Seoul Bike Sharing Demand Prediction

Saugata Deb, Shreyash Movale, Ankit Patil, Naga sai Kiran

Data science trainees,

Almabetter

AlmaBetter, Bangalore

Abstract:

Bike sharing as we know is a transport service primary focus to lend conventional or electrical bikes to an individual or a group of individuals in order to let them travel in city or outskirt in rent for an hour, a day or for a month depending on the needs.

In market share we can see that Bike Sharing system has a global market share which was valued around $3.39 billion in 2019 and is projected to grow to $6.98 billion 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.

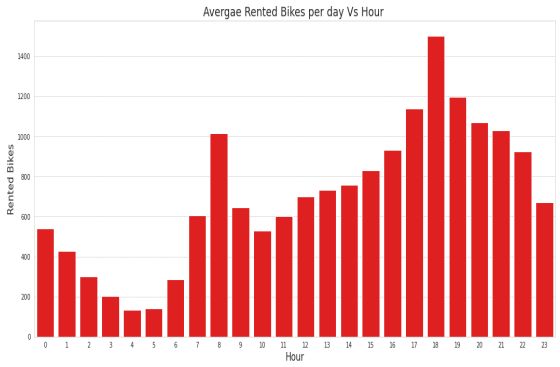
According to our problem our main aim is to build a predictive model so as to find the number of bikes rented based on the given dataset.

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.

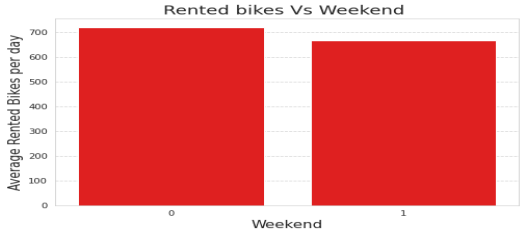
3.1 Hour based Analysis

The hour-based Analysis showed that the bike demand is at its peak at 08:00 AM and in the evening between 05:00 PM to 09:00 PM. We can conclude that most of the bike users belong to the working category as the time indicating bike count at the peak is mostly the working hours start and end time.



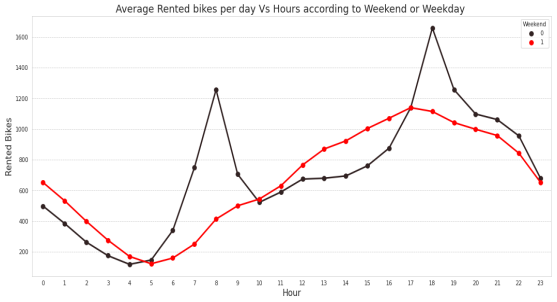
3.2WeekdayandWeekoffbasedAnalysis

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



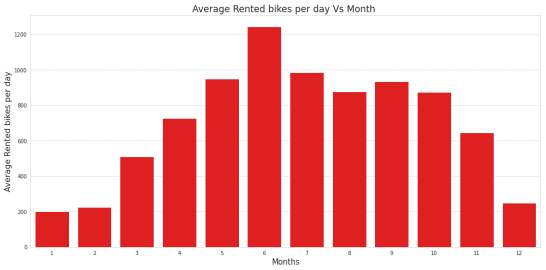
3.3 Hours v/s Weekday and Week off based Analysis

The below plot shows that for weekends the rented bike counts remain in saddle condition while for weekdays it shows a peak at 8:00 AM and 6:00 PM which may be the result of working-class traffic while the trend in weekend pattern corresponds to probably tourists who typically are casual users who rent/drop off bikes uniformly during the day and tour the city.



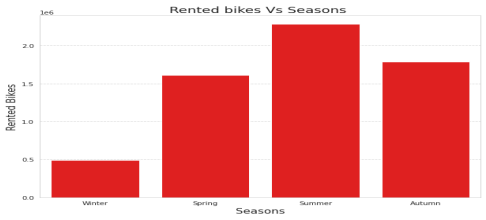
3.4 Month based Analysis

The below bar chart contains the average bike count over each month of a calendar year. We can see here that the graph shows more entries in months number 5 to 10 i.e., May to October which mostly correlates to season data. We can see that the most rentals are in June and May while the least are in January and February.



