Capstone Project Regression

(Abhishek Mishra, Kurva Mallesh, Arunesh Mishra)
Data science trainees,
Alma Better, Bangalore

Abstract:

Bike sharing market is growing all over the world these years. It is meaningful to do some analysis about it because more and more companies are getting into this business. Since bike sharing is a recent phenomenon, not that many related analyses have been done upon this topic. However, more and more bike sharing companies have started to realize the importance of data driven decision making. One of the major aspects that can be addressed by data analysis is to predict the demand of bikes on any given day. Knowing the demand would help us in creating a better supply and subsequently reduce the gap between supply and demand.

Our EDA can make us understand data which variable is very important and check how every variable connected with dependent variable.

We make some models to predict the label column based on feathers.

Keywords: EDA, Seoul Bike Sharing

1. 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.

Ddareungi (Korean) is Seoul's bike sharing system, which was set up in 2015. It is also named **Seoul Bike** in English.

Ddareungi was first introduced in Seoul in October 2015 in select areas of the right bank of the Han River. After a few months, the number of stations reached 150 and 1500 bikes was made available. In 2016, the number of stations has increased steadily to cover new districts.[3] As of July 2016, there were about 300 stations and 3000 bikes available, and Seoul mayor Park Won-soon confirmed his intention to increase the number of bikes available to 20,000.

Data Description: -

- **Date: -** year-month-day
- **Rented Bike count: -** Count of bikes rented at each hour.
- **Hour: -** Hour of the day
- **Temperature: -** Temperature in Celsius
- **Humidity: -** Humidity %
- **Wind speed** - Wind speed m/s
- **Visibility -** Visibility 10m

- **Dew point temperature –** Celsius
- **Solar radiation -** MJ/m^2
- **Rainfall -** Rainfall mm
- **Snowfall –** cm
- **Seasons -** winter, spring, summer, autumn
- **Holiday -** Holiday/No holiday
- **Functional Day –** No Function (Non-Functional Hours), Function (Functional hours)

2. EDA

Exploratory Data Analysis (EDA):

After the data wrangling step, we performed EDA by comparing different parameters which are involved in the dataset. EDA helps us to find the different relations among the parameters. It involves the visualization of the data by comparing the different parameters to find out the best among all.

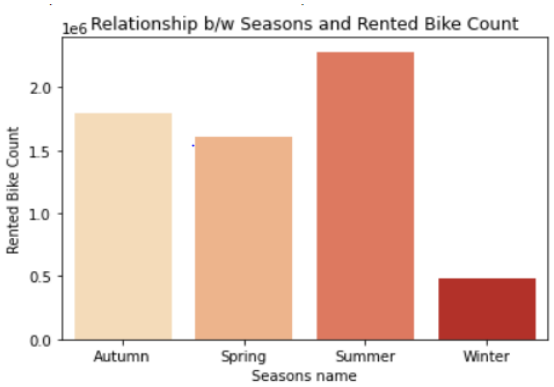
In the process of understanding the data we found that Rented Bike count is the label column and rest all are feathers we did some analysis. We explain each one by one.

Rented Bike Count vs Seasons plot?

1. According to graph we can say that in "Summer" there are 2283234 rented bike bookings happened.
2. We can see that in "autumn" season there are 1790002 rented bike bookings happened.

	Seasons	Rented Bike Count
0	Autumn	1790002
1	Spring	1611909
2	Summer	2283234
3	Winter	487169

3. And we can say that in "Spring" there are 1611909 rented bike bookings happened.
4. According to graph we can say that in "Winter" there are 487169 rented bike bookings happened.

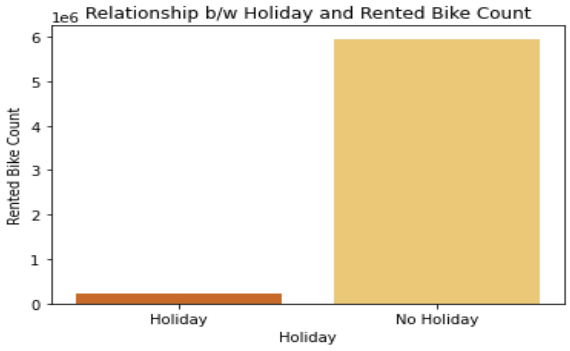


Relationship b/w Holiday and Rented Bike Count?

