TIM SERIES AND REGRESSION ANALYSIS

Metro-interstate Traffic Volume Analysis

MODULE LEADER

DR ALESSANDRO DI STEFANO

A.DISTEFANO@TEES.AC.UK

TEESSIDE UNIVERSITY

STUDENT

MOHANA KAMANOORU

A0223038@LIVE.TEES.AC.UK

TEESSIDE UNIVERSITY

Table of Contents

1. Abstract

With the increase in road traffic volumes every year worldwide, it is crucial to consider required road maintenance projects, alternate route plans to avoid traffic congestions and reduce accidents. Hence, it is vital to understand the traffic trends and patterns for better planning and hence forecasting road traffic volume becomes critical. In this paper, different learning algorithms are studied and applied on a dataset consisting of hourly traffic data for more than six years. Research questions are answered by performing regression and time-series analysis of machine learning methodologies. Different models are developed using the appropriate algorithms, and the efficiency and performance of each model are compared and analysed.

The data in this study contains dependent and independent variables. It also includes time-series data. Regression Analysis and Timeseries analysis is performed on the data to forecast the traffic volume for 12 months. The accuracy achieved using different algorithms are as follows. Timeseries analysis with Holt-winters and ARIMA is 68% when used interpolate to fill the missing time-series values. Regression analysis with Multiple Linear regression is 14%. With K Neighbor Regressor is 82.4% with n=2, the powerful Support Vector Regressor is 78.98% and the most accurate model from all these algorithms Multi-Layer Perceptron Model (ANN) 94% accuracy.

2. Introduction

The evolution of technology is constantly progressing every year, the power of breaking down complex problems is exponential by applying many advanced machine learning algorithms by analysing the big data, which might be humanly not possible. This research main intention is to build a machine learning model with higher efficiency to predict traffic volume from the available past data.

3. Dataset Description

Data for this study is obtained from UCI Machine Learning Repository[1]. This dataset provides information about hourly traffic volume between Interstate 94 Westbound ATR station 301, between Minneapolis and St Paul, MN. Details of weather conditions and holiday details are provided. It contains 48204 records with nine columns. Detailed column description and analysis are provided in this research paper.

4. Exploratory Data Analysis and Visualisation

The dataset consists of 48,204 rows and nine columns. The column names and column values are shown below.

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
0	None	288.28	0.0	0.0	40	Clouds	scattered clouds	2012-10-02 09:00:00	5545
1	None	289.36	0.0	0.0	75	Clouds	broken clouds	2012-10-02 10:00:00	4516
2	None	289.58	0.0	0.0	90	Clouds	overcast clouds	2012-10-02 11:00:00	4767
3	None	290.13	0.0	0.0	90	Clouds	overcast clouds	2012-10-02 12:00:00	5026
4	None	291.14	0.0	0.0	75	Clouds	broken clouds	2012-10-02 13:00:00	4918

Figure 1 Raw Dataset Columns (Head)

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
48199	None	283.45	0.0	0.0	75	Clouds	broken clouds	2018-09-30 19:00:00	3543
48200	None	282.76	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 20:00:00	2781
48201	None	282.73	0.0	0.0	90	Thunderstorm	proximity thunderstorm	2018-09-30 21:00:00	2159
48202	None	282.09	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 22:00:00	1450
48203	None	282.12	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 23:00:00	954

Figure 2 Raw Dataset Columns (Tail)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48204 entries, 0 to 48203
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   holiday              48204 non-null  object
 1   temp                 48204 non-null  float64
 2   rain_1h              48204 non-null  float64
 3   snow_1h              48204 non-null  float64
 4   clouds_all           48204 non-null  int64
 5   weather_main         48204 non-null  object
 6   weather_description  48204 non-null  object
 7   date_time            48204 non-null  object
 8   traffic_volume       48204 non-null  int64
dtypes: float64(3), int64(2), object(4)
memory usage: 3.3+ MB
```

Figure 3 Raw dataset (info)

4.1. Univariate Analysis

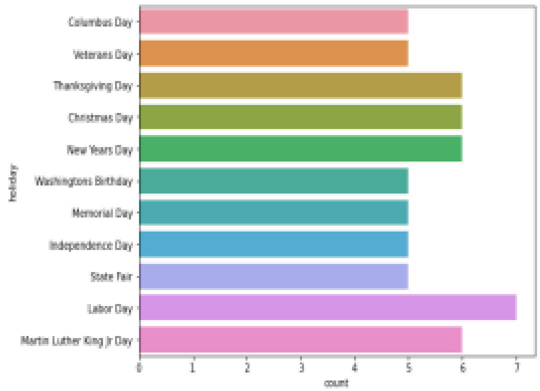
The column name and data types o the columns of the dataset are mentioned in the below table.[1]

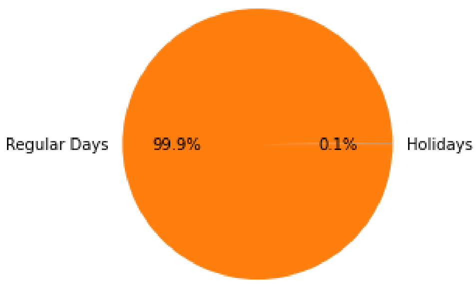
Holiday	Minnesota State holidays (regional and national)	Categorical	String
Temp	Temperature in kelvin (Average)	Numeric	Float
Rain	Hourly rain in millimetres	Numeric	Float
Snow	Hourly snow in millimetres	Numeric	Float
Clouds all	Clouds cover in percentage	Numeric	int
Weather main	Weather description in short	Categorical	String
Weather description	Comprehensive weather description	Categorical	String
Date time	Time and date the data is recorded	Time Series	Date Time
Traffic volume	Traffic volume per hour	Numeric	int

Figure 4 Column and data types

Column values analysis and visualisation

Column 1: holiday

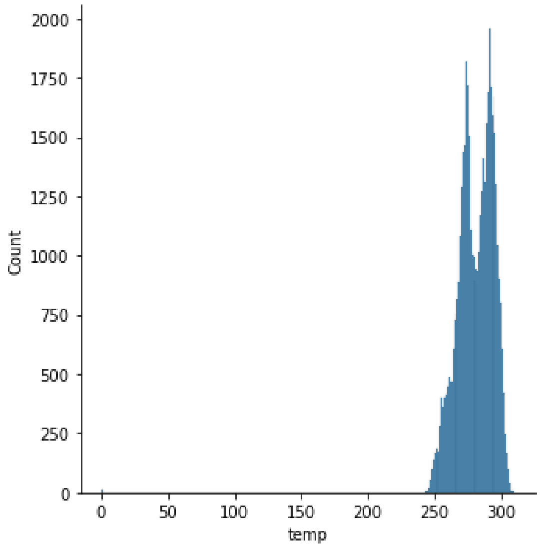


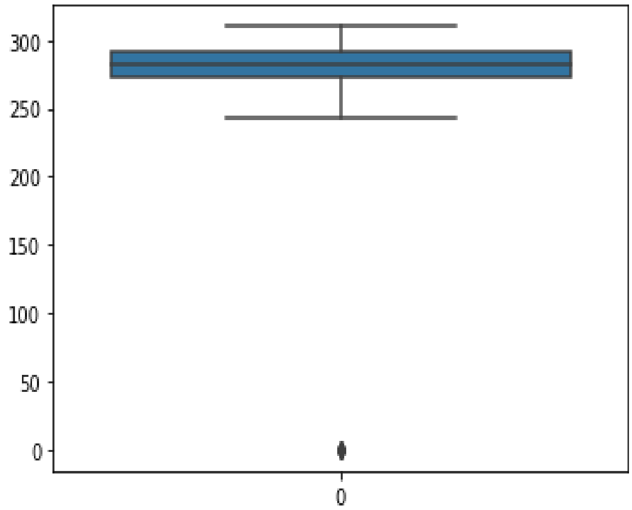


Number of holidays: 61
Unique number of holidays: 11
number of regular days :48143

The dataset contains 11 different holidays recorded and rest of the records are for regular days. The data for regular days is huge (99.9%) compared to the holiday data (0.1%). So, analysing the dataset for holiday related information may not be suitable for machine learning algorithms since the amount of data recorded is only 68 rows for holidays.

Column 2: Temperature

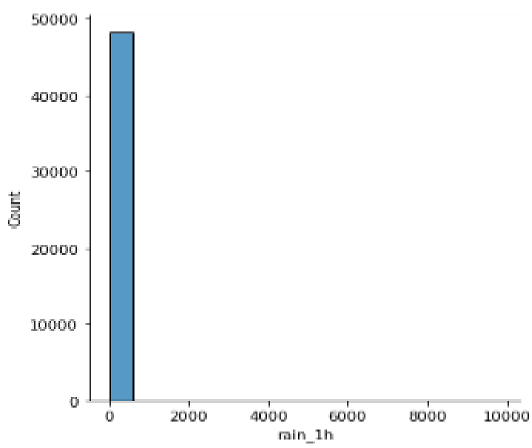


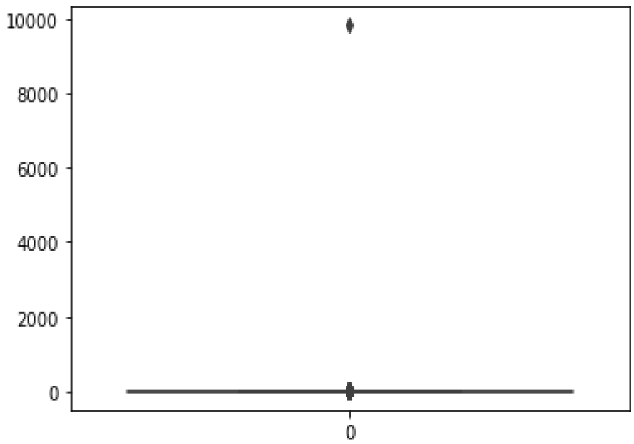


Observation:

Temperature is recorded in kelvin, could be converted to Celsius for easy understanding. possible outlier identified. Temp data to be converted to Celsius, practically temperature will be 0 kelvin which is -273 Celsius. so, this is a definite wrong data present in the dataset.

Column 3: Rain





```
max_rain = traffic[traffic['rain_1h'] == 9831.300000]
max_rain
```

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
24872	None	302.11	9831.3	0.0	75	Rain	very heavy rain	2016-07-11 17:00:00	5535

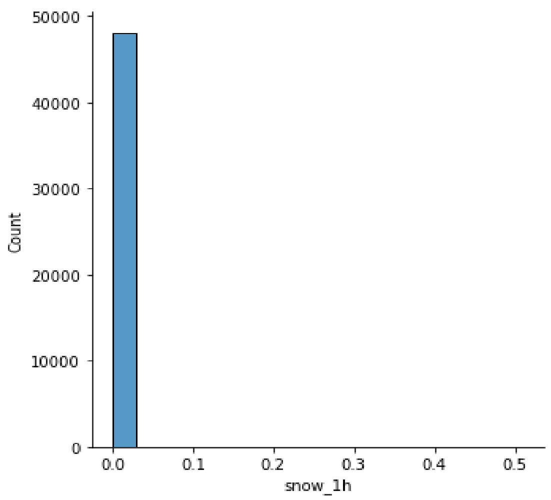
Observation:

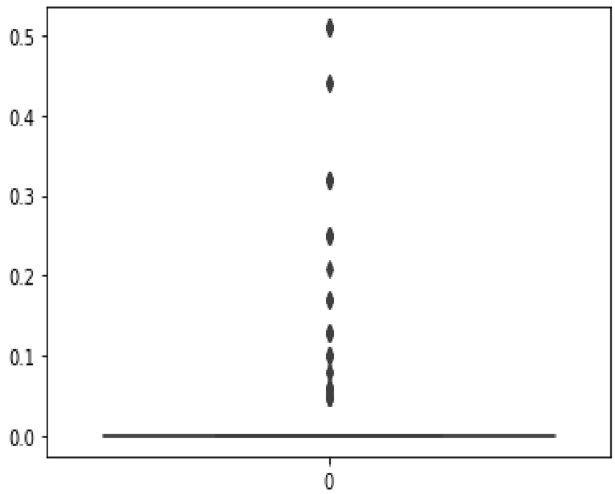
Amount in mm of rain that occurred in the hour. possible outlier identified on 11th July 2017. From wunderground website, when looked for rain and temp.

max temp = 90 F --> 305.372 Kelvin
avg temp = 79.2 F --> 299.3722 Kelvin
min temp = 72 F --> 295.372 Kelvin

no abnormal precipitation recorded on 11th but rain with precipitation of 2.17 inches (= 55mm) is recorded on 24th Sunday July 2017. Recorded temp is 302 which matches with the data on wunderground, but no heavy rain is recorded, hence this must be corrupt data.

Column 4: Snow





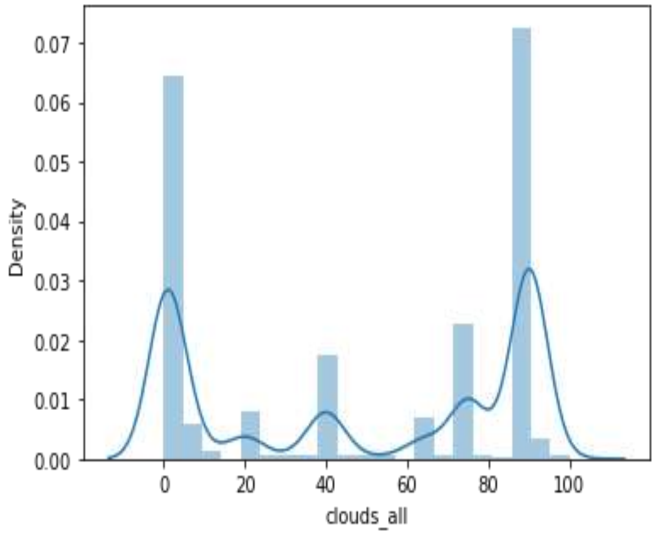

