Capstone Project-2

Supervised ML on Seoul Bike Sharing Dataset (Regression)

- Soma Pavan Kumar(Pro Flex)
- spkumar1998@gmail.com

Contents

Problem Statement

- Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time.

- Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

Introduction

- Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city.

- Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world.

Variable Information

1. Date : year-month-day
2. Rented Bike count - Count of bikes rented at each hour
3. Hour - Hour of he day
4. Temperature-Temperature in Celsius
5. Humidity - %
6. Windspeed - m/s
7. Visibility - 10m
8. Dew point temperature - Celsius
9. Solar radiation - MJ/m2
10. Rainfall - mm
11. Snowfall - cm
12. Seasons - Winter, Spring, Summer, Autumn
13. Holiday - Holiday/No holiday
14. Functional Day – No Func(Non Functional Hours), Fun(Functional hours)

Data Wrangling

- Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis.

- Our Dataset includes about 14 columns and about 8760 observations

- We observed a huge correlation between the variables Temperature and Dew point Temperature. So, we had to drop the column Dew point variable so it doesn't introduce bias in our predictions.

After cleaning the snippet of data looks like:

```
[ ]  # This will show us the first 5 observations of our data
     df.head()
```

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
2	01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	Yes
3	01/12/2017	107	3	-6.2	40	0.9	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
4	01/12/2017	78	4	-6.0	36	2.3	2000	-18.6	0.0	0.0	0.0	Winter	No Holiday	Yes

The df.head() method shows us the first 5 rows of the Dataset.

Let us look at some statistics of the Data

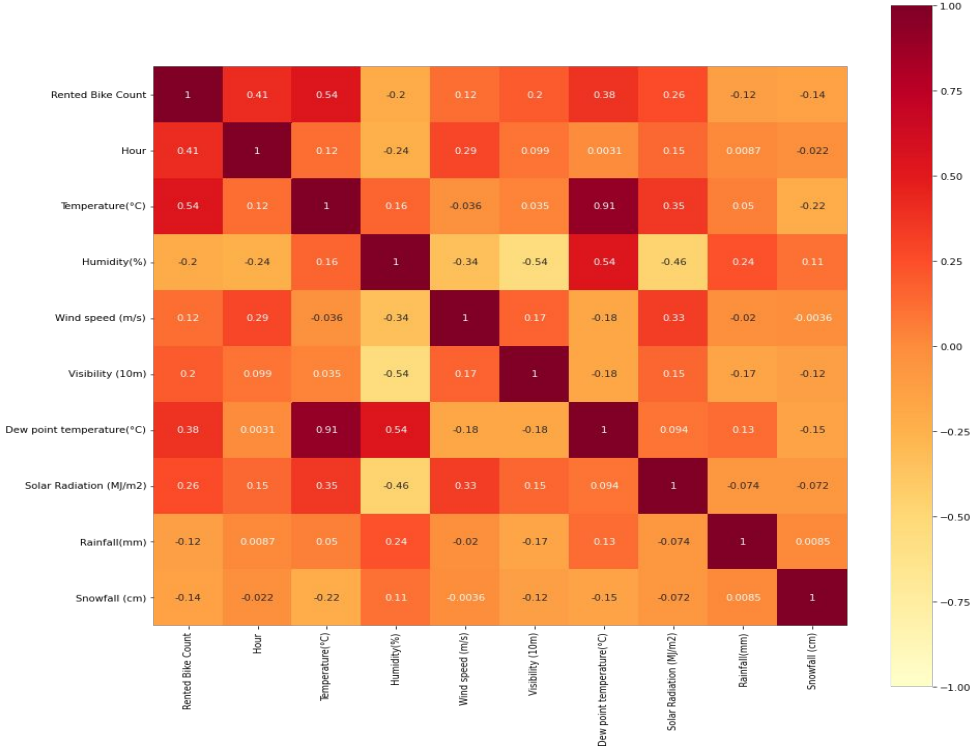
- The Statistics of the data could be found out from an inbuilt function in pandas library called describe() .

```
[ ]  df.describe()
```

