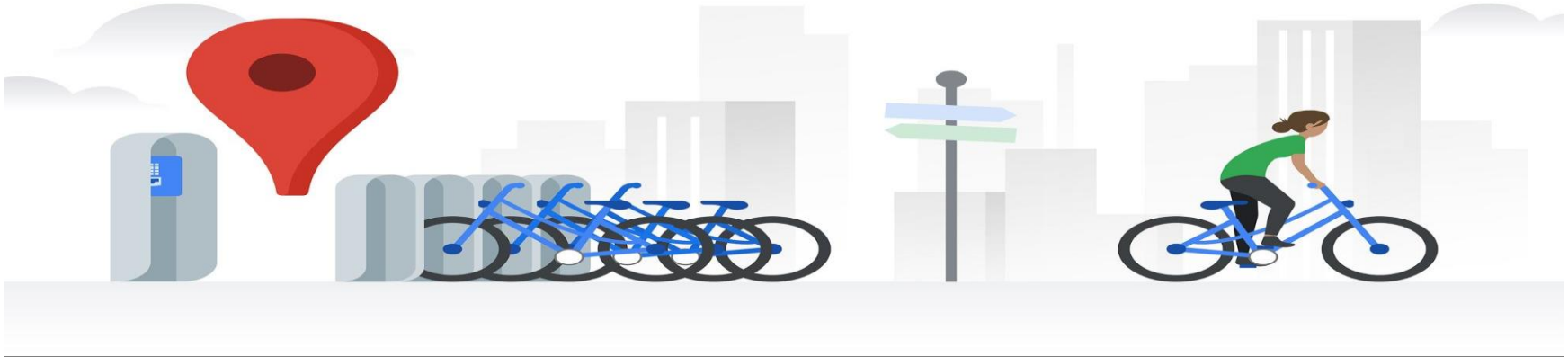
Capstone Project

Bike Sharing Demand Prediction

-Kritisha Panda

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.



Primary objective:

- To build a statistical model to predict the number of bikes that can be rented each hour given the factors which can affect the outcome

Secondary objectives:

- To learn how data is represented is datasets
- To understand the pre-processing of dataset
- To study and compare various ML models and conclude the best predictor model
- To solve a real life business problem

The Dataset

- The dataset has 8760 rows and 14 features
- All the features are in correct data type except the ‘Date’ feature
- The data set has no null, missing or duplicate values
- The dataset shows hourly rental data for 1 year (1 December 2017 to 31 November 2018).We take this as a single year

	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

Descriptive Statistics:

- Measures of Frequency :- Count, Percent, Frequency.
- Measures of Central Tendency :- Mean, Median, and Mode.
- Measures of Dispersion or Variation:- Range(min,max),Variance, Standard Deviation
- Measures of Position :- Percentile Ranks, Quartile Ranks

