BIKE RENTAL PROJECT

-By,
Jeevan Hemmanna B.


Rent a bike
Pincasso
shutterstock
IMAGE ID: 151664957
www.shutterstock.com

CONTENTS

1. **Problem Statement**: The Bike Rental Data contains the daily count of rental bikes between the year 2011 and 2012 with corresponding weather and seasonal information. We would like to predict the daily count of rental count in order to automate the system.

2. **Data used:** Data used here should be used to build a regression model which can predict the number of bikes that would be rented on a day by analyzing the weather and season and given below is the data snapshot of the data.

```
bike_df=pd.read_csv("day.csv")

#Print the `head` of the data
bike_df.head(10)
```

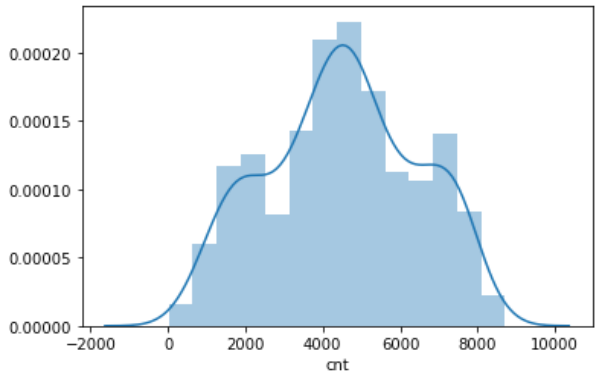
	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600
5	6	2011-01-06	1	0	1	0	4	1	1	0.204348	0.233209	0.518261	0.089565	88	1518	1606
6	7	2011-01-07	1	0	1	0	5	1	2	0.196522	0.208839	0.498696	0.168726	148	1362	1510
7	8	2011-01-08	1	0	1	0	6	0	2	0.165000	0.162254	0.535833	0.266804	68	891	959
8	9	2011-01-09	1	0	1	0	0	0	1	0.138333	0.116175	0.434167	0.361950	54	768	822
9	10	2011-01-10	1	0	1	0	1	1	1	0.150833	0.150888	0.482917	0.223267	41	1280	1321

Below is the variable used in the data –

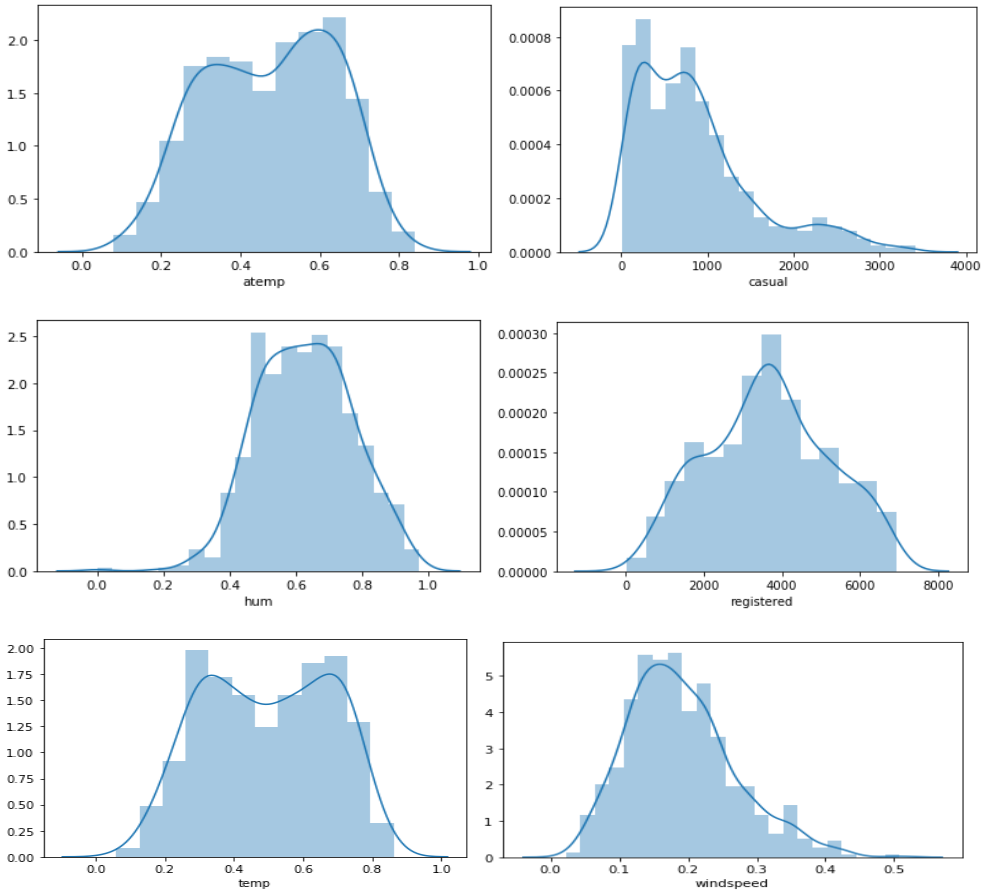
s.no	Variables
1	Dteday
2	Season
3	Yr
4	Mnth
5	Holiday
6	Weekday
7	workingday
8	weathersit
9	Temp
10	Atemp
11	Hum
12	windspeed
13	Casual
14	registered

