



AGH

Metody Inteligencji Obliczeniowej

Predykcja nowych przypadków COVID-19

Jan Zajda, Wojciech Jędraski, Tomasz Gajda

Predykcja nowych przypadków COVID-19	1
1. Temat projektu	1
2. Założenia technologiczne	2
3. Źródło danych wejściowych	2
4. Parsowanie danych	3
5. Wizualizacja danych	4
6. Macierz korelacji i wykresy autokorelacji	7
7. Dobór danych wejściowych	9
8. Predykcja - uczenie SSN	10
9. Przykładowe wyniki predykcji	11
Predykcja następnego tygodnia	11
Predykcja średniej ilości dziennych zakażeń następnych tygodni	12
10. Dokumentacja użytkownika	13

1. Temat projektu

Projekt polega na stworzeniu narzędzia, pozwalającego na predykcję nowych przypadków **COVID-19**. Do tworzenia predykcji wykorzystane zostaną **Sztuczne Sieci Neuronowe**. Z racji szeregu czasowego dane zostaną podzielone na **uczące** i **testujące** zgodnie z osią czasu. Jako dane wejściowe użyte zostaną niektóre wartości szeregu czasowego z przeszłości - do ich ustalenia przeprowadzona zostanie **analiza autokorelacji**, z której wyłoniony powinien zostać kluczowy input.

2. Założenia technologiczne

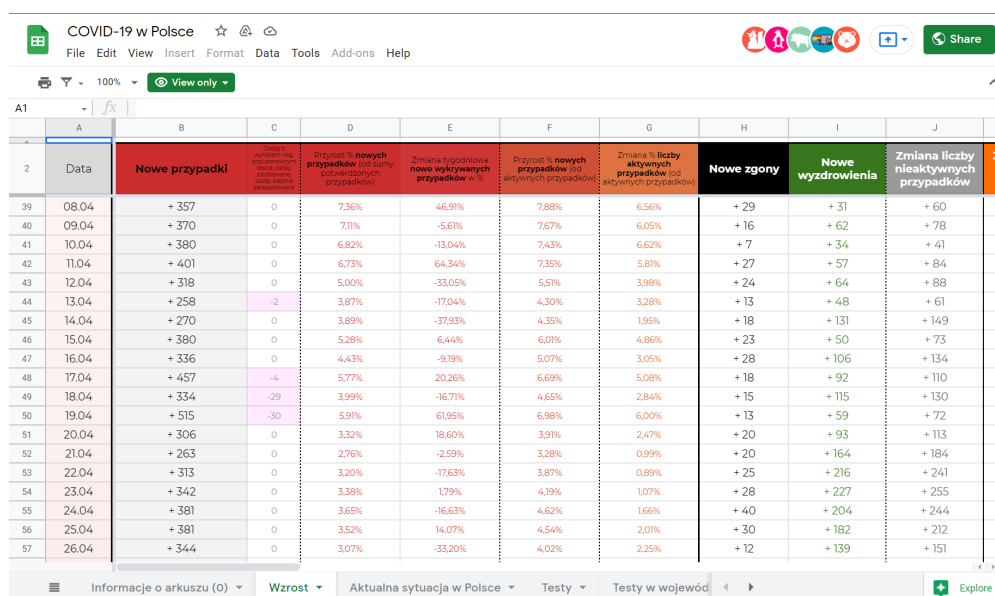
Do stworzenia narzędzia korzystającego ze **Sztucznej Sieci Neuronowej** wykorzystany został **Python** w najnowszej wersji. Główne biblioteki, z których korzystaliśmy to **pandas**, **plotly** i **scikit-learn**.

3. Źródło danych wejściowych

Do wytworzenia predykcji korzystamy z danych przygotowanych przez Pana **Michała Rogalskiego**. Dane są w postaci plików CSV, umieszczone są w folderze o nazwie **data**. Wszystkie dane oraz pliki zawarte są publicznym arkuszu pod tym linkiem:

<http://bit.ly/covid19-poland>

Dane dla większej przejrzystości zostały podzielone na trzy główne pliki - **wzrost**, **testy i szczepienia**. Dla wygody, korzystaliśmy z pobranych arkuszy, dlatego w projekcie zawarte są dane w zakresie od **2 marca 2020**, do **17 maja 2021***.



