Bike Rental Count Report

Author: Abhishek Goswami

15/10/2019

Contents

Chapter 1

Introduction:

Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become advanced. Through these systems user is able to easily rent a bike from a particular position and return back at another position. Currently there are about 500 bike-sharing programs around the world which is composed of over 500 thousands of bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

Bike-sharing rental process is highly correlated to the environmental and seasonal settings. For Instance, weather conditions, precipitation, day of week, season etc. can affect the rental behaviours. The task given this assignment is predication of bike rental count daily based on the Environmental and seasonal settings.

1.1 Problem Statement:

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.

1.2 Data:

Understanding of data is the very first and important step in the process of finding solution of any business problem. Here in our case we were provided with dataset with following features, we need to go through each and every variable of it to understand and for better functioning.

Size of Dataset Provided: - 731 rows, 17 Columns (including dependent variable)

Missing Values: No

Outliers Presented: Yes

Below mentioned is a list of all the variable names with their meanings:

- **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.
- **atemp**: Normalized feeling temperature in Celsius.
- **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

Chapter 2

Methodology:

❖ 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. EDA involves some of the steps below mentioned.

- Data exploration and Cleaning
- Missing value treatment
- Outlier Analysis
- Feature Selection
- Features Scaling
- Visualization
- Skewness and Log transformation

❖ Modelling

Once all the Pre-Processing steps has been done on our data set, we will now further move to our next step which is modelling. Modelling plays an important role to find out the good inferences from the data. Choice of models depends upon the problem statement and data set. As per our problem statement and dataset, we will try some models on our preprocessed data and post comparing the output results we will select the best suitable model for our problem. As per our data set following models need to be tested:

- Linear regression
- Decision Tree
- Random forest

❖ Model Selection

The final step of our methodology will be the selection of the model based on the different output and results shown by different models. We have multiple parameters which we will study further in our report to test whether the model is suitable for our problem statement or not.

Chapter 3

Pre-Processing

3.1 Data exploration and Cleaning (Missing Values and Outliers)

The very first step which comes with any data science project is data exploration and cleaning which includes following points as per this project:

1. Separate the combined variables like dteday
2. Remove the outliers
3. Convert to proper datatypes

3.2 Creating some new variables out of the given variable.

Here in our data set our variable name **dteday** contains date which is of type timestamp. So we tried to extract some important variables from this variable:

date : The date (extracted from the given column).
So our new set of variables are:

- Instant
- dteday
- season
- yr
- mnth
- hr
- holiday
- weekday
- workingday
- weathersit
- temp
- atemp

- hum
- windspeed
- casual
- registered
- cnt
- date

3.3 Selection of variables

Now as we have extracted the meaningful information from the given variables so we will drop the redundant variables which are as follows:

- Instant: which is nothing but the index.
- dteday: as date have extracted date , mnth and yr are already present

Dropping is done later after Outlier Analysis.

Some more data exploration

In this report we are trying to predict the count of bikes of a bike rental company. So here we have a data set of 731 observations with 17 variables including one dependent variable.

3.4.1 Below are the names of Independent variables:

Instant, dteday, season, yr, mnth, hr, holiday, weekday, workingday, weathersit, temp, atemp, hum, windspeed, casual, registered, date

Our Dependent variable is: **cnt**

3.4.2 Dividing the variables into two categories on basis of their data types:

Continuous variables – ‘temp', 'atemp’, ‘hum’, ‘windspeed’, ‘casual’, ‘registered’, ‘date’

Categorical Variables - 'season', 'yr', 'mnth', 'hr', 'holiday', ‘weekday’, ‘workingday’, ‘weathersit’

3.4.3 Exploratory Data Analysis

In exploring the data, we have

While doing EDA converted season, yr, mnth, holiday, workingday, weekday, weathersit into

Categorical variable.

Missing Value Analysis

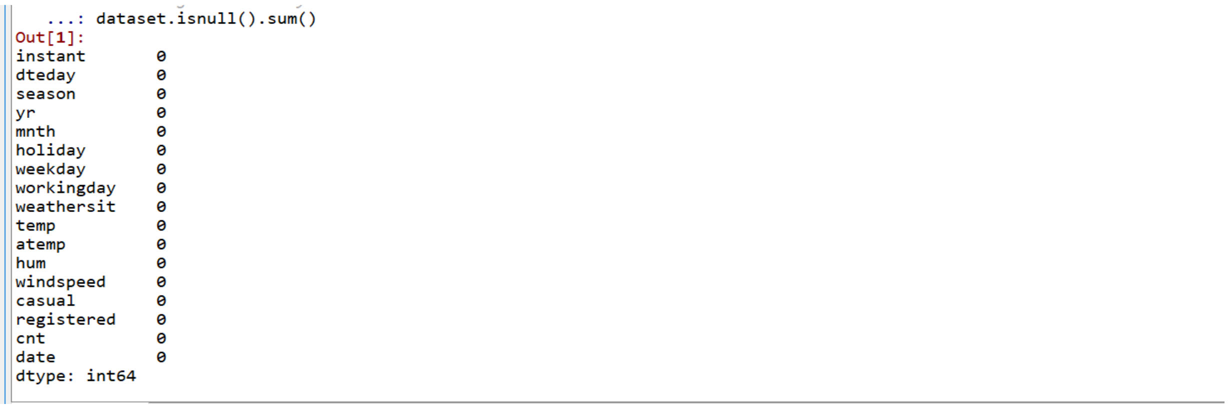
Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behavior and relationship with other variables correctly. It can lead to wrong prediction or classification.

Treating Missing Values:-

First we will check how much percentage of the data has missing value, if it is greater than 30% then we will discard the data set. On the other hand, if it is not then either we can delete those observation or can impute it. Deleting observation may lead to loss of important information so it is better to impute it. These are the following ways to impute a missing value:-

- Deletion
- Mean/Mode/Median Imputation

In our data there are no missing values present.



```
   ...: dataset.isnull().sum()
Out[1]:
instant       0
dteday        0
season        0
yr            0
mnth          0
holiday       0
weekday       0
workingday    0
weathersit    0
temp          0
atemp         0
hum           0
windspeed     0
casual        0
registered    0
cnt           0
date          0
dtype: int64
```

Outlier Analysis

Outlier is a commonly used terminology used by analysts and data scientists as it needs close attention else it can result in wildly wrong estimation. Simply speaking, Outlier is an observation that appears far away and diverge from an overall pattern in a sample.

Outlier can be of two types: Univariate and Multivariate. Here we have discussed the example of univariate outlier. These outliers can be found when we look at distribution of a single variable. Multivariate outliers are outliers in an n-dimensional space. In order to find them, we have to look at distributions in multi-dimensions.

Below are the box plots of variables hum, windspeed and casual each of which contains outliers In them:

