Raport jakości powietrza

Alicja Szymikowska,
Julia Wenta
Wydział Fizyki Technicznej i Matematyki Stosowanej
Politechnika Gdańska, 2020

Spis treści

1. Wstęp, motywacja oraz cele

W obecnych czasach żyjemy w coraz bardziej zanieczyszczonym środowisku, ma to ogromny wpływ na nasze zdrowie i samopoczucie. Oddychanie jest podstawą funkcjonowania wszystkich organizmów żywych, jest to zatem zagadnienie dotyczące nas wszystkich.
Motywacją naszej pracy jest jednak przede wszystkim nabycie umiejętności obróbki, analizy, modelowania danych. Za cel obrałyśmy również lepsze poznanie bibliotek do języka Python, w tym Numpy, Pandas, Matplotlib. Celem była też chęć bliższego zapoznania się z tematem dotyczącym jakości powietrza.

2. Opis danych - struktura zbiorów, opis zmiennych, pochodzenie

Zbiór danych zawiera odpowiedzi uzyskane od wielosensorowego urządzenia umieszczonego na polu we włoskim mieście. Rejestrowane są średnie godzinne odpowiedzi wraz z referencjami stężenia gazu, pochodzące z certyfikowanego analizatora. [1]

Charakterystyka zestawu danych	*Wielowymiarowy, Szereg czasowy*	**Ilość rekordów/ instancji**	*9358*	**Obszar**	*Technologia*
Rodzaj danych	*Rzeczywiste*	**Ilość właściwości**	*15*	**Data**	*2016-03-23*
Zadanie	*Regresja*	**Brakujące wartości?**	*Tak (oznaczenie "-200")*		

Dane opisane są przez następujące cechy / właściwości: [1]

- *Date* - Data (DD/MM/YYYY)
- *Time* - Czas (HH.MM.SS)
- *CO(GT)* - Rzeczywiste uśrednione godzinne stężenie CO w mg / m^3 (analizator referencyjny)
- *PT08.S1* - (tlenek cyny) uśredniona co do godziny odpowiedź czujnika (nominalnie ukierunkowana na CO)
- *NMHC(GT)* - Rzeczywiste uśrednione godzinne ogólne stężenie węglowodorów niemetalicznych w mikrog / m^3 (analizator referencyjny)
- *C6H6(GT)* - Rzeczywiste uśrednione godzinne stężenie benzenu w mikrog / m^3 (analizator referencyjny)

- *PT08.S2* - Uśredniona co do godziny, odpowiedź czujnika (nominalnie docelowy NMHC)
- *NOx(GT)* - Rzeczywiste uśrednione godzinne stężenie NOx w ppb (analizator referencyjny)
- *PT08.S3* - (tlenek wolframu) uśredniona co do godziny, odpowiedź czujnika (nominalnie docelowy NOx)
- *NO2(GT)* - Rzeczywiste uśrednione godzinne stężenie NO2 w mikrog / m^3 (analizator referencyjny)
- *PT08.S4* - (tlenek wolframu) uśredniona co do godziny odpowiedź czujnika (nominalnie docelowy NO2)
- *PT08.S5* - (tlenek indu) uśredniona co do godziny odpowiedź czujnika (nominalnie docelowy O3)
- *T* - Temperatura w °C
- *RH* - Względna wilgotność (%)
- *AH* - Wilgotność właściwa

3. Opis procesu przygotowywania danych do analizy - kolejne kroki

Dane, które wykorzystałyśmy do analizy pochodzą ze strony zawierającej zestawy danych[1]. Dzięki tej stronie, miałyśmy możliwość pobrania danych w pliku zip. W nim znajdowały się dane zapisane w plikach csv oraz xlsx. Dzięki temu bez najmniejszych przeszkód mogłyśmy zaimportować dane do naszej aplikacji napisanej w języku Python. Do wczytywania danych została wykorzystana biblioteka Pandas.

Po wczytaniu danych, sprawdziłyśmy ilość wierszy w których znajdowały się brakujące dane, ich ilość wyniosła 8530. Bardzo znacząca liczba spowodowała, że konieczne było przygotowanie danych do analizy w taki sposób aby, mogły zostać poddane późniejszej obróbce.
Na początku, przy pomocy funkcji *dtypes* [2] sprawdziłyśmy typy danych - część danych posiadała typ *object*, a druga typ *float64*. Typ danych, dla wszystkich cech, za wyjątkiem Daty oraz Czasu zmieniłyśmy na *float64*.
Następnie dzięki funkcjom *isnull()* [3]*.sum()*[4] otrzymałyśmy informacje o pustych wierszach oraz kolumnach w naszym zestawie danych. Zestaw posiadał dwie puste kolumny o nazwie “Unnamed”, które zostały usunięte. Pozostałe kolumny zawierały po 114 pustych wierszy, te wiersze również zostały usunięte, za pomocą komendy: *air_data = air_data.dropna(how="all")* [5]*,* gdzie *air_data* to nasz zestaw danych.
Puste serie danych prawdopodobnie były spowodowane odcięciem od zasilania urządzenia sensorowego.
Po usunięciu pustych komórek, przeszłyśmy do komórek, które zawierały brakujące wartości, oznaczone przez “-200”. Na początek wszystkie takie wartości,