3. **<u>Data Preprocessing</u>:** For any modeling to be done on this data first we need to preprocess the data this process includes analysis of the variables , establishing the target variable, outlier Analysis, Feature selection and Feature scaling then we can apply the regression modeling and find which model have a good accuracy in prediction.

   - **Univariant Analysis:** For this Analysis we plot a density graph and find normality of the variable.

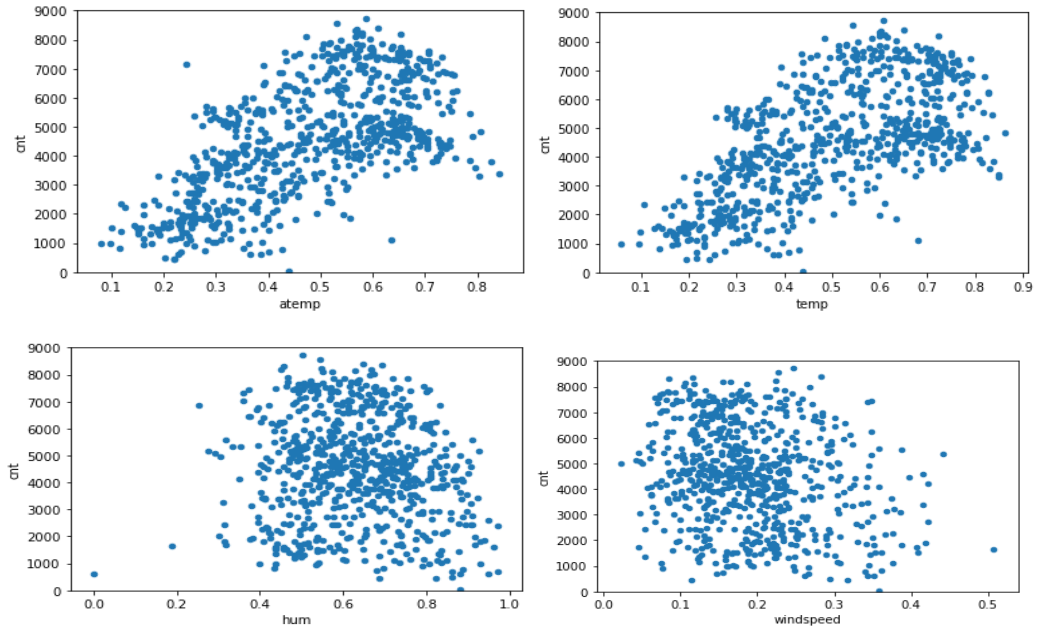


From the graph we can see that cnt (target variable) is normally distributed.



