Seoul Bike Sharing Demand Prediction

Md Mahfooz Alam Ansari
Data Science Enthusiast,
Almabetter

Abstract:

Bike sharing is a transport service which mainly focuses to lend conventional or electrical bikes to an individual or a group of individuals for an hour, a day or for a month depending on the needs. Using these systems, people are able rent a bike from a one location and return it to a different place on an asneeded basis

In market share we can see that Bike Sharing system has a global market share which was valued around 3.39 billion Dollars in 2019 and is projected to grow to 6.98 billion Dollars by 2027 with a compound annual growth rate of around 14% indicatively from 2020 to 2027.

Several factors such as low bike rent, increase in capital investments, introduction of e-bikes in the market, technological advancement and government schemes for development of several bike-sharing infrastructure has increased the overall market share and led to the introduction of several opportunities during the forecasted year. However, rise in bike theft and huge initial investment are some of the key factors in order to hinder expected market growth.

Keywords: Bike-Sharing, Data Mining, Predictive Analysis, Linear Regression, Machine Learning.

1. Problem Statement

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour, and date information.
Based on the given data we have to build a machine learning model which will be helping us to predict the number of bikes that must be made available by predicting the demand for bikes rented per day.

- **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 Hours), Fun(Functional hours)

2. Introduction:

Bike sharing system demand nowadays is increasing in proportional manners globally. This system has gained a lot of attention with its cost-effective system and easy to use nature. This system has already attracted a huge customer base globally like in South
Korea, São Paulo, China and Australia.

Bike sharing system generally rents bikes on an hour, day and month basis and is generally based on static pricing inclusive of hour, days or month. Because of its affordability and easy renting system anyone can commute on arrival.

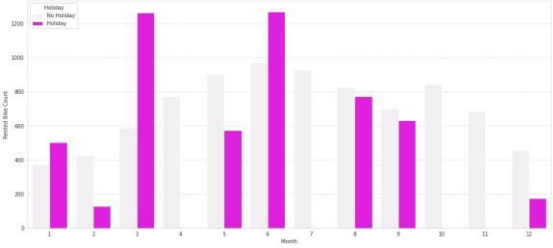
According to our problem our main aim is to build a predictive model so as to find the number of bikes rented based on the given dataset.

3. Exploratory Data Analysis

Exploratory Data Analysis (EDA) plays a vital role in the analysis of the data variables which are important from the aspect of feature engineering. It will help us to distribute and relate between dependent and independent variables. We have gone through an analysis of every independent as well as the dependent variable to check which independent factor affects the dependent factor.

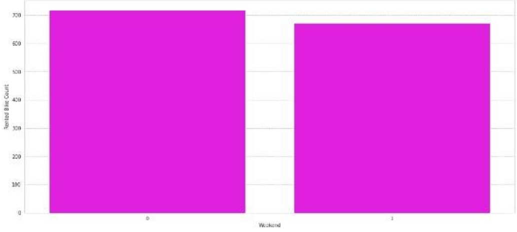
3.1 Month based Analysis for Holiday & Non-Holiday

The month-based Analysis showed that In June maximum bike rented near 1000. and January, February enjoys less rented bike demand near 400. March and June enjoy more bike sharing demand in holiday than nonholidays. February, December have less bike sharing demand for both in holidays and nonholidays. It also shows April, October, December months have nearly zero bike sharing demand in holidays.



3.2Weekday and Weekend based Analysis

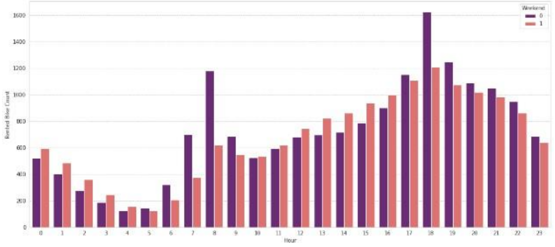
The Weekday and Week off based Analysis shows almost equal weightage on rented bike count



3.3 Hourly Rented Bike Count in Weekdays and Weekend

The graph shows that Hour-18 or 6pm shows maximum rented bike demand in both weekdays (above 1600) and weekend (above 1200). Hour 4 & Hour 5 (means 4 and 5 am) shows very less rented bike demand in both weekend and weekdays.

Hour 8 (8 am) shows good rented bike demand in weekdays (near 1200) but in weekend 8am hour has not so good bike sharing demand (near 600)

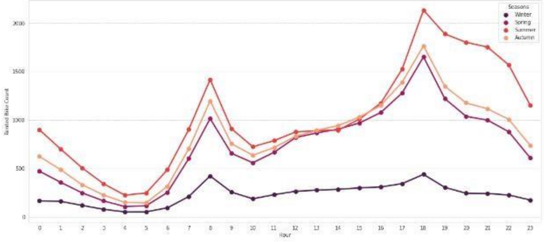


3.4 Season-wise Hourly Rented Bike Count Analysis

From this graph we can conclude that summer season enjoys overall best and least bike sharing demand and winter has overall less demand than any other season. Hour-18(6pm) and Hour8(8am) are two best peak time in any season when bike sharing is in high-demand. there is not so much difference in demand (near 1000) from Hour-12 to Hour-16(12pm-4pm) among summer, spring, autumn season.