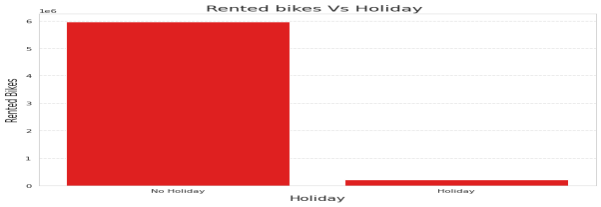
3.5 Season-wise Analysis

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.



3.5 Function day and Holiday

The below figure shows the dependency of rented bike count on functioning days and holidays respectively. Although its values are unidirectional it may not be a key part to predict the bike-sharing demand

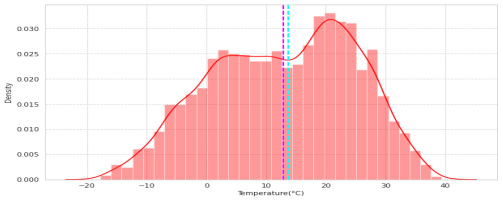


3.6 Analysing Numerical Variables

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.

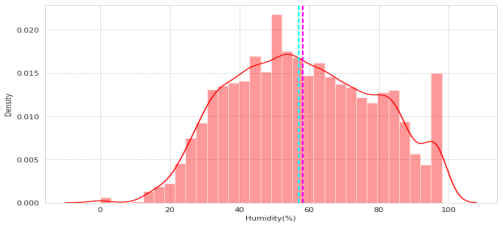
3.6.1. Temperature

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.



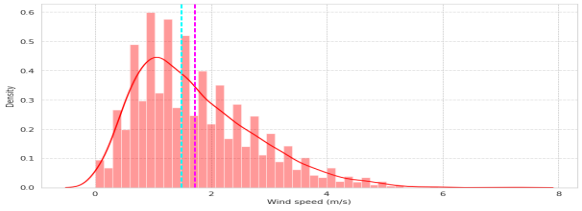
3.6.2 Humidity (%)

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.



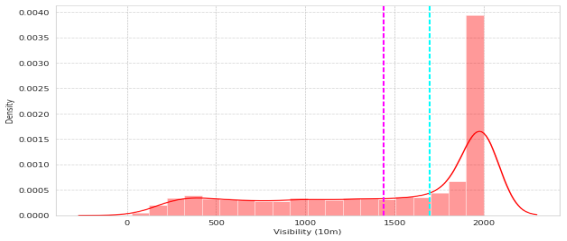
3.6.3 Wind Speed (m/s):

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.



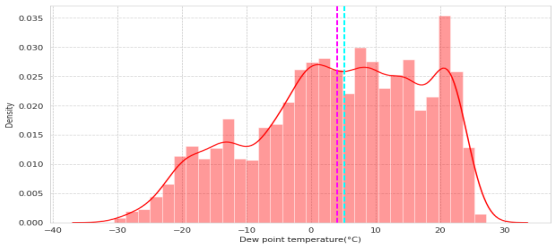
3.6.4 Visibility

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.



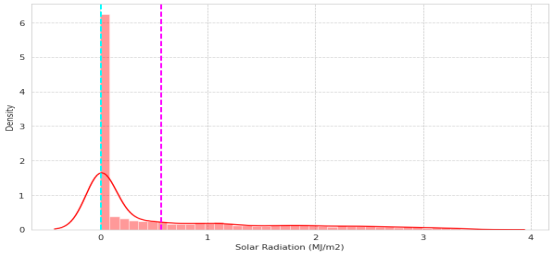
3.6.5 Dew Point Temperature (°C)

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.