	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)
count	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000
mean	704.602055	11.500000	12.882922	58.226256	1.724909	1436.825799	4.073813	0.569111	0.148687	0.075068
std	644.997468	6.922582	11.944825	20.362413	1.036300	608.298712	13.060369	0.868746	1.128193	0.436746
min	0.000000	0.000000	-17.800000	0.000000	0.000000	27.000000	-30.600000	0.000000	0.000000	0.000000
25%	191.000000	5.750000	3.500000	42.000000	0.900000	940.000000	-4.700000	0.000000	0.000000	0.000000
50%	504.500000	11.500000	13.700000	57.000000	1.500000	1698.000000	5.100000	0.010000	0.000000	0.000000
75%	1065.250000	17.250000	22.500000	74.000000	2.300000	2000.000000	14.800000	0.930000	0.000000	0.000000
max	3556.000000	23.000000	39.400000	98.000000	7.400000	2000.000000	27.200000	3.520000	35.000000	8.800000

Lets begin with visualizations

1. Lets plot the Correlations between the independent variables/features in the dataset.

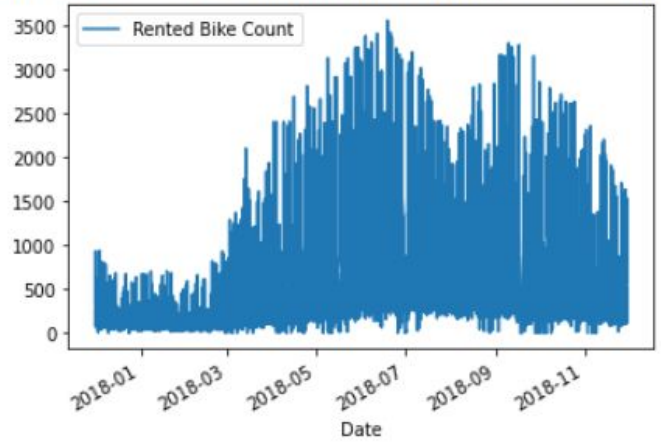


- We find out that there is very high correlation between the Variables between Temperature and Dew point Temperature.
- Also, there is significant correlation between Rented bike count and Temperature.
- And also there is a significant correlation between Dew point Temperature and Humidity

2. Plotting graph and a chart to find out the maximum number of riders according to the seasons:

```
df.plot(x='Date',y='Rented Bike Count',kind='line')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba3b861890>
```

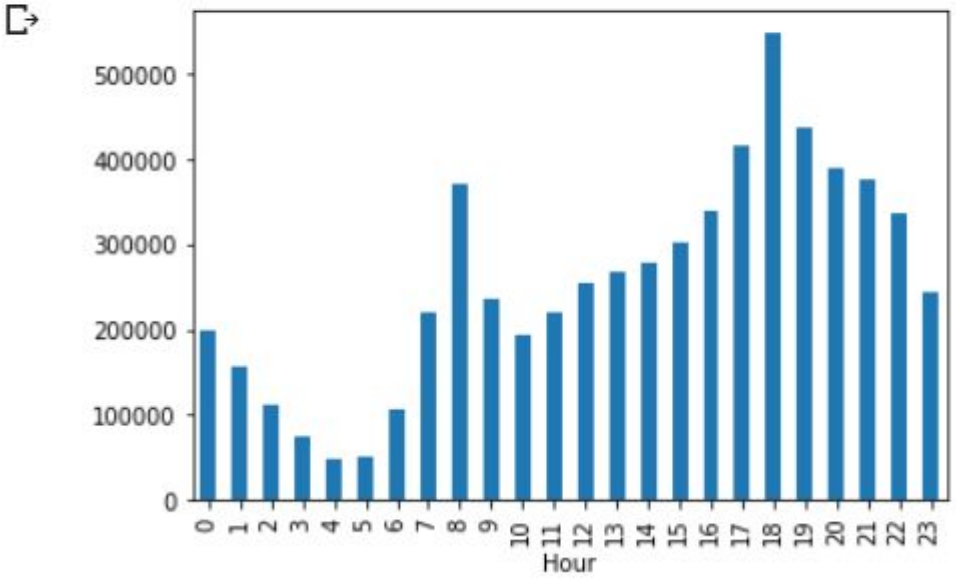


Seasons	Rented Bike Count
Summer	2283234
Autumn	1790002
Spring	1611909
Winter	487169

- The Graphs and the chart show that most number of bikes are ridden in Summer and consequently in autumn.

