BIKE SHARING DEMAND PREDICTION

[CSP 571: Data Preparation and Analysis Project]

ABSTRACT

Capital Bikeshare program is a service of providing rental bikes for the users which is the most popular program going around the world. We sought to create a predictive model for the demand of rental bikes for Capital Bikeshare program in Washington D.C. The analysis was based on three different regression models such as Linear, Random Forest, and Gradient Boosting. To evaluate testing results, the Root Mean Squared Logarithmic Error (RMSLE) was identified. Before building the prediction model more insights from data extracted in Exploratory Data Analysis stage. The best model observed is Gradient Boosting Machine Model/Gradient Boosting Regression Tree (GBRT) with a RMSLE value equal to 0.29 for testing dataset and 0.241 for training dataset.

Submitted By: -

Pritam Gajbhiye
[A20452320]
Chirag Bhansali
[A20436467]

Under the Guidance: -

Prof. Jawahar Panchal

1.1 Overview

Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world.

The data generated by these systems makes them attractive for researchers because the duration of travel, departure location, arrival location, and time elapsed is explicitly recorded. Bike sharing systems therefore function as a sensor network, which can be used for studying mobility in a city. In this project, we are trying to combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C.

1.2 Objective

Our objective is to combine the historical usage patterns with the weather data to predict the future bike rental demand in the Capital Bikeshare program in Washington, D.C. Through this process we are aiming to create a predictive model which will help to optimize, deciding future strategies, supply needed, etc. for the Capital Bikeshare program.

1.3 Specific Questions/Research Goals

We seek to answer these specific questions in this research:

- ❑ How different features such as time of day, weather, holidays and seasons impact rental bicycle demand?
- ❑ Which features are the most important for predicting the bicycle rental demands?
- ❑ How accurately can we predict the future demand?

2 Hypothesis Generation

Before exploring the data to understand the relationship between the variables, we focus on basic hypothesis generation based on the business problem, domain knowledge, and some hand experience of divvy bike sharing program at Chicago.

Here are some of the hypothesis which we thought could influence the demand of bikes:

- **Hourly trend**: There must be high demand during office timings. Early morning and late evening can have different trend (cyclist) and low demand during 10:00 pm to 4:00 am.
- **Daily Trend:** Registered users demand more bike on weekdays as compared to weekend or holiday.

- **Rain:** The demand of bikes will be lower on a rainy day as compared to a sunny day. Similarly, higher humidity will cause to lower the demand and vice versa.
- **Temperature:** In India, temperature has negative correlation with bike demand. But, after looking at Washington's temperature graph, I presume it may have positive correlation.
- **Pollution:** If the pollution level in a city starts soaring, people may start using Bike (it may be influenced by government / company policies or increased awareness).
- **Time:** Total demand should have higher contribution of registered user as compared to casual because registered user base would increase over time.
- **Traffic:** It can be positively correlated with Bike demand. Higher traffic may force people to use bike as compared to other road transport medium like car, taxi etc.

3 Data preparation and Cleaning

The dataset is taken from Kaggle dataset collection. The dataset is a .csv file with hourly rental data for two years (i.e. 2011 and 2012) from Capital Bike Sharing program, Washington D.C. The dataset contains already separated files for training and testing. The dependent variables are taken out from the testing file. We first combine the both the two files to understand the independent variables on more observations.

Now the dataset contains 17379 observations and 12 attributes.

Independent Variables

datetime: Date and hour in "mm/dd/yyyy hh:mm" format

season: Four categories-> 1 = spring, 2 = summer, 3 = fall, 4 = winter

holiday: whether the day is a holiday or not (1/0)

workingday: whether the day is neither a weekend nor holiday (1/0)

weather: Four Categories of weather

1-> Clear, Few clouds, Partly cloudy, Partly cloudy

2-> Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

3-> Light Snow and Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered

4-> Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

temp: hourly temperature in Celsius

atemp: "feels like" temperature in Celsius

humidity: relative humidity

windspeed: wind speed

Dependent Variables

registered: number of registered users

casual: number of non-registered users

count: number of total rentals (registered + casual)

3.1 Data type Identification

```
'data.frame':  17379 obs. of  12 variables:
$ datetime  : Factor w/ 17379 levels "2011-01-01 00:00:00",..: 1 2 3 4 5 6 7 8 9 10 ...
$ season    : int  1 1 1 1 1 1 1 1 1 1 ...
$ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
$ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
$ weather   : int  1 1 1 1 1 2 1 1 1 1 ...
$ temp      : num  9.84 9.02 9.02 9.84 9.84 ...
$ atemp     : num  14.4 13.6 13.6 14.4 14.4 ...
$ humidity  : int  81 80 80 75 75 75 80 86 75 76 ...
$ windspeed : num  0 0 0 0 0 ...
$ casual    : num  3 8 5 3 0 0 2 1 1 8 ...
$ registered: num  13 32 27 10 1 1 0 2 7 6 ...
$ count     : num  16 40 32 13 1 1 2 3 8 14 ...
```

Fig 1. Data types of each variables

Sample Dataset looks like:

1	datetime	season	holiday	workingda	weather	temp	atemp	humidity	windspeed	casual	registered	count
2	2011/1/1 0:00	1	0	0	1	9.84	14.395	81	0	3	13	16
3	2011/1/1 1:00	1	0	0	1	9.02	13.635	80	0	8	32	40
4	2011/1/1 2:00	1	0	0	1	9.02	13.635	80	0	5	27	32
5	2011/1/1 3:00	1	0	0	1	9.84	14.395	75	0	3	10	13
6	2011/1/1 4:00	1	0	0	1	9.84	14.395	75	0	0	1	1
7	2011/1/1 5:00	1	0	0	2	9.84	12.88	75	6.0032	0	1	1
8	2011/1/1 6:00	1	0	0	1	9.02	13.635	80	0	2	0	2
9	2011/1/1 7:00	1	0	0	1	8.2	12.88	86	0	1	2	3
10	2011/1/1 8:00	1	0	0	1	9.84	14.395	75	0	1	7	8
11	2011/1/1 9:00	1	0	0	1	13.12	17.425	76	0	8	6	14
12	2011/1/1 10:00	1	0	0	1	15.58	19.695	76	16.9979	12	24	36
13	2011/1/1 11:00	1	0	0	1	14.76	16.665	81	19.0012	26	30	56
14	2011/1/1 12:00	1	0	0	1	17.22	21.21	77	19.0012	29	55	84
15	2011/1/1 13:00	1	0	0	2	18.86	22.725	72	19.9995	47	47	94
16	2011/1/1 14:00	1	0	0	2	18.86	22.725	72	19.0012	35	71	106
17	2011/1/1 15:00	1	0	0	2	18.04	21.97	77	19.9995	40	70	110
18	2011/1/1 16:00	1	0	0	2	17.22	21.21	82	19.9995	41	52	93
19	2011/1/1 17:00	1	0	0	2	18.04	21.97	82	19.0012	15	52	67
20	2011/1/1 18:00	1	0	0	3	17.22	21.21	88	16.9979	9	26	35

Fig 2. Sample Dataset

3.2 Feature Engineering

From the above result we can see the variables “season”, “workingday”, “weather”, and “holiday” having integer data type, so we need to change them to categorical data type as “season” is to categorize as spring, summer, fall, and winter, “weather” as good, normal, bad, and worse, “workingday” and “holiday” as either yes or no.

Also, we create new variables “date”, “hour”, “weekday”, “year”, and “month” by extracting the required information for each variable from “datetime” variable and drop that variable.

datetime
2011-01-01 00:00:00
2011-01-01 01:00:00
2011-01-01 02:00:00
2011-01-01 03:00:00
2011-01-01 04:00:00
2011-01-01 05:00:00
2011-01-01 06:00:00
2011-01-01 07:00:00
2011-01-01 08:00:00
2011-01-01 09:00:00

date	year	month	hour	wkday
2011-01-01	2011	1	0	7
2011-01-01	2011	1	1	7
2011-01-01	2011	1	2	7
2011-01-01	2011	1	3	7
2011-01-01	2011	1	4	7
2011-01-01	2011	1	5	7
2011-01-01	2011	1	6	7
2011-01-01	2011	1	7	7
2011-01-01	2011	1	8	7
2011-01-01	2011	1	9	7

Fig 3. Extracting information from datetime variable

4 Exploratory Data Analysis

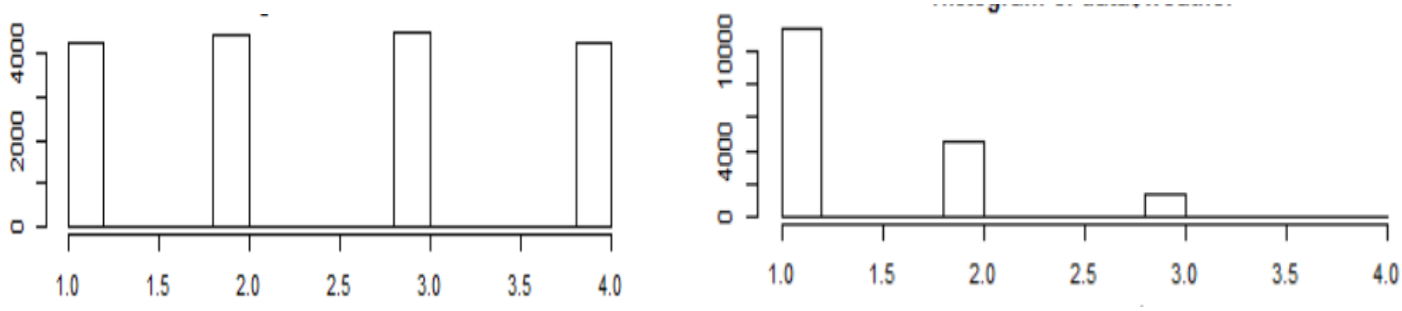
Here we work on missing value analysis, understanding the distribution of numerical variables, generating various graph in order to extract and explore insights hidden in our data. Also, we addressed some of the hypothesis we assumed above before exploring the data.