```
traffic[traffic['snow_1h'] > 0.4]
```

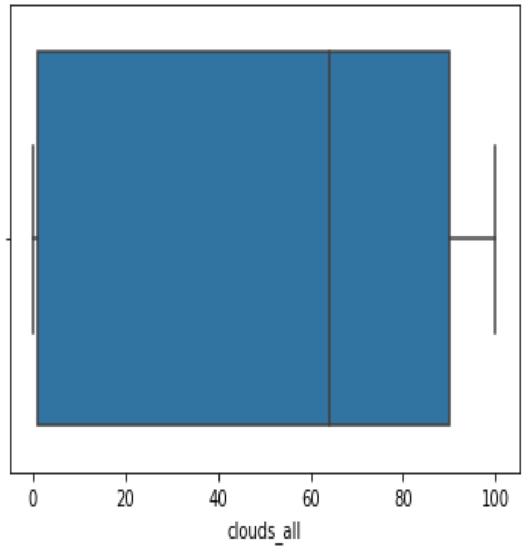
	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
20158	None	274.33	0.98	0.51	90	Rain	moderate rain	2015-12-23 12:00:00	5167
20159	None	274.33	0.98	0.51	90	Snow	snow	2015-12-23 12:00:00	5167
20160	None	274.33	0.98	0.51	90	Mist	mist	2015-12-23 12:00:00	5167
20161	None	274.33	0.98	0.51	90	Fog	fog	2015-12-23 12:00:00	5167
20268	None	267.14	0.00	0.44	90	Snow	snow	2015-12-28 22:00:00	2165
20269	None	267.14	0.00	0.44	90	Mist	mist	2015-12-28 22:00:00	2165
20270	None	267.06	0.00	0.51	90	Snow	snow	2015-12-28 23:00:00	888
20271	None	267.06	0.00	0.51	90	Mist	mist	2015-12-28 23:00:00	888

Observation:

Amount in mm of snow that occurred in the hour. All data recorded in December so possible snow time. From wunderground website, when looked for the amount of snow and temp 12 .No heavy snow is recorded, hence not processing the recorded data

Column 5: Clouds

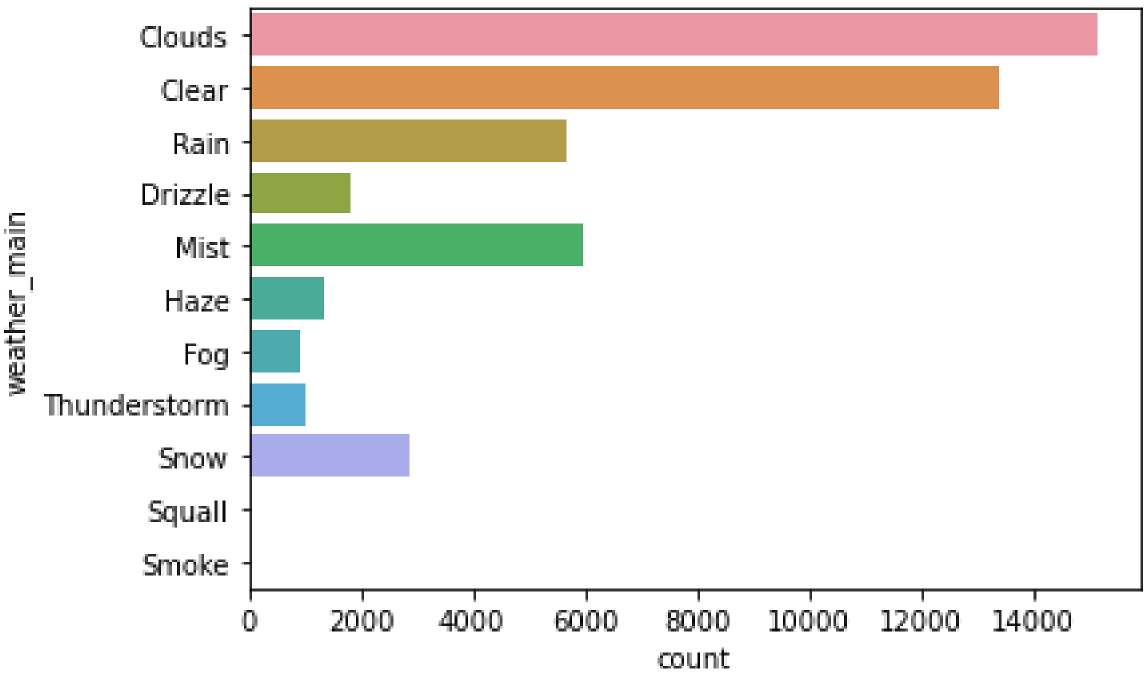




Observation:

Nothing abnormal identified, this column shows percentage of cloud cover and is a categorical variable.

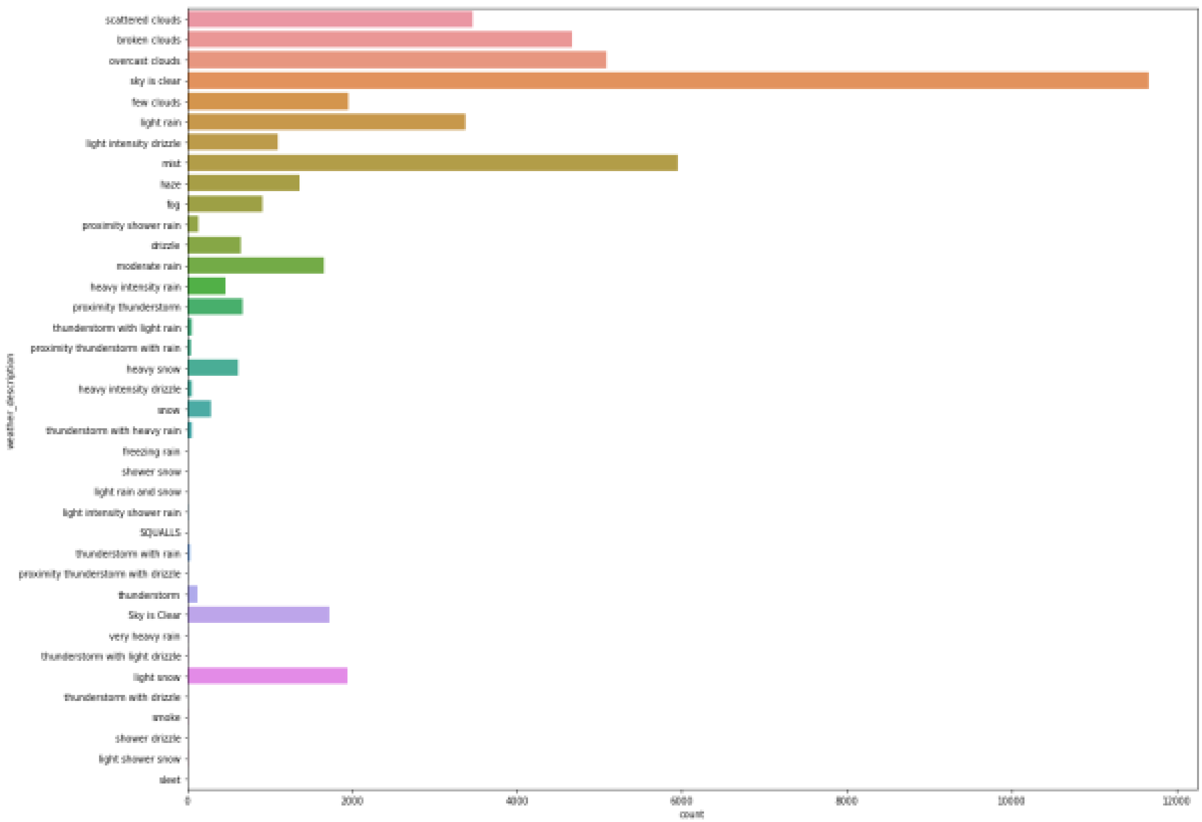
Column 6: Weather – Short note



```
traffic['weather_main'].value_counts()
```

```
Clouds          15164
Clear           13391
Mist             5950
Rain             5672
Snow             2876
Drizzle          1821
Haze             1360
Thunderstorm     1034
Fog               912
Smoke              20
Squall              4
Name: weather_main, dtype: int64
```

Column 7: Weather – Description



Observation:

Sky is clear for most of the days.

Column 8: Date and Time

