Bike Sharing Demand Prediction

Final Report

Name: Xingkai Wu

Python Code(Github) Link:
https://github.com/xwu0223/Bike-Sharing-Prediction-Project/blob/master/Bike_Sharing_Demand_Final.ipynb

Problem Statement:

After the completion of this project, the Washington D.C city bike sharing program can predict the user count of bikes in different locations and therefore to relocate bikes at low demand area to high demand areas in order to increase the utilization ratio of the bikes.

What will client do or decide based on analysis

By forecasting bike rental demand of bike sharing program in Washington D.C based on historical usage patterns in relation with weather, time and other factors. The city will relocate bicycles based on predictions to meet the demand and maximize the utilization ratio of the bikes, and therefore generate less pollution in the city.

Data Sets:

The data sets are available from this link in .csv format, the data is available from year 2010 to 2012, the data is relatively clean and easy to use. It has the columns of
datetime - hourly date + timestamp
season - 1 = spring, 2 = summer, 3 = fall, 4 = winter
holiday - whether the day is considered a holiday
workingday - whether the day is neither a weekend or holiday
weather -
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 - temperature in Celsius
atemp - "feels like" temperature in Celsius
humidity - relative humidity
windspeed - wind speed
casual - number of non-registered user rentals initiated
registered - number of registered user rentals initiated
count - number of total rentals

Data Wrangling

Data wrangling steps:
Checking if there is any missing values(NaN),
From the code and result shown below, there is no missing values.

```
In [63]: hour.isnull().sum().sum()
Out[63]: 0

In [64]: day.isnull().sum().sum()
Out[64]: 0
```

The following data is the first 5 rows in hour.csv and day.csv respectively.

```
hour.head()
```

	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

```
day.head()
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

We can start with wrangling with day.csv, the season, month, holiday and workingday columns are well organized.

Temperature is a common sense factor, and it is normalized by 41, so, we need to multiply the temp value by 41 for better data visualization.

The cnt column is better to categorize into range, that is, 0-500, 501-1000 … .... 5501-6000, and 6000+ .

As the data type of dteday is string, we need to convert the entire column to date type:

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt	cnt_range
1	2011-01-01 00:00:00	1	0	1	0	6	0	2	14.110847	0.363625	0.805833	0.160446	331	654	985	501-1000
2	2011-01-02 00:00:00	1	0	1	0	0	0	2	14.902598	0.353739	0.696087	0.248539	131	670	801	501-1000
3	2011-01-03 00:00:00	1	0	1	0	1	1	1	8.050924	0.189405	0.437273	0.248309	120	1229	1349	1001-1500
4	2011-01-04 00:00:00	1	0	1	0	2	1	1	8.200000	0.212122	0.590435	0.160296	108	1454	1562	1501-2000
5	2011-01-05 00:00:00	1	0	1	0	3	1	1	9.305237	0.229270	0.436957	0.186900	82	1518	1600	1501-2000

Now, we need to check if there is repeated dates or missing dates, the code below shows that there is no repeated dates or missing dates.

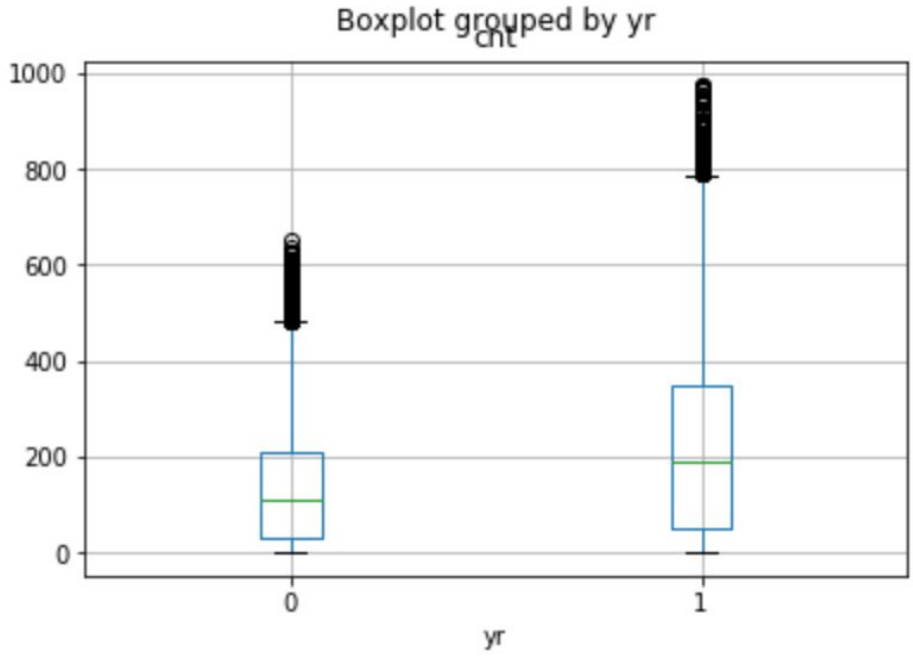
```
day.dteday.nunique()
```