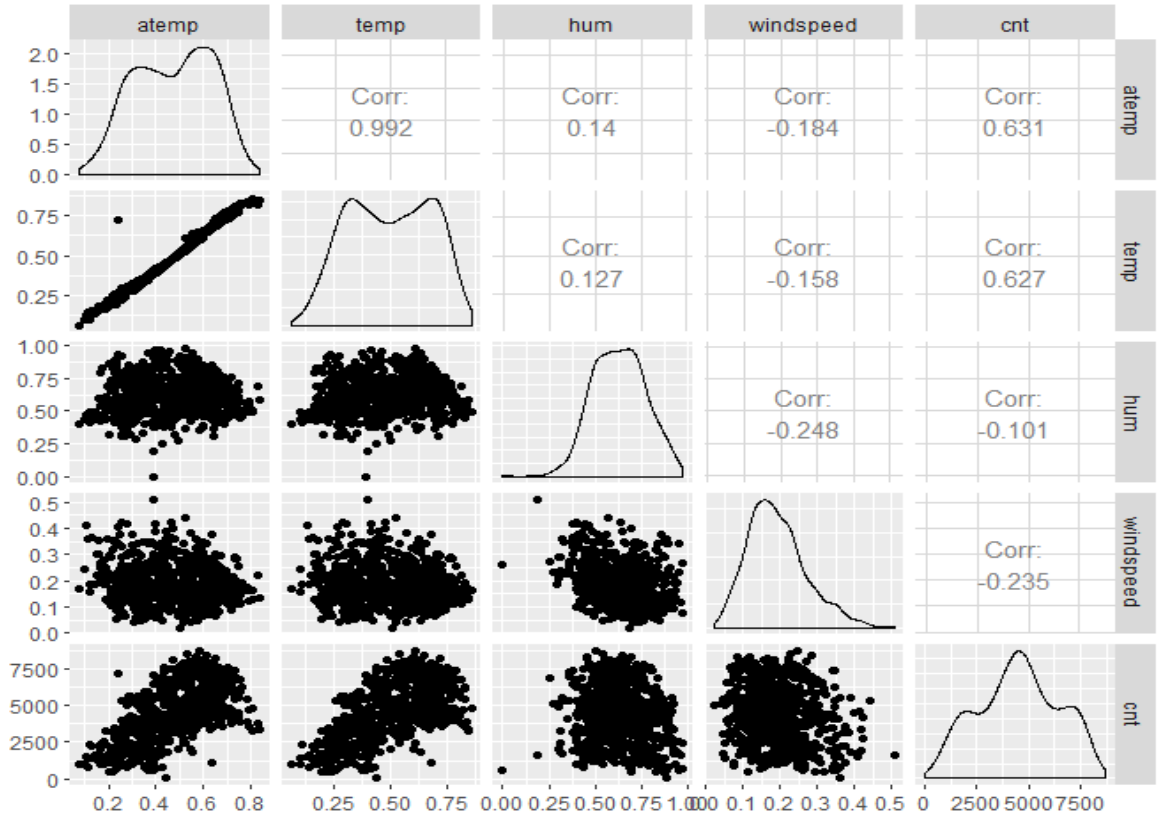
From the graph we can see that temp, atemp, and registered are normally distributed. And casual is skewed to right and there might be outlier in this data. And hum is skewed left and this data seems to be normalized.

- **Bivariant Analysis:** we use bivariant analysis for finding weather the target variable and numerical independent variable related.



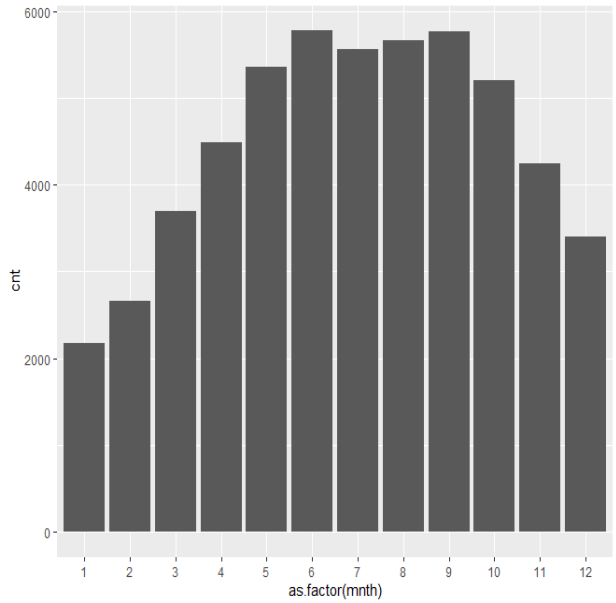
From the graph we can see that temp and atemp has a very good relationship between cnt and windspeed and cnt does not have a good relationship and hum has a average relationship.