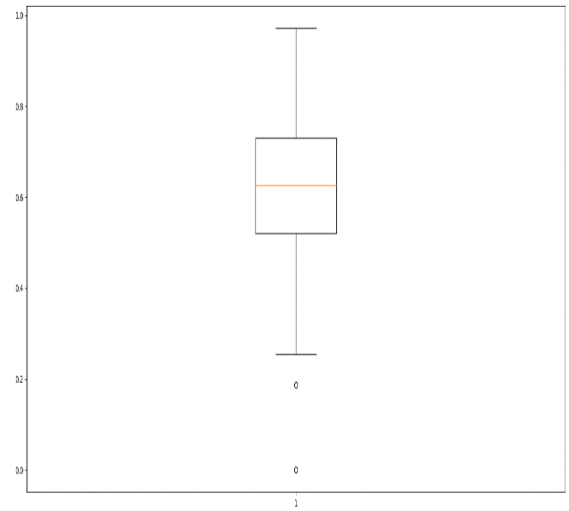


Figure 1 HUM

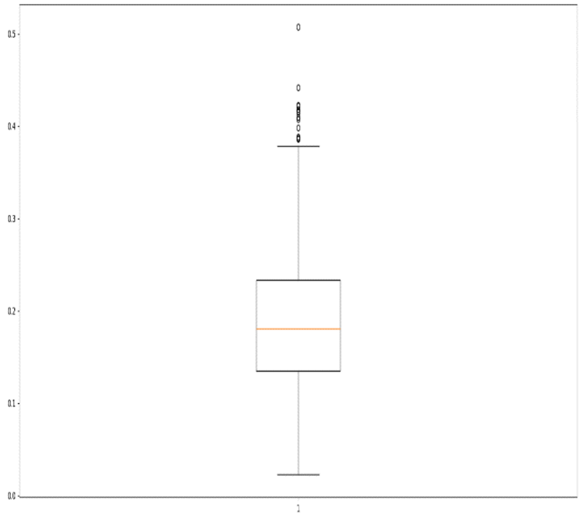


Figure 2 WINDSPEED

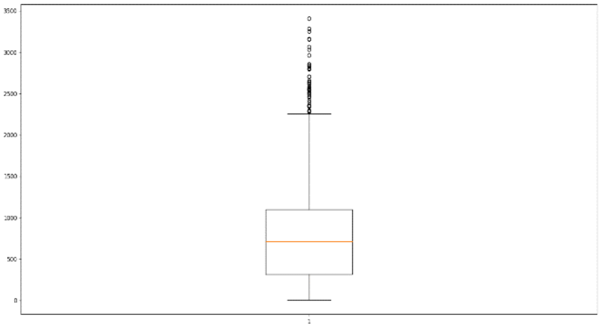


Figure 3 CASUAL

After removal of outliers

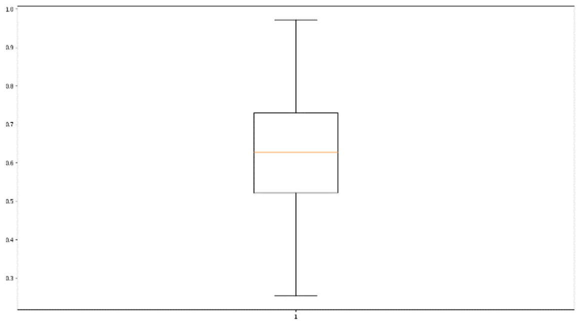


Figure 1 HUM

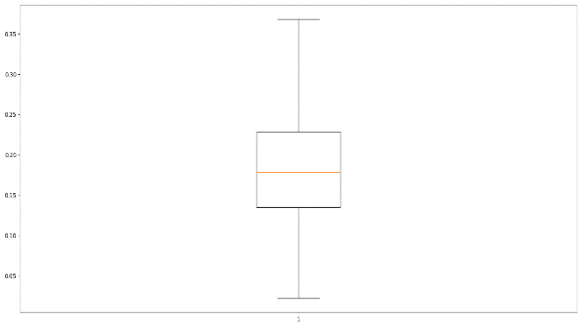


Figure 2 WINDSPEED

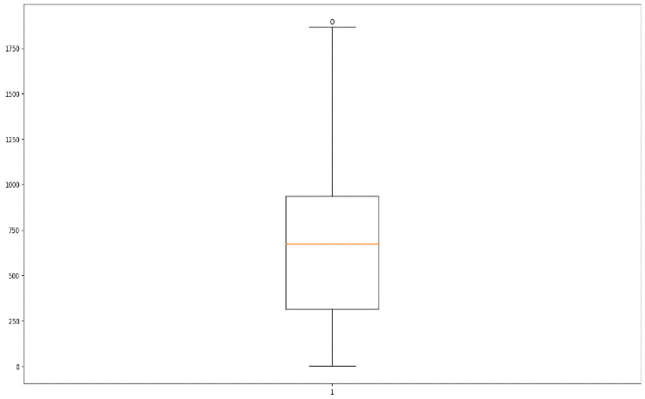
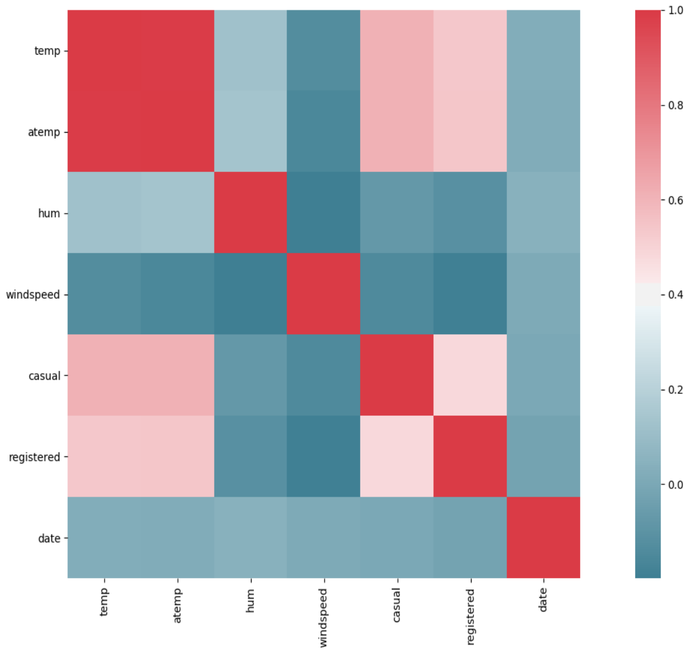


Figure 3 CASUAL

Feature Selection

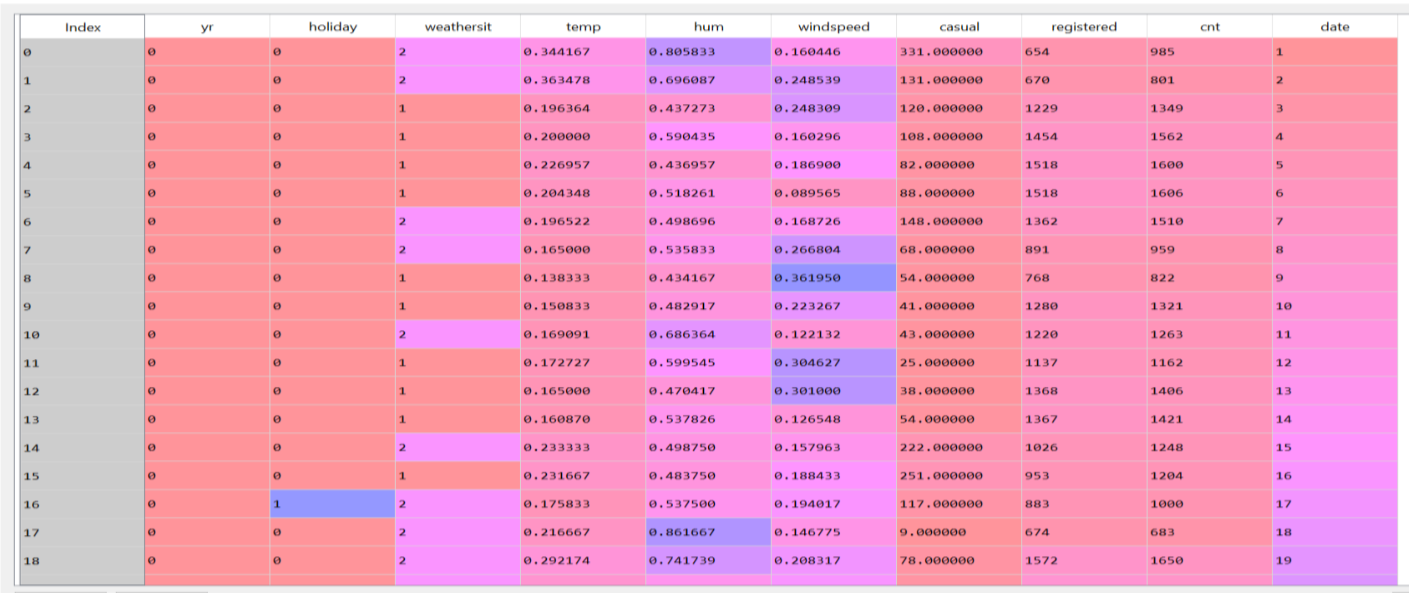
Here in feature selection we plot the correlational analysis of the continuous predictor variables. We look for low correlation between the predictors. If there is high correlation between two Predictor variables then we will delete one of them. This is called Dimension Reduction. Secondly, we do chi-square test among categorical variables to check whether the predictors are independent or not. If p-value of a categorical variable w.r.t to another categorical variable is less than 0.05 then we have to reject the null hypothesis and say that the variables are not independent.

