**Project Report:**

**BIKE RENTING**

**Chander Mohan**

**Contents**

**1. Introduction**

1.1 Problem Statement . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.2 Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.3 Exploratory Data Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

**2. Methodology**

2.1 Pre Processing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.1.1 Missing Value Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .6

2.1.2 Outlier Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2.1.3 Feature Selection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .8

2.1.4 Feature Scaling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

2.1.5 Principal Component Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

Extra Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

2.2 Modeling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . 22

2.2.1 Linear Regression. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

2.2.2 Decision Tree. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .22

2.2.3 Random Forest. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

2.2.4 Support Vector Regression . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2.2.5 Gradient Boosting. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

**3. Conclusion**

3.1 Model Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

3.2 Model Selection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

**Chapter 1**

**Introduction**

* 1. **Problem Statement**

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings. The details of data attributes in the dataset are as follows -

* 1. **Data**

**There are 16 variables in our data in which 15 are independent variables and 1 (cnt) is dependent variable. Since our target variable is continuous in nature, this is a regression problem.**

**Variables Information:**

instant: Record index

dteday: Date

season: Season (1:springer, 2:summer, 3:fall, 4:winter)

yr: Year (0: 2011, 1:2012)

mnth: Month (1 to 12)

holiday: weather day is holiday or not (extracted from Holiday Schedule)

weekday: Day of the week

workingday: If day is neither weekend nor holiday is 1, otherwise is 0.

weathersit: (extracted fromFreemeteo) 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: Normalized temperature in Celsius. The values are derived via (t-t\_min)/(t\_max-t\_min), t\_min=-8, t\_max=+39 (only in hourly scale)

atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t\_min)/(t\_maxt\_min), t\_min=-16, t\_max=+50 (only in hourly scale)

hum: Normalized humidity. The values are divided to 100 (max)

windspeed: Normalized wind speed. The values are divided to 67 (max)

casual: count of casual users

registered: count of registered users

cnt: count of total rental bikes including both casual and registered

**1.3 Exploratory Data Analysis**

Exploratory Data Analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics. In the given data set there are 16 variables and data types of all variables are either float64 or int64. There are 731 observations and 16 columns in our data set. Missing value is not present in our data.

**List of columns and their number of unique values** -

instant 731

dteday 731

season 4

yr 2

mnth 12

holiday 2

weekday 7

workingday 2

weathersit 3

temp 499

atemp 690

hum 595

windspeed 650

casual 606

registered 679

cnt 696

dtype: int64

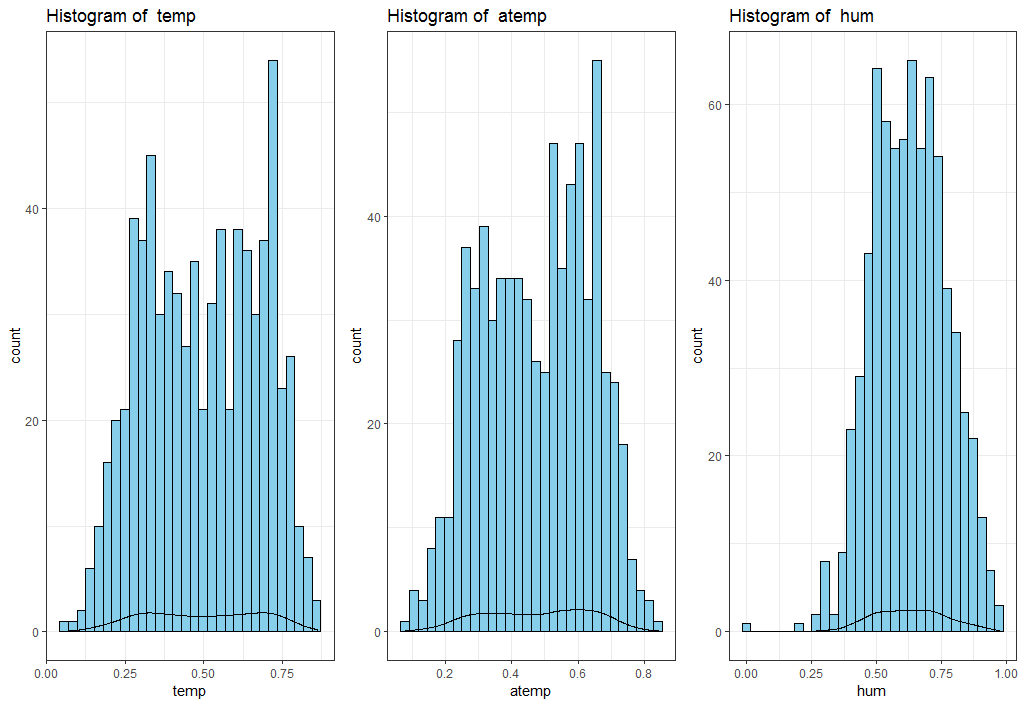
**From EDA we have concluded that there are 7 continuous variable and 9 categorical variable in nature.**

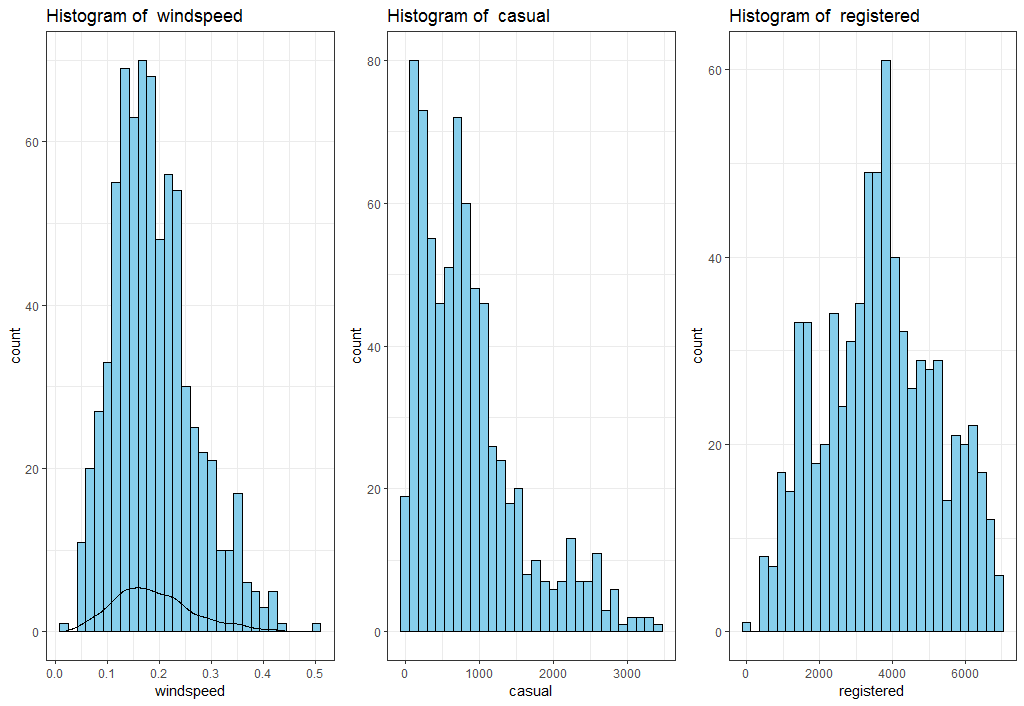
**Chapter 2**

**Methodology**

Before feeding the data to the model we need to clean the data and convert it to a proper format. It is the most crucial part of data science project we spend almost 80% of time in it.