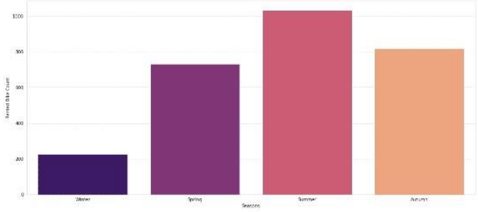
For every season Hour-4 and 5(4 & 5 am) shows low demand in bike sharing.

After Hour-10(10am) bike sharing demand is increasing up to Hour-18(6pm) then it is decreasing.



3.5 Season-wise Analysis

During the season-wise analysis, it was found that the month plays a significant role in rented bike demands. The demands are most likely to be high during summer followed by autumn and spring while winter shows the least demand.

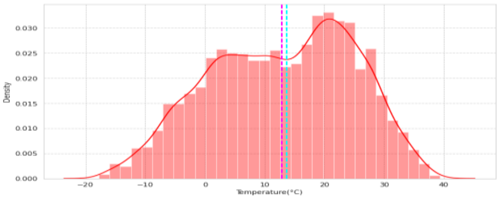


3.6 Analysing Numerical Variables

The numerical variables of the data set include Temperature(°C), Humidity (%), Wind Speed (m/s), Visibility (10m), Dew Point Temperature(°C), Solar Radiation (MJ/m^2), Rainfall (mm) Snowfall (cm). All the independent variables listed here represent the weather of the city which has a crucial role in rented bike demand deviation.

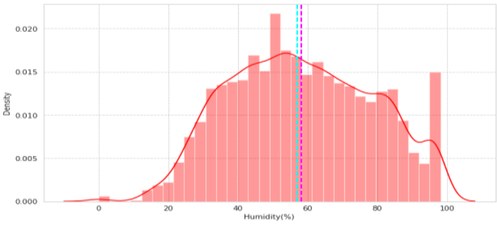
3.6.1. Temperature

In the density plot for **Temperature** we can see that the median is greater than the mean we can say to some extent that this is negatively skewed.



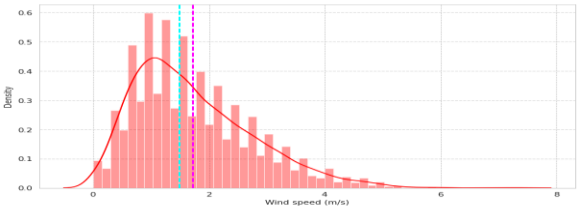
3.6.2 Humidity (%)

In the density plot for **Humidity** we can see that the mean is greater than the median we can say to some extent that this is positively skewed.



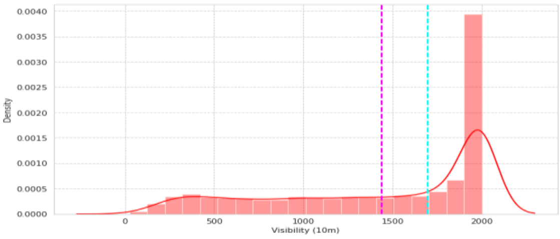
3.6.3 Wind Speed (m/s):

In density plot for **Windspeed** we can see that mean is greater than the median we can say to some extent that this is positively skewed.



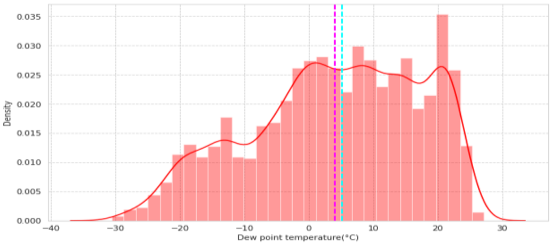
3.6.4 Visibility

In the density plot for **Visibility,** we can see that median is greater than mean we can say to some extent that this is negatively skewed.



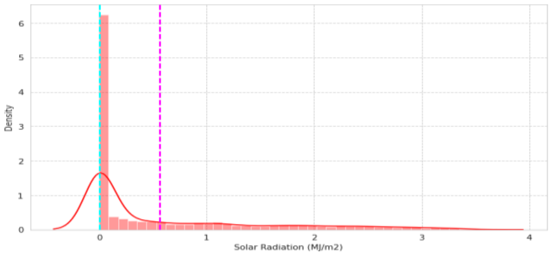
3.6.5 Dew Point Temperature (°C)

In the density plot for **Dewpoint Temperature** we can see that median is greater than mean we can say to some extent that this is negatively skewed.



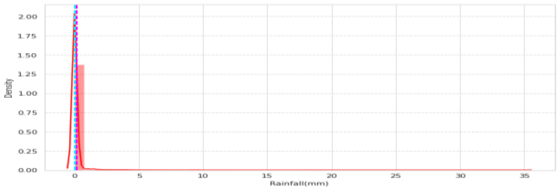
3.6.6 Solar Radiation

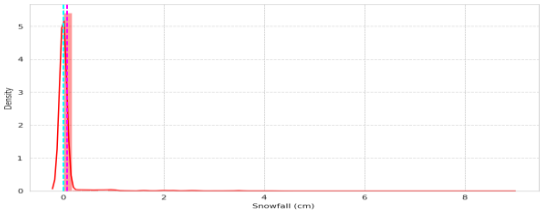
In density plot for **Solar Radiation** we can see that mean is greater than median we can say that this is positively skewed.