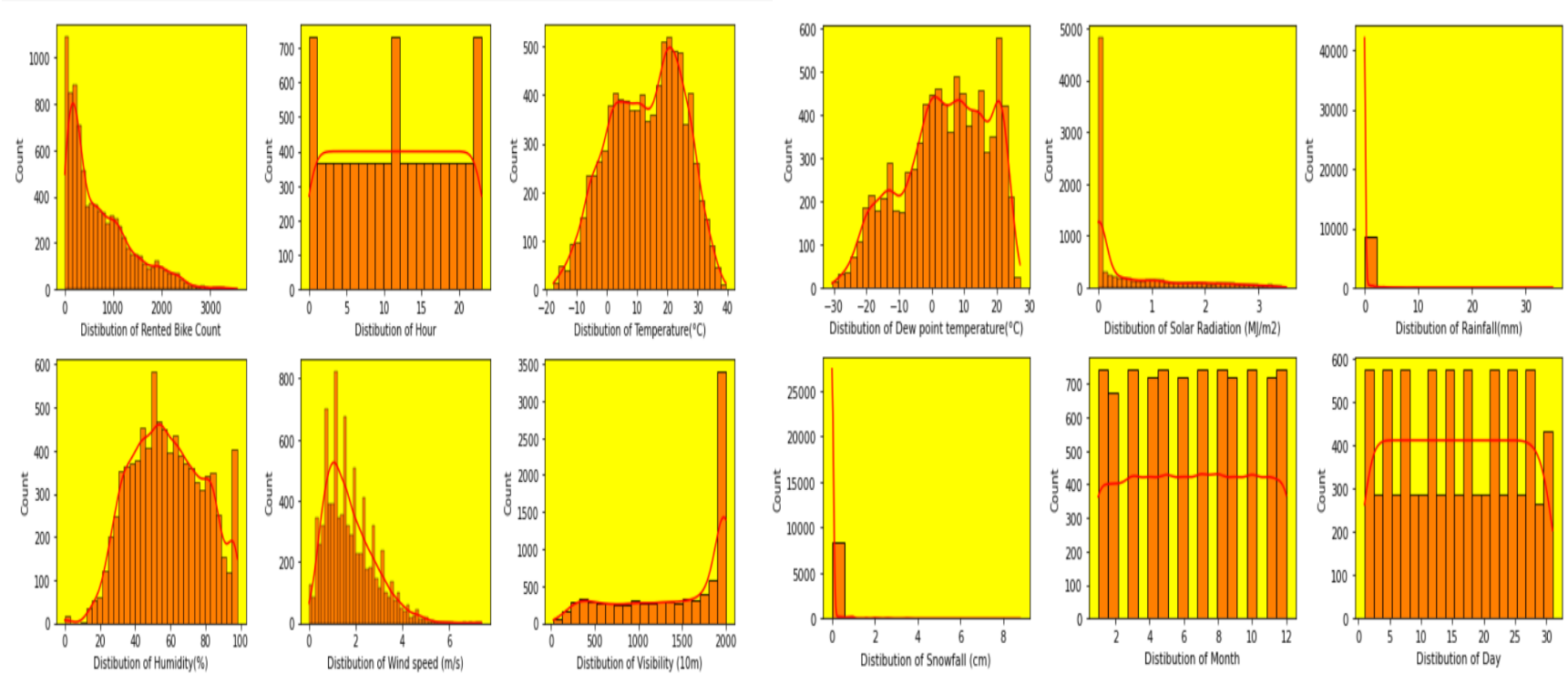
	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)
count	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000
mean	704.602055	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

Treating the ‘Date’ feature

- Since the ‘Date’ column is not in the appropriate format, we change the dtype to datetime64[ns] and extract the Month, Day and Year to three columns.

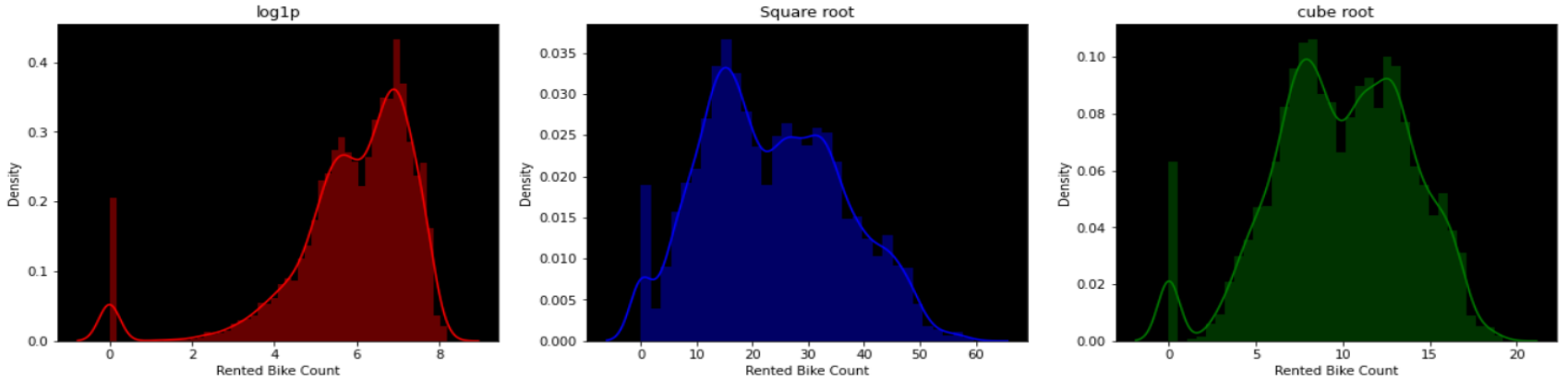
```
[12] #We need to convert date column to DateTime
     bikeData['Date'] = pd.to_datetime(bikeData['Date'], format='%d/%m/%Y')

     #Extracting month,day and year from the date
     bikeData['Month']=bikeData['Date'].dt.month
     bikeData['Day']=bikeData['Date'].dt.day
     bikeData['Year']=bikeData['Date'].dt.year
```

