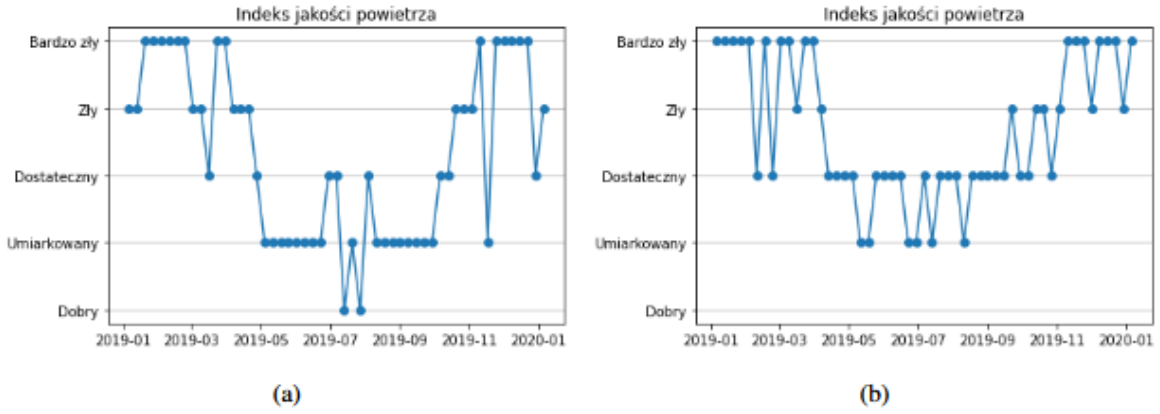
# Air quality prediction for Kraków using statistical models

Weronika Niedźwiedź

## 1. Project description

In this project, historical data pertaining to air pollution collected by one of the air pollution sensors in Kraków was used to predict its upcoming values. Multiple of the documented pollutants were the subject of separate predictions, namely PM10, PM2.5, NO2, SO2 and O3. As a result, a yearly prediction of each was obtained, as well as a prediction of the value of the overall air quality index commonly used in Poland - Indeks Jakości Powietrza (IJP).

The data used in predictions is sourced from the databases of Główny Inspektorat Ochrony Środowiska. They collect hourly measurements of various pollutants from multiple sensors located in most urban areas in Poland, including Kraków.




(a) final prediction for IJP (b) real IJP values

## 2. Data preparation

Data was read into a dataframe by using the Pandas library for Python.

Multiple statistical methods were used. In cases where missing data occurred during the working time of the sensor, the values were calculated by using interpolation as implemented in Pandas:

```
dane=dane.interpolate(limit_area="inside")
```

Due to the data being collected hourly for years (in some cases, over a decade), the amount of data used in the models had to be reduced. Instead of using hourly values, maximum values for each week were picked, so that the worst air quality index for each week could be calculated. The data was also limited to that from years 2014-2018. The measurements from the year 2019, which at the time was the latest available dataset, were used for testing the predictions.

## 3. The Box-Jenkins method

The Box-Jenkins method is used in order to choose and apply autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) to the known values of a time series.

Both ARMA and ARIMA implement autoregression and moving averages:

**AR (autoregression) -** assumes that there is an autocorrelation in the series (each value depends on the past values). It's expressed with a parameter *p* - the number of past calculations taken into account when calculating the present one. Each past value is assigned a certain weight, which is obtained during optimization of the model.

**MA (moving average)** - used to provide a prediction based on past and current error terms. It works like the AR component, but uses error terms instead of measured values. It's number of past errors used, parameter *q*, has to be well adjusted - if it's too small, the prediction will be overly sensitive to random momentary deviations, but with a too big value it will not respond to changes in the time series overall.

