Харківський національний університет імені В. Н. Каразіна
Факультет математики і інформатики
Кафедра прикладної математики

Кваліфікаційна робота

освітньо-кваліфікаційний рівень: **бакалавр**

на тему

«Математичне моделювання пандемії ковід-19 методами аналізу "великих даних"»

Виконав: студент групи МП41 IV курсу
(перший бакалаврський рівень)
спеціальності 113
«Прикладна математика»
освітньої програми
«Прикладна математика»
Фальченко Іван Русланович
Керівник: доктор фіз.-мат. наук ,
професор кафедри
прикладної математики
Кізілова Наталія Миколаївна
Рецензент: доцент закладу вищої освіти,
кафедри теоретичної та прикладної
інформатики,
кандидат технічних наук
Меняйлов Євген Сергійович

Харків – 2024 рік

Зміст:

1. Анотації

Фальченко І. Р. Математичне моделювання пандемії ковід-19 методами аналізу "великих даних"

В основі цієї роботи лежить кластеризація європейських країн на основі даних про захворюваність на COVID-19 з 2020 по 2022 роки, які містять кількість нових випадків на мільйон населення та загальну кількість випадків на мільйон населення. Основними методами кластеризації, які використовуються в дослідженні, є модифікації метода k-середніх на основі евклідової метрики та відстані Махаланобіса, які були розглянуті як швидша альтернатива до методу найближчого сусіда. Через особливості методу k-середніх, цей алгоритм було застосовано до 2 наборів початкових центроїдів: 4 початкових центроїдів, які були побудовані на основі надійних результатів кластеризації того ж самого набору даних за допомогою методу найближчого сусіда з дослідження [18] та 7 початкових центроїдів на основі географічних підгруп європейських країн. Метод найближчого сусіда був додатково застосований до кластерів, виявлених за допомогою кластеризації k-середніх, щоб дослідити внутрішню структуру цих кластерів. Результати підтвердили важливість географічної близькості країн, виділяючи при цьому незвичайні сполуки географічно віддалених країн, які неодноразово з’являлися в різних результатах кластеризації.

Falchenko I. R. Mathematical modeling of the covid-19 pandemic using big data analysis methods

This paper is dedicated to the clusterization of European countries based on their COVID-19 morbidity data from 2020 to 2022, which include the number of new cases per million population as well as the total number of cases per million population. The main methods of clusterization used in the research are the variations of k-means clustering based on Euclidean metrics and Mahalanobis distance, which were tested as a faster alternative to the nearest neighbor method. Due to the special features of k-means clustering this algorithm was applied to 2 sets of initial centroids: 4 initial centroids that were constructed based on the reliable results of clusterization of the same dataset using the nearest neighbor method from the research [18] and 7 initial centroids based on the geographical subgroups of European countries. The nearest neighbor method was additionally applied to the clusters revealed by k-means clustering in order to investigate the inner structure of these clusters. The results confirmed importance of geographic vicinity proximity while showing the unexpected combinations of geographically distant countries that repeatedly appeared in different clusterizations.

2. Вступ

В останні часи проблема пандемії COVID-19 нібито втрачає актуальність в суспільній думці, але попри зростання ролі інших глобальних проблем, це питання все ще залишається нагальним. Протягом 2023 року швидкість розповсюдження коронавірусу в деяких країнах Європи досягала більш ніж 500 нових випадків захворювання на день, і цей показник є вочевидь далеким від прийнятного мінімуму. Ця статистика демонструє, що фінальне рішення все ще не знайдене і наукова спільнота має продовжувати роботу в цьому напрямку. [41]

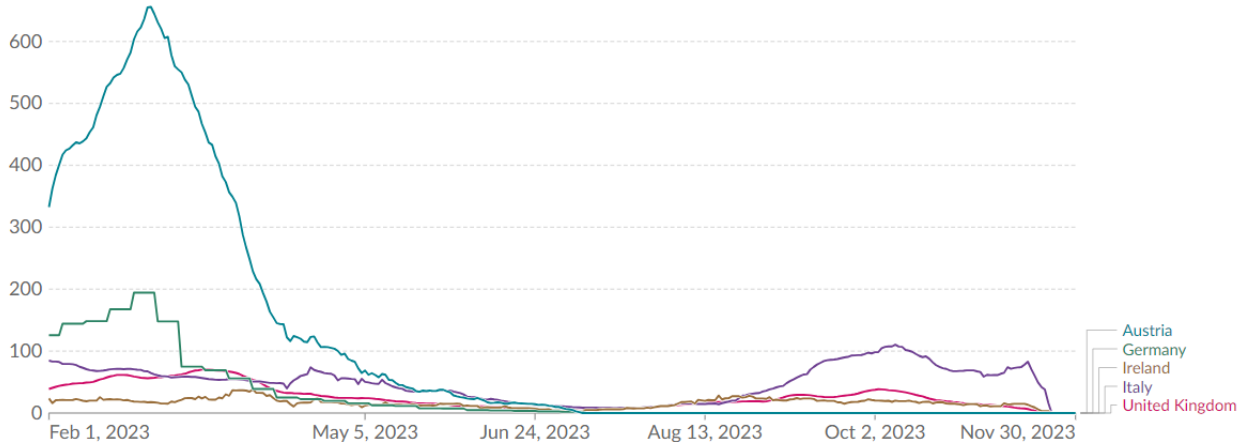

Рис. 1 Графіки нових випадків захворювання на COVID-19 на мільйон осіб для Австрії, Німеччини, Ірландії, Італії та Великобританії з 8 січня 2023 р. по сьогоднішній день. Дані взяті з джерела [32]

COVID-19 – це висококонтагіозна вірусна хвороба, спричинена коронавірусом гострого респіраторного синдрому SARS-CoV-2. Перші випадки цього захворювання були зареєстровані наприкінці 2019 року в місті Ухань провінції Хубей (Китай). Згодом, наприкінці грудня 2019 року, SARS-CoV-2 швидко поширився по всьому світу і, згідно з останніми оновленими статистичними даними, виміряними в 231 країні та країні, він спричинив понад 7

010 681 смертей і 675 619 811 людей одужало. У 2020-2021 роках були виявлені наступні нові варіанти вірусу [4]: Альфа у Великобританії (UK) наприкінці грудня 2020 року; Бета в Південній Африці в грудні 2020 року; Гамма в Бразилії на початку січня 2021 року; Дельта в Індії в грудні 2020 року; Лямбда в Перу в червні 2021 року; і Омікрон в Південній Африці в листопаді 2021 року. Крім того, ще кілька варіантів були класифіковані як Епсилон, Дзета, Ета, Тета, Йота, Каппа і Мю [4]. Ці спостереження демонструють, що географічний фактор має істотне значення у появі нових осередків хвороби та її поширенні навколо них. [41]

В рамках дослідження пандемії COVID-19 дуже важливу роль відіграє аналіз статистичних даних, таких як кількість нових випадків захворювання, нових смертей, нових тестів та щеплення, наявність лікарів та лікарень, коефіцієнт репродукції хвороби (він обчислюється як кількість вторинних випадків зараження однією інфікованою особиною) та деяких інших параметрів, що впливають на прийняття рішень щодо заходів протидії пандемії в різних країнах і регіонах [1,3]. В залежності від від типу аналізу можна отримати інформацію про ймовірні причини, впливові фактори, схожість чи відмінність ситуації в окремих групах населення та оцінити ефективність уже застосованих заходів [5]. Різні статистичні підходи на основі багатовимірних лінійних та нелінійних регресійних моделей, моделей нелінійної цілочисельної регресії, моделей глибоких нейронних мереж та інших моделей машинного навчання були перевірені на відкритих наборах даних (ВООЗ, статистика коронавірусу Worldometer, Google Health та деякі інші). Математичне моделювання динаміки пандемії [6-8], аналіз стабільності та ефективне керування [9,10] разом з розробкою нових вакцин [11] є важливими для подальшого розвитку математичної епідеміології [12] як галузі міждисциплінарних досліджень. [40, 41]

У цьому дослідженні кластерний аналіз, що являє собою швидкий та надійний математичний інструмент для вирішення задач розпізнавання та

класифікації, був застосований до наборів даних про випадки захворюваності на COVID-19 у 2020, 2021 та 2022 роках. Кластерний аналіз, як частина цього розвиненого наукового апарату, досліджує саме ступінь подібності даних і дозволяє побудувати класифікацію в рамках різноманітних соціальних груп (наприклад, населення різних країн або статей). Результати кластерного аналізу надають нам системне уявлення про ситуацію та дозволяють робити обґрунтовані припущення щодо впливу інших показників на динаміку захворюваності при одночасному порівнянні схожості за статистичними даними та схожості за значеннями цих показників. [40, 41]

В рамках кластерного аналізу даних про коронавірус природнім чином встає питання вибору між різними методами. Кожного разу виникає потреба знайти метод, що є оптимальним за швидкодією, точністю та наочністю результатів для конкретної задачі. Одним з найпростіших засобів порівняння різних методів є порівняння їх роботи на однакових даних. В цьому дослідженні були порівняні результати роботи методу найближчого сусіда та методу k-середніх завдяки застосуванню цих методів до даних про захворюваність на COVID-19 в країнах Європи. Задля дослідження особливостей роботи методу k-середніх він також був застосований до кумулятивних даних з джерела [32] (тобто даних про загальну кількість хворих на COVID-19 в країнах Європи) та була розглянута модифікація цього метода з відстанню Махаланобіса. [40, 41]

3. Аналіз літератури

Загальна описова статистика [5,13], кореляція динаміки захворюваності на covid-19 із соціально-економічними факторами [6], вплив віку та статі [7] та часова динаміка передачі COVID-19 [14,15] з 2020 по 2023 рік були проаналізовані за допомогою різних статистичних методів. Також були

опубліковані локальні дослідження для населення Південно-Східної Азії та Індії [3], Бразилії [16], України [17], Європи [18,19], Польщі [10], Перу [20], Індонезії [21], Китаю [22] , США [23] та Канади [24]. Статистичний аналіз і математичні моделі виявились дуже ефективними в прогнозуванні динаміки пандемії у різних країнах [22,23], соціально-економічних факторів [6,25] і урядових заходів проти COVID-19 [26]. Найбільш перспективні результати були отримані на основі світових даних із більш ніж 180 країн [25,27,28]. Підхід до багатовимірного аналізу даних [29], імовірнісні та стохастичні методи [16,30] і глибоке машинне навчання [5] визнано найбільш перспективними методами аналізу великих даних пандемії та ефективної ідентифікації кластерів COVID-19 у світі [28]. Серед усього різноманіття статистичних методів кластерний аналіз є важливим засобом рішення задач класифікації, особливо в випадках, коли щільність ймовірності для вихідних даних (що зазвичай подані в вигляді часових рядів $\left\{X(t_j)\right\}_{j=1}^{N}$, де N – це кількість спостережень, тобто днів) значно відрізняється від нормальному розподілу і тому кореляційний аналіз може дати помилкові результати в задачах розпізнавання та класифікації [31]. [41]

Застосування методу k-середніх в цій роботі буде спиратися на результати дослідження «Аналіз «великих даних» і математичне моделювання епідемії covid-19 в країнах Європи» (джерело [18]), а саме на дерево кластеризації, що було отримано за допомогою методу найближчого сусіда. Як буде показано далі, це рішення є цілком природнім, оскільки метод k-середніх вимагає задання початкових центроїдів, які було б доречно визначити за допомогою вже існуючих результатів кластеризації обраних даних.

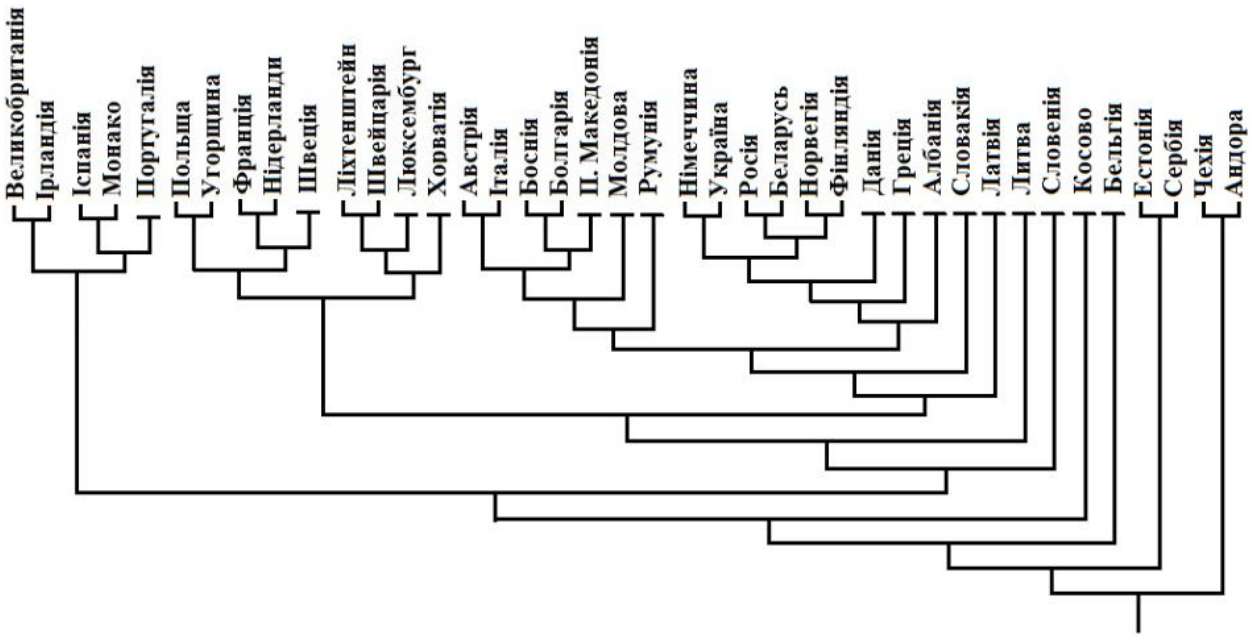

Рис. 2 Дендрограма, що ілюструє результати кластеризації даних про захворюваність в країнах Європи за допомогою методу найближчого сусіда (взято з джерела [18])

У дослідженні [18] кластери для динаміки COVID-19 з березня 2020 року до жовтня 2021 року у 40 європейських країнах були визначені методом найближчого сусіда та візуалізовані у вигляді дендрограм, що показують результати ієрархічної кластеризації (Рис.2). Було помічено, що багато країн з найближчих кластерів мають спільні географічні кордони, наприклад Великобританія та Ірландія, Іспанія та Португалія, Австрія та Італія, Росія та Білорусь, Норвегія та Фінляндія (рис.2). Однак деякі з найближчих сусідів в динаміці COVID-19 знаходяться далеко один від одного на мапі. Цю динаміку можна пояснити спільними заходами уряду проти COVID-19 або іншими соціально-економічними факторами у різних країнах. Також важливо відзначити, що застосування методу найближчого сусіда не є доцільним для великої кількості країн (більш ніж 200), кожній з яких відповідають досить довгі часові ряди

(зазвичай з кількохсот днів спостереження), оскільки алгоритм має досить велику обчислювальну складність (щонайменше $O(n^2)$ в залежності від реалізації, де n – кількість країн). [41]

Неоднозначність результатів разом з занадто малою швидкістю кластеризації можна виправити, якщо застосувати до тих самих даних інший алгоритм кластеризації, наприклад метод k-середніх. Метод k-середніх вже був успішно застосований для класифікації випадків COVID-19 на основі неповних гетерогенних наборів даних [32,33] разом з поточними параметрами атмосфери [34], а також як метод виявлення COVID-19 за послідовностями генома людини [35]. Окрім цього, метод k-середніх продемонстрував свою ефективність в рамках класифікації за допомогою моделей машинного навчання [33]. [41]

На дендрограмі з дослідження [18] можна виділити 4 кластери, за якими буде розрахований один з наборів початкових центроїдів для методу k-середніх.