Below fig show the correlation between our predictors:



As we can see above temp and atemp variables are highly correlated to one another. Hence we need to remove one from our dataset.

Also, through chi square test we find out that variables season, mnth, and weathersit are dependent on each other and variables holiday, weekday and workingday are also dependent. Hence, we need to remove such variables. We decide to keep temp, weathersit and holiday in our dataset and remove all remaining dependent variables. Hence the final data with which we proceed further is as shown in the fig below.



Index	yr	holiday	weathersit	temp	hum	windspeed	casual	registered	cnt	date
0	0	0	2	0.344167	0.805833	0.160446	331.000000	654	985	1
1	0	0	2	0.363478	0.696087	0.248539	131.000000	670	801	2
2	0	0	1	0.196364	0.437273	0.248309	120.000000	1229	1349	3
3	0	0	1	0.200000	0.590435	0.160296	108.000000	1454	1562	4
4	0	0	1	0.226957	0.436957	0.186900	82.000000	1518	1600	5
5	0	0	1	0.204348	0.518261	0.089565	88.000000	1518	1606	6
6	0	0	2	0.196522	0.498696	0.168726	148.000000	1362	1510	7
7	0	0	2	0.165000	0.535833	0.266804	68.000000	891	959	8
8	0	0	1	0.138333	0.434167	0.361950	54.000000	768	822	9
9	0	0	1	0.150833	0.482917	0.223267	41.000000	1280	1321	10
10	0	0	2	0.169091	0.686364	0.122132	43.000000	1220	1263	11
11	0	0	1	0.172727	0.599545	0.304627	25.000000	1137	1162	12
12	0	0	1	0.165000	0.470417	0.301000	38.000000	1368	1406	13
13	0	0	1	0.160870	0.537826	0.126548	54.000000	1367	1421	14
14	0	0	2	0.233333	0.498750	0.157963	222.000000	1026	1248	15
15	0	0	1	0.231667	0.483750	0.188433	251.000000	953	1204	16
16	0	1	2	0.175833	0.537500	0.194017	117.000000	883	1000	17
17	0	0	2	0.216667	0.861667	0.146775	9.000000	674	683	18
18	0	0	2	0.292174	0.741739	0.208317	78.000000	1572	1650	19

.

3.5. Feature Scaling

Data Scaling is very important step in data pre-processing when we are dealing with variables which have different scales. Scaling of such variables are required otherwise they will cause anomaly in the final model as the variable with larger scales will completely dominate the model result. There are two ways of scaling:

- Normalization
- Standardization (z-score)

However, the numeric variables given to us are already scaled hence feature scaling is not Required here.

Chapter 4

Modelling:

In this case we have to predict the count of bike renting according to environmental and seasonal condition. So the target variable here is a continuous variable. For Continuous we can use various Regression models. Model having less error rate and more accuracy will be our final model.

Models built are: -

- Linear regression
- c50 (Decision tree for regression target variable)
- Random Forest

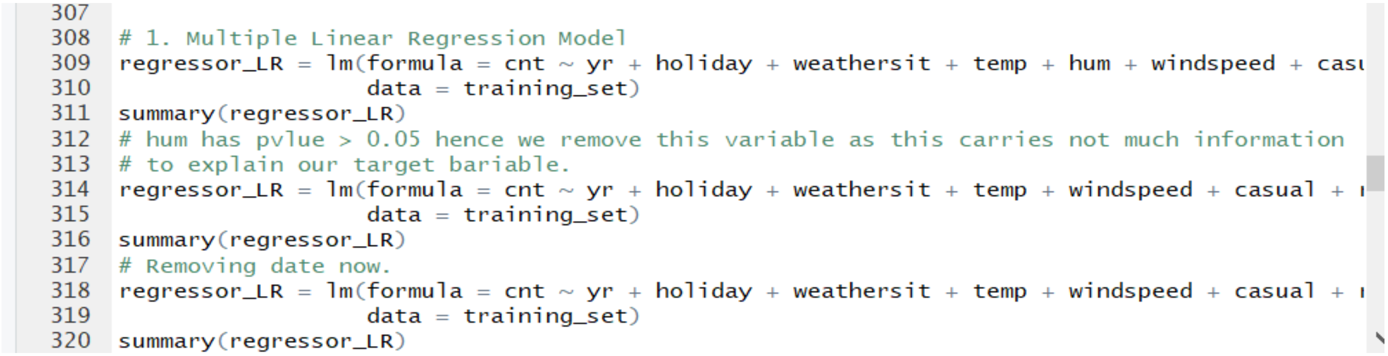
Before running any model, we will split our data into two parts which is train and test data. In our case we have taken 80% of the data as train data and 20% of the data as test data.

4.1 Linear Regression

It is one of the most widely known modeling technique. Linear regression is usually among the first few topics which people pick while learning predictive modeling. In this technique, the dependent variable is continuous, independent(s) can be continuous or discrete and nature of regression line is Linear.

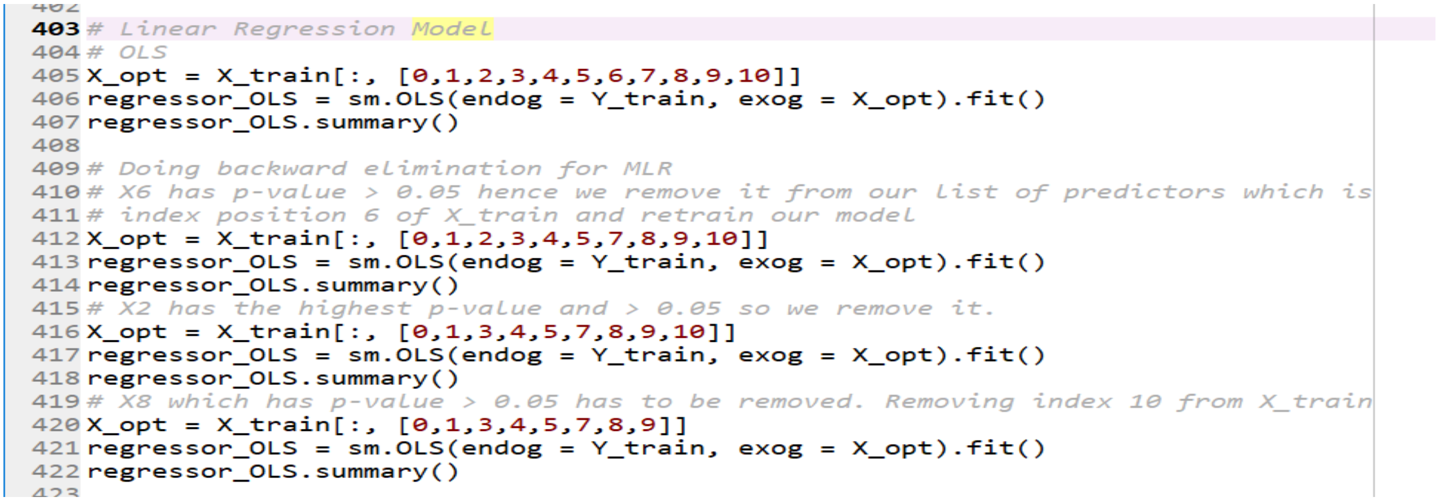
Creating Model: -

In R:



```
# 1. Multiple Linear Regression Model
regressor_LR = lm(formula = cnt ~ yr + holiday + weathersit + temp + hum + windspeed + casu
                  data = training_set)
summary(regressor_LR)
# hum has pvlue > 0.05 hence we remove this variable as this carries not much information
# to explain our target bariable.
regressor_LR = lm(formula = cnt ~ yr + holiday + weathersit + temp + windspeed + casual +
                  data = training_set)
summary(regressor_LR)
# Removing date now.
regressor_LR = lm(formula = cnt ~ yr + holiday + weathersit + temp + windspeed + casual +
                  data = training_set)
summary(regressor_LR)
```

