Statistics for Data Science Assignment Report

Subhadeep Dash,

S20160010021

Abstract:

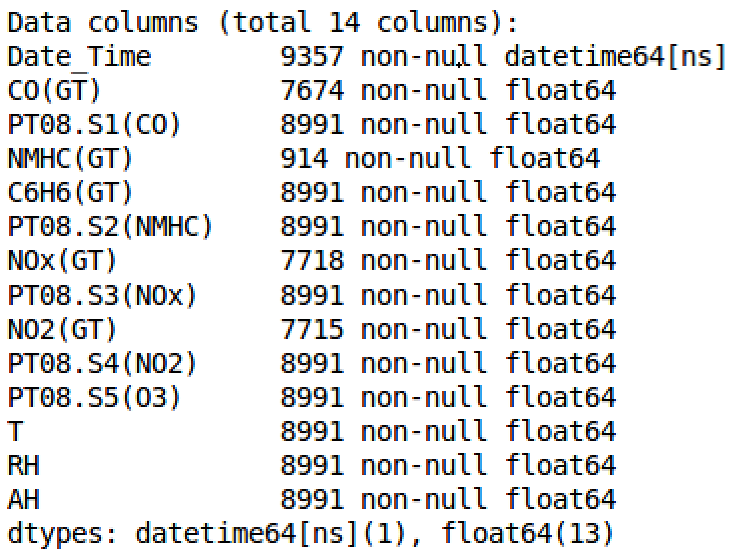
This paper is a report on the data analysis, regression performed and time series analysis performed on the Air Quality Dataset. In this paper, various ways of representing the statistical data and further processing on the dataset will be discussed.

Problem Statement:

Perform Regression, Data Analysis and Time Series Analysis on the Air Quality Dataset provided.

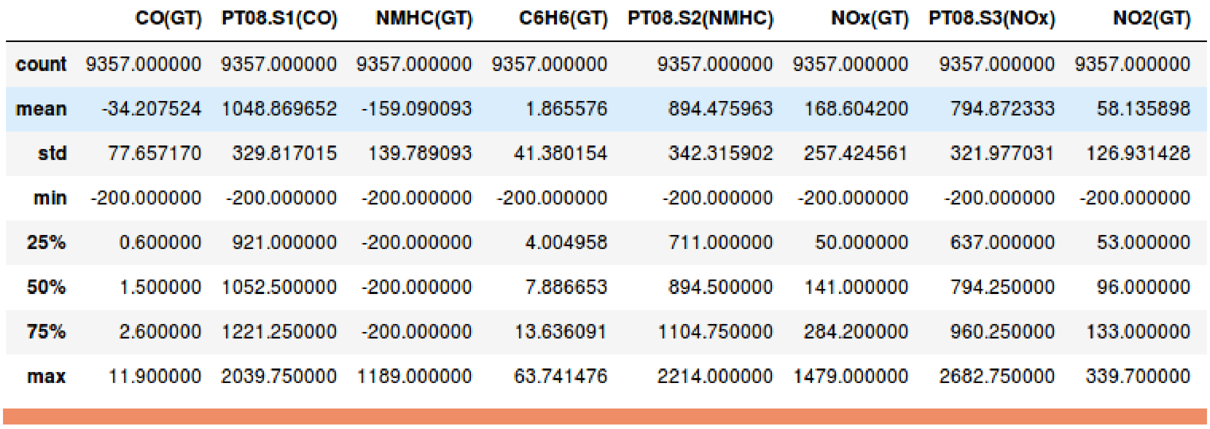
Initial Data Pre-processing:

The dataset was scanned and various details were checked upon. According to the dataset report, the null values in the data-set were replaced by -200 which could be seen in the overall summary of the data. The statistics after-200 was replaced with null from the dataset is as follows:

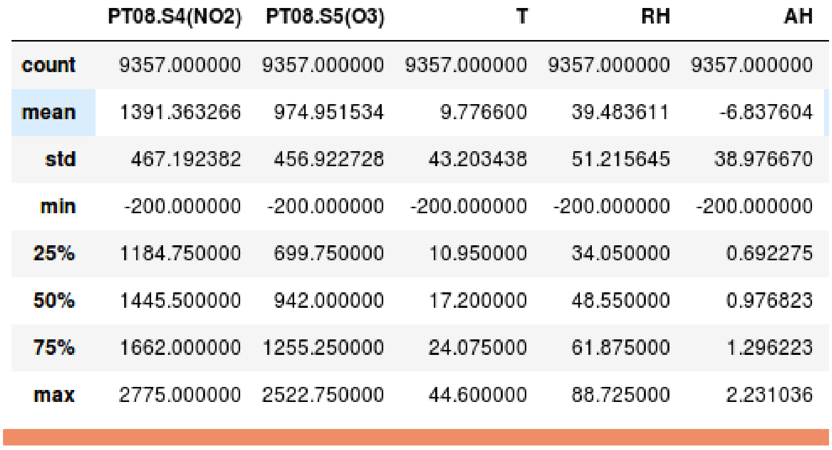


```
Data columns (total 14 columns):
Date_Time        9357 non-null datetime64[ns]
CO(GT)           7674 non-null float64
PT08.S1(CO)      8991 non-null float64
NMHC(GT)         914 non-null float64
C6H6(GT)         8991 non-null float64
PT08.S2(NMHC)    8991 non-null float64
NOx(GT)          7718 non-null float64
PT08.S3(NOx)     8991 non-null float64
NO2(GT)          7715 non-null float64
PT08.S4(NO2)     8991 non-null float64
PT08.S5(O3)      8991 non-null float64
T                8991 non-null float64
RH               8991 non-null float64
AH               8991 non-null float64
dtypes: datetime64[ns](1), float64(13)
```

We can observe that there are 9357 records are present out of which no field is completely filled for all records. The detailed description of the data is available below. The count includes the null attributes and the minimum is displayed as -200 for most of the columns because the -200 values have not been removed.



