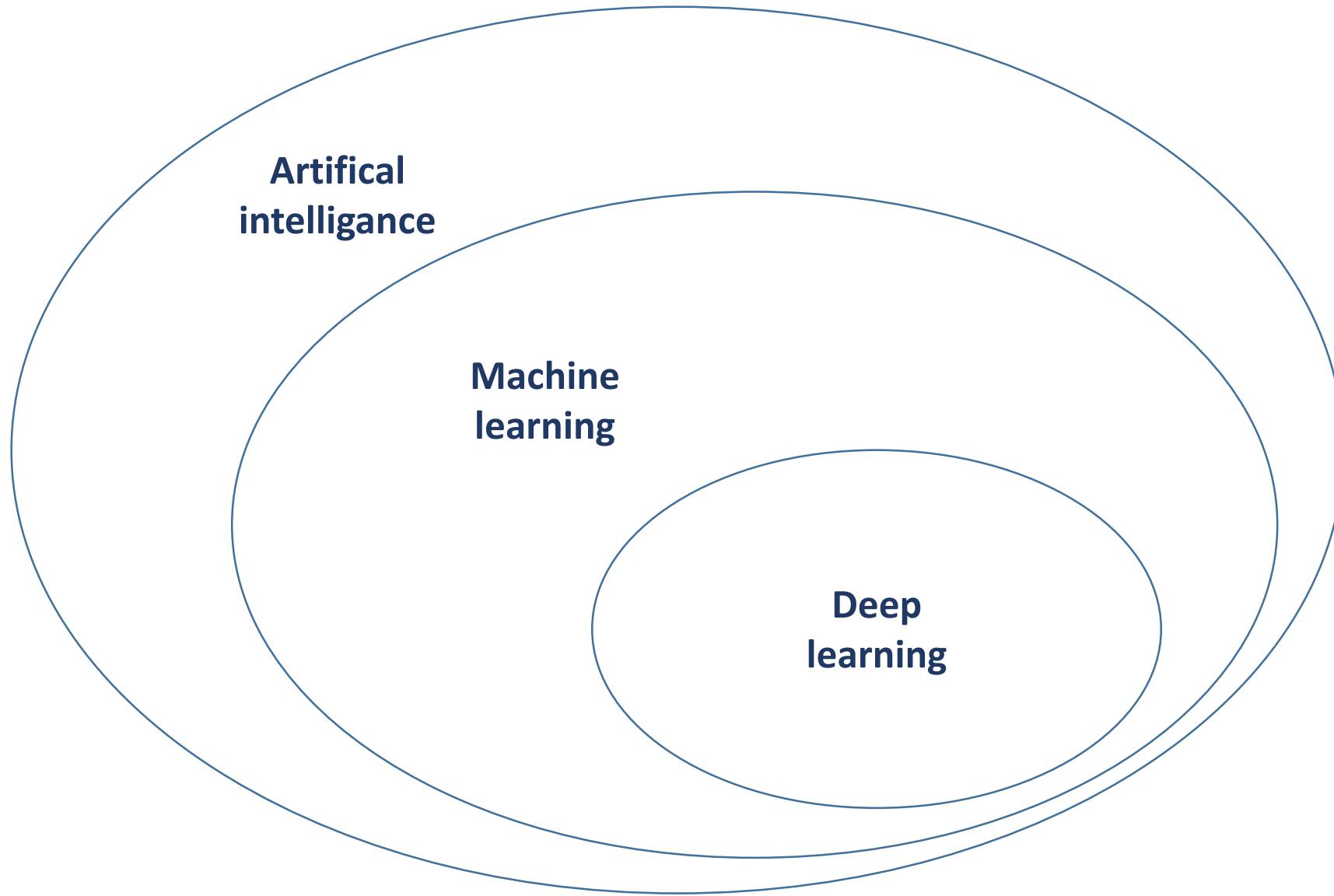


Geo data science – nowe spojrzenie na GIS

Jacek Urbański
Centrum GIS UG

Nowe możliwości jakie daje powiązanie **GIS** z **AI** (sztuczną inteligencją)

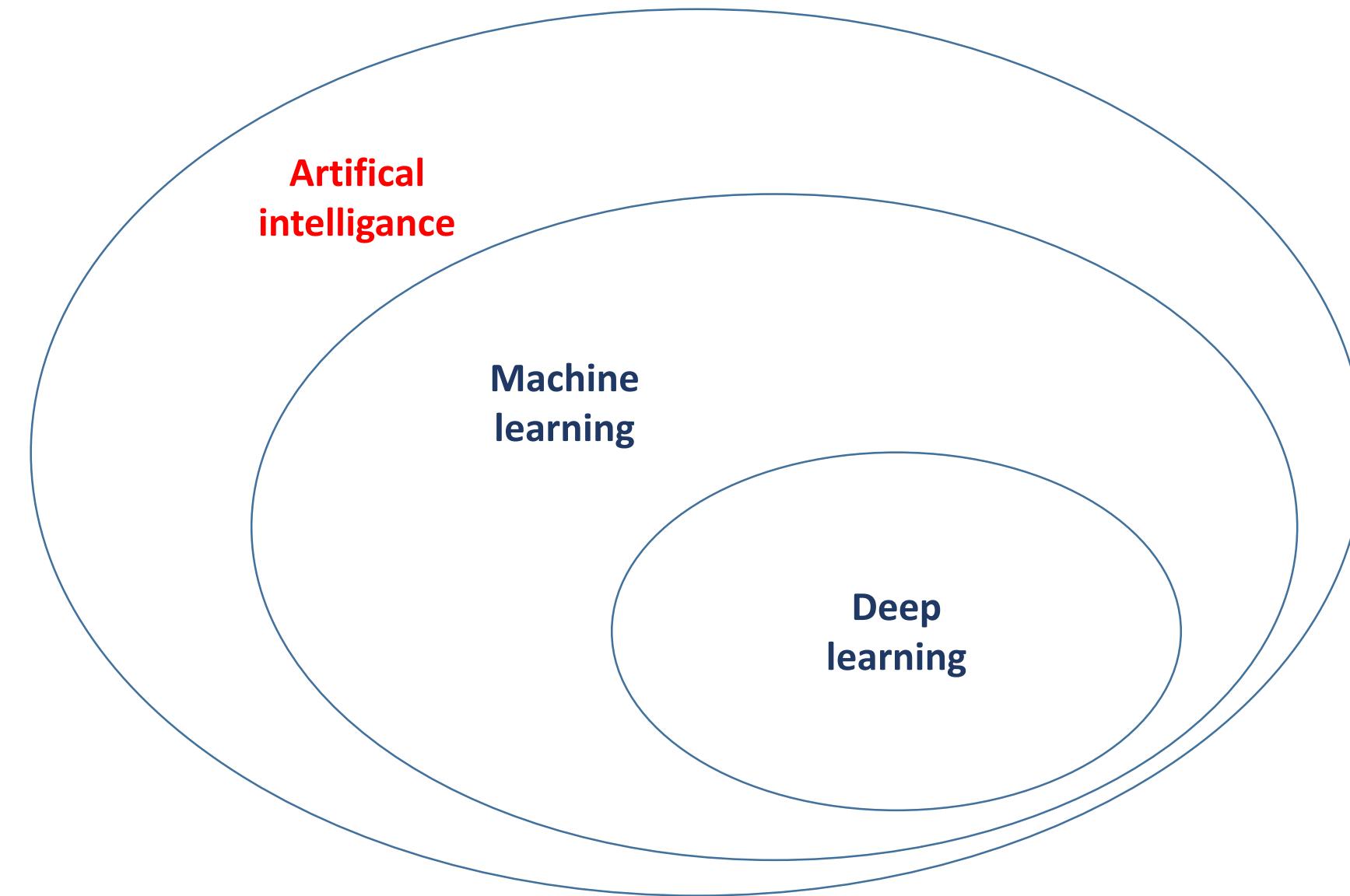
Co to jest AI ?



Artifical
intelligence

Machine
learning

Deep
learning



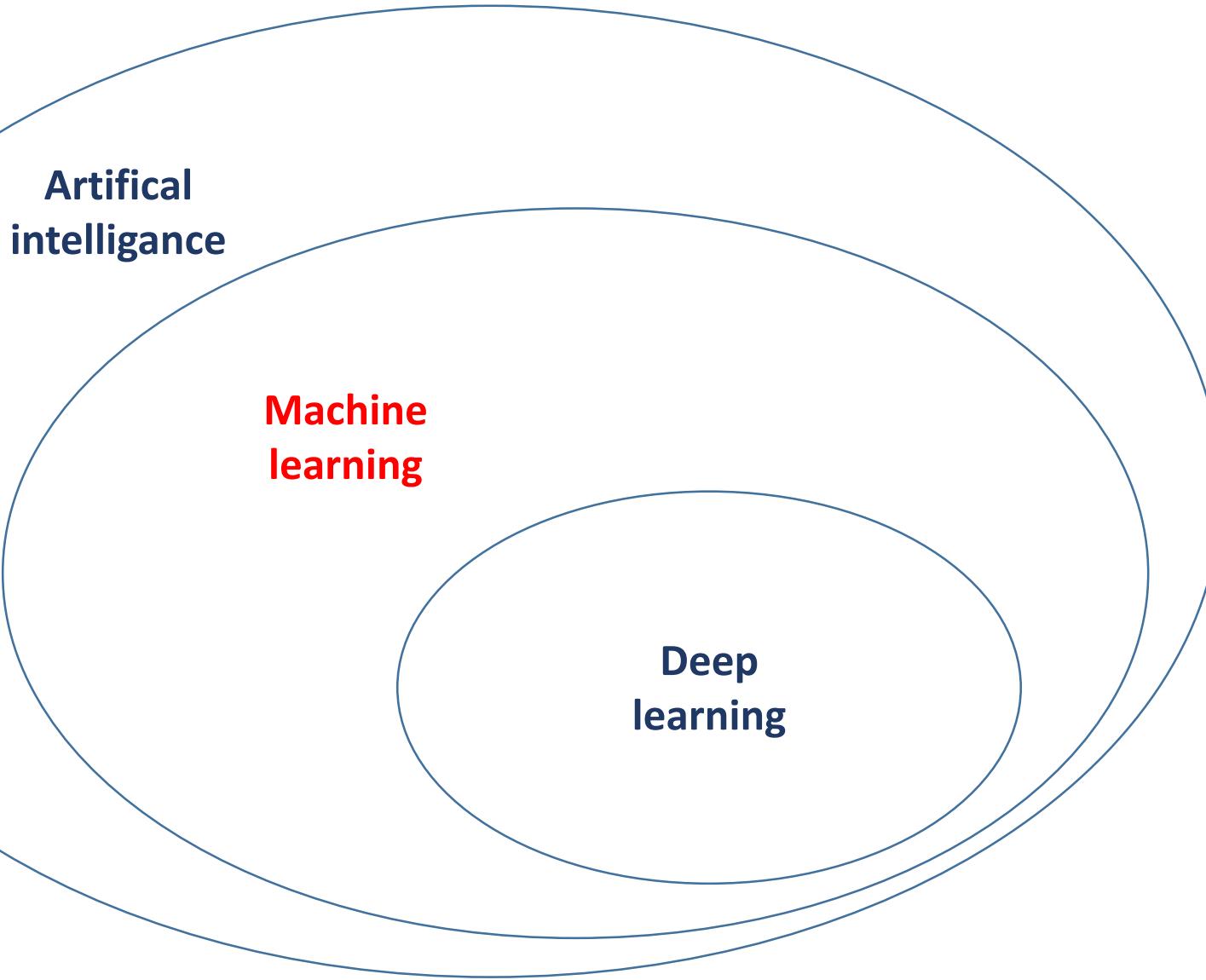
AI – sztuczna inteligencja

Rok urodzenia ok. 1950

Cel: próba automatyzacji zadań wymagających intelektu człowieka

Hipoteza: można to osiągnąć za pomocą zaprogramowania dużej liczby zasad do symulowania wiedzy = *symbolic AI*. Stała się ona podstawą rozkwitu *Expert systems* w latach 1980.

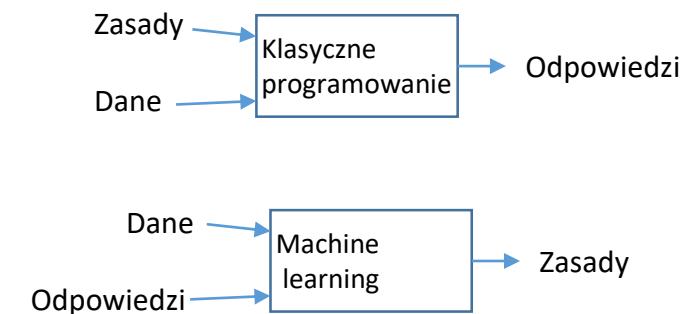
Ograniczenia: działała dobrze dla dobrze zdefiniowanych problemów logicznych (gra w szachy) ale już nie przy problemach rozmytych (klasyfikacja obrazów).



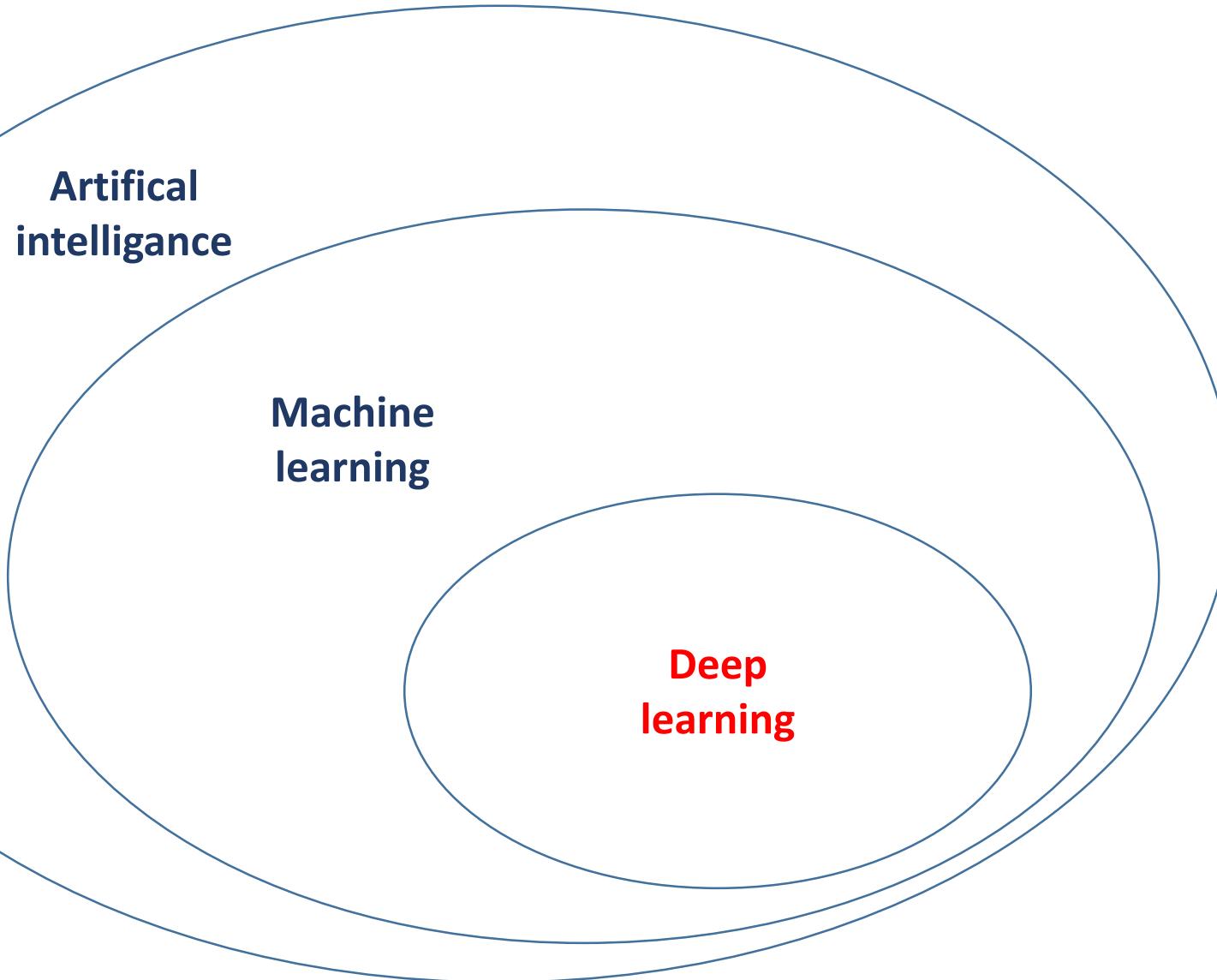
Lata 1990 do czasów obecnych.