In Python:



```
# Linear Regression Model
# OLS
X_opt = X_train[:, [0,1,2,3,4,5,6,7,8,9,10]]
regressor_OLS = sm.OLS(endog = Y_train, exog = X_opt).fit()
regressor_OLS.summary()

# Doing backward elimination for MLR
# X6 has p-value > 0.05 hence we remove it from our list of predictors which is
# index position 6 of X_train and retrain our model
X_opt = X_train[:, [0,1,2,3,4,5,7,8,9,10]]
regressor_OLS = sm.OLS(endog = Y_train, exog = X_opt).fit()
regressor_OLS.summary()
# X2 has the highest p-value and > 0.05 so we remove it.
X_opt = X_train[:, [0,1,3,4,5,7,8,9,10]]
regressor_OLS = sm.OLS(endog = Y_train, exog = X_opt).fit()
regressor_OLS.summary()
# X8 which has p-value > 0.05 has to be removed. Removing index 10 from X_train
X_opt = X_train[:, [0,1,3,4,5,7,8,9]]
regressor_OLS = sm.OLS(endog = Y_train, exog = X_opt).fit()
regressor_OLS.summary()
```

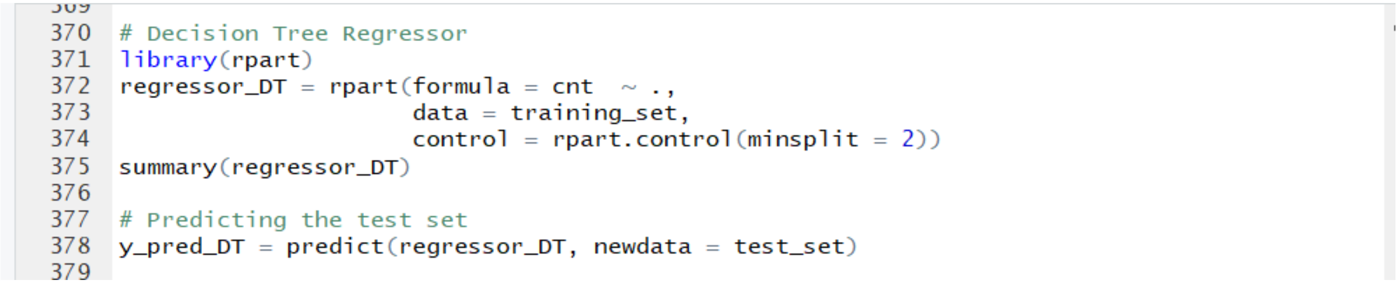
4.2 Decision Tree

This model is also known as Decision tree for regression target variable.

For this model we have divided the dataset into train and test part using random sampling. Where train contains 80% data of dataset and test contains 20% data of dataset. Decision Tree is a nonlinear and non-continuous model.

Creating Model: -

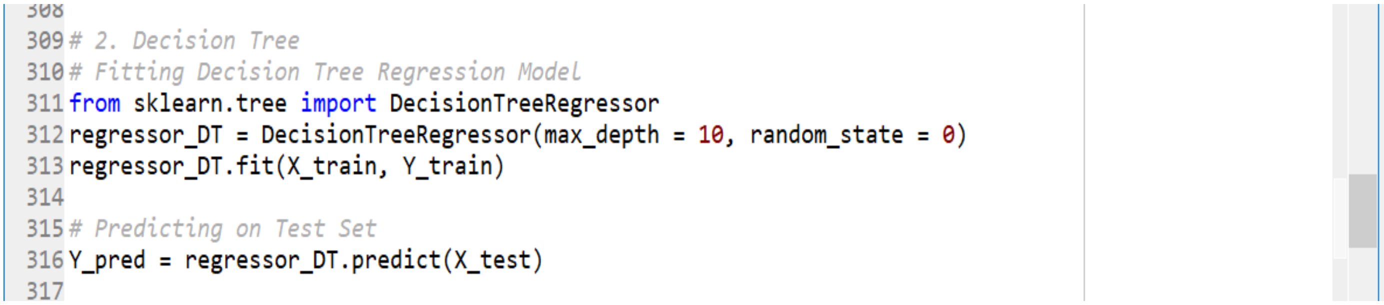
In R:



```
# Decision Tree Regressor
library(rpart)
regressor_DT = rpart(formula = cnt  ~ .,
                     data = training_set,
                     control = rpart.control(minsplit = 2))
summary(regressor_DT)

# Predicting the test set
y_pred_DT = predict(regressor_DT, newdata = test_set)
```

In python:



```
# 2. Decision Tree
# Fitting Decision Tree Regression Model
from sklearn.tree import DecisionTreeRegressor
regressor_DT = DecisionTreeRegressor(max_depth = 10, random_state = 0)
regressor_DT.fit(X_train, Y_train)

# Predicting on Test Set
Y_pred = regressor_DT.predict(X_test)
```

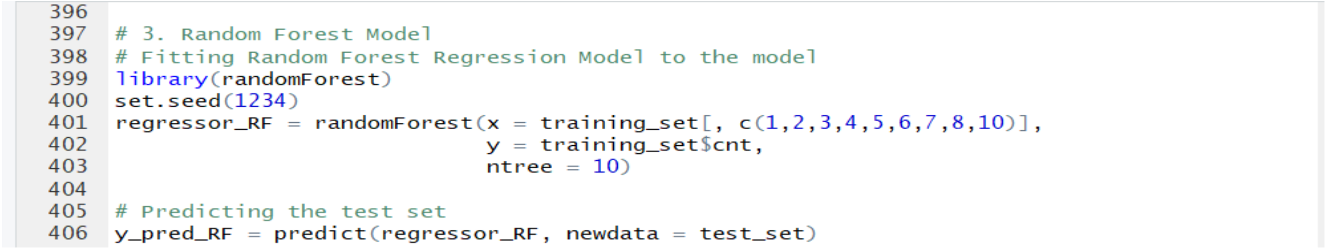
4.3 Random Forest

Random forests or random decision forests is an ensemble learning method for classification, regression and other task, which operates 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 all the individual trees.

To say it in simple words: Random forest builds multiple decision trees and Outputs the prediction voted by all the individual tress

Creating Model: -

In R:

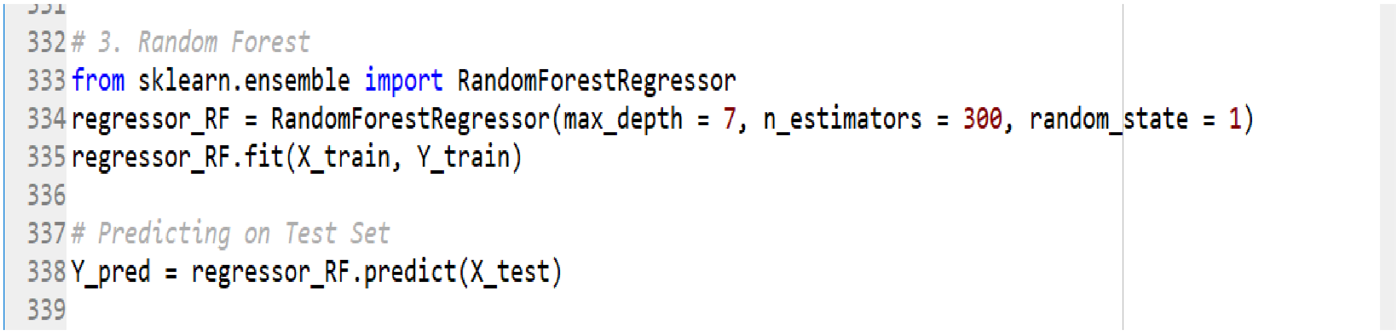


```r

# 3. Random Forest Model
# Fitting Random Forest Regression Model to the model
library(randomForest)
set.seed(1234)
regressor_RF = randomForest(x = training_set[, c(1,2,3,4,5,6,7,8,10)],
                            y = training_set$cnt,
                            ntree = 10)

# Predicting the test set
y_pred_RF = predict(regressor_RF, newdata = test_set)
```

