Seoul Bike Sharing Demand Prediction

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

Data science trainees,

Almabetter

Abstract:

Currently, the rental bike system plays a crucial part in public transport to increase the mobility of traffic in any city. The more important part of any bike-sharing system is to predict its bike availability in its key locations and prediction of its demand. This study helps us to predict the approximate bike demands throughout the city to unleash the pressure of citizens who are reliable on bikes mostly for transportation purposes. A data mining technique is employed for overcoming the hurdles for the prediction of hourly rental bike demand based on the data which includes whether the information (Temperature, Humidity, Visibility, Windspeed, Dew point TemperAture, Solar Radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information, weekdays or weekend information, Holiday or No Holiday, etc. We also tried to eliminate the features which do not generally contribute to bike-sharing demand prediction. On this data and after Exploratory Data Analysis, we tried to build multiple preferable machine learning algorithms which contributed toward demand prediction and came up with the most accurate one.

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 he 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:

Currently, the bike-sharing scheme is well-received throughout the world. It is a shared bike service for individuals, which is free of charge and for a short-term basis at a minimal rate. Most bike-sharing systems permit people to borrow and return a bike from a bike station to another station that belongs to the same network. Bike-sharing gains a vast range of attention in recent years as part of initiatives to boost the use of cycles, improve the first mile/last mile link to other modes of transportation, and minimize the negative effect of transport activities on the environment. Bike-sharing has significant impacts on establishing a larger cycling community, increasing the use of transportation, minimizing greenhouse gas emissions, enhancing public health, and also traffic troubles.

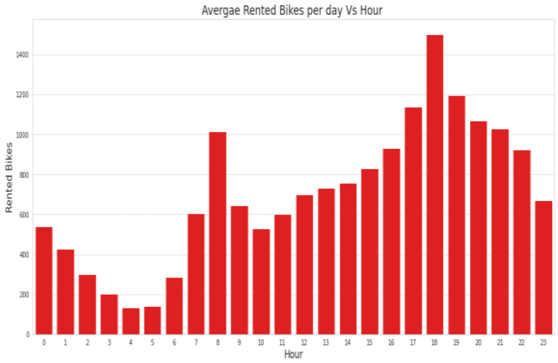
So, the constant raise of users necessitates the prediction of the number of rental bikes that were needed to make the bike-sharing system consistently work. Therefore, we aim to use machine learning to predict the required number of rental bikes required at each hour.

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.

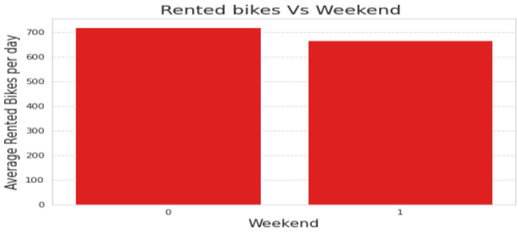
3.1 Hour based Analysis

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.



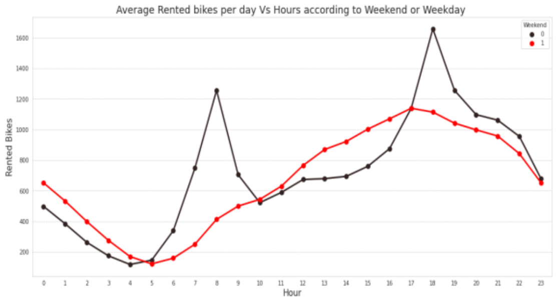
3.2WeekdayandWeekoffbasedAnalysis

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

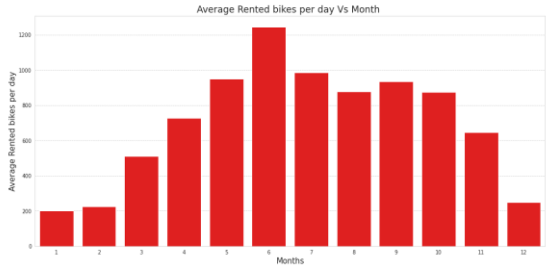
The below plot shows that for weekends the rented bike counts remain in saddle condition while for weekdays it shows a peak at 8:00 AM and 6:00 PM which may be the result of working-class traffic while the trend in weekend pattern corresponds to probably tourists who typically are casual users who rent/drop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



3.4 Month based Analysis

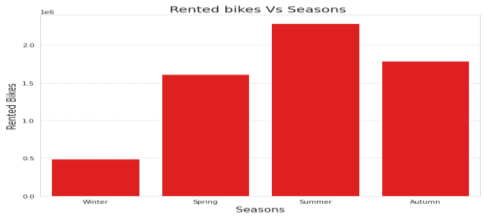
The below bar chart contains the average bike count over each month of a calendar year.