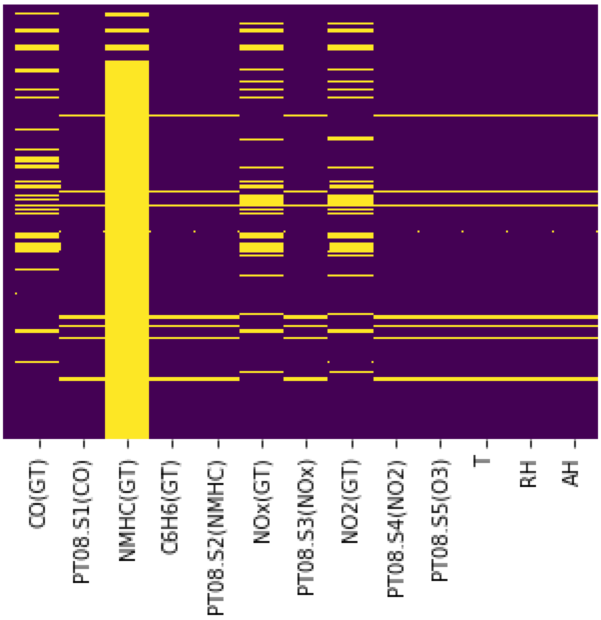
	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)
count	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000
mean	-34.207524	1048.869652	-159.090093	1.865576	894.475963	168.604200	794.872333	58.135898
std	77.657170	329.817015	139.789093	41.380154	342.315902	257.424561	321.977031	126.931428
min	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000
25%	0.600000	921.000000	-200.000000	4.004958	711.000000	50.000000	637.000000	53.000000
50%	1.500000	1052.500000	-200.000000	7.886653	894.500000	141.000000	794.250000	96.000000
75%	2.600000	1221.250000	-200.000000	13.636091	1104.750000	284.200000	960.250000	133.000000
max	11.900000	2039.750000	1189.000000	63.741476	2214.000000	1479.000000	2682.750000	339.700000



	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
count	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000
mean	1391.363266	974.951534	9.776600	39.483611	-6.837604
std	467.192382	456.922728	43.203438	51.215645	38.976670
min	-200.000000	-200.000000	-200.000000	-200.000000	-200.000000
25%	1184.750000	699.750000	10.950000	34.050000	0.692275
50%	1445.500000	942.000000	17.200000	48.550000	0.976823
75%	1662.000000	1255.250000	24.075000	61.875000	1.296223
max	2775.000000	2522.750000	44.600000	88.725000	2.231036

Tables displaying the data before missing values replacement.

The heat map for the missing values in each field for the data-set is as follows:

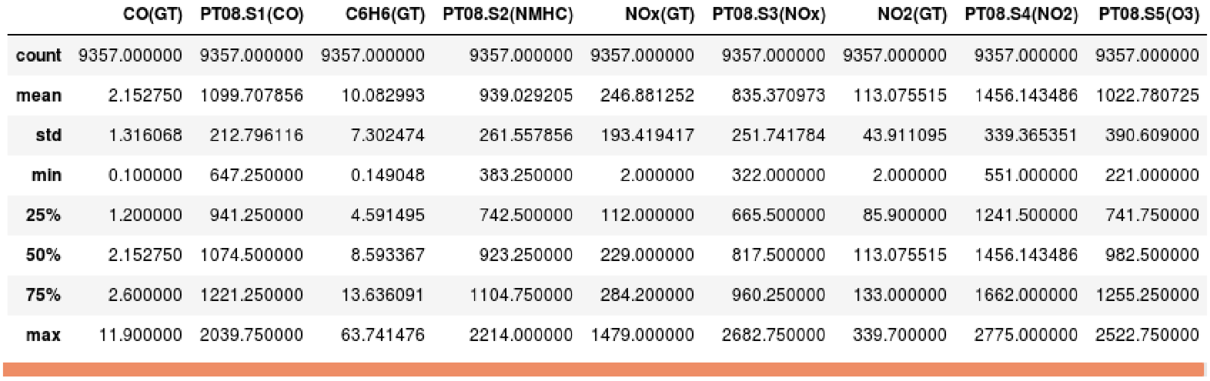


Heat Map Displaying Null values at various positions of the dataset

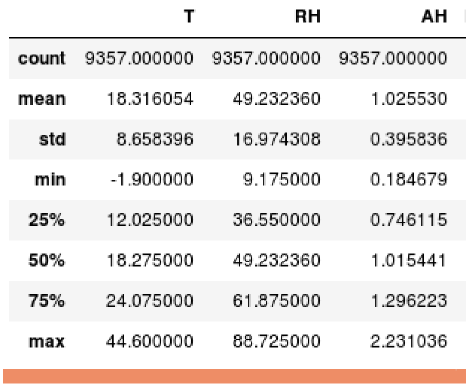
It can be seen from the head map that the NMHC(GT) column consists of a lot of null values, so it is better to discard it. For other columns, we replace the missing values in them with the means of their respective fields.