In Python:



```python
# 3. Random Forest
from sklearn.ensemble import RandomForestRegressor
regressor_RF = RandomForestRegressor(max_depth = 7, n_estimators = 300, random_state = 1)
regressor_RF.fit(X_train, Y_train)

# Predicting on Test Set
Y_pred = regressor_RF.predict(X_test)

```

Chapter 5

Conclusion

5.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

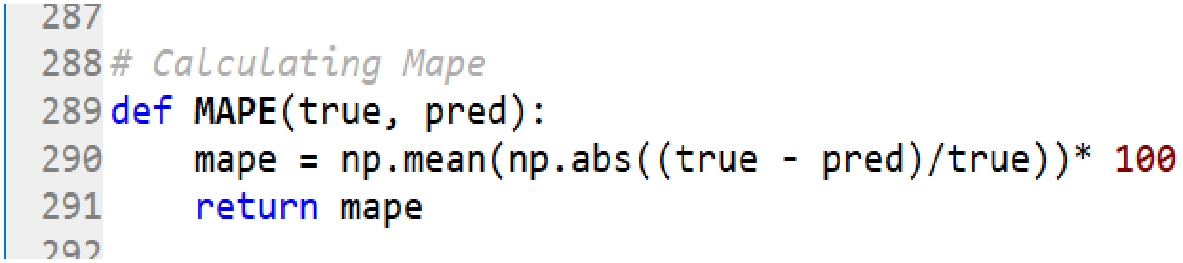
1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case of Bike Rental Predictions, the latter two, Interpretability and Computation Efficiency, do not hold much significance. Therefore, we will use Predictive performance as the criteria to compare and evaluate models.

Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

Mean Absolute Percentage Error (MAPE)

MAPE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in our project.

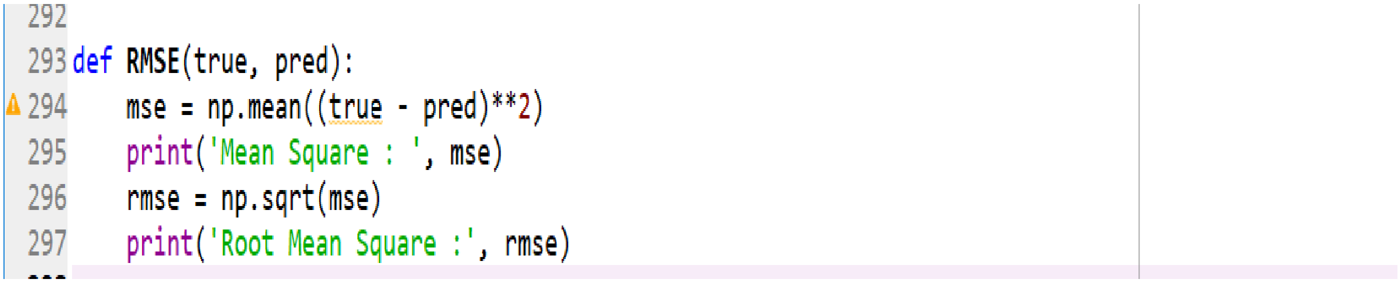


```
# Calculating Mape
def MAPE(true, pred):
    mape = np.mean(np.abs((true - pred)/true))* 100
    return mape
```

In above function ‘true’ is the actual value and ‘pred’ is the predicted value. It will provide the error percentage of model.

Root Mean Squared Error (RMSE)

RMSE is the most popular evaluation metric used in regression problems. It follows an assumption that the errors are unbiased and follow a normal distribution.

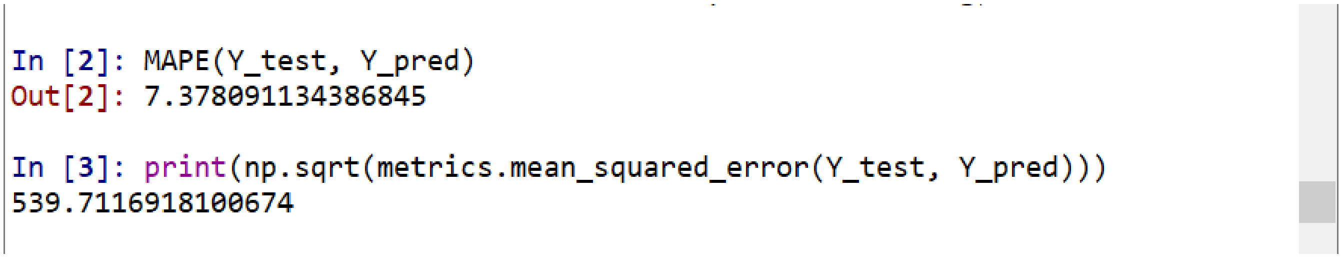


```python

def RMSE(true, pred):
    mse = np.mean((true - pred)**2)
    print('Mean Square : ', mse)
    rmse = np.sqrt(mse)
    print('Root Mean Square :', rmse)
```

MAPE, RMSE and R-squared values for different models in Python are as follows:

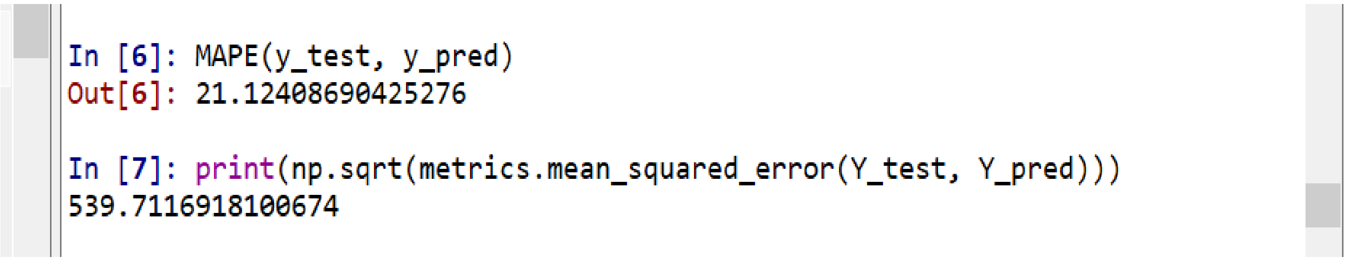
1. Linear Model



```
In [2]: MAPE(Y_test, Y_pred)
Out[2]: 7.378091134386845

In [3]: print(np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
539.7116918100674
```

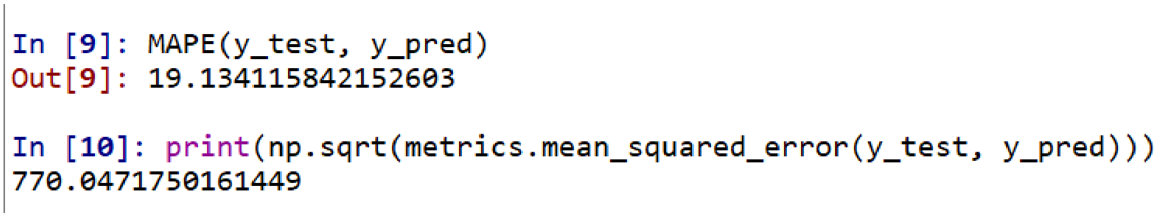
2. Decision Tree Model



```
In [6]: MAPE(y_test, y_pred)
Out[6]: 21.12408690425276

In [7]: print(np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
539.7116918100674
```

