

Міністерство освіти і науки України  
Харківський національний університет імені В. Н. Каразіна  
Факультет математики і інформатики  
Кафедра прикладної математики

*До захисту допущено*

*кафедрою прикладної математики, протокол №5 від 12 червня 2026 р.*

*завідувач кафедри*

*прикладної математики*

*доктор фіз.-мат. наук, професор*

*Валерій КОРОБОВ*

## **КВАЛІФІКАЦІЙНА РОБОТА**

здобувача першого (бакалаврського) рівня вищої освіти

**«Статистичний аналіз та прогнозування динаміки викидів CO<sub>2</sub> у країнах Європи»**

Спеціальність 113 Прикладна математика

Освітня програма Прикладна математика

Здобувач

Еліна ЗАВІР'ЮХА

Науковий керівник

доктор фіз.-мат. наук, професор,  
професор кафедри прикладної математики  
Наталія КІЗІЛОВА

Харків – 2026

## АНОТАЦІЯ

Кваліфікаційну роботу присвячено аналізу та прогнозуванню викидів вуглекислого газу (CO<sub>2</sub>) у європейських країнах із використанням методів кластерного аналізу та прогнозування часових рядів. Дослідження виконано на основі відкритих статистичних даних платформи Our World in Data за період 2000–2024 років для двадцяти європейських країн.

У роботі розглянуто два показники: викиди CO<sub>2</sub> на душу населення (CO<sub>2</sub> per capita) та загальні викиди CO<sub>2</sub> (CO<sub>2</sub> total). Для забезпечення коректного порівняння даних застосовано стандартизацію за допомогою Z-оцінок. Кластеризацію країн виконано методами K-Means та ієрархічної кластеризації. Якість отриманих кластерів оцінювалася за допомогою коефіцієнта силуєту.

У межах дослідження побудовано прогнози показників CO<sub>2</sub> на 2025-2028 роки за допомогою моделей ARIMA та Holt Exponential Smoothing. Для оцінювання точності прогнозування використано показники MAE, RMSE та MAPE. За результатами порівняння встановлено, що для більшості досліджуваних країн метод Holt Exponential Smoothing забезпечує вищу точність прогнозування, тому його результати були використані для подальшого аналізу.

На основі отриманих прогнозів виконано аналіз прогнозованої кластерної належності країн. Для кожної країни визначено найближчий центр кластера, отриманий методом K-Means. Проведений аналіз дозволив оцінити стабільність сформованих кластерів та виявити країни, для яких у прогнозованому періоді можливі зміни кластерної належності.

Усі етапи обробки даних, кластеризації, прогнозування та аналізу результатів були реалізовані мовою програмування Python.

**Ключові слова:** викиди CO<sub>2</sub>, кластерний аналіз, K-Means, ієрархічна кластеризація, прогнозування, ARIMA, Holt Exponential Smoothing, часові ряди.

## ABSTRACT

### **Zaviriukha Elina. Statistical cluster analysis and forecasting of CO<sub>2</sub> emissions in European countries.**

This qualification thesis is devoted to the analysis and forecasting of carbon dioxide (CO<sub>2</sub>) emissions in European countries using cluster analysis and time series forecasting methods. The study is based on open statistical data from the Our World in Data platform covering the period from 2000 to 2024 for twenty European countries.

The thesis examines two indicators: CO<sub>2</sub> emissions per capita and total CO<sub>2</sub> emissions. To ensure the comparability of the data, standardization using Z-scores was applied. Countries were clustered using the K-Means and hierarchical clustering methods. The quality of the obtained clusters was evaluated using the silhouette coefficient.

Within the study, forecasts of CO<sub>2</sub> indicators for the period 2025–2028 were developed using the ARIMA and Holt Exponential Smoothing models. The forecasting accuracy was assessed using the MAE, RMSE, and MAPE metrics. The comparison of the obtained results showed that the Holt Exponential Smoothing method provided higher forecasting accuracy for most of the analyzed countries; therefore, its results were used for further analysis.

Based on the obtained forecasts, an analysis of the projected cluster membership of the countries was carried out. For each country, the nearest cluster center obtained by the K-Means method was determined. The analysis made it possible to assess the stability of the formed clusters and identify countries for which changes in cluster membership may occur during the forecast period.

All stages of data processing, clustering, forecasting, and result analysis were implemented using the Python programming language.

**Keywords:** CO<sub>2</sub> emissions, cluster analysis, K-Means, hierarchical clustering, forecasting, ARIMA, Holt Exponential Smoothing, time series.

## ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. КЛАСТЕРИЗАЦІЯ ЄВРОПЕЙСЬКИХ КРАЇН ЗА ПОКАЗНИКАМИ ВИКИДІВ CO <sub>2</sub>	
1.1. Дані про викиди CO <sub>2</sub> та їх нормалізація.....	6
1.2. Теоретичні основи кластерного аналізу.....	7
1.3. Метод k-середніх.....	7
1.4. Ієрархічна кластеризація методом найвіддаленішого сусіда.....	8
1.5. Оцінка якості кластеризації методом коефіцієнта силуету.....	9
РОЗДІЛ 2. ПРОГНОЗУВАННЯ ВИКИДІВ CO <sub>2</sub> МЕТОДАМИ ARIMA ТА EXPONENTIAL SMOOTHING	
2.1. Теоретичні основи моделей ARIMA.....	10
2.2. Автокореляція та вибір параметрів ARIMA.....	13
2.3. Побудова та прогнозування моделей ARIMA.....	14
2.4. Метод Holt Exponential Smoothing.....	17
2.5. Аналіз залишків та оцінка точності прогнозування (MAE, RMSE, MAPE).....	19
2.6. Порівняння моделей ARIMA та Exponential Smoothing.....	20
РОЗДІЛ 3. АНАЛІЗ ПРОГНОЗОВАНИХ ЗНАЧЕНЬ У МЕЖАХ СФОРМОВАНИХ КЛАСТЕРІВ	
3.1. Визначення прогнозованої кластерної належності країн.....	21
3.2. Порівняння прогнозних та історичних кластерів.....	22
ВИСНОВКИ.....	23
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	24
ДОДАТКИ.....	26

## ВСТУП

Аналіз даних і прогнозування сьогодні використовуються в багатьох сферах діяльності людини та охоплюють такі напрями, як економіка, екологія, промисловість, фінанси та інформаційні технології. Зростання обсягів статистичної інформації створює потребу у використанні сучасних методів її обробки, які дозволяють виявляти закономірності, досліджувати структуру даних та оцінювати можливі тенденції їх зміни в майбутньому.

Одним із важливих напрямів застосування статистичних методів є дослідження екологічних показників. Особливу увагу привертають викиди вуглекислого газу ( $\text{CO}_2$ ), які є одним із основних чинників зміни клімату. Рівень викидів суттєво відрізняється між країнами та залежить від особливостей економічного розвитку, структури промисловості, чисельності населення та енергетичної політики.

Для дослідження подібних даних широко використовуються методи кластерного аналізу, які дозволяють об'єднувати об'єкти у групи за ступенем їх подібності. Водночас важливим є не лише аналіз наявних даних, а й прогнозування майбутніх значень показників. Поєднання методів кластеризації та прогнозування дає змогу оцінити не тільки поточну структуру даних, а й можливі зміни кластерної належності об'єктів у майбутньому.

У кваліфікаційній роботі досліджуються показники викидів  $\text{CO}_2$  у країнах Європи. Для аналізу використовуються методи кластерного аналізу та прогнозування часових рядів. Практична частина роботи включає побудову кластерів, оцінювання їх якості, прогнозування показників  $\text{CO}_2$  та визначення прогнозної кластерної належності країн на основі отриманих прогнозів.

# РОЗДІЛ 1. КЛАСТЕРИЗАЦІЯ ЄВРОПЕЙСЬКИХ КРАЇН ЗА ПОКАЗНИКАМИ ВИКИДІВ CO<sub>2</sub>

## 1.1. Дані про викиди CO<sub>2</sub> та їх нормалізація

У дослідженні використано відкриті статистичні дані платформи *Our World in Data* [1], що містять інформацію про викиди CO<sub>2</sub> у різних країнах світу. Для проведення порівняльного аналізу було обрано 20 європейських країн.