Distribution of numerical variables

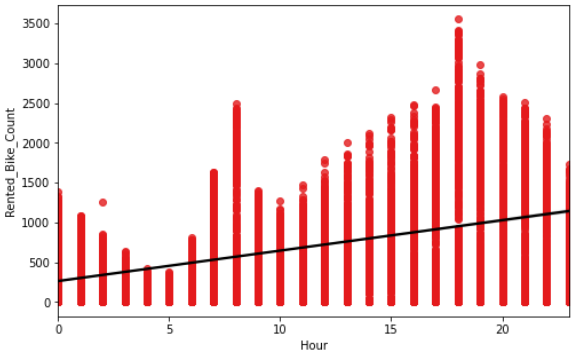


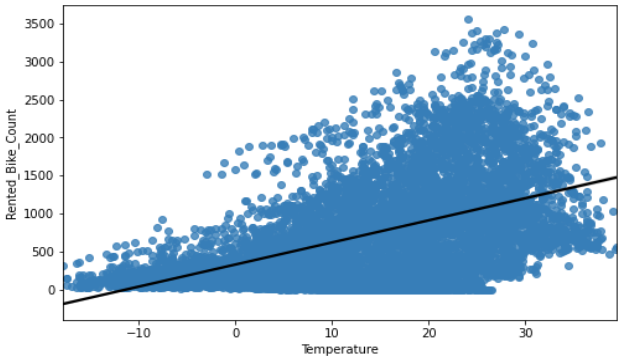
Feature Transformation

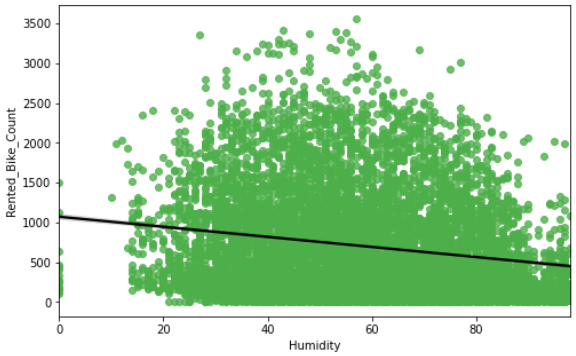
- Most of the Machine learning models like linear models, naive Bayes generally assume that the data is normally distributed (symmetrical shape), But the data we got from the real world scenario is not normally distributed, indeed it's skewed (left-skewed or right-skewed).

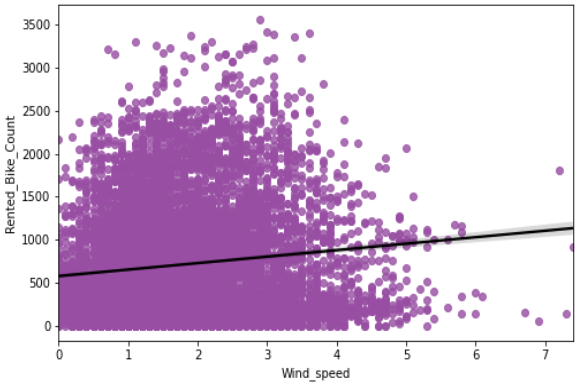


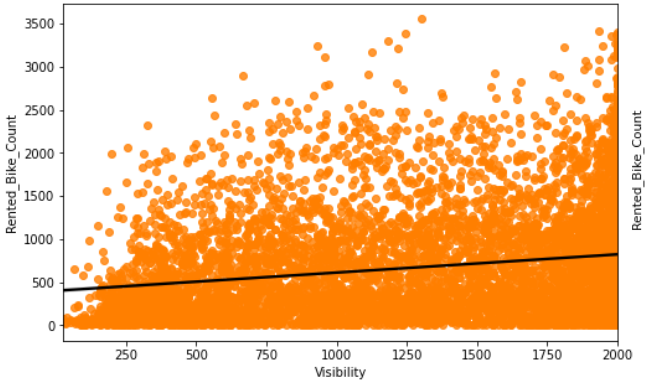
Plotting the numerical features and fitting a LR line:

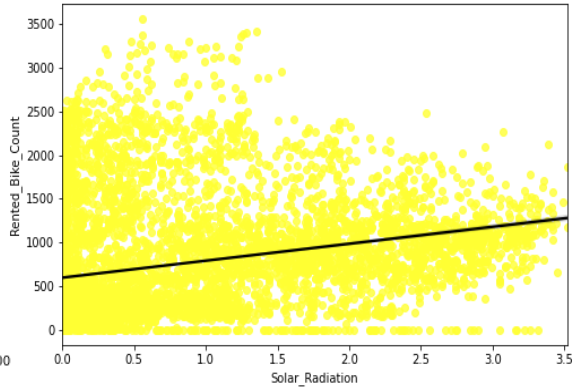




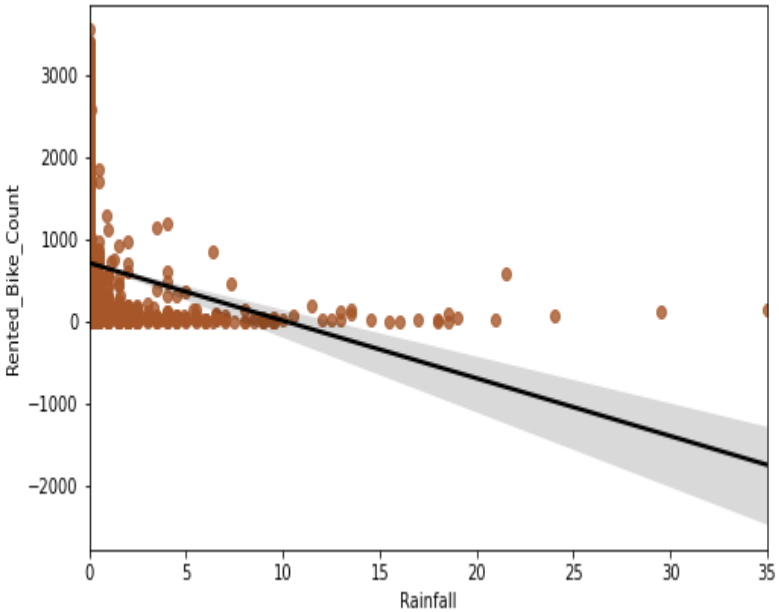


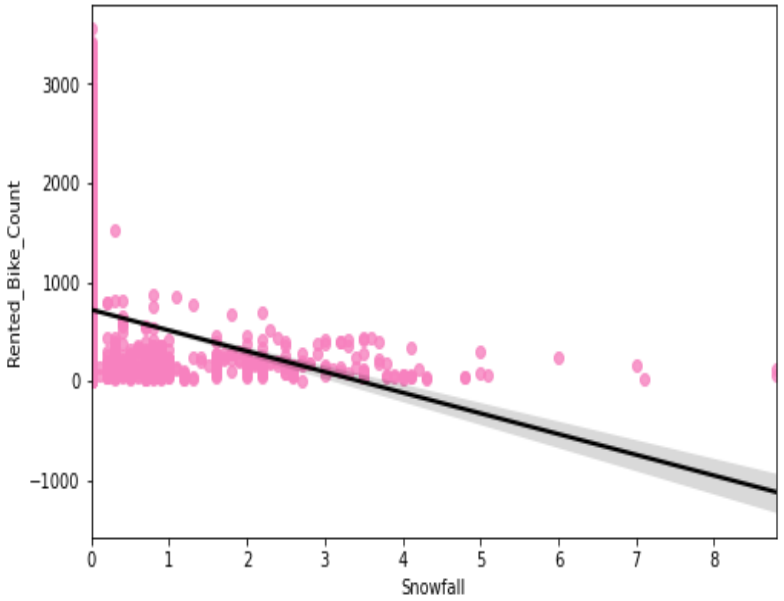






- As expected, rainfall and snowfall have a negative correlation with Rented Bike Count
- On the days of heavy rainfall or snowfall people prefer to stay indoors, hence the demand for bikes is very low