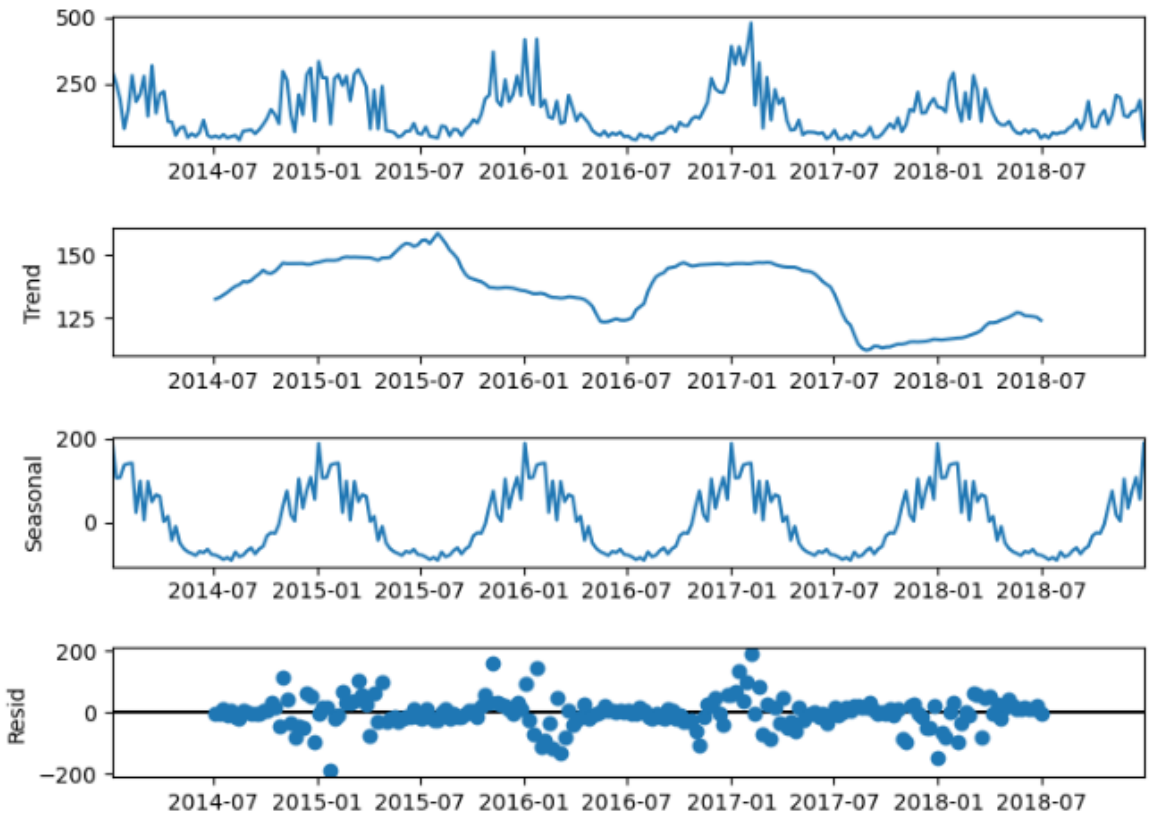
**I (integrated)** - present in ARIMA, but not ARMA. It's used to remove non-stationarity from the data, if it exists - the time series is differentiated until it's reached. The related parameter, d, is the degree of differentiation.

**Seasonality** - ARIMA adjusted to data with seasonal trends is known as **SARIMA**. It requires additional parameters *P, Q* and *D* - equivalents to *p, q* and *d* for seasonality. It also requires parameter *m* - duration of a seasonal cycle.

**Box-Jenkins method** implements the following steps:

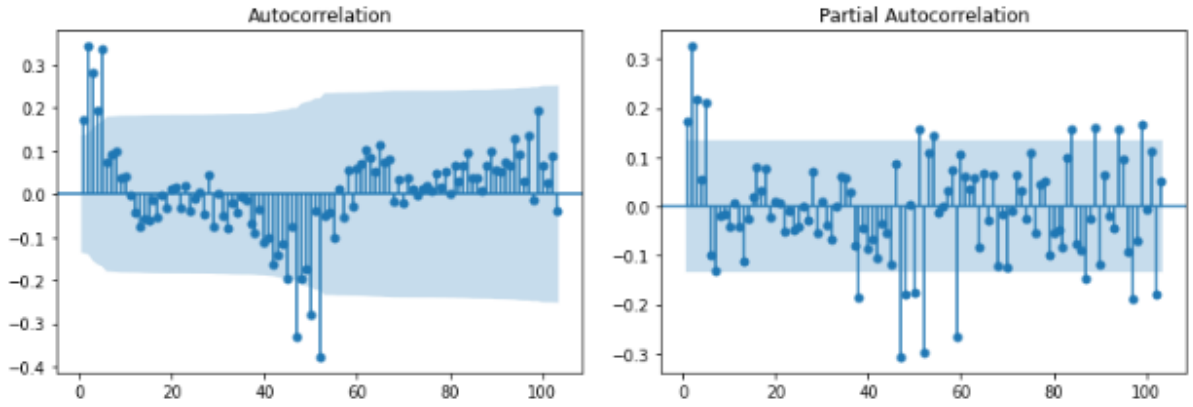
1. **Identification** - it serves to detect if the time series is stationary, and if not, to perform integration. It also estimates the initial values of $p$ and $q$. ACF and PACF are used for it.
2. **Parameter estimation** - estimating the best parameters for the used data. In Python, the library Pmdarima implements this as a function *auto_arima*. It checks various parameter values according to a chosen estimate of correctness.
3. **Statistical model checking** - making sure that the model with the chosen parameters works correctly. It usually involves checking if the function of the error term resembles a random series - if not, the parameters probably haven't been chosen correctly.

