Capstone Project-2

Seoul Bike Sharing Demand Prediction

Aashruti A
Raneev K
Kunal M

Flow of Presentation

- Agenda
- Data Summary
- EDA
- Data Preprocessing
- ML models
- Conclusion

Agenda

- 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.
- The objective of this project is to build a ML model which is optimal and which is able to predict the bike count required based on available features.

Data Summary

- Date : The date of the day, type:str
- Rented Bike Count - Number of rented bikes per hour and it is also a dependent variable, type:int
- Hour - Hour of the day ranging from 0-23, type: int
- Temperature (°C)- Temperature in Celsius, type:float
- Humidity(%) - Humidity in the air in %, type: int
- Wind speed (m/s) - Speed of the wind in m/s, type: float
- Visibility (10m) - Visibility in m, type: int
- Dew point temperature(°C) - The temperature at which the water starts to condense out of the air, type: float
- Solar Radiation (MJ/m2) - Electromagnetic radiation emitted by the Sun, type: float
- Rainfall(mm) - Amount of rainfall in mm, type: float italicized text
- Snowfall(cm) - Amount of snowfall in cm, type: float
- Seasons - Season of the year, type: str
- Holiday - If the day is holiday or not, type: str
- Functioning Day - Whether the day is functional or not, type:str

Cleaning of dataset

Data Preparation and Cleaning

```
[ ]  # Missing Values Check
     bike_df.isnull().sum()
```

```
Date                       0
Rented Bike Count          0
Hour                       0
Temperature(°C)            0
Humidity(%)                0
Wind speed (m/s)           0
Visibility (10m)           0
Dew point temperature(°C)  0
Solar Radiation (MJ/m2)    0
Rainfall(mm)               0
Snowfall (cm)              0
Seasons                    0
Holiday                    0
Functioning Day            0
dtype: int64
```

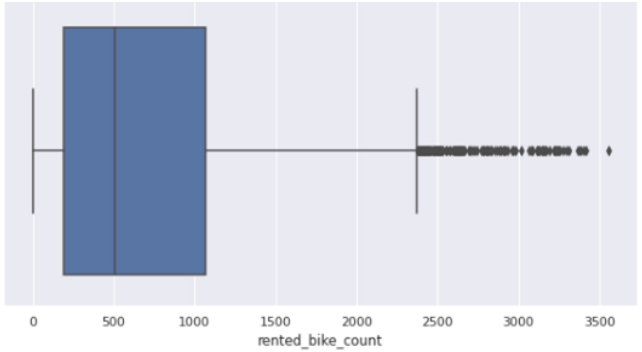
As we can see from above result that there are no missing values in the dataset.

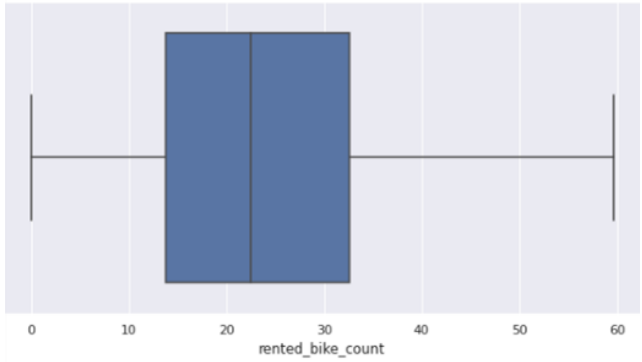
Duplicate Values Check

```
[ ]  # Checking whether there are duplicate records in the dataset.
     dup_no = len(bike_df[bike_df.duplicated()])
     print('The number of duplicated rows in the dataset are: ', dup_no)
```

```
The number of duplicated rows in the dataset are:  0
```

- There are not any duplicate rows and missing values in the dataset
- Some feature names are quite lengthy, lets rename the features





Data Preprocessing

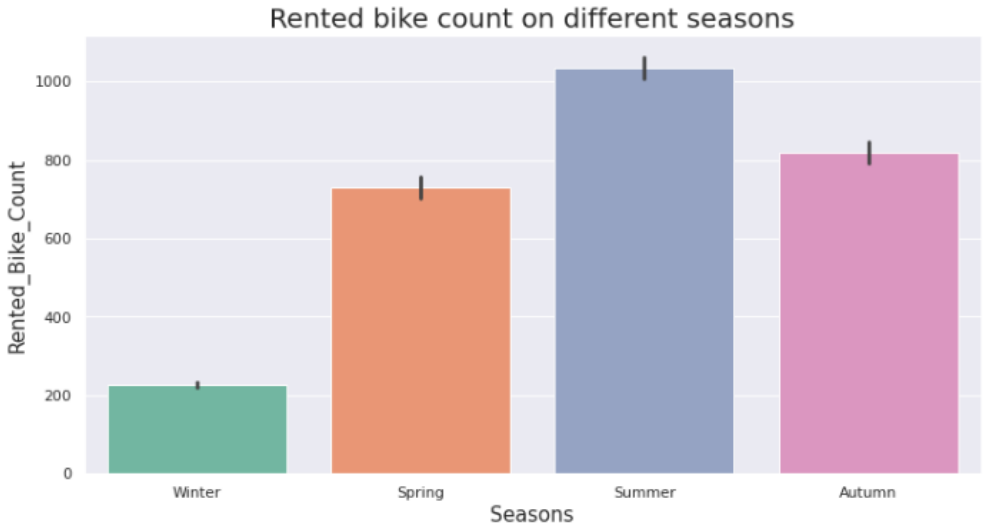
```
[70] # # One hot encoding
     # #creating a copy of the dataframe
     bike_data = bike_df.copy()

     def one_hot_encoding(data, column):
       data = pd.concat([data, pd.get_dummies(data[column], prefix = column, drop_first = True)], axis =1)
       data = data.drop([column], axis = 1)
       return data

     for col in cat_feat:
       bike_data = one_hot_encoding(bike_data, col)
```

