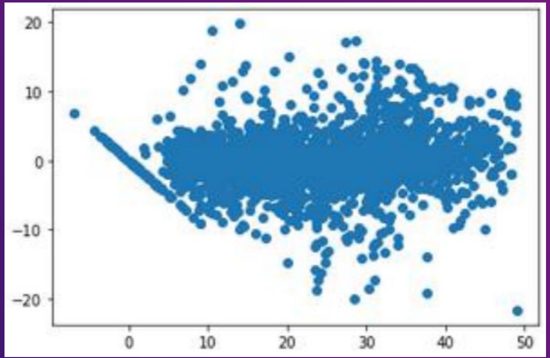
	Feature	Feature Importance
1	Temperature	0.27
0	Hour	0.24
15	Functioning_Day_Yes	0.15
2	Humidity	0.11
7	Rainfall	0.06
13	Seasons_Winter	0.05
6	Solar_Radiation	0.04
5	Dew_point_temperature	0.02
9	Month	0.02
10	Weekdays_or_weekend	0.02
3	Wind_speed	0.01
4	Visibility	0.01
8	Snowfall	0.00
11	Seasons_Spring	0.00
12	Seasons_Summer	0.00
14	Holiday_No Holiday	0.00

7) Gradient Boosted Regression



Result on Train Test

Model Score: 0.9001696544113085
MSE : 15.372942681922371
RMSE : 3.9208344369435406
MAE : 2.8013746972643125
R2 : 0.9001696544113085
Adjusted R2 : 0.8994345943425468

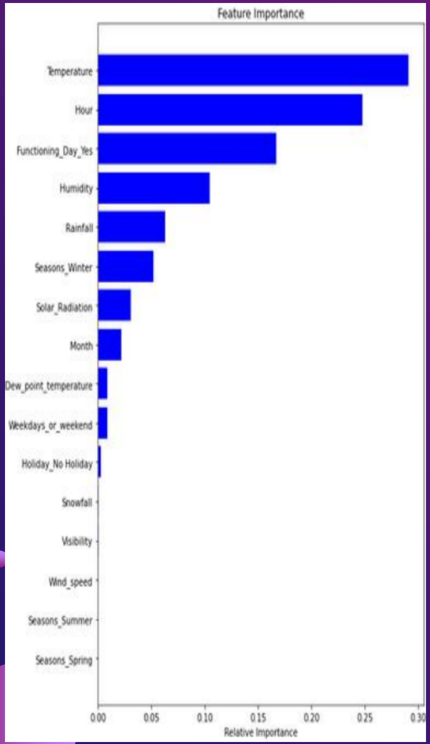


Result on Test Set

MSE : 17.588973973146366RMSE : 4.193921073786006MAE : 2.991383727015173R2 : 0.8886537035680477Adjusted R2 : 0.887833850488015

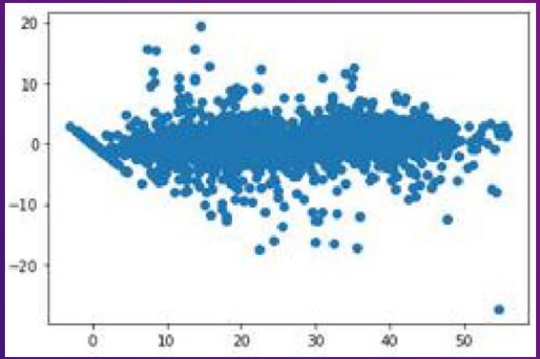
	Feature	Feature Importance
1	Temperature	0.29
0	Hour	0.25
15	Functioning_Day_Yes	0.17
2	Humidity	0.10
7	Rainfall	0.06
13	Seasons_Winter	0.05
6	Solar_Radiation	0.03
9	Month	0.02
5	Dew_point_temperature	0.01
10	Weekdays_or_weekend	0.01
3	Wind_speed	0.00
4	Visibility	0.00
8	Snowfall	0.00
11	Seasons_Spring	0.00
12	Seasons_Summer	0.00
14	Holiday_No Holiday	0.00