The Box-Jenkins method was applied to the investigated data. For the identification step, decomposition as implemented in the Statsmodels library was used. The following are its results for PM10 measurements:




It suggests a discernable seasonal pattern lasting a year - so, 52 measurements. Therefore differentiation is necessary.

The following are ACF and PACF of the differentiated data:




In ACF, there are 5 values that go above the confidence interval before one appears inside of it. It means that the initial value of *q* will be 5.

In PACF that number is 3, which will be the initial value of *p*.

For P and Q we consider whether there are significant jumps above or below the confidence level at the indexes that are multiples of the seasonal period. Here, good initial values might be P=2 and Q=1.

The series was differentiated once seasonally, therefore d=1 and D=0.

The initial parameters of the model therefore make up the following model: SARIMA(3,0,5)x(2,1,1)

Therefore the function for choosing the final values was:

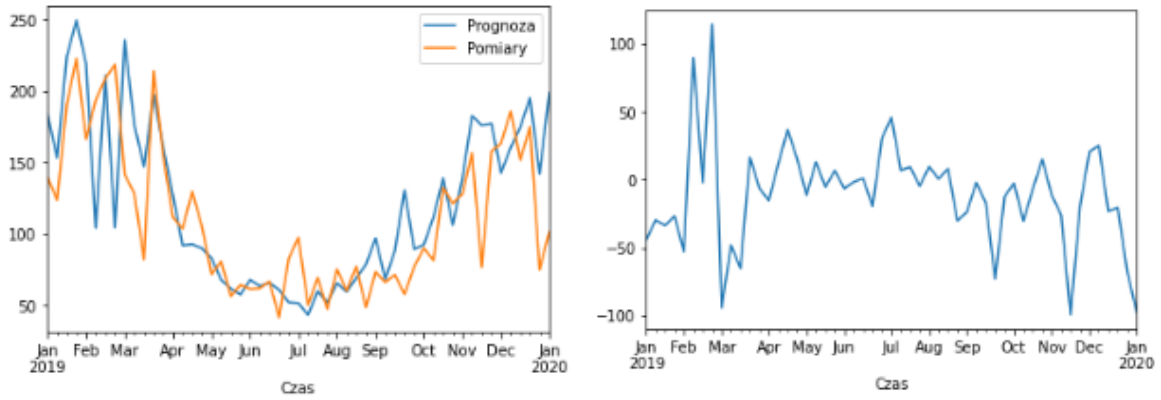
```
fit = auto_arima(data.astype(int), start_p = 4, start_q = 5,
                                   start_P = 2, start_Q = 1,
                                   m = 52, seasonal = True)
```

The following are the resulting models picked for each dataset:

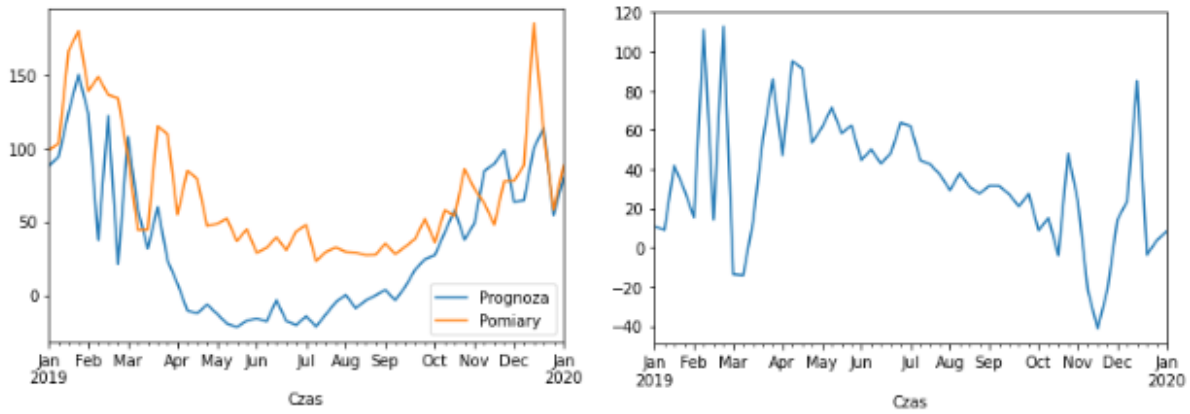
| Pomiar | Model |
|---|---|
| PM10 | SARIMA(1,0,2)x(0,1,1) |
| PM2.5 | SARIMA(1,0,1)x(1,1,1) |
| NO2 | SARIMA(0,0,1)x(0,1,1) |
| SO2 | SARIMA(3,0,1)x(2,1,0) |
| O3 | SARIMA(0,0,1)x(1,1,1) |