```
# min date recorded in the dataset
traffic.date_time.min()
```

```
'2012-10-02 09:00:00'
```

```
#max date recorded in the dataset
traffic.date_time.max()
```

```
'2018-09-30 23:00:00'
```

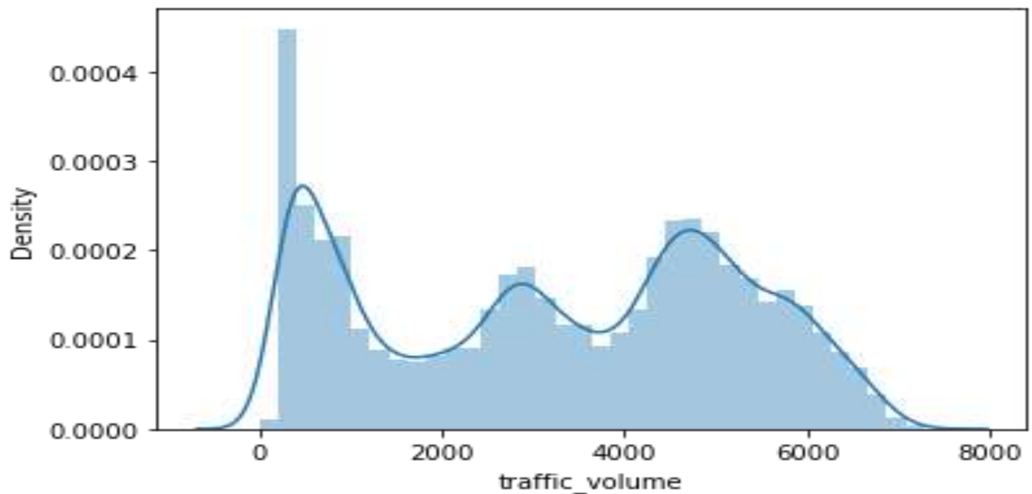
```
traffic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48204 entries, 0 to 48203
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   holiday              48204 non-null  object
 1   temp                 48204 non-null  float64
 2   rain_1h              48204 non-null  float64
 3   snow_1h              48204 non-null  float64
 4   clouds_all           48204 non-null  int64
 5   weather_main         48204 non-null  object
 6   weather_description  48204 non-null  object
 7   date_time            48204 non-null  object
 8   traffic_volume       48204 non-null  int64
dtypes: float64(3), int64(2), object(4)
memory usage: 3.3+ MB
```

Observation:

Traffic data is recorded from 2nd October 2012 to 30th September 2018, datatype of the column is object type.

Column 9: Traffic Volume

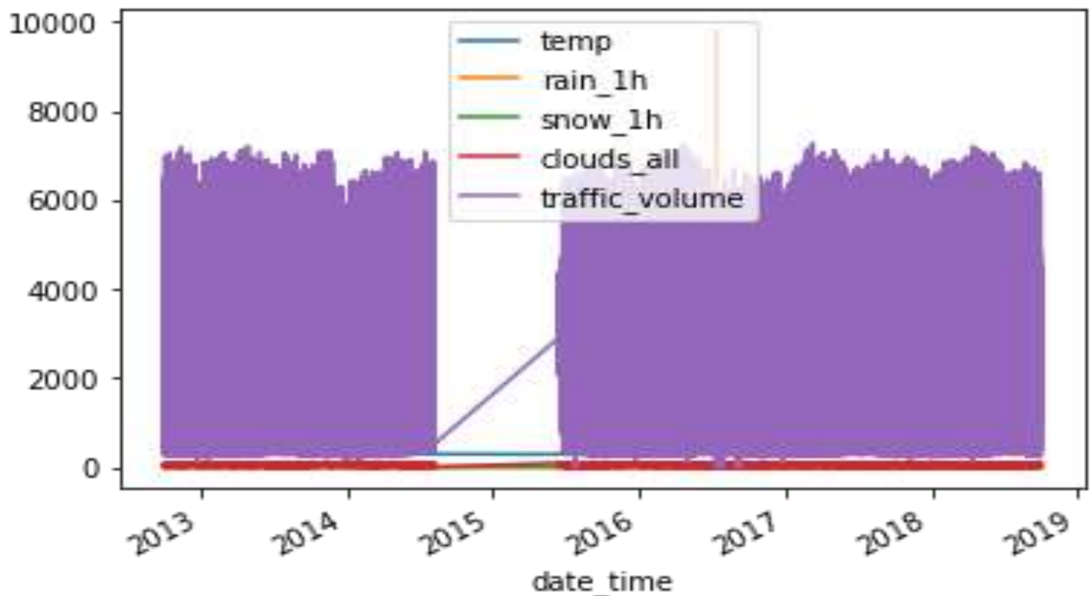


	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
0	None	288.28	0.0	0.0	40	Clouds	scattered clouds	2012-10-02 09:00:00	5545
1	None	289.36	0.0	0.0	75	Clouds	broken clouds	2012-10-02 10:00:00	4516
2	None	289.58	0.0	0.0	90	Clouds	overcast clouds	2012-10-02 11:00:00	4767
3	None	290.13	0.0	0.0	90	Clouds	overcast clouds	2012-10-02 12:00:00	5026
4	None	291.14	0.0	0.0	75	Clouds	broken clouds	2012-10-02 13:00:00	4918

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume
48199	None	283.45	0.0	0.0	75	Clouds	broken clouds	2018-09-30 19:00:00	3543
48200	None	282.76	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 20:00:00	2781
48201	None	282.73	0.0	0.0	90	Thunderstorm	proximity thunderstorm	2018-09-30 21:00:00	2159
48202	None	282.09	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 22:00:00	1450
48203	None	282.12	0.0	0.0	90	Clouds	overcast clouds	2018-09-30 23:00:00	954

4.2. Multivariate Analysis

We are analysing all Columns of the dataset against the datetime column.




```
# checking if there is any data present in between 9th aug 2014 and 10th june 2015 ( missing data)
traffic.loc[(traffic['date_time'] >= '2014-08-09') & (traffic['date_time'] <= '2015-06-10')]
```

holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume

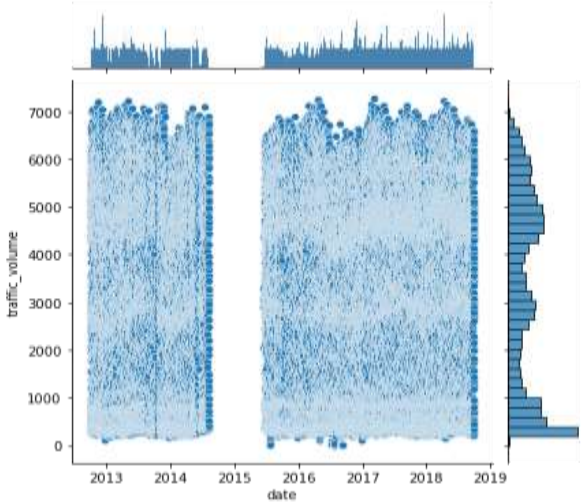
Observation:

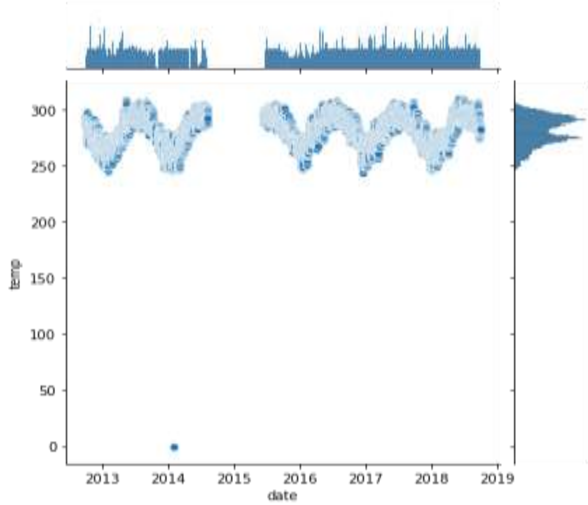
Data has not been recorded from 9th Aug 2014 to 10th June 2015, (period of 10 months), training machine learning model with this missing data might result in less performance.

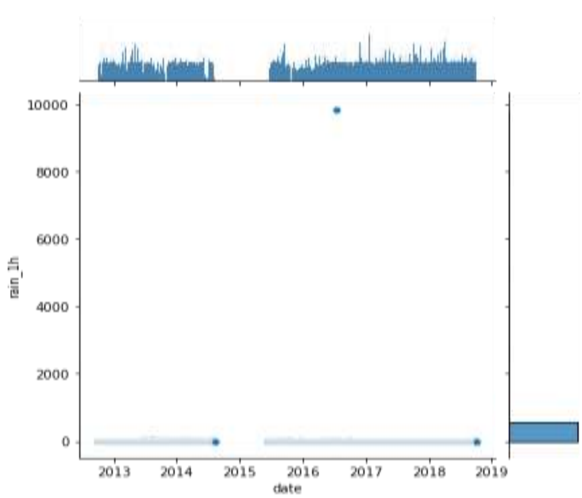
Trial and Error:

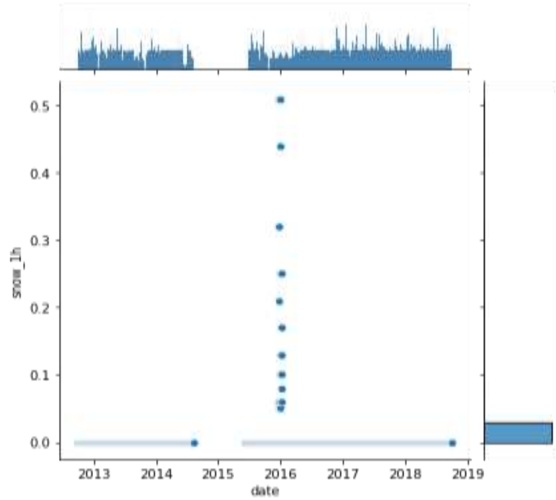
Cannot comment on handling this missing data, only after performing the data pre-processing and then manually checking the model accuracy with different test and train data splits and by filling these missing values.

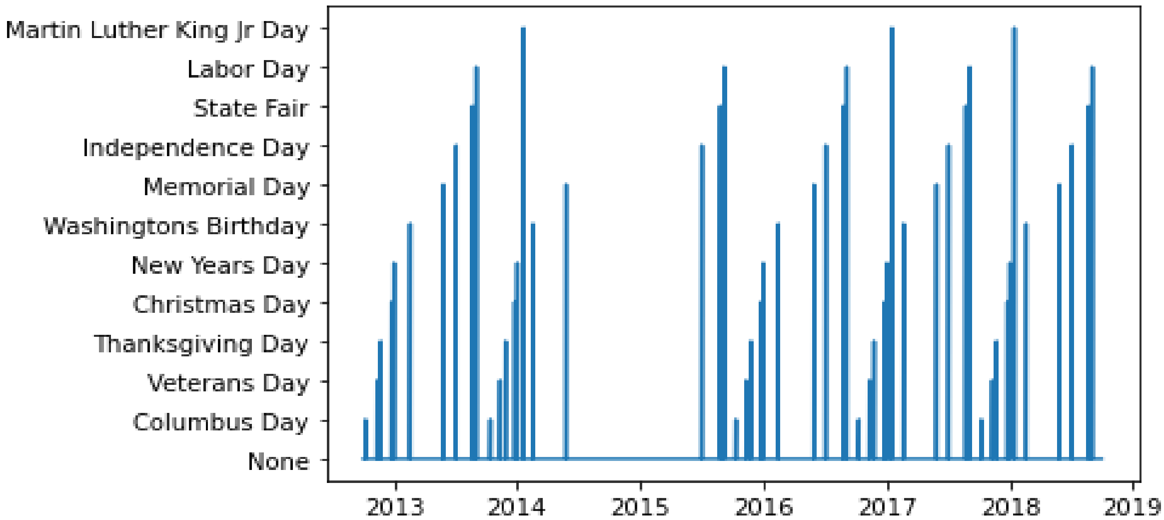
Plots of individual columns against date_time column,

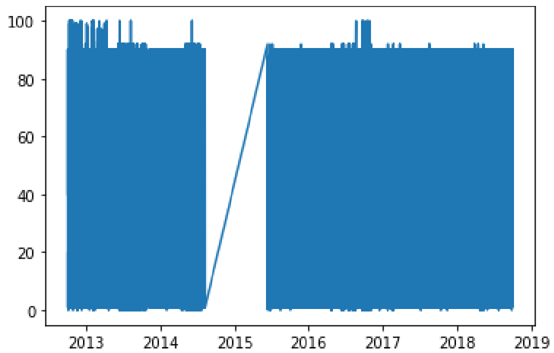


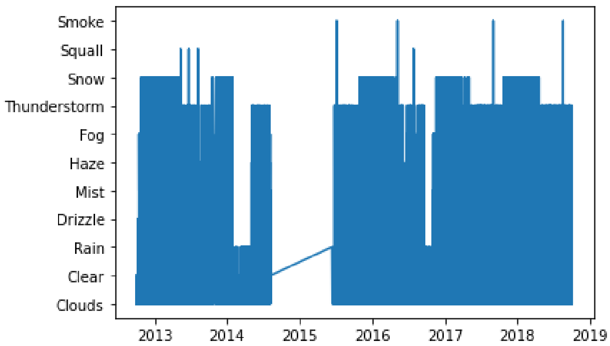




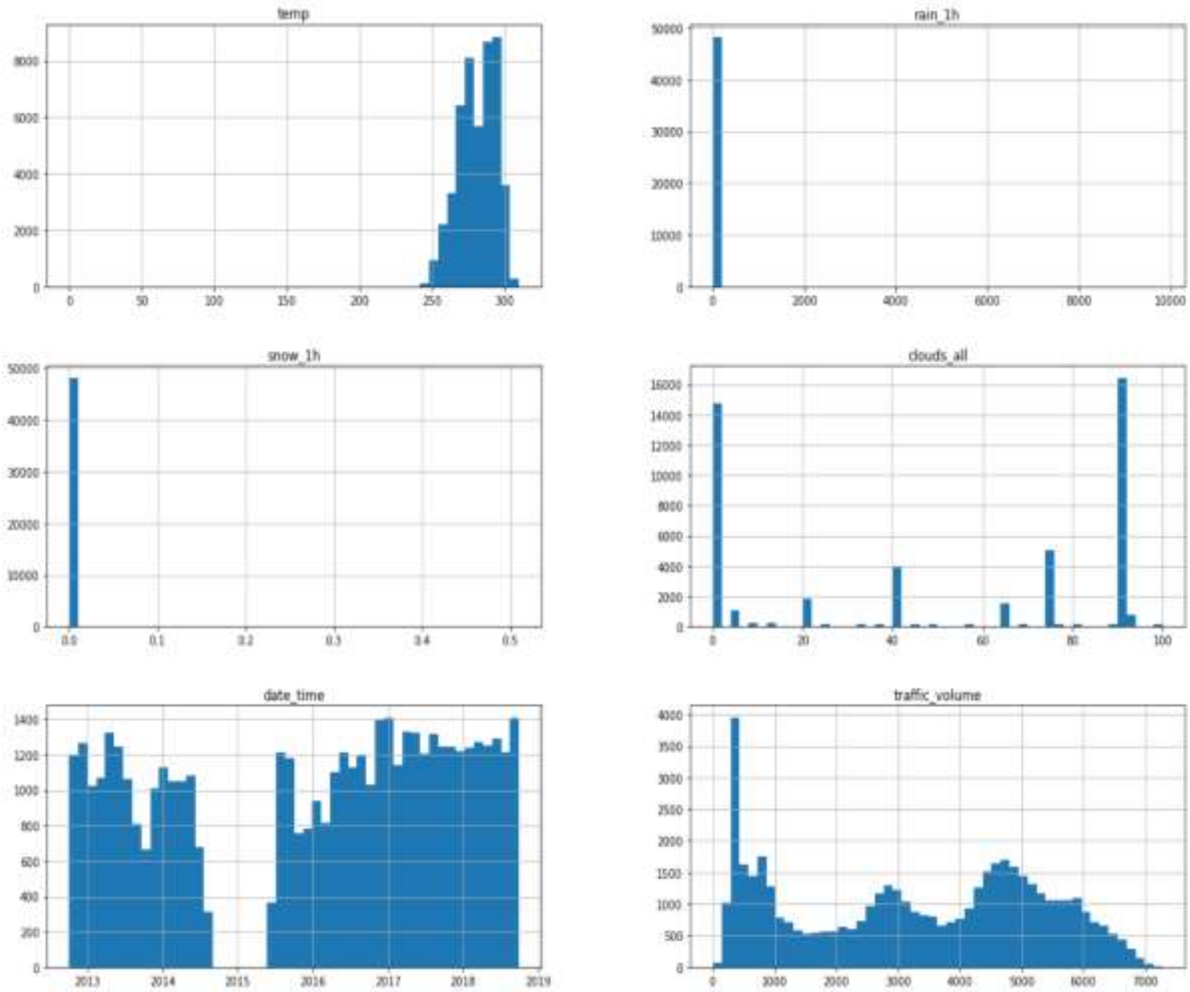




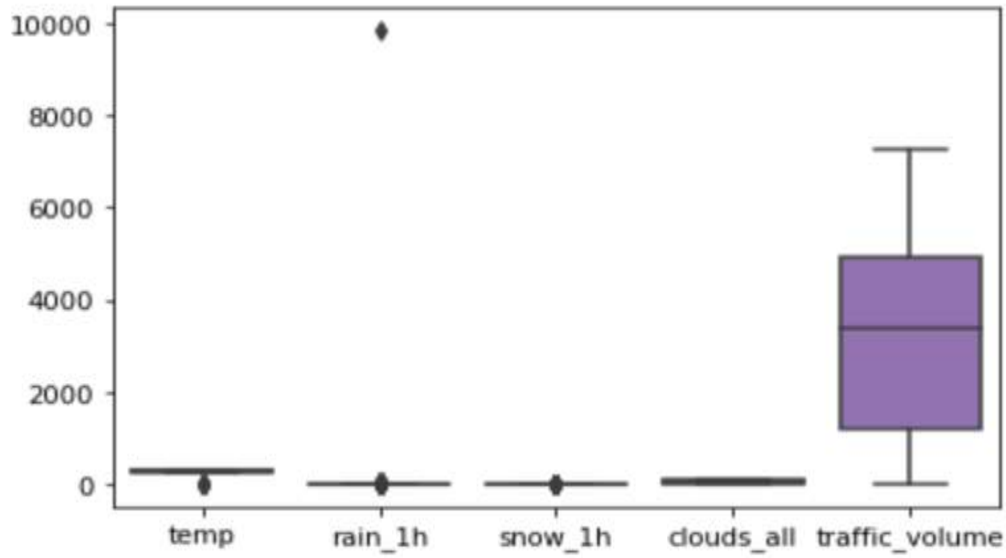




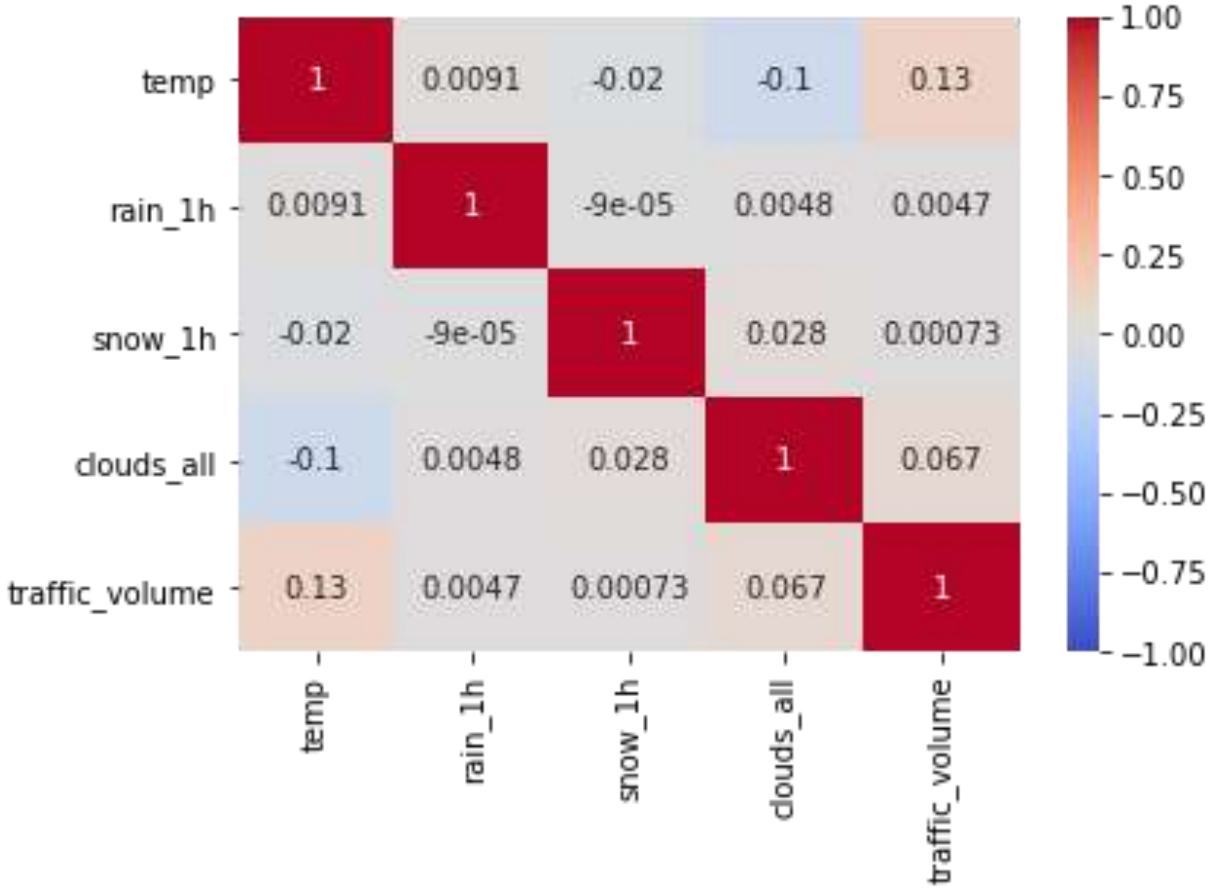