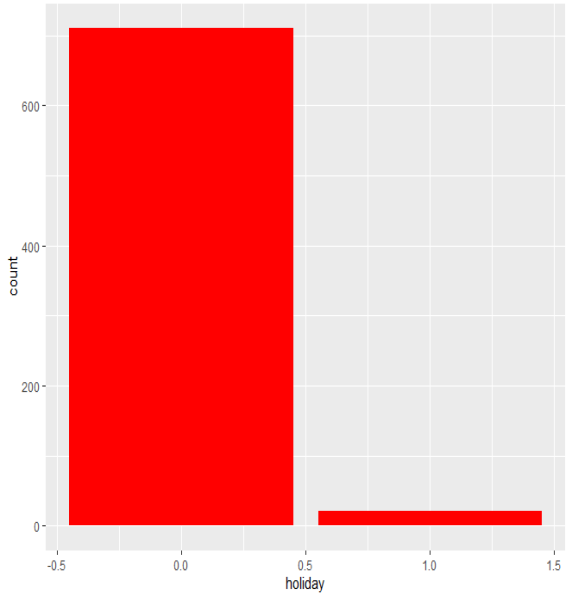
Let's see the same thing in the ggpair plot.



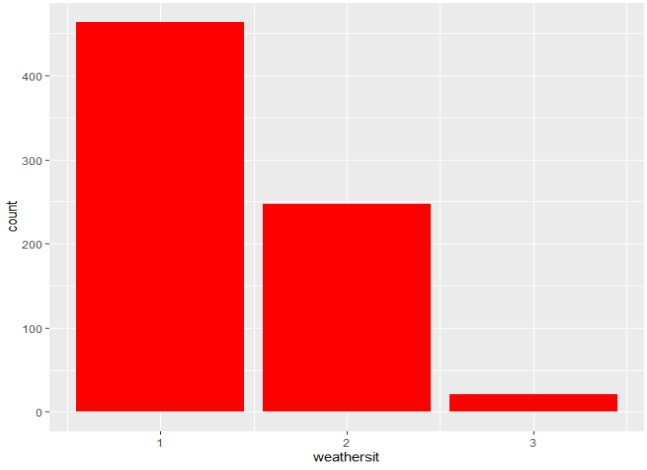
From the plot we can see temp and atemp has a very good relationship.

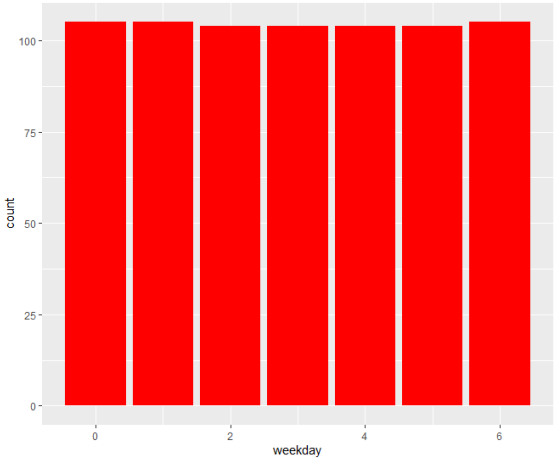
Let's see for the categorical independent variables -





For month we can see that June has the greatest number of bike rentals and also, we can see from the holiday graph that highest bike rentals happen on holidays.





From the weathersit graph we can see that weather type 1 that is "Clear, few clouds, partly cloudy, partly cloudy" has the greatest number of bike rentals. and from weekday graph we can see that the rental count is almost same on all the weekdays.

- **Missing value Analysis:** Missing value Analysis is one of the most common analysis that needs to be done on any data set and we do it here also, this we do in order to reduce the bias and also produce a strong model.

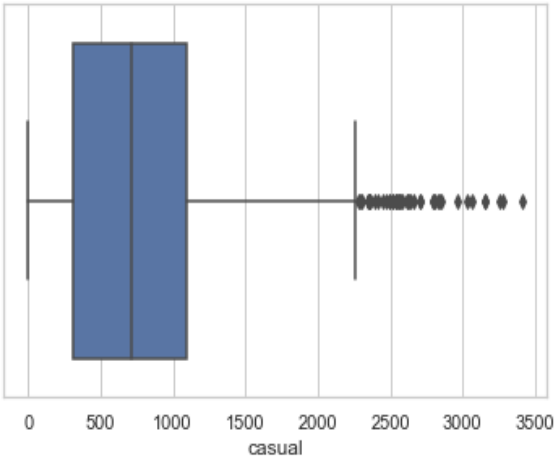
  Below is the table for missing value analysis:

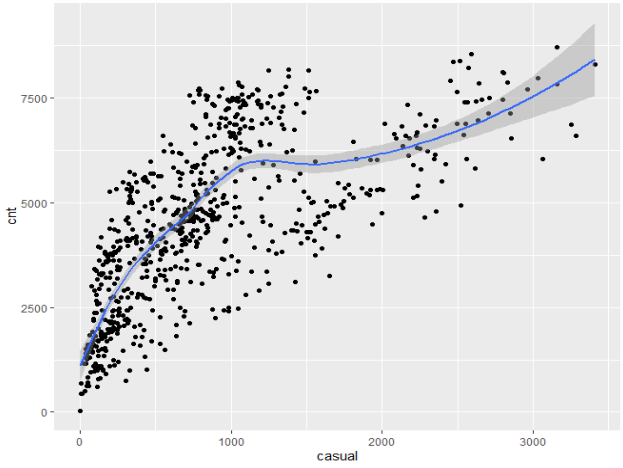
```
total_missing_values = bike_df.isnull().sum().sort_values(ascending=False)
total_missing_values
```

```
cnt           0
registered    0
casual        0
windspeed     0
hum           0
atemp         0
temp          0
weathersit    0
workingday    0
weekday       0
holiday       0
mnth          0
yr            0
season        0
dteday        0
instant       0
dtype: int64
```

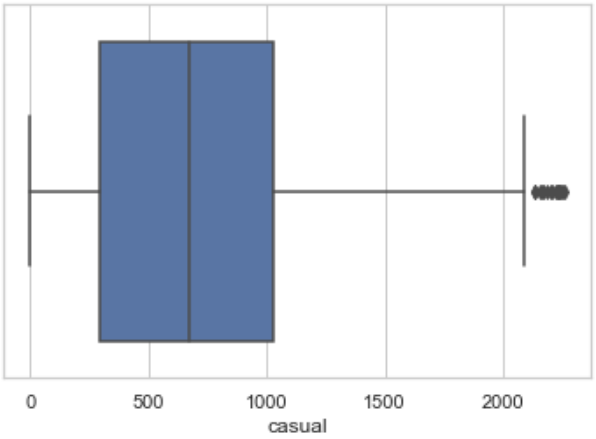
- **Outlier Analysis** - The Other steps of Preprocessing Technique is Outliers analysis; an outlier is an observation point that is distant from other observations. Outliers in data can distort predictions and affect the accuracy, if you don't detect and handle them appropriately especially in regression models.

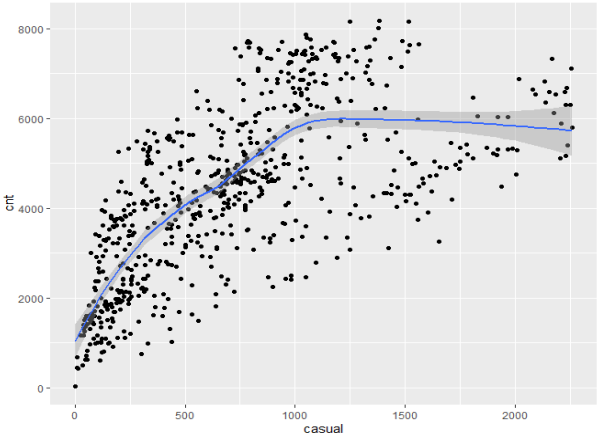
  As we are observed that from the above figure the data is skewed so, there is chance of outlier in independent variable 'casual', one of the best methods to detect outliers is Boxplot.





from the above figures we can see that presence of Outliers in variable 'casual' and relationship between 'casual' and 'cnt' before removing Outliers. And parson correlation value before outlier removal is 0.6728044333386834





From the above figures we can see boxplot of 'casual' after removing outliers and relationship between 'casual' and 'cnt' after removing outliers. And parson correlation value after outlier removal is 0.6460020508747337

Since there is significant difference between Pearson coefficient correlation between before and after outlier detection for 'casual' and 'cnt' and losing nearly 40 observation so, we are not going to treat the outliers.

- **Feature Selection and dimensionality reduction -** In the part we select only required variable that has most effect in our model building. For this we have this following the criteria – The relationship between 2 independent variables should be less and relationship between independent and target variable should be high.

	temp	atemp	hum	windspeed	casual	registered	cnt
temp	1.0	0.99	0.13	-0.16	0.54	0.54	0.63
atemp	0.99	1.0	0.14	-0.18	0.54	0.54	0.63
hum	0.13	0.14	1.0	-0.25	-0.077	-0.091	-0.1
windspeed	-0.16	-0.18	-0.25	1.0	-0.17	-0.22	-0.23
casual	0.54	0.54	-0.077	-0.17	1.0	0.4	0.67
registered	0.54	0.54	-0.091	-0.22	0.4	1.0	0.95
cnt	0.63	0.63	-0.1	-0.23	0.67	0.95	1.0

From the correlation plot we can see that there is a strong relationship between "temp" and "atemp" so in this case we can choose any one of the variable and drop one variable and also there is very negative correlation between "cnt" (target variable) and "hum" so we can drop "hum" variable also. So, after this from the dataset now we can drop "hum" and "atemp" variable from the dataset.