3. Random Forest Model

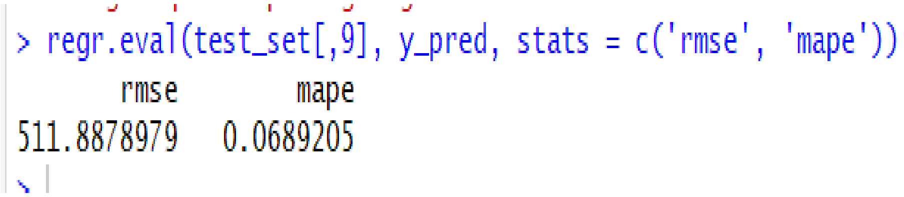


```
In [9]: MAPE(y_test, y_pred)
Out[9]: 19.134115842152603

In [10]: print(np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
770.0471750161449
```

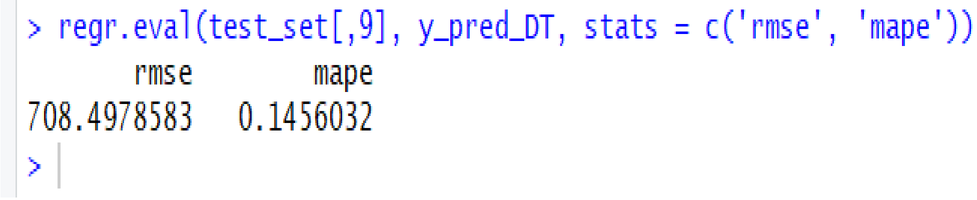
MAPE and RMSE values in R:

1. Linear Model



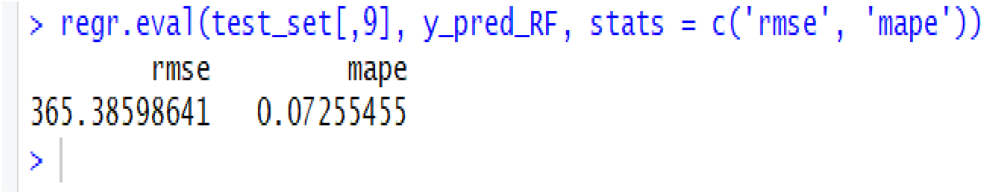
```
> regr.eval(test_set[,9], y_pred, stats = c('rmse', 'mape'))
       rmse        mape 
511.8878979   0.0689205 
> 
```

2. Decision Tree Model



```
> regr.eval(test_set[,9], y_pred_DT, stats = c('rmse', 'mape'))
       rmse        mape 
708.4978583   0.1456032 
> 
```

3. Random Forest



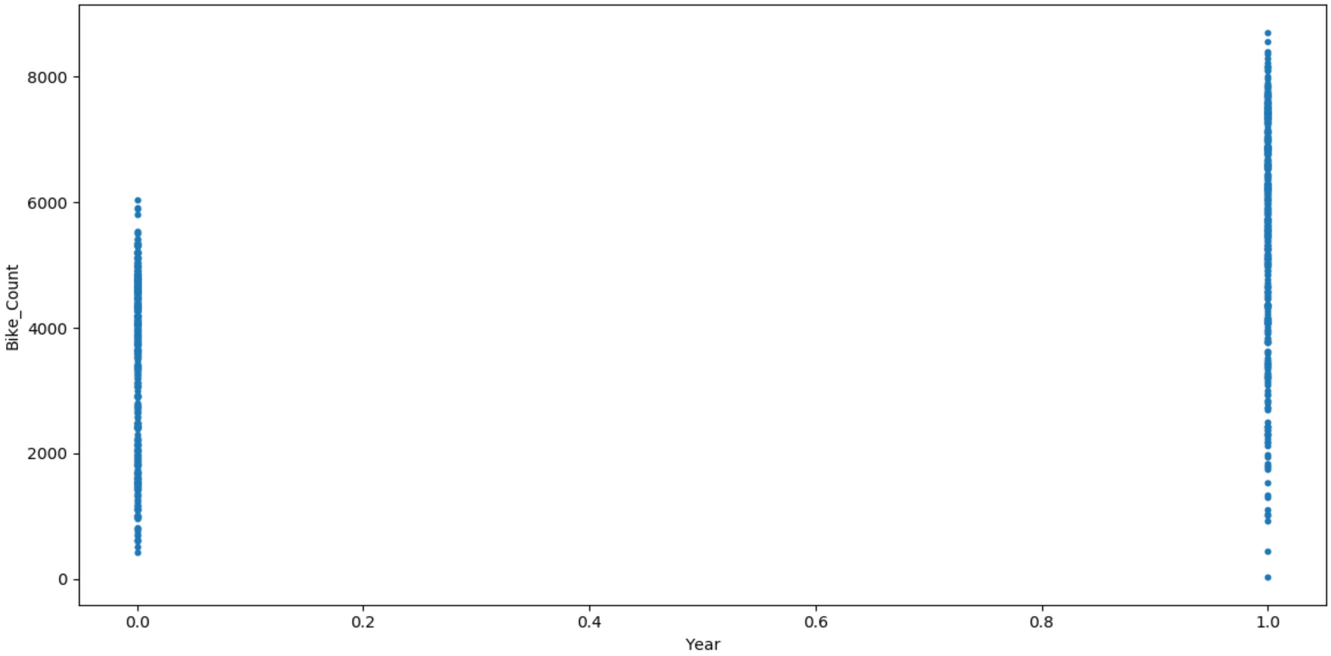
```
> regr.eval(test_set[,9], y_pred_RF, stats = c('rmse', 'mape'))
        rmse         mape 
365.38598641   0.07255455 
> 
```

5.2 Model Selection

We can see that in both R and Python Linear Regression Model fits the best out of Decision Tree and Random Forest. Mape is best for Linear Regression Model. Although the performance of random forest was also good but given the regression problem as Linear Regression has the best Mape and is a very good model for regression problems Hence, I have chosen Linear Regression Model as the best model for our problem statement.