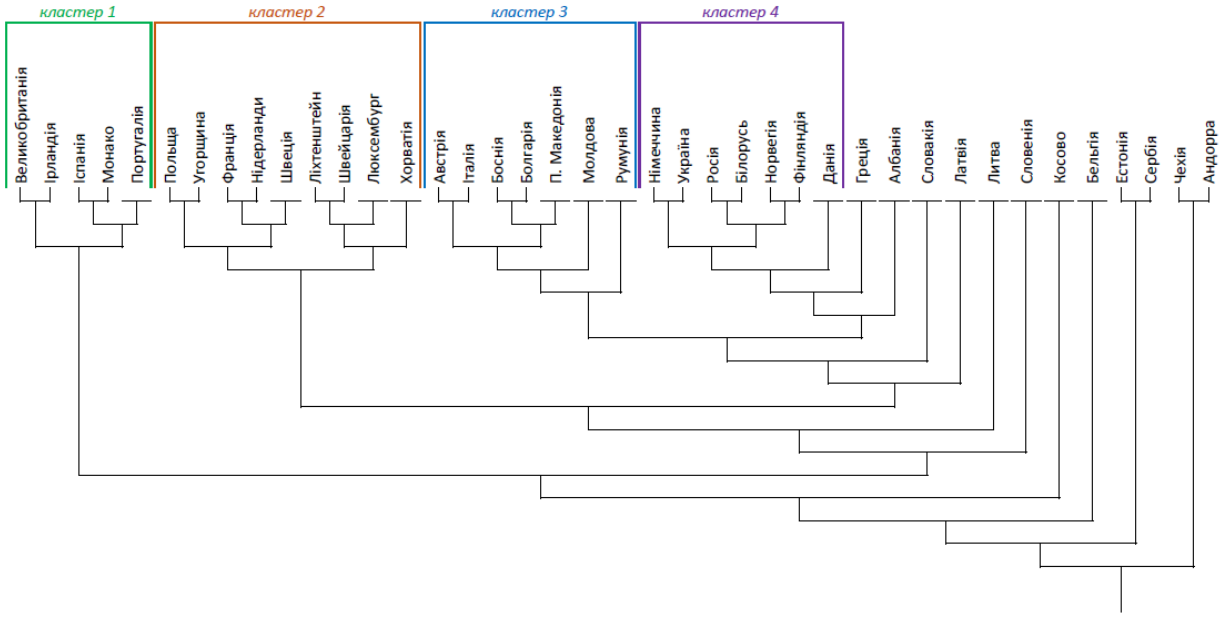

Рис. 3 Дендрограма з рис. 2 з виділеними кластерами

Треба також відзначити, що існуюча наукова практика показує ефективність і точність відстані Махаланобіса в багатьох задачах класифікації, кластеризації та виявлення викидів (включаючи кластеризацію даних COVID-19), оскільки вона враховує кореляції набору даних і тому її використання призводить до більш коректних результатів, особливо коли характеристики в наборі даних корельовані або коли набір даних в цілому є зміщеним та дуже нерівномірним. [33]. Ці переваги відстані Махаланобіса будуть використані в подальшому задля того, щоб скорегувати попередні результати кластеризації, які були отримані при застосуванні евклідової метрики. [41]

4. Матеріали та методи

На першому етапі були використані статистичні дані про кількість нових випадків захворюваності на COVID-19 на мільйон населення в країнах Європи з 14 березня 2020 р. до 15 жовтня 2021 р. (ці дані були взяті з джерела [32] та використані в дослідженні [18] для побудови дендрограми на рис. 2), що подані у вигляді часових рядів (часовий ряд – це послідовність спостережень, які впорядковані за часом у вигляді масиву точок або неперервної функції). Для роботи з ними було використане програмне забезпечення Excel.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	date	Europe	Albania	Andorra	Austria	Belarus	Belgium	Bosnia	Bulgaria	Croatia	Czech	Denmark	Estonia	Finland	France	Germany	Greece	Hungary	Ireland	Italy
2	2020-03-14	12,683545	2,0260158	68,832241	20,874806	0,4671075	14,40458	1,7529678	1,4445959	2,7028571	5,5343915	16,842364	17,95212	6,5793623	13,659773	16,238063	3,5498701	0,697436	9,2778613	51,949037
3	2020-03-15	14,562239	2,1707159	90,040798	25,023577	0,5127493	17,152858	2,1563037	1,674432	3,4472449	6,6027359	16,854137	19,188853	7,2879988	15,787485	19,317181	3,8267868	0,8380524	11,627659	57,190917
4	2020-03-16	16,595285	2,3366447	114,05455	29,709643	0,5504952	20,322023	2,611096	1,9147762	4,337551	7,7575214	16,769903	20,051269	8,0130029	18,12648	22,623187	4,0825327	1,0038871	14,299259	62,412185
5	2020-03-17	18,776234	2,526928	140,24892	34,896988	0,5772153	23,974733	3,1096885	2,1510729	5,3686735	8,9884169	16,705293	20,65314	8,7629288	20,652557	26,123034	4,3316052	1,197117	17,26675	67,487364
6	2020-03-18	21,088072	2,7656076	167,80013	40,462363	0,5938363	28,194304	3,6467218	2,37645	6,5233673	10,284703	16,782175	21,150983	9,5452799	23,358379	29,765826	4,5793453	1,4167143	20,475049	72,273355
7	2020-03-19	23,511321	3,0600915	196,44434	46,236878	0,6007551	33,055995	4,208514	2,5855294	7,7691837	11,6444	17,091324	21,724885	10,361705	26,257279	33,500799	4,838404	1,6630262	23,850854	76,632379
8	2020-03-20	26,013252	3,4054985	225,47621	52,095261	0,6021624	38,6278	4,7893211	2,7757143	9,0668367	13,041101	17,688121	22,578822	11,190584	29,359495	37,303984	5,1189025	1,9325523	27,29194	80,430237
9	2020-03-21	28,558209	3,7847226	254,07196	57,779778	0,6079384	44,908714	5,4025439	2,9430591	10,375	14,473177	18,599363	23,790423	12,021558	32,660667	41,141289	5,425621	2,2222241	30,777327	83,546205
10	2020-03-22	31,095335	4,1764519	281,25165	62,980148	0,6307405	51,864686	6,0628596	3,0873344	11,63102	15,918377	19,823331	25,376347	12,820547	36,128197	44,976639	5,7587218	2,5259479	34,268985	85,910994
11	2020-03-23	33,554452	4,5576043	305,71767	67,375817	0,6890037	59,359584	6,7806068	3,2081175	12,793469	17,344898	21,360384	27,287877	13,547544	39,659891	48,715962	6,1078867	2,8354248	37,696366	87,507125
12	2020-03-24	35,862628	4,9106368	326,60855	70,739037	0,8065439	67,157421	7,5589758	3,3095752	13,828673	18,71616	23,205222	29,419097	14,185289	43,179802	52,254222	6,4572153	3,1405381	41,015422	88,344302
13	2020-03-25	37,954964	5,2220641	343,04127	72,919111	1,002768	74,960407	8,3878842	3,3986526	14,709694	19,996163	25,332555	31,628576	14,725525	46,594102	55,503349	6,7997222	3,433117	44,211023	88,466714
14	2020-03-26	39,755825	5,4830387	354,31588	73,845621	1,3043165	82,482216	9,2649084	3,4768596	15,422245	21,130772	27,712435	33,757596	15,179066	52,499619	58,377842	7,1183411	3,7036501	47,273559	87,954387
15	2020-03-27	41,201595	5,6958771	360,30853	73,39386	1,7436085	89,419903	10,174606	3,5439546	15,949796	22,0982	30,315419	35,641781	15,59831	63,481443	60,770209	7,3960858	3,9491978	50,248582	86,897368
16	2020-03-28	42,245673	5,8768142	361,04077	71,675239	2,3468438	95,563477	11,069883	3,6002395	16,285408	22,859805	33,098017	37,26573	16,038806	79,370769	62,598199	7,6152062	4,1666043	53,12832	85,414084
17	2020-03-29	42,855104	6,0358526	357,30947	68,890843	3,1533669	100,72754	11,907858	3,6437818	16,45102	23,409801	35,980421	38,570041	16,568865	99,984367	63,797495	7,7582324	4,363521	56,003129	83,626671
18	2020-03-30	43,062983	6,1754573	350,37455	65,303314	4,1963923	104,85936	12,843095	3,6760308	16,465612	23,745778	38,852797	39,526113	17,226917	125,28281	64,380223	7,8196606	4,5517926	59,039426	81,623516
19	2020-03-31	42,945497	6,2898301	341,24285	61,14298	5,5031358	108,07283	13,256577	3,6927593	16,347857	23,880083	41,595355	40,090576	18,046441	155,5584	64,377332	7,7972795	4,7501816	62,39385	79,525421
20	2020-04-01	42,588073	6,3691145	331,18506	56,671581	7,1063694	110,54028	13,752134	3,6873844	16,128265	23,817805	44,078277	40,259344	19,023605	191,14851	63,83317	7,6889983	4,9818159	66,260315	77,409413
21	2020-04-02	42,118573	6,4080962	320,83654	52,117095	9,0264273	112,37485	14,118152	3,6595439	15,83898	23,586555	46,160347	40,031159	20,12763	224,27495	62,791756	7,5004769	5,2682245	70,880231	75,342404
22	2020-04-03	41,66901	6,395616	310,43417	47,761758	11,269881	113,77285	14,352336	3,61111	15,526939	23,181051	47,690826	39,398481	21,267008	247,18083	61,311845	7,2429575	5,6225056	76,519759	73,387039
23	2020-04-04	41,373066	6,3189188	300,33332	43,727664	13,827511	114,87106	14,484293	3,5486047	15,228265	22,612905	48,549708	38,320137	22,338521	260,56061	59,459489	6,9365589	6,056374	83,440906	71,574835
24	2020-04-05	41,370888	6,1816302	290,17325	40,07372	16,63995	115,78445	14,542741	3,4823553	14,952347	21,87761	48,695794	36,851444	23,221239	265,19258	57,31427	6,6009175	6,553661	91,60462	69,9204
25	2020-04-06	41,729589	5,9985381	279,69551	36,793616	19,651311	116,58377	14,564563	3,4178571	14,702245	20,997887	48,146495	35,072861	23,855708	261,7387	54,934245	6,2454706	7,0881254	100,8266	68,412164
26	2020-04-07	42,398746	5,7964631	268,59858	33,873768	22,798386	117,20138	14,550652	3,366162	14,474388	19,979793	46,953969	33,136504	24,175496	249,59781	52,374017	5,877808	7,6240146	110,85951	67,019701
27	2020-04-08	43,318711	5,6029492	256,53786	31,241173	26,004472	117,58924	14,484288	3,3422447	14,245102	18,855484	45,222888	31,151432	24,177601	228,11817	49,675966	5,4996168	8,1178821	121,26311	65,711392
28	2020-04-09	44,372964	5,4408984	243,5726	28,820576	29,191849	117,84806	14,375681	3,3586131	13,983571	17,658084	43,112664	29,225448	23,887398	204,51625	46,912506	5,11893	8,5338896	131,43284	64,442589
29	2020-04-10	45,439135	5,3346631	229,9343	26,518801	32,281081	117,87386	14,236571	3,4241983	13,652449	16,456021	40,830207	27,470754	23,39489	186,00022	44,169254	4,7399654	8,8438513	140,59162	63,181065
30	2020-04-11	46,362017	5,3003332	215,84372	24,28772	35,227706	117,60585	14,060831	3,5414786	13,222143	15,302187	38,567956	25,951464	22,799467	171,1057	41,526554	4,3659767	9,0251587	147,9362	61,896369
31	2020-04-12	46,963952	5,3361678	201,64546	22,108081	38,042003	117,04213	13,83468	3,7112391	12,691939	14,250329	36,457451	24,682666	22,223786	158,29148	39,027454	4,0039534	9,0924669	153,11777	60,560279
32	2020-04-13	47,103403	5,4397005	187,18875	20,00047	40,72803	116,18666	13,539227	3,9346876	12,066429	13,325963	34,594052	23,632928	21,729402	146,015	36,701341	3,6708038	9,0716277	155,89361	59,167423

Рис. 4 Частина вихідних статистичних даних з джерела [32]

До вихідних даних можна також віднести дендрограму на рис. 2, оскільки на її основі ми будемо задавати один з наборів початкових центроїдів для методу k-середніх.

Тепер перейдемо до основної складової цієї роботи, а саме кластерного аналізу. Головне призначення кластерного аналізу - розбивка безлічі досліджуваних об'єктів і ознак на однорідні у відповідному розумінні групи або кластер, що є задачею класифікації даних і виявлення в них структури в ній.

Задача кластерного аналізу полягає в тому, щоби різнорідні дані, які містяться в множині Q, розбити на ціле число m підмножин (кластерів) $Q_1, \dots, Q_m$ так, щоб кожен об'єкт Gj належав до однієї і тільки однієї підмножині розбиття, і щоб об'єкти, які належать одному і тому ж кластеру, були подібними, в той час як об'єкти, які належать різним кластерам, були різнорідними. [40]

Загальна схема кластерного аналізу на першому етапі має наступний вигляд:

I. Проводимо кластеризацію методом k-середніх. В якості результату отримуємо набір кластерів та їх центри. Для подальшого аналізу обчислюємо радіуси кластерів та відстані від кожного елемента до центра того кластера, якому він належить.