3.6.7 Rainfall and Snowfall

The average rainfall and snowfall in Seoul are 2mm and 2cm respectively. The regression plot shows a similar decrease in the Rented Bike Count with an increase in rainfall and snowfall. It is obvious that the less the rainfall and snowfall is, the more the rented bike count which indicates the public prefers to stay in shelter during heavy rain or snowfall.



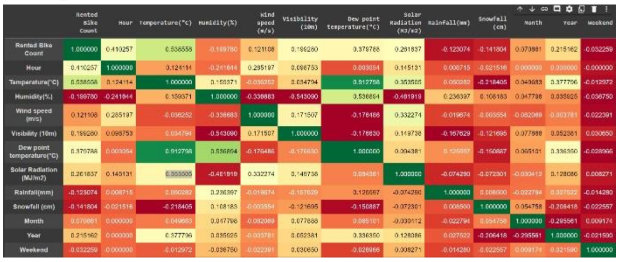


4. Correlation Analysis

The correlation analysis has been done to get a better understanding of dependent and independent variables' multicollinearity. Multicollinearity may not affect the accuracy of the model as much but we might lose reliability in determining the effects of individual independent features on the dependent feature in your model and that can be a problem when we want to interpret your model.

4.1 Heatmap

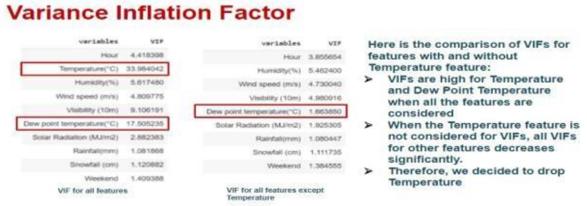
Let's check the heatmap plotted concerning independent variables.



	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)	Month	Year	Weekend
Rented Bike Count	1.000000	0.410257	0.538558	-0.199780	0.121108	0.199280	0.379788	0.261837	-0.123074	-0.141804	0.070861	0.215162	-0.032259
Hour	0.410257	1.000000	0.124114	-0.241644	0.285197	0.098753	0.003054	0.145131	0.008715	-0.021516	0.000000	0.000000	-0.000000
Temperature(°C)	0.538558	0.124114	1.000000	0.159371	-0.036252	0.034794	0.912798	0.353505	0.050282	-0.218405	0.049683	0.377796	-0.012972
Humidity(%)	-0.199780	-0.241644	0.159371	1.000000	-0.336683	-0.543090	0.536894	-0.461919	0.236397	0.108183	0.047798	0.035925	-0.036750
Wind speed (m/s)	0.121108	0.285197	-0.036252	-0.336683	1.000000	0.171507	-0.176486	0.332274	-0.019674	-0.003554	-0.082069	-0.003781	-0.022391
Visibility (10m)	0.199280	0.098753	0.034794	-0.543090	0.171507	1.000000	-0.176630	0.149738	-0.167629	-0.121695	0.077888	0.052381	0.030650
Dew point temperature(°C)	0.379788	0.003054	0.912798	0.536894	-0.176486	-0.176630	1.000000	0.094381	0.125597	-0.150887	0.065101	0.336350	-0.028966
Solar Radiation (MJ/m2)	0.261837	0.145131	0.353505	-0.461919	0.332274	0.149738	0.094381	1.000000	-0.074290	-0.072301	-0.030412	0.128086	0.008271
Rainfall(mm)	-0.123074	0.008715	0.050282	0.236397	-0.019674	-0.167629	0.125597	-0.074290	1.000000	0.008500	-0.022794	0.027522	-0.014280
Snowfall (cm)	-0.141804	-0.021516	-0.218405	0.108183	-0.003554	-0.121695	-0.150887	-0.072301	0.008500	1.000000	0.054758	-0.206418	-0.022557
Month	0.070861	0.000000	0.049683	0.047798	-0.082069	0.077888	0.065101	-0.030412	-0.022794	0.054758	1.000000	-0.295561	0.009174
Year	0.215162	0.000000	0.377796	0.035925	-0.003781	0.052381	0.336350	0.128086	0.027522	-0.206418	-0.295561	1.000000	-0.021590
Weekend	-0.032259	-0.000000	-0.012972	-0.036750	-0.022391	0.030650	-0.028966	0.008271	-0.014280	-0.022557	0.009174	-0.021590	1.000000

We can infer the following from the above heatmap

Temperature and Dew Point Temperature (feels like temperature) are highly correlated, as one would expect. Let's check the variance inflation factor for the data



Variance Inflation Factor

variables	VIF
Hour	4.418398
Temperature(°C)	33.984042
Humidity(%)	5.617480
Wind speed (m/s)	4.809775
Visibility (10m)	9.106191
Dew point temperature(°C)	17.505235
Solar Radiation (MJ/m2)	2.882383
Rainfall(mm)	1.081868
Snowfall (cm)	1.120882
Weekend	1.409388

VIF for all features

variables	VIF
Hour	3.855654
Humidity(%)	5.462400
Wind speed (m/s)	4.730040
Visibility (10m)	4.980916
Dew point temperature(°C)	1.663850
Solar Radiation (MJ/m2)	1.925305
Rainfall(mm)	1.080447
Snowfall (cm)	1.111735
Weekend	1.384555