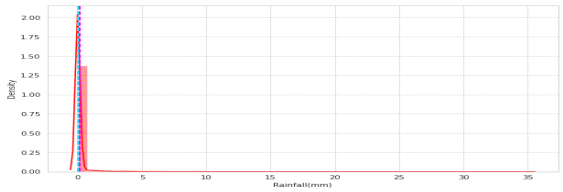
3.6.6 Solar Radiation

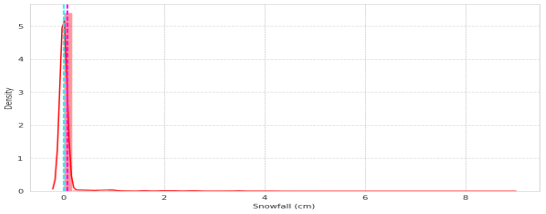
In density plot for **Solar Radiation** we can see that mean is greater than median we can say that this is positively skewed.



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.



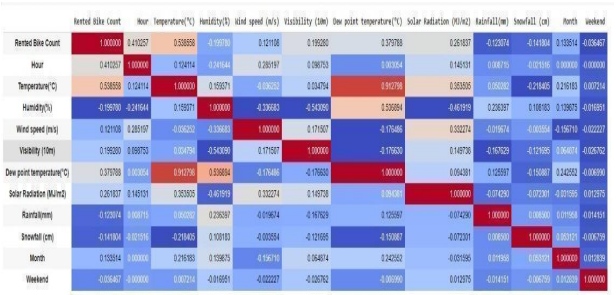


4. Correlation Analysis

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

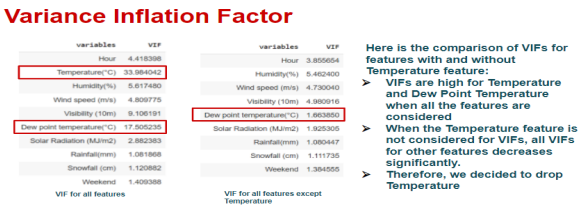
Let's check the heatmap plotted concerning independent variables.



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



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.

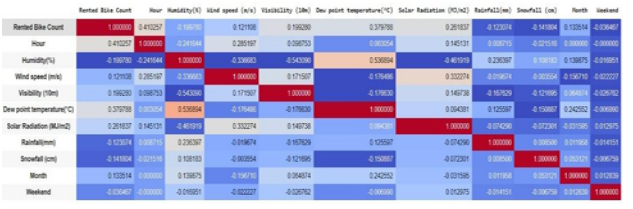
$$VIF = \frac{1}{(1-R^2)}$$

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.



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 24-hour 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 litre of water per metre 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 Normalisation

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:

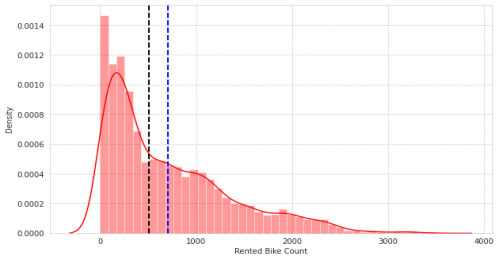


Fig 5.1.1

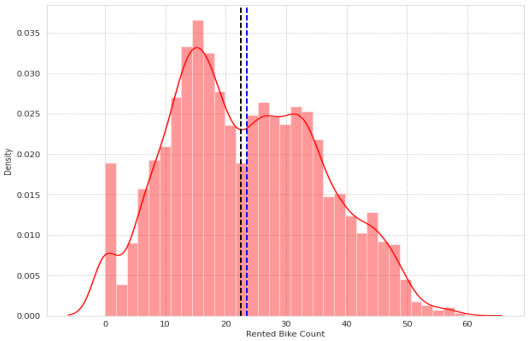


fig 5.1.2

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

R^2 errors on the test data = 0.779 and training data = 0.774 are almost the same. So, we can conclude that Linear Regression model is definitely not an overfit model. Still we will go ahead with other models in a search of more precise one

models and select the regression algorithm based c the lowest MSE on the Test data. The final mod used for submission is then obtained by aga training the selected Regression Algorithm on t entire Input Data set

7.1 Train/Test Split

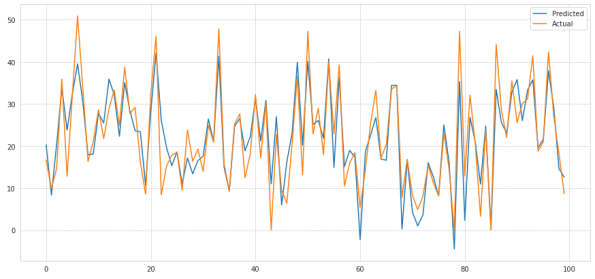
The train/test split was done as 80/20 % c data with a random state of 12. The final datas was of shape (8760, 50) which was split to (700 50) as Train data and (1752, 50) as Test data.