```
731
```

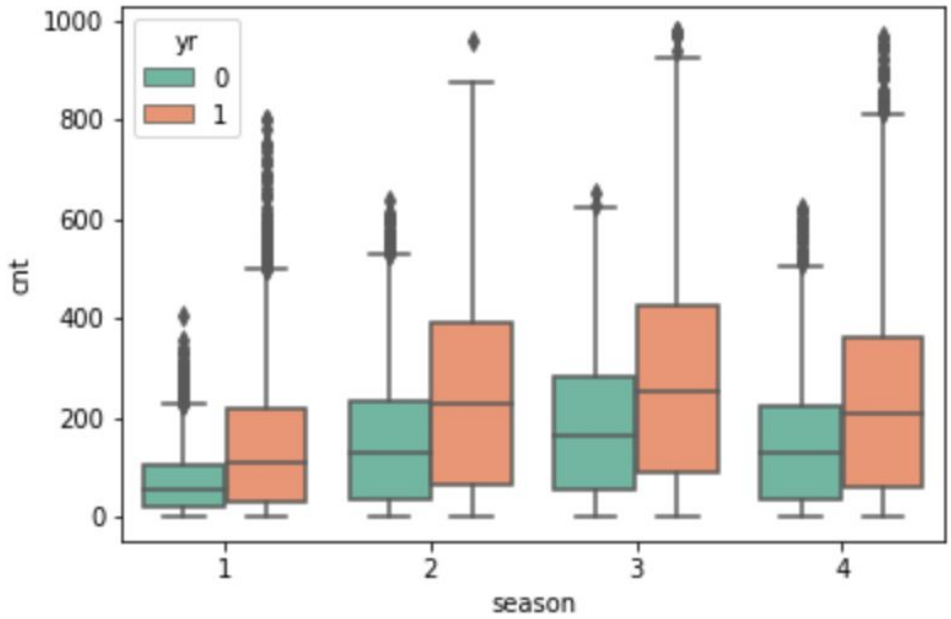
```
len(day)
```

```
731
```

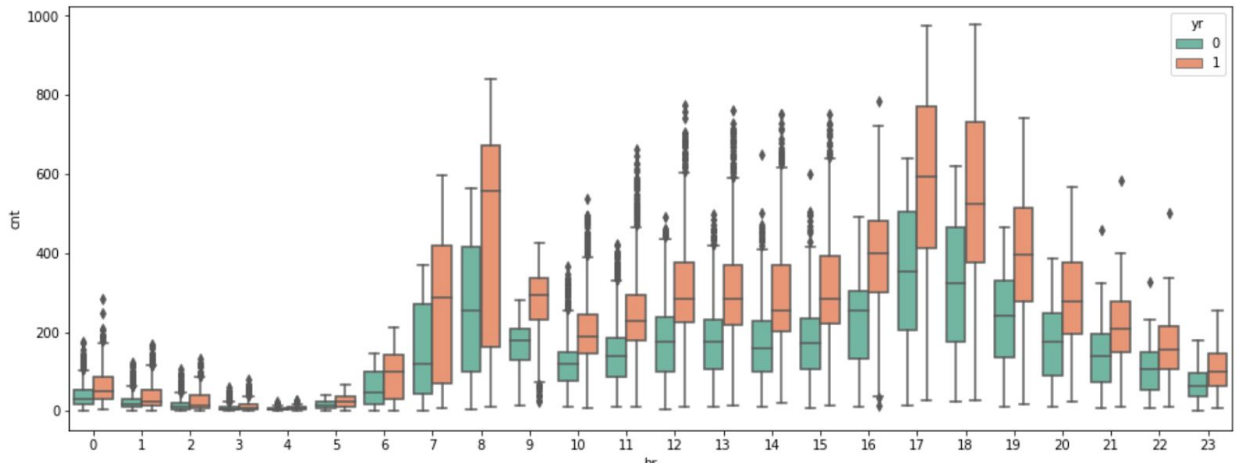
Data Storytelling



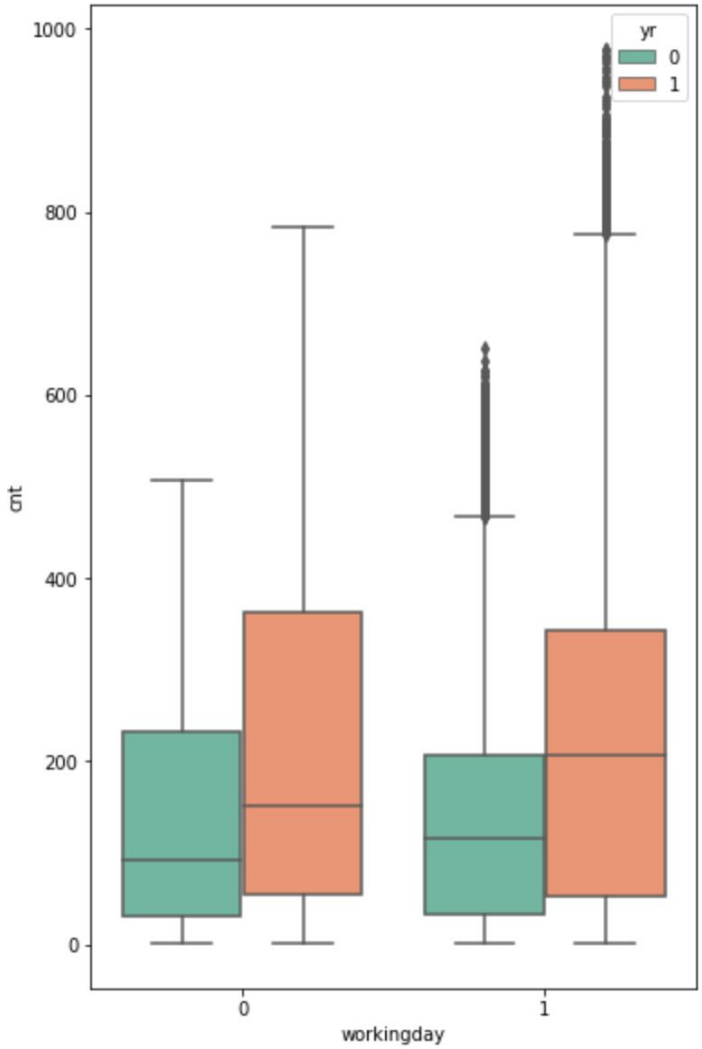
Year 2012 has a big increase in counts compared to 2011



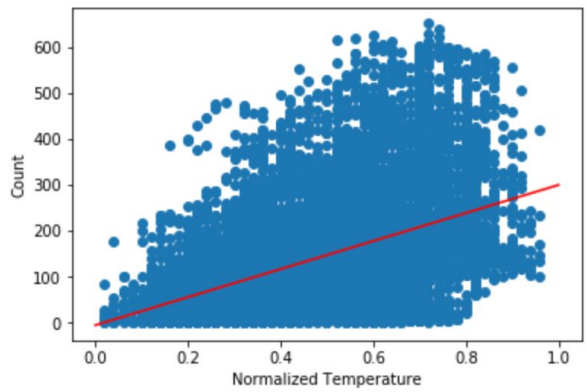
The Spring season has got relatively lower count, Summer and Fall have relatively higher count The median values in the box plot give evidence for it.

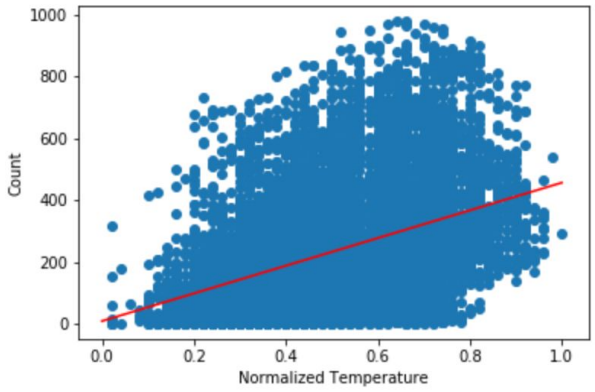


The median values are relatively higher at 7AM - 8AM and 5PM - 6PM. The regular school and work users contributed that part

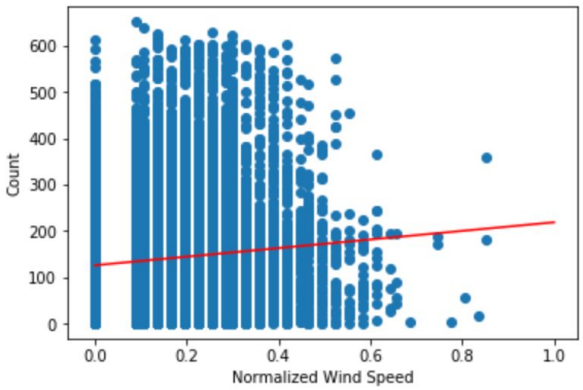


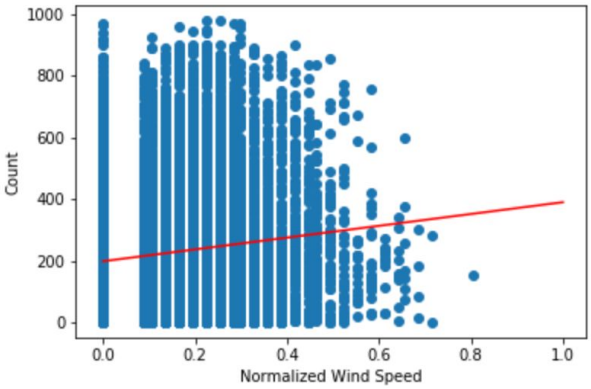
It is very obvious that all the outlier points are contributed by the Working Day.



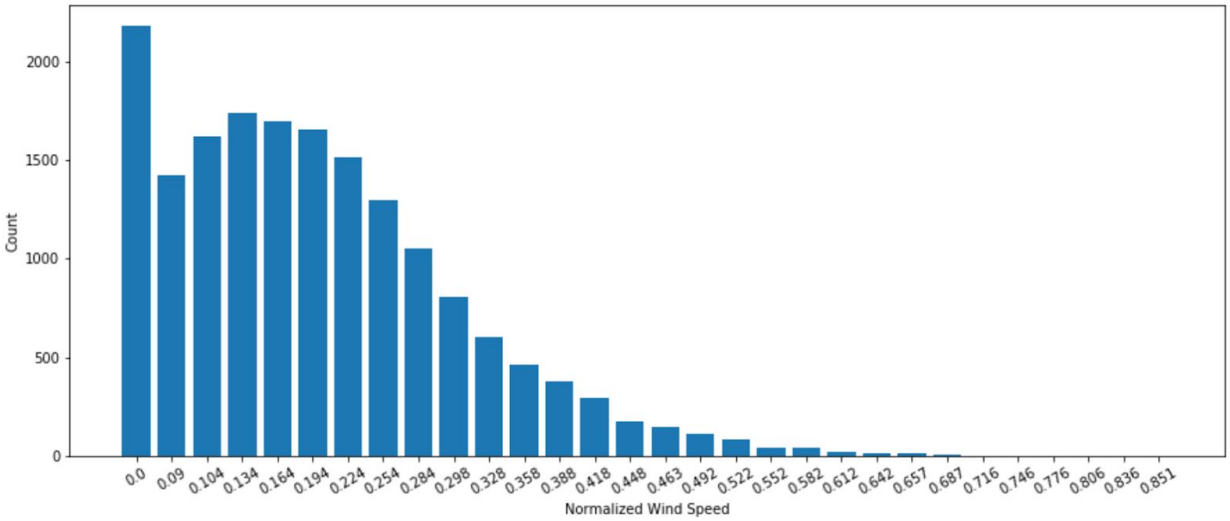


The linear Regression



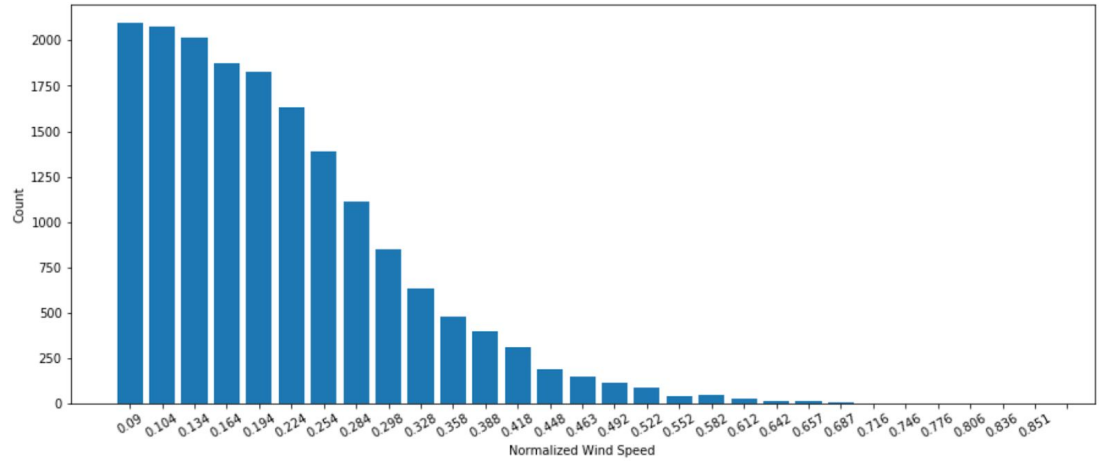


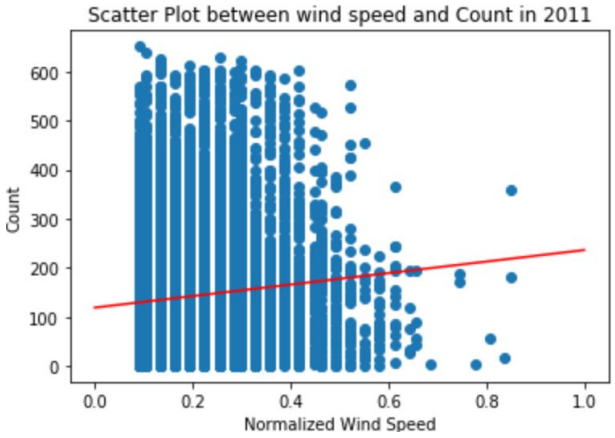
It is very odd that there are huge amount count when the wind speed is zero. And by common sense, it's nearly impossible. The histogram plot below shows that there are over 2000 counts when wind speed is 0. So in this case, I consider this as missing value.

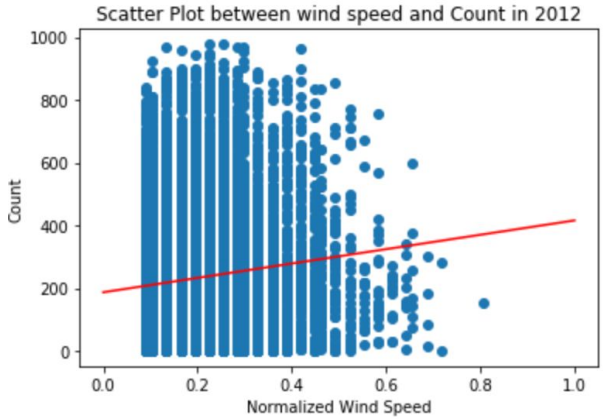


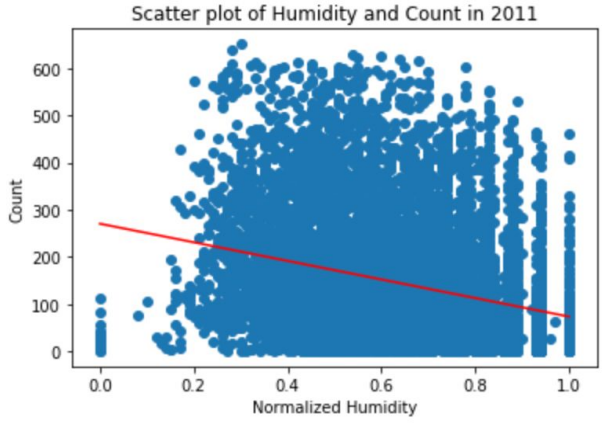
By applying Random Forest Classifier from sklearn package, based on the existing values of season, weathersit, hum, mnth, temp, yr, atemp from non- zero wind speed, the Random Forest