Pytanie: Czy zamiast ręcznego wprowadzania zasad, komputer może je sam wyprowadzić na podstawie analizy dostępnych danych.

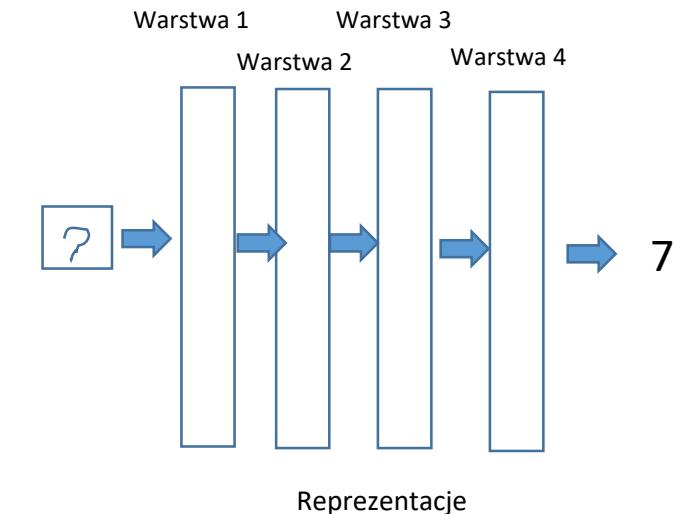
Nowe zasady programowania:



Wykorzystanie zbiorów treningowych do budowy modelu (zasad) i jego testowanie za pomocą zbiorów testowych.



Lata 2010 do czasów obecnych
Wykorzystanie sieci neuronowych
(neural networks)



Sieć neuronowa transformuje cyfrowy obraz w kolejne reprezentacje coraz bardziej różniące się od początkowego obrazu i coraz bardziej informacyjne co do wyniku.

Wieloetapowa droga do poznania reprezentacji danych.

Zastosowanie AI

DATA SCIENCE

DATA ENGINEERING

COMPUTATIONAL DATA SCIENCE

POZYSKIWANIE

PRZYGOTOWANIE

ANALIZA

RAPORTOWANIE

DZIAŁANIE

PLIKI TEKSTOWE

BAZY DANYCH

Web serwisy

No SQL storage

Eksploracja EDA

PREPROCESSING

Classification

Regression

Clustering

Graph Analytics

.....

Korelacja

Trend

Outliers

Statystyki

Czyszczenie

Duplicate

Inconsistent

Missing

Outlier

Transformacja

Scailing

Transformation

Feature selection

Dimensions PCA

Wizualizacja

Histogramy

Boxplots

Line graph

GIS

WSTĘPNE PRZETWARZANIE / SPATIAL DATA ENGINEERING

ANALIZA PRZESTRZENNA

POZYSKIWANIE

PRZYGOTOWANIE

ANALIZA

TWORZENIE MAP

PLIKI TEKSTOWE

BAZY Geo-DANYCH

WIZUALIZACJA

PREPROCESSING

Mapy

Czyszczenie

Histogramy

Duplicate

Boxplots

Inconsistent

Line graph

Missing

Outlier

EDB

Outliers

Statystyki

Struktura funkcjonalna Data Science i GIS

Struktura danych DATA SCIENCE - **Tidy data - organizacja danych w tablicach**

Podsumowanie:

1. Każda zmienna, która jest mierzona (pozyskiwana) powinna być w jednej kolumnie.
2. Każda oddzielnna obserwacja szeregu zmiennych powinna tworzyć oddzielny wiersz.
3. Dla każdego typu zmiennych powinna być oddzielna tabela.
4. Jeżeli posiadamy parę tablic, powinny zawierać kolumnę, która umożliwi ich łączenie.

Tablica
Atrybutowa
GIS

country	year	cases	population
Afghanistan	1990	745	1507071
Afghanistan	2000	2566	2059360
Brazil	1999	37737	17206362
Brazil	2000	80488	17404898
China	1999	212258	127215272
China	2000	21766	128042583

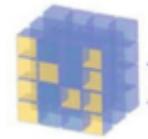
variables

country	year	cases	population
Afghanistan	1990	745	1507071
Afghanistan	2000	2566	2059360
Brazil	1999	37737	17206362
Brazil	2000	80488	17404898
China	1999	212258	127215272
China	2000	21766	128042583

observations

country	year	cases	population
Afghanistan	1990	745	1507071
Afghanistan	2000	2566	2059360
Brazil	1999	37737	17206362
Brazil	2000	80488	17404898
China	1999	212258	127215272
China	2000	21766	128042583

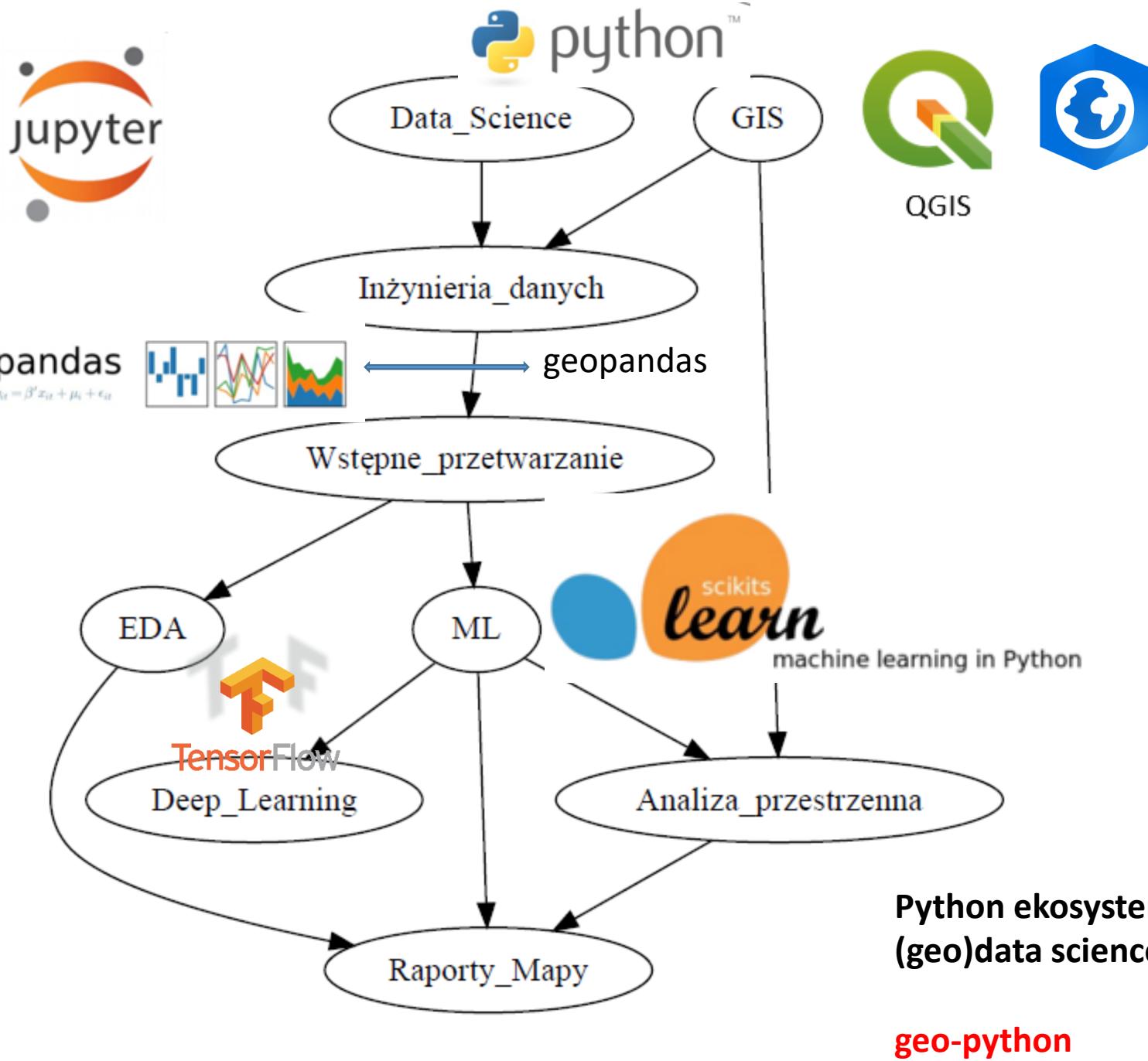
values

 NumPy

 SciPy

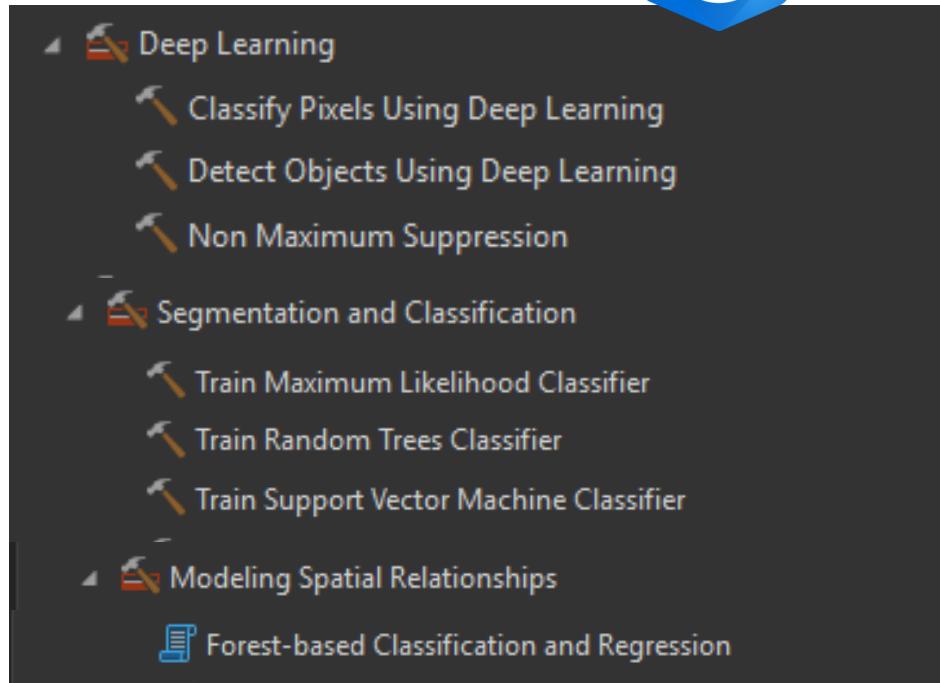
 scikits-image
image processing in python

 matplotlib



Data Science → GIS

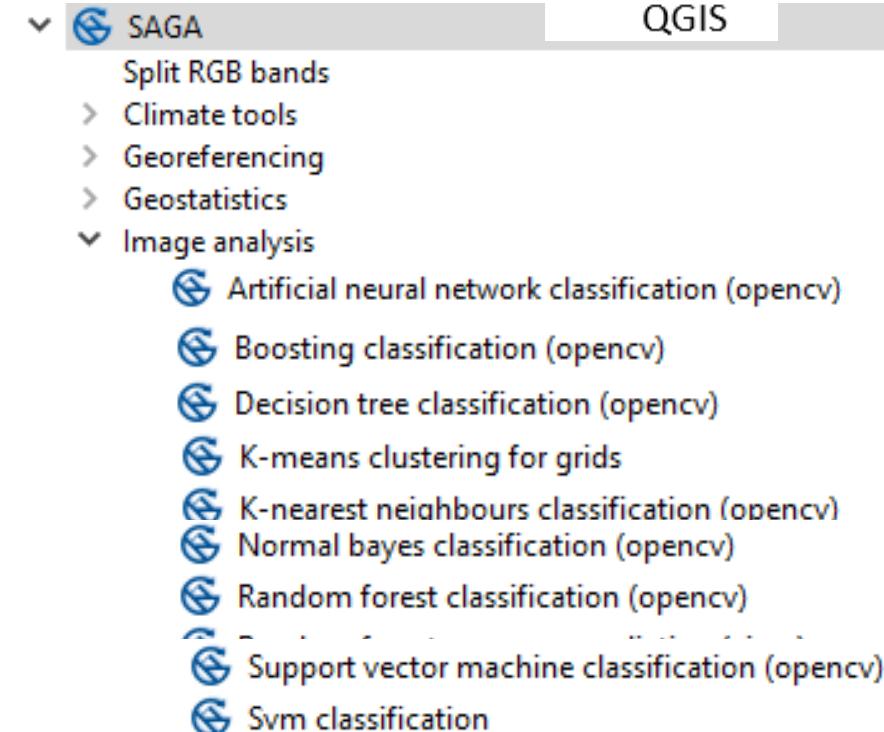
Arc GIS Pro



Algorytmy (metody) data science



QGIS



GIS → Data Science

Atrybuty (zmienne) przestrzenne dla obserwacji



PROBLEM (analiza SPM10 w Polsce)

1 Countrycode,Namespace,AirQualityNetwork,AirQualityStation,AirQualityStationEoICode,
2 PL,PL.CIEP.AQ,NET_PL_MP_AQ,STA_PL0126A,PL0126A,SPO_PL0126A_5_001,SPP_PL0126A_5_001,
3 PL,PL.CIEP.AQ,NET_PL_MP_AQ,STA_PL0126A,PL0126A,SPO_PL0126A_5_001,SPP_PL0126A_5_001,
4 PL,PL.CIEP.AQ,NET_PL_MP_AQ,STA_PL0126A,PL0126A,SPO_PL0126A_5_001,SPP_PL0126A_5_001,
5 PL,PL.CIEP.AQ,NET_PL_MP_AQ,STA_PL0126A,PL0126A,SPO_PL0126A_5_001,SPP_PL0126A_5_001,
6 PL,PL.CIEP.AQ,NET_PL_MP_AQ,STA_PL0126A,PL0126A,SPO_PL0126A_5_001,SPP_PL0126A_5_001,

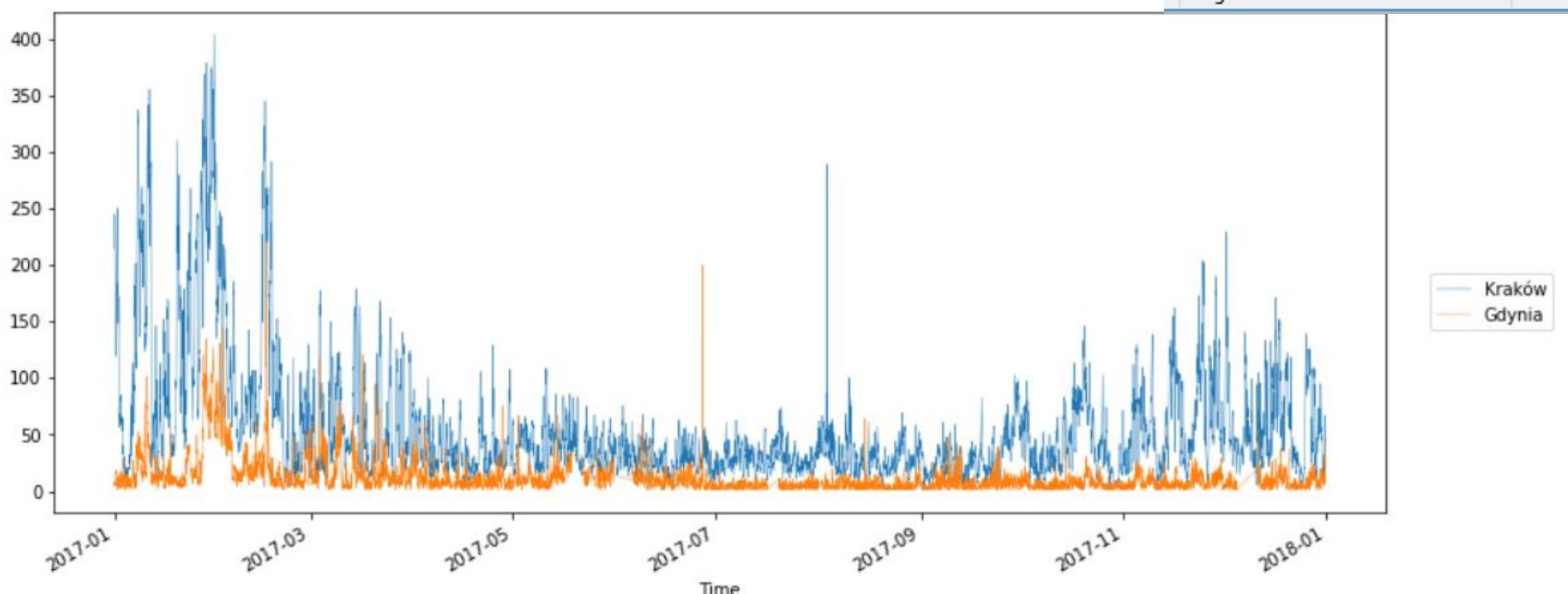
ime,Concentration,UnitOfMeasurement,DatetimeBegin,DatetimeEnd,Validity,Verification
our,20.29449999,µg/m³,2016-01-10 18:00:00 +01:00,2016-01-10 19:00:00 +01:00,1,1
our,6.18107,µg/m³,2016-01-11 15:00:00 +01:00,2016-01-11 16:00:00 +01:00,1,1
our,68.87179999,µg/m³,2016-01-10 21:00:00 +01:00,2016-01-10 22:00:00 +01:00,1,1
our,68.55689999,µg/m³,2016-01-09 06:00:00 +01:00,2016-01-09 07:00:00 +01:00,1,1
our,23.28399999,µg/m³,2016-01-08 22:00:00 +01:00,2016-01-08 23:00:00 +01:00,1,1

Pytania np..