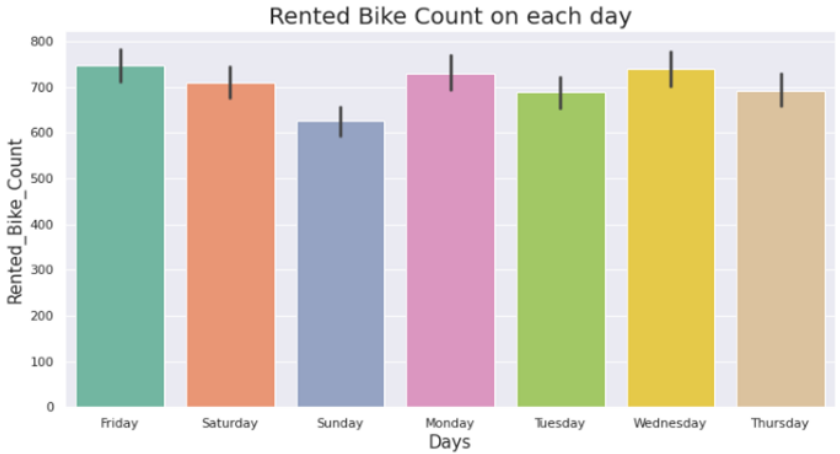
- Creating dummy variables for categorical variables using one hot encoding.
- A one hot encoding allows the representation of categorical data to be more expressive.
- Many machine learning algorithms cannot work with categorical data directly.
- The categories must be converted into numbers.
- This is required for both input and output variables that are categorical

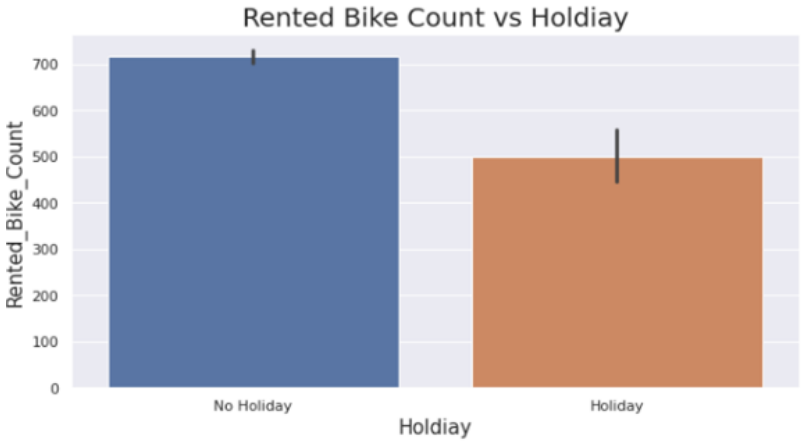
Data Visualization



- From above graph it is observed that, people prefer renting bikes highest at summer and least at winter.

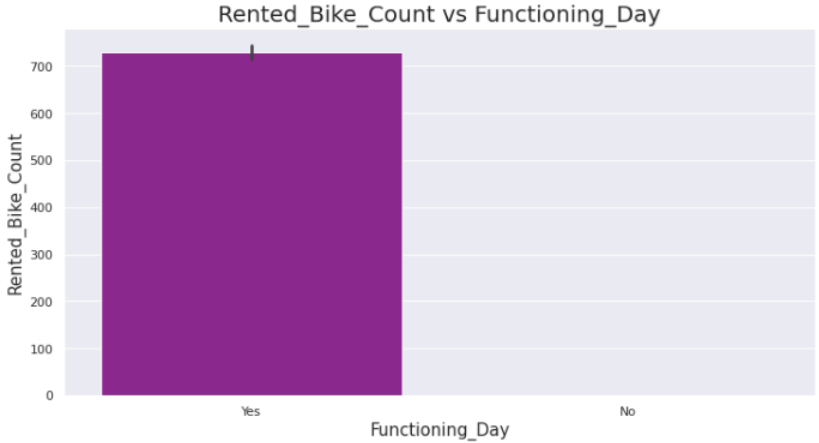
Data Visualization

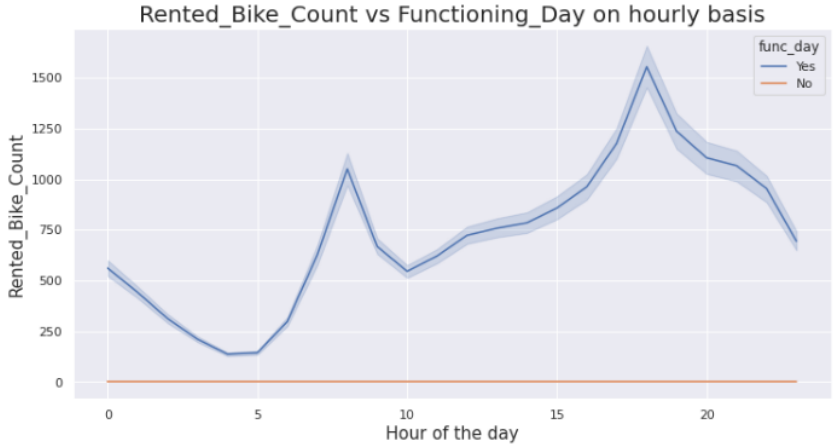




- The rented bike count is pretty much the same throughout the week being least on Sunday.
- People are renting bikes more on a non holiday than a holiday.