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.

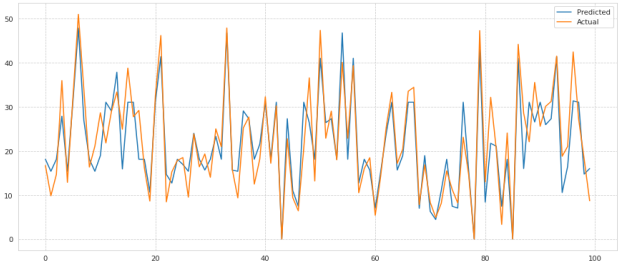
The predicted values show nearly optimal fit behaviour concerning the actual data. The train and test errors are shown below the plot.




```
Training Errors
MSE: 34.443723451189115
MAE: 4.436644249627593
R2: 0.779

Testing Errors
MSE: 34.12057506681097
MAE: 4.365698635890322
R2: 0.774
```

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

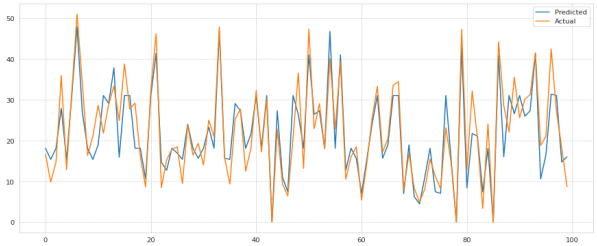


R^2 errors on the test data = 0.9 and training data = 0.93 are almost the same. So, we can conclude that the data ofa on Polynomial regression model has not been overfitted. The Efficiency of this model shows a greater difference than the Linear regression.

```
Training Errors
MSE: 11.516976335573187
MAE: 2.25335580563619
R2: 0.93

Testing Errors
MSE: 14.65901362869785
MAE: 2.5455574028490577
R2: 0.9
```

7.4 Decision Tree Regressor

Decision Trees (DTs) are a non-parametric 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



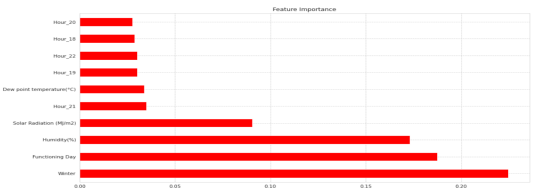

```
Training Errors
MSE: 25.793982334841058
MAE: 3.7210179188275037
R2: 0.835

Testing Errors
MSE: 29.901751933312354
MAE: 3.9701362007157677
R2: 0.802
```

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 0.835 for training data and 0.802 for test data. Therefore, we can say that the model is optimally fit for the data.

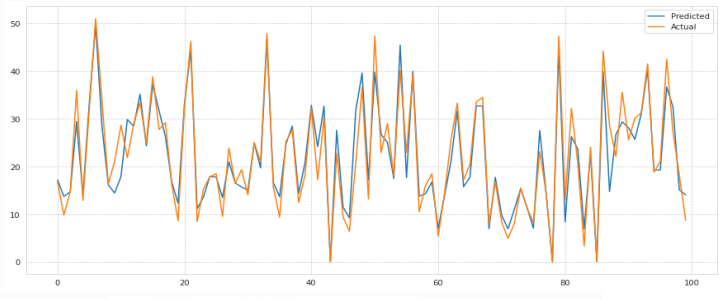
7.4.1 Feature importance in the decision tree



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.