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.



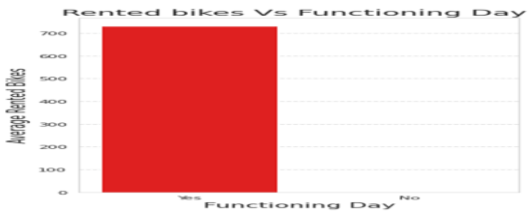
3.5 Season-wise Analysis

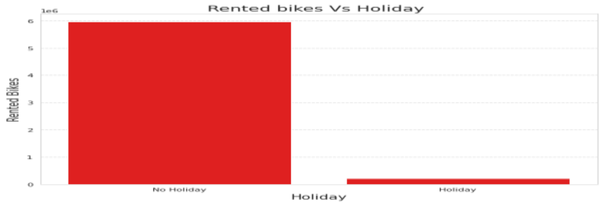
During the season-wise analysis, it was found that the month plays a significant role in rented bike demands. The demands are most likely to be high during summer followed by autumn and spring while winter shows the least demand.



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



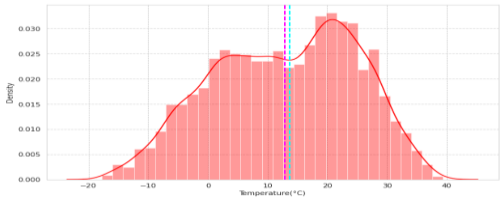


3.6 Analyzing Numerical Variables

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.

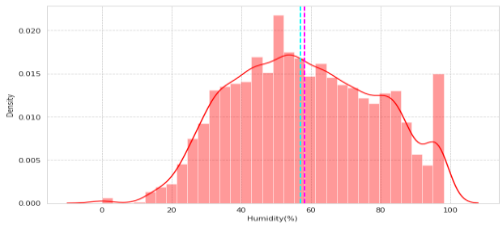
3.6.1.Temperature

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.



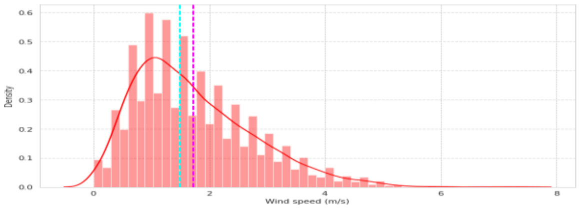
3.6.2 Humidity(%)

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.



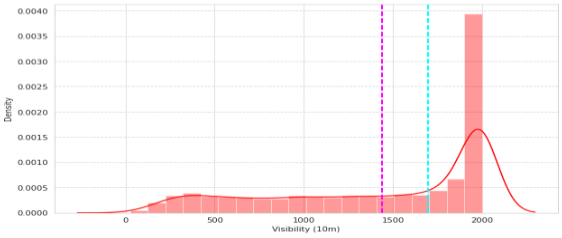
3.6.3 Wind Speed (m/s):

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.



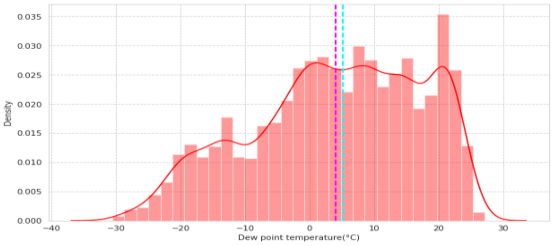
3.6.4 Visibility

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.



