



Bike Sharing Program

07.09.2018

Rashi Nigam
Springboard Data Science Track
Fremont, CA 94539



Overview

A **bicycle-sharing system**, is a service in which **bicycles** are made available for shared use to individuals on a short-term basis for a price or free. Many bike share systems allow people to borrow a bike from a "dock" and return it at another dock belong to the same system.

Why the Need

- Increase personal mobility, providing people with better access to destinations throughout the City
- Integrate bike share as an extension of public transit network
- Develop an innovative transportation system that improves livability and economic competitiveness
- Reduce the environmental impact of transportation and help achieve goal of 'Go Green'
- Develop a system that serves users in minority and low-income communities and improves their access to key destinations, such as jobs and recreation
- safe mode of transportation that promotes active and healthy living
- Create a system that is financially sustainable, transparently operated, and accountable to the public.

History of Bike sharing systems go way back since year 1965 in Amsterdam however most of the major North American Systems started around 2010 and Capital Bikeshare - Washington, DC and Arlington, VA (1,600 bikes/191 stations) is among the most active systems. It is as well the client of this project. Capital Bikeshare has grown steadily, which has driven demand for more stations and bikes. Thus Client needed a research analysis to optimize their service & operations that

- Predict the Bike Rental volume/count
- Factors or features that influence Bike Rental Count

Goals

Predict the Bike Rental volume from the dataset given by Capital Bike Sharing System.

Determine the factors or features that influence Bike Rental Count most.

The client is Capital Bikeshare System and this research to predict bike rental count will be useful to them in knowing:

- What features in the dataset influence the bike rental count
- When is the demand for bike share program maximum during the day, season, quarter or year.
- Does weather conditions like temperature, humidity, windspeed have any impact on the demand? If yes, then is it to advantage or adverse.
- Are bike users whether Registered or Casual drive Bike Rental Count? If yes, do they have similar influence on Bike Rental Count distribution?

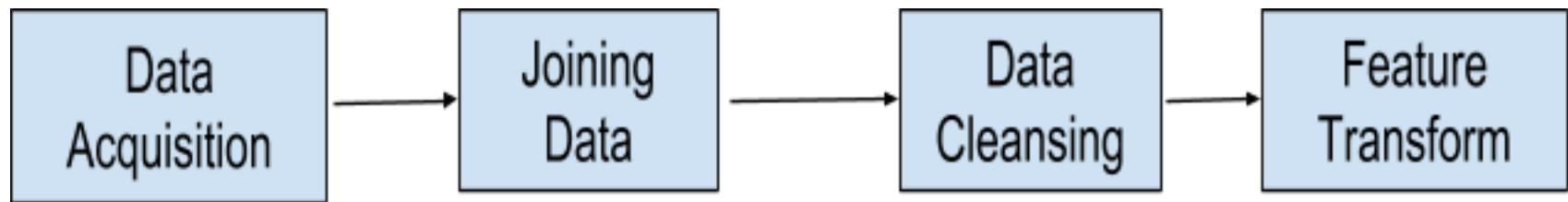
Data Wrangling

Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis.

Goals of Data Wrangling are:

- Reveal a “deeper intelligence” within your data, by gathering data from multiple sources
- Provide accurate, actionable data in the hands of business analysts in a timely matter
- Reduce the time spent collecting and organizing unruly data before it can be utilized
- Enable data scientists and analysts to focus on the analysis of data, rather than the wrangling
- Drive better decision-making skills by senior leaders in an organization

Key steps in Data Wrangling



Capital Bikeshare posts quarterly data reports of bike trip times, start and end locations, and type of user (registered or casual). Each trip is on one line of data. These data are readily and publicly available at <https://www.kaggle.com/c/bike-sharing-demand/data> and appear as below:

1	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
2	1	1/1/11	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0	3	13	16
3	2	1/1/11	1	0	1	1	0	6	0	1	0.22	0.2727	0.8	0	8	32	40
4	3	1/1/11	1	0	1	2	0	6	0	1	0.22	0.2727	0.8	0	5	27	32
5	4	1/1/11	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0	3	10	13
6	5	1/1/11	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0	0	1	1
7	6	1/1/11	1	0	1	5	0	6	0	2	0.24	0.2576	0.75	0.0896	0	1	1
8	7	1/1/11	1	0	1	6	0	6	0	1	0.22	0.2727	0.8	0	2	0	2
9	8	1/1/11	1	0	1	7	0	6	0	1	0.22	0.2727	0.8	0	1	0	2

The Capital bikeshare datasets required data wrangling in terms of extracting dataset from Capital Bikeshare public shared repository followed by identifying meaningful dataset, renaming a few columns based on preference, feature transforming date timestamp to day, month, year; formatting the date and time columns to match with the weather data.

1. Renaming the columns as below:

```

'instant': 'rec_id',
'dteday':'datetime',
'holiday': 'is_holiday',
'workingday': 'is_workingday',
'weathersit': 'weather_condition',
'hum': 'humidity',
'mnths': 'month',
'cnt': 'total_count',
  
```




```
'hr': 'hour',
```

```
'yr': 'year'
```

2. There were not any missing values to drop or replace. Type casting the attributes as 'datetime' or 'category' shown below

```
stats['datetime'] = pd.to_datetime(stats.datetime)#dae time conversion
```

```
# categorical variables
```

```
stats['season'] = stats.season.astype('category')
```

```
stats['is_holiday'] = stats.is_holiday.astype('category')
```

```
stats['weekday'] = stats.weekday.astype('category')
```

```
stats['weather_condition'] = stats.weather_condition.astype('category')
```

```
stats['is_workingday'] = stats.is_workingday.astype('category')
```

```
stats['month'] = stats.month.astype('category')
```

```
stats['year'] = stats.year.astype('category')
```

```
stats['hour'] = stats.hour.astype('category')
```


Model Workflows

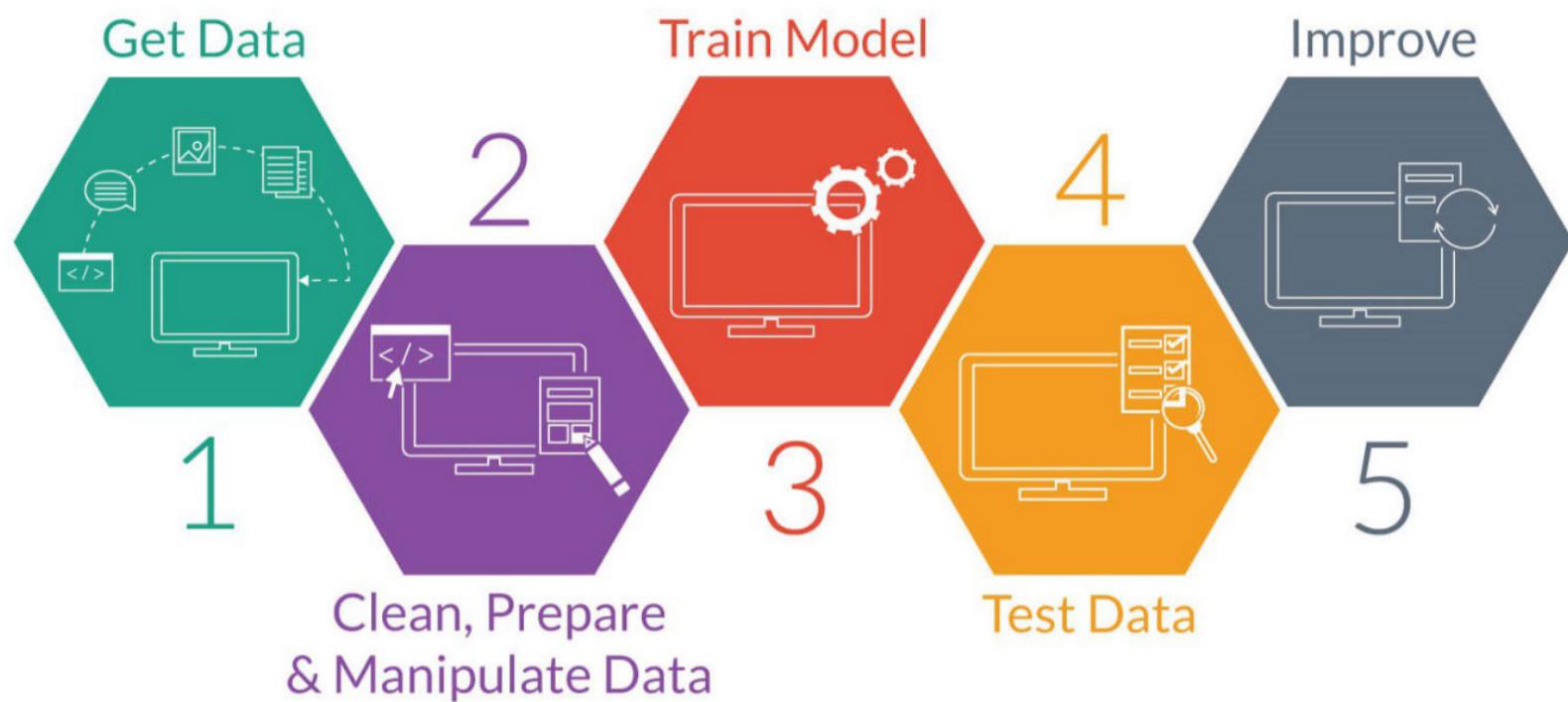


Image courtesy: <https://towardsdatascience.com/building-a-deployable-ml-classifier-in-python-46ba55e1d720>

Any Machine Learning model comprises two stages where Stage I constitutes:

- Get Data/ Data Extraction from authentic/authorized source
- Data Cleaning/Feature Transform/Exploratory Data Analysis
- Train Model using ML techniques after data partitioning into training dataset and Test dataset

Stage II , once training dataset is validated

- Test data is used for the same Machine Learning model
- Compare the results to verify model efficiency and suggest any improvements/recommendations

Exploratory Data Analysis/Inferential Statistics

Of the 16 features, following features showed correlation with the target labeled variable 'cnt' i.e. Bike Rental Count:

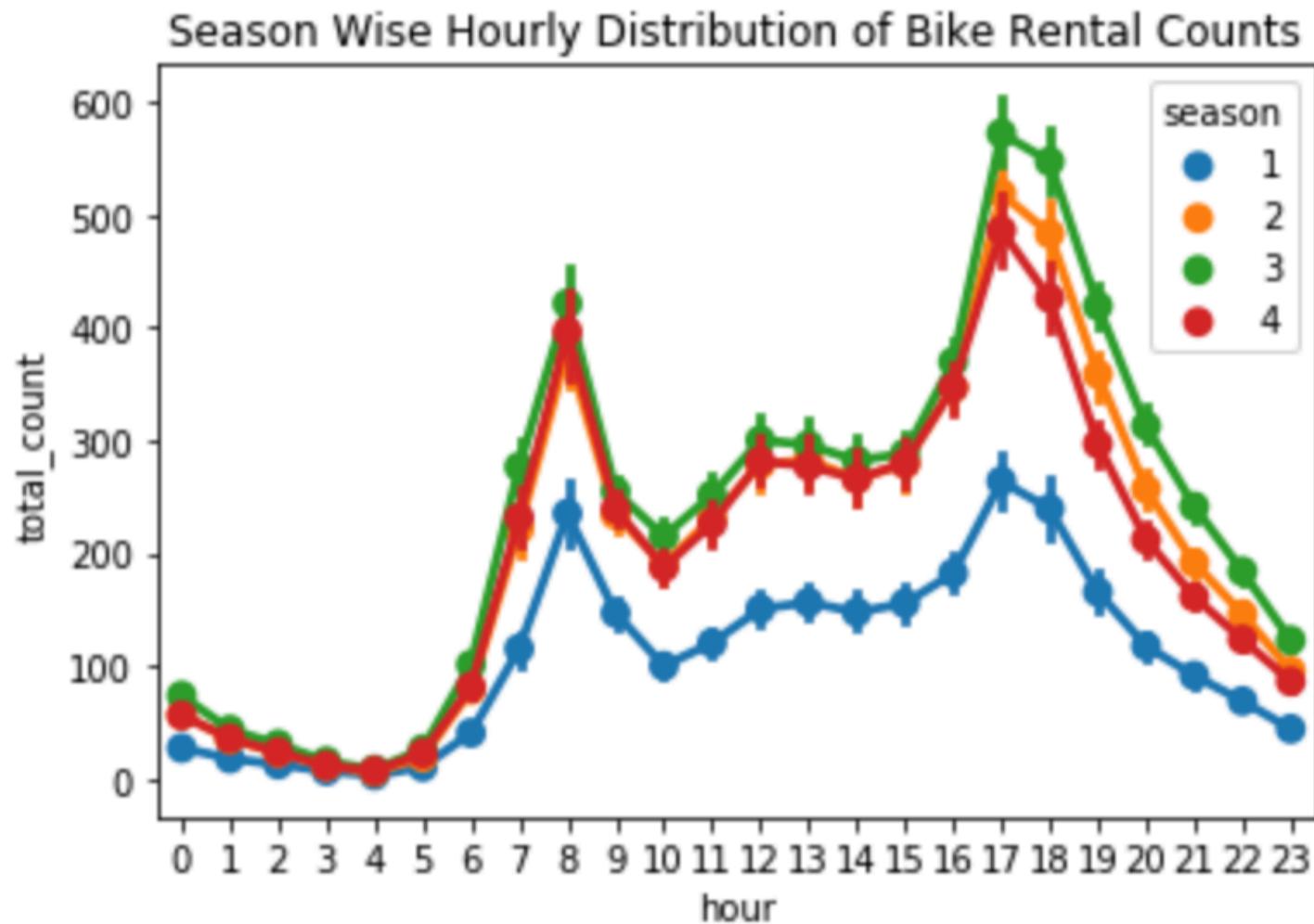
- Season
- Month
- Temperature (temp)
- Humidity (hum)
- Windspeed

- Casual
- Registered
- Hour (hr)

Above dependencies can be verified with following visualization graph plots and inferential statistics (code book attached too)

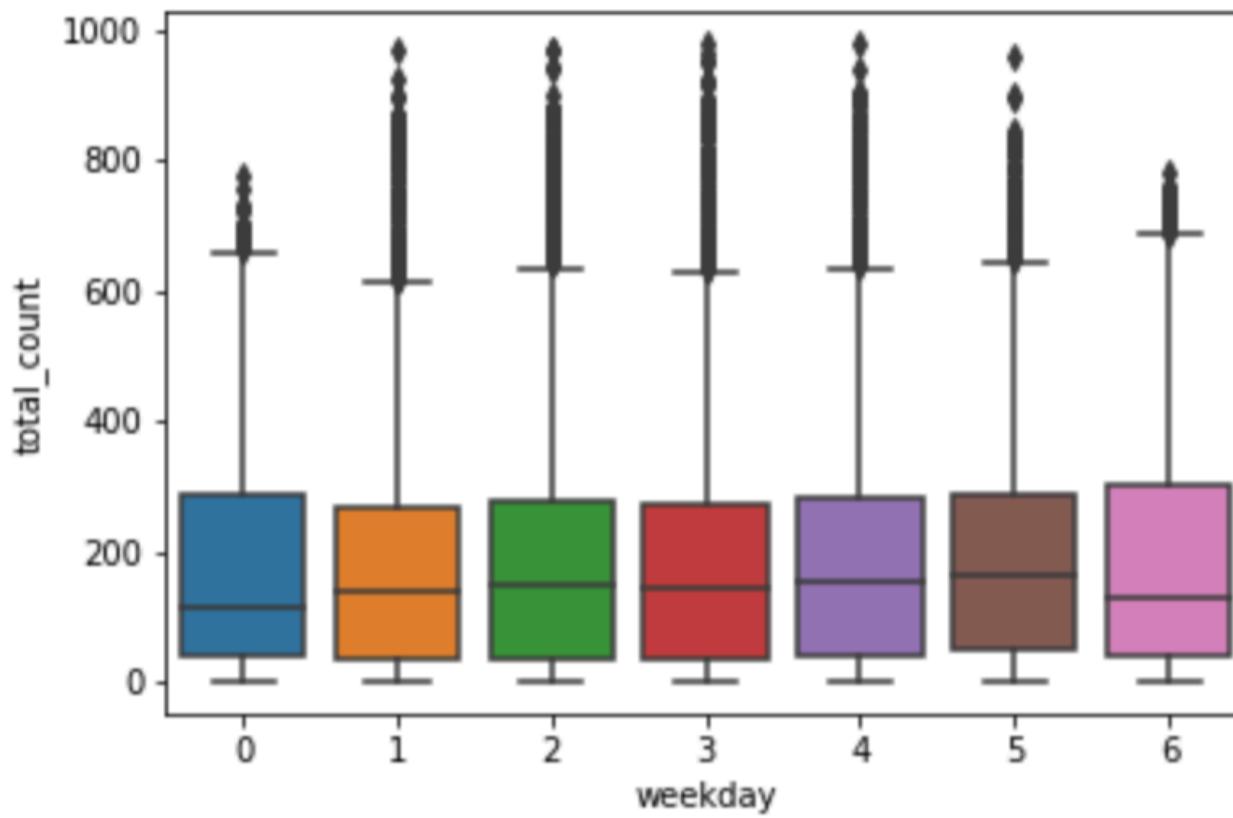
<https://github.com/rashi-n/Machine-Learning-Projects/blob/master/Capstone%20Projects/CS%20I%20-%20EDA%20%26%20Inferential%20Statistics.ipynb>:

1. Line Chart between Bike Rental Count vs. Hour across Seasons



Season 1 = Spring, 2 = Summer, 3= Fall, 4=Winter. Above graph shows similar trends for all seasons with counts peaking in the morning between 7 -9 AM and in the evening between 4-6 PM for the reason those are business hours. The counts are lowest for spring season (Legend 1) while highest for Fall (Legend 3) across 24 hours

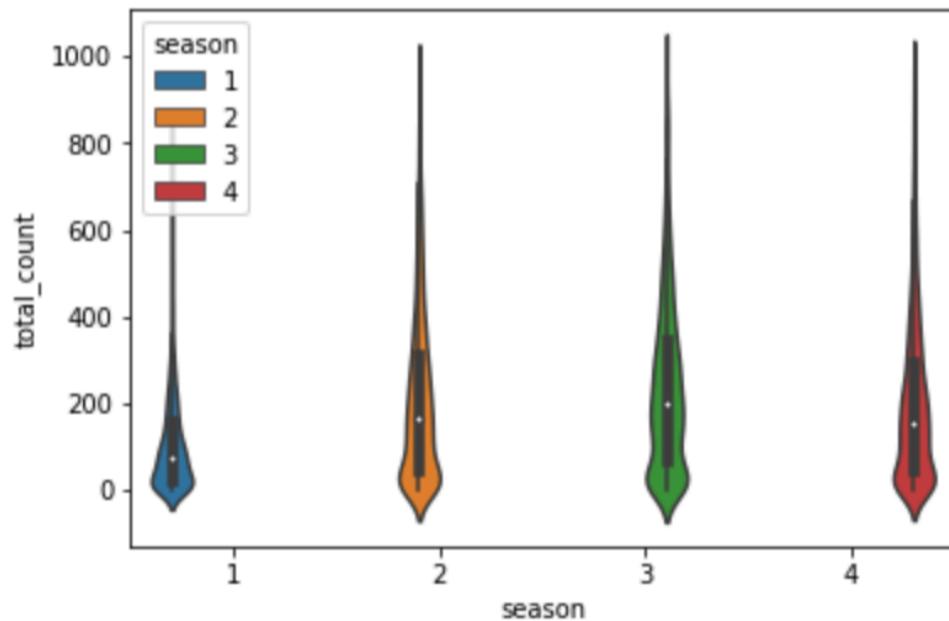
2. Box Plot between Bike Rental Count vs Weekday



