ABSTRACT :

In our project, we focus on predicting the number of bike rentals for a city bike-share system in Washington D.C., as part of a Kaggle competition. Through our project, we identified several important feature engineering ideas that helped us create more predictive features. We then trained and tuned several different models with our data, and experimented with ensembling methods as well. In the end, we found that decision-tree based models perform well on the bike-share data.

The programming language Python was used to build the model.

INTRODUCTION :

In recent years bike sharing schemes have become more commonplace in major cities all over the world. The service involves making bikes available for shared use, generally for a short term basis. Bike share system, the bike is taken from one location in the city and returned to another, generally the user’s destination. Bike-sharing has experienced huge growth in recent years .The huge rise in popularity in these schemes has been the motivation for choosing this topic.

OBJECTIVE :

Our main aim is to build a multi variate linear regression model that will be used to predict the number of bikes that will be required at any given hour of the day relative to the weather conditions present at that particular time. Our model will take a number of variables, mainly variables attributed to the weather and try to predict the number of bikes that would be needed to supply the demand at that time. Our main objective will be preceded by a statistical analysis of all the attributes within the data set.

SOLUTION OVERVIEW :

In our analysis,we have used Jupyter Notebook for analysis. The file was downloaded as a Comma Separated file (CSV) from this website https://archive.ics.uci.edu .The correlation between the variables are plotted using modules like Seaborn, Matlab and Pandas. We use Multiple Linear Regression,Cross Validation,Random forest to fit the model properly.

DATASET :

The following is a list of the attributes from the dataset, and a brief description of their purpose:

1. Instant: The number of the instant (there are 17,379 instances in the hourly data)

2. Dteday: Date of the year (range: 1st Jan 2011 – 31st Dec 2012)

3. Season: 1: Winter, 2: Spring, 3: Summer, 4 : Autumn

4. Yr: The year, either: 0: 2011 or 1: 2012

5. Mnth: The month of the year: 1: Jan, 2: Feb, 3: Mar, 4: Apr, 5: May, 6: Jun, 7: Jul, 8: Aug, 9: Sep, 10: Oct, 11: Nov, 12: Dec

6. Hr: Hour of the day, working over a twenty-four hour period. (0-23 hrs)

7. Holiday: This refers to whether the day is a public holiday or not. 0: No, 1: Yes

8. Weekday: Refers to the day of the week, 0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday.

9. Workingday: If day is a working day: 1, if day is weekend/holiday: 0

10. Weathersit: this breaks the day up into 4 weather categories: 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

11. temp : Normalized temperature in Celsius. The values are divided to 41 (max)

12. atemp: Normalized feeling temperature in Celsius.

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

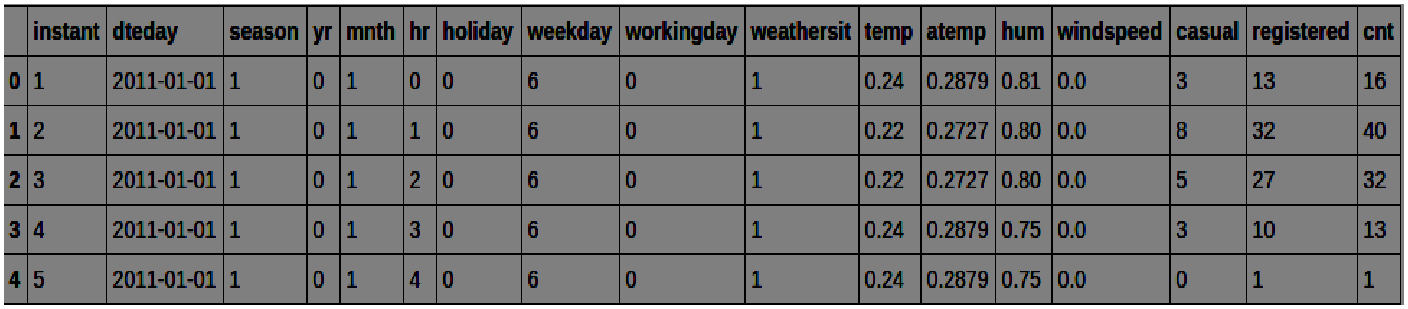
14. windspeed: Normalized wind speed. The values are divided to 67 (max)

15. casual: count of casual bike users.

16. registered: count of registered bike users.

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

Here is the top view of the dataset.



	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0	3	13	16
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0	8	32	40
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0	5	27	32
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0	3	10	13
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0	0	1	1

EDA :

Now we are going to analyse the problem. The dataset is imported to notebook using Pandas module. Now we are going to find out the relation between the attributes.