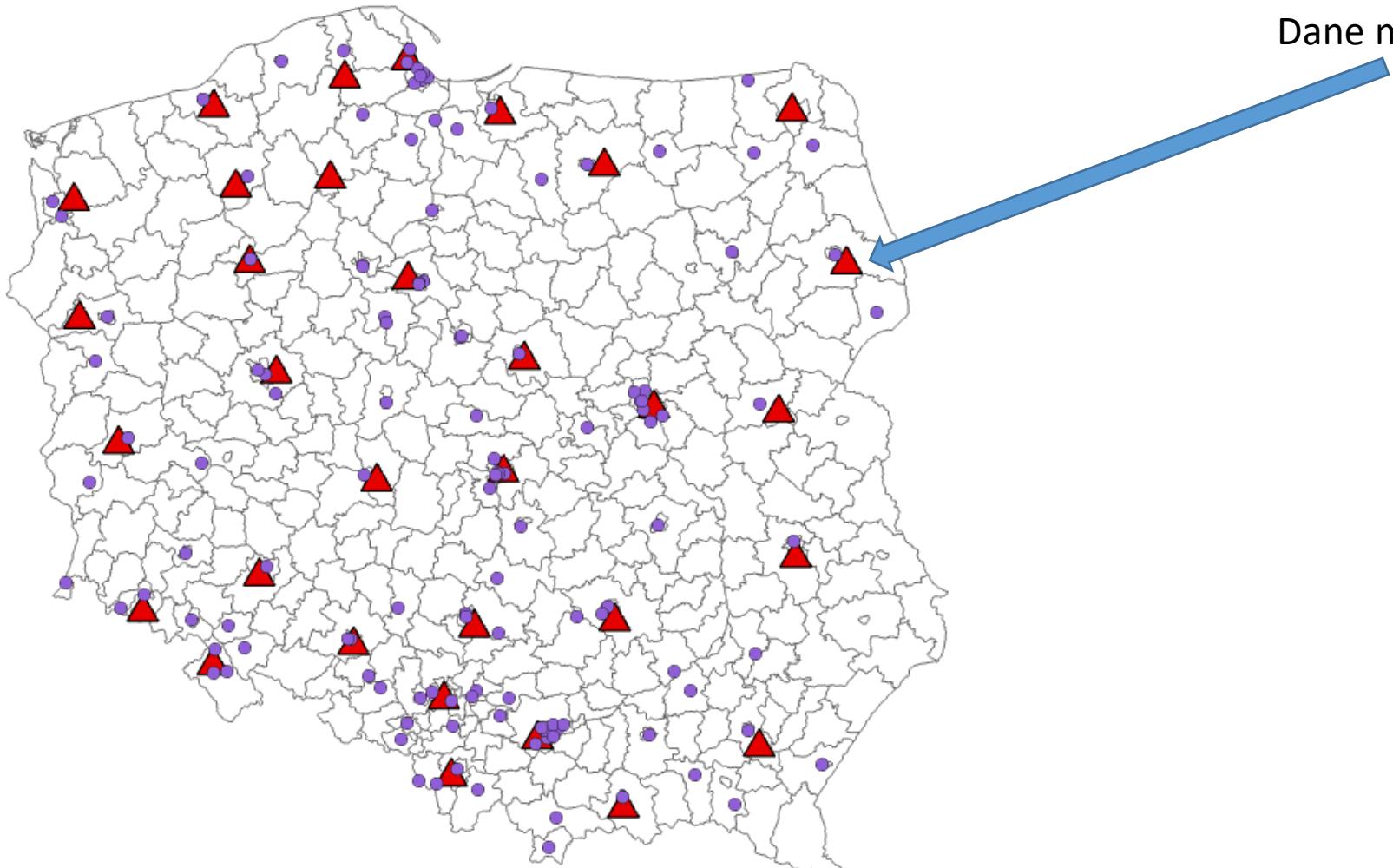
Jak zanieczyszczenie się różni w czasie i przestrzeni ?

Od czego zależy ?

Jak można je modelować?



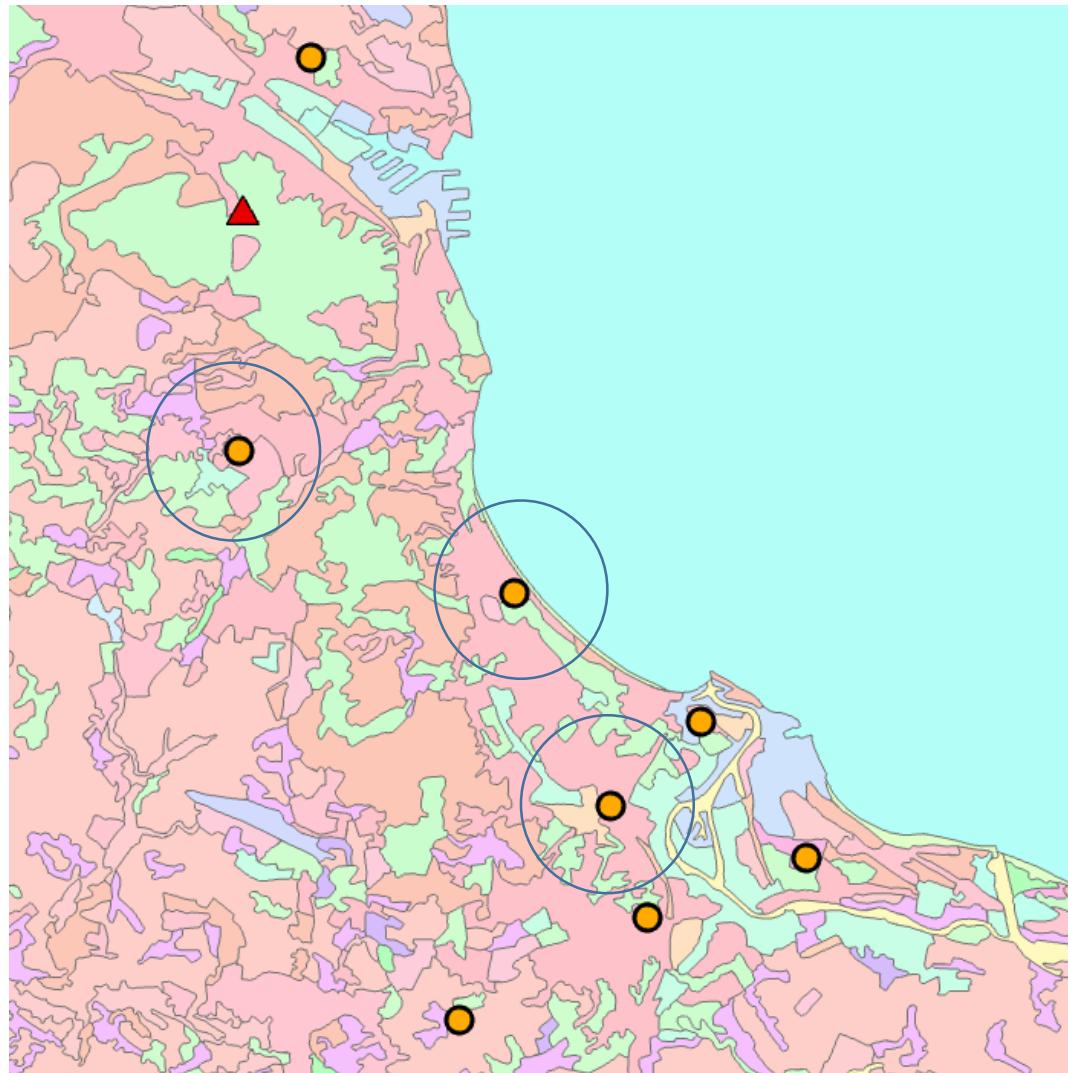
Dane meteo do stacji PM10



Przestrzenna inżynieria
danych (dodawanie
atrybutów)

```
1 "Local time in Gdynia-Oksywie (airbase)";"T";"Po";"P";"Pa";"U";"DD";"Ff";"ff10";"ff3";"N";"WW";"W1";"W2"  
2 "27.05.2018 23:00";"14.9";"765.1";"769.3";"";"93";"Wind blowing from the north-west";"1";"";"";"no clouds";  
3 "27.05.2018 22:00";"15.7";"765.0";"769.1";"";"89";"variable wind direction";"1";"";"";"no clouds";" "  
4 "27.05.2018 21:00";"15.3";"764.8";"769.0";"";"91";"Wind blowing from the north-northwest";"1";"";"";"40%";  
5 "27.05.2018 20:00";"16.7";"764.5";"768.5";"";"83";"Wind blowing from the north";"3";"";"";"no clouds";  
6 "27.05.2018 19:00";"17.8";"764.3";"768.4";"";"78";"Wind blowing from the north";"3";"";"";"40%";" "  
7 "27.05.2018 18:00";"18.1";"764.5";"768.5";"";"79";"Wind blowing from the north-northeast";"3";"";"";"5"  
8 "27.05.2018 16:00";"18.5";"765.0";"769.0";"";"78";"Wind blowing from the east-northeast";"5";"";"";"10"  
9 "27.05.2018 15:00";"18.7";"765.2";"769.3";"";"79";"Wind blowing from the north-east";"4";"";"";"40%";"
```

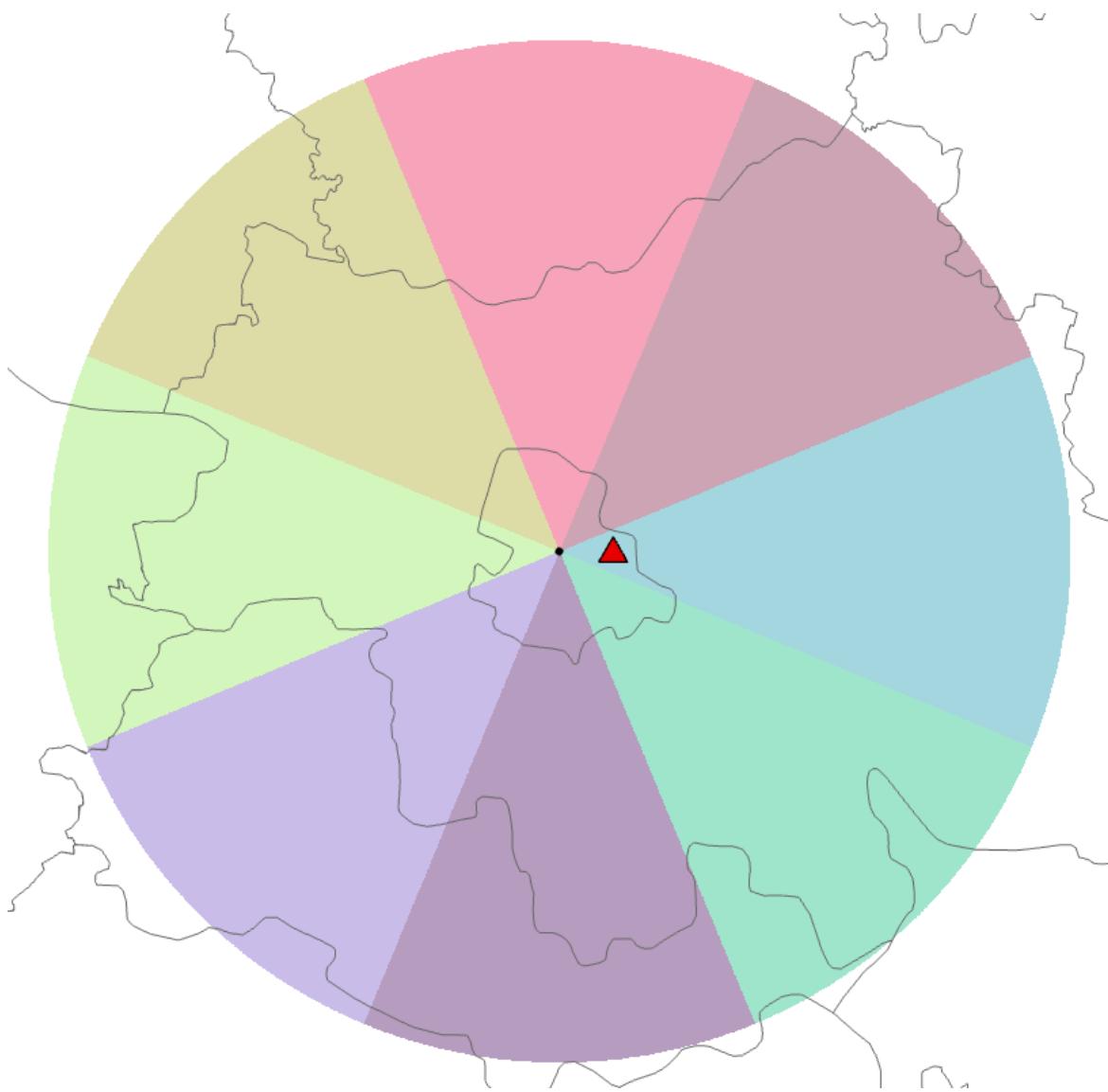
Uwzględnienie pokrycia terenu w buforze



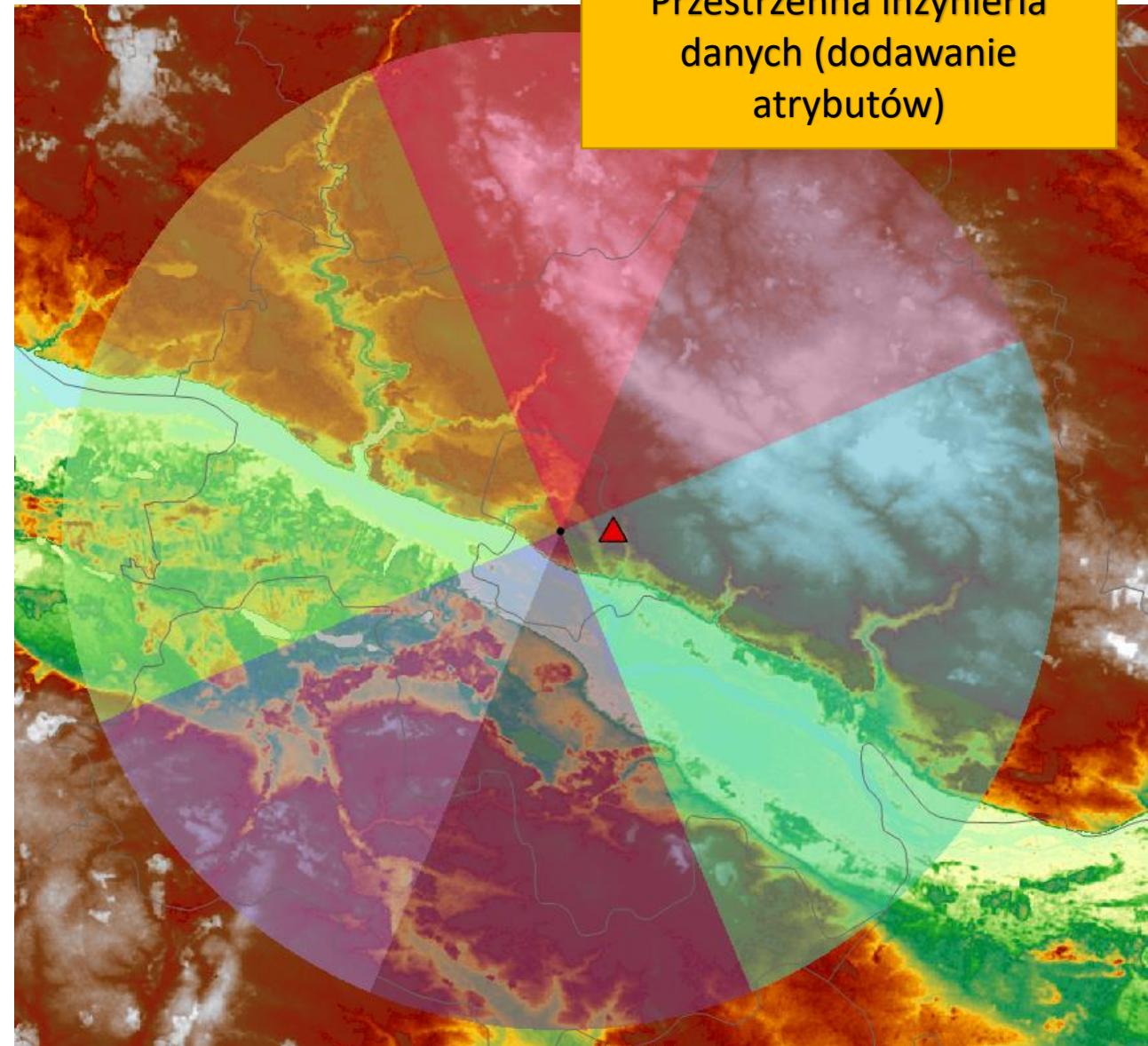
Przestrzenna inżynieria
danych (dodawanie
atrybutów)

