Air Quality report

Alicja Szymikowska,
Julia Wenta
Faculty of Applied Physics and Mathematics
Gdansk University of Technology, 2020

Table of contents

1. Introduction, motivation and goals

Nowadays, we live in an increasingly polluted environment, it has a huge impact on our health and well-being. Breathing is the basis for the functioning of all living organisms, so it is an issue that concerns us all.
However, the motivation of our work is, above all, acquiring the skills to process, analyze and model data. We also aimed to better understand Python libraries, including Numpy, Pandas, and Matplotlib. The goal was also to learn more about the topic of air quality.

2. Description of data - structure of sets, description of variables

The dataset contains the answers obtained from a multi-sensor device located on a field in an Italian city. Average hourly responses are recorded along with gas concentration references from a certified analyzer. [1]

Data set Characteristics:	*Multivariate, Time-Series*	**Number of Instances:**	*9358*	**Area:**	*Computer*
Attribute Characteristics:	*Real*	**Number of Attributes:**	*15*	**Date**	*2016-03-23*
Associated Tasks:	*Regression*	**Missing Values?**	*Yes (signed with "-200")*		

Attribute Information:

- *Date* (DD/MM/YYYY)
- *Time* (HH.MM.SS)
- *CO(GT)* - True hourly averaged concentration CO in mg / m^3 (reference analyzer)
- *PT08.S1* (tin oxide) - Hourly averaged sensor response (nominally CO targeted)
- *NMHC(GT)* - True hourly averaged overall Non Metanic HydroCarbons concentration in microg / m^3 (reference analyzer)

- *C6H6(GT)* -True hourly averaged Benzene concentration in microg / m^3 (reference analyzer)
- *PT08.S2* - (titania) hourly averaged sensor response (nominally NMHC targeted)
- *NOx(GT)* - True hourly averaged NOx concentration in ppb (reference analyzer)
- *PT08.S3* - (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)
- *NO2(GT)* - True hourly averaged NO2 concentration in microg/ m^3 (reference analyzer)
- *PT08.S4* - (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
- PT08.S5 - (indium oxide) hourly averaged sensor response (nominally O3 targeted)
- *T* - Temperature in °C
- *RH* - Relative Humidity (%)
- *AH* - Absolute Humidity

3. Description of the process of preparing data for analysis

The data we used for the analysis come from a page with data sets [1]. We imported downloaded data to our application written in Python. The Pandas library was used to load the data.

After loading the data, we checked the number of rows in which there were missing data, their number was 8530. This very large number caused that it was necessary to prepare the data for analysis in such a way that they could be subjected to subsequent processing.
At the beginning, with the help of the *dtypes* [2] function, we checked the data types - some of the data had the object type, and the other the float64 type. The data type has been changed to float64 for all features, except Date and Time.
Then, thanks to the *isnull ()* [3] *.sum ()* [4] functions, we received information about empty rows and columns in our data set. The set had two empty columns called "Unnamed" that were removed. The remaining columns contained 114 empty rows, these rows were also removed using the command: *air_data = air_data.dropna (how = "all")* [5], where *air_data* is our data set.
Empty data series probably resulted from a power cut from the sensor device.
After removing the empty cells, we went to the cells that contained the missing values, marked by "-200". To begin with, we changed all of these values into empty cells to supplement them with new values after correlation.

We checked subsequent correlations between data using the *corr ()* [6] method.
The correlation results were placed as a comment in the clearData.py file.
For example, the largest CO (GT) correlation occurs with C6H6 (GT), the remaining correlations are shown in Figure 3.1.