4.1 Missing value analysis

We check for any missing values in the dataset by using “*is.na ()”* command and observe that there are neither missing values nor NA values in the dataset.

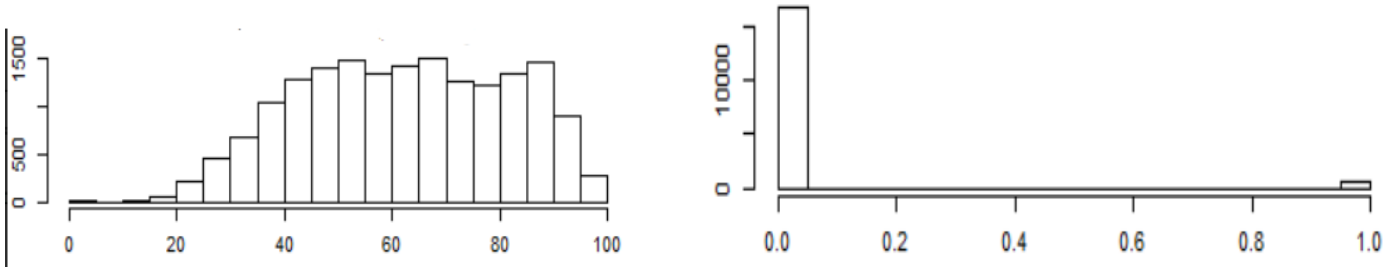
4.2 Understanding distribution of Numerical variables

The numerical variables in our dataset are *“season”, “weather”, “humidity”, “holiday”,” workingday”, “temp”, “atemp”, and “windspeed”*. We plot the histogram plot for each variable and come with their distribution also, some inferences from the plots.



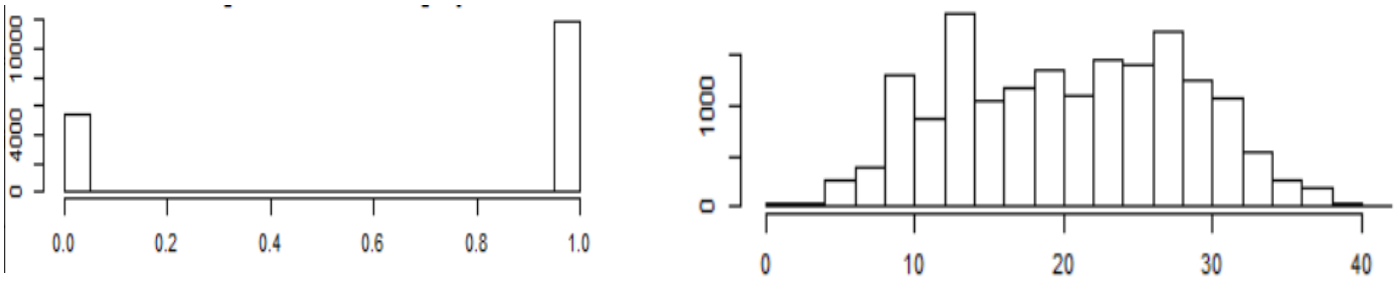
(a) (b)

Fig 4. Histogram plot for (a) season, (b) weather



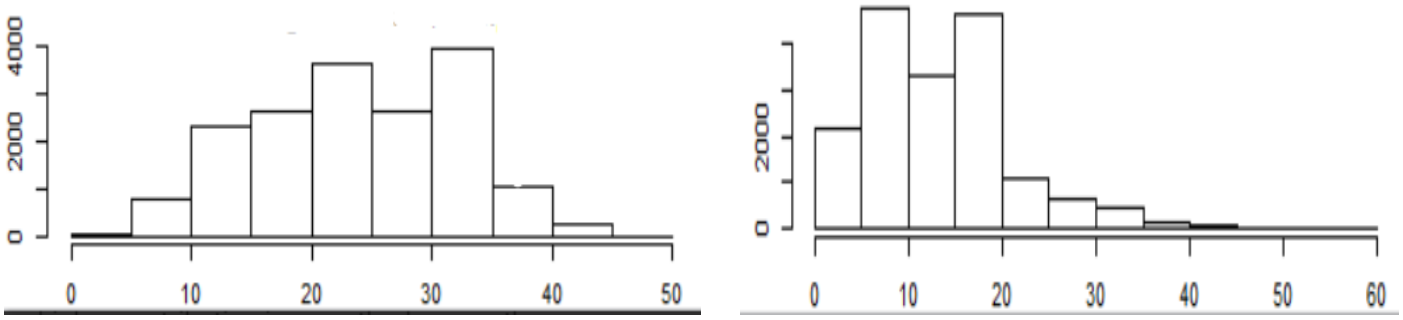
(a) (b)

Fig 5. Histogram plot for (a) humidity, (b) holiday



(a) (b)

Fig 6. Histogram plot for (a) workingday, (b) temp



(a) (b)

Fig 7. Histogram plot for (a) atemp, (b) windspeed

From above histogram plots few inferences can be drawn such as season variable has four categories of almost equal distribution, in weather variable weather 1 (i.e. good) has higher contribution, variables workingday and holiday shows the similar inference as expected and the remaining variables humidity, temp, and atemp shows nearly normal distribution.

4.3 Outlier Analysis

For outlier analysis we plot the boxplot of each variable and observe the outlier's points which are above the fourth quartile.

While observing count variable boxplot which contains lots of outlier data point i.e. the distribution is skewed towards right side.

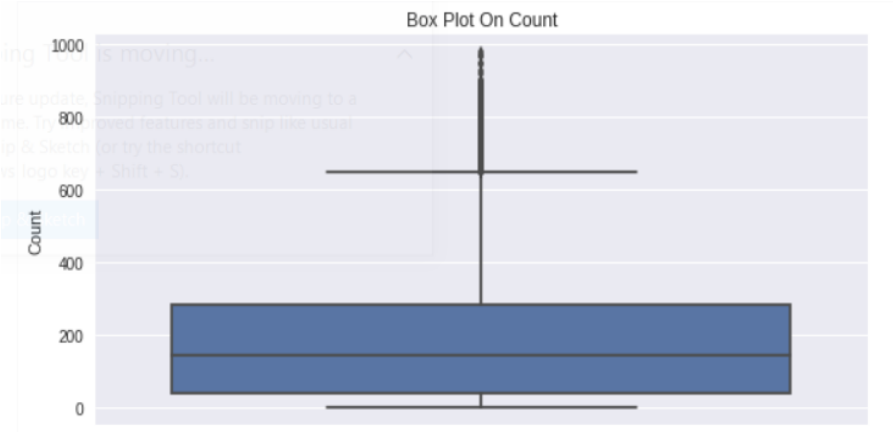


Fig 8. Boxplot of count variable

Looking at the season variable boxplot we can clearly observe that spring season has got relatively lower count.

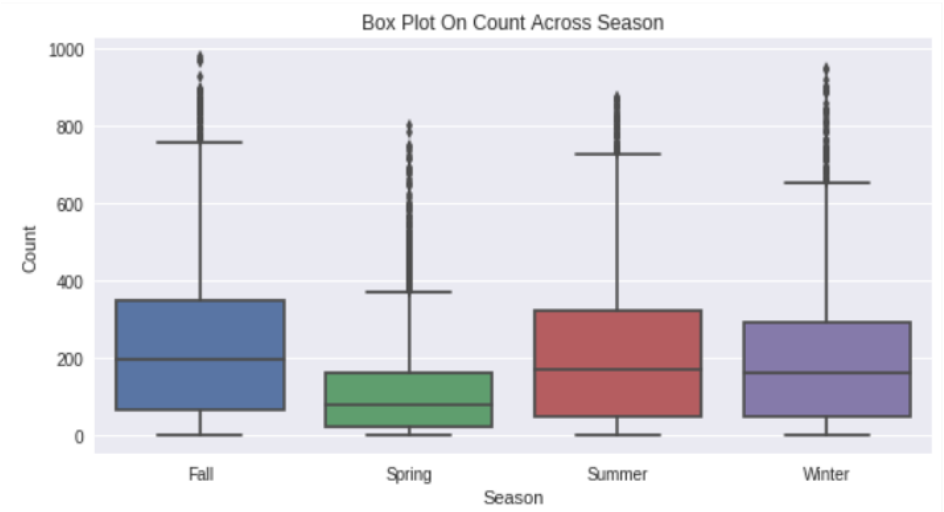


Fig 9. Boxplot of season variable

We can observe an interesting fact from the hour variable that the median values are relatively higher at 7am-8am and 5pm-6pm. From this we can inferred that the time corresponds to regular school and office users taking the bicycle for their travel from their home to office/school and back.