UR200	WA200	GR200	IN200	RO200	UR600	WA600	GR600	IN600	RO600
62.93	0.0	2.66	20.68	6.74	52.55	0.0	12.26	16.26	13.1
62.93	0.0	2.66	20.68	6.74	52.55	0.0	12.26	16.26	13.1
62.93	0.0	2.66	20.68	6.74	52.55	0.0	12.26	16.26	13.1
62.93	0.0	2.66	20.68	6.74	52.55	0.0	12.26	16.26	13.1
62.93	0.0	2.66	20.68	6.74	52.55	0.0	12.26	16.26	13.1

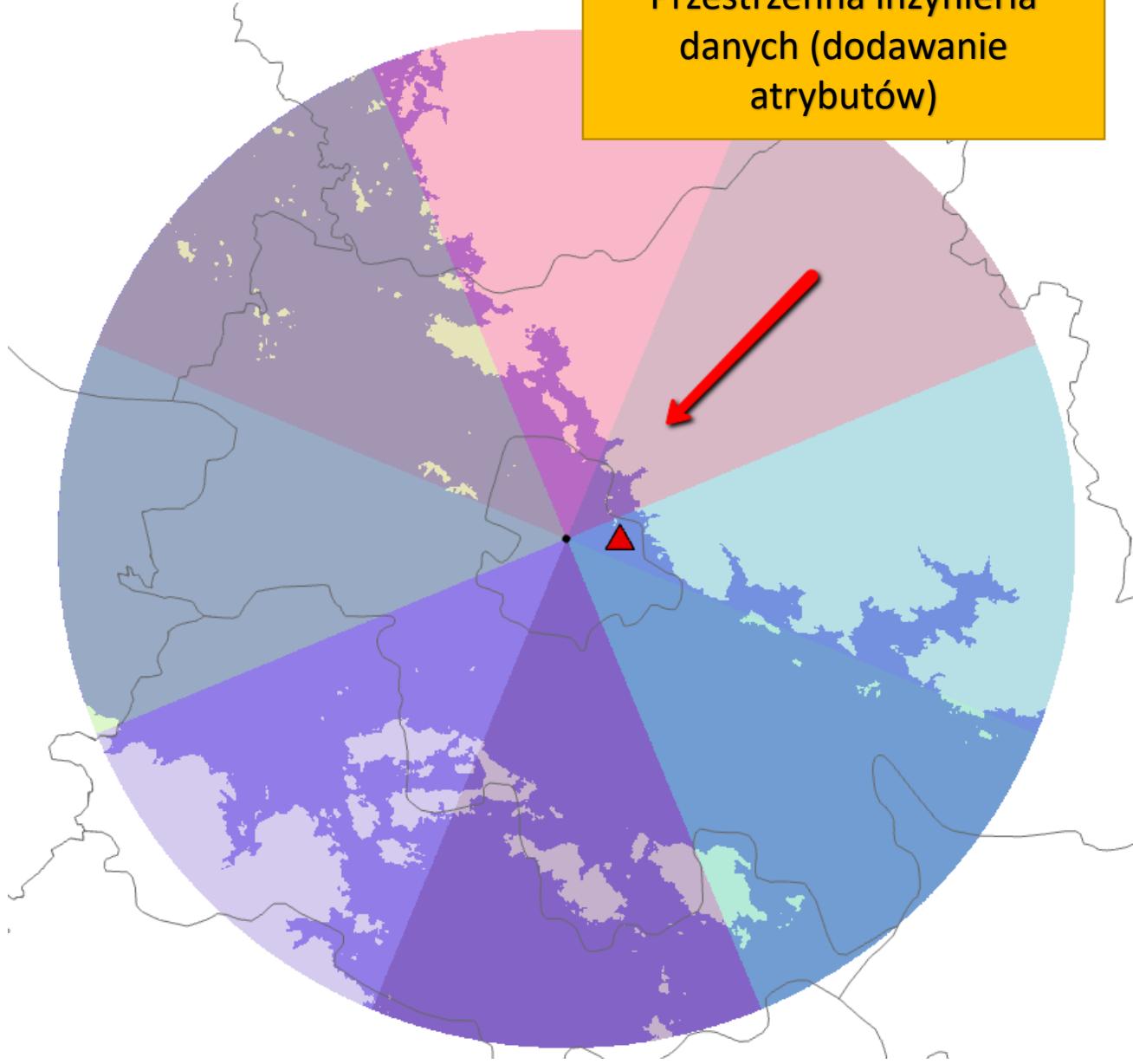
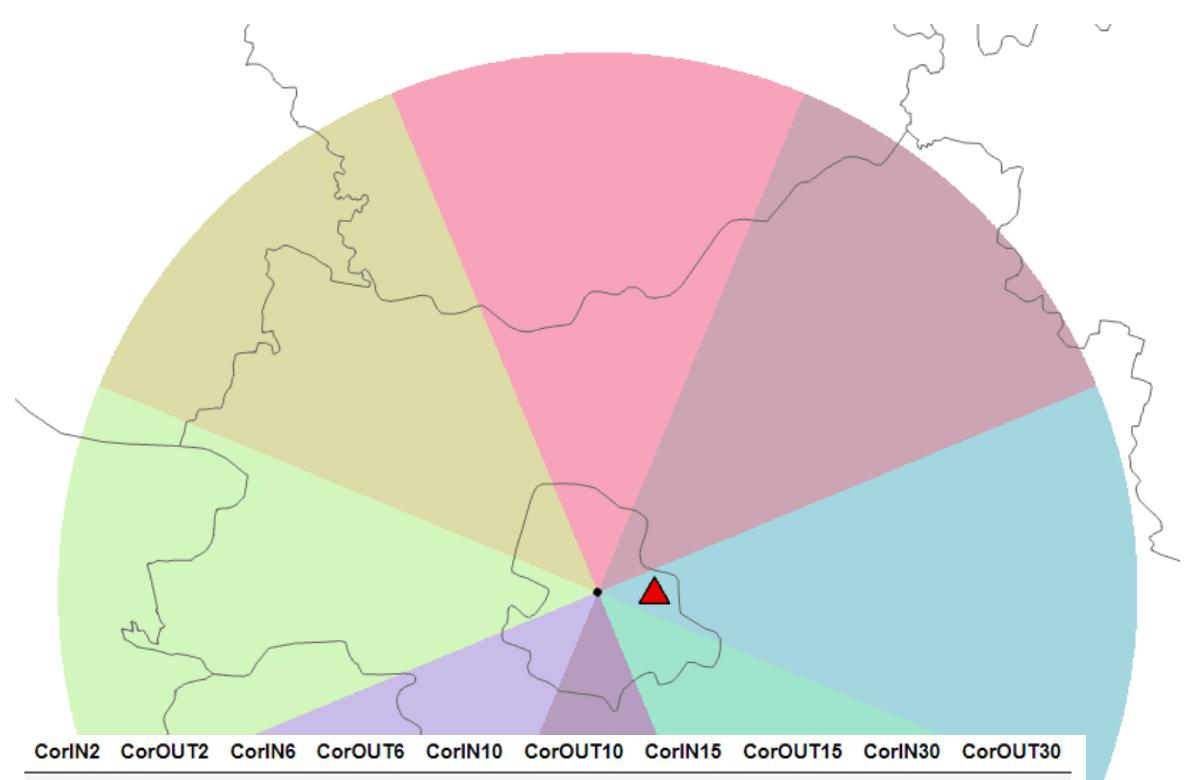
DEM korytarze wiatrowe



Przestrzenna inżynieria
danych (dodawanie
atrybutów)



Przestrzenna inżynieria
danych (dodawanie
atrybutów)



GPS – liczba aut (Open Street map)

Przestrzenna inżynieria
danych (dodawanie
atrybutów)



AHA200	AHA600	AHA2000
3.94	60.51	50.73
3.94	60.51	50.73
3.94	60.51	50.73
3.94	60.51	50.73
3.94	60.51	50.73

Field:	Add	Delete	Calculate	Selection:
OBJECTID	Shape	Field1	Field2	Field3
1	Point	1	53.453716	14.579609
2	Point	2	53.45385	14.579581
3	Point	3	53.453985	14.579542
4	Point	4	53.454113	14.579467
5	Point	5	53.454246	14.579406
6	Point	6	53.454389	14.5794
7	Point	7	53.45453	14.579307
8	Point	8	53.454622	14.57924
9	Point	9	53.454706	14.579288

Przebieg analizy w Jupiter Notebook (geo-python)



Analiza i modelowanie smogu PM10 w Polsce

Jupiter Notebook (inżynieria danych, EDA, wizualizacja i analiza)



Import potrzebnych bibliotek

```
In [1]: 1 import numpy as np  
2 import pandas as pd  
3 import geopandas as gpd  
4 import matplotlib.pyplot as plt  
5 import seaborn as sns  
6 sns.set_style("whitegrid")
```

```

1 Code,Year,Month,Day,Concentration,Concentrat_min,Concentrat_max,T,Trange,H%,Wind_vel,UR200,WA
2 PL0496A,2015,12,31,38.2,38.2,38.2,-13.6,0.0,84.0,1.0,62.93,0.0,2.66,20.68,6.74,52.55,0.0,12.2
3 PL0496A,2016,12,31,46.44,39.88,61.6,-1.1,2.7,95.6,3.08,62.93,0.0,2.66,20.68,6.74,52.55,0.0,12
4 PL0496A,2016,12,1,27.14,9.6,81.1,0.46,2.8,91.4,2.38,62.93,0.0,2.66,20.68,6.74,52.55,0.0,12.26
5 PL0496A,2016,12,2,14.78,5.6,32.11,-0.76,3.9,91.0,3.33,62.93,0.0,2.66,20.68,6.74,52.55,0.0,12.....

```

Wprowadzenie danych

length : 21 248 064 lines : 96 681

In [2]:

```

1 smog=pd.read_csv('PM10_day2.csv')
2 smog.head()

```

Out[2]:

	Code	Year	Month	Day	Concentration	Concentrat_min	Concentrat_max	T
0	PL0496A	2015	12	31	38.20	38.20	38.20	-13.60
1	PL0496A	2016	12	31	46.44	39.88	61.60	-1.10
2	PL0496A	2016	12	1	27.14	9.60	81.10	0.46
3	PL0496A	2016	12	2	14.78	5.60	32.11	-0.76
4	PL0496A	2016	12	3	8.27	1.23	15.66	-0.78

5 rows × 49 columns

	CorIN10	CorOUT10	CorIN15	CorOUT15	CorIN30	CorOUT30	AHA200	AHA600	AHA2000
	0.26	0.92	0.12	0.83	0.32	0.84	3.94	60.51	50.73
	0.89	0.31	0.93	0.20	0.96	0.25	3.94	60.51	50.73
	0.74	0.46	0.75	0.34	0.82	0.37	3.94	60.51	50.73
	0.74	0.46	0.66	0.41	0.64	0.55	3.94	60.51	50.73
	0.91	0.28	0.81	0.13	0.83	0.33	3.94	60.51	50.73

Atrybut	Opis
Code	Kod stacji pomiarowej
Year	Rok
Month	Miesiąc
Day	Dzień
Concentration	Średnia dobowa koncentracja PM10
Concentrat_min	Minimalna dobowa koncentracja PM10
Concentrat_max	Maksymalna dobowa koncentracja PM10
T	Średnia dobowa temperatura powietrza
Trange	Zakres temperatury w czasie doby
H%	Średnia wilgotność względna powietrza w %
Wind_vel	Średnia dobowa prędkość wiatru w ms^{-1}
UR200	Część (0-1) obszaru o promieniu 200m od punktu pomiarowego pokryta przez zabudowę miejską
WA200	Część (0-1) obszaru o promieniu 200m od punktu pomiarowego pokryta przez wodę
GR200	Część (0-1) obszaru o promieniu 200m od punktu pomiarowego pokryta przez zieleń
IN200	Część (0-1) obszaru o promieniu 200m od punktu pomiarowego pokryta przez infrastrukturę przemysłową
RO200	Część (0-1) obszaru o promieniu 200m od punktu pomiarowego pokryta przez drogi
CorIN2	Część (0-1) dostępnego korytarza powietrznego w sektorze 30 stopni zgodnego z kierunkiem wiatru na obszarze o promieniu 2 km
CorOUT2	Część (0-1) dostępnego korytarza powietrznego w sektorze 30 stopni przeciwnego do kierunku wiatru na obszarze o promieniu 2 km
AHA200	Liczba aut na hektar w promieniu 200m od punktu pomiarowego (całkowita liczba rejestracji GPS z Open Street data)

Pokrycia dla 200, 600, 1000, 2000, 4000 m

Korytarze dla 2, 6, 10, 15, 30 km

Auta dla 200, 600, 2000 m

Podstawowe pytania

Podstawowe standardy czystości powietrza w EU wynoszą:

50 µg/m³ 24 hours(mean) nie więcej niż 35 razy do roku (WHO 3 dni)

40 µg/m³ 1 year (mean) (WHO bez szkody dla zdrowia 20 µg/m³)

Dla 2016, 2017 wyznaczyć dla poszczególnych stacji średnie roczne PM10

Dla 2016, 2017 wyznaczyć dla poszczególnych stacji liczbę dni z przekroczeniem dobowego limitu PM10.