1. According to the graph we can say in the non-holiday time there are 5956419 bikes rented count

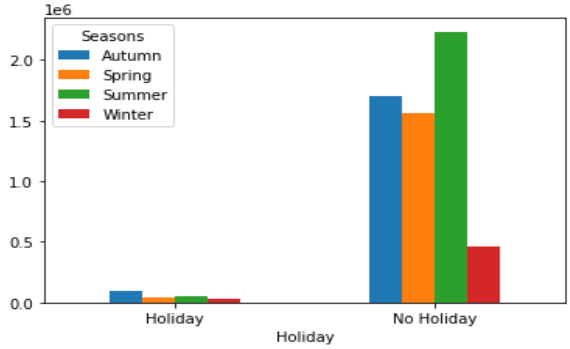
	Holiday	Rented Bike Count
0	Holiday	215895
1	No Holiday	5956419

2. In Holiday only 215895 counts happened.
3. Again, we can understand that Seasons is impacting on Holiday.
4. Normally we know there are high bike bookings in Summer.
5. But during holiday time in Autumn season there was very high rented bike count.



Also using season and holiday rented bike count: -

Seasons	Autumn	Spring	Summer	Winter
Holiday				
Holiday	91018	45742	49063	30072
No Holiday	1698984	1566167	2234171	457097

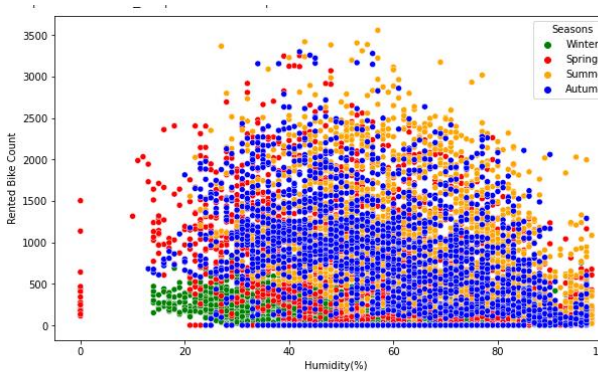


Relationship b/w seasons humidity and bike count

1. According to bar plot Humidity is not making huge impact on Rented Bike Count.

	Seasons	Humidity_min	Humidity_max
0	Autumn	13	97
1	Spring	0	98
2	Summer	21	98
3	Winter	14	97

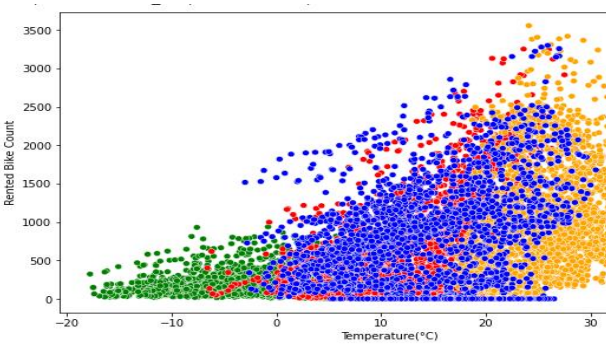
2. We found there are some rows that have Humidity as 0; it is not possible so we need to check this.



Temperature, Rented Bike Count for different Seasons: -

	Seasons	Temperature_min	Temperature_max
0	Autumn	-3.0	30.5
1	Spring	-6.6	29.4
2	Summer	16.3	39.4
3	Winter	-17.8	10.3

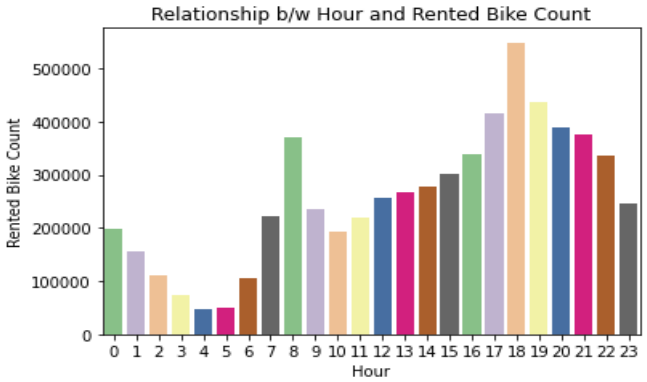
1. According to scatter plot Rented Bikeike Count is very low at Temperature(°C) less than -10
2. In Winter **Season** Temperature goes min -17.8°C



3. This type of condition comes in winter season. So we can suggest the company in the winter season to check the Temperature (°C) tends for each day and plan accordingly.
4. The temperature(°C) less than -10 have impact on Rented Bike Count.

