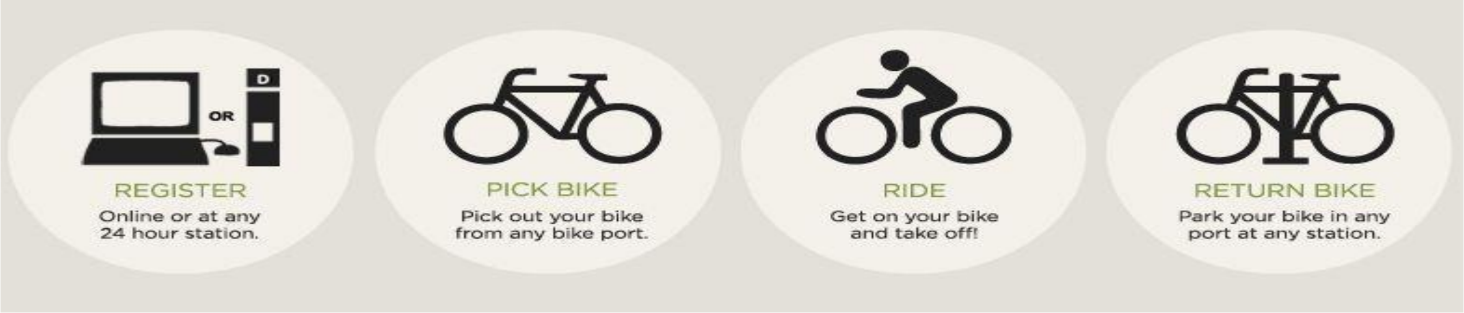
Capstone Project - 2

Bike Sharing Demand Prediction

Supervised ML Regression Model

Vineeta Singh
Tushar Gupta



REGISTER
Online or at any 24 hour station.

PICK BIKE
Pick out your bike from any bike port.

RIDE
Get on your bike and take off!

RETURN BIKE
Park your bike in any port at any station.

Concept of Bike Sharing

The basis concept of bike sharing is sustainable transportation. Bike sharing programs have expanded rapidly in recent years as cities search for ways to increase bike usage, increase mobility choices and reduce adverse environmental impacts.

It is a convenient, environmentally friendly way to get around town, but has flaws too. There are certain conditions that must be met for proper implementation of this program. Bike sharing Demand is affected by multiple factors including temperature, resources and many more.

Let’s predict Bike Sharing Demand in Seoul

1. 1. Defining problem statement

2. 2. Data Summary

3. 2. EDA and feature Engineering

4. 3. Feature selection

5. 4. Preparing Dataset for modelling

6. 5. Implementing Regression models

7. 7. Model Validation & Selection

8. 6. Challenges

9. 7. Conclusions

10. 8. Q&A

Problem statements

- Prediction of bike count required at each hour.
- What factors affect bike sharing count ?
- Reduce waiting time of public.

Need of machine learning to predict bike demand:

The idea of this project is to create a predictive model that identifies upcoming trends in bike sharing demand.

It is crucial to keep in mind that machine learning can only be used to memorize patterns that are present in the training data, so we can only recognize what we have seen before. When using Machine Learning we are making the assumption that the future will behave like the past, and this isn’t always true.

Data pipeline

- *Preparing the data_1* : In this first part, we’ve done data inspection where we checked null or missing values and did multiple operations to make sure our dataset is up to the mark.

- *Preparing the data_2* : Checked all the features, extracted data feature to get more data. Now as dataset is ready, we moved to the next step.

- *EDA* : In this part, in order to see the trends we did some exploratory data analysis on the features and checked distribution of data points and correlation with dependent variable.

- *Create model* : After Data preparation, We feed our machine learning model with numerical data. This process is called model building. We start with simple algorithm but model complexity increases for better performance.

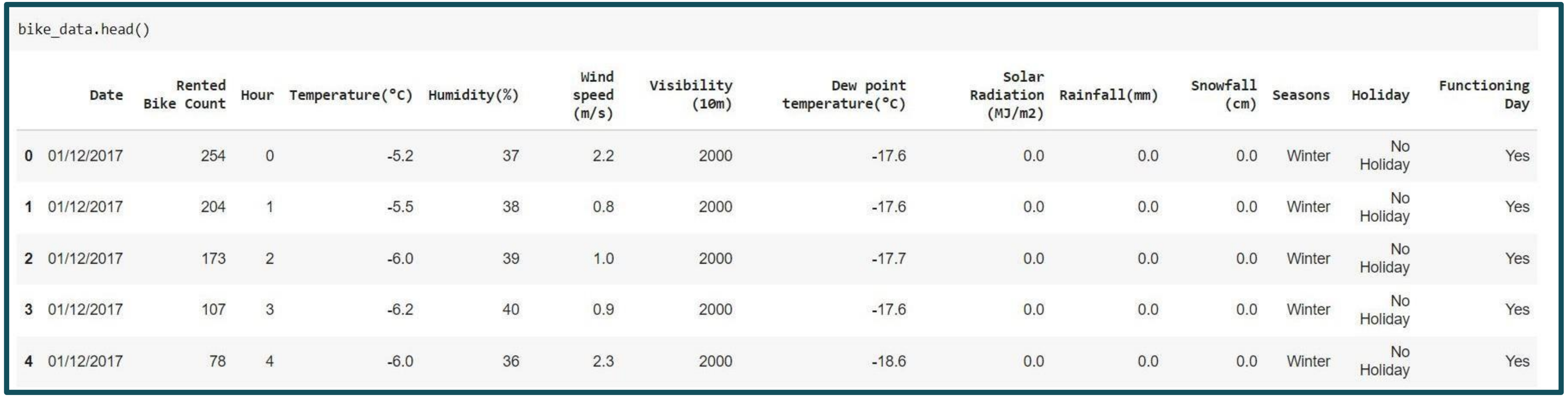
- *Choose a Measure of Success* : After applying every model, we measure it’s accuracy by some evaluation matrices.

Data summary

- 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)

Basic Data Exploration

- This Dataset has 8760 Row and 14 Columns.
- There are 3 categorical features ‘Seasons’, ‘Holiday’, & ‘functioning Day’.
- From datetime string we extract lots of features like day,year and month
- Dataset contains no null values
- No missing or duplicates values.

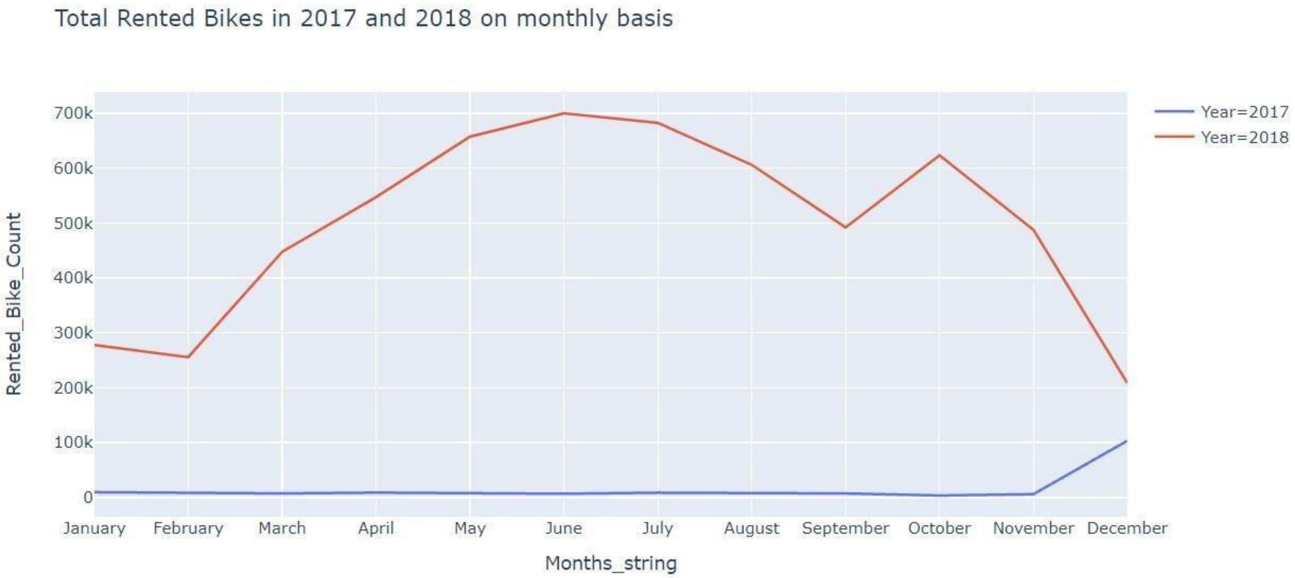


```
bike_data.head()
```

	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

EDA

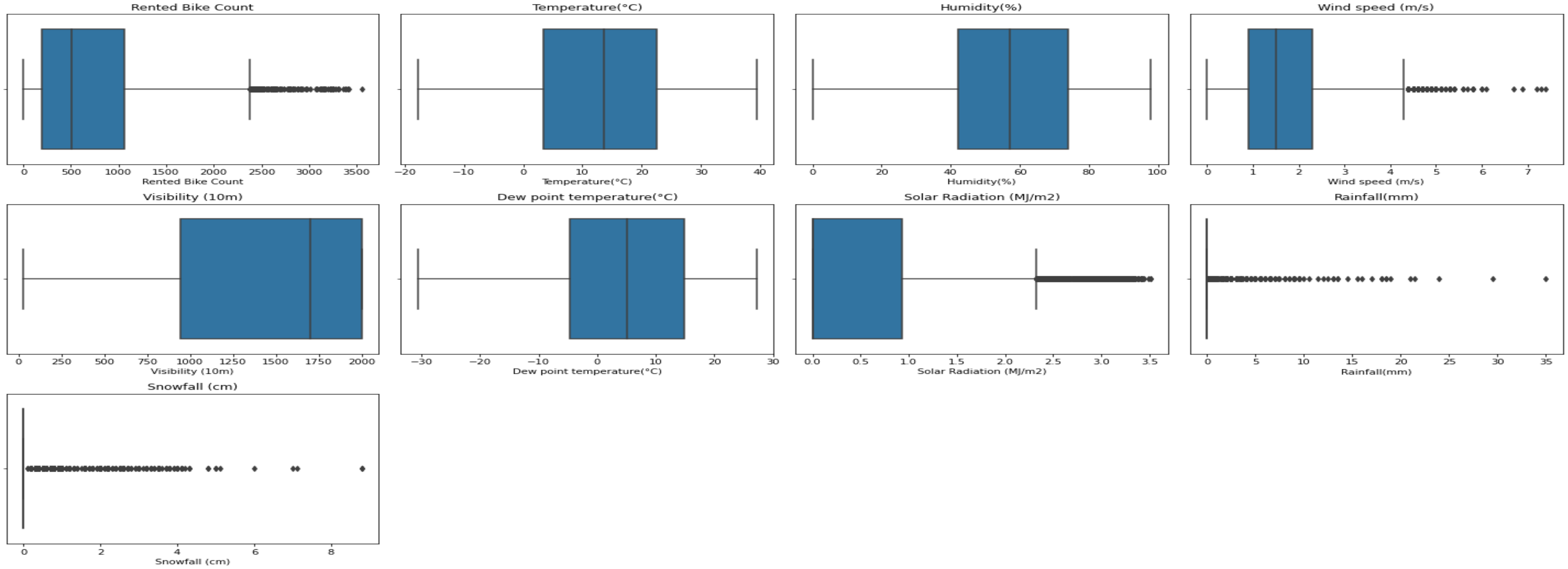
Rented bike count on Different months in 2017 and 2018



- Rented bike count was very low in 2017.
- In 2018, There is a sharp increase in Rented bike count.