The following are the results of prediction with these models(left) and the difference between them and the actual data (right):

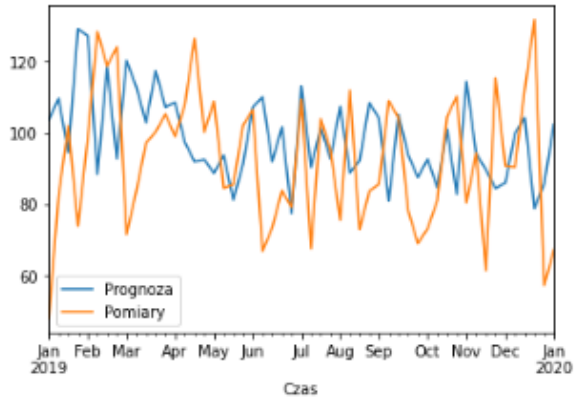
a) PM10

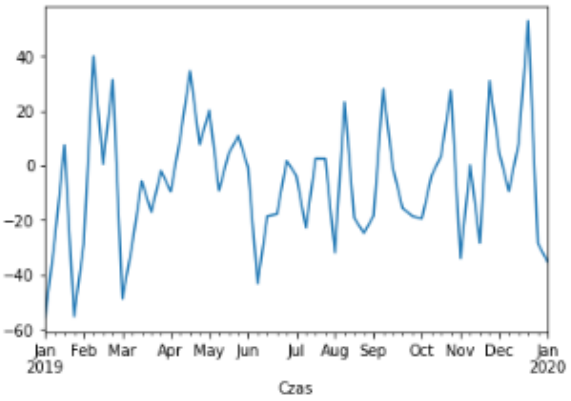



b) PM2.5

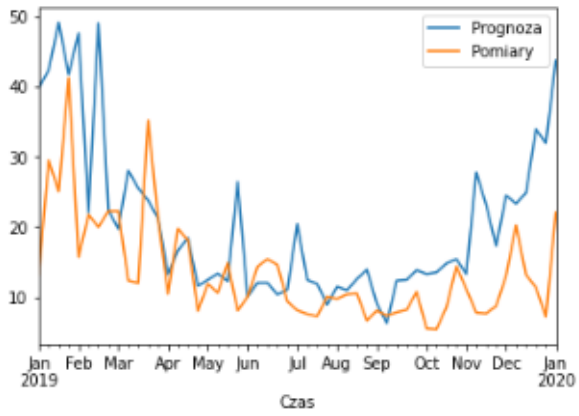


c) NO2



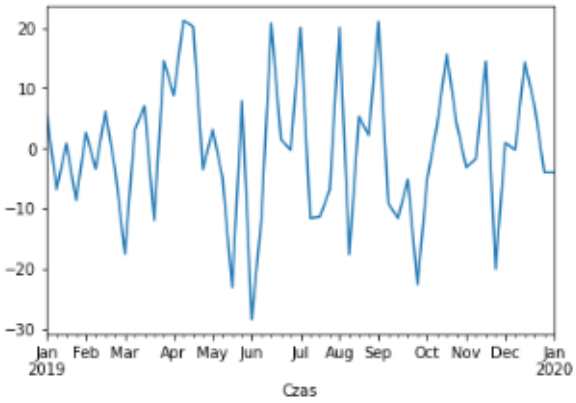




d) SO2





e) O3

The following are MAPE values for each prediction:

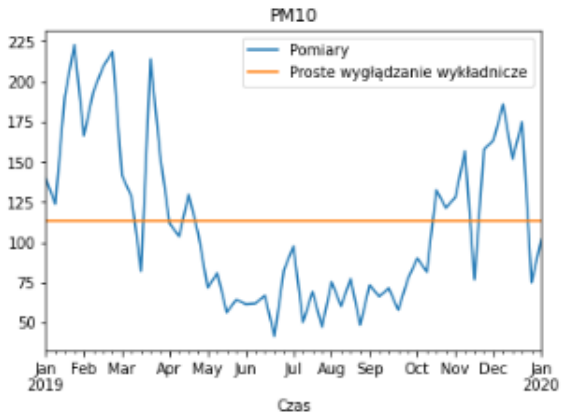
| Pomiar | MAPE |
|---|---|
| PM10 | 27.608 |
| PM2.5 | 74.598 |
| NO2 | 23.627 |
| SO2 | 71.832 |
| O3 | 10.546 |