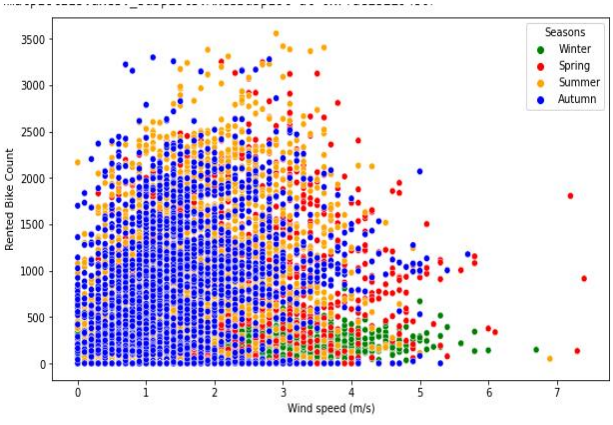
Rented Bike Count vs Month plot

1. According to the graph we can observe that at 8am there are good number of bookings.
2. At **6pm** there are **very high** bookings.



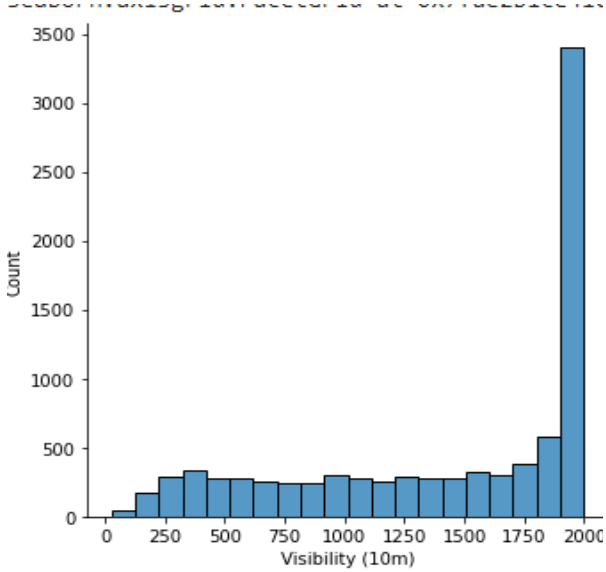
3. We can suggest the company to give high preference to 8pm.
4. Also we can see there are very less bookings b/w 3pm - 6pm

Relationship b/w seasons Wind speed and bike count



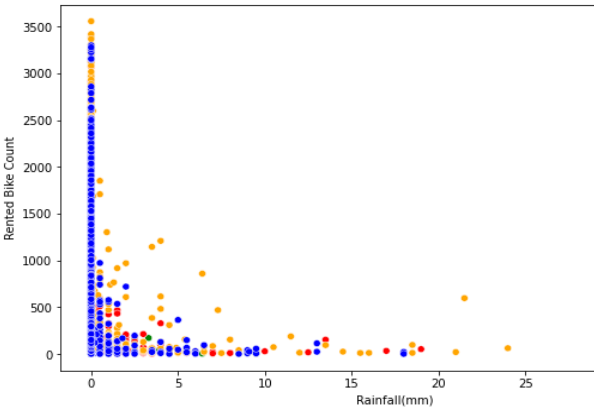
❖ We can see there are not much bike rented count after Wind speed 5(m/s)