(i) Box plot between temperature and seasons :

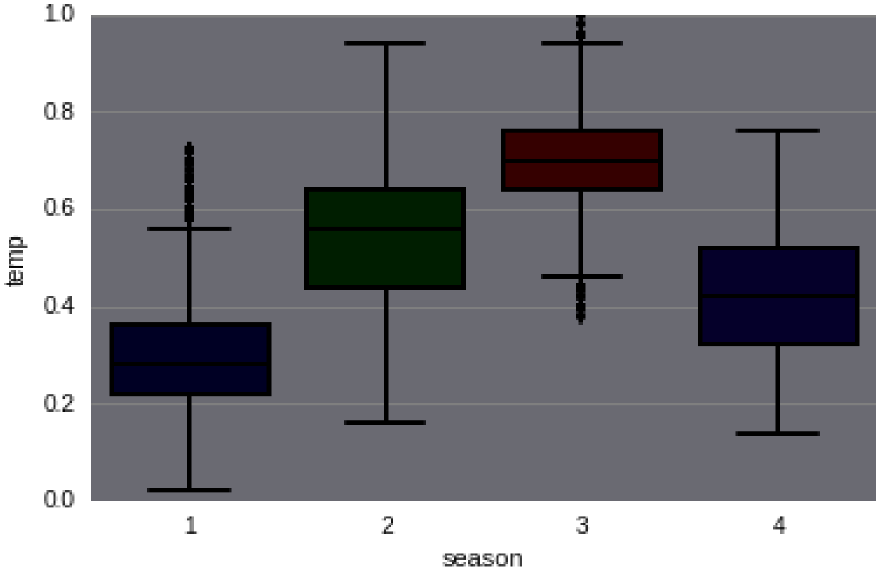


Figure-1

The above box plot shows the relation between temperature and the seasons. We can see that season 3(i. e. summer season) temperature is highest among the all seasons. And Winter season has the lowest temperature.

(ii) Box plot between casual users with seasons and registered users with seasons :

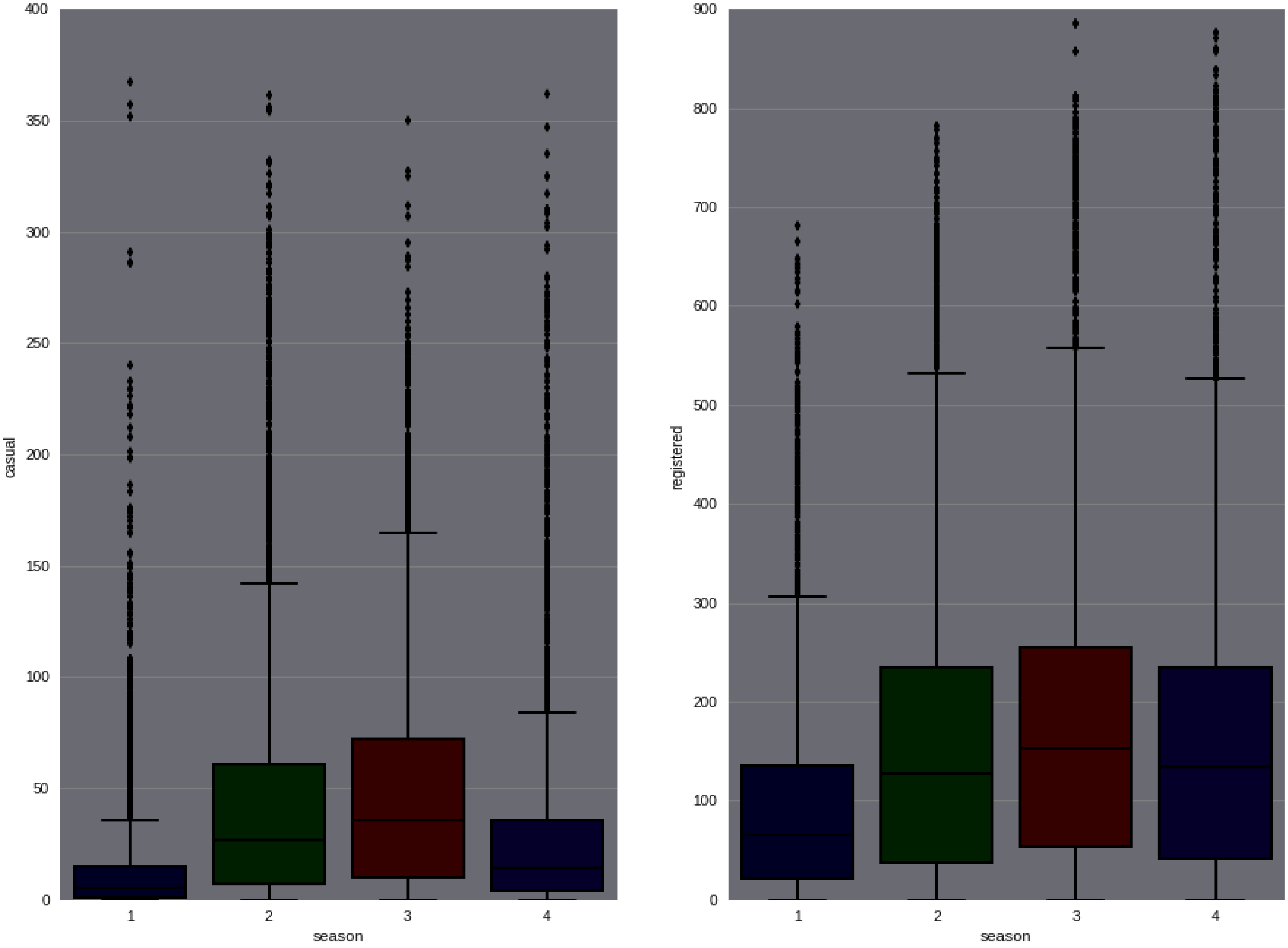


Figure-2

The above plot shows that casual users increases in Summer season and a few users use the system in Winter season. So Summer season can make a good business in bike share system.