1 Jakie stacje mamy do dyspozycji?

In [3]:

```
1 stacje_all=list(smog['Code'].unique())
2 print(stacje_all)
```

Lista z nazwami stacji

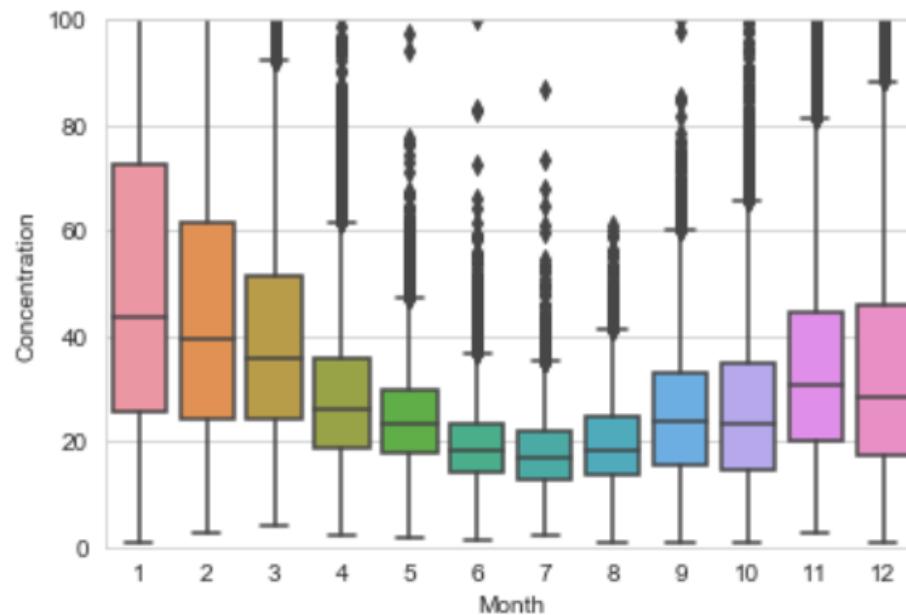
```
['PL0496A', 'PL0151A', 'PL0651A', 'PL0234A', 'PL0236A', 'PL0294A', 'PL0568A',
'PL0241A', 'PL0243A', 'PL0526A', 'PL0184A', 'PL0295A', 'PL0311A', 'PL0559A', 'P
L0051A', 'PL0052A', 'PL0049A', 'PL0047A', 'PL0520A', 'PL0045A', 'PL0048A', 'PL0
050A', 'PL0046A', 'PL0209A', 'PL0575A', 'PL0585A', 'PL0634A', 'PL0503A', 'PL003
1A', 'PL0008A', 'PL0222A', 'PL0306A', 'PL0237A', 'PL0218A', 'PL0242A', 'PL0529
A', 'PL0239A', 'PL0552A', 'PL0238A', 'PL0240A', 'PL0283A', 'PL0596A', 'PL0563
A', 'PL0187A', 'PL0192A', 'PL0321A', 'PL0191A', 'PL0504A', 'PL0633A', 'PL0298
A', 'PL0039A', 'PL0643A', 'PL0012A', 'PL0501A', 'PL0641A', 'PL0273A', 'PL0642
```

Charakterystyczną cechą jest zmienność sezonowa.

In [4]:

```
1 # Analiza po miesiącach
2 plt.ylim((0,100))
3 sns.set(rc={'figure.figsize':(6,4)})
4 sns.boxplot(data=smog, x='Month', y='Concentration');
```

Eksploracyjna analiza danych



- | | |
|---|------------------------------|
| 1 | Zmiennaść pomiędzy stacjami. |
| 2 | Zmiennaść średniej rocznej. |

In [6]:

```

1 # Średnie roczne zanieczyszczenie dla stacji
2 #sr_roczna2016=np.zeros((len(stacje_all),2))
3 sr_roczna2017=np.zeros((len(stacje_all),2))
4 rok=int(2017)
5 for i in range(0,len(stacje_all)):
6     smog1=smog[(smog.Code==stacje_all[i]) & (smog.Year==rok)]
7     if len(smog1)>200: # co najmniej 200 dni
8         pp=smog['Concentration'][((smog.Code==stacje_all[i]) & (smog.Year==rok)].mean()
9         pps=smog['Concentration'][((smog.Code==stacje_all[i]) & (smog.Year==rok)].std()
10        if rok==2016:
11            sr_roczna2016[i,0]=round(pp,2)
12            sr_roczna2016[i,1]=round(pps,2)
13        elif rok==2017:
14            sr_roczna2017[i,0]=round(pp,2)
15            sr_roczna2017[i,1]=round(pps,2)

```

Utworzenie tablic ze średnim zanieczyszczeniem dla każdej stacji

In [7]:

```

1 out_plik = open(r"C:\POLUTION\PROJ\s्र_roczne2.txt","w")
2
3 sss='STACJA,SR2016,SR2017,OD2016,OD2017'+"\n"
4 out_plik.write(sss)
5 for i in range (len(stacje_all)):
6     #print(stacje_all[i],sr_roczna2016[i],sr_roczna2017[i])
7     sss=stacje_all[i]+','+str(sr_roczna2016[i,0])+','+str(sr_roczna2016[i,1])
8     +str(sr_roczna2017[i,0])+','+str(sr_roczna2017[i,1])+'\n'
9     out_plik.write(sss)
10
11 out_plik.close()

```

Zapisanie tablicy do pliku tekstowego

1	STACJA, SR2016, SR2017, OD2016, OD2017
2	PL0496A, 24.05, 23.26, 13.59, 15.6
3	PL0151A, 23.62, 24.92, 14.51, 16.77
4	PL0651A, 27.52, 0.0, 16.51, 0.0
5	PL0234A, 35.72, 37.31, 30.37, 40.19
6	PL0236A, 29.88, 31.9, 22.21, 31.06

```
In [8]: 1 pm10sr=pd.read_csv('sr_roczne2.txt')
2 pm10sr.loc[pm10sr['SR2017']==0,'SR2017']=np.nan
3 pm10sr.loc[pm10sr['SR2016']==0,'SR2016']=np.nan
4 pm10sr.loc[pm10sr['OD2017']==0,'OD2017']=np.nan
5 pm10sr.loc[pm10sr['OD2016']==0,'OD2016']=np.nan
```

```
In [9]: 1 powiatypl=gpd.read_file('geo/pl_powiaty.shp')
2 pm10=gpd.read_file('geo/stacje_PM10.shp')
```

Utworzenie geo data frame dla powiatów i stacji

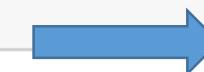
```
In [10]: 1 pm10.head(2)
```

Out[10]:

	AirQuality	AirQuali_1	AirQuali_2	AirPolluta	Altitude	count	mean	STACJA_MET	IDPP	geometry	SR2016	SR2017	OD2016	OD2017
0	SIKatoKossut	PL0008A	urban	PM10	273.0	20654.0	41.369689	Katowice	1	POINT (498220.7060315014 266381.2517729597)	38.64	41.01	25.47	42.92

1	MpKrakAlKras	PL0012A	urban	PM10	207.0	20567.0	57.291317	Kraków	2	POINT (566277.409905806 243790.1838724455)	56.58	54.67	34.44	48.44
---	--------------	---------	-------	------	-------	---------	-----------	--------	---	---	-------	-------	-------	-------

```
In [11]: 1 pm10B=pm10.set_index('AirQuali_1').join(pm10sr.set_index('STACJA'))
2 pm10B.head()
```

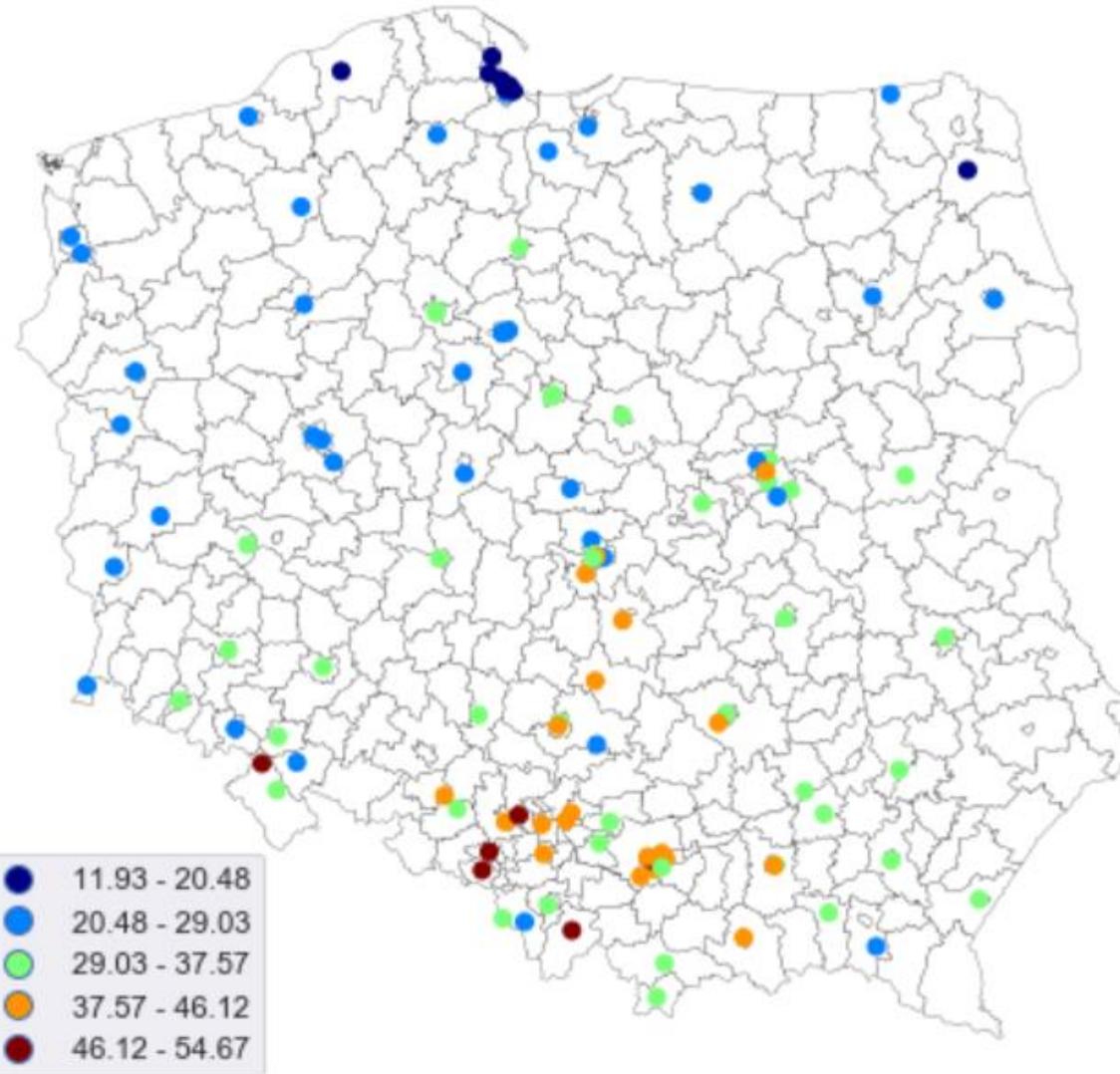


Połączenie data frames

26.73	28.06	15.45	24.85
41.17	42.16	29.21	38.85
NaN	23.97	NaN	15.86

In [13]:

```
1 plt.rcParams['figure.figsize'] = [8,8]
2
3 ax = powiatypl.plot(color='white', edgecolor='black', linewidth=0.2);
4 pm10B[pm10B.SR2017>0].plot(column='SR2017',cmap='jet', scheme='equal_interval',
5                               ax=ax,legend='True');
6 ax.axis('off');
```



Średnie roczne stężenie PM10 w roku 2017,
Powinno być poniżej 40 (WHO 20)

Liczba dni w roku z PM10 > 50 µg/m3

In [14]:

```
1 out_plik = open(r"C:\POLUTION\PROJ\dobowe_dni50a.txt","w")
2
3 sss='STACJA,DNI17_50'+'\n'
4 out_plik.write(sss)
5
6
7 prog=50
8 rok=2017
9 for i in range(0,len(stacje_all)):
10     smog1=smog[(smog.Code==stacje_all[i]) & (smog.Year==rok)]
11     #print(stacje_all[i],len(smog1['Concentration'][smog1['Concentration']>50]),len(smog1))
12     if len(smog1)>=200:
13         ldni=len(smog1['Concentration'][smog1['Concentration']>50])
14         sss=stacje_all[i]+','+str(ldni)+'\n'
15         out_plik.write(sss)
16
17 out_plik.close()
```

Tworzy plik tekstowy z liczbą dni powyżej 50 dla każdej stacji

1	STACJA, DNI17_50
2	PL0496A, 15
3	PL0151A, 17
4	PL0234A, 73
5	PL0236A, 54
6	PL0294A, 90
7	PL0568A, 33

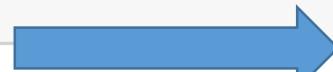
In [15]:

```
1 pm10d50=pd.read_csv('dobowe_dni50a.txt')
```

In [16]:

```
1 pm10[ 'STACJA']=pm10.AirQuali_1
2
3 pm10C=pd.merge(pm10,pm10d50,how='outer',on='STACJA')
4
5 pm10C.head(3)
```

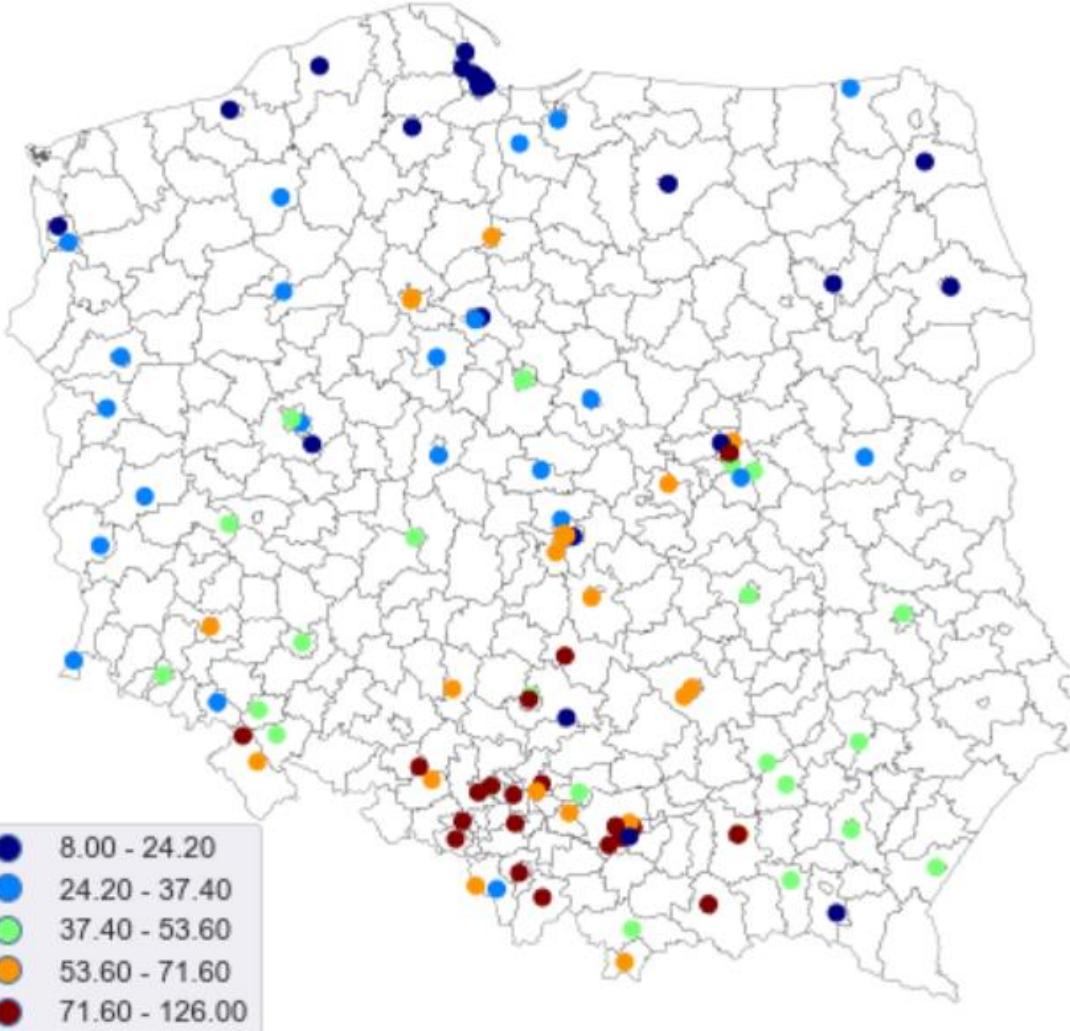
Połączenie data frames



	geometry	STACJA	DNI17_50
1	POINT (498220.7060315014 266381.2517729597)	PL0008A	78.0
2	POINT (566277.409905806 243790.1838724455)	PL0012A	126.0
3	POINT (450086.9278154476 484649.3748249114)	PL0031A	30.0

In [17]:

```
1 plt.rcParams['figure.figsize'] = [8,8]
2
3 ax = powiatypl.plot(color='white', edgecolor='black', linewidth=0.2);
4 pm10C[pm10C.DNI17_50>0].plot(column='DNI17_50',cmap='jet', scheme='quantiles',
5                                 ax=ax,legend='True');
6 ax.axis('off');
```



Liczba dni w 2017 roku ze stężeniem PM10 powyżej 50.
Powinno być poniżej 35 dni (WHO 3)

