Capstone Project - 2

Seoul Bike Sharing Demand Prediction

OverView

- 1. Defining Problem Statement
- 2. Cleaning Data and EDA
- 3. Preparing Data set for modelling
- 4. Applying Model
- 5. Feature Selection
- 6. Model Selection and Validation

Problem Description

1. Due to global warming, increasing pollution and depletion of energy resources, many countries are focused on using renewable sources of energy where ever they can. South Korea is among them.
2. Seoul being the capital of South Korea has a Rented Bike Sharing system, which allows user to rent bike for certain amount of period for certain amount of money.
3. We have been provided with Bike Sharing data of two years and we are expected to build a model which can predict further demand of bikes.

Data Pipeline

- 1. Data Cleaning :- In this we go through features to check if null values were present in them and either to replace or drop them.
- 2. Outlier Detection :- Most the time, data do contain outliers in them, so it is necessary to check them as well.
- 3. EDA :- We did Exploratory Data Analysis (EDA) to gain important insights.
- 4. Create a Model :- In this we tried different model in-order to show the approach towards solving a problem and simultaneously increasing it productivity.

Data Summary

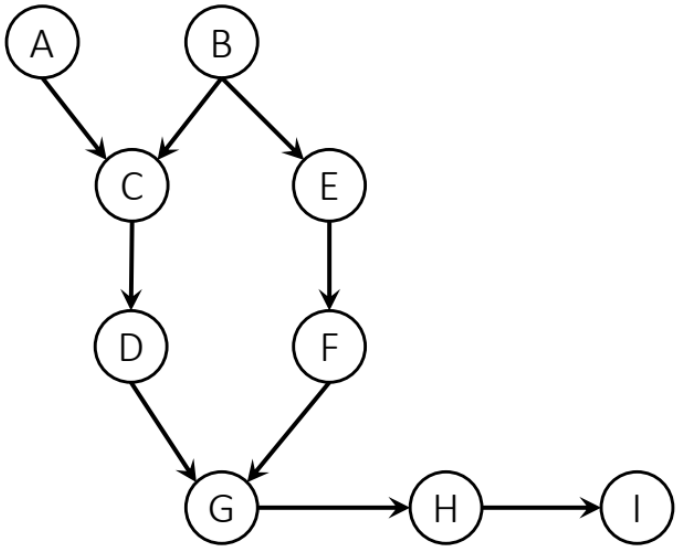
The dataset contains 8760 rows and 14 columns

Out of 14 columns, we have 2 categorical column and rest numerical column

- Date : year-month-day
- Rented Bike count - Count of bikes rented at each hour
- Hour - Hour of he day
- Temperature-Temperature in Celsius of that particular day
- Humidity - %
- Windspeed - 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 - No Func (Non Functional Hours), Fun(Functional hours)