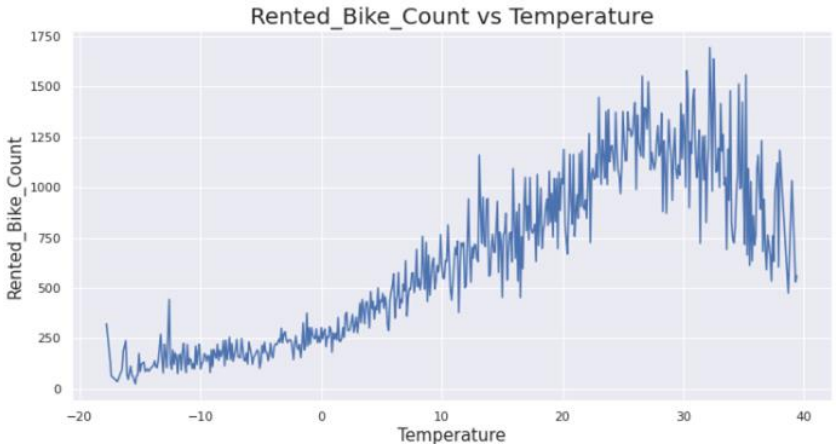
Data Visualization

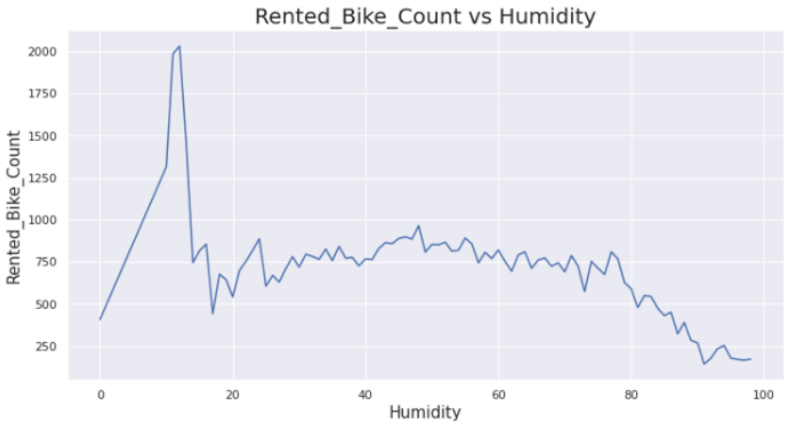




- The rented bike count is zero for a non functioning day which suggests that if no one rents a bike on a particular day then it is a non functioning day for the company.
- From above graph it is observed that, people prefer renting bikes around morning 7 - 9 am and in the evening 5 - 7 pm.

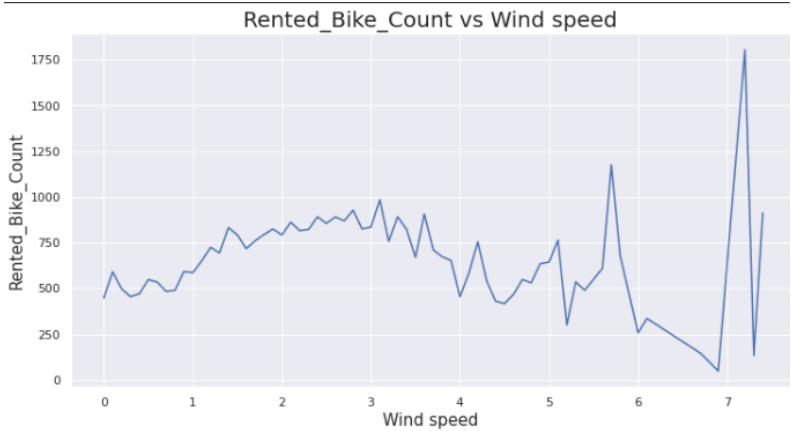
Data Visualization

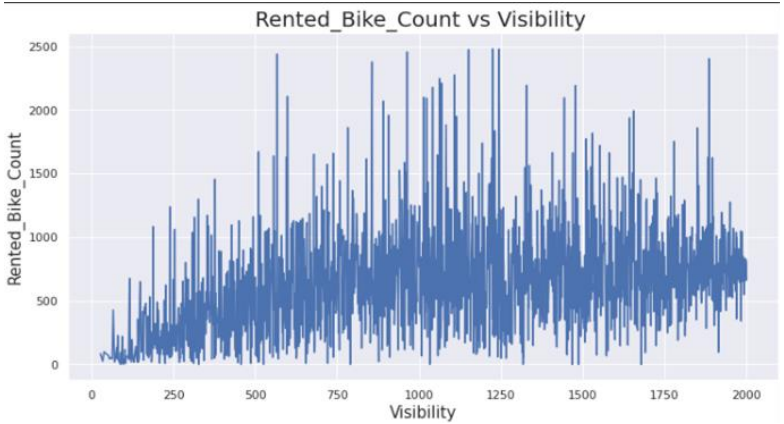




- The bike count increases with temperature. The demand is maximum around 25-35 °C.
- The bike count decreases as the humidity increases. The demand is maximum around 15% humidity.