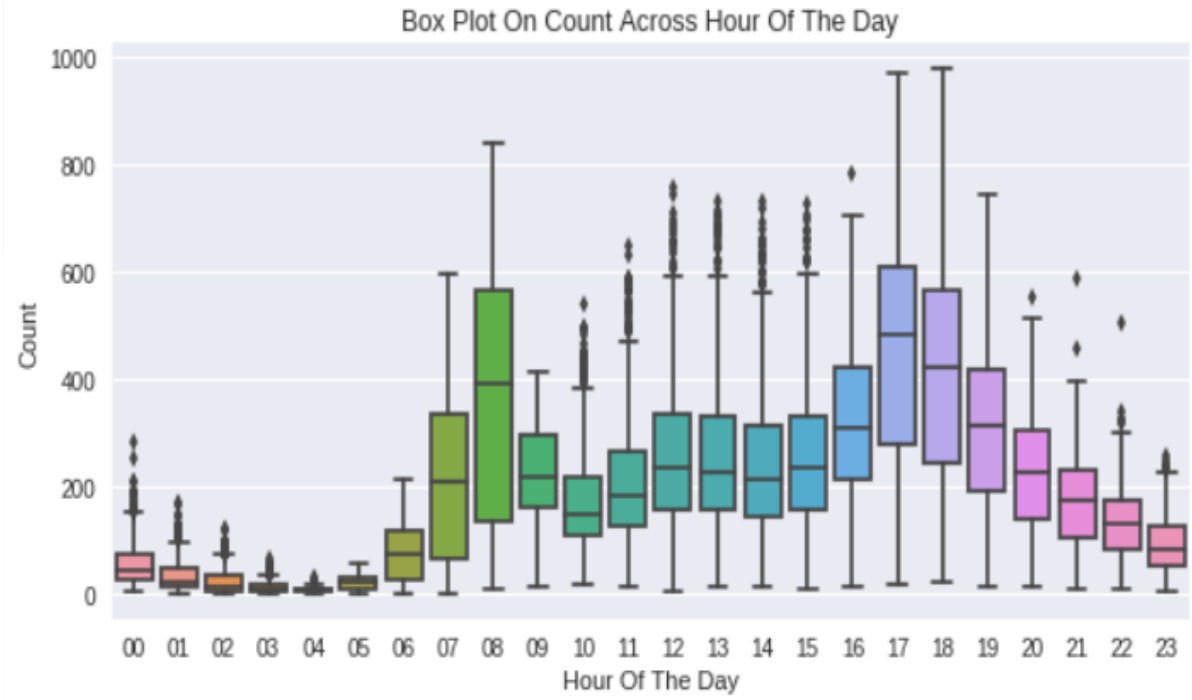


Fig 10. Boxplot of Hour Variable

When we plot the boxplot of working day variable, we can observe that most of the outlier's are from working day than non-working day.

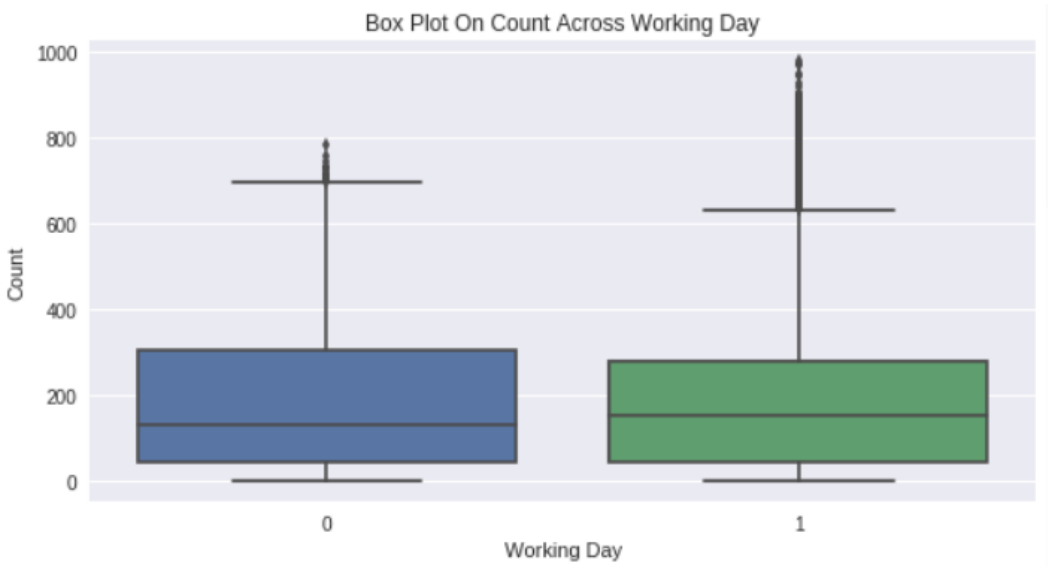
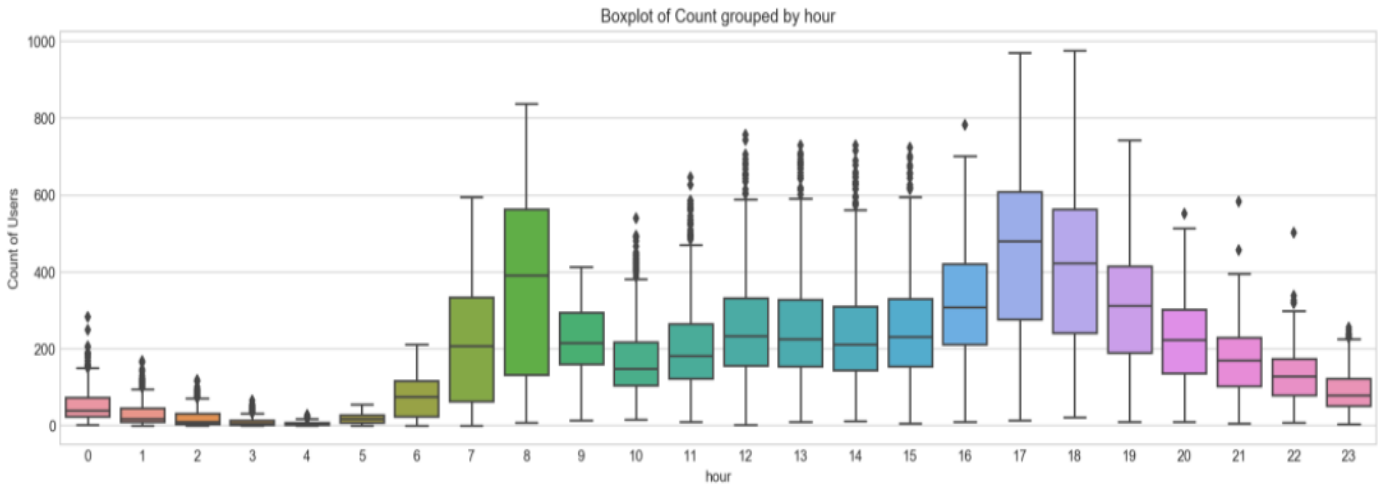


Fig 11. Boxplot of working day variable

4.4 Hypothesis Testing (Previously assumed hypothesis)

Now, we tested our previously assumed hypothesis which were based on our prior knowledge and past hands on experience. Here we check that whether the variables support our prior assumption or not, if not we gather the hidden insights based on the data. To achieve this, we plot various plots with respect to the count variable which is a combined count of registered and casual users.

4.4.1 Hourly Trend



From above plot we can inferred that the bike rental demand is high between 7-8 and 17-19 hours, this inference satisfies our prior assumed hypothesis that the demand is high during the office/school timings. To get more insights we plot the hourly trend for both types of users i.e.

registered and casual from which we can clearly infer that casual users mostly take bike in the afternoon hours which is a prime time for most of us to go outside and enjoy the weather and the most of the registered users might be those who had taken the subscription for travelling back and forth for their office/school from their home as we can see the major count of registered users is between 7-8 and 17-19 hours.

