Bike Renting

Gaurav Malik

Contents

Introduction

1.1 Problem Statement

Bike Renting

This project is on making prediction of bike rental count on daily basis based on various environmental conditions and seasonal settings.

Our motive is to make a model which provides us an insight of the data through various plots and find the best model of Machine Learning algorithm which would make the prediction for the given dataset.

1.2 Data Interpretation

The dataset contains the following features

- 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 [Web Link])
- weekday : day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.
- weathersit :

  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_max-t_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

The description of the data is as follows:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
instant       731 non-null int64
dteday        731 non-null object
season        731 non-null int64
yr            731 non-null int64
mnth          731 non-null int64
holiday       731 non-null int64
weekday       731 non-null int64
workingday    731 non-null int64
weathersit    731 non-null int64
temp          731 non-null float64
atemp         731 non-null float64
hum           731 non-null float64
windspeed     731 non-null float64
casual        731 non-null int64
registered    731 non-null int64
cnt           731 non-null int64
dtypes: float64(4), int64(11), object(1)
```

From the above dataset description we can see that there are total of 16 features with no missing data.

Methodology

2.1 Pre Processing

Data Pre Processing is just as important as creating a machine learning model. Before using the models of Machine Learning it is very essential that we first make the data which can be better suited to the algorithms. This process mostly includes cleaning the data, look for missing values, handling categorical data, looking for outliers. During our pre processing we also need to need to plot various bars and plots to look for outliers and the distribution of the data too.

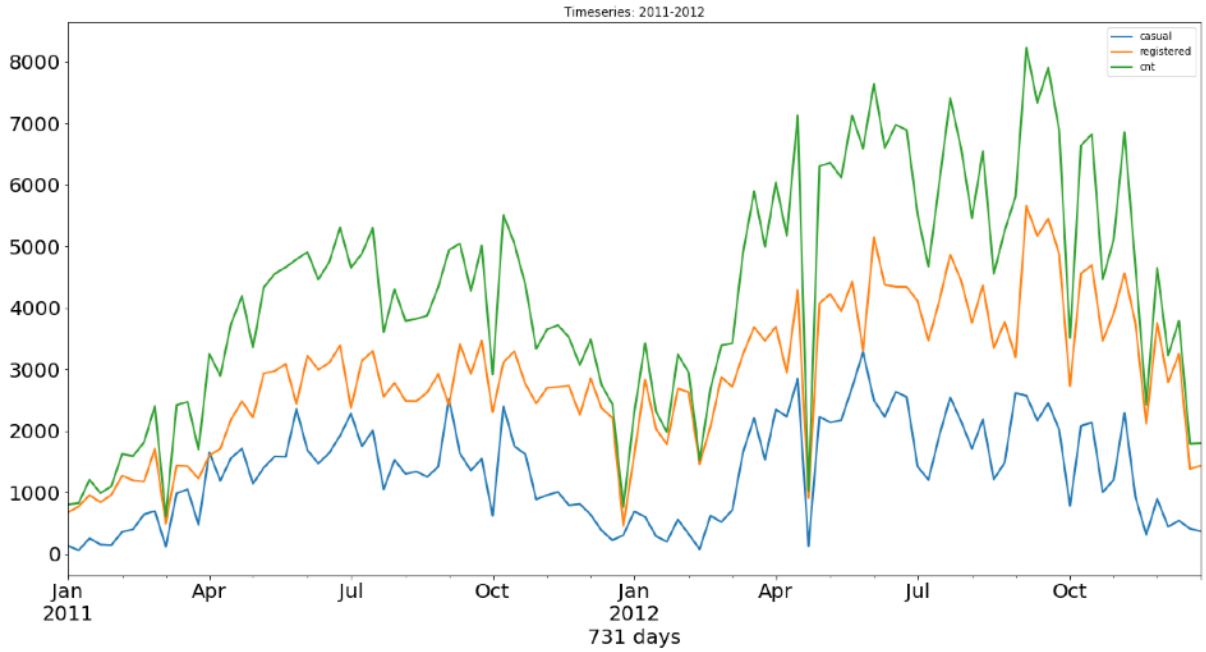
2.1.1 Imputing NA Values

As observed in the previous section that there are no null values in our dataset so there is no need for us to perform any imputations and we can directly look at the distribution of our data.

2.1.2 Distribution Histograms

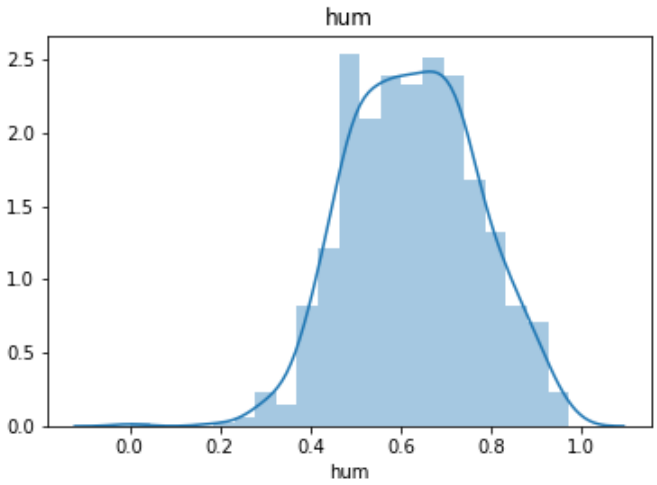
While we do our pre processing it becomes very important for us to observe the distribution of data. To at least have a sense of the data distribution in our dataset we must have an insight of the data distribution and we do this by plotting histograms of the data features. And as we have our data properly stored with dates we can make a time series plot to gain a proper insight and just look at the distribution of the data.