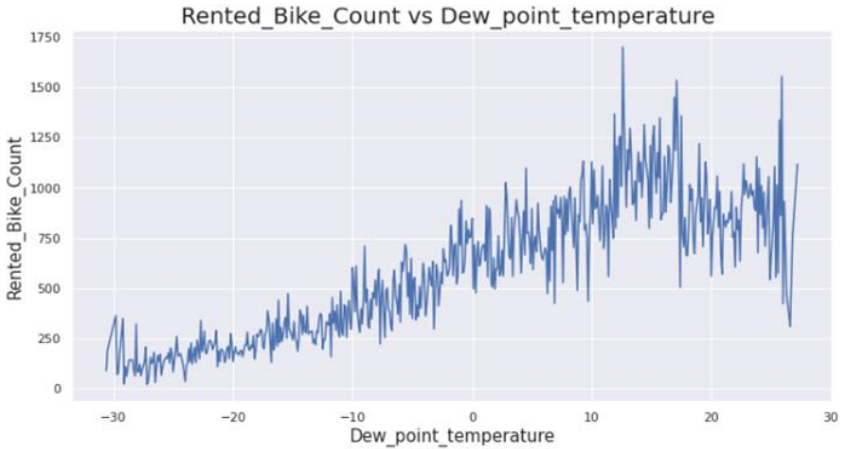
Data Visualization

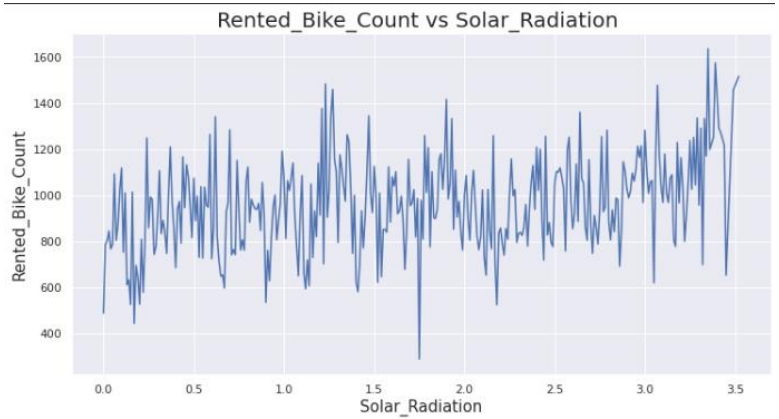




- As evident from above graph, the rented bike count decreases with increase in wind speed, but there is a spike in count at around 7m/s of wind speed.
- Visibility seems to follow a linear relationship with rented bike count upto a particular value after that visibility doesn't make any impact on rented bike count

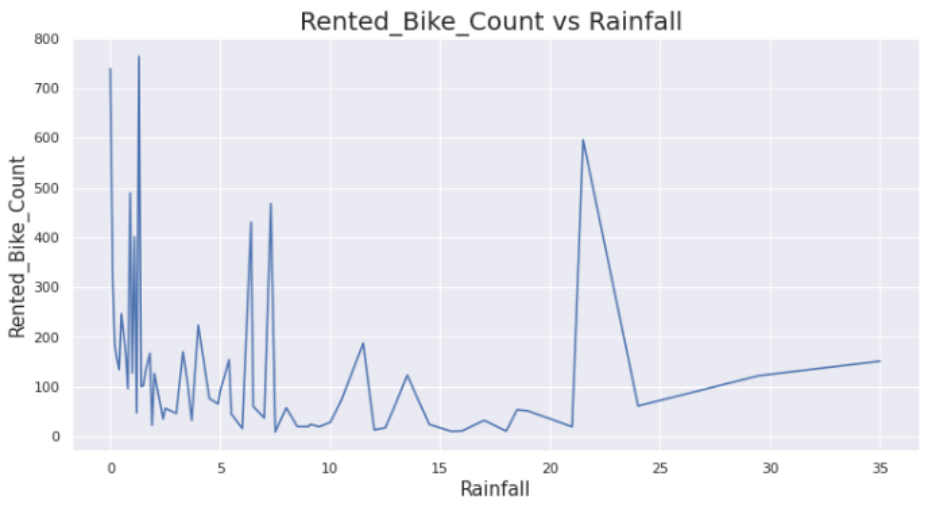
Data Visualization

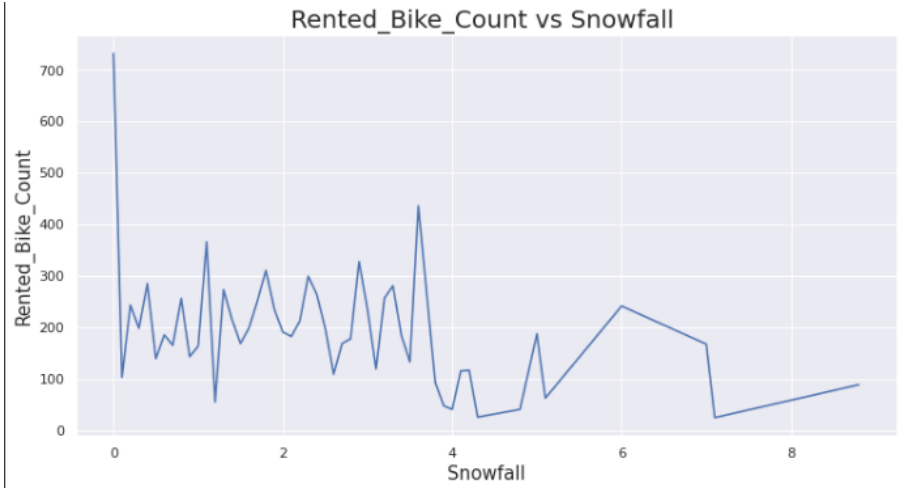




- We can observe that rented bike count increases with increase in dew point temperature till about 20°C.
- Solar radiation does not have much impact on the number of bikes rented.

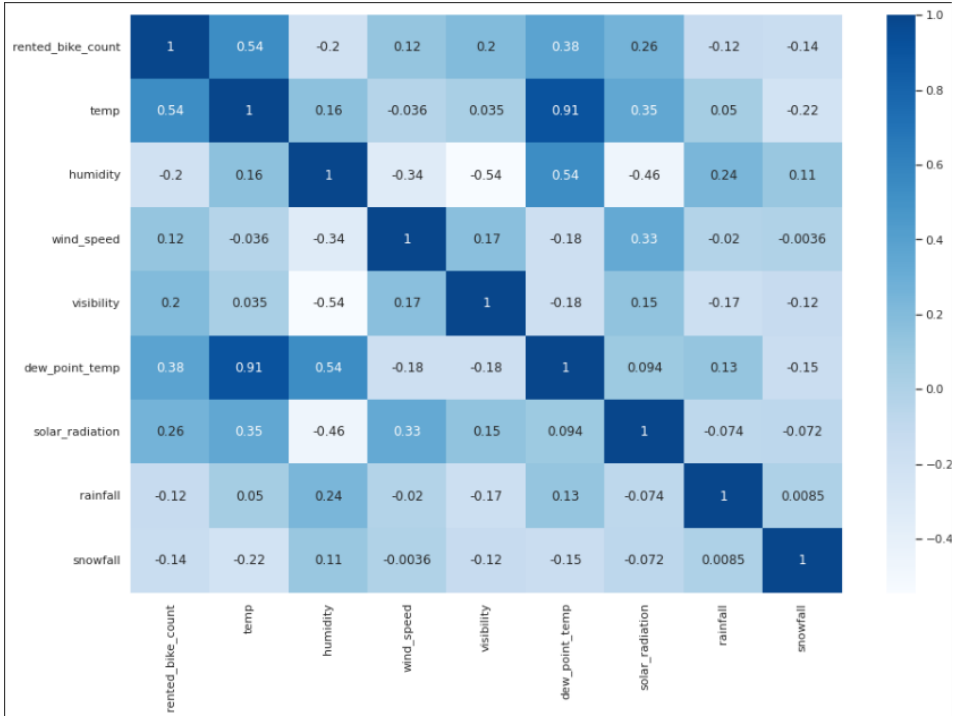
Data Visualization





- Looking at the above graphs, it is evident that people don't prefer to rent bikes during rainfall and snowfall.