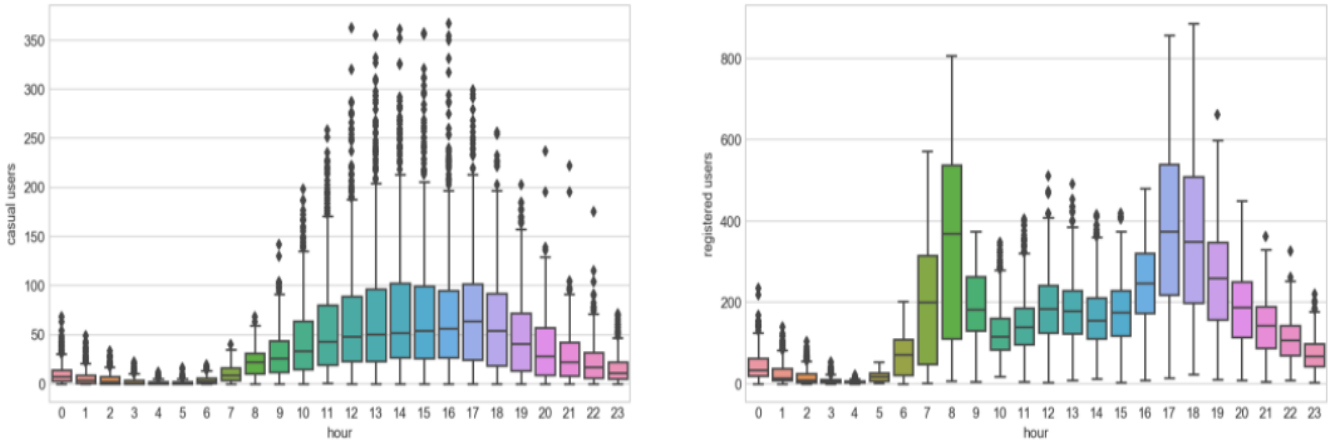


Fig 12. Boxplot for casual and registered users corresponding to hour

4.4.2 Daily Trend

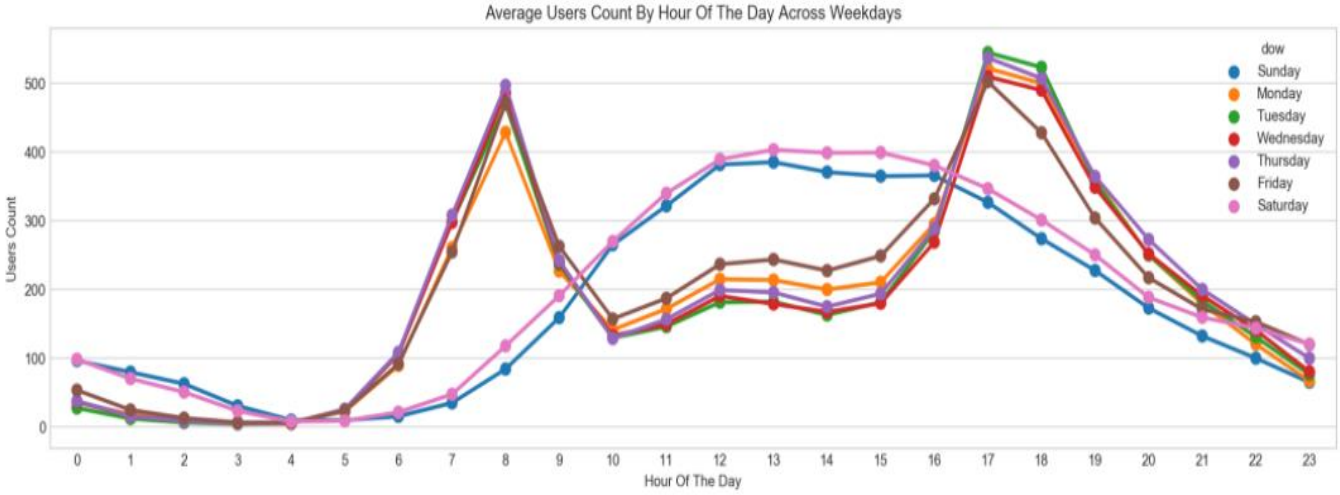


Fig 13. Daily trend of count

From the above plot clearly satisfies our prior assumed hypothesis regarding daily trend that user demands more on weekdays as compared to weekends. Also, we plot the daily trend for registered and casual users separately where we clearly see that the registered users count is less on weekends as compared to weekdays which strongly support our belief that the most of the registered users are from office/schooling background.

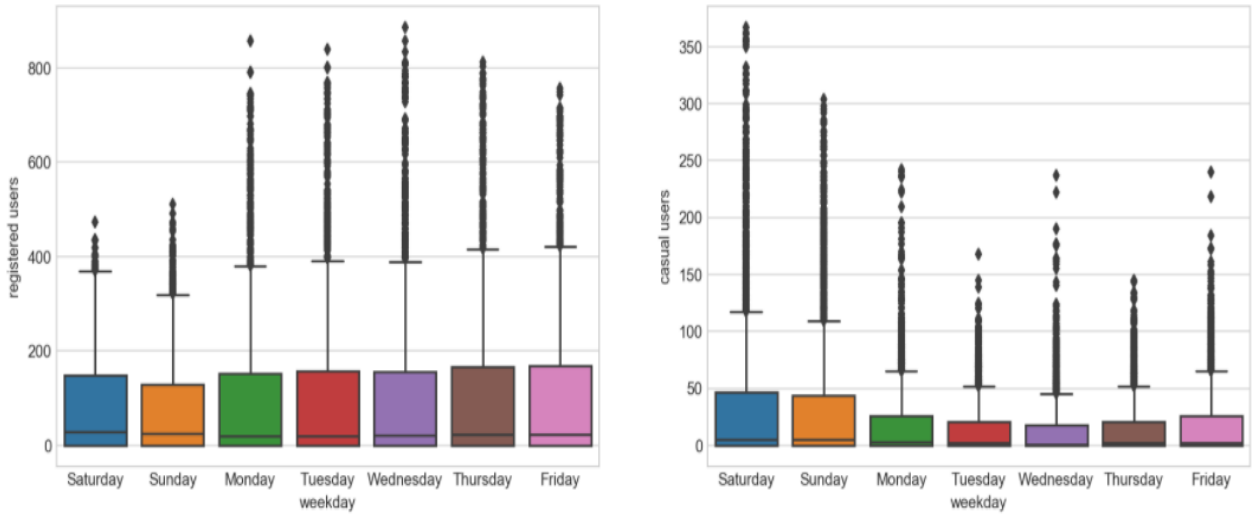


Fig 14. Daily trend boxplots for registered and casual users

4.4.3 Rain

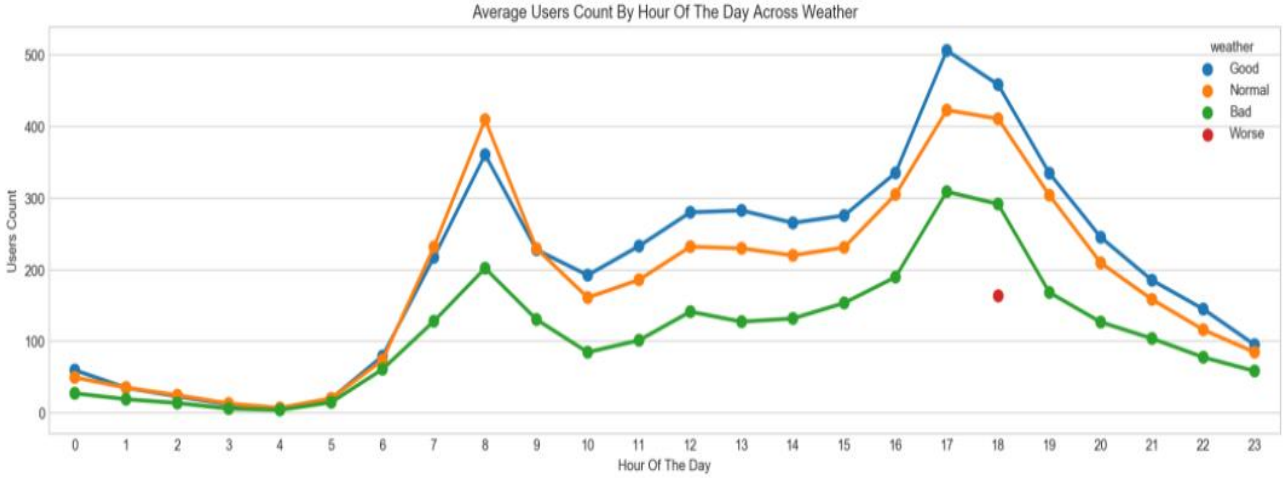
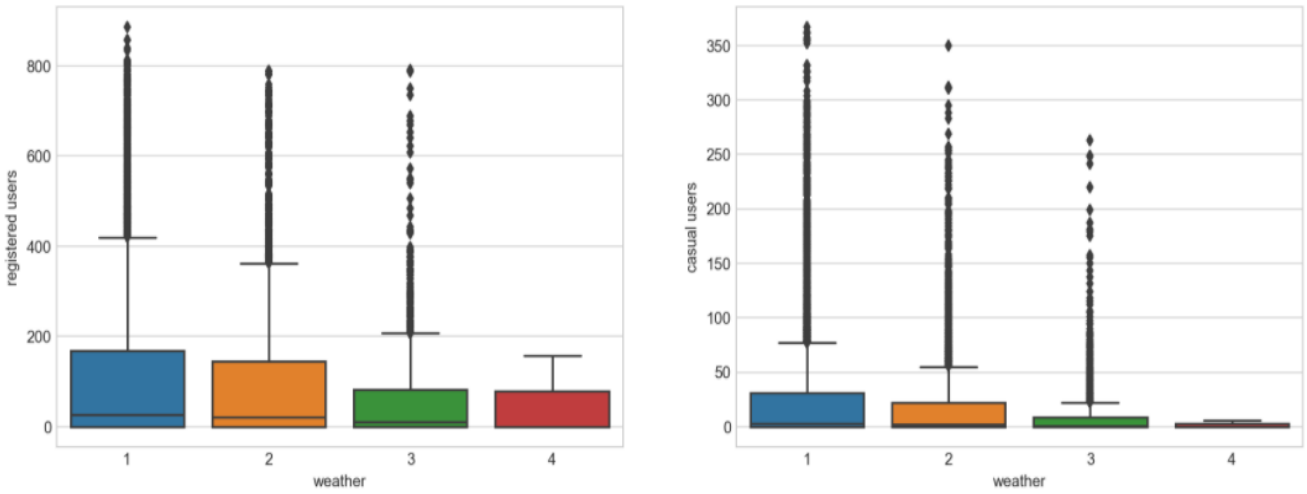


Fig 15. Weather plot

There are no specific variables for rain in our dataset, but we have taken the weather variable where 3 represent light rain and 4 represent heavy rain. By plotting the boxplots of weather variable separately for registered and casual users, we clearly infer that the counts for registered users is more as compared to casual users. In 4 (i.e. heavy rain) we can see that in casual users the counts nearly equal 5-10 which is totally negligible but as compared to registered users we can see some count near to 200 which might be the number of users who totally depends on the bike for their travelling



.