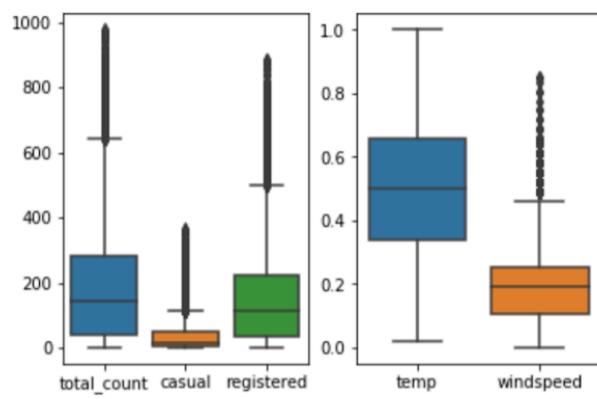
Weekday 0= Sunday, 1= Monday, 2= Tuesday & so on. During weekdays Mon -Fri, I see median of Bike Rental count is similar as opposed to weekends.

3. Violin Plot between Bike Rental Count vs Seasons

```
z = sn.violinplot(data=stats, x='season', y= 'total_count', hue = 'season')
```

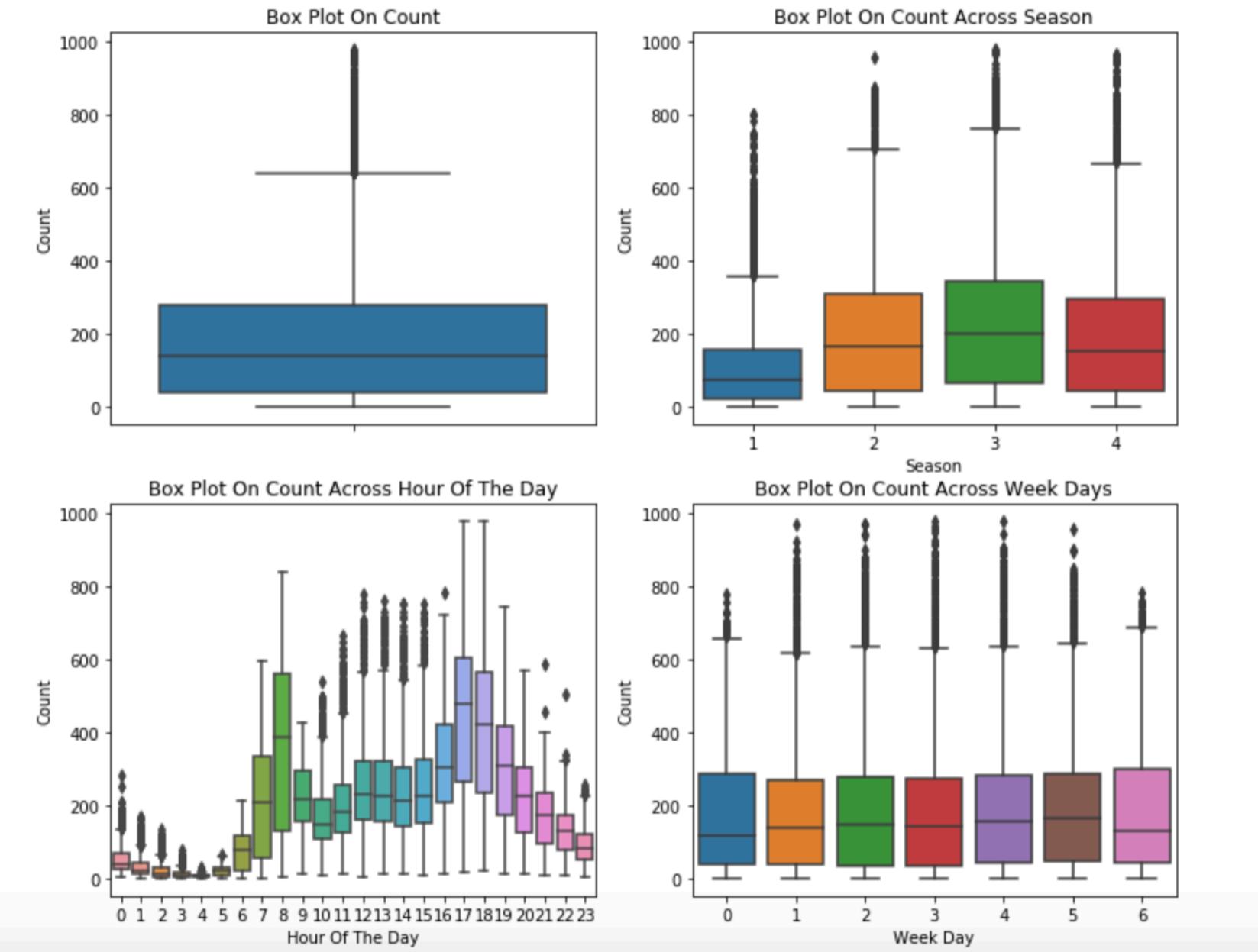


4. Box Plot between Bike Rental Count vs Casual, Registered Users



The total, casual & registered type users show sizeable number of outlier values, however casual show lower numbers though. For weather attributes of temperature and wind speed, we see outliers only in the case of windspeed.

5. Outlier Analysis



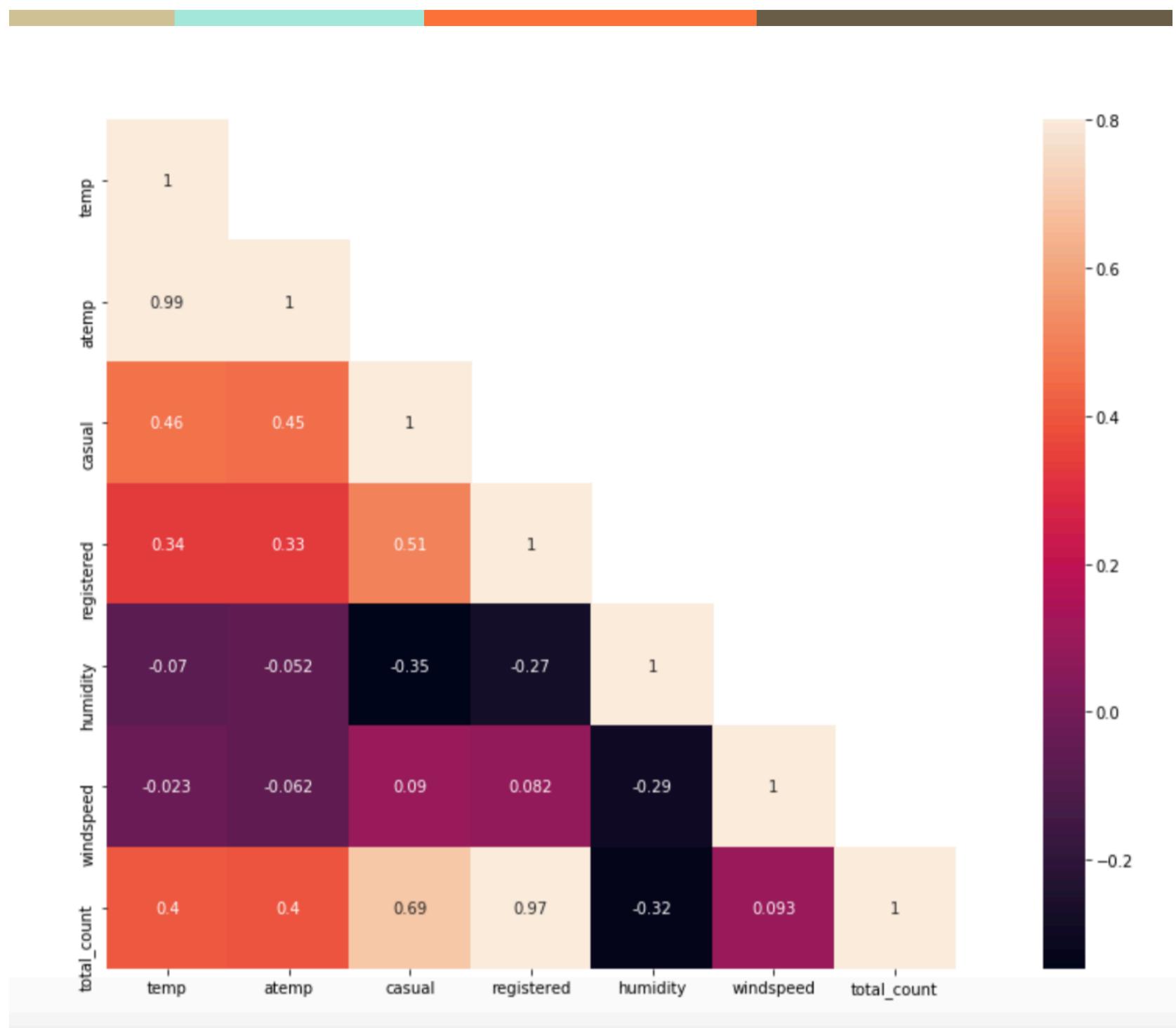
At first look, "count" variable contains several outlier data points which skew the distribution towards right (as there are more data points beyond Outer Quartile Limit). But in addition to that, following inferences can also be made from the simple boxplots given below.