**2.1 Pre Processing**

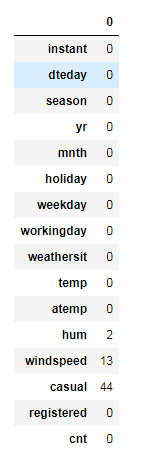
Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms looking at data refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis. To start this process we will first try and look at all the probability distributions of the variables. Most analysis like regression, require the data to be normally distributed. We can visualize that in a glance by looking at the probability distributions or probability density functions of the variable.



**2.2.1 Missing Value Analysis**

In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data. If a columns has more than 30% of data as missing value either we ignore the entire column or we ignore those observations But in this problem we don’t have any missing value.

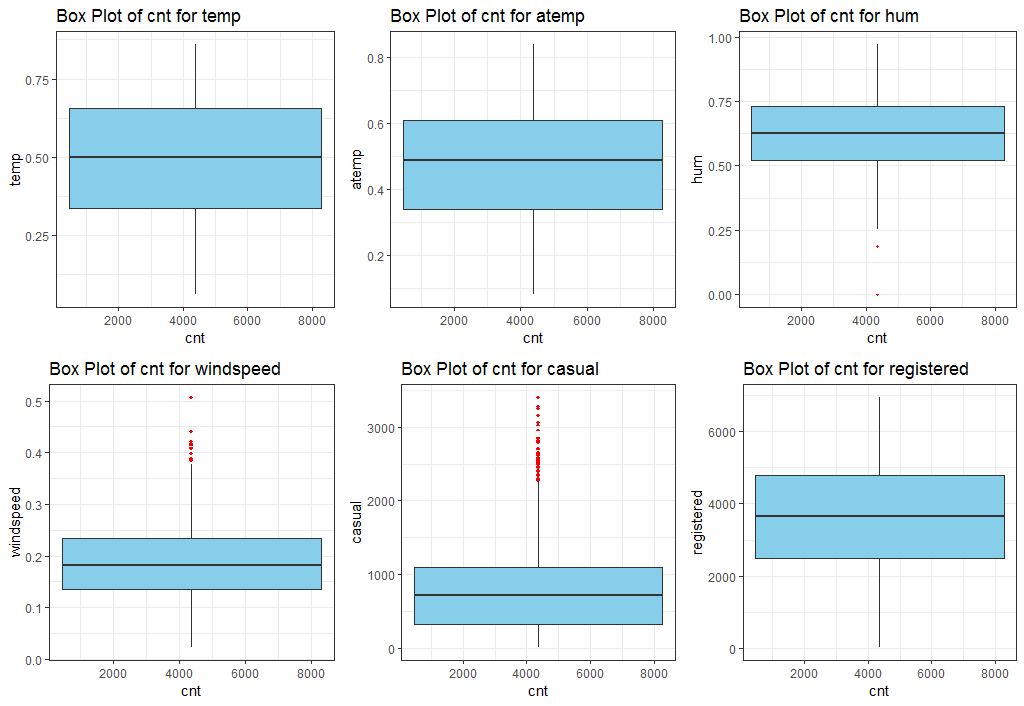
Missing\_val # -> There is no missing value found.



**2.1.2 Outlier Analysis**

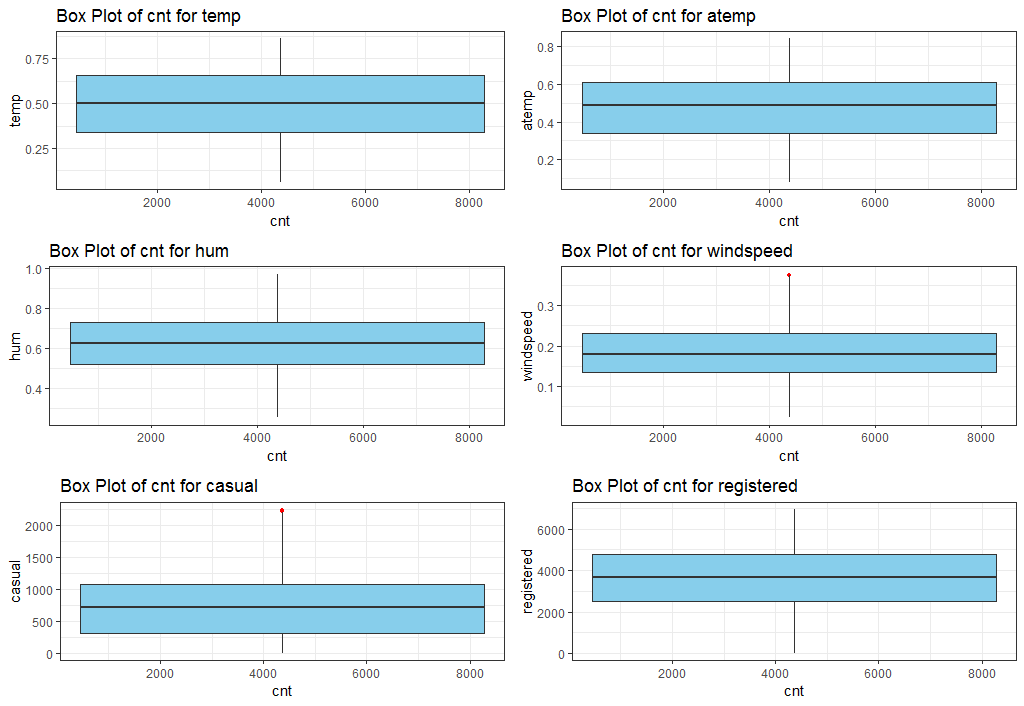
We can clearly observe from these probability distributions that most of the variables are skewed. The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data. One of the other steps of pre-processing apart from checking for normality is the presence of outliers. In this case we use a classic approach of removing outliers. We visualize the outliers using boxplots.

In figure we have plotted the boxplots of the 6 predictor variables with respect to **cnt**. A lot of useful inferences can be made from these plots. First as we can see, we have a lot of outliers and extreme values in each of the data set.



From the boxplot “hum”, “windspeed”, and “casual” consists of outliers. We have converted the outliers (data beyond minimum and maximum values) as NA i.e. missing values and fill them by **KNN** imputation method in case of R and mean method in case of Python

Below we can see new Boxplot to make sure that we have almost removed outliers:



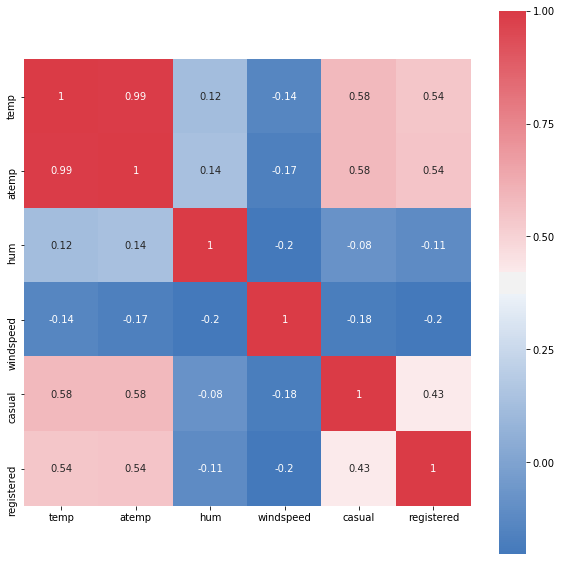
**2.1.3 Feature Selection**

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class and regression prediction. Selecting subset of relevant columns for the model construction is known as Feature Selection.

We cannot use all the features because some features may be carrying the same information or irrelevant information which can increase overhead. To reduce overhead we adopt feature selection technique to extract meaningful features out of data. This in turn helps us to avoid the problem of multi collinearity.

In this project we have selected **Correlation Analysis** for numerical variable, **ANOVA** (Analysis of variance) for categorical variable and in case of R we have also used Backward Elimination Method by lm().

