Przetwarzanie i Analiza Danych

Laboratorium 12: Prognozowanie wartości w szeregach czasowych na przykładzie danych dotyczących epidemii COVID-19

Cel ćwiczenia

Celem jest wykonanie prostej predykcji wzrostu zachorowań za pomocą modelu Gompertza, ponadto celem praktycznym jest dalsze zapoznawanie się z bibliotekami **pandas**, **numpy** oraz **scipy**, dalej zakładamy, że aliasem dla biblioteki **pandas** jest `pd`, natomiast dla pkgnumpy jest to `np`. Przed przystąpieniem do ćwiczenia należy zapoznać się z klasami `np.datetime64` oraz `np.timedelta64` oraz funkcją `scipy.optimize.curve_fit`.

Zadania

1. Wczytać dane dotyczące rozwoju epidemii COVID-19 (odpowiednio przypadki potwierdzone, śmiertelne, wyleczone) – Google: *github covid* (`https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv`, `https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv`, `https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_recovered_global.csv`), za pomocą `pd.read_csv(file_name)` do trzech tabel np.: `dc`,`dd`,`dr`.
2. Dla każdej z tabel ustawić indeks na czterch początkowych zmiennych za pomocą metody `set_index(columns_as_list)`.
3. Zwalcować każdą z tabel za pomocą metody `stack`, w wyniku powinno się uzyskać serię danych Tab. 1 posiadającą wielowymiarowy indeks. Proszę zwrócić uwagę, że data została włączona jako nienazwany poziom 4 do indeksu.
4. Złączyć uzyskane trzy serie danych w jedna tabelę (np. przekazując odpowiedni słownik (`{nazwa_kolumny1: dane1, ..., nazwa_kolumny3: dane3}`)) do konstruktora klasy `DataFrame`).

5. Usunąć indeks za pomocą metody `reset_index()`

6. Kolumna o indeksie 4 (piąta kolumna) zawiera datę, która jest rozpoznawana jak `string`, należy dodać nową kolumnę o nazwie `czas` zwierającą poprawnie sparsowaną datę (konwesji można dokonać za pomocą funkcji (pd.to_datetime(data_series))).

7. Utworzyć nową kolumnę `t`, która będzie przechowywała czas w dniach od ustalonej daty, np. od 1.03.2020 r. (Wystarczy np. policzyć różnicę dat `df['czas'] - pd.Timestamp('2020/03/01'), uwaga:` **pandas** `używa obiektów datetime64` (oraz `timedelta64`, które mają rozdzielczość nanosekund, przejście do skali dni wykonujemy za pomocą odpowiedniego dzielenia))

8. Wybrać dane dotyczące Polski (lub dowolnego innego kraju) np. za pomocą polecenia: `pol = dane[dane[''Country/Region'']=='Poland']`, wynik podobny do Tab. 2 i zwizualizować na dwóch podwykresach liczbę przypadków dziennie oraz za pomocą polecenia `diff` wyznaczyć i zwizualizować przyrosty dzienne.

9. Zdefiniować funkcję `gompertz(t, N0, b, c)` określoną następująco:

$$f(t) = N0\mathrm{e}^{-b\mathrm{e}^{-ct}}$$

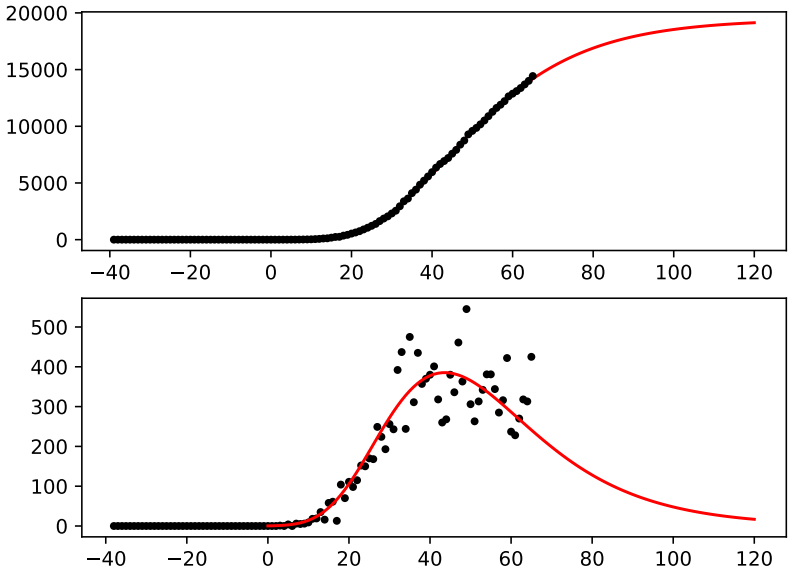
10. Za pomocą funkcji `curve_fit` dopasować do danych krzywą Gompertza (np.: `popt, pcov = curve_fit(gompertz, x, y, p0=[1.,1.,1.])`), a następnie zwizualizować krzywą skumulowaną oraz natężenie przypadków, wyniki powinny być zbliżone do tych, jak na rysunku 1). Poleceniem `diff` należy zamienić skumulowaną liczbę zachorowań na przyrosty dzienne. Przedstawić graficznie przyrosty dzienne oraz pochodną z dopasowanej krzywej Gompertza (jest to natężenie liczby przypadków). Pochodną wyznaczyć numerycznie np. za pomocą różnicy symetrycznej.