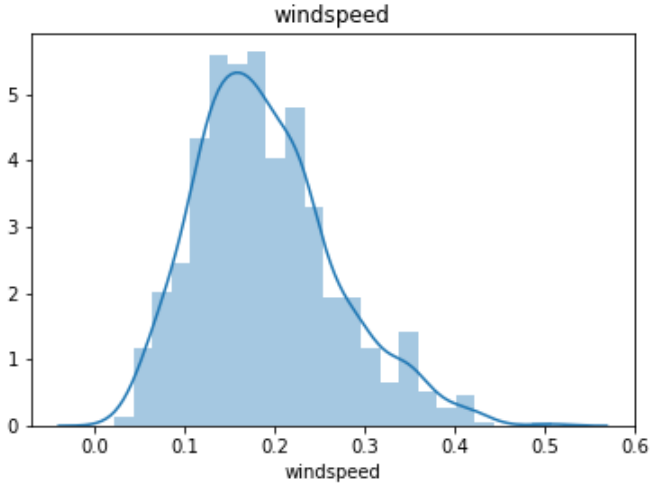
The time series of the data is plotted below:

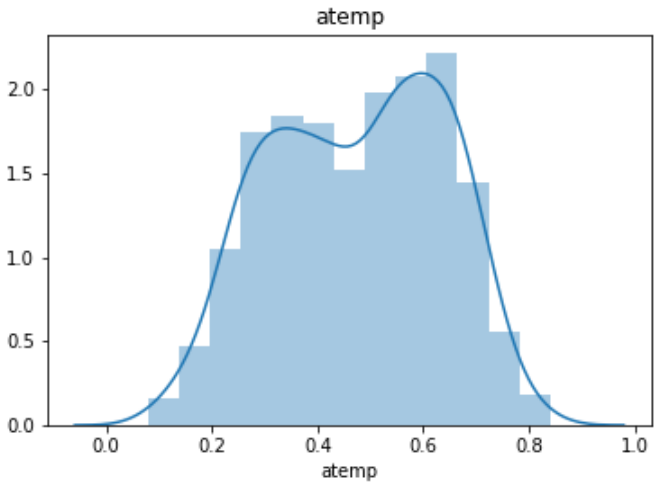


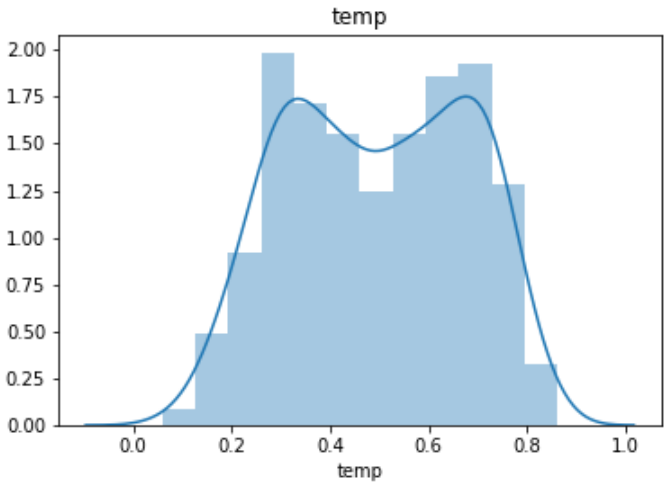
We can observe from the above time series plot that there is a sharp decrease in bike renting in the time period between October and February and this pattern is repeated at the end of the plot too. By this we can infer that during winter people are less likely to book a bike than any other season.

Now let us take a look at the data distribution. Following are the histograms of columns indicating their distribution









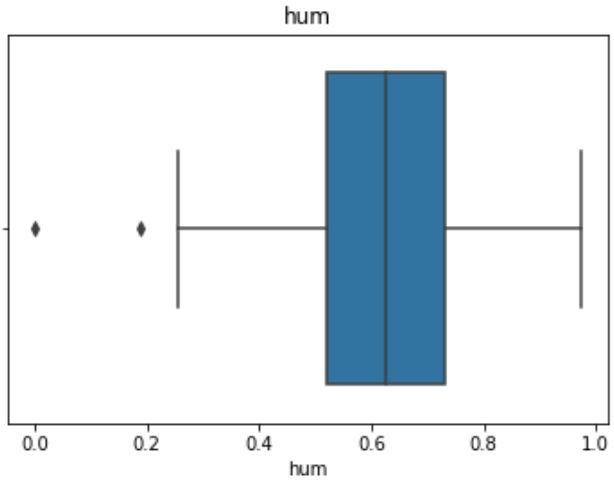
Before we discuss about the above distribution plots, let's first reflect on all the remaining columns or features. As the data was recorded daily for two years. There won't be any need for us look at features containing data about weeks, months, weekday, working day, etc. but that does not mean that they are not important. They are but as categorical values they won't reflect on some specific insights.

