COMPREHENSIVE PROJECT REPORT

★ Contents

2.5 KNN model

2.7 Lasso Regression model

2.8 Gradient Boosting algorithm

2.9 Extreme Gradient Boosting (XGB) model

2.10 Support Vector Regression model

3. **Conclusion report**

1. Preprocessing of Data

1.1 Loading Data: The data is taken from a csv file named “day.csv” and has the following structure:

'data.frame': 731 obs. of 16 variables:

$ instant : int 1 2 3 4 5 6 7 8 9 10 ...

$ dteday : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...

$ season : int 1 1 1 1 1 1 1 1 1 1 ...

$ yr : int 0 0 0 0 0 0 0 0 0 0 ...

$ mnth : int 1 1 1 1 1 1 1 1 1 1 ...

$ holiday : int 0 0 0 0 0 0 0 0 0 0 ...

$ weekday : int 6 0 1 2 3 4 5 6 0 1 ...

$ workingday: int 0 0 1 1 1 1 1 0 0 1 ...

$ weathersit: int 2 2 1 1 1 1 2 2 1 1 ...

$ temp : num 0.344 0.363 0.196 0.2 0.227 ...

$ atemp : num 0.364 0.354 0.189 0.212 0.229 ...

$ hum : num 0.806 0.696 0.437 0.59 0.437 ...

$ windspeed : num 0.16 0.249 0.248 0.16 0.187 ...

$ casual : int 331 131 120 108 82 88 148 68 54 41 ...

$ registered: int 654 670 1229 1454 1518 1518 1362 891 768 1280 ...

$ cnt : int 985 801 1349 1562 1600 1606 1510 959 822 1321 ...

Data Sample:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.226642	0.652917	0.350133	247	1867	2114
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.255046	0.590000	0.155471	644	2451	3095
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.242400	0.752917	0.124383	159	1182	1341
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.231700	0.483333	0.350754	364	1432	1796
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.223487	0.577500	0.154846	439	2290	2729

1.2 Python packages and methods used:

1) import pandas as pd
2) import numpy as np
3) from sklearn.model_selection import train_test_split
4) from sklearn.tree import DecisionTreeRegressor
5) from sklearn.metrics import mean_squared_error, r2_score
6) from sklearn.metrics import mean_absolute_error

7) from sklearn.metrics import confusion_matrix

8) from sklearn.preprocessing import StandardScaler

9) from sklearn.ensemble import RandomForestRegressor

10) from sklearn.neighbors import KNeighborsRegressor

11) from sklearn.linear_model import BayesianRidge, LinearRegression

12) from sklearn.linear_model import Lasso

13) from sklearn import svm

14) from sklearn import linear_model

15) import matplotlib.pyplot as plt

16) import seaborn as sns

1.3 R packages and methods used:

1) ggplot2

2) corrgram

3) DMwR

4) caret

5) randomforest

6) unbalanced

7) C50

8) inTrees

9) dummies

10) e1071

11) Information

12) MASS

13) rpart

14) gbm

15) ROSE

16) sampling

17) RRF

18) AICcmodavg

19) Metrics

20) glmnet

1.4 Inputing data to predict:

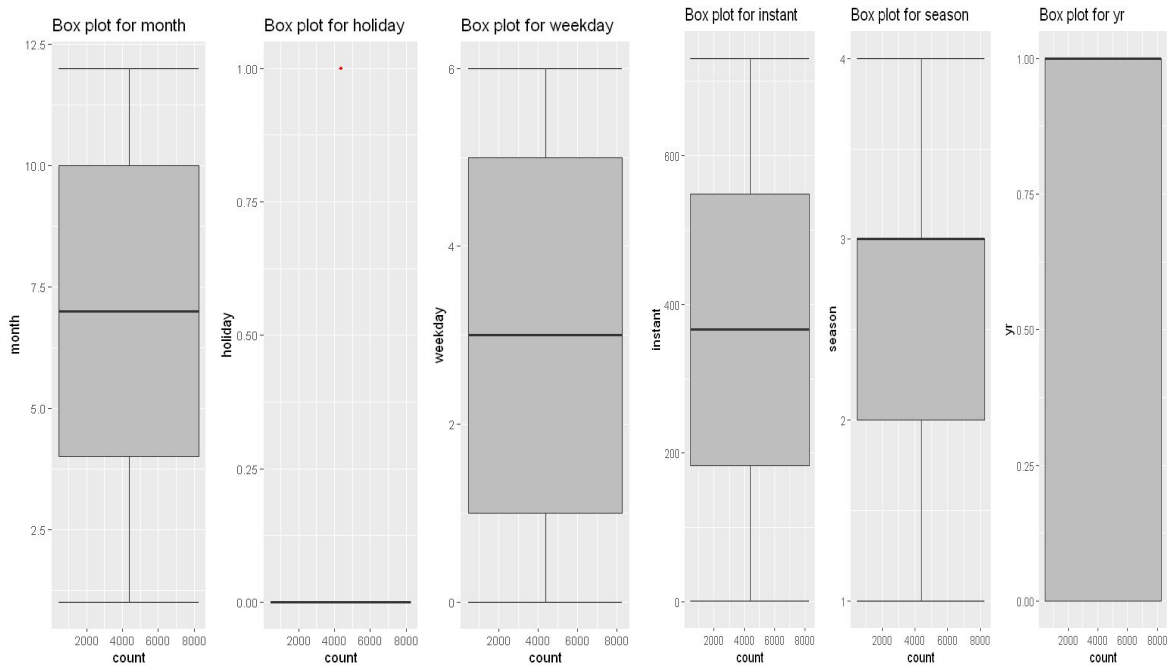
NOTE: These values have been taken from the dataset itself to check how models are going to predict in case of new data.

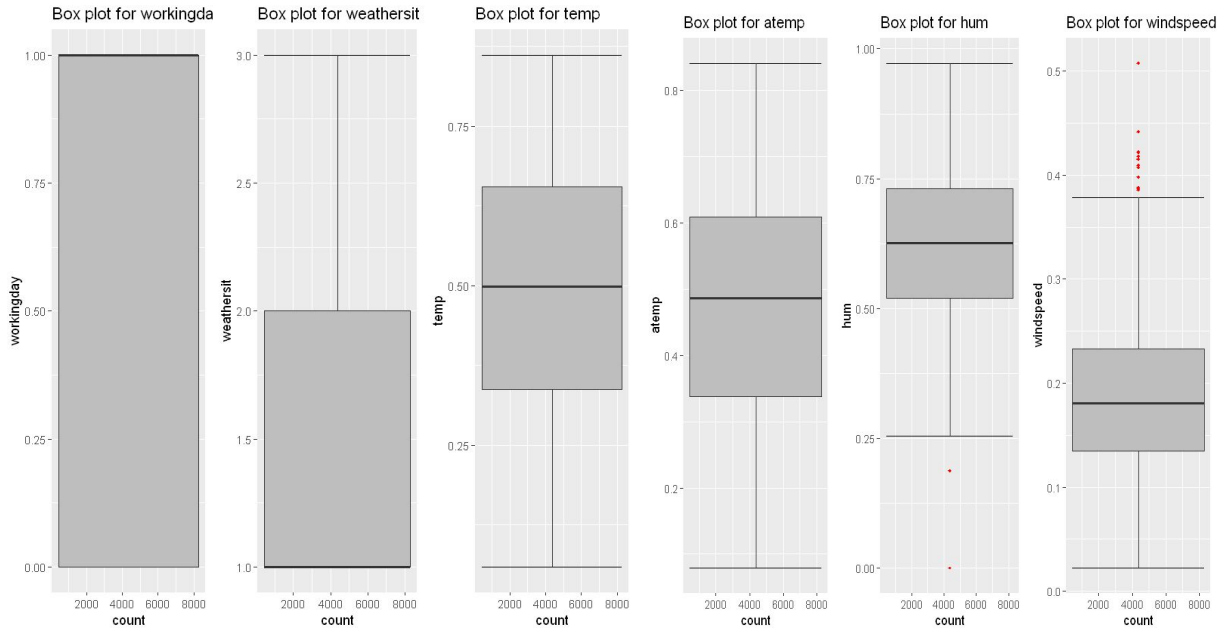
Sample Data: (date:2011-01-05, season:1, yr:0, month:1, holiday:0, weekday:3, workingday:1, weathersit:1, temp:0.226957, atemp:0.229270, humidity:0.436957, windspeed:0.186900, c-82, r-1518, cnt-1600)

Enter date 2011-01-05
Enter season 1
Enter yr 0
Enter month 1
Enter holiday 0
Enter weekday 3
Enter workingday 1
Enter weathersit 1
Enter temp 0.226957

Enter atemp 0.229270
Enter humidity 0.43695
Enter windspeed 0.1869

1.5 Boxplot for variables: Visualizing values of independent variables to check outliers present in the data set with the help of boxplots of variables**.**





1.6 Outlier Analysis: Removing outliers from the Bike renting dataset with all independent variables, excluding the "holiday" variable as the number of rows where holiday is 1 (representing a holiday) is very less but at the same time its significant for us to predict the count value effectively when there is a holiday.

