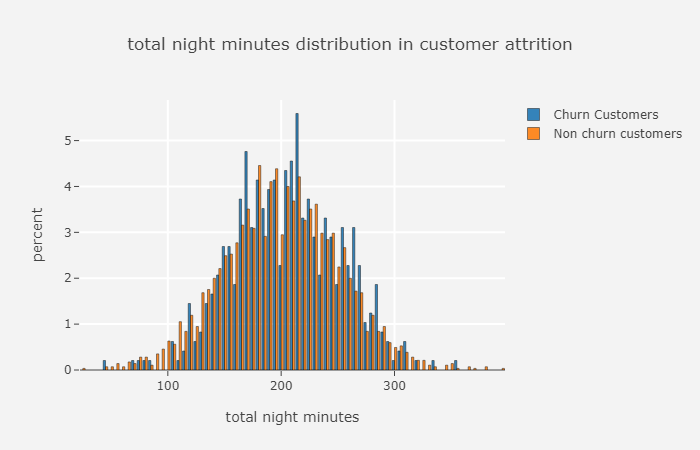
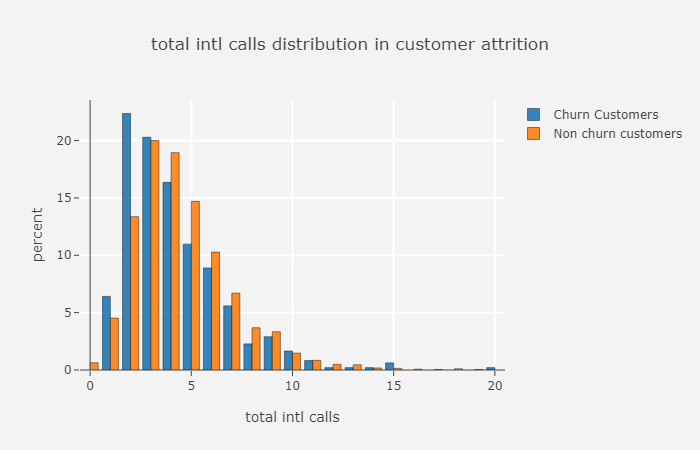
Data Fields - 

season - 1 = spring, 2 = summer, 3 = fall, 4 = winter  
holiday - whether the day is considered a holiday  
workingday - whether the day is neither a weekend or holiday  
weather -  
1: Clear, Few clouds, Partly cloudy, Partly cloudy  
2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist  
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds  
4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog  
temp - temperature in Celsius  
atemp - "feels like" temperature in Celsius  
hum- relative humidity  
windspeed - wind speed  
casual - number of non-registered user rentals initiated  
registered - number of registered user rentals initiated  
cnt - number of total rentals (Dependent Variable)

## Data Summary

As a first step lets do three simple steps on the dataset

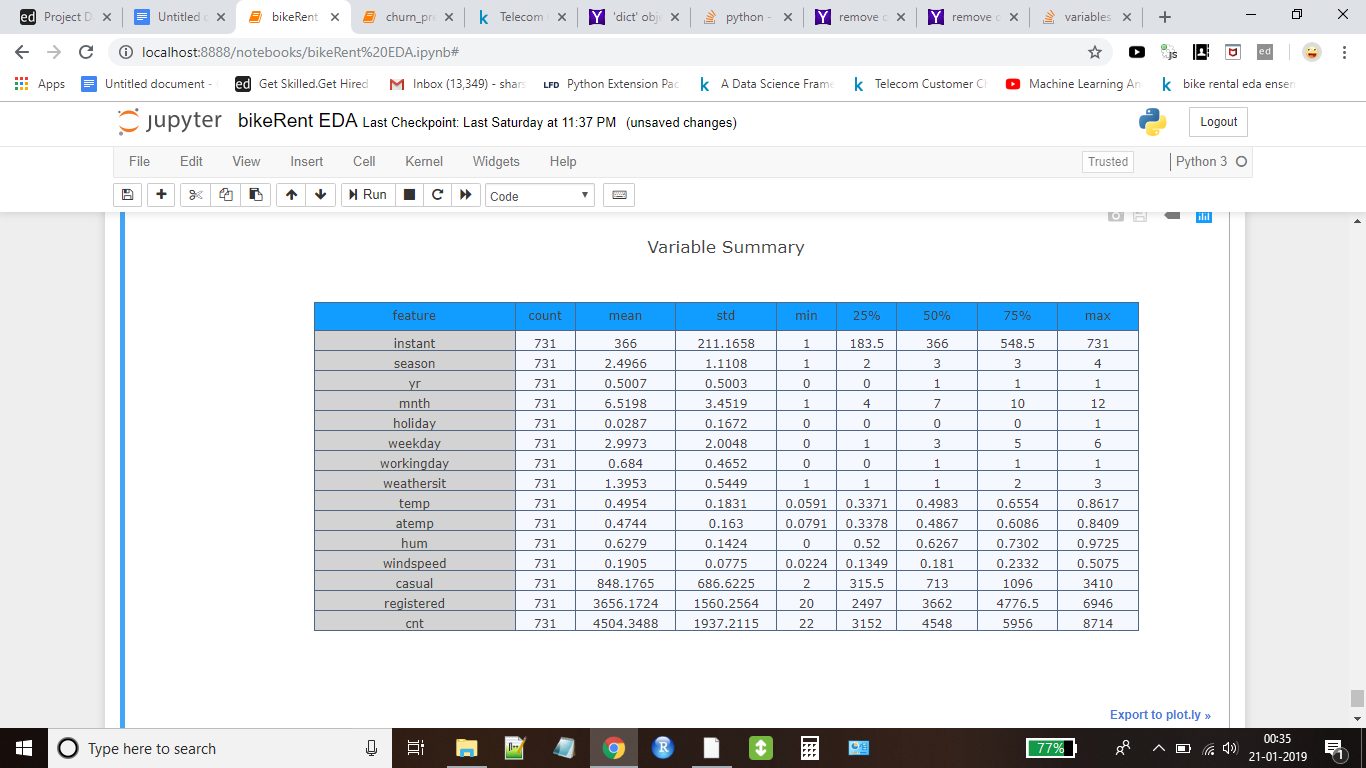
* Size of the dataset
* Get a glimpse of data by printing few rows of it.
* What type of variables contribute our data

