BRIEF DESCRIPTION

The gas sensor dataset contains datas for 4 different gases which are carbon(II) Oxide, methane, ethylene and ethylacetone. The dataset was collected over gas sensor arrays (containing 8 different sensors) with 5 replicate of the gas sensor array. The data collected spanned for 21 days considering 10 different concentration levels for each of the gases making a total of 40 concentration levels.

DATA STRUCTURE

In a day the data is recorded at a frequency of 0.1Hz and taken for 600seconds, for each individual board at a particular concentration level, considering a particular gas for 4 different replicate of a particular gas, on a particular board and a particular concentration level. The records are labelled in this format, B1_C0_F20_R1.txt implies the text file contains recorded values sampled for 0.1Hz at a total time of 600 seconds; considering board one (with 8 different sensors), gas CO was passed at a concentration of 20ppm repitition 1. This was repeated 4 times for the same concentration level, same gas, I.e 4 different text files containing B1_CO_F20_R1.txt, B1_CO_F20_R2.txt, B1_CO_F20_R3.txt, B1_CO_F20_R4.txt.

The experiment was conducted for 21 days, different time and different boards used. The values recorded are the corresponding value of each sensor on the board being considered, that is, output value from sensor one, sensor two, sensor three, sensor four, sensor five, sensor six, sensor seven and sensor eight. This files are raw files gotten from the sensor whose result from the individual sensor could be affected by temperature, humidity or any other climatic change.

The sensor operation are individual, that is, sensor A operation does not affect sensor B operation; they are all working individually.

The readings was taken for each gas considering 10 different concentration levels for 4 individual repitition which makes a total of 160 different text files collected for the four gases on a particular board. Since, we have 5 boards then we have a total of 480 text files collected.

The reason for the data gathered in this structure is for instrument calibration and correction; also to test for variation in values gotten from our sensor with respect to climatic condition so as to determine the accuracy of our sensor devices. Since, that's not our target then we will randomly pick 3 random concentration level for each gases restricting our task to board One; this makes us have 64 text files to work with, the text files used are alongside their repitition:

B1_FCO_70_(R1, R2,R3,R4).txt - merged into merged_FCO_70.txt
B1_FCO_80_(R1,R2,R3,R4).txt - merged into merged_FCO_80.txt
B1_FCO_90_(R1,R2,R3,R4).txt – merged into merged_FCO_90.txt

B1_FEA_20_(R1,R2,R3,R4).txt – merged into merged_FEA_20.txt
B1_FEA_30_(R1,R2,R3,R4).txt – merged into merged_FEA_30.txt
B1_FEA_40_(R1,R2,R3,R4).txt – merged into merged_FEA_40.txt

B1_FEY_40_(R1,R2,R3,R4).txt – merged into merged_FEY_40.txt
B1_FEY_50_(R1,R2,R3,R4).txt – merged into merged_FEY_50.txt
B1_FEY_60_(R1,R2,R3,R4).txt – merged into merged_FEY_60.txt

B1_FME_60_(R1,R2,R3,R4).txt – merged into merged_FME_60.txt
B1_FME_70_(R1,R2,R3,R4).txt – merged into merged_FME_70.txt

B1_FME_80_(R1,R2,R3,R4).txt – merged into merged_FME_80.txt

for example considering gas CO at concentration 20ppm, preprocessing repitition1, repitition2, repitition3, repitition4 into a single file. The screenshot below shows that we loaded in text file B1_GCO_F040_R1.txt, B1_GCO_F040_R2.txt, B1_GCO_F040_R3.txt, B1_GCO_F040_R4.txt which will be merged into a single processed file.

```
In [21]: import os
         import sys
         os.listdir()

Out[21]: ['.ipynb_checkpoints',
          'B1_GCO_F040_R1.txt',
          'B1_GCO_F040_R2.txt',
          'B1_GCO_F040_R3.txt',
          'B1_GCO_F040_R4.txt',
          'Box_plot_of_sense_eightboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_eight_no_outliers_board_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_fiveboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_five_no_outliers_board_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_fourboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_four_no_outliers_board_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_oneboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_one_no_outliers_board_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_sevenboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_seven_no_outliers_board_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_sixboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_six_no_outliers_board_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_threeboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_three_no_outliers_board_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_twoboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_sense_two_no_outliers_board_one_CO_at_40ppm.jpeg',
          'Box_plot_of_timeboard_one_CO_at_40ppm.jpeg',
          'Box_plot_of_time_no_outliers_board_one_CO_at_40ppm.jpeg',
          'merge_co_40.ipynb']
```

The screenshot below shows a visual representation of the data gotten from the 4 text files R1, R2, R3, R4 in a pandas dataframe.