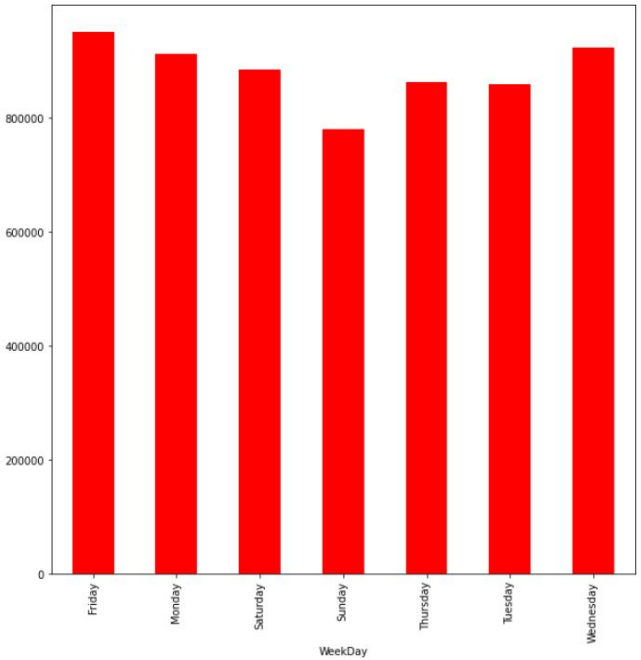
3. Plotting the frequency of number of rides according to the hour of the day

```
df.groupby('Hour').sum()['Rented Bike Count'].plot.bar()
plt.rcParams["figure.figsize"] = (10,10)
```



- By plotting this graph, we see the less obvious that people use the rented bikes at 08:00 in the morning and 18:00 in the evening. Which are the most obvious times for a person to rent a bike while heading to their offices and back.

4. Plotting the frequency of number of rides taken according to day of the week.
To find out that Friday is the busiest in terms of the number of rented bike counts.

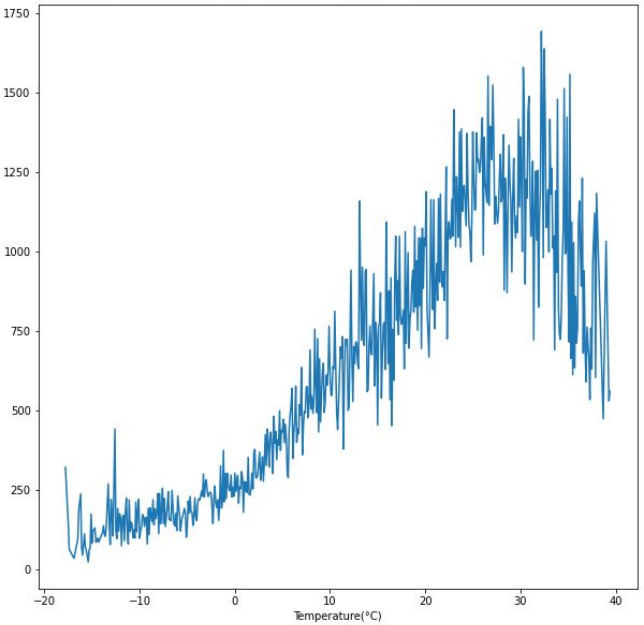


WeekDay	Rented Bike Count
Friday	950334
Wednesday	923956
Monday	911743
Saturday	885492
Thursday	861999
Tuesday	858596
Sunday	780194

5. Plotting the Temperatures in which Koreans would like to ride their bikes

```
df.groupby('Temperature(°C)').mean()['Rented Bike Count'].plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba3ade0710>
```