```
traffic.hist(bins=50, figsize=(20,15))
plt.show()
```



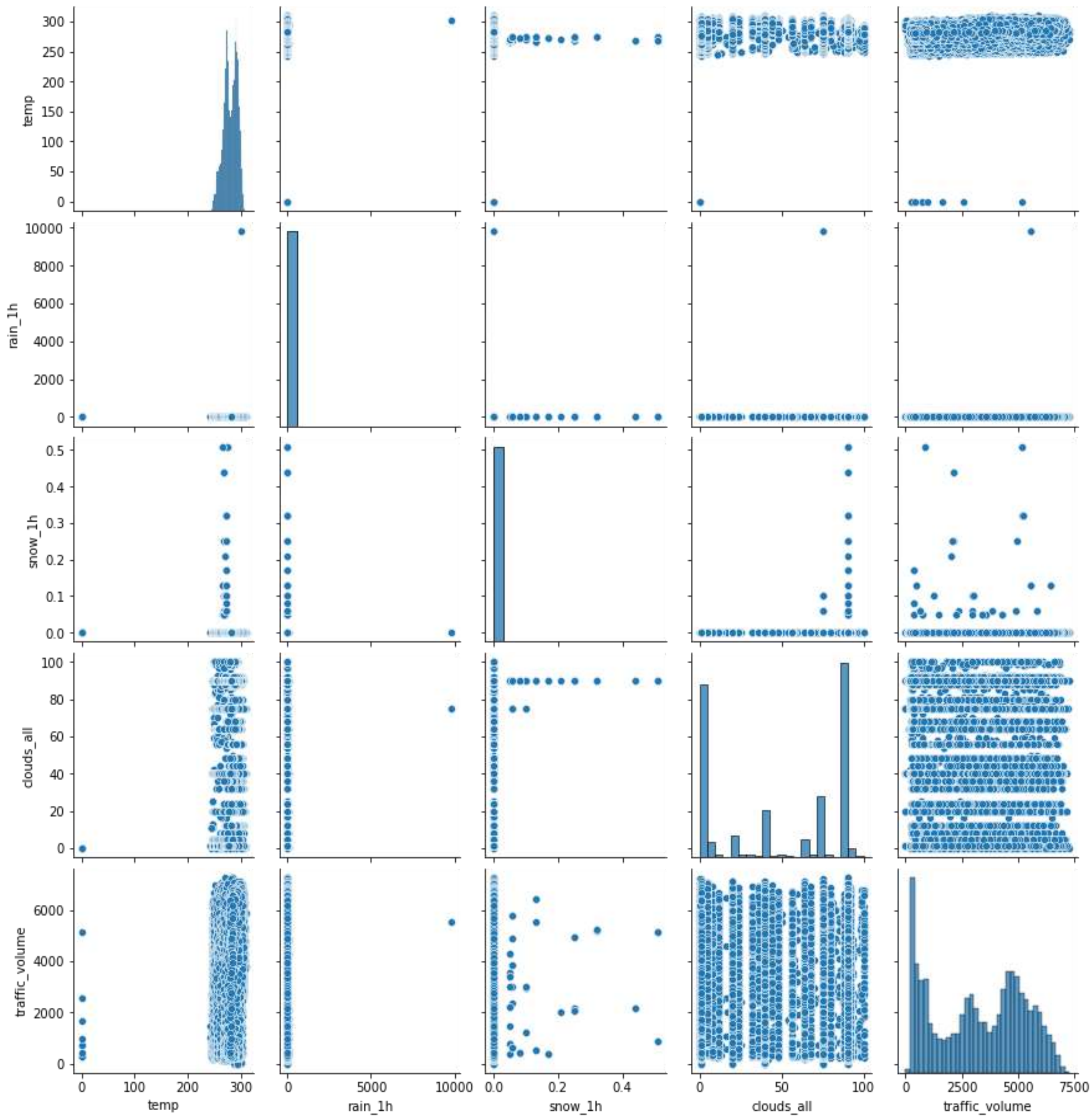
```python
#box plot for all the columns to check outliers
ax = sns.boxplot(data=traffic)
```



Pearson's heat map.



Pair plot to understand the correlation between columns



<u>Observations</u>:

1. Holiday column - US National holidays plus regional holiday, Minnesota State Fair
   Categorical - String
   Replace None with "regular day", other days as Holidays
   Regular days = 48143,
   Holiday count = 61 days,

2. Temperature Column - Average temp in kelvin
   Numeric - Float (Continuous)
   Covert temp from Kelvin to Celsius
   Data missing from Sep 2014 to May 2015
   Observed Outlier

3. Rain Column - Amount in mm of rain that occurred in the hour
   Numeric - Float (Continuous)
   Data missing from Sep 2014 to May 2015
   Observed Outlier

4. Snow Column - Amount in mm of snow that occurred in the hour
   Numeric - Float (Continuous)
   Data missing from Sep 2014 to May 2015
   Observed Outlier

5. Clouds Column - Percentage of cloud cover
   Numeric - int (Continuous)
   Data missing from Sep 2014 to May 2015

6. Weather Main - Short textual description of the current weather
   Categorical - String
   Data missing from Sep 2014 to May 2015

7. Weather Description - Longer textual description of the current weather
   Categorical - String
   Data missing from Sep 2014 to May 2015

8. Date and Time Column - Hour of the data collected in local CST time
   Time Series - Date Time
   Change data type from object to datatime64[ns]

9. Traffic Volume Column - Hourly I-94 ATR 301 reported westbound traffic volume
   Numeric - int64 (Discrete)
   Data missing from Sep 2014 to May 2015

5. Research Question

This research aims to analyse the traffic volume trend and build a machine learning model with higher accuracy to predict the traffic volume for 12 months. Since the dataset consists of dependent and independent variables, regression analysis is chosen to be performed. Also, there is a date and time column as the traffic volume is recorded on an hourly basis time-series analysis is also performed.

This analysis can further understand pollution, air quality information, highway maintenance and make better decisions regarding road works and closures. This traffic volume indirectly represents the town's growing or fading popularity, and better-informed town planning decisions can be made. Also, the steadiness of the business economy in the town could be analysed.

5.1. Feature Selection

5.1.1. Time-Series Analysis

The traffic dataset also contains a time series variable, 'date_time'. This research focuses on univariate Time-series analysis, where the two variables are traffic_volume and time series itself to forecast data for the coming 12 months.[2]

Target: this study aims to forecast the traffic volume with the past data, so the traffic_volume is out output variable.
Labels: For univariate time series, there are only two columns in the dataset, i.e., output variable traffic_volume and the timeseries itself.

5.1.2. Regression Analysis

The target variable is "traffic_volume" (discrete numerical data), a dependant variable and other columns holiday, temperature, rain, and snow are independent variables. We perform a regression analysis to identify the relationship between these dependent and independent variables and build our model. This model predicts the traffic volume information for the given independent variables(features).[3]

Target: this study aims to forecast the traffic volume with the past data, so the traffic_volume is out output variable.
Labels: For the regression analysis all the continuous variables are considered to predict the traffic_volume.

6. Data pre-processing

Variable types	Dataset consists of Categorical, Numerical and Timeseries variables.
Accuracy	The data is mostly correct, except some outliers in rain and temp columns.
Completeness	No null values, the data is missing between 2014-08-09 and 2015-06-10.
Consistency	Data is consistent, when verified for weather info from other online sources.
Timeliness	Data is appropriately recorded.
Believability	Data is genuine and is obtained from UCI machine learning repository.
Interpretability	Data is easily mapped with numeric scales, and all features are well described.

6.1. Replacing column values

The holiday column values are converted from categorical to numerical values. If the values are None replaced with 1, else replaced with 0.

6.2. Convert data type

Converted temperature from Kelvin to Celsius and changed the data type of date_time column to Date data type.

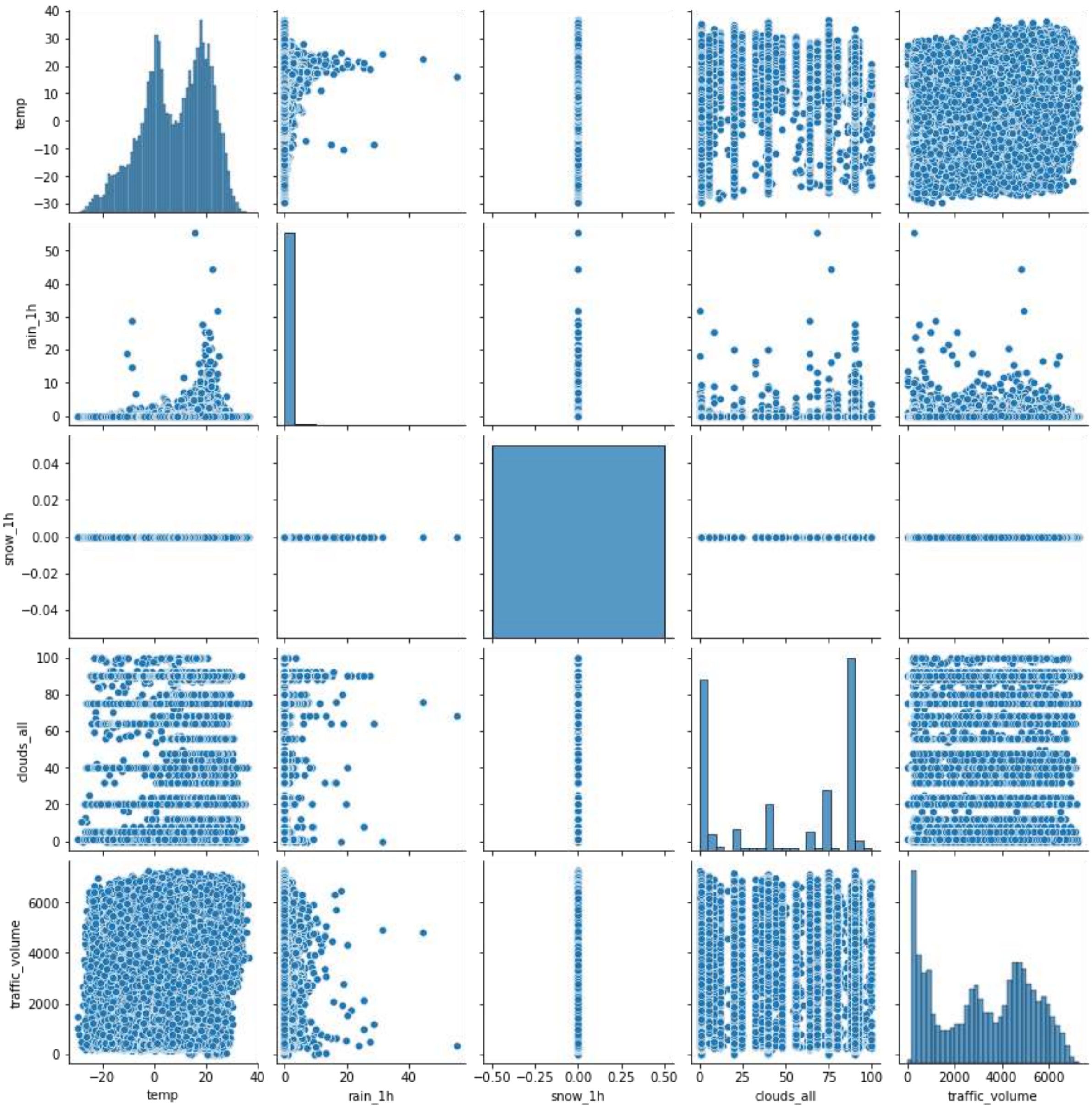
6.3. Handling/ Removing Outliers

the outlier data from the temp column and rain column are removed after cross verifying the weather data from other online sources.