II. Оскільки результати застосування методу k-середніх та методу найближчого сусіда мають різний вигляд, треба застосувати деякі перетворення, так щоб подати кластери, що були отримані методом k-середніх, у вигляді дендрограми. Для цього визначимо нову відстань між елементами, спираючись на відстані елементів до центру відповідного кластера, та застосуємо метод найближчого сусіда з цією метрикою всередині кожного кластера. Таким чином

отримуємо окремі дерева для кожного кластера, які можна буде порівняти з кластерами на дендрограмі на рис. 3.

Розглянемо основні характеристики та особливості методів кластеризації, що будуть застосовані у подальшому:

1) Метод k-середніх:

Оскільки метод k-середніх, як і багато інших методів кластеризації, має багато модифікацій, сконцентруємося перш за все на тому варіанті, що реалізований в цій роботі.

1.1) Постановка задачі:

Розглядаємо n спостережень $\bar{X} = (x_1, \dots, x_n)$, кожне з яких характеризується m ознаками. Серед інших алгоритмів кластеризації метод k-середніх виділяється тим, що ми наперед задаємо кількість кластерів k, на які буде розбита множина спостережень.

1.2) Алгоритм:

1.2.1) Задаємо k центроїдів – точок, що будуть тимчасовими центрами кластерів. Можна їх обрати випадковим чином або взяти конкретні елементи з множини спостережень, але зазвичай цей вибір робиться свідомо з урахуванням специфіки конкретної задачі, оскільки початковий вибір центроїдів має істотній вплив на роботу методу.

1.2.2) Перебираємо усі точки та знаходимо відстані до кожного з центроїдів. Для цього можна використовувати багато різних метрик, але на першому етапі використовується евклідова метрика $\rho(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$. Точка приєднується до того кластера чий центроїд є найближчим. Ця ітерація завершується на тому етапі, коли усі точки розподілені по кластерам.

1.2.3) Обираємо нові центроїди. Це також можна робити різними засобами, але в цій реалізації новий центроїд будемо обчислювати наступним чином:

$c_i = \frac{1}{m_i} \cdot \sum_{x \in C_i} x$, де C_i – кластер; c_i – його новий центроїд; m_i – кількість точок, що були приєднані до цього кластера. Пізніше буде відзначено, що такий вибір засобу перерахунку центроїдів є оптимальним для евклідової метрики.

1.2.4) Виконуємо ті ж самі дії, що і на кроці 2, для нового набору центроїдів. Алгоритм зупиняється коли нове розбиття на кластери не відрізняється від того, що було отримано на попередній ітерації.

1.3) Обчислювальна складність:

1.3.1) Часова складність:

Найбільш корисною на практиці є оцінка, коли ми наперед обмежуємо кількість ітерацій алгоритму. Цей підхід справджується, оскільки необхідна для збіжності кількість ітерацій зазвичай невелика (щонайменше в порівнянні з об'ємом даних). Якщо вважати за елементарні операції арифметичні дії, то часова складність методу k-середніх – це $O(I \cdot k \cdot m \cdot n)$, де I – наперед задана кількість ітерацій, k – кількість кластерів, n – кількість спостережень, m – кількість ознак кожного спостереження [35]. В рамках цієї задачі доцільно розглядати обчислення відстані між елементами як елементарну операцію, і тоді часова складність дорівнює $O(I \cdot k \cdot n)$. Бачимо, що ця складність є лінійною відносно кількості спостережень, що разом з малою кількістю ітерацій для збіжності та простотою реалізації робить метод k-середніх швидким та гнучким для великого об'єму даних у порівнянні з багатьма іншими алгоритмами (навіть у тих випадках, коли збільшення об'єму даних зменшує ефективність цього методу, цю проблему можна вирішити за допомогою різноманітних модифікацій).

1.3.2) Просторова складність:

Метод k-середніх є також досить економним з точки зору пам’яті, бо окрім початкових даних для кластеризації зберігається лише k центроїдів, тому просторова складність буде $O\big((n+k)\cdot m\big)$, де k – кількість кластерів, n – кількість спостережень, m – кількість ознак кожного спостереження. [35]

1.4) Збіжність відносно цільової функції

Важливою частиною методу кластеризації є вибір цільової функції, що допомагає оцінити оптимальність деякого розбиття на кластери. Для методу k-середніх існує декілька комбінацій метрики, правила перерахування центроїдів та цільової функції, що гарантують збіжність відносно цієї цільової функції. Однією з таких комбінацій є евклідова метрика $\rho(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$, правило перерахування центроїдів $c_i=\frac{1}{m_i}\cdot\sum_{x\in C_i}x$ та цільова функція SSE (sum of the squared error) [35], яка для метрики ρ записується наступним чином:

$$SSE=\sum_{i=1}^{k}\sum_{x\in C_i}\rho(x,c_i)^2$$

Треба зауважити, що навіть в цьому випадку метод k-середніх гарантує збіжність лише до локального мінімуму цільової функції, оскільки невдалий вибір початкових центроїдів може значно погіршити якість кластеризації. [35, 36]

1.5) Переваги та недоліки:

Підіб’ємо підсумки та розглянемо головні позитивні та негативні особливості методу k-середніх.

Переваги:

- Гнучкість та простота реалізації [36]
- Швидка збіжність [36]

- Мала просторова складність
- Поширеність та велика кількість модифікацій

Недоліки:

- Необхідність наперед задавати кількість кластерів та початкові центроїди
- Ефективність методу дуже сильно залежить від вдалості вибору початкових центроїдів [36]
- Можна гарантувати збіжність лише до локального мінімуму цільової функції [35, 36]

Метод k-середніх був реалізований за допомогою програми в Python (вона наведена у додатку З).

2) Метод найближчого сусіда

Оскільки метод найближчого сусіда не є основним в цій роботі, ми розглянемо його лише на базовому рівні. Перш за все, нам треба розуміти його алгоритм та мати змогу порівняти його з методом k-середніх.

2.1) Постановка задачі:

Розглядаємо n спостережень $\bar{X} = (x_1, \dots, x_n)$, кожне з яких характеризується m ознаками. На відміну від методу k-середніх, кількість кластерів не є наперед заданою. До того ж метод сусіду є ієрархічним агломеративним методом кластеризації (тобто ми починаємо з того, що кожне спостереження є окремим кластером, і потім крок за кроком з'єднуємо їх у кластери, просуваючись вгору за ієрархією), тому як результат ми маємо отримати ієрархічну структуру, яку зручно ілюструвати за допомогою дендрограми.

2.2) Алгоритм:

На кожному кроці обираються два найближчих спостереження (відстань між спостереженнями визначається за допомогою метрики, в даній реалізації це евклідова метрика $p(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$) і «зливаються» в одне (два часових ряди замінюються на один ряд, елементи якого є середніми арифметичним відповідних елементів цих 2 рядів). Кожне таки злиття фіксується задля подальшої побудови дендрограми. Повторюємо ці дії доки не залишається один часовий ряд. На основі послідовності «злиття» часових рядів та відстаней між ними будуємо дендрограму, що наочно демонструє міру подібності між елементами.

2.3) Обчислювальна складність:

2.3.1) Часова складність:

В цьому дослідженні використана так званий «наївний алгоритм» для методу найближчого сусіда, що має часову складність $O(n^3)$, де n – кількість спостережень [37]. Існує декілька модифікацій, що можуть пришвидшити цей алгоритм, але найменша можлива часова складність для методу найближчого сусіда дорівнює $O(n^2)$ [38]. Можемо побачити, що при зростанні n метод найближчого сусіда буде працювати повільніше, ніж метод k-середніх, тому він гірше підходить для великих даних.

2.3.2) Просторова складність:

Оскільки в цій конкретній реалізації ми на кожному кроці будуємо матрицю відстаней між кластерами, просторова складність буде також більшою ніж у методу k-середніх.

Метод найближчого сусіда був реалізований за допомогою програми у Python (вона наведена у додатку 3).

Незважаючи на такі переваги евклідової метрики як простота та гарантована збіжність за цільової функцією SSE (sum of the squared error), вона також має один істотній недолік, що може негативно впливати на коректність кластеризації: вона не враховує взаємні кореляції між величинами у багатомірному векторі (в нашій задачі ми іноді будемо розглядати часові ряди як вектори з деякого багатовимірного розподілу). Як вже було відзначено, цю проблему можна подолати за допомогою переходу від евклідової метрики до відстані Махаланобіса. Тому на другому етапі ми будемо спиратися на відстань Махаланобіса при застосуванні методу k-середніх, щоб провести порівняння з результатами, які були отримані на першому етапі, і зробити більш точні висновки щодо закономірностей, які спостерігаються в даних про захворюваність на COVID-19.

Відстань Махаланобіса обчислюється за наступною формулою: $d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}$, де S – це коваріаційна матриця багатовимірного розподілу, до якого належать вектори $\vec{x}$ і $\vec{y}$. Звідси вже можна побачити, що застосування відстані Махаланобіса до двох векторів спирається на припущення, що ці вектори взяті з одного розподілу ймовірностей [39]. В рамках задачі про дослідження статистики COVID-19 у різних країнах це припущення можна сформулювати так, що пандемія відбувається в усіх країнах за деяким єдиним спільним законом. [41]

Перед застосуванням відстані Махаланобіса слід розглянути деякі особливості нашої задачі. По-перше, було б недоцільно обчислювати відстань Махаланобіса безпосередньо часових рядів, тому що в цьому випадку нам довелося б обчислити коваріаційну матрицю N × N, а через те, що число N у нашій задачі дуже велике, обчислювальна складність також стає надто великою. Щоб уникнути цього, замість часових рядів будуть розглядатися їх наступні статистичні показники: середнє арифметичне часового ряду, мінімальне та максимальне значення в часовому ряді та коефіцієнт асиметрії. Коефіцієнт

асиметрії являє собою міру асиметрії ймовірнісного розподілу дійсної випадкової величини відносно її середнього значення та обчислюється за формулою $\Gamma_1 = \mu_3 / \sigma^3$, де μ_3 — це центральний момент третього порядку, а σ — це середнє квадратичне відхилення. [41]

Зв'язок між значенням коефіцієнта асиметрії та графіком функції щільності демонструється на наступному графіку:

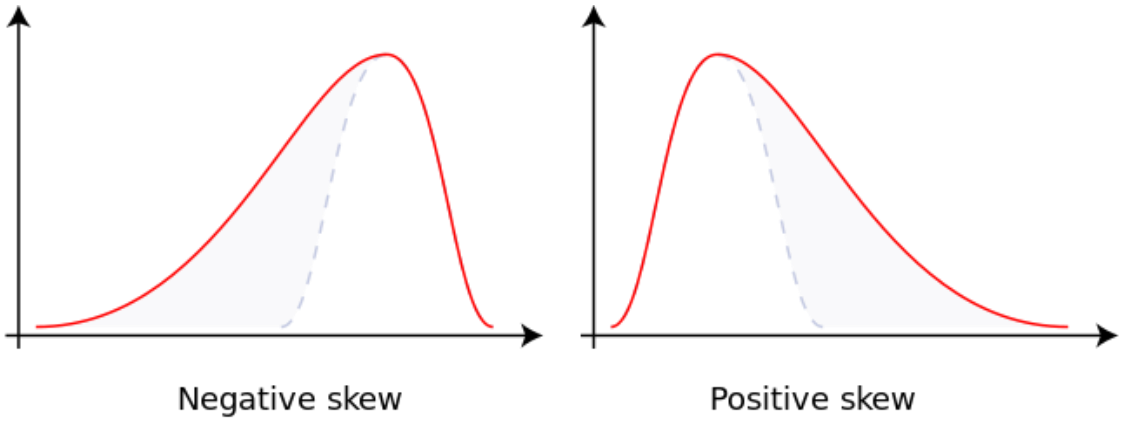

Рис. 5 Графік функції щільності в залежності від значення коефіцієнта асиметрії

Задля урівноваження впливу різних показників на значення відстані, деякі з них були пронормовані. Фінальний набір даних, до якого буде застосований метод k-середніх з відстанню Махаланобіса, виглядає наступним чином:

	Europe	Albania	Andorra	Austria	Belarus	Belgium	Bosnia	Bulgaria	Croatia	Czech	Denmark	Estonia	Finland	France
срзнач	0,4181124	0,3114335	1	0,4309021	0,3038822	0,5556092	0,3754154	0,392133	0,5270824	0,8042901	0,3218038	0,6486527	0,1349642	0,5780258
макс	0,2690729	0,2739007	0,9321128	0,5645353	0,1569713	1	0,3536472	0,3835695	0,6267098	0,8264677	0,4185573	0,8021729	0,0990633	0,5572296
мин	3,9165059	0,3414478	0,0016626	0,9778906	0,1442364	2,3727871	0,5412926	0,4460715	0,0192519	1,4090995	1,0043828	0,2079108	0,2981796	2,063287
скос	0,2674092	0,6618773	0,8174951	1,3627679	0,0262101	2,5409319	0,9422859	0,8769077	1,1504929	0,8792718	1,6419017	1,1854094	0,4861813	0,738863

Рис. 6 Новий набір даних, до якого буде застосований метод k-середніх з відстанню Махаланобіса

Вплив переходу до іншого набору даних на результати кластеризації буде детально проаналізовано в подальшому.

Оскільки в нашій задачі коваріаційна матриця загального ймовірнісного розподілу S невідома, замість неї була використана вибіркова коваріаційна матриця (в якості вибірки був використаний весь новий набір даних). Вибіркова коваріаційна матриця $S = [s_{jk}]$ була обчислена за допомогою функції numpy.cov() за наступною формулою:

$s_{jk} = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \overline{x_j})(x_{ik} - \overline{x_k})$, де N – це розмір відповідної вибірки, $\overline{x_j}$ та $\overline{x_k}$ – відповідні вибіркові середні

Фінальним кроком в цій роботі є застосування методу k-середніх з евклідовою метрикою та з відстанню Махаланобіса до кумулятивних даних з джерела [32] (тобто даних про загальну кількість хворих на COVID-19 в країнах Європи). Загальна ідея кластерного аналізу та ж, що і на попередніх етапах: спочатку застосовується метод k-середніх з евклідовою метрикою, потім створюється новий набір даних с чотирма статистичними показниками і вже до нього застосовується метод k-середніх з відстанню Махаланобіса, наприкінці порівнюються результати цих двох засобів кластеризації.