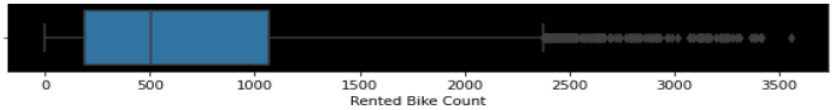


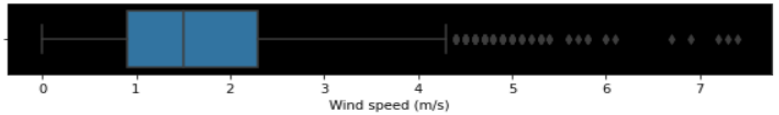


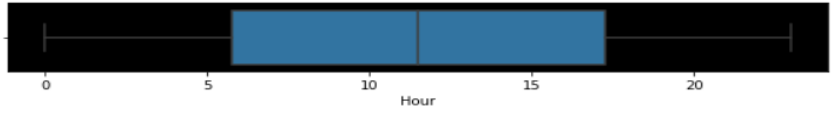
Outlier detection

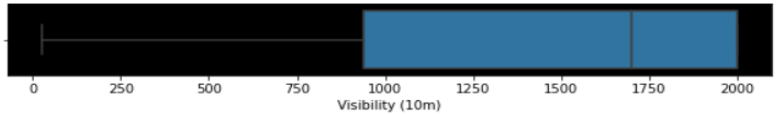
There are two main reasons why giving outliers special attention is a necessary aspect of the data analytics process:

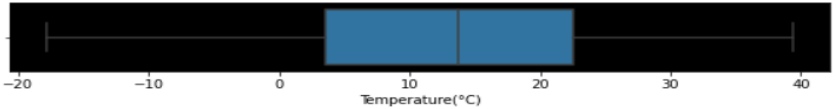
- Outliers may have a negative effect on the result of an analysis
- Outliers—or their behavior—may be the information that a data analyst requires