Spring season has got relatively lower count. The dip in median value in boxplot gives evidence for it. The boxplot with "Hour Of The Day" is quiet interesting. The median values are relatively higher at 7AM - 8AM and 5PM - 6PM. It can be attributed to regular school and office users at that time. Most of the outlier points are mainly contributed from "Working Day" than "Non Working Day". It is quiet visible from figure 4.

6. Correlation plot between "count" and ["temp","atemp","humidity","windspeed"]

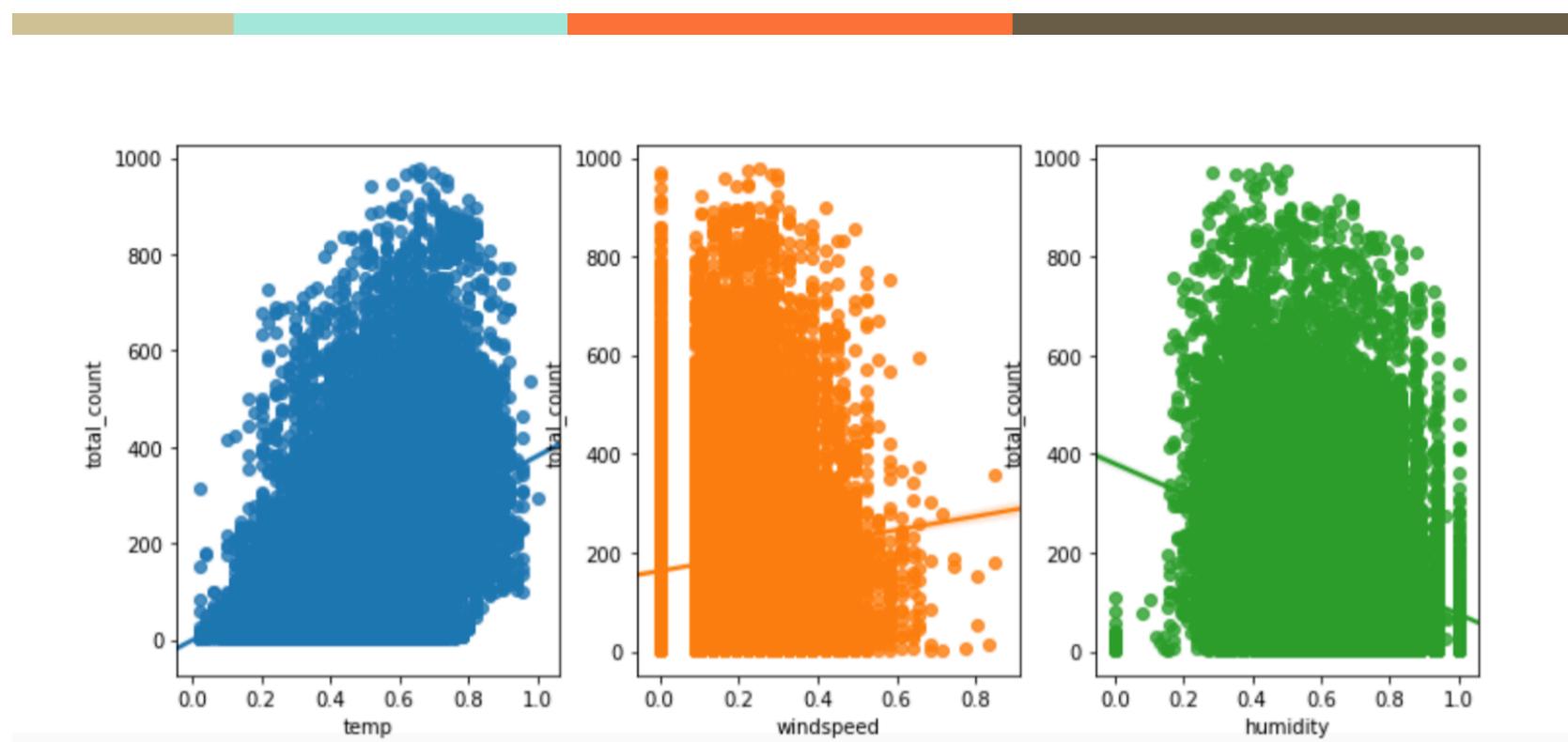
One common to understand how a dependent variable is influenced by features (numerical) is to find a correlation matrix between them. Let's plot a correlation plot between "count" and ["temp","atemp","humidity","windspeed"].

Features temp and humidity have got positive and negative correlation with count respectively. Although the correlation between them is not very prominent still the count variable has got little dependency on "temp" and "humidity". Feature 'windspeed' is not going to be a really useful numerical feature and it is visible from its correlation value with "count". Feature "atemp" is a variable not taken into account since "atemp" and "temp" has got strong correlation with each other. During model building any one of the variables must be dropped since they will exhibit multicollinearity in the data. "Casual" and "Registered" are also not taken into account since they are leakage variables in nature and need to be deleted during model building. Regression plot from seaborn is one useful way to depict the relationship between pairs of features. Here we consider "count" vs "temp", "humidity", "windspeed".



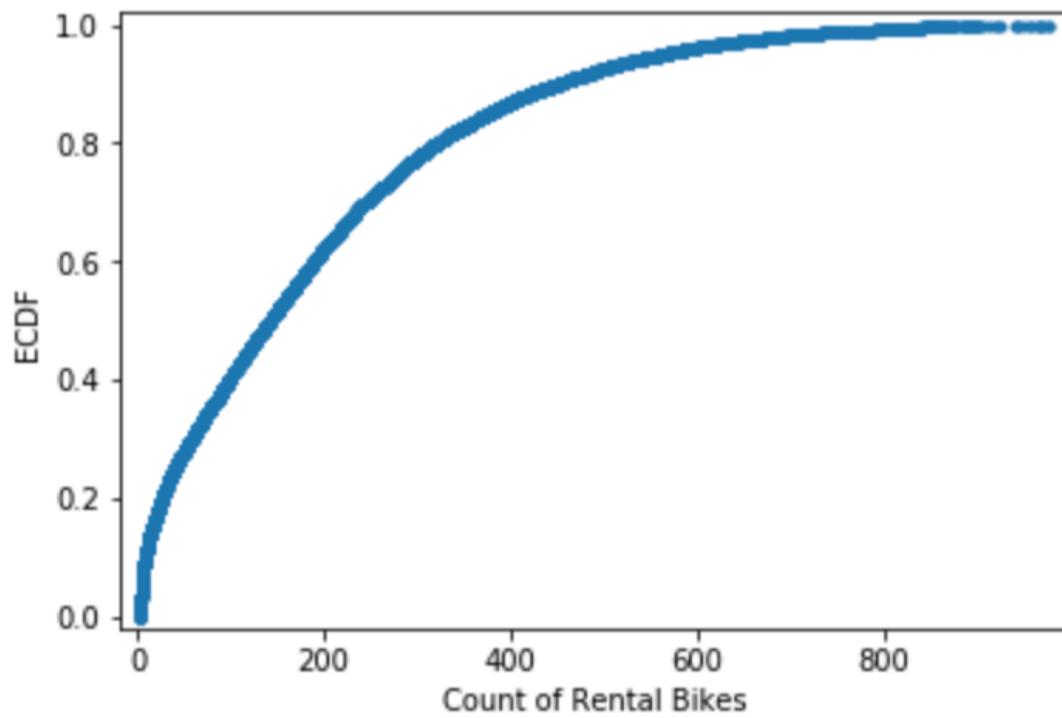
Correlation between Bike Rental Volume (total_count) and 'registered' user type is the highest. Followed by 'casual' user type. I will explore this dependency of Bike rental volume by User Type in Null Hypothesis under Inferential statistics coming later in EDA. There is moderate collinearity between 'total_count' and 'temp'(temperature)too.