5. Результати

На першому етапі після застосування методу k-середніх з початковими центроїдами, що були отримані за допомогою кластерів з дослідження [18] (обчислюємо центроїди як середні арифметичні часових рядів для країн, що входять до нього, як і було в описі методу. У другому кластері не беремо піддерево з Ліхтенштейну, Швейцарії, Люксембургу та Хорватії, оскільки до

нього входять дуже малі країни, які мають дуже специфічну статистику захворюваності і тому можуть негативно вплинути на роботу методу), і обчислення радіусів кластерів та відстаней до центру, отримуємо наступні 4 кластери (Таблиця 1).

Одразу можемо порівняти склад отриманих кластерів з тими, що були в дослідженні [18]. Для цього стане в нагоді наведена нижче порівняльна діаграма для результатів кластеризації методом найближчого сусіда та методом k-середніх (Рис.7).

Таблиця 1. Чотири кластери, що були отримані методом k-середніх з початковими центроїдами на основі результатів дослідження [18], відстані до центроїдів та радіуси кластерів

Кластер 1			Кластер 2		
Країни	Відстань до центру	Радіус	Країни	Відстань до центру	Радіус
Монако	2019,54	2933,8	Швейцарія	2418,38	5399,47
Іспанія	2092,76		Нідерланди	2544,36	
Ірландія	2452,35		Словенія	2738,23	
Португалія	2813,49		Люксембург	2742,08	
Великобританія	2933,8		Франція	3094,83	
			Ліхтенштейн	3144,83	
			Бельгія	3474,64	
			Литва	4746,46	
			Андорра	5081,48	

			Чехія	5399,47	
Кластер 3			Кластер 4		
Країни	Відстань до центру	Радіус	Країни	Відстань до центру	Радіус
Болгарія	1282,05	5373,97	Білорусь	816,13	1850,46
Македонія	1631,92		Німеччина	1027,87	
Боснія	1773,06		Росія	1053,73	
Австрія	1808,17		Норвегія	1195,49	
Італія	1882,26		Фінляндія	1475,34	
Молдова	1898,58		Україна	1599,45	
Польща	2378,5		Греція	1746,58	
Румунія	2404,54		Данія	1802,82	
Словаччина	2629,36		Албанія	1850,46	
Угорщина	2682,12				
Хорватія	2973,06				
Швеція	2976,58				
Латвія	3307,32				
Косово	4160,92				
Естонія	4551,05				
Сербія	5373,97				

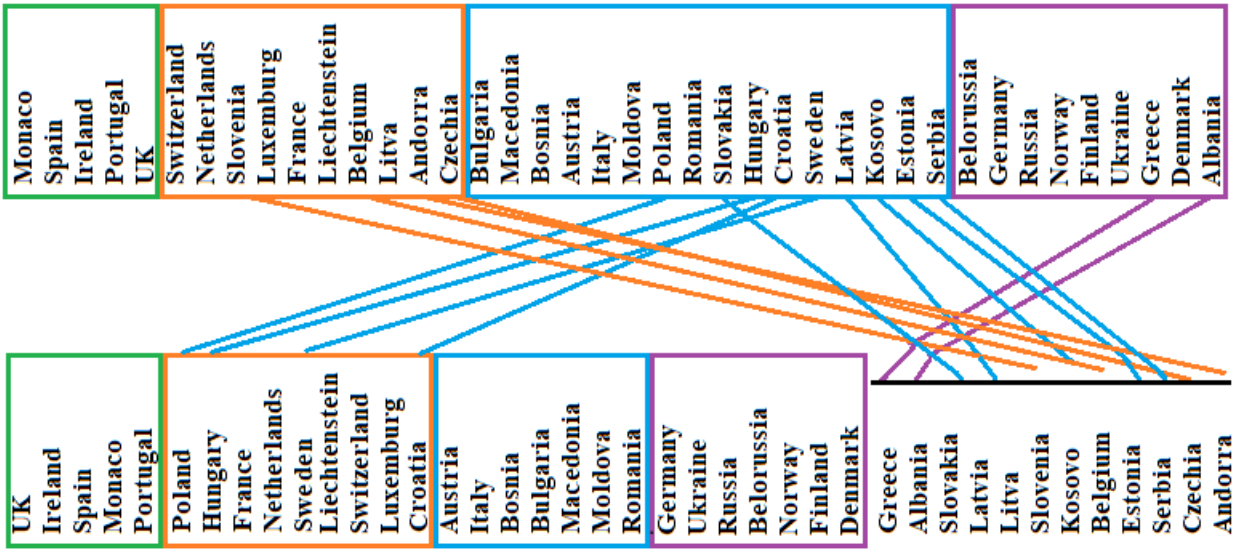

Рис. 7 Співставлення кластерів, що були знайдені в дослідженні [18] за допомогою методу найближчого сусіда (внизу) та кластерів, що були знайдені методом k-середніх (лінії позначають перехід з одного кластера в інший)

Подивимось перш за все на країни, на основі яких були сформовані центроїди. Тут результати досить добре збігаються з методом найближчого сусіду: лише Польща, Угорщина та Швеція перейшли з 2 до 3 кластеру, інші ж країни залишилися там, де і були. Цікаво, що Ліхтенштейн, Люксембург та Швейцарія також залишилися в 2 кластері, хоча вони і не використовувалися для обчислення центроїдів. Інші країни стояли на рис. 3 як би окремо від чотирьох кластерів, але при застосуванні методу k-середніх вони розподілились між цими кластерами.

Тепер задля того, щоб представити отримані кластери у вигляді дерев та порівняти їх внутрішню структуру з кластерами з дослідження [18], застосовуємо метод найближчого сусіда до кожного з кластерів, тільки в цьому випадку оберемо в якості метрики модуль різниці між відстанями до центру відповідного кластера. Отримуємо наступні дендрограми (номер біля дендрограми – номер

відповідного кластера) (Рис.8). Бачимо, що структура цих кластерів дуже відрізняється від тих, що були в дослідженні [18].

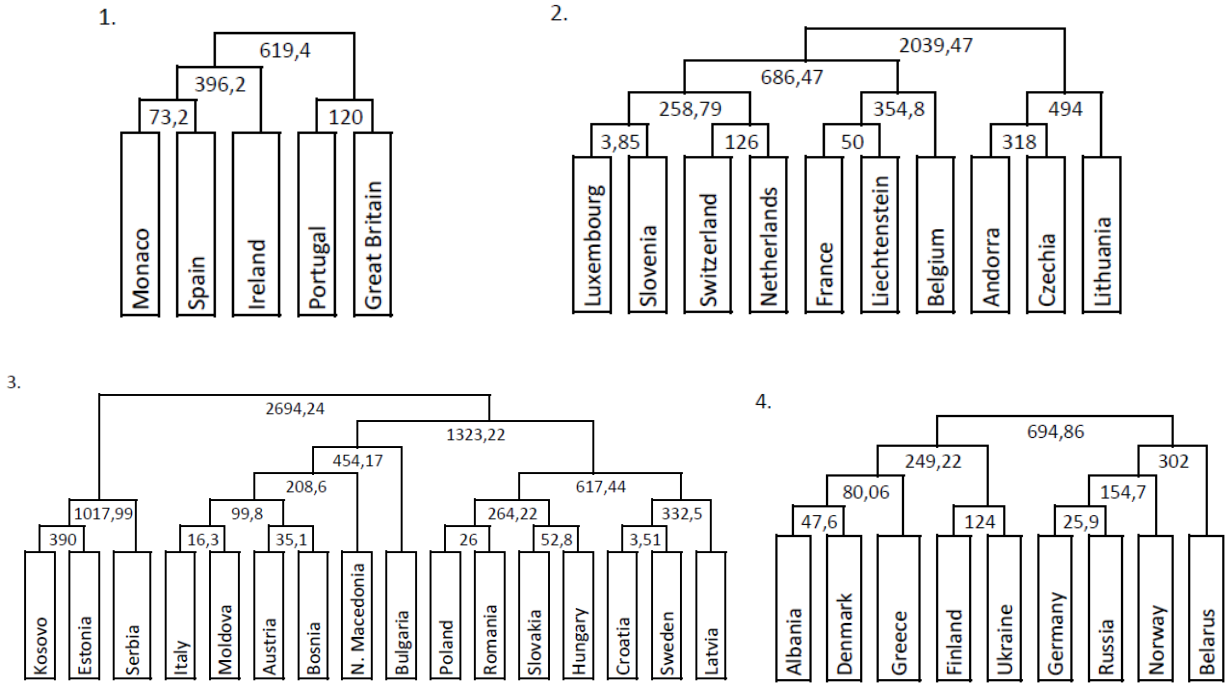

Рис. 8 Дендрограми для 4 кластерів, що були отримані за допомогою методу k-середніх (початкові центроїди на основі результатів дослідження [18])

Ми вже відмічали, що вибір початкових центроїдів має дуже великий вплив на результати роботи методу k-середніх, тому спробуємо побудувати кластеризацію спираючись на інші початкові центроїди. Для цього розіб'ємо усю вибірку європейських країн (за винятком дуже малих країн, оскільки вони мають дуже специфічну статистику і це може негативно вплинути на роботу методу k-середніх) на 7 груп країн, що близькі одна до одної, та обчислимо центроїди як середні арифметичні часових рядів для країн, що входять до відповідної групи, як і було в описі методу k-середніх. [41]

Групи географічно близьких країн:

Група 1: Іспанія, Португалія

Група 2: Франція, Бельгія, Нідерланди

Група 3: Великобританія, Ірландія

Група 4: Німеччина, Австрія, Швейцарія, Італія

Група 5: Данія, Швеція, Норвегія, Фінляндія

Група 6: Польща, Україна, Естонія, Латвія, Литва, Чехія, Словаччина, Угорщина

Група 8: Словенія, Хорватія, Боснія, Сербія, Македонія, Албанія, Греція, Румунія, Молдова

Отримуємо наступні 7 кластерів:

Таблиця 2. Чотири кластери, що були отримані методом k-середніх з початковими центроїдами на основі географічних груп, відстані до центроїдів та радіуси кластерів

Кластер 1			Кластер 2		
Країни	Відстань до центру	Радіус	Країни	Відстань до центру	Радіус
Іспанія	1436,7	2198.95	Франція	2412,36	4434,38
Монако	1892,58		Нідерланди	2553,26	
Португалія	2198,95		Бельгія	2628,78	
			Андорра	4434,38	
Кластер 3			Кластер 4		
Країни	Відстань до центру	Радіус	Країни	Відстань до центру	Радіус

Ірландія	1700,35	1700,35	Австрія	1212,4	2906,48
Великобританія	1700,35		Італія	1446,75	
			Швейцарія	1632,44	
			Боснія	2055,13	
			Польща	2245,91	
			Люксембург	2693,95	
			Ліхтенштейн	2906,48	

Кластер 5			Кластер 6		
Країни	Відстань до центру	Радіус	Країни	Відстань до центру	Радіус
Беларусь	861,58	1908,65	Словаччина	2445,08	5191,29
Німеччина	1045,53		Швеція	2613,18	
Росія	1139,71		Словенія	3213,89	
Норвегія	1311,43		Латвія	3311,61	
Україна	1464,43		Угорщина	3320,54	
Фінляндія	1586,58		Естонія	3767,42	
Албанія	1766,09		Чехія	5191,29	
Греція	1771,89				
Данія	1830,83				
Молдова	1908,65				
Кластер 7					

Країни	Відстань до центру	Радіус
Македонія	1976,87	4423,76
Болгарія	2016,37	
Хорватія	2394,35	
Румунія	2640,43	
Литва	3951,12	
Косово	4136,76	
Сербія	4423,76	

Одразу можемо порівняти склад кластерів зі складом початкових груп. Бачимо дуже багато відмінностей: хоча менші кластери (1, 2, 3,7) змінилися не так сильно, склад більших кластерів (4, 5, 6) вже має мало спільного з початковими групами.

Тепер, як і в попередньому випадку, застосуємо метод найближчого сусіда до отриманих кластерів із метрикою у вигляді модулю різниці відстаней до центру відповідних кластерів. В якості результату отримаємо дендрограму на рис. 12, на якій можна побачити, що внутрішня структура кластерів, побудованих за географічними групами, кардинально відрізняється від структури кластерів на рис. 3. Ця дендрограма та мапи, які допомагають відстежувати географічні закономірності в результатах кластеризації, наведені в додатку 1.

Треба зауважити, що кластер 3 має цікаву особливість: оскільки він складається з двох країн і його центроїд є середнім арифметичним часових рядів для цих країн, вони мають однакові відстані до центру. Тому якщо ми будемо

розраховувати відстань між елементами як модуль різниці відстаней до центра, отримаємо 0, що, звісно, не відповідає дійсності. Цей приклад показує недолік відстані, що була введена заради побудови дендрограм методом найближчого сусіда для кластерів, які були отримані за допомогою метода k-середніх.

На другому етапі застосовуємо метод k-середніх із відстанню Махаланобіса. Слід враховувати, що перехід до іншого набору даних може істотно вплинути на результати кластеризації. Тому для нового набору даних зі статистичними показниками використовуються як декартова метрика, так і відстань Махаланобіса, щоб відокремити вплив зміни метрики від впливу зміни набору даних. Як і на першому етапі, будуємо кластеризацію за допомогою метода k-середніх з 4 початковими центроїдами на основі результату методу найближчого сусіда та з 7 початковими центроїдами на основі географічних груп європейських країн. Результати кластеризації зручно проілюструвати за допомогою кольорових мап (Рис.9-10, Cartesian metrics означає евклідову метрику).