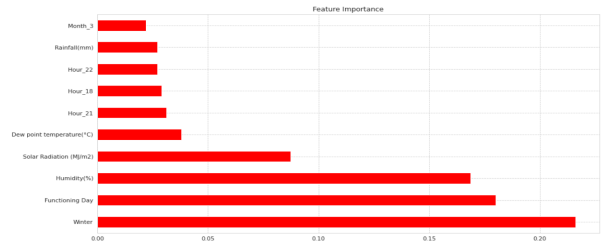
7.5 Random Forest

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 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.




```
Training Errors
MSE: 17.410712899141867
MAE: 3.107133318857302
R2: 0.888

Testing Errors
MSE: 18.8454619617235
MAE: 3.1542327512216732
R2: 0.875
```

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 0.888 for training data and 0.875 for test data, even the mean squared error is less as compared to linear regression and decision tree regressor 6.5.1 Feature importance

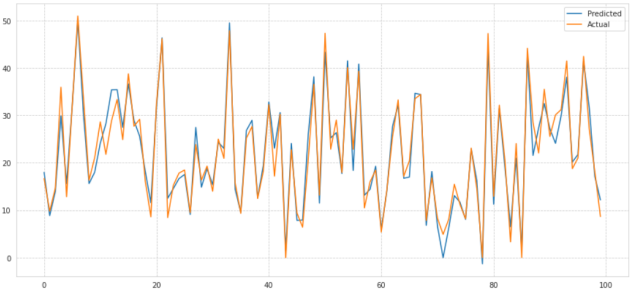


in Random Forest.

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

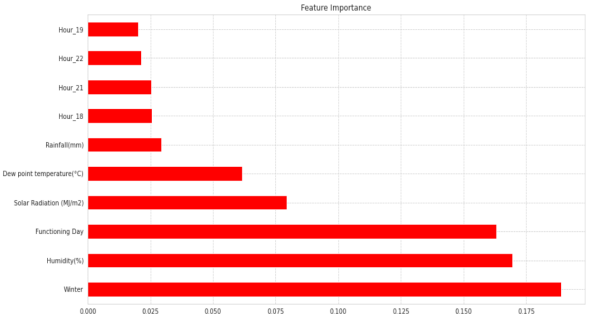
Gradient Boosting algorithm is used to generate an ensemble model by combining the weak learners or weak predictive models. Gradient boosting builds an additive mode by using multiple decision trees of fixed size as weak learners or weak predictive models. The parameter, n_estimators, decides the number 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 searc cross validation, we gave those parameters to th algorithm and got R^2 of 0.958 on training data an 0.933 for test data which is highest in our model tests. Mean squared errors and mean absolute erro are also least for Gradient boost with be parameters

```
Training Errors
MSE: 6.502066630324401
MAE: 1.712901821228588
R2: 0.958


Testing Errors
MSE: 10.078320275215765
MAE: 2.167583140792035
R2: 0.933
```

7.6.1 Feature importance in the Gradient Boost Regressor with GridsearchCV



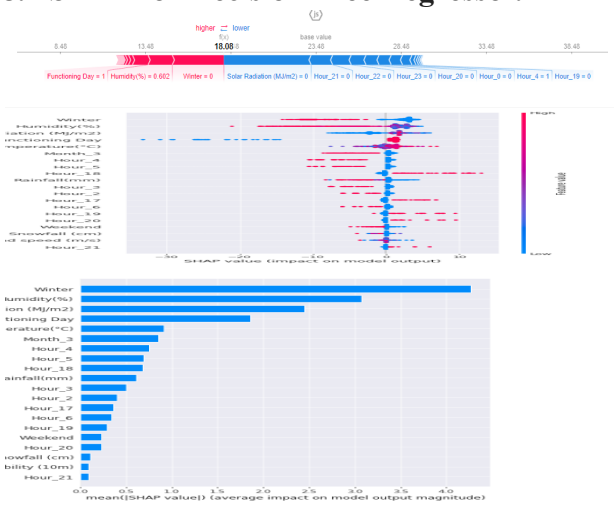
We can see from the graph, scores given to each feature for Random Forest. Higher scores indicate higher importance given to the feature. For decision tree regressor, Winter, Functioning Day and humidity has gotten highest importance.