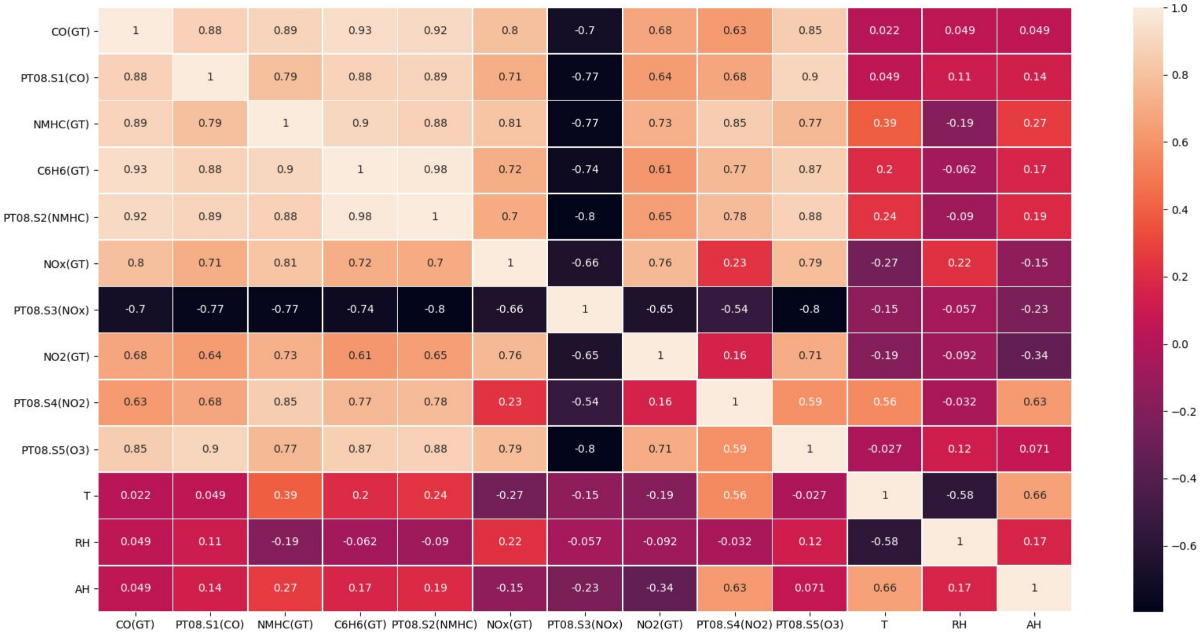

Figure 3.1. Results of correlation between data

The appropriate values for each of the cells were obtained and updated using the *groupby* [7], *apply* [8], *ffill* [9] and *bfill* [10] methods.
After completing the steps described above, the data did not contain empty records. The updated data was saved to a csv file for further use during analysis and modeling.

4. Data analysis - assumptions, a brief description of the methods and chosen methodology of analysis

Model 1: After preparing the data in the clearData.py file, we were able to proceed to the data analysis. Again, we can take a closer look at the data contained in the AirQuality_Cleared.csv file. After loading this file, we have used the describe() [11] method. As a result, we received the necessary statistical information on the data contained in the file, e.g: average value, standard deviation, minimum value, maximum value and 25%, 50% and 75% quartiles. Then, to make reading the data more user-friendly, we visualized them using charts. The following charts show the approximate trend of values by date, the dates on graph below (Figure 4.1) can be replaced with other dates to find other features.