	A	B	C	D	E	F	G	H	I	J
	Data	Nowe przypadki	Nowe zgon	Przyrost % nowych przypadków (od sumy poprzedzających przypadków)	Zmiana tygodniowa nowo wykrywanych przypadków w %	Przyrost % nowych przypadków (od aktywnych przypadków)	Zmiana % liczby aktywnych przypadków (od aktywnych przypadków)	Nowe zgony	Nowe wyzdrowienia	Zmiana liczby nieaktywnych przypadków
39	08.04	+ 357	0	7,36%	46,91%	7,88%	6,56%	+ 29	+ 31	+ 60
40	09.04	+ 370	0	7,11%	-5,61%	7,67%	6,05%	+ 16	+ 62	+ 78
41	10.04	+ 380	0	6,82%	-13,04%	7,43%	6,62%	+ 7	+ 34	+ 41
42	11.04	+ 401	0	6,73%	64,34%	7,35%	5,81%	+ 27	+ 57	+ 84
43	12.04	+ 318	0	5,00%	-33,05%	5,51%	3,98%	+ 24	+ 64	+ 88
44	13.04	+ 258	-2	3,87%	-17,04%	4,30%	3,28%	+ 13	+ 48	+ 61
45	14.04	+ 270	0	3,89%	-37,93%	4,35%	1,95%	+ 18	+ 131	+ 149
46	15.04	+ 380	0	5,28%	6,44%	6,01%	4,86%	+ 23	+ 50	+ 73
47	16.04	+ 336	0	4,43%	-9,19%	5,07%	3,05%	+ 28	+ 106	+ 134
48	17.04	+ 457	-4	5,77%	20,26%	6,69%	5,08%	+ 18	+ 92	+ 110
49	18.04	+ 334	-29	3,99%	-16,77%	4,65%	2,84%	+ 15	+ 115	+ 130
50	19.04	+ 515	-30	5,91%	61,95%	6,98%	6,00%	+ 13	+ 59	+ 72
51	20.04	+ 306	0	3,32%	18,60%	3,91%	2,47%	+ 20	+ 93	+ 113
52	21.04	+ 263	0	2,76%	-2,59%	3,28%	0,99%	+ 20	+ 164	+ 184
53	22.04	+ 313	0	3,20%	-17,63%	3,87%	0,89%	+ 25	+ 216	+ 241
54	23.04	+ 342	0	3,38%	1,79%	4,19%	1,07%	+ 28	+ 227	+ 255
55	24.04	+ 381	0	3,65%	-16,63%	4,62%	1,66%	+ 40	+ 204	+ 244
56	25.04	+ 381	0	3,52%	14,07%	4,54%	2,01%	+ 30	+ 182	+ 212
57	26.04	+ 344	0	3,07%	-33,20%	4,02%	2,25%	+ 12	+ 139	+ 151

Rys. 1 - Plik **Google Spreadsheets** stworzony przez Michała Rogalskiego, zawierający wszystkie dostępne dane na temat przebiegu COVID-19 w Polsce

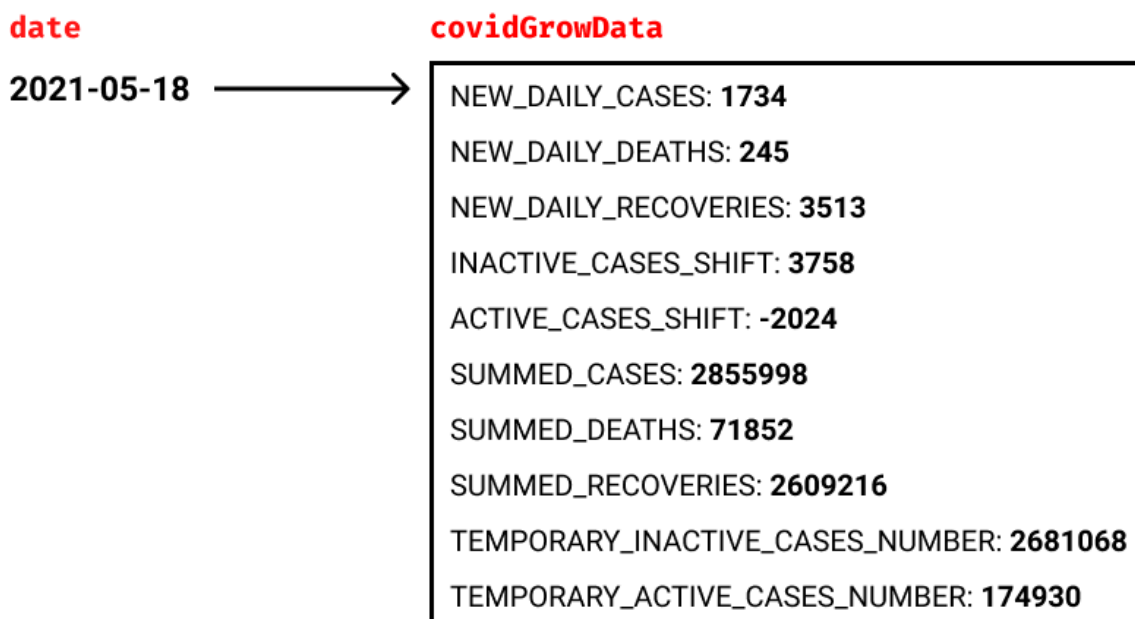
* mówiąc o danych na temat **dzienniej ilości zakażeń i testów** - z wiadomych względów dane na temat dzienniej ilości podawanych szczepionek zaczynają się dopiero **17 lutego 2021 roku**

4. Parsowanie danych

Dane z plików CSV wymagają parsowania - w folderze **reader** znajduje się plik o nazwie **csvReader.py**, który służy do przetworzenia danych z plików CSV na struktury danych dostępne w Pythonie.

W pliku utworzone zostały **trzy klasy** - po jednej na rodzaj (**CovidGrow**, **CovidTest** i **Vaccination**) - w których dostępne są metody czytające dane z plików CSV. Metody te przetwarzają dane na słowniki, które poszczególnym datom przypisują wszystkie dane dotyczące specyficznego rodzaju. Przykładowo wyciągając ostatni element z obiektu klasy CovidGrow, otrzymujemy:

```
covidGrowDetails = readCovidGrow()  
date, covidGrowData = covidGrowDetails.popitem()
```



Rys. 2 - Graficzne przedstawienie **formatu danych** stosowanych w projekcie

Taka struktura pozwala nam na łatwą manipulację danymi i przystępny dostęp do wszystkich informacji.