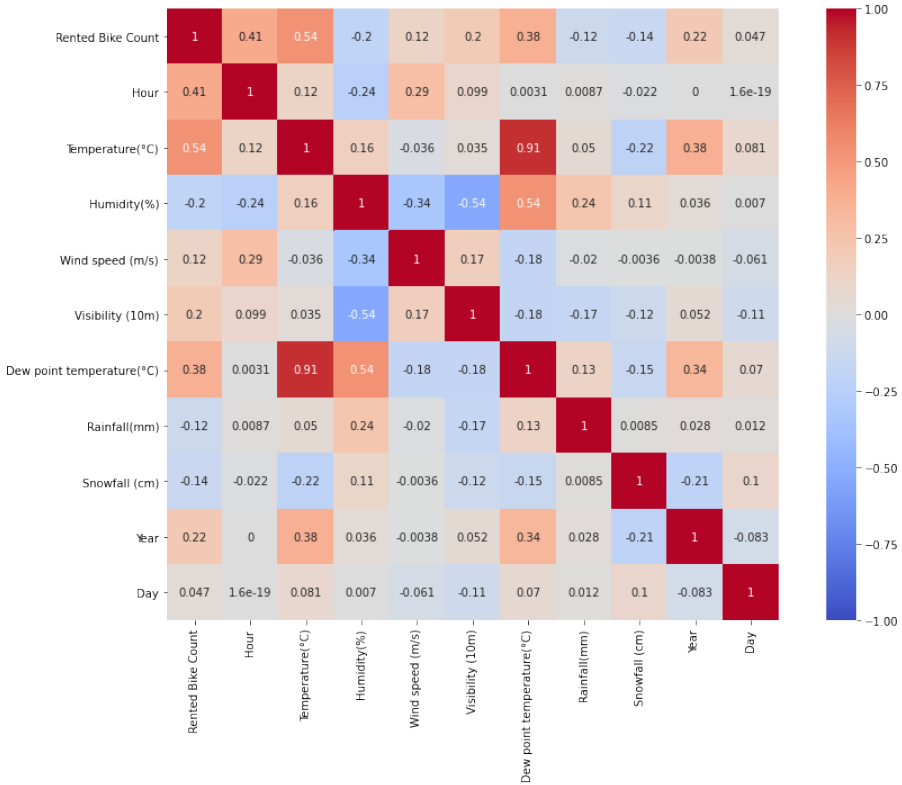
Independent Parameters

1. Month :- Month of the year
2. Season :- Type of season
3. No Holiday :- If holiday is there then 0, else 1
4. Night :- If Bike is rented in night or day.
5. Yes :- If the day is functional day or not.
6. Hour :- Hour of the day
7. Temperature :- In degree Celsius
8. Humidity
9. Rainfall :- If any
10. Snowfall :- If any
11. Visibility :- Due to rainfall and snowfall
12. Windspeed :- in m/s
13. Day of week