Fig 15. Boxplot of weather for registered and casual users

4.4.4 Temperature

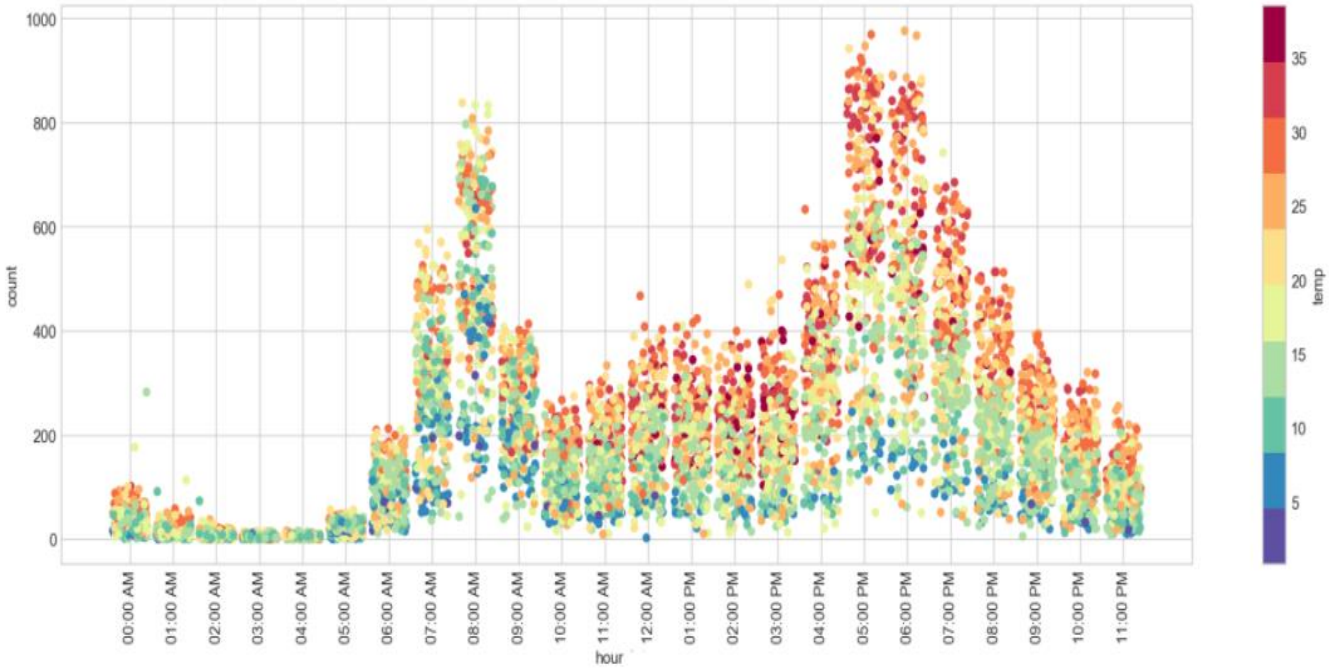


Fig 16. Counts with respect to temperature and hour

From the above plot we can inferred that most bikes are rented on warm mornings and evening. Also, we prior assumed that the temperature variable might show some positive correlation and we can see it is nearly true. This plot shows that the count of renting bikes increases with respect to warm weather, indirectly we can say that people use to ride bicycle in warm weather might be to enjoy the weather with friends and colleagues.

To observe the proper correlation of counts with respect to temperature, humidity, windspeed, casual and registered users we plot the correlation plot for these variables.

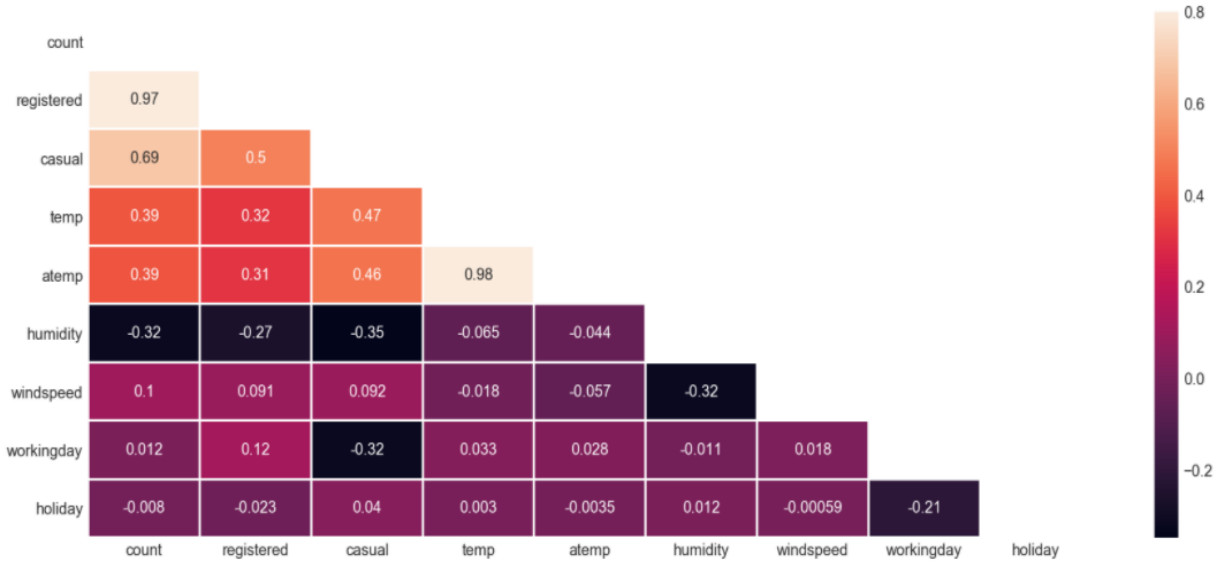
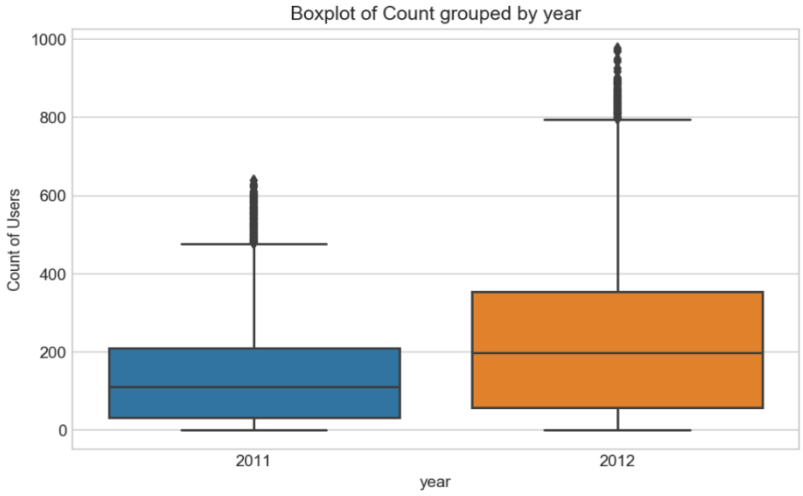


Fig 17. Correlation plot

From the above correlation plot we can infer that the temperature variable is positively correlated with the counts also, we can see that it is more correlated to casual users than registered users. From which we can say that people use to enjoy warm weather and take the rental bikes to enjoy the streets with friends and colleagues.

4.4.5 Time

We previously assumed that with time the counts will increase but the registered users might contribute more i.e. registered users count must increase with time. To be confident on our assumption we plot the boxplot for the two year (2011 and 2012) and we confirmed that with time the counts increased.



Also, there is one more categorical variable “season “so we plot the average users count with respect to hours and categorizing based on seasons. From the plot we clearly see that the average counts are higher fall, summer and winter as compared to spring. As spring moths are more windy and colder people don’t use to rent the bike which is a genuine reason to stay at home or use car or public transport for travelling.