5. Wizualizacja danych

W folderze o nazwie **visualization**, możemy znaleźć klasę o nazwie **CovidVisualization** zawierającą pięć metod służących do wizualizacji danych:

- **linear_covid_data_plots** - metoda służąca do tworzenia wykresu liniowego podanych wartości,
- **bar_autocorrelation_plots** - metoda przedstawiająca wykresy autokorelacji dla różnych danych wejściowych,
- **correlation_matrix_plot** - metoda przedstawiająca macierz korelacji zmiennych na wykresie,
- **bar_plot** - metoda służąca do tworzenia wykresu słupkowego podanych wartości,
- **week_avg_prediction** - metoda porównująca na wykresie wyniki przewidywań średniej ilości dziennych zachorowań w trzech kolejnych tygodniach z realnymi wartościami,
- **plot_prediction** - metoda porównująca na wykresie wyniki przewidywań ilości dziennych zachorowań w określonym okresie czasu - metoda składa wykresy przewidywań, danych testowych i treningowych w jeden wykres,

Po wywołaniu metody, lokalnie w przeglądarce zostaje włączony interaktywny wykres, na którym możemy zobaczyć dane przekazane do metod. Do stworzenia wizualizacji skorzystaliśmy z biblioteki **plotly**.



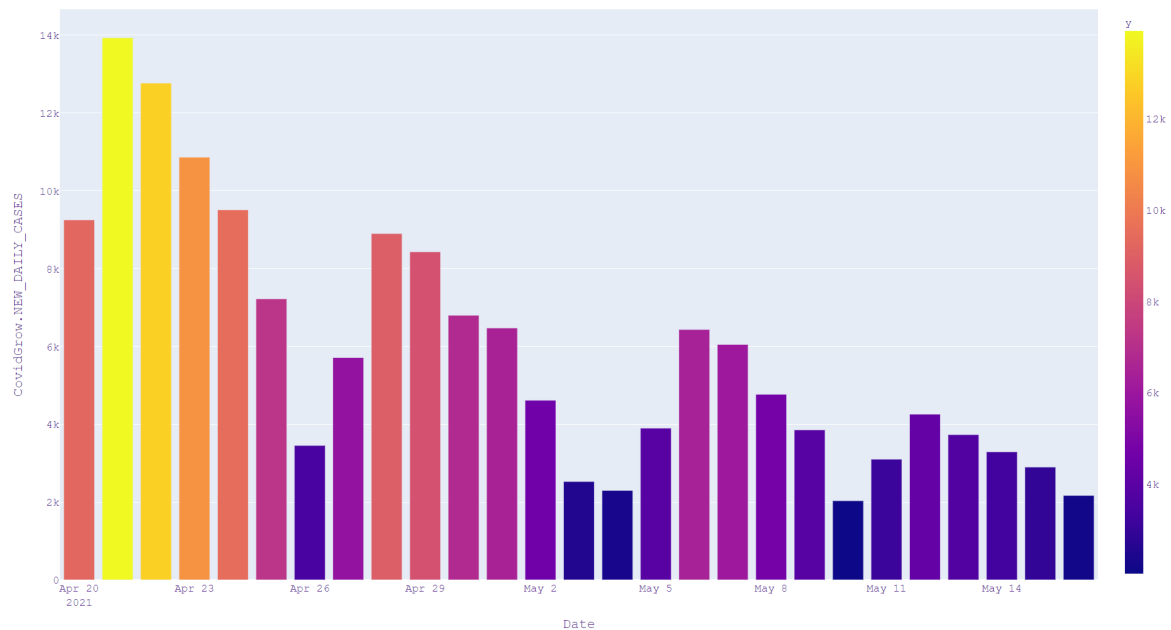
Rys. 3 - Logo biblioteki **plotly**

Po wywołaniu metody, lokalnie w przeglądarce zostaje włączony interaktywny wykres, na którym możemy zobaczyć dane przekazane do metod.