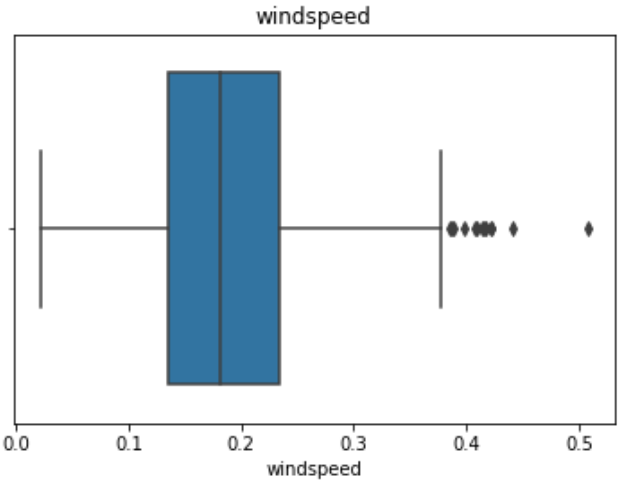
Now, looking at the above plots we can say that feature containing windspeed data is not skewed so we have to deal with it, as we will do it in the next section.

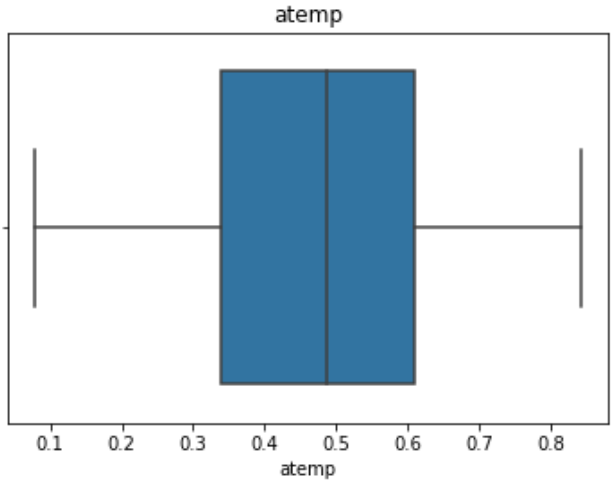
2.1.3 Outliers Analysis

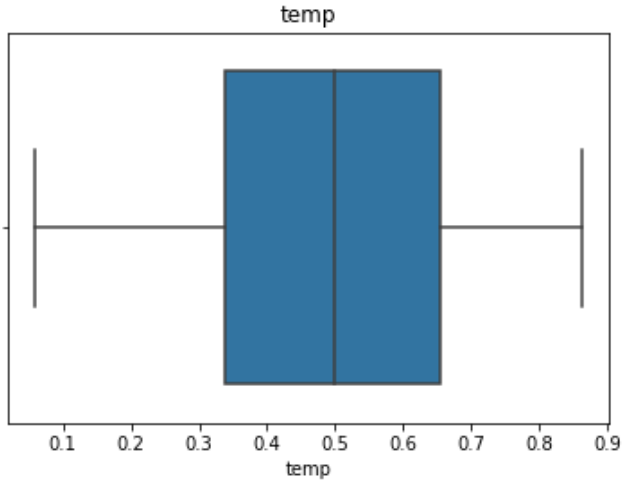
When we are cleaning our data, analysing outliers becomes among essential things to do. With the help of outliers box plot we can see whether the data which is too far from the rest of the data points will affect the prediction and remove them replacing with better values like mean of the data.