After removal of outliers :

'data.frame': 715 obs. of 15 variables:

$ instant : int 1 2 3 4 5 6 7 8 9 10 ...

$ season : int 1 1 1 1 1 1 1 1 1 1 ...

$ yr : int 0 0 0 0 0 0 0 0 0 0 ...

$ month : int 1 1 1 1 1 1 1 1 1 1 ...

$ holiday : int 0 0 0 0 0 0 0 0 0 0 ...

$ weekday : int 6 0 1 2 3 4 5 6 0 1 ...

$ workingday: int 0 0 1 1 1 1 1 0 0 1 ...

$ weathersit: int 2 2 1 1 1 1 2 2 1 1 ...

$ temp : num 0.344 0.363 0.196 0.2 0.227 ...

$ atemp : num 0.364 0.354 0.189 0.212 0.229 ...

$ hum : num 0.806 0.696 0.437 0.59 0.437 ...

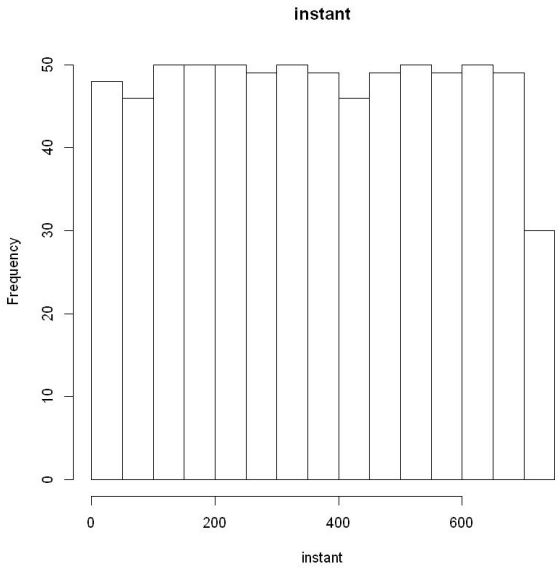
$ windspeed : num 0.16 0.249 0.248 0.16 0.187 ...

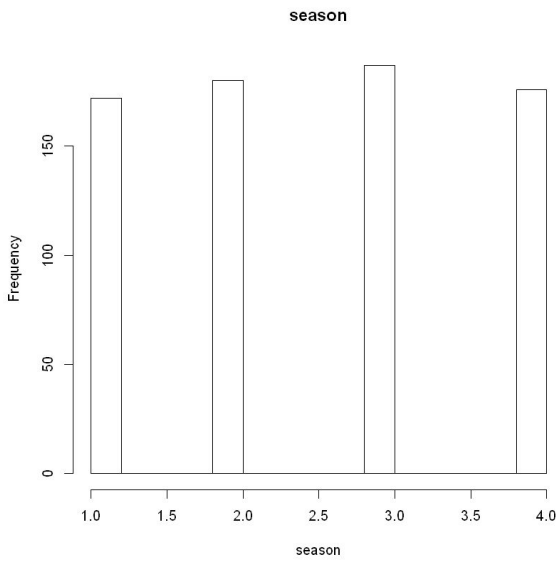
$ casual : int 331 131 120 108 82 88 148 68 54 41 ...

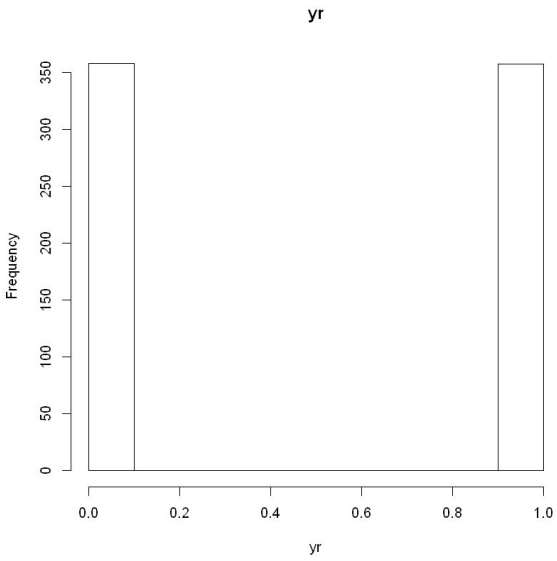
$ registered: int 654 670 1229 1454 1518 1518 1362 891 768 1280 ...

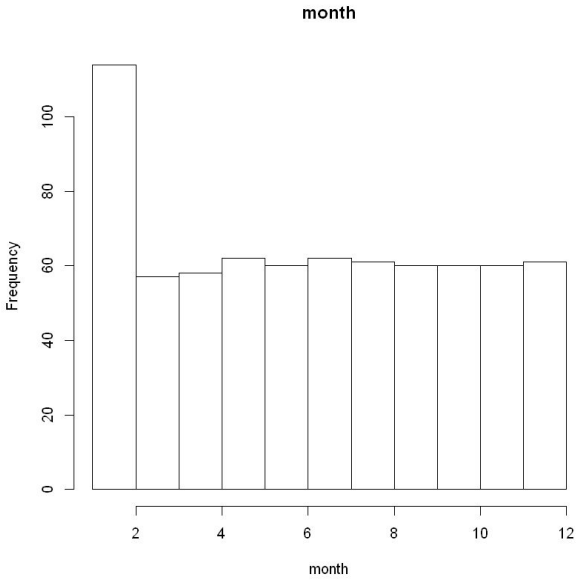
$ count : int 985 801 1349 1562 1600 1606 1510 959 822 1321 ...

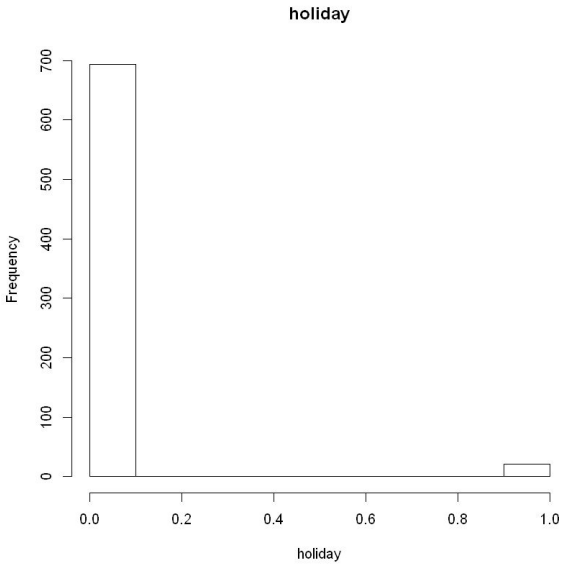
1.7 Histogram to check spread for data : These histograms plots will help us to find which method we should consider to scale the data i.e Normalization or Standardisation, by checking the skewness present in individual variables of data.

