RTSM Report
MA31020

Shubham Mawa - 16IM10033

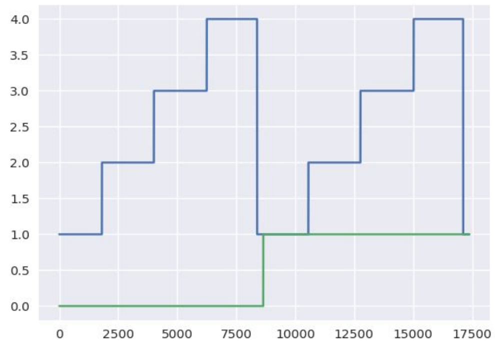
Background and Data description

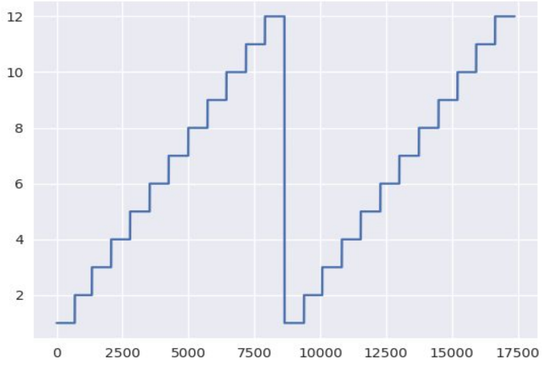
Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become automatic.Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.
Bike-sharing rental process is highly correlated to the environmental and seasonal settings. For instance, weather, precipitation, day, season, hour, etc. can affect the rental behaviors.
The dataset consists of columns like season, year, month, holiday, weather situation, temperature, humidity, wind speed and the total count of rental bikes for each observation.

Data Analysis

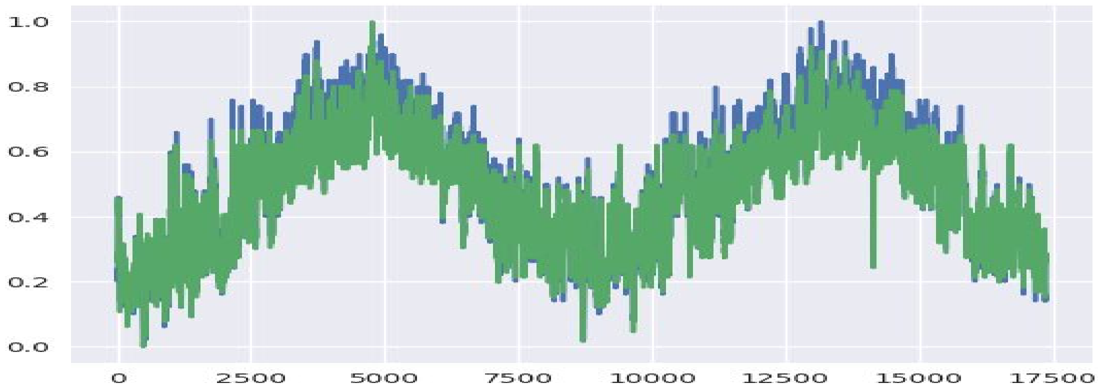
Visualization

The individual variable vectors are extracted from the hour.csv file and plotted to check for any observable pattern or structure in the data and also to study the relationship between different variables if there exists. Some of the plots are shown here





These plots clearly show that there is some seasonality in the data as it was expected, because the bike rental count would depend on season, time of the day, temperature, humidity etc.



We make another plot of the variables temp and atemp, and from the plot it looks that they are highly correlated. We find the correlation coefficient to verify our observation. **Python cmds:**

```
>>print(pearsonr(y10,y11))
>>0.98 : Highly correlated
```

Thus we should check the adjusted r_square value of our fitted model with and without including one of these variables when we do the model adequacy checking and possibly eliminate one of these variables when we finally fit the model.

Correlation coefficients for some other variables are also calculated and found to be close to zero and hence we will assume them to be uncorrelated while fitting the model.