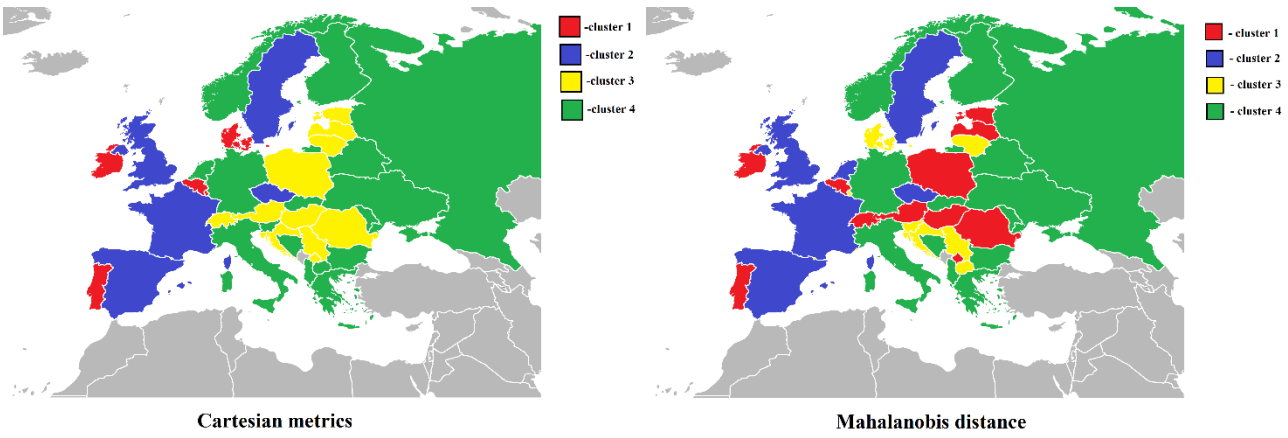

Рис. 9 Візуалізація на мапі Європи 4 кластерів (з початковими центроїдами на основі результатів метода найближчого сусіда) отриманих при застосуванні метода k-середніх до набора даних зі статистичними показниками

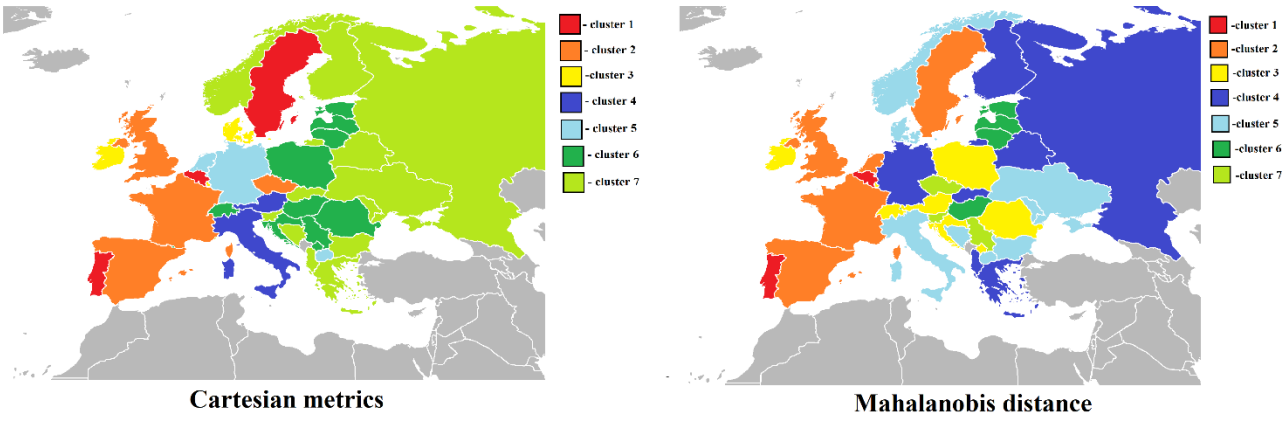

Рис. 10 Візуалізація на мапі Європи 7 кластерів (з початковими центроїдами на основі груп європейських країн) отриманих при застосуванні метода k-середніх до набора даних зі статистичними показниками

Після порівняння результатів кластеризації нового набору даних із попередніми результатами кластеризації ми можемо зробити такі висновки: в нових результатах стало менше відповідності між географічною близькістю країн та подібністю їхньої динаміки захворюваності на COVID-19, але більшість основних змін пояснюється переходом до іншого набору даних. Що стосується наслідків зміни метрики, то вона нечасто розбиває групи країн, що раніше були разом в одному кластері, натомість може помістити їх усіх разом в інший кластер. Наприклад, ми бачимо на рис. 9, що група з Польщі, Швейцарії, Австрії, Угорщини та Румунії належить до одного кластеру як після застосування декартової метрики, так і після застосування відстані Махаланобіса, але в першому випадку вона належить до кластера 3, а в останньому випадку він належить до кластера 1. [41]

Для оцінки впливу географічного фактору також корисно розглянути розмір найбільшої групи країн зі спільними кордонами в кластері для різних результатів кластеризації. Відповідні дані містяться в таблицях 3 та 4, які

наведені в додатку 2. Вони демонструють, що загалом перехід від набору даних с часовими рядами до набору даних з їх статистичними показниками та від евклідової метрики до відстані Махаланобіса призводить до зникнення географічних закономірностей, що були виявлені раніше, та робить результати кластеризації складнішими для інтерпретації.

Незважаючи на те, що результати кластеризації на другому етапі показують, що відповідність між географічною близькістю країн та їхньою схожістю за статистикою захворюваності на коронавірус є досить нестійкою, все ж є деякі закономірності, які представляють великий інтерес для подальших досліджень. Можна помітити декілька груп країн, які географічно не близькі та на перший погляд мають небагато спільних рис, але дуже часто потрапляють до одного кластера, наприклад Франція та Чехія (вони належать до одного кластера в усіх випадках, окрім результатів кластеризації з 7 географічними центроїдами з використанням евклідової метрики для часових рядів та відстані Махаланобіса для набору даних зі статистичними показниками) або Німеччина, Білорусь та Росії (вони належать до одного кластеру в усіх випадках, окрім результатів кластеризації з 7 географічними центроїдами з використанням евклідової метрики набору даних зі статистичними показниками).

В цій ситуації варто було б окремо перевірити, чи дійсно країни в цих нетипових групах схожі за статистикою COVID-19. Щоб оцінити їхню подібність, ми будуємо два окремих графіки для цих груп (на першому графіку ми також додаємо криву для Нідерландів, оскільки ця країна часто знаходиться в одному кластері з Францією та Чехією, але географічно близька до Франції і тому інтуїтивно здається більш схожою на неї) і один спільний графік для обох груп.

На цих графіках можна побачити, що криві для країн з однієї групи досить схожі за піками захворюваності та їх крутизною (піки захворюваності Білорусі, Німеччини та Росії зміщені вперед порівняно з піками захворюваності в Чехії,

Франції та Нідерландах і є значно менш крутими), тоді як криві для країн з різних груп помітно відрізняються. Це спостереження доводить достовірність отриманих результатів кластеризації. Усі три порівняльні графіки наведені в додатку 1.

На останньому етапі застосовуємо метод k-середніх із евклідовою метрикою та відстанню Махаланобіса до кумулятивних даних (тобто часових рядів із загальною кількістю нових випадків на мільйон). Як і на другому етапі, для відстані Махаланобіса використовується набір даних із чотирма статистичними показниками. Якщо знов проілюструвати результати кластеризації за допомогою кольорових мап, то можна побачити, що вони демонструють не більш стійкий зв'язок між географічною близькістю та статистикою COVID-19, ніж на попередніх етапах. Однак на усіх чотирьох мапах чітко виділяється новий кластер зі східноєвропейських і балканських країн (він зображений окремо на наступній мапі). Усі 4 мапи з результатами кластеризації кумулятивних даних для різних початкових центроїдів та метрик можна знайти в додатку 1.

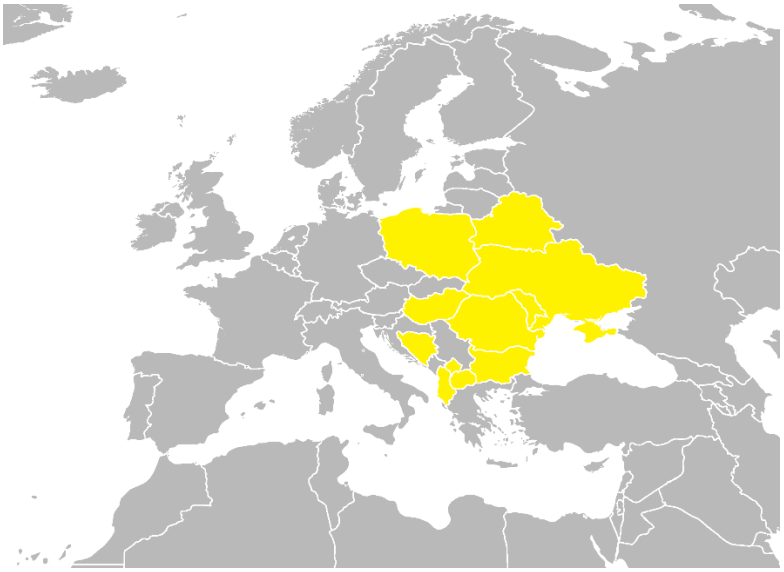

Рис. 11 Кластер зі східноєвропейських і балканських країн, що спостерігається в результатах кластеризації кумулятивних даних

Незважаючи на те, що деякі з цих країн часто потрапляли в один кластер в попередніх результатах кластеризації, не було об’єднаного географічного блоку країн, який вже можна чітко побачити чітко на мапах для кумулятивних даних. Оскільки цей кластер не розпадається після зміни початкових центроїдів і метрики, подібність між цими країнами можна вважати сильною та стійкою. Цей висновок підтверджується графіком загальної кількості випадків на мільйон у Боснії, Польщі та Румунії (ці три країни були взяті як вибірка з кластеру) та графіком для усіх країн з кластера, які наведені у додатку 1. Помітно, що ці три країни мають багато спільного у динаміці захворюваності, а графіки в цілому дуже близькі один до одного, особливо в часовому проміжку з березня 2020 року до березня 2021 року.

6. Висновки

Результати кластеризації методом k-середніх в цілому збігаються з результатами, отриманими методом найближчого сусіда (принаймні для тих країн, які були враховані при розрахунку початкових центроїдів). Оскільки результати, отримані методом найближчого сусіда, вважаються досить надійними та коректними, можна стверджувати про коректність результатів метода k-середніх із зазначеним вибором початкових центроїдів. Але структура дендрограм, які були побудовані за результатами кластеризації за допомогою метода k-середніх, вже суттєво відрізняється від структури дендрограм, які були побудовані за методом найближчого сусіда. Крім того, обрана для побудови нових дендрограм відстань не завжди дає правильні результати, як це було з двоелементним кластером. Отже, можна зробити висновок, що для визначення

складу кластерів доцільно використовувати метода k-середніх, але для побудови ієрархічних структур краще окремо використовувати інші методи. [41]

Щоб зробити результати кластеризації більш правильними завдяки урахуванню кореляцій у наборі даних, був застосований метод k-середніх з метрикою Махаланобіса. Задля того, щоб цей метод можна було раціонально застосувати, довелося перейти до нового набору даних зі статистичними показниками, і виявилося, що цей перехід набагато більше впливає на результати кластеризації, ніж зміна метрики. [41]

Великий інтерес представляє питання того, наскільки схожими є географічно близькі країни за статистикою захворюваності. Аналізуючи результати метода k-середніх, можна сказати, що географічно близькі країни часто потрапляють в один кластер, навіть частіше, ніж при використанні метода найближчого сусіда. Тим не менш, виявлені географічні закономірності часто є нестабільними і можуть зникнути при переході до нового набору даних або при зміні евклідової метрики на відстань Махаланобіса. Також у результатах обох методів ми можемо побачити багато випадків, коли країни, які знаходяться далеко одна від одної, потрапляють в один кластер, і деякі з цих незвичайних сполук утворюються як при застосуванні метода k-середніх (і майже всіх його реалізованих варіацій), так і при застосуванні метода найближчого сусіда: наприклад, Росія, Німеччина і Білорусь часто потрапляють в один кластер (зазвичай разом з кількома країнами Північної Європи). Такі незвичайні сполуки навряд чи є цілком випадковими і тому потребують подальшого дослідження. Однак кластерний аналіз кумулятивних даних (тобто часових рядів для загальних випадків захворюваності на мільйон населення) виділяє стабільний і географічно однорідний кластер східноєвропейських і балканських країн, який відповідає географічній логіці набагато краще, ніж попередні сполуки. Порівняння динаміки захворюваності на коронавірус у деяких країнах з цих сполук довело, що вони

справді схожі за статистикою захворюваності та що результати кластеризації є коректними. [41]

Отримані результати показують, що географічна близькість країн має значний вплив на їх статистику захворюваності, але є й інші важливі фактори, які пояснюють подібність статистики коронавірусу в географічно віддалених країнах. [41]

7. Список використаних джерел:

1. Covid-19 Pandemic. Problems Arising in Health and Social Policy. /Ch. Aspalter (Ed.). Springer Singapore. 2023. 303 p. https://doi.org/10.1007/978-981-99-2497-4
2. Koley T.K., Dhole M. The COVID-19 Pandemic: The Deadly Coronavirus Outbreak. Taylor & Francis, 2020. - 174 p.
3. Nandiyanto A.B.D. COVID-19: From Health, Education, Economic, to Science and Technology in South East Asia and India. Nova Science Publ., Inc. 2021. - 385 p.
4. Cascella M., Rajnik M., Aleem A., et al. Features, Evaluation, and Treatment of Coronavirus (COVID-19). In: StatPearls [Internet]. Treasure Island: StatPearls Publishing. 2024. Available from: https://www.ncbi.nlm.nih.gov/books/NBK554776/
5. Spiegelhalter D., Masters A. Covid by numbers: Making sense of the pandemic with data. Penguin UK, 2022. - 320 p. https://doi.org/10.1093/occmed/kqac013
6. Zakharova A.A., Kizilova N.M. Study of correlations of the dynamics of COVID-19 disease with some socio-economical factors. Bull. of V.N. Karazin Kharkiv National University. ser. "Mathematical modeling. Information Technology. Automated control systems". – 2020. – N48. – P.49-56. (in Ukrainian)
7. Fox W.P. Mathematical Modeling in the Age of the Pandemic. CRC Press, Taylor & Francis Group, 2021. 164 p.
8. Vytla V., Ramakuri S.K., Peddi A., et al Mathematical Models for Predicting Covid-19 Pandemic: A Review. J. Phys.: Conf. Ser. 2020. Vol.1797, 012009. doi: 10.1088/1742-6596/1797/1/012009
9. Krishna M.V. Mathematical modelling on diffusion and control of COVID–19. Infectious Disease Modelling. 2020, Vol.5, P.588-597. https://doi.org/10.1016/j.idm.2020.08.009
10. Kizilova N. Nonlinear Dynamics, Stability and Control Strategies: Mathematical Modeling on the Big Data Analyses of Covid-19 in Poland. // 16th Intern. Conf. "Dynamical Systems: Theory and Applications". Lodz University of Technology. - 2021. - P.467-468.