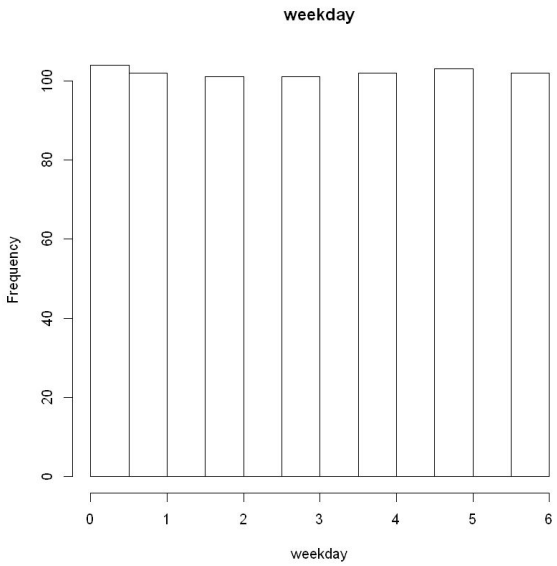


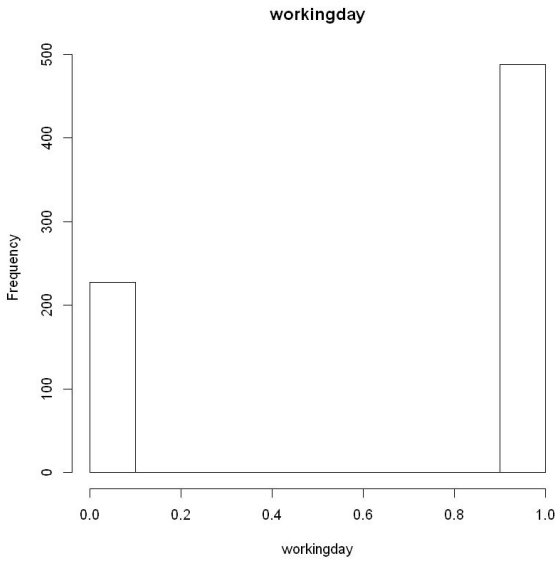


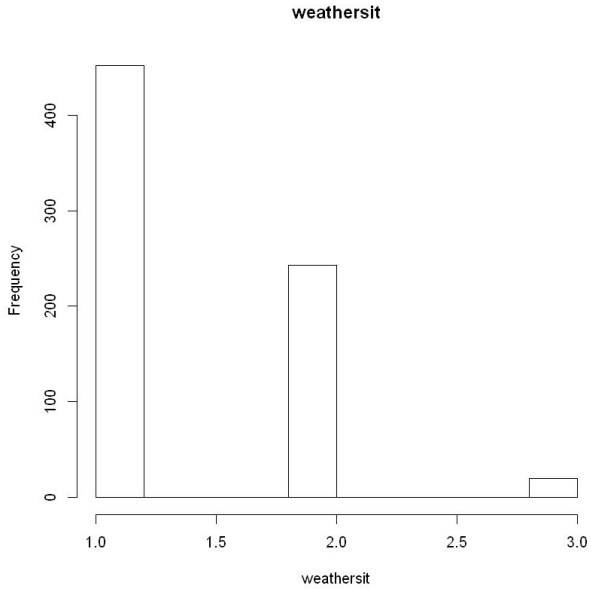

yr
Frequency
0
50
100
150
200
250
300
350
0.0
0.2
0.4
0.6
0.8
1.0
yr

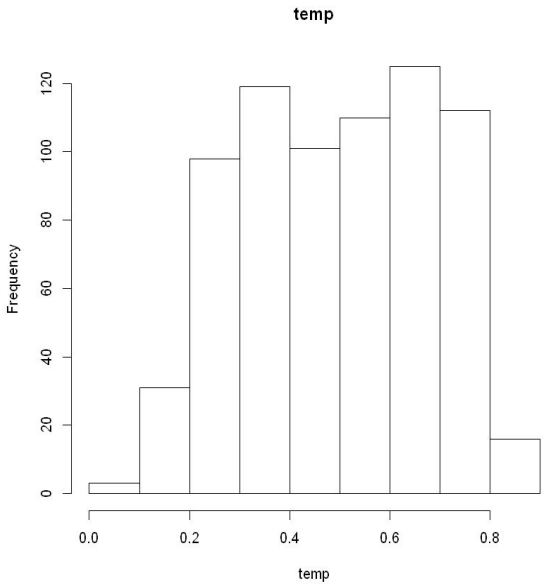

month
Frequency
0
20
40
60
80
100
2
4
6
8
10
12
month

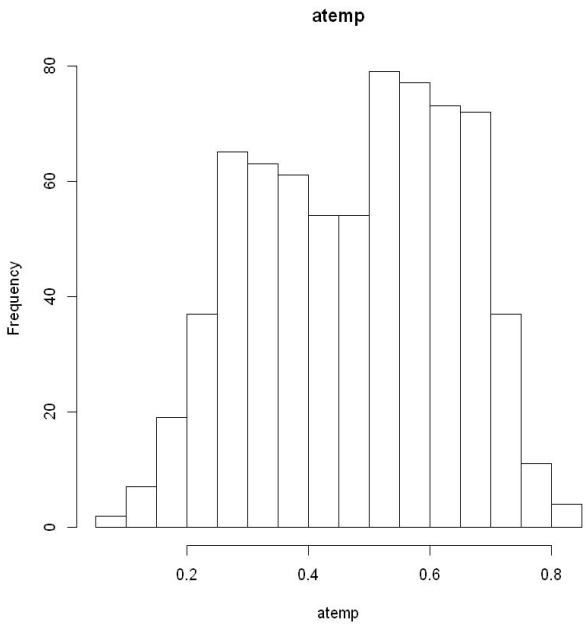

holiday
Frequency
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0
holiday

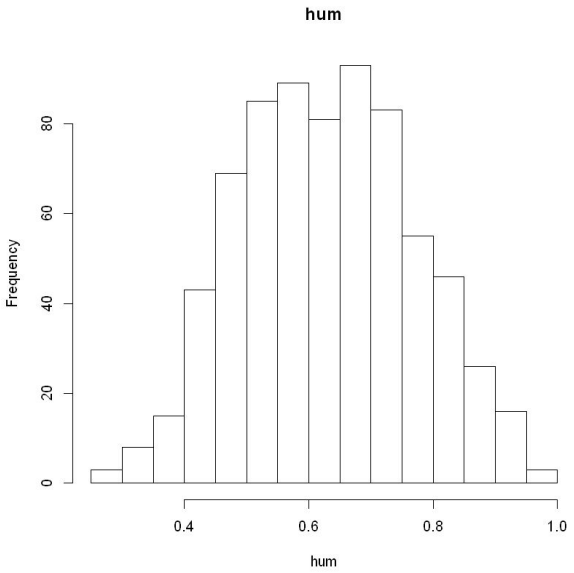

weekday
Frequency
0
20
40
60
80
100
0
1
2
3
4
5
6
weekday

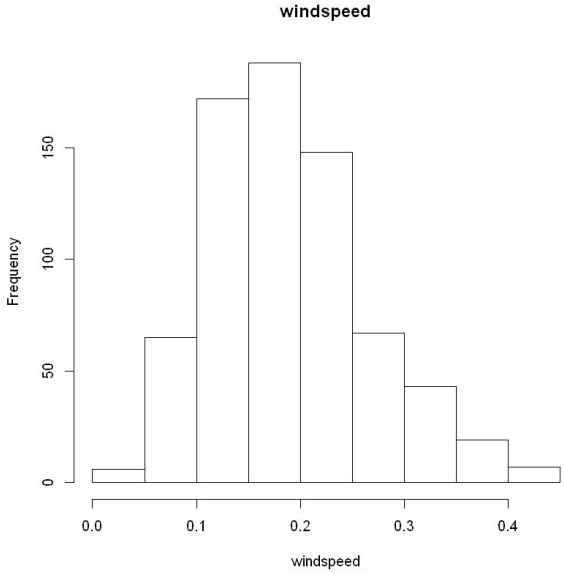

workingday
Frequency
0
100
200
300
400
500
0.0
0.2
0.4
0.6
0.8
1.0
workingday

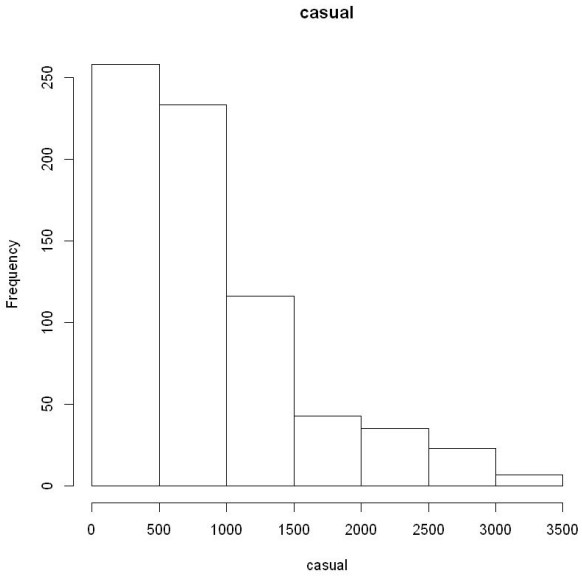

weathersit
Frequency
0
100
200
300
400
1.0
1.5
2.0
2.5
3.0
weathersit

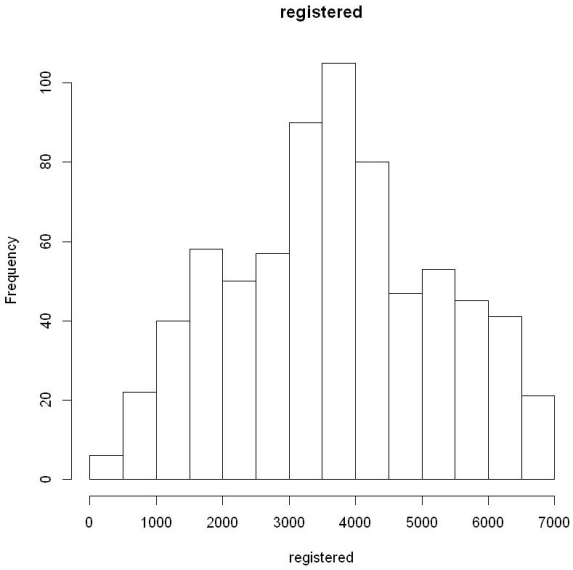