- **Feature Scaling –** From the dataset we can see that "casual" and "registered" variable are not scaled so for the model to scale these two variables we use normalization method. After this we can see that the data point will be between 0 and 1. Below is the formula for normalization.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

below is a snapshot after normalization.

casual	registered
0.096537559	0.09153913
0.037852113	0.09384926
0.034624413	0.17455963
0.031103286	0.20704591
0.023474178	0.21628646
0.025234742	0.21628646
0.042840376	0.19376263

4. **Building Predictive Models** – We will be using regression model in order predict the bike rental count for a day. We will be using following Random forest, decision tree and liner regression and after that we will be selecting a model which has better accuracy out of all three model.

- **Evaluating the model –** Here we will be using a concept called residual, which is basically difference between actual value of the targeted variable to the predicted value.

  We will be using 2 method for evaluating our model –

  - **MAPE -** (Mean Absolute Percentage Error) measures the size of the error in percentage terms. It usually expresses the accuracy as a ratio defined by the formula:

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|,$$

  - **RMSE –** (Root Mean Square Error) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. This is calculated by a formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

The function is used in the code is below –

```
#Calculate MAPE
def MAPE(y_true, y_pred):
    mape = np.mean(np.abs((y_true - y_pred) / y_true))*100
    print(f'MAPE : {mape}')

def RMSE(y_test,y_predict):
    mse = np.mean((y_test-y_predict)**2)
    print("Mean Square : ",mse)
    rmse=np.sqrt(mse)
    print("Root Mean Square : ",rmse)
```

- **Decision Tree -** A tree has many analogies in real life and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.

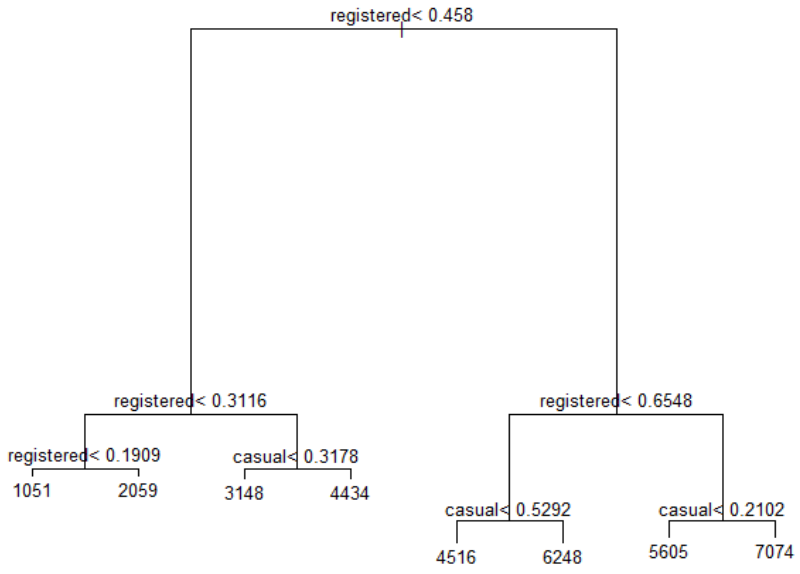
  Below is the code that is written for the Decision tree.

```
from sklearn.tree import DecisionTreeRegressor

DT1 = DecisionTreeRegressor()
DT1 = DT1.fit(bike_train_feature, bike_train_target)
print(DT1)
```

```
DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=None,
                      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, presort='deprecated',
                      random_state=None, splitter='best')
```

Below is the graphical representation of the decision tree.



Here we can see that the decision tree has mostly 2 independent variable “registered” and “casual” so we can say that the model is biased towards the 2-independent variable and so, from this we can see that this model is not so good.