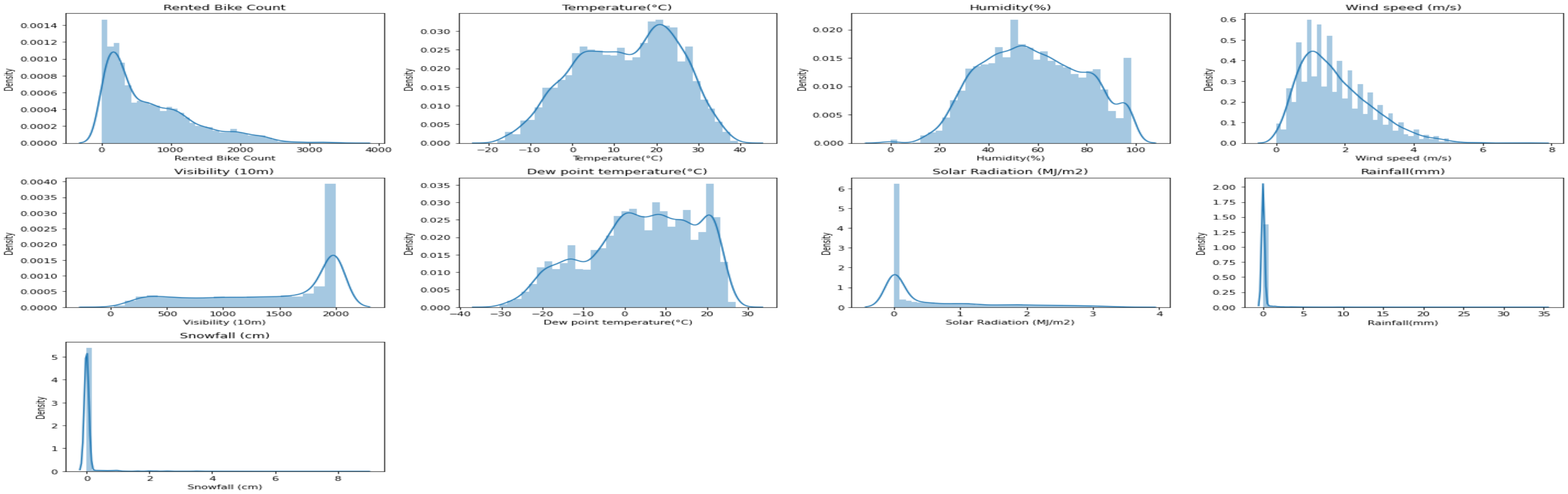
EDA

Box plot for numerical features



- Box plot shows some outliers present in data.

EDA Skewness

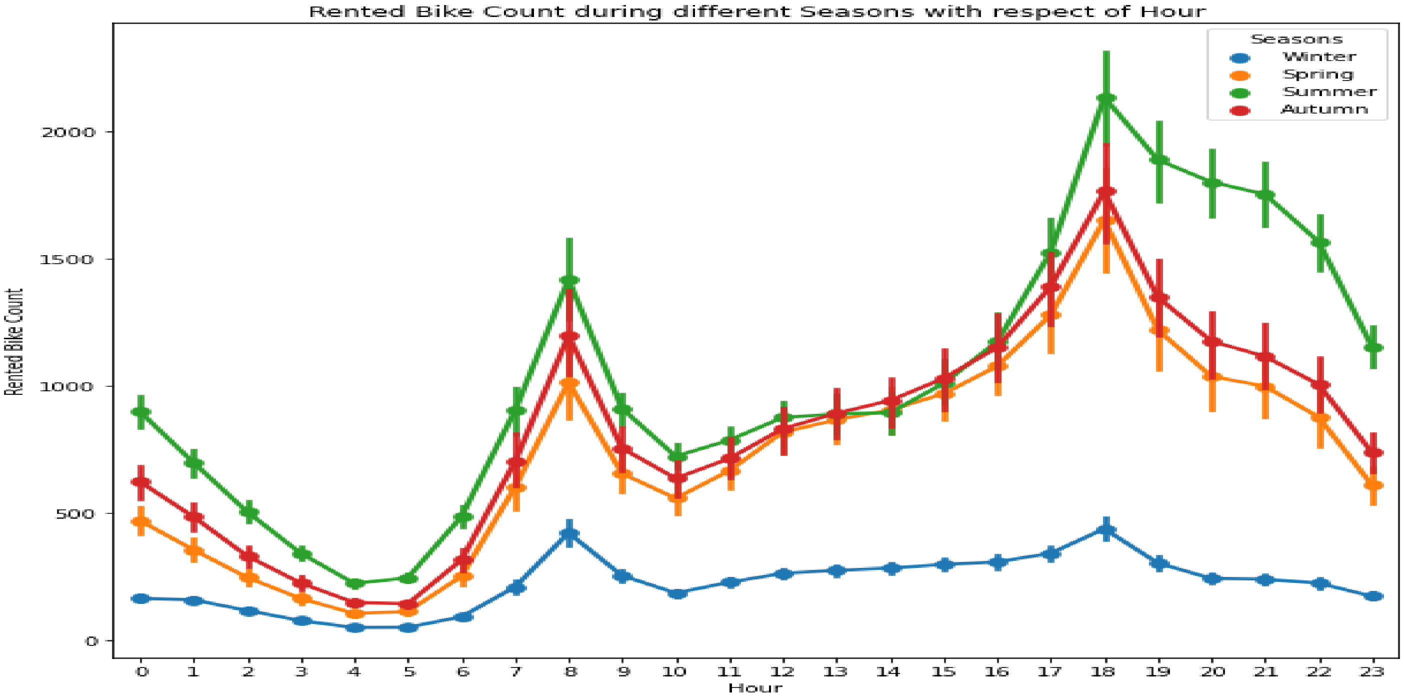


In this plots we observe that some of our columns is right skewed and some are left skewed we have to remember this things when we apply algorithms

Right skewed columns are

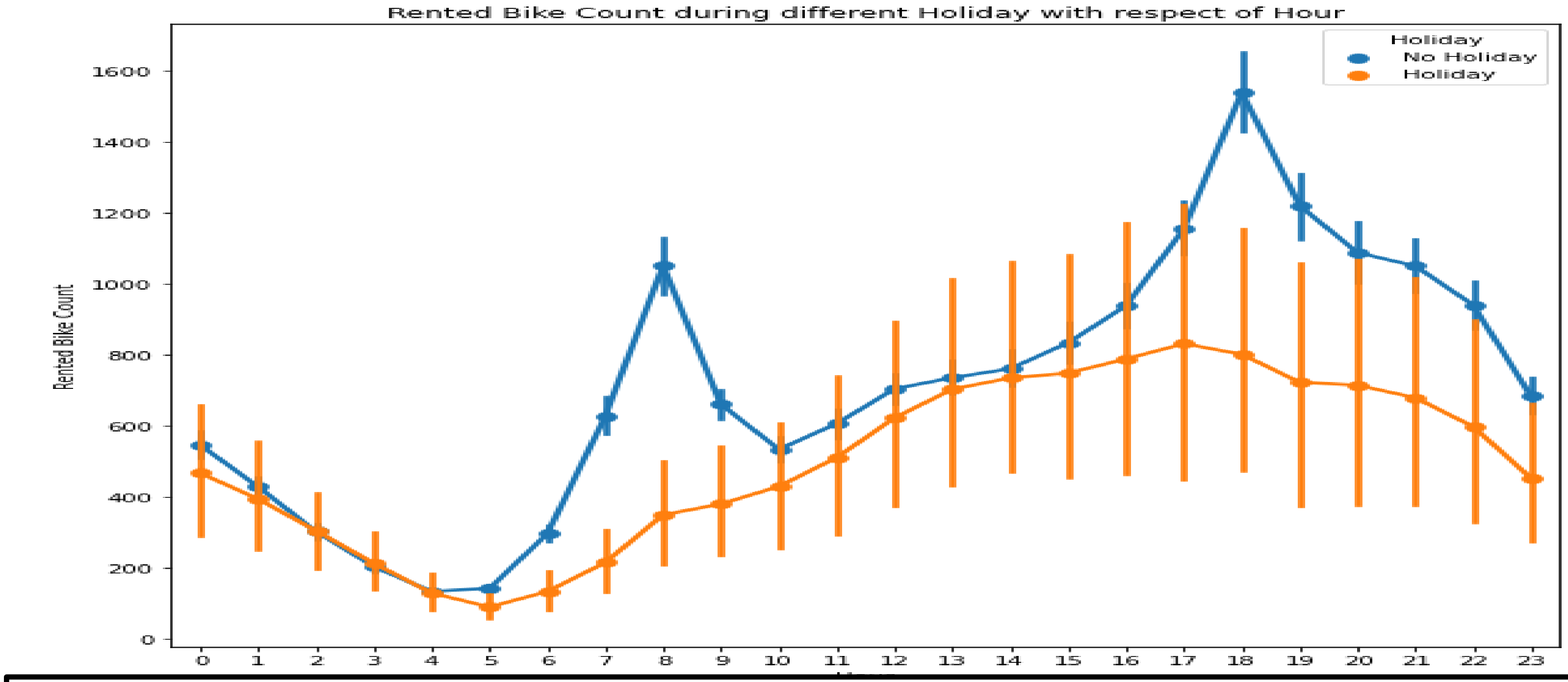
Rented Bike Count (Its also our Dependent variable), Wind speed (m/s), Solar Radiation (MJ/m2), Rainfall(mm), Snowfall (cm)