Following are the box plots for the data we have on customers:



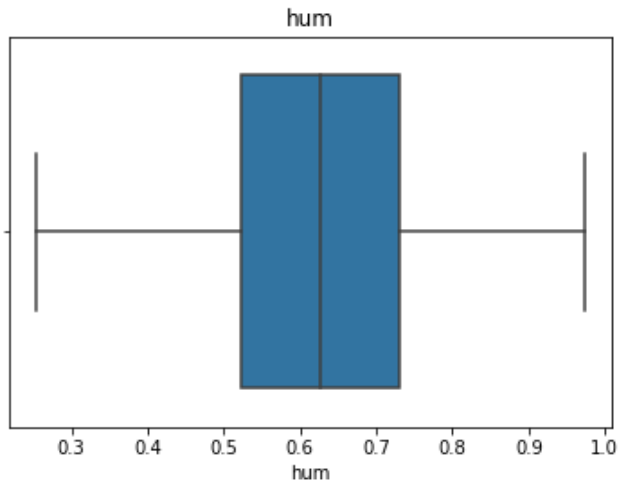


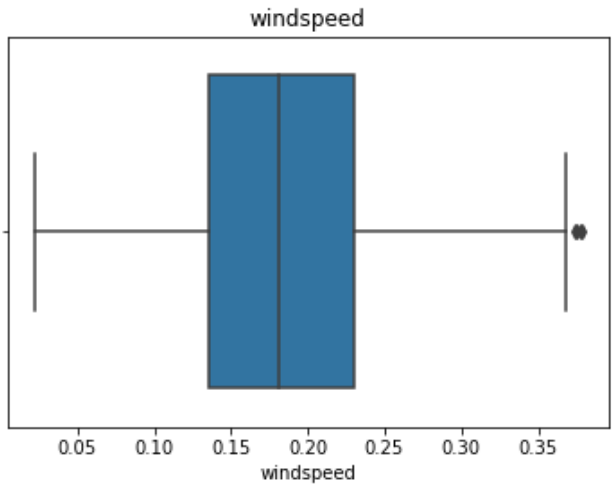


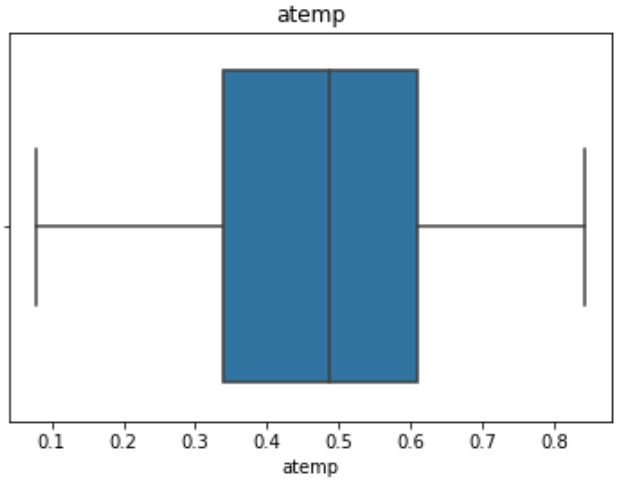


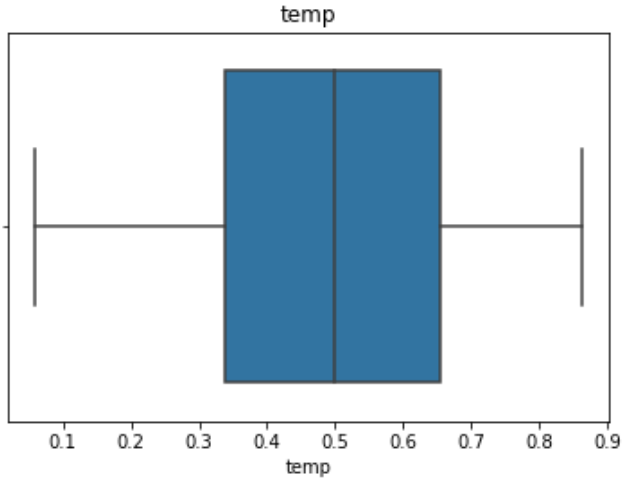
We can observe that there are some outliers in the columns ‘hum’ and ‘windspeed’. So we remove those outliers give them some suitable values as the scaling has been done using MinMaxScaler which are very sensitive of outliers.

So after removing the outliers and replacing them with suitable values i.e. the median of the columns, the following is the plot after treating the outliers.









2.1.4 Feature Selection

We have looked upon our data, got a general hint of what it includes, how it is distributed. During data pre processing there comes a problem of whether to select all the features of the data or not. At times the features are all independent of each other but sometimes they are not. So, we have to filter out those highly correlated features.