**Correlation Plot:**



From correlation analysis we have found that **temp** and **atemp** has high

correlation (=0.99), so we have excluded the **atemp** column both in **R and PYTHON**

**ANOVA in PYTHON:**



Here P value of **every variable** is 0.0 which is less than the **threshold value** 0.05.

**My Hypothesis is**-

**H0= Independent does not variable explain our Target variable.**

**H1= Independent variable does not explain our Target variable**.

If p> 0.05 then Reject the **NULL Hypothesis (H0)**, that means this particular Independent variable is not going to explain my **Target Variable**.

Now, in my case -

P value for every variable, p < 0.05 that means every categorical variable explain my target variables

**Conclusion: -**

**we have not Removed any categorical variables from ANOVA. But**

**we have removed “atemp” correlation Plot Analysis.**

**we will also remove 'instant' and "dteday" whcih is irrelevant for model learning & prediction.**

And finally I have 13 variables remaining in Python –

Index(['season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday',

'weathersit', 'temp', 'hum', 'windspeed', 'casual', 'registered'

'cnt']

**Dummy variables:**

I have converted categorical variables (“season”, “mnth”, “weekday”, and “weathersit”) into dummy variables. Because the categories in these variables are Nominal data. The factor (1,2,3,4….) which represent these categories can be understood as ordinal data by Machine learning Model because of these order (1,2,3,4..). Therefore, to avoid this confusion sometimes it is compulsory to convert these variables into dummy variables where each categories will become a feature with 0/1 representation.

1 represents presence of categories

And

0 represents absence of categories.

And after all have remaining 31 features with dummy variables.

**ANOVA in R:**

instant 1 1.083e+09 1.083e+09 476.8 <2e-16 \*\*\*

season 1 4.518e+08 451797359 144 <2e-16 \*\*\*

yr 1 8.798e+08 879828893 344.9 <2e-16 \*\*\*

mnth 1 2.147e+08 214744463 62.01 1.24e-14 \*\*\*

holiday 1 1.280e+07 12797494 3.421 0.0648 .

weekday 1 1.246e+07 12461089 3.331 0.0684 .

workingday 1 1.025e+07 10246038 2.737 0.0985 .

weathersit 1 2.423e+08 242288753 70.73 <2e-16 \*\*\*

**But here I have only deleted “holiday” whose p>0.05 because the other two variables (“weekday” and “workingday”) contribute to explain my target variable (cnt).**

**Backward elimination in R:**

Backward elimination method is used to delete the irrelevant variables one by one by keeping In mind that p value of each variable must be less than 0.05. And also the best possible value of R^2 and adjusted R^2. Here, first we include all the variables and try to get the best possible team of Features.

Call:

lm(formula = cnt ~ ., data = dataset\_deleted)

Residuals:

Min 1Q Median 3Q Max

-353.93 -120.99 -34.17 44.25 1783.12

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 36.52445 74.27285 0.492 0.62304

instant -0.49723 1.08896 -0.457 0.64809

season -6.49720 17.36816 -0.374 0.70845

yr 221.07385 402.06259 0.550 0.58259

mnth 12.03869 33.59409 0.358 0.72018

holiday -19.06951 59.85767 -0.319 0.75014

weekday 14.63003 4.83612 3.025 0.00257 \*\* 🡪 Good predictor

workingday -269.70513 36.39110 -7.411 3.53e-13 \*\*\* 🡪 Good predictor

weathersit 21.49505 25.07165 0.857 0.39154

temp 214.62734 85.24629 2.518 0.01203 \* 🡪 Good predictor

hum -54.19694 97.64473 -0.555 0.57904

windspeed -68.40549 145.57847 -0.470 0.63858

casual 0.95419 0.03452 27.639 < 2e-16 \*\*\* 🡪 Good predictor

registered 1.03449 0.01513 68.385 < 2e-16 \*\*\* 🡪 Good predictor

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 257.3 on 717 degrees of freedom

Multiple R-squared: 0.9827, Adjusted R-squared: 0.9824

F-statistic: 3128 on 13 and 717 DF, p-value: < 2.2e-16

**From Backward elimination I have deleted one by one these variables (dteday,atemp,holiday,season,windspeed,yr) whose P value is >0.05**

**Finally I received these predictors which contribute much to explain my target variables.**

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 21.90354 55.97710 0.391 0.69570

instant 0.10481 0.08058 1.301 0.19377

mnth -7.82514 3.38035 -2.315 0.02090 \* 🡪 Good predictor

weekday 14.79016 4.79622 3.084 0.00212 \*\* 🡪 Good predictor

workingday -266.38390 35.01198 -7.608 8.69e-14 \*\*\* 🡪 Good predictor

weathersit 20.36923 24.37128 0.836 0.40355

temp 204.81049 83.83195 2.443 0.01480 \* 🡪 Good predictor

hum -44.82217 93.05406 -0.482 0.63018

casual 0.95727 0.03401 28.146 < 2e-16 \*\*\* 🡪 Good predictor

registered 1.03457 0.01384 74.738 < 2e-16 \*\*\* 🡪 Good predictor

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 256.7 on 721 degrees of freedom

Multiple R-squared: 0.9827, Adjusted R-squared: 0.9824

F-statistic: 4538 on 9 and 721 DF, p-value: < 2.2e-16

**And after all I got only 10 predictor variable in R –**

"instant" "mnth" "weekday" "workingday" "weathersit"

"temp" "hum" "casual" "registered" "cnt"

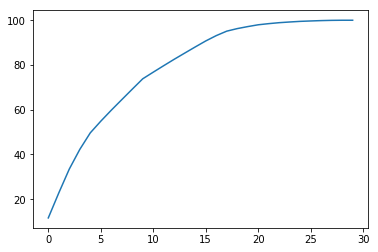
**2.2.4 Feature Scaling**

**Feature scaling** is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. Since our data is not uniformly distributed as we have seen in (**2.1 Pre Processing**) we have used **Normalization** as Feature Scaling Method.

**5 Principal Component Analysis**

Principal component analysis is a method of extracting important variables (in form of components) from a large set of variables available in a data set. It extracts low dimensional set of features from a high dimensional data set with a motive to capture as much information as possible. With fewer variables, visualization also becomes much more meaningful. PCA is more useful when dealing with 3 or higher dimensional data. After creating dummy variable of categorical variables the shape of our data became 31 columns and 731 observations, this high number of columns leads to bad accuracy.