11. Fraguela-Collar A. Moving From COVID-19 Mathematical Models to Vaccine Design: Theory, Practice and Experiences. Bentham Sci. Publ. 2022. 583 p.
12. Brauer, F., Mathematical epidemiology: Past, present, and future. Infect. Dis. Model. 2(2), 113–127 (2017).
13. Overton Ch.E., Stage H.B., Ahmad S., et al. Using statistics and mathematical modelling to understand infectious disease outbreaks: COVID-19 as an example. Infectious Disease Modelling, 2020, Vol.5. P.409-441. https://doi.org/10.1016/j.idm.2020.06.008.
14. He X., Lau E.H.Y., Wu P., et al. Temporal dynamics in viral shedding and transmissibility of COVID-19, Nat. Med. 2020, Vol. 26, P. 672–675.
15. Chun, J.Y., Baek, G., Kim, Y., Transmission onset distribution of COVID-19, Intern. J. Infect. Dis., 99, 403–407 (2020).
16. Danelon A.F., Kumbhakar S.C. Estimating COVID-19 under-reporting through stochastic frontier analysis and official statistics: A case study of São Paulo State, Brazil. Socio-Economic Planning Sciences, 2023. Vol. 90, 101753. https://doi.org/10.1016/j.seps.2023.101753.
17. Kostecka V.V., Kizilova N.M. Mathematical modeling of dynamics of the covid-19 epidemic. Bull. of V.N. Karazin Kharkiv National University. ser. "Mathematical modeling. Information Technology. Automated control systems". – 2020. – N48. – P.65-71. (in Ukrainian)
18. Kulik D.A., Kurkchi Ye.P., Kizilova N.M. Analysis of "big data" and mathematical modeling of the covid-19 epidemic in European countries. Bull. of V.N. Karazin Kharkiv National University. ser. "Mathematical modeling. Information Technology. Automated control systems". – 2021. – N52. – P.35-42. (in Ukrainian) https://doi.org/10.26565/2304-6201-2021-52-05
19. Voloshyna K.I., Kizilova N.M., Kiporenko P.V. Study of the dynamics of four waves of covid-19 in European countries. Bull. of V.N. Karazin Kharkiv National

University. ser. "Mathematical modeling. Information Technology. Automated control systems". – 2021. – N54. – P.6-15. (in Ukrainian)

20. Sempé L., P. Lloyd-Sherlock, R. Martínez, S. Ebrahim, M. McKee, E. Acosta Estimation of all-cause excess mortality by age-specific mortality patterns for countries with incomplete vital statistics: a population-based study of the case of Peru during the first wave of the COVID-19 pandemic. The Lancet Regional Health - Americas, 2021, Vol.2, 100039. https://doi.org/10.1016/j.lana.2021.100039.

21. Aldila, S.H.A., Khoshnaw, E., Safitri, et al., A mathematical study on the spread of COVID-19 considering social distancing and rapid assessment: The case of Jakarta, Indonesia, Chaos, Solitons & Fractals, 139, 110042 (2020).

22. Li Q., Guan X., Wu P., et al.Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia. New England Journal of Medicine, 2020. Vol. 382, 1199-1207. htpps://doi.org/ 10.1056/NEJMoa2001316

23. Asadi-Zeydabadi M., Buscema M., Lodwick W., et al. Analysis of COVID-19 pandemic in USA, using Topological Weighted Centroid. Computers in Biology and Medicine. 2021. Vol.136, 104670. htpps://doi.org/ 10.1016/j.compbiomed.2021.104670

24. Sun J. Forecasting COVID-19 pandemic in Alberta, Canada using modified ARIMA model. Computer Methods and Programs in Biomedicine. 2021. Vol. 22, 100029. https://doi.org/10.1016/j.cmpbup.2021.100029

25. Galanis G., Georgiadis A. Socioeconomic conditions and contagion dynamics of the COVID-19 pandemic with and without mitigation measures: Evidence from 185 countries. World Development, 2024, Vol.175, 106477. https://doi.org/10.1016/j.worlddev.2023.106477.

26. Žid P., M. Haindl, V. Havlíček Governmental Anti-Covid Measures Effectiveness Detection. Procedia Computer Science, 2023, Vol.225, P.2922-2931. https://doi.org/10.1016/j.procs.2023.10.285.

27. Tomer V., Gupta S., Manwal M., Singh D.P. How statistics of World health index react against COVID-19. Materials Today: Proceedings, 2021. Vol.46, P.11267-11273. https://doi.org/10.1016/j.matpr.2021.03.486.

28. de Araújo Morais L.R., da Silva Gomes G.S. Applying Spatio-temporal Scan Statistics and Spatial Autocorrelation Statistics to identify Covid-19 clusters in the world - A Vaccination Strategy?
Spatial and Spatio-temporal Epidemiology. 2021. Vol.39. 100461. https://doi.org/10.1016/j.sste.2021.100461.

29. Ramadan A., Kamel A., Taha A., et al. A multivariate data analysis approach for investigating daily statistics of countries affected with COVID-19 pandemic. Heliyon, 2020. Vol.6, N11, e05575.
https://doi.org/10.1016/j.heliyon.2020.e05575.

30. Na J., Tibebu H., De Silva V., Kondoz A., Caine M. Probabilistic approximation of effective reproduction number of COVID-19 using daily death statistics. Chaos, Solitons & Fractals, 2020. Vol.140,110181. https://doi.org/10.1016/j.chaos.2020.110181.

31. Big Data Analysis: New Algorithms for a New Society, ed. by N. Japkowicz, J. Stefanowski, Springer (2016).

32. Research and data to make progress against the world's largest problems. [Electronic resource]. Available: https://ourworldindata.org/coronavirus

33. Ahmed Hamed, Ahmed Sobhy, Hamed Nassar. Accurate Classification of COVID-19 Based on Incomplete Heterogeneous Data using a KNN Variant algorithm. Arabian Journal for Science and Engineering 2021.
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7931985/

34. Кізілова Н.М. Методи аналізу «великих даних»: методичні рекомендації з курсу «Прикладні задачі аналізу великих даних» для здобувачів вищої освіти спеціальності «Прикладна математика». – Харків : Харківський національний університет імені В.Н. Каразіна, 2023.

35. Pan-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. Introduction to Data Mining (Second Edition). 2019. Chapter 7. P. 10-20

36. О. М. Ткаченко, О. Ф. Грійо Тукало, О. В. Дзісь, С. М. Лаховець. Метод кластеризації на основі послідовного запуску k-середніх з удосконаленим вибором кандидата на нову позицію вставки. НаукПраці ВНТУ. 2012. С. 1

37. Murtagh F, Contreras P (2012). "Algorithms for hierarchical clustering: an overview". Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2 (1). Wiley Online Library 86–97. doi:10.1002/widm.53

38. Gower JC, Ross GJ (1969). "Minimum spanning trees and single linkage cluster analysis". Journal of the Royal Statistical Society, Series C. 18 (1): 54–64. doi:10.2307/2346439. JSTOR 2346439. MR 0242315.

39. De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D. L. (2000). "The Mahalanobis distance". Chemometrics and Intelligent Laboratory Systems. 50 (1): 1–18. doi:10.1016/s0169-7439(99)00047-7

40. Фальченко І. Р. Кізілова Н. М. Аналіз «великих даних» динаміки COVID-19 у країнах Європи методами кластерізації з різними метриками. Збірник тез доповідей XVIII Міжнародної науково-практичної конференції студентів та молодих вчених «Сучасні проблеми математики та її застосування в

природничих науках та інформаційних технологіях». – Харків : Харківський національний університет імені В.Н. Каразіна, 10-11 травня 2024р. С. 45-48

41. Falchenko I. R., Kizilova N. M. Cluster Classification with Different Metrics for Big Data Analysis of Covid-19 Dynamics. Proceedings ICTERI 2024 (submitted)

8. Додаток 1

У цьому додатку наведені дендрограми, кольорові мапи з кластерами та графіки, які були згадані у цій роботі.

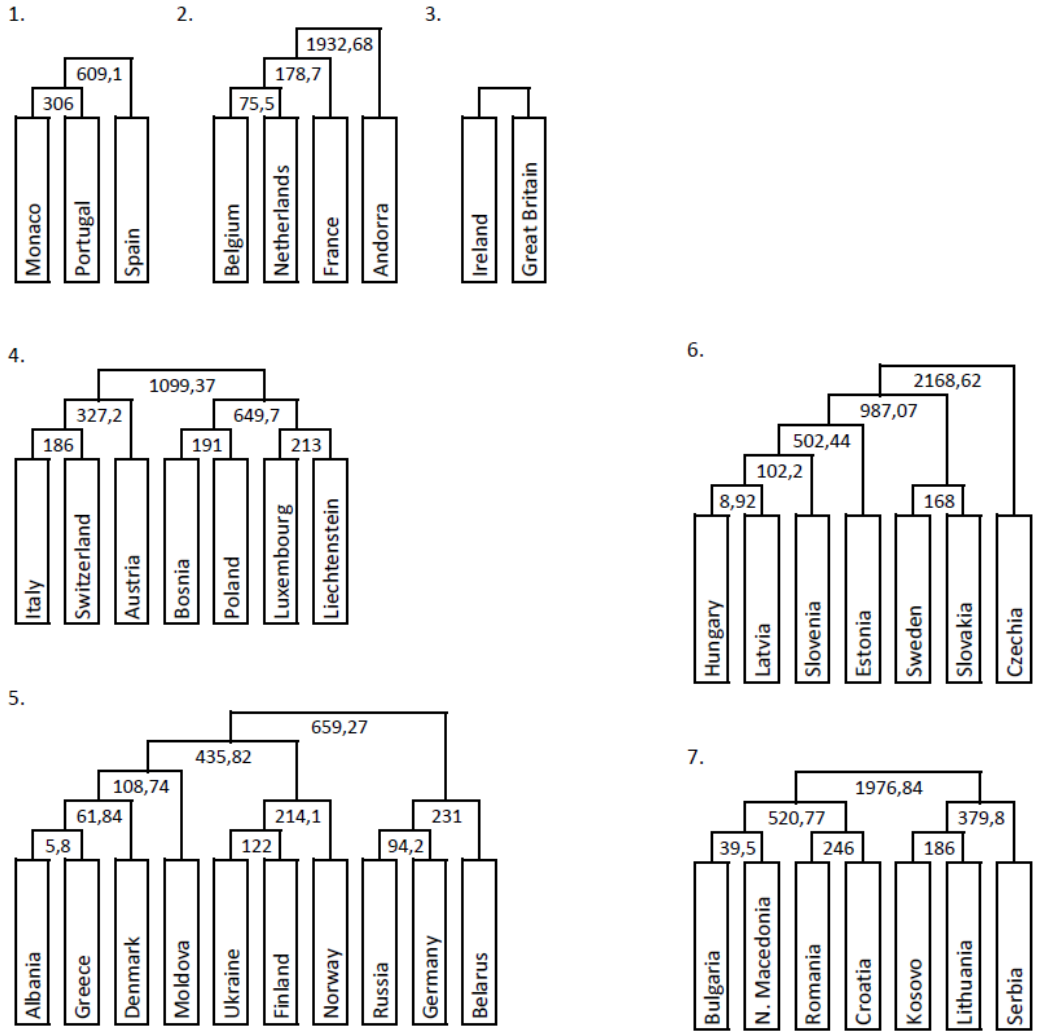

Рис. 12 Дендрограми для 7 кластерів, що були отримані за допомогою методу k-середніх (початкові центроїди на основі груп географічно близьких країн)

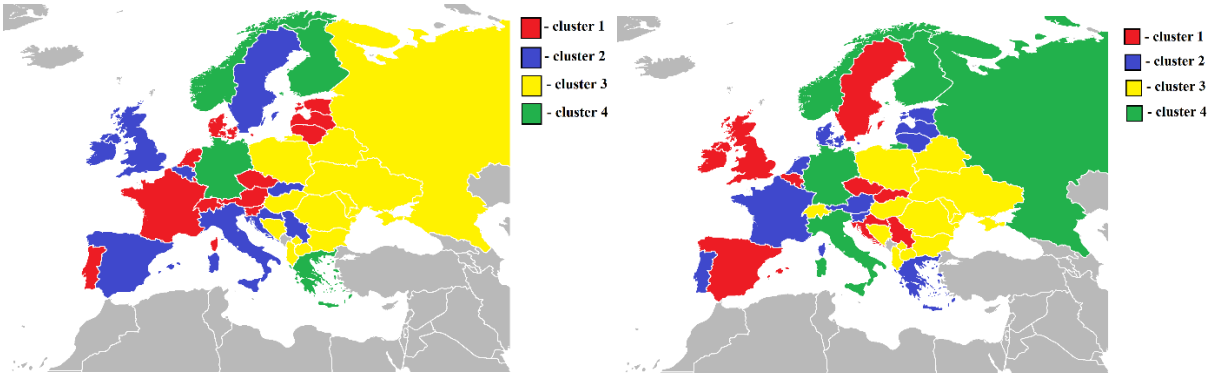

Рис. 13 Візуалізація на мапі Європи 4 кластерів (з початковими центроїдами на основі результатів метода найближчого сусіда) отриманих при застосуванні метода k-середніх до набора даних зі статистичними показниками (кумулятивні дані)

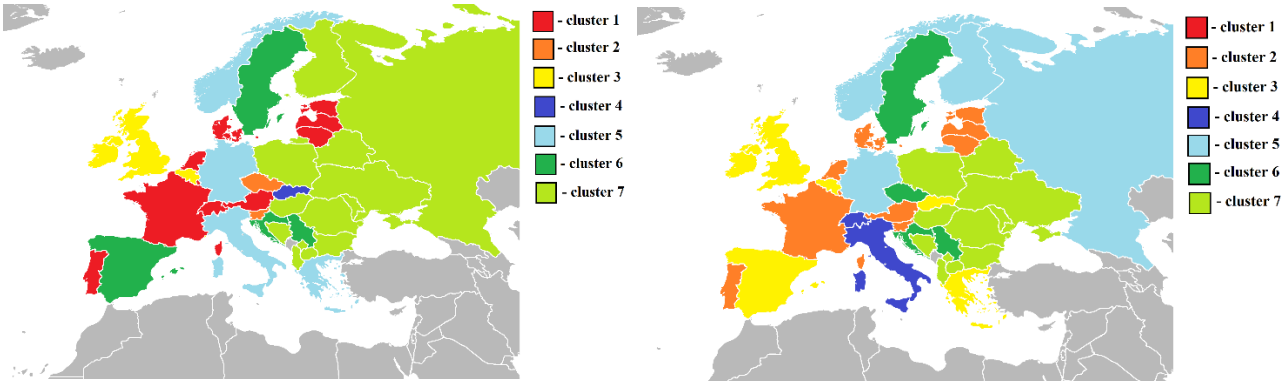

Рис. 14 Візуалізація на мапі Європи 7 кластерів (з початковими центроїдами на основі груп європейських країн) отриманих при застосуванні метода k-середніх до набора даних зі статистичними показниками (кумулятивні дані)