temp
Frequency
0
20
40
60
80
100
120
0.0
0.2
0.4
0.6
0.8
temp

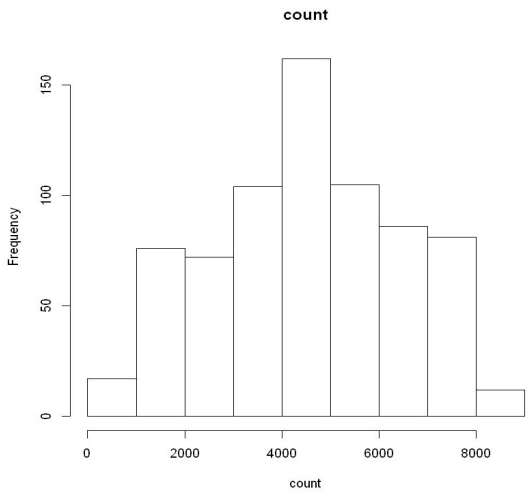

atemp
Frequency
0
20
40
60
80
0.2
0.4
0.6
0.8
atemp


hum
Frequency
0
20
40
60
80
0.4
0.6
0.8
1.0
hum

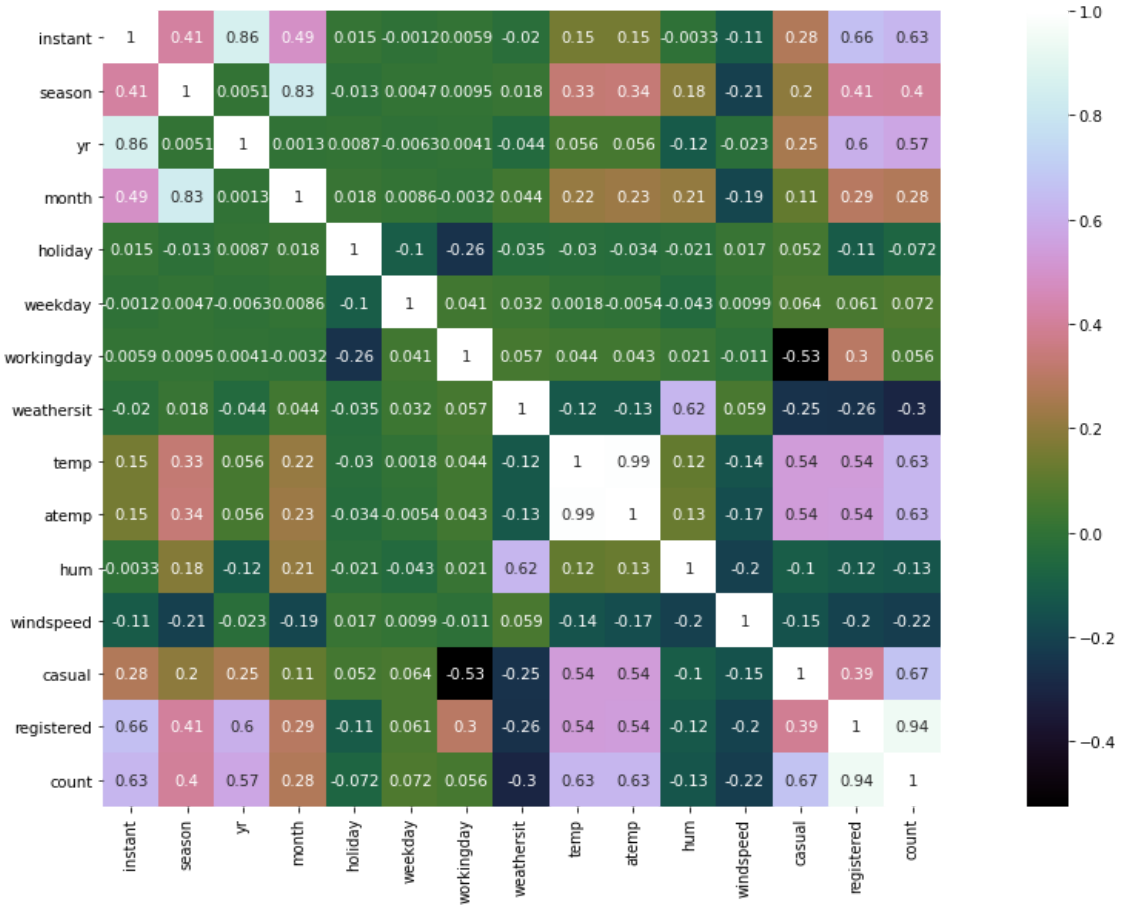

windspeed
Frequency
0
50
100
150
0.0
0.1
0.2
0.3
0.4
windspeed


casual
Frequency
0
50
100
150
200
250
0
500
1000
1500
2000
2500
3000
3500
casual


registered
Frequency
0
20
40
60
80
100
0
1000
2000
3000
4000
5000
6000
7000
registered



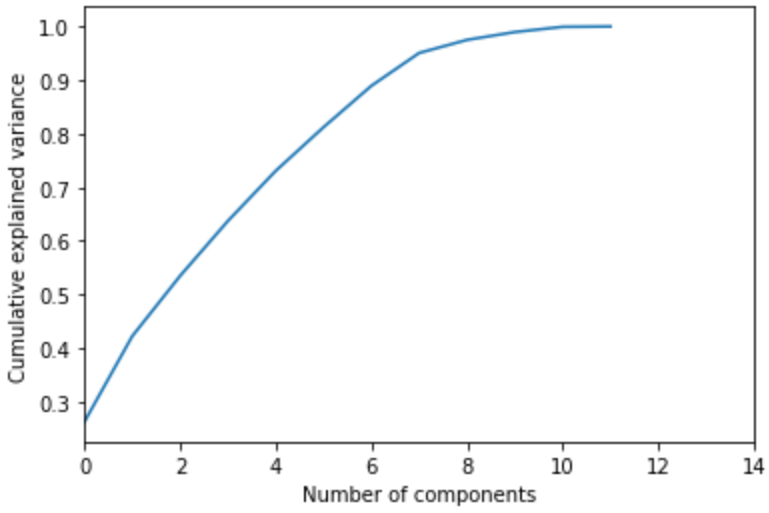
1.8 Correlational Analysis: To check individual correlation (whether positive or negative) between variables, and removal of variables having high positive correlation with each other as they would only give us redundant information.