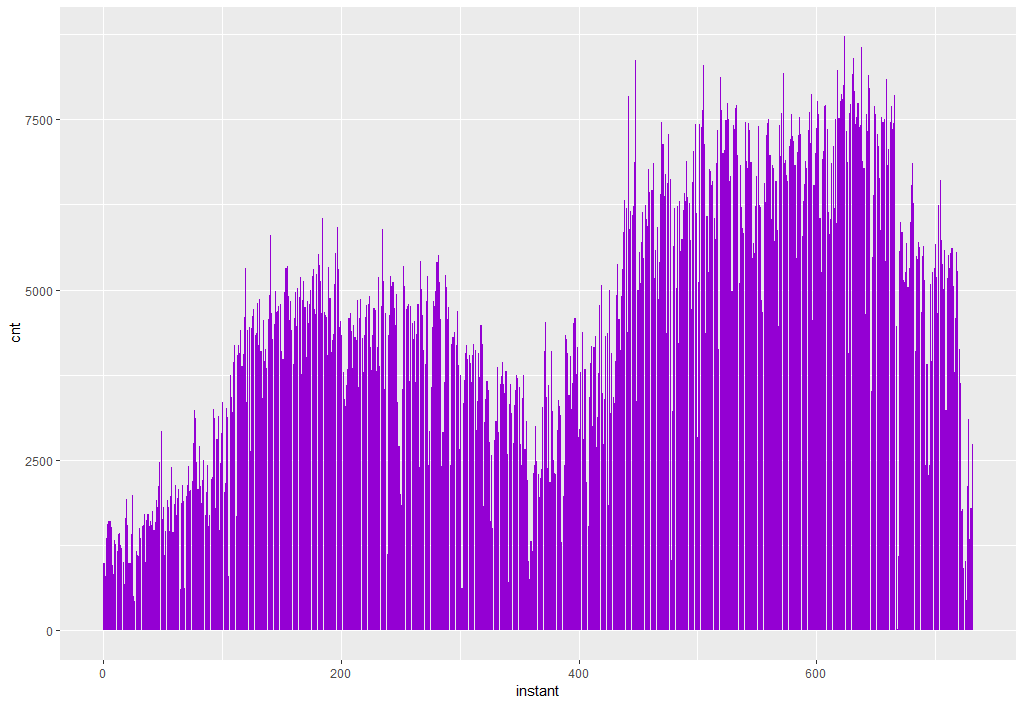
We have used PCA only in Python



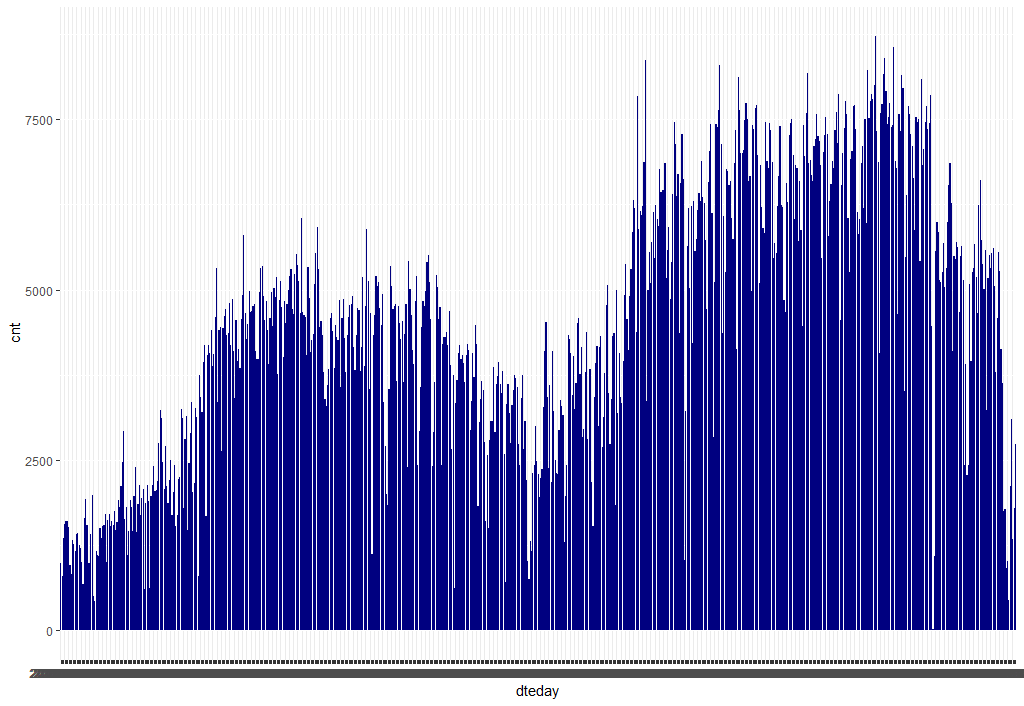
We have applied PCA algorithm on our data and from the above graph. we have concluded that 25 variables out of 31 explains more than 90% of data. So we have selected only those 25 variables to feed our models.

**Extra Figures**

1. **From below graph we can conclude that in 2012 customer take more bike in RENT as compared to 2011 data.**

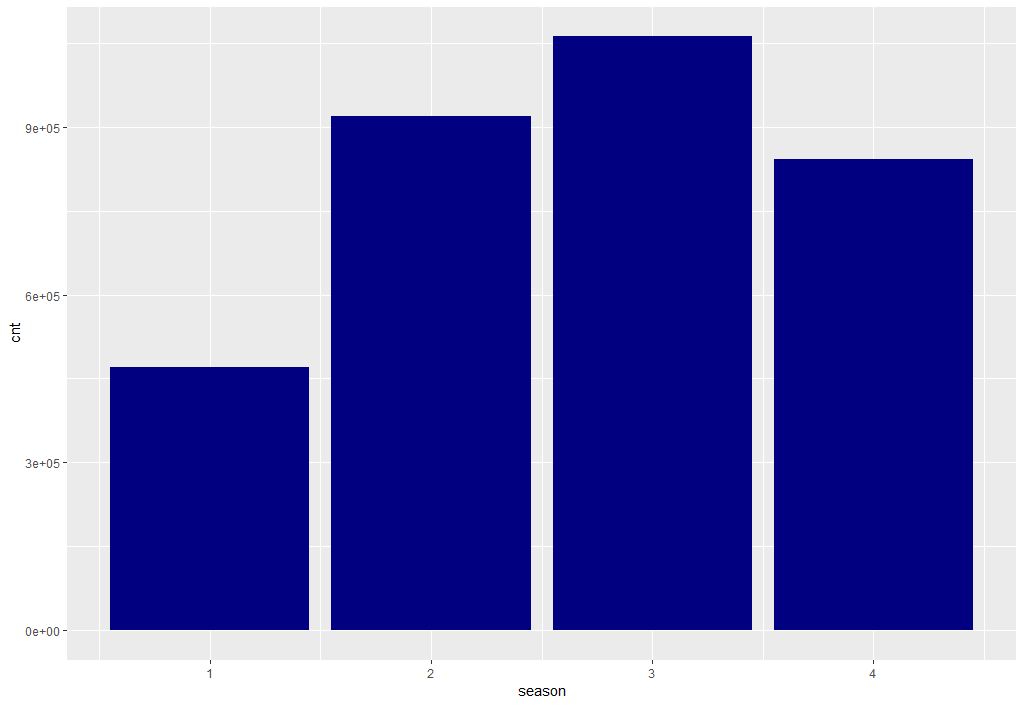


1. **Below graph is similar like above but this is a bar graph of each unique date with cnt.**