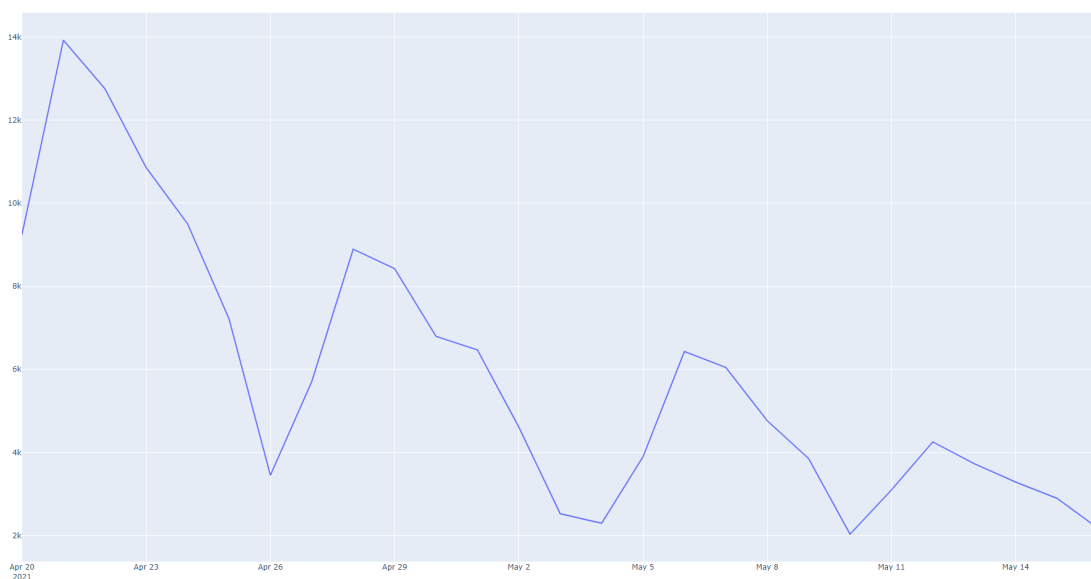
Przykłady działania wizualizacji przedstawione zostaną przy użyciu danych z dni **od 20 kwietnia 2021 roku do 17 maja 2021 roku**.

Przedstawiają one odpowiednio:

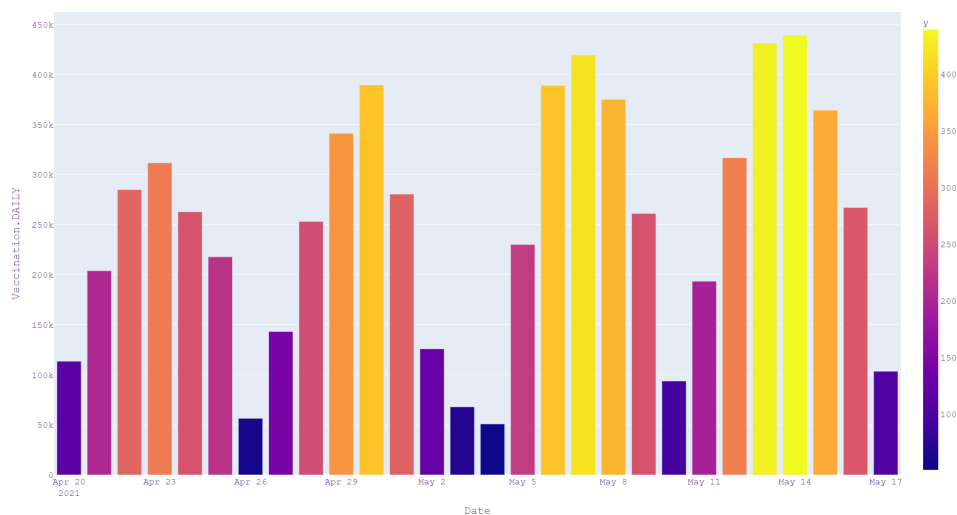
- Ilość dziennych przypadków zakażeń (w postaci liniowej i słupkowej),
- Ilość dziennych dawek szczepionek (w postaci liniowej i słupkowej)



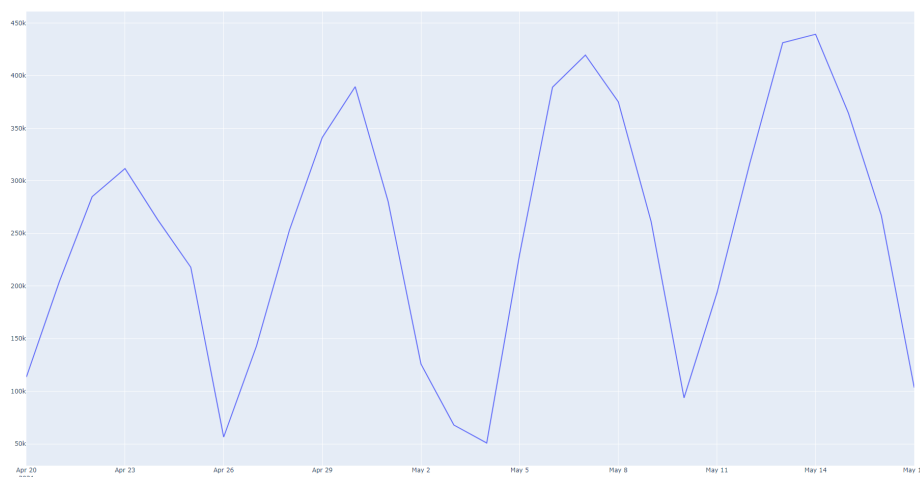
Rys. 4 - Wykres **słupkowy** przedstawiający ilość przypadków (20.04 - 17.05.2021)



Rys. 5 - Wykres **liniowy** przedstawiający ilość przypadków (20.04 - 17.05.2021)