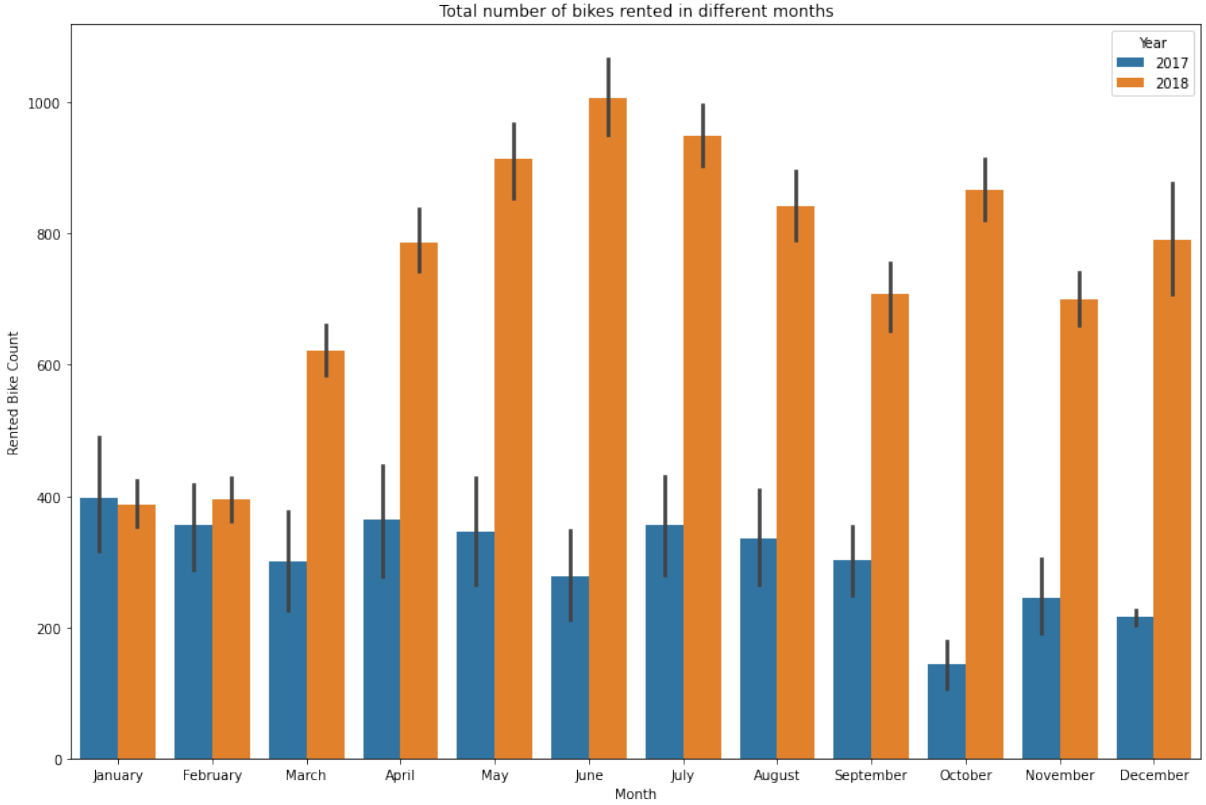


Exploratory Data Analysis

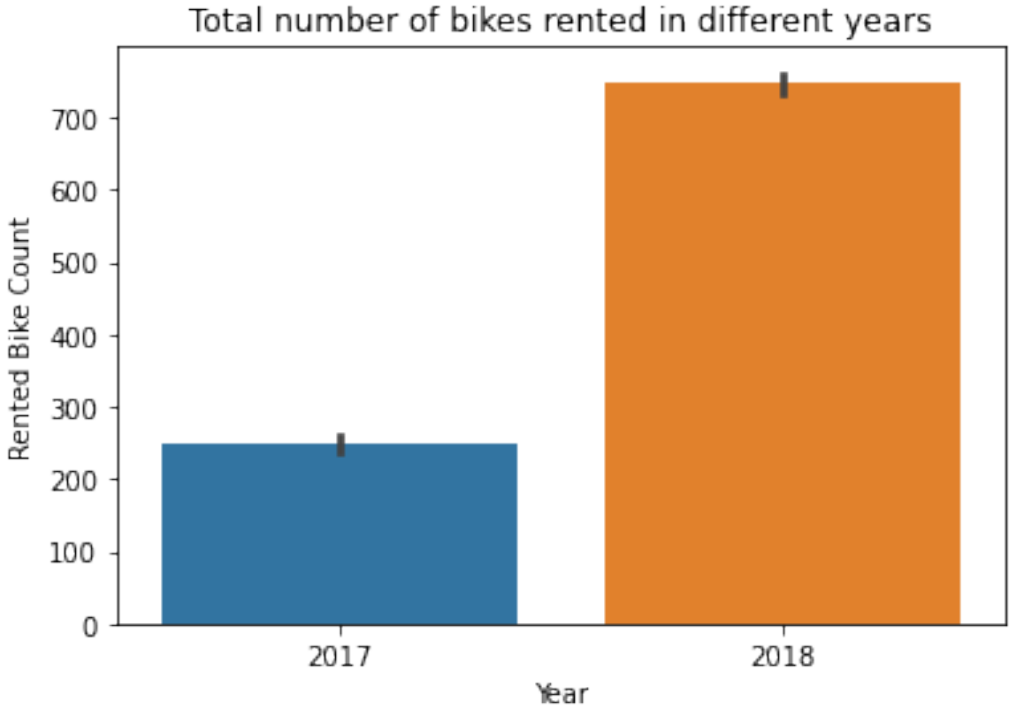
Correlation Heat-Map



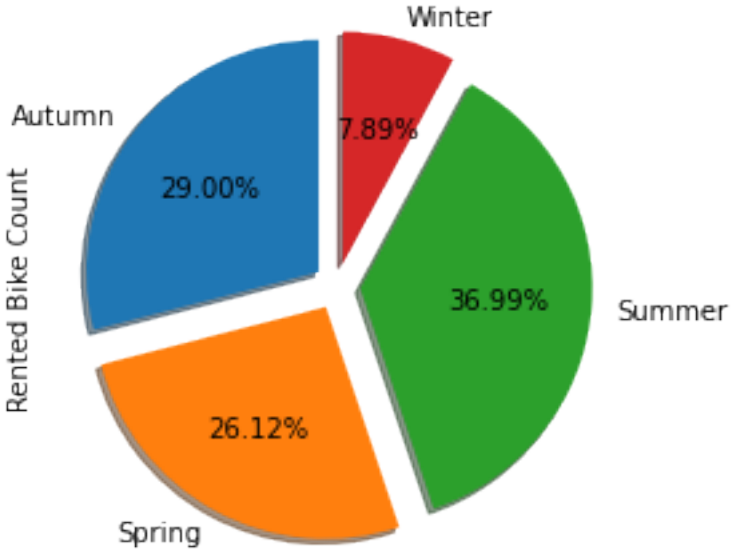
Total Number of Bikes Rented in different months



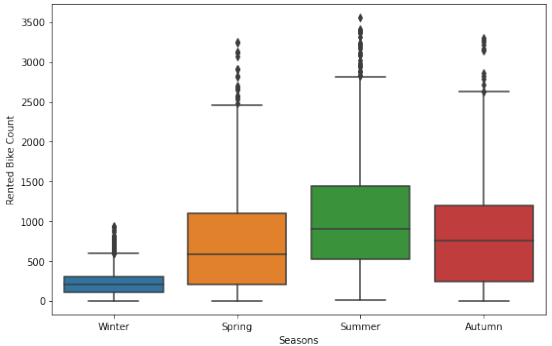
EDA (contd.)