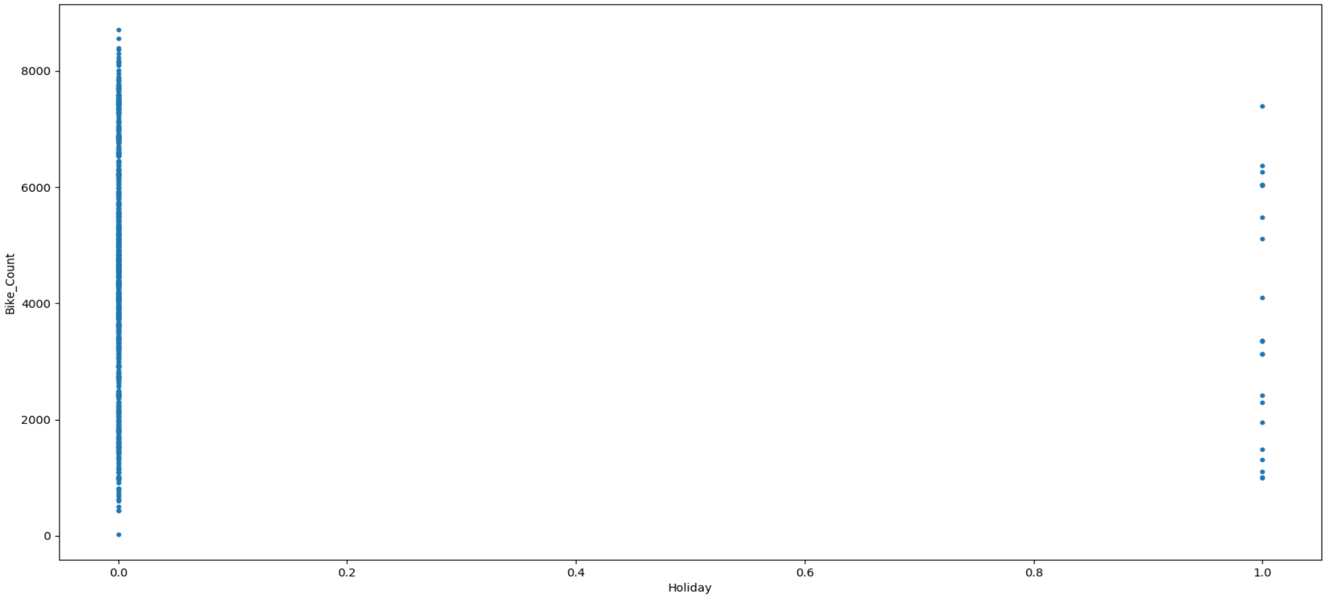
5.3 Visualization

i. yr Vs cnt



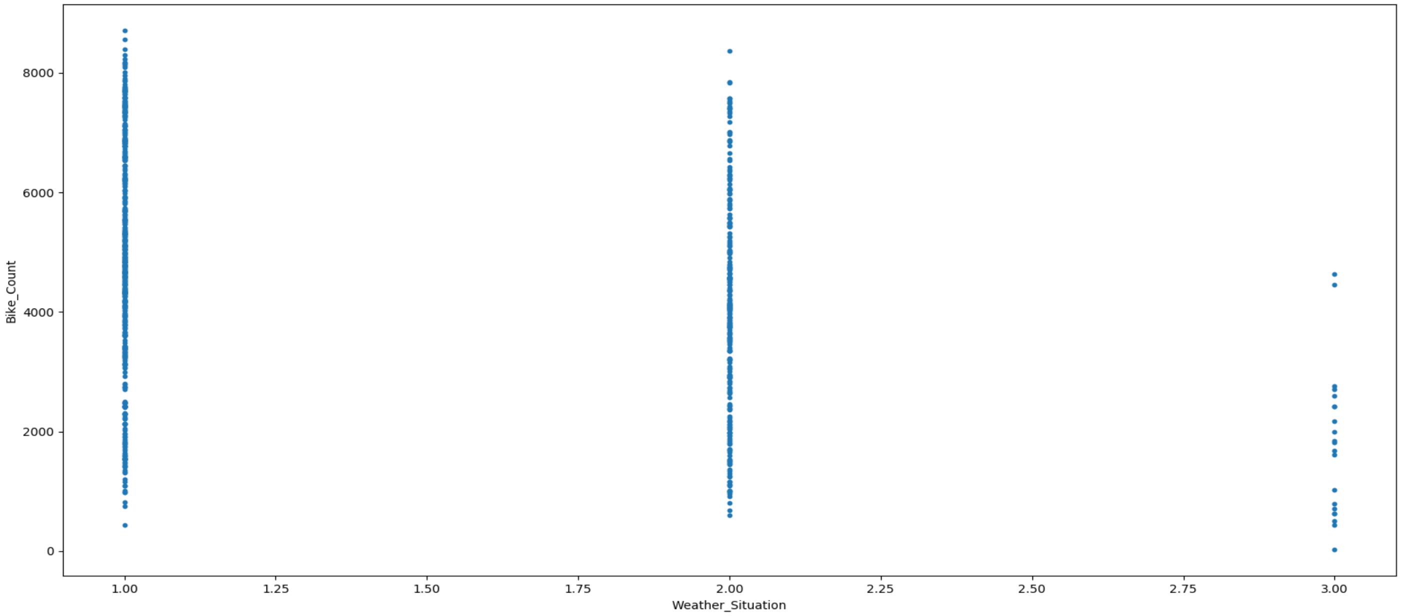
We could conclude that the bike rentals were more for the second year i.e. 2012 which could mean that people were using these services more in 2012 which might be because of some advertisements or offers provided to the customers.

ii. holiday Vs cnt



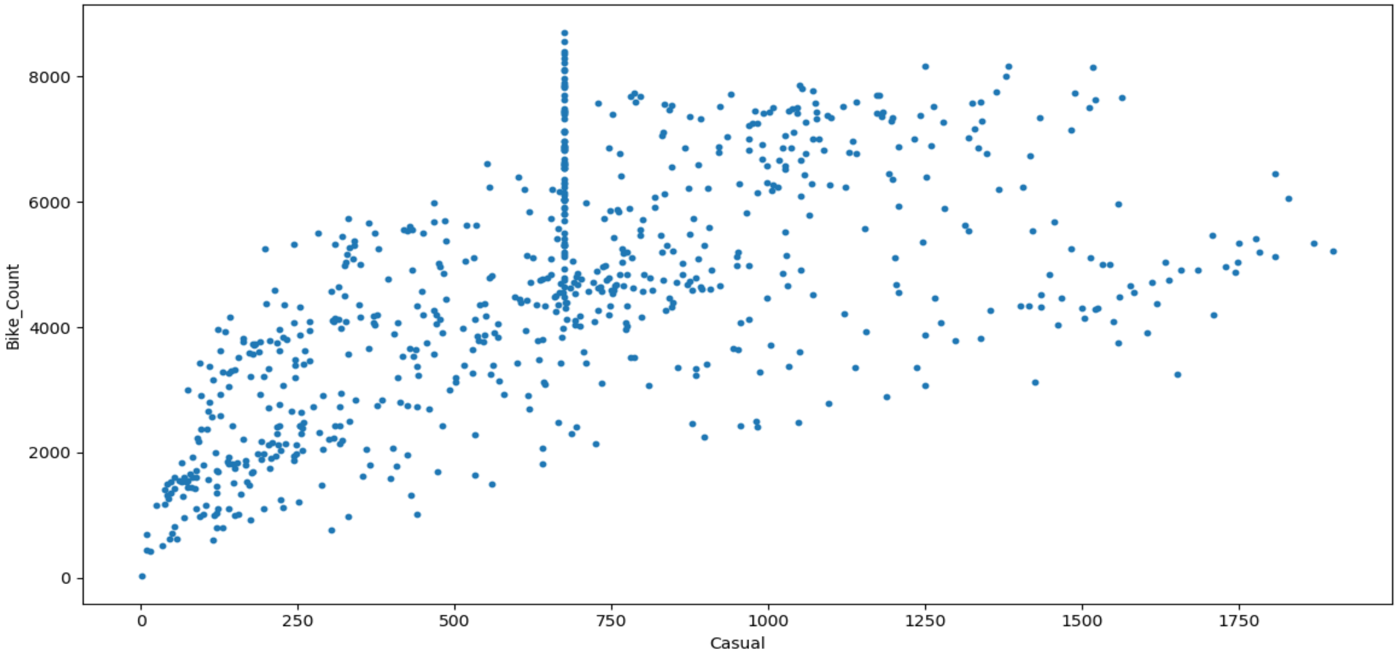
We could say that bike rentals were more when it was not a holiday. Maybe customers who rent the bikes, use it for going to office.