EDA Bike count Vs Different Seasons



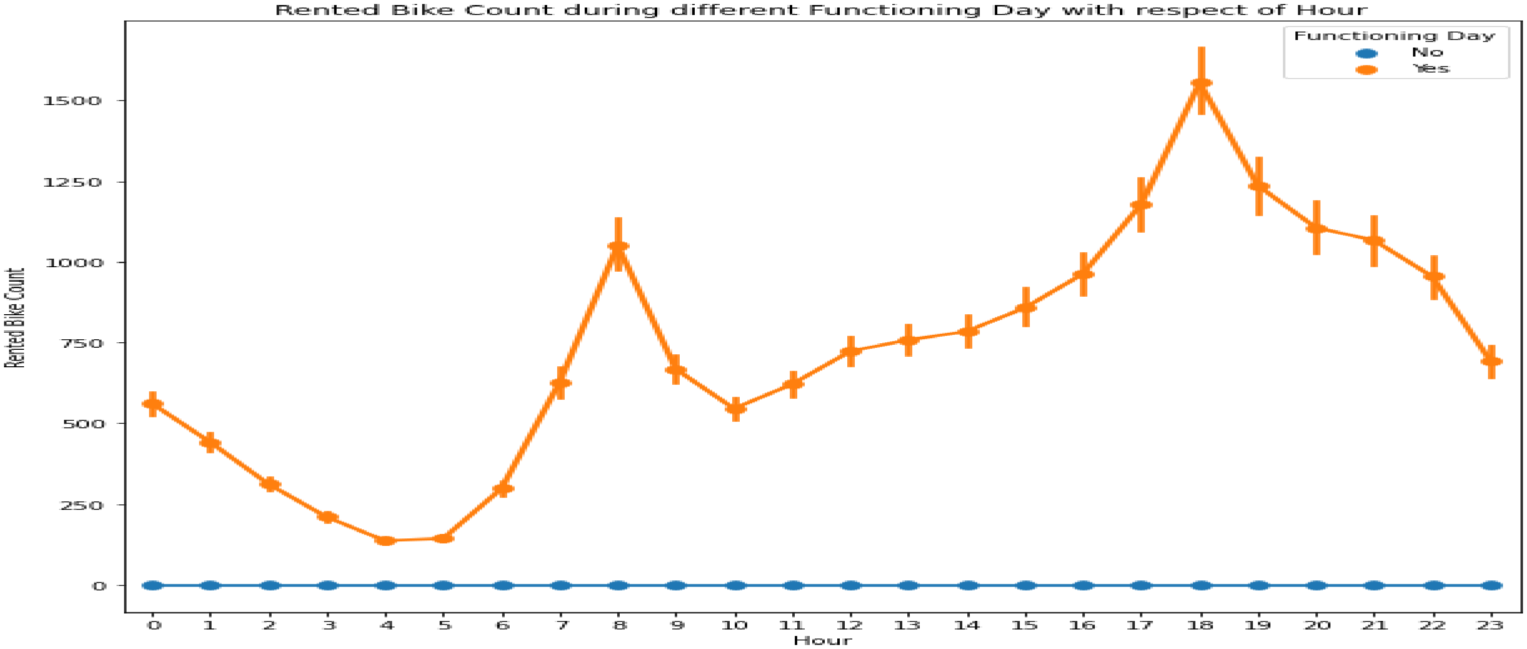
In the season column, we are able to understand that the demand is low in the winter season and highest in summer

EDA Bike count Vs Holiday



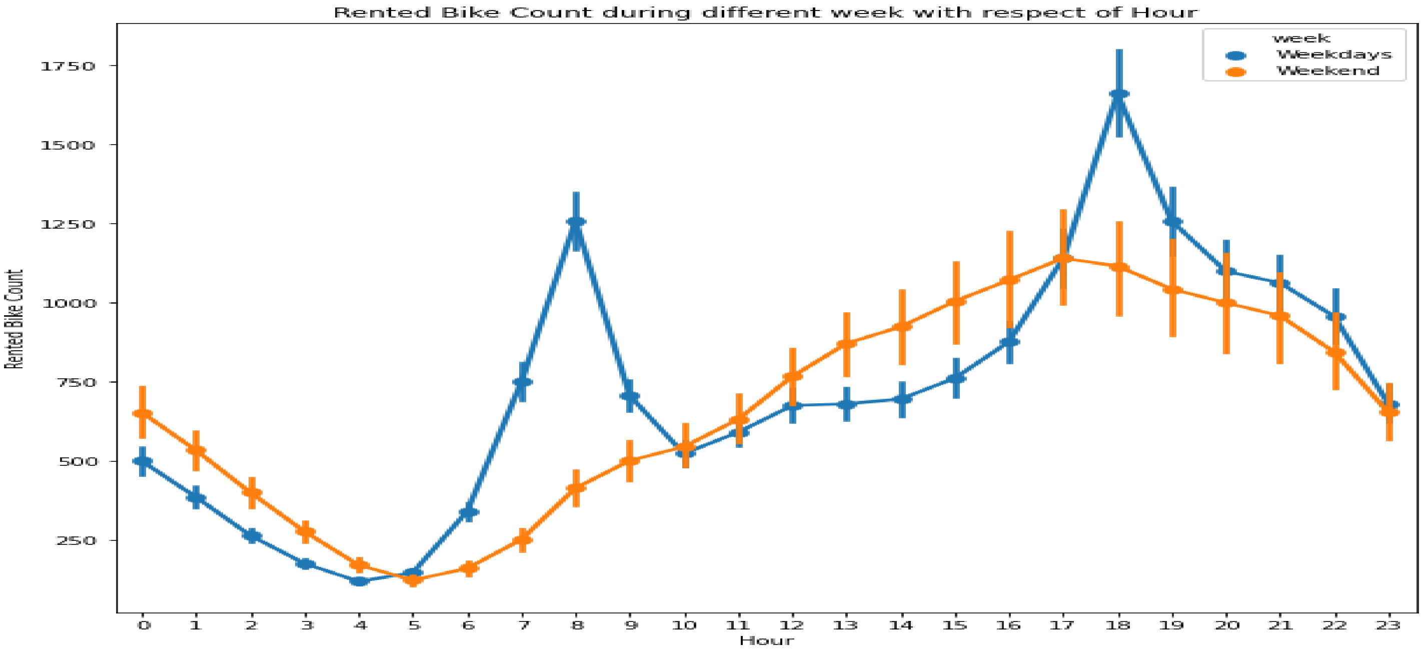
In the Holiday column, The demand is low during holidays, but in no holidays the demand is high, it may be because people use bikes to go to their work.

EDA Bike count Vs Functioning Day



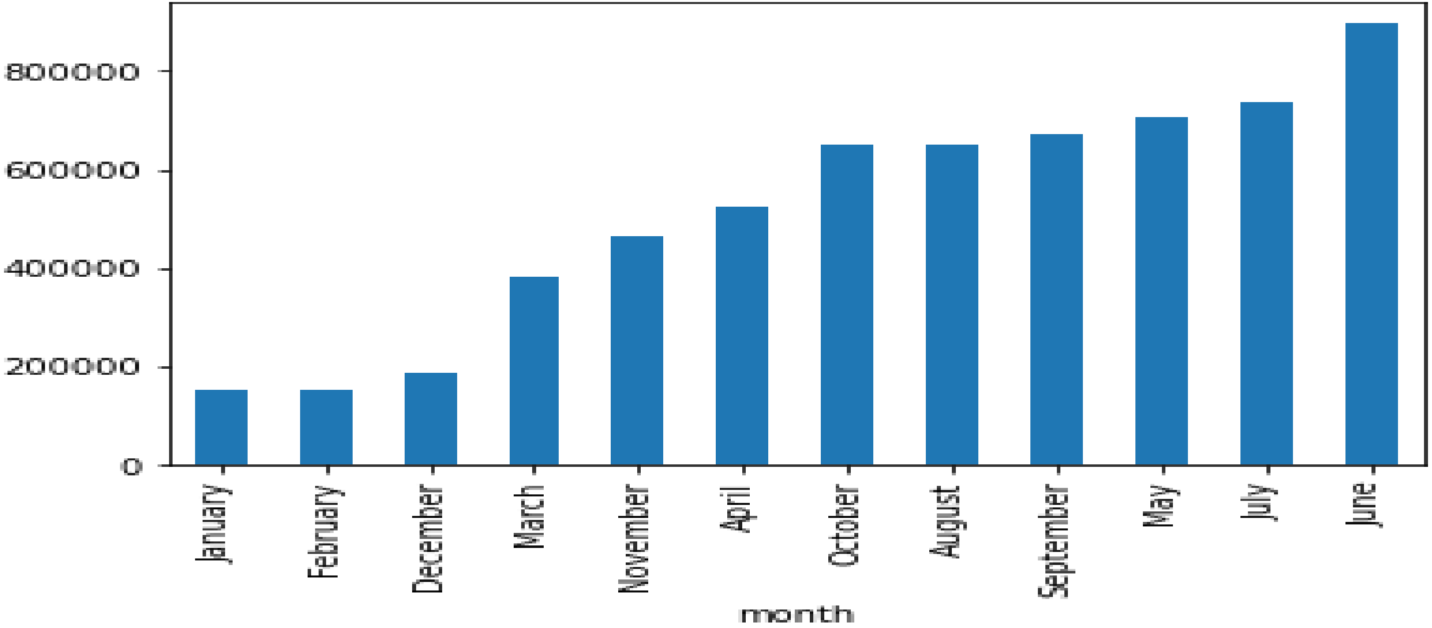
In the Functioning Day column, If there is no Functioning Day then there is no demand