(iii)Plot between hours in x-axis and casual users in y-axis:

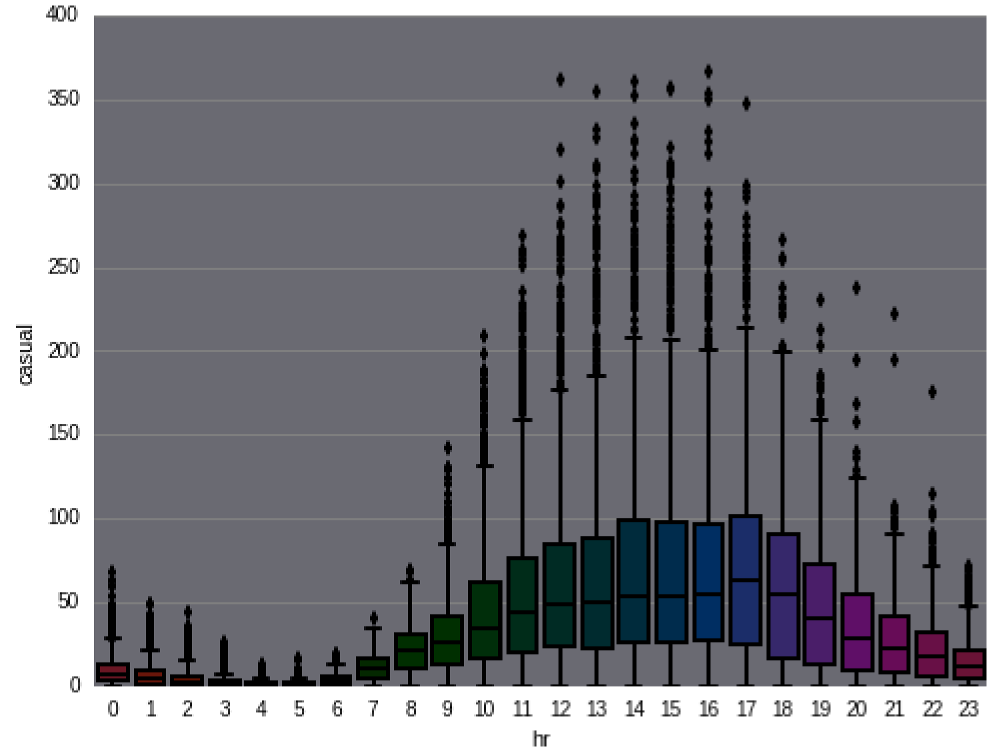


Figure-3

The above plot shows increases and decreases of number of users among the day 10.00 to 19.00 is a good time for bike-share system for casual users.

(iv)Plot between hours in x-axis and registered users in y-axis:

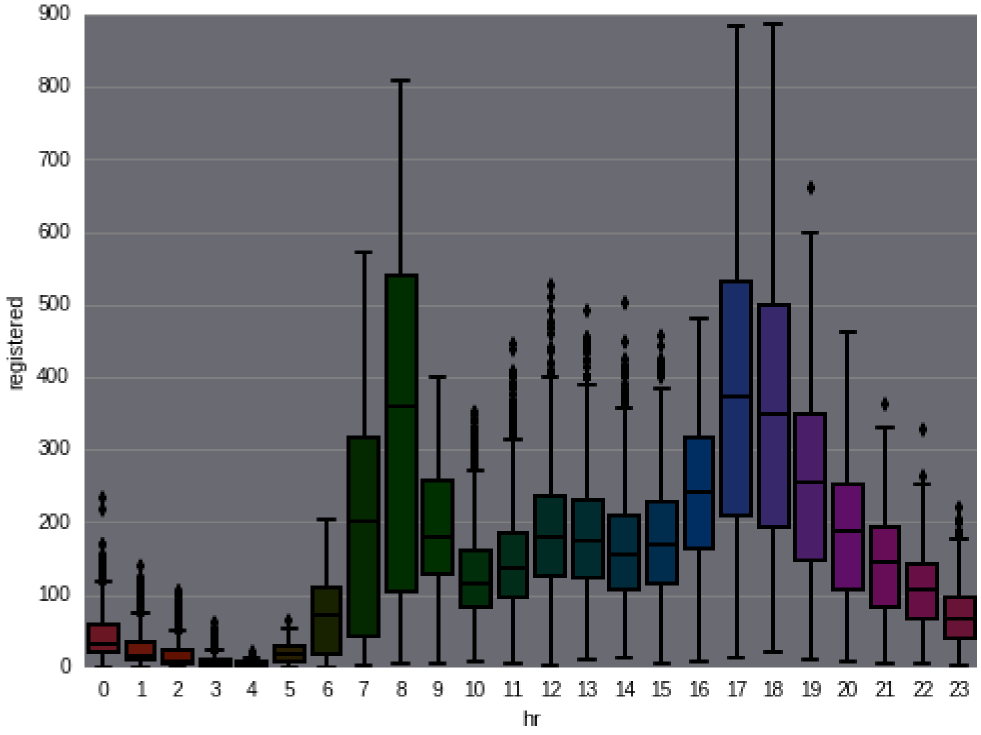


Figure-4

The bike rentals for registered users can be segregated into three categories:

1. High : 7:00-9:00 and 16:00-20:00

2. Average : 10:00-15:00 and 20:00-22:00

3. Low : 0:00-6:00 and 22:00-24:00

(v) Correlation Matrix :

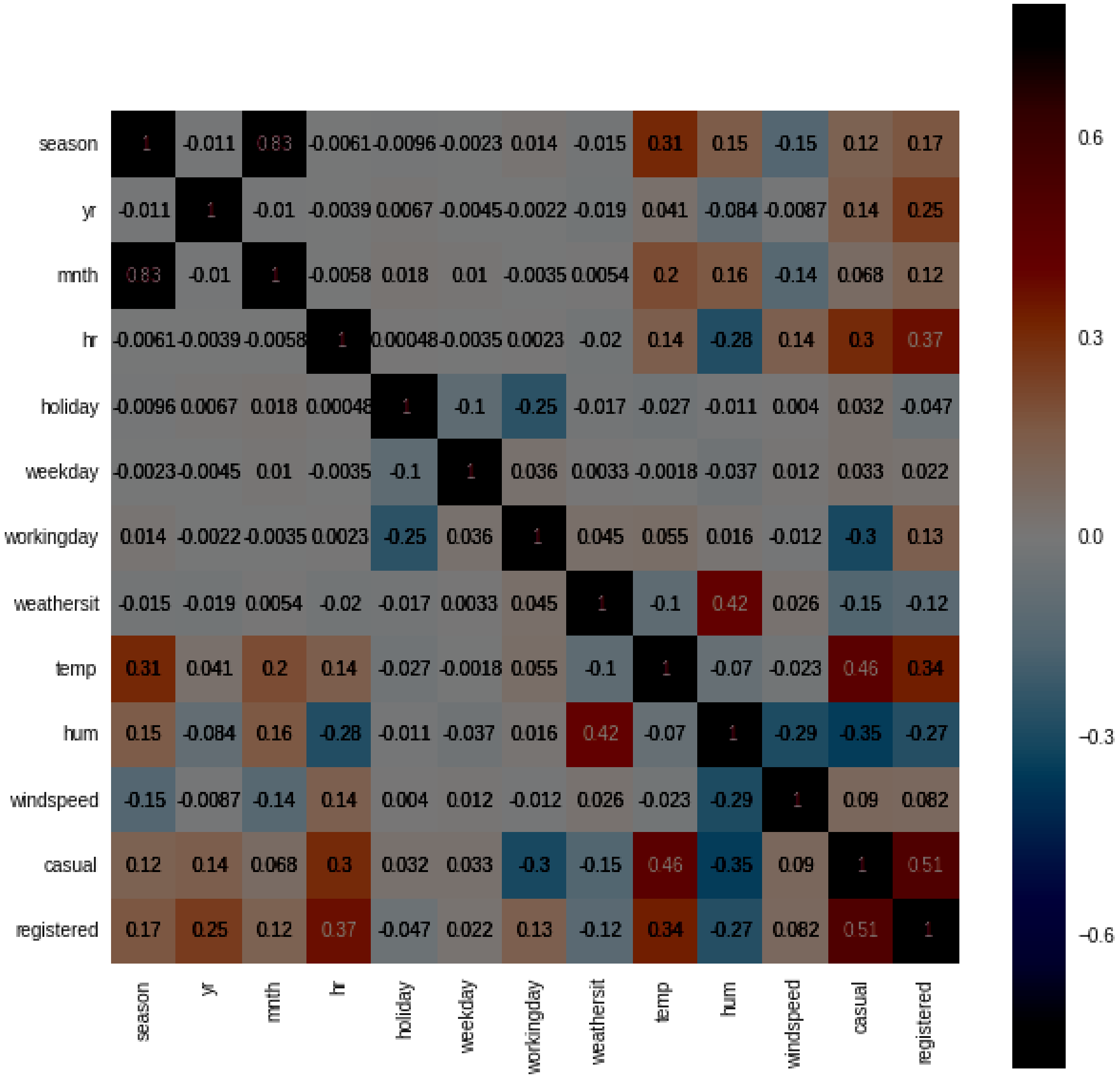


Figure-5

The above plot shows correlation between the attributes. There are a number of variables which are negatively correlated. It should be noted that we are particularly interested in identifying the variables that are strongly correlated, when building our model.

Some of the observations obtained from the correlation matrix are:

- Temperature has a higher degree of positive correlation with casual bike rental count as compared to registered bike rental count. That means people prefer to ride a bike at cool to medium temperatures than cold temperatures in Washington DC.

- Humidity has a higher degree of negative correlation with casual users count as compared to registered users count. This suggests that high humidity days like rainy season attracts less riders.

- Wind Speed is positively correlated with bike rental count but the correlation is low.