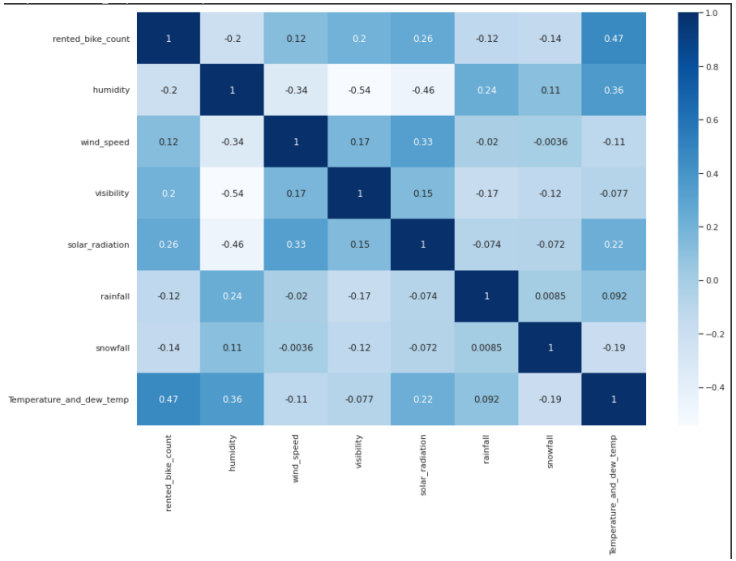
Data Visualization



- The correlation between temperature and dew point temperature is 0.91 which is quite high. To eliminate multicollinearity we will merge those 2 features to form a single feature.

Feature Engineering

```
[66] # creating new feature from temperature and dew point temperature as they both are related to each other.
     bike_df['Temperature_and_dew_temp'] = bike_df['temp'] + bike_df['dew_point_temp']
     bike_df.head()
```



Combining the high correlated features to form single feature called temperature and dew point feature.

Model Building

- Linear Regression
- Lasso Regression
- Ridge Regression
- Decision Trees Regressor
- Random Forest Regressor
- Gradient Boosted Regressor
- Gradient Boosting Regressor With Gridsearch cv

Evaluation Metrics - Linear Regression Model

Training Data

MSE : 100605.9122673585
RMSE : 317.1843506028608
MAE : 215.73174171121056
R2 : 0.7577179934011766
Adjusted R2 : 0.7503026524105123

Testing Data

MSE : 98047.31356421151
RMSE : 313.1250765496298
MAE : 212.05237632910647
R2 : 0.7657299360440506
Adjusted R2 : 0.7585598104844806

Evaluation Metrics - Lasso Regression

Training Data

MSE : 100800.30348062365
RMSE : 317.4906352644494
MAE : 215.86389963123858
R2 : 0.7572498549771652
Adjusted R2 : 0.7498201860300273

Testing Data

MSE : 98235.1747597836
RMSE : 313.4249108794379
MAE : 212.2031411267829
R2 : 0.7652810685258937
Adjusted R2 : 0.7580972048197998

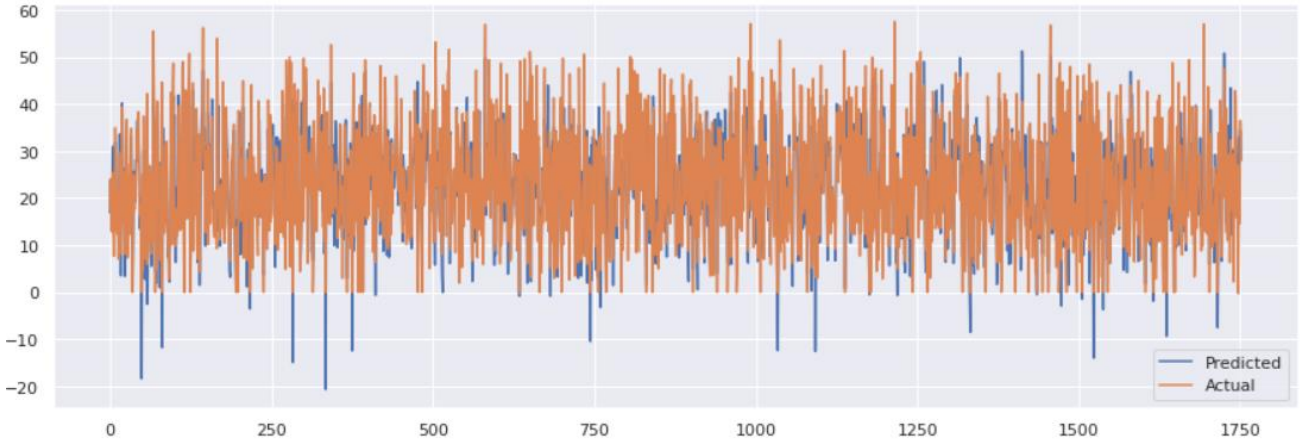
Evaluation Metrics - Ridge Regression

Training Data

MSE : 100807.65245404908
RMSE : 317.50220858137203
MAE : 215.87971845825064
R2 : 0.7572321569712746
Adjusted R2 : 0.7498019463547392

Testing Data

MSE : 98257.40907824635
RMSE : 313.460378801287
MAE : 212.23370338006276
R2 : 0.7652279427948672
Adjusted R2 : 0.7804081853188356