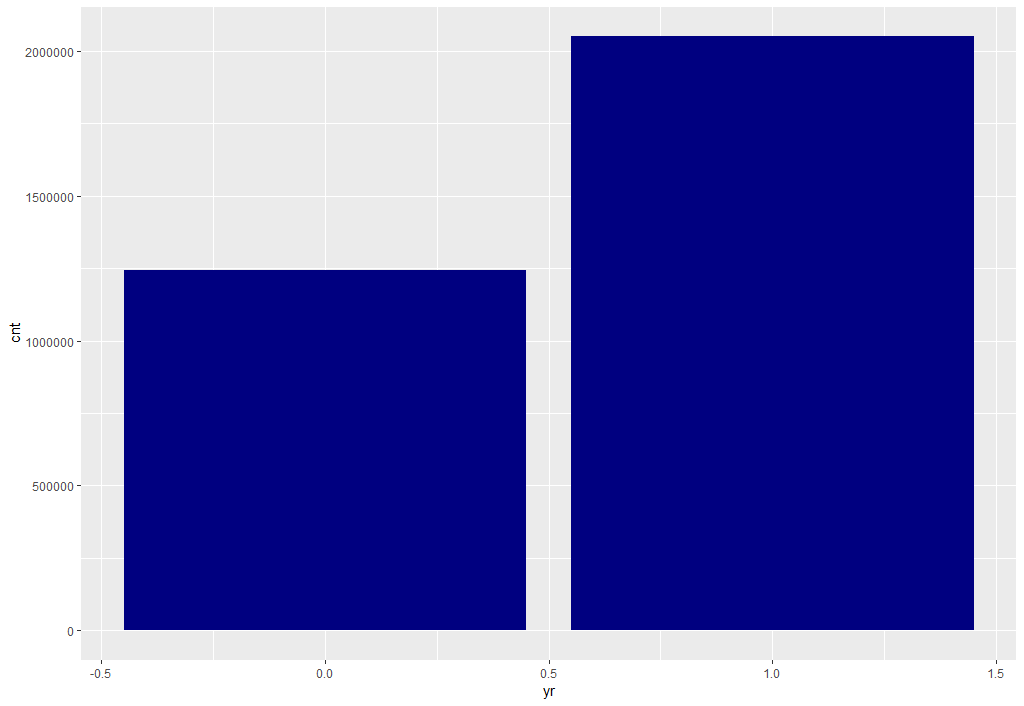


1. Season (1:springer, 2:summer, 3:fall, 4:winter)

**From below graph we can see that in fall (Rainy) season more bikes have been rented.**



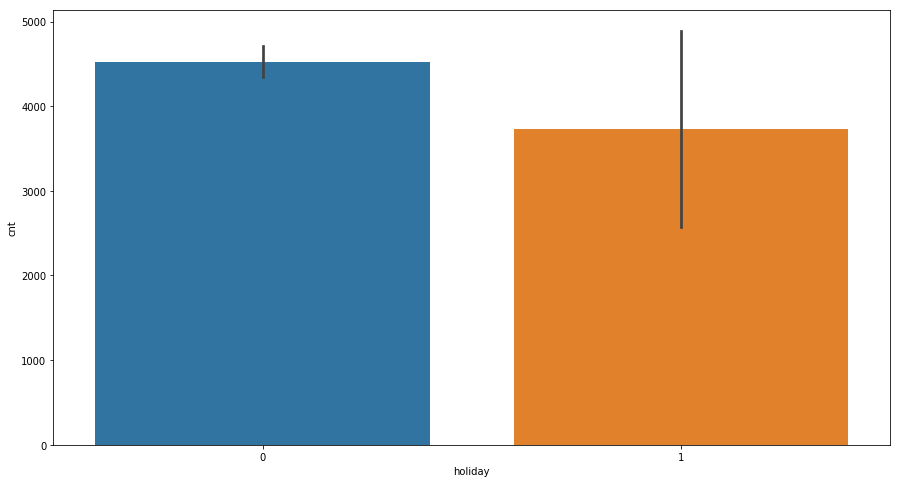
1. **In 2012**  **more bikes have been rented.**



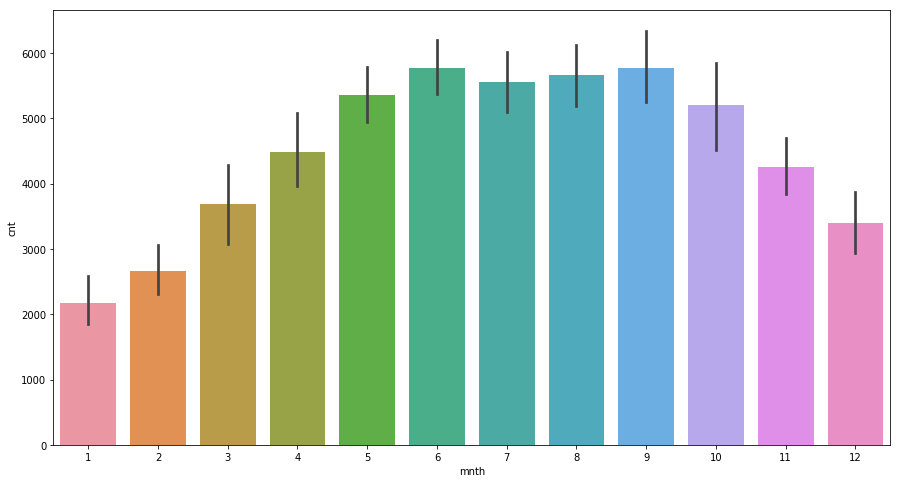


****

**By comparing above and below graph there I much less holiday in one year but from below bar graph in those holidays customer take bike in Rent more as compare to non holiday in Ratio.**

****

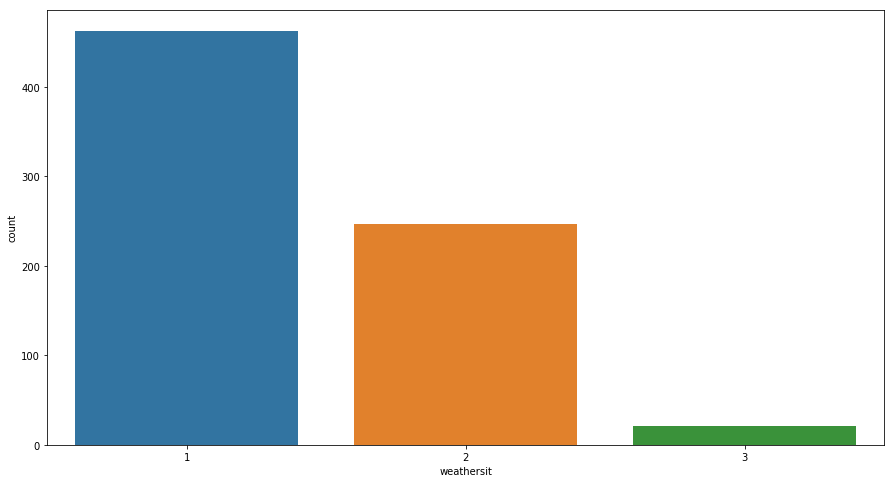
1. **In mid year from July to October customer take more bike in Rent.**

****

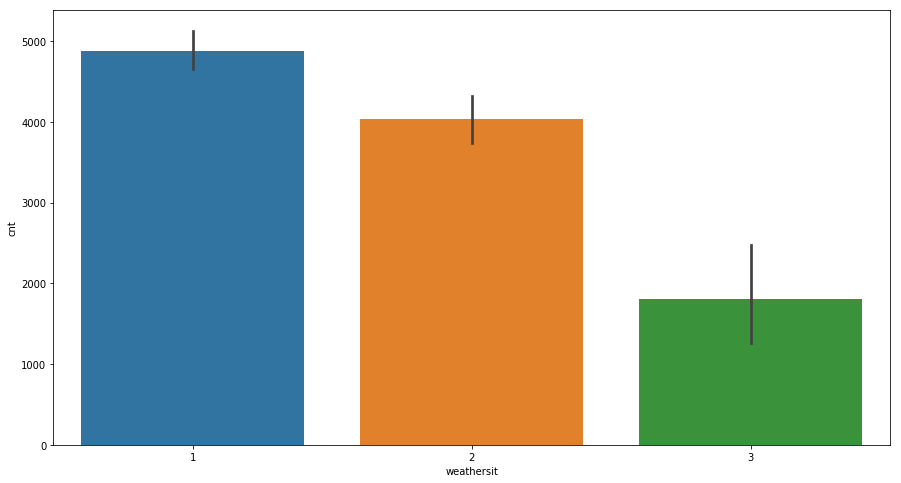
1. 1: Clear, Few clouds, Partly cloudy, Partly cloudy