EDA Bike count Vs Week



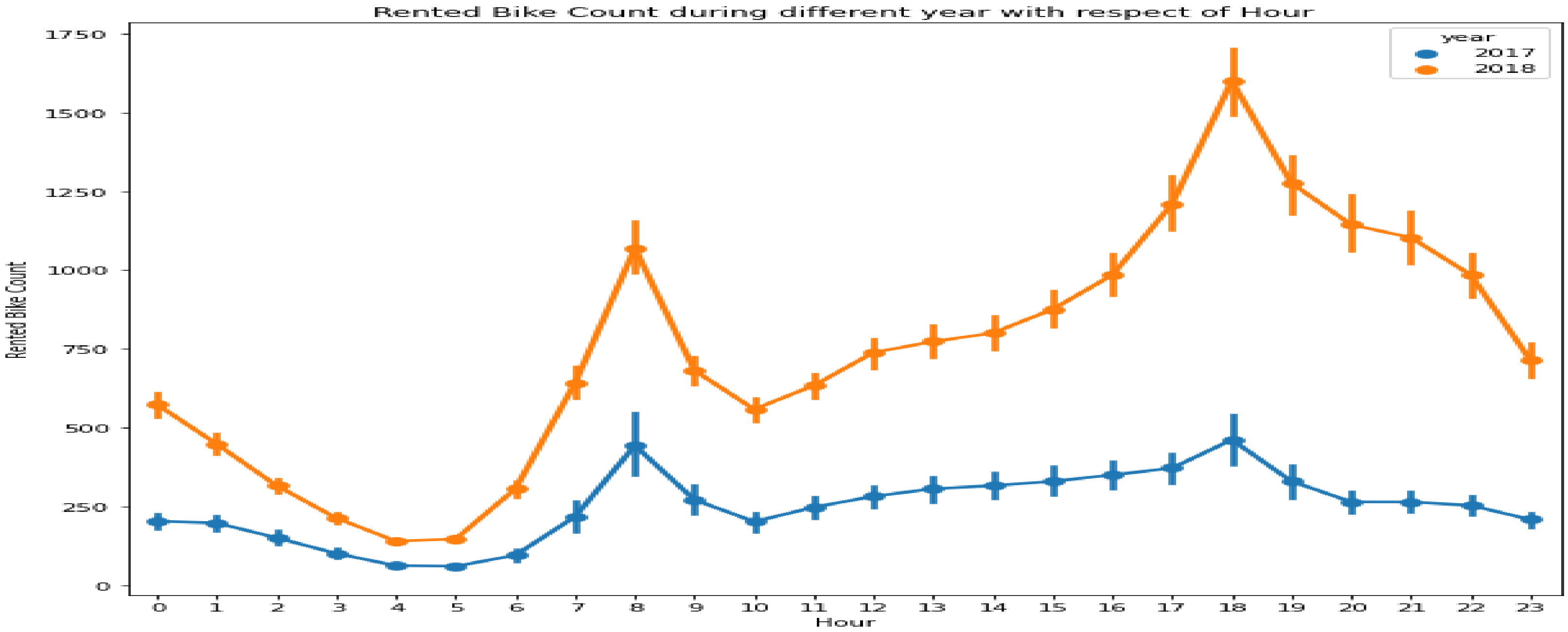
In the week column, We can observe from this column that the pattern of weekdays a nd weekends is different. In the weekend the demand becomes high in the afternoon as people loves to travel during this hour . While the demand during office timing is high in weekdays

EDA Bike count Vs Different Month



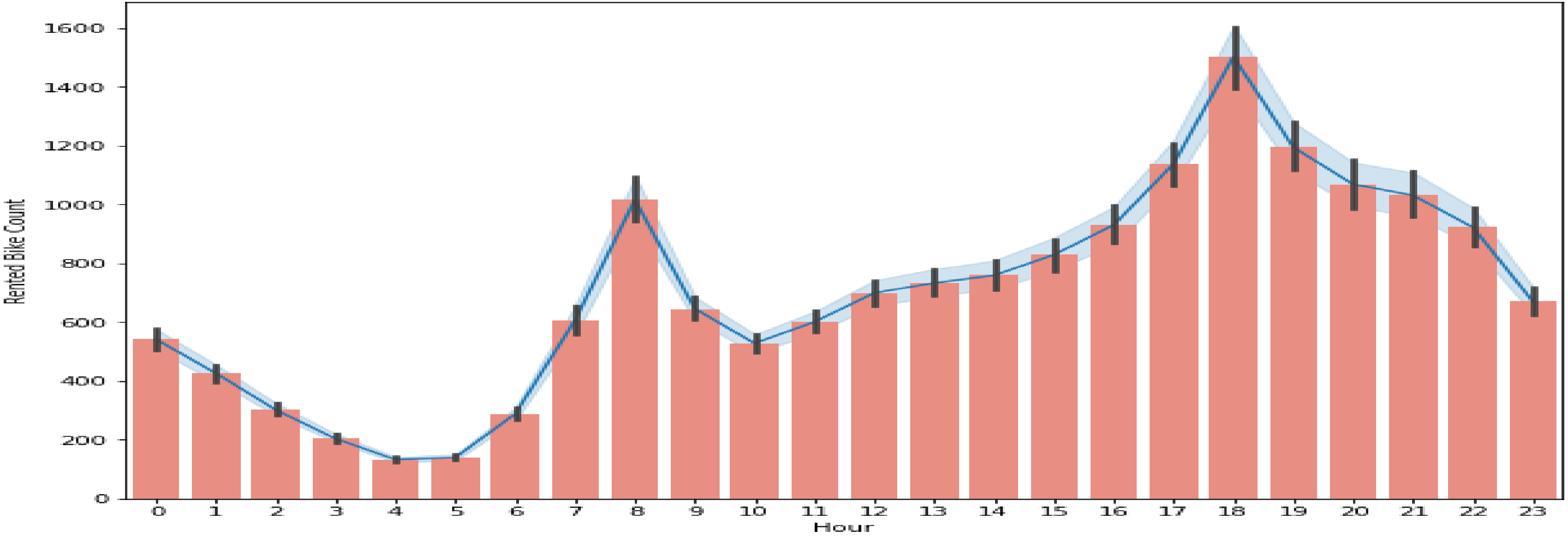
In the month column, We can clearly see that the demand is low in December January & February, It is cold in these months and we have already seen in season column that demand is less in winters. It is high during July and June time as there might be vacations and people love to enjoy outings

EDA Bike count Vs Year



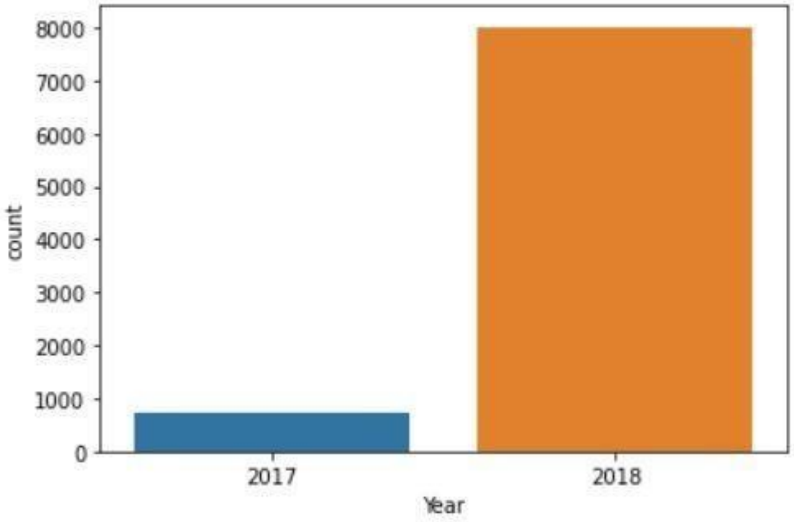
The demand was less in 2017 and higher in 2018, it may be because it was new in 2017 and people did not know much about it.

EDA - Rented bike count per hour



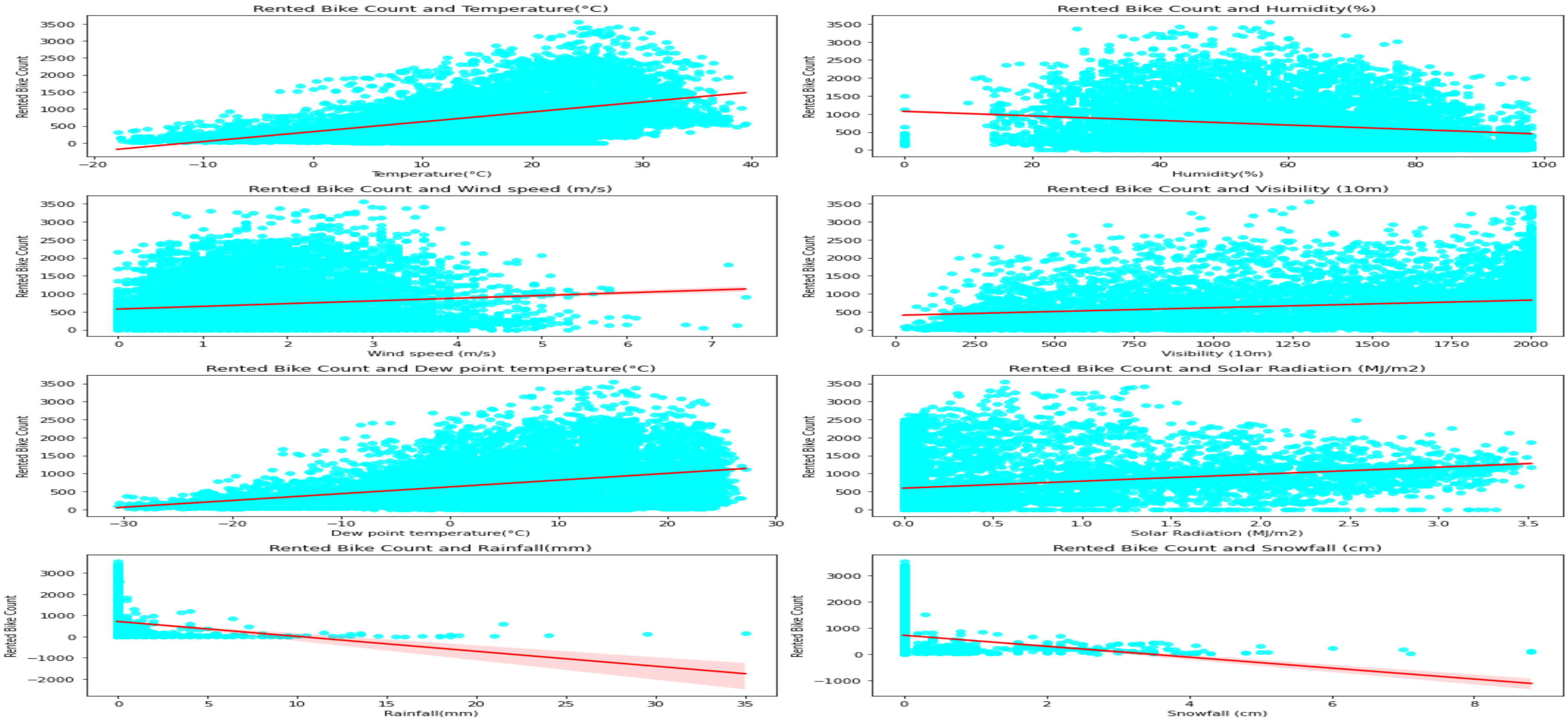
- **Hour:** Demand for bike is mostly in morning (7 to 8) and in the evening (5 to 9).

EDA - Bike count in 2017 and 2018



- Our dataset mostly contains information of year 2018 and very little information of year 2017.