Modifying missing information:

After the removal of null values and discarding unnecessary fields, we obtain the following modified information from the data:



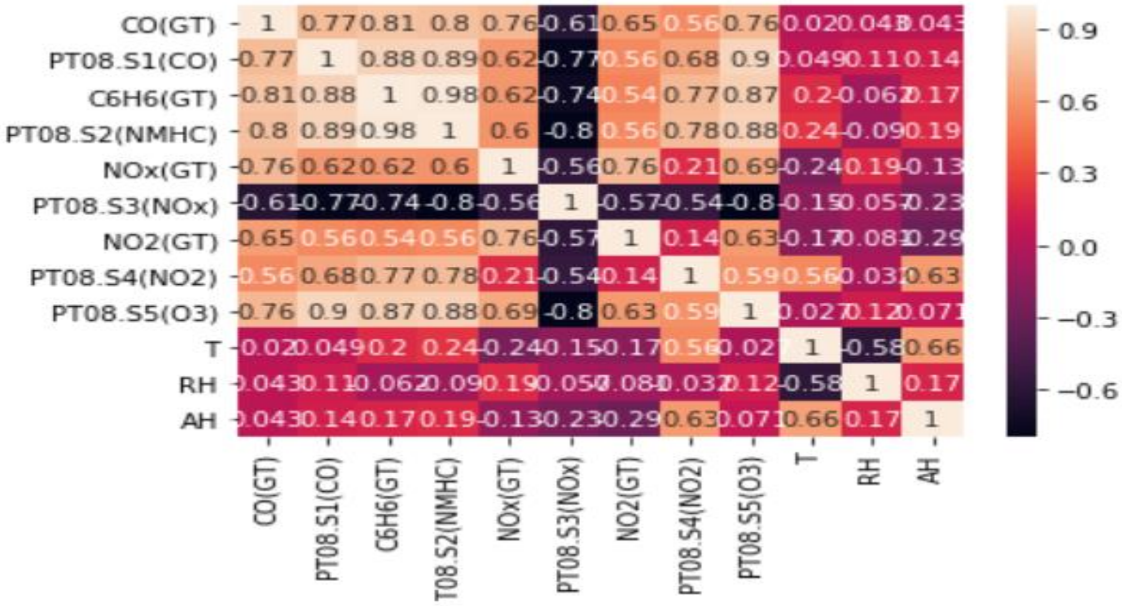
	CO(GT)	PT08.S1(CO)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)
count	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000	9357.000000
mean	2.152750	1099.707856	10.082993	939.029205	246.881252	835.370973	113.075515	1456.143486	1022.780725
std	1.316068	212.796116	7.302474	261.557856	193.419417	251.741784	43.911095	339.365351	390.609000
min	0.100000	647.250000	0.149048	383.250000	2.000000	322.000000	2.000000	551.000000	221.000000
25%	1.200000	941.250000	4.591495	742.500000	112.000000	665.500000	85.900000	1241.500000	741.750000
50%	2.152750	1074.500000	8.593367	923.250000	229.000000	817.500000	113.075515	1456.143486	982.500000
75%	2.600000	1221.250000	13.636091	1104.750000	284.200000	960.250000	133.000000	1662.000000	1255.250000
max	11.900000	2039.750000	63.741476	2214.000000	1479.000000	2682.750000	339.700000	2775.000000	2522.750000



	T	RH	AH
count	9357.000000	9357.000000	9357.000000
mean	18.316054	49.232360	1.025530
std	8.658396	16.974308	0.395836
min	-1.900000	9.175000	0.184679
25%	12.025000	36.550000	0.746115
50%	18.275000	49.232360	1.015441
75%	24.075000	61.875000	1.296223
max	44.600000	88.725000	2.231036