Shape of data : 731 rows , 16 columns

#### **Sample Of First Few Rows**

#### **Variables Data Type**

instant int64  
dteday object  
season int64  
yr int64  
mnth int64  
holiday int64  
weekday int64  
workingday int64  
weathersit int64  
temp float64  
atemp float64  
hum float64  
windspeed float64  
casual int64  
registered int64  
cnt int64  
dtype: object



## **Feature Engineering**

As we see from the above results, the columns "season","holiday","workingday" and "weathersit" should be of "categorical" data type.But the current data type is "int" for those columns. We will transform the dataset in the following ways so that we can get started up with our EDA

Coerce the datatype of "season","holiday","workingday" and weather to category

We don't require dteday column as we already have separate columns as “yr” for year and “mnth” for months . We also don't require instant column , we will drop both of these from our dataset

## **Missing Values Analysis**

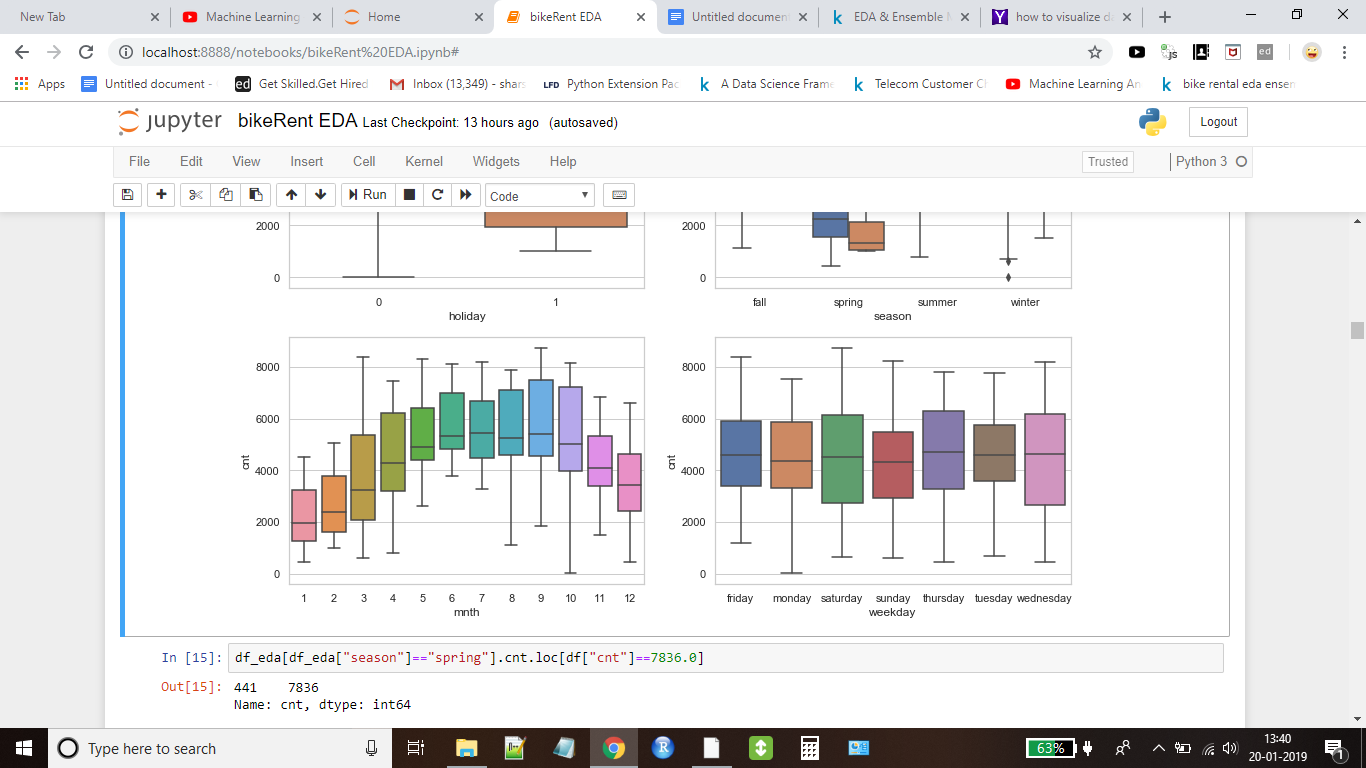
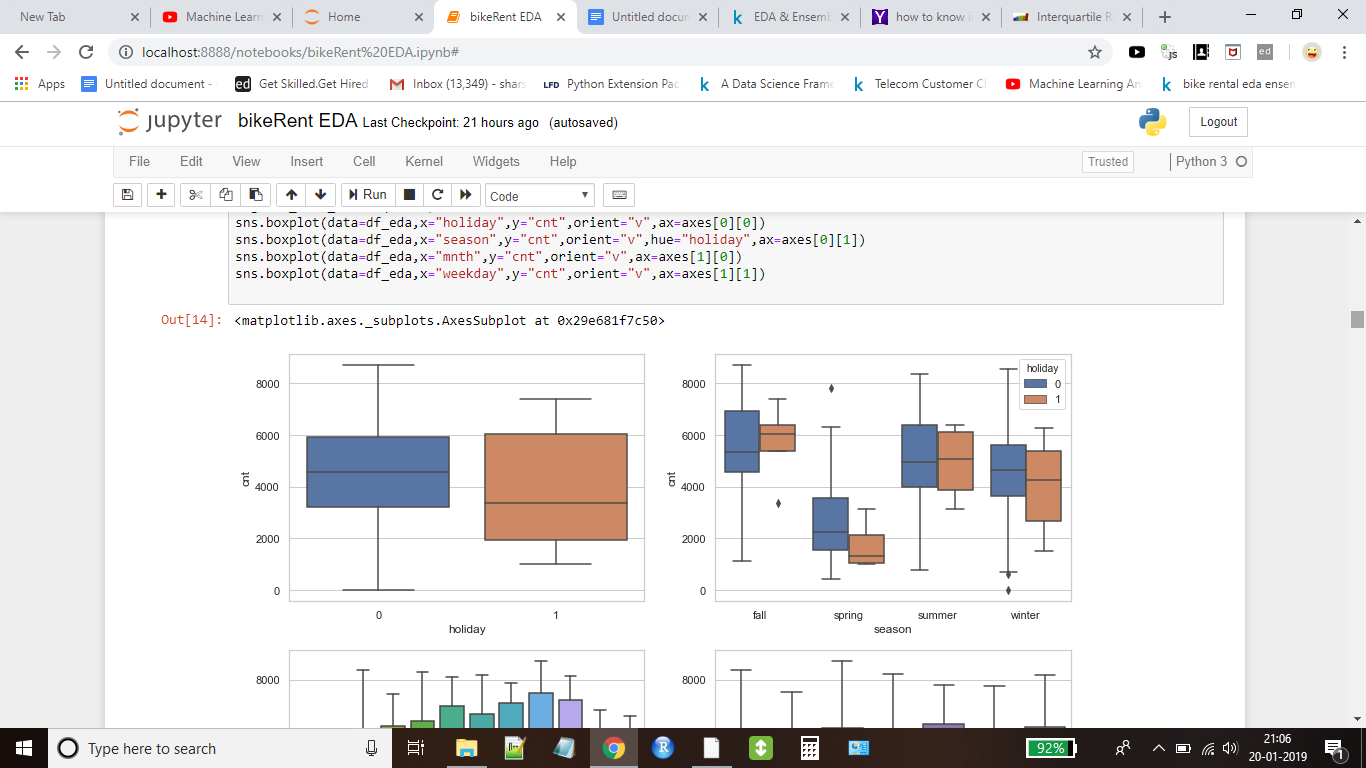
Once we get hang of the data and columns, next step we generally is to find out whether we have any missing values in our data. Luckily we don't have any missing value in the dataset.