EDA Relation of Bike count Vs Different features


Rented Bike Count and Temperature(°C)

Rented Bike Count and Humidity(%)

Rented Bike Count and Wind speed (m/s)

Rented Bike Count and Visibility (10m)

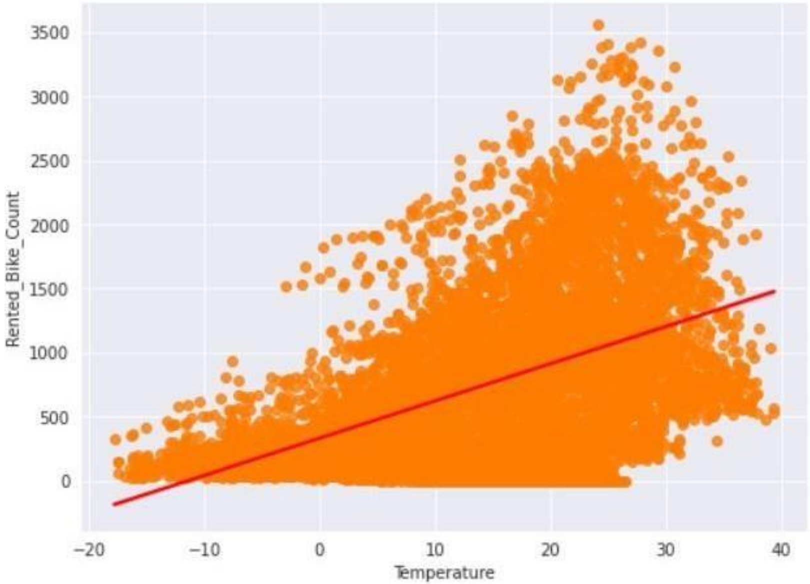
Rented Bike Count and Dew point temperature(°C)

Rented Bike Count and Solar Radiation (MJ/m2)

Rented Bike Count and Rainfall(mm)

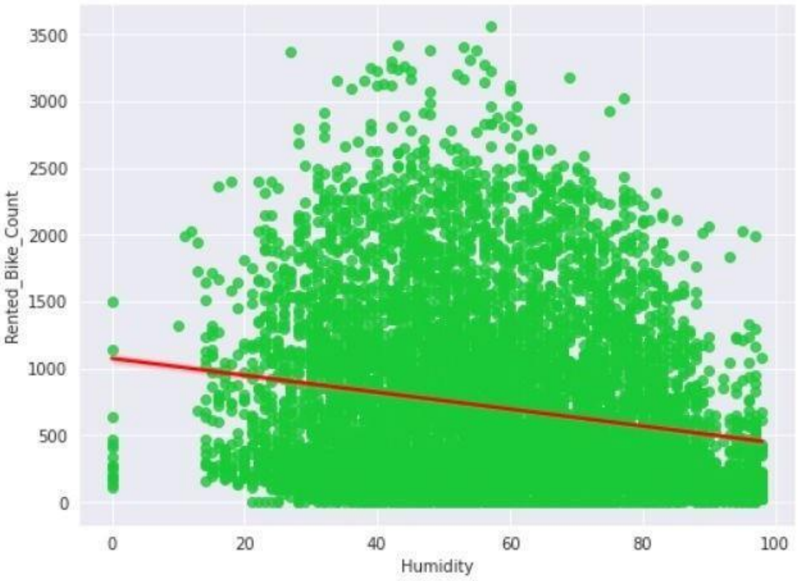
Rented Bike Count and Snowfall (cm)

EDA - Relationship between bike count and Temperature



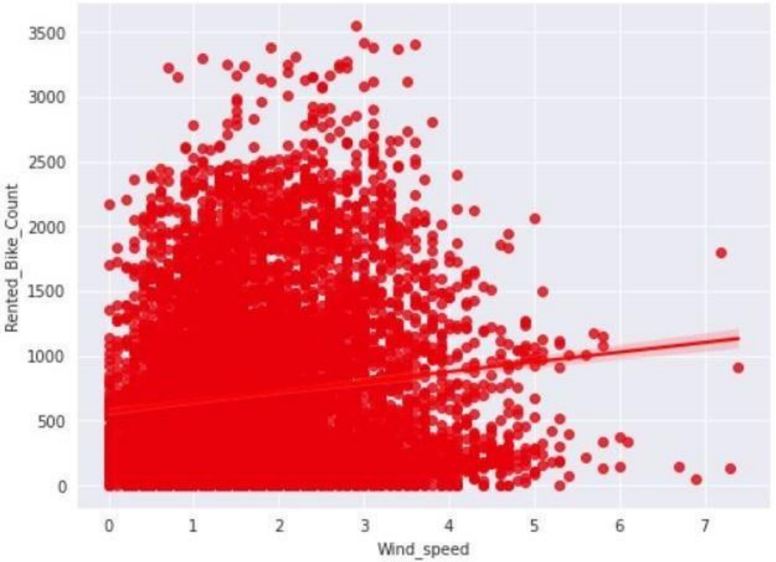
Temperature : Temperature is positively correlated. Rented bike count is highest between 20 °C and 30 °C. So, it means temperature has an effect.

EDA - Relationship between bike count and Humidity



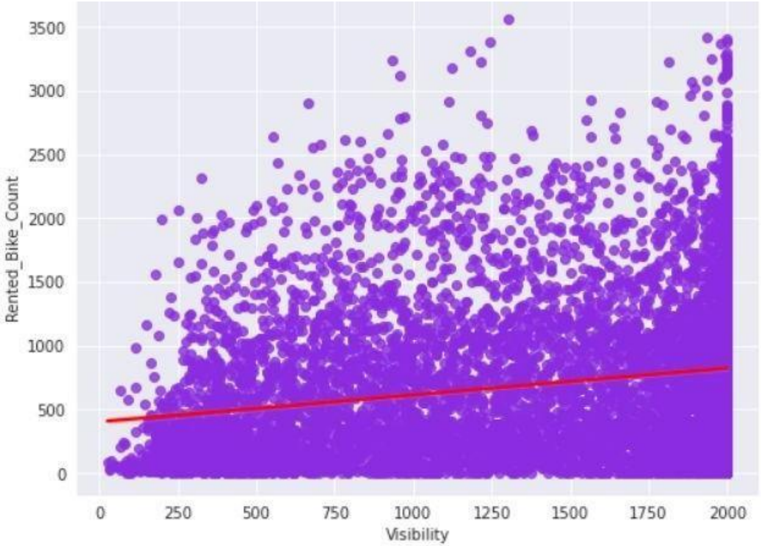
Humidity : Humidity is the amount of water vapor in the air. So, People preferring to borrow bike When there is less humidity.

EDA - Relationship between bike count and Windspeed



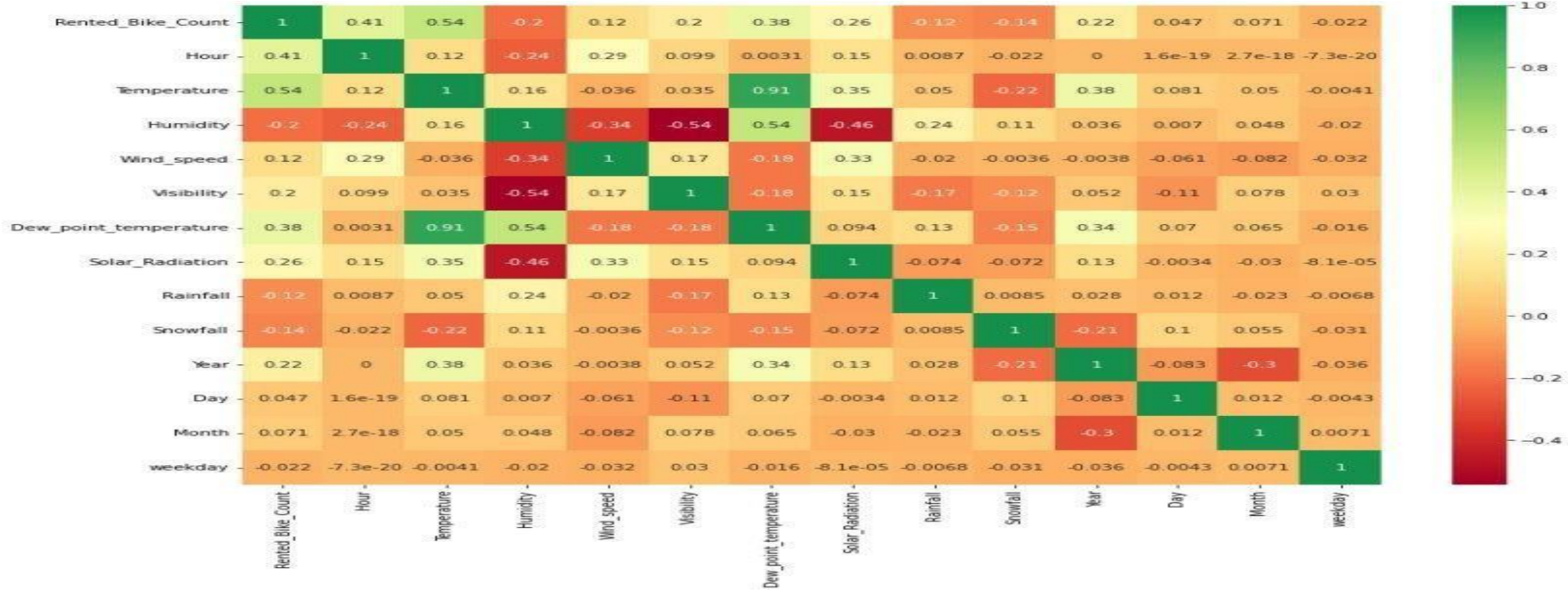
- **Windspeed :** Consumers prefer bikes when wind speed is in particular range but looking at plot, we can conclude that wind speed doesn't affect our data much.

EDA - Relationship between bike count and Visibility



- **Visibility:** Visibility doesn't affect our results much but all we know is that it is positively correlated with bike count.