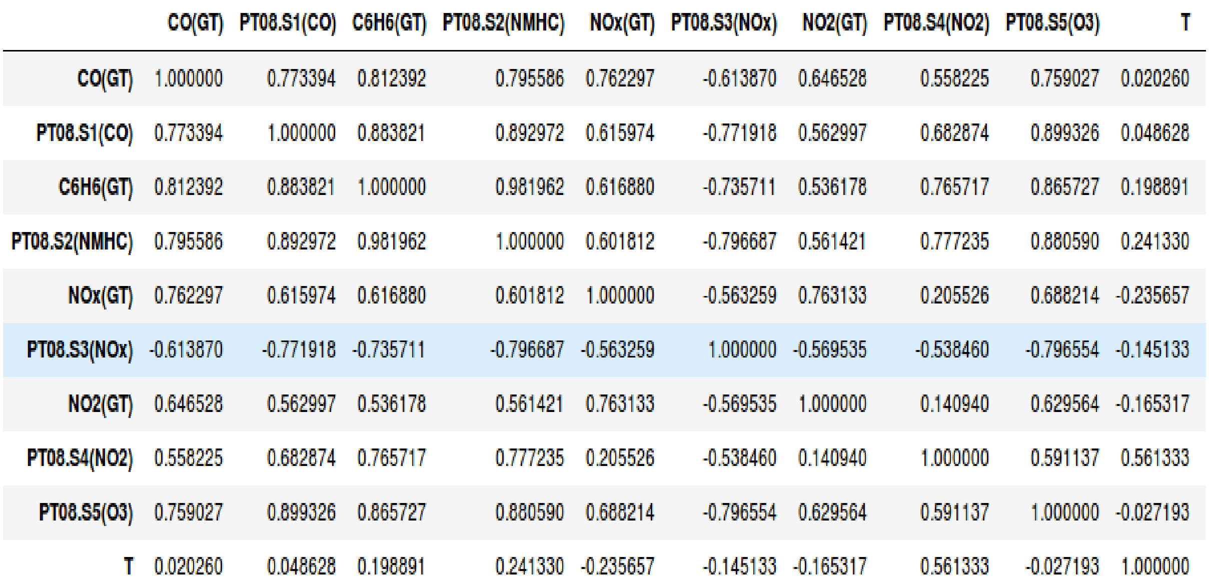
Tables representing modified values after removal of NULL from the dataset

Let's have a look on the correlation matrix obtained with the help of a heat map.



After removal of a field and filling missing values with mean, we obtain the above correlation matrix from which we can observe that it is symmetric and the diagonals represent the correlation with the element itself. Therefore, there are only ones across the diagonal which are represented in white colour. Some variables can be seen as negatively correlated which are towards the darker side of the heat map while others are positive correlated which have lighter colours in the heat map.

Our data attributes which we will be predicted through regression are AH and RH i.e. Absolute Humidity and Relative Humidity. Detailed and evident correlation with more precision between the features can be looked below:



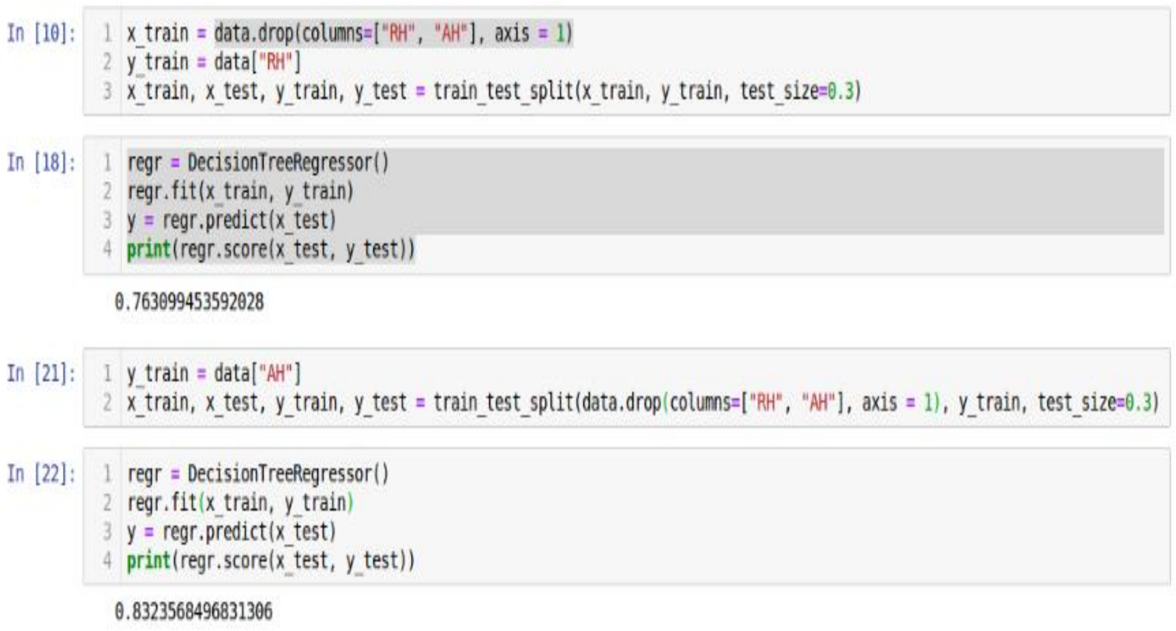
	CO(GT)	PT08.S1(CO)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T
CO(GT)	1.000000	0.773394	0.812392	0.795586	0.762297	-0.613870	0.646528	0.558225	0.759027	0.020260
PT08.S1(CO)	0.773394	1.000000	0.883821	0.892972	0.615974	-0.771918	0.562997	0.682874	0.899326	0.048628
C6H6(GT)	0.812392	0.883821	1.000000	0.981962	0.616880	-0.735711	0.536178	0.765717	0.865727	0.198891
PT08.S2(NMHC)	0.795586	0.892972	0.981962	1.000000	0.601812	-0.796687	0.561421	0.777235	0.880590	0.241330
NOx(GT)	0.762297	0.615974	0.616880	0.601812	1.000000	-0.563259	0.763133	0.205526	0.688214	-0.235657
PT08.S3(NOx)	-0.613870	-0.771918	-0.735711	-0.796687	-0.563259	1.000000	-0.569535	-0.538460	-0.796554	-0.145133
NO2(GT)	0.646528	0.562997	0.536178	0.561421	0.763133	-0.569535	1.000000	0.140940	0.629564	-0.165317
PT08.S4(NO2)	0.558225	0.682874	0.765717	0.777235	0.205526	-0.538460	0.140940	1.000000	0.591137	0.561333
PT08.S5(O3)	0.759027	0.899326	0.865727	0.880590	0.688214	-0.796554	0.629564	0.591137	1.000000	-0.027193
T	0.020260	0.048628	0.198891	0.241330	-0.235657	-0.145133	-0.165317	0.561333	-0.027193	1.000000