## **Outliers Analysis**

We can see that there are only few outliers present in count variable with respect to season .

Following inferences can also been made from the simple box plots given below.

Spring season has got relatively lower count.The dip in median value in boxplot gives evidence for it.The boxplot with “Month” is quite interesting.The median value are relatively higher from august to september .



## 

We can infer from the figure above is that

In year 2012 , count value is more relatively than 2011

Value of count is comparatively lesser in the case when weather represents light snow,rain and thunderstorm , as it is quite obvious in extreme weather ,demand drops

There is lesser demand in case of holiday ..

## 

## 

## Now will see how the data is distributed in other independent variables

## 

## 

## 

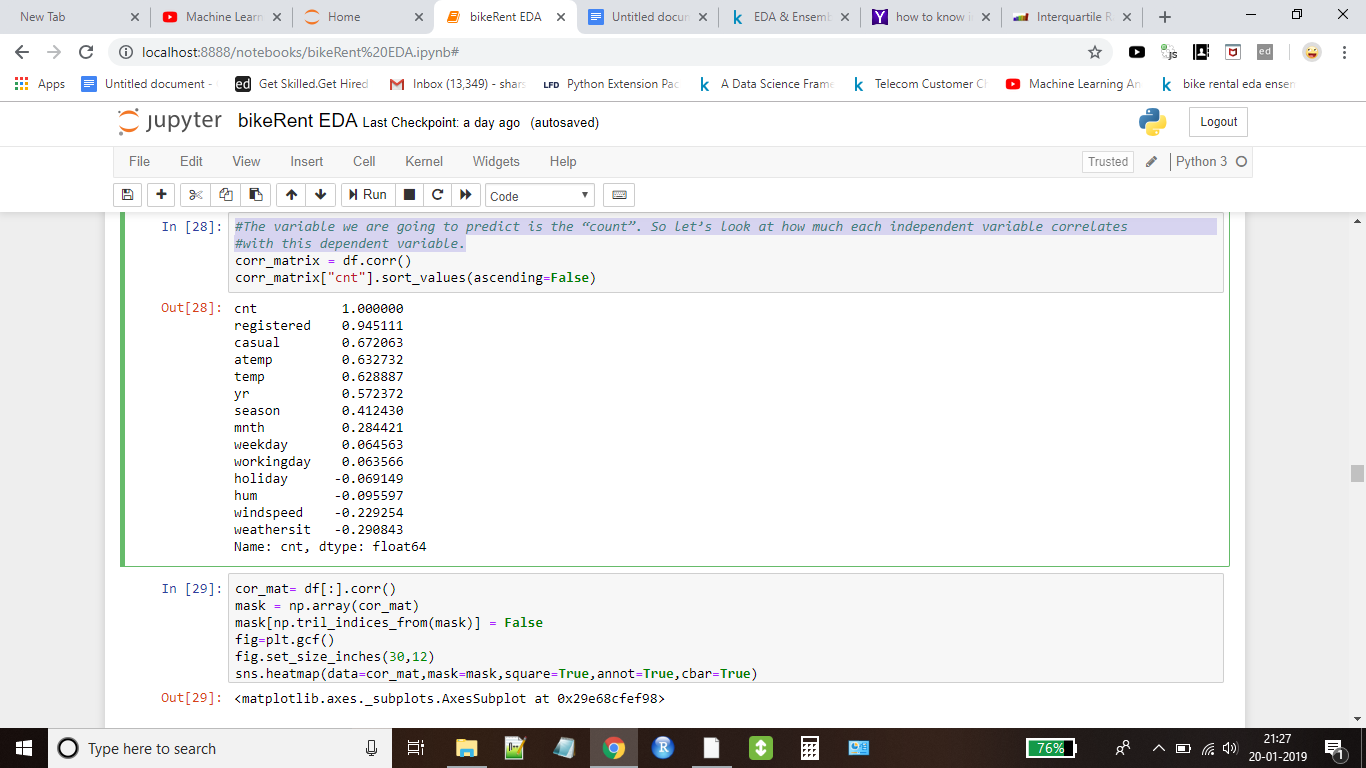
## 

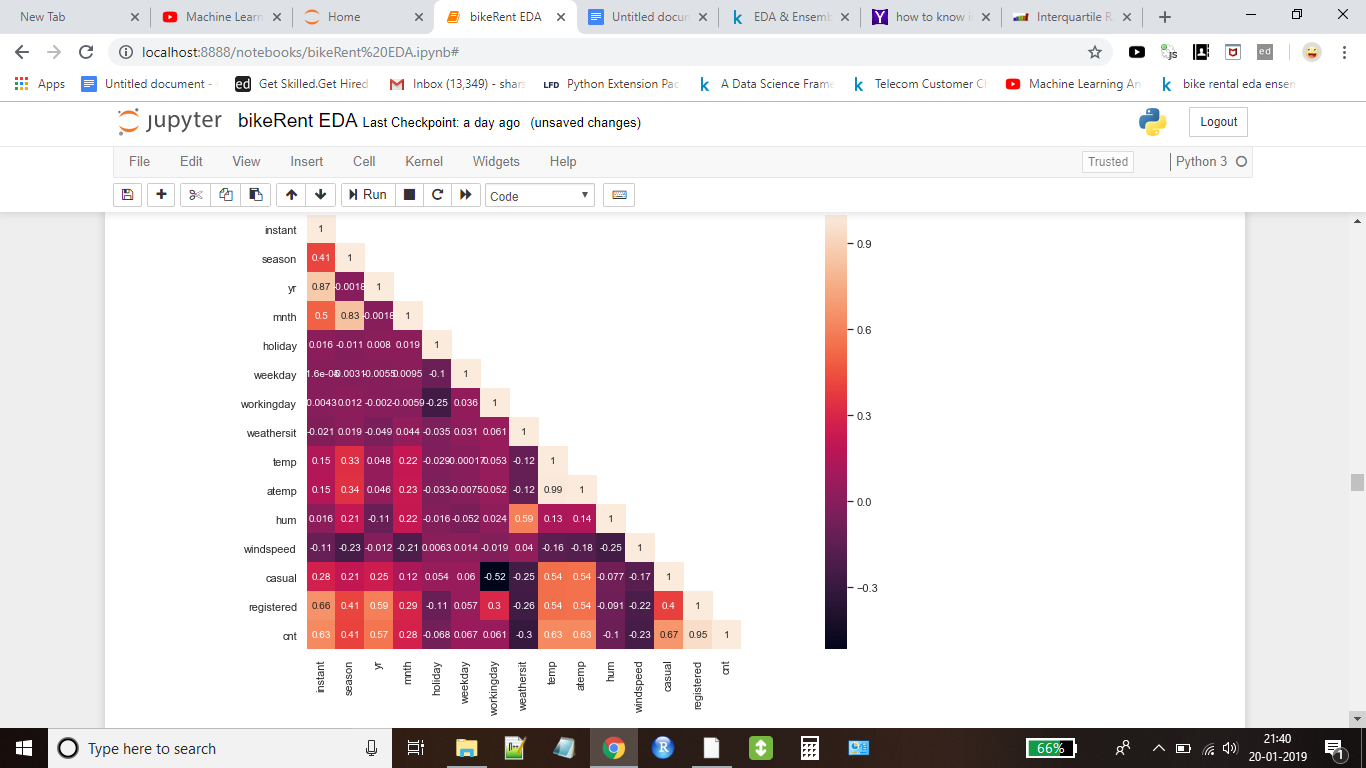
## 

## **Correlation Analysis**

One common to understand how a dependent variable is influenced by features (numerical) is to find a correlation matrix between them. Lets plot a correlation plot

The variable we are going to predict is the “count”. So let’s look at how much each independent variable correlates with this dependent variable.