- **Model Evaluation –** From the below figure we can see the performance of the model.

```
#MAPE
MAPE(bike_test_target,prediction_DT)

#RMSE
RMSE(bike_test_target,prediction_DT)
```

```
MAPE : 3.30962390430595
Mean Square :  31036.299319727892
Root Mean Square :  176.17122159912466
```

From the model performance Evaluation, we can see that MAPE is 3.3 and we can calculate model accuracy by 100 – MAPE which is around 96 percent and RMSE is 176 which is high and which clearly stats that model is overfitted. We can reduce the overfitting by tuning the model parameter.

- **<u>Random Forest</u>** - Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

  Below is the code for Random Forest:

```
from sklearn.ensemble import RandomForestRegressor

RF_one = RandomForestRegressor(n_estimators= 500, random_state=100).fit(bike_train_feature,bike_train_target)
RF_one
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=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=500, n_jobs=None, oob_score=False,
                      random_state=100, verbose=0, warm_start=False)
```

- **Model Evaluation –** From the below we can see the Performance of the model

```
#MAPE
MAPE(bike_test_target,RF_predict_one)

#RMSE
RMSE(bike_test_target,RF_predict_one)
```

```
MAPE : 1.6979751778680088
Mean Square :  19812.448761959182
Root Mean Square :  140.75670059346794
```

Here we can see that MAPE is around 1.6 and the accuracy is 98.4 and RMSE is 103 which is better than Decision tree.

Let's See how Linear regression will perform.

- **Linear Regression** - Multiple linear regression is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical.

  Below is the code for the Linear regression:

```
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor

model = sm.OLS(bike_train_target,bike_train_feature).fit()
predictions = model.predict(bike_test_feature)

model.summary()
```

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	1.000
Model:	OLS	**Adj. R-squared (uncentered):**	1.000
Method:	Least Squares	**F-statistic:**	9.316e+07
Date:	Fri, 07 Feb 2020	**Prob (F-statistic):**	0.00
Time:	19:36:50	**Log-Likelihood:**	-1608.4
No. Observations:	584	**AIC:**	3237.
Df Residuals:	574	**BIC:**	3281.
Df Model:	10		
Covariance Type:	nonrobust		

```
Call:
lm(formula = cnt ~ ., data = train_feature_lr)

Residuals:
       Min         1Q     Median         3Q        Max
-3.590e-12 -2.350e-13  1.500e-14  2.300e-13  1.145e-12

Coefficients:
             Estimate Std. Error    t value Pr(>|t|)
(Intercept)  2.200e+01  1.015e-13  2.167e+14  < 2e-16 ***
season      -6.678e-14  3.101e-14 -2.154e+00   0.0317 *
yr           3.044e-13  5.713e-14  5.328e+00 1.43e-07 ***
mnth        -1.331e-15  8.962e-15 -1.490e-01   0.8820
holiday     -7.651e-14  1.127e-13 -6.790e-01   0.4973
weekday      6.990e-15  8.616e-15  8.110e-01   0.4175
workingday   3.529e-13  6.531e-14  5.404e+00 9.58e-08 ***
weathersit   1.495e-15  3.612e-14  4.100e-02   0.9670
temp         2.024e-13  1.422e-13  1.423e+00   0.1553
windspeed    3.249e-13  2.322e-13  1.399e+00   0.1623
casual       3.408e+03  1.607e-13  2.121e+16  < 2e-16 ***
registered   6.926e+03  1.869e-13  3.707e+16  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.123e-13 on 575 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 1.196e+33 on 11 and 575 DF,  p-value: < 2.2e-16
```

Here residual Standard error is quite less so the distance between predicted values and actual values are very less so this model is predicted almost accurate values.
And Multiple R-Square value is 1 so, we can explain about 100 % of the data using our multiple linear regression model. This is very impressive.

- **Model Evaluation -** From the below we can see the Performance of the model:

```
#MAPE
MAPE(bike_test_target,predictions)

#RMSE
RMSE(bike_test_target,predictions)
```

```
MAPE : 0.09693464336052213
Mean Square :  13.560249593721965
Root Mean Square :  3.6824244179238717
```

From the above values we can see that value of MAPE is very less which mean that accuracy is very high which is around 99% and the Value of RMSE is also very less. This is extraordinary result.

5. Conclusion

As we predict the bike rental values using Random Forest, Decision Tree and Linear regression and after we compared the performance of the model by using RMSE and MAPE we can make a conclusion that Linear Regression model can be used for the bike rental prediction since it has a very less MAPE and RMSE value.