Correlation Matrix of training features

Regression:

Regression is a process of predicting a value of a specific attribute by implementing a well-trained model. Regression may seem similar to classification but classification involves assigning each data record ‘d’ to a well-defined class ‘c’ by predicting based on the values of the attributes whereas Regression predicts the value of a specific feature ‘f’ by working on the data of several attributes i.e. Regression is used for continuous labelling whereas Classification is used for discrete labelling.

In this assignment, the values of RH and AH were predicted by using a Decision Tree Regressor. The implementation was using Python and Sci-kit learn library was used for training the construction of the regressor. SVMs weren’t chosen, as they were taking a lot of time to train and Decision Trees were built quickly were compared to them. A Regression Score of 76.31 was obtained when the regressor was used to predict the Relative Humidity value. When the Absolute Humidity was predicted using the same regressor, a score of 83.24 was obtained. The code snippet can be seen in the figure below:



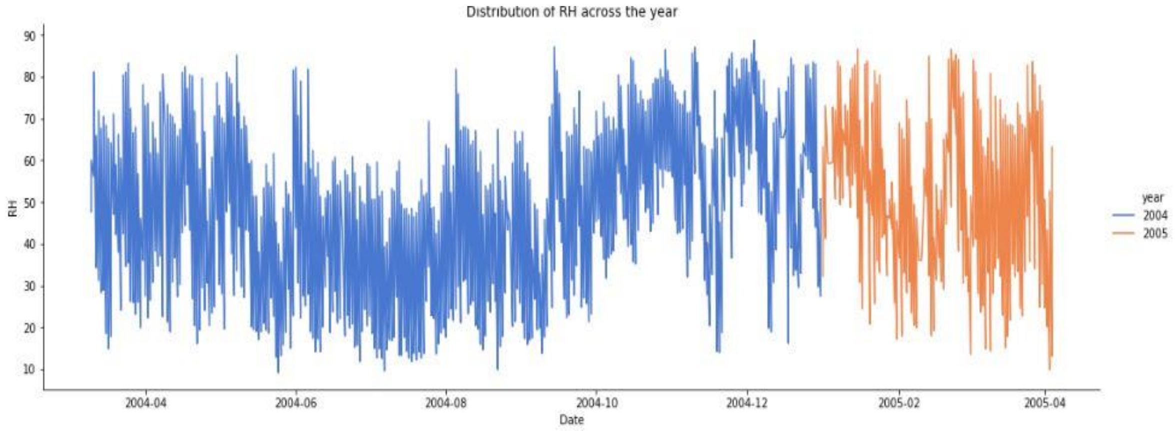
```
In [10]: x_train = data.drop(columns=["RH", "AH"], axis = 1)
         y_train = data["RH"]
         x_train, x_test, y_train, y_test = train_test_split(x_train, y_train, test_size=0.3)

In [18]: regr = DecisionTreeRegressor()
         regr.fit(x_train, y_train)
         y = regr.predict(x_test)
         print(regr.score(x_test, y_test))

         0.763099453592028

In [21]: y_train = data["AH"]
         x_train, x_test, y_train, y_test = train_test_split(data.drop(columns=["RH", "AH"], axis = 1), y_train, test_size=0.3)

In [22]: regr = DecisionTreeRegressor()
         regr.fit(x_train, y_train)
         y = regr.predict(x_test)
         print(regr.score(x_test, y_test))

         0.8323568496831306
```

Code Snippet of training and testing in Python

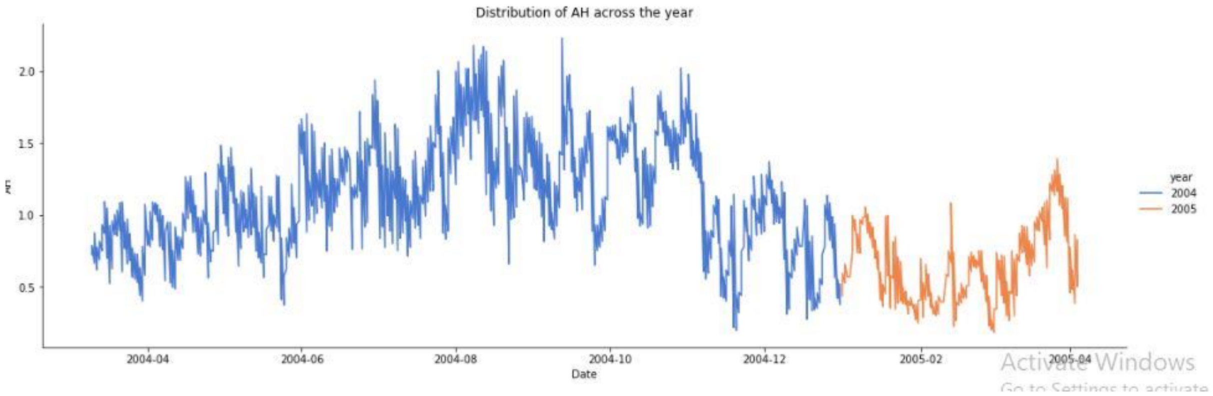
The same data had been divided into two parts out of which one was used for training and the other was used for testing. Decision Trees are considered one of the best structures involved in classification and regression. They are easy to interpret and just like the ones how human brain involves decision making(step-by-step).