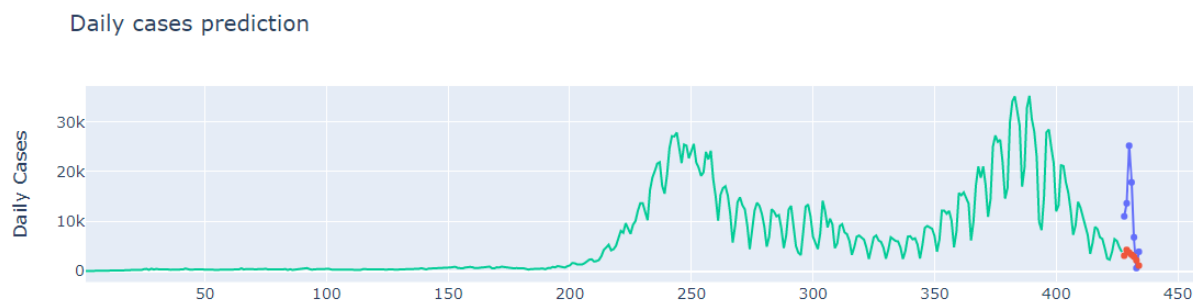


Rys. 6 - Wykres **słupkowy** przedstawiający ilość podanych szczepionek (20.04 - 17.05.2021)



Rys. 7 - Wykres **liniowy** przedstawiający ilość podanych szczepionek (20.04 - 17.05.2021)

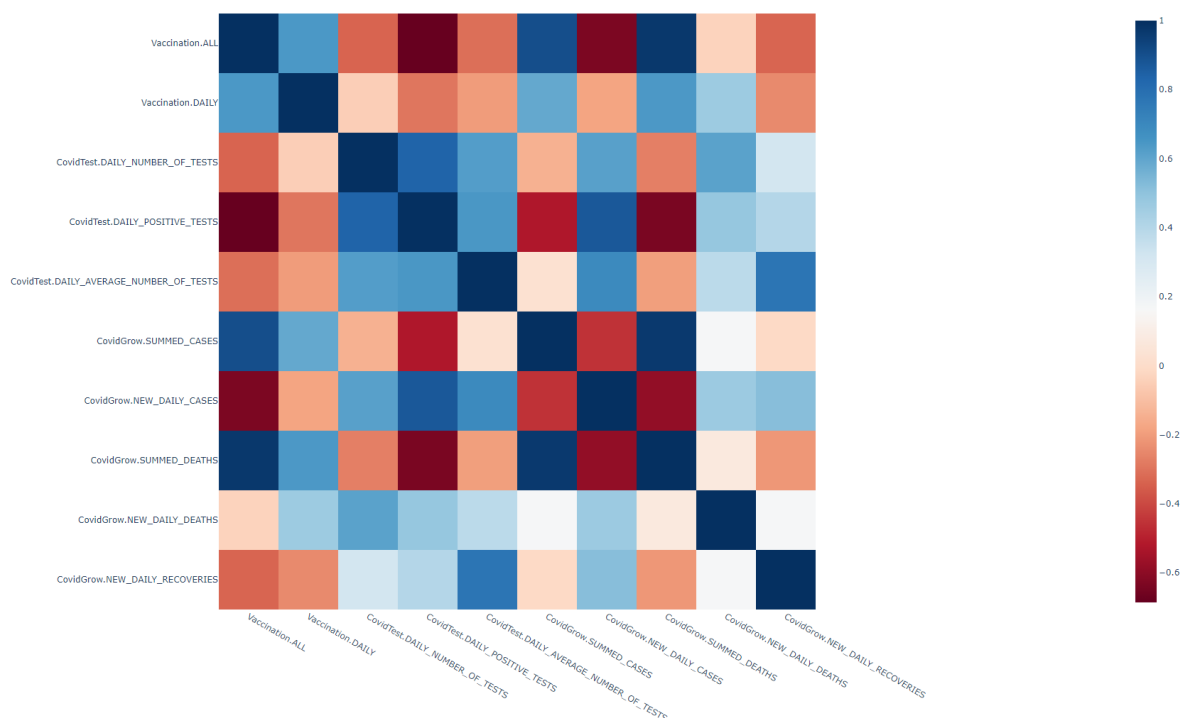
Jeden z wykresów predykcji pokazuje również wyniki przedstawione na tle danych treningowych dostarczonych w procesie:



Rys. 8 - Wizualizacja predykcji na tle **danych treningowych**

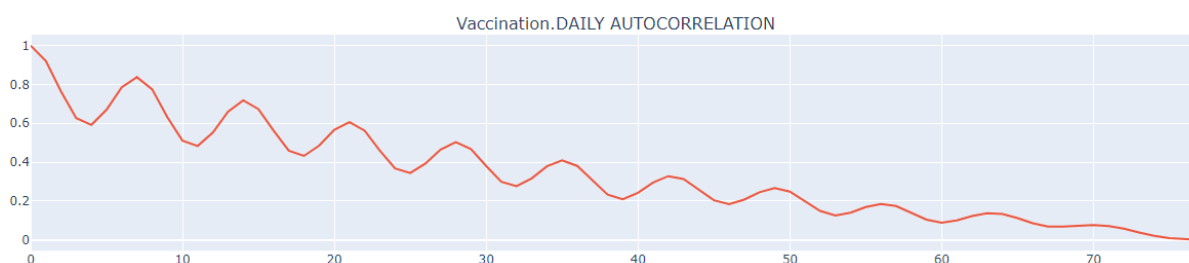
6. Macierz korelacji i wykresy autokorelacji