(vi)Plot between temp and atemp:

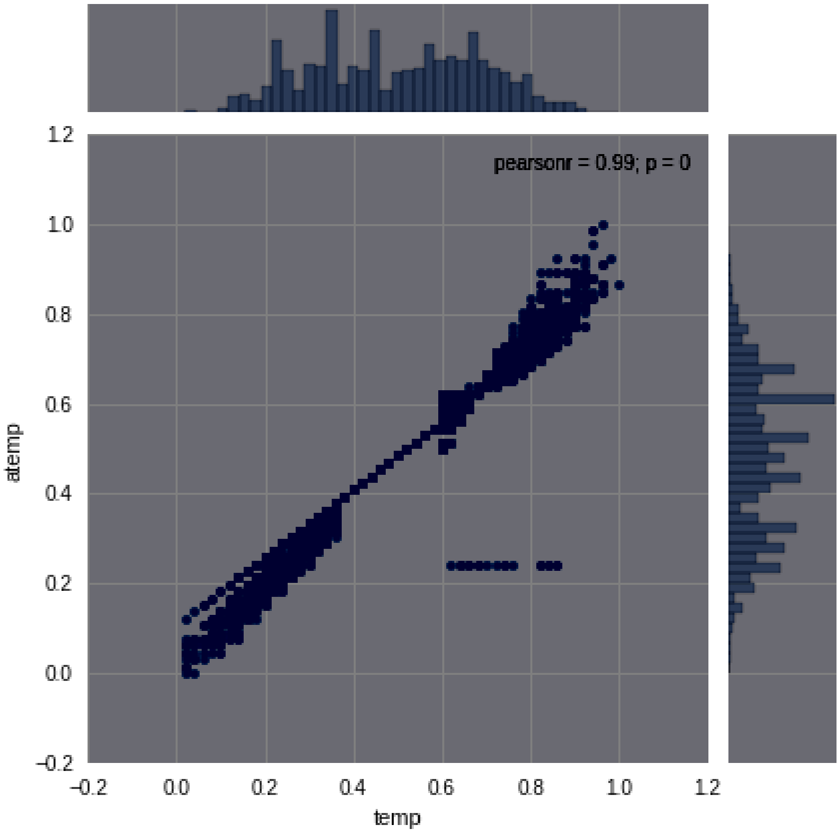


Figure-6

From the above joint plot we can see temperature felt (atemp) is very highly correlated with the actual temperature and hence it would be best to consider the only the temperature felt.

EDA Summary:

We observed during initial analysis of data that registered and casual users had different cycling behavior. The behavior for casual users was much more dependent on wind-speed, temperature, season etc. Also,while the bike renting increased during morning and evening for registered users,it increased around evening for casual users. Thus we decided to create separate models for the two types of users.

Dist plot of the target variable(casual users) :

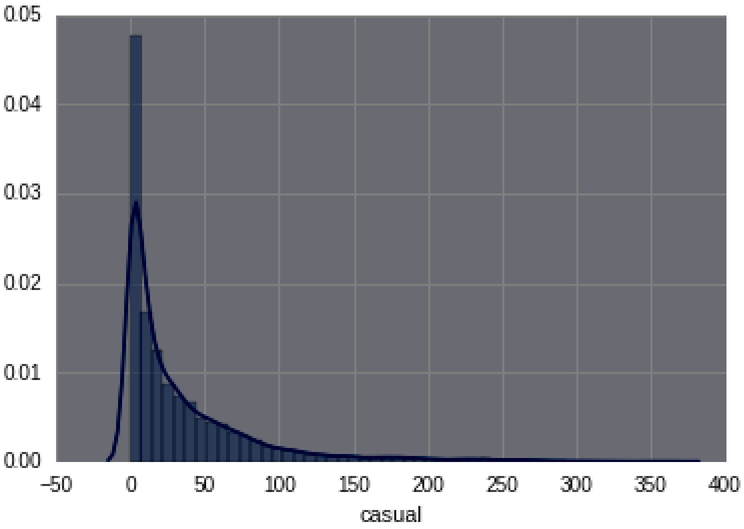


Figure-7

Dist plot of the target variable(registered users) :

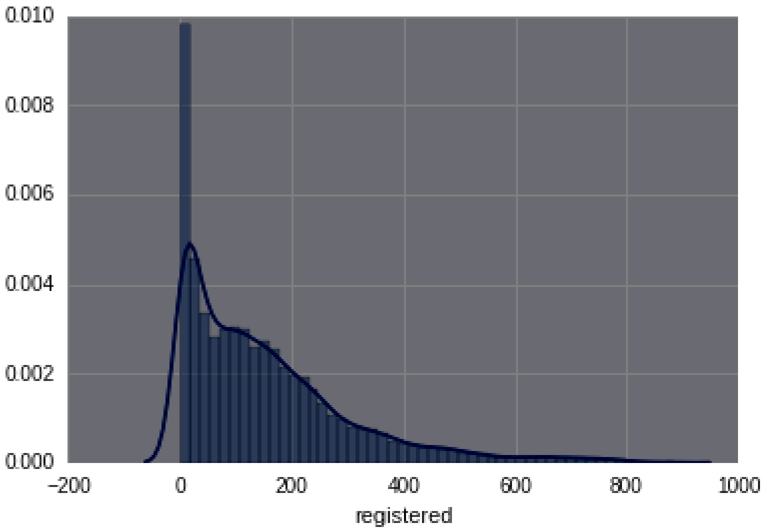


Figure-8

MODELING :

1.Multiple Linear Regression :

Multiple linear Regression model uses multiple independent variables to predict one dependent variable.