zamieniłyśmy na puste komórki, aby po skorelowaniu uzupełnić je nowymi wartościami.
Za pomocą metody *corr()*[6] sprawdzałyśmy kolejne korelacje między danymi.
Wyniki korelacji zostały umieszczone w formie komentarza w pliku clearData.py.
Przykładowo, największa korelacja CO(GT) występuje z C6H6(GT), pozostałe korelacje widoczne są na rysunku 3.1.

Rys 3.1 Wyniki korelacji pomiędzy danymi

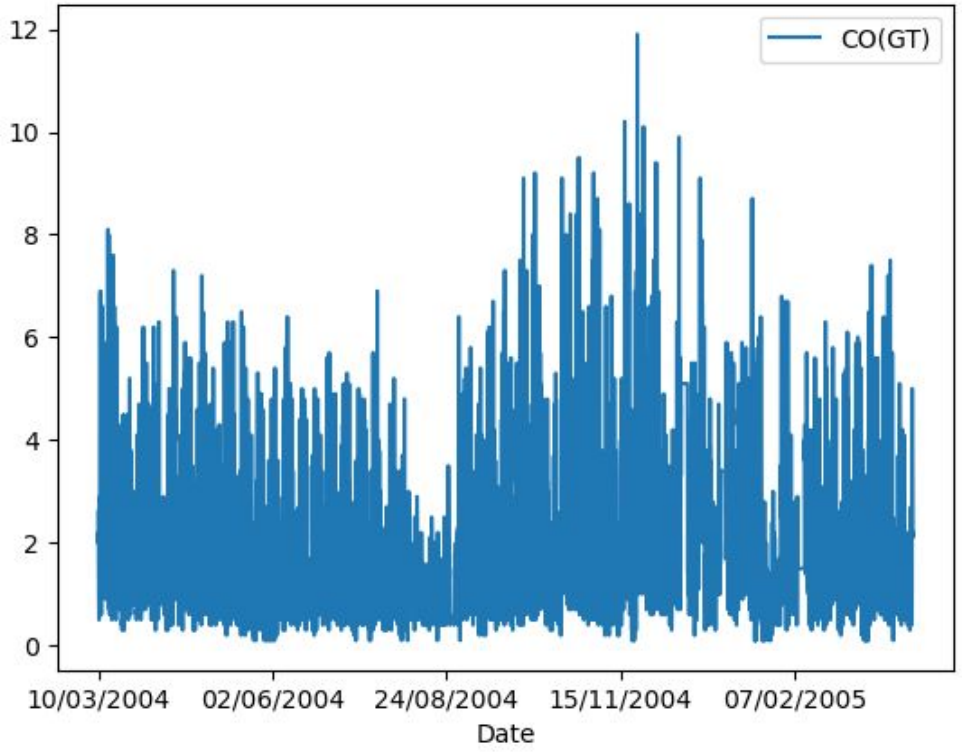
Odpowiednie wartości dla każdej z komórek zostały uzyskane i uzupełnione przy pomocy metod *groupby*[7], *apply*[8], *ffill*[9] oraz *bfill*[10].
Po wykonaniu wyżej opisanych kroków, dane nie zawierały pustych rekordów. Uzupełnione dane zostały zapisane do pliku csv w celu dalszego wykorzystania podczas analizy i modelowania.