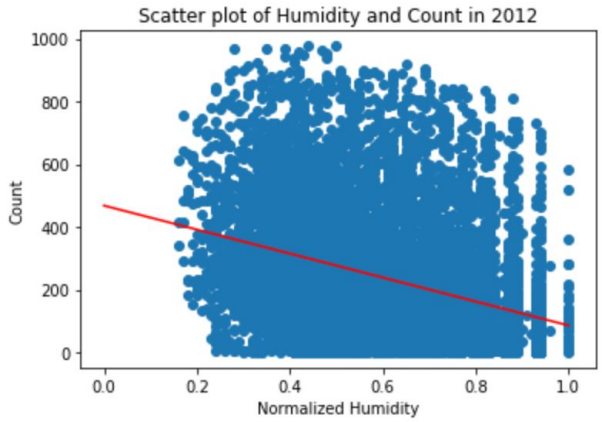
Classifier function will estimate the actual wind speed for the zero wind speed. The following histogram plot illustrates the new wind speed distribution, and it is very obvious that it is normally distributed.





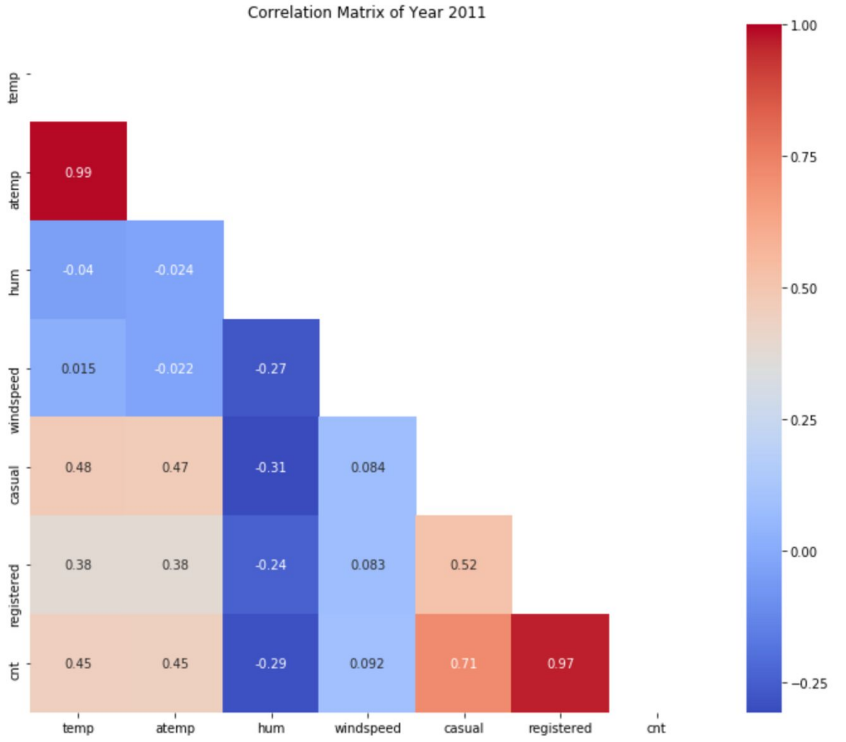


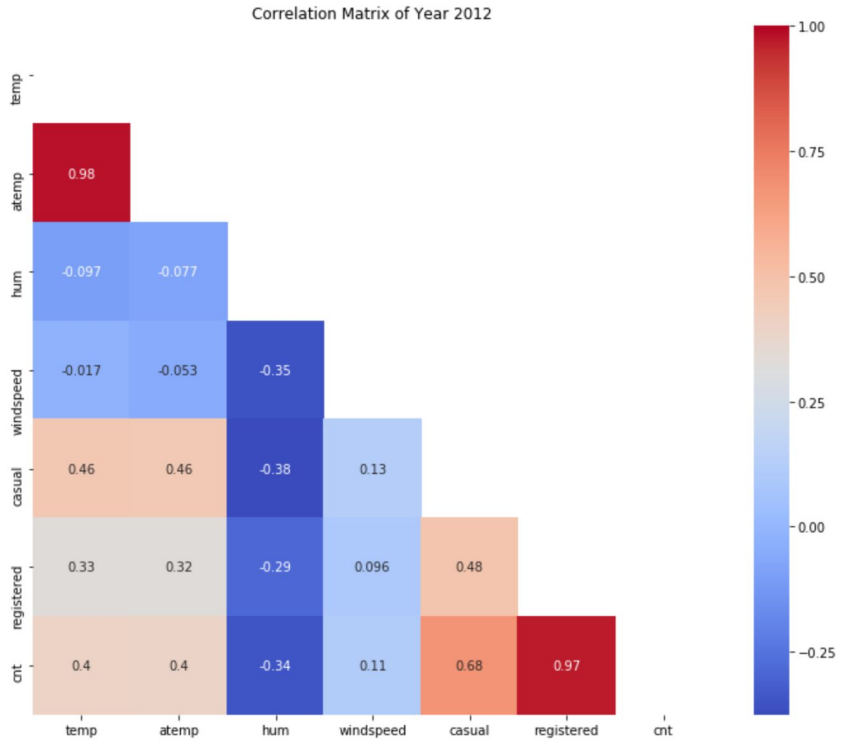


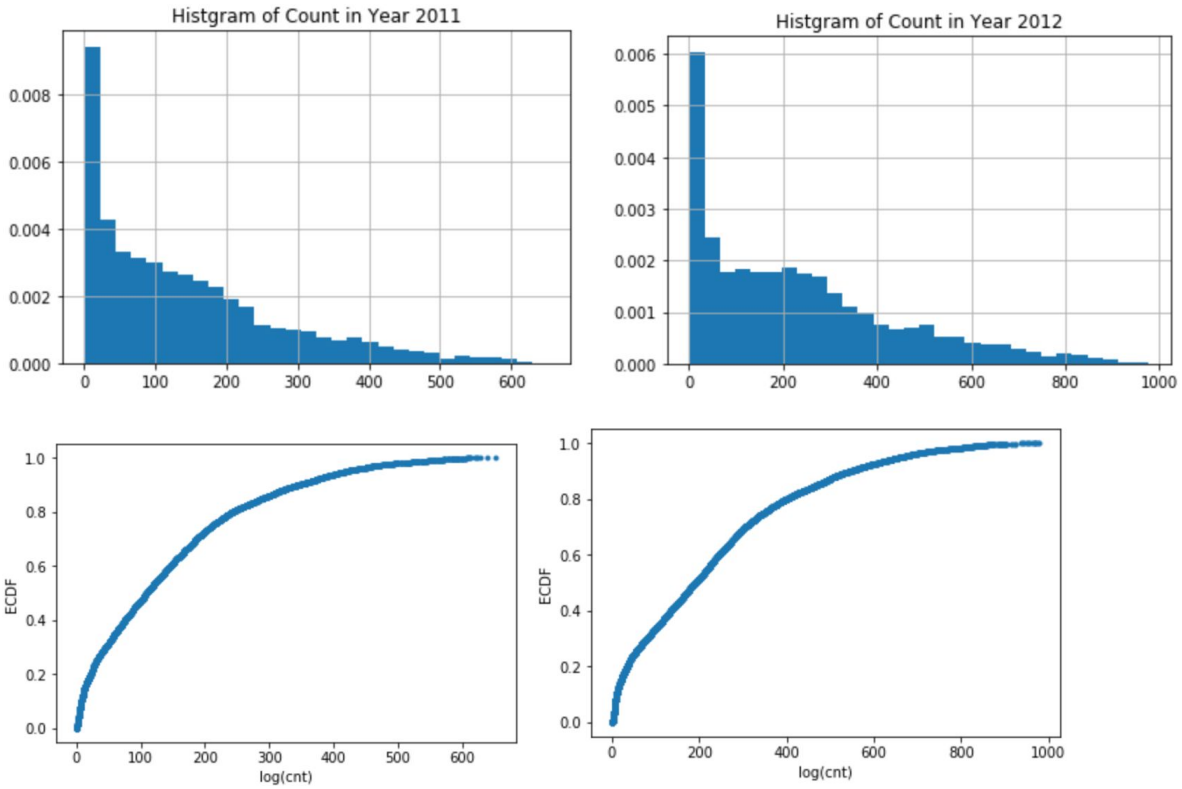


Regression plot in seaborn library in python is one useful way to depict the relationship between two features of dataset. But in this dataset, it is obvious that temperature, wind speed and humidity have small correlation to count from the scatter plots above.

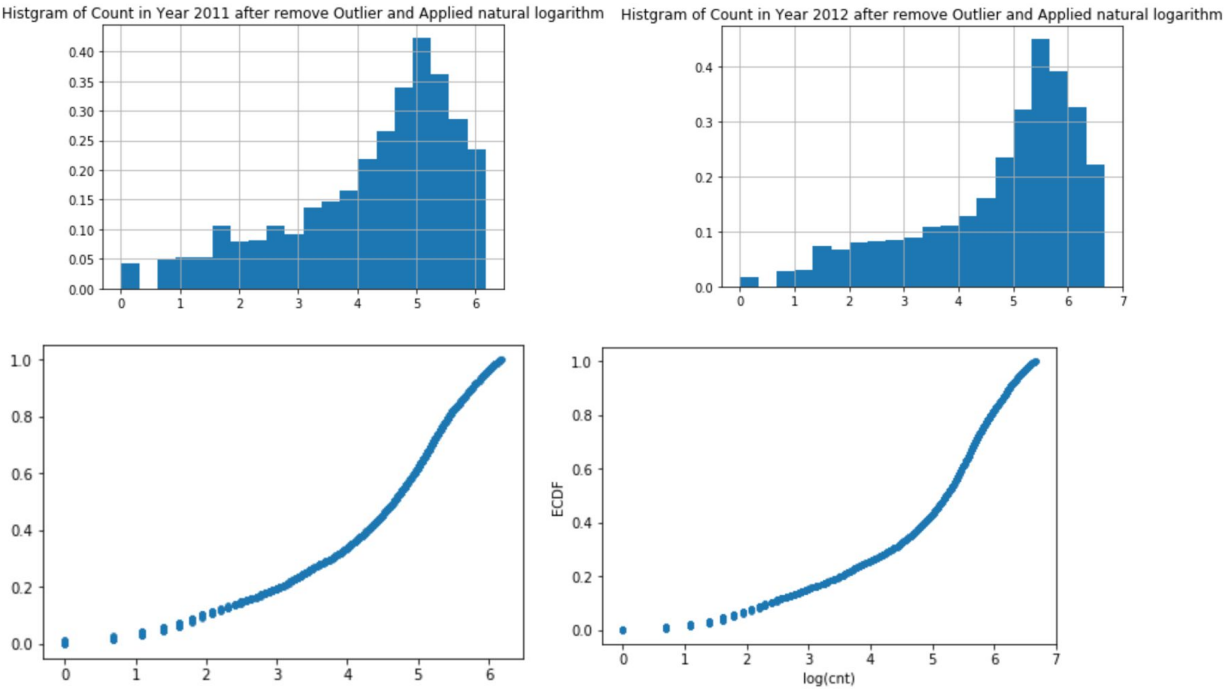
The following correlation matrix illustrated the correlation relationship between variables. As shown, the temperature and feeling temperature are highly correlated.



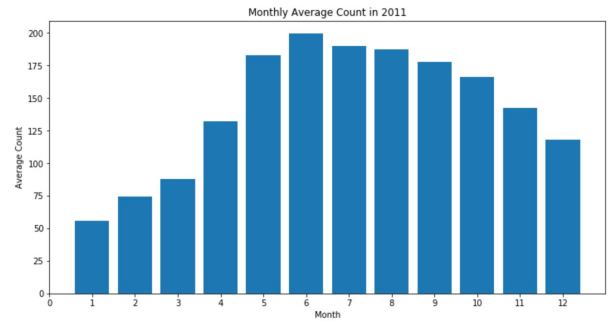


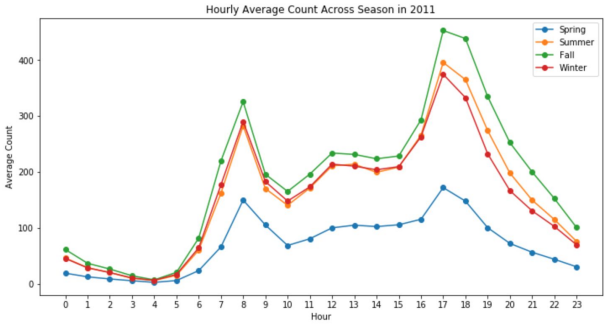


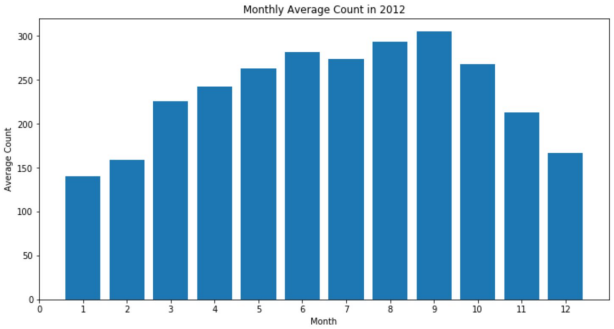
The histogram plots shown above are right skewed, and the ECDF plots are not even close to normal distribution. Iit is better to transform it to normal distribution or close to normal distribution.

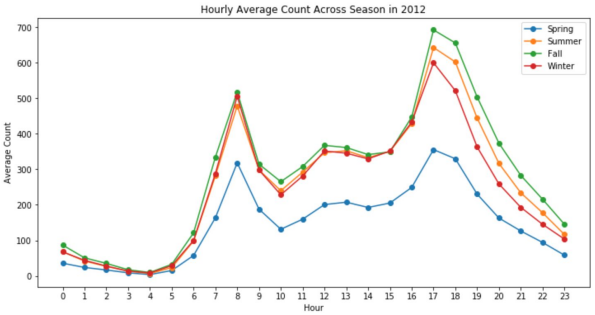


After dropping the outliers, then applying the natural logarithm, the histogram plots and ECDF plots are much closer to normal distribution.