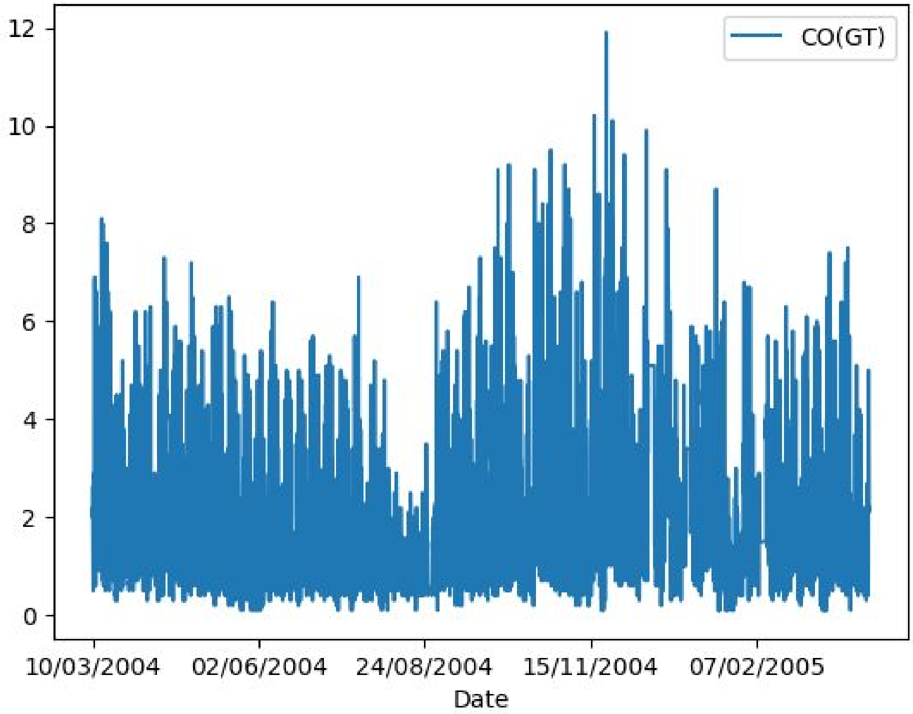

Figure 4.1 Graph of carbon dioxide dependence on date

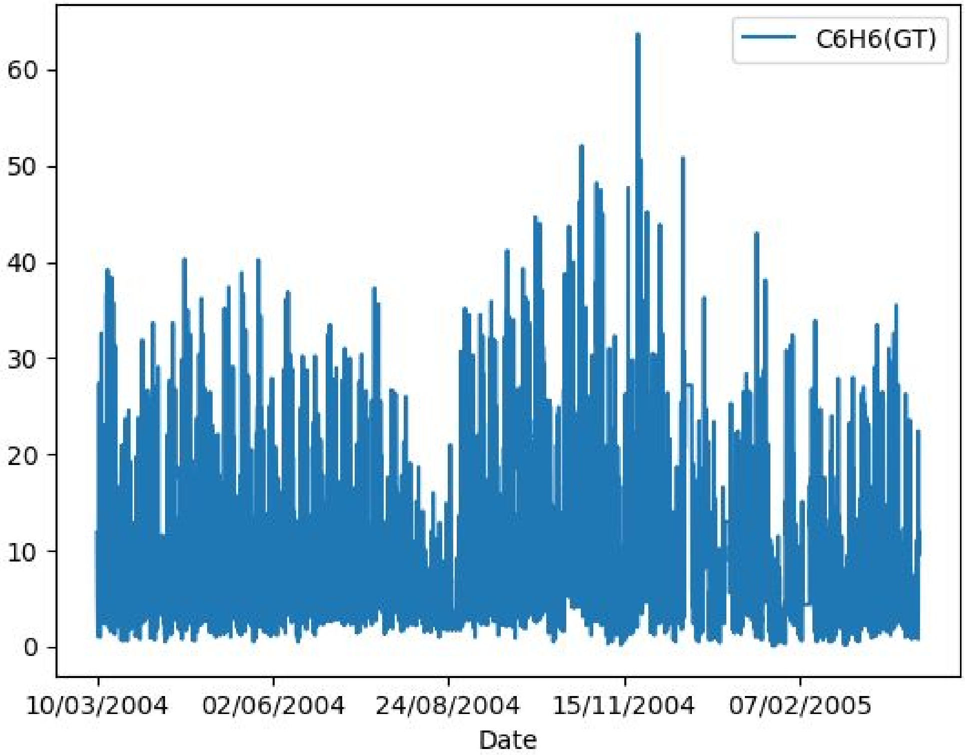

Figure 4.2 Graph of benzene concentration dependence date

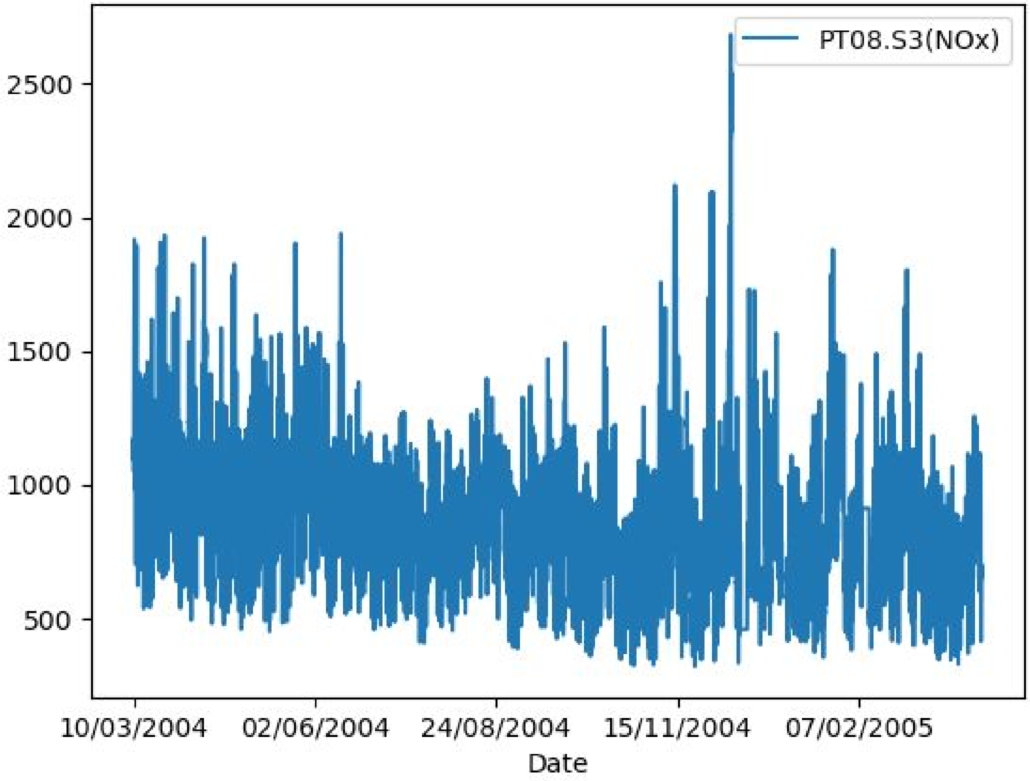

Figure 4.3 Graph of tungsten oxide dependence on date

All charts were included in the repository [13]. The next step was to add a new column *'dateAndTime'.* It contains data from the columns *'date'* and *'time'*. Then the new column was formatted. Based on this step, subsequent ones were made, where the dates were divided into months, days of the week and also hours. This is due to the fact that one of the most important variables describing in linear regression is time. Most artificial and natural phenomena operate in hourly, daily and monthly cycles.