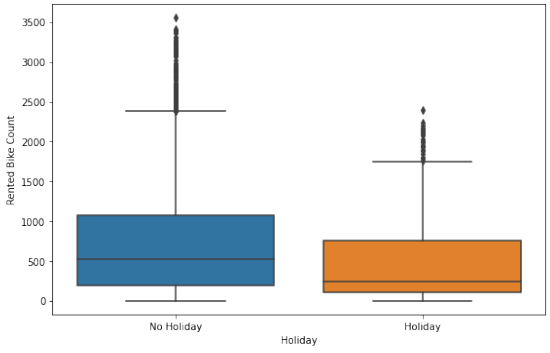

Total number of bikes rented in different years
Rented Bike Count
0
100
200
300
400
500
600
700
2017
2018
Year

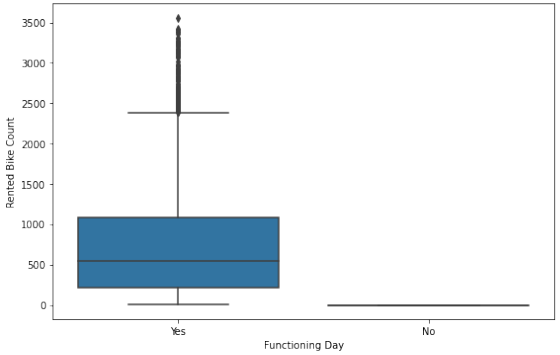
Rented Bike Count in Different season

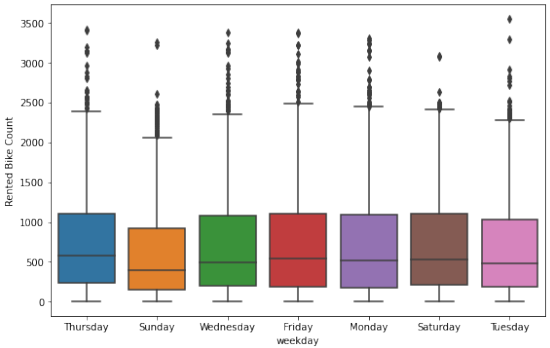


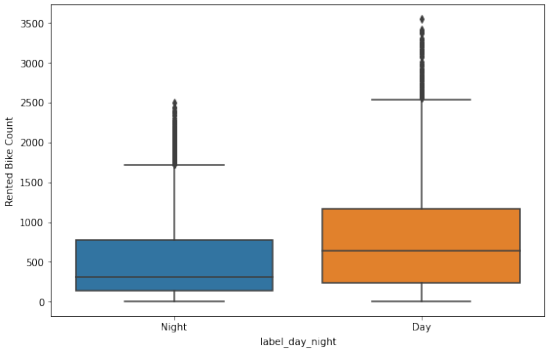
EDA (contd.)