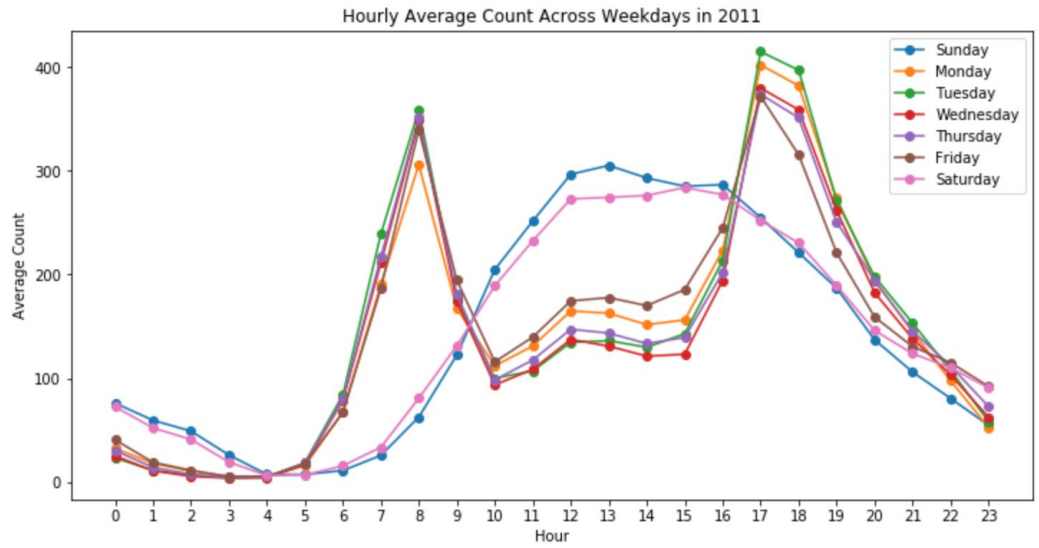


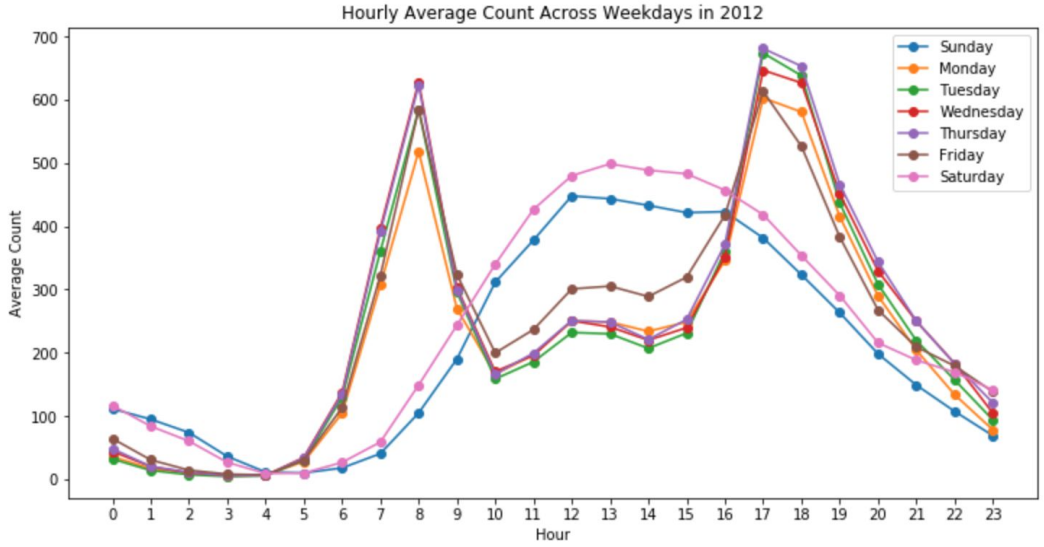




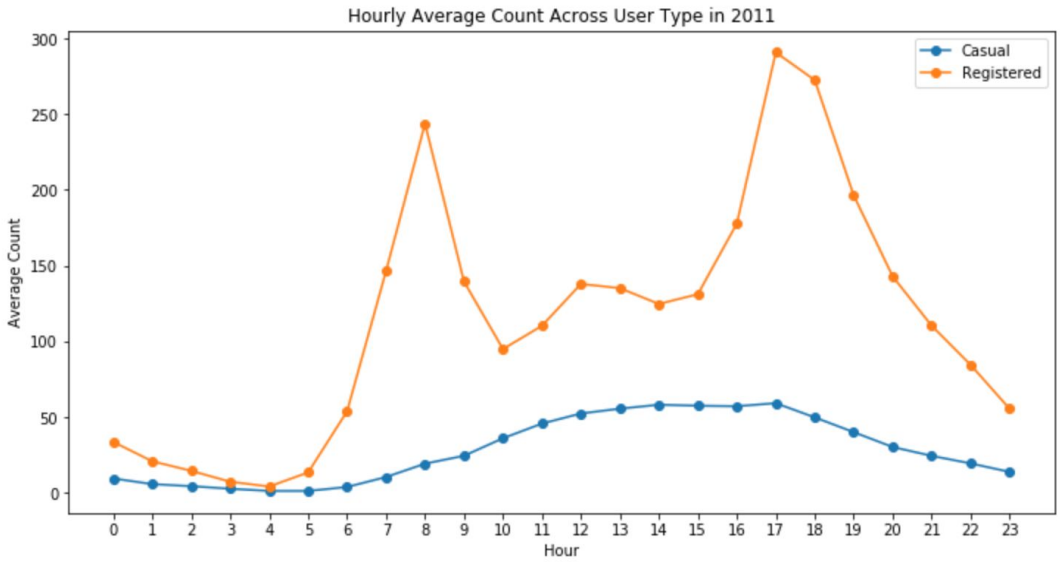


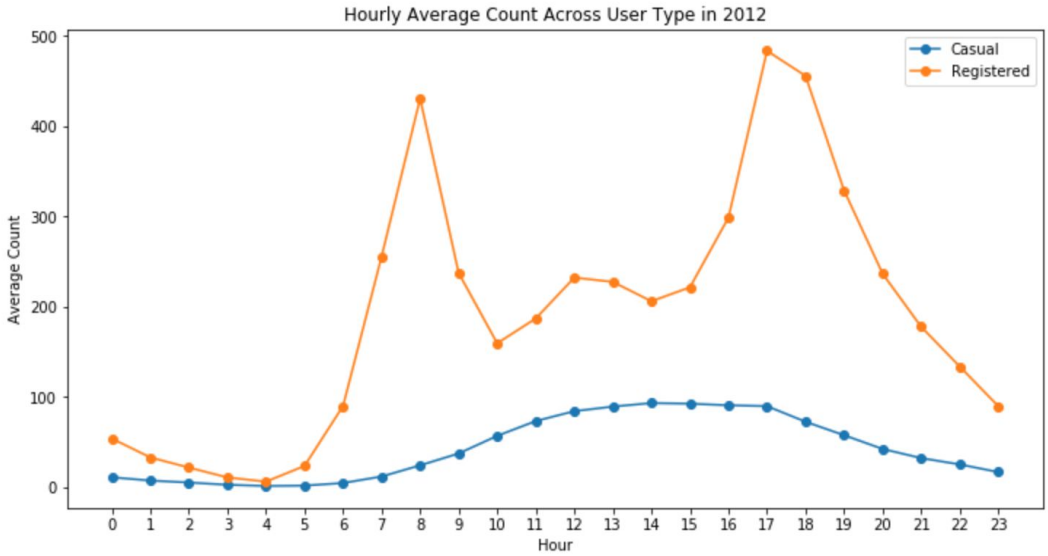
From the hourly average count across the year and seasons plots, It is obvious to tell that people tend to rent a bike during Summer and Fall since it is really conducive and enjoyable to ride bike in those seasons. Therefore May, June, July, August,September and October has relatively higher demand for bicycle.





From the Hourly average count across weekdays plots. It's telling me that, on weekdays, more people tend to rent a bicycle around 7AM-8AM and 5PM-6PM. As I mentioned earlier this can be attributed to regular school and work commuters. On weekends, the demand trend is very close to normal distribution, and more people tend to ride a bike between 10AM - 4PM





From the average count across user type plots, It is obvious that the peak user count around *7AM-8AM* and *5PM-6PM* is purely contributed by registered user.

Hypothesis Generation

- **Hourly trend**: There must be high demand during peak commuting hours, 7AM - 8AM and 5PM - 6PM, and low demand during 10:00 pm to 4:00 am.
- **Daily Trend:** Registered users demand more bike on weekdays as compared to weekends or holidays.
- **Weather:** The demand of bikes will be lower on a **rainy** day as compared to a sunny day. Rainy days usually have high **humidity**, therefore high humidity will cause to lower the demand and vice versa.