In the next part, we correlated variables with the variable 'CO'. This is visualized in the chart below (Figure 4.4).

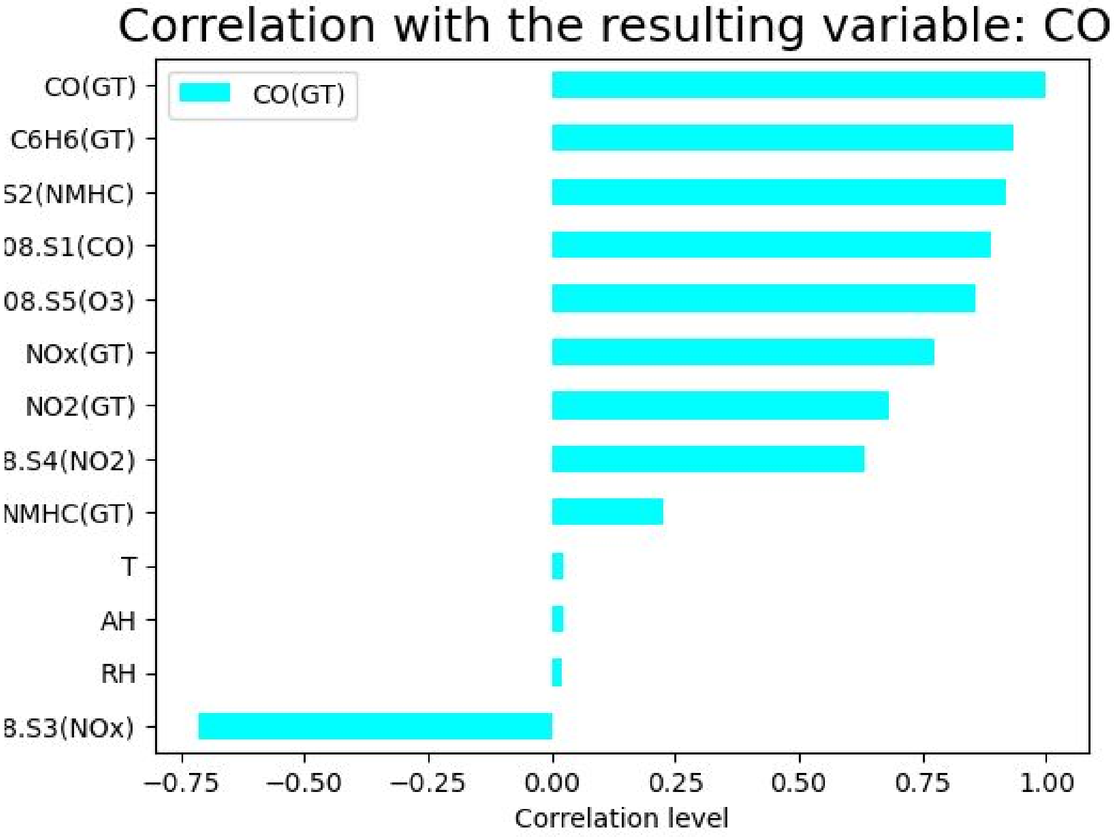

Figure 4.4 Correlation graph for the CO variable

We want to check how weather and time affect the level of air pollution. Therefore, we will be interested in the three least correlated variables 'T', 'AH', 'RH'.
Each of these variables was checked using the *correlationFunction* and *checkShift* functions. Using these functions, we obtained:

Optimal shift for RH: 12
0.3920431367189807

Optimal shift for AH: 12
0.04375636410267759

Optimal shift for T: 12
-0.22446569561762525

The above graph (Figure 4.4) shows that the variables ‘T’ and ‘RH’ correlate with the CO (GT) variable twelve hours after the pollution, while the ‘AH’ variable does not correlate with the CO (GT) variable. We used the *shiftDataFrame12* method to create a new *DataFrame*, thanks to which we obtained the above results.

Model 2: We also attempted to analyze the data taking into account the variable C6H6 (Actual hourly average benzene concentration). It should be noted that benzene is toxic and harmful to humans.
First, we made charts using the pairpolt *[14]* function.