In [22]:
```
import pandas as pd
repitition_one = pd.read_csv(r'B1_GCO_F040_R1.txt',sep = '\t',header = None)
repitition_one.columns = ['time','sense_one', 'sense_two', 'sense_three','sense_four','sense
_five',
                          'sense_six','sense_seven','sense_eight']
repitition_one.head()
```

Out[22]:

	time	sense_one	sense_two	sense_three	sense_four	sense_five	sense_six	sense_seven	sense_eight
0	0.00	39.22	18.64	21.73	5.58	73.08	45.87	55.81	7.01
1	0.01	39.22	18.61	21.73	5.58	73.30	45.87	55.81	7.01
2	0.02	39.22	18.61	21.69	5.58	73.08	45.67	55.81	7.01
3	0.03	39.22	18.61	21.73	5.58	73.08	45.67	55.81	7.01
4	0.04	39.22	18.61	21.69	5.58	73.30	45.77	55.81	7.01

In [23]:
```
repitition_two = pd.read_csv(r'B1_GCO_F040_R2.txt',sep = '\t',header = None)
repitition_two.columns = ['time','sense_one', 'sense_two', 'sense_three','sense_four','sense
_five',
                          'sense_six','sense_seven','sense_eight']
repitition_two.head()
```

Out[23]:

	time	sense_one	sense_two	sense_three	sense_four	sense_five	sense_six	sense_seven	sense_eight
0	0.00	36.43	17.09	19.84	5.81	68.66	41.78	52.64	7.03
1	0.01	36.43	17.09	19.84	5.81	68.66	41.87	52.51	7.03
2	0.02	36.50	17.09	19.84	5.81	68.86	41.87	52.76	7.03
3	0.03	36.50	17.14	19.86	5.81	68.66	41.78	52.51	7.03
4	0.04	36.43	17.11	19.84	5.81	68.86	41.78	52.64	7.03

In [24]:
```
repitition_three = pd.read_csv(r'B1_GCO_F040_R3.txt',sep = '\t',header = None)
repitition_three.columns = ['time','sense_one', 'sense_two', 'sense_three','sense_four','sen
se_five',
                          'sense_six','sense_seven','sense_eight']
repitition_three.head()
```

Out[24]:

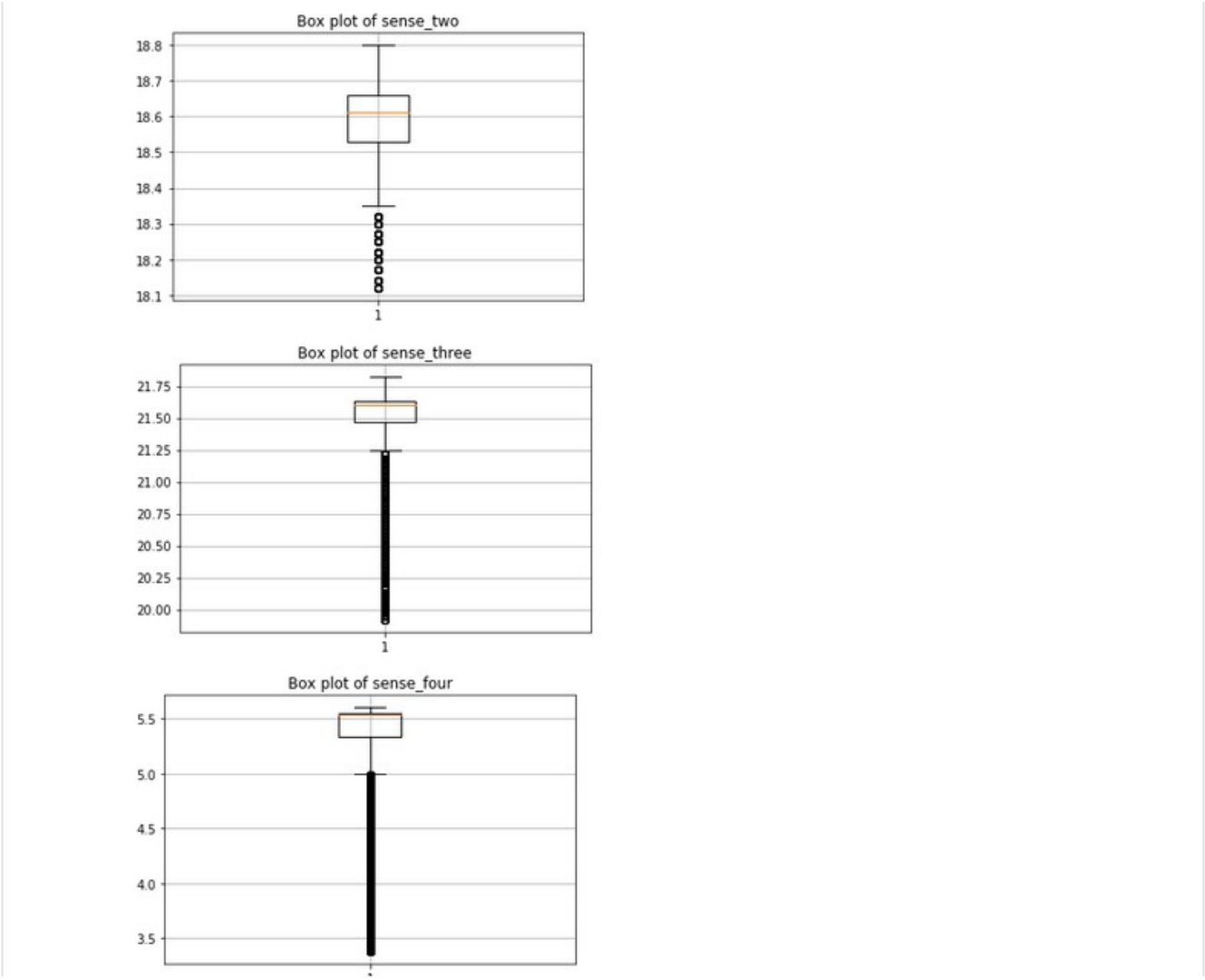
	time	sense_one	sense_two	sense_three	sense_four	sense_five	sense_six	sense_seven	sense_eight
0	0.00	44.11	21.83	24.66	8.04	80.85	49.28	65.38	8.53
1	0.01	44.11	21.83	24.66	8.04	80.58	49.28	65.38	8.53
2	0.02	44.11	21.80	24.66	8.04	80.58	49.16	65.19	8.53
3	0.03	44.11	21.80	24.66	8.04	80.58	49.16	65.38	8.54
4	0.04	44.11	21.83	24.66	8.04	80.58	49.16	65.19	8.53

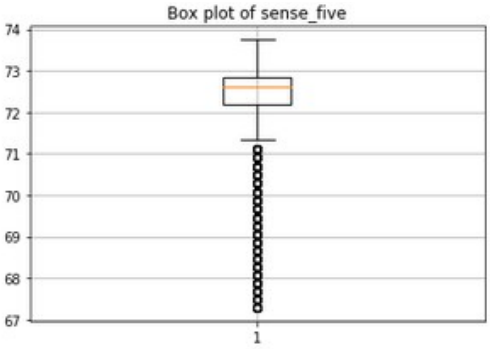
In [25]:
```
repitition_four = pd.read_csv(r'B1_GCO_F040_R4.txt',sep = '\t',header = None)
repitition_four.columns = ['time','sense_one', 'sense_two', 'sense_three','sense_four','sens
e_five',
                          'sense_six','sense_seven','sense_eight']
repitition_four.head()
```

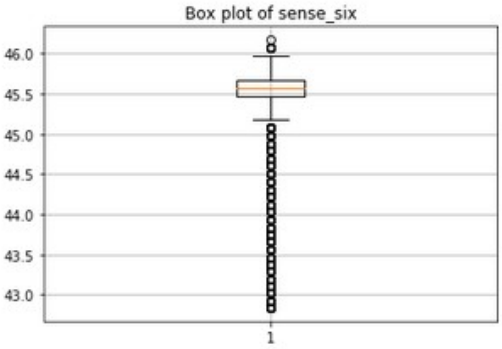
Out[25]:

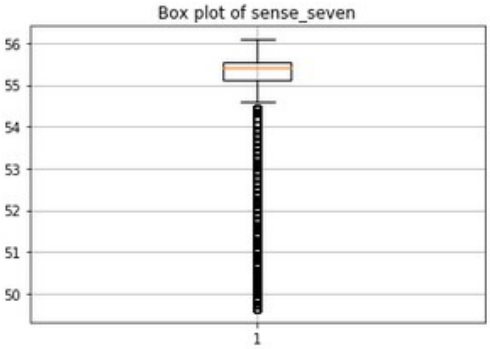
	time	sense_one	sense_two	sense_three	sense_four	sense_five	sense_six	sense_seven	sense_eight
0	0.00	46.47	23.05	26.07	7.98	85.89	50.90	69.01	8.14
1	0.01	46.58	23.05	26.16	7.98	85.89	50.90	69.21	8.13
2	0.02	46.47	23.05	26.16	7.98	85.89	50.90	69.21	8.13
3	0.03	46.47	23.09	26.16	7.98	85.89	50.78	69.01	8.13
4	0.04	46.47	23.05	26.16	7.98	85.89	50.90	69.01	8.13

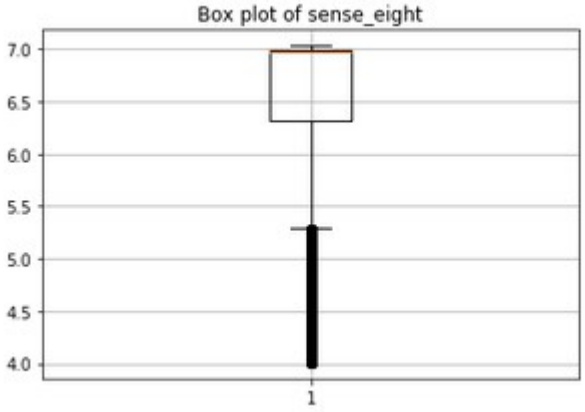
Inorder to have a perfect visual of or data considering each column tagged sense_one(sensor one), sense_two(sensor two), sense_three(sensor three), sense_four(sensor four), sense_five(sensor five), sense_six(sensor six), sense_seven(sensor seven), sense_eight(sensor eight); we went ahead to use a boxplot representation of the data using the matplotlib library in python. The screenshots below indicates that:











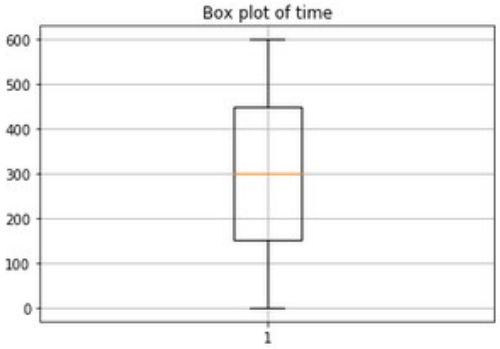
from the graphical visuals show above using the matplotlib library we could notice dark and thick lines on each graph representing each sensor data (spanning between their respective ranges); those dark and thick lines represents outliers present in our data which could be due to environmental influence while gathering the data and other possible factors that are unknown. Therefore a little transformation was made using the source code below to filter out the outliers present in each sensor data gathered:

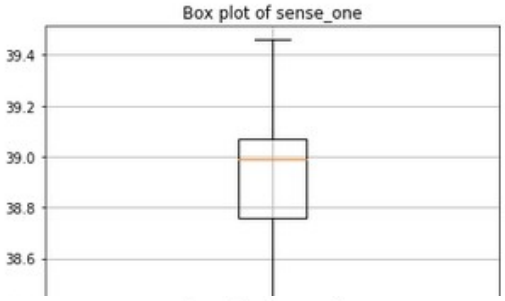
```
In [27]: repitition_one['sense_one'][repitition_one['sense_one'] < 38.4] = 38.4

In [28]: repitition_one['sense_two'][repitition_one['sense_two'] <= 18.36] = 18.36

In [29]: repitition_one['sense_three'][repitition_one['sense_three'] < 21.25] = 21.25

In [30]: repitition_one['sense_four'][repitition_one['sense_four'] <= 5.0 ]= 5.0

In [31]: repitition_one['sense_five'][repitition_one['sense_five'] <= 71.5] = 71.5

In [40]: repitition_one['sense_six'][repitition_one['sense_six'] < 45.20] = 45.20
         repitition_one['sense_six'][repitition_one['sense_six'] > 45.96] = 45.96

In [33]: repitition_one['sense_seven'][repitition_one['sense_seven'] < 54.6] = 54.6

In [34]: repitition_one['sense_eight'][repitition_one['sense_eight'] <5.3] = 5.3
```