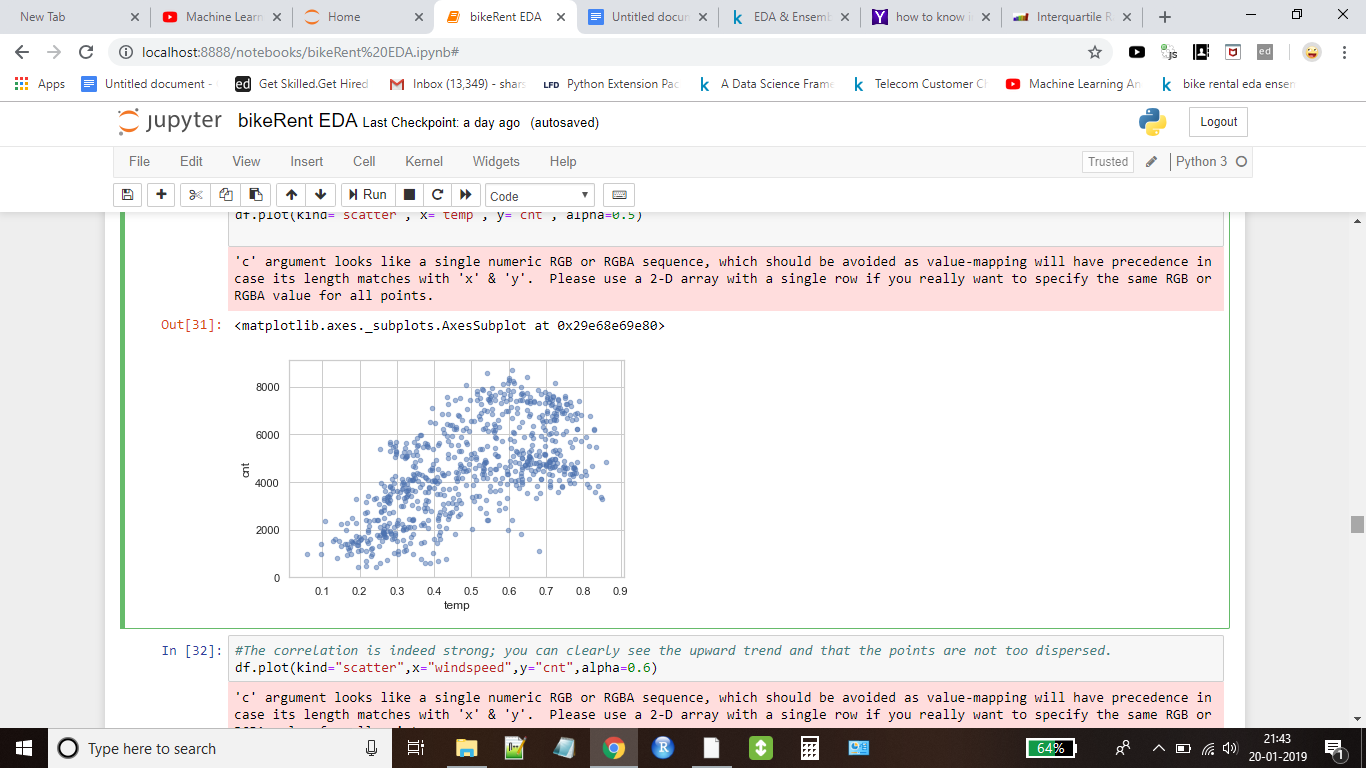


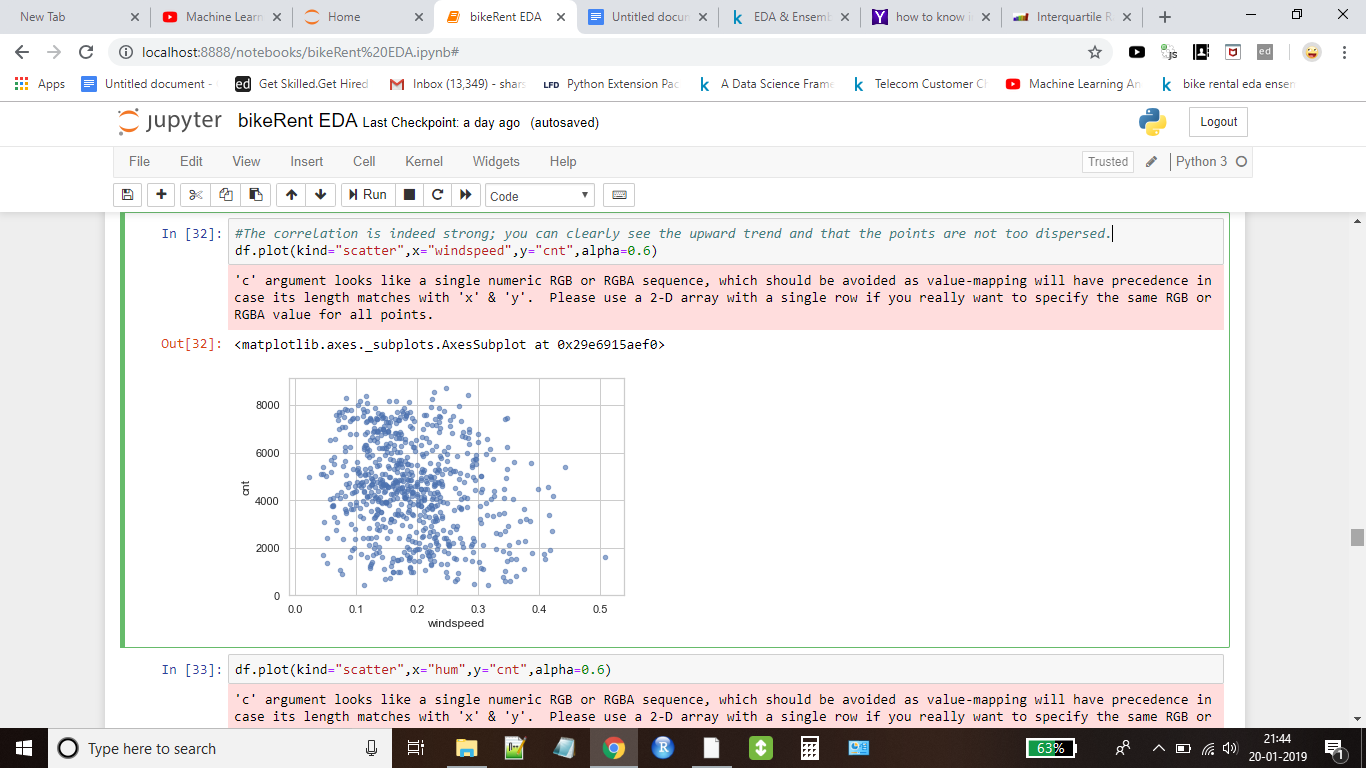