У роботі розглядаються два показники: CO<sub>2</sub> per capita (викиди на душу населення) та CO<sub>2</sub> per country (загальний обсяг викидів). Дані охоплюють період 2000-2024 років, що дозволяє простежити зміни показників у часі та порівняти країни між собою.

Оскільки масштаб цих двох показників суттєво відрізняється (загальні викиди вимірюються в мільйонах тонн, тоді як per capita - у значно менших значеннях), їх необхідно привести до порівняльного вигляду. Для цього застосовано нормалізацію, що дає змогу коректно використовувати обидва показники у межах одного кластерного аналізу.

Використаємо метод нормалізації Z-оцінок, який полягає у перетворенні кожного значення показника відповідно до його відхилення від середнього та величини стандартного відхилення.

У результаті всі значення подаються у спільній порівняльній шкалі, де:

- середньому значенню відповідає Z-оцінка 0;
- значення, менші за середнє, мають негативні Z-оцінки;
- значення, більші за середнє - позитивні.

Формула стандартизації Z-оцінок:

$$Z = \frac{X - \text{mean}(X)}{\text{SD}(X)} \quad (1.1)$$

де:

$X$  - початкове (вихідне) значення показника (наприклад, рівень CO<sub>2</sub> для певної країни);

$\text{mean}(X)$  - середнє значення показника у вибірці (середній рівень CO<sub>2</sub> серед усіх країн);

$SD(X)$  - стандартне відхилення показника (міра того, наскільки значення  $CO_2$  розкидані відносно середнього) [2].

## 1.2. Теоретичні основи кластерного аналізу

Кластеризація - це статистична процедура, метою якої є поділ множини об'єктів  $X = \{x_1, x_2 \dots x_n\}$  на ціле число неперетинних підмножин  $C_1, C_2, \dots, C_k$ , які називаються кластерами, тобто:

1. Повнота покриття множини:  $C_1 \cup C_2 \cup \dots \cup C_k = X$
2. Неперетинність кластерів:  $C_i \cap C_j = \emptyset, i \neq j$
3. Унікальна належність кожного об'єкта:  $x_n \in C_k$  для рівно одного  $k$

Основною вимогою кластерного аналізу є те, щоб об'єкти, що належать одному кластеру, були максимально подібними між собою, тоді як об'єкти з різних кластерів - максимально відмінними.

Подібність або відмінність визначається за допомогою певної міри відстані або схожості, залежно від природи даних [3].

## 1.3. Метод k-середніх

Метод k-середніх належить до прототипних алгоритмів кластеризації, які виконують однорівневе розбиття даних на кластери. Існує велика кількість його різновидів, проте в цьому дослідженні використовується класична версія, що базується на мінімізації відстаней до центроїдів та евклідовій метриці. Такий варіант є коректним при кластеризації числових характеристик, зокрема показників  $CO_2$ .

Нехай маємо множину спостережень:  $X = \{x_1, x_2 \dots x_n\}$ , де кожне спостереження - це точка у двовимірному просторі ознак ( $CO_2$  per capita та  $CO_2$  per country).

Мета алгоритму - розбити множину  $X$  на  $k$  кластерів:  $C_1, C_2, \dots, C_k$  та знайти центроїди цих кластерів.

### Алгоритм:

- Вибрати  $k$  початкових центроїдів шляхом випадкового вибору  $k$  спостережень із множини даних.

- Для кожного об'єкта  $x_i$  визначити найближчий центр за евклідовою відстанню:

$$p(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1.2)$$

Ітерація присвоєння завершується тоді, коли всі точки розподілені по кластерах.

- Для кожного кластера обчислити новий центроїд за формулою середнього:

$$y = \frac{1}{m_k} \sum_{x \in C_k} x, \quad (1.3)$$

де  $C_k$  – кластер;  $y$  – його новий центроїд;  $m_k$  – кількість точок, що належать кластеру  $C_k$ .

- Повторити кроки 2-3, доки центри кластерів не перестануть змінюватися (або зміни стануть незначними) [4].

#### 1.4. Ієрархічна кластеризація методом найвіддаленішого сусіда

Алгоритми кластеризації поділяються на ієрархічні та неієрархічні.

В ієрархічних методах формується дендрограма - дерево послідовних об'єднань.

Ієрархічні методи бувають двох типів:

- *Агломеративні* - кожен об'єкт починає як окремий кластер, а на кожному кроці об'єднуються два кластери з найменшою міжкластерною відстанню.
- *Дивізивні* - усі об'єкти спочатку утворюють один кластер, який рекурсивно розділяється на менші.

У цьому дослідженні застосовуємо агломеративний підхід, зокрема метод найвіддаленішого сусіда. Цей метод визначає відстань між двома кластерами як максимальну відстань між будь-якими двома їх елементами.

Відстань між кластерами  $G$  і  $H$  задається як:

$$d_{\text{complete}}(G, H) = \max_{i \in G, j \in H} d_{ij} \quad (1.4)$$

де  $d_{ij}$  - відстань між об'єктами  $i$  та  $j$ , визначена за евклідовою метрикою.

Таким чином, на кожному кроці об'єднуються ті два кластери, для яких значення  $d_{complete}(G, H)$  є мінімальним серед усіх пар кластерів [5].

### 1.5. Оцінка якості кластеризації методом коефіцієнта силуету

Для оцінювання отриманих кластерів у дослідженні використано метод коефіцієнта силуету. Цей метод дозволяє кількісно визначити, наскільки добре об'єкти вписуються у свої кластери, порівнюючи їхню внутрішньокластерну та міжкластерну віддаленість.

Нехай об'єкт  $i$  належить кластеру  $A$ . Тоді:

- Внутрішньокластерна відстань

$a(i) = \text{average}_{j \in A, j \neq i} d(i, j)$ , тобто середня відстань між об'єктом  $i$  та всіма іншими елементами його кластера  $A$ .

- Найближча міжкластерна відстань

Для кожного іншого кластера  $C \neq A$ . обчислюється:

$$d(i, C) = \text{average}_{j \in C} d(i, j),$$

Потім обирається мінімальне значення:  $b(i) = \min_{C \neq A} d(i, C)$  яке є мінімальною середньою міжкластерною відстанню від об'єкта  $i$  до будь-якого кластера  $C \neq A$ .

- Коефіцієнт силуету

Використовуючи значення внутрішньокластерної відстані  $a(i)$  та найближчої міжкластерної відстані  $b(i)$ , силует для спостереження  $i$  визначають як:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (1.5)$$

- Інтерпретація

Коефіцієнт силуету задовольняє умову:  $-1 \leq s(i) \leq 1$

- 1)  $s(i) \approx 1$ : об'єкт добре вписаний у свій кластер; кластер компактний і добре відокремлений.
- 2)  $s(i) \approx 0$ : об'єкт лежить на межі між кластерами.
- 3)  $s(i) < 0$ : об'єкт, імовірно, помилково віднесений до кластера.

Для оцінювання якості всієї кластеризації використовується середнє значення силуетів:

$$S = \frac{1}{n} \sum_{i=1}^n s(i) \quad (1.6)$$

Чим вище  $S$ , тим якіснішою вважається кластеризація [6].

## РОЗДІЛ 2. ПРОГНОЗУВАННЯ ВИКИДІВ CO<sub>2</sub> МЕТОДАМИ ARIMA ТА EXPONENTIAL SMOOTHING

### 2.1. Теоретичні основи моделей ARIMA

**ARIMA (AutoRegressive Integrated Moving Average)** є одним із найпоширеніших методів аналізу та прогнозування часових рядів. Вона використовується для дослідження даних, упорядкованих у часі, та виявлення закономірностей, які можуть бути використані для оцінки майбутньої поведінки досліджуваного показника.

Модель ARIMA враховує типові структури, що зустрічаються в часових рядах, і поєднує відносну простоту побудови з можливістю описувати широкий спектр часових процесів. Назва ARIMA є скороченням від *AutoRegressive Integrated Moving Average* і відображає три основні складові моделі: авторегресію (AR), інтегрування або диференціювання (I) та ковзне середнє (MA).

Поєднання цих компонентів дає змогу враховувати залежність поточних значень від попередніх спостережень, усувати нестационарність ряду за допомогою диференціювання та описувати випадкові коливання.