The shape of the dataset is 17379.We split the dataset using Sklearn module to training and testing dataset. We took 30% dataset for testing and 70% for training. Season,Year,Month,Hour,Holiday,Weekday, Weathersit, Temperature, Humidity, Wind-speed are the independent variables that will use in multiple regression model. We have built two separate models for casual users and registered users .We have used the multiple linear regression to our model.

Result of Multiple Linear Regression Model :

The scores of Casual and registered users are given below :

Score of	Casual users	Registered users
Training set	0.45292493123339106	0.33767442848493234
Testing set	0.45894678334104122	0.32865949785119852

2.Log-linear Regression Model :

As our target variables 'casual' and 'registered' follow poisson distribution ,for that multiple linear regression is not a good approach for this dataset. We are going to use log-linear model for this model.

Result of Log-linear Regression Model :

The results are given below :

	Casual users	Registered users
Training set	0.59323638559793235	0.46535246190137702
Testing set	0.59376246937866395	0.46151500292045505

Residual :

The difference between the observed value of the dependent variable and the predicted value is called the **residual** . Each data point has one residual.

Residual = Observed value - Predicted value

A **residual plot** is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a non-linear model is more appropriate.

Scattered plot of Residual Testing of casual users:

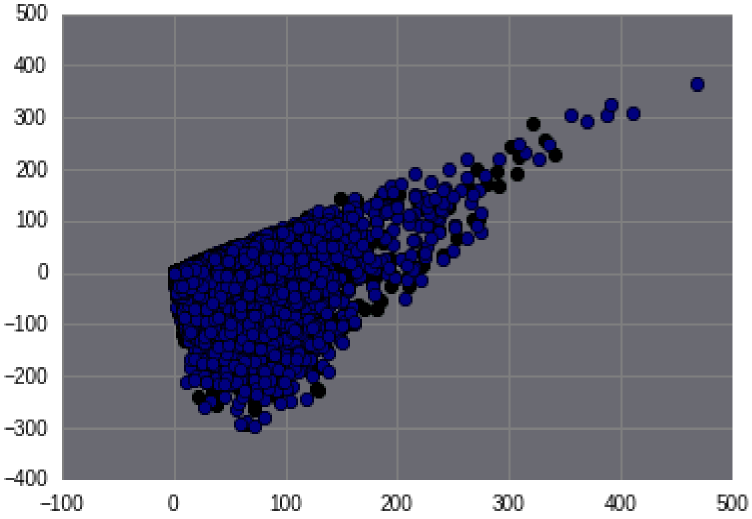


Figure-9

Scattered plot of Residual Testing of registered users:

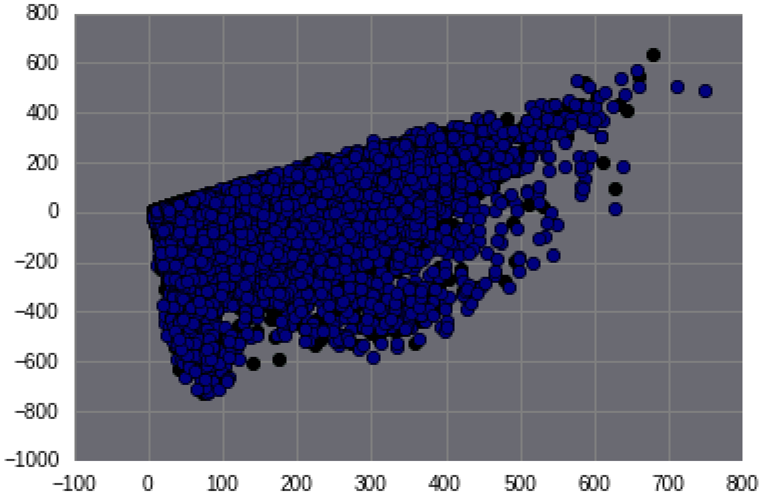


Figure-10

3.Random Forest :

Random forest operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

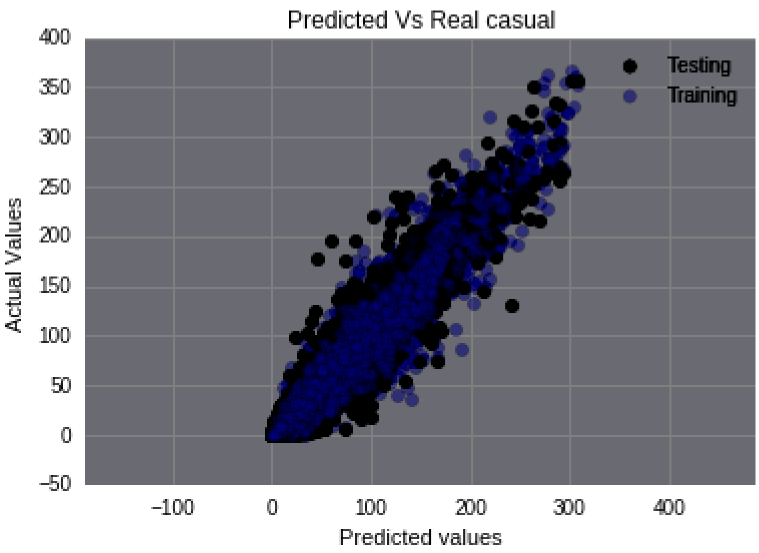
We are going to use Random forest for this model.