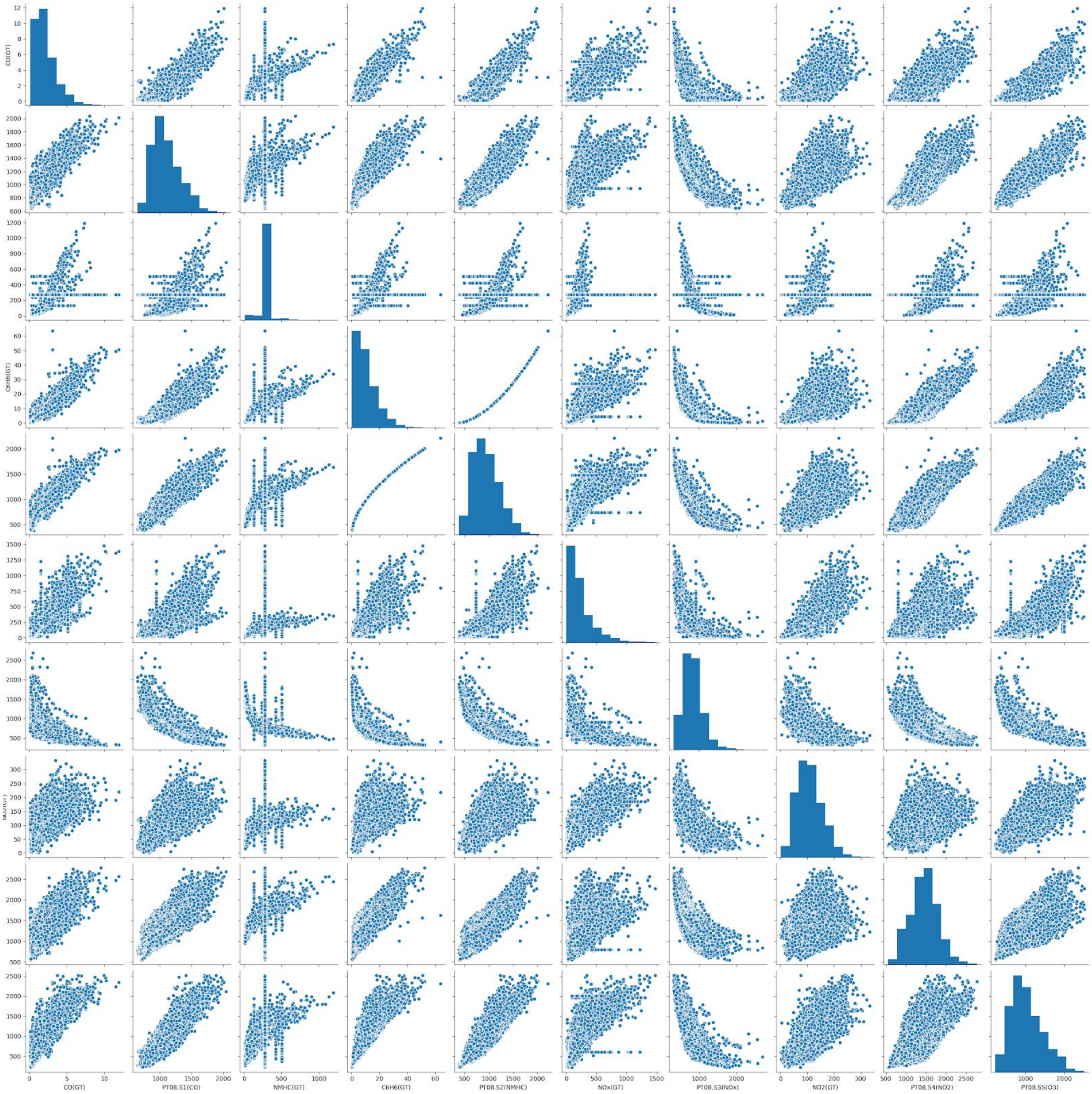

Figure 4.5 Chart showing that data is linear

Thanks to this procedure, you can easily tell if the data is linear. In the next step, we used the *'dateAndTime'* parameters created earlier. We have obtained graphs of concentration levels depending on the month:

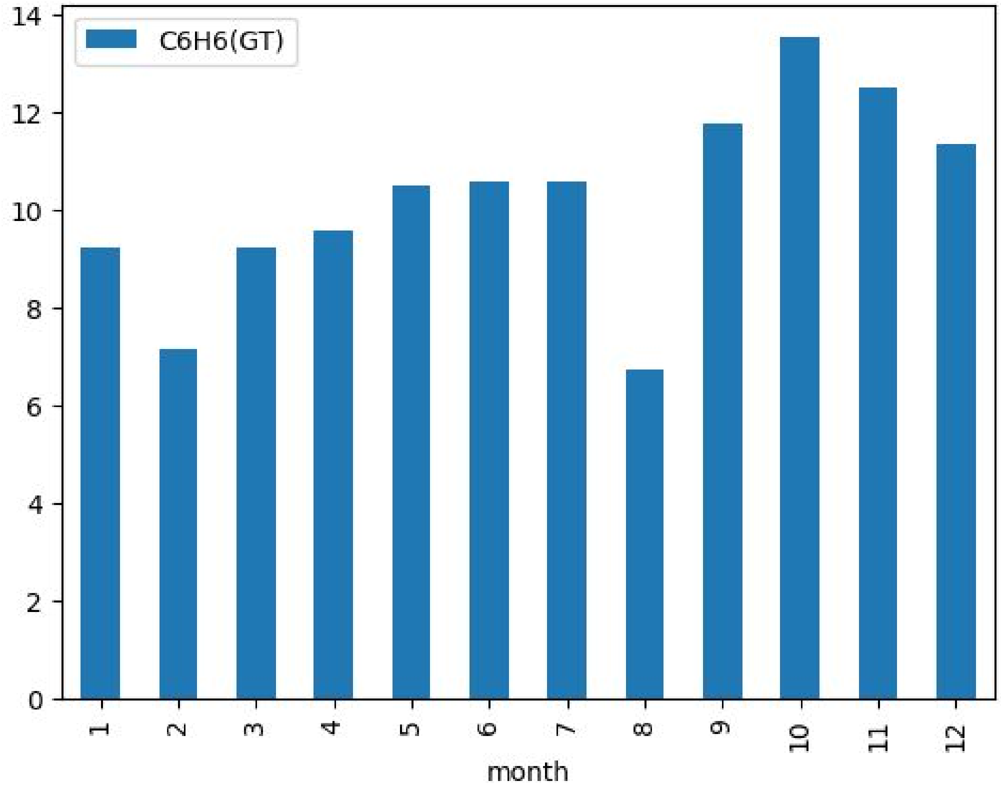

Figure 4.6 Graph of benzene concentration depending on the month

Graph (Figure 4.6) shows that the highest concentration of benzene occurred in October. Continuing this action, we also received a graph of the dependence of the weekday on concentration. He did not make any additional conclusions, except that on working days the concentration level is higher than on the weekend. We've also created a chart due to hours:

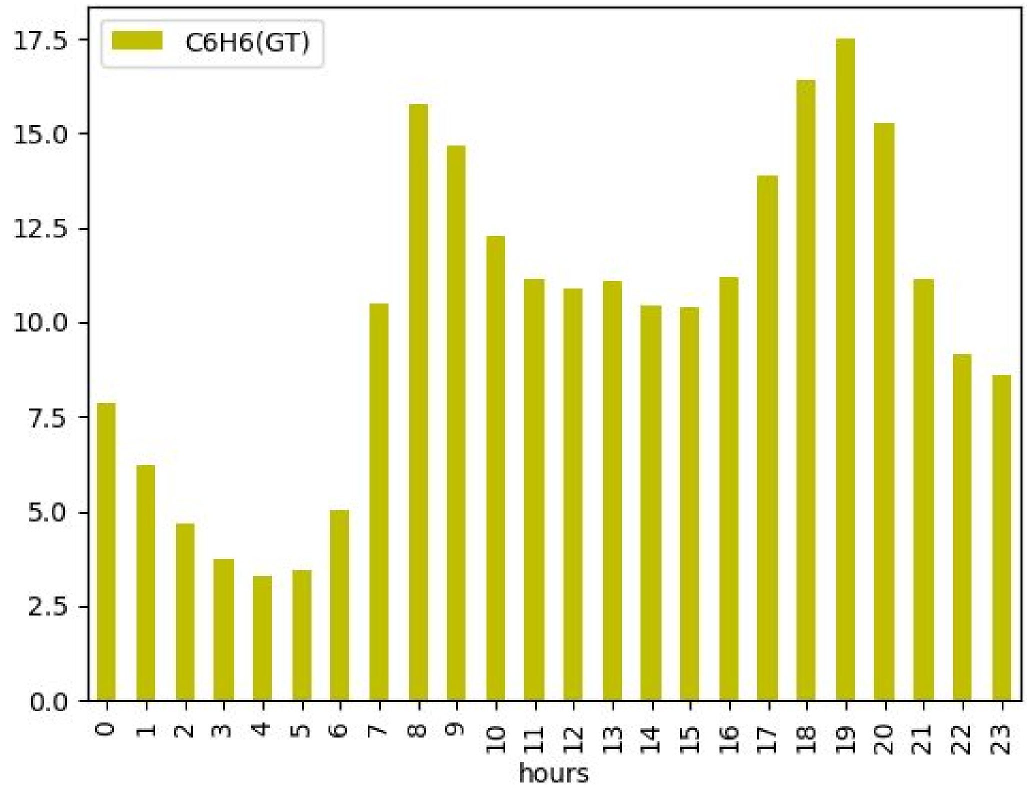

Figure 4.7 Graph of benzene concentration depending on the time of day

Here the conclusion is also obvious, the highest concentration is in the morning peak and in the late afternoon and evening.
Next, Pearson's [12] correlation was performed and the calculation of the final features for better accuracy.