4. Analiza danych - przyjęte założenia, krótki opis metod i obranej metodologii analizy

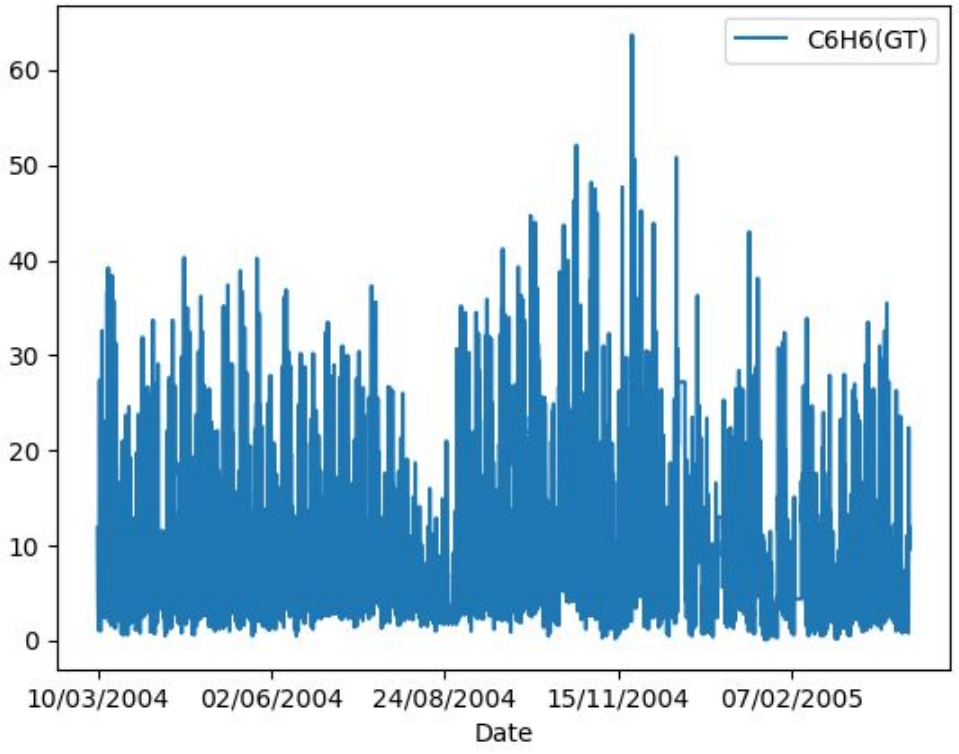
Model 1: Po przygotowaniu danych w pliku clearData.py mogłyśmy przystąpić do analizy danych. Ponownie możemy bliżej przyjrzeć się danym zawartym w pliku AirQuality_Cleared.csv. Po wczytaniu tego pliku wykonano na nim metodę *describe()*[11]. Dzięki temu otrzymałyśmy niezbędne informacje statystyczne dotyczące danych zawartych w pliku, między innymi: średnią wartości, odchylenie standardowe, wartość minimalną, wartość maksymalną i kwartyle 25%, 50% i 75%. Następnie, aby ułatwić sobie odczytanie danych, zwizualizowałyśmy je za pomocą wykresów.Poniższe wykresy pokazują przybliżony trend wartości według dat,

poniższy wykres (Rys 4.1) można zastąpić innymi datami w celu znalezienia wyraźnych cech.

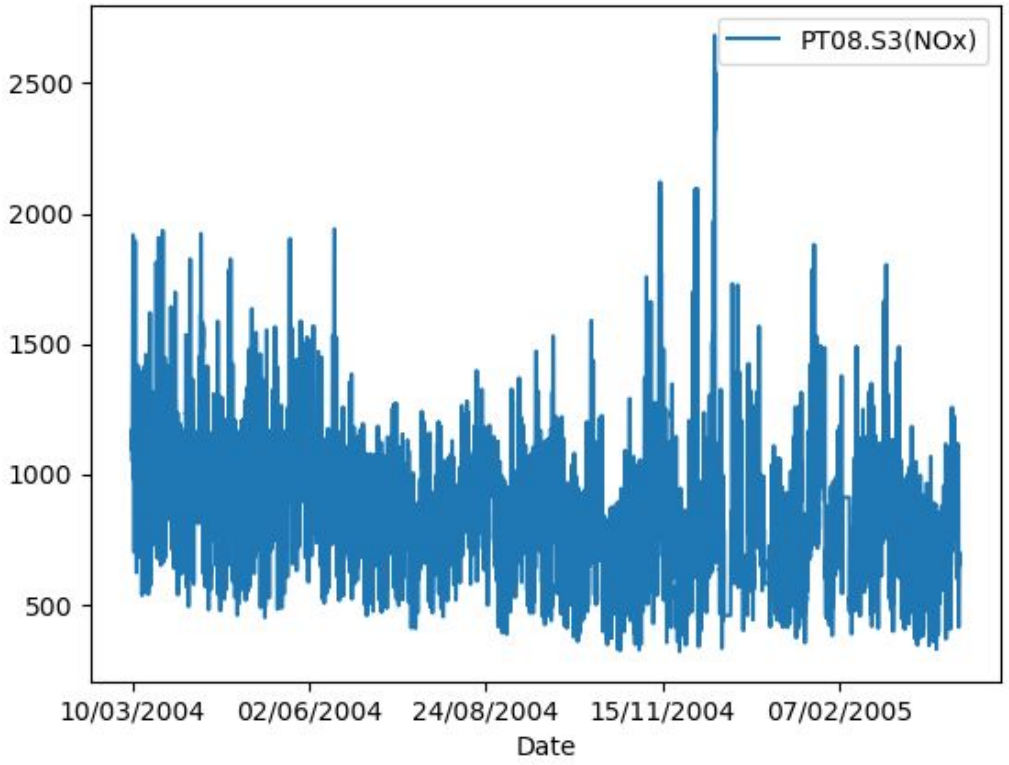
Poniżej znajdują się trzy wybrane wykresy:



Rys 4.1 Wykres zależności dwutlenku węgla od daty



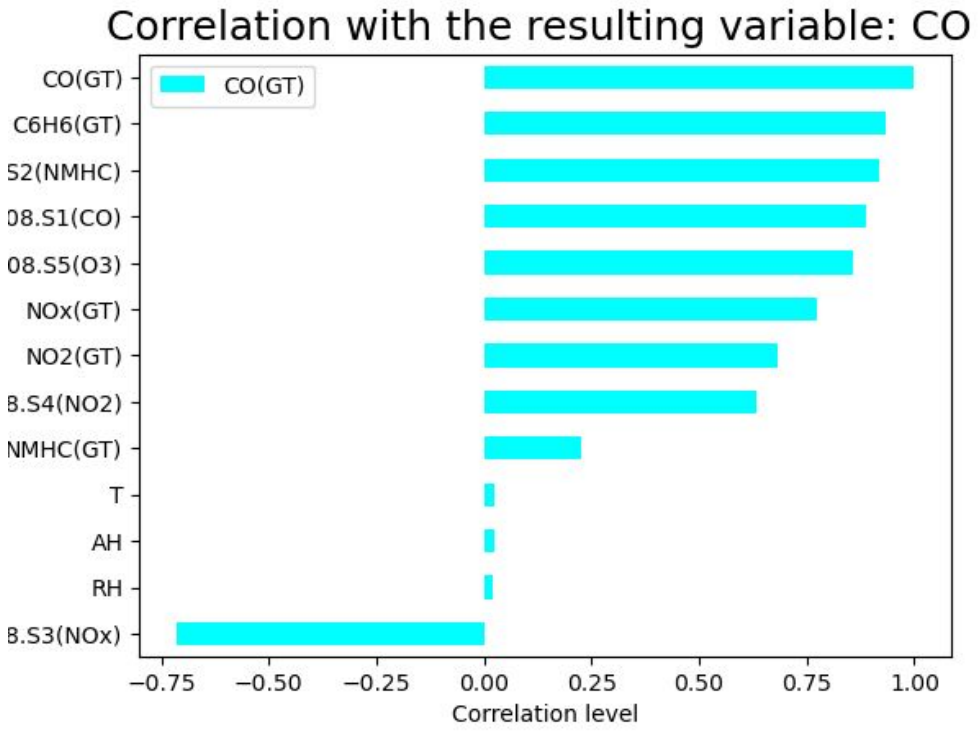
Rys 4.2 Wykres zależności stężenia benzenu od daty



Rys 4.3 Wykres zależności tlenek wolframu od daty

Wszystkie wykresy zostały zawarte w repozytorium[13]. Kolejnym krokiem było dodanie nowej kolumny *'dateAndTime'*. Zawiera ona dane z kolumny *'date'* oraz *'time'*. Następnie nowa kolumna została sformatowana. Na podstawie tego kroku, wykonano kolejne, gdzie nastąpił podział daty na miesiące, dni tygodnia a także godziny. Wynika to z tego, że jedną z najważniejszych zmiennych opisujących w regresji liniowej jest czas. Większość zjawisk sztucznych i naturalnych działa w cyklach godzinowych, dziennych i miesięcznych.

W dalszej części wykonałyśmy korelację względem zmiennej 'CO'. Wizualizuje to poniższy wykres (Rys. 4.4).



Rys 4.4 Wykres korelacji dla zmiennej CO

Chcemy sprawdzić w jaki sposób pogoda i czas wpływają na poziom zanieczyszczenia powietrza. Zatem będą interesować nas trzy najmniej skorelowane zmienne ‘T’,’AH’,’RH’.
Każda z tych zmiennych została sprawdzona za pomocą funkcji *correlationFunction* oraz *checkShift*. Korzystając z tych funkcji otrzymano:

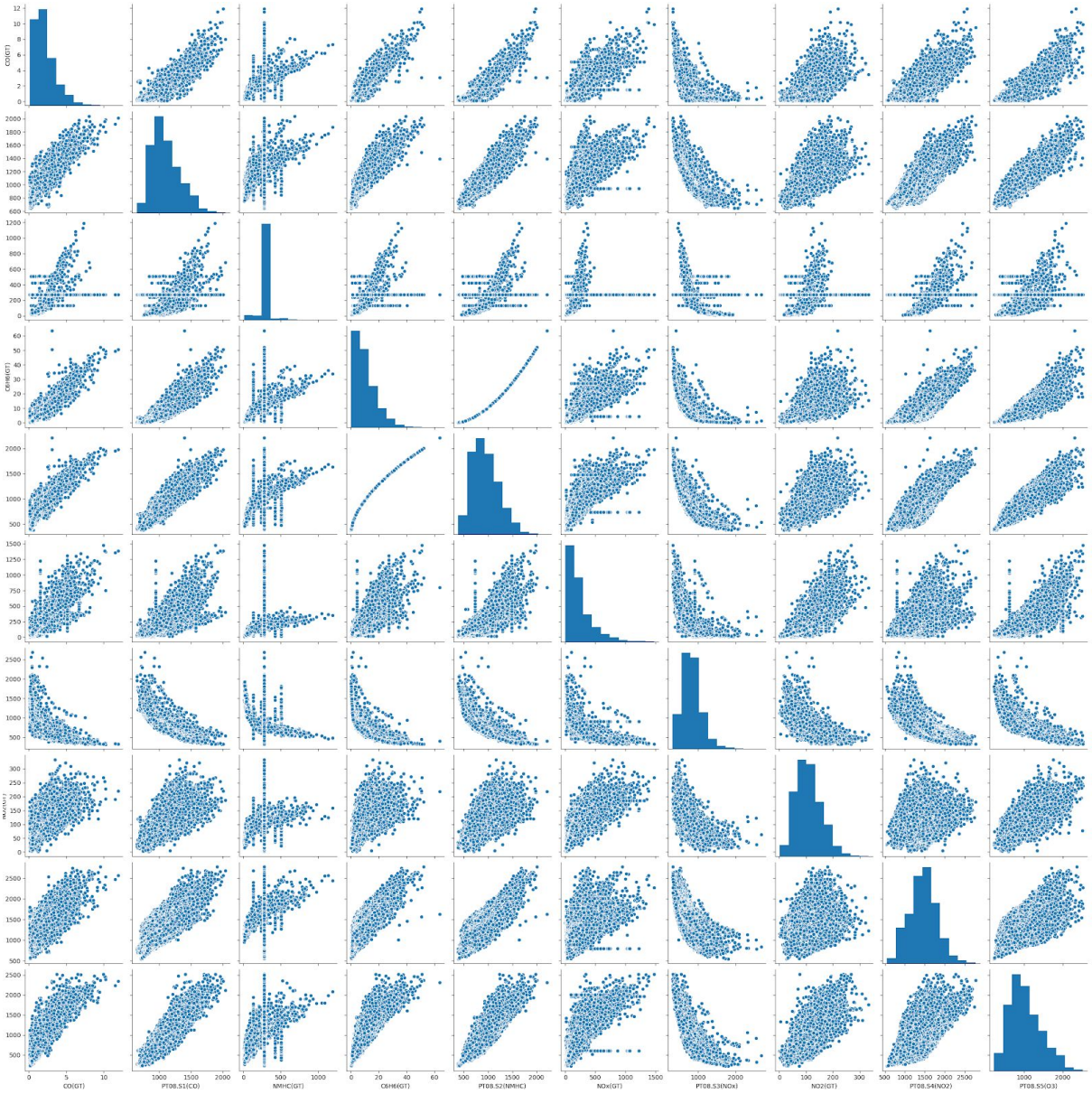
Optimal shift for RH: 12
0.3920431367189807

Optimal shift for AH: 12
0.04375636410267759

Optimal shift for T: 12
-0.22446569561762525

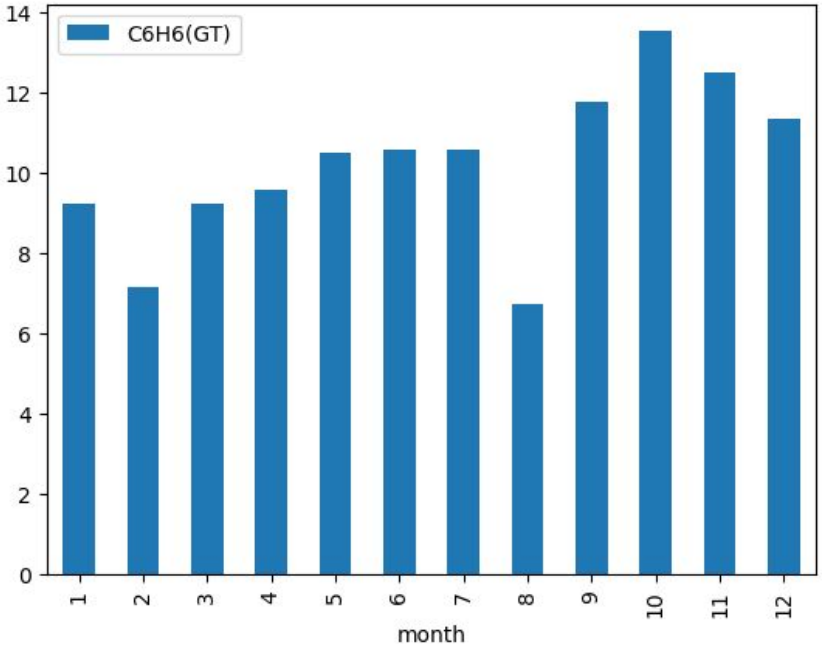
Z powyższego wykresu (Rys. 4.4) wynika, że zmienne ‘T’ oraz ‘RH’ korelują ze zmienną CO (GT) po dwunastu godzinach od zanieczyszczenia, natomiast zmienna ‘AH’ wcale nie koreluje ze zmienną CO(GT). Wykorzystałyśmy metodę *shiftDataFrame12* do stworzenia nowego DataFrame, dzięki niej otrzymałyśmy powyższe wyniki.