VIF for all features except Temperature

Here is the comparison of VIFs for features with and without Temperature feature:

- VIFs are high for Temperature and Dew Point Temperature when all the features are considered
- When the Temperature feature is not considered for VIFs, all VIFs for other features decreases significantly.
- Therefore, we decided to drop Temperature

4.2 VIF (Variance Inflation Factor):

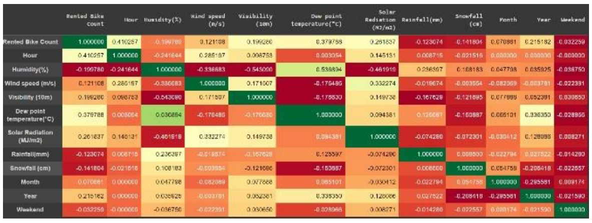
Variance Inflation Factor (VIF) is used to detect the presence of multicollinearity. Variance inflation factors (VIF) measure how much the variance of the estimated regression coefficients is inflated as compared to when the predictor variables are not linearly related. It is obtained by regressing each independent variable, say X on the remaining independent variables (say Y and Z) and checking how much of it (of X) is explained by these variables.

$$VIF = \frac{1}{(1-R^2)}$$

VIF shows similar results as a heatmap. Temperature and Dew Point Temperature show more correlation so the best way to eliminate prediction errors is to drop any temperature as it has more VIF than DPT. The VIF before and after dropping temperature is shown below in **fig 4.2.1** and **fig 4.2.2** respectively.

Fig. 4.2.1 VIF before dropping Temperature Fig. 4.2.2 VIF after dropping Temperature

After dropping the temperature data, we get the correlation heatmap as below.



	Rented Bike count	Hour	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Month	Year	Weekend
Rented Bike Count	1.000000	0.410257	-0.199780	0.121108	0.199280	0.379788	0.261837	-0.123074	-0.141804	0.070861	0.215162	-0.032259
Hour	0.410257	1.000000	-0.241644	0.285197	0.098753	0.003054	0.145131	0.008715	-0.021516	0.000000	0.000000	-0.000000
Humidity(%)	-0.199780	-0.241644	1.000000	-0.336683	-0.543090	0.536894	-0.461919	0.236397	0.108183	0.047798	0.035925	-0.036750
Wind speed (m/s)	0.121108	0.285197	-0.336683	1.000000	0.171507	-0.176486	0.332274	-0.019674	-0.003554	-0.082069	-0.003781	-0.022391
Visibility (10m)	0.199280	0.098753	-0.543090	0.171507	1.000000	-0.176630	0.149738	-0.167629	-0.121695	0.077888	0.052381	0.030650
Dew point temperature(°C)	0.379788	0.003054	0.536894	-0.176486	-0.176630	1.000000	0.094381	0.125597	-0.150887	0.065101	0.336350	-0.028966
Solar Radiation (MJ/m2)	0.261837	0.145131	-0.461919	0.332274	0.149738	0.094381	1.000000	-0.074290	-0.072301	-0.030412	0.128086	0.008271
Rainfall(mm)	-0.123074	0.008715	0.236397	-0.019674	-0.167629	0.125597	-0.074290	1.000000	0.008500	-0.022794	0.027522	-0.014280
Snowfall (cm)	-0.141804	-0.021516	0.108183	-0.003554	-0.121695	-0.150887	-0.072301	0.008500	1.000000	0.054758	-0.206418	-0.022557
Month	0.070861	0.000000	0.047798	-0.082069	0.077888	0.065101	-0.030412	-0.022794	0.054758	1.000000	-0.295561	0.009174
Year	0.215162	0.000000	0.035925	-0.003781	0.052381	0.336350	0.128086	0.027522	-0.206418	-0.295561	1.000000	-0.021590
Weekend	-0.032259	-0.000000	-0.036750	-0.022391	0.030650	-0.028966	0.008271	-0.014280	-0.022557	0.009174	-0.021590	1.000000

5.Feature Description:

- **Date**: Date feature which is **str** type is needed to convert it into Datetime format DD/MM/YYYY.
- **Rented Bike Count**: Number of bikes rented which is our Dependent variable according to our problem statement which is **int** type.
- **Hour**: Hour feature which is in 24hour format which tells us number bike rented per hour is **int** type.
- **Temperature(°C)**: Temperature feature which is in Celsius scale(°C) is **Float** type.
- **Humidity (%)**: Feature humidity in air (%) which is **int** type.
- **Wind speed (m/s)**: Wind Speed feature which is in (m/s) is **float** type.
- **Visibility (10m)**: Visibility feature which is in 10m, is **int** type.
- **Dew point temperature(°C)**: Dew point Temperature in (°C) which tells us temperature at the start of the day is **Float** type.
- **Solar Radiation (MJ/m2)**: Solar radiation or UV radiation is **Float** type.
- **Rainfall(mm)**: Rainfall feature in mm which indicates 1 mm of rainfall which is equal to 1 liter of water per meter square is **Float** type.
- **Snowfall (cm)**: Snowfall in cm is Float type. Seasons: Season, in this

feature four seasons are present in data is **str** type.

- **Holiday**: whether no holiday or holiday can be retrieved from this feature is **str** type. ● **Functioning Day**: Whether the day is Functioning Day or not can be retrieved from this feature is **str** type.