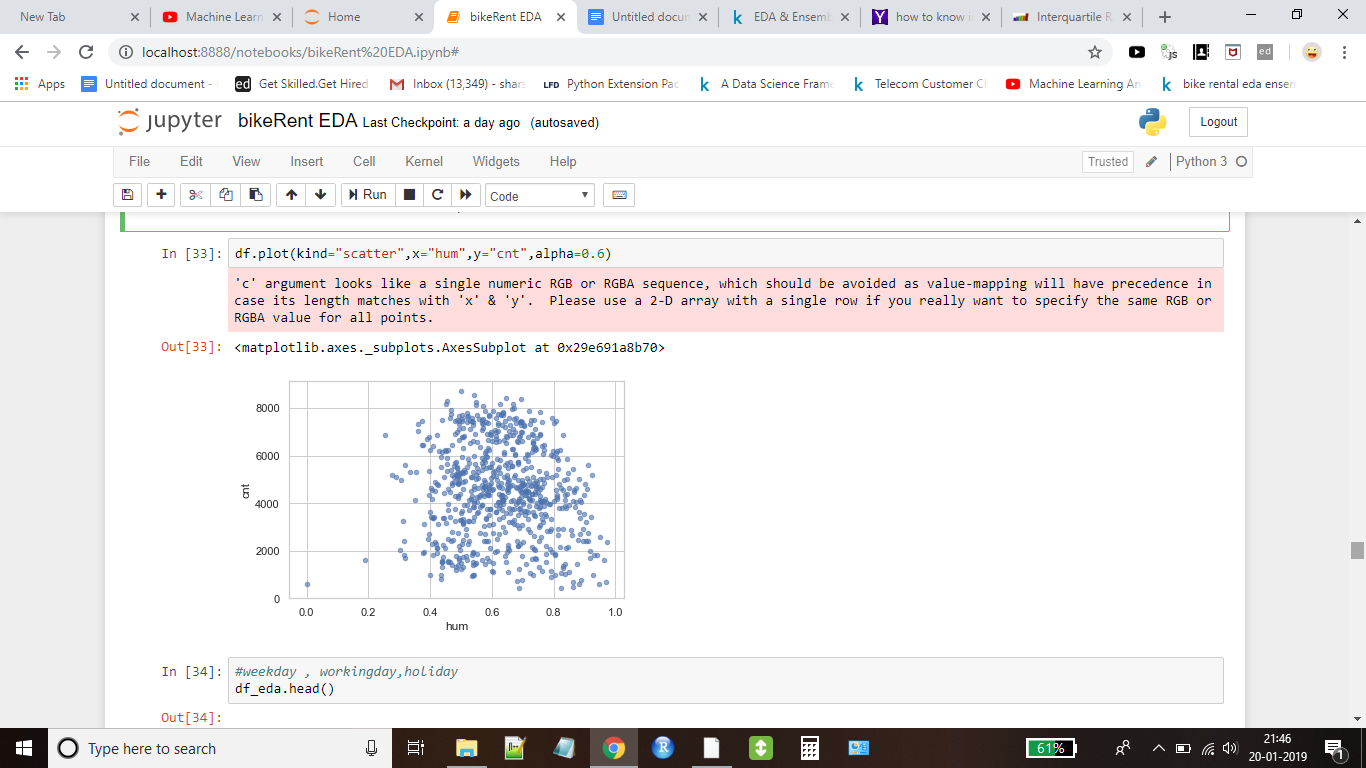