Model 2: Podjęłyśmy również próbę analizy danych biorąc pod uwagę zmienną C6H6(Rzeczywiste uśrednione godzinne stężenie benzenu). Warto zwrócić uwagę, że benzen jest toksyczny i szkodliwy dla człowieka.
W pierwszej kolejności wykonałyśmy wykresy przy pomocy funkcji *pairpolt* [14].



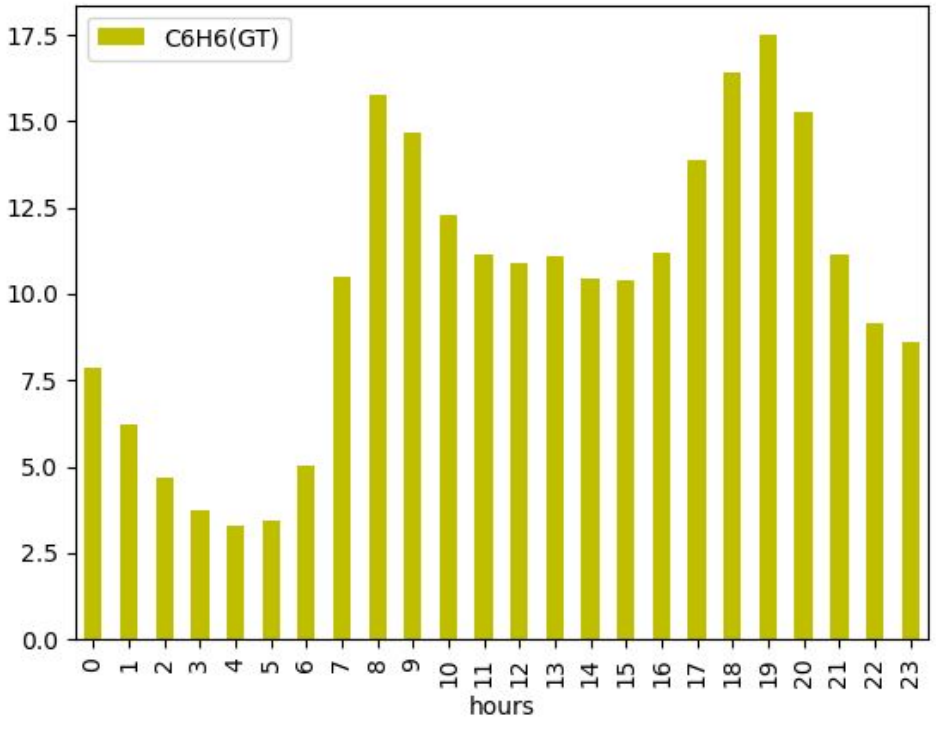
Rys 4.5 Wykres wskazujący na liniowość danych

Dzięki temu zabiegowi, od razu widać czy dane są liniowe. W następnym kroku wykorzystałyśmy stworzone wcześniej parametry *‘dateAndTime”.* Uzyskałyśmy wykresy zależności poziomu stężenia od miesiąca:



Rys 4.6 Wykres zależności poziomu stężenia benzenu w zależności od miesiąca

Co ciekawe wynika z niego, że największe stężenie benzenu występowało w październiku. Kontynuując to działanie otrzymałyśmy również wykres zależności dnia tygodnia od stężenia. Nie wniósł on żadnych dodatkowych wniosków, poza tym, że w dni robocze poziom stężenia jest większy niż w weekend. Stworzyłyśmy również wykres w podziale na godziny:



Rys 4.7 Wykres zależności poziomu stężenia benzenu w zależności od pory dnia

Tutaj wniosek również jest oczywisty, ponieważ największe stężenie panuje w porannym szczycie oraz późnym popołudniem i wieczorem.
W dalszej kolejności została wykonana korelacja *Pearsona* [12] oraz obliczenie ostatecznych cech dla lepszej dokładności.

5. Modelowanie danych - przyjęte założenia, krótki opis metod i obranej metodologii budowania modeli

Model 1: W pierwszym przypadku, do budowy modelu wykorzystałyśmy zmienne RH, T oraz CO. Następnie dokonano podziału zbioru wartości na treningowy i testowy.