**Авторегресія (AR, AutoRegression)** є складовою моделі ARIMA, яка описує залежність поточного значення часового ряду від його попередніх значень. У такій моделі минулі спостереження використовуються як предиктори для пояснення поточного значення ряду.

Авторегресійна модель порядку  $p$  враховує  $p$  попередніх значень часового ряду. Загальний вигляд авторегресійної моделі можна подати у вигляді:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + e_t \quad (2.1)$$

де  $y_t$  - значення часового ряду в момент часу  $t$ ,  $c$  - константа,  $\phi_1, \phi_2, \dots, \phi_p$  - параметри моделі, а  $e_t$  - випадкова похибка (білий шум).

Таким чином, поточне значення ряду визначається як комбінація кількох попередніх значень та випадкової складової. Чим більший порядок  $p$ , тим більше попередніх спостережень враховується моделлю.

У найпростішому випадку використовується модель **AR(1)**, у якій поточне значення ряду визначається одним попереднім значенням та випадковою похибкою:

$$y_t = c + \phi_1 y_{t-1} + e_t \quad (2.2)$$

Така модель є базовим випадком авторегресійної моделі та використовується для опису залежності між послідовними значеннями часового ряду.

**Диференціювання (Integrated, I).** Компонент **I (Integrated)** у моделі ARIMA пов'язаний із застосуванням диференціювання до початкових спостережень. Диференціювання полягає у відніманні від поточного значення часового ряду його попереднього значення. Така процедура використовується для перетворення часового ряду у стаціонарний вигляд.

Необхідність диференціювання зумовлена тим, що багато часових рядів містять тренд або інші зміни рівня, які ускладнюють побудову моделі. Виконання диференціювання дозволяє усунути ці зміни та зробити статистичні характеристики ряду більш стабільними в часі.

У моделі ARIMA параметр  $d$  визначає кількість разів, яку застосовується операція диференціювання. Цей параметр називають порядком або ступенем диференціювання. Якщо  $d=0$ , диференціювання не виконується; якщо  $d=1$ , використовується перша різниця; якщо  $d=2$ , застосовується повторне диференціювання вже до отриманого ряду.

У записі моделі ARIMA процес диференціювання задається оператором:

$$(1 - B)^d y_t, \quad (2.3)$$

де  $B$  - оператор зсуву (backshift operator), для якого  $By_t = y_{t-1}$ .

Відповідно, після застосування диференціювання ряд  $(1 - B)^d y_t$  описується моделлю ARMA. Таким чином, модель ARIMA поєднує авторегресійну складову та ковзне середнє з попереднім перетворенням даних за допомогою диференціювання.

### **Ковзне середнє (Moving Average, MA)**

Складова MA (Moving Average) враховує вплив випадкових похибок попередніх періодів на поточне значення часового ряду. На відміну від авторегресійної частини, де використовуються минулі значення самого показника, тут основну роль відіграють залишкові помилки, отримані в попередні моменти часу.

Модель ковзного середнього порядку  $q$  можна подати у вигляді:

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q}, \quad (2.4)$$

де  $e_t$  є випадковою похибкою (білим шумом), а  $\theta_1, \theta_2, \dots, \theta_q$  - параметрами моделі.

Параметр  $q$  визначає кількість попередніх похибок, які враховуються при формуванні поточного значення ряду. Таким чином, модель описує короткостроковий вплив випадкових відхилень, що виникали в минулому.

Для моделей ковзного середнього важливою є властивість **оборотності (invertibility)**. Вона означає, що процес MA( $q$ ) за певних умов може бути представлений через нескінченний авторегресійний процес. Наявність цієї властивості спрощує оцінювання параметрів та практичне застосування моделі.

У моделі ARIMA складова MA поєднується з авторегресійною частиною та операцією диференціювання. У результаті під час аналізу враховуються як попередні значення часового ряду, так і похибки минулих періодів, що дозволяє точніше відобразити внутрішню структуру часових даних.

Таким чином, для опису конкретної моделі використовується запис ARIMA( $p, d, q$ ), де параметр  $p$  задає порядок авторегресійної складової,  $d$  характеризує ступінь диференціювання, а  $q$  визначає порядок компоненти

ковзного середнього. Поєднання цих параметрів дає змогу адаптувати модель до особливостей конкретного часового ряду [7][8, с.72-75].

## 2.2. Автокореляція та вибір параметрів ARIMA

Перед побудовою моделі ARIMA доцільно проаналізувати автокореляційну структуру часового ряду. Для цього використовуються графіки ACF та PACF, які відображають залежність між поточними та попередніми значеннями ряду.

Автокореляційна функція ACF (Autocorrelation Function) показує загальну кореляцію між значенням  $y_t$  та його лаговими значеннями  $y_{t-k}$ . Вона дає змогу оцінити, наскільки зв'язок між спостереженнями зберігається через один, два або більше часових кроків.

Часткова автокореляційна функція PACF (Partial Autocorrelation Function) характеризує зв'язок між  $y_t$  та  $y_{t-k}$  після усунення впливу проміжних лагів  $1, 2, \dots, k - 1$ . Таким чином, PACF дозволяє визначити прямий вплив конкретного попереднього значення на поточне спостереження.

У моделі ARIMA графіки ACF та PACF традиційно використовуються для попереднього визначення її структури. За графіком PACF можна орієнтовно оцінити порядок авторегресійної складової  $p$ , а за графіком ACF - порядок складової ковзного середнього  $q$ . Якщо після певного лагу значення перестають бути статистично значущими, відповідний лаг може розглядатися як можливий порядок моделі.

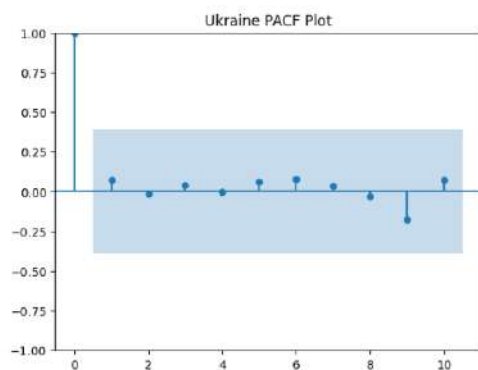


Рисунок 2.1 - Часткова автокореляційна функція (PACF) часового ряду викидів  $CO_2$  для України

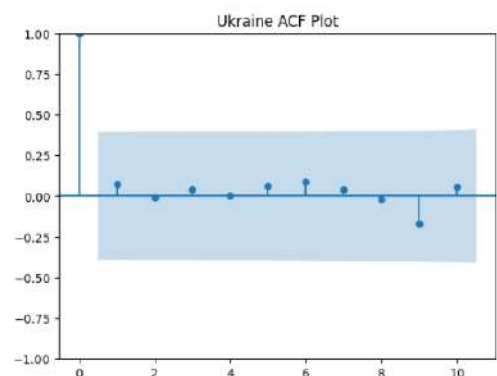


Рисунок 2.2 – Автокореляційна функція (ACF) часового ряду викидів  $CO_2$  для України

У даному дослідженні графіки ACF та PACF використовуються для аналізу часової структури даних.

Після попереднього аналізу часових рядів виконується автоматичний підбір моделі ARIMA. Для цього розглядаються різні комбінації параметрів  $p$ ,  $d$  та  $q$ , а оцінювання коефіцієнтів здійснюється методом максимальної правдоподібності (Maximum Likelihood Estimation, MLE).

Для порівняння отриманих моделей використовується інформаційний критерій Акаїке (Akaike Information Criterion, AIC). Цей критерій враховує як якість узгодження моделі з наявними даними, так і кількість параметрів, що входять до її складу. Завдяки цьому досягається баланс між точністю опису часового ряду та складністю моделі.

Значення AIC обчислюється за формулою

$$AIC = -2 \log(L) + 2(p + q + k + 1), \quad (2.5)$$

де  $L$  - значення функції правдоподібності моделі,  $p$  - порядок авторегресійної складової,  $q$  - порядок ковзного середнього, а  $k$  враховує наявність константи в моделі [8, с.76-77].

### 2.3. Побудова та прогнозування моделей ARIMA

**Стаціонарність** є однією з основних властивостей часових рядів. Вона означає, що статистичні характеристики процесу не змінюються з часом. Для стаціонарного ряду математичне сподівання та дисперсія залишаються сталими, а коваріація між двома спостереженнями залежить лише від відстані між ними (лагу), а не від конкретного моменту часу.

