

Demand Forecasting problem :

Steps followed to attack the problem –

1. Reading the input Json file to get the data points. Pandas data frame is used to do this.
2. Now this data frame need to be converted in to time series object.
3. As we get time series object we an aggregate the timestamps of 15 minute chunks.
4. Now look for any missing values. If there any then replacing it with 1 or 0, depends on the stationarity of the series.
5. Now we have time series, we need to calculate the stationarity check on series.
6. I used rolling mean and rolling standard deviation for checking the stationarity of the series.
7. Rolling mean and Rolling standard deviation has very less variability with respect to time. This proves that series is stationarity. [Also tried dickey fuller stationarity check.]
8. Now plotted the ACF and PAF plots for getting the order of auto regressive terms and moving average terms, which need to pass as an argument in ARIMA model.
9. Then we build the model, using following values of p, d and q:
10. Then we forecast the demands using previously trained model.
11. Then calculated the errors between actual and forecasted demands.

Answer 1 :

- Reading of data and converting in to time series object.
- Aggregating in to 15 minute interval and plotting the aggregated data to visualize the distribution.
- For visualization I used simple line chart and box plot.
- Box plot is the best way to get the 1st quartile, median, 3rd quartile and 4th quartile of the data, which shows the variability of the data in best way.
- Box plot is also very useful to get the outliers observations in the data.
- By looking In to the line chart of the aggregated data, it is very much clear that there is no upward or downward trend present in the data. Since we have only 4 month of data seasonality can not be seen in the observations.

Answer 2 :

- For Building the forecasting model, first step is to check if time series is stationary or not? To check stationarity statistical properties (mean, variance, standard deviation) of the series should not change with respect to time.
- For checking stationarity of series I tried plotting rolling mean and rolling standard deviation. Variance of rolling mean and standard deviation was not varying with time.
- Also tried running dickey fuller stationarity check.
- By all the above observations we ensured that time series is stationary.
- Now since TS is stationary, we don't need to run any differencing or transformation operations on TS.

- Now next step was to figure out the order of ARIMA model.
- For that I plotted ACF and PACF plots, and with the help of these plots I came up with order of ARIMA model.
- I choose p (autoregressive terms)=4, d (differencing)=0 since series was already stationary and q (moving average term)=2.
- Then I simply run ARIMA model to train the model.
- And I predicted some numbers using the trained model.
- Then I calculated error matrices like APE (Absolute percentage error) and AWV (Absolute weighted variance.)
- I tried predicting on training set itself, and it was giving me following results:
 - APE = 61.58%
 - AWV = 40.69%

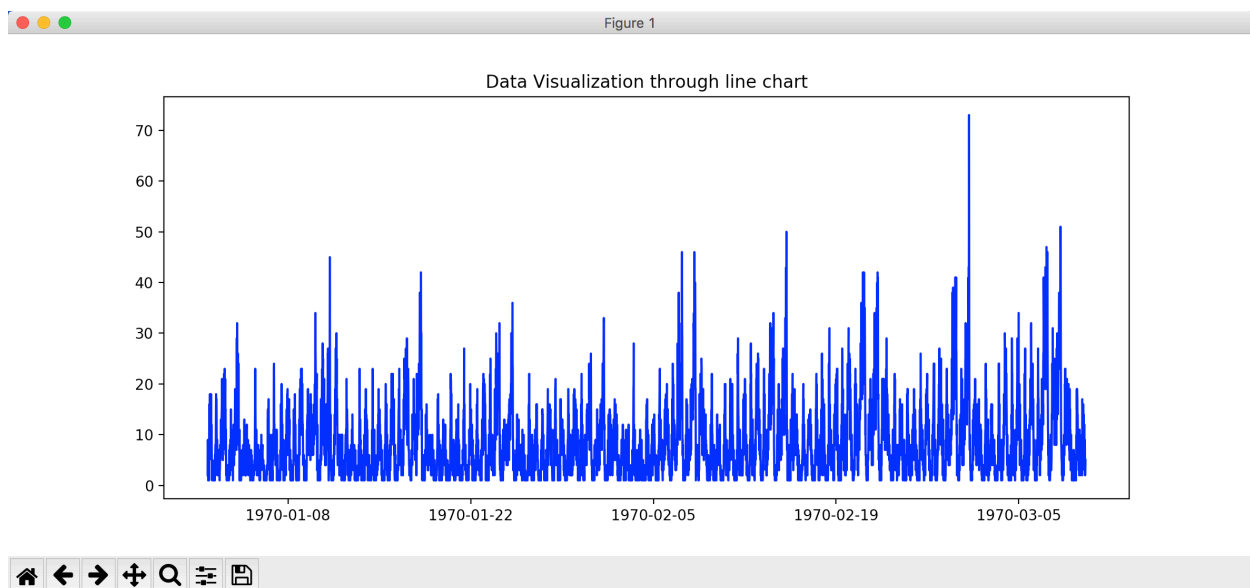
Due to time constraints I did not work that much on accuracy. But a lot can be done to improve accuracy.