1.9 Feature Scaling: Scaling of features into comparable values to avoid ambiguity in data models, by applying Standardisation method.

	season	month	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	count
0	-1.36604	-1.613231	1.501831	-1.474435	1.112499	-0.831608	-0.687581	1.252861	-0.360230	-0.764420	-1.937104	-1.834469
1	-1.36604	-1.613231	-1.497653	-1.474435	1.112499	-0.726523	-0.748120	0.465104	0.867791	-1.054597	-1.926838	-1.929662
2	-1.36604	-1.613231	-0.997739	0.677281	-0.727995	-1.635908	-1.754443	-1.392663	0.864585	-1.070557	-1.568163	-1.646153
3	-1.36604	-1.613231	-0.497825	0.677281	-0.727995	-1.616122	-1.615332	-0.293266	-0.362321	-1.087967	-1.423795	-1.535957
4	-1.36604	-1.613231	0.002089	0.677281	-0.727995	-1.469430	-1.510324	-1.394932	0.008540	-1.125690	-1.382731	-1.516297

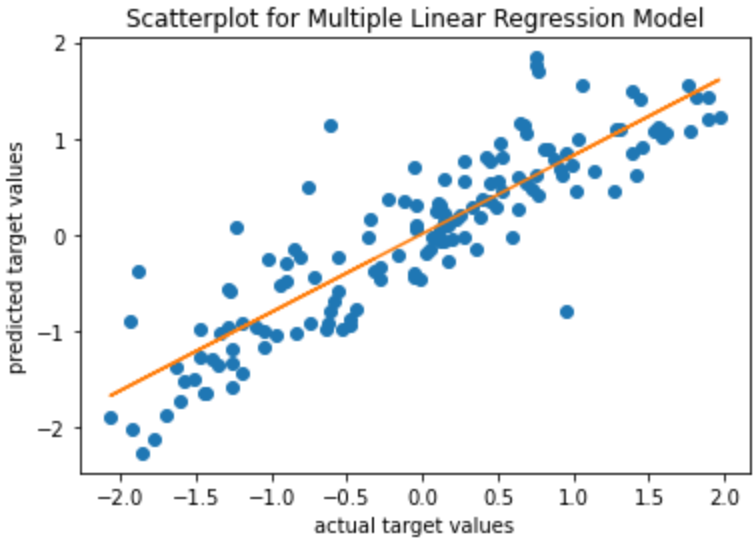
1.10 Principal Component Analysis : Applying Principal Component Analysis to check at least how many variables are needed to cover the variations present in the dataset.

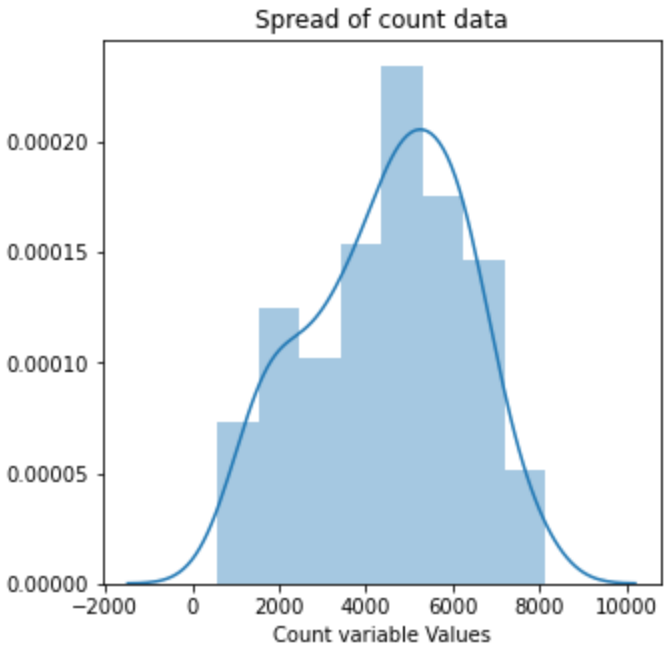


2. Models with different error metrics

2.1 Multiple Linear Regression : In the Multiple Linear Regression model, we have an moderately good r-square value as well as a low mean square error, so until we don't find a better model let's hold on to this model on the basis of its performance given by the following error metrics and its predicted value for the input data given by the user.

- **For Testing dataset:**
  - mean square error 0.26
  - mean absolute error 0.37
  - Root mean square error 0.51
  - r2 score: 0.75

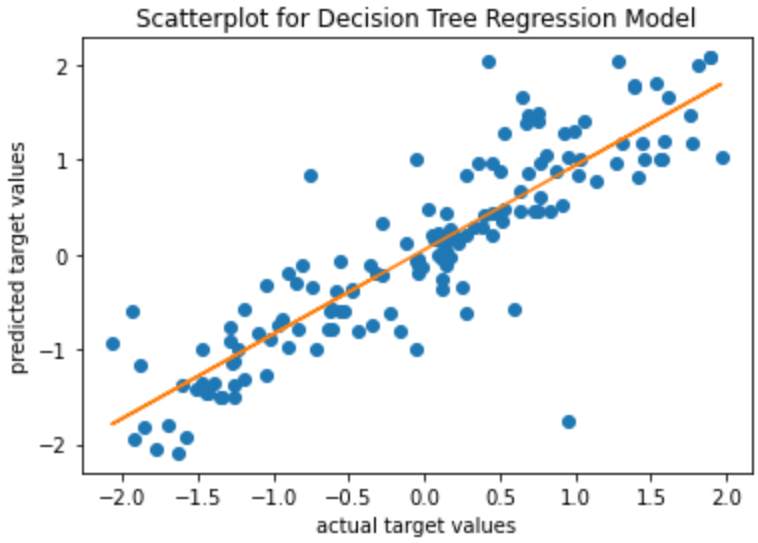


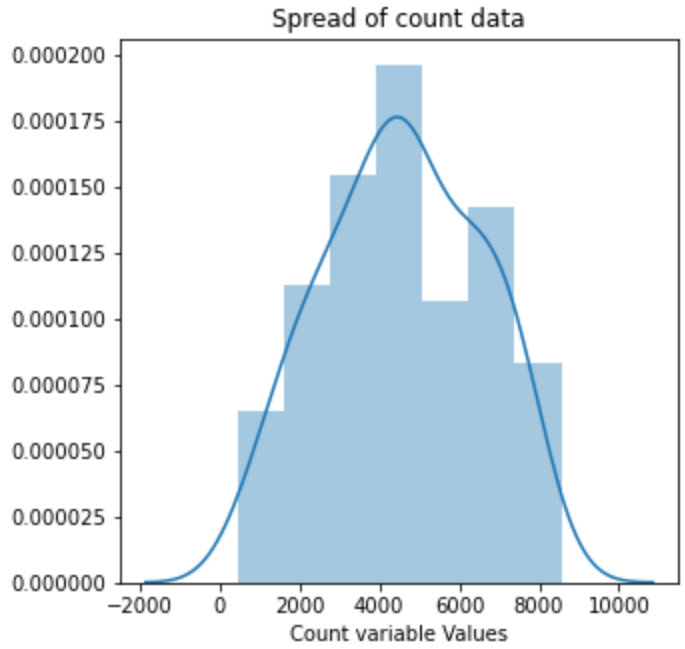


- **For input data:**
  - Actual value: 3118
  - Predicted value: 3622
  - mean square error 3.08
  - mean absolute error 1.75
  - Root mean square error 1.75

2.2 Decision Tree model: In the Decision Tree Regression model, we have an moderately medium r-square value as well as a low mean square error, but considering the performance of multiple regression model, decision tree model underperforms with respect to the multiple regression model, so we would reject this model based on the comparison of error metrics of this model compared to the multiple regression model.