Розрізняють строгу та слабку стаціонарність. Строго стаціонарний процес має однакові ймовірнісні властивості після будь-якого зсуву в часі.

У практичних дослідженнях частіше використовується поняття слабкої (або коваріаційної) стаціонарності.

Часовий ряд  $Y_t$  є слабо стаціонарним, якщо виконуються такі умови:

1. Середнє значення є сталим:  $E(Y_t) = \mu$
2. Дисперсія не залежить від часу:  $Var(Y_t) = \sigma^2$
3. Коваріація залежить лише від лагу  $k$ :  $\gamma_k = Cov(Y_t, Y_{t-k})$

Для стаціонарного процесу автокореляційна функція визначається як

$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad (2.6)$$

де  $\gamma_0$  – дисперсія ряду.

Одним із найважливіших прикладів стаціонарного процесу є білий шум (White Noise). Для нього середнє значення є сталим, а автокореляція відсутня для всіх лагів, окрім нульового [9].

Стаціонарність є важливою передумовою для побудови моделей прогнозування, зокрема моделей сімейства ARIMA, оскільки більшість статистичних методів передбачають незмінність характеристик ряду протягом часу.

### **Тест Дікі–Фуллера та диференціювання часових рядів**

На практиці багато економічних та екологічних часових рядів є нестаціонарними через наявність тренду. Одним із найпоширеніших прикладів є випадкове блукання (Random Walk):

$$y_t = y_{t-1} + \varepsilon_t, \quad (2.7)$$

де  $\varepsilon_t$  випадкова похибка.

Такий процес містить одиничний корінь (Unit Root) і є нестаціонарним. Для перевірки наявності одиничного кореня використовується тест Дікі–Фуллера (Dickey-Fuller Test).

Для моделі

$$y_t = \phi y_{t-1} + \varepsilon_t \quad (2.8)$$

гіпотези тесту мають вигляд:

$H_0: \phi = 1$  (ряд має одиничний корінь і є нестаціонарним)

$H_1: \phi < 1$  (ряд є стаціонарним)

У практичних застосуваннях найчастіше використовується розширений тест Дікі–Фуллера (Augmented Dickey–Fuller, ADF), який враховує додаткові лагові залежності та підходить для ширшого класу часових рядів.

Результат тесту оцінюється за  $p$ -value:

- $p < 0.05$  - нульова гіпотеза відхиляється, ряд вважається стаціонарним;
- $p \geq 0.05$  - нульова гіпотеза не відхиляється, ряд вважається нестаціонарним.

Якщо ряд є нестаціонарним, для його перетворення часто застосовується диференціювання. Перша різниця визначається як

$$\Delta y_t = y_t - y_{t-1}. \quad (2.9)$$

Для випадкового блукання  $y_t = y_{t-1} + \varepsilon_t$  після диференціювання отримуємо

$$\Delta y_t = \varepsilon_t, \quad (2.10)$$

тобто стаціонарний процес білого шуму.

Якщо для досягнення стаціонарності необхідно виконати одне диференціювання, ряд називають інтегрованим першого порядку та позначають  $I(1)$ . Якщо потрібно виконати  $d$  диференціювань, ряд називають інтегрованим порядку  $I(d)$  [10].

### **Довірчі інтервали прогнозу**

Для оцінки невизначеності прогнозу в моделях ARIMA використовуються довірчі інтервали прогнозу. Найчастіше застосовується 95%-й довірчий інтервал, який визначається за формулою

$$\hat{y}_{T+h|T} \pm 1.96\sqrt{v_{T+h|T}}, \quad (2.11)$$

де  $\hat{y}_{T+h|T}$  - прогнозоване значення, а  $v_{T+h|T}$  - оцінена дисперсія прогнозу.

Для моделей ARIMA довірчі інтервали залежать від дисперсії прогнозу та параметрів моделі. У випадку багатокрокового прогнозування розмір довірчого інтервалу збільшується зі зростанням горизонту прогнозування. Це пов'язано з накопиченням невизначеності під час прогнозування майбутніх значень часового ряду.

Розрахунок довірчих інтервалів ґрунтується на припущенні, що залишки моделі є некорельованими та мають нормальний розподіл. Водночас отримані інтервали можуть бути дещо звуженими, оскільки при їх побудові зазвичай не

враховується невизначеність оцінювання параметрів моделі. Крім того, передбачається, що закономірності, виявлені в історичних даних, зберігатимуться протягом прогнозного періоду, а майбутні похибки залишатимуться некорельованими [8, с.82].

#### 2.4. Метод Holt Exponential Smoothing

Експоненціальне згладжування (Exponential Smoothing) є одним із найпоширеніших методів аналізу та прогнозування часових рядів. Основна ідея методу полягає у згладжуванні вихідного ряду даних для побудови прогнозів майбутніх значень. На відміну від звичайного ковзного середнього, експоненціальне згладжування надає більшої ваги останнім спостереженням, тоді як вплив старіших даних поступово зменшується за експоненціальним законом. Завдяки цьому модель швидше реагує на зміни в часовому ряді та забезпечує більш актуальні прогнози. Прогноз формується на основі експоненціально зваженого середнього попередніх спостережень, де найбільшу вагу мають найновіші значення ряду, а вага більш ранніх спостережень поступово зменшується [11].

У найпростішому варіанті експоненціального згладжування прогноз базується на оцінці поточного рівня ряду (level). Рівень характеризує середнє значення показника на певний момент часу та оновлюється після надходження кожного нового спостереження. Для простого експоненціального згладжування оцінка рівня визначається як зважена комбінація поточного значення ряду та попередньої оцінки рівня:

$$l_t = \alpha y_t + (1 - \alpha)l_{t-1} \quad (2.12)$$

де:

- $y_t$  - фактичне значення ряду в момент часу  $t$ ;
- $l_t$  - оцінка рівня ряду;
- $\alpha$  - параметр згладжування,  $0 \leq \alpha \leq 1$ .

Параметр  $\alpha$  визначає, наскільки сильно нові спостереження впливають на оновлення моделі. Великі значення  $\alpha$  надають більшої ваги останнім даним, тоді

як малі значення забезпечують більш плавне згладжування та зменшують вплив випадкових коливань.

Для часових рядів, у яких спостерігається стійка тенденція до зростання або зниження, простого експоненціального згладжування недостатньо. У таких випадках використовується метод Холта (Holt Exponential Smoothing), який додатково враховує компоненту тренду. Тренд відображає напрямок і швидкість зміни показника в часі.

У методі Холта прогноз формується з використанням двох компонентів: рівня та тренду. Прогноз на  $h$  кроків уперед визначається як:

$$\hat{y}_{t+h|t} = l_t + hb_t \quad (2.13)$$

де  $b_t$  характеризує оцінку тренду в момент часу  $t$ .

Оновлення рівня та тренду здійснюється за такими рівняннями:

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (2.14)$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \quad (2.15)$$

де:

- $l_t$  - поточна оцінка рівня;
- $b_t$  - поточна оцінка тренду;
- $\alpha$  - параметр згладжування рівня;
- $\beta$  - параметр згладжування тренду.

Параметр  $\beta$  визначає швидкість адаптації тренду до нових даних. Якщо значення  $\beta$  є високим, модель швидше реагує на зміни тенденції. За низьких значень  $\beta$  оцінка тренду змінюється більш плавно [8, с.34-38].

У даній роботі використовується адитивний тренд (additive trend), який передбачає, що зміна показника з часом відбувається приблизно на однакову абсолютну величину. Такий підхід є доцільним для аналізу річних даних про викиди CO<sub>2</sub>, оскільки дозволяє враховувати довгострокову тенденцію зміни показників та будувати прогнози на наступні роки.

## 2.5. Аналіз залишків та оцінка точності прогнозування (MAE, RMSE, MAPE)

Після побудови моделі Holt Exponential Smoothing були розраховані оцінені значення (fitted values), які є результатом апроксимації вихідного часового ряду моделлю. Вони використовуються для оцінки того, наскільки добре модель відтворює наявні дані.