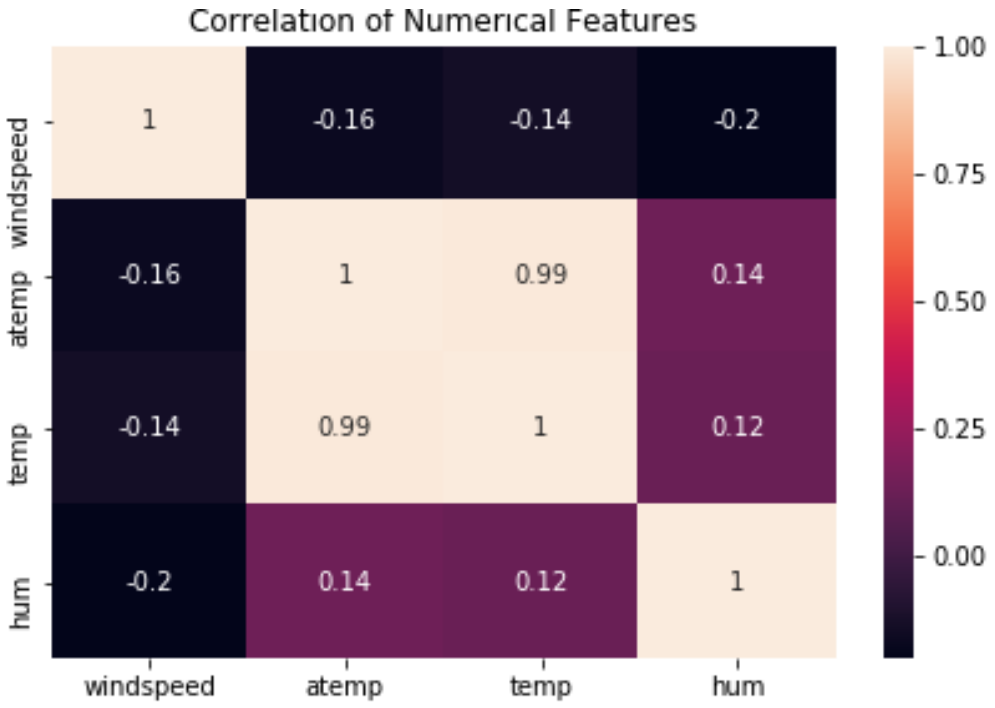
Correlated features in general do not improve the models. There are various benefits to removing the correlated features-

- Making the learning of algorithms faster
- Decreases the biasing in data
- Make the model much simpler and interpretable

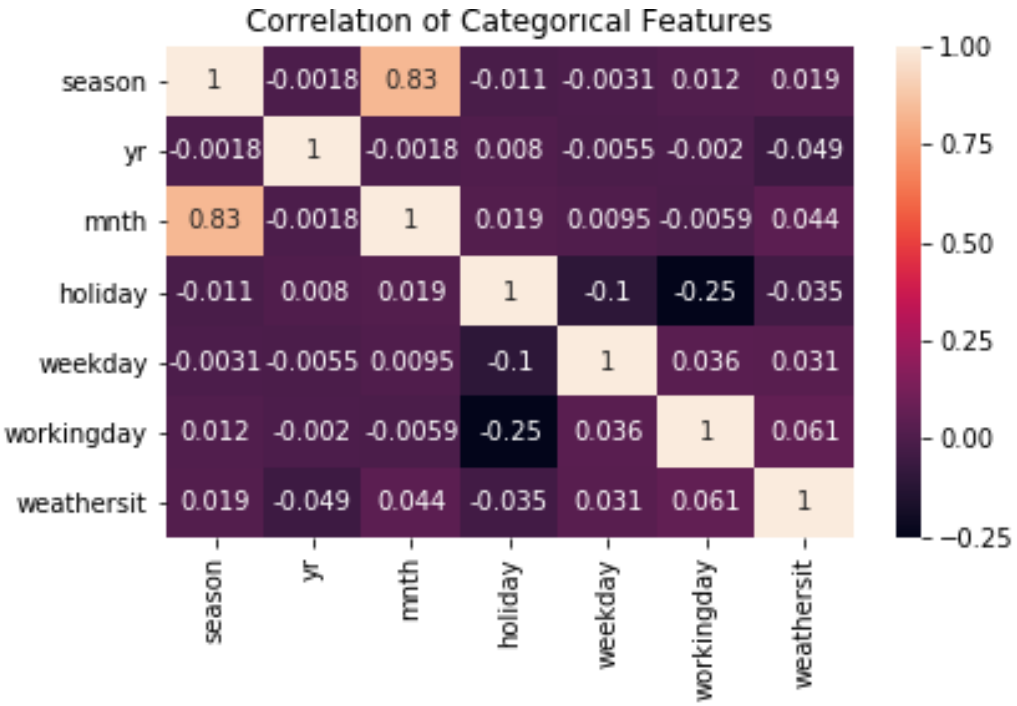
For linear models like linear regression, logistic regression multicollinearity can make the prediction highly unstable. For models like Naive Bayes actually benefits from “positive” correlated data and for random forest regression the benefits are indirect as random forest is good at detecting interactions between different features.

So, removing these features can at times be necessary to speed up the learning. As the aim of the data scientist is to make the data interpretable it becomes essential to make the data simple.