6. Feature Engineering

The provided data in its raw form wasn't directly used as an input to the model. Several feature engineering was carried out where few features were modified, few were dropped, and few were added. Below is a summary of the feature engineering carried out with the provided data set

- The *Date Time* column which contained the date-time stamp in 'YYYY-MM-DD HH:MM: SS' format was split into individual ['month', 'date', 'day', 'hour'] categorical columns
- Drop *season* column: This is because the season column falls under four categorical data, autumn, summer, spring, and winter and we have added each category individually after encoding.
- Drop *date* column: Intuitively, there should be no dependency on the date. Hence drop this column
- Drop *temperature* column: temp and Dew point temperature are very highly correlated and essentially indicate the same thing. Hence retain only the dew point temperature column
- *One Hot Encoding* of categorical feature:

  a. *Hours*: Split hour column to hour_0, hour_1, ..., hour_23. Drop the hour column since they are a function of the rest of the retained hour columns.

  b. *Month*: Split month column to month_1, month_2, ..., month_12. Drop month columns since they are a function of the rest of the retained month columns

  c. *Seasons*: Split the season's column into autumn, summer, spring, and winter. Drop the seasons column since it is the function of the rest of the season's columns

- *Ordinal Encoding:* The Holiday and Functioning day columns have been encoded using ordinal encoding to provide equal weightage to the deciding entries.

5.1 Normalization

The univariate analysis of rented bike data shows a positive skewness which would have been a problem while predicting the values on the test data set. So, to ensure the minimization of errors we have taken the square root of the rented bike count data which tends the data for equal weightage. The need for normalization is basically for making sure that a table contains only data directly related to the primary key, that each data field contains only one item of data, and that redundant (duplicated and unnecessary) data is eliminated.

The difference between the rented bike count data plot before and after normalization is shown below in fig 5.1.1 and fig 5.1.2 respectively:

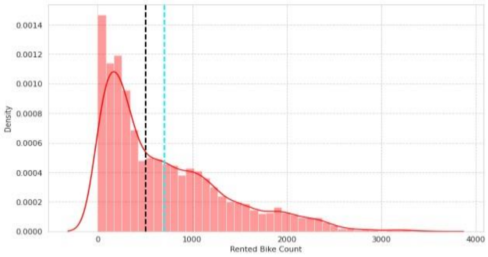


Fig 5.1.1

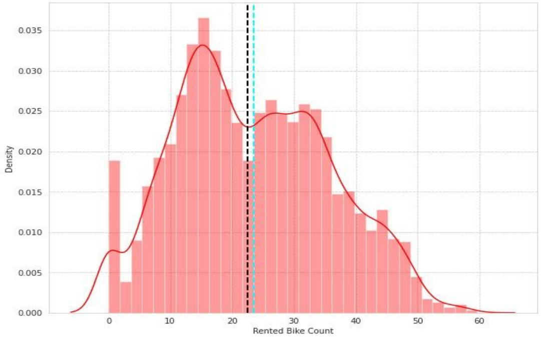
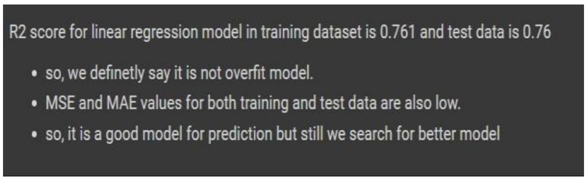


fig 5.1.2

7. Building Machine Learning Algorithm

The provided data is first cleaned and transformed using Feature Engineering. We then split the data into the Train set (for Hyperparameter tuning) and Test set (for Model Evaluation). Using MSE as our evaluation metric, we compare various models and select the regression algorithm-based o the lowest MSE on the Test data. The final mode used for submission is then obtained by again training the selected Regression Algorithm on entire Input Data set



R2 score for linear regression model in training dataset is 0.761 and test data is 0.76

- so, we definetly say it is not overfit model.
- MSE and MAE values for both training and test data are also low.
- so, it is a good model for prediction but still we search for better model

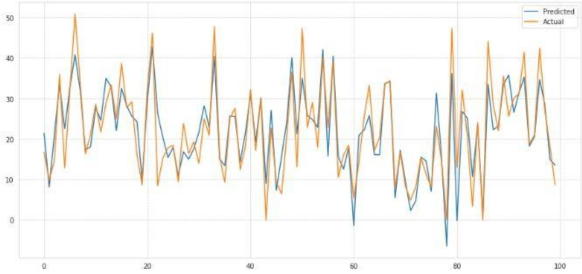
7.1 Train/Test Split

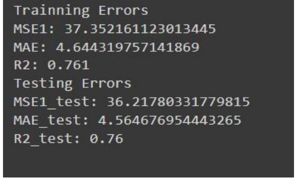
The train/test split was done as 80/20 % o data with a random state of 12. The final dataset was of shape (8760, 50) which was split to (7008 50) as Train data and (1752, 50) as Test data.
To normalize the data after the split, using the Min-Max Scalar module will give equal weightage to all the parameters to retain data from one-way deviation.

7.2 Linear Regression

After proper analysis of the data, many features were dropped or modified by the regression model requirement.

The predicted values show nearly optimal fit behavior concerning the actual data. The train and test errors are shown below the plot.