Для оцінки якості моделі було побудовано графік Actual vs Fitted. Чим ближче точки розташовані до діагональної лінії, тим меншою є різниця між фактичними та оціненими значеннями і тим точніше модель описує досліджуваний часовий ряд.

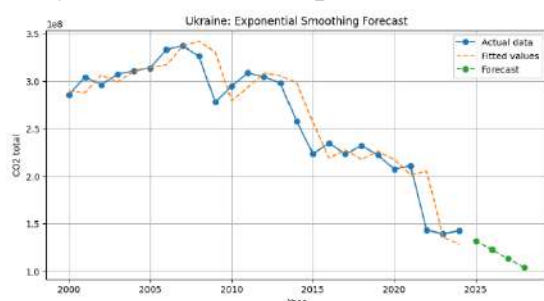


Рисунок 2.3 - Реальні, оцінені та прогнозовані значення загальних викидів CO<sub>2</sub> для України за моделлю Holt Exponential Smoothing

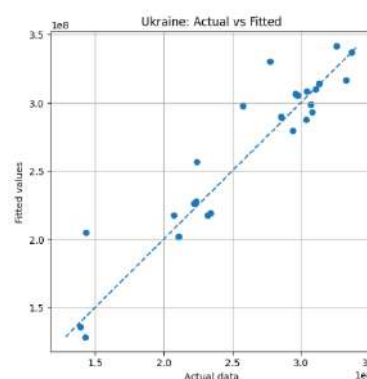


Рисунок 2.4 - Співвідношення фактичних та оцінених значень для України

Відповідні графіки прогнозу для інших країн подано в Додатку 3.

Для перевірки адекватності моделі було виконано аналіз залишків (Residual Analysis). Залишок визначається як різниця між фактичним та оціненим значенням показника:

$$e_t = y_t - \hat{y}_t \quad (2.16)$$

де:

$y_t$  - фактичне значення показника;

$\hat{y}_t$  - оцінене моделлю значення;

$e_t$  - залишок.

Аналіз залишків дозволяє оцінити, наскільки добре модель описує часовий ряд. Якщо модель є адекватною, залишки повинні випадково коливатися навколо нуля та не містити виражених закономірностей або трендів [12].



Рисунок 2.5 - Графік залишків моделі ARIMA для України



Рисунок 2.6 - Графік залишків моделі Holt Exponential Smoothing для України

Для кількісної оцінки точності прогнозування використано метрики MAE, RMSE та MAPE.

Середня абсолютна похибка (MAE) визначається за формулою:

$$MAE = \frac{1}{n} \sum_{t=1}^n |e_t| \quad (2.17)$$

Дана метрика показує середню величину відхилення прогнозованих значень від фактичних і вимірюється в тих самих одиницях, що й вихідні дані.

Середньоквадратична похибка (RMSE) обчислюється як:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2} \quad (2.18)$$

Ця метрика більш чутлива до великих помилок, оскільки відхилення підносяться до квадрату.

Середня абсолютна відносна похибка (MAPE) визначається за формулою:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{e_t}{y_t} \right| \quad (2.19)$$

На відміну від попередніх показників, MAPE подається у відсотках і дозволяє порівнювати точність прогнозування для різних часових рядів [13].

## 2.6. Порівняння моделей ARIMA та Exponential Smoothing

Для порівняння моделей ARIMA та Holt Exponential Smoothing було використано показник MAPE (Mean Absolute Percentage Error). Оскільки ця метрика подається у відсотках, вона є зручною для порівняння точності моделей між різними країнами та показниками незалежно від масштабу даних.

У даній роботі значення MAPE розраховувалися на тестовому періоді 2020–2024 років шляхом порівняння фактичних та прогнозованих значень. Модель з меншим значенням MAPE вважалася такою, що забезпечує більш точне прогнозування відповідного часового ряду.

*Детальні результати прогнозування на 2025–2028 роки та значення метрик MAE, RMSE і MAPE наведені в Додатку 2.2 та 2.3.*

Країна	Краща модель для CO <sub>2</sub> per capita	Краща модель для CO <sub>2</sub> total
Бельгія	Holt Exponential Smoothing	Holt Exponential Smoothing
Хорватія	Holt Exponential Smoothing	Holt Exponential Smoothing
Чехія	Holt Exponential Smoothing	ARIMA
Данія	ARIMA	Holt Exponential Smoothing
Фінляндія	Holt Exponential Smoothing	Holt Exponential Smoothing
Франція	ARIMA	ARIMA
Німеччина	ARIMA	ARIMA
Греція	Holt Exponential Smoothing	Holt Exponential Smoothing
Угорщина	Holt Exponential Smoothing	Holt Exponential Smoothing
Ірландія	ARIMA	Holt Exponential Smoothing
Італія	Holt Exponential Smoothing	Holt Exponential Smoothing
Нідерланди	ARIMA	ARIMA
Норвегія	ARIMA	ARIMA
Польща	Holt Exponential Smoothing	Holt Exponential Smoothing
Португалія	Holt Exponential Smoothing	ARIMA
Словаччина	Holt Exponential Smoothing	Holt Exponential Smoothing
Іспанія	Holt Exponential Smoothing	Holt Exponential Smoothing
Швеція	Holt Exponential Smoothing	ARIMA
Україна	Holt Exponential Smoothing	ARIMA
Велика Британія	Holt Exponential Smoothing	ARIMA

*Таблиця 2.1 - Порівняння моделей ARIMA та Holt Exponential Smoothing за найменшим значенням MAPE*

## РОЗДІЛ 3. АНАЛІЗ ПРОГНОЗОВАНИХ ЗНАЧЕНЬ У МЕЖАХ СФОРМОВАНИХ КЛАСТЕРІВ

### 3.1. Визначення прогнозної кластерної належності країн

Після побудови кластерів на основі історичних даних за 2000–2024 роки було виконано аналіз прогнозованих значень викидів CO<sub>2</sub> для періоду 2025-2028 років. Для прогнозування було використано метод експоненціального згладжування (Exponential Smoothing), оскільки він продемонстрував кращі

показники точності порівняно з моделлю ARIMA для більшості досліджуваних країн.

Для кожної країни були отримані прогностні значення двох показників: викидів CO<sub>2</sub> на душу населення (CO<sub>2</sub> per capita) та загальних викидів CO<sub>2</sub> (CO<sub>2</sub> total). Оскільки прогноз будувався окремо для кожного року, на наступному етапі було обчислено середні прогностні значення за період 2025–2028 років.

Далі прогностні значення були приведені до того самого масштабу, що й дані, використані під час кластерного аналізу. Процедура нормалізації та відповідні формули наведені в підрозділі 1.1.

Для визначення прогностної належності кожної країни використовувалися центри кластерів, отримані методом K-Means. Країна відносилася до того кластера, центр якого знаходився найближче до її прогностної позиції у просторі ознак. Визначення відстані між об'єктами здійснювалося за допомогою евклідової відстані, описаної в підрозділі 1.2.

### **3.2. Порівняння прогностних та історичних кластерів**

Після визначення прогностного кластера результати були порівняні з історичними кластерами, сформованими за даними 2000–2024 років. Таке порівняння дозволило оцінити стабільність кластерної структури та виявити країни, для яких прогнозовані значення можуть призвести до переходу в інший кластер.

Якщо історичний та прогностний кластери збігалися, вважалося, що країна зберігає свою кластерну належність у прогнозованому періоді. У випадку розбіжності кластерів робився висновок про можливу зміну позиції країни відносно інших досліджуваних держав за показниками викидів CO<sub>2</sub>.

Отримані результати дають змогу оцінити не лише поточну структуру кластерів, а й потенційні зміни у майбутньому, що можуть бути пов'язані зі зміною динаміки викидів CO<sub>2</sub>.

*Результати кластеризації прогностних значень наведені в Додатку 2.4.*

## ВИСНОВКИ

У кваліфікаційній роботі проведено статистичний аналіз викидів CO<sub>2</sub> у двадцяти європейських країнах на основі даних платформи Our World in Data за період 2000–2024 років. Для дослідження використовувалися два показники: викиди CO<sub>2</sub> на душу населення (CO<sub>2</sub> per capita) та загальні викиди CO<sub>2</sub> (CO<sub>2</sub> total).