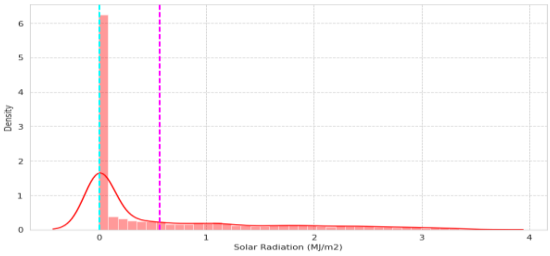
3.6.5 Dew Point Temperature (°C)

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



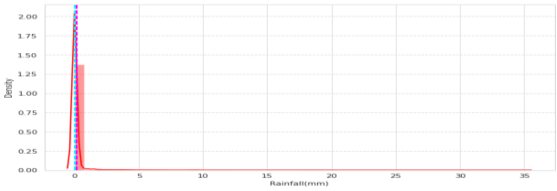
3.6.6 Solar Radiation

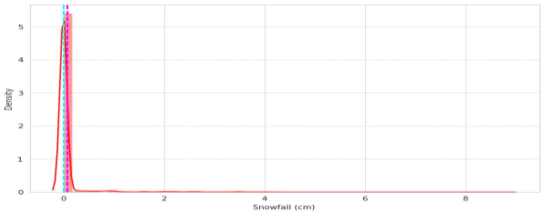
In density plot for **Solar Radiation** we can see that mean is greater than median we can say that this is positively skewed.



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.



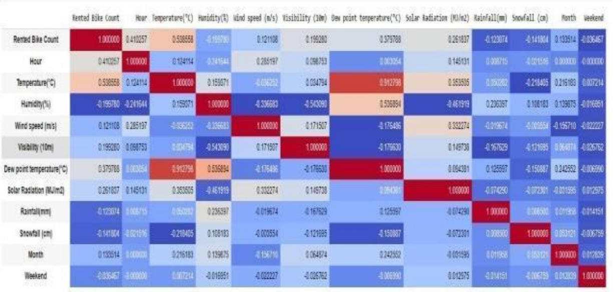


4. Correlation Analysis

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

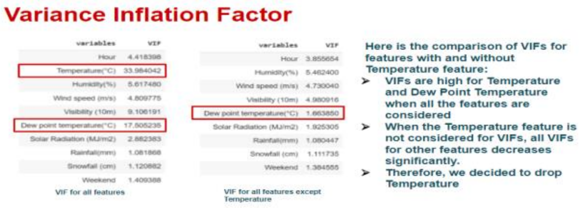
Let's check the heatmap plotted concerning independent variables.



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


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.

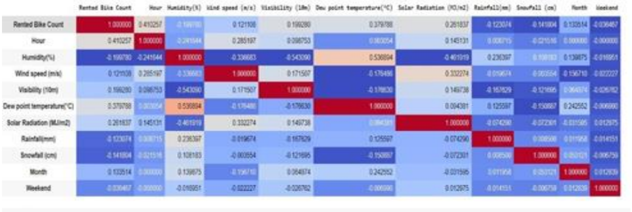
$$VIF = \frac{1}{(1-R^2)}$$

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.



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 bike 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:

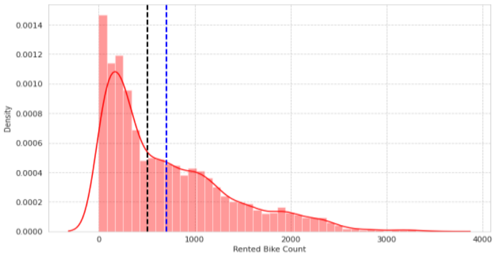


Fig 5.1.1

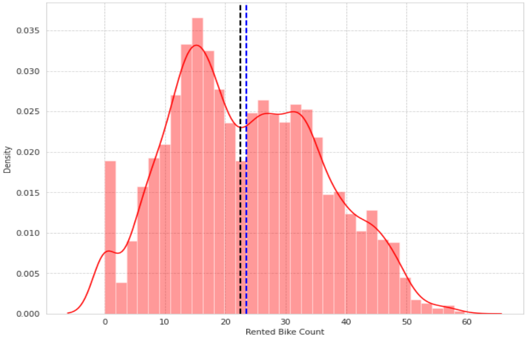


fig 5.1.2

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

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

7.1 Train/Test Split

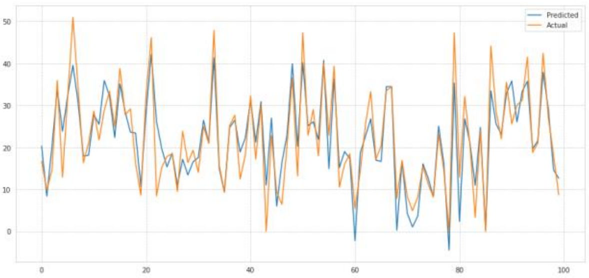
The train/test split was done as 80/20 % of data with a random state of 12. The final dataset was of shape (8760, 50) which was split to (7008, 50) as Train data and (1752, 50) as Test data.