Model Selection and Fitting

A linear regression model is fitted to the data. The total rental count is the dependent variable y, and the various features like season, month, hour, holiday, weekday, temp, humidity etc are the regressor variables.

Before fitting the data an important preprocessing step needs to be done. Some of the features are categorical variables i.e they take some finite values. Such type of data often causes the performance of some models to drop (because for example month 1,2,3..12 represent the months but there is no mathematical relation between them like 2 is twice of 1 so Feb is twice of Jan) and thus a one-hot representation of data is used to solve this problem. A one-hot matrix will be formed with no of columns = no of variables, and rows= no of observations, for each observation one such column will be 1 and others will be 0. **Python cmds:**

```
>>from keras.utils import to_categorical
>>x2 = to_categorical(y2)
>>x4 = to_categorical(y4)
>>x5 = to_categorical(y5)
```

After these operations we finally concatenate all the feature vectors into one X matrix.

```
>>X = np.concatenate((x2,x3,x4,x5,x6,x7,x8,x9,x11,x12,x13),axis=1)
```

A linear regression model is then fitted.

```
>>reg = LinearRegression()
>>reg.fit(X1,y)
```

The coefficients are obtained using: >>**print(reg.coef_)**

```
[ 5.02542897e+14  5.02542897e+14  5.02542897e+14 5.02542897e+14  8.59375000e+01
5.31788576e+14  5.31788576e+14  5.31788576e+14  5.31788576e+14  5.31788576e+14
5.31788576e+14  5.31788576e+14 5.31788576e+14   5.31788576e+14  5.31788576e+14
5.31788576e+14 5.31788576e+14  2.02560661e+14   2.02560661e+14  2.02560661e+14
2.02560661e+14 2.02560661e+14  2.02560661e+14   2.02560661e+14  2.02560661e+14
2.02560661e+14  2.02560661e+14  2.02560661e+14  2.02560661e+14  2.02560661e+14
2.02560661e+14  2.02560661e+14  2.02560661e+14  2.02560661e+14  2.02560661e+14
2.02560661e+14  2.02560661e+14  2.02560661e+14  2.02560661e+14  2.02560661e+14
2.02560661e+14  3.68259758e+14  5.76622594e+14  2.08362836e+14  2.08362836e+14
2.08362836e+14  2.08362836e+14  2.08362836e+14  5.76622594e+14  3.68259758e+14
1.67833051e+12  7.38687194e+14  7.38687194e+14  7.38687194e+14  7.38687194e+14
1.16968750e+02  1.27781250e+02 -8.21250000e+01 -2.87578125e+01]
```

The r_square and adjusted r_square values are calculated.

```
>>r_square = reg.score(X1,y)
>>adjusted_r_square = 1-(1 - r_square)*(n-1)/(n-p-1)
```

n = No of observations, **p** = no of variables

Values : (when temp was removed due to high correlation with atemp)
r_ square = 0.685967576251913
Adjusted_r_square = 0.6849159665188074