Relationship b/w Visibility and bike count



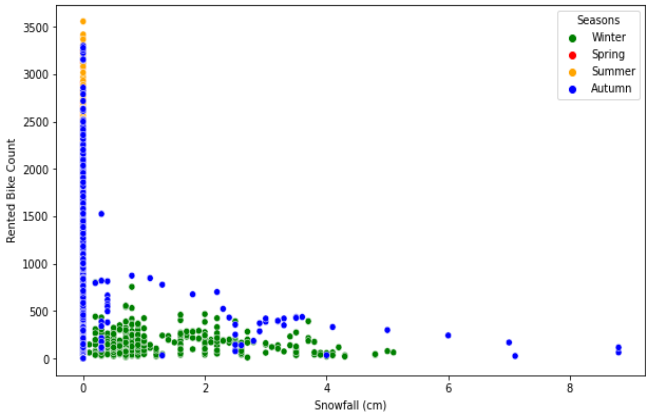
- According to the distribution plot we can see there are very high values of 2000(10m) Visibility.

Relationship b/w seasons Rainfall and bike count



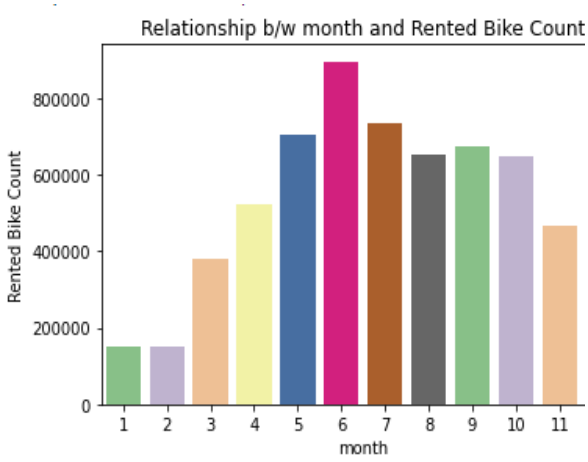
- According to scatter plot we can say that there is very less Rented bike count when Rainfall greater than 5(mm).
- Also, we can see that there is very less possibility of more than 5(mm) Rainfall in the Winter season.

Relationship b/w seasons Snowfall (cm) and bike count



- According to previous scatter plot we can see that Snowfall happens only in Winter and Autumn seasons.
- Also, we can see Snowfall directly impacting on Rented Bike Count.
- And there is no Snowfall in Summer and Spring seasons.

Relationship b/w Date and bike count: -



- According to bar plot we can say that Rented Bike Count is very high in June month.

5. Exploratory Data Analysis (EDA) Conclusion: -

- ❖ Seasons are making huge impact on **Rented Bike Count**. There was very high count in **summer (2283234)** and very low count in **Winter (487169)** season.
- ❖ We know that there is **high Rented Bike Count** in **summer** but during **Holiday** people like to book the bikes in **Autumn** season more. So, we can say Autumn season is best for Holidays.
- ❖ We found there are some rows that have **Humidity** as **0** so it is not possible so we need to check this.
- ❖ The **Temperature**(°C) Less than **-10** making huge impacting on **Rented Bike Count.**
- ❖ We suggest the company to give **high preference** to **8pm**. Because at that time **Rented Bike Count** is very high.
- ❖ We can say that there is **not much** bike ranted count after **Wind speed 5(m/s)**
- ❖ We can say that there are very less **Rented bike count** when **Rainfall** is greater than **5(mm)**.
- ❖ Also, we can see **Snowfall** directly **impacting** on **Rented Bike Count**.
- ❖ According to bar plot we can say that **Rented Bike Count** is very high in **June** month.

6. Feature Engineering: -

- ❖ Encoding on date column, Split month from the date.
- ❖ Creating new feature week day and weekend.
- ❖ Do some feature engineering on hour column and make them as evening, morning, noon, night so the model can easily understand.
- ❖ Used correlation plot and variance inflation factor (VIF) for features selections.
- ❖ In data pre-processing we used power transformer, minimum, maximum scaler for standardize the data.