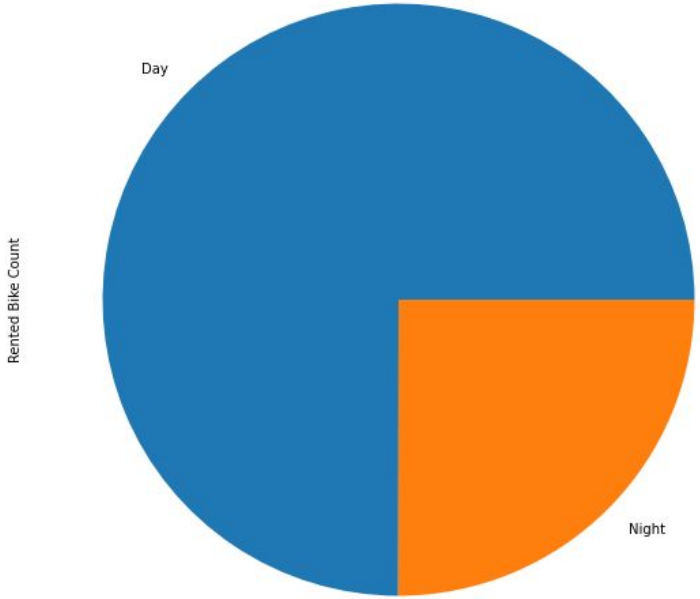


- The sweet spot of temperature in which Korean people like to ride their bikes is around 25 degree Celsius to 30 degree Celsius which can get hot.

6. Plotting which part of the day (Day or Night) do the people of Korea like to ride their bikes:

```
df.groupby('day_or_night').sum()['Rented Bike Count'].plot.pie()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba3af7af90>
```

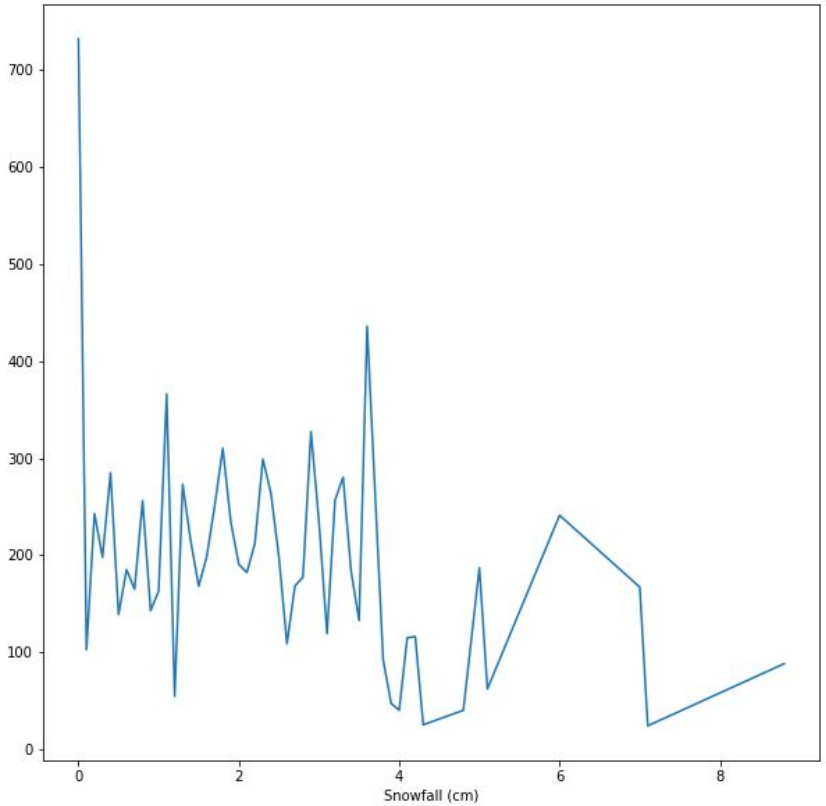


- So, we find out that as usual and sane, more than 75% of Korean people like to ride their bikes in the daylight for their commute to offices back and forth. And only a few people like to ride their bikes after 08:00 in the night.