Modelowanie

```
In [25]: 1 smoga=smog.dropna()
```

```
In [26]: 1 collist=['Concentration', 'T', 'H%', 'Wind_vel',
 2           'UR200', 'WA200', 'GR200', 'IN200', 'RO200', 'UR600', 'WA600', 'GR600',
 3           'IN600', 'RO600', 'UR1000', 'WA1000', 'GR1000', 'IN1000', 'RO1000',
 4           'UR2000', 'WA2000', 'GR2000', 'IN2000', 'RO2000', 'UR4000', 'WA4000',
 5           'GR4000', 'IN4000', 'RO4000', 'CorIN2', 'CorOUT2', 'CorIN6', 'CorOUT6',
 6           'CorIN10', 'CorOUT10', 'CorIN15', 'CorOUT15', 'CorIN30', 'CorOUT30',
 7           'AHA200', 'AHA600', 'AHA2000']
 8 smog1=smoga[collist]
```

Współczynnik korelacji z koncentracją (wybór istotnych atrybutów)

```
In [27]: 1 correlations_data = smog1.corr()['Concentration'].sort_values()
 2 print(correlations_data, '\n')
```

T	-0.432010		
Wind_vel	-0.298404		
WA4000	-0.090318		
GR200	-0.080640		
GR4000	-0.078984		
GR600	-0.074964	IN4000	0.082824
GR2000	-0.070282	UR2000	0.092711
WA2000	-0.068625	RO200	0.096458
		RO2000	0.098587
		RO4000	0.106903
		H%	0.111193
		UR4000	0.124739
		AHA2000	0.130930

```
In [5]: 1 collist2=['Concentration', 'T', 'H%', 'Wind_vel', 'UR4000', 'WA4000',
2           'GR200', 'IN4000', 'RO4000','CorIN30', 'CorOUT15','AHA2000']
3 smog2=smoga[collist2]
```

```
In [6]: 1 smog2.shape
```

```
Out[6]: (68162, 12)
```

Podział na data frame X i y

```
In [ ]: 1 X=smog2[['T', 'H%', 'Wind_vel', 'RO4000', 'UR4000', 'WA4000',
2           'GR200', 'IN4000', 'CorOUT15', 'CorIN30', 'AHA2000']]
3 y=smog2[['Concentration']]
```

```
In [7]: 1 # rozwiązanie alternatywne
2 X=smog2[['T', 'H%', 'Wind_vel']]
3 y=smog2[['Concentration']]
```

```
In [31]: 1 X
```

```
Out[31]:
```

	T	H%	Wind_vel
0	-13.60	84.0	1.00
1	-1.10	95.6	3.08
2	0.46	91.4	2.38
3	-0.76	91.0	3.33

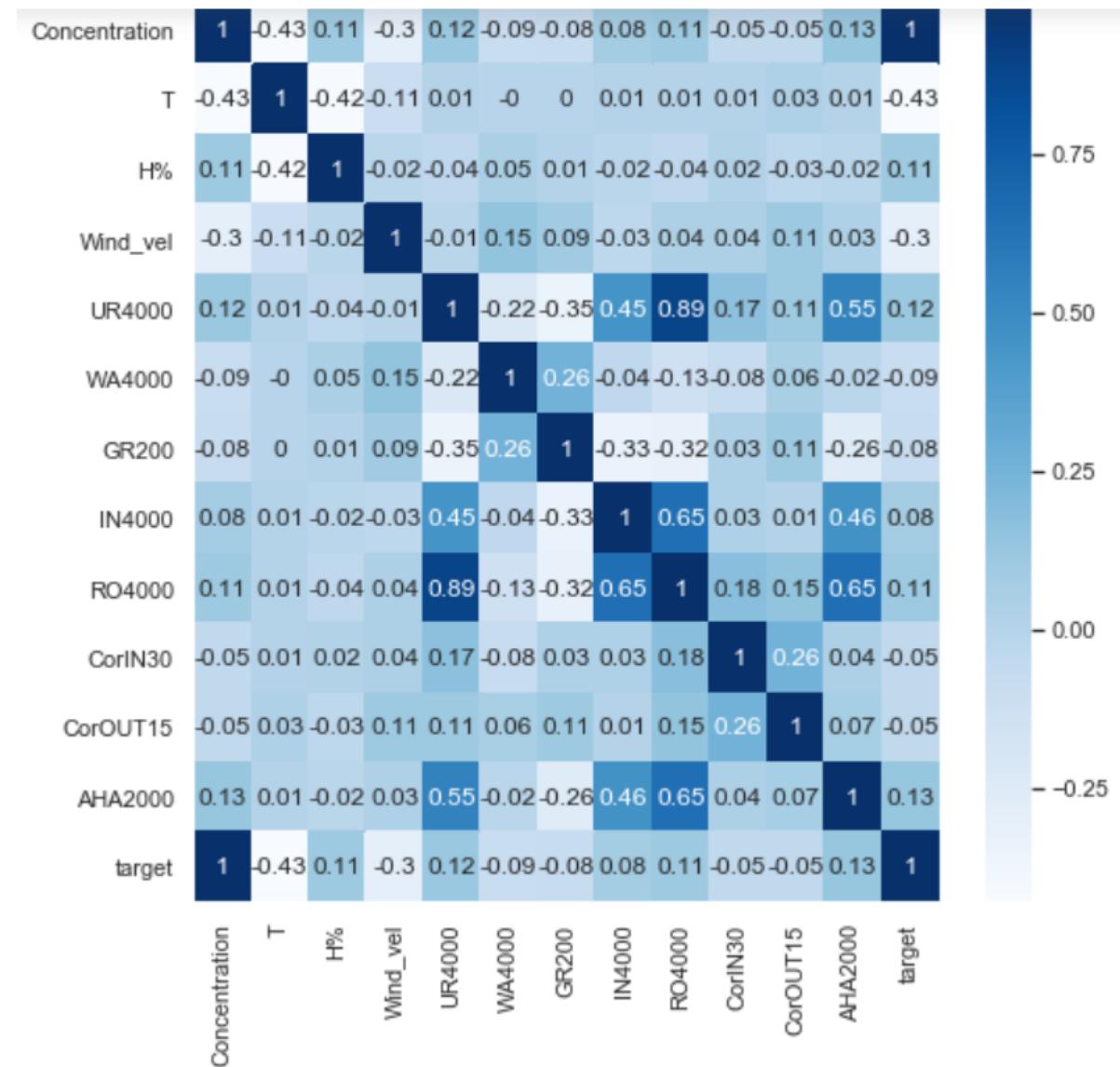
```
1 y.head(4)
```

	Concentration
0	38.20
1	46.44
2	27.14
3	14.78

Macierz korelacji

In [35]:

```
1 plt.subplots(figsize=(8,8))
2 sns.heatmap(smog2.assign(target = y).corr().round(2), cmap = 'Blues',
3             annot = True).set_title('Correlation matrix', fontsize = 16);
```



```
In [9]: 1 from sklearn.model_selection import train_test_split  
2  
3 X_train, X_test, y_train, y_test = train_test_split(X, y,  
4 test_size=0.15, random_state=107)
```

Tworzenie zbiorów treningowych i testowych

```
In [10]: 1 from sklearn.ensemble import RandomForestRegressor,  
2 from sklearn import metrics
```

Pobranie bibliotek

```
In [40]: 1 y_traina=np.array(y_train).reshape(-1)  
2 y_testa=np.array(y_test).reshape(-1)  
3 random_forest = RandomForestRegressor(random_state=70,n_estimators=100,oob_score = True)  
4 random_forest.fit(X_train,y_traina) Proces uczenia
```

Zdefiniowanie modelu regresora

```
out[40]: RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,  
max_features='auto', max_leaf_nodes=None,  
min_impurity_decrease=0.0, min_impurity_split=None,  
min_samples_leaf=1, min_samples_split=2,  
min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None,  
oob_score=True, random_state=70, verbose=0, warm_start=False)
```

```
In [41]: 1 predictions = random_forest.predict(X_test) Proces predykcji
```

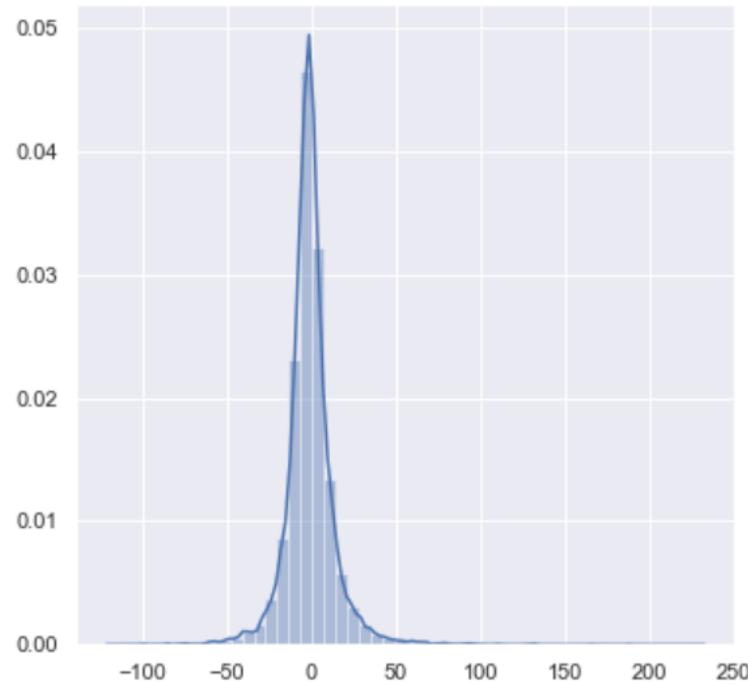
```
array([32.28359167, 26.48136944, 27.42514333, ..., 78.76309357,  
14.33207 , 24.40668 ])
```

```
In [42]: 1 print('MAE:', metrics.mean_absolute_error(y_testa, predictions))  
2 print('MSE:', metrics.mean_squared_error(y_testa, predictions))  
3 print('RMSE:', np.sqrt(metrics.mean_squared_error(y_testa, predictions)))  
4 print('R^2 Training Score: {:.2f} \nOOB Score: {:.2f} \nR^2 Validation Score: {:.2f}'.format(random_forest.score(X_train, y_t  
5  
6  
random_forest.oob_score_,  
random_forest.score(X_test, y_test)))
```

MAE: 9.431809936769259
MSE: 218.9984449584984
RMSE: 14.798596046872095
R^2 Training Score: 0.90
OOB Score: 0.67
R^2 Validation Score: 0.70

In [45]:

```
1 plt.subplots(figsize=(6,6))
2 sns.distplot((y_testa-predictions),bins=50);
```



In [55]:

```

1 # jakie czynniki były ważne
2 base_imp = imp_df(X_train.columns, random_forest.feature_importances_)
3 base_imp

```

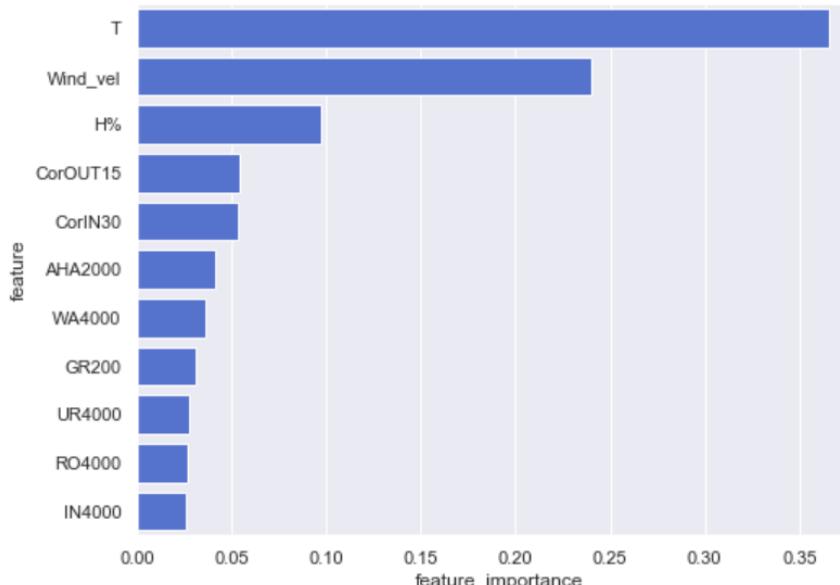
Określenie ważności poszczególnych atrybutów

In [56]:

```

1 plt.subplots(figsize=(8,6))
2 var_imp_plot(base_imp, '');

```



In [57]:

```

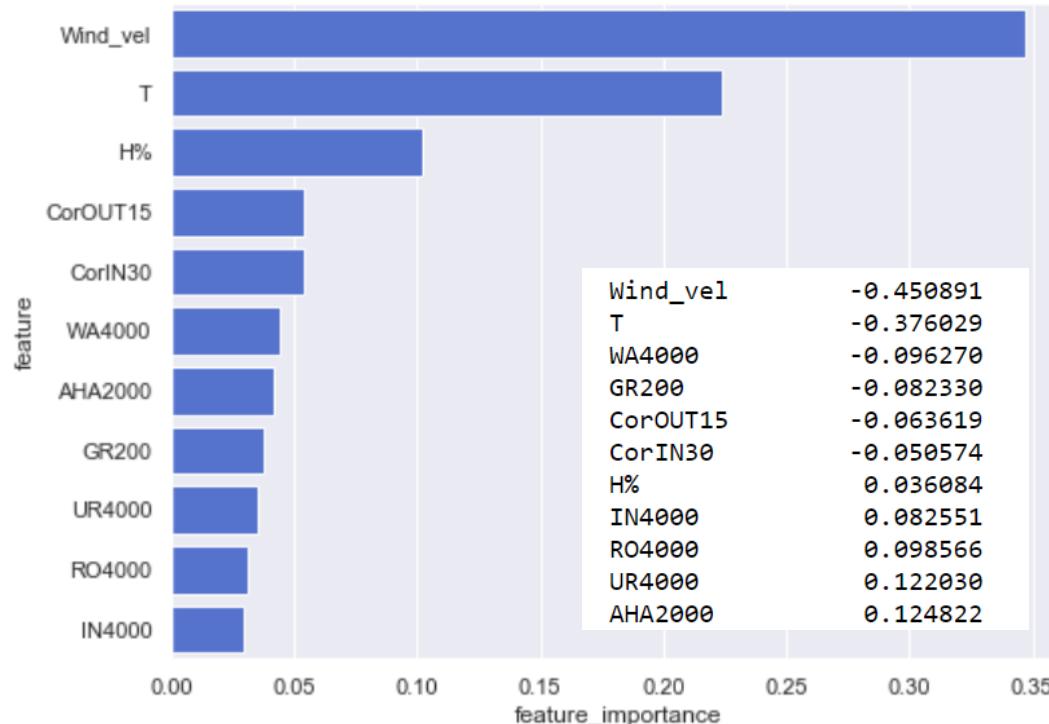
1 # jakie czynniki były ważne
2 import eli5
3 from eli5.sklearn import PermutationImportance
4 perm = PermutationImportance(random_forest, random_state=1).fit(X_test, y_test)
5 eli5.show_weights(perm, feature_names = X_test.columns.tolist())

```

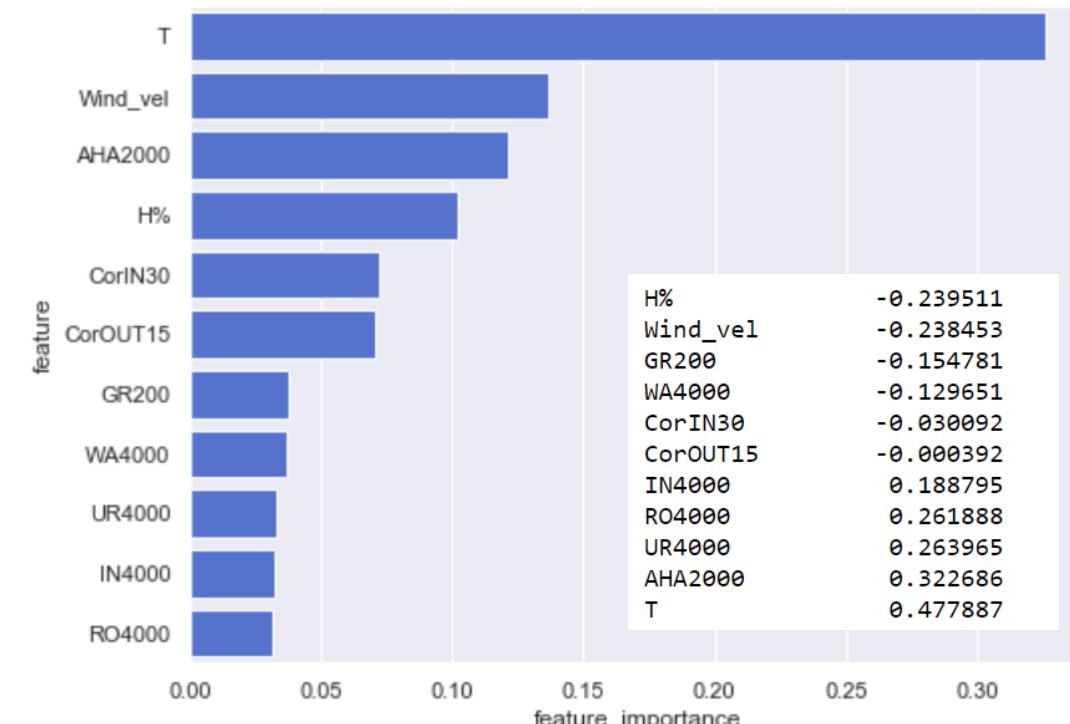
Out[57]:

Weight	Feature
0.9247 ± 0.0357	T
0.5710 ± 0.0261	Wind_vel
0.1653 ± 0.0126	H%
0.0547 ± 0.0017	AHA2000
0.0494 ± 0.0216	WA4000
0.0460 ± 0.0022	UR4000
0.0265 ± 0.0033	RO4000
0.0265 ± 0.0074	GR200
0.0262 ± 0.0058	CorOUT15
0.0261 ± 0.0022	CorIN30
0.0205 ± 0.0034	IN4000

styczeń, luty, marzec



czerwiec, lipiec sierpień



Out[66]:

Weight	Feature
0.7354 ± 0.0269	Wind_vel
0.4148 ± 0.0282	T
0.1197 ± 0.0103	H%
0.0516 ± 0.0039	UR4000
0.0462 ± 0.0075	AHA2000
0.0270 ± 0.0116	WA4000
0.0269 ± 0.0037	CorOUT15
0.0260 ± 0.0041	RO4000
0.0226 ± 0.0200	GR200
0.0212 ± 0.0038	IN4000
0.0195 ± 0.0029	CorIN30

Weight	Feature
0.4847 ± 0.0419	T
0.2059 ± 0.0295	Wind_vel
0.1841 ± 0.0119	AHA2000
0.0662 ± 0.0054	H%
0.0378 ± 0.0073	WA4000
0.0303 ± 0.0062	GR200
0.0283 ± 0.0061	CorOUT15
0.0281 ± 0.0082	CorIN30
0.0275 ± 0.0067	UR4000
0.0271 ± 0.0038	RO4000
0.0126 ± 0.0048	IN4000

```
In [31]: 1 XX=smog[['T', 'H%', 'Wind_vel']]
```

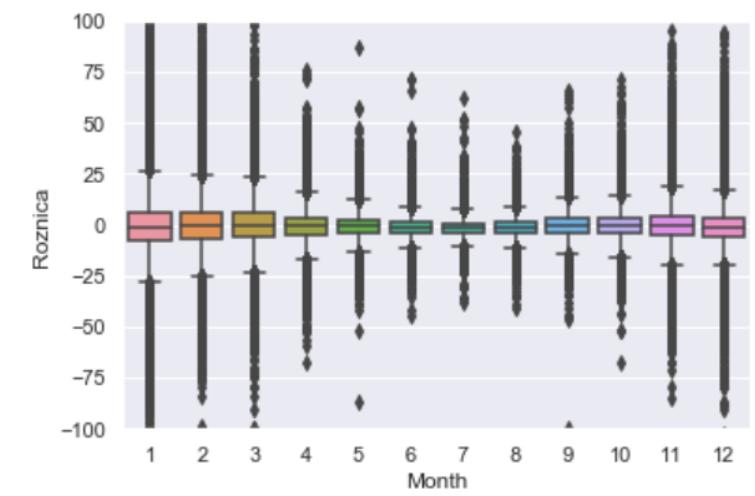
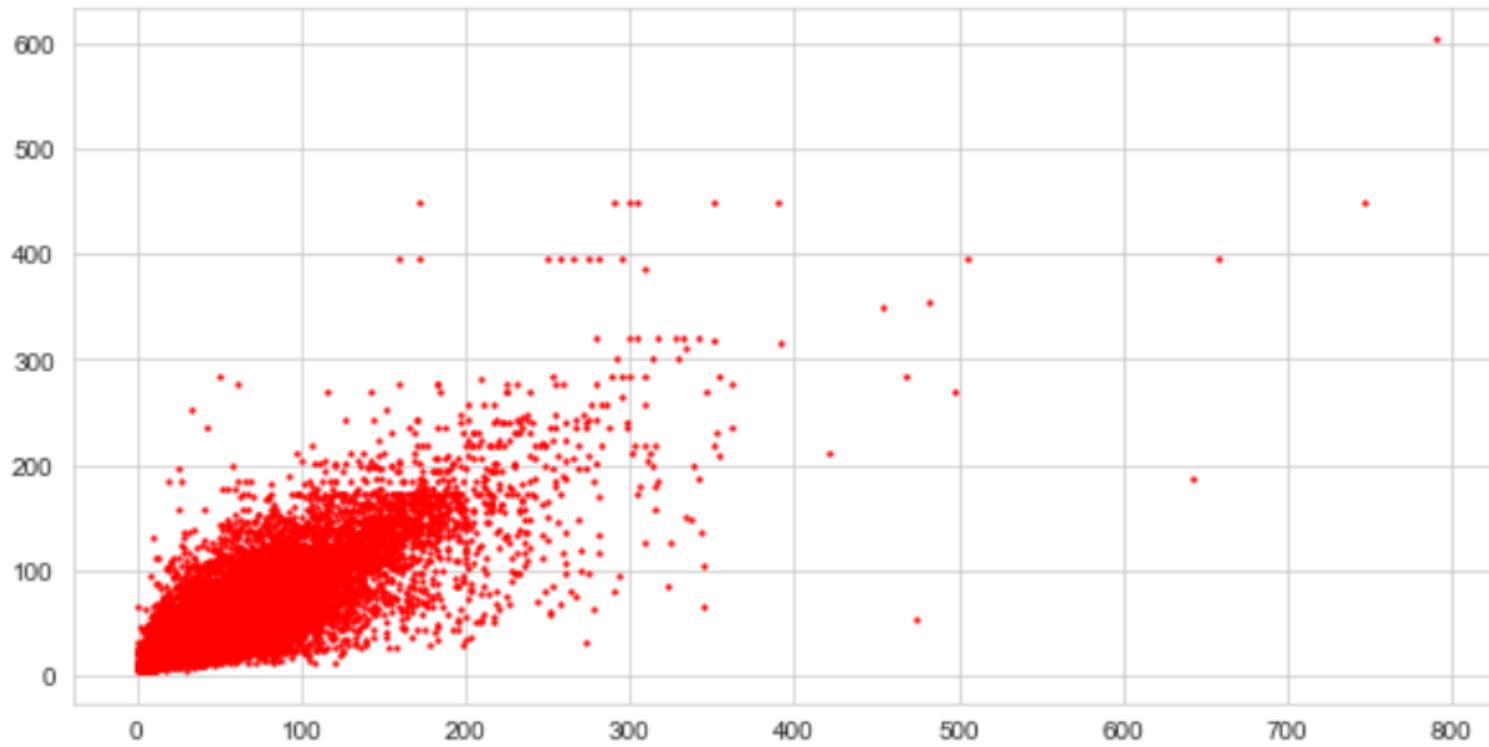
Predykcja dla roku 2017

```
In [32]: 1 predictions = random_forest.predict(XX)
```

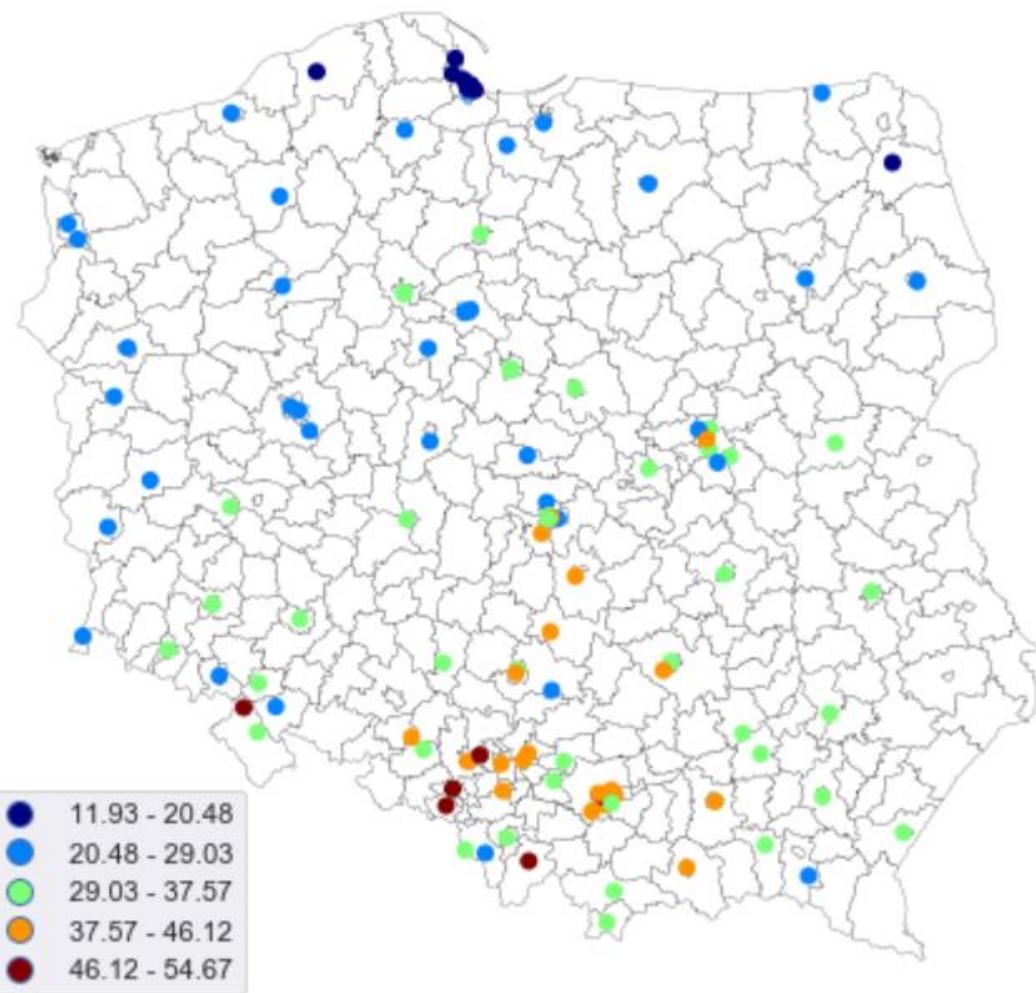
```
In [33]: 1 len(predictions)
```

```
Out[33]: 96680
```

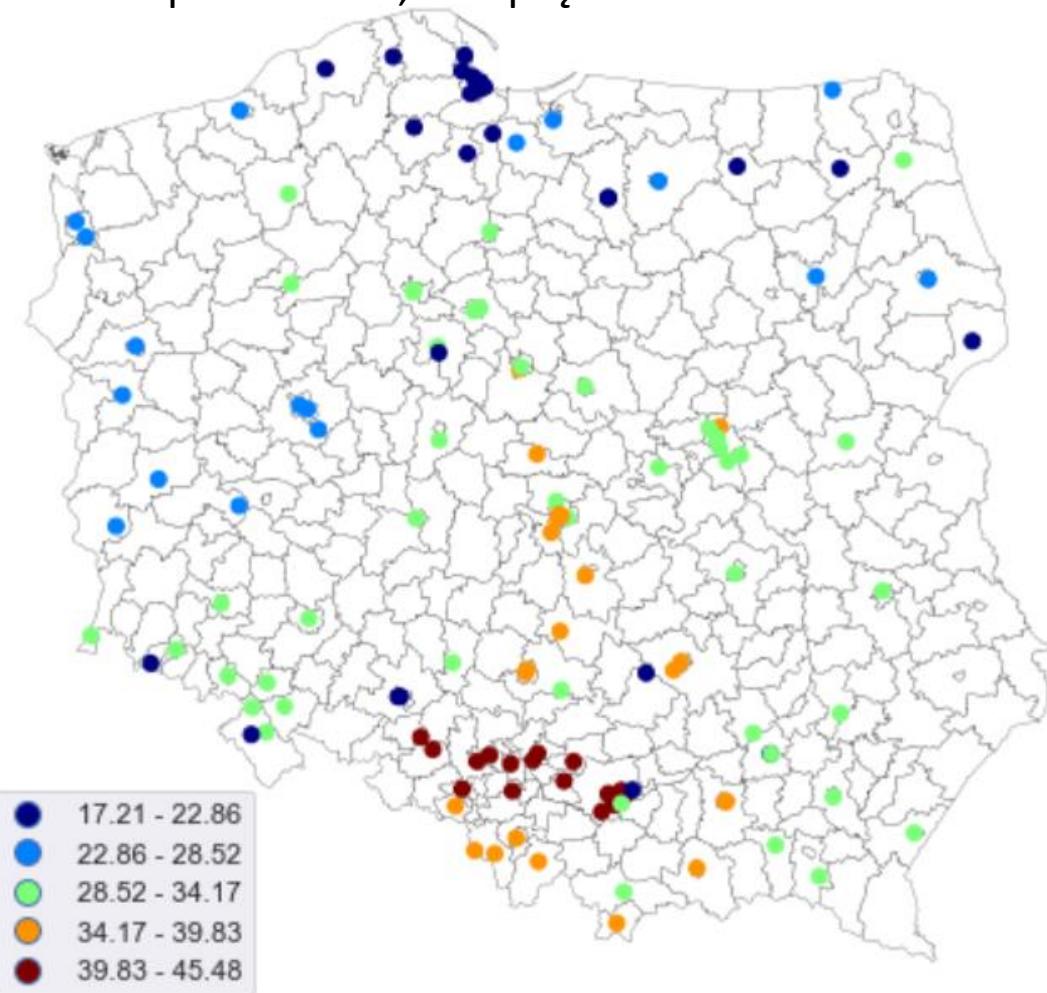
```
In [35]: 1 fig, ax = plt.subplots(figsize=(10, 5))
2 ax.scatter(smog.Concentration,predictions,label='skitscat',color='r',marker='o',s=2)
```