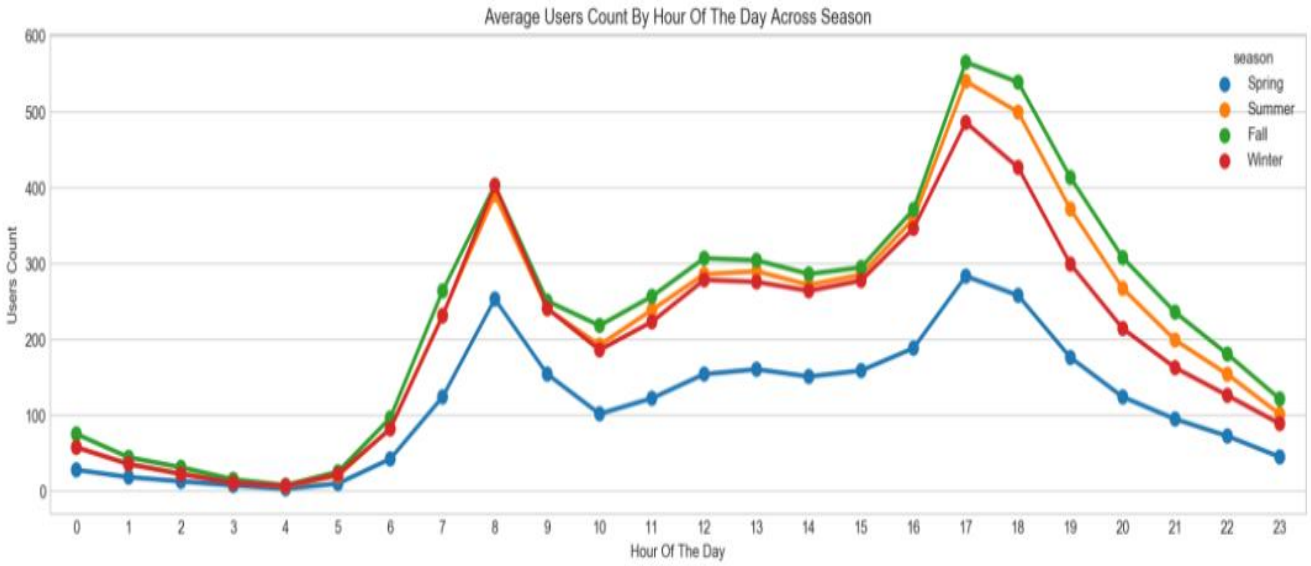


Fig 18. Season plot

5 Modeling (Model Training and Validation)

First, we begin with Linear Regression model statistical technique as it is always a good option to start with simple model rather than complex machine learning algorithms. After that we implement other statistical learning techniques such as Regularization Model (Ridge and Lasso), Random Forest Model and Gradient Boost Model where we observe that our evaluation criteria value gets improved as compared Linear Regression Model.

As the given dataset contains two separates .csv files i.e. train.csv and test.csv but in test file the dependent variables (count, registered, casual) are missing. Here we split the train file dataset into two separate train and test dataset by contributing 80% of the observations to train dataset and rest to test dataset. We will fit our models on train dataset and evaluate our model on test dataset and jot down the RMSLE value to compare with the other models RMSLE values.

5.1 Evaluation Criteria

Here we have used Root Mean Squared Logarithmic Error (RMSLE) for evaluating the model. Usually we use Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) for regression evaluation but RMSLE is helpful to penalize an under-predicted estimate greater than an over-predicted estimate.

RMLSE is given as: -

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$$

Where, n = total number of samples

p_i = predicted value

a_i = actual value

5.2 Linear Regression Model

Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x)

We take the split train dataset to fit our linear regression model, where we describe count as a dependent variable in our formula and rest variables as independent variable. We then calculate the RMSLE value for both train and test dataset, we found out that the train RMSLE value nearly equal to 1.027. We predict the count variable values for the test dataset and calculate the RMSLE value which is nearly equal to 1.004.

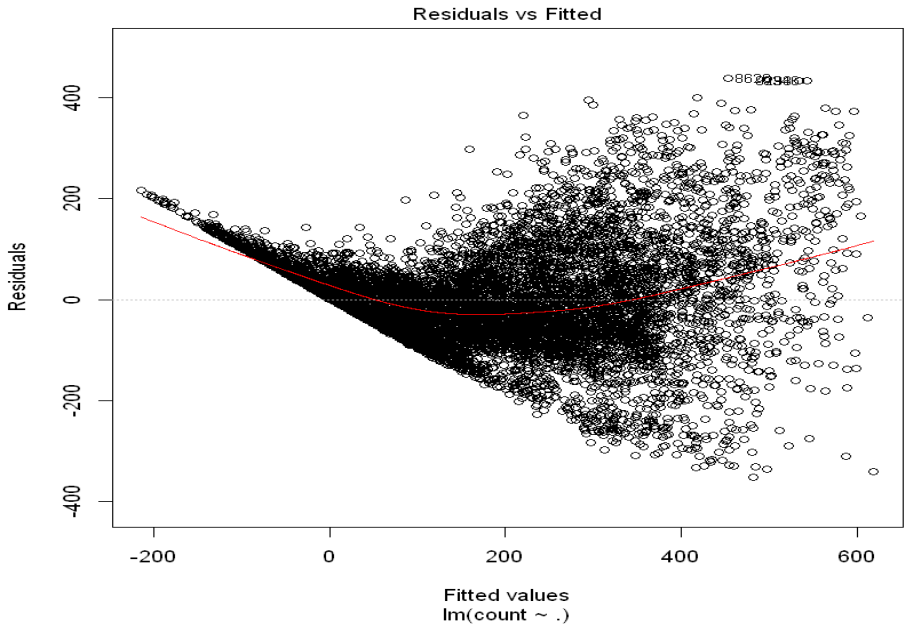


Fig 19. Residual vs Fitted plot for Linear Regression

To properly visualize how is our model performed we plot the count variable of test dataset and plot the predicted values of count variable, where we compared their distribution and find out that the Linear Regression model doesn't fit properly on our test data.

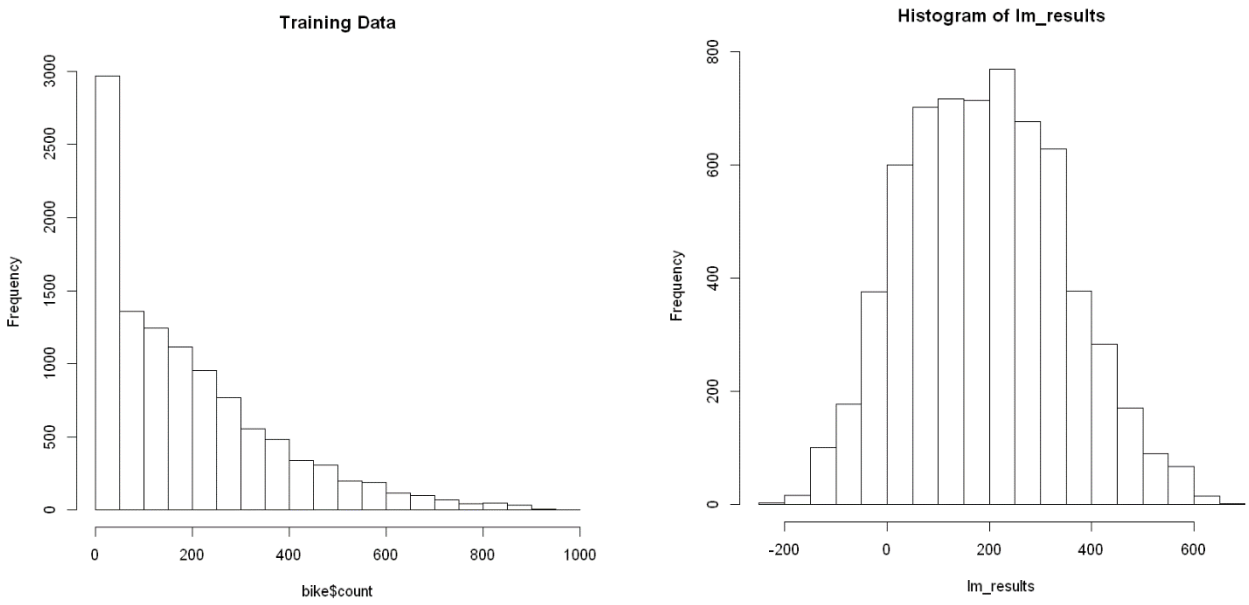


Fig 20. Count variable plot (a) training data (b) predicted counts by Linear regression

5.3 Random Forest Model

From the above result of Linear Regression model, we decide to improve our model prediction results. So, we choose to fit/execute Random Forest modeling on our data and check the RMSLE value corresponding to it, so that at the end we select best modeling approach for such kind dataset with more accurate results i.e. low RMSLE value.

Ensemble models are nothing but an art of combining a diverse set of individual weak learners (models) together to improve the stability and predictive capacity of the model. Ensemble Models improves the performance of the model by

- Averaging out biases
- Reducing the variance
- Avoiding overfitting

Before moving directly towards fitting the random forest model on data, we fit two different models on casual and registered users. To fit the random forest model, we need to decide the values for two parameters *"ntree"* and *"mtry".*

The variable *"ntree"* sets how large our Random Forest is going to be. Or in other words, how many trees should be contained in our ensemble. We use 500 here to strike some balance between fitness and computation time.

The variable *"mtry"* controls the number of variables randomly sampled at each split. The random value here is one third of all the variables given to Random Forest. In our example we use the value equals to 5.

After, fitting both models we sum the results so that we can compare the results and check the RMSLE value for the Random Forest regression model.