```
# check the data type of the columns
traffic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48130 entries, 0 to 48203
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   holiday              48130 non-null  object
 1   temp                 48130 non-null  float64
 2   rain_1h              48130 non-null  float64
 3   snow_1h              48130 non-null  float64
 4   clouds_all           48130 non-null  int64
 5   weather_main         48130 non-null  object
 6   weather_description  48130 non-null  object
 7   date_time            48130 non-null  datetime64[ns]
 8   traffic_volume       48130 non-null  int64
```

Pair plot after data pre-processing



7. Model and Algorithm Selection

7.1. Timeseries Analysis

7.1.1. Data Processing

Interpolate

The dataset contains missing data, these missing values result in lower accuracy and Holt-Winters, and ARIMA models cannot be applied to the discontinuous series. So, these missing values can be filled using the "**interpolate**" method and "**ffill**" methods.

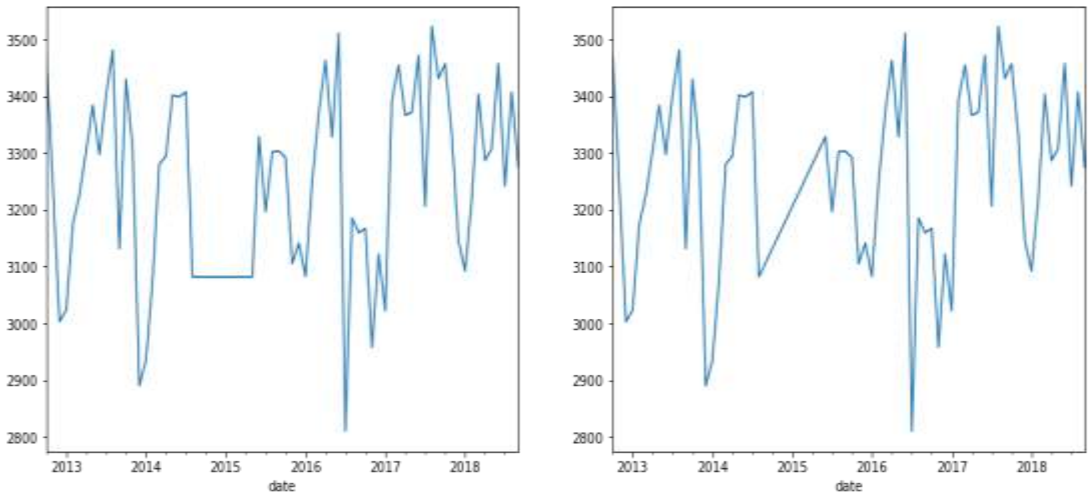
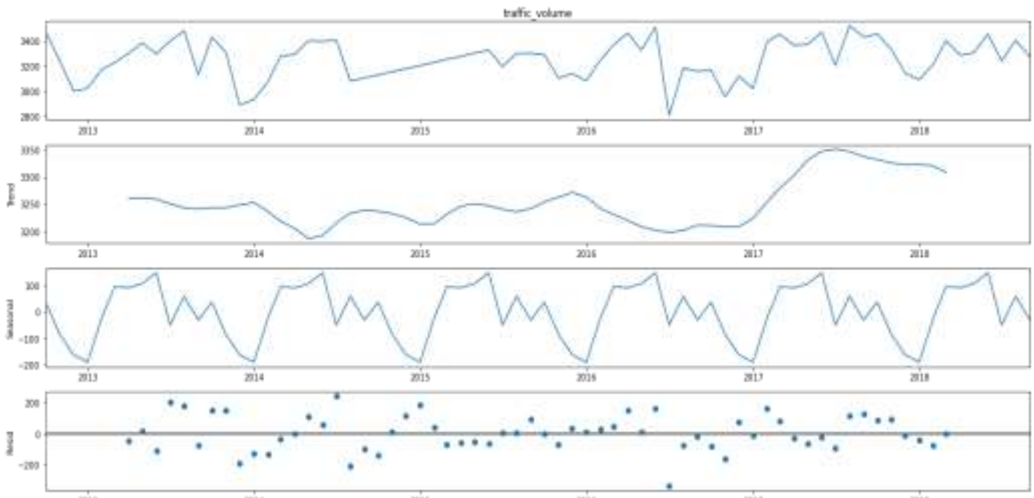


Figure 5 ffill method (Left), interpolate (right)

On performing experiments with various methods, interpolate method had higher accuracy than the ffill method because the ffill method fills the missing values with the last value in the series. In contrast, the interpolate applies linear interpolation to the missing values.

Decomposition

Also, Time series decomposition gives a clear breakdown of its components trend, seasonality irregularity and cyclicity.



The overall trend of the data is almost horizontal until 2017 and has a rising trend later. The seasonal component is present too. The seasonality of the series should also be verified before applying machine learning models. [4]

The time series curve is said to be stationary if it has a constant mean, constant variance and if the covariance is independent of time. To determine the stationarity Augmented Dickey-Fuller test and KPSS (Kwiatkowski-Phillips-Schmidt-Shin) Test. Results of these tests show series is not stationary.

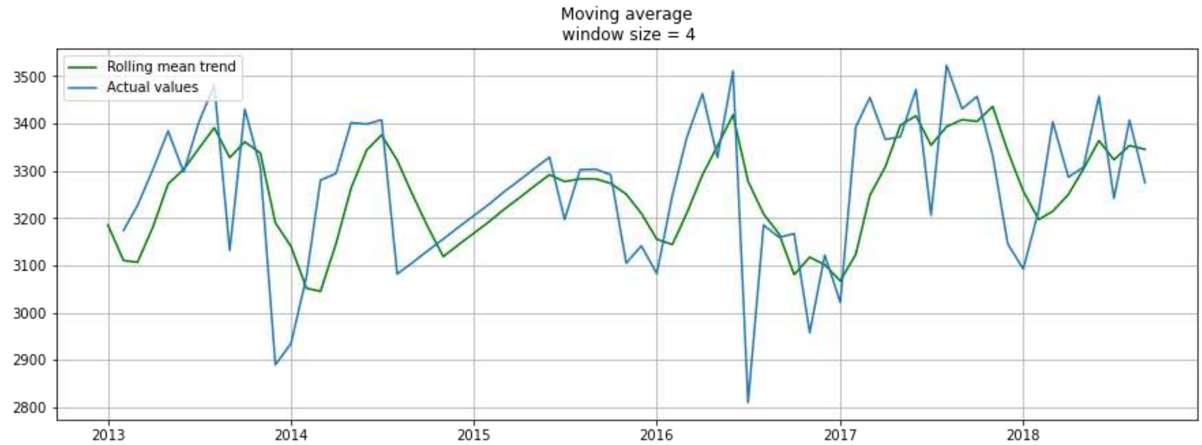
```
ADF Statistic: -5.857644
p-value: 0.000000
Critical Values:
	1%: -3.526
	5%: -2.903
	10%: -2.589
```

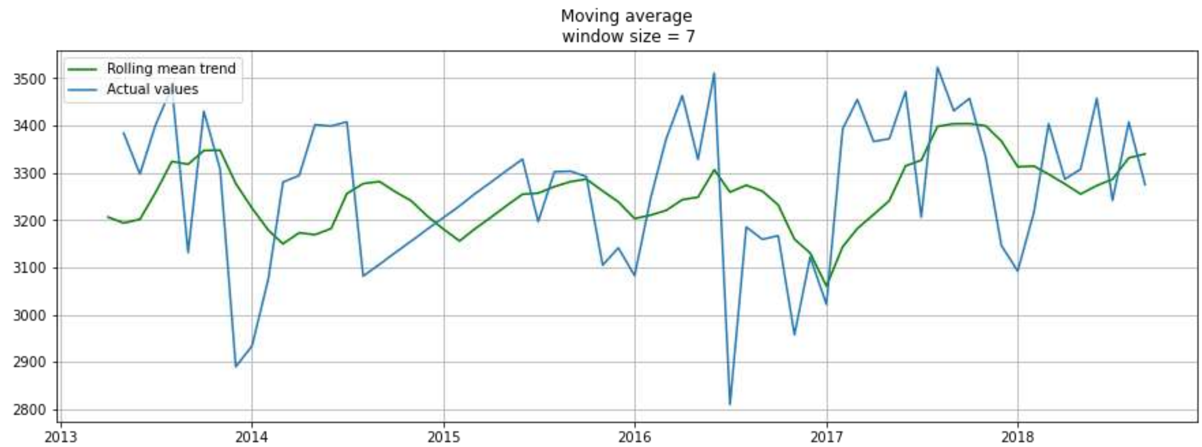
```
Results of KPSS Test:
Test Statistic            0.295293
p-value                   0.100000
Lags Used                12.000000
Critical Value (10%)      0.347000
Critical Value (5%)       0.463000
Critical Value (2.5%)     0.574000
Critical Value (1%)       0.739000
```

ADF test shows the series is stationary, but KPSS shows the series is not stationary since the p-value is not <=0.05, so the series is differentiated twice to achieve stationarity.

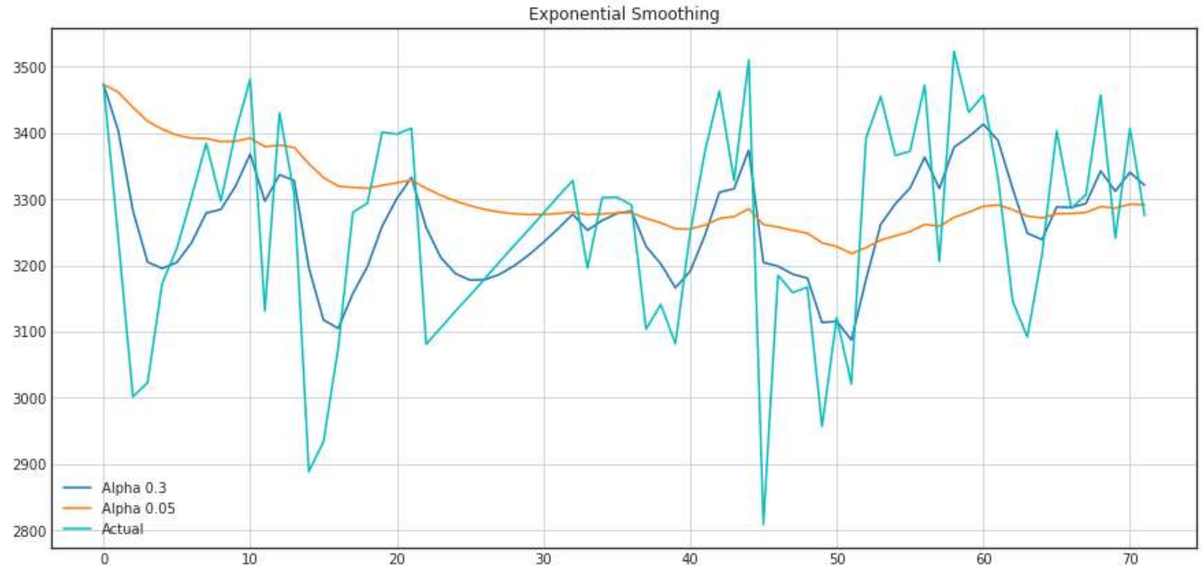
7.1.2. Experiments – Time Series

A. Moving Average

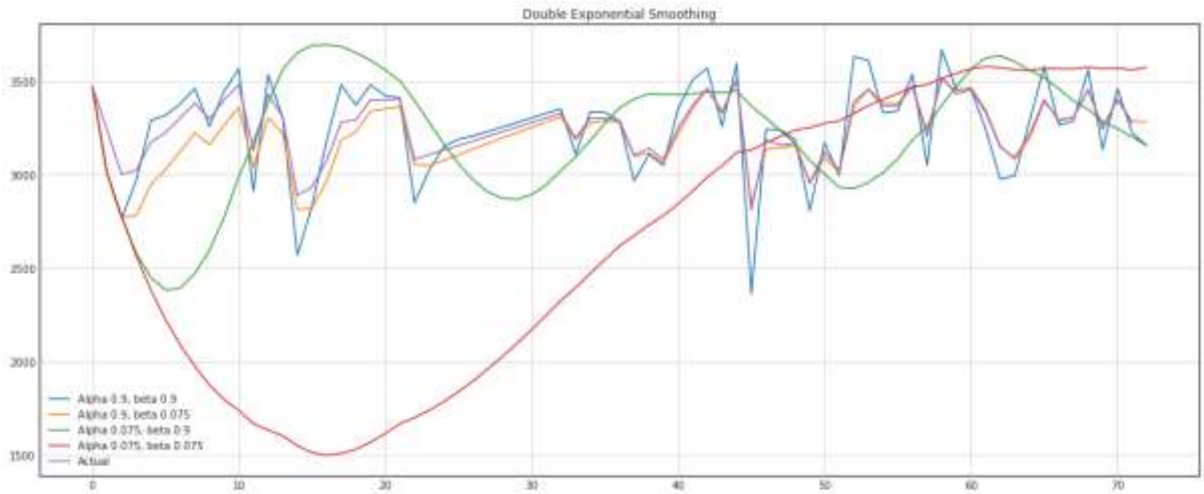




B. Weighted Average (Exponential Smoothing)[5]



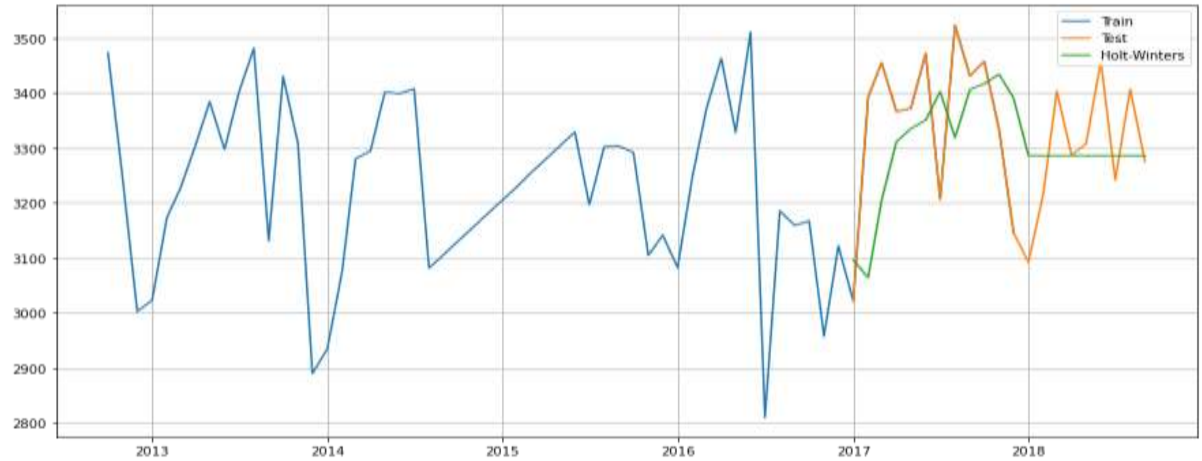
C. Double Exponential Smoothing



<u>Observations:</u>

From the above three methods, moving average, Weighted average, and exponential smoothing with various values of alpha and beta, the traffic volume curve has a trend the moving average accuracy is not great looking at the plots. Also, with exponential and double exponential smoothing efficient forecast model cannot be built because the traffic volume curve also contains seasonality. (Observed from the timeseries decomposition)

D. Unsupervised – Holt-winters

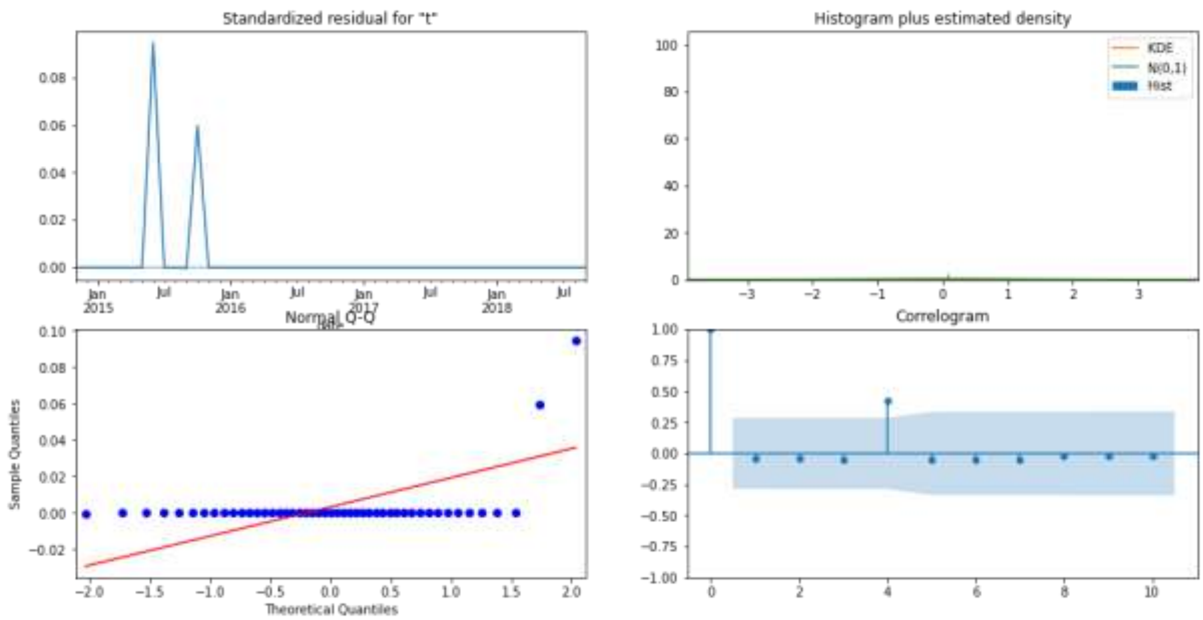