iii. weathersit Vs cnt



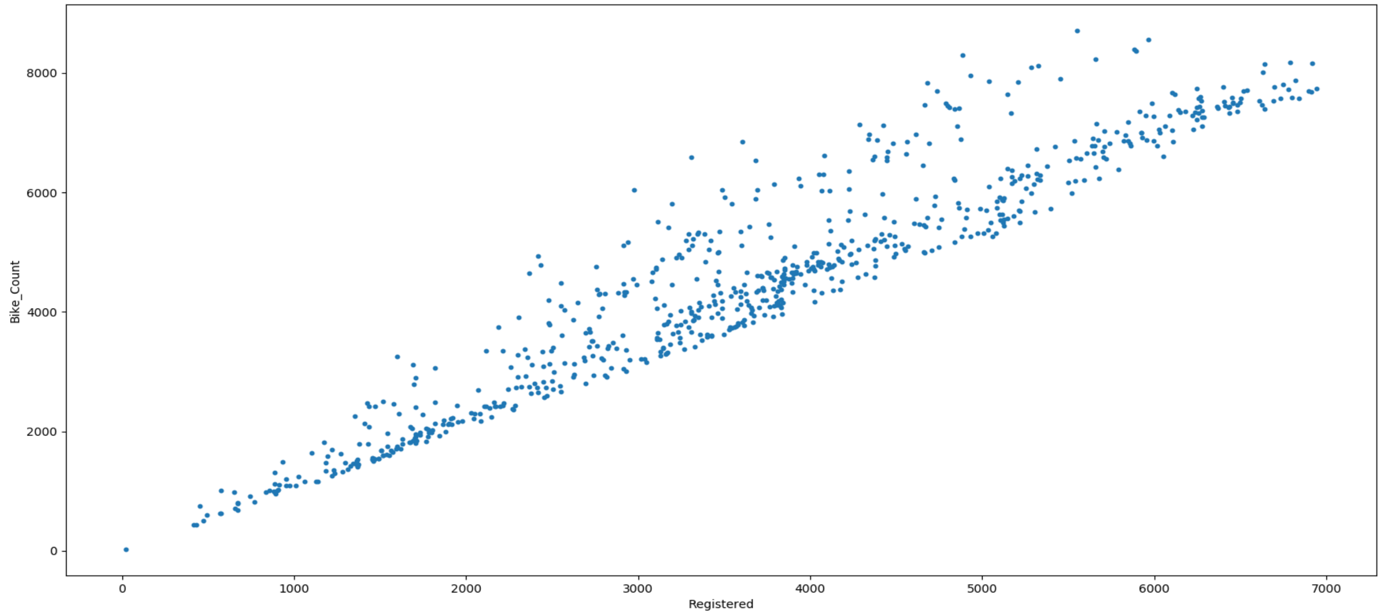
We could conclude that rentals were mode when the weather situation was of the category 1 or 2. Category 3 has very few rentals i.e. people don’t use the rentals when there is light snow or light rain or thunderstorms or scattered clouds are there.

iv. casual Vs cnt



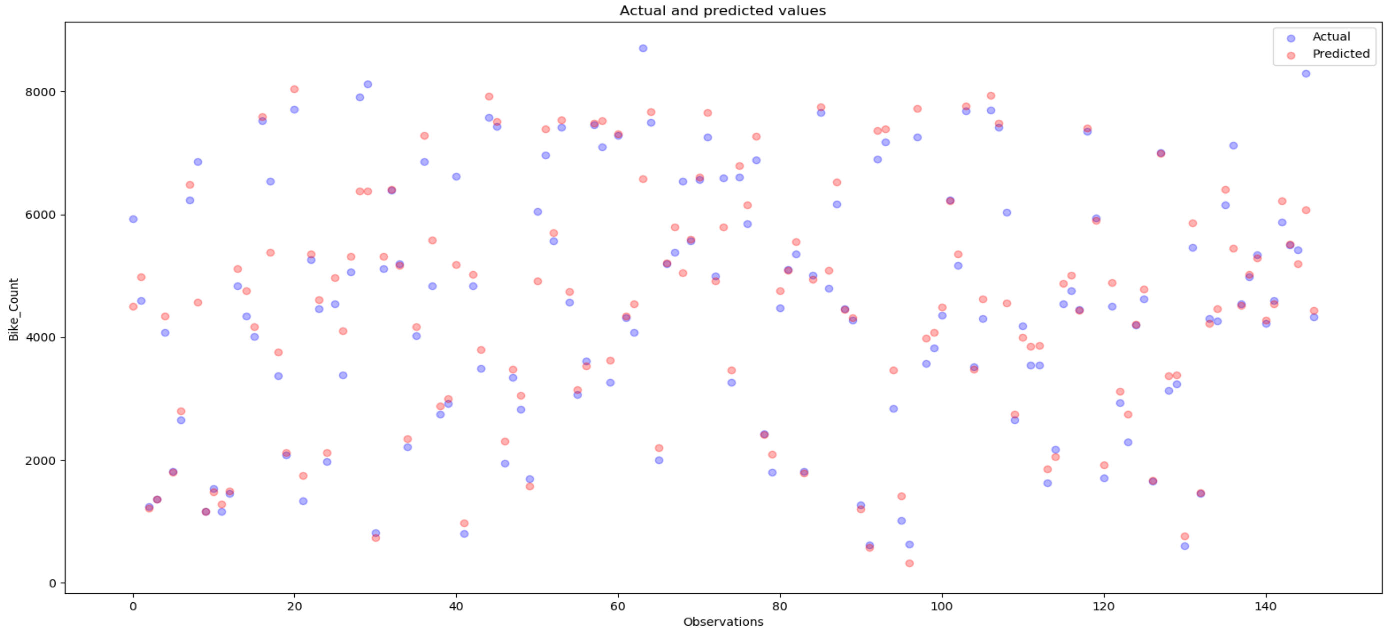
We could see that over time the bookings from casual accounts were very few.

v. registered Vs cnt

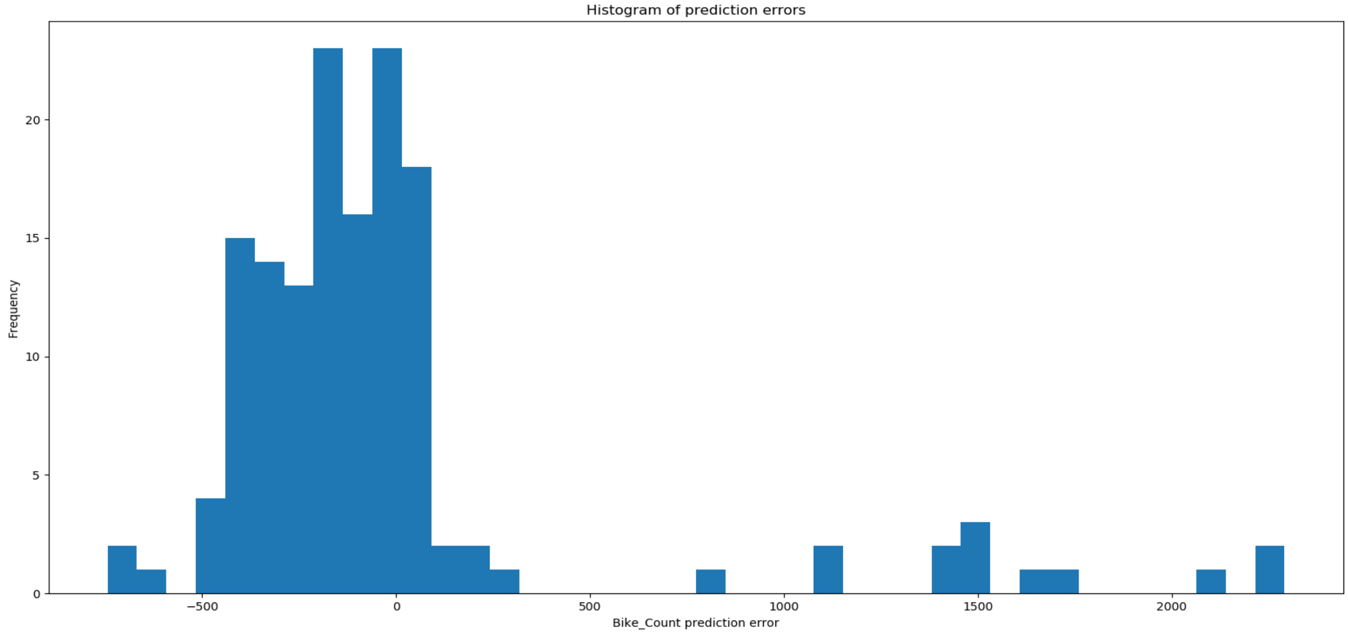


We could conclude that registered users and bike counts are linearly increasing. Maybe because of some offers given to registered users.

vi. Prediction and Actual values of Linear Model



vii. Histogram of residuals (Linear Regression)



<u>END OF REPORT</u>