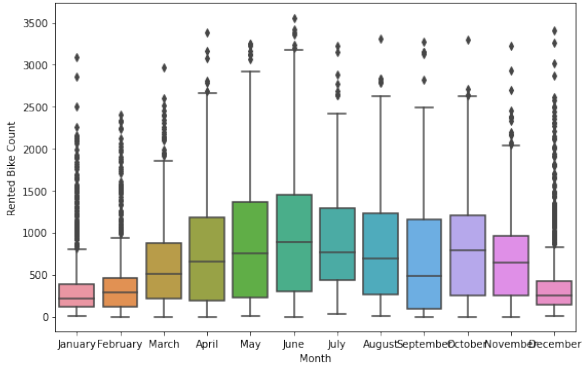

Rented Bike Count
Winter
Spring
Summer
Autumn
Seasons


Rented Bike Count
No Holiday
Holiday
Holiday

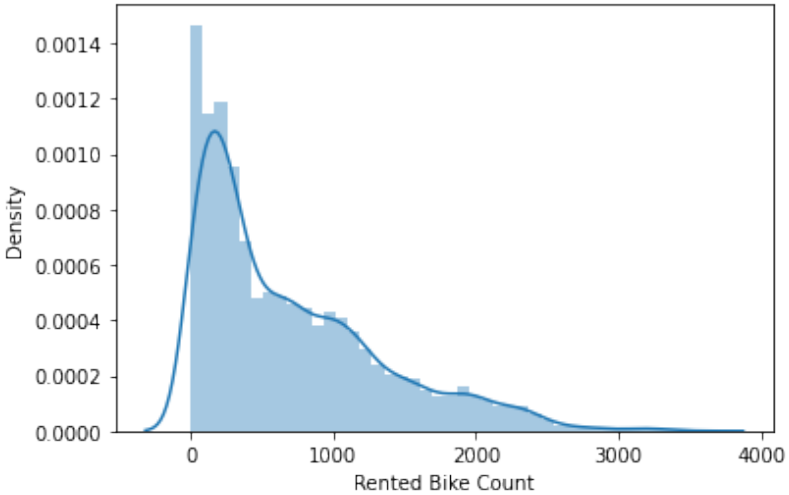

Rented Bike Count
Yes
No
Functioning Day

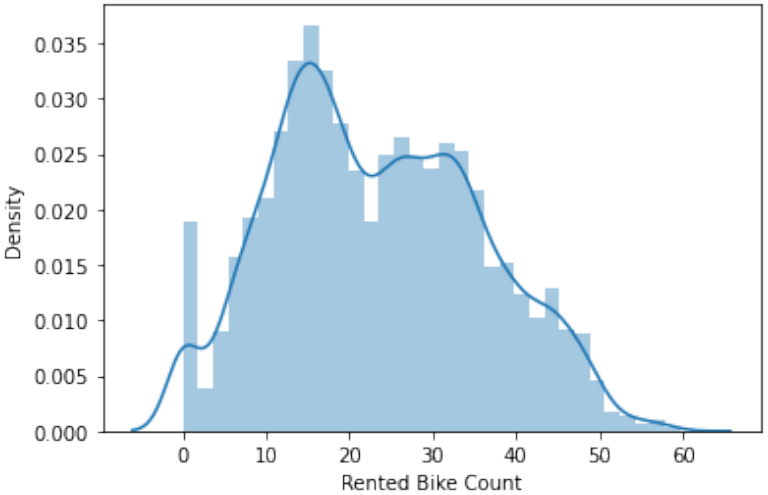

Rented Bike Count
Thursday
Sunday
Wednesday
Friday
Monday
Saturday
Tuesday
weekday


Rented Bike Count
Night
Day
label_day_night


Rented Bike Count
January
February
March
April
May
June
July
August
September
October
November
December
Month

Dependent Variable - EDA (contd.)





Insights

1. Temperature and dew point temperature are highly correlated so there is no point of considering both the features. We drop dew point temperature
2. From the month of march in 2018 there is high increase in sales as compared to 2017.
3. We can conclude that the people prefer summer seasons to rent bikes.
4. Winter is the least favourable season
5. Temperature affects the the number of rented bike directly, People prefer to rent bikes in a hotter days.
6. Humidity does not affect the number of rented bikes much
7. If the windspeed is high, people are less likely to rent a bike
8. When the visibility is low, the number of rented bikes is also low
9. When there is a heavy snowfall people are less likely to rent a bike

Preparing Data for Modelling

	January	Date	September	October	March	June	Spring	May	July	Sunday	November	No Holiday	Saturday	Snowfall (cm)	Wind speed (m/s)	Visibility (10m)
0	1	2017-01-12	0	0	0	0	0	0	0	0	0	1	0	0.0	2.2	2000
1	1	2017-01-12	0	0	0	0	0	0	0	0	0	1	0	0.0	0.8	2000
2	1	2017-01-12	0	0	0	0	0	0	0	0	0	1	0	0.0	1.0	2000
3	1	2017-01-12	0	0	0	0	0	0	0	0	0	1	0	0.0	0.9	2000
4	1	2017-01-12	0	0	0	0	0	0	0	0	0	1	0	0.0	2.3	2000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8755	0	2018-11-30	0	0	0	0	0	0	0	0	1	1	0	0.0	2.6	1894
8756	0	2018-11-30	0	0	0	0	0	0	0	0	1	1	0	0.0	2.3	2000
8757	0	2018-11-30	0	0	0	0	0	0	0	0	1	1	0	0.0	0.3	1968
8758	0	2018-11-30	0	0	0	0	0	0	0	0	1	1	0	0.0	1.0	1859
8759	0	2018-11-30	0	0	0	0	0	0	0	0	1	1	0	0.0	1.3	1909

8760 rows × 31 columns

Train set :- (6132, 31)

Test Set :- (2628, 31)

Task :- Linear Regression

Feature Selection and Model Training

Rescaling

- **MinMaxScaler** scales all the data features in the range [0, 1] or else in the range [-1, 1] if there are negative values in the dataset. This scaling compresses all the inliers in the narrow range [0, 0.005].
- In the presence of outliers, StandardScaler does not guarantee balanced feature scales, due to the influence of the outliers while computing the empirical mean and standard deviation. This leads to the shrinkage in the range of the feature values.