- **For Testing dataset:**
  - mean square error 0.3128805714251789
  - mean absolute error 0.3707398831085963
  - Root mean square error 0.5593572842335915
  - r2 score: 0.7044906830648324

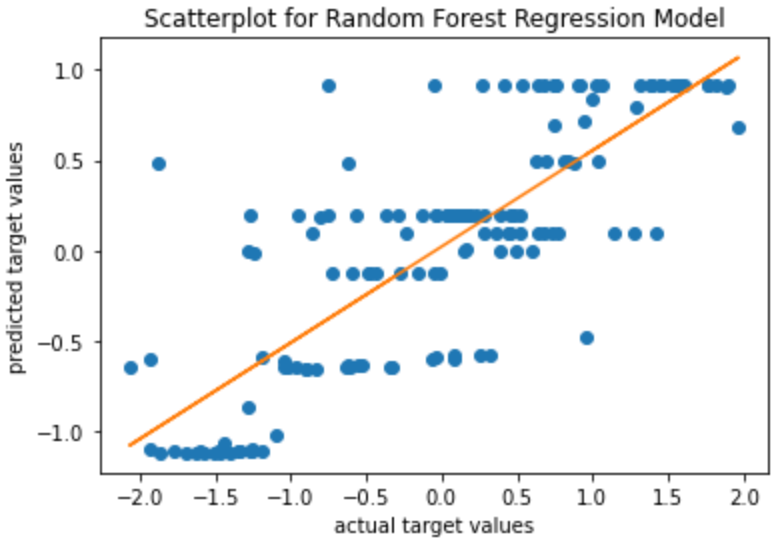


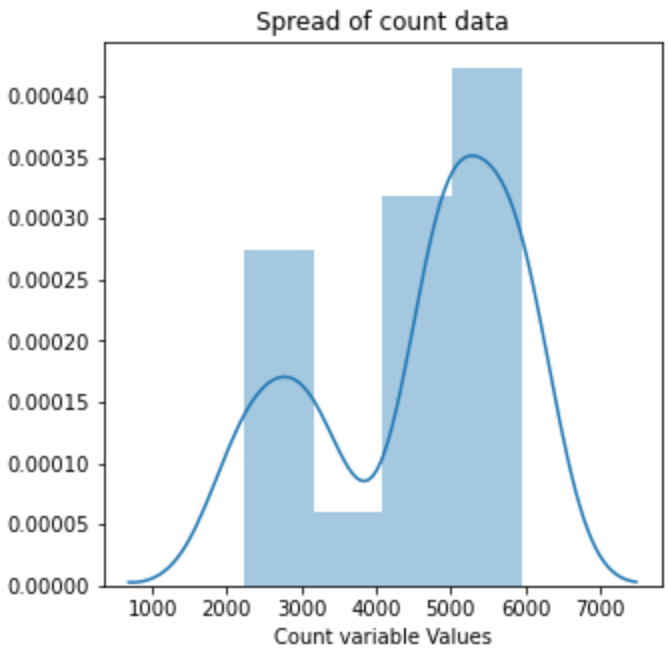


- **For input data:**
  - Actual value: 3118
  - Predicted value: 1495
  - mean square error 4.08
  - mean absolute error 1.80
  - Root mean square error 2.02

2.3 Random Forest Regression model: In the random forest regression model the values are spread very apart, the root mean square value is very high not to mention the r-square value is also quite low, which in a considerable case should be just the inverse of the current scenario. Hence, we'll reject this model based on the below mentioned values of different error metrics for the given problem statement.

- **For Testing dataset:**
  - mean square error 0.36
  - mean absolute error 0.46
  - Root mean square error 0.60
  - r2 score: 0.65

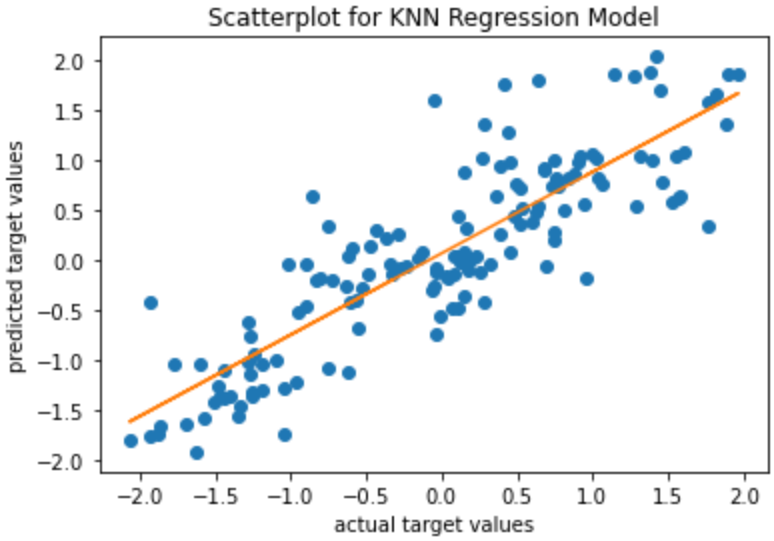


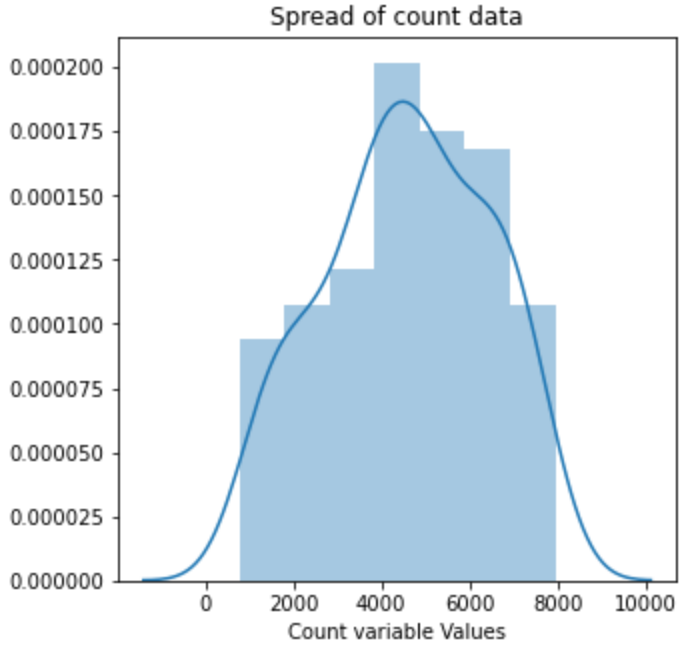


- **For input data:**
  - Actual value: 3118
  - Predicted value: 3755
  - mean square error 1.85
  - mean absolute error 1.35
  - Root mean square error 1.36

2.4 KNN model: In the KNN model for the given problem statement the r-square value is not good enough to consider this model to predict the count variable, also the root mean square error is very high which makes us reject this model.

- **For Testing dataset:**
  - mean square error 0.39
  - mean absolute error 0.44
  - Root mean square error 0.62
  - r2 score: 0.62