We can see that there is pretty much strong correlation of count with respect to atemp,temp,yr,season and month. We will ignore casual and registered variables as they are directly related with count variable. Windspeed shows negative correlation as it is obvious with high value of windspeed , demand will definitely drop. Same is the case with weather variable.

Now we will see how these variables are related with respect to each other.

temp and atemp are highly correlated,correlation value is 0.99. We will drop atemp variable as both are same.

The most promising variable for predicting the count is the temp, so lets look at their correlation scatter plot.



The correlation is indeed strong; you can clearly see the upward trend and that the points are not too dispersed. In case of humidity and windspeed , data is too much dispersed , no correlation at all.

**Model building**

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.3,random\_state=30)

Shape of train and test dataset

Train data (511, 10)  
Test data (219, 10)

#**Linear regression**

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None, normalize=True)

R squared value :: 0.8113

Linear Regression Root Mean Squared Error : 831.9710

Linear Regression Root Mean Squared Log Error: 0.2442

Linear Regression Mean Absolute Error : 617.5498

# **Lasso** (**least absolute shrinkage and selection operator**)

Lasso(alpha=1.0, copy\_X=True, fit\_intercept=True, max\_iter=1000,  
 normalize=False, positive=False, precompute=False, random\_state=None,  
 selection='cyclic', tol=0.0001, warm\_start=False)

R squared value :: 0.810241940891103

Lasso Root Mean Squared Error : 834.3916

Lasso Root Mean Squared Log Error: 0.2455

Lasso Mean Absolute Error : 619.674127

**#Gradient boost Regressor**

GradientBoostingRegressor(alpha=0.9, criterion='friedman\_mse', init=None,  
 learning\_rate=0.1, loss='ls', max\_depth=3, max\_features=None,  
 max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,  
 min\_impurity\_split=None, min\_samples\_leaf=1,  
 min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,  
 n\_estimators=100, n\_iter\_no\_change=None, presort='auto',  
 random\_state=None, subsample=1.0, tol=0.0001,  
 validation\_fraction=0.1, verbose=0, warm\_start=False)

Gradient boost regressor R squared value :: 0.8910702019919841

Gradient boost regressor Root Mean Squared Error : 632.1835049178984

Gradient boost regressor Root Mean Squared Log Error: 0.2096

Gradient boost regressor Mean Absolute Error : 461.9865

**#Regularization Model - Ridge**

We will use the grid search also along with this.We have given a list of values for alpha parameter.

ridge\_params\_= {'max\_iter' : [3000] , 'alpha' : [0.1, 1, 2, 3, 4, 10, 30,100,200,300,400,800,900,1000]}