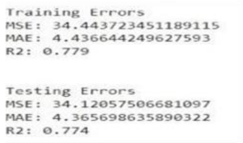
To normalize the data after the split, using the Min-Max Scalar module will give equal weightage to all the parameters to retain data from one-way deviation.

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 behaviour concerning the actual data. The train and test errors are shown below the plot.

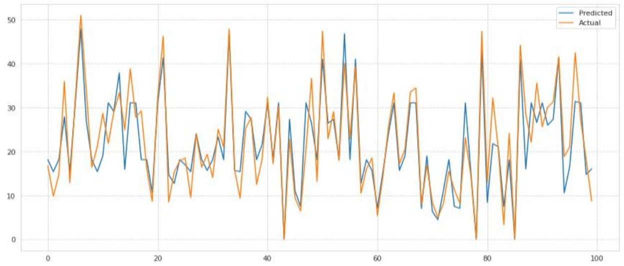




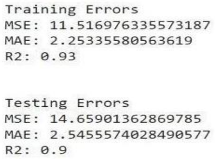
```
Training Errors
MSE: 34.443723451189115
MAE: 4.436644249627593
R2: 0.779

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



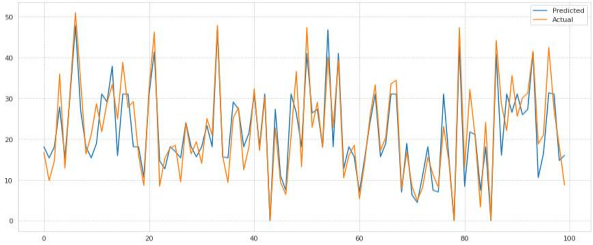
R^2 errors on the test data = 0.9 and training data = 0.93 are almost the same. So, we can conclude that the data ofa on Polynomial regression model has not been overfitted. The Efficiency of this model shows a greater difference than the Linear regression.

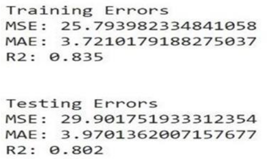