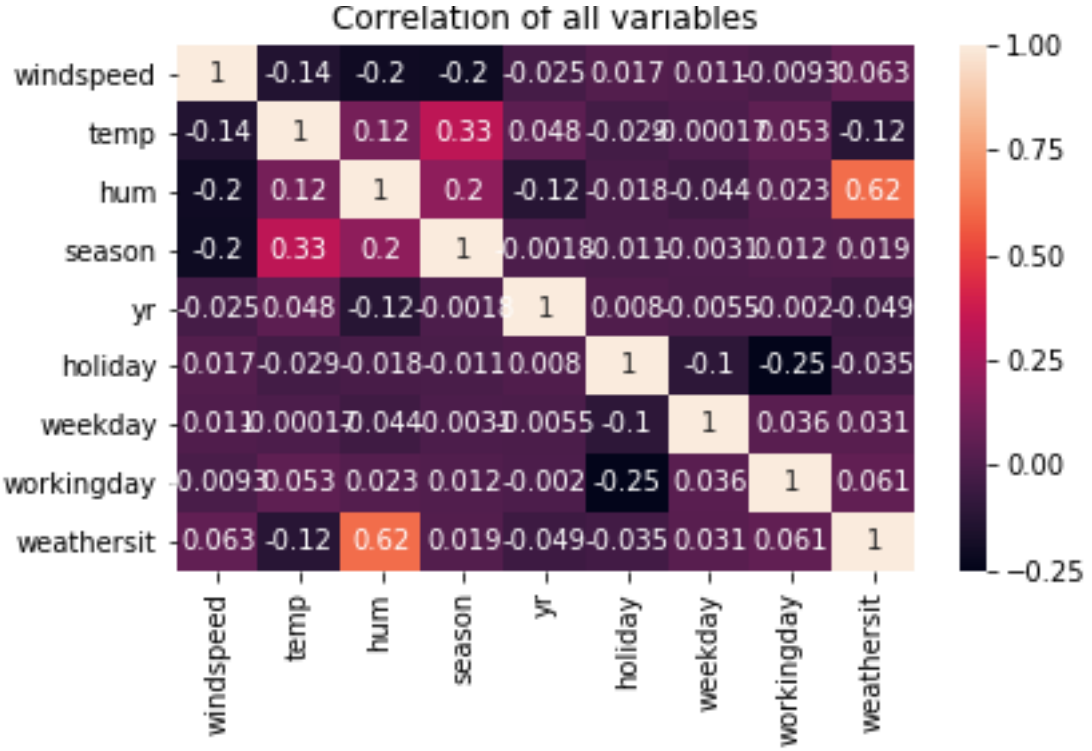
Following shows the heat map of our data:



As observed ‘temp’ and ‘atemp’ are highly correlated, we remove ‘atemp’ from the dataset.



In this correlation plot we can see that ‘mnth’ and ‘season’ are highly correlated so we remove the ‘mnth’ feature. Now lets check out all the features’ correlation plot.



And finally we remove ‘hum’ as it is highly correlated to ‘weathersit’.

2.1.5 Encode Categorical Data

What is Categorical Data?

Categorical data are variables that contains labels as their value instead of having numbers.

Problem With Categorical Data

Many machine learning algorithms requires input and output variables to be numeric. They cannot work on categorical data directly, unlike decision tree which can work on categorical data without any problem.

In order to avoid these problems it is preferred to convert all categorical data to numeric form. In case the output variable is categorical variable too, we have to convert it back to categorical variable to present them or to use in applications.

Different Methods To Encode Categorical Data

We can encode categorical data using either of the two steps:

- Integer Encoding
- One-Hot Encoding

In integer encoding each unique categorical variable gets a unique number. This is also called label encoding.

For example in a data set we have three cities: New York, Mumbai, Moscow

In label encoding each of the variables get a number. Like, “New York” would be 1, “Mumbai” would be 2, and “Moscow” would be 3.

With label encoding there comes an uninvited guest called natural ordering in the data. This means that the machine automatically assumes that there is a natural order in the categories with one being higher than the other. To get rid of this problem we do One-Hot

Encoding. One-Hot Encoding makes columns same as the number of categorical variables and that too of boolean type.

For example,

NEW YORK	MUMBAI	MOSCOW
1	0	0
0	1	0
0	0	1

The binary variables are often referred as dummy variables.

Steps Followed in this project:

The data we have is already encoded but this data will create a problem as it will rank the features according to their numbers so we used OneHotEncoder in order to avoid the troubles.

2.1.6 Feature Scaling

Why Scaling?

Most of the time the data in different features of dataset have high varying magnitude of data. This is a problem in most of the algorithms as they mostly use Eucledian distance to do their computation. This high magnitude features will weigh more than others features in the distance calculations with low magnitude.

To suppress this problem we need to scale the features at a same level for which we do Feature Scaling.

What Happens while Scaling?

When we use algorithms to scale, all the data is reduced to between -1 to 1 or 0 to 1 depending on algorithms and even the parameters.

When to Scale?

There are various algorithms where scaling makes a difference and some where it does not.

- k-Nearest Neighbours is very sensitive with magnitudes as they depend on Eucledian distance and scaling is needed for this algorithm.
- Models which are tree based like Decision Tree & Random Forest do not depend on distance due to which they can handle ranges of features which vary a lot.

Steps Followed in this project:

Our dataset already scaled the numerical data using MinMaxScaler. We could have scaled the categorical variables too but this time we won't scale. Instead we will let them be and fit the model to the data. The only thing this would affect is the time to train the model. As the magnitude of the data is not too high so it won't affect the model performance.