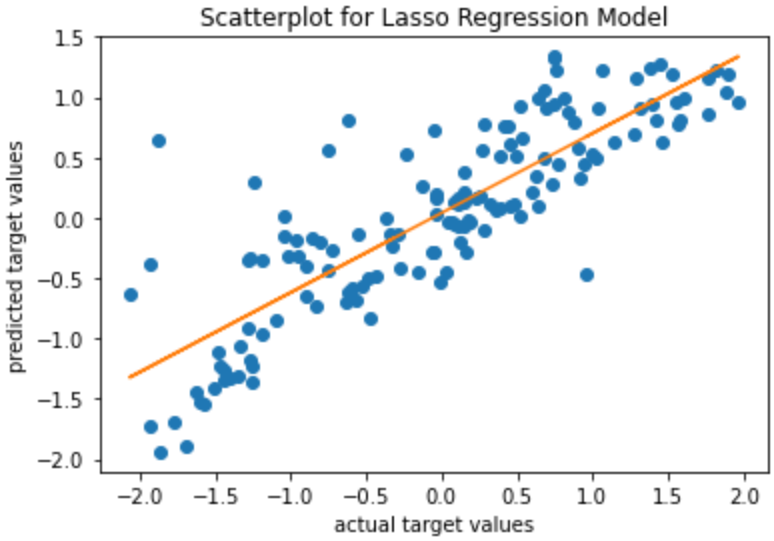


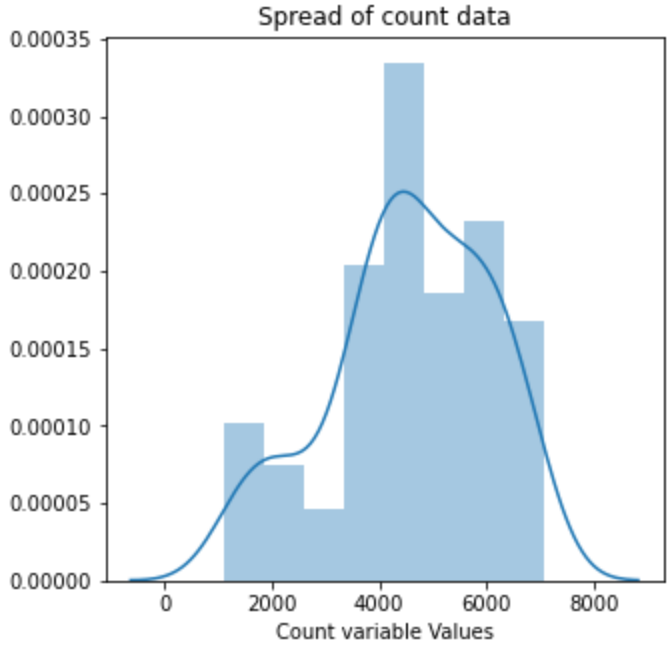


- **For input data:**
  - Actual value: 3118
  - Predicted value: 3927
  - mean square error 0.984
  - mean absolute error 0.981
  - Root mean square error 0.992

2.5 Lasso Regression model : The Lasso regression generally performs well, when the dependent variable has a low standard deviation from the mean, this implies that the data should be spread near the mean value of the dependent variable for the model to predict precise values, but such is not the case with count variable. Hence, the r-square value is quite low and thus we would not consider the model.

- **For Testing dataset:**
  - mean square error 0.34
  - mean absolute error 0.41
  - Root mean square error 0.58
  - r2 score: 0.67

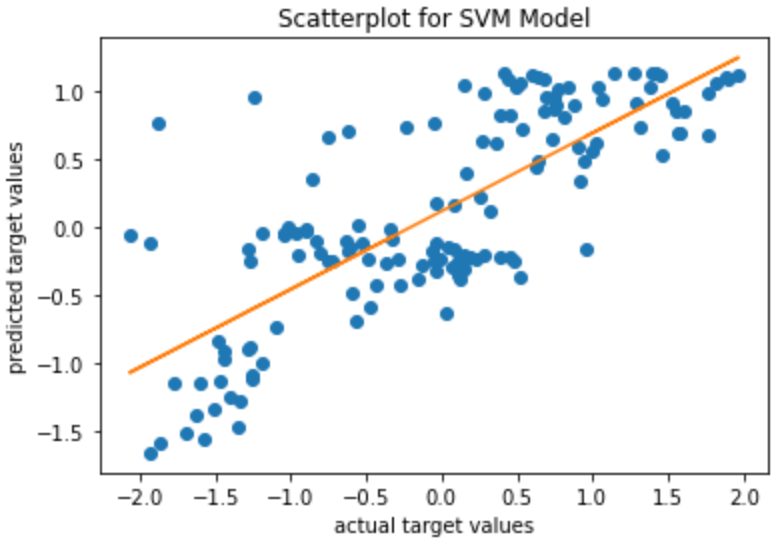


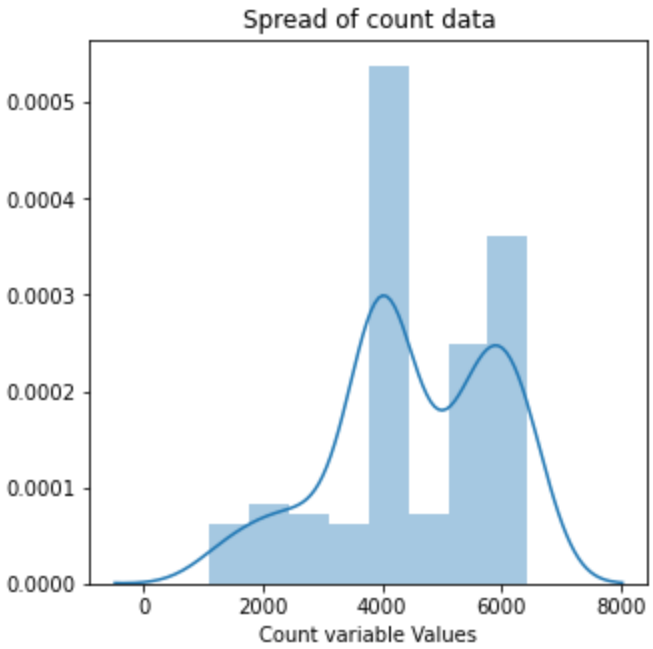


- **For input data:**
  - Actual value: 3118
  - Predicted value: 4165
  - mean square error 2.68
  - mean absolute error 1.62
  - Root mean square error 1.63

2.6 Support Vector Regression model: The Support Vector Regressor doesn't have any parameter value which pique's our interest to consider the model. Hence for the given problem statement we'll reject this model based upon the given below error metrics values.

- **For Testing dataset:**
  - mean square error 0.64
  - mean absolute error 0.56
  - Root mean square error 0.80
  - r2 score: 0.39



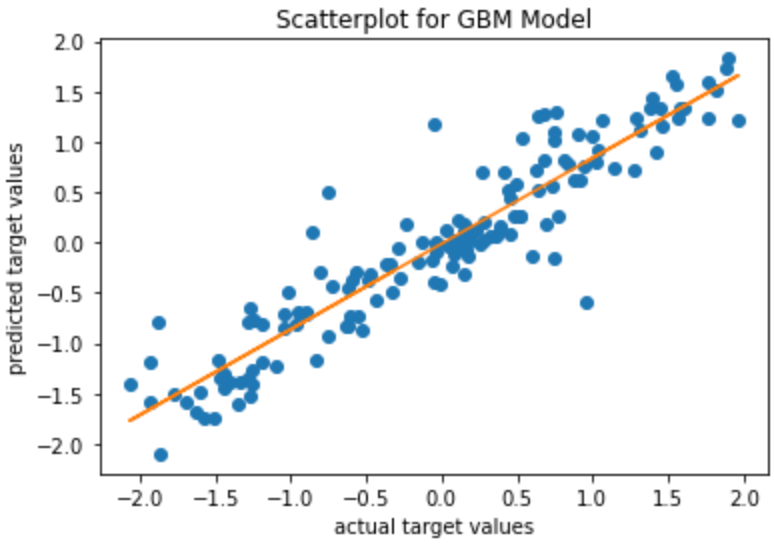


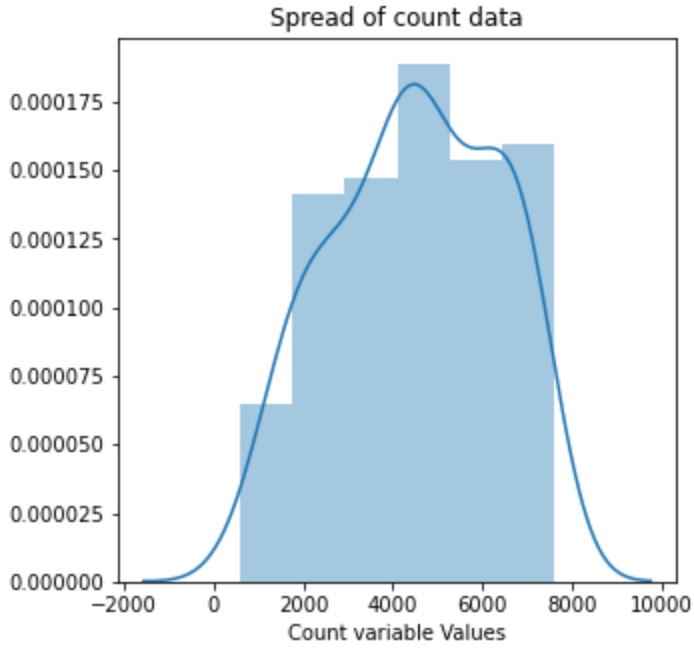
- **For input data:**
  - Actual value: 3118
  - Predicted value: 4029
  - mean square error 1.10
  - mean absolute error 1.04
  - Root mean square error 1.04

2.7 Gradient Boosting algorithm: In the Gradient Boosting model we acquire a considerably good r-square value as well as very less mean square error in the prediction on Testing dataset, hence until a better model shows up we can use this model for predicting the count values.