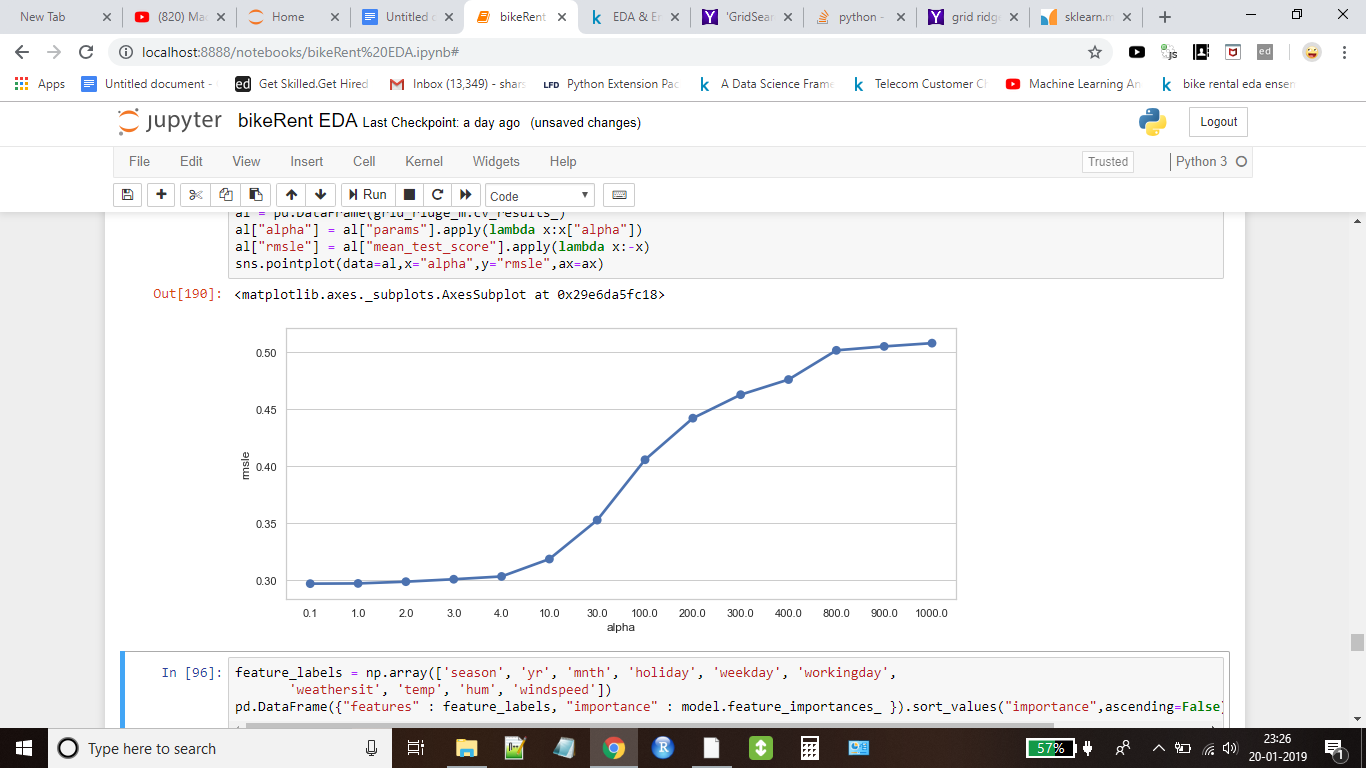
After fitting model to the data, we get

**Best Estimators**

Ridge(**alpha=0.1**, copy\_X=True, fit\_intercept=True, **max\_iter=3000**,  
 normalize=False, random\_state=None, solver='auto', tol=0.001)