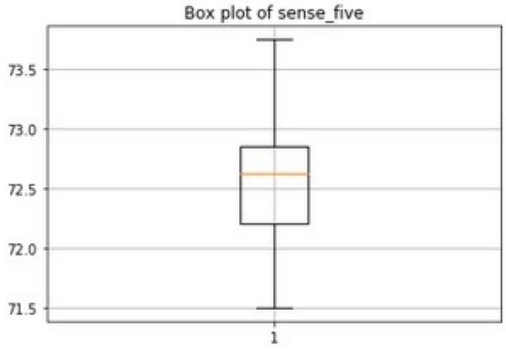
The source code above in jupyter notebook works this way:
In line 27 filter out all data collected by sensor one below 38.4 and then set them to 38.4
In line 28 filter out all data collected by sensor two below 18.36 and then set them to 18.36
In line 29 filter out all data collected by sensor three below 21.25 and then set them to 21.25
In line 30 filter our all data collected by sensor four below 5.0 and then set them to 5.0
In line 31 filter out all data collected by sensor five below 71.5 and then set them to 71.5
In line 40 filter out all data collected by sensor six outside the range of 45.20 and 45.96
In line 33 filter out all data collected by sensor seven below 54.6 and then set them to 54.6
In line 34 filter out all data collected by sensor eight below 5.3 and then set them to 5.3

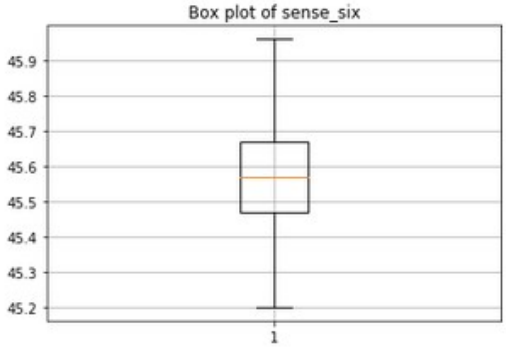
The essence of doing this, is to preprocess our data so that we can feed in a clean data to our algorithm to be able to perform well and not do worse; on doing this, checking out for outliers again which could be a threat to our learning algorithm we got a clean data as confirmed below by visualization:

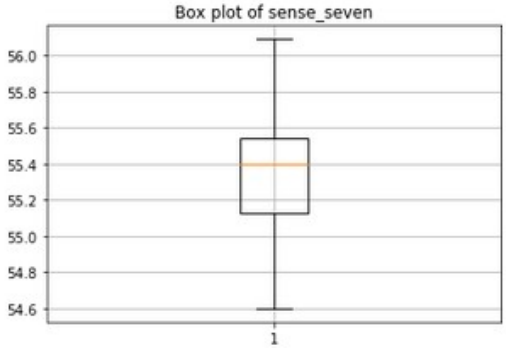
In [41]:
```
import matplotlib.pyplot as plt

for key in repitition_one.keys():
    plt.boxplot(repitition_one[key])
    plt.grid('on')
    plt.title('Box plot of '+str(key))
    plt.show()
    plt.savefig('Box_plot_of_'+ str(key)+ '_no_outliers_board_one_CO_at_40ppm.jpeg')
```

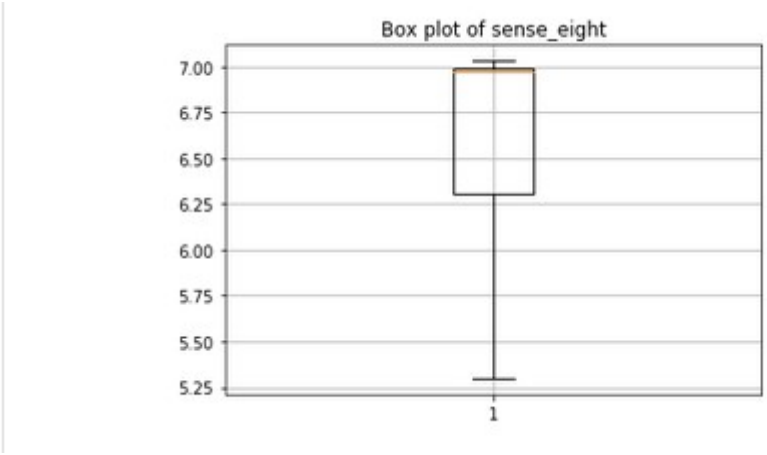












Now we have a perfect and clean data in repitition one, this same analysis was conducted likewise for repitition2, repitition3, repitition4 so that on merging the repititions into a single file we will have all to be a clean data, we got a clean result.