```
Unsupervised - without missing values
Holt-Winters
----------------
MAE : 115.67
MSE : 21468.11
RMSE  : 146.520002
R2_SCORE  : -0.249372
```

Observations:

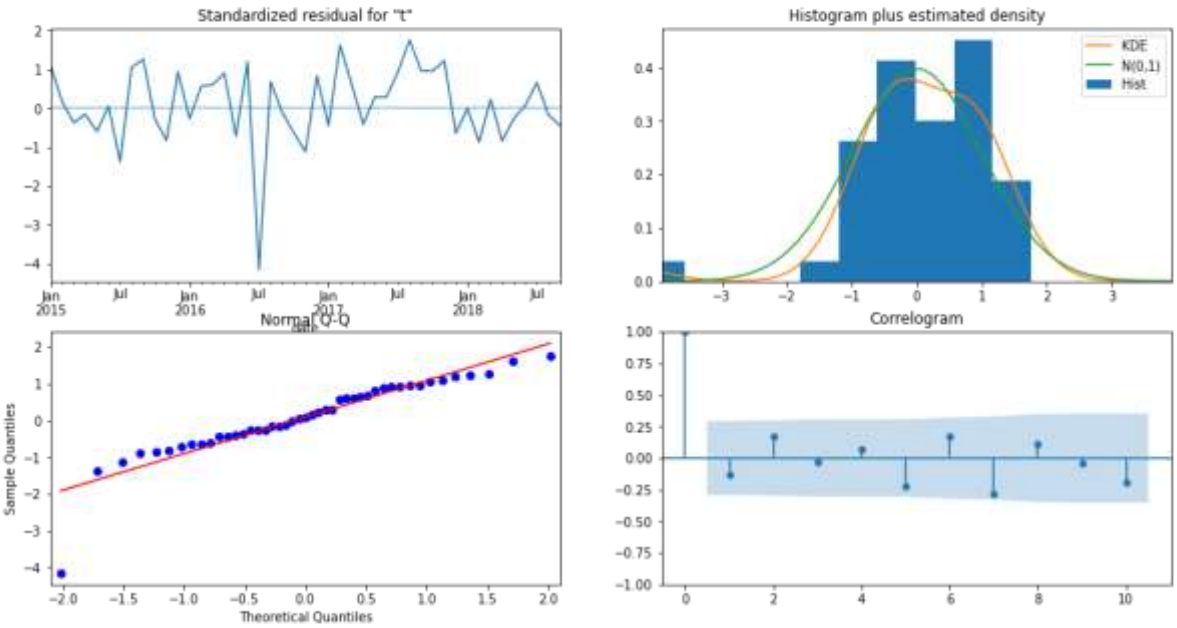
The coefficient of determination r2_score is -0.24, this unsupervised model has negative score, which indicate this could result in worse predictions arbitrarily. Unsupervised models are used in real time data and tend to have lower accuracies, these models are mostly applied for association and clustering analysis.

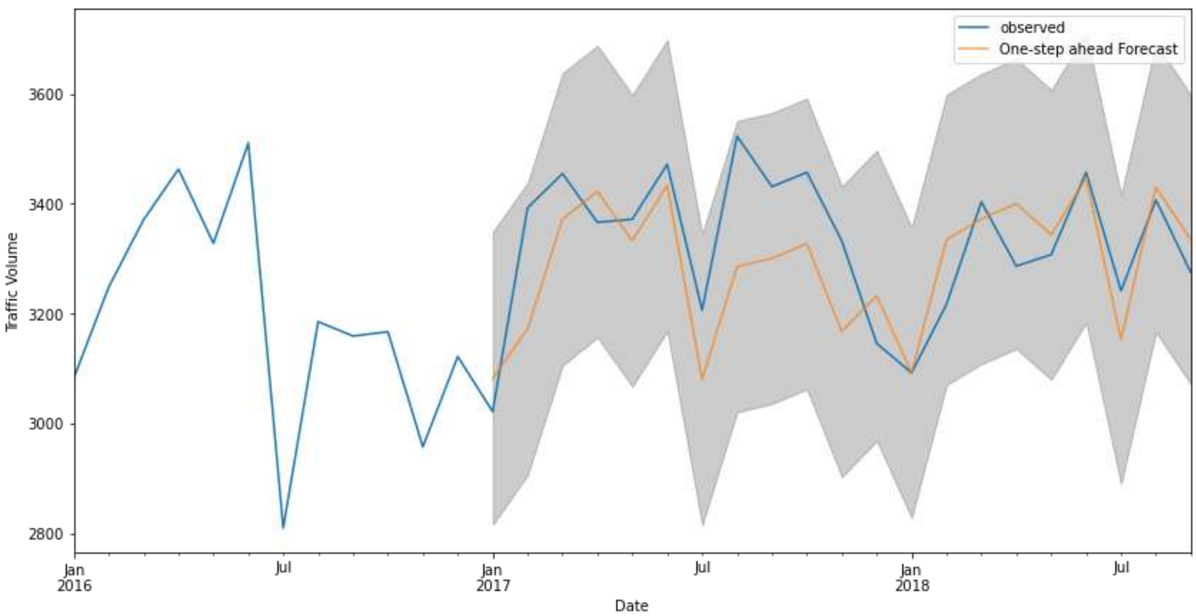
E. Unsupervised – ARIMA[6]



Observations:

Time series with missing values has less accuracy and has skewed predictions. so, the timeseries data filled with interpolate for the missing traffic volume information might improve prediction accuracy.