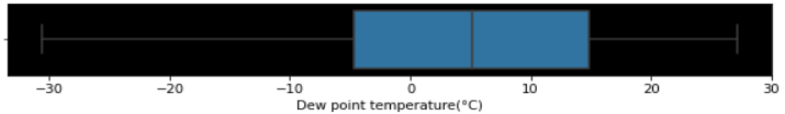


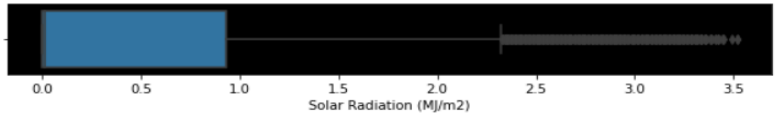




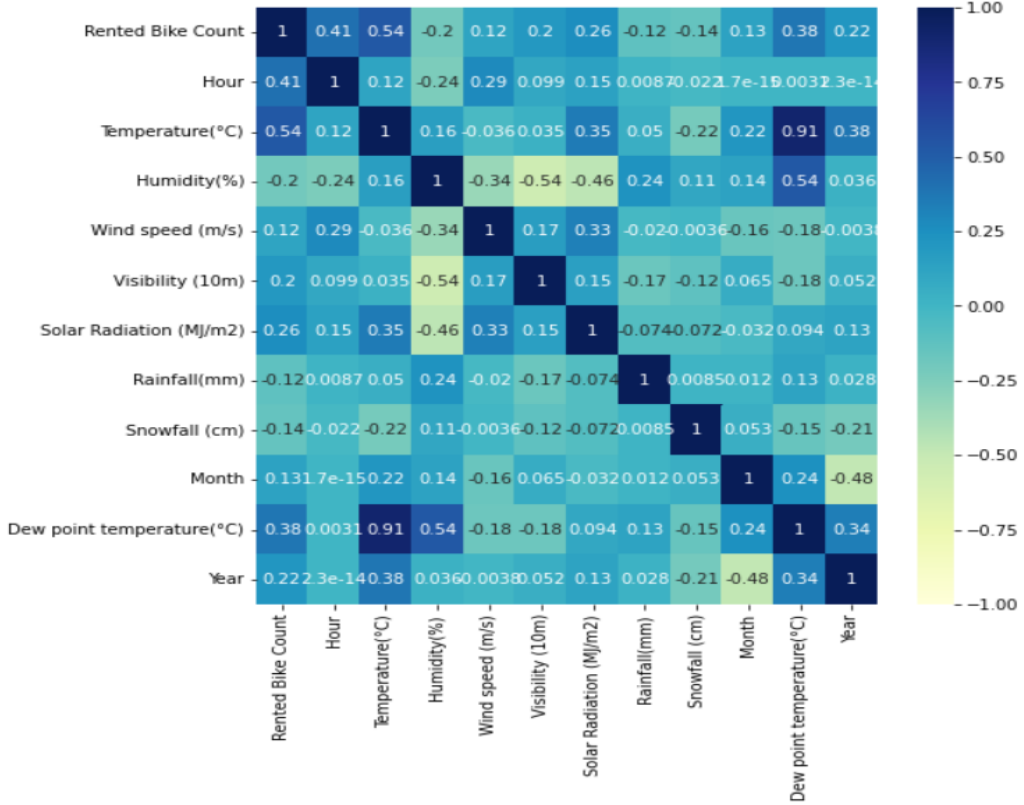








Correlation Heatmap:



- A correlation heatmap is a visual representation of collinearity present between the features
- Temperature and Dew point temperature are strongly correlated
- Temperature and Hour share a strong correlation with the dependent variable Rented Bike Count

Variance Inflation Factor

- VIF is an indicator of how much the features are correlated. A high VIF indicates a higher correlation.
- In our dataset, Dew point temperature and Year features have the highest VIFs hence they're dropped.

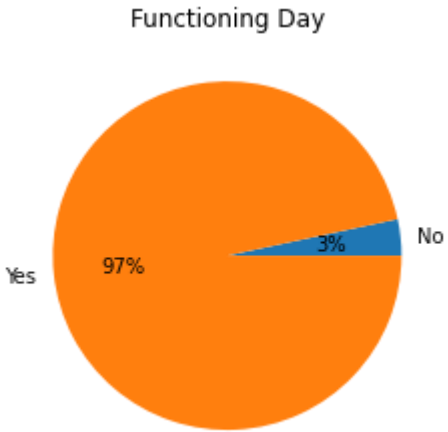
VIF before dropping features

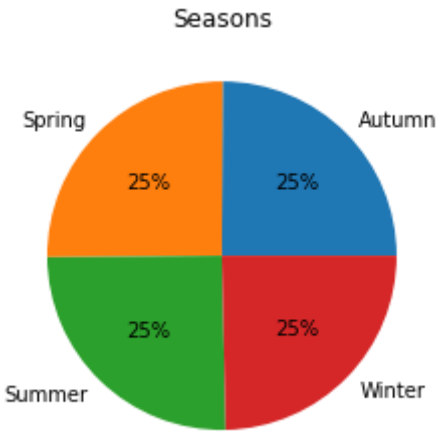
	feature	VIF
0	Hour	4.458880
1	Temperature(°C)	188.666573
2	Humidity(%)	187.533688
3	Wind speed (m/s)	4.890096
4	Visibility (10m)	10.788995
5	Dew point temperature(°C)	126.954261
6	Solar Radiation (MJ/m2)	2.904971
7	Rainfall(mm)	1.103386
8	Snowfall (cm)	1.155412
9	Month	5.108772
10	Year	407.025112
11	Day	4.379818

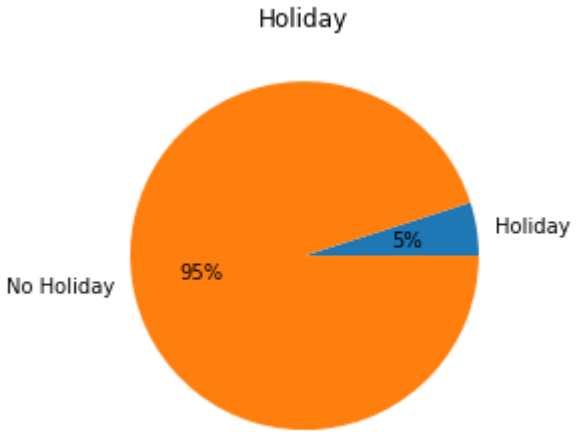
VIF after dropping features

	feature	VIF
0	Hour	3.997641
1	Temperature(°C)	3.288024
2	Humidity(%)	6.802299
3	Wind speed (m/s)	4.667341
4	Visibility (10m)	5.471035
5	Solar Radiation (MJ/m2)	2.275006
6	Rainfall(mm)	1.080689
7	Snowfall (cm)	1.139759
8	Month	5.027060
9	Day	3.776455