7. Model Implementation: -

Linear Regression: -

Using Linear regression, we get our scores like-

- Training score = 0.7528304949472668
- MAE: 4.840998284567279
- MSE: 38.178135932975174
- RMSE: 6.178845841496224

- R2: 0.7477419157511107
- Adjusted R2: 0.7444734167418701
- Coefficient

  [5.95508209 -1.62770952 0.2515433 0.5636376 0.72879401 -3.06033017

  0.07512372 -1.90288105 -5.19005316 -3.693359 -0.10264436 -0.29086177

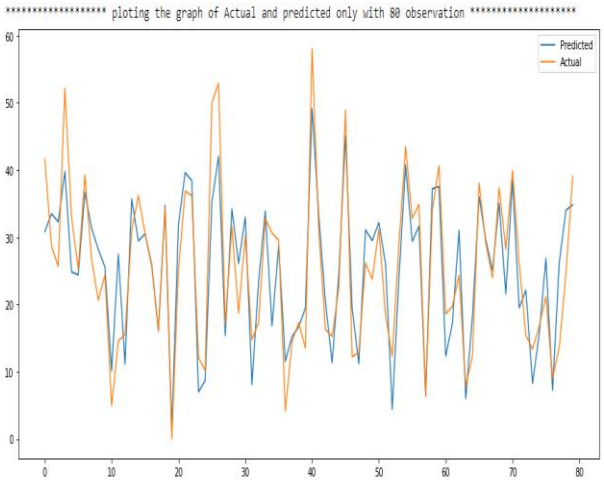
  -0.99552432 0.67296183 5.36751057 -1.12073129 -0.06739495 -0.86094349

  -0.64704363 -0.42827696 1.11789926 -0.52207466 0.54913309 0.70144529

  1.10463537 0.32806082 0.06739495 -0.62810677]

- Intercept = 23.442819487037085

  Below is the plot for actual and predicted values using Linear Regression model.

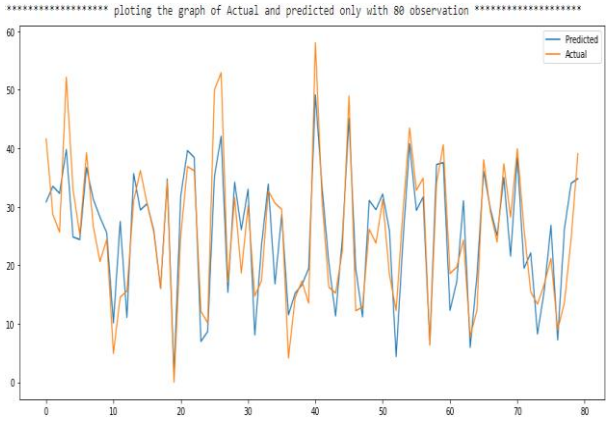


Lasso (hyper parameter tuning): -

Using Lasso regression, we get our scores like-

- Training score = 0.752830494923367
- The best parameters found out to be: {'alpha': 1e-05}
- Where model best score is: 0.7504355889277325
- MAE: 4.8409962817034815
- MSE: 38.17811853331392
- RMSE: 6.178844433493525
- R2: 0.7477420307175728
- Adjusted R2: 0.7444735331979486

Below is the plot for actual and predicted values using Lasso Regression model.

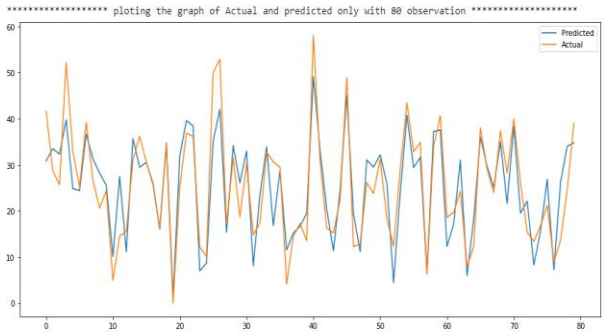


Ridge (hyper parameter tuning): -

Using Ridge regression, we get our scores like-

- Training score = 0.752830271580107
- The best parameters found out to be: {'alpha': 1.9}
- Where model best score is: 0.7504361064287439
- MAE: 4.840845216063553
- MSE: 38.175954219950114
- RMSE: 6.178669292003749
- R2: 0.7477563311942599
- Adjusted R2: 0.7444880189654025

Below is the plot for actual and predicted values using Ridge Regression model.