```
Unsupervised - without missing values
ARIMA(1, 1, 1)x(0, 1, 1, 12)12
---------------------------------
MAE : 88.27
MSE : 11781.93
RMSE  : 108.544611
R2_SCORE  : 0.314331
```

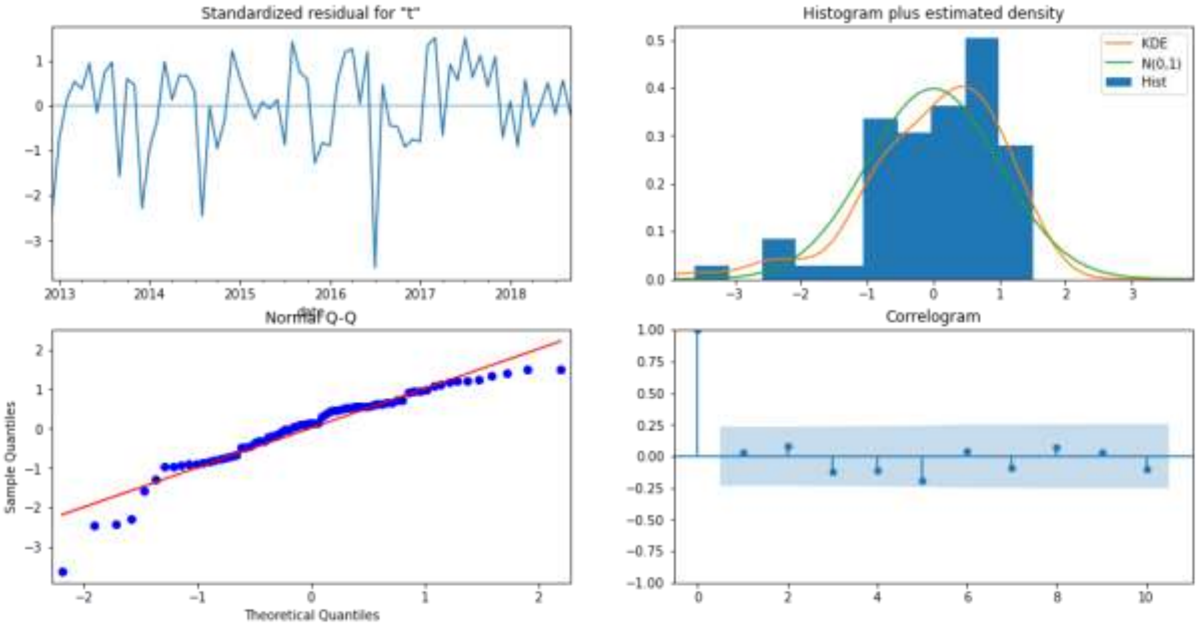
Observations:

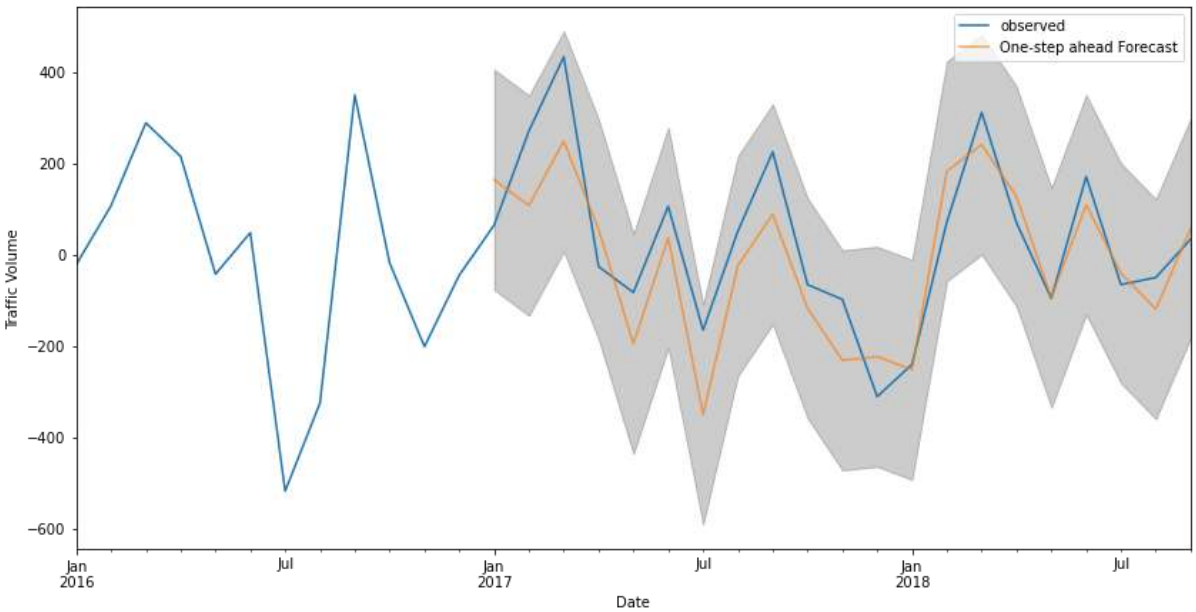
Data is skewed, there is abnormal peak in the residual mid of 2015. Possible reasons could be the bias introduced by interpolate method to fill the missing time series values and outliers. Since the unsupervised model applied data is not processed before applying the algorithm. Hence the accuracy is only 31%. A higher accuracy can be achieved by applying supervised learning methods.

To improve Forecast Accuracy:

1. We can apply Supervised Learning with train and test split and achieve higher accuracy.
2. To apply a time series forecast, we should first check the stationarity.

F. Supervised - ARIMA





```
Supervised Learning- without missing values
ARIMA(2, 0, 2)x(0, 0, 3, 12)12
---------------------------------
MAE : 86.36
MSE : 10107.47
RMSE  : 100.535892
R2_SCORE  : 0.683024
```

Observations:

By applying the supervised ARIMA with seasonal component and by differentiating the time series to make it stationary, resulted in better accuracy of 68%. As the series also has a seasonal component the values P, D and Q are predicted first based on the AIC value. Lower the AIC value more appropriate the P, D and Q are. Accuracy of 68% is achieved with ARIMA (2,0,2)(0,0,3,12)12 with AIC=412.9 .

7.2. Regression Analysis

Regression Analysis is applied to the dataset only if linear relationship between the independent and dependent variables exist. Also, the observations must be independent of each other in the dataset, and the output variable must be normally distributed for the fixed input variables. Since the traffic volume dataset does not have a strong linear relationship between the target and the output variables, low accuracy is predicted with the machine algorithms. [3]

7.2.1. Data Processing

Data Encoding

Machine learning algorithms are complex math functions built together. These accept only numerical data inputs, so we convert our essential features to numeric from categorical variable to apply these algorithms.

```
# visualising top 6 rows od the dataframe
traffic.head()
```

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	date_time	traffic_volume	date
0	1	15.13	0.0	0.0	40	Clouds	scattered clouds	2012-10-02 09:00:00	5545	2012-10-02
1	1	16.21	0.0	0.0	75	Clouds	broken clouds	2012-10-02 10:00:00	4516	2012-10-02
2	1	16.43	0.0	0.0	90	Clouds	overcast clouds	2012-10-02 11:00:00	4767	2012-10-02
3	1	16.98	0.0	0.0	90	Clouds	overcast clouds	2012-10-02 12:00:00	5026	2012-10-02
4	1	17.99	0.0	0.0	75	Clouds	broken clouds	2012-10-02 13:00:00	4918	2012-10-02

	holiday	temp	rain_1h	snow_1h	clouds_all	time	traffic_volume	year	month	day	weekday
0	1	15.13	0.0	0.0	40	9	5545	2012	10	2	2
1	1	16.21	0.0	0.0	75	10	4516	2012	10	2	2
2	1	16.43	0.0	0.0	90	11	4767	2012	10	2	2
3	1	16.98	0.0	0.0	90	12	5026	2012	10	2	2
4	1	17.99	0.0	0.0	75	13	4918	2012	10	2	2
5	1	18.57	0.0	0.0	1	14	5181	2012	10	2	2
6	1	20.02	0.0	0.0	1	15	5584	2012	10	2	2
7	1	20.71	0.0	0.0	1	16	6015	2012	10	2	2
8	1	20.99	0.0	0.0	20	17	5791	2012	10	2	2
9	1	19.95	0.0	0.0	20	18	4770	2012	10	2	2

Data Scaling

Since the data is represented in different magnitudes, we normalise the scales using standard scalar from sklearn to perform data processing.

	holiday	temp	rain_1h	snow_1h	clouds_all	time	traffic_volume	year	month	day	weekday
0	0.035623	0.550906	-0.1299	0.0	-0.238914	-0.345808	1.149803	-1.854341	1.029048	-1.575065	-0.493685
1	0.035623	0.635865	-0.1299	0.0	0.658234	-0.201715	0.631880	-1.854341	1.029048	-1.575065	-0.493685
2	0.035623	0.653172	-0.1299	0.0	1.042725	-0.057622	0.758215	-1.854341	1.029048	-1.575065	-0.493685
3	0.035623	0.696438	-0.1299	0.0	1.042725	0.086471	0.888576	-1.854341	1.029048	-1.575065	-0.493685
4	0.035623	0.775890	-0.1299	0.0	0.658234	0.230564	0.834217	-1.854341	1.029048	-1.575065	-0.493685

7.2.2. Experiments – Regression Analysis

A. Test and Train Data split

The entire traffic dataset is split into the training set and testing set using the train_test_split package from sklearn. The data split ratio is 80% train and 20% test the data.