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.

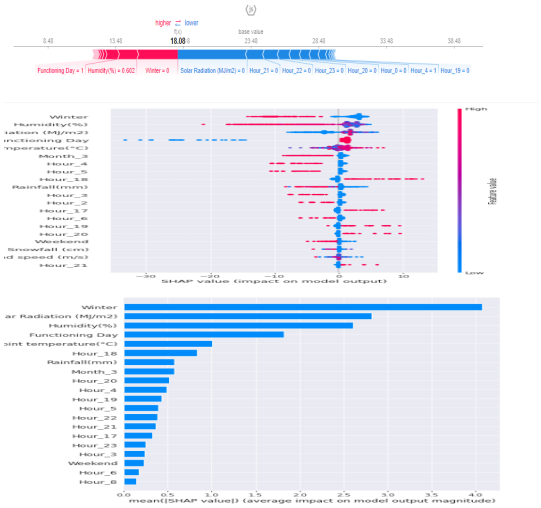
8.1 SHAP for Decision Tree Regressor.



- Here we can see a negative feature or blue color block pushes the prediction toward left over base value and causes prediction negative.
- Also we can see from SHAP summary that high Hour_18 value increasing predicted bike demand. Also, we can see low Snowfall value increasing predicted bike demand.

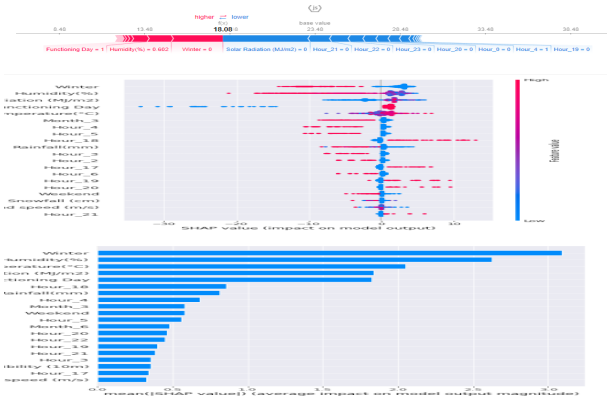
- In the bar graph we can see Winter has the highest feature value while Hour_21 has the Lowest feature value.

8.2 SHAP for Random Forest Regressor



- Here we can see a negative feature or blue color block pushes the prediction toward left over base value and causes prediction negative.
- Also, we can see from SHAP summary that Hour_18 value is increasing predicted bike demand. Also, we can see low Weekend value increasing predicted bike demand.
- In the bar graph we can see Winter has the highest feature value while Hour_8 has the Lowest feature value.

7.3. SHAP for Gradient Boost with Gridsearch



- Here we can see a negative feature or blue color block pushes the prediction toward left over base value and causes prediction negative.
- Also we can see from SHAP summary that Hour_18 value increased predicted bike demand. Also we can see low Weekend value increasing predicted bike demand.
- In bar graph we can see Winter has the highest feature value while Wind Speed has the Lowest feature value

9. Conclusion:

- So we have come to the end of our project Bike Sharing Demand Prediction. What we did let's take a short recap. We have found our dataset info where we have found there are 8760 rows and 14 columns with 13 features as independent and one as dependent according to our problem statement that is Rented bike count on which we have done our prediction. We didn't find any null values or duplicates.

- In feature engineering we have done one hot encoding, ordinal encoding on independent features. Then in correlation analysis we have dropped the temperature feature as it was showing very high correlation with Dew Point Temperature which would be very difficult for model prediction so as to interpret as well.

- In model selection we have selected GradientBoost Model with GridSearch to get the best estimator for our model prediction as it has given very less mean squared error and high r2 score of above 90% for both train and test data with a split of 50% each.

- In model explainability we found that Winter Feature is giving high shap value while hour_21,hour_8,WindSpeed is not contributing in Tree Based Models.

10. References:

1.GeekforGeeks
2.Kaggle
3.Analytics Vidya