Correlation between Different factors by using Heatmap



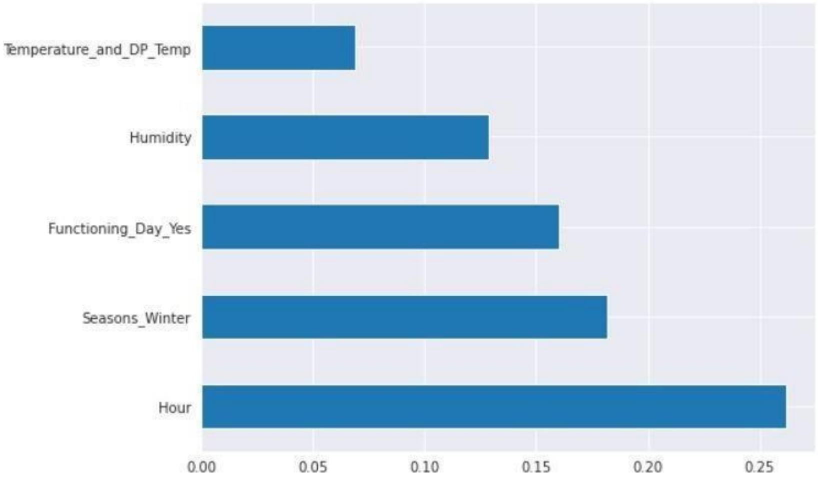
	Rented_Bike_Count	Hour	Temperature	Humidity	Wind_speed	Visibility	Dew_point_temperature	Solar_Radiation	Rainfall	Snowfall	Year	Day	Month	weekday
Rented_Bike_Count	1	0.41	0.54	-0.2	0.12	0.2	0.38	0.26	-0.12	-0.14	0.22	0.047	0.071	-0.022
Hour	0.41	1	0.12	-0.24	0.29	0.099	0.0031	0.15	0.0087	-0.022	0	1.6e-19	2.7e-18	-7.3e-20
Temperature	0.54	0.12	1	0.16	-0.036	0.035	0.91	0.35	0.05	-0.22	0.38	0.081	0.05	-0.0041
Humidity	-0.2	-0.24	0.16	1	-0.34	-0.54	0.54	-0.46	0.24	0.11	0.036	0.007	0.048	-0.02
Wind_speed	0.12	0.29	-0.036	-0.34	1	0.17	-0.18	0.33	-0.02	-0.0036	-0.0038	-0.061	-0.082	-0.032
Visibility	0.2	0.099	0.035	-0.54	0.17	1	-0.18	0.15	-0.17	-0.12	0.052	-0.11	0.078	0.03
Dew_point_temperature	0.38	0.0031	0.91	0.54	-0.18	-0.18	1	0.094	0.13	-0.15	0.34	0.07	0.065	-0.016
Solar_Radiation	0.26	0.15	0.35	-0.46	0.33	0.15	0.094	1	-0.074	-0.072	0.13	-0.0034	-0.03	-8.1e-05
Rainfall	-0.12	0.0087	0.05	0.24	-0.02	-0.17	0.13	-0.074	1	0.0085	0.028	0.012	-0.023	-0.0068
Snowfall	-0.14	-0.022	-0.22	0.11	-0.0036	-0.12	-0.15	-0.072	0.0085	1	-0.21	0.1	0.055	-0.031
Year	0.22	0	0.38	0.036	-0.0038	0.052	0.34	0.13	0.028	-0.21	1	-0.083	-0.3	-0.036
Day	0.047	1.6e-19	0.081	0.007	-0.061	-0.11	0.07	-0.0034	0.012	0.1	-0.083	1	0.012	-0.0043
Month	0.071	2.7e-18	0.05	0.048	-0.082	0.078	0.065	-0.03	-0.023	0.055	-0.3	0.012	1	0.0071
weekday	-0.022	-7.3e-20	-0.0041	-0.02	-0.032	0.03	-0.016	-8.1e-05	-0.0068	-0.031	-0.036	-0.0043	0.0071	1

- Multicollinearity : Dew point temperature and Temperature are variables which are highly correlated.

Plot graph of feature importance

These 5 features affect our dependent variable most.

Hour feature has highest importance so far.

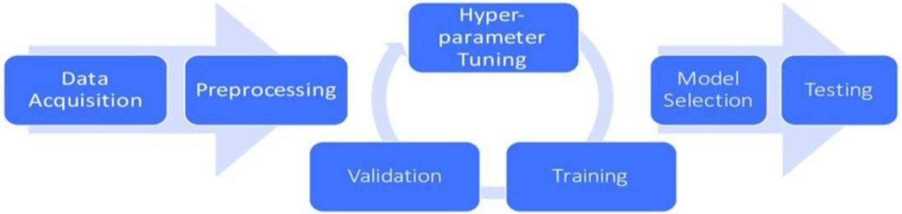


Preparing Dataset for Modelling

- Dataset:

  Train set: (7008, 16)

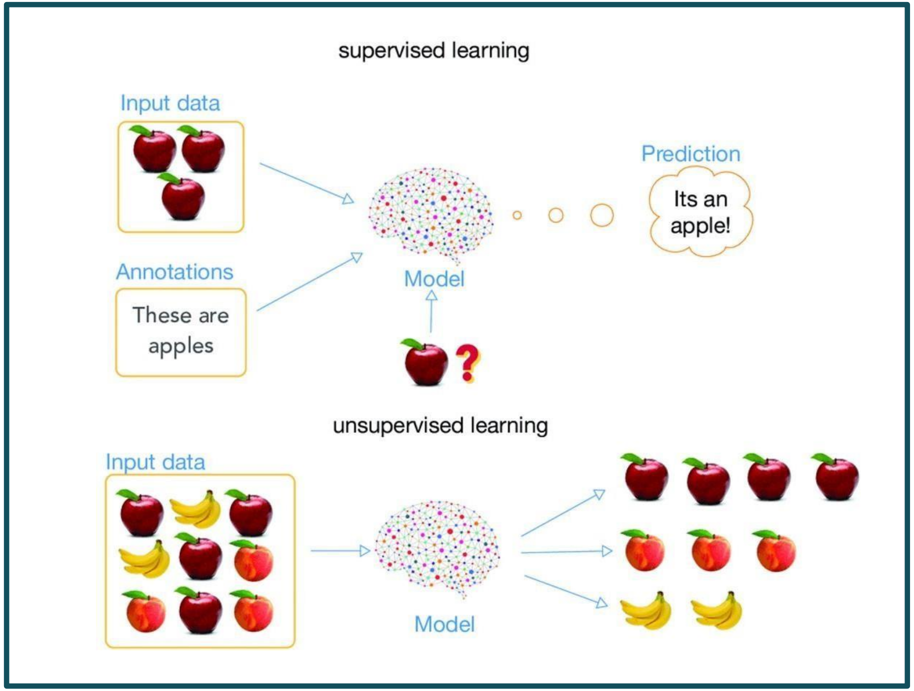
  Test set: (1752, 16)
- Handled ‘Dew point Temperature’ and ‘Temperature’ as they were highly correlated so Dew Point Temp column dropped.
- Dropped ‘Date’ column after extracting useful features from it.
- Carefully handled feature selection part as it affects R2 score.



Modeling Overview

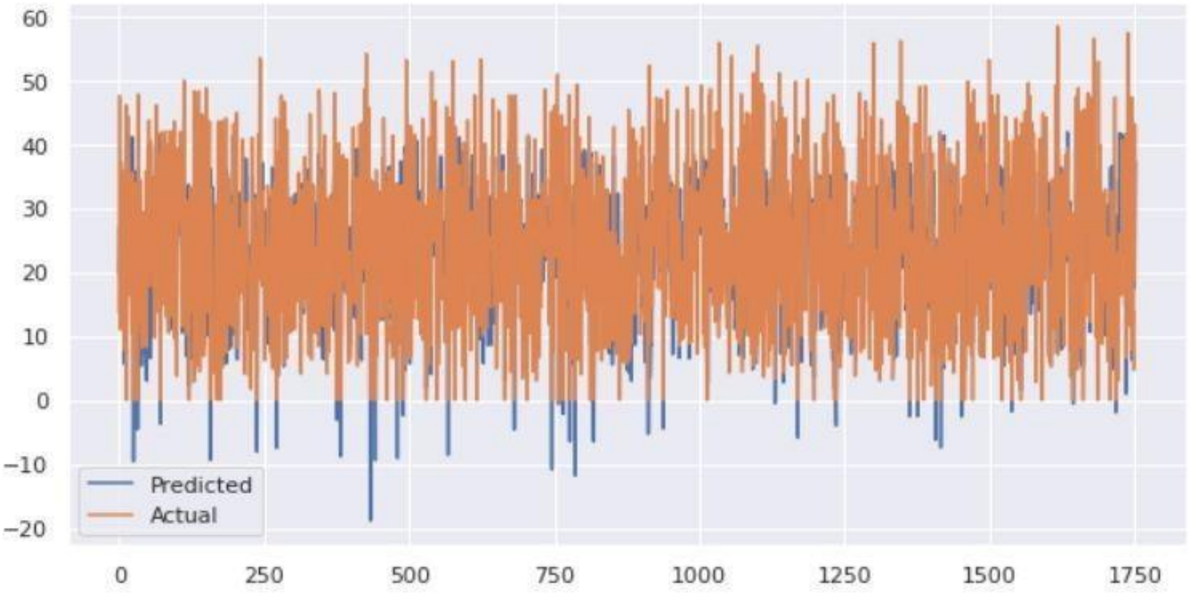
- Type – Supervised Learning
- In this project, we are using eight model on our data set for getting best performance :

1. LINEAR REGRESSION
2. LASSO REGRESSION
3. RIDGE REGRESSION
4. ELASTIC NET REGRESSION
5. POLYNOMIAL FEATURE
6. GRADIENT BOOSTING
7. DECISION TREE
8. RANDOM FOREST
9. XGBOOST



Implementing linear regression (Baseline Model)

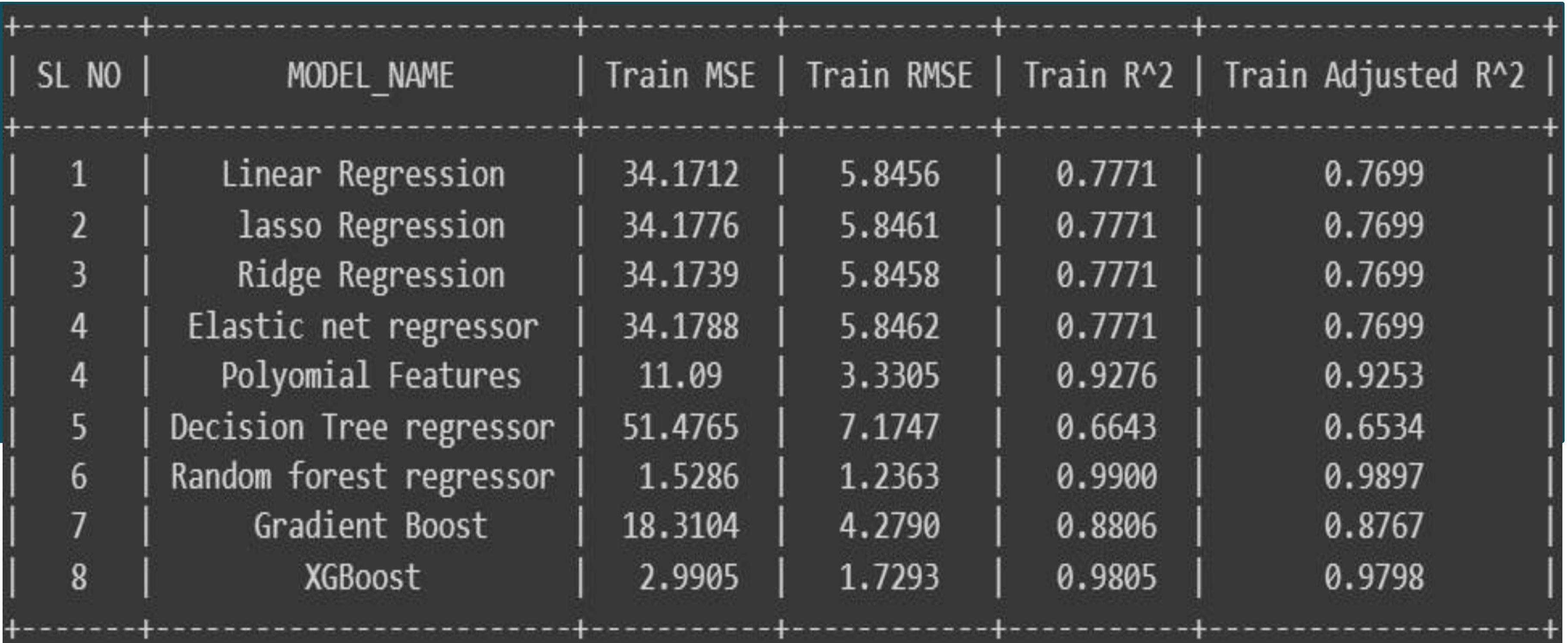
```
MSE : 36.44312929853443
RMSE : 6.036814499264859
R2 : 0.7743096534955213
Adjusted R2 : 0.7669906858907181
```



- Accuracy is just 77%, we need to look for more algorithms to see if we can get better results.