By wybrać spośród dużej ilości potencjalnych danych te, które mają znaczenie w przypadku próby predykcji przypadków COVID-19, skorzystaliśmy z analizy **korelacji** między zmiennymi oraz analizy **autokorelacji**. Korelację między zmiennymi opisaliśmy na macierzy:

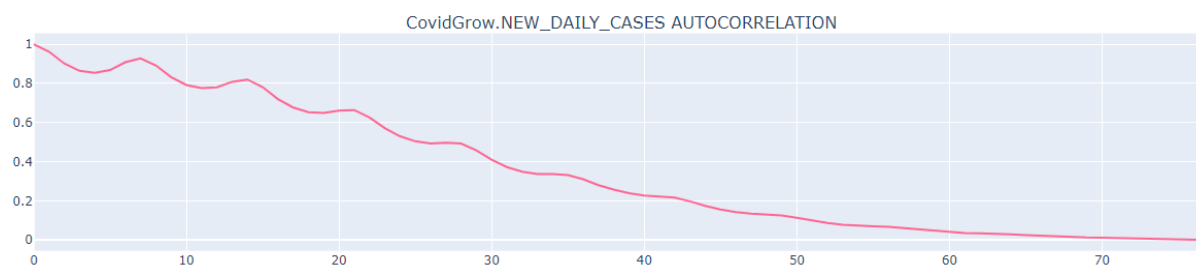


Rys. 9 - Wizualizacja **macierzy korelacji** potencjalnych wartości wejściowych

Autokorelacja (korelacja sygnału z poprzednimi jego stanami), została przedstawiona na osobnym wykresie dla każdej ze zmiennych, na przykład:

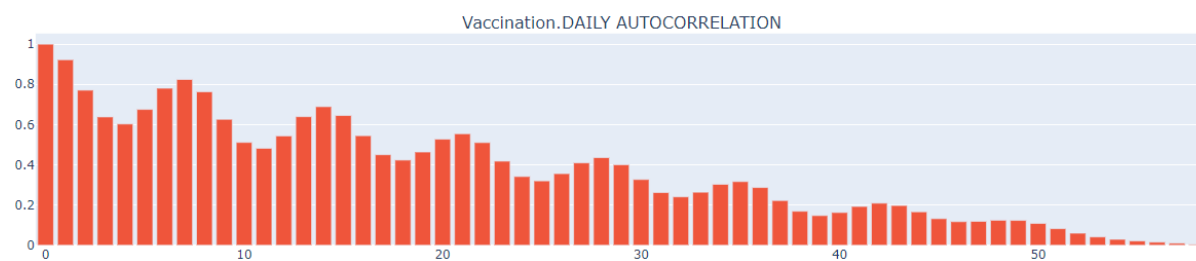


Rys. 10 - Wizualizacja **autokorelacji** dla **dziennej ilości podanych szczepionek** (liniowy)

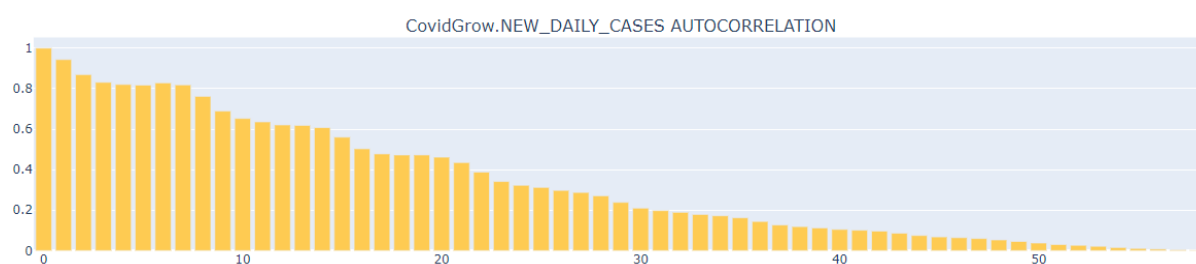


Rys. 11 - Wizualizacja **autokorelacji** dla **dziennej ilości przypadków** (liniowy)

Na początku wykresy autokorelacji przedstawione zostały na wykresach **liniowych**, ale zgodnie z zaleceniami zmienione zostały na wykresy **słupkowe**.



Rys. 12 - Wizualizacja **autokorelacji** dla **dziennej ilości podanych szczepionek** (słupkowy)



Rys. 13 - Wizualizacja **autokorelacji** dla **dziennej ilości przypadków** (słupkowy)

7. Dobór danych wejściowych

Po przejrzeniu wyników analizy autokorelacji przedstawionej w postaci wykresów, ustaliliśmy że nie dają nam one jasnych wyznaczników mówiących o tym, z jakiego okresu dane powinniśmy brać. Model niewiele różnić się będzie przy zmianie okresu (z np. 7 na 14 dni) - w tym przypadku liczy się **tendencja wzrostowa** i lepszej aproksymacji krzywej zachorowań nie otrzymamy stosując inny zakres dat.

```
def choose_columns(corr_data):
    goal = str(CovidGrow.NEW_DAILY_CASES)
    corr_type = 'corrMatrix'

    corr = corr_data[corr_type][goal]
    data = dict()

    data[Vaccination] = []
    data[CovidGrow] = []
    data[CovidTest] = []

    for key in Vaccination:
        if corr[str(key)] > HIGH_CORR or corr[str(key)] < -HIGH_CORR:
            data[Vaccination].append(key)

    for key in CovidGrow:
        if corr[str(key)] > HIGH_CORR or corr[str(key)] < -HIGH_CORR:
            data[CovidGrow].append(key)

    for key in CovidTest:
        if corr[str(key)] > HIGH_CORR or corr[str(key)] < -HIGH_CORR:
            data[CovidTest].append(key)

    return data
```