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

Potem wykonano trening i test zbioru. Poniżej zostały zamieszczone otrzymane wartości:

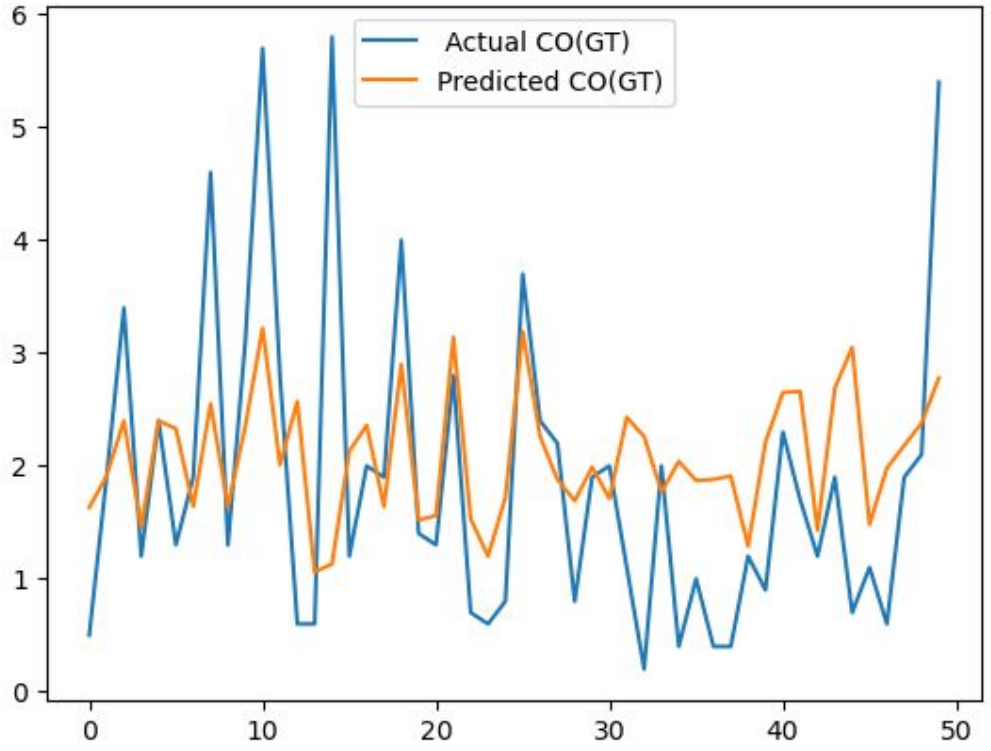
	Actual CO(GT)	Predicted CO(GT)
0	0.5	1.63
1	1.9	1.91
2	3.4	2.40
3	1.2	1.45
4	2.4	2.40
5	1.3	2.33
6	1.9	1.64
7	4.6	2.55
8	1.3	1.64
9	3.1	2.35

Obliczono także współczynnik R^2 :
trainRegression.score(X_test, y_test)
Jego wartość w tym modelu wyniosła: 0.1544555274683902

Oraz błędy jakie otrzymałyśmy:

Mean Squared Error: 1.779567238605898
Mean Absolute Error: 1.0011099195710456
Median Absolute Error: 0.7799999999999998

I wykres wartości (50 wartości):



Rys 5.1 Porównanie wartości rzeczywistych z oczekiwanymi

Model 2: Do budowy modelu 2 wykorzystaliśmy zmienną C6H6 oraz dane zawarte w zmiennej airData2 - czyli te same dane, zawarte w kolumnach pliku AirQuality_Cleared.csv oprócz kolumn *'Date', 'Time', 'T','RH','AH','NMHC(GT)','C6H6(GT)'*. Tak jak w pierwszym modelu i tutaj dokonałyśmy podziału zbioru wartości na treningowy i testowy:

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

Jednakże w tym przypadku zbiór testowy został nieznacznie powiększony(*test_size=0.3*).
Następnie dokonano treningu i testu:

	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

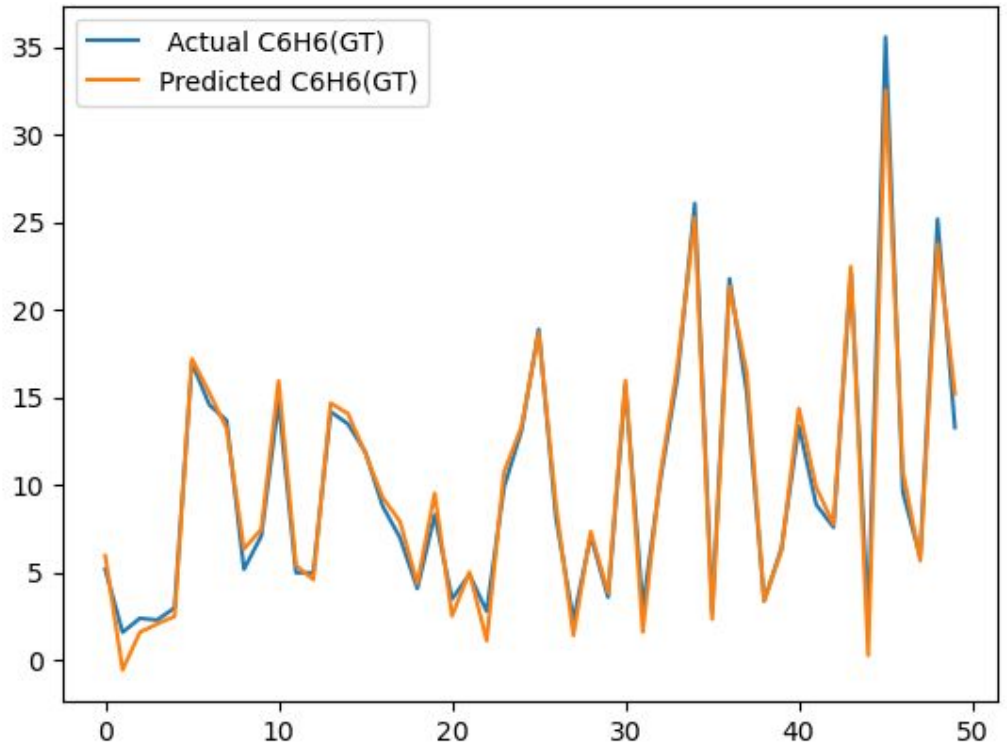
7	13.7	13.239920
8	5.2	6.362239
9	7.1	7.481272