Data Visualisation:

We visualize the temperature and humidity values in the years 2014 and 2015 by different colours in the following plot. The first plot is one of RH (Relative Humidity) and second one is of AH (Absolute Humidity).



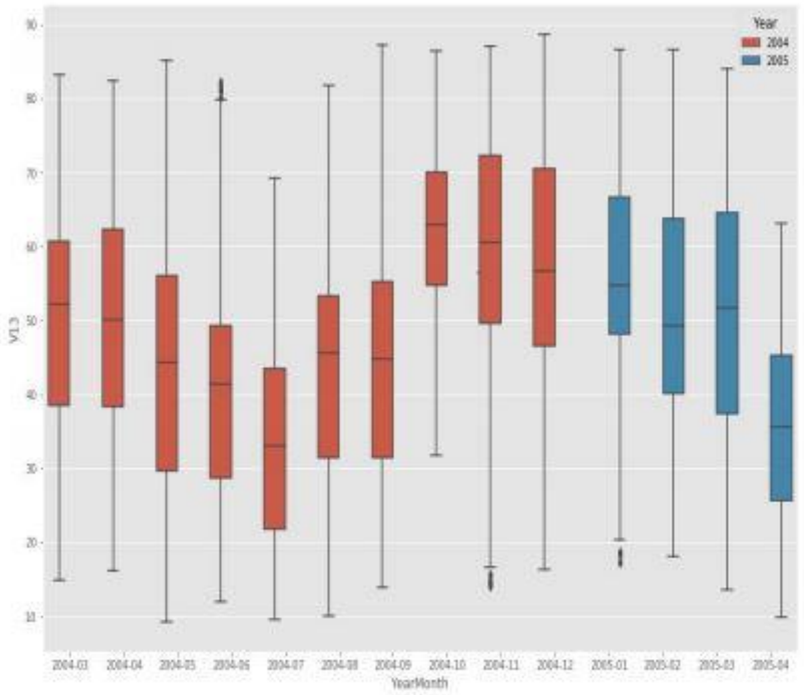
Time Series plot of Relative Humidity



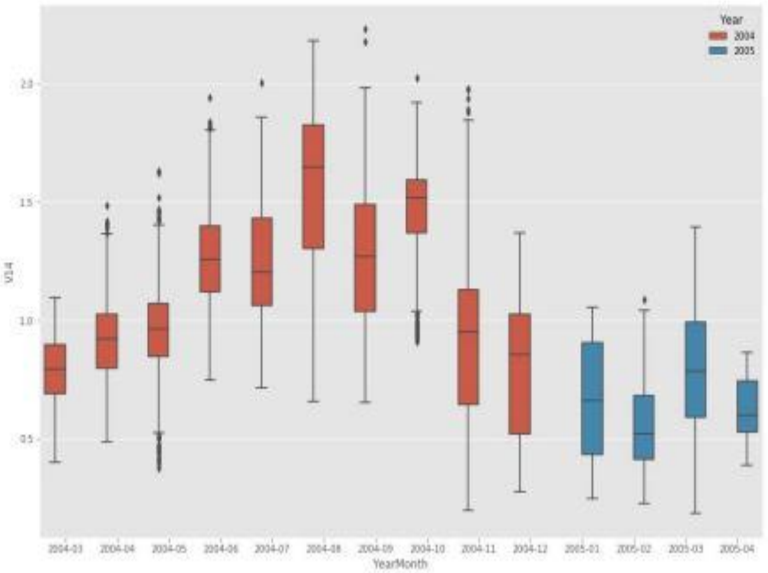
Time Series plot of Absolute Humidity

Boxplots:

The box plots can also be plotted in python using the data and it has been implemented to draw the box plots for the output parameters RH and AH to represent them month-wise. The plot can be seen in much detail below.