Rys. 14 - Funkcja odpowiedzialna za dobór danych wejściowych

Przy każdym włączeniu predykcji, **dynamicznie** obliczane są wartości korelacji zmiennych z poszukiwaną przez nas **dzienną liczbą nowych przypadków** - korelacja sprawdzana jest dla okresu czasu wyznaczonego przez podane w argumentach granice. W programie podać możemy minimalną wartość korelacji tych zmiennych - w ten sposób automatycznie wybierane są dane które weźmiemy pod uwagę w momencie uczenia sieci.

8. Predykcja - uczenie SSN

Do predykcji wykorzystaliśmy bibliotekę **scikit-learn**.



Rys. 15 - Logo biblioteki **scikit-learn**

W programie znaleźć można dwa różne rodzaje predykcji:

1. Przewidywanie ilości przypadków w wyznaczonym przez nas okresie, bazując na określonym zbiorze danych uczących

Ten rodzaj predykcji pozwala na wybranie kilku **parametrów**, które definiują sposób i dane do trenowania naszej sieci. Parametry do ręcznego ustalenia:

- **START_DATE** - data - definiuje dzień, od którego zaczynamy brać dane do trenowania sieci,
- **DAYS_TO_PREDICT** - int - ilość dni, które chcemy przewidzieć (domyślnie 7),

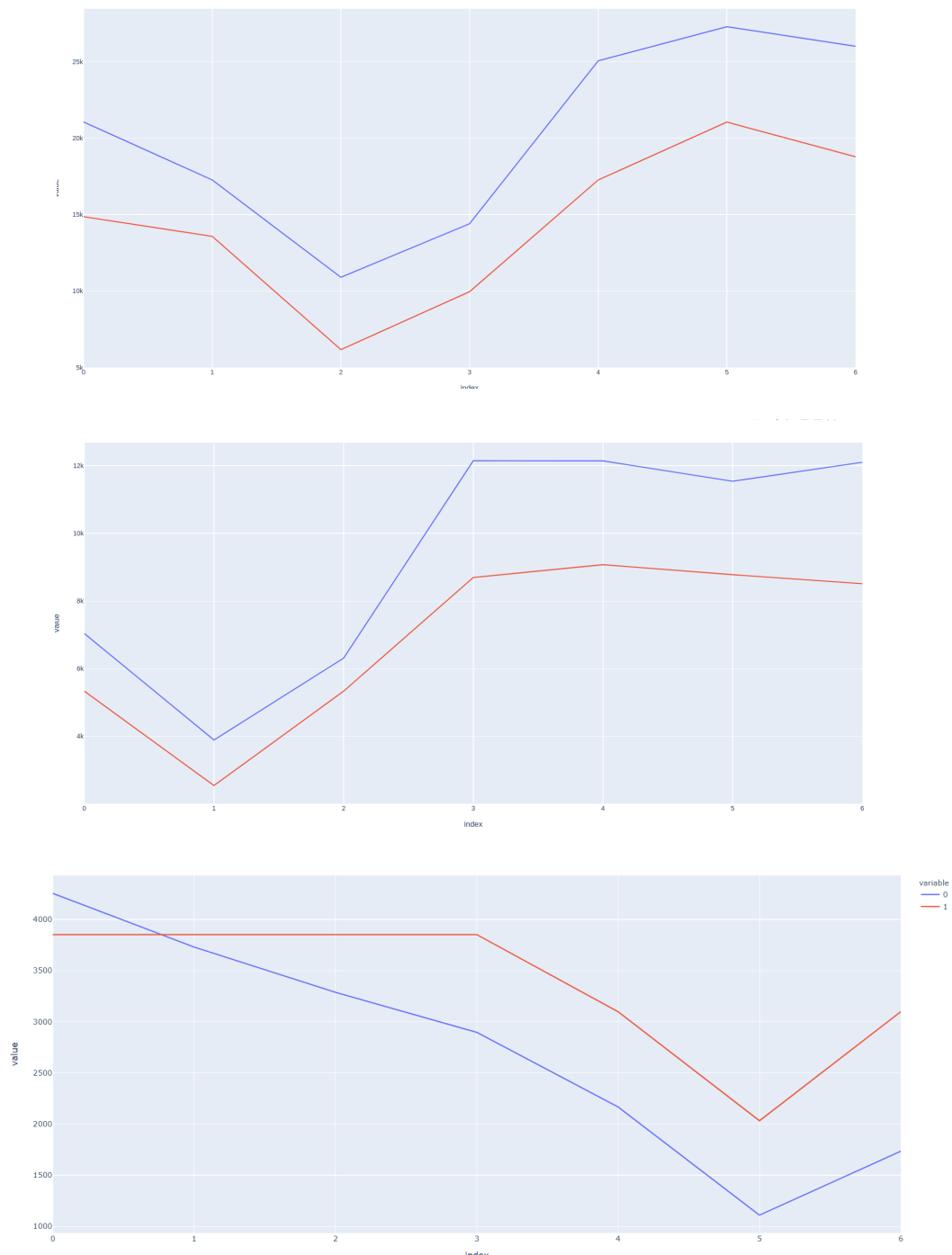
Ta predykcja bazując na wybranej przez nas ilości danych i wybranych przez selekcję na podstawie korelacji typów zmiennych, umożliwia nam ustalenie przybliżonych ilości dziennych zakażeń w {DAYS_TO_PREDICT} kolejnych dniach.