7. Plotting how many people ride in Snowfall and how much of snowfall.

```
df.groupby('Snowfall (cm)').mean()['Rented Bike Count'].plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fba3ac81850>
```



- People like to ride their rented bikes in Absolute no rainfall and there are people who like to ride their bikes in a little snowfall.

AI

Data modelling

Data Modelling is the process of analyzing the data objects and their relationship to the other objects. It is used to analyze the data requirements that are required for the business processes.

Train test split

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

- The Data is split into X and Y . Where X is all our independent Variables and Y is dependent variables(Rented Bike Count).

- The first five observations of X is shown below

```
X=df.drop(columns=['Rented Bike Count','Date','Holiday'],axis =1)
y=df.iloc[:,1]
```

X contains all the input features important for predicting the y class variable(Rented Bike Count). All the features that do not give much information are removed. Holiday and Functioning Day features are redundant, only one of them is enough.

```
X.head()
```

	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Functioning Day	day_or_night	WeekDay	Month
0	0	-5.2	37	2.2	2000	0.0	0.0	0.0	3	1	1	0	12
1	1	-5.5	38	0.8	2000	0.0	0.0	0.0	3	1	1	0	12
2	2	-6.0	39	1.0	2000	0.0	0.0	0.0	3	1	1	0	12
3	3	-6.2	40	0.9	2000	0.0	0.0	0.0	3	1	1	0	12
4	4	-6.0	36	2.3	2000	0.0	0.0	0.0	3	1	1	0	12

Lets implement Machine Learning Algorithms!

1. **Multiple Linear Regression:**

It is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. Multiple regression is an extension of linear (OLS) regression that uses just one explanatory variable.

After implementation, the intercepts and coefficients of the model are given below.

```
[48] model_linear_reg.intercept_

-235.49099828907356

[49] model_linear_reg.coef_

array([ 2.57430220e+01,  3.03446120e+01, -8.15383426e+00,  1.46762785e+01,
        1.72990710e-02, -1.22523348e+02, -6.29579524e+01,  2.29365116e+01,
       -9.74933145e+01,  9.56278858e+02, -1.28932855e+02,  1.05131519e+00,
        2.80846063e+00])
```

- The R2 scores for Multiple Linear Regression for train dataset is around 55% and same goes for test dataset which is also around 55%. The RMSE value is around 420 as shown in the figure below:

```
[81] #printing the R2 scores of the training and testing dataset.
     Train_r2=r2_score(y_train,pred_train)
     Test_r2=r2_score(y_test,pred_test)
     mse=mean_squared_error(y_test,pred_test)
     rmse=np.sqrt(mse)
     print("The Multiple Linear Regression r2 score for train is {} and for test is {}".format(Train_r2,Test_r2))
     print("The Multiple Linear Regression root mean squared error is : ",rmse)
```

```
The Multiple Linear Regression r2 score for train is 0.5528116899122115 and for test is 0.5546907091912411
The Multiple Linear Regression root mean squared error is :  425.18668060482105
```

Results of the Multiple Linear Regression Model:

1. The Rsquared score for training and testing is around 55% and 55% respectively.
2. The RMSE value is around 420. Trying out other models is advised as the r2 score is too low.

2. Decision Tree Regressor

- Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

- The Decision tree regressor when fitted to our data gives us better results than linear regression model as it is more complex than our previous model.

- Hence as expected, the accuracy scores and R2 scores are better than our previous model.