7. Linear Regression plot between Bike Rental Count vs Temp, humidity, Windspeed



There is direct positive relation between Bike Rental volume(total_count) vs 'temp' while negative relation with 'windspeed'

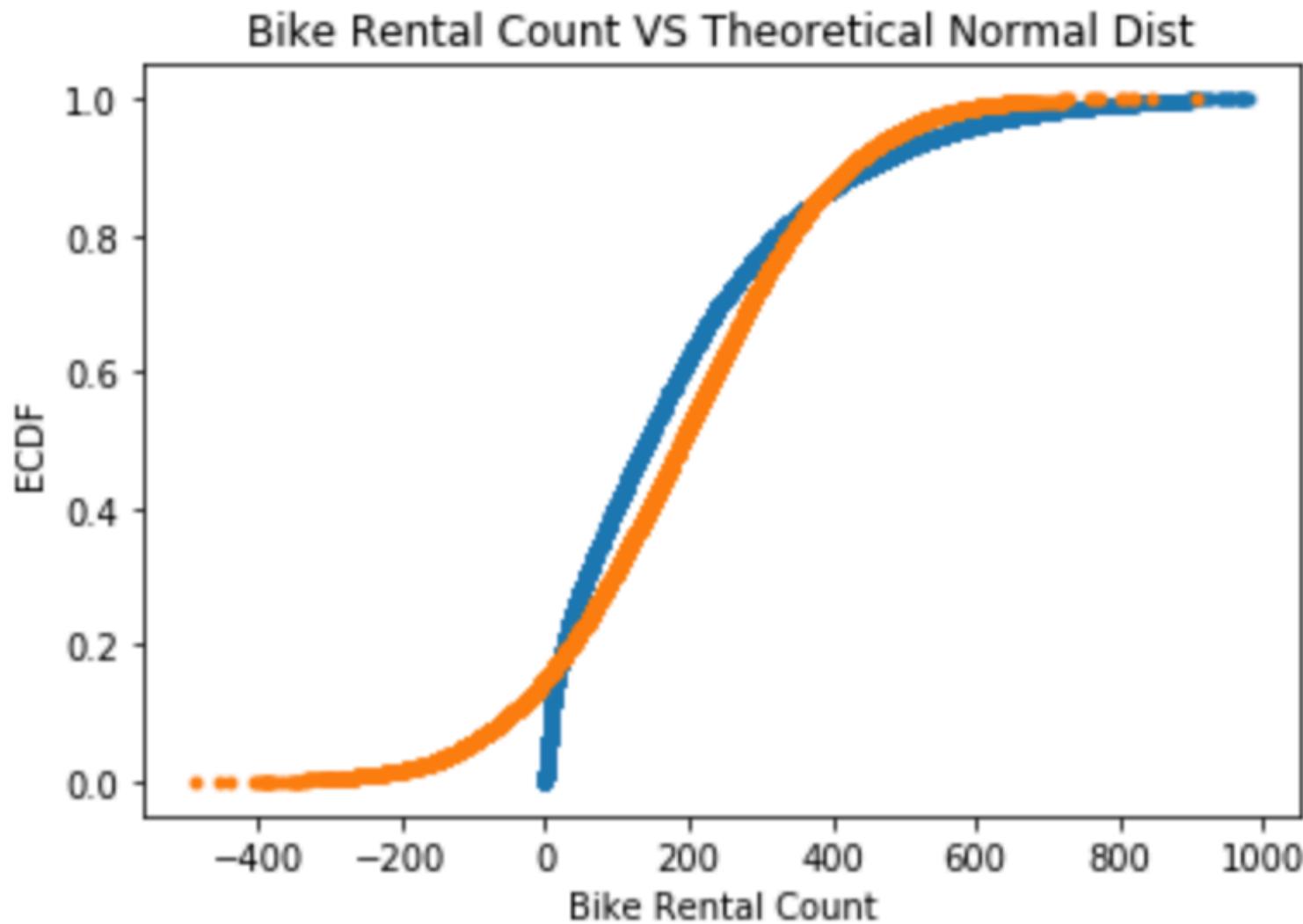
8. Empirical Continuous Distribution ECDF plot for Bike Rental Count



```
[33]: np.percentile(stats['total_count'], [25, 50, 75, 90, 98, 100])
```

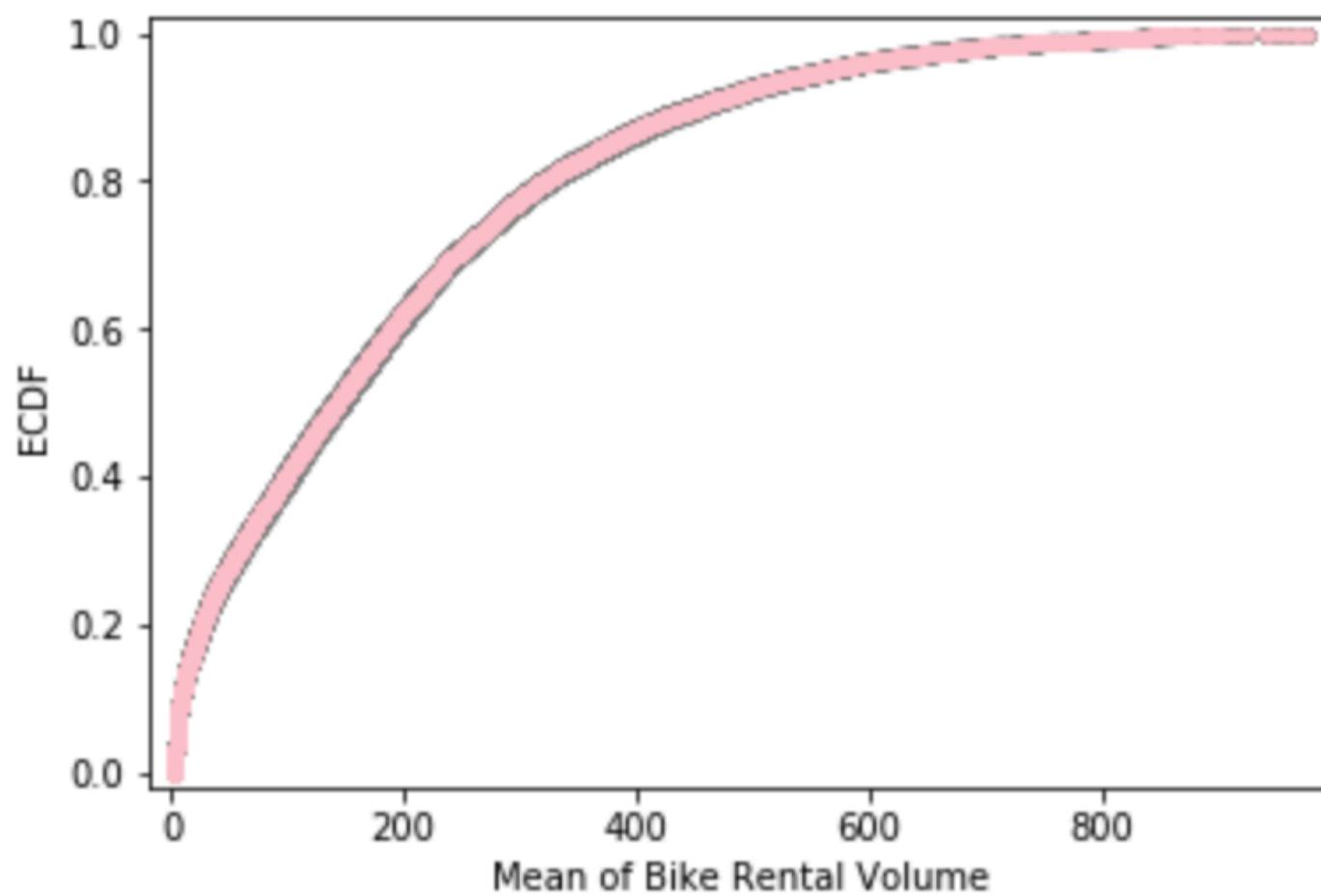
```
[33]: array([ 40. , 142. , 281. , 451.2, 690. , 977. ])
```

9. Checking ECDF Distribution of Bike Rental Count across two years (2011 & 2012) and theoretical samples of data



Compare the distribution of the data to the theoretical distribution of the data. This is done by comparing the ecdf First define a function for computing the ecdf from a data set. Next use `np.random.normal` method to sample the theoretical normal distribution and overlay the ecdf of both data sets to compare distribution. We see how closely the real data set follows the theoretical normal distribution curve.

10. Visualizing ECDF using bootstrap samples



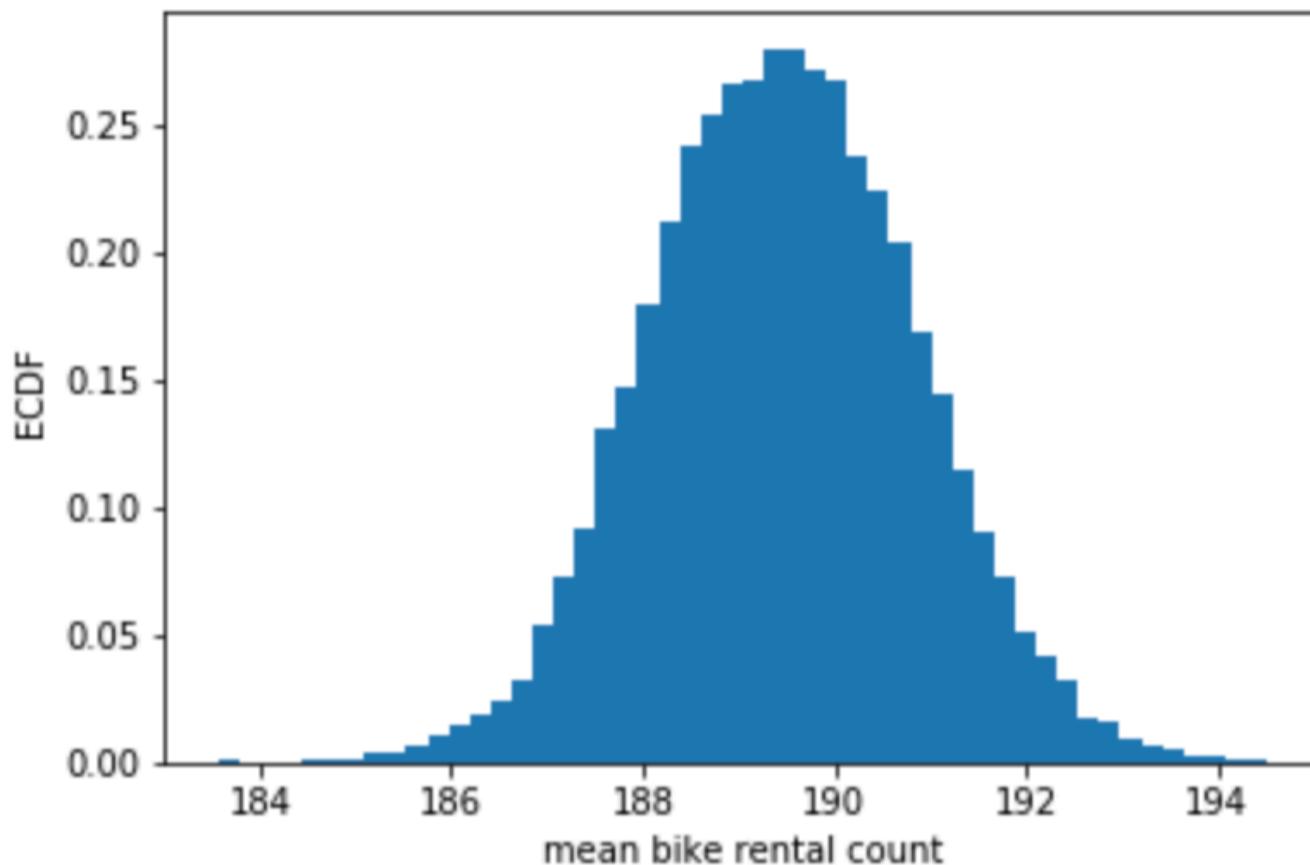
By graphically displaying the bootstrap samples with an ECDF, I see how bootstrap sampling allows probabilistic distribution of data.