RFE Feature Selection

- Recursive Feature Elimination, or RFE for short, is a popular feature selection algorithm.
- RFE is popular because it is easy to configure and use and because it is effective at selecting those features (columns) in a training dataset that are more or most relevant in predicting the target variable.
- There are two important configuration options when using RFE: the choice in the number of features to select and the choice of the algorithm used to help choose features. Both of these hyperparameters can be explored, although the performance of the method is not strongly dependent on these hyperparameters being configured well.

First OLS Regression Result

OLS Regression Results

Dep. Variable:	Rented Bike Count	R-squared:	0.
Model:	OLS	Adj. R-squared:	0.
Method:	Least Squares	F-statistic:	62
Date:	Sat, 31 Jul 2021	Prob (F-statistic):	0
Time:	06:19:22	Log-Likelihood:	427
No. Observations:	6132	AIC:	−85
Df Residuals:	6119	BIC:	−84
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025
const	−0.2173	0.016	−13.670	0.000	−0.248
Hour	0.1842	0.005	33.929	0.000	0.174
Temperature(°C)	0.3621	0.016	23.222	0.000	0.332
Humidity(%)	−0.1927	0.008	−23.022	0.000	−0.209
Rainfall(mm)	−0.5620	0.046	−12.349	0.000	−0.651
Snowfall (cm)	0.0612	0.032	1.911	0.056	−0.002
Spring	−0.0485	0.005	−10.547	0.000	−0.057
Summer	−0.0519	0.006	−8.669	0.000	−0.064
Winter	−0.1093	0.006	−16.885	0.000	−0.122
No Holiday	0.0312	0.007	4.260	0.000	0.017
Yes	0.2671	0.009	29.496	0.000	0.249
June	0.0677	0.006	11.323	0.000	0.056
May	0.0383	0.006	6.413	0.000	0.027

Omnibus:	1024.580	Durbin-Watson:	2.
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1989.
Skew:	1.025	Prob(JB):	0
Kurtosis:	4.893	Cond. No.	5

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corr

Final OLS Regression Result

OLS Regression Results

Dep. Variable:	Rented Bike Count	R-squared:	0.
Model:	OLS	Adj. R-squared:	0.
Method:	Least Squares	F-statistic:	54
Date:	Sat, 31 Jul 2021	Prob (F-statistic):	0
Time:	06:19:23	Log-Likelihood:	362
No. Observations:	6132	AIC:	−72
Df Residuals:	6122	BIC:	−71
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025
const	0.2417	0.011	22.682	0.000	0.221
Hour	0.2064	0.006	35.205	0.000	0.195
Humidity(%)	−0.2356	0.009	−26.097	0.000	−0.253
Snowfall (cm)	0.0387	0.036	1.089	0.276	−0.031
Spring	−0.0390	0.005	−7.772	0.000	−0.049
Summer	0.0625	0.005	12.351	0.000	0.053
Winter	−0.1877	0.005	−37.002	0.000	−0.198
No Holiday	0.0227	0.008	2.798	0.005	0.007
June	0.0449	0.007	6.795	0.000	0.032
May	0.0731	0.006	11.295	0.000	0.060

Omnibus:	644.447	Durbin-Watson:	2.
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1059.
Skew:	0.747	Prob(JB):	8.73e-
Kurtosis:	4.383	Cond. No.	3