- **Temperature:** Temperature has a positive influence with demand, It is hard to ride a bike at the lower temperatures
- **Type of user:** Total demand should have higher contribution of registered user as compared to casual because registered user base would increase year by year.

Statistical Data Analysis

- Are there variables that are particularly significant in terms of explaining the answer to your question?

  Temperature and wind speed are the most important independent variables in my project, these 2 variables are seasonal indicators, Summer and Fall have higher temperature than in Spring and Winter, therefore higher demand in the Summer and Fall.

- What are the most appropriate tests to use to analyze these relationships?

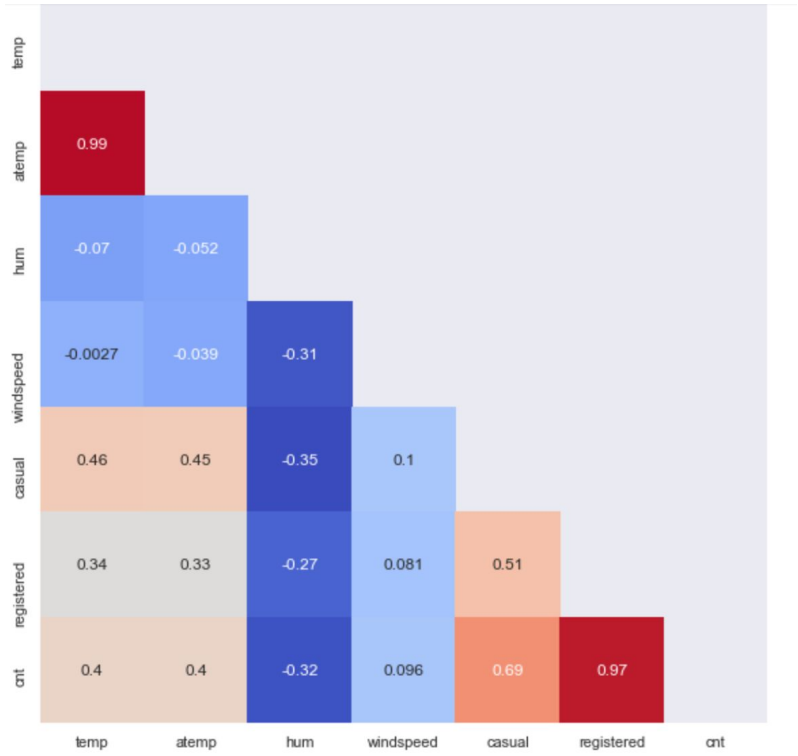
  Frequentist inference is the most appropriate test to use to analyze the relationship.

  Using the idea of continuous variable, to demonstrate the relationship between independent variables and dependent variable(user count)

- Are there significant differences between subgroups in your data that may be relevant to your project aim?

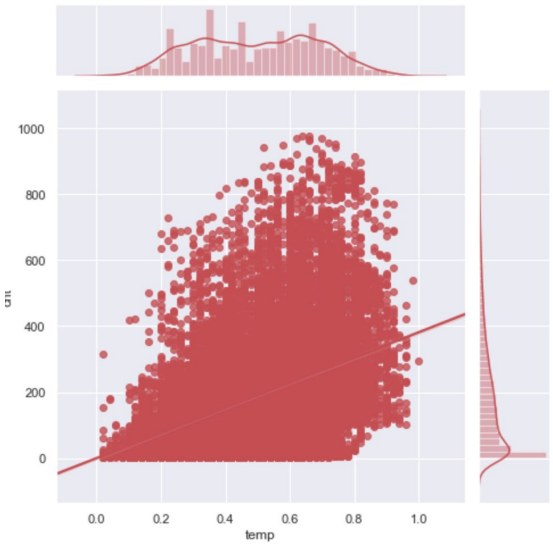
  The demand in year 2012 is significantly higher than that in year 2011

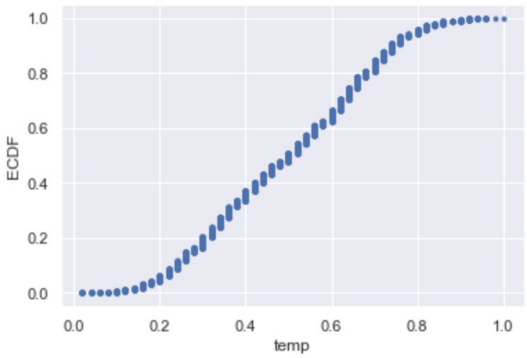
- Are there strong correlations between pairs of independent variables or between an independent and a dependent variable?