Result of Random Forest :

We have calculated Root mean squared error (RMSE) for both training and testing set and R-Squared (training and testing accuracy).The results are given below in the table.

	Casual Users	Registered Users
RMSE for Training Set	12.154631177006404	28.690022489822496
RMSE for Testing Set	15.387032650913973	35.656216740581513
Training accuracy	0.94%	0.96%
Testing accuracy	0.90%	0.94%

Plot between Predicted values versus Real values :



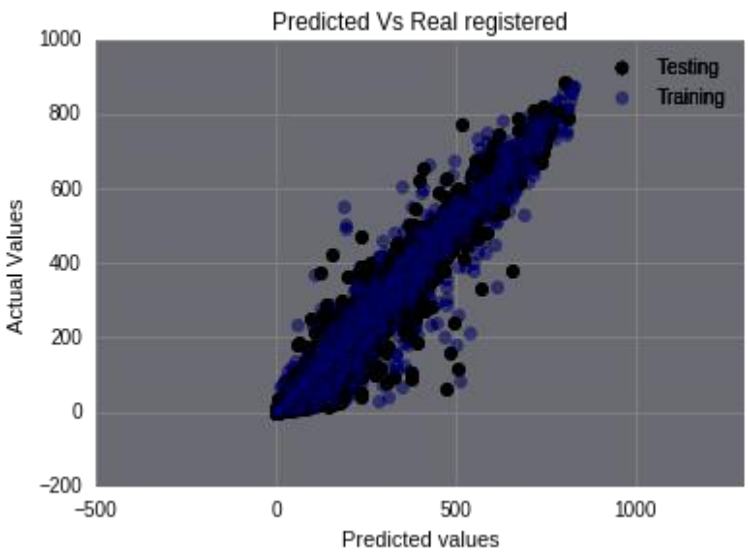
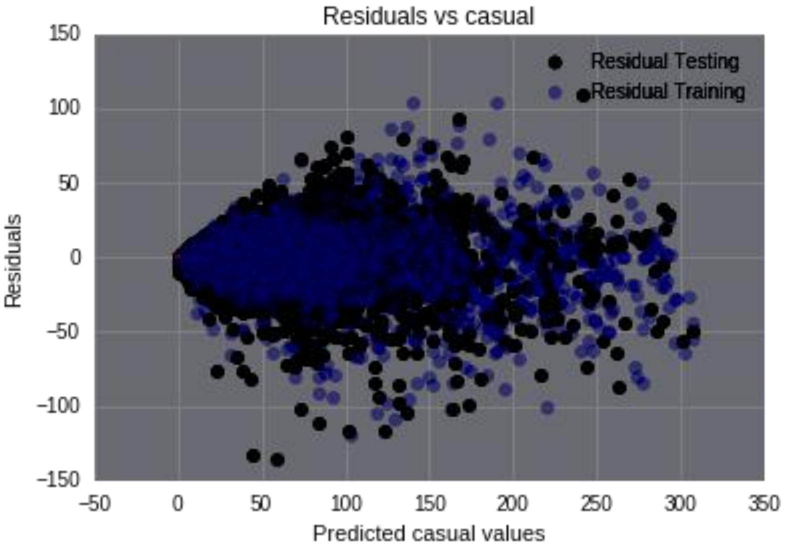


Figure-11 Figure-12

Homoscedasticity :

The assumption of homoscedasticity (meaning “same variance”) is central to linear regression models. Homoscedasticity describes a situation in which the error term (that is, the random disturbance in the relationship between the independent variables and the dependent variable) is the same across all values of the independent variables. Heteroscedasticity (the violation of homoscedasticity) is present when the size of the error term differs across values of an independent variable. The impact of violating the assumption of homoscedasticity is a matter of degree, increasing as heteroscedasticity increases.



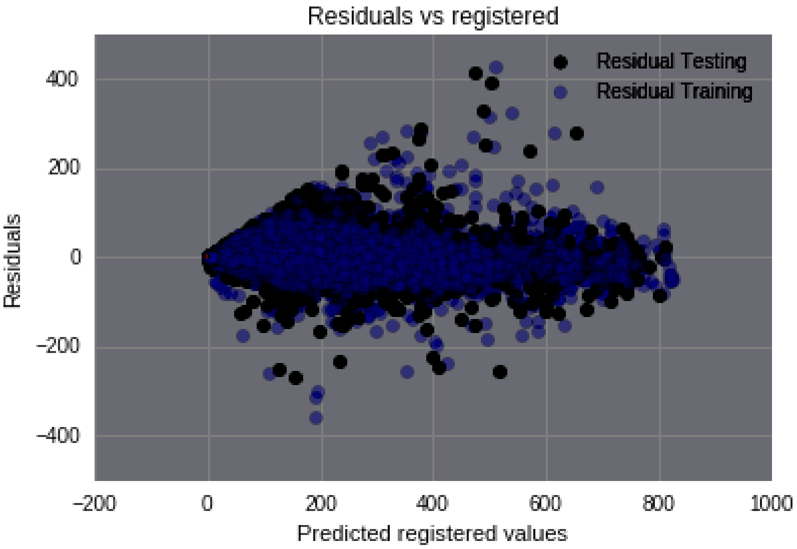


Figure-13 Figure-14

From the above plots,we can see the homoscedastic behavior of residuals versus target variable. Figure(13) is more heteroscedastic than that of figure (14).