У результаті кластерного аналізу було виконано групування країн за допомогою методу K-Means та ієрархічної кластеризації методом найвіддаленішого сусіда. Обидва методи сформували подібну структуру кластерів. Методи виділили чотири кластери, при цьому Німеччина утворила окремий кластер через найвищі загальні викиди CO<sub>2</sub>. Порівняння результатів показало, що відмінності між методами стосуються лише Данії та Греції. Значення середнього коефіцієнта силуету 0,533 для методу K-Means та 0,460 для ієрархічної кластеризації свідчать про те, що метод K-Means забезпечує більш якісне розділення країн на кластери.

Для прогнозування показників CO<sub>2</sub> були досліджені моделі ARIMA та Holt Exponential Smoothing. Оцінювання точності прогнозів за показниками MAE, RMSE та MAPE показало, що для більшості досліджуваних країн метод Holt Exponential Smoothing забезпечує кращі результати. Тому саме ця модель була використана для подальшого аналізу прогнозованої кластерної належності.

На основі прогнозованих значень викидів CO<sub>2</sub> на 2025–2028 роки було визначено прогнозні кластери країн. Для цього прогнозні значення були приведені до масштабу історичних даних та порівняні з центрами кластерів, отриманими методом K-Means. Проведене порівняння історичних і прогнозних кластерів показало, що частина країн зберігає свою кластерну належність, тоді як для інших можливий перехід до іншого кластера. Це свідчить про потенційні зміни у структурі розподілу країн за рівнем викидів CO<sub>2</sub> у найближчі роки.

Основні результати дослідження були представлені на науковій конференції [14]. Також робота брала участь у Всеукраїнському конкурсі студентських наукових робіт у 2025–2026 навчальному році (спеціальність «Прикладна математика»), де була відзначена дипломом III ступеня.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Our World in Data. CO<sub>2</sub> and Greenhouse Gas Emissions. URL: <https://ourworldindata.org/co2-and-greenhouse-gas-emissions> (last accessed:15.05.2026).
2. Larose D. Discovering Knowledge in Data. Wiley, 2005. P.37, P.149.
3. КНУ ім. Тараса Шевченка, факультет комп'ютерних наук та кібернетики. Основні поняття кластеризації та постановка задачі: Лекція №11, с.1. URL: [https://csc.knu.ua/media/study/asp/mod\\_probl\\_inf\\_tech\\_sys\\_analysis\\_ivohin/lecture/lec11.pdf](https://csc.knu.ua/media/study/asp/mod_probl_inf_tech_sys_analysis_ivohin/lecture/lec11.pdf) (дата звернення:10.04.2026).
4. Pan-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. Introduction to Data Mining (Second Edition). 2021. Chapter 7 P. 534-538. URL: <https://www-users.cse.umn.edu/~kumar001/dmbook/index.php> (last accessed:23.04.2026).
5. Tibshirani R. Hierarchical clustering. Data Mining, 2013. P. 10. URL: <https://www.stat.cmu.edu/~ryantibs/datamining/lectures/05-clus2.pdf> (last accessed:05.05.2026).
6. Rousseeuw P.J. Silhouettes: A graphical aid to cluster analysis, 1987. Pp. 55-57. URL: <https://wis.kuleuven.be/stat/robust/papers/publications-1987/rousseeuw-silhouettes-jcam-sciencedirectopenarchiv.pdf> (last accessed:05.05.2026).
7. Brownlee J. Introduction to Time Series Forecasting with Python : How to Prepare Data and Develop Models to Predict the Future. Machine Learning Mastery, 2020. Pp. 208-209. URL: [https://www.inf.szte.hu/~korosig/teach/books/Jason%20Brownlee%20-%20Introduction%20to%20Time%20Series%20Forecasting%20with%20Python%20-%20How%20to%20Prepare%20Data%20and%20Develop%20Models%20to%20Predict%20the%20Future-v1.9%20\(2020\).pdf](https://www.inf.szte.hu/~korosig/teach/books/Jason%20Brownlee%20-%20Introduction%20to%20Time%20Series%20Forecasting%20with%20Python%20-%20How%20to%20Prepare%20Data%20and%20Develop%20Models%20to%20Predict%20the%20Future-v1.9%20(2020).pdf) (last accessed:10.05.2026).
8. Hyndman R. J. Forecasting: Principles & Practice. 2014. Pp. 34-38, 72-77, 82. URL: <https://robjhyndman.com/uwafiles/fpp-notes.pdf>

- (last accessed:10.05.2026).
9. Cryer J. D., Chan K.-S. Time Series Analysis with Applications in R. 2nd ed. 2008. Pp. 16-19. URL:  
<https://mybiostats.wordpress.com/wp-content/uploads/2015/03/time-series-analysis-with-applications-in-r-cryer-and-chan.pdf> (last accessed:14.05.2026).
  10. Unit Root & Augmented Dickey-Fuller (ADF) Test. Stony Brook University. URL: [https://www.ams.sunysb.edu/~zhu/ams586/UnitRoot\\_ADF.pdf](https://www.ams.sunysb.edu/~zhu/ams586/UnitRoot_ADF.pdf) (last accessed:17.05.2026).
  11. Kumar K. Econometric Modeling and Forecasting. Exponential Smoothing. P. 1. URL:  
[https://www.du.ac.in/du/uploads/departments/Operational%20Research/24042020\\_Course%20MOR204\\_Econometric%20Modeling%20and%20Forecasting\\_Kaushal%20Kumar.pdf](https://www.du.ac.in/du/uploads/departments/Operational%20Research/24042020_Course%20MOR204_Econometric%20Modeling%20and%20Forecasting_Kaushal%20Kumar.pdf) (last accessed:19.05.2026).
  12. Kim H.-Y. Statistical notes for clinical researchers: simple linear regression 3-residual analysis. Restorative Dentistry & Endodontics. 2019. Pp. 2-4. URL:  
<https://rde.ac/upload/pdf/rde-44-e11.pdf> (last accessed:20.05.2026).
  13. Hyndman R. J., Athanasopoulos G. Forecast accuracy. Forecasting: Principles and Practice. URL: <https://otexts.com/fpp3/accuracy.html> (last accessed:23.05.2026).
  14. Zaviriukha E.D. MULTICOMPONENT ANALYSIS OF TIME SERIES BASED ON CO<sub>2</sub> EMISSION DATA. *Сучасні проблеми математики та її застосування в природничих науках та інформаційних технологіях: тези доповідей XX Міжнародної науково-практичної конференції студентів та молодих вчених (5 - 6 травня 2026 р. м. Харків, Україна) - Харків : ХНУ імені В.Н. Каразіна, 2026. С.81-84.*