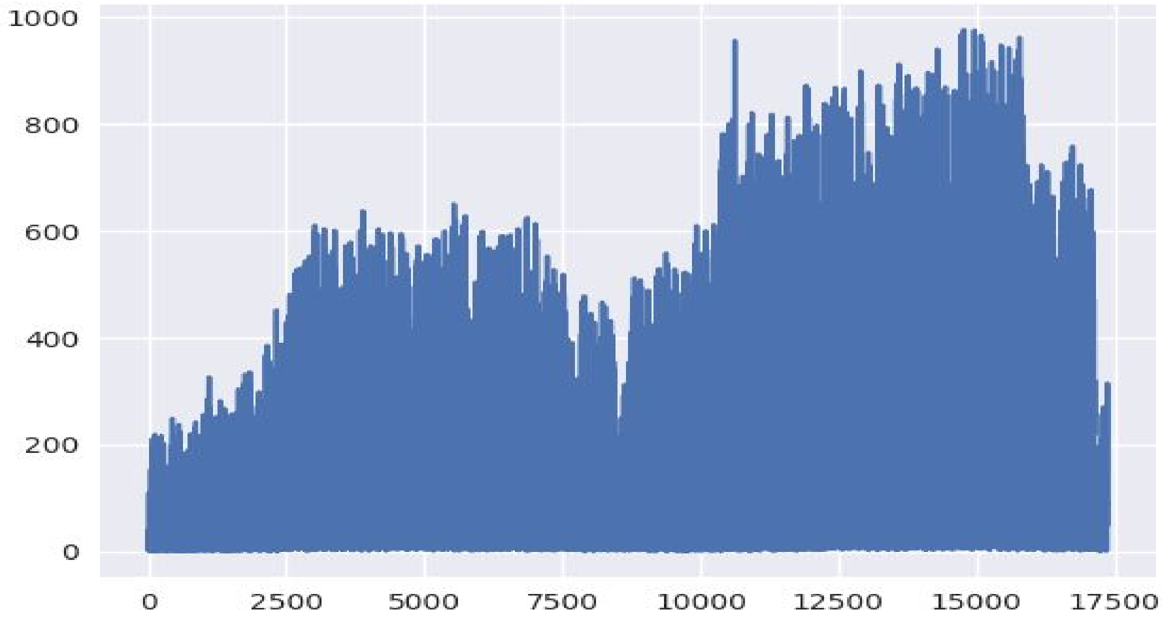
>>X = np.concatenate((x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13),axis=1) (x10 is included here)
Values : (when temp and atemp are both included while fitting the model)
r_ square = 0.6862732031998036
Adjusted_r_square = 0.6852044416655804

There is no significant change in the r_square and adjusted r_square values. We expected adjusted r_square to decrease a little bit due to high correlation between temp and atemp but the values indicate that inclusion or exclusion of an extra variable in this case does not cause much difference in model fitting.

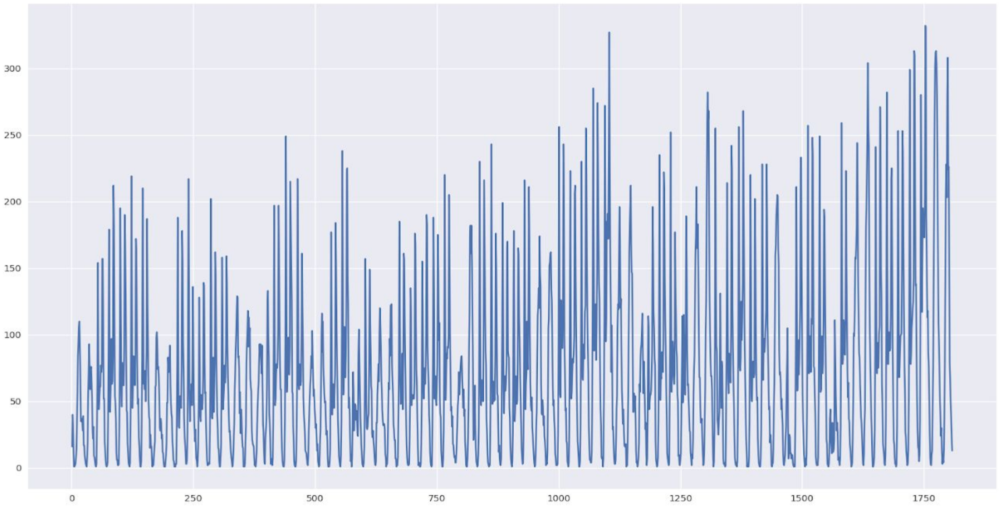
Alternative approach:

The data can also be modelled as a time series. We take only the total rental count variable and consider it as a time series. In this approach we are not using the data of the other variables and thus the effect of those variables on the dependent variable is not studied.

Modelling the seasonality in the data is not straightforward in this case as there can be periodicity in the data even when we consider the time period as an hour, 3-4 hours, a day, a month or a season. Plotting all the data together does not give a very good visualization.



Plotting the data for season 1 (similarly can be done for all the data)



Thus one approach would be to apply different ARIMA models to different parts of the data.
A single Seasonal ARIMA model when applied to the model did not yield very good results.
The Linear Regression model yields better results(adjusted_r_square= 0.6852044416655).