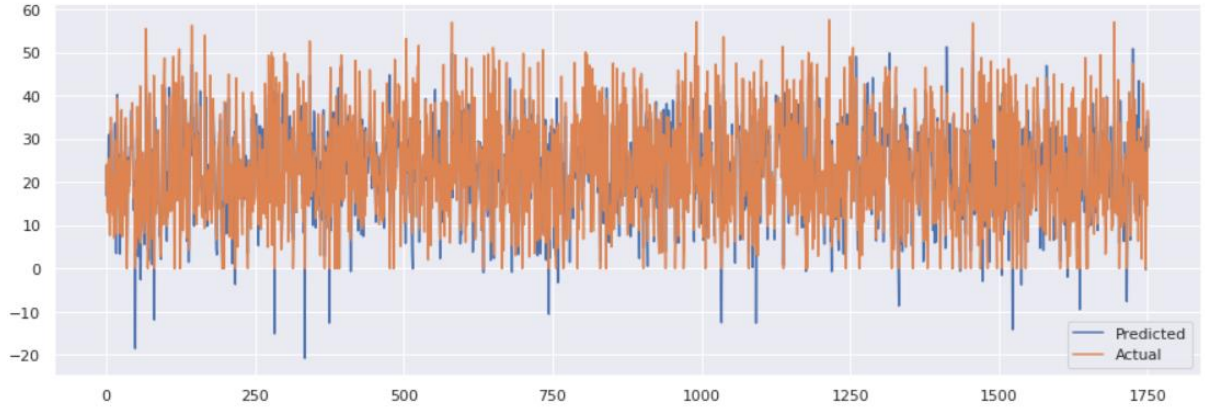
Evaluation Metrics - Elastic Net Regression

Training Data

MSE : 100613.77639400275
RMSE : 317.19674713654103
MAE : 215.7375838946109
R2 : 0.7576990547887176
Adjusted R2 : 0.7502831341583547

Testing Data

MSE : 98055.5049129043
RMSE : 313.1381562711646
MAE : 212.05953020308382
R2 : 0.7657103639852926
Adjusted R2 : 0.7585396393986152