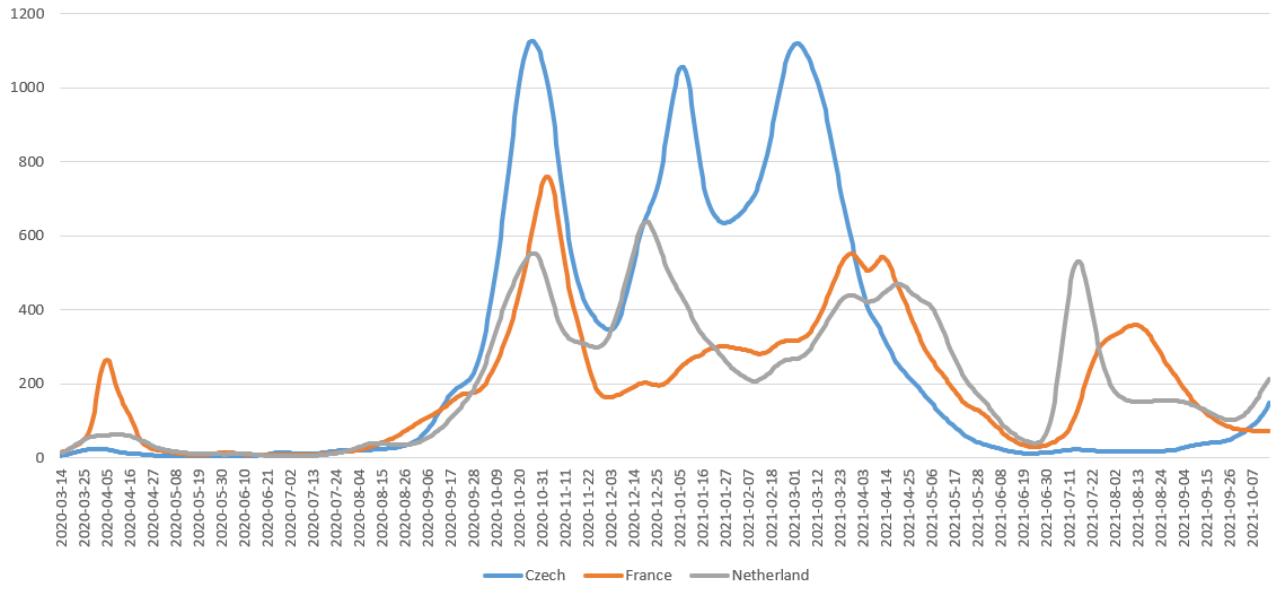

Рис. 15 Порівняльний графік нових випадків захворювання на мільйон населення для Франції, Чехії та Нідерландів

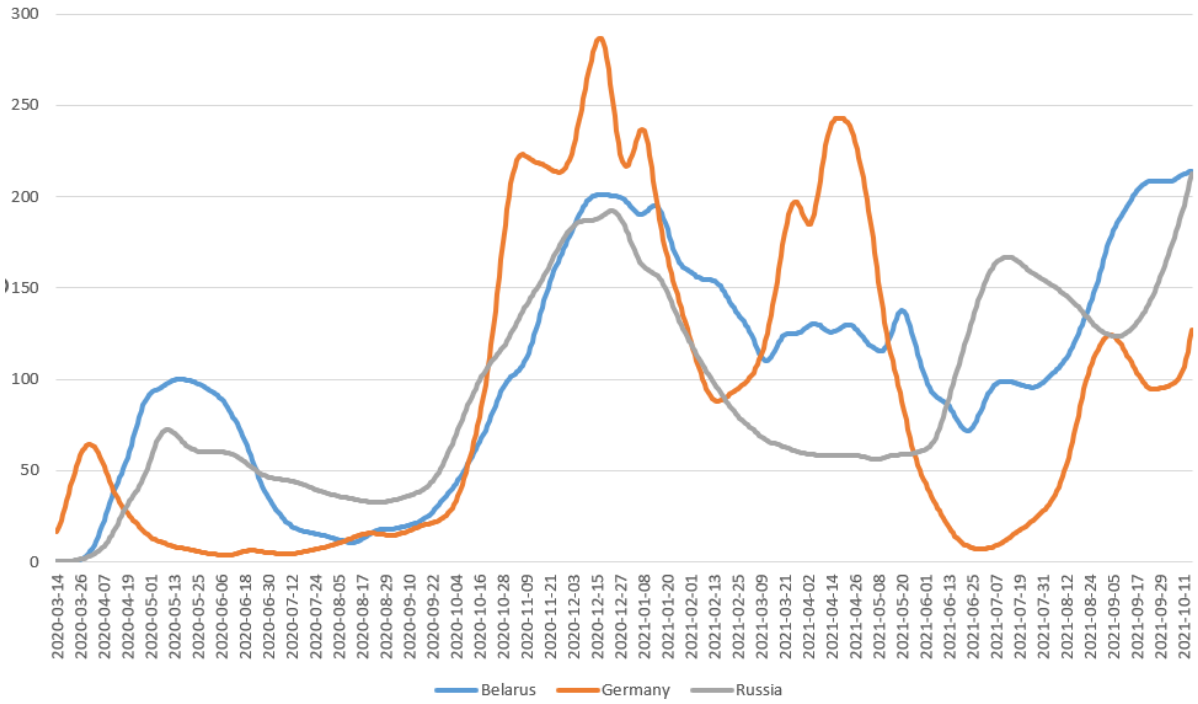

Рис. 16 Порівняльний графік нових випадків захворювання на мільйон населення для Беларусі, Німеччини та Росії

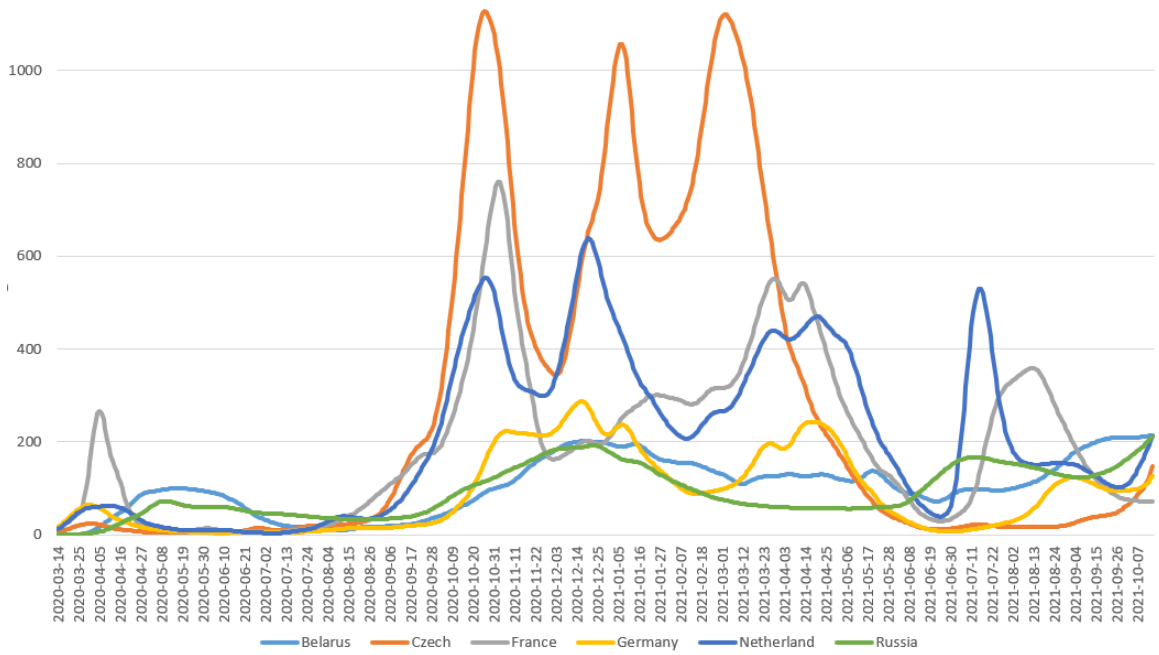

Рис. 17 Загальний порівняльний графік нових випадків захворювання на мільйон населення для Беларусі, Німеччини, Росії, Франції, Чехії та Нідерландів

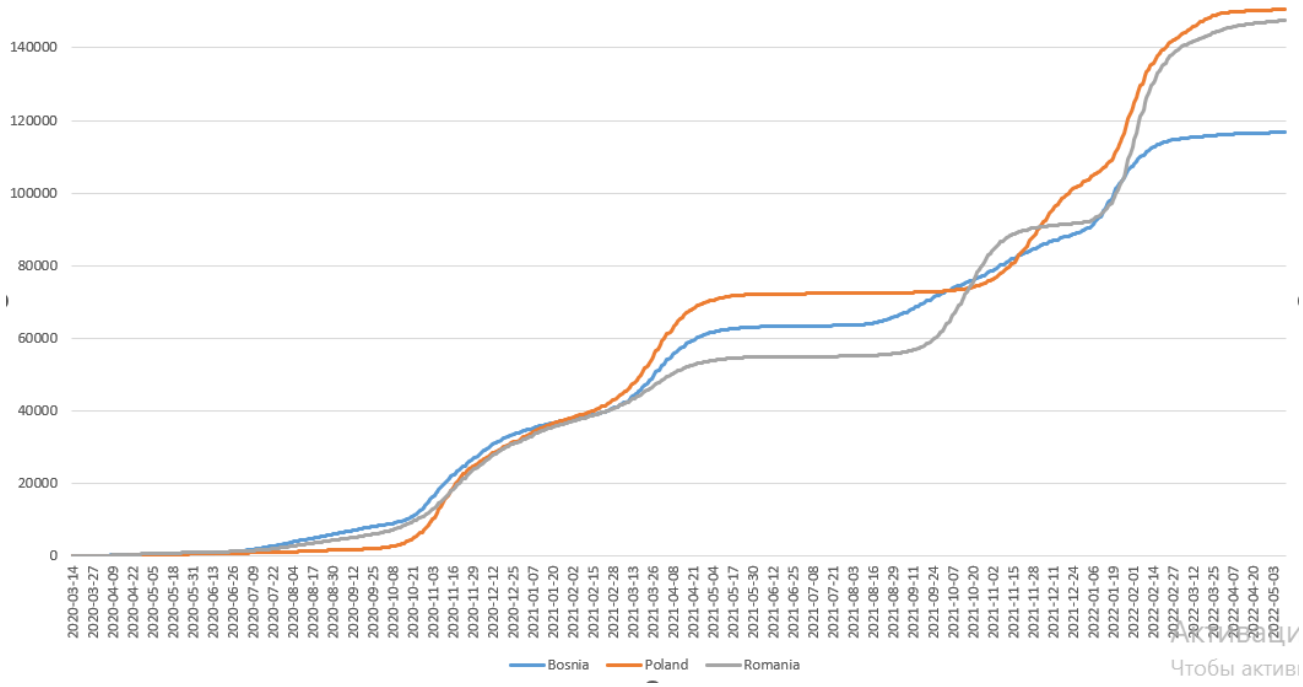

Рис. 18 Порівняльний графік загальної кількості випадків захворювання на мільйон населення для Боснії, Польщі та Румунії

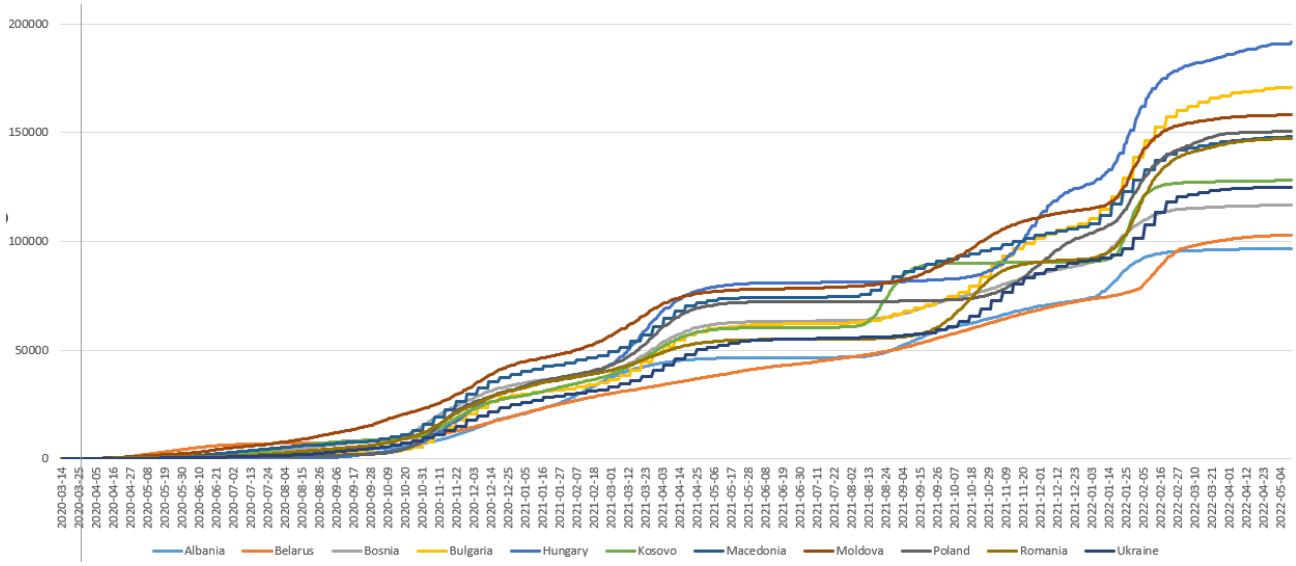

Рис. 19 Порівняльний графік загальної кількості випадків захворювання на мільйон населення для балканських та східноєвропейських країн з кластера, який був виявлений для кумулятивних даних

Додаток 2

У цьому додатку наведені таблиці 3 та 4.

Таблиця 3. Розмір найбільшої групи країн зі спільними кордонами в кластері для різних результатів кластеризації (початкові центроїди на основі результатів метода найближчого сусіда)

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Initial dataset, Euclidean metric	2	5	13	5
New dataset, Euclidean metric	2	2	8	7

New dataset, Mahalanobis distance	5	2	4	7

Таблиця 4. Розмір найбільшої групи країн зі спільними кордонами в кластері для різних результатів кластеризації (початкові центроїди на основі груп географічно близьких країн)

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
Initial dataset, Euclidean metric	2	4	2	4	6	4	6
New dataset, Euclidean metric	1	2	1	2	2	5	7
New dataset, Mahalanobis distance	1	2	2	3	2	3	1

Додаток 3

У цьому додатку наведені програми в Python, за допомогою яких були реалізовані методи кластеризації.