- **For Testing dataset:**

  - mean square error 0.14850671700386742
  - mean absolute error 0.25964238863914435
  - Root mean square error 0.38536569256210057
  - r2 score: 0.8592363809527677

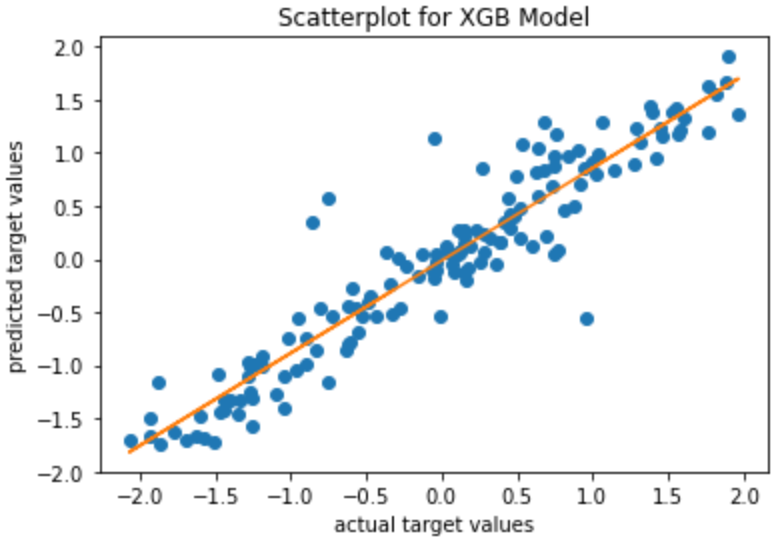


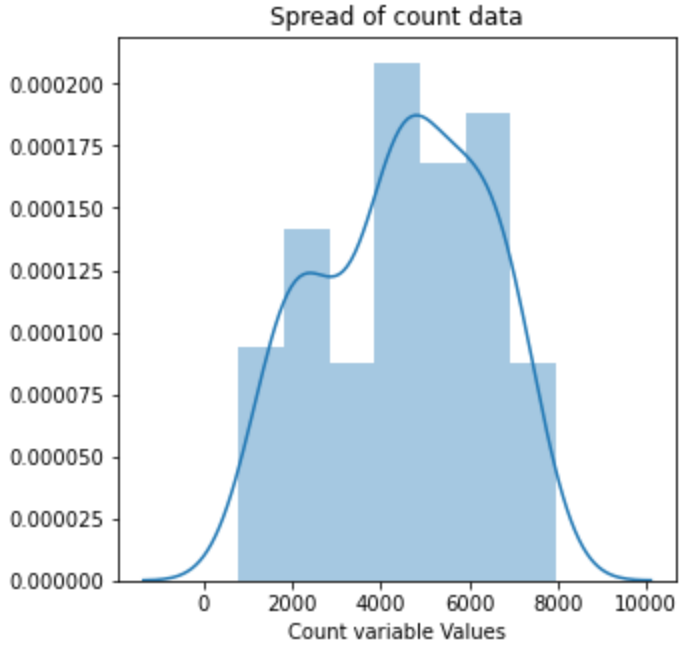


- **For input data:**
  - Actual value: 3118
  - Predicted value: 3976
  - mean square error 2.94
  - mean absolute error 1.70
  - Root mean square error 1.71

2.8 Extreme Gradient Boosting (XGB) model: In the results of the xgb model we get a considerably good r-square value with respect to the previous r-square values of other models also not to mention its mean square error is also very low, hence we can consider this model for predicting values of the count variable.

- **For Testing dataset:**
  - mean square error 0.12
  - mean absolute error 0.23
  - Root mean square error 0.35
  - r2 score: 0.88

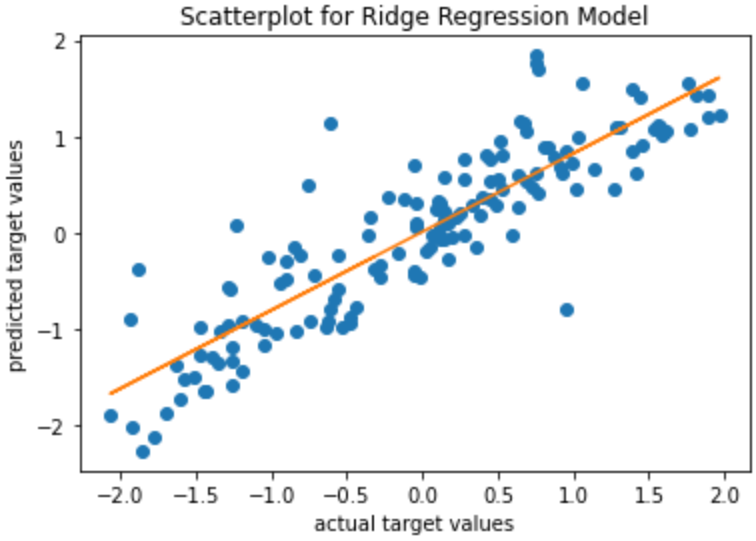


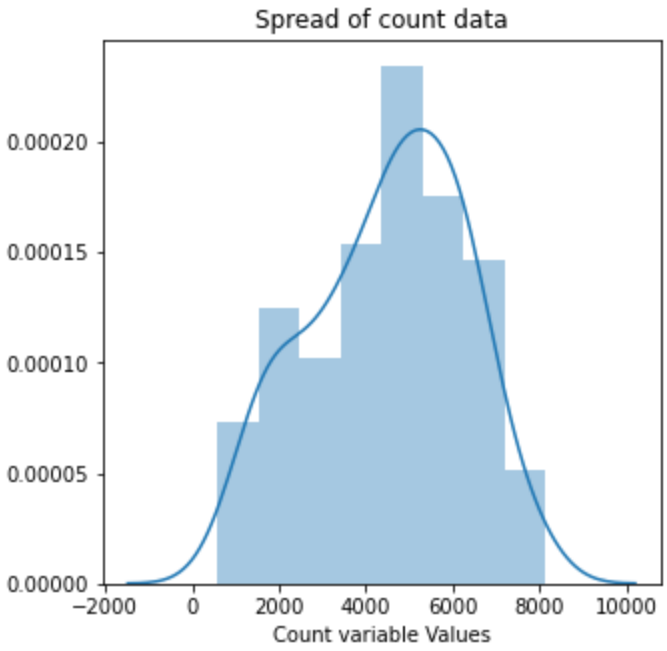


- **For input data:**
  - Actual value: 3118
  - Predicted value: 3819
  - mean square error 2.56
  - mean absolute error 1.60
  - Root mean square error 1.60

2.9 Ridge Regression model : In the ridge regression model the variance of variables are large so they are a bit far from the true value as we can see in the scatterplot, also the ridge regression works well when the variables have high multicollinearity with each other. Also, the r-square value is not quite high enough to consider it as a good regression model for the given problem statement.

- **For Testing dataset:**
  - mean square error 0.26
  - mean absolute error 0.36
  - Root mean square error 0.51
  - r2 score: 0.75





- **For input data:**
  - Actual value: 3118
  - Predicted value: 3622
  - mean square error 3.08
  - mean absolute error 1.75
  - Root mean square error 1.75

3. Conclusion report : So, now to finalize the model after going through various error metrics and scatterplots of various models, we came to a conclusion which is, the Extreme Gradient Boosting model to provides the most satisfactory results where mean square error is 0.12 and the r-square value is 0.88 which is very much acceptable. So we'll accept the XGB model for predicting the count variable.