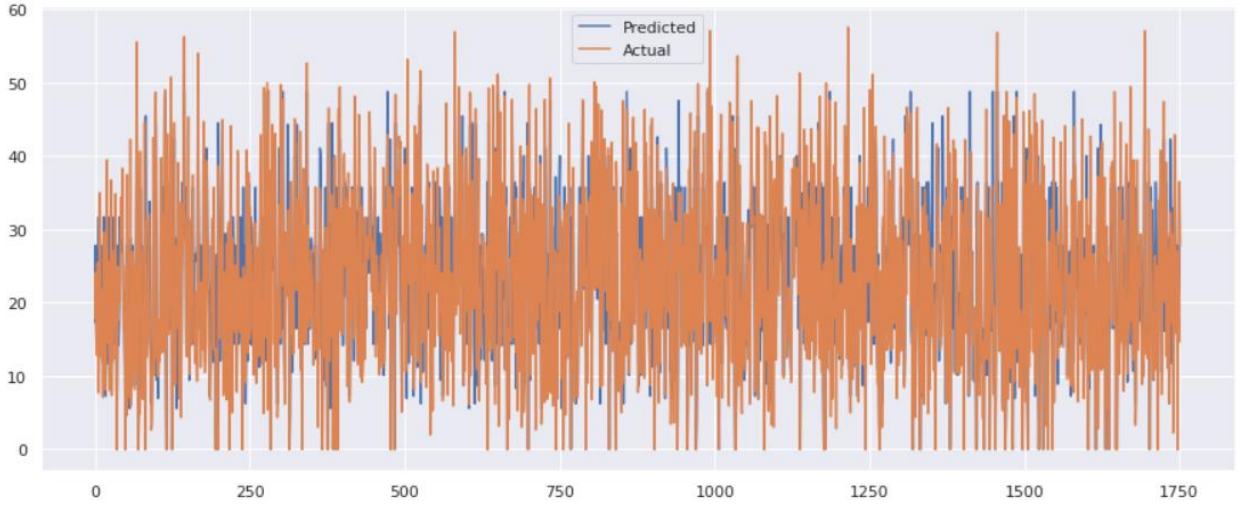
Evaluation Metrics - Decision Tree

Training Data

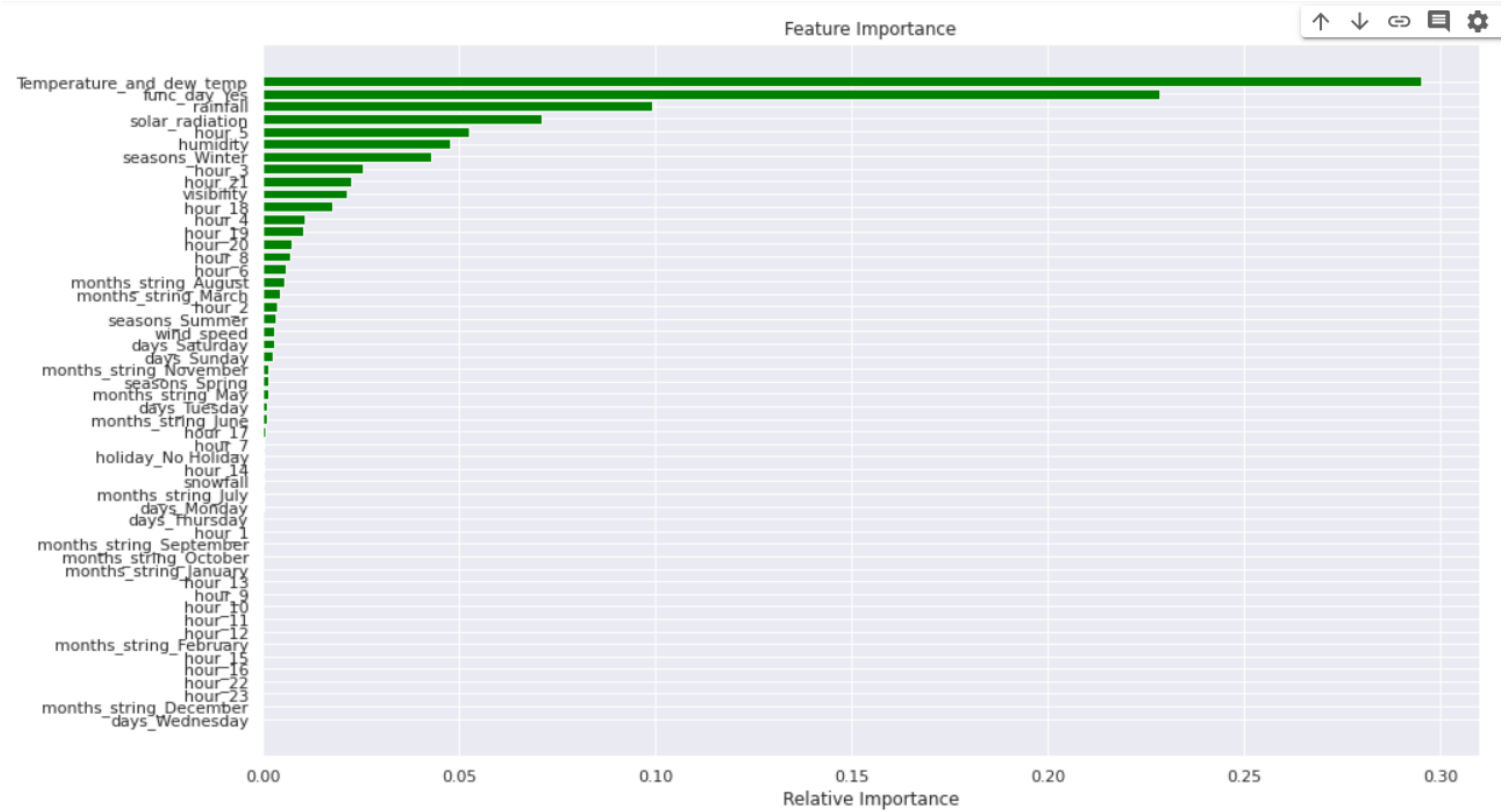
Model Score: 0.7581755514301581
MSE : 129012.39905028316
RMSE : 359.1829604119371
MAE : 237.00706845725372
R2 : 0.6893086875951706
Adjusted R2 : 0.6703175507788324

Testing Data

MSE : 133881.92968315782
RMSE : 365.8987970507116
MAE : 240.89976181684554
R2 : 0.6801082345935101
Adjusted R2 : 0.6703175507788324



Feature Importance - Decision Tree



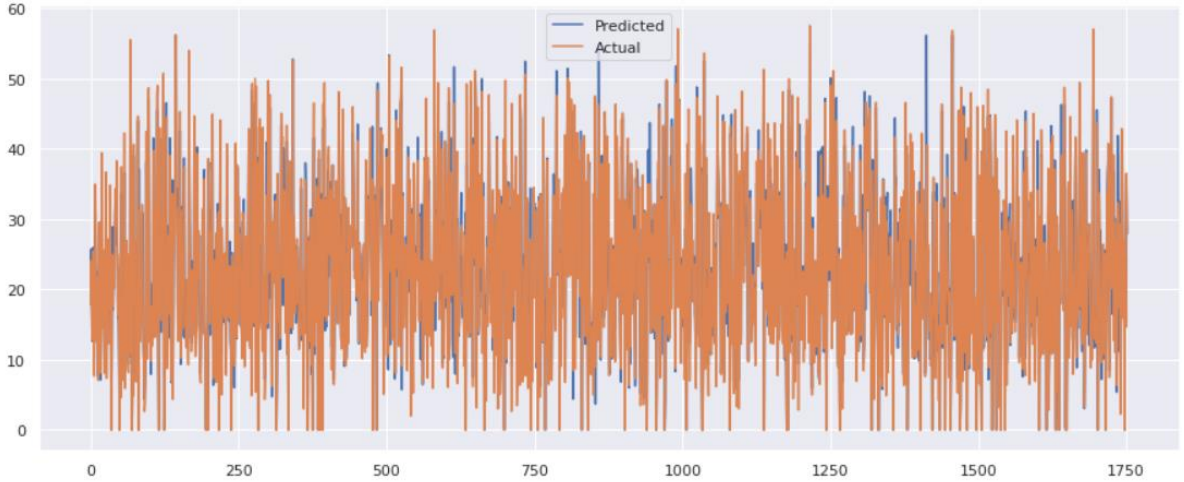
Evaluation Metrics - Random Forest

Training Data

Model Score: 0.9886831314935745
MSE : 4964.034325599896
RMSE : 70.45590341199164
MAE : 41.21999795530623
R2 : 0.9880454719794635
Adjusted_R2 : 0.987679588838164

Testing Data

MSE : 38346.11644521783
RMSE : 195.82164447582863
MAE : 111.61475186737812
R2 : 0.9083774269225618
Adjusted R2 : 0.9055732045564484



Evaluation Metrics - RF Gridsearch CV

Training Data

```
Model Score: 0.7853321986045929
MSE : 114972.78162227821
RMSE : 339.0763654728507
MAE : 225.0484408092352
R2 : 0.7231192918198736
Adjusted_R2 : 0.7146450147007644
```

Testing Data

```
MSE : 126863.72747761484
RMSE : 356.17934734851605
MAE : 231.18930377601367
R2 : 0.6968772272336967
Adjusted R2 : 0.6875997792149517
```

Best parameters by Gridsearch CV

```
[138] rf_grid.best_params_

{'max_depth': 8,
 'min_samples_leaf': 40,
 'min_samples_split': 50,
 'n_estimators': 50}
```