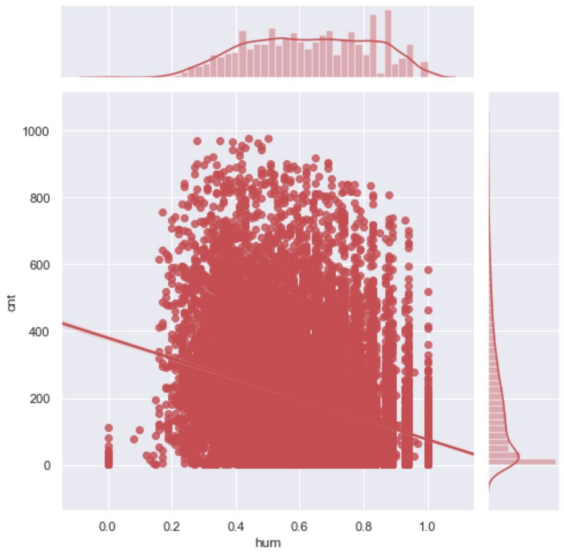


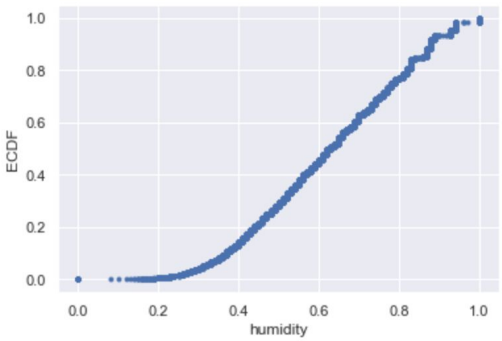
Temperature and humidity have strong correlations with user count

Temperature has a stronger correlation with casual user than with registered user, therefore membership promotion is suggested in good weather so that higher demand will be seen in the future as the majority of the user counts are contributed by registered user.



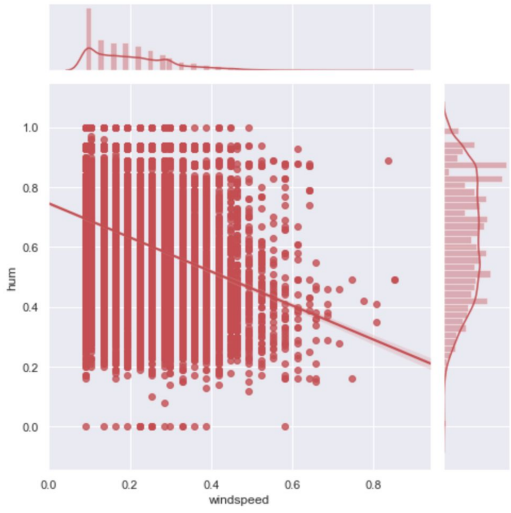


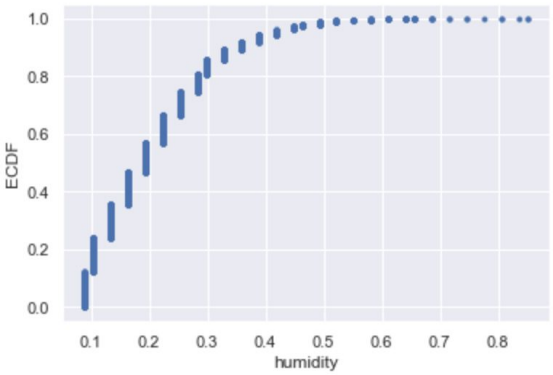




Wind speed and humidity have strong correlation with each other, this is common sense as higher wind speed makes water vapor in the air evaporates more quickly.

From the ECDF plot, it is clear to tell that wind speed is right half normal distribution.





In- Depth Analysis

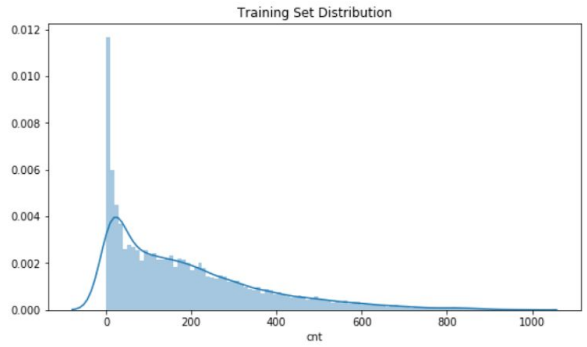
Linear Regression Model

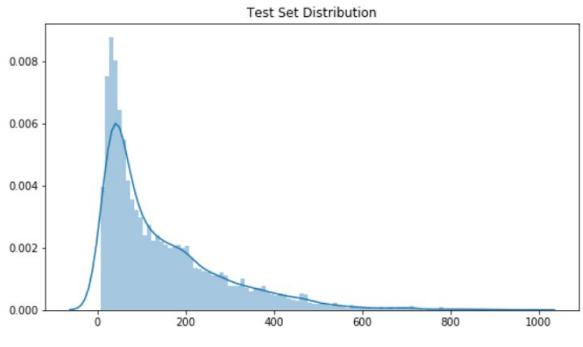
RMSLE (Root Mean Squared Logarithmic Error) is to evaluate the performance in predicting.

Visualizing Distribution Of Train And Test

By splitting the data into training data and testing data, the following plots illustrated the distribution of the training(left) and testing(right) data.

The RMSLE value for linear regression in validation is 0.991, this is not anywhere close to ideal.





Regularization