5. Data modeling - a brief description of the methods and chosen modeling methodology

Model 1: In the first case, we used the variables RH, T and CO to build the model. Then the set of values was divided into training and test.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

Then training and test were done. The values received are below:

```
   Actual CO(GT)  Predicted CO(GT)
0            0.5              1.63
1            1.9              1.91
2            3.4              2.40
3            1.2              1.45
4            2.4              2.40
5            1.3              2.33
6            1.9              1.64
7            4.6              2.55
8            1.3              1.64
9            3.1              2.35
```

The R^2 coefficient was also calculated:
trainRegression.score (X_test, y_test)
The value in this model was: 0.1544555274683902

Errors that we've got:
Mean Squared Error: 1.779567238605898
Mean Absolute Error: 1.0011099195710456
Median Absolute Error: 0.7799999999999998

Value graph (50 values):

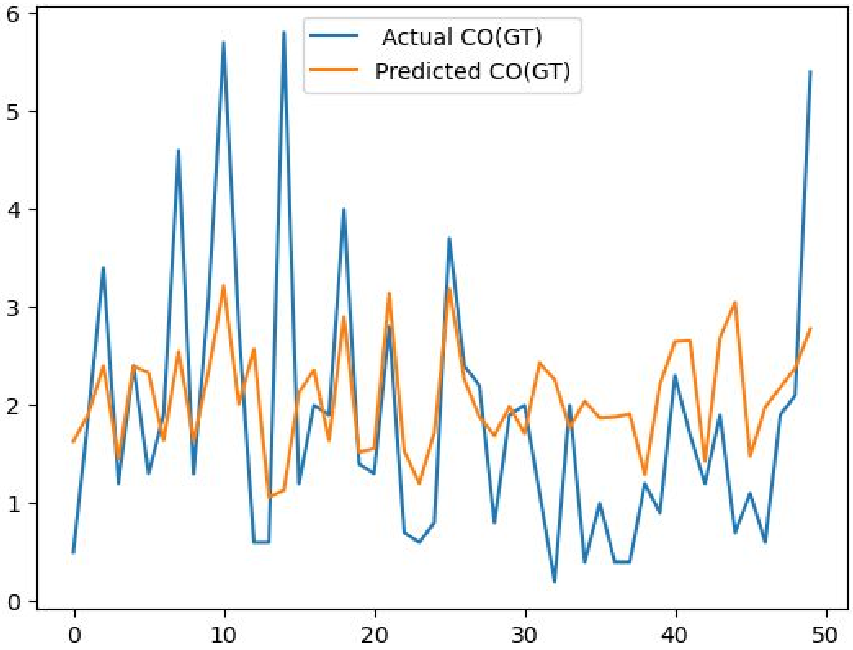

Figure 5.1 Comparison of actual and expected values

Model 2: To build model 2, we used the C6H6 variable and the data contained in the airData2 variable - i.e. the same data contained in the columns of the AirQuality_Cleared.csv file except the columns *'Date', 'Time', 'T', 'RH', 'AH' , 'NMHC (GT) "," C6H6 (GT)'.* As in the first model, here we divided the set of values into training and test:

```
X_train, X_test, y_train, y_test = train_test_split(airData2Values, C6H6, test_size=0.3, random_state=0)
```

However, in this case the test set was slightly increased (*test_size* = 0.3). Then the training and test were done:

```
Actual C6H6(GT)  Predicted C6H6(GT)
0            5.2          5.974521
1            1.6         -0.549634
2            2.4          1.604310
3            2.3          2.079293
4            3.0          2.507987
5           17.0         17.226395
6           14.6         15.293614
7           13.7         13.239920
8            5.2          6.362239
9            7.1          7.481272
```

The R^2 coefficient was also calculated:

```
linearRegression2.score(X_test, y_test)
```

The value in this model was: 0.9787568291122973

Errors that we’ve got:

```
Mean Squared Error:      1.1542890292986514
Mean Absolute Error:     0.8002900518899405
Median Absolute Error:   0.6506932639140158
```

Value graph (50 values):