Box plot : Relative Humidity vs YearMonth



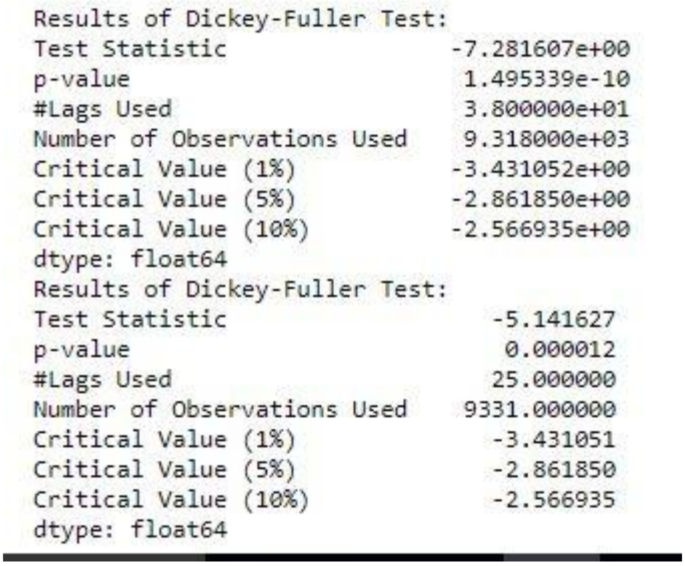
Box plot : Absolute Humidity vs YearMonth

Time series Analysis:

DQ test:

The time series is non stationary- This is the null hypothesis. If statistic is less than the critical value, reject the null hypothesis.

Results:



```
Results of Dickey-Fuller Test:
Test Statistic                -7.281607e+00
p-value                        1.495339e-10
#Lags Used                     3.800000e+01
Number of Observations Used    9.318000e+03
Critical Value (1%)           -3.431052e+00
Critical Value (5%)           -2.861850e+00
Critical Value (10%)          -2.566935e+00
dtype: float64
Results of Dickey-Fuller Test:
Test Statistic                   -5.141627
p-value                           0.000012
#Lags Used                       25.000000
Number of Observations Used    9331.000000
Critical Value (1%)              -3.431051
Critical Value (5%)              -2.861850
Critical Value (10%)             -2.566935
dtype: float64
```

The test statistic is less than the crtical value. So, RH and AH series are stationary.

ARIMA model:

This is used to forecast the data. ARIMA(p,d,q) p- number of AR terms

d- number of differences

q- number of moving average terms

Now, we plot ACF(Auto Correlation Function) and PACF(Partial Autocorrelation Function).

In the plot: p→ lag value where PACF plot crosses upper confidence interval for the first time

q→ lag value where ACF plot crosses upper confidence interval for the first time

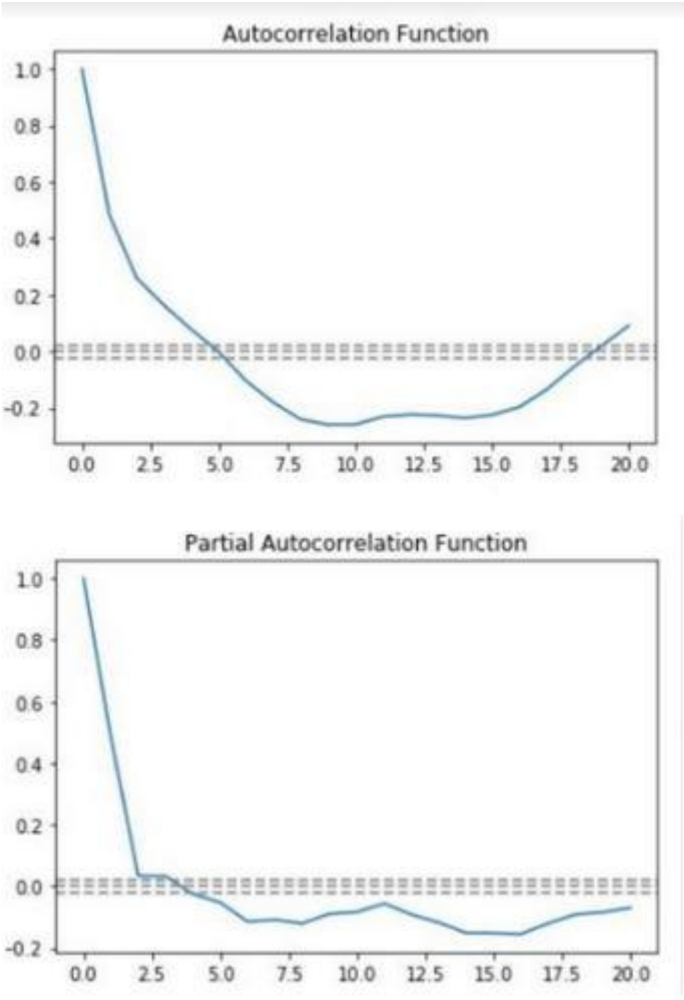


Figure: ACF and PACF for RH.