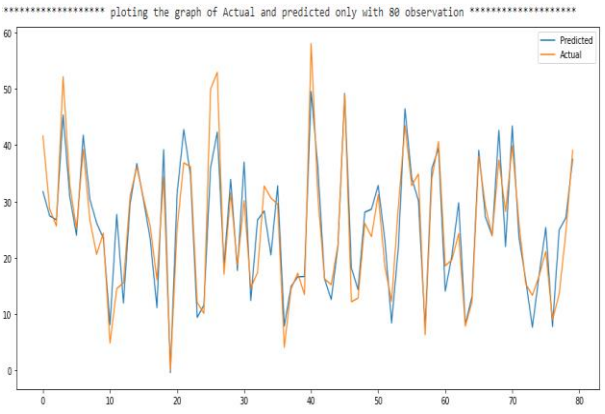


Linear Regression (Polynomial Features): -

Using Linear Regression (Polynomial Features) regression, we get our scores like-

- Training score = 0.8639314249868966
- MAE: 3.5308132120461275
- MSE: 23.097768928108806
- RMSE: 4.806013829371365
- R2: 0.8473838809087664
- Adjusted R2: 0.8095343872914992

Below is the plot for actual and predicted values using Linear Regression (Polynomial Features) model.

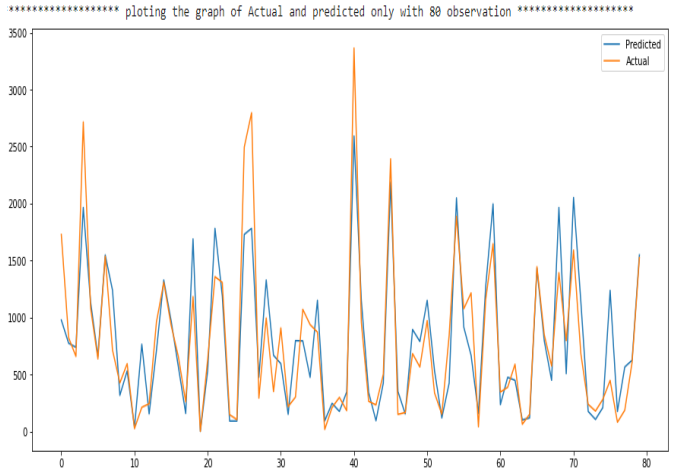


Decision Tree: -

Using Decision Tree regression, we get our scores like-

- Training score = 0.8337445528699623
- The best parameters found out to be: {'criterion': 'mse', 'max_depth': 15, 'max_features': 24, 'min_samples_split': 50, 'splitter': 'random'}
- Where model best score is: 0.7931489412927507
- MAE: 194.41948198148998
- MSE: 87753.14028462046
- RMSE: 296.23156530764993
- R2: 0.7855500258095656
- Adjusted R2 : 0.782670836341268

Below is the plot for actual and predicted values using Decision Tree model.



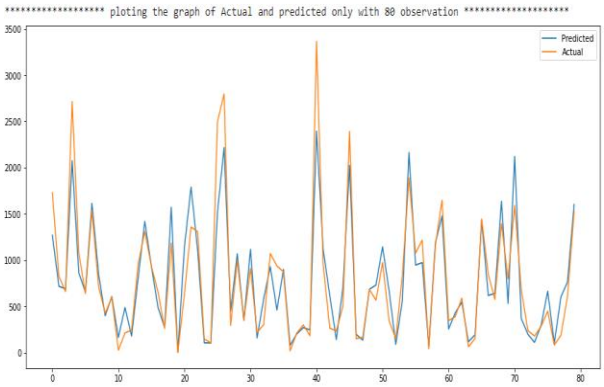
Using Random Forest Regression: -

Using random forest regression, we get our scores like-

- Training score = 0.9511223280260815

- The best parameters found out to be: {'max_depth': 20, 'max_features': 24, 'min_samples_split': 10, 'n_estimators': 100}
- where model best score is: 0.8577875917155999
- MAE: 154.6444976595282
- MSE: 57299.259482874615
- RMSE: 239.37263728938322
- R2: 0.8599728206035834
- Adjusted R2: 0.8580928260653907

Below is the plot for actual and predicted values using Random Forest Regression model.