2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

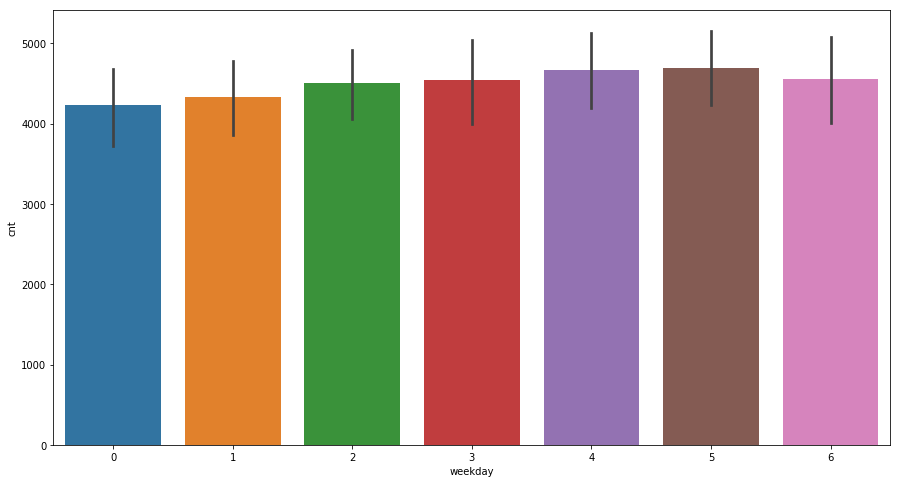
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds

****

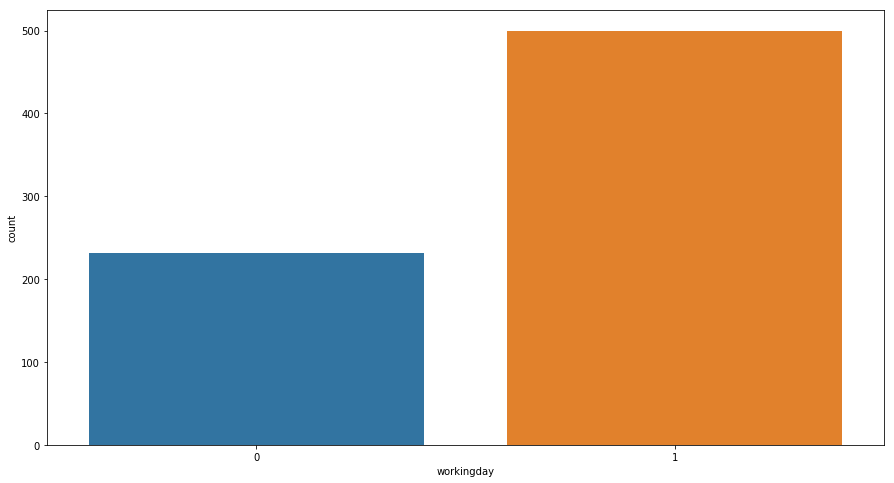
**In clear sky with few clouds more bike is rented**

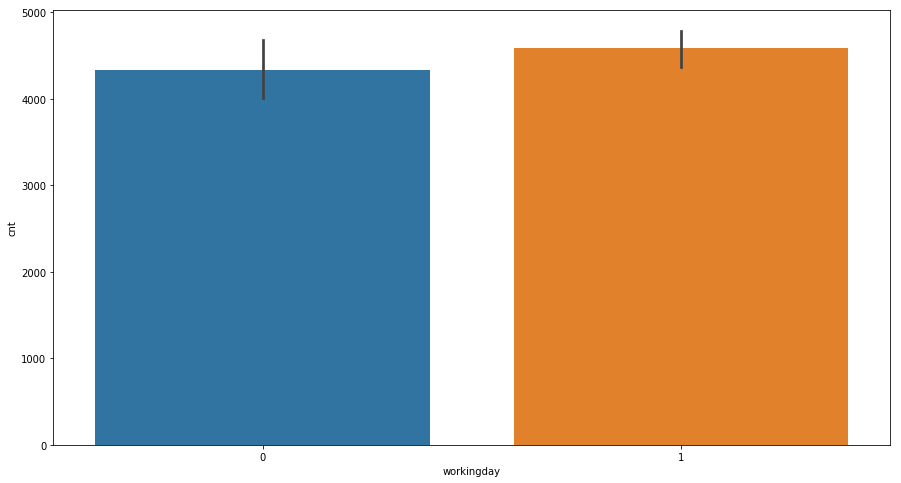
****

1. **In Saturday more bike is rented.**

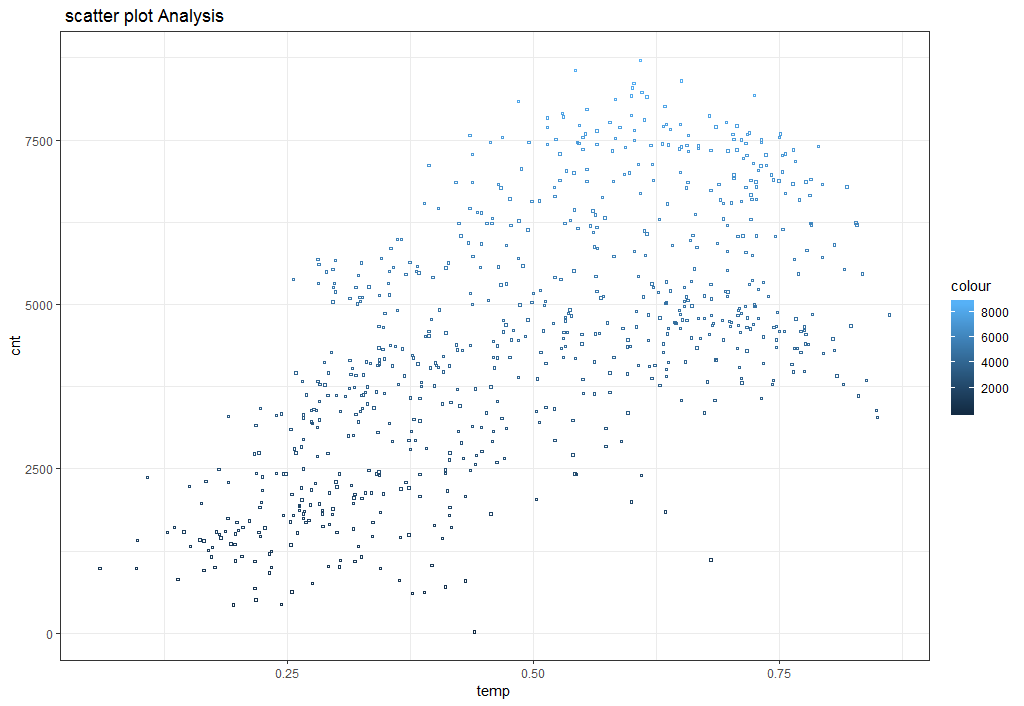
****

1. **In working day is more in a year but bike is rented more in holiday in Ratio by comparing the below two graph.**