Model Validation & Selection

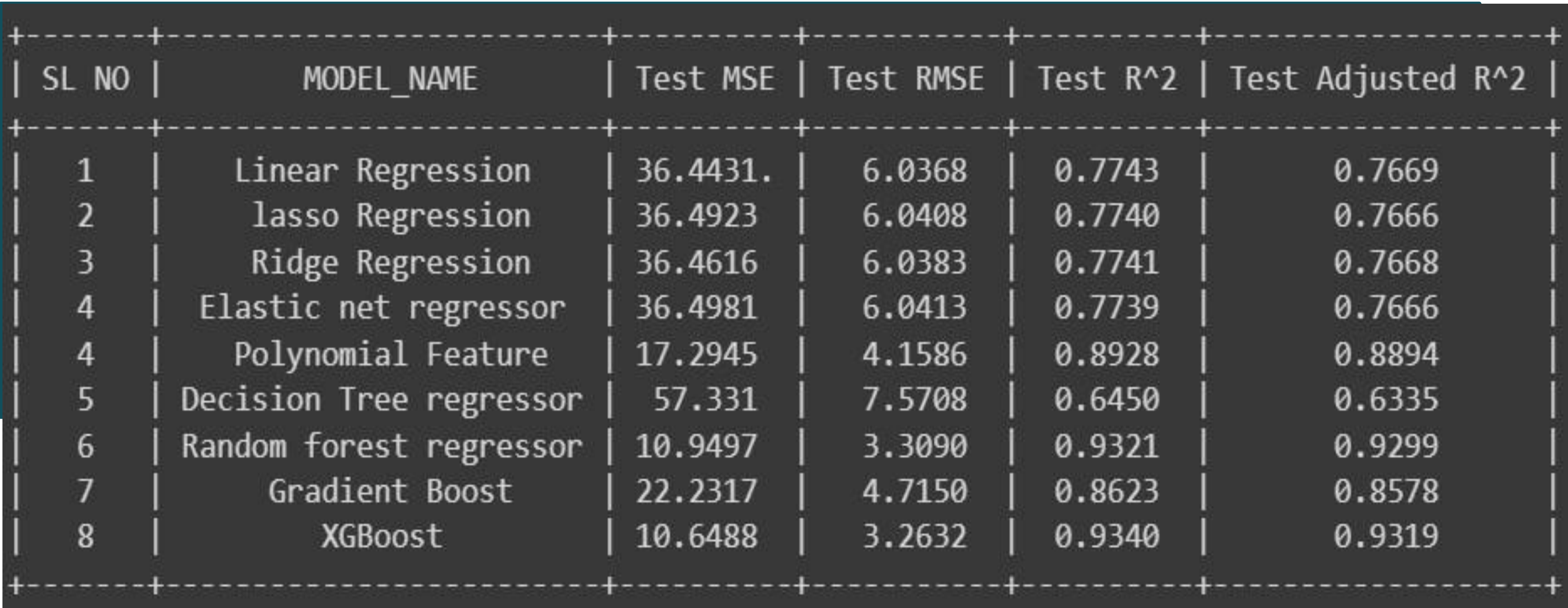
- Model Summary for the train data set (Hyperparameter tuned)



```
+-------+--------------------------+------------+-------------+------------+---------------------+
| SL NO |        MODEL_NAME        | Train MSE  | Train RMSE  | Train R^2  | Train Adjusted R^2  |
+-------+--------------------------+------------+-------------+------------+---------------------+
|   1   |    Linear Regression     |  34.1712   |   5.8456    |   0.7771   |       0.7699        |
|   2   |     lasso Regression     |  34.1776   |   5.8461    |   0.7771   |       0.7699        |
|   3   |     Ridge Regression     |  34.1739   |   5.8458    |   0.7771   |       0.7699        |
|   4   |   Elastic net regressor  |  34.1788   |   5.8462    |   0.7771   |       0.7699        |
|   4   |    Polyomial Features    |   11.09    |   3.3305    |   0.9276   |       0.9253        |
|   5   | Decision Tree regressor  |  51.4765   |   7.1747    |   0.6643   |       0.6534        |
|   6   | Random forest regressor  |   1.5286   |   1.2363    |   0.9900   |       0.9897        |
|   7   |      Gradient Boost      |  18.3104   |   4.2790    |   0.8806   |       0.8767        |
|   8   |         XGBoost          |   2.9905   |   1.7293    |   0.9805   |       0.9798        |
+-------+--------------------------+------------+-------------+------------+---------------------+
```

Model Validation & Selection

- Model Summary for the test data set (Hyperparameter tuned)



```
+-------+-------------------------+----------+-----------+----------+-------------------+
| SL NO |        MODEL_NAME       | Test MSE | Test RMSE | Test R^2 | Test Adjusted R^2 |
+-------+-------------------------+----------+-----------+----------+-------------------+
|   1   |    Linear Regression    | 36.4431. |   6.0368  |  0.7743  |       0.7669      |
|   2   |     lasso Regression    | 36.4923  |   6.0408  |  0.7740  |       0.7666      |
|   3   |     Ridge Regression    | 36.4616  |   6.0383  |  0.7741  |       0.7668      |
|   4   |  Elastic net regressor  | 36.4981  |   6.0413  |  0.7739  |       0.7666      |
|   4   |    Polynomial Feature   | 17.2945  |   4.1586  |  0.8928  |       0.8894      |
|   5   | Decision Tree regressor |  57.331  |   7.5708  |  0.6450  |       0.6335      |
|   6   | Random forest regressor | 10.9497  |   3.3090  |  0.9321  |       0.9299      |
|   7   |      Gradient Boost     | 22.2317  |   4.7150  |  0.8623  |       0.8578      |
|   8   |         XGBoost         | 10.6488  |   3.2632  |  0.9340  |       0.9319      |
+-------+-------------------------+----------+-----------+----------+-------------------+
```

Model Validation & Selection

Observation 1:From model summary table, Linear regression is not giving us accurate result.

Observation 2: lasso, ridge and elastic net regressor didn’t give any good results either. So, we went for more M.L. models.

- Observation 3: Decision tree not giving worst results of all and Random forest gave decent results with approx. 93% accuracy.

- Observation 4: Gradient Boost and XGBoost gave us better results. We came to the conclusion that XGBoost is performing better than all other algorithms with accurate.

Scenario while applying Gradient Boost

On using GridSearchCV for the Gradient boost regressor, we get best hyperparameters. They are given below.

Best hyperparameters:

base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1, gamma=0, importance_type='gain', learning_rate=0.1, max_delta_step=0, max_depth=8, min_child_weight=1, min_samples_leaf=40, min_samples_split=50, missing=None, n_estimators=100, n_jobs=1, nthread=None, objective='reg:linear', random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None, silent=None, subsample=1, verbosity=1)

Scenario while applying XGBoost

On training XGBoost regressor with Grid search, best hyperparameters obtained are given below.

Best parameters:

base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1, gamma=0, importance_type='gain', learning_rate=0.1, max_delta_step=0, max_depth=8, min_child_weight=1, min_samples_leaf=40, min_samples_split=50, missing=None, n_estimators=100, n_jobs=1, nthread=None, objective='reg:linear', random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None, silent=None, subsample=1, verbosity=1

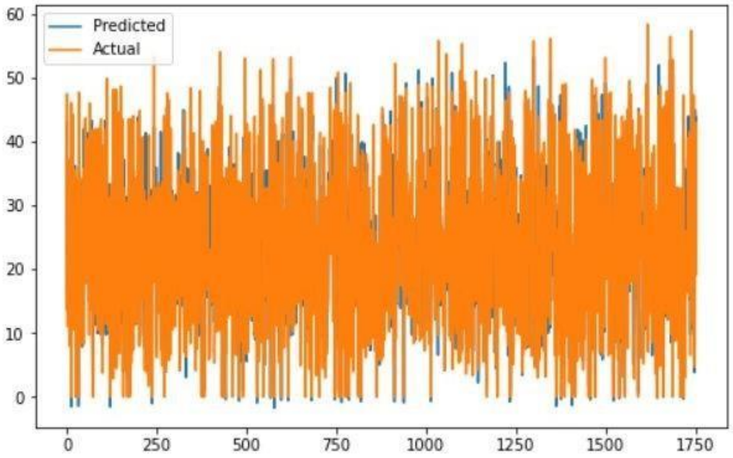
Gradient Boost(test data)

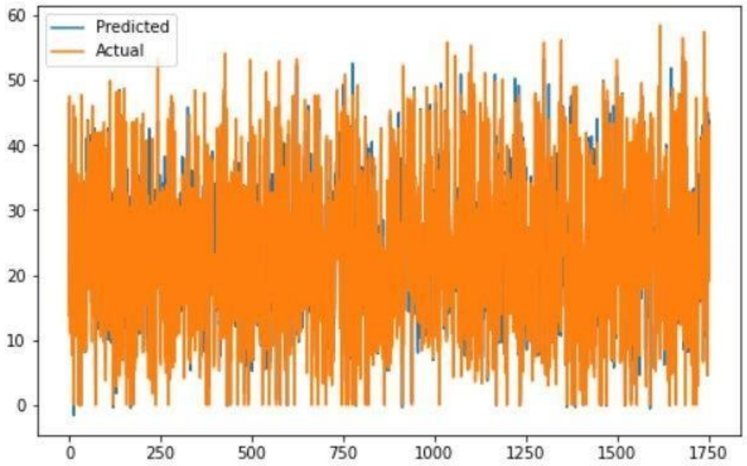
```
MSE : 7.702191174843459
RMSE: 2.775282179318611
R2   : 0.9497801781511271
Adjusted R2 : 0.9481515872303206
```

We can now conclude that XGBOOST has higher accuracy.