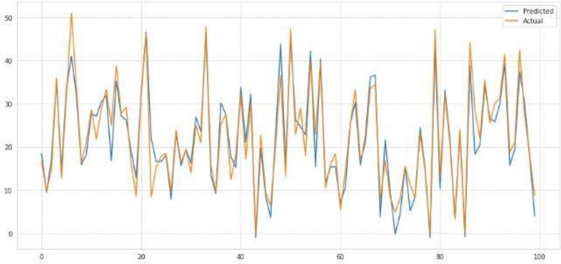




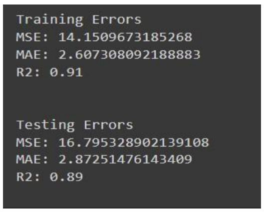
```
Trainning Errors
MSE1: 37.352161123013445
MAE: 4.644319757141869
R2: 0.761
Testing Errors
MSE1_test: 36.21780331779815
MAE_test: 4.564676954443265
R2_test: 0.76
```

7.3 Polynomial Regression

The same data set is then trained and tested using polynomial fit regression with degree taken as 2 and based on the values of the evaluation matrix the errors are calculated and the graph is plotted as shown



R^2 errors on the test data = 0.89 and training data = 0.91 are almost the same. So, we can conclude that the data on Polynomial regression model has not been overfitted. The Efficiency of this model shows a greater difference than the Linear regression.

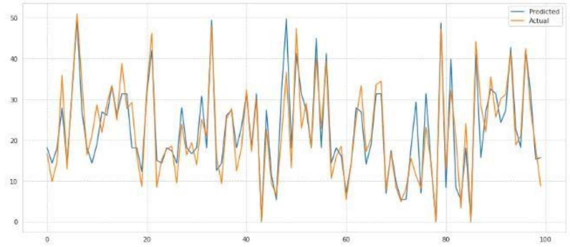


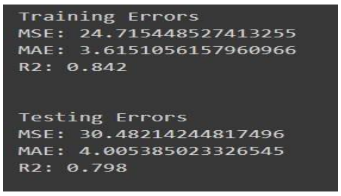
```
Training Errors
MSE: 14.1509673185268
MAE: 2.607308092188883
R2: 0.91

Testing Errors
MSE: 16.795328902139108
MAE: 2.87251476143409
R2: 0.89
```

7.4 Decision Tree Regressor

Decision Trees (DTs) are a nonparametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. Since decision trees are prone to overfitting, we have given parameters like maximum depth, maximum leaf nodes etc. to the model





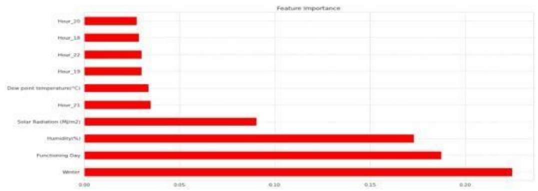
```
Training Errors
MSE: 24.715448527413255
MAE: 3.6151056157960966
R2: 0.842


Testing Errors
MSE: 30.48214244817496
MAE: 4.005385023326545
R2: 0.798
```

The Decision Tree Regression Model seems to approximate the Rented Bike Count better than the Linear Regression Model, but not as good as the Polynomial Regression Model. We can see this by comparing the parameters of the Root Mean Squared Error, the Mean Absolute Error, and the R-squared value. Also, we can visualize the better accuracy of this new model by looking at the above line plot. Of course, the Decision Tree Regression Model is not perfect and it has various disadvantages, we list some of them:

- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.
- We got an R2 score of 0.842 for training data and 0.798 for test data. Therefore, we can say that the model is optimally fit for the data.

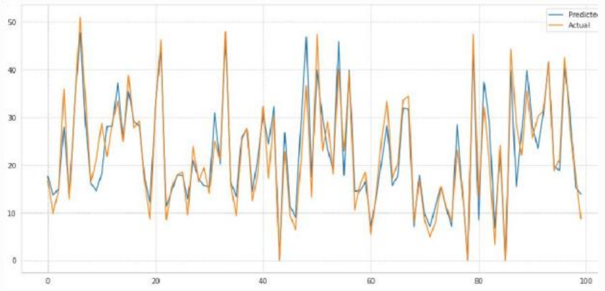
7.4.1 Feature importance in the decision tree



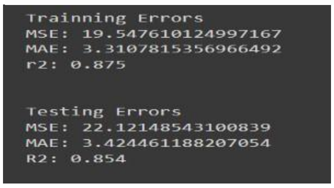
We can see from the graph, scores given to each feature for Decision Tree Regressor. Higher scores indicate higher importance given to the feature. For decision tree regressor, Winter, Functioning Day and humidity has gotten highest importance.

7.5 Random Forest

to find the optimal split at every node of every tree. Then the information from the n trees is aggregated for classification and prediction. Random forests also provide the importance of each feature by accumulated Gini gains of all splits in all trees representing the variable discrimination ability.