- The R2 scores for Decision Tree Regressor algorithm is:

```
[84] #printing the best parameters and also the R2 score of the training dataset
     print('The best Decision Tree R2 score is : {:.2f} \n with max depth = {:.2f} \n \
     '.format(gridSearch_DecisionTree.best_score_,gridSearch_DecisionTree.best_params_['max_depth'] ))
```

```
The best Decision Tree R2 score is : 0.84
 with max depth = 12.00
```

```
[85] #printing the best parameters and also the R2 score of the testing dataset.
     print('The best R2 test score is : {:.2f}\n with max depth = {:.2f}\n \
     '.format(bestDecisionTree_testScore,gridSearch_DecisionTree.best_params_['max_depth']))
```

```
The best R2 test score is : 0.77
 with max depth = 12.00
```

```
[86] #let us calculate the RMSE value
     rmse_dt=np.sqrt(mean_squared_error(y_test,pred_test_dt))
```

```
     print("The root mean squared error for the Decision Trees Regressor on the test dataset is: ", rmse_dt)
```

```
The root mean squared error for the Decision Trees Regressor on the test dataset is:  303.19389395292643
```

- The R2 scores for train dataset is 0.84 or 84%with a max depth of 12.
- And the Best R2 scores for test dataset is 0.77 or 77% with the same max_depth parameter. The RMSE value is around 300.

3. Random forest regressor

- Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

- The Random forest is a tad bit more complex than decision trees as it uses multiple Decision trees and hence it is expected to have more accuracy in this model.

```
[91] print("The r2 score obtained for the testing dataset is:",r2_score(y_train,predict_RF_train))
```

```
The r2 score obtained for the testing dataset is: 0.9876771514636705
```

```
[92] print("The r2 score obtained for the training dataset is:",r2_score(y_test,predict_RF))
```

```
The r2 score obtained for the training dataset is: 0.8984730121084518
```

```
[93] #calculating the rmse value for random forest model
     rmse_rf=np.sqrt(mean_squared_error(y_test,predict_RF))
     print("The root mean squared error for the Decision Trees Regressor on the test dataset is: ", rmse_rf)
```

```
The root mean squared error for the Decision Trees Regressor on the test dataset is:  203.02032753725263
```

- The R2 scores for Random Forest train dataset is 0.98 or 98%.
- The R2 scores for Random Forest test dataset is 0.89 or 89%.

The RMSE value is around 200.

4. XGBoost regressor

XGBoost is an efficient implementation of the gradient boosting algorithm that can be used for regression predictive modelling. It is fast because of the capacity to do parallel computation on a single machine.It is one of the best Regression models that can be used. It is also used in finding important features of a dataframe.

```
[98] gridSearch_xgb.best_score_

     0.9029464316771395

[99] bestxbg_testScore

     0.8852651783968575
```

Result of the XGBoost Regressor Model:

1. The R2 score obtained with XGBoost Regressor is around: 90%
2. The RMSE value for the XGBoost Regressor without hyperparamter tuning is around 200
3. The R2 score obtained with XGBoost Regressor after hyperparamter tuning on the testing dataset is around 88.5%.

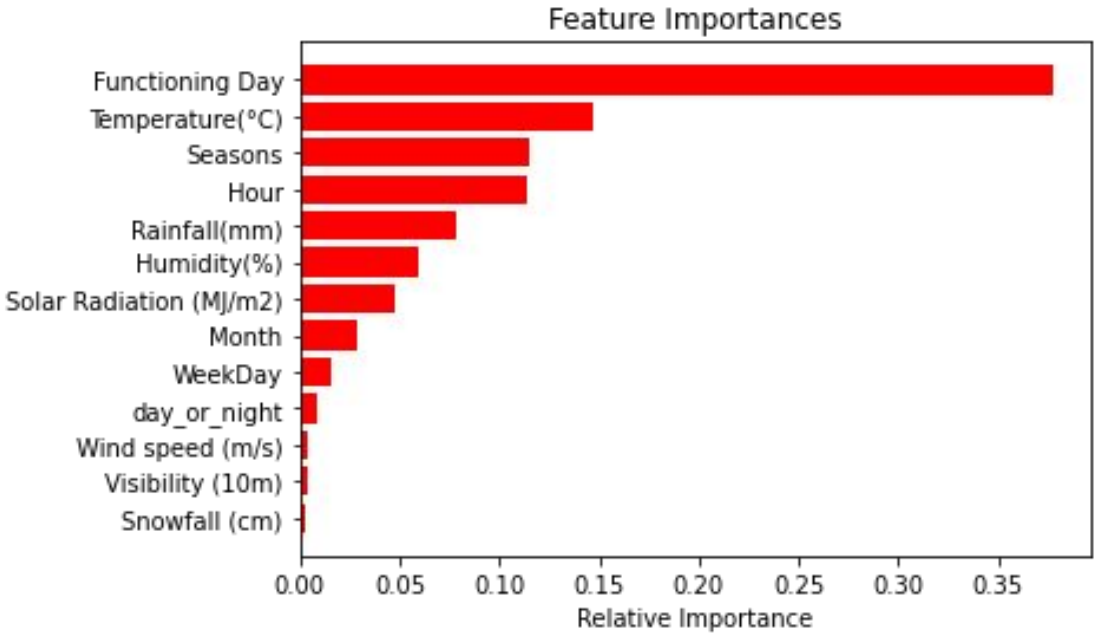
- The accuracy scores or R2 scores for XGBoost is very good, almost as good as Random Forest Regressor.