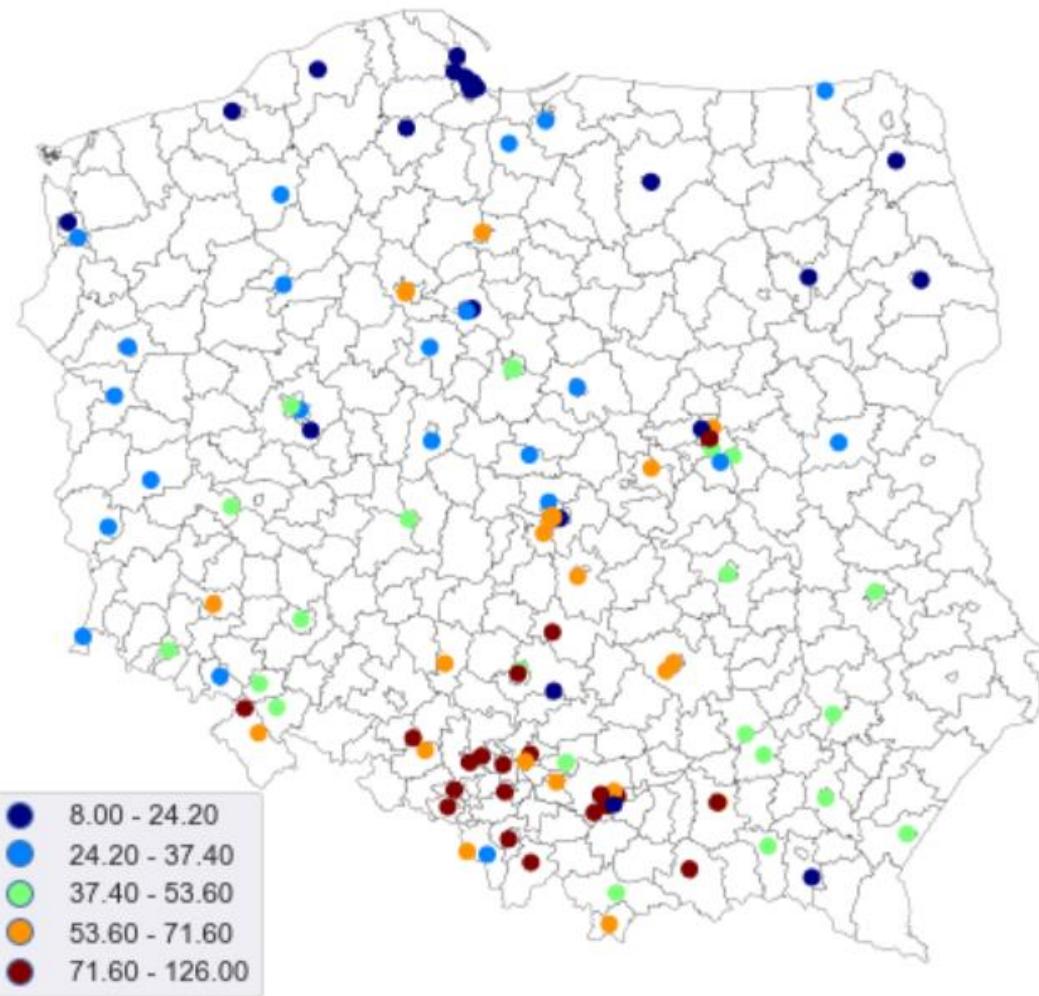
Średnie roczne w 2017



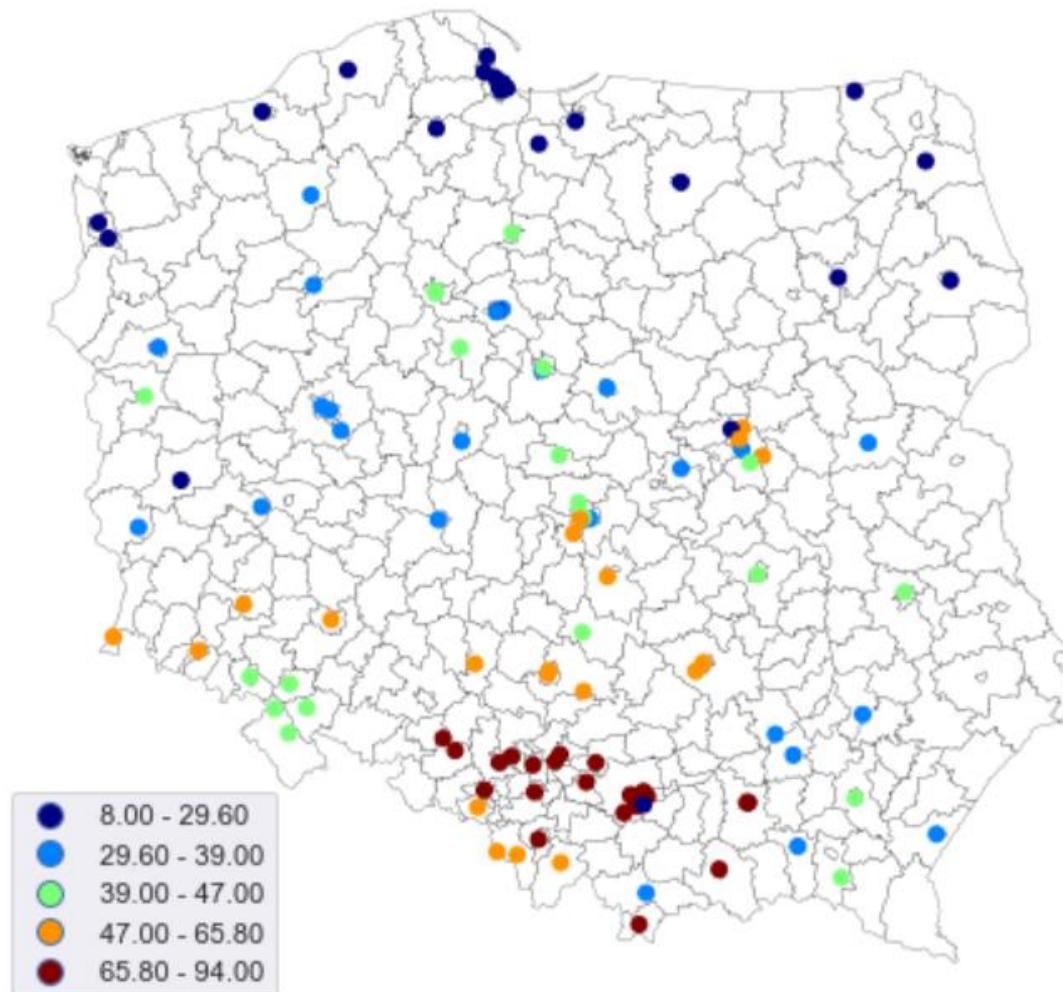
Modelowane średnie roczne w 2017
Na podstawie T,H% i prędkości wiatru



Liczba dni z PM powyżej 50 w 2017



Modelowana liczba dni z PM powyżej 50 w 2017
Na podstawie T,H% i prędkości wiatru



<https://github.com/urbanskigis>

Github (raportowanie i współpraca)

The screenshot shows a GitHub profile page for a user named Jacek Urbanski. The profile picture is a 4x4 grid of yellow squares. The profile summary includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and status indicators for repositories (4), projects (0), stars (0), followers (0), and following (0). The 'Overview' tab is selected. Below it, under 'Popular repositories', there are four repository cards:

- HEL-geodata-science**: A repository for a seminarium geoscience na Helu. It is a Jupyter Notebook and has 2 stars.
- Glacier-Terminus-Tracking**: A GIS tool for two-dimensional glacier-terminus change tracking. It has 1 star and 1 fork.
- urban-atlas-ludnosc**: A Jupyter Notebook repository.
- Smog-PM10**: A Jupyter Notebook repository. This card has a red arrow pointing to it from the top right.

On the right side of the profile page, there is a 'Customize your pins' section and a 'Set status' button.

Jacek Urbanski
urbanskigis

Overview Repositories 4 Projects 0 Stars 0 Followers 0 Following 0

Popular repositories

Customize your pins

HEL-geodata-science
Seminarium geoscience na Helu
Jupyter Notebook ★ 2

Glacier-Terminus-Tracking
A GIS tool for two-dimensional glacier-terminus change tracking
★ 1 ⌂ 1

urban-atlas-ludnosc
Jupyter Notebook

Smog-PM10
Analiza i modelowanie smogu PM10 w Polsce.
Jupyter Notebook

urbanskigis / Smog-PM10

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Analiza i modelowanie smogu PM10 w Polsce.

Edit

Manage topics

14 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

Branch: master Smog-PM10 / geo / ←

urbanskigis Add files via upload Latest commit 12ab3b2 15 minutes ago

geo Delete dat.txt 17 minutes ago

img Add files via upload 24 minutes ago

PM10_day2.csv Add files via upload 15 minutes ago

README.md Create README.md 30 minutes ago

Smog PM10 model.ipynb Add files via upload 21 minutes ago

README.md

Smog-PM10

Analiza i modelowanie smogu PM10 w Polsce.

Analiza wykorzystuje co godzinne pomiary stężenia PM10 na kiludziesięciu stacjach pomiarowych w Polsce zmierzone od stycznia 2016 do maja 2018. Obserwacje zostały pobrane ze stron EU. Zostały one uzupełnione o zmienne meteorologiczne i szereg zmiennych o charakterze przestrzennym. Do analizy wykorzystano metody data science.

W przedstawionym notebooku wykorzystano zmodyfikowane dane - średnie dobowe. Wykonano podstawowe mapy służące do oceny stopnia zanieczyszczenia, mapę średnich rocznych wartości PM10 oraz mapę liczby dni z wartościami powyżej 50 $\mu\text{g}/\text{m}^3$. Analizie poddano różne czynniki. Wykazano, że w skali roku podstawowymi czynnikami zmniejszającymi smog są temperatura, pokrycie terenu takie jak woda i zieleń, a także istnienie korytarzy powietrznych. Natomiast czynnikami

[Code](#)[Issues 0](#)[Pull requests 0](#)[Projects 0](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

Branch: master ▾

[Smog-PM10 / Smog PM10 model.ipynb](#)[Find file](#) [Copy path](#)

Add files via upload

0f11543 23 minutes ago

1 contributor

1.28 MB

[Download](#) [History](#)

Analiza i modelowanie smogu PM10 w Polsce



Wstęp

Projekt ten przedstawia analizę przestrzennej zmienności smogu w Polsce, na przykładzie stężenia PM10 w $\mu\text{g}/\text{m}^3$. W pierwszej części projektu przeprowadzono analizę rozmieszczenia smogu biorąc po uwagę podstawowe kryteria czystego powietrza EU i WHO (średnie dobowe stężenie roczne oraz liczbę dni ze stężeniem powyżej $50 \mu\text{g}/\text{m}^3$). W drugiej części zbudowano i przetestowano model predykcji średniego stężenia PM10 na dowolnym obszarze zabudowanym Polski w dowolnym czasie. Określono także wpływ różnych czynników na jakość powietrza w ciągu roku oraz w sezonie zimowym i letnim.

Import potrzebnych w projekcie bibliotek

urban-atlas-ludnosc

Jupyter Notebook

urbanskigis / urban-atlas-ludnosc

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Branch: master ▾ urban-atlas-ludnosc / Dane o ludności w URBAN ATLAS.ipynb Find file Copy path

urbanskigis Add files via upload 675161e on Apr 26

1 contributor

437 lines (436 sloc) | 13.9 KB

Dodanie atrybutów ze spisu powszechnego 2011 do obiektów zabudowy urban-atlas 2012

[Spis powszechny](#): (aktualności z datą 12.01.2017)

[Urban atlas](#):

Program poniżej dodaje dla każdego obiektu dowolnej polskiej warstwy pokrycia urban-atlas dane o ludności pozyskane w spisie powszechnym 2011 i udostępnione w siatce kilometrowej.

Do programu (ArcGIS Pro) wprowadzamy pustą geobazę, SHP urban-atlas dla danego obszaru i część siatki ze spisu powszechnego pokrywającą ten obszar. Obie warstwy powinny być sprowadzone do jednego układu współrzędnych (np. ETRS 1989 Poland CS92).

In [3]:

```
import numpy as np
import arcpy

arcpy.env.workspace=r"C:\SP8\POPULACJA\LudCorUrb\LudCorUrb.gdb"

corine_urb=r"C:\SP8\POPULACJA\CORINE_URBAN_POMORSKIE.shp" # ETRS 1989 Poland CS92
corine_urb
geostat_pop=r"C:\SP8\POPULACJA\PD_STAT_GRID_CELL_2011_URB_POM.shp" # ETRS 1989 Poland CS92 siatka pokrywająca corine_urb
corine_urb2=r"C:\SP8\POPULACJA\CORINE_URBAN_POP2.shp"

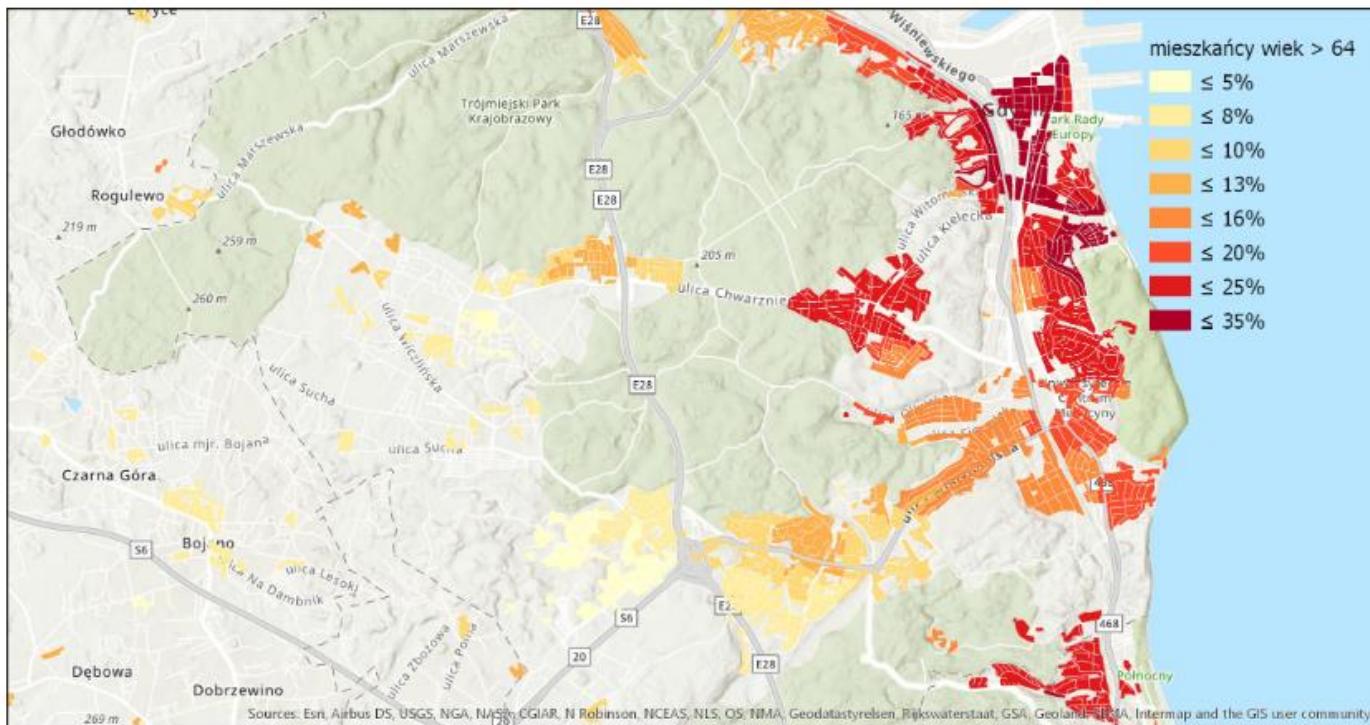
# do dopasowania z powyższych (pole z ID w warstwie po Union)
```

```
In [19]: # Czyszczenie warstw tymczasowych w geobazie
```

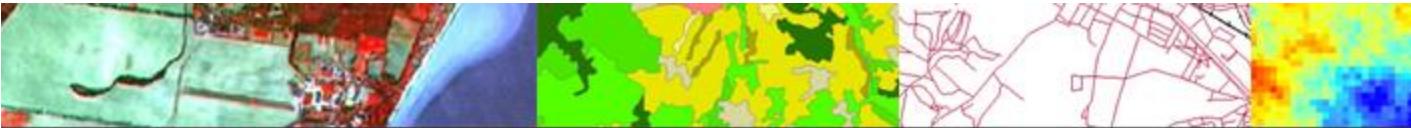
```
arcpy.Delete_management('temp_1')
arcpy.Delete_management('temp_2')
arcpy.Delete_management('temp_2s')
arcpy.Delete_management('temptab1')
arcpy.Delete_management('temptab2')
```

```
Out[19]: <Result 'true'>
```

Przykładowa mapa wynikowa przedstawia procentowy udział mieszkańców o wieku 65 lat i więcej



```
In [ ]:
```



Centrum GIS Wydziału Oceanografii i Geografii Uniwersytetu Gdańskiego

Geograficzne Systemy Informacyjne (GIS) wspierają badania naukowe i analizę danych. Wykorzystując analizę przestrzenną i modelowanie są coraz częściej używane do poszukiwania odpowiedzi na pytania badawcze. Dają możliwość łączenia różnorodnych danych i pomagają zrozumieć złożone systemy posiadające odniesienie przestrzenne.

Ideą utworzenia Centrum GIS na UG jest promowanie wykorzystania technologii GIS w pracy naukowej przez pracowników, doktorantów i studentów UG.

Seminaria programowania organizowane przez Centrum GIS

Kurs programowania w ArcGIS Pro za pomocą Pythona
Najbliższy planowany termin kursu: luty 2020

Kurs programowania w QGIS za pomocą Pythona
Najbliższy planowany termin kursu: 23 - 27.09.2019

Seminarium Python GeoData Science
Najbliższy planowany termin kursu:

Zapisy: dagmara.litwicka@gmail.com