```
Training Errors
MSE: 11.516976335573187
MAE: 2.25335580563619
R2: 0.93

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



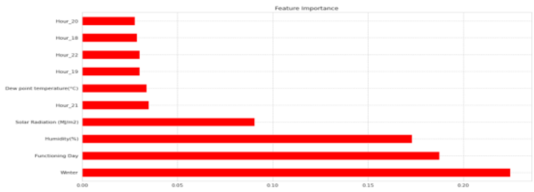


```
Training Errors
MSE: 25.793982334841058
MAE: 3.7210179188275037
R2: 0.835

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.

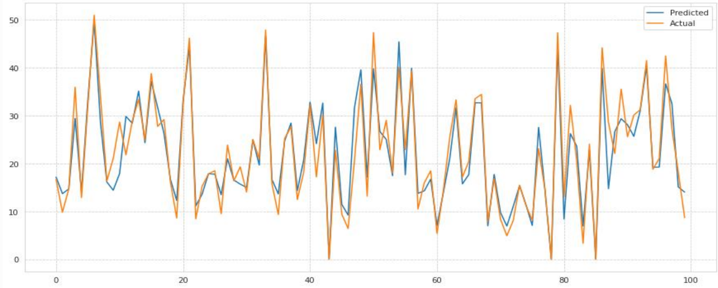
7.4.1 Feature importance in the decision tree

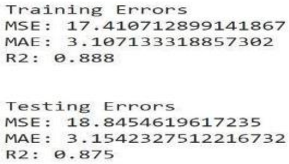


We can see from the graph, scores given to each feature for Decision Tree Regressor. Higher scores indicate higher importance given to the feature. For decision tree regressor, Winter, Functioning Day and humidity has gotten highest importance.

7.5 Random Forest

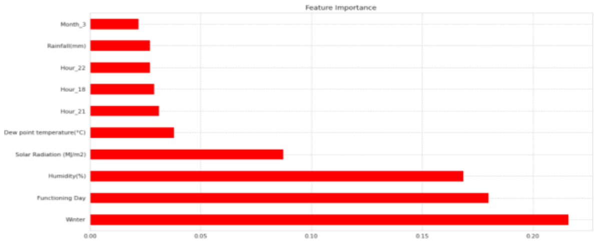
Random forest is an almighty tool that ensembles decision trees and bagging. The base learner of random forests is a binary tree constructed by recursive partitioning (RPART) and then developed using classification and regression trees. Binary splits of the parent node of a random forest split data into 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

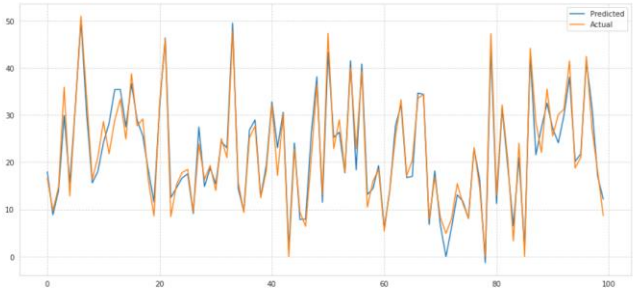


in Random Forest.

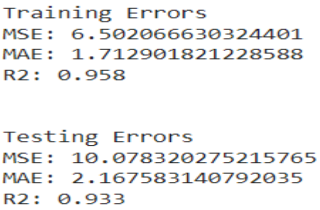
We can see from the graph, scores given to each feature for Random Forest. Higher scores indicate higher importance given to the feature. For decision tree regressor,Winter, Functioning Day and humidity has gotten highest importance

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.

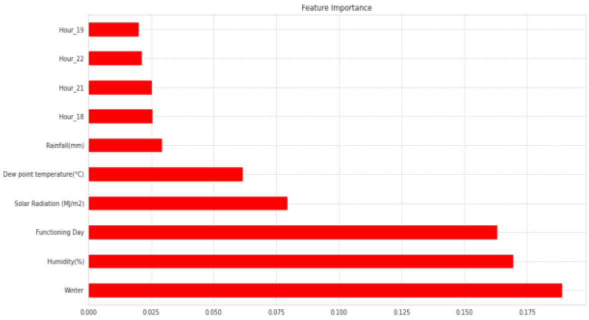


After getting the best parameters from grid search cross validation, we gave those parameters to the algorithm and got R^2 of 0.958 on training data and 0.933 for test data which is highest in our model's tests. Mean squared errors and mean absolute errors are also least for Gradient boost with best 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



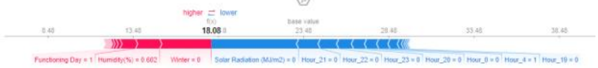
We can see from the graph, scores given to each feature for Random Forest. Higher scores indicate higher importance given to the feature. For decision tree regressor, Winter, Functioning Day and humidity has gotten highest importance.

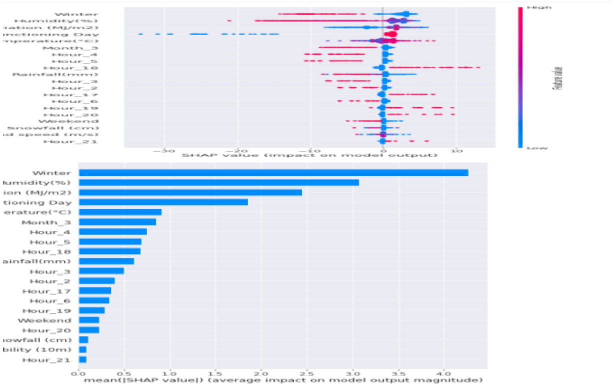
8. Model Explainability:

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 pushes the prediction negatively
- Block size: the block size shows the feature importance. larger the block size larger will the feature importance value**.**

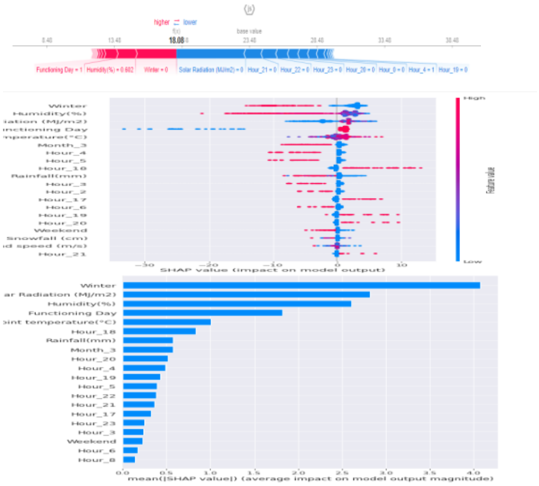
8.1 SHAP for Decision Tree Regressor.





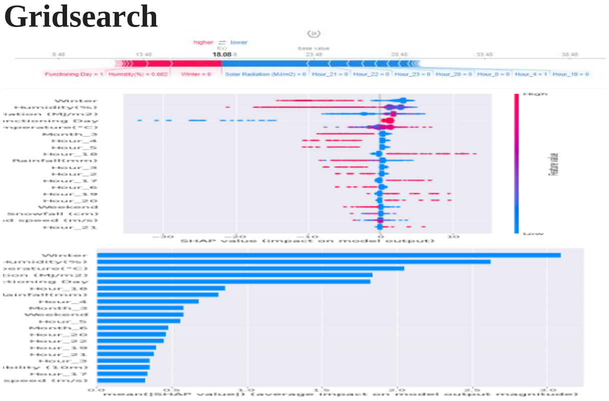
- Here we can see a negative feature or blue color block pushes the prediction toward left over base value and causes prediction negative.
- Also we can see from SHAP summary that high Hour_18 value increasing predicted bike demand. Also we can see low Snowfall value increasing predicted bike demand.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

1. In the summer season the highest number of bikes was rented as compared to other seasons with the count touching at 3500 while in the winter season the lowest number of bikes was rented touching the count of close to just 1000. From this, we can assume that people tend to rent more bikes in summer as compared to other seasons. Also people tend to rent fewer bikes in the winter season.