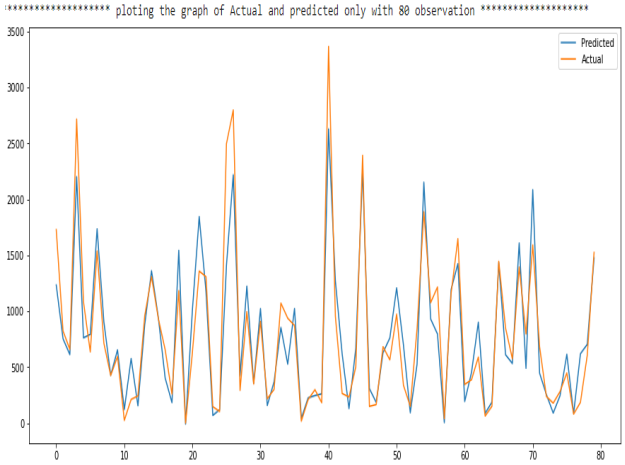


Gradient Boosting Regression: -

Using Gradient Boosting Regression, we get our scores like-

- Fitting 5 folds for each of 80 candidates, totaling 400 fits
- Training score = 0.9600157356513978
- The best parameters found out to be: {'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 250}
- where model best score is: 0.8589398718814742
- MAE: 163.202950996003
- MSE: 61375.71470538196
- RMSE: 247.74122528433162
- R2: 0.8500108327542623
- Adjusted R2: 0.8479970893051296

Below is the plot for actual and predicted values using Gradient Boosting Regression model.

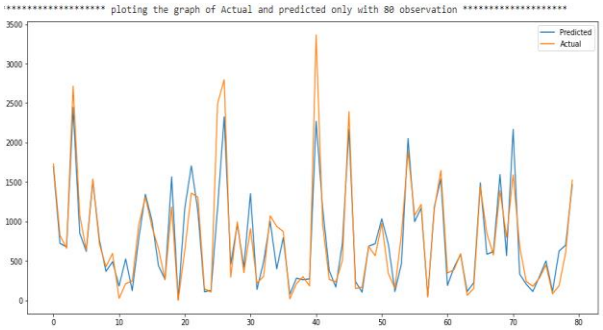


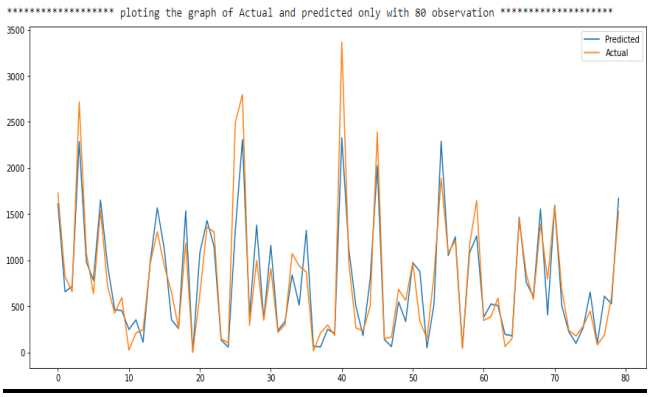
Adaboost Boost Regression:-

Using Adaboost Boost Regression we get our scores like-

- Training score = 0.9979923433449015
- The best parameters found out to be: {'base_estimator': DecisionTreeRegressor(), 'learning_rate': 1.5, 'n_estimators': 150}
- Where model best score is: 0.8586673277122084
- MAE: 151.6972602739726
- MSE: 60155.32739726028
- RMSE: 245.26583006456542
- R2: 0.8529931992642262
- Adjusted R2: 0.8510194968469403

Below is the plot for actual and predicted values using Adaboost Boost Regression model.





XGBoost Regression: -

Using XGBoost Regression, we get our scores like

- Training score = 0.9999999999081989
- The best parameters found out to be: {'learning_rate': 0.5, 'max_depth': 15, 'n_estimators': 100}
- where model best score is: 0.8312029908133877
- MAE: 159.7988008220582
- MSE: 63704.871485765885
- RMSE: 252.39823986265412
- R2: 0.8443188699388147
- Adjusted R2: 0.8422287066185488

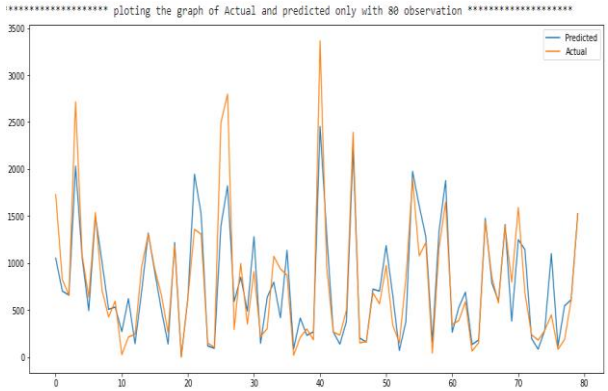
Below is the plot for actual and predicted values using XGBoost Regression model.