## 4. Exponential smoothing

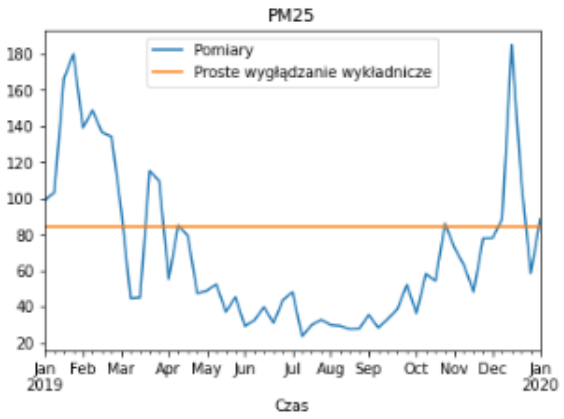
There are multiple methods implementing prediction using exponential smoothing.

1. **Single Exponential Smoothing** - uses the previous value of the time series and the moving average at the previous point of time summed up with certain weights to generate the next value. It's used with data that fluctuates around a certain level that doesn't change or changes slowly.
2. **Double Exponential Smoothing** - it's used for data with a distinguishable trend. It sums the prediction used in SES with a similar prediction for the trend. To account for the changing trend, this model frequently also implements trend damping, either additive or multiplicative.
3. **Holt-Winters Method** - also known as triple exponential smoothing; it is used in data which also includes a seasonal cycle. It has two variants, determining how the seasonal component is used to influence the final value of the prediction:
    a. **additive** - used when the seasonal changes are constant in time; that is, if the rising or falling of the trend doesn't cause the seasonal fluctuation to become larger or smaller.
    b. **multiplicative** - used when change in the value of the trend influences the value of seasonal fluctuation.

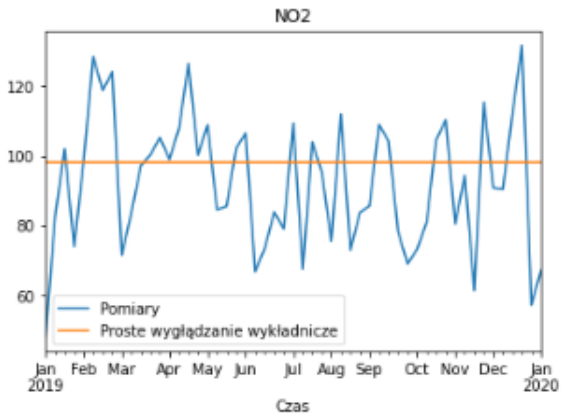
All three methods were performed on every dataset, using the implementations from the Statsmodels library. The following are the results for SES and MAPE values for each prediction:



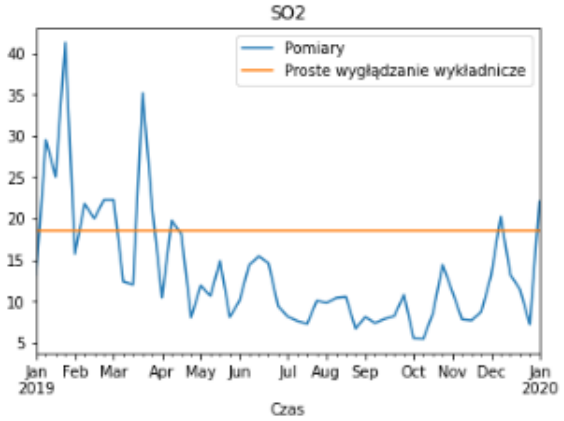

(a)




(b)