Analysing categorical variable







One Hot Encoding

- One Hot Encoding is used to convert categorical features to numerical for the ease of processing and modelling.

```
bikeData_copy=bikeData.copy()
#Dummification of categorical features
columns_=['Seasons','Holiday','Functioning Day']
dummy_categorical_features = pd.get_dummies(bikeData[columns_],drop_first=True)
dummy_categorical_features.head(3)
```

	Seasons_Spring	Seasons_Summer	Seasons_Winter	Holiday_No Holiday	Functioning Day_Yes
0	0	0	1	1	1
1	0	0	1	1	1
2	0	0	1	1	1

Final data for modelling:-

	Rented Bike Count	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Month	Day	Seasons_Spring	Seasons_Summer	Seasons_Winter	Holiday_No Holiday	Functioning Day_Yes	2ndhalf	3rdhalf	4thhalf
0	254	-5.2	37	2.2	2000	0.00	0.0	0.0	12	1	0	0	1	1	1	0	0	0
1	204	-5.5	38	0.8	2000	0.00	0.0	0.0	12	1	0	0	1	1	1	0	0	0
2	173	-6.0	39	1.0	2000	0.00	0.0	0.0	12	1	0	0	1	1	1	0	0	0
3	107	-6.2	40	0.9	2000	0.00	0.0	0.0	12	1	0	0	1	1	1	0	0	0
4	78	-6.0	36	2.3	2000	0.00	0.0	0.0	12	1	0	0	1	1	1	0	0	0
5	100	-6.4	37	1.5	2000	0.00	0.0	0.0	12	1	0	0	1	1	1	0	0	0
6	181	-6.6	35	1.3	2000	0.00	0.0	0.0	12	1	0	0	1	1	1	1	0	0
7	460	-7.4	38	0.9	2000	0.00	0.0	0.0	12	1	0	0	1	1	1	1	0	0
8	930	-7.6	37	1.1	2000	0.01	0.0	0.0	12	1	0	0	1	1	1	1	0	0
9	490	-6.5	27	0.5	1928	0.23	0.0	0.0	12	1	0	0	1	1	1	1	0	0

Models Implemented:

- Linear Regression (Train set=81%,Test set=80%)
- Decision Tree (Train set=99%,Test set=96%)
- Random Forest (Train set=99%,Test set=98%)
- XG Boost (Train set=99%,Test set=97%)
- Bagging (Train set=95%,Test set=94%)
- Stacking (Train set=99%,Test set=98%)

Evaluation Metrics used:

- R2 Score
- MSE (Mean Squared Error)
- RMSE (Root Mean Squared Error)
- Cross Validation Accuracy
- Cross Validation Standard Deviation

Model Comparison Chart:

	R-Square	MSE	RMSE	CV Accuracy	CV std
Stacking Regressor	0.98	2.4	1.5	0.98	0.0008
Random forest Regression	0.98	2.6	1.6	0.98	0.0014
Bagging Regressor	0.98	3.0	1.7	0.98	0.00083
Decision Tree Regression	0.97	3.9	2.0	0.97	0.0022
XG Boost	0.95	7.1	2.7	0.95	0.0018
Linear Regression	0.81	2.7e+01	5.2	0.81	0.0063

Maximum Score in each Column

	R-Square	MSE	RMSE	CV Accuracy	CV std
Stacking Regressor	0.98	2.4	1.5	0.98	0.0008
Random forest Regression	0.98	2.6	1.6	0.98	0.0014
Bagging Regressor	0.98	3.0	1.7	0.98	0.00083
Decision Tree Regression	0.97	3.9	2.0	0.97	0.0022
XG Boost	0.95	7.1	2.7	0.95	0.0018
Linear Regression	0.81	2.7e+01	5.2	0.81	0.0063

Conclusion:

- Overfitting was checked and models were found to perform well on both train and test sets
- Hyperparameters were chosen after tuning wherever it was necessary
- Stacking Regressor is the best model to solve our business problem
- RMPSE of stacking regressor is 12% which is quite acceptable
- Accuracy of stacking regressor is 98%
- RMSE is 1.5 for stacking regressor which is minimum of all the models