2. During the working day people tend to rent more bikes as around 3500 from this we can assume that on holidays people tend to rent fewer bikes.

3. Also, we can see people tend to rent fewer or no bikes during non functioning days.

4. In Hour vs Rented Bike Count we can see that during 18:00 Hrs (i.e. 6:00 PM)

the highest number of bikes was rented as compared to 5:00 Hrs (i.e. 5:00 AM). This means people tend to rent fewer bikes in the early morning.

5. In Rainfall vs Rented Bike Count and similarly with Snowfall vs Rented Bike Count we can see that people tend to rent the highest number of bikes during 0.00mm of Rainfall or no rainfall and 0.00cm of snowfall or no snowfall as compared to when there is rainfall or snowfall. In other words, people rent fewer bikes or no bikes with the increase in rainfall or snowfall.

6. In month vs Rented Bike Count we can see that people tend to rent more bikes in 6 or June month as compared to fewer bikes during Dec or January. From this, we can assume that people tend to rent more bikes in summer as compared to winter.
7. In the weekend vs Rented Bike count we can see that people tend to rent more bikes during weekdays as compared to weekends.

8. In Average Bike Rented vs Hour we can see that at 6:00 PM the average number of bikes rented by the people was 1550. While at 00.00 or midnight average bike rental was lowest with just around 550 bikes.

9. In Average Bike Rented vs Month we can see that Average Bike rented in July was highest at around 1250 and Average Bike Rented during February was the lowest with just 200 average bikes.

10. After applying the linear regression model, we got R2 score of 0.779 for training data and R2 score of 0.774 for test data, which signifies that model is optimally fit on both training and test data i.e. no overfitting is seen.

11. Therefore, for even better fir, we applied polynomial regression model with degree = 2, we got R2 score of 0.933 for training data and 0.90 for test data

12. 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 we got R2 score of 0.835 for training data and 0.803 for test data which is less than polynomial regression.

13. To get better accuracy on a tree-based model, we applied Random forest with n_estimator as 180 and with maximum depth as 13, with that we got an R2 score of 0.888 for training data and 0.875 for test data.

14. Finally, we applied Gradient boost with parameters selected after grid search which resulted in highest R2 score of 0.958 for training data and 0.933 for test data with very less mean squared error of 6 and 10 in training as well as in test data.
15. Also we can see from SHAP summary that high **Hour_18** value increasing prediction. Also we can see low **Snowfall** value increasing prediction and it is a common phenomenon in all the models.
16. Lastly, In bar graph from SHAP we can see **Winter** has the highest feature value while **Wind Speed** has the Lowest shap value.We can conclude that Hour_21,Hour_8 and Wind Speed is not contributing in Decision Tree,Random Forest and Gradient Boost in model prediction.

10. References:

1.GeekforGeeks
2.Kaggle
3.Analytics Vidya