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is corr

Variance Inflation Factor

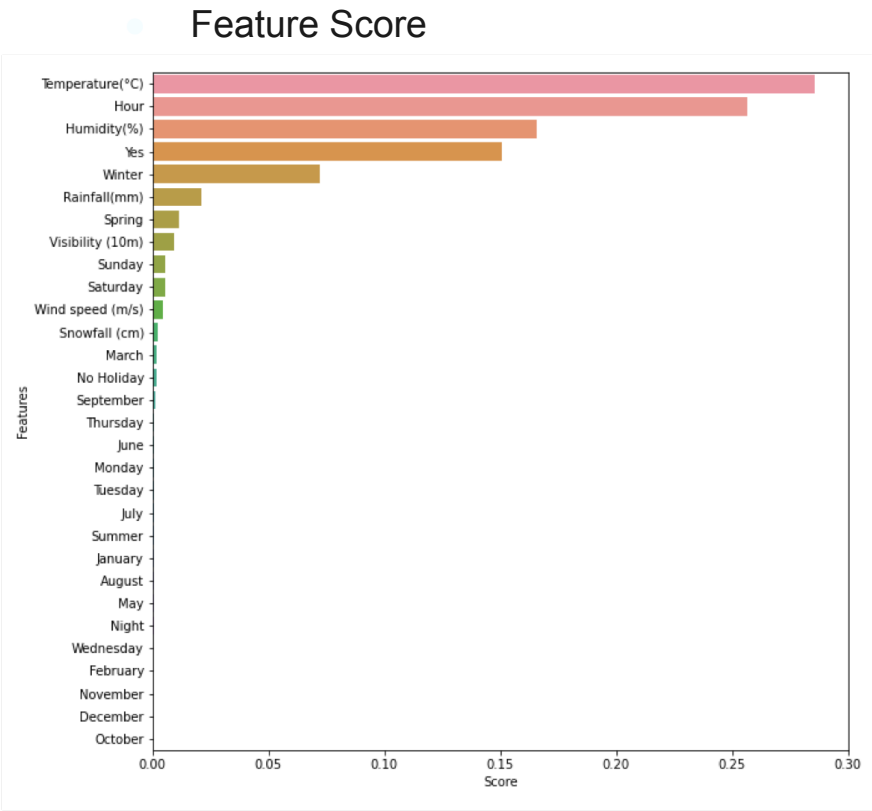
- Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables.
- Mathematically, the VIF for a regression model variable is equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable.
- This ratio is calculated for each independent variable. A high VIF indicates that the associated independent variable is highly collinear with the other variables in the model.

Decision tree Regressor

- Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application.
- It is a tree-structured classifier with three types of nodes.
- The ***Root Node*** is the initial node which represents the entire sample and may get split further into further nodes.
- The ***Interior Nodes*** represent the features of a data set and the branches represent the decision rules.
- Finally, the ***Leaf Nodes*** represent the outcome. This algorithm is very useful for solving decision-related problems.

Decision Tree Regression

- r-square score on test data :0.8627875530521529
- adjusted r-square score on test data :0.8603957033087274
- homogeneity score on test data :0.8536419828090458
- mean squared error score on test data :21.609074707974006



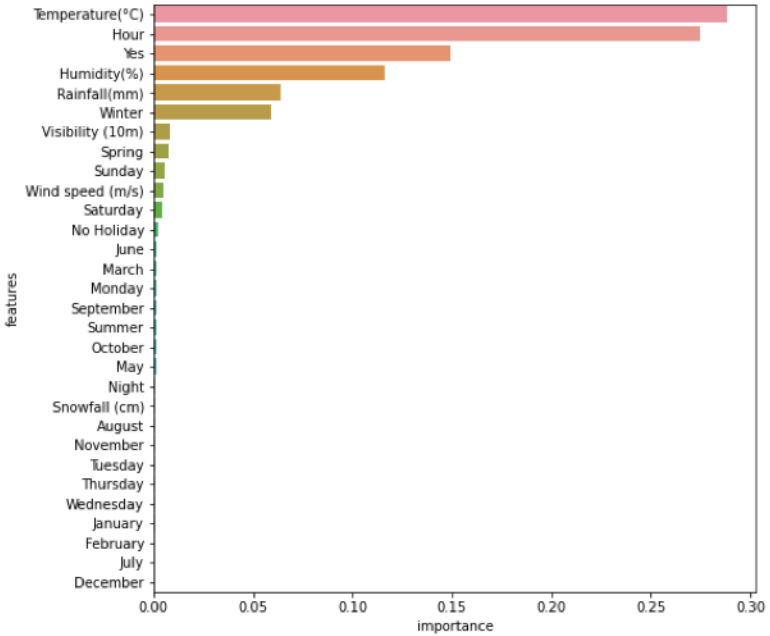
Gradient Boost

- Boosting" in machine learning is a way of combining multiple simple models into a single composite model. This is also why boosting is known as an additive model, since simple models (also known as weak learners) are added one at a time, while keeping existing trees in the model unchanged.
- As we combine more and more simple models, the complete final model becomes a stronger predictor.
- The term "gradient" in "gradient boosting" comes from the fact that the algorithm uses gradient descent to minimize the loss.