The feature importances of each of the feature used is our model shown below.

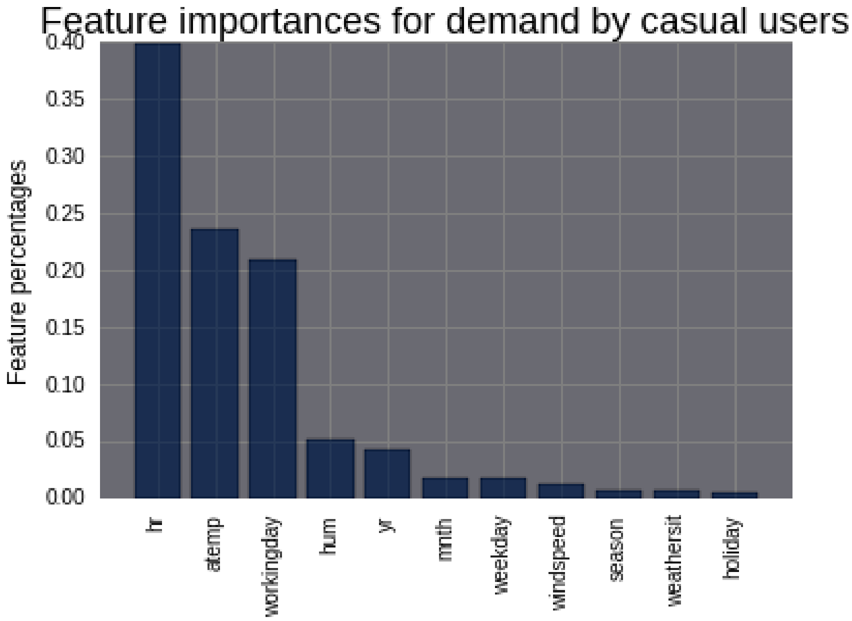


Figure-15

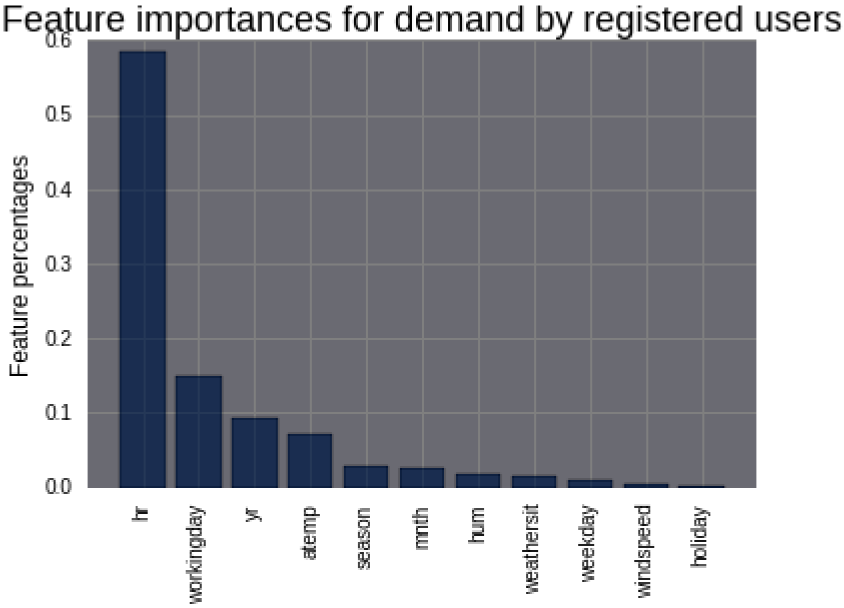


Figure-16

From Figure(15) and Figure(16) , we can see that for both demands by casual users and registered users, hour is the most important feature, the feature importance has higher value in case of casual users. Temperature has higher importance in case of demand by casual users.

CONCLUSION :

We established significant relationship between several independent variables and Bike-sharing ridership. We developed a regression model that can be applied directly to bike station business to predict hourly demand. We also found that the usage of bike rental is far more high for registered users as compared to casual user and the demand is maximum during morning and evening travel hours. Also temperature has significant effect on bike ridership. We trained various different models and found that Random Forest was best to capture the variance and non linearity of the dataset. It gave an RMSE of 90% for casual users and 94% for registered users on test set.

REFERENCE:

[1] https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

[2] https://en.wikipedia.org/wiki/Random_forest

[3] http://stattrek.com/regression/residual-analysis.aspx?Tutorial=AP

[4] http://www.statisticssolutions.com/homoscedasticity/

[5] http://cs229.stanford.edu/proj2014/Jimmy%20Du,%20Rolland%20He,%20Zhivko%20Zhechev,%20Forecasting%20Bike%20Rental%20Demand.pdf

[6] http://scikit-learn.org/stable/modules/sgd.html