11. Confidence Interval

Assuming 95% Confidence interval i.e. give the 2.5th and 97.5th percentile of bootstrap replicates is stored as bs_replicates

```
np.percentile(bs_replicates, [2.5, 97.5])  
O/p: array([186.82940186, 192.19181915])
```

Verifying it with histogram for bootstrap replicates



This is bootstrap estimate of the probability distribution function of the mean Bike Rental Count at the Capital Bikeshare System. Remember, we are estimating the mean Bike Rental Count we would get if the Capital Bikeshare System could repeat all the measurements from 2011 to 2012 over and over again. This is a probabilistic estimate of the mean. I plot the PDF as a histogram, and I see that it is not Normal as it has slightly longer left tail.

In fact, it can be shown theoretically that under not-too-restrictive conditions, the value of the mean will always be Normally distributed. (This does not hold in general, just for the mean and a few other statistics.) The standard deviation of this distribution, called the standard error of the mean, or SEM, is given by the standard deviation of the data divided by the square root of the number of data points. I.e., for a data set. Notice that the SEM we got from the known expression and the bootstrap replicates is the same and the distribution of the bootstrap replicates of the mean is Normal.

12. Extending Confidence Interval Concept to Pairs Bootstrap

Finding pairs bootstrap for slope & intercept of a linear function between Bike Rental Count and Registered User Type

15. ('Registered Sample Size:', 17379, '\nRegistered User Type Mean:',
153.78686920996606)

('Casual Sample Size:', 17379, '\nCasual User Type Mean:', 35.67621842453536)

There is a difference between the mean of registered and casual User Type in the sample data, but a statistical analysis will help determine if the difference is significant. Null Hypothesis: There is no significant difference between registered and casual user type mean on Bike Rental Count.

$$H_0: \mu_{\text{registered}} - \mu_{\text{casual}} = 0$$

Significance Level: 95% Confidence

$$\alpha = 0.05$$

16. Difference of Means by Permutation Samples

Using np.concatenate, np.random.permutation methods is
('Difference of Means', 118.11065078543069)
('p-value =', 0)

17. T-Test and P-Value

Using scipy import stats method performed the T-Test and P-value

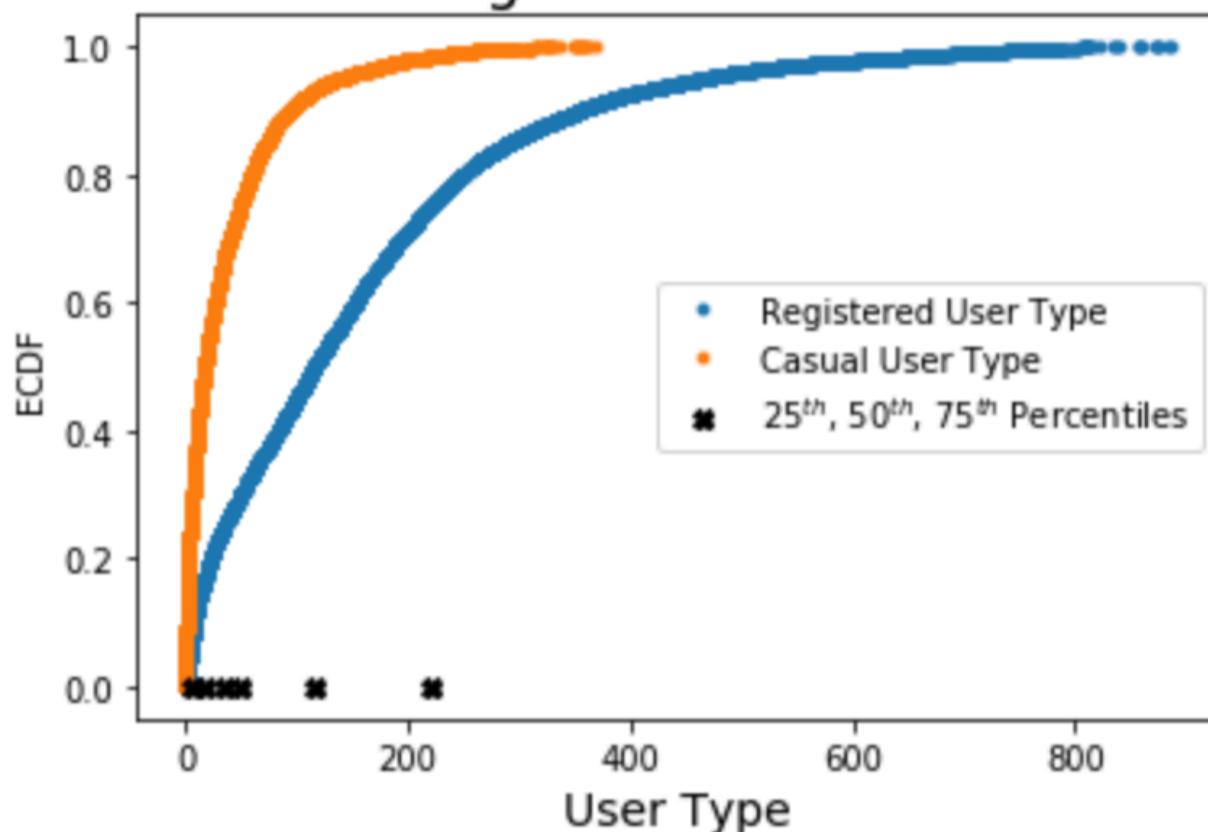
('t-statistic:', 97.81332643791566)
('p-value:', 0.0)

This confirms Null Hypothesis may be rejected as means of Bike Rental Count by User Type is not same.

18. ECDF Plots for Bike Rental Count Distribution by Registered and casual User Types



Distribution of Registered and Casual User Type



Hence by above plot, we may reject the Null hypothesis since there is significant difference between Bike Rental Count distribution of Registered and Casual User Type.

EDA Outcomes

Based on above EDA & Inferential Statistics, following are the outcomes:

- Bike rental count (volume) distribution is normal distribution curve and can be predicted based on given dataset.
- The feature set that drives this prediction is:
 - Casual
 - Registered
 - Hour (hr)
 - Windspeed
 - Humidity (hum)
 - Season
 - Month
 - Temperature (temp)
- ‘Registered’ User type is renting bikes more than ‘Casual’ during any period. Total count of ‘Bike Rent in a Day’ is sum of Registered and Casual bike rental count.
- Bikes are rented more during the working days than on weekends and holidays.