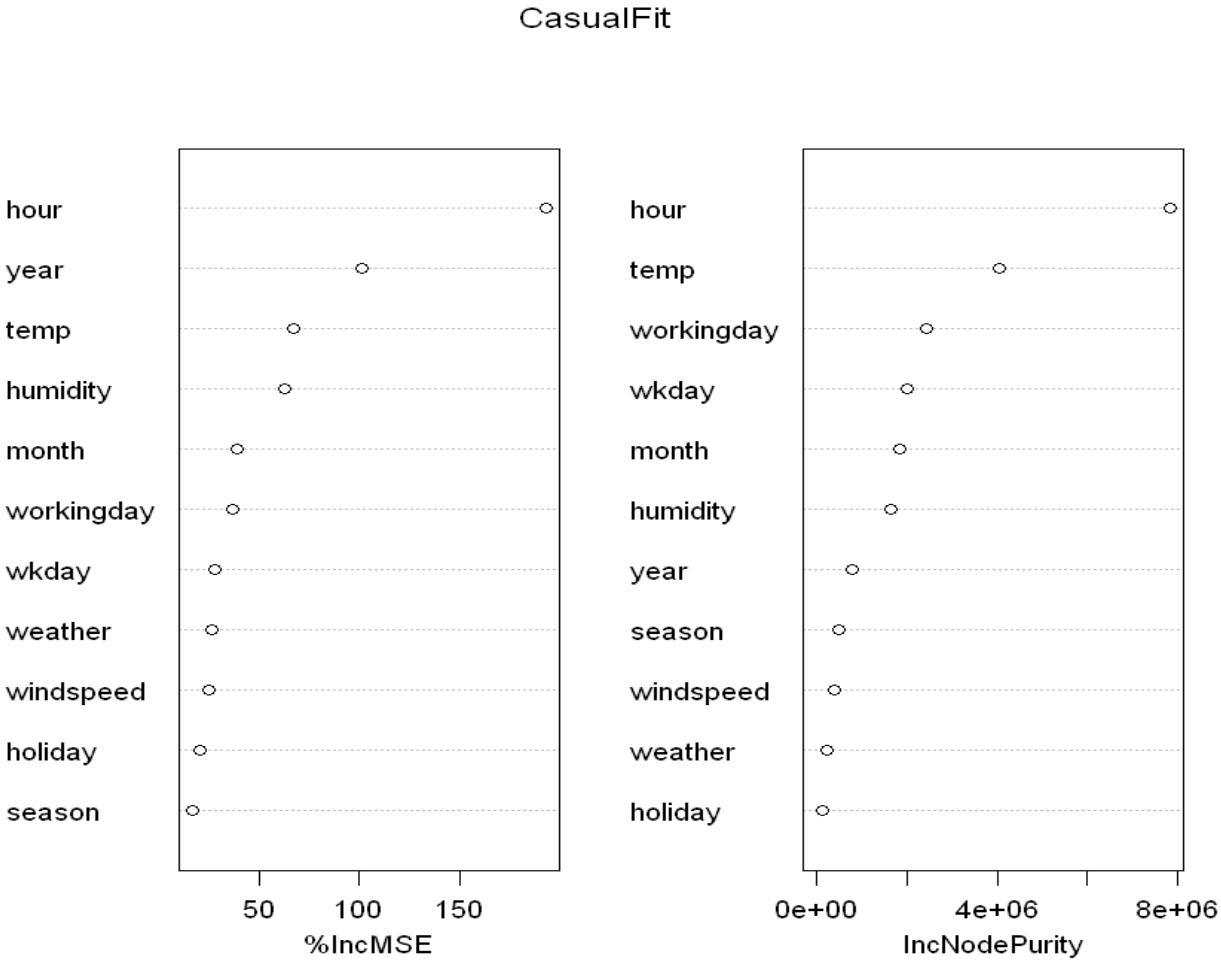


Fig 21. Variable importance plot for Casual users fit

From the above plot we can infer that variables season, holiday, windspeed and weather are not must significant here.

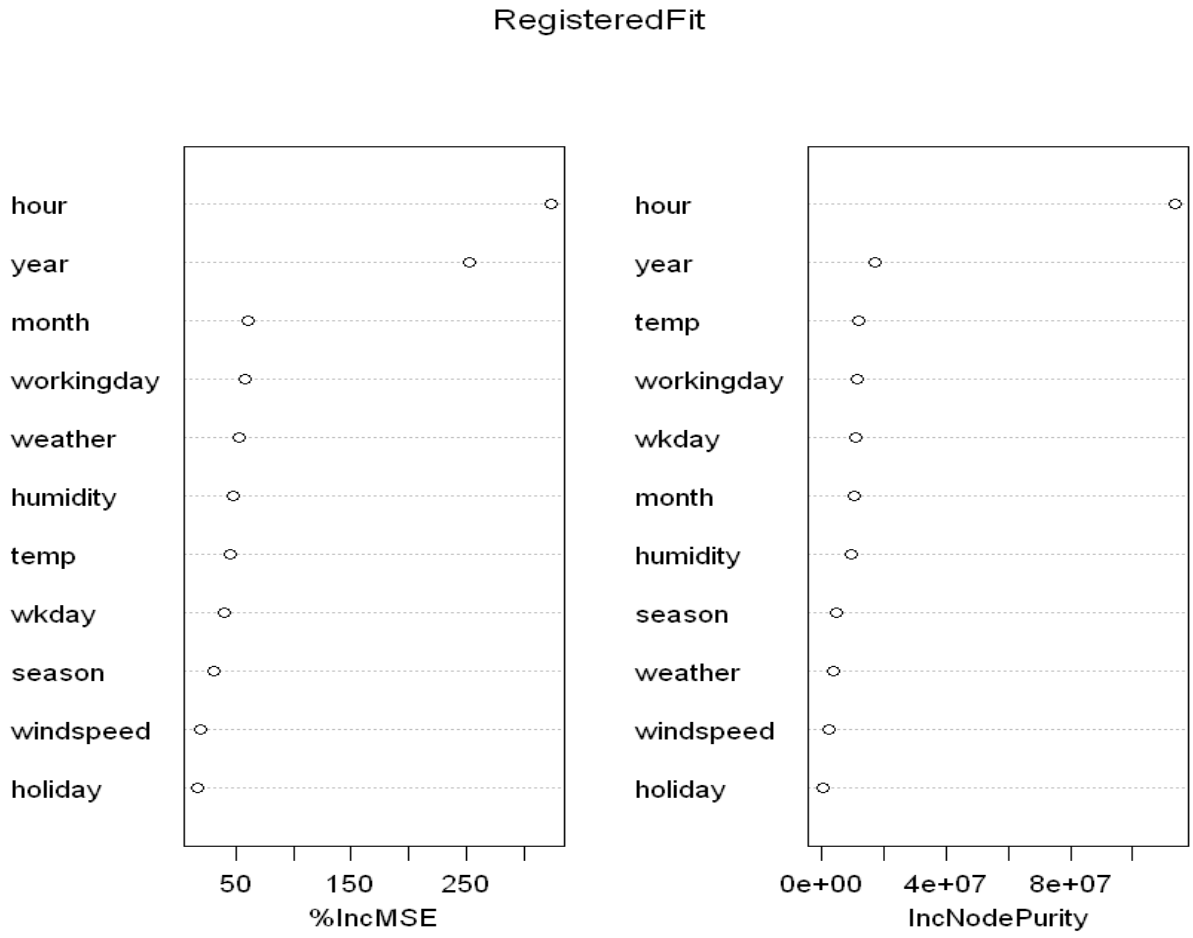


Fig 22. Variable importance plot for Registered users fit

From the above plot we can infer that the variables season, holiday, windspeed and weekday are not much significant here.

From the below plotted results, we can clearly see that the distribution of count variable from random forest regression results nearly matches the actual distribution. And we can conclude that the random forest modeling is better than the linear regression model for such dataset.

The training RMSLE for random forest model is 0.256 which is much better as compared to linear regression model training RMSLE value. The test RMSLE value for random forest model is 0.421 which is also much better than previous linear regression model testing RMSLE.

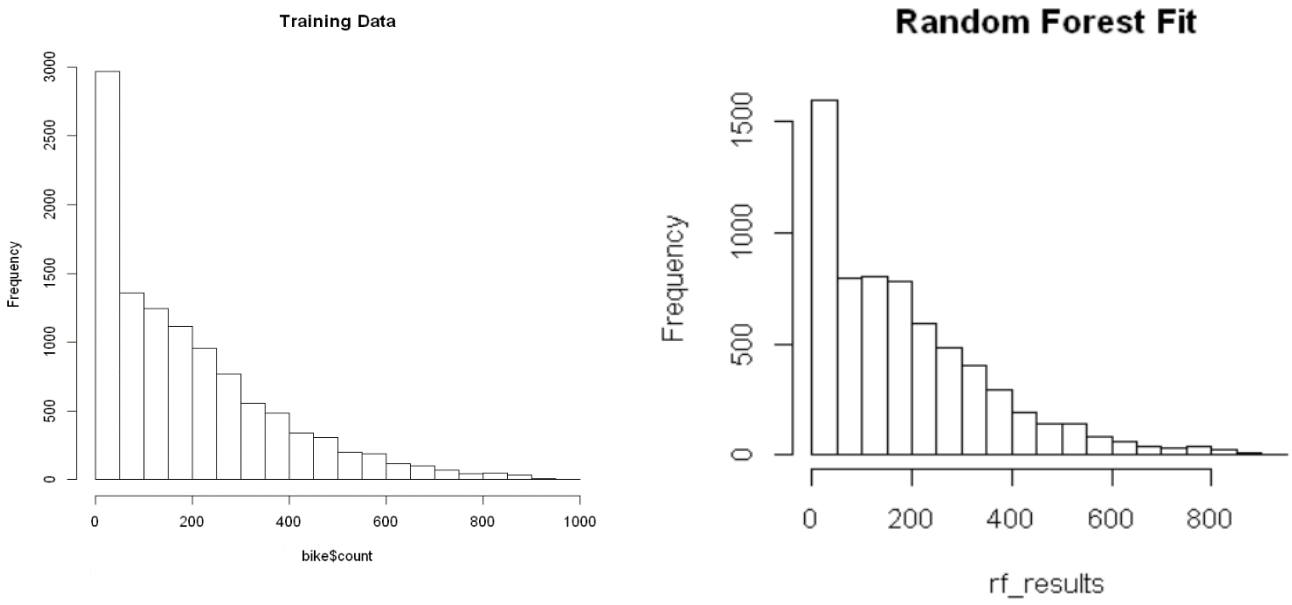


Fig 23. Count variable plot (a) training data (b) predicted counts by Random Forest model

As we inferred two main things from variable importance plots of casual and registered fit, we remove those variables from the data and fit the random forest again and check for any improvements we found out that both the training and testing RMSLE value get reduced.