1) Програма для метода найближчого сусіда та метода k-середніх з Евклідовою метрикою:

```
import pandas as pd
import numpy as np
import copy
import xlwings as xw



def clusterize_with_set_centers(df, centers, k):
    clusters = [[] for i in range(k)]
    countries = df.columns
    for country in countries:
        min_dist = np.linalg.norm(df[country] - centers[0])
        closest_center_num = 0
        for i in range(len(centers)):
            if np.linalg.norm(df[country] - centers[i]) < min_dist:
                min_dist = np.linalg.norm(df[country] - centers[i])
                closest_center_num = i
        clusters[closest_center_num].append(country)
    return clusters


def cluster_average(cluster):
    n = len(cluster)
    cluster_sum = sum([df[country] for country in cluster])
    return cluster_sum/n
```

```
def radius_and_distances(number, center, cluster):
    if len(cluster)==2:
        dist = [np.linalg.norm(df[country] - center) for country in cluster]
        print('Radius of the cluster', number, ': ', max(dist))
        print('Distances:')
        for i in range(2):
            print(cluster[i], ': ', dist[i])
        return dist
    else:
        dist = [np.linalg.norm(df[country] - center) for country in cluster]
        print('Radius of the cluster', number, ': ', max(dist))
        distance_dict = {key:value for (key,value) in zip(dist, cluster)}
        dist_sorted = copy.deepcopy(dist)
        dist_sorted.sort()
        print('Distances:')
        for d in dist_sorted:
            print(distance_dict[d], ': ', d)
        return dist


def neighbour_method(data, n):
    col = list(data.columns)
    last_col = col[-1]
    last_row = len(data)-1
    if n==0:
```

```
        return
    else:
        col = list(data.columns)
        dist = np.array([[np.linalg.norm(data.iloc[0:-1,i] - data.iloc[0:-1,j]) for j in range(0, len(col))] for i in range(len(col))])

        mins = [min(dist[i, (i+1):len(col)]) for i in range(len(col)-1)]

        minimum = min(mins)
        min_indices = list(zip(*np.where(dist == minimum)))
        i_min = min_indices[0][0]
        j_min = min_indices[0][1]

        print('merge: ', data[i_min][last_row], 'and', data[j_min][last_row])
        print('distance: ', minimum)

        temp1 = data[last_col-1]
        temp2 = data[last_col]
        min1 = data[i_min]
        min2 = data[j_min]
        if (i_min == (last_col-1)) and (j_min != last_col):
            data[last_col] = min2
            data[j_min] = temp2
        if (i_min != (last_col-1)) and (j_min == last_col):
            data[last_col-1] = min1
```

```
            data[i_min] = temp1
        if (j_min == last_col-1):
            data[last_col] = min1
            data[i_min] = temp2
        if (i_min != (last_col-1)) and (j_min != last_col) and (j_min != (last_col-1)):
            data[last_col-1] = min1
            data[last_col] = min2
            data[i_min] = temp1
            data[j_min] = temp2

        data.iloc[0:-1, last_col-1] = (data.iloc[0:-1, last_col-1]+data.iloc[0:-1, last_col])/2
        data[last_col-1][last_row] = data[last_col-1][last_row] + '+' + data[last_col][last_row]
        data1 = data.iloc[0:len(data), 0:last_col]
        return neighbour_method(data1, n-1)

k = 4
wb = xw.Book('Final COVID data.xls')
data_excel = wb.sheets['new cases']
df = data_excel.range('C1:AP582').options(pd.DataFrame, header = True, index = False).value
print('NEIGHBOR CLUSTERS')
neighbor_clusters = [['UK', 'Ireland', 'Spain', 'Monaco', 'Portugal'], \
                     ['Poland', 'Hungary', 'France', 'Netherland', 'Sweden'], \
                     ['Austria', 'Italy', 'Bosnia', 'Bulgaria', 'Macedonia', 'Moldova', 'Romania'], \
                     ['Germany', 'Ukraine', 'Russia', 'Belarus', 'Norway', 'Finland', 'Denmark']]
centers = [cluster_average(neighbor_clusters[i]) for i in range(k)]
```

```
clusters_old = clusterize_with_set_centers(df, centers, k)
while True:
    new_centers = [cluster_average(clusters_old[j]) for j in range(len(clusters_old))]
    clusters_new = clusterize_with_set_centers(df, new_centers, k)
    if clusters_new == clusters_old:
        for i in range(k):
            print('Cluster ', i+1, ': ', clusters_new[i])
            distances = radius_and_distances(i+1, new_centers[i], clusters_new[i])
            print(' ')
            pre_dist_df = [distances, clusters_new[i]]
            dist_df = pd.DataFrame(pre_dist_df, columns = [i for i in range(len(clusters_new[i]))])
            neighbour_method(dist_df, len(clusters_new[i])-1)
            print(' ')
        break
    else:
        clusters_old = copy.deepcopy(clusters_new)
```

```
k1 = 7
print('GEOGRAPHIC CLUSTERS')
geographic_clusters = [['Spain', 'Portugal'], \
                       ['France', 'Belgium', 'Netherland'], \
                       ['UK', 'Ireland'], \
                       ['Germany', 'Austria', 'Switzerland', 'Italy'], \
```

```
                ['Denmark', 'Sweden', 'Norway', 'Finland'], \
                ['Poland', 'Ukraine', 'Estonia', 'Latvia', 'Lithuania', 'Czech', 'Slovakia', 'Hungary'], \
                ['Slovenia', 'Croatia', 'Bosnia', 'Serbia', 'Macedonia', 'Albania', 'Greece', 'Romania', 'Moldova']]
centers_g = [cluster_average(geographic_clusters[i]) for i in range(k1)]
clusters_old_g = clusterize_with_set_centers(df, centers_g, k1)
while True:
    new_centers_g = [cluster_average(clusters_old_g[j]) for j in range(len(clusters_old_g))]
    clusters_new_g = clusterize_with_set_centers(df, new_centers_g, k1)
    if clusters_new_g == clusters_old_g:
        for i in range(k1):
            print('Cluster ', i+1, ': ', clusters_new_g[i])
            distances_g = radius_and_distances(i+1, new_centers_g[i], clusters_new_g[i])
            print(' ')
            pre_dist_df_g = [distances_g, clusters_new_g[i]]
            dist_df_g = pd.DataFrame(pre_dist_df_g, columns = [i for i in range(len(clusters_new_g[i]))])
            #print(dist_df)
            if len(clusters_new_g[i]) > 2:
                neighbour_method(dist_df_g, len(clusters_new_g[i])-1)
                print(' ')
            else:
                print('The cluster consists of two or less elements, therefore the nearest neighbor algorithm is
useless and non applicable')
                print(' ')
        break
    else:
```

```
    clusters_old_g = copy.deepcopy(clusters_new_g)
```

2) Програма для метода k-середніх з метрикою Махаланобіса:

```
import pandas as pd

import numpy as np

import scipy

import copy

import xlwings as xw

from math import sqrt


wb = xw.Book('COVID data for Mahalanobis distance 1.xls')

data_excel = wb.sheets['new cases']

df = data_excel.range('D589:AQ593').options(pd.DataFrame, header = True, index = False).value

#print(df)

#print(df.to_numpy())

covariance_matrix = np.cov(df.to_numpy())

eps = 0

print('eps =', eps)

covariance_matrix = covariance_matrix + eps*np.identity(4)

covariance_matrix_inverse = np.linalg.inv(covariance_matrix)

#print(np.linalg.eigvals(covariance_matrix_inverse))


def mahalanobis(x, y, inv_cov):

    left_product = np.dot((x-y), inv_cov)

    #print('temp = ', np.dot(left_product, (x-y).T))
```

```
    dist = sqrt(np.dot(left_product, (x-y).T))
    return dist


def clusterize_with_set_centers(df, centers, k):
    clusters = [[] for i in range(k)]
    countries = df.columns
    for country in countries:
        min_dist = mahalanobis(df[country].to_numpy(), centers[0].to_numpy(), covariance_matrix_inverse)
        closest_center_num = 0
        for i in range(len(centers)):
            if mahalanobis(df[country].to_numpy(), centers[i].to_numpy(), covariance_matrix_inverse) <
min_dist:
                min_dist = mahalanobis(df[country].to_numpy(), centers[i].to_numpy(),
covariance_matrix_inverse)
                closest_center_num = i
        clusters[closest_center_num].append(country)
    return clusters


def cluster_average(cluster):
    n = len(cluster)
    cluster_sum = sum([df[country] for country in cluster])
    return cluster_sum/n


def radius_and_distances(number, center, cluster):
    if len(cluster)==2:
```

```
        dist = [mahalanobis(df[country].to_numpy(), center.to_numpy(), covariance_matrix_inverse) for country in cluster]
        print('Radius of the cluster', number, ': ', max(dist))
        print('Distances:')
        for i in range(2):
            print(cluster[i], ': ', dist[i])
        return dist
    else:
        dist = [mahalanobis(df[country].to_numpy(), center.to_numpy(), covariance_matrix_inverse) for country in cluster]
        print('Radius of the cluster', number, ': ', max(dist))
        distance_dict = {key:value for (key,value) in zip(dist, cluster)}
        dist_sorted = copy.deepcopy(dist)
        dist_sorted.sort()
        print('Distances:')
        for d in dist_sorted:
            print(distance_dict[d], ': ', d)
        return dist


def neighbour_method(data, n):
    col = list(data.columns)
    last_col = col[-1]
    last_row = len(data)-1
    if n==0:
        return
    else:
```

```
    col = list(data.columns)
    #dist = np.array([[np.linalg.norm(data.iloc[0:-1,i] - data.iloc[0:-1,j]) for j in range(0, len(col))] for i in range(len(col))])
    dist = np.array([[abs(float(data.iloc[0:-1,i] - data.iloc[0:-1,j])) for j in range(0, len(col))] for i in range(len(col))])

    mins = [min(dist[i, (i+1):len(col)]) for i in range(len(col)-1)]

    minimum = min(mins)
    min_indices = list(zip(*np.where(dist == minimum)))
    i_min = min_indices[0][0]
    j_min = min_indices[0][1]

    print('merge: ', data[i_min][last_row], 'and', data[j_min][last_row])
    print('distance: ', minimum)

    temp1 = data[last_col-1]
    temp2 = data[last_col]
    min1 = data[i_min]
    min2 = data[j_min]
    if (i_min == (last_col-1)) and (j_min != last_col):
        data[last_col] = min2
        data[j_min] = temp2
    if (i_min != (last_col-1)) and (j_min == last_col):
        data[last_col-1] = min1
        data[i_min] = temp1
```

```
    if (j_min == last_col-1):
        data[last_col] = min1
        data[i_min] = temp2
    if (i_min != (last_col-1)) and (j_min != last_col) and (j_min != (last_col-1)):
        data[last_col-1] = min1
        data[last_col] = min2
        data[i_min] = temp1
        data[j_min] = temp2

    data.iloc[0:-1, last_col-1] = (data.iloc[0:-1, last_col-1]+data.iloc[0:-1, last_col])/2
    data[last_col-1][last_row] = data[last_col-1][last_row] + '+' + data[last_col][last_row]
    data1 = data.iloc[0:len(data), 0:last_col]
    return neighbour_method(data1, n-1)


'''
dist_test = mahalanobis(df['Albania'].to_numpy(), df['Bosnia'].to_numpy(), covariance_matrix_inverse)
print('dist_custom = ', dist_test)
print('dist_scipy =  ', scipy.spatial.distance.mahalanobis(df['Albania'].to_numpy(), df['Bosnia'].to_numpy(),
covariance_matrix_inverse))
'''

k = 4
print('                MAHALANOBIS')
print('NEIGHBOR CLUSTERS')
```

```
neighbor_clusters = [['UK', 'Ireland', 'Spain', 'Monaco', 'Portugal'], \
                     ['Poland', 'Hungary', 'France', 'Netherland', 'Sweden'], \
                     ['Austria', 'Italy', 'Bosnia', 'Bulgaria', 'Macedonia', 'Moldova', 'Romania'], \
                     ['Germany', 'Ukraine', 'Russia', 'Belarus', 'Norway', 'Finland', 'Denmark']]
centers = [cluster_average(neighbor_clusters[i]) for i in range(k)]
clusters_old = clusterize_with_set_centers(df, centers, k)
while True:
    new_centers = [cluster_average(clusters_old[j]) for j in range(len(clusters_old))]
    clusters_new = clusterize_with_set_centers(df, new_centers, k)
    if clusters_new == clusters_old:
        for i in range(k):
            print('Cluster ', i+1, ': ', clusters_new[i])
            distances = radius_and_distances(i+1, new_centers[i], clusters_new[i])
            print(' ')


            pre_dist_df = [distances, clusters_new[i]]
            dist_df = pd.DataFrame(pre_dist_df, columns = [i for i in range(len(clusters_new[i]))])
            #print(dist_df)
            neighbour_method(dist_df, len(clusters_new[i])-1)
            print(' ')
        break
    else:
        clusters_old = copy.deepcopy(clusters_new)


k1 = 7
```

```
print('GEOGRAPHIC CLUSTERS')
geographic_clusters = [['Spain', 'Portugal'], \
                       ['France', 'Belgium', 'Netherland'], \
                       ['UK', 'Ireland'], \
                       ['Germany', 'Austria', 'Switzerland', 'Italy'], \
                       ['Denmark', 'Sweden', 'Norway', 'Finland'], \
                       ['Poland', 'Ukraine', 'Estonia', 'Latvia', 'Lithuania', 'Czech', 'Slovakia', 'Hungary'], \
                       ['Slovenia', 'Croatia', 'Bosnia', 'Serbia', 'Macedonia', 'Albania', 'Greece', 'Romania', 'Moldova']]
centers_g = [cluster_average(geographic_clusters[i]) for i in range(k1)]
clusters_old_g = clusterize_with_set_centers(df, centers_g, k1)
while True:
    new_centers_g = [cluster_average(clusters_old_g[j]) for j in range(len(clusters_old_g))]
    clusters_new_g = clusterize_with_set_centers(df, new_centers_g, k1)
    if clusters_new_g == clusters_old_g:
        for i in range(k1):
            print('Cluster ', i+1, ': ', clusters_new_g[i])
            distances_g = radius_and_distances(i+1, new_centers_g[i], clusters_new_g[i])
            print(' ')
            pre_dist_df_g = [distances_g, clusters_new_g[i]]
            dist_df_g = pd.DataFrame(pre_dist_df_g, columns = [i for i in range(len(clusters_new_g[i]))])
            #print(dist_df)
            if len(clusters_new_g[i]) > 2:
                neighbour_method(dist_df_g, len(clusters_new_g[i])-1)
                print(' ')
            else:
```

```
            print('The cluster consists of two or less elements, therefore the nearest neighbor algorithm is useless and non applicable')
            print(' ')
        break
    else:
        clusters_old_g = copy.deepcopy(clusters_new_g)
```