```
from sklearn import model_selection

#split data as training and testing set 80% and 20% respectively
from sklearn.model_selection import train_test_split

ft_train, ft_test, lb_train, lb_test = train_test_split(data , target, test_size=0.20, random_state = 2)
display(ft_train.shape,ft_test.shape)
```

```
(38504, 9)

(9626, 9)
```

B. Multiple Linear Regression

Training the multiple linear regressor[7] from the sklearn package and the stats models yields similar traffic data accuracy. Data is skewed like unsupervised ARIMA model, although this cannot be compared because ARIMA is applied on time-series and the traffic volume columns only, whereas linear regressor is applied on all the columns of the dataset (this might result in lower accuracy also the linear relationship between the features and labels is not stronger as reflected from correlation heatmap)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         traffic_volume   R-squared:                       0.144
Model:                            OLS   Adj. R-squared:                  0.144
Method:                 Least Squares   F-statistic:                     718.1
Date:                Sun, 11 Apr 2021   Prob (F-statistic):               0.00
Time:                        13:36:14   Log-Likelihood:                -51633.
No. Observations:               38504   AIC:                         1.033e+05
Df Residuals:                   38494   BIC:                         1.034e+05
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0002      0.005      0.044      0.965      -0.009       0.009
x1             0.1272      0.004     30.681      0.000       0.119       0.135
x2             0.0108      0.004      2.500      0.012       0.002       0.019
x3            -0.2191      0.005    -48.435      0.000      -0.228      -0.210
x4            -0.1620      0.005    -35.273      0.000      -0.171      -0.153
x5             0.0160      0.005      3.425      0.001       0.007       0.025
x6            -0.0493      0.005    -10.428      0.000      -0.059      -0.040
x7             0.1993      0.005     41.123      0.000       0.190       0.209
x8             0.0465      0.005      9.024      0.000       0.036       0.057
x9            -0.0337      0.006     -5.620      0.000      -0.046      -0.022
==============================================================================
Omnibus:                    19641.360   Durbin-Watson:                   2.004
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2247.760
Skew:                           0.176   Prob(JB):                         0.00
Kurtosis:                       1.870   Cond. No.                         1.45
==============================================================================
```

```
Multiple Linear Regression
------------------------------
MAE : 0.82
MSE : 0.86
RMSE  : 0.925102
R2_SCORE  : 0.146603
```

<u>Observations:</u>

Accuracy of the linear regression model is 14%, it is very low because there is no stronger linear correlation between the target variable (traffic_volume) and the input variables (holiday, week, day, month, year, snow, rain, temp, weather, and clouds information)

C. Support Vector Regressor

SVR [8] algorithm identifies the non-linearity in the dataset and provides a prediction model with greater efficiency and accuracy than the multiple linear regressor. Now, the machine learning model is built on the training dataset and accuracy is predicted against the testing data set to split.

Fine-tuning the model with various epsilon values, the higher accuracy obtained with this model is 78.99%, epsilon defines the tolerance margin. Larger values of epsilon introduce more significant error in the prediction model. Similarly, if epsilon is 0, then every erroneous prediction in the model might have many support vectors to sustain that. From few experiments, the following is observed for different values of epsilon. This algorithm did not affect the accuracy with the ffill or interpolate method used to fill missing values.

```
SVR: Accuracy is: 0.7754497284824359 C=10, epsilon=0.05
SVR: Accuracy is: 0.7862211864782305 c=10, epsilon=0.1
SVR: Accuracy is: 0.7885247983876983 C=10, epsilon=0.3
SVR: Accuracy is: 0.7899783329541525 C=10, epsilon=0.4
SVR: Accuracy is: 0.789005715668167 C=10, epsilon=0.45
SVR: Accuracy is: 0.7864581179705342 C=10, epsilon=0.5
SVR: Accuracy is: 0.7294206288648982 C=10, epsilon=0.9
```

```
Support Vector Regressor
------------------------------
Accuracy : 0.7883733013567891
MAE : 0.35
MSE : 0.21
RMSE  : 0.460680
R2_SCORE  : 0.788373
```

Observations:

SVR model has better accuracy compared to multiple linear regressor, because this model acknowledges the linearity in the data and with optimal values of epsilon=0.4 has resulted in 78.99% accuracy. As the value of epsilon away from 0.4 the accuracy of the model comes down.

D. KNN Regressor

KNN algorithm is a mathematical algorithm resembling the association between the input and the output variables by calculating the averages of the observations in the same neighbourhood values.[9] KNN is also a non-parametric method, so considering the neighbourhood value is critical. In the SVR model, moving away from epsilon decreases the accuracy. The significantly less or very high number of neighbours have a similar impact on accuracy. The ffill method had higher accuracy with n=2 84.8%, but with the interpolating, the accuracy dropped to 82.4%

```
KNeighborsRegressor: n = 1, Accuracy is: 0.7916049845276343
KNeighborsRegressor: n = 2, Accuracy is: 0.8241106578987706
KNeighborsRegressor: n = 3, Accuracy is: 0.8159313128278626
KNeighborsRegressor: n = 4, Accuracy is: 0.8119873797309655
KNeighborsRegressor: n = 5, Accuracy is: 0.8090679203892419
KNeighborsRegressor: n = 6, Accuracy is: 0.8085563529726397
KNeighborsRegressor: n = 7, Accuracy is: 0.8096194946240436
KNeighborsRegressor: n = 8, Accuracy is: 0.8070245284166122
KNeighborsRegressor: n = 9, Accuracy is: 0.8029249466155639
```

```
KNeighborsRegressor
------------------------------
Accuracy : 0.8241106578987706
MAE : 0.32
MSE : 0.2
RMSE  : 0.444560
R2_SCORE  : 0.802925
```

Observations:

KNN is one of the most powerful algorithms, with the traffic dataset it has shown higher accuracy of 82% which is not very far from SVR algorithm. On increasing the neighborhood value beyond 2 the accuracy tends to fall.

E. Multi-Layer Perceptron Model

Artificial Neural Network algorithms are known to be highly efficient models with higher accuracy. These models learn from information mapping, and the learnt data is stored as weights, like the neurons in the human brain.

MLP the data flows forward from the input layer to the output layer through the hidden layer. These are trained using a backpropagation algorithm that minimises the loss function provided all the input and output variables are standardised before applying MLP. This algorithm is a supervised learning algorithm it can learn from the non-linear inputs for both regression and classification problems.[10]

This model had an accuracy of 93.4 with seed s=5 when filled the missing values using the ffill method, but the accuracy improved to 94% using interpolate method to fill the missing data.

```
Multi Layer Perceptron Model
------------------------------
Accuracy : 0.940257108012639
MAE : 0.17
MSE : 0.06
RMSE  : 0.244770
R2_SCORE  : 0.940257
```

Observations:

The model built with Multi-layer perceptron has the highest accuracy of 94% compared to all the other regression models. (Multiple Linear regression, Support Vector Regressor and K Neighbor Regressor).

8. Results

METHOD	*MAE*	*MSE*	*RMSE*	*R2_score*	*Accuracy*
Holt-winters (Unsup)	115.67	21468.11	146.52	-0.249	-
ARIMA (1, 1, 1) (1, 1, 1, 12)	88.27	11781.93	108.544611	0.314331	31.4%
Holt-winters (Sup)	114.08	21790.03	147.61	-0.268	-
ARIMA (2, 0, 2) (0, 0, 3, 12)	86.36	10107.47	100.535	0.6830	68%
Multiple Linear Regression	0.82	0.86	0.925102	0.146603	14%
Support Vector Regression	0.35	0.21	0.46	0.788	78.8%
K Nearest Regressor	0.32	0.2	0.444560	0.802925	82.4%
Multi-Layer Perceptron	0.17	0.06	0.244770	0.940257	94%

9. Discussion

In this current research, different algorithms have been applied and every algorithm has different accuracy, it is interesting to analyse why do different algorithms have different accuracies. Only few models are more efficient depending on the chosen dataset. For the current metro traffic dataset, accuracy of time series achieved is only 31% unsupervised and 68% with supervised, this is still low because, while building the model, selected features are only date and traffic volume, other dependent variables have been ignored, the dimensionality of real time data has been reduced to univariate. Also, the least accuracy was observed with MLR model, this is because the linear model assumes that there is linear relationship between input and output variables, which is not true in real-time, as seen from the correlation matrix the linear relation between variables is not satisfactory hence 14% accuracy is achieved. Similarly, the support vector regression and K nearest regressor identifies the appropriate line in the hyperplane and enables you to choose the minimum error while tuning the models, since the traffic volume dataset had no stronger linearity between variables the best possible accuracy achieved is 78.8% and 82.4% after tuning, which are nearly good models for prediction. On the other hand, the accuracy achieved with neural network (MLP) algorithm is the best 94%. This is because, MLP model has lot more coefficients to learn in the hidden layers, and uses activation functions. Feed forward networks, back propagation algorithms in neural network replicate the neural model of human brain and are often black box implementations with hidden layers. These algorithms have proven to be efficient with higher accuracies and always come with the risk of being black box implementations.

10. Conclusions

The dataset has no data recorded from mid-2014 till mid-2015, and the series has seasonality and trend pattern. Also, the series is not stationary. The data is processed to eliminate the missing data with interpolate and ffill methods, and the time-series curve is differentiated twice to make it stationary. Then ARIMA (p=2, d=0, q=2) model from sklearn is 68% accurate (30.5% with ffill). From this research, the highest accuracy achieved with time-series is 68% and with regression analysis is 94%. However, this cannot be compared because the time series model is built on the date and traffic volume (univariate) and regression is multivariate analysis.

11. Future Work

I intend to study sktime packages, compare the time-series model against the sklearn model, analyse how it affects accuracy and predictions and understand how the pitfalls of sklearn are handled in the sktime package.[11]

12. References

[1] "UCI Machine Learning Repository: Metro Interstate Traffic Volume Data Set." https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume (accessed Apr. 14, 2021).

[2] "An End-to-End Project on Time Series Analysis and Forecasting with Python | by Susan Li | Towards Data Science." https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b (accessed Apr. 14, 2021).

[3] "Regression Techniques in Machine Learning." https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/ (accessed Apr. 14, 2021).

[4] “Deep Learning for Time Series Forecasting.” https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/ (accessed Apr. 14, 2021).

[5] “Time Series in Python — Exponential Smoothing and ARIMA processes | by Benjamin Etienne | Towards Data Science.” https://towardsdatascience.com/time-series-in-python-exponential-smoothing-and-arima-processes-2c67f2a52788 (accessed Apr. 14, 2021).

[6] “Unsupervised Machine Learning Approaches for Outlier Detection in Time Series - Tech Rando.” https://techrando.com/2019/08/23/unsupervised-machine-learning-approaches-for-outlier-detection-in-time-series/ (accessed Apr. 14, 2021).

[7] “Introduction to Multivariate Regression Analysis.” https://www.mygreatlearning.com/blog/introduction-to-multivariate-regression/ (accessed Apr. 14, 2021).

[8] “Support Vector Regression In Machine Learning.” https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/ (accessed Apr. 14, 2021).

[9] “2 K-nearest Neighbours Regression | Machine Learning for Biostatistics.” https://bookdown.org/tpinto_home/Regression-and-Classification/k-nearest-neighbours-regression.html (accessed Apr. 14, 2021).

[10] “Multilayer Perceptron - an overview | ScienceDirect Topics.” https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron (accessed Apr. 14, 2021).

[11] “Univariate time series classification with sktime — sktime documentation.” https://www.sktime.org/en/latest/examples/02_classification_univariate.html (accessed Apr. 14, 2021).