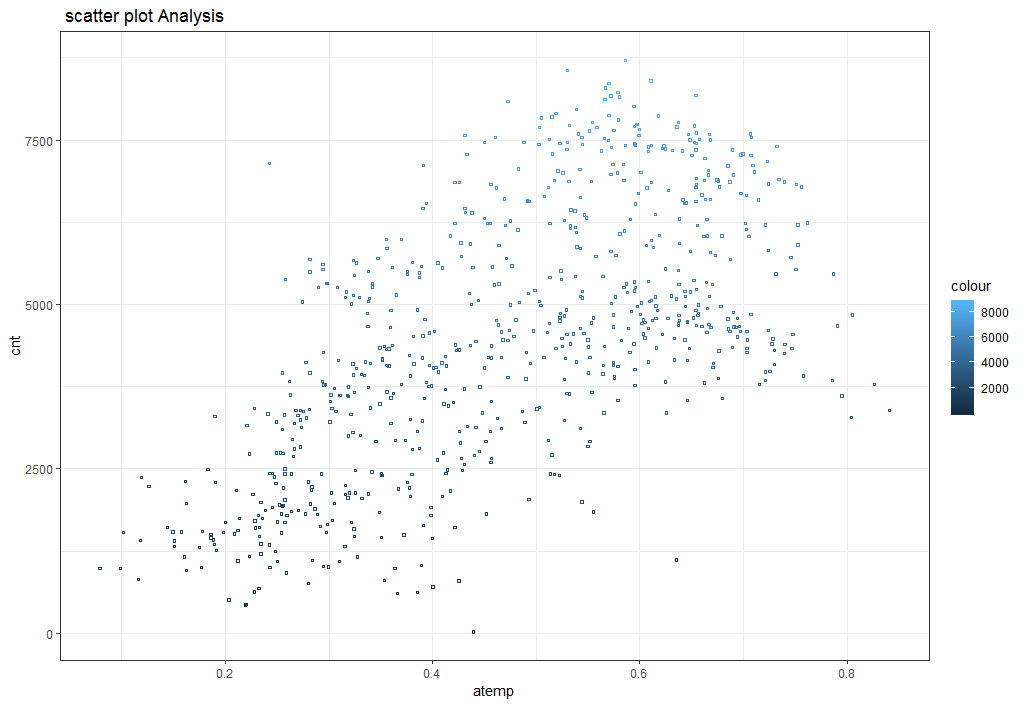
****

****

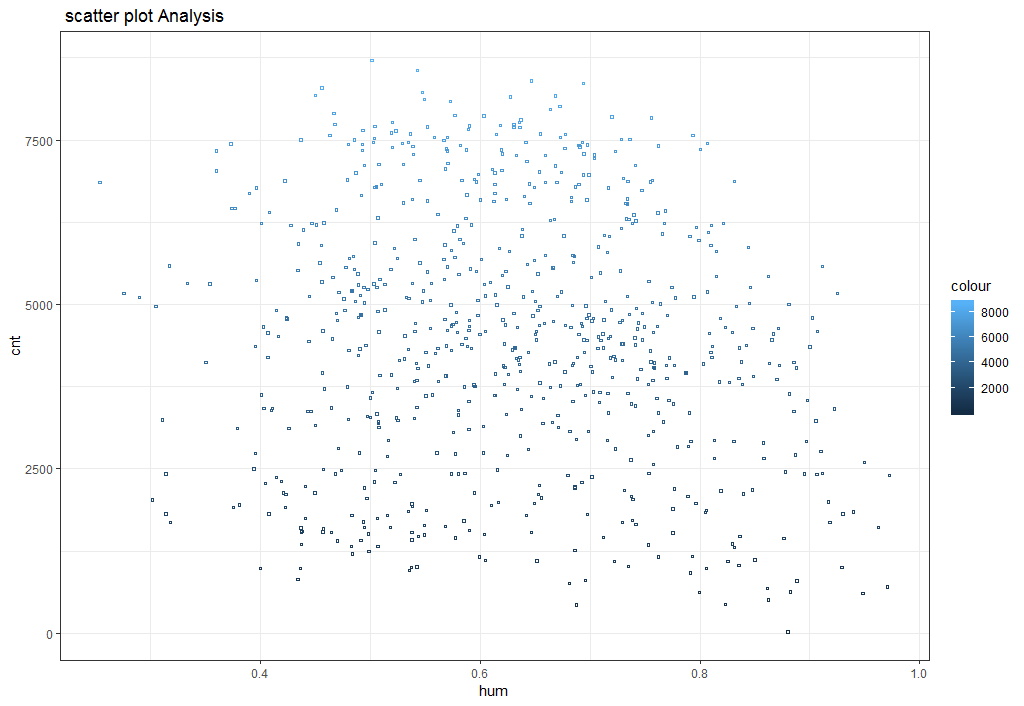
1. **Positive correlation between “temp” and “cnt”**



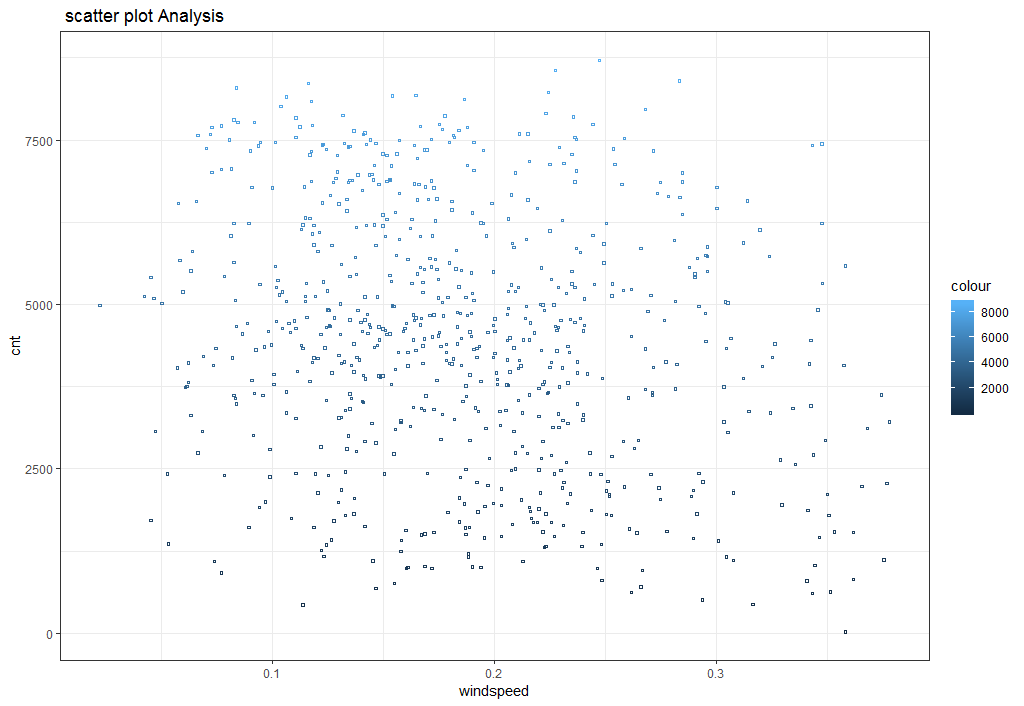
1. **Positive correlation between “atemp” and “cnt”**



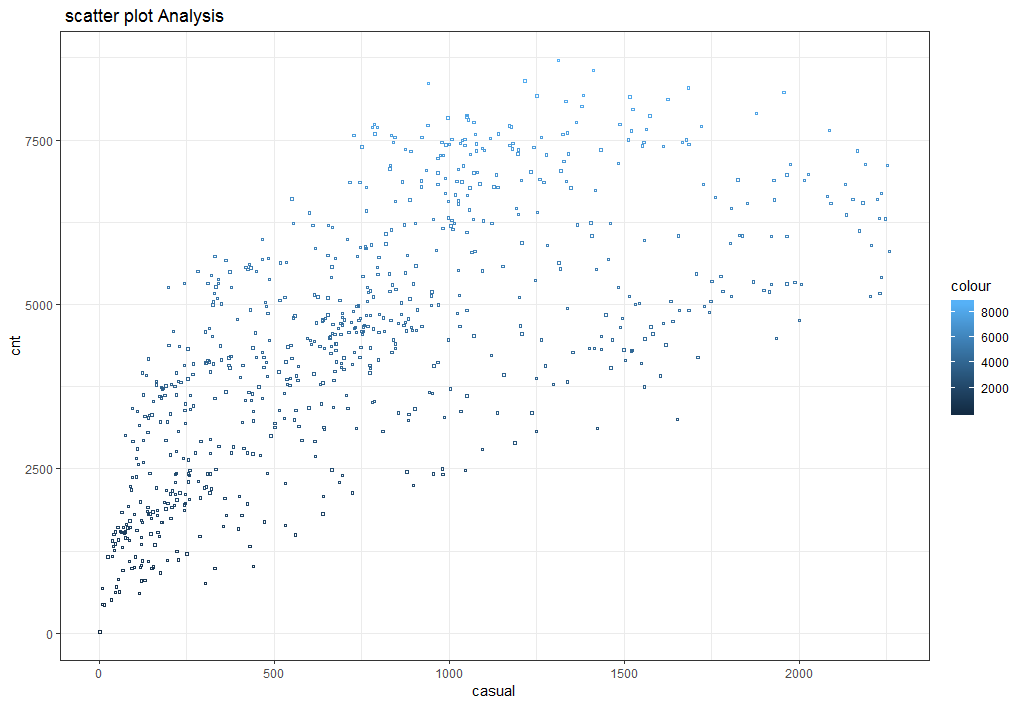
1. **No correlation between “hum” and “cnt”**