Tabela 1: Tabela etap 1.

				0
Province/State	Country/Region	Lat	Long	
Zhejiang	Afghanistan	33.000000	65.000000	1/23/20 0
				1/24/20 0
				1/25/20 0
				1/26/20 0
				1/27/20 0

Tabela 2: Tabela etap 2.

	Province/State	Country/Region	Lat	Long	level_4	c	d	r	czas	t
19216	NaN	Poland	51.9194	19.1451	1/23/20	0.0	0.0	0.0	2020-01-23	-38.0
19217	NaN	Poland	51.9194	19.1451	1/24/20	0.0	0.0	0.0	2020-01-24	-37.0
19218	NaN	Poland	51.9194	19.1451	1/25/20	0.0	0.0	0.0	2020-01-25	-36.0
19219	NaN	Poland	51.9194	19.1451	1/26/20	0.0	0.0	0.0	2020-01-26	-35.0



Rysunek 1: Przykładowe wyniki.

```
def diff_fun(fun, h=1e-7):
return lambda x : (fun(x + h) - fun(x - h)) / 2 / h

t = np.linspace(0,120,121)
ym = fun(t, *popt)
plt.subplot(2,1,1)
plt.plot(t, ym, "g-")
plt.plot(x,y, ".")
plt.subplot(2,1,2)
plt.plot(x[1:],np.diff(y),'k.')
plt.plot(t, diff_fun(lambda x: fun(x, *popt))(t), 'r-')
```

11. Z modelu odczytać prognozowaną za pomocą tego modelu łączną liczbę zachorowań na COVID-19 w Polsce, określić kiedy model przewiduje szczyt zachorowań oraz kiedy będzie prognozowany koniec epidemii, (np. czas do osiągniecie 99% oraz 50% wszystkich zachorowań), kiedy będzie połowa epidemii. Do rozwiązywania równań można użyć funkcji `brenth` pakietu `scipy.optimize` (Uwaga: Do interpretacji wyników tego modelu proszę podchodzić z dużą ostrożnością)

Opcja1 Powtórzyć obliczenia przepróbkowując wielokrotnie dane (np. 100 razy, `train_test_split`) i uśredniając wyniki (parametry modelu), wykreślić modele dla skrajnych wartości parametrów. Odpowiedzieć na pytania dot. oszacowania prognoz poprzedniego punktu.

Opcja2 Powtórzyć obliczenia budując model w sposób kroczący (uczymy na próbkach początkowych poniżej ustalonej chwili t, testujemy model na 5 kolejnych próbkach powyżej t), biorąc t co 5 kolejnych dni licząc od 20.03. Wyznaczyć i przedstawić graficznie jak zmienia się: 1) błąd testowania, 2) współczynniki modelu oraz 3) prognozowana liczba przypadków, w kolejnych chwilach t. Wypowiedzieć się na tej podstawie na temat stacjonarności zjawiska.

Opcja3 Dla każdego kraju (zmienna `Country/Region`) wyznaczyć: 1) czas rozpoczęcia epidemii (1% przypadków), 2) czas zakończenia (99% przypadków), 3) okres trwania epidemii (różnica pomiędzy 1 i 2), 4) prognozowaną liczbę przypadków. (Uwaga: Przed przystąpieniem do analizy dane należy zagregować po kraju (zsumwać), grupując (`groupby`, `sum`, `agg`), gdyż niektóre kraje mają podane dane do poziomu prowincji.). Dalej wykonać co następuje:

   (a) Dokonać grupowania tych danych na 6 grup za pomocą wybranego algorytmu grupowania. Zwizualizować skupienia w rzucie na dwie pierwsze składowe główne z pomocą przekształcenia PCA.

   (b) Sprawdzić, w której grupie są: Polska, Chiny, USA, Włochy, Australia, Iran, Brazylia, Szwecja.

   (c) W obrębie każdej z grup znaleźć obserwacje (państwa) odstające (punkty w największej odległości od centrum, wskazówka `np.argmax`)

12. Napisać podsumowanie i wnioski z eksperymentów.