Answer 3 :

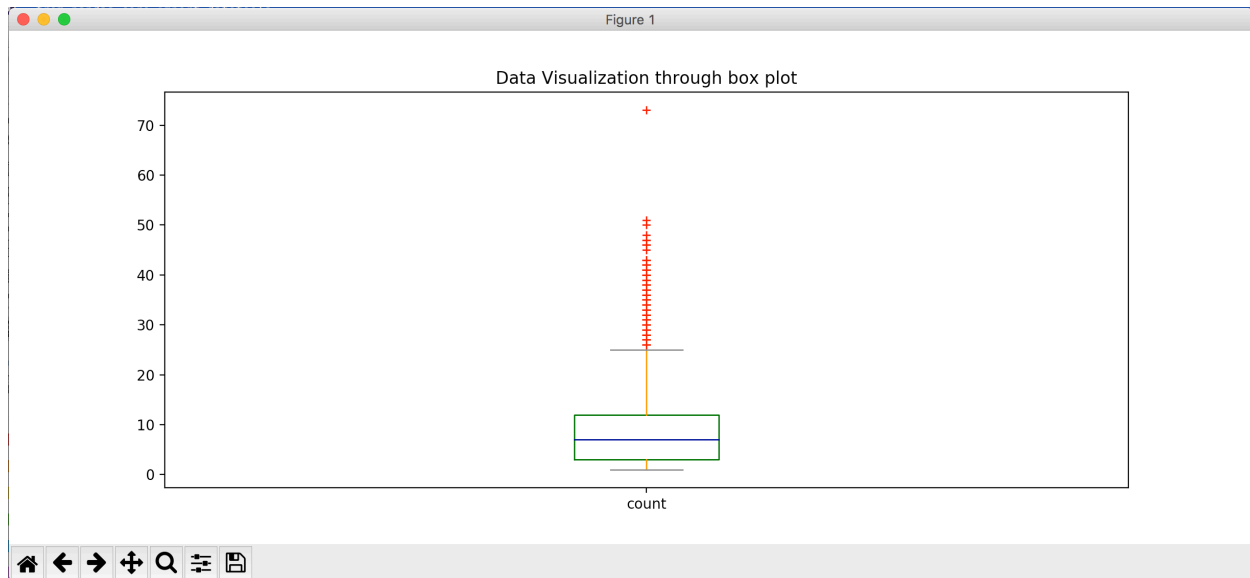
This part is already covered in answer 2.

Here are plots and graphs :