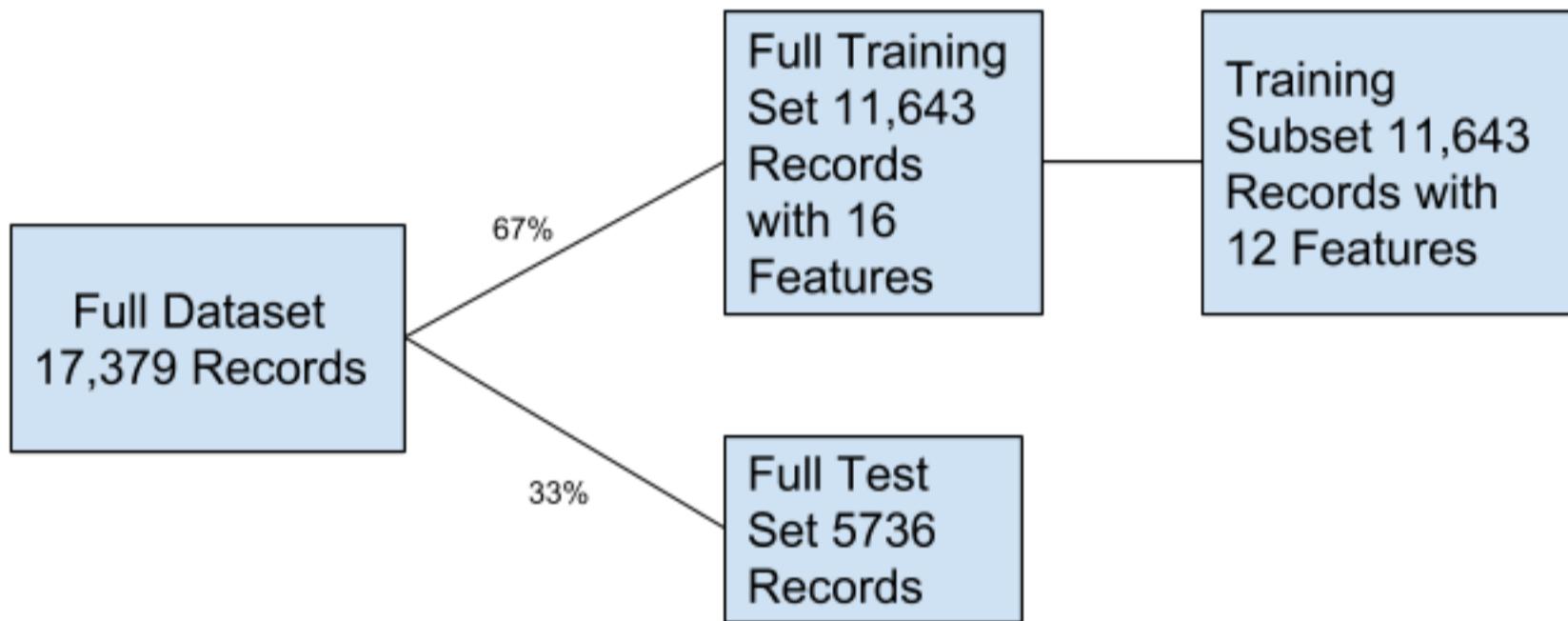
<https://github.com/rashi-n/Machine-Learning-Projects/blob/master/Capstone%20Projects/CS%20I%20-%20In%20Depth%20Analysis%20Regression.ipynb>

<https://github.com/rashi-n/Machine-Learning-Projects/blob/master/Capstone%20Projects/CS%20I%20-%20In%20depth%20Analysis.ipynb>

Step 2:

Develop Feature set for each train test split.

The no. of observations for each data split can be found in below figure:



Training subset is basically including all the features apart from 'Registered' and 'Casual' since they are in direct linear equation to 'Rental Bike Count' target label and so omitting them makes better sense in predictive analytics.

Step 3:

Total of 4 Machine Learning models were analyzed for this Supervised Linear Regression problem:

1. Linear Regression
2. Ridge Regression
3. Lasso Regression
4. Decision Tree Regression



Step 4, 6, 7:

Each of the above models were cross validated under 5 folds or 10 folds and models were trained using number of hyperparameters to ensure that the model is not underperforming due to lack of tuning.

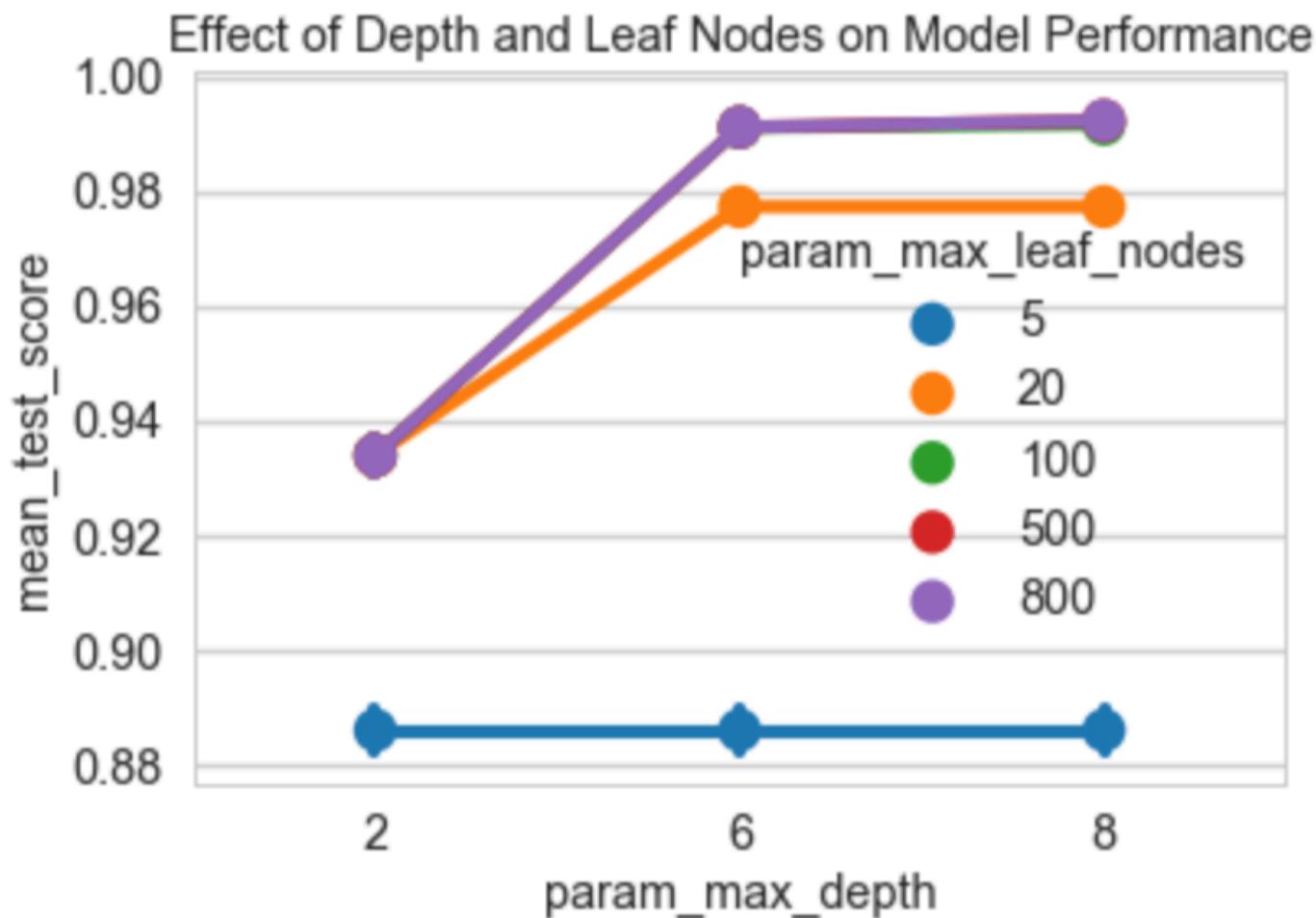
Decision Tree Regression was performed on all feature set with One-Hot Encoding applied for Categorical variables and

Performance Metric $R^2 = 0.8857$

Tuning was performed using Grid Search with Cross Validation technique and

Performance Metric $R^2 = 0.8582$

This improves the performance & almost close to 1.0



Residual Graph Plot between Observed Y-value and predicted Y -value for training data set and test dataset showed similar structure and thus affirmed the predictability of the model:

features ('Registered', 'Casual') from the training set as to perform deeper analysis on rest of the feature set.

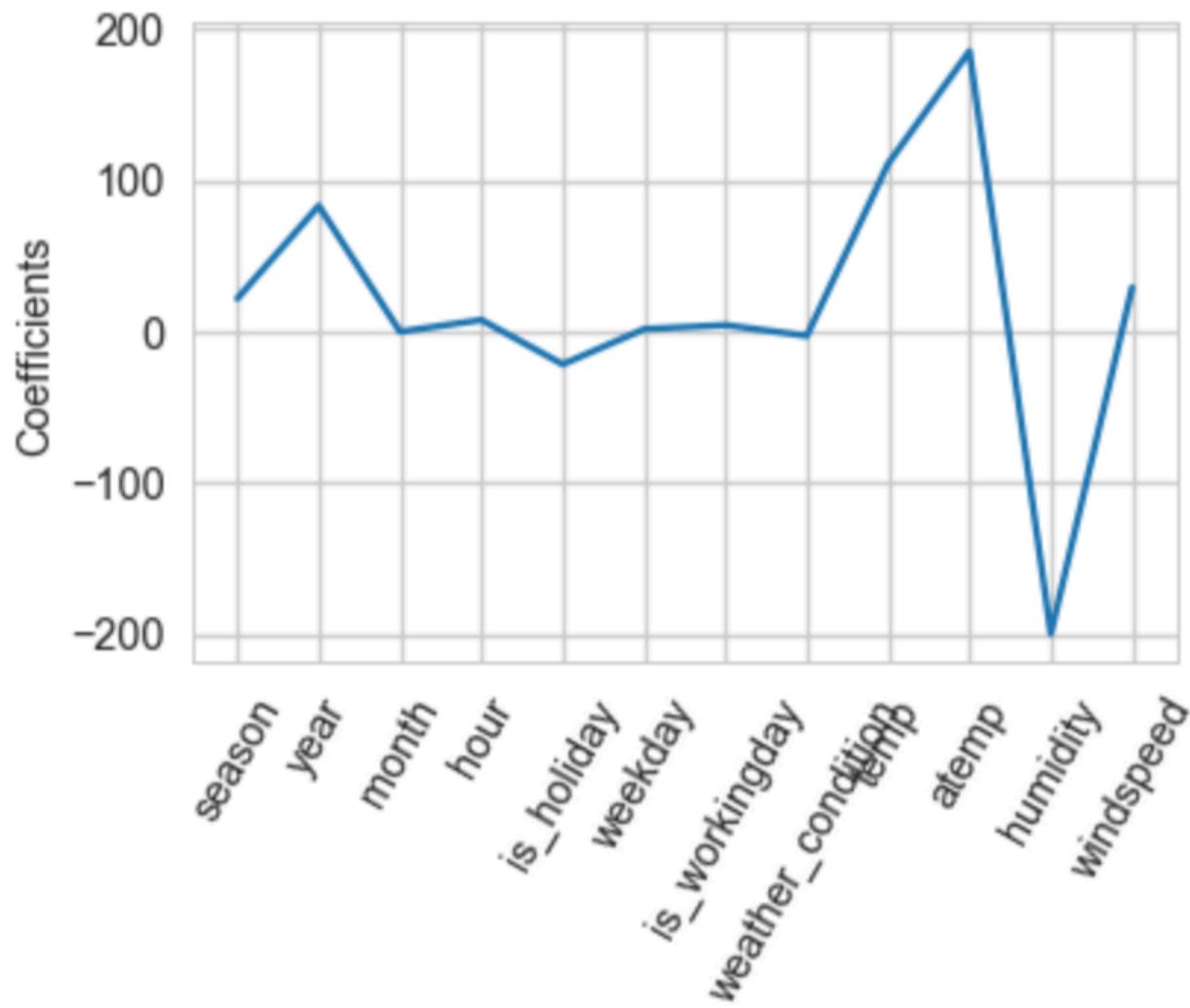
Step 5 & 6: Select Features and Models

```
x, x_test, y, y_test = train_test_split(stat.iloc[:,2:-3], stat.iloc[:,16],
                                         test_size=0.33, random_state=42)
```