Boosting the Model (Gradient Boost)

- Gradient boost score on train data: 0.9539400456545698
- Gradient boost score on test data: 0.9190353708501229
- r-square score on test data : 0.9190353708501229
- adjusted r-square score on test data: 0.9176240176400727
- homogeneity score on test data :0.9999999999999999
- mean squared error score on test data :12.750816408573327

Feature Score



What is SHAP?

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions (see papers for details and citations)

SHAP

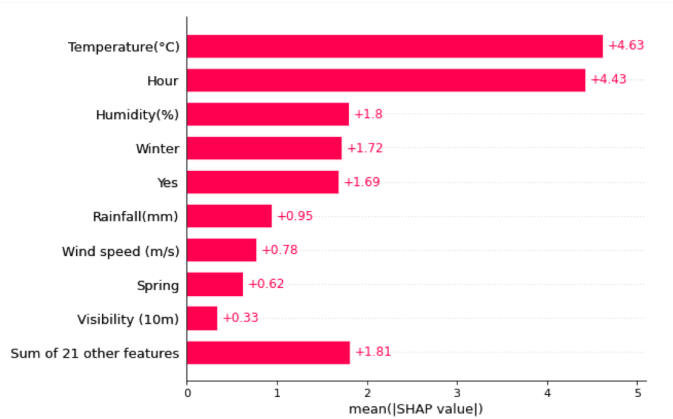
Shap Values

```
.values =
array([[ 3.99978544e-01, -3.33773919e+00,  9.90905310e-01, ...,
         1.99949336e-03,  2.33259085e-02, -1.04568951e-03],
       [ 1.36447099e+01,  8.00476173e+00,  1.86569400e+00, ...,
        -2.32769361e-02,  4.60020180e-02,  1.64377961e-01],
       [-1.00433369e+00,  2.89122244e+00,  8.89110520e-01, ...,
        -3.42972576e-02,  1.73848294e-02,  2.10488437e-01],
       ...,
       [-2.50982157e+00,  5.65487566e+00,  1.17631442e+00, ...,
        -2.42634148e-02,  5.41436470e-02,  3.80207584e-02],
       [-2.42230276e+00, -1.46818153e+00,  6.84433302e-01, ...,
        -6.98675104e-02, -1.66111731e-03, -9.99922059e-02],
       [ 3.32131854e+00, -4.79761778e+00, -2.08980977e+00, ...,
         5.99490264e-03,  6.17217552e-05,  2.16361373e-02]])

.base_values =
array([23.95706901, 23.95706901, 23.95706901, ..., 23.95706901,
       23.95706901, 23.95706901])

.data =
array([[15. ,  8.2, 62. , ...,  0. ,  0. ,  0. ],
       [18. , 28.4, 57. , ...,  0. ,  0. ,  0. ],
       [11. , 29.9, 57. , ...,  0. ,  0. ,  0. ],
       ...,
       [11. , 25.5, 57. , ...,  0. ,  0. ,  0. ],
       [ 0. ,  8.3, 59. , ...,  0. ,  0. ,  1. ],
       [20. ,  7.1, 83. , ...,  0. ,  0. ,  0. ]])
```

Shap Values Bar-plot



Conclusion

- In the problem statement it is mentioned that the company started in the year 2017 that's why the sales are low in the 2017 and higher in 2018.
- The Hour of the day affects the bike sales which is because of the probably office hours
- The `temperature` `humidity` `windspeed` also plays a crucial role in the the sales of rented bikes
- higher temperature tends to attract more customer that is why we see peak in sales in the summer seasons
- After trying combinations of features with linear regression the model underfit. It seemed obvious because data is spread too much. It didn't seem practical to fit a line.
- With Decision tree we reached at the model R squared value of 0.86. We only fitted with minimum number of leaf hyper-parameter. With default parameters it overfitted and reached R-squared at 1 with train dataset but 0.86 with test.
- Lastly Gradient boost gives the best results with R squared value for train data 0.95 and test data 0.92
- The **Feature_importance** is almost the same in both the tree based models. Gradient boost fine-tunes with error of the prior trees this is why it gets better accuracies.