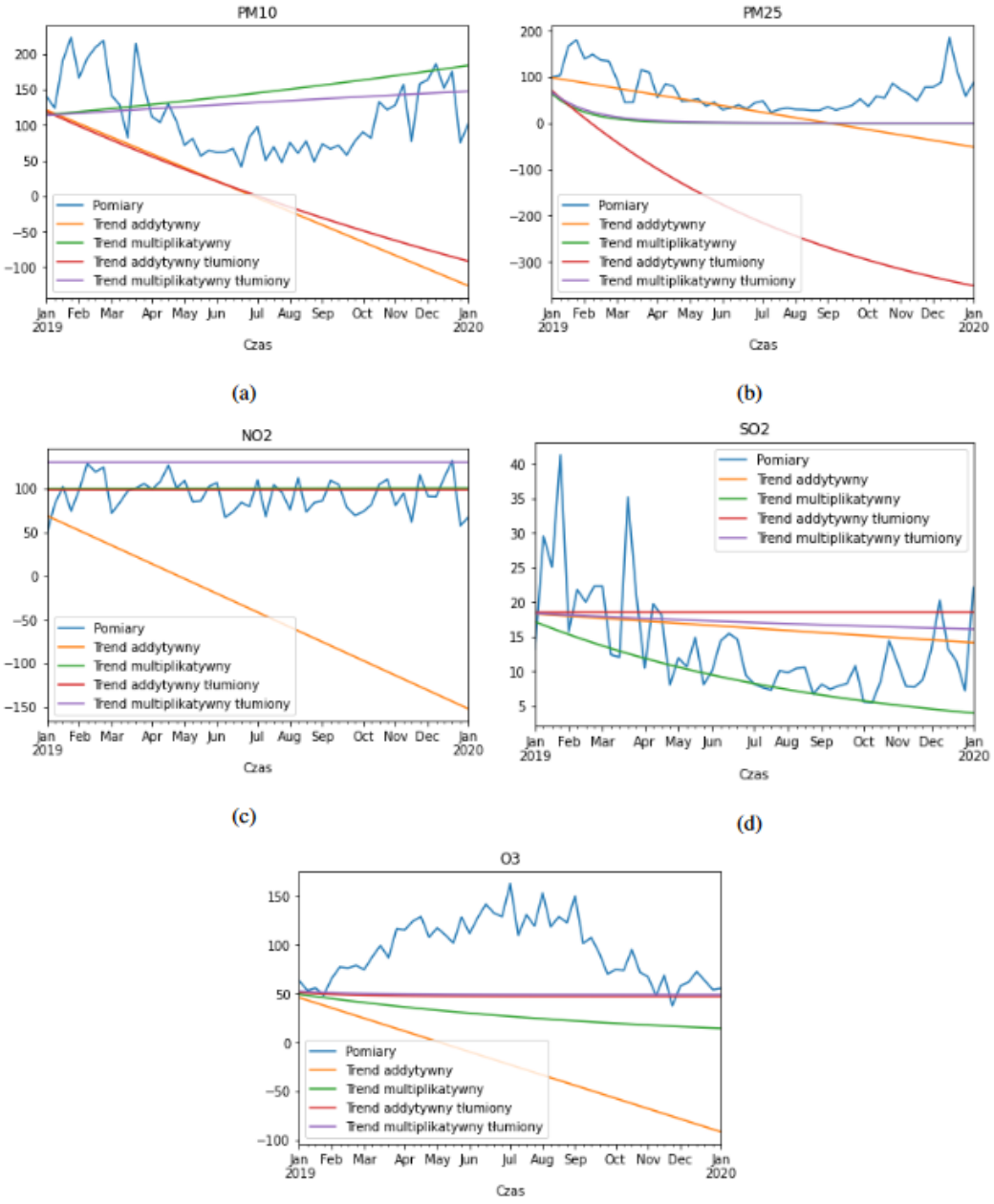

(c)




(d)


| Pomiar | MAPE |
|---|---|
| PM10 | 48.405 |
| PM2.5 | 82.772 |
| NO2 | 20.657 |
| SO2 | 77.305 |
| O3 | 37.828 |

The following are DES predictions with various methods of trend damping, as well as their MAPE values:

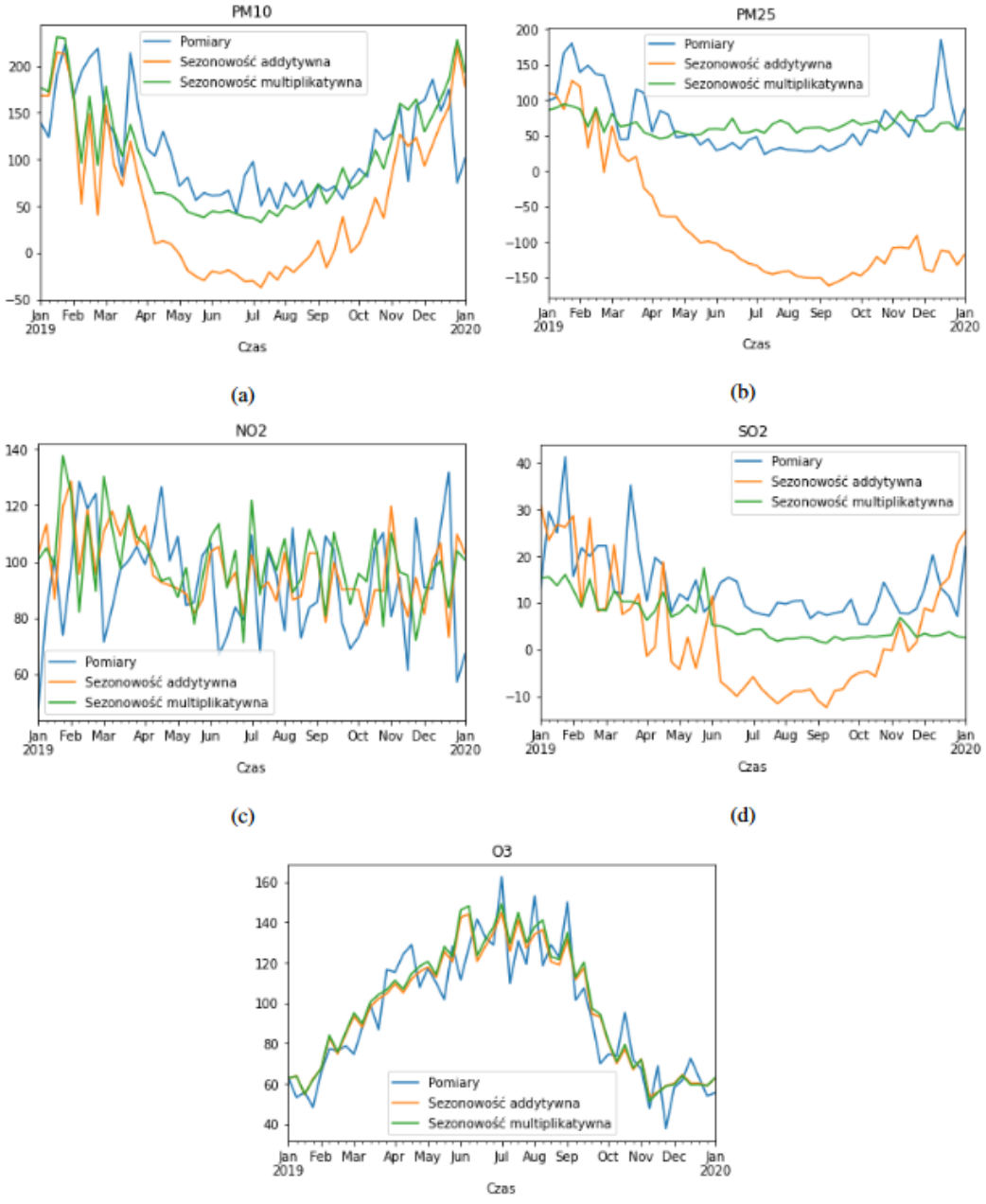



| Pomiar | Trend | | | |
|---|---|---|---|---|
| | Addytywny | Multiplikatywny | Addytywny tłumiony | Multiplikatywny tłumiony |
| PM10 | 109.655 | 75.887 | 101.822 | 62.158 |
| PM2.5 | 70.963 | 94.890 | 519.017 | 93.102 |
| NO2 | 150.613 | 21.390 | 20.620 | 48.174 |
| SO2 | 57.004 | 32.067 | 77.325 | 64.957 |
| O3 | 130.669 | 66.330 | 44.041 | 42.391 |

On that basis, the best method of handling the trend was chosen in each case:

| Pomiar | Trend |
|---|---|
| PM10 | Multiplikatywny tłumiony |
| PM2.5 | Addytywny |
| NO2 | Addytywny tłumiony |
| SO2 | Multiplikatywny |
| O3 | Brak |

Using these choices, Holt-Winters method was used:



Based on the following MAPE values, the best way of calculating seasonal was chosen, and in result the best exponential smoothing model was chosen for each dataset:

| Pomiar | Sezonowość | |
|---|---|---|
| | Addytywna | Multiplikatywna |
| PM10 | 79.418 | 31.192 |
| PM2.5 | 303.926 | 51.360 |
| NO2 | 24.050 | 25.608 |
| SO2 | 116.414 | 56.469 |
| O3 | 11.180 | 11.386 |

| Pomiar | Trend | Sezonowość |
|---|---|---|
| PM10 | Multiplikatywny tłumiony | Multiplikatywna |
| PM2.5 | Addytywny | Multiplikatywna |
| NO2 | Addytywny tłumiony | Addytywna |
| SO2 | Multiplikatywny | Multiplikatywna |
| O3 | Brak | Addytywna |