2. Przewidywanie tygodniowo średnich ilości dziennych przypadków zarażeń dla 3 kolejnych tygodni

Ten rodzaj również pozwala na zmianę parametrów, lecz te zmieniamy w pliku **prepareData.py**. Ta predykcja bazując na tygodniowych średnich z przeszłości pozwala nam ustalić średnie dla trzech kolejnych tygodni.

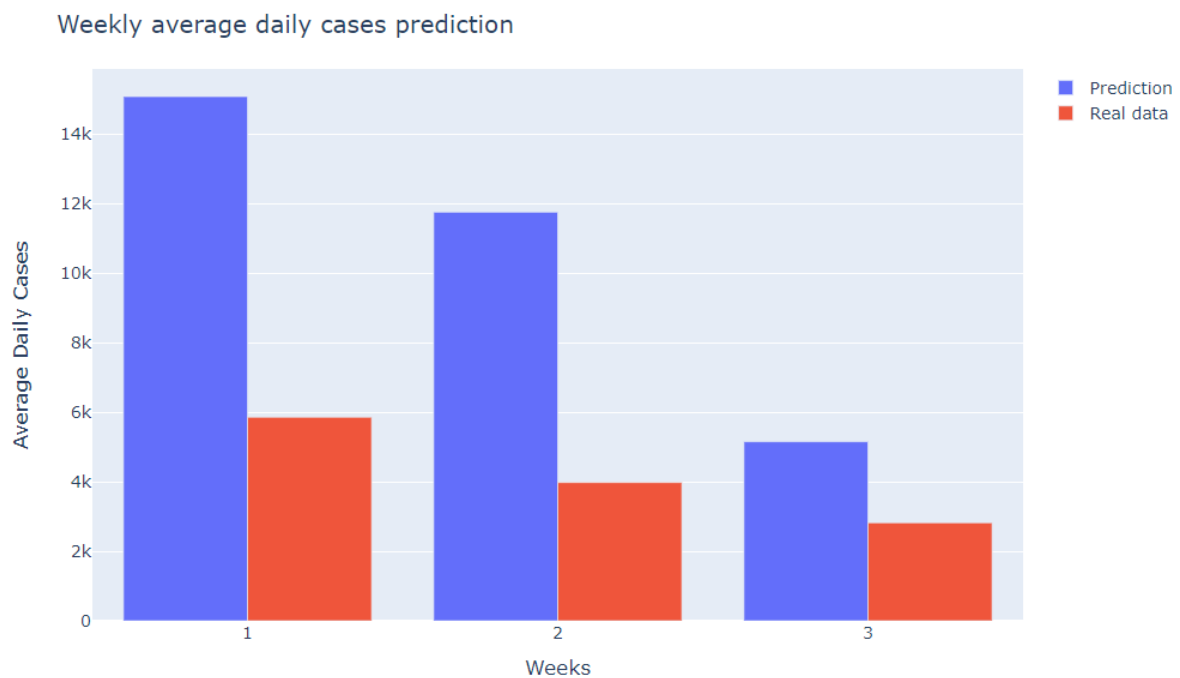
9. Przykładowe wyniki predykcji

Predykcja następnego tygodnia



Rys. 16 - Przykładowe wyniki **predykcji** **dziennej** **ilości** **zachorowań** następnego tygodnia

Predykcja średniej ilości dziennych zakażeń następnych tygodni



Rys. 17 - Przykładowy wynik **predykcji tygodniowej średniej ilości dziennych zachorowań** trzech kolejnych tygodni

10. Dokumentacja użytkownika

By skorzystać ze stworzonego przez nas narzędzia, wystarczy postępować zgodnie z instrukcją umieszczoną poniżej.

1. Pobierz repozytorium

Wszystkie pliki źródłowe znajdują się w repozytorium na platformie GitHub - <https://github.com/Equilibrium23/MIO-Project>

2. Pobierz najnowszą wersję Python'a

Wszystkie informacje i potrzebne pliki można znaleźć na oficjalnej stronie Python'a - <https://www.python.org/downloads/>

3. Pobierz wszystkie zależności z pliku requirements.txt

Korzystając z komendy poniżej pobierz wszystkie zewnętrzne biblioteki

```
python3 -m pip install -r requirements.txt
```

(komenda może się różnić w zależności od wersji Pythona)

4. Z poziomu folderu ze wszystkimi plikami, uruchom main.py

Plik **main.py** zawiera przedstawienie większości funkcji, które dostępne są w naszym programie. Wewnątrz znajdują się wypunktowane opcje, które można **odkomentować** i uruchomić ponownie plik aby uzyskać opisany wyżej w komentarzu efekt.

```
python3 main.py
```

(komenda może się różnić w zależności od wersji Pythona)