Evaluation Metrics - Gradient Boost

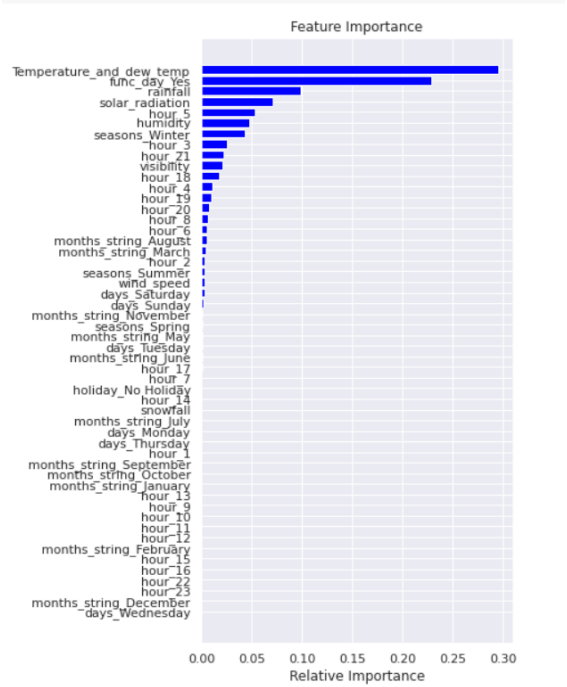
Training Data

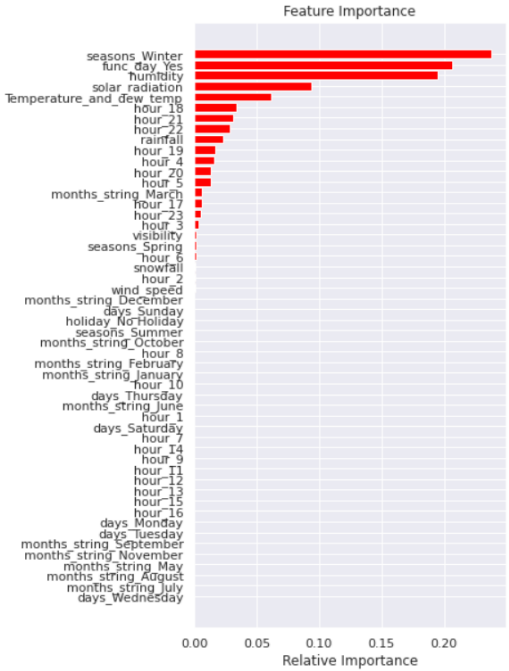
```
Model Score: 0.9500234383572045
MSE : 24181.081510888762
RMSE : 155.50267364546747
MAE : 96.60380653555514
R2 : 0.941766434813308
Adjusted R2 : 0.9399841244014728
```

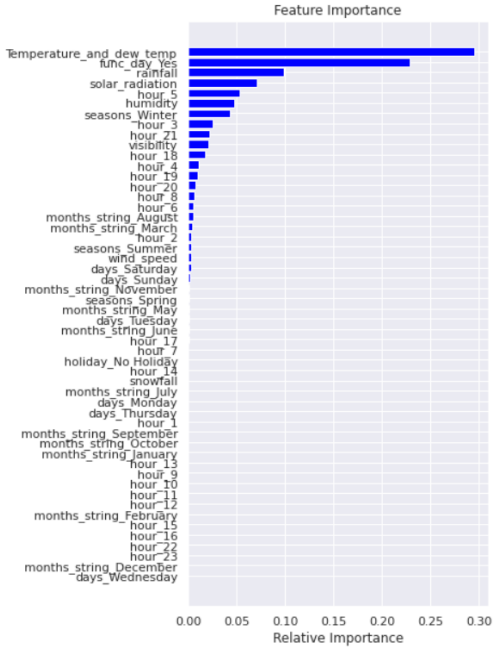
Testing Data

```
MSE : 39358.19179686625
RMSE : 198.3889911181219
MAE : 119.91132705642147
R2 : 0.905959217297637 8
Adjusted R2 : 0.9030809826298787
```

Feature Importance(RF vs RF Gridsearch CV vs Gradient Boosting)







Summary

		Model	MAE	MSE	RMSE	R2_score	Adjusted R2
Training set	**0**	Linear regression train	215.732	100605.912	317.184	0.758	0.750
	1	Lasso Regression train	215.864	100800.303	317.491	0.757	0.750
	2	Ridge regression train	215.880	100807.652	317.502	0.757	0.750
	3	Elastic net regression train	215.738	100613.776	317.197	0.758	0.750
	4	Decision tree train	237.007	129012.399	359.183	0.689	0.670
	5	Random forest regression	41.220	4964.034	70.456	0.988	0.990
	6	Random forest regression train with GridCV	225.048	114972.782	339.076	0.723	0.710
	7	Gradient Boosting train gridsearchcv	96.604	24181.082	155.503	0.942	0.940
Test set	**0**	Linear regression test	212.052	98047.314	313.125	0.766	0.760
	1	Lasso regression test	212.203	98235.175	313.425	0.765	0.760
	2	Ridge regression test	212.234	98257.409	313.460	0.765	0.780
	3	Elastic net regression Test	212.060	98055.505	313.138	0.766	0.759
	4	Decision tree test	240.900	133881.930	365.899	0.680	0.670
	5	Random forest regression	111.615	38346.116	195.822	0.908	0.906
	6	Random forest regression test with GridCV	231.189	126863.727	356.179	0.697	0.688
	7	Gradient Boosting gridsearchcv	119.911	39358.192	198.389	0.906	0.900

Conclusion

1. Almost all algorithms performed really well on both training dataset and testing dataset so we can say that variance is less and no issues of overfitting are present.

1. Both "Random forest regression" and "Gradient Boosting regression(GridSearch cv) has highest R2 score of 98.8% and 94% respectively.

1. Performance on "Decision Tree" algorithm is comparatively less with an R2 score of 68%.

We know that this data is time dependent, the values for variables like temperature, solar_radiation, wind_speed etc., will not always be consistent. Therefore, there will be scenarios where the model might not perform well. As Machine learning is an exponentially evolving field, we will have to be prepared for all contingencies and also keep checking our model from time to time. Therefore, having a quality knowledge and keeping pace with the ever evolving ML field would surely help one to stay a step ahead in future.

THANK YOU