Regularization is extremely useful in any of the following cases. We do not face all the cases mentioned below but overfitting and multicollinearity may pose some issues for us.

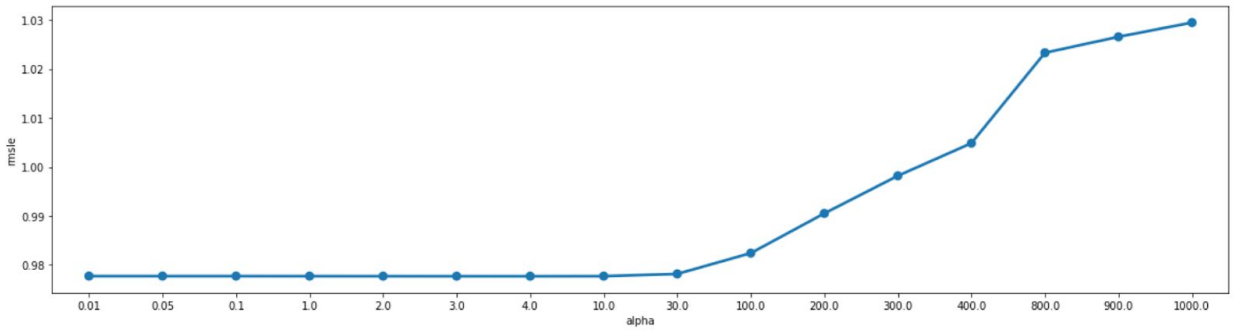
Overfitting.

Large number of variables.

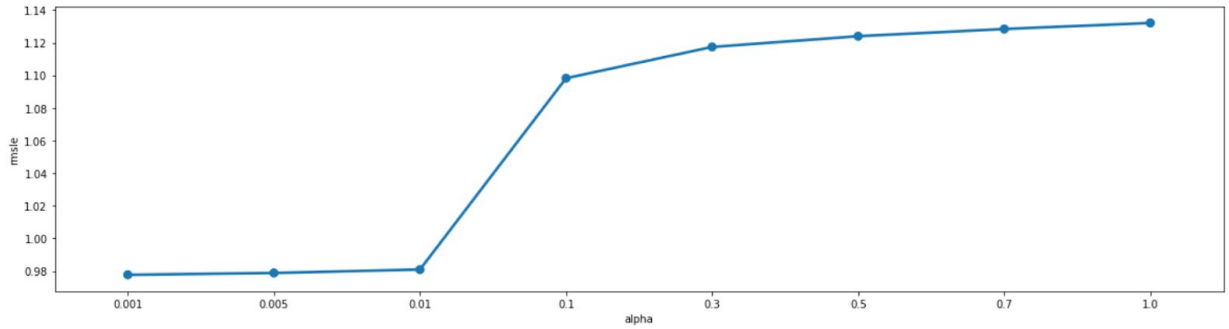
Low ratio of number of observations to the number of variables.

Multicollinearity.

L2 regularization (Ridge Regression) is extremely helpful for third case where we have more number of attributes than observation. But we are doing fine with that as of now since we have just 12 attributes and dataset has 10886 records. *Ridge Regression* is also quite useful when there is high *collinearity* between predictor variable. It may happen in our dataset since we have highly correlated variables such as temp-atemp and month-season.
Both Ridge and Lasso give RMSLE value of 0.99
Regularization Model - Ridge :



Having really large number of variable may again result in overfitting. This is because when we have more variables model becomes more complex and sometimes lowers its predicting and generalization power. L1 regularization (Lasso Regression) comes handy in these situations by reducing the coefficients to zero thereby producing simpler. Models.

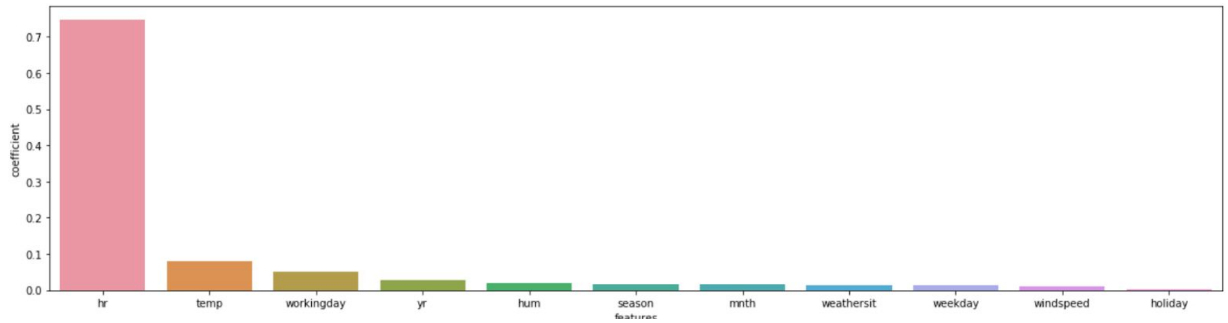


Ensemble Models - Random Forest

Ensemble models are nothing but art of combining diverse set of individual weak learners(models) together to improve stability and predictive capacity of the model. Ensemble Models improves the performance of the model by Averaging out biases.