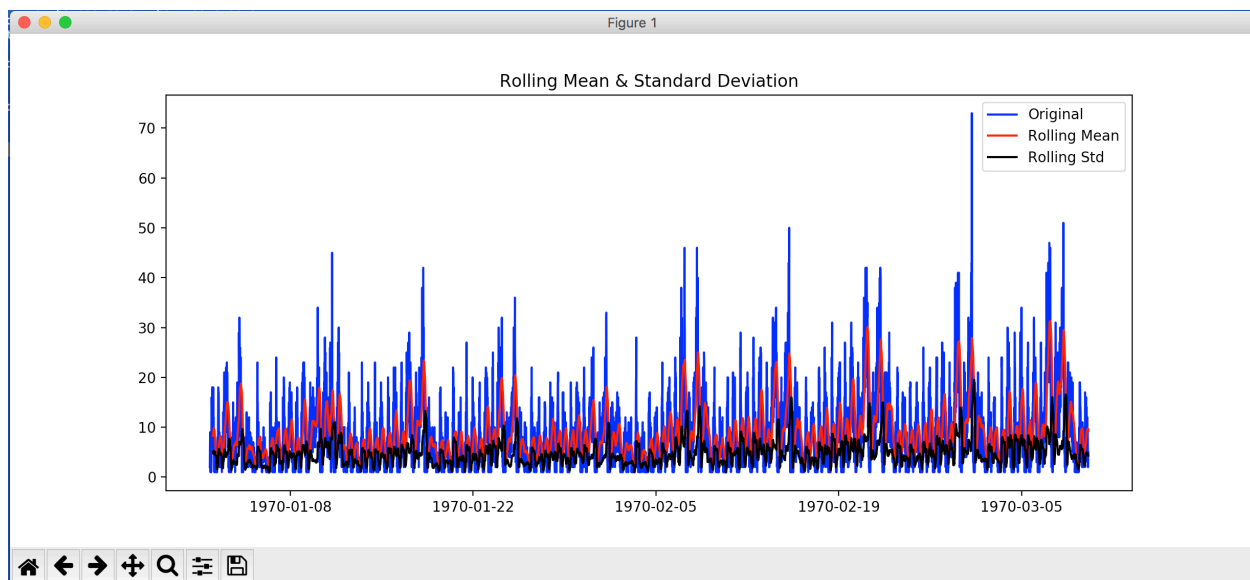
1. Line chart for data visualization. Here I have plotted less data because for complete dataset this chart was getting messed up. So this chart is only for 66% of the data.



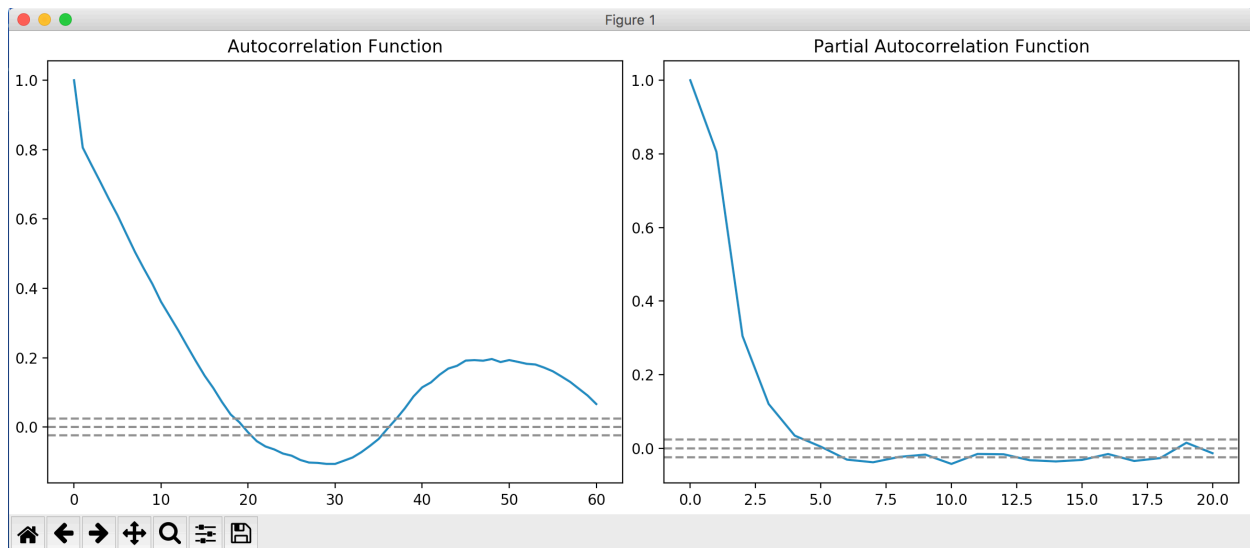
2. Here is box plot to know the variability of the data. Also to know the outliers in the data.