Model Selection

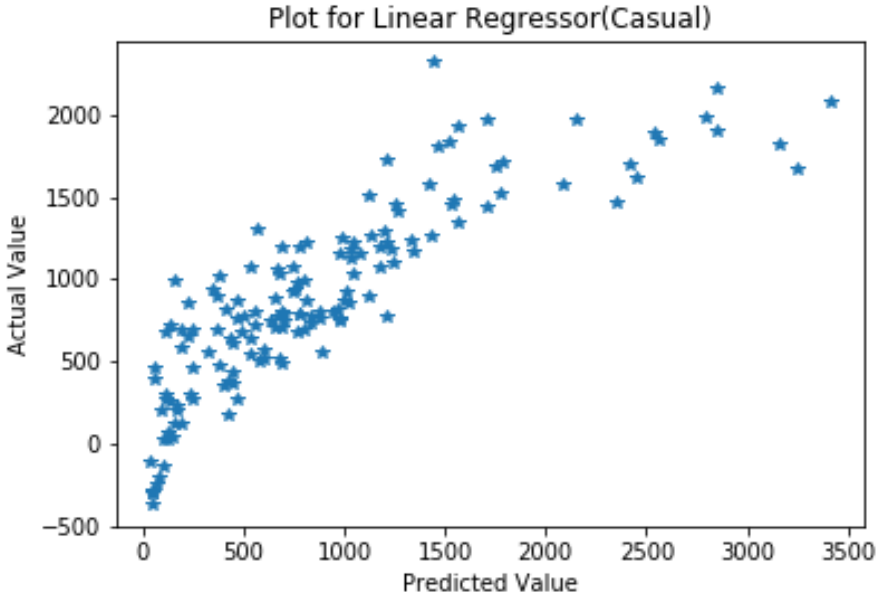
We need to predict the count of bikes renting for a day in terms of ‘registered’ and ‘casual’. For this we need to find the best model which can fit our data and produce acceptable results. As the results are in count then we will be using few regression models like:

- Multiple Linear Regression
- SVM
- Random Forest

Multiple Linear Regression

Using multiple regression we were able to produce the following results:

R-square	- 0.715
Mean Square Error	- 150480.98
Mean Absolute Error	- 277.14
Root Mean Square Error	- 387.92



Support Vector Regression

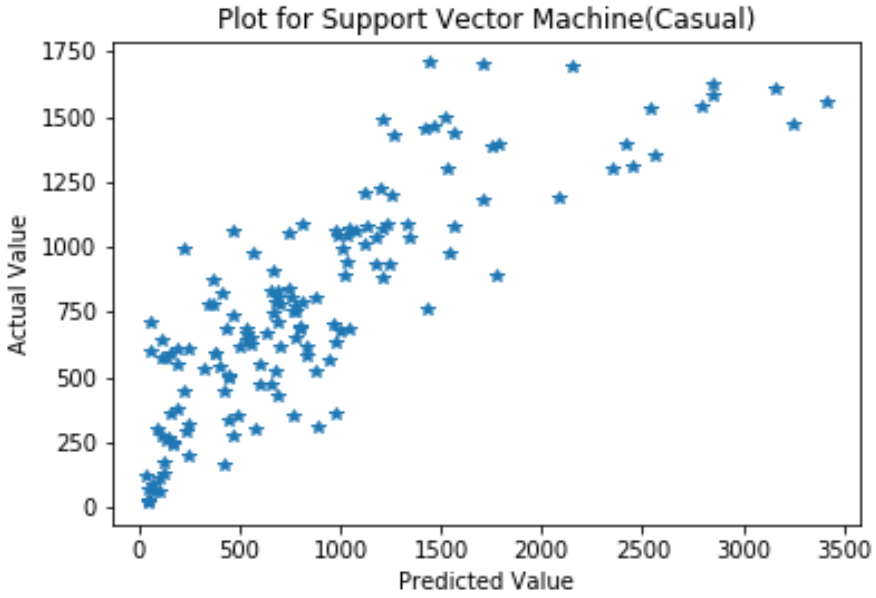
Support Vector Machine is producing the following result after much optimisation:

R-square - 0.61

Mean Square Error - 209387.48

Mean Absolute Error - 293.79

Root Mean Square Error - 457.59



Random Forest Regression

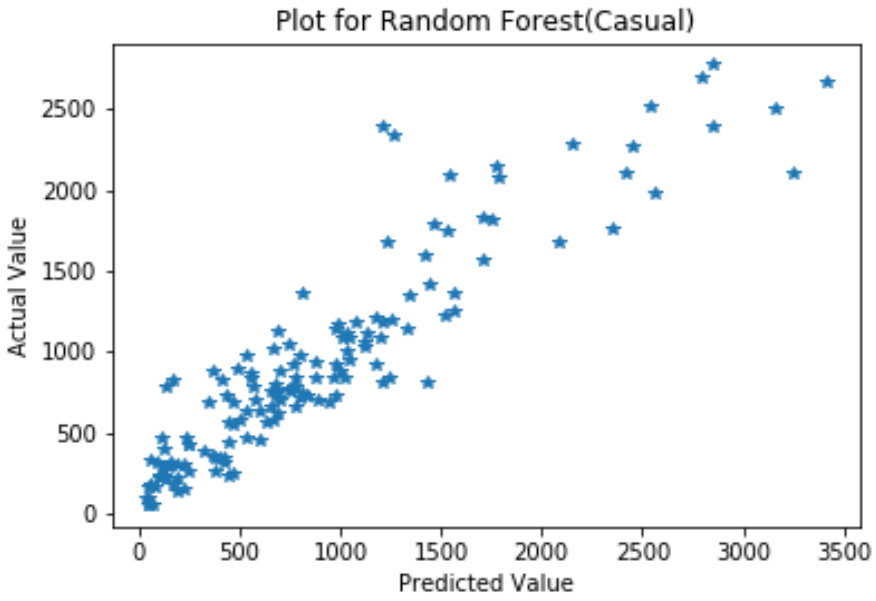