## ДОДАТОК 1

У цьому додатку наведена програмна реалізація в Python

1.1) Код виконує кластеризацію країн за показниками викидів CO<sub>2</sub> методами k-середніх та complete linkage.

```
1 import pandas as pd
2 import numpy as np
3 from matplotlib import pyplot as plt
4 from matplotlib.patches import Ellipse
5
6 from sklearn.cluster import KMeans, AgglomerativeClustering
7 from sklearn.metrics import silhouette_samples, silhouette_score
8
9 df_capita = pd.read_csv("/data/notebook_files/co2-emissions-per-capita.csv")
10
11 df_country = pd.read_csv("/data/notebook_files/annual-co2-emissions.csv")
12
13 df_capita = df_capita.rename(columns={"Entity": "Country"})
14 df_country = df_country.rename(columns={"Entity": "Country"})
15
16
17 df_capita = df_capita[
18     (df_capita["Year"] >= 2000) & (df_capita["Year"] <= 2024)
19 ]
20
21 df_country = df_country[
22     (df_country["Year"] >= 2000) & (df_country["Year"] <= 2024)
23 ]
24
25 capita_mean = (
26     df_capita
27     .groupby("Country", as_index=False)["CO2 emissions per capita"]
28     .mean()
29     .rename(columns={"CO2 emissions per capita": "CO2_capita_mean"})
30 )
31
32 country_mean = (
33     df_country
34     .groupby("Country", as_index=False)["Annual CO2 emissions"]
35     .mean()
36     .rename(columns={"Annual CO2 emissions": "CO2_total_mean"})
37 )
38
39 df_features = pd.merge(capita_mean, country_mean, on="Country")
40
41 print("\nПобудовані ознаки")
42 print(df_features.head())
43
44 for col in ["CO2_capita_mean", "CO2_total_mean"]:
45     mean_val = df_features[col].mean()
46     std_val = df_features[col].std(ddof=1)
47
48     z_col = col + "_z"
49     df_features[z_col] = (df_features[col] - mean_val) / std_val
50
51 print("\nНормалізовані показники Z-оцінки")
52 print(df_features[["Country", "CO2_capita_mean_z", "CO2_total_mean_z"]].head())
53
54 X = df_features[["CO2_capita_mean_z", "CO2_total_mean_z"]].values
55
56 # Метод k-середніх
57 K = 4
58
59 kmeans = KMeans(
60     n_clusters=K,
61     random_state=42,
62     n_init=10
63 )
64
65 labels_kmeans = kmeans.fit_predict(X)
66 df_features["cluster_kmeans"] = labels_kmeans + 1
67
68 print("\nРезультати методу k-середніх")
69 print(df_features[["Country", "cluster_kmeans"]].sort_values("cluster_kmeans"))
70
71 # Ієрархічна кластеризація complete linkage
72 hier_complete = AgglomerativeClustering(
73     n_clusters=K,
74     linkage="complete",
75     metric="euclidean"
76 )
77
78 labels_complete = hier_complete.fit_predict(X)
79 df_features["cluster_complete"] = labels_complete + 1
80
81 print("\nРезультати ієрархічної кластеризації complete linkage")
82 print(df_features[["Country", "cluster_complete"]].sort_values("cluster_complete"))
83
84 print("\nПерехресна таблиця кластерів")
85 print(
86     pd.crosstab(
87         df_features["cluster_kmeans"],
88         df_features["cluster_complete"]
89     )
90 )
```

```

91
92 map_complete_to_kmeans = {
93     1: 3,
94     2: 2,
95     3: 1,
96     4: 4
97 }
98
99 df_features["cluster_complete_mapped"] = df_features["cluster_complete"].map(
100     map_complete_to_kmeans
101 )
102
103 # Оцінка якості кластеризації – коефіцієнт силуєту
104 # K-means
105 sil_samples_k = silhouette_samples(X, labels_kmeans, metric="euclidean")
106 sil_mean_k = silhouette_score(X, labels_kmeans, metric="euclidean")
107
108 df_features["silhouette_kmeans"] = sil_samples_k
109
110 print(
111     "\nСередній коефіцієнт силуєту для k-середніх: {:.3f}".format(
112         sil_mean_k
113     )
114 )
115
116 # Complete linkage
117 sil_samples_c = silhouette_samples(X, labels_complete, metric="euclidean")
118 sil_mean_c = silhouette_score(X, labels_complete, metric="euclidean")
119
120 df_features["silhouette_complete"] = sil_samples_c
121
122 print(
123     "\nСередній коефіцієнт силуєту для complete linkage: {:.3f}".format(
124         sil_mean_c
125     )
126 )
127
128 print("\nДатумкова таблиця")
129 print(
130     df_features[
131         [
132             "Country",
133             "CO2_capita_mean_z",
134             "CO2_total_mean_z",
135             "cluster_kmeans",
136             "cluster_complete",
137             "silhouette_kmeans",
138             "silhouette_complete",
139         ]
140     ].sort_values("cluster_kmeans")
141 )
142
143 result_table = df_features[
144     [
145         "Country",
146         "CO2_capita_mean_z",
147         "CO2_total_mean_z",
148         "cluster_kmeans",
149         "cluster_complete",
150         "silhouette_kmeans",
151         "silhouette_complete",
152     ]
153 ].sort_values("cluster_kmeans")
154
155 result_table.to_excel(
156     "cluster_results_2000_2024.xlsx",
157     index=False
158 )
159
160 print("\nExcel-файл збережено: cluster_results_2000_2024.xlsx")
161
162 def draw_cluster_ellipses(data, cluster_column):
163     ax = plt.gca()
164
165     for cluster in sorted(data[cluster_column].unique()):
166         cluster_data = data[data[cluster_column] == cluster]
167
168         ellipse = Ellipse(
169             (
170                 cluster_data["CO2_capita_mean_z"].mean(),
171                 cluster_data["CO2_total_mean_z"].mean()
172             ),
173             cluster_data["CO2_capita_mean_z"].max()
174             - cluster_data["CO2_capita_mean_z"].min()
175             + 0.5,
176             cluster_data["CO2_total_mean_z"].max()
177             - cluster_data["CO2_total_mean_z"].min()
178             + 0.5,
179             fill=False,
180             linewidth=2,
181             linestyle="--"
182         )
183
184         ax.add_patch(ellipse)
185
186 colors = ["red", "green", "blue", "purple"]
187

```

```

188 plt.figure(figsize=(8, 6))
189
190 for row in df_features.itertuples():
191     plt.scatter(
192         row.CO2_capita_mean_z,
193         row.CO2_total_mean_z,
194         color=colors[row.cluster_kmeans - 1],
195         s=60
196     )
197
198 draw_cluster_ellipses(df_features, "cluster_kmeans")
199
200 plt.title("Кластери країн (метод k-середніх)")
201 plt.xlabel("CO2 per capita (Z-оцінка)")
202 plt.ylabel("CO2 total (Z-оцінка)")
203 plt.grid(True, alpha=0.3)
204 plt.show()
205
206 plt.figure(figsize=(8, 6))
207
208 for row in df_features.itertuples():
209     plt.scatter(
210         row.CO2_capita_mean_z,
211         row.CO2_total_mean_z,
212         color=colors[row.cluster_complete_mapped - 1],
213         s=60
214     )
215
216 draw_cluster_ellipses(df_features, "cluster_complete_mapped")
217
218 plt.title("Кластери країн (ієрархічний метод, complete linkage)")
219 plt.xlabel("CO2 per capita (Z-оцінка)")
220 plt.ylabel("CO2 total (Z-оцінка)")
221 plt.grid(True, alpha=0.3)
222 plt.show()
223
224 import numpy as np
225 import matplotlib.pyplot as plt
226
227 def compute_centers_radii(df, label_col):
228     centers = []
229
230     for c1 in sorted(df[label_col].unique()):
231         subset = df[df[label_col] == c1]
232
233         cx = subset["CO2_capita_mean_z"].mean()
234         cy = subset["CO2_total_mean_z"].mean()
235
236         dists = np.sqrt(
237             (subset["CO2_capita_mean_z"] - cx) ** 2 +
238             (subset["CO2_total_mean_z"] - cy) ** 2
239         )
240
241         R = dists.max() + 0.25
242
243         centers.append({
244             "cluster": int(c1),
245             "cx": cx,
246             "cy": cy,
247             "radius": R
248         })
249
250     return centers
251
252 centers_kmeans = compute_centers_radii(df_features, "cluster_kmeans")
253 centers_complete = compute_centers_radii(df_features, "cluster_complete_mapped")
254
255 centers_kmeans_df = pd.DataFrame(centers_kmeans)
256 centers_complete_df = pd.DataFrame(centers_complete)
257
258 print("\nЦентри і радіуси кластерів k-means")
259 print(centers_kmeans_df)
260
261 print("\nЦентри і радіуси кластерів complete linkage")
262 print(centers_complete_df)
263
264 with pd.ExcelWriter("cluster_results_2000_2024_extended.xlsx") as writer:
265     result_table.to_excel(writer, sheet_name="Cluster results", index=False)
266     centers_kmeans_df.to_excel(writer, sheet_name="KMeans centers", index=False)
267     centers_complete_df.to_excel(writer, sheet_name="Complete centers", index=False)
268
269 def plot_clusters_with_circles(df, label_col, centers, title):
270     fig, ax = plt.subplots(figsize=(10, 7))
271
272     colors = ["red", "green", "blue", "purple"]
273
274     offsets = {
275         "Denmark": (0.15, -0.12),
276         "Greece": (0.10, 0.10),
277         "Slovakia": (0.10, -0.02),
278         "Sweden": (-0.18, 0.08),
279         "Hungary": (0.14, -0.14),
280         "Croatia": (-0.20, -0.08),
281         "Norway": (-0.15, 0.08),
282         "Ireland": (0.08, -0.08),