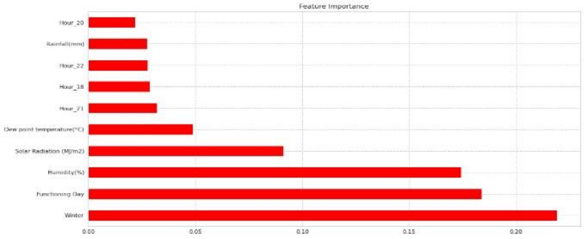
Random forest is an almighty tool that ensembles decision trees and bagging. The base learner of random forests is a binary tree constructed by recursive partitioning (RPART) and then developed using classification and regression trees. Binary splits of the parent node of a random forest split data into two children's nodes and increase homogeneity in children nodes compared to parent nodes. Note that a random forest does not split tree nodes based on all variables; instead, it chooses random variable subsets as candidates



```
Trainning Errors
MSE: 19.547610124997167
MAE: 3.3107815356966492
r2: 0.875


Testing Errors
MSE: 22.12148543100839
MAE: 3.424461188207054
R2: 0.854
```

For Random Forest we gave n_estimators, Maximum depth per tree and maximum leaf nodes as parameters to get a better fit model for that, we got R^2 of 0.875 for training data and 0.854 for test data, even the mean squared error is less as compared to linear regression and decision tree regressor.

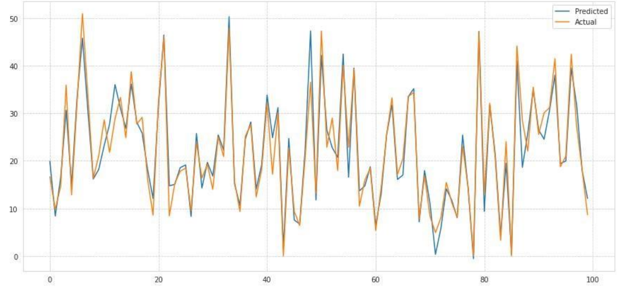


6.5.1 Feature importance in Random Forest.

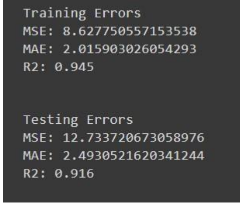
We can see from the graph, scores given to each feature for Random Forest. Higher scores indicate higher importance given to the feature. For decision tree regressor, Winter, Functioning Day and humidity has gotten highest importance

7.6 Gradient Boost Regressor with GridsearchCV

Gradient Boosting algorithm is used to generate an ensemble model by combining the weak learners or weak predictive models. Gradient boosting builds an additive mode by using multiple decision trees of fixed size as weak learners or weak predictive models. The parameter, n_estimators, decides the number of decision trees which will be used in the boosting stages. For parameters, we have used grid search cross validation, which takes a list of parameters and returns the best parameters for a certain dataset on a certain model.



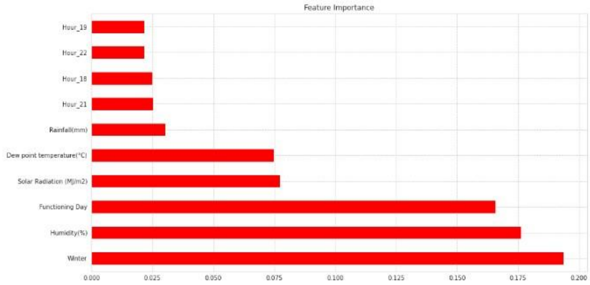
After getting the best parameters from grid search cross validation, we gave those parameters to the algorithm and got R^2 of 0.945 on training data an 0.916 for test data which is highest in our model tests. Mean squared errors and mean absolute error are also least for Gradient boost with be parameters



```
Training Errors
MSE: 8.627750557153538
MAE: 2.015903026054293
R2: 0.945


Testing Errors
MSE: 12.733720673058976
MAE: 2.4930521620341244
R2: 0.916
```

7.6.1 Feature importance in the Gradient Boost Regressor with GridsearchCV



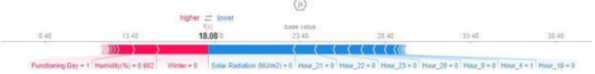
We can see from the graph, scores given to each feature for Random Forest. Higher scores indicate higher importance given to the feature. For decision tree regressor, Winter, Functioning Day and humidity has gotten highest importance.

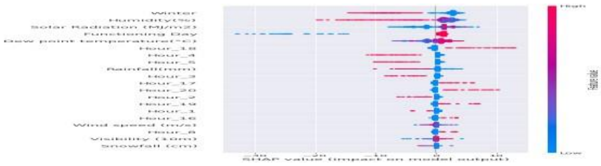
8. Model Explainability:

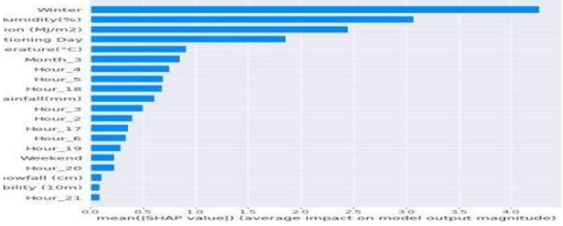
SHAP Interpretation

- Base value: This is the average feature value. This value is used to determine if the prediction is true or false.
- Red color Block: This represents the feature for which the prediction is positive. Higher this value will push the prediction positively.
- Blue color block: This represents the feature for which the prediction is negative. higher this value will push the prediction negatively
- Block size: the block size shows the feature importance. larger the block size larger will the feature importance value**.**