Random Forest algorithm creates one of the best results for this data set. The following is the result of the model:

R-square - 0.84

Mean Square Error - 84262.67

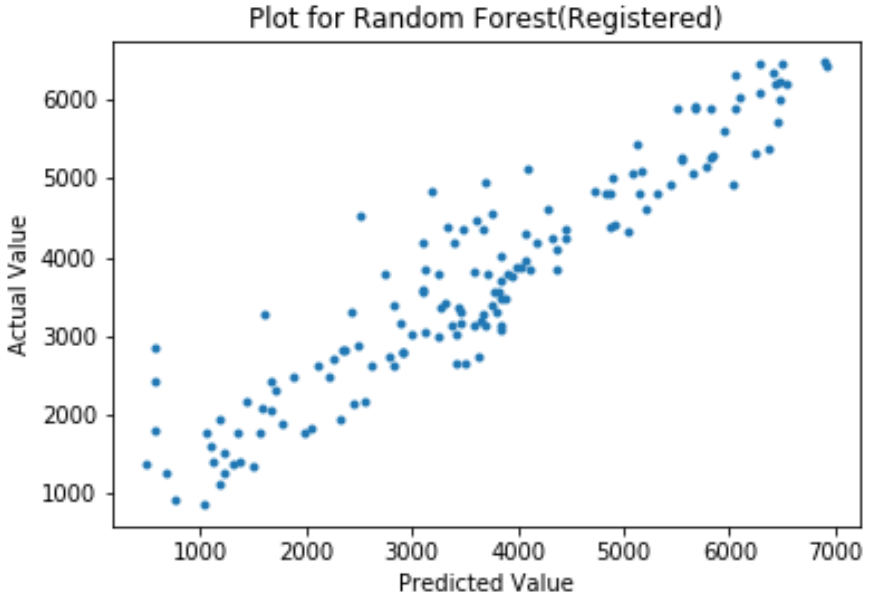
Mean Absolute Error - 198.65

Root Mean Square Error - 290.28



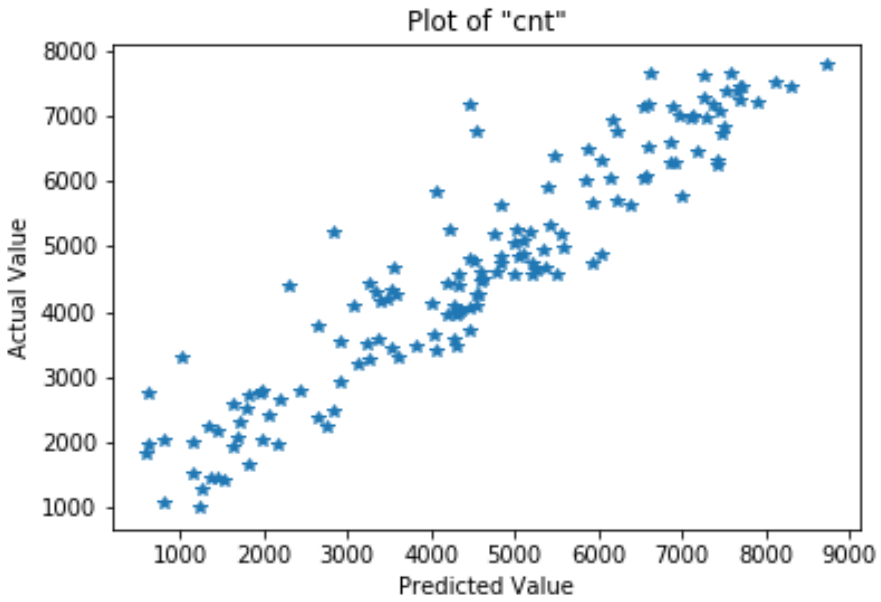
As the best model, Random Forest Regressor, we will now fit for registered bike sharing prediction. The results are as follows:

R-square	- 0.87
Mean Square Error	- 360174.74
Mean Absolute Error	- 449.40
Root Mean Square Error	- 600.15



Conclusion

Now we finally see the plot of count as we add the regular and casual prediction:



The plot seems good as almost all the predicted and actual values lie in a straight line with a particular width of the range. But seems perfect as a model.

References

- Medium
- www.towardsdatascience.com
- www.stackoverflow.com
- https://stats.stackexchange.com/
- Analytics Vidya
- Research Gate