- The Train dataset R2 score is 0.9 or 90% with estimators of 1000 and best learning rate is 0.1

- The Test dataset scores R2 score is 0.885 or 88.5% with the same hyper parameters.

- The RMSE value is around 200.

Feature Importances using XGBoost

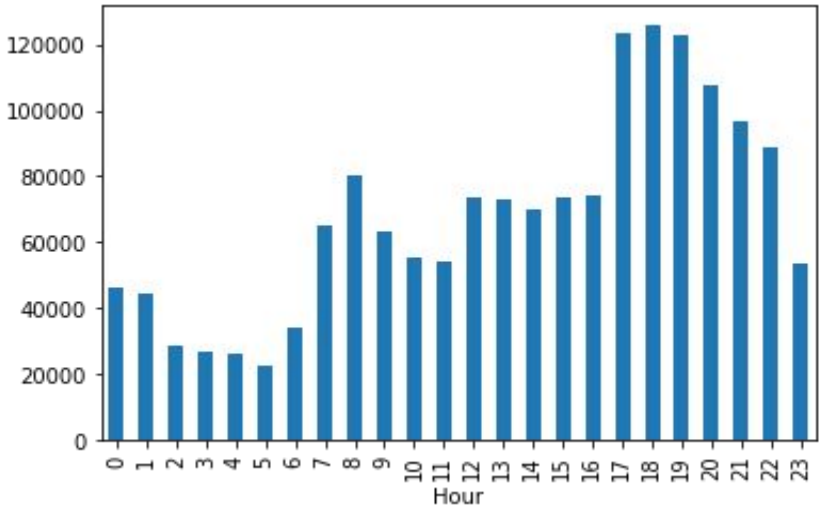


The most important features are:

1. Functioning Day
2. Temperature
3. Seasons

Hence, these are the features that are the most important in determining the number of bikes rented that particular day.

Results: Per Hour Demand as per the Random Forest Model



Hence, at 8 AM in the morning and from 5pm to 8 pm the demand in very high. The is slight rise in demand at 10pm. These are the timings at which the bikes should be available in good number in order to decrease the waiting time.

Here, the testing dataset is used as input to the XGBoost Regressor model and the results are plotted in the form of a graph.

Conclusion

- The Dataset I choose to Explore is the data on Seoul Bike Sharing Dataset. Which includes the data which influences people renting a bicycle may it be Temperature, Humidity and so on.

- The Parameter that does not affect the number bikes ridden is the week day or which day of the week. Regardless of which week day it is the number of rides taken remains more or less the same.

- We also saw that people like to rent and ride their bikes in office hours that is 08:00 in the morning and 18:00 in the evening.

- And People like to ride when there is the least rainfall and snowfall.

- So, to predict we would like to use the Random Trees Regressor algorithm to make further predictions.

- The Results were predicted and were plotted in the form of a graph to find out at hour of the day the demand is high, so that the availability of the bikes can be ensured in order to reduce the waiting time.