The training RMSLE value is 0.214 and testing RMSLE value is 0.349 for the reduced dataset.

5.4 Gradient Boosting Machine (Gradient Boosted Regression Trees)

The Gradient Boosting method is an ensemble method using boosting technique, while the Random Forest method is using bagging technique. It is a forward stage-wise additive model uses greedy strategy.

There are some parameters that can be imposed on the construction of decision trees:

- Number of trees: generally adding more trees to the model can be very slow to overfit. We use a large number as “n.trees” = 4,000 to ensure there are enough trees.
- Tree depth: deeper trees are more complex. We use “interaction.depth” = 3.

The other parameters are left as default.

Here we approach the same way as we approached for random forest modeling, we model the gradient boosting model on those predictors or variables we found to be more important or significant in random forest modeling step.

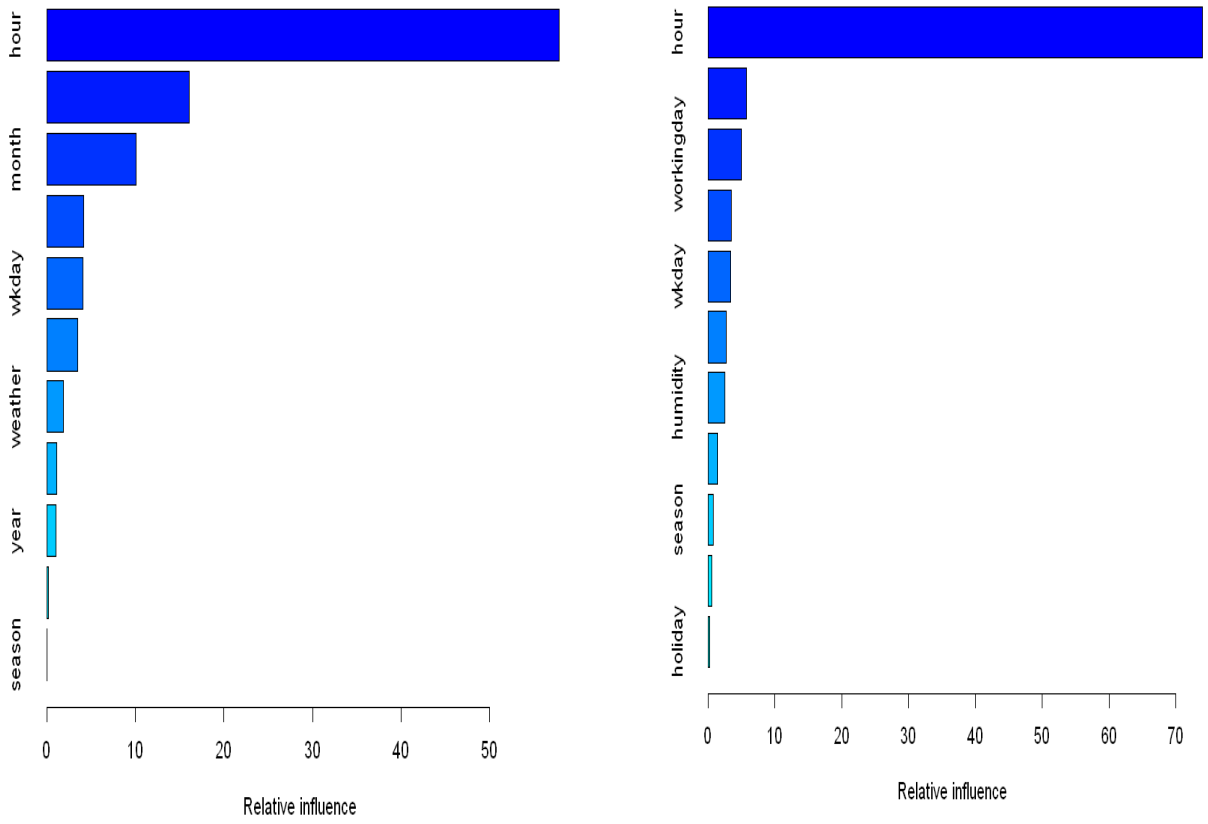


Fig 24. Variable importance for (a) casual (b) registered users

From the results of fitting the GBM the training RMSLE value for gradient boosting model is 0.241 which is nearly equal to the training RMSLE value of random forest model and much better than linear regression model. The testing RMSLE value for gradient boosting model is 0.29 which is little bit improved from random forest model. Also, we plot the distribution of GBM results on count and compared it with training count distribution where we confirm that the GBM performs well.

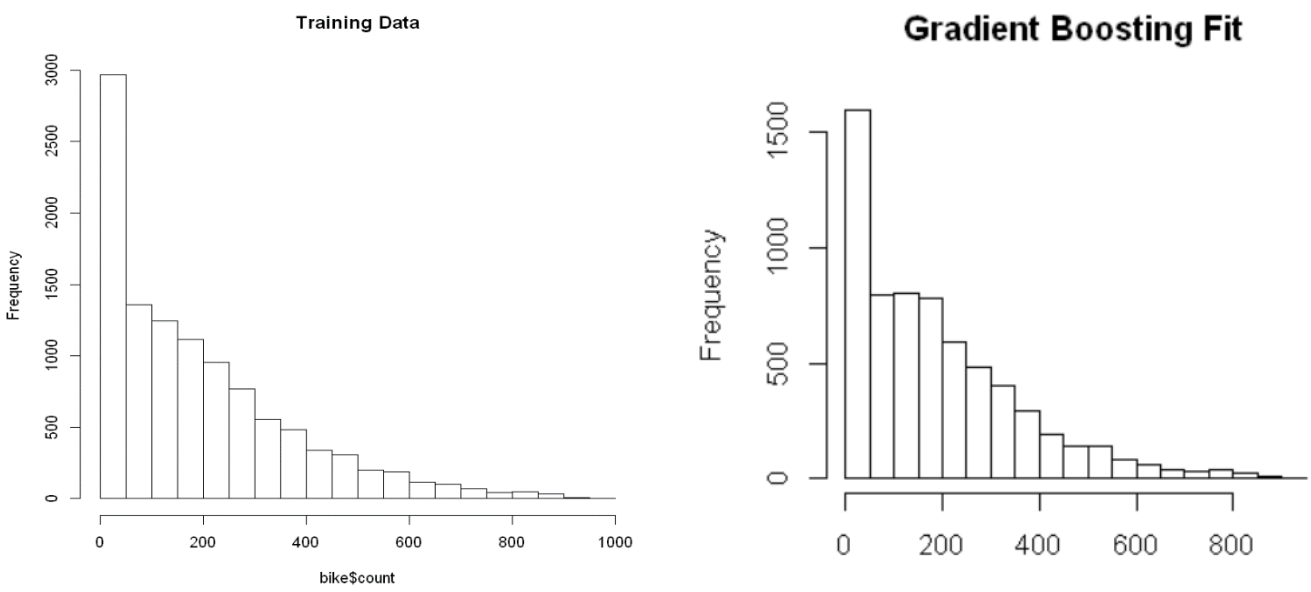


Fig 25. Count variable plot (a) training data (b) predicted counts by GBM model

6 Conclusion

After executing Linear Regression, Random Forest and Gradient Boosting Machine models we found out that gradient boosting model works better on our given bike share dataset of Capital Bike Sharing program, Washington D.C. The gradient boosting machine model gives RMSLE value of 0.29, using this model we can predict the future demand of rental bikes needed per hour in Washington D.C.

From exploratory data analysis stage we found out that the demand of rental bikes is high at office/school timings , most casual users rent bike in the afternoon, the count is higher in warm weather, the count of registered users is more in rainy season where as casual users is negligible showing that some of the registered users rely on subscription of the this bike rental program and the variable such as temperature is highly correlated with the count. Based on the prediction model Capital Bikeshare program can decide their future business strategies and implement in their program.

This model can be further improved if more data is available or we can add some census data and biodiversity data to it. So that we can really come up with some questions such as does the biodiversity (greenery/parks) affect the rental demand in that area? Which race mostly use such programs? What is the annual income of the users? Etc.