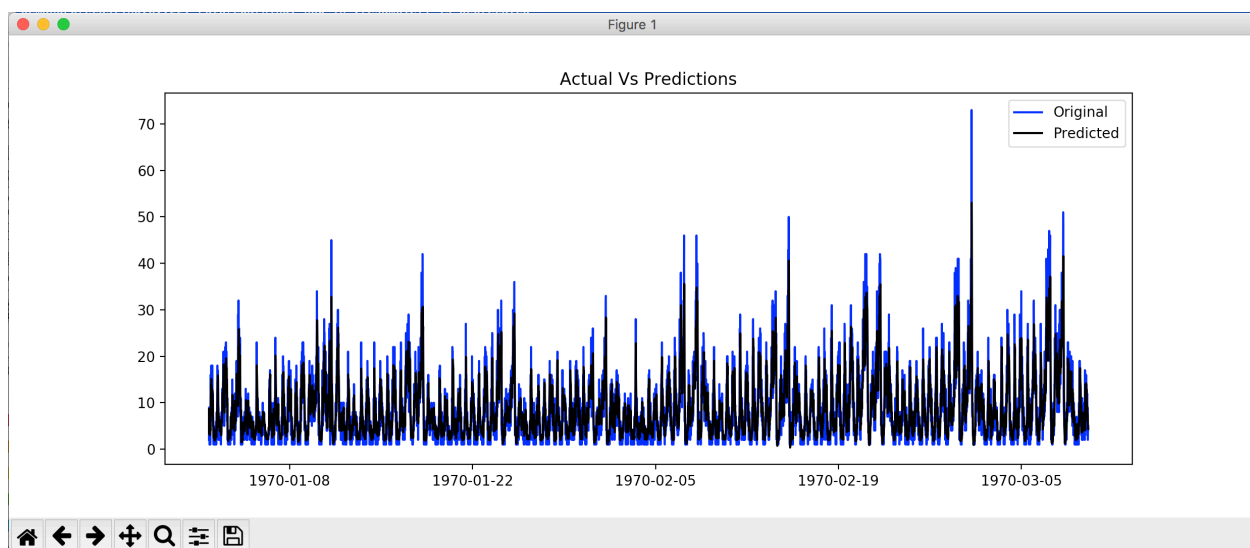
3. Here is rolling mean and rolling standard deviation to know of series is stationary or not? Clearly there is no upward or downward trend in the data. So series seems to be stationary.



4. Here is ACF and PACF plots. Which gives a sense of choosing order for ARIMA model.



5. Here is actual and predicted series. Blue one is actual and black is predicted. We can see from the chart that picks and valleys are captured in most of the cases.



6. Here is screenshot of error matrices. AWV and APE is shown in pic.

```
test — -bash — 149x37
...achineLearning/test — -bash  .../python/pythoncode — -bash  ~/Dipesh/DFE/oozie — -bash  ...IF_train_test/nif_train — -bash ...  ...IF_train_test/nif_train — -bash ...
3c15c2d502ee:test z001l92$ python demandF0recasting.py
/Users/z001l92/anaconda2/lib/python2.7/site-packages/statsmodels/compat/pandas.py:56: FutureWarning: The pandas.core.datetools module is deprecated a
nd will be removed in a future version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
demandF0recasting.py:22: FutureWarning: how in .resample() is deprecated
the new syntax is .resample(...).sum()
  agg_df = df.resample('15T',how='sum')
demandF0recasting.py:49: FutureWarning: pd.rolling_mean is deprecated for DataFrame and will be removed in a future version, replace with
  DataFrame.rolling(window=24,center=False).mean()
  rolmean = pd.rolling_mean(agg_df, window=24)
demandF0recasting.py:50: FutureWarning: pd.rolling_std is deprecated for DataFrame and will be removed in a future version, replace with
  DataFrame.rolling(window=24,center=False).std()
  rolstd = pd.rolling_std(agg_df, window=24)
('Absolute percentage error is : ', 61.582198325235005)
('Absolute weighted variance is : ', 40.696404420590241)
3c15c2d502ee:test z001l92$
```

Thanks
Dipesh
dipeshnjoshi@gmail.com
9743754445