8) Gradient Boosting Regressor with Gridsearchcv



Result on Train Test

Model Score: 0.9688779196560818
MSE : 4.792510277791052
RMSE : 2.189180275306502
MAE : 1.421909456944972
R2 : 0.9688779196560818
Adjusted R2 : 0.9686487648997529



Result on Test Set

MSE : 8.999159624968756RMSE :
2.999859934225056MAE :
1.951219472831746R2 :
0.9430311798306117Adjusted R2 :
0.9426117131381542

	Feature	Feature Importance
1	Temperature	0.27
0	Hour	0.26
15	Functioning_Day_Yes	0.15
2	Humidity	0.11
7	Rainfall	0.06
13	Seasons_Winter	0.06
6	Solar_Radiation	0.03
9	Month	0.02
10	Weekdays_or_weekend	0.02
5	Dew_point_temperature	0.01
3	Wind_speed	0.00
4	Visibility	0.00
8	Snowfall	0.00
11	Seasons_Spring	0.00
12	Seasons_Summer	0.00
14	Holiday_No Holiday	0.00

05 CONCLUSION

During the time of our analysis, we initially did EDA on all the features of our datset. We first analyzed our dependent variable, 'Rented Bike Count' and also transformed it. Next we analyzed categorical
variable and dropped the variable who had majority of one class, we also analysed numerical variable, found out the correlation, distribution and their relationship with the dependent variable. We also removed some numerical features who had mostly 0 values and hot encoded the categorical variables. Next we implemented 8 machine learning algorithms Linear Regression, Lasso Regression, Ridge Regression , Elastic-net Regression , Decision Tree Regression, Random Forest Regression, Gradient Boosted Regression and Gradient Boosting Regressor with Gridsearchcv. We did hyper parameter tuning to improve our model performance. The results of our evaluation are:

		Model	MAE	MSE	RMSE	R2_score	Adjusted R2
Training set	0	Linear regression	5.586	53.081	7.286	0.655	0.65
	1	Lasso regression	6.660	80.535	8.974	0.477	0.47
	2	Ridge regression	5.586	53.081	7.286	0.655	0.65
	3	Elastic net regression	6.071	64.131	8.008	0.584	0.58
	4	Dicision tree regression	3.609	24.739	4.974	0.839	0.84
	5	Random forest regression	0.720	1.321	1.149	0.991	0.99
	6	Gradient boosting regression	2.801	15.373	3.921	0.900	0.90
	7	Gradient Boosting gridsearchcv	1.422	4.793	2.189	0.969	0.97
Test set	0	Linear regression	5.608	52.846	7.270	0.665	0.66
	1	Lasso regression	6.865	86.437	9.297	0.453	0.45
	2	Ridge regression	5.608	52.846	7.270	0.665	0.66
	3	Elastic net regression Test	6.196	66.729	8.169	0.578	0.57
	4	Dicision tree regression	3.822	28.895	5.375	0.817	0.82
	5	Random forest regression	1.995	9.814	3.133	0.938	0.94
	6	Gradient boosting regression	2.991	17.589	4.194	0.889	0.89
	7	Gradient Boosting gridsearchcv	1.951	8.999	3.000	0.943	0.94

- No overfitting is seen.
- Random forest Regressor and Gradient Boosting gridsearchcv gives thehighestR2scoreof 99% and 95% respectively for TrainSetand92% for Test set.
- Feature Importance value for Random Forest and Gradient Boost are different.
- We can deploy this model.

However, this is not the ultimate end. As this data is time dependent, the values for variables like temperature, windspeed, solar radiation, etc., will not always be consistent. Therefore, there will be scenarios where the model might not perform well. As Machine learning is an exponentially evolving field, we will have to be prepared for all contingencies and also keep checking our model from time to time. Therefore, having a quality knowledge and keeping pace with the ever evolving ML field would surely help one to stay astep ahead in future.

THANK YOU!