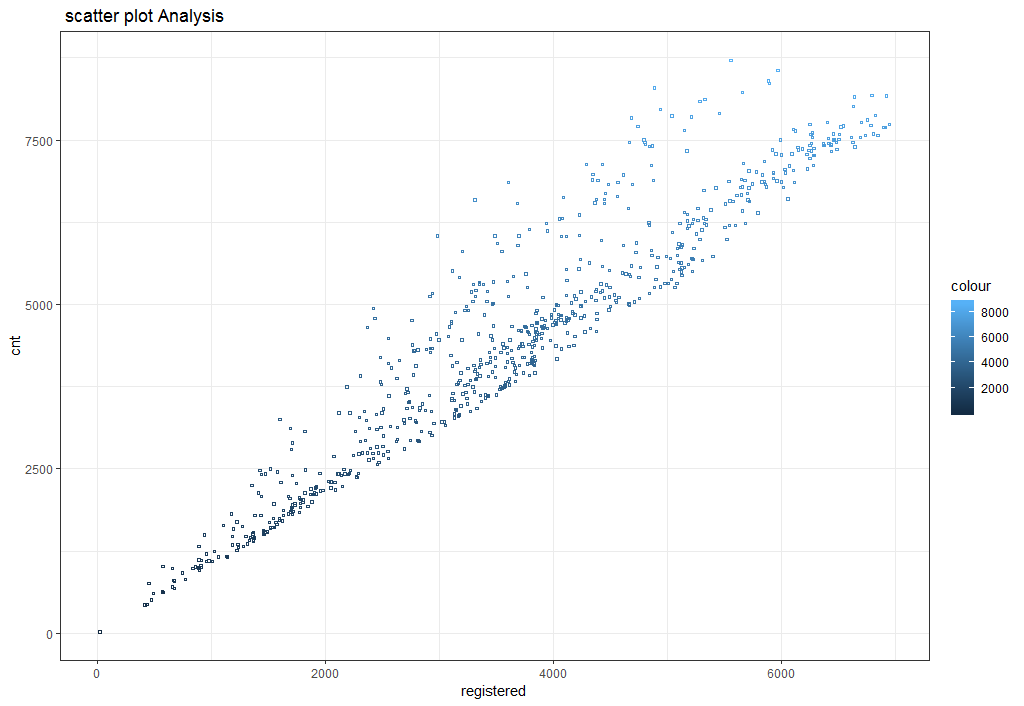
1. **No correlation between “windspeed” and “cnt”**



1. **A slightly positive correlation between “casual’ and “ cnt”.**



1. **A high positive correlation between “registered” and “cnt”.**



**2.2 Modeling**

After a thorough preprocessing we will be using some regression models on our processed data to predict the target variable. Following are the models which we have built –

**2.2.1 Liner Regression**

Linear Regression is one of the statistical methods of prediction. It is applicable only on continuous data. To build any model we have some assumptions to put on data and model. Here are the assumptions to the linear regression model.

|  |  |  |
| --- | --- | --- |
| **Linear Regression** | **R** | **PYTHON** |
| **RMSE Train** | 265.3598716 | 348.6541043531216 |
| **RMSE Test** | 212.2170503 | 387.8544256571429 |
| **R^2 Test** | 0.9807779 | 0.9646306127764841 |

**2.2.2 Decision Tree**

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Each branch connects nodes with “and” and multiple branches are connected by “or”. It can be used for classification and regression. It is a supervised machine learning algorithm. Accept continuous and categorical variables as independent variables. Extremely easy to understand by the business users. Split of decision tree is seen in the below tree. The RMSE value and R^2 value for our project in R and Python are –

|  |  |  |
| --- | --- | --- |
| **Decision Tree** | **R** | **PYTHON** |
| **RMSE Train** | 503.005411 | 0.0 |
| **RMSE Test** | 598.3854972 | 872.348507402487 |
| **R^2 Test** | 0.9821495 | 0.8210753906837198 |

**2.2.3 Random Forest**

Random Forest is an ensemble technique that consists of many decision trees. The idea behind Random Forest is to build n number of trees to have more accuracy in dataset. It is called random forest as we are building n no. of trees randomly. In other words, to build the decision trees it selects randomly n no of variables and n no of observations to build each decision tree. It means to build each decision tree on random forest we are not going to use the same data. The RMSE value and R^2 value for our project in R and Python are –

|  |  |  |
| --- | --- | --- |
| **Random Forest** | **R** | **PYTHON** |
| **RMSE Train** | 129.4719764 | 232.90808643404776 |
| **RMSE Test** | 240.1007197 | 629.230001844388 |
| **R^2 Test** | 0.9865681 | 0.9069087468979179 |

**2.2.4 Support vector Regression (SVR)**

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.

|  |  |  |
| --- | --- | --- |
| **SVR** | **R** | **PYTHON** |
| **RMSE Train** | 218.790003 | 1896.7030417458407 |
| **RMSE Test** | 221.1328272 | 2055.676382291139 |
| **R^2 Test** | 0.9880324 | 0.006427584876783965 |

**2.2.5 Gradient boosting**

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

|  |  |
| --- | --- |
| **Gradient boosting** | **PYTHON** |
| **RMSE Train** | 218.7014471159227 |
| **RMSE Test** | 517.4852151253407 |
| **R^2** | 0.9381141770882626 |

**Chapter 3**

**Conclusion**

In this chapter we are going to evaluate our models, select the best model for our dataset to predict the target variable with good accuracy.

**3.1 Model Evaluation**

In the previous chapter we have seen the **Root Mean Square Error** (RMSE) and **R-Squared** Value of different models. **Root Mean Square Error** (RMSE) is the standard deviation of the residuals (prediction **errors**). Residuals are a measure of how far from the regression line data points are, RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Whereas **R**-**squared** is a relative measure of fit, **RMSE** is an absolute measure of fit. As the square root of a variance, **RMSE** can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of **RMSE** and higher value of **R-Squared Value** indicate better fit.

**3.2 Model Selection**

**In case of Python:**

From the observation of all **RMSE Value** and **R-Squared** Value we have concluded that **Linear Regression Model**  has minimum value of RMSE test and it’s **R-Squared** Value is also maximum (i.e. 0.9646) in Python.

The RMSE value of Testing data and Training does not differs a lot this implies that it is not the case of overfitting.

**Therefore Multiple Linear Regression is good model in Python.**

**In case of R:**

**Support Vector Regression has 2nd minimum RMSE test value (**221.13) **after Linear Regression RMSE test value (**212.21**) .**

**But RMSE train value in case of Linear Regression is 265.35, seems like model is under fitting.**

**And RMSE train value in case of SVR is 218.79 ,which seems like model is not under fitting as well as the RMSE value of Testing data and Training does not differs a lot this implies that it is not the case of overfitting. Its R^2 values is 0.9880324 which is maximum.**

**Therefore, SVR is good model in R.**