Subset of training set was obtained per above train_test split and fitted for Linear Regressor, Ridge, Lasso and Decision Tree Regressor, following were the R² readings respectively for Test set(X_test, y_test):

0.387, 0.390, 0.378, 0.886

Lasso Coefficients Predictor graph for the Feature dataset is as shown below:



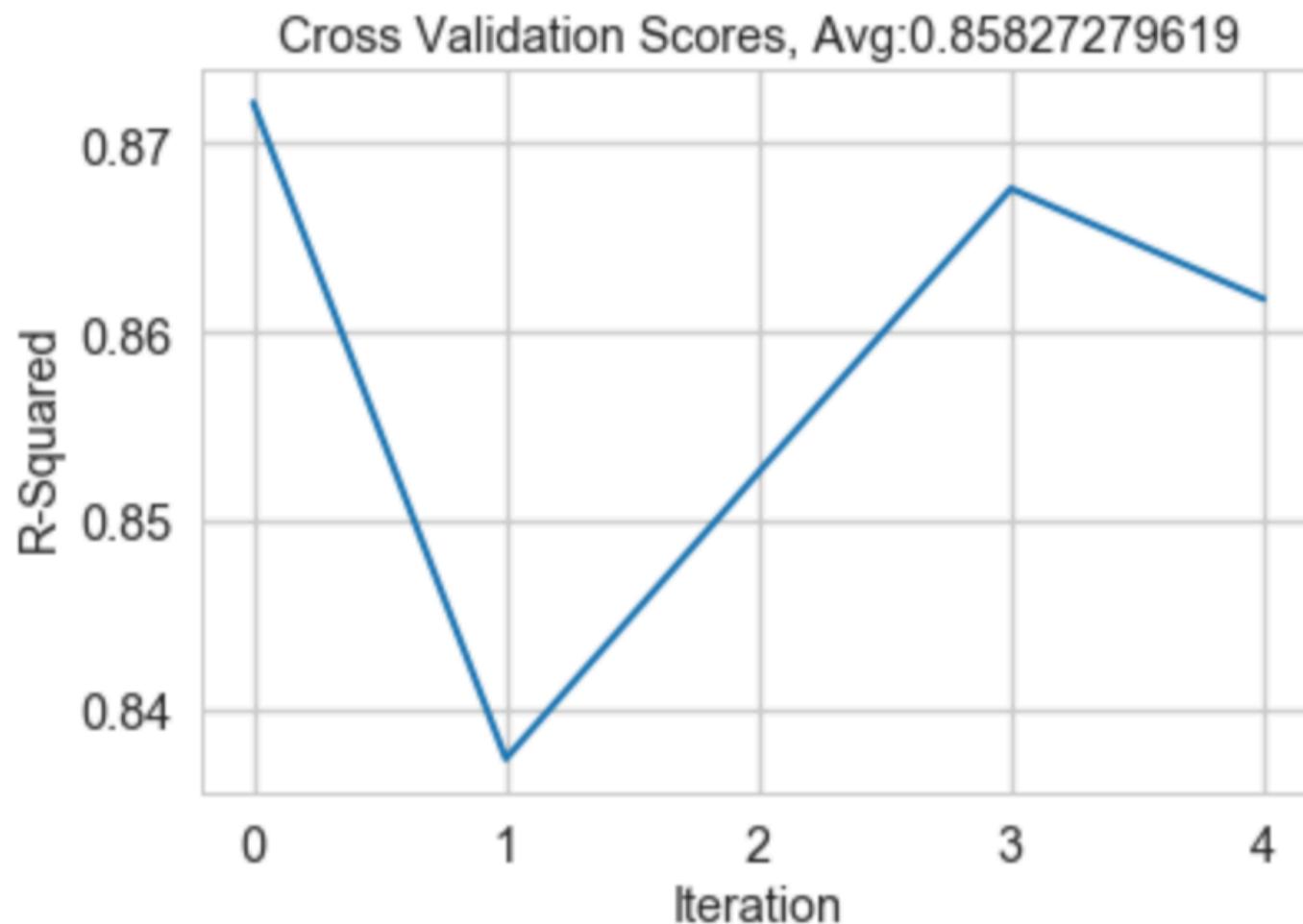
Above graph shows Temp and aTemp are most influencing predictors (after 'Registered', 'Casual') for Target variable 'Bike Rental Count'.

Step 7: Select Best Model and Tune Hyperparameters

Cross Validation and hyperparameter tuning was performed using Pipeline in case of Linear Regression model since I had combination of parameters to tune (elasticnet_l1_ratio, R^2).

```
Tuned ElasticNet Alpha: {'elasticnet_l1_ratio': 1.0}
Tuned ElasticNet R squared: 0.389258056093
```

To tune performance parameter R^2 in Decision Tree Regression model, cross validation 5 folds technique was applied and following was the mean of R^2:



On applying cross validation R2 average is lowered to 0.86 from 0.885 Decision Tree regressor score. This may be due to overfitting. Hence limiting the training set to subset of features and applying cross validation technique is not improving the model.

Other approach was taken to run Grid Search with Cross-Validation on entire feature set was taken, this resulted in best hyperparameter values as shown below:

```
R-Squared::0.996848172075
Best Hyperparameters::
{'min_samples_split': 10, 'max_leaf_nodes': 500, 'criterion': 'mse', 'max_depth': 8, 'min_samples_leaf': 20}
```




	mean_fit_time	mean_score_time	mean_test_score	mean_train_score	param_criterion	param_max_depth	param_max_leaf_nodes	param_min_samples_leaf
0	0.024276	0.004257	0.893902	0.897234	mse	2	5	20
1	0.021610	0.004356	0.893902	0.897234	mse	2	5	20
2	0.024005	0.004129	0.893902	0.897234	mse	2	5	20
3	0.023785	0.004360	0.893902	0.897234	mse	2	5	40
4	0.022789	0.004187	0.893902	0.897234	mse	2	5	40

Step 8 : Evaluate the Model

The model achieved highest R^2 value of 0.996 and best hyperparameters of:

Minimum Samples Split: 10

Max Leaf Nodes: 500

Criterion: MSE

Max Depth: 8

Min Samples Leaf: 20

Machine Learning Model Comparison

	R squared	MSE	Grid Search Cross Validation R squared/Elastic Net using Pipeline
Linear Regression(Test set with partial feature set)	0.387	-4525.62	0.389, 1.0
Ridge Regression(Test set with partial feature set)	0.390		0.389, 1.0
Lasso Regression(Test set with full feature set)	0.378		0.389, 1.0

Model	R squared	AIC	BIC
Decision Tree Regression(Test set with partial feature set)	0.886	-97.24	0.858

Based on ML analysis, the best model is devised using Decision Tree regressor (R squared 0.886) .

Recommendations and Future Work

With this predictive model, Client may benefit in better prediction of Bike Rental Count -

- Demand for bike share program is maximum during the day between 7 AM to 9 AM and 4 PM to 6 PM
- weather conditions like temperature, humidity, windspeed direct correlation with Bike Rental Count. Windspeed has inverse collinearity while temperature & humidity are positive correlated.
- Registered and Casual users are in equation to total bike rental count. Bike Rental Count distribution by Registered and Casual for given years is good to predict the Bike Rental Count for any future years as verified in Exploratory Data Analysis of Bootstrap samples.

There is lot of potential to enhance the model by

- Collection of more features in the dataset like Gender and Age to help customer know if bike rental preference is by any age or gender group
- Model improvement using other Regression models like Random Forest, Support Vector

References

https://www.washingtonpost.com/local/trafficandcommuting/capital-bikeshare-gears-up-for-another-expansion/2017/10/02/bcf81b4a-a2fe-11e7-ade1-76d061d56efa_story.html?utm_term=.7690885fb8e3

<http://scholarcommons.usf.edu/cgi/viewcontent.cgi?article=1196&context=jpt>

https://en.wikipedia.org/wiki/Data_wrangling

<https://www.datawatch.com/what-is-data-wrangling/>