## 5. Final results

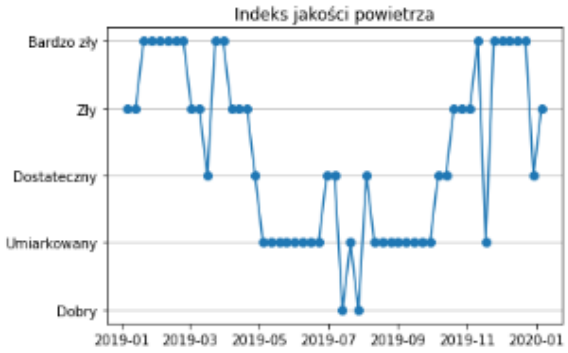
The following table lists the lowers MAPE value obtained for each dataset with each method:

| Pomiar | Model | | | |
|---|---|---|---|---|
| | SARIMA | Pojedyncze wygładzanie wykładnicze | Podwójne wygładzanie wykładnicze | Metoda Holta-Winters'a |
| PM10 | 27.608 | 48.405 | 62.158 | 31.192 |
| PM2.5 | 74.598 | 82.772 | 70.963 | 51.360 |
| NO2 | 23.627 | 20.657 | 20.620 | 24.050 |
| SO2 | 71.832 | 77.305 | 32.067 | 56.469 |
| O3 | 10.546 | 37.828 | 42.391 | 11.180 |

Therefore, the following seemed to be the most appropriate models for each:

| Pomiar | Model dający najlepsze wyniki |
|---|---|
| PM10 | SARIMA |
| PM2.5 | Metoda Holta-Winters'a |
| NO2 | Podwójne wygładzanie wykładnicze |
| SO2 | Podwójne wygładzanie wykładnicze |
| O3 | SARIMA |

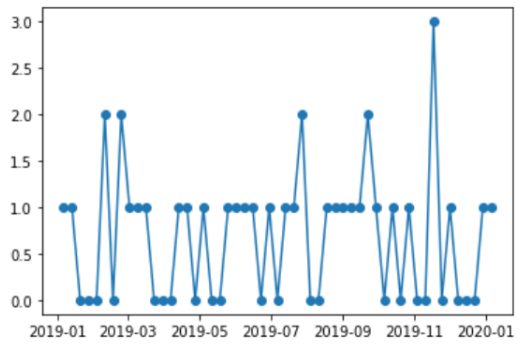
These models, as calculated previously, were used to predict the worst air quality index (IJP) for each week of the year 2019. The following are the real values (a) compared to the predicted values (b):

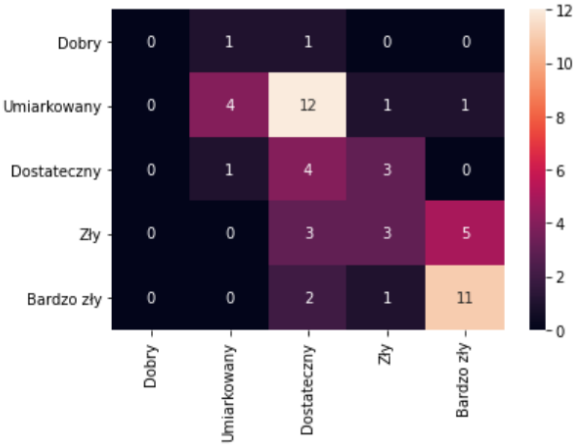



(a)


(b)

Below there is a graph of absolute difference between the two, as well as the error matrix:



## 6. Discussion

In the final results for the IJP, the most frequently occurring error is a difference of only one degree of the IJP. If we consider this an acceptable error given the long term of the prediction, then over 90% of the predictions could be considered correct.

The predictions usually indicated a worse value of IJP than the real value. It might mean that considering the trends, pollution in 2019 was not as bad as it was in 2014-2018.

The most challenging part of the work was choosing and using SARIMA models. It involves lengthy calculations, which in some cases took hours to compute. These models would probably prove to be impossible to use if the data from all available years (2000-2019) was used, instead of just weekly maxima of 4 years. If the predictions need to be made in real time, or if many need to be made in a short time frame, exponential smoothing methods might be a better choice, as in many cases they displayed similar or better overall results than much more complicated SARIMA methods.