Ridge Root Mean Squared Log Error : 0.24499171538368814

Model has also given the best value of alpha parameter and that is 0.1

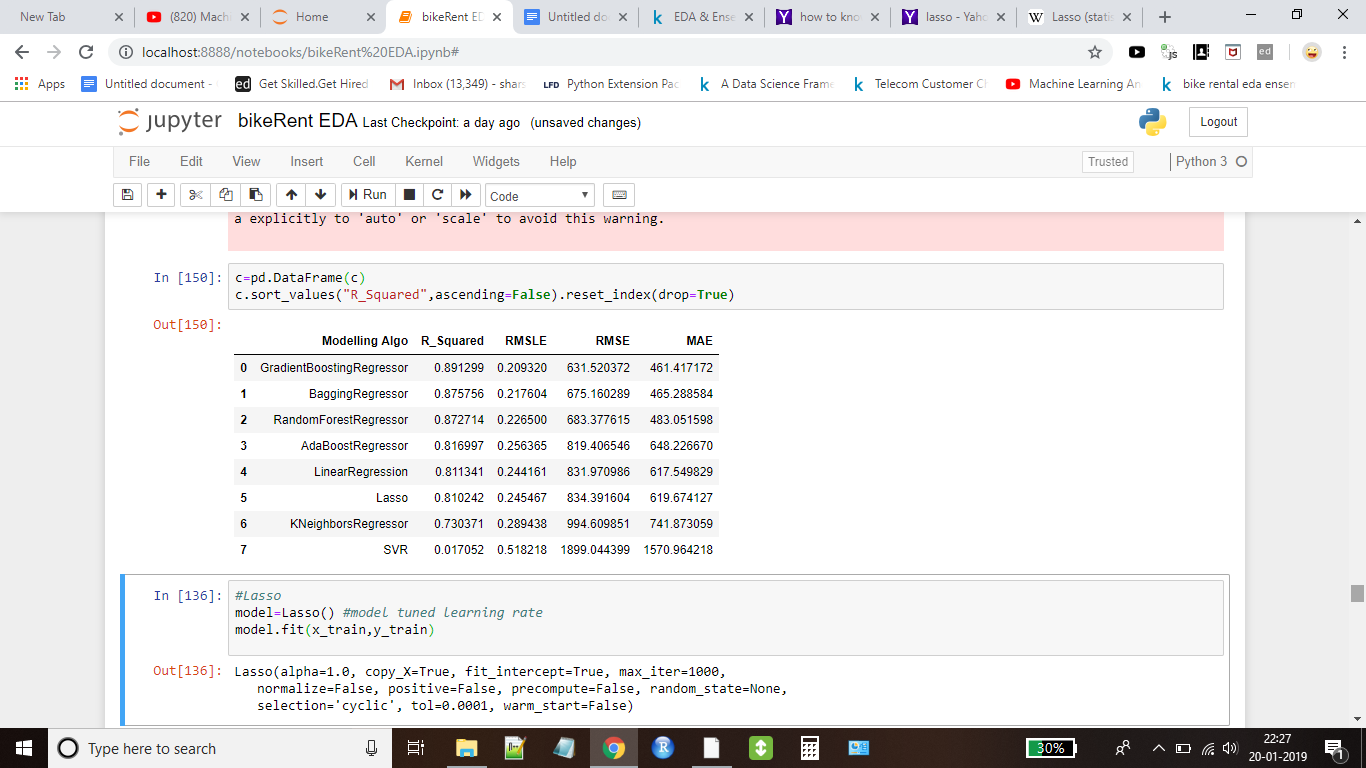


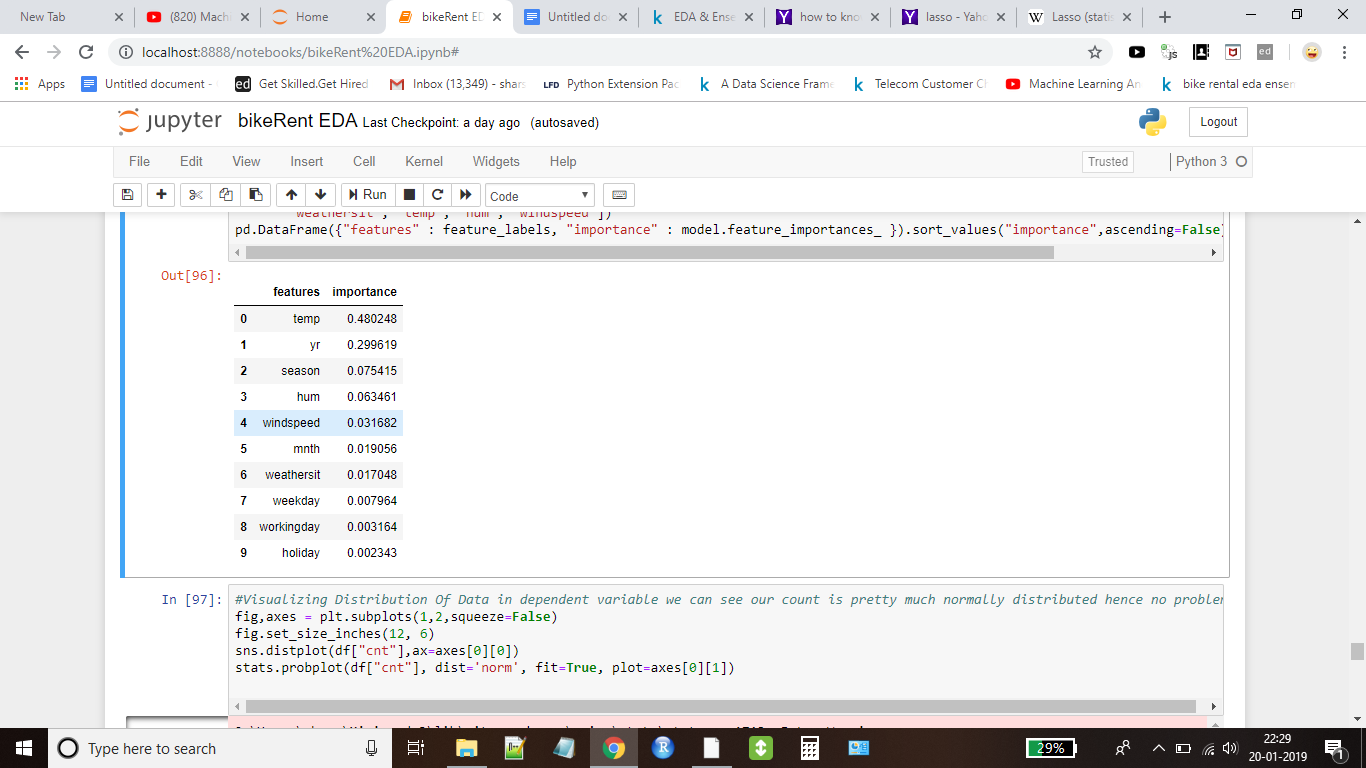
From the above figure , it is clear that for alpha = 0.1 , rmsle value is the lowest.

We have applied some other model also like **RandomForestRegressor,AdaBoostRegressor,BaggingRegressor,SVR,**

**KNeighborsRegressor**

And we have framed the results in data frame and sorted the values based on greater R\_Squared values. We have also plotted feature importance alongside , once again it is proved that temp has contributed a lot in predicting count values.





From the above figure , it is clear that Gradient Boosting Regressor is predicting bike rental count values with least error. We will try to tune the parameters of model and optimize it.

**Model Tuning ( Gradient Boosting Regressor)**

Now we will tune the parameters of gradient boost regressor to help optimize our model

We will set learning rate = 0.1904 and random\_state=30 . We will see whether this improves the model statistics metrics or not

GradientBoostingRegressor(alpha=0.9, criterion='friedman\_mse', init=None,  
 **learning\_rate=0.1904**, loss='ls', max\_depth=3,  
 max\_features=None, max\_leaf\_nodes=None,  
 min\_impurity\_decrease=0.0, min\_impurity\_split=None,  
 min\_samples\_leaf=1, min\_samples\_split=2,  
 min\_weight\_fraction\_leaf=0.0, n\_estimators=100,  
 n\_iter\_no\_change=None, presort='auto', **random\_state=30**,  
 subsample=1.0, tol=0.0001, validation\_fraction=0.1, verbose=0,  
 warm\_start=False)

By fitting the model again with data and new parameters , we get the following values :

**New Values** **Old Values**

**R squared value** :: 0.9023663611875501 0.8910702019919841

**Root Mean Squared Error** : 598.5074334190634 632.1835049178984

**Root Mean Squared Log Error**: 0.2002 0.2096

**Mean Absolute Error** : 438.9757 461.9865

We can see that MSE, RMSLE and MAE now have reduced value, that means our tuned model is predicting with less error. This is the best we can get from the model with new parameters.

So we have build several models , and we have chosen the **Gradient Boosting Regressor** as the best model that is going to predict bike rental count with least error.

**Model Performance Comparison**