Obliczono także współczynnik R^2 :
linearRegression2.score(X_test, y_test)

Jego wartość w tym modelu wyniosła: 0.9787568291122973

Oraz błędy jakie otrzymano:
Mean Squared Error: 1.1542890292986514
Mean Absolute Error: 0.8002900518899405
Median Absolute Error: 0.6506932639140158

Poniżej przedstawiono wykres (50 wartości):



Rys 5.2 Porównanie wartości rzeczywistych z oczekiwanymi

6. Rezultaty, wnioski i ich dyskusja

Zestawem danych, na którym pracowałyśmy, był zbiór danych zawierający informacje o jakości powietrza, zbierane co godzinę przez rok (Marzec 2004 - Luty 2005) poprzez wielosensorowe urządzenie.
Zestaw danych wymagał przygotowania do analizy - dataset zawierał puste rekordy, oraz brakujące wartości, które zostały uzupełnione za pomocą metody korelacji.
Następnie dokonałyśmy analizy, początkowo wydobyłyśmy podstawowe informacje o naszym zestawie danych, takie jak: wartości minimalne, maksymalne czy odchylenie standardowe.
W dalszej części raportu wykonałyśmy korelację względem zmiennej 'CO', która wskazała na powiązania z pozostałymi danymi. Model pierwszy wskazał na to, że zanieczyszczenia tlenkiem węgla nie można przewidzieć na podstawie wilgotności i temperatury. Obserwując otrzymane wyniki widzimy rozbieżność pomiędzy wartościami Actual CO(GT) i Predicted CO(GT). Tak samo wartości błędów *Mean Squared Error, Mean Absolute Error i Median Absolute Error* mają duże wartości. Jest to informacja, że model nie poradził sobie przy takim doborze danych. Utwierdza nas w tym także wykres Rys.5.1.gdzie gołym okiem widzimy różnice wartości. Warto również dodać że współczynnik R^2 jest bardzo mały, co jest oznaką, że jest to dopasowanie niezadowalające.
Drugi model skupiał się wokół C6H6(Rzeczywiste uśrednione godzinne stężenie benzenu). Wskazał on, że największe stężenie panuje w porannym szczycie oraz późnym popołudniem i wieczorem. Analizując otrzymane wyniki widzimy, że ten model poradził sobie lepiej niż poprzedni. Ważną różnicą między nimi jest fakt, że zbiór testowy był trochę większy. Tutaj wartości *Actual C6H6(GT)* i *Predicted C6H6(GT)* wyglądają lepiej. Oczywiście odbiegają od siebie w pewnym stopniu. Wartości błędów *Mean Squared Error, Mean Absolute Error i Median Absolute Error* nie są zbyt duże. Współczynnik R^2 jest bardzo jest duży - prawie równy 1, co oznacza, że jest to bardzo dobre dopasowanie. Zwróćmy jeszcze uwagę na wykres Rys.5.2. Gołym okiem możemy dostrzec, że w większości linie pokrywają się.
Badanie jakości powietrza na podstawie danych było ciekawym i nowym doświadczeniem dla nas. Zmotywowało nas do nauki i zgłębienia języka Python oraz jego bibliotek. W naszej opinii zagadnienie jakości powietrza jest bardzo ważnym tematem. Świat cały czas się rozwija, a wraz z jego rozwojem niestety także przybywa nowych zanieczyszczeń, z których nawet nie zdajemy sobie sprawy. A jest to temat niezwykle istotny dla nas jak i dla przyszłych pokoleń. Bez wątpienia ważne jest to, aby uświadomić sobie, że jakość powietrza ma wpływ na nasze zdrowie.

7. Bibliografia

[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