KNN Regression:-

Using KNN Regression we get our scores like-

- Training score = 0.8873684520023236
- MAE: 182.88365296803656
- MSE: 79982.99894063927
- RMSE: 282.8126569668325
- R2: 0.8045385953954318
- Adjusted R2: 0.8019143450558335

Below is the plot for actual and predicted values using KNN Regression model.



8. Creating Data Frame of all Evaluation Matrix with respect of models: -

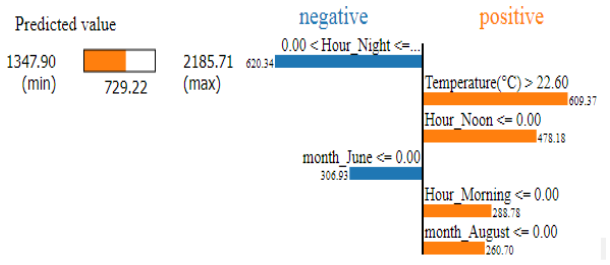
	Linear	Lasso	Ridge	Polynomial	Decision_Tree	Random_Forest
Mean_Absolute_error	4.840998	4.840996	4.840845	3.530813	194.419482	154.644498
Mean_square_error	38.178136	38.178119	38.175954	23.097769	87753.140285	57299.259483
Root_Mean_square_error	6.178846	6.178844	6.178669	4.806014	296.231565	239.372637
Training_score	0.752830	0.752830	0.752830	0.863931	0.833745	0.951122
R2	0.747742	0.747742	0.747756	0.847384	0.785550	0.859973
Adjusted_R2	0.744473	0.744474	0.744488	0.809534	0.782671	0.858093

In the second picture all rows are dependent with 1st image.

Gradient_Boosting_Regressor	Ada_Boost_Regressor	XGBoost_Regressor
163.202951	151.697260	159.798801
61375.714705	60155.327397	63704.871486 79
247.741225	245.265830	252.398240
0.960016	0.997992	1.000000
0.850011	0.852993	0.844319
0.847997	0.851019	0.842229

9. model expansibility: -

Model explainability using lime



10. Conclusion from Model Training: -

About some Feature engineering

- First, we converted categorical values to num values.
- Converting date column D type objects to date. Split **day of week**, **month** and **year** in three columns
- Creating new feature **Week Day** and **weekend**. And dropping the days of week column from data and from categorical feature.
- Converted hour column in three parts like Night, Noon, Evening, Morning For better prediction.
- Using correlation plot we found **"Dew point temperature(°C)"** variable is highly correlated.
- Using VIF there is '**Dew point temperature(°C)**' is **multicollinearity**. So, decided to drop Dew point temperature(°C).

About models

- Also using **Power Transformer** for scaling and transform it after train test split so we can avoid over fitting.
- Create a function for print the scores so we can use this function.
- Fist we use some **Linear Regression** and then use **Lasso** and **Ridge** with the help of **Hyper parameter Tuning**
- Using **Polynomial Features** with **Linear Regression**. but we did not get good scores.
- Then we use Tree Base Models because we know **multicollinearty not** affect **tree base models**. So used Random **Forest Regression** and **Decision Tree**.
- Then we decided to use some boosting regression for better predictions. So, we

used **Gradient Boosting Regression**, **Adaboost Boost Regression**, **XGBoost Regression**. We got our best **Rmse** score with Linear, Lasso, Ridge, Polynomial. We got our best **Training score** with Random Forest. We got our best R2, Adjusted_R2 with Decision Tree, Random Forest, Bagging.

References-

1. scikit-learn
2. Matplotlib
3. Seaborn
4. Machine Learning Mastery
5. Geeks for Geeks
6. Analytics Vidhya
7. Wikipedia