8.1 SHAP for Decision Tree Regressor.

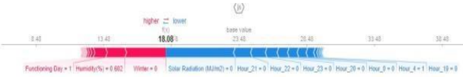


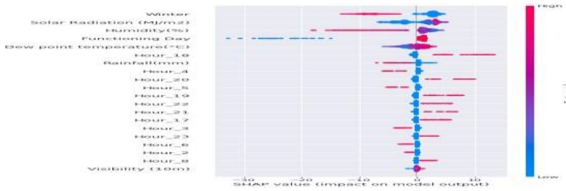


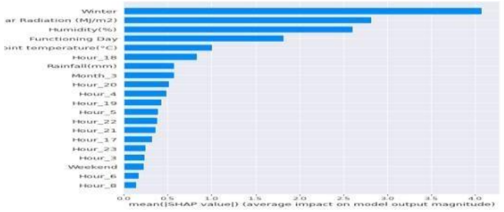


- Here we can see negative feature or blue color block pushes the prediction toward left over base value and causing prediction negative.
- We can see from SHAP summary that high Hour_18 value increases predicted bike demand ● Low snowfall value also increasing predicted bike sharing demand ● Humidity is highly negatively correlated with bike share demand
- In bar graph we can see winter has the highest feature value while snowfall has the lowest feature value

8.2 SHAP for Random Forest Regressor



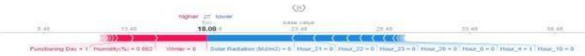


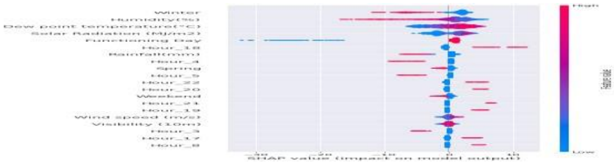


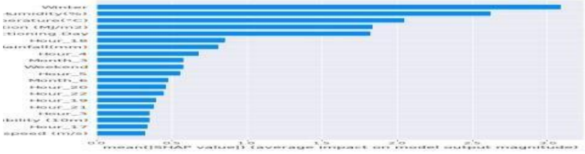
- Here we can see negative feature or blue color block pushes the prediction toward left over base value and causing prediction negative.

- We can see from SHAP summary that high Hour_18 value increases predicted bike demand
- Low snowfall value also increasing predicted bike sharing demand
- Humidity is highly negatively correlated with bike share demand
- In bar graph we can see winter has the highest feature value while snowfall has the lowest feature value

7.3. SHAP for Gradient Boost with Gridsearch







- we can see negative feature or blue color block pushes the prediction toward left over base value and causing prediction negative.
- Also, we can see from SHAP summary that Hour_18 value increasing predicted bike demand.
- Here also humidity and winter season is highly negatively correlated with predicted bike sharing demand

- In bar graph we can see Winter has the highest feature value while Hour_8 has the Lowest feature_value

9. Conclusion:

1) In June maximum bike rented near 1000. and January, February enjoys less rented bike demand near 400. March and June enjoy more bike sharing demand in holiday than non-holidays February, December have less bike sharing demand for both in holidays and non-holidays. April, October, December months have nearly zero bike sharing demand in holidays.

2) weekdays have more rented bike demand than weekend

3)In weekdays 6 pm and 8 am but in weekend only 6 pm are peak time of bike sharing demand. 4 and 5 am has lowest bike sharing demand in both weekend and weekdays

4) Summer season enjoys overall best and least bike sharing demand and winter has overall less demand

5) From the regression plots we can conclude that the columns

'Rainfall', 'Snowfall', 'Humidity' these features are negatively related with the dependent variable 'Rented Bike Count'. This means when Rainfall, snowfall, humidity is higher bike sharing demand is lower. 'Temperature', 'Wind_speed','Visibility', 'Dew_point_temperature', 'Solar_Radiation' are positively correlated with the dependent variable 'Rented Bike Demand'. This means if 'Temperature', 'Wind_speed','Visibility', 'Dew_point_temperature', 'Solar_Radiation' are higher or lower then bike sharing demand maybe higher or lower respectively.

6) After applying linear regression model, we got r2 score of 0.761 for training dataset and 0.76 for test dataset which defines that model is optimally fit for training and test data i.e. no overfitting

7) Therefore, for even better fit, we applied polynomial regression model with degree = 2, we got R2 score of 0.91 for training data and 0.89 for test data

8) We also tried Tree based classifiers for our data, we applied Decision Tree Regressor, since decision tree is prone to overfit, we gave certain parameters like maximum depth of the tree, maximum leaf nodes etc., with that we got R2 score of 0.842 for training data and 0.798 for test data which is less than polynomial regression.

10) To get better accuracy on tree-based model, we applied Random Forest with n_estimator as 180 and with maximum depth as 12, with that we got R2 score of 0.875 for training data and 0.854 for test data.

11) Finally, we applied Gradient boost with parameters selected after grid search which resulted in highest R2 score of 0.945 for training data and 0.916 for test data with very less mean squared error of 8.6 and 12.4 in training as well as in test data.

Therefore, we can say that it gives us optimal result in term of test dataset. It is best for final prediction

12) Lastly, in bar graph we can see Winter has the highest feature value. We can conclude that Hour_8, Visibility and Wind Speed is not contributing in Decision Tree, Random Forest and Gradient Boost in model prediction

than any other season. There are very less bike sharing
demand in morning 4 and 5 am

10. References:

1.GeekforGeeks

2.Kaggle

3.Analytics Vidya