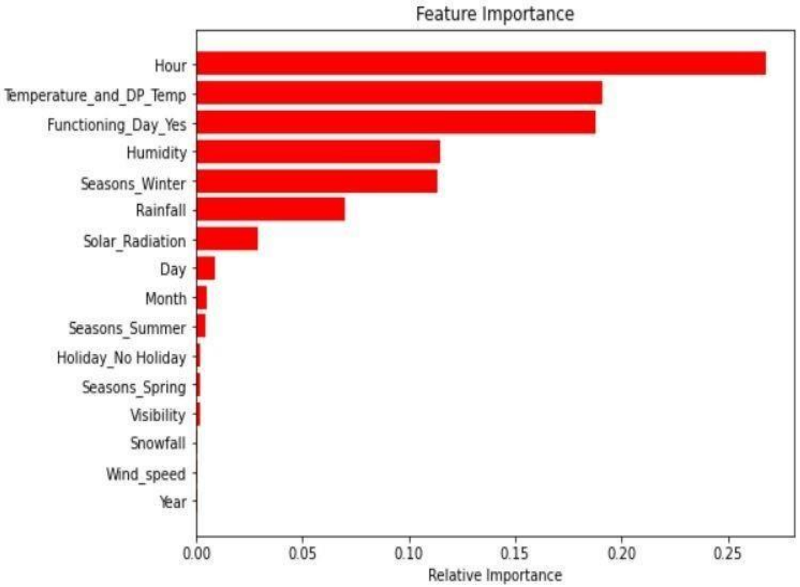
XG boost(test data)

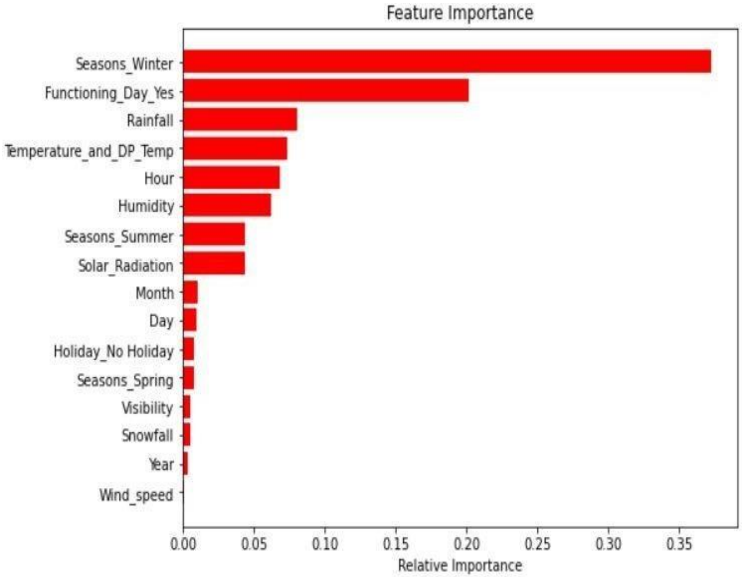
```
MSE :10.648825082485597
RMSE:3.2632537569863604
R2   :0.934052397008936
Adjusted R2 : 0.9319137660157117
```





Feature Importance



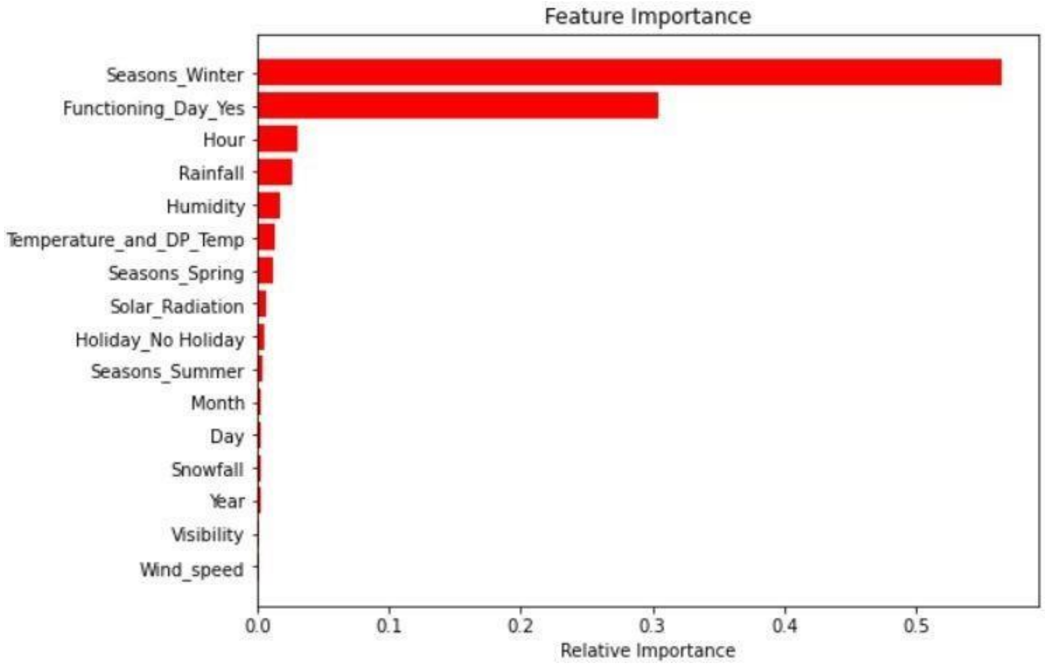


Gradient boost

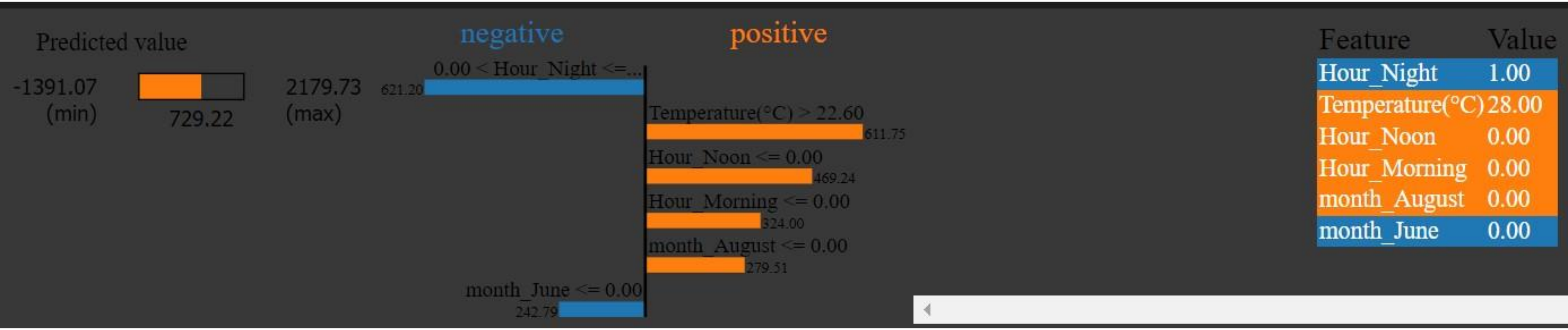
XGBOOST

Feature importance with XGBoost model Grid Search-cv

Winter season is the most relevant feature here.



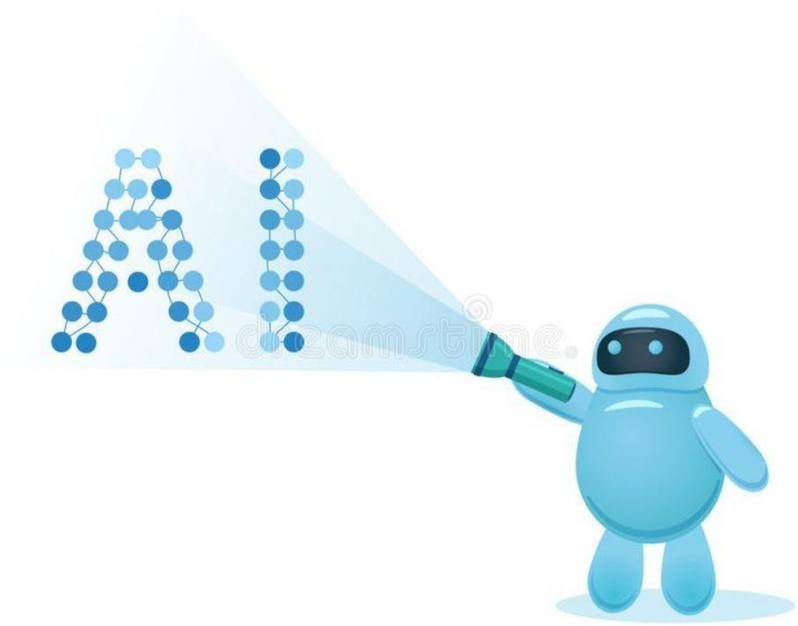
Feature-visualization (LIME)



Feature	Value
Hour_Night	1.00
Temperature(°C)	28.00
Hour_Noon	0.00
Hour_Morning	0.00
month_August	0.00
month_June	0.00

Challenges

- Large dataset to handle
- Feature engineering
- Feature selection - Making sure we don’t miss any important feature.
- Careful tuning of hyperparameters as it affects R2 score.
- Computation time



Conclusion

- We implemented 9 M.L. models. After comparing the mean square error and mean root square error of all the models, XG Boost has least mean square error and root mean square error. XG Boost has highest accuracy of 93.4% among all algorithms. So, we can conclude that XG Boost is the best model to predict rented bike count.
- The number of business hours of the day and the demand for rented bikes were most correlated and it makes sense also. Highest number of bikes rented at the 18th hour of day.
- In the week column, We can observe from this column that the pattern of weekdays and weekends is different. In the weekend the demand becomes high in the afternoon as people loves to travel during this hour . While the demand during office timings is high in weekdays.

Conclusion

- Total number of bike count increased when there was favorable temperature. So, this can be an important factor in predicting underlying patterns of rented bike count.
- In the month column, We can clearly see that the demand is low in December January & February, It is cold in these months and we have already seen in season column that demand is less in winters. It is high during July and June time as there might be vacations and people love to enjoy outings.
- The important features for bike rented prediction is temperature followed by humidity and hour of day.

THANK YOU

Q & A