Reducing the variance.
Avoiding overfitting.

The RMSLE value for Random Forest is 0.299, this is much better than the linear regression model.

Feature Importance By Random Forest

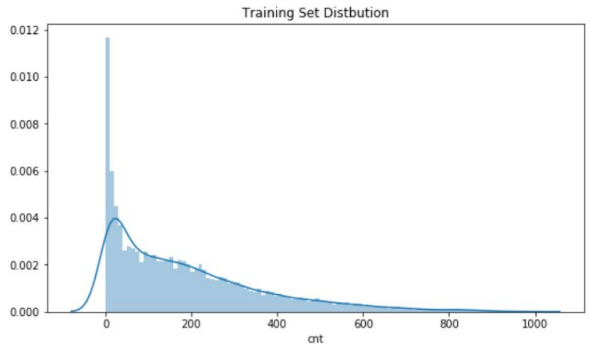


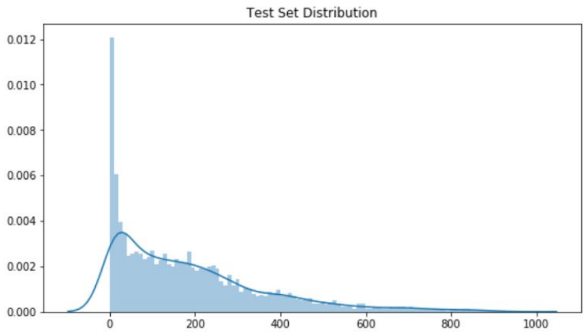
From the bar plot above, it is obvious that time of the day is the most important feature in the data set that influence the number of user count, temperature and working- day have relatively smaller influences than time of the day.

Ensemble Model - Gradient Boost

The RMSLE value for Random Forest is 0.273, this value has reached the top 10% percentile of the Kaggle scoreboard.

By using the Gradient Boost model to make the prediction with testing data, the comparison between training data and predicted data. The distribution of the predicted result is almost identical to the training data.





Conclusion

The Bike sharing demand prediction from the Kaggle competition is especially helpful for beginners in the data science world. It is a fairly simple dataset suitable for applying some concrete statistical techniques like Regression and also for some advance ensemble models such as Random Forest and Gradient Boosting algorithms.
The first capstone project ended with a fairly small RMSLE value of 0.273 with Gradient Boost model.