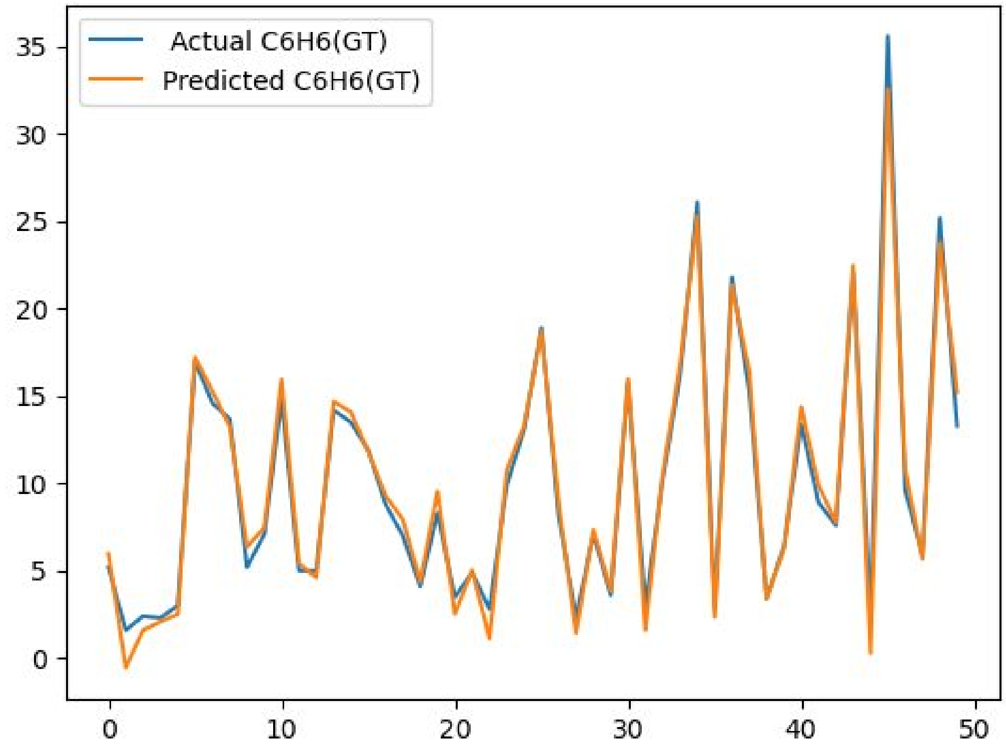

Figure 5.2 Comparison of actual and expected values

6. Results and conclusion

The data set on which we worked was a data set containing information about air quality, collected every hour for a year (March 2004 - February 2005) through a multi-sensor device.

The data set required preparation for analysis - the dataset contained empty records and missing values, which were supplemented by a correlation method.

Then we performed the analysis, initially extracting basic information about our data set, such as: minimum, maximum or standard deviation.

In the next part of the report, we performed a correlation to the 'CO' variable, which indicated links to other data. The first model pointed out that carbon monoxide pollution cannot be predicted on the basis of humidity and temperature. Observing the results obtained, we see a discrepancy between the values of Actual CO (GT) and Predicted CO (GT). Similarly, Mean Squared Error, Mean Absolute Error and Median Absolute Error error values have large values. This is information that the model did not cope with such data selection. This is also confirmed by chart Fig. 5.1, where we can see the difference in values easily. It is also worth adding that the R^2 coefficient is very small, which is a sign that this match is unsatisfactory.

The second model focused around C6H6 (Actual hourly average benzene concentration). It pointed out that the highest concentration prevails in the morning peak and in the late afternoon and evening. Analyzing the results obtained, we can see that this model made better work than the previous one. An important difference between them is the fact that the test set was slightly larger, here, the values of Actual C6H6 (GT) and Predicted C6H6 (GT) look better. Of course, they diverge to some extent. Error values Mean Squared Error, Mean Absolute Error and Median Absolute Error are not too large. The R^2 factor is very large - almost equal to 1, which means that it is a very good fit. Let's pay attention to the graph Fig. 5.2. We can see with the naked eye that most lines overlap.

The study of air quality based on data was an interesting and new experience for us. It motivated us to learn and deepen Python and its libraries. In our opinion, the issue of air quality is a very important topic. The world is constantly evolving, and with its development, unfortunately, new pollutants are also arriving, of which we do not even realize. And this topic is extremely important for us and for future generations. Undoubtedly, it is important to realize that air quality has an impact on our health.

7. Bibliography

[1]
Dane oraz ich opis: http://archive.ics.uci.edu/ml/datasets/Air+Quality
[2] https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dtypes.html
[3]https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isnull.html?highlight=isnull#pandas.DataFrame.isnull
[4]
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sum.html?highlight=sum#pandas.DataFrame.sum
[5]
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html?highlight=dropna#pandas.DataFrame.dropna
[6]
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html?highlight=corr#pandas.DataFrame.corr
[7]
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html?highlight=groupby#pandas.DataFrame.groupby
[8]
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.core.groupby.GroupBy.apply.html?highlight=apply#pandas.core.groupby.GroupBy.apply
[9]
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.core.groupby.DataFrameGroupBy.ffill.html?highlight=ffil#pandas.core.groupby.DataFrameGroupBy.ffill
[10]
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.core.groupby.DataFrameGroupBy.bfill.html?highlight=bfill#pandas.core.groupby.DataFrameGroupBy.bfill
[11]
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.describe.html?highlight=describe#pandas.DataFrame.describe
[12] https://pl.wikipedia.org/wiki/Wsp%C3%B3%C5%82czynnik_korelacji_Pearsona
[13] https://github.com/juliawenta/dataAnalysis
[14] https://seaborn.pydata.org/generated/seaborn.pairplot.html