We can conclude that p=3 and q=4

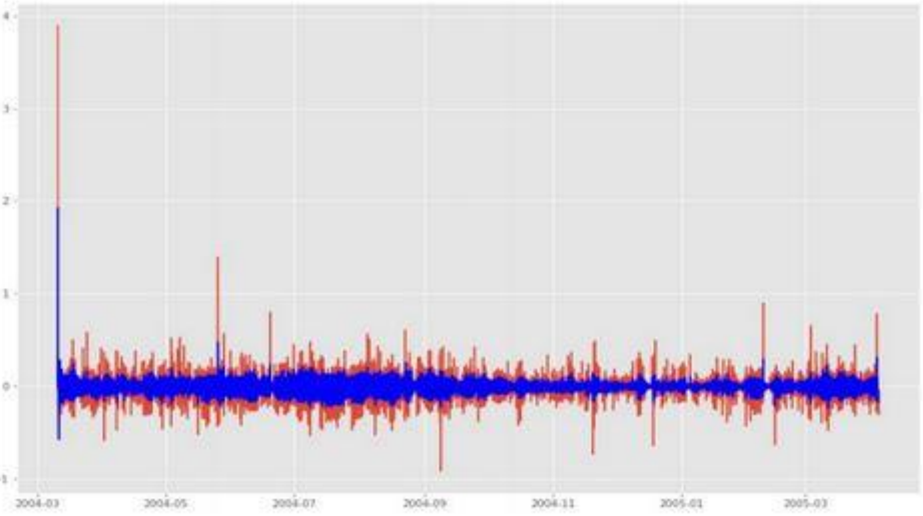


Figure: ARIMA model for RH

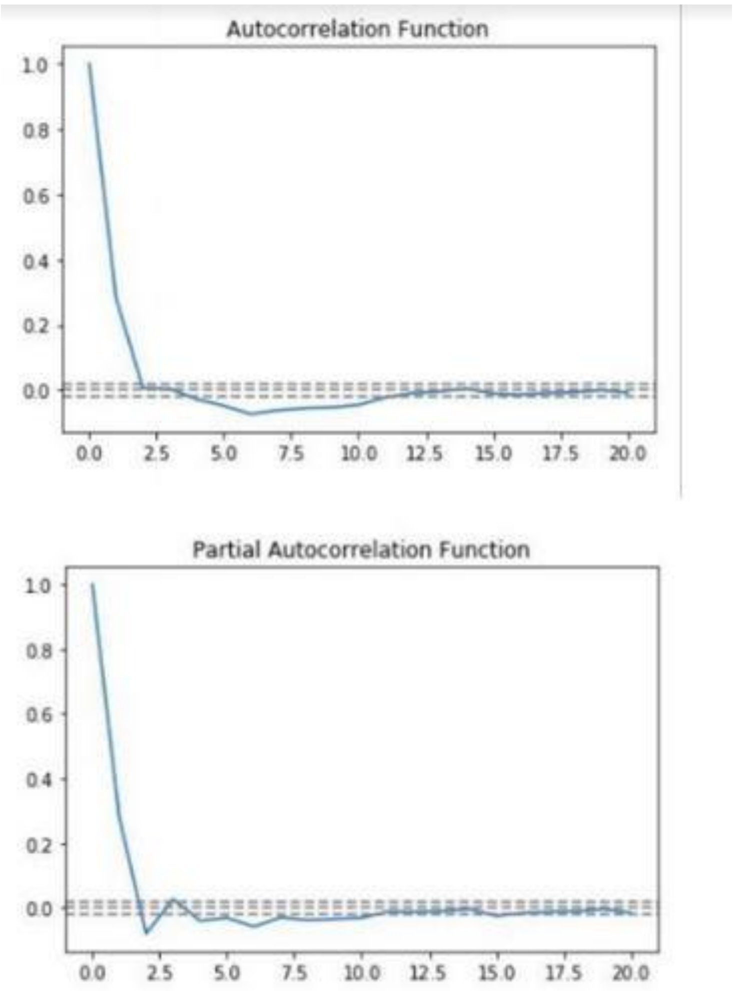


Figure: ACF and PACF for AH.

We can conclude that p=2 and q=2

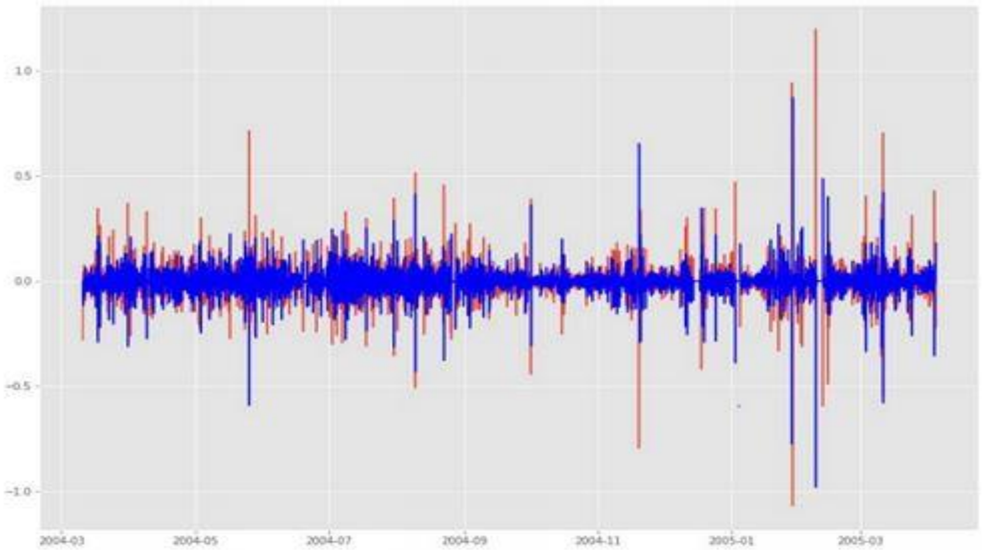


Figure: ARIMA model for AH

Conclusion:

Hence we have been able to perform data analysis, filling of missing values, regression and time series analysis for the given Air Quality Dataset using Python.