```

```

283     "Finland": (0.08, -0.10),
284     "Belgium": (0.08, 0.08),
285     "Netherlands": (0.08, 0.10),
286     "Czechia": (0.08, -0.06),
287     "France": (0.08, 0.08),
288     "Ukraine": (0.08, -0.10),
289     "Spain": (0.08, -0.08),
290     "Italy": (0.08, 0.08),
291     "United Kingdom": (0.08, 0.08),
292     "Poland": (0.08, -0.08),
293     "Germany": (0.08, 0.08)
294 }
295
296 for _, row in df.iterrows():
297     x = row["CO2_capita_mean_z"]
298     y = row["CO2_total_mean_z"]
299     cl = int(row[label_col])
300
301     ax.scatter(
302         x,
303         y,
304         color=colors[cl - 1],
305         s=60
306     )
307
308     dx, dy = offsets.get(row["Country"], (0.05, 0.05))
309
310     ax.annotate(
311         row["Country"],
312         xy=(x, y),
313         xytext=(x + dx, y + dy),
314         fontsize=8,
315         arrowprops=dict(
316             arrowstyle="-",
317             linewidth=0.5,
318             color="gray"
319         ),
320         bbox=dict(
321             boxstyle="round,pad=0.2",
322             fc="white",
323             ec="none",
324             alpha=0.7
325         )
326     )
327
328     for c in centers:
329         circle = plt.Circle(
330             (c["cx"], c["cy"]),
331             c["radius"],
332             fill=False,
333             linestyle="--",
334             linewidth=2,
335             color="black"
336         )
337
338         ax.add_patch(circle)
339
340         ax.text(
341             c["cx"],
342             c["cy"],
343             "C" + str(c["cluster"]),
344             fontsize=11,
345             fontweight="bold",
346             ha="center",
347             va="center"
348         )
349
350     ax.set_xlabel("CO2 per capita (Z-оцінка)")
351     ax.set_ylabel("CO2 total (Z-оцінка)")
352     ax.set_title(title)
353     ax.grid(True, alpha=0.3)
354
355     plt.tight_layout()
356     plt.show()
357
358 plot_clusters_with_circles(
359     df_features,
360     label_col="cluster_kmeans",
361     centers=centers_kmeans,
362     title="Кластери та кола навколо центрів, метод k-середніх, 2000-2024"
363 )
364
365 plot_clusters_with_circles(
366     df_features,
367     label_col="cluster_complete_mapped",
368     centers=centers_complete,
369     title="Кластери та кола навколо центрів, complete linkage, 2000-2024"
370 )

```

## 1.2) Наступний код виконує прогнозування часових рядів за допомогою моделі ARIMA

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 from statsmodels.tsa.arima_model import ARIMA
6 from statsmodels.tsa.stattools import adfuller
7 from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
8
9 per_capita = pd.read_csv("/data/notebook_files/co2-emissions-per-capita.csv")
10 total = pd.read_csv("/data/notebook_files/annual-co2-emissions.csv")
11
12 print(per_capita.columns)
13 print(total.columns)
14
15 per_capita = per_capita.rename(columns={
16     per_capita.columns[0]: "Country",
17     per_capita.columns[1]: "Code",
18     per_capita.columns[2]: "Year",
19     per_capita.columns[3]: "CO2_per_capita"
20 })
21
22 total = total.rename(columns={
23     total.columns[0]: "Country",
24     total.columns[1]: "Code",
25     total.columns[2]: "Year",
26     total.columns[3]: "CO2_total"
27 })
28
29 per_capita["Year"] = pd.to_numeric(per_capita["Year"], errors="coerce")
30 total["Year"] = pd.to_numeric(total["Year"], errors="coerce")
31
32 per_capita["CO2_per_capita"] = pd.to_numeric(
33     per_capita["CO2_per_capita"], errors="coerce"
34 )
35
36 total["CO2_total"] = pd.to_numeric(
37     total["CO2_total"], errors="coerce"
38 )
39
40 per_capita = per_capita[
41     (per_capita["Year"] >= 2000) &
42     (per_capita["Year"] <= 2024)
43 ]
44
45 total = total[
46     (total["Year"] >= 2000) &
47     (total["Year"] <= 2024)
48 ]
49
50 def calculate_metrics(actual, forecast):
51     actual = pd.Series(actual).reset_index(drop=True)
52     forecast = pd.Series(forecast).reset_index(drop=True)
53
54     df_metrics = pd.DataFrame({
55         "actual": actual,
56         "forecast": forecast
57     }).dropna()
58
59     errors = df_metrics["actual"] - df_metrics["forecast"]
60
61     mae = np.mean(np.abs(errors))
62     rmse = np.sqrt(np.mean(errors ** 2))
63
64     nonzero = df_metrics["actual"] != 0
65     mape = np.mean(
66         np.abs(errors[nonzero] / df_metrics["actual"][nonzero])
67     ) * 100
68
69     return mae, rmse, mape
70
71 def select_best_arima(series, max_p=3, max_d=2, max_q=3):
72     best_aic = np.inf
73     best_order = None
74     best_model = None
75
76     series = series.dropna()
77
78     for p in range(max_p + 1):
79         for d in range(max_d + 1):
80             for q in range(max_q + 1):
81                 try:
82                     model = ARIMA(series, order=(p, d, q))
83                     fitted = model.fit()
84
85                     if fitted.aic < best_aic:
86                         best_aic = fitted.aic
87                         best_order = (p, d, q)
88                         best_model = fitted
89
90                 except:
91                     continue
92
93     return best_model, best_order, best_aic
94
95 def make_forecast(data, indicator_col, indicator_name):
96     results = []
97     forecast_years = [2025, 2026, 2027, 2028]
98
99     countries = data["Country"].unique()
100
101     for country in countries:
102         country_data = data[data["Country"] == country].sort_values("Year")
```

```

183
184     series = country_data.set_index("Year")[indicator_col].dropna()
185
186     if len(series) < 15:
187         continue
188
189     train = series[series.index <= 2019]
190     test = series[(series.index >= 2020) & (series.index <= 2024)]
191
192     if len(train) < 10 or len(test) == 0:
193         continue
194
195     try:
196         adf_p_value = adfuller(series)[1]
197     except:
198         adf_p_value = np.nan
199
200     train_model, order, train_aic = select_best_arima(train)
201
202     if train_model is None:
203         continue
204
205     test_forecast = train_model.get_forecast(steps=len(test))
206     test_forecast_mean = test_forecast.predicted_mean
207
208     mae, rmse, mape = calculate_metrics(
209         test.values,
210         test_forecast_mean.values
211     )
212
213     final_model = ARIMA(series, order=order).fit()
214
215     forecast = final_model.get_forecast(steps=4)
216     forecast_mean = forecast.predicted_mean
217     conf_int = forecast.conf_int()
218
219     plt.figure(figsize=(10, 5))
220
221     plt.plot(
222         series.index,
223         series.values,
224         label="Actual data"
225     )
226
227     plt.plot(
228         forecast_years,
229         forecast_mean.values,
230         label="Forecast"
231     )
232
233     plt.fill_between(
234         forecast_years,
235         conf_int.iloc[:, 0],
236         conf_int.iloc[:, 1],
237         alpha=0.3,
238         label="Confidence interval"
239     )
240
241     plt.title(f"{country} - {indicator_name} ARIMA Forecast")
242     plt.xlabel("Year")
243     plt.ylabel(indicator_name)
244     plt.legend()
245     plt.grid(True)
246     plt.show()
247
248     residuals = final_model.resid
249
250     plt.figure(figsize=(10, 4))
251
252     plt.plot(residuals)
253     plt.axhline(y=0, linestyle="--")
254
255     plt.title(f"{country} Residual Plot")
256     plt.xlabel("Time")
257     plt.ylabel("Residuals")
258     plt.grid(True)
259     plt.show()
260
261     plt.figure(figsize=(8, 4))
262     plot_acf(residuals, lags=10)
263     plt.title(f"{country} ACF Plot")
264     plt.show()
265
266     plt.figure(figsize=(8, 4))
267     plot_pacf(residuals, lags=10)
268     plt.title(f"{country} PACF Plot")
269     plt.show()
270
271     for i, year in enumerate(forecast_years):
272         results.append({
273             "Country": country,
274             "Indicator": indicator_name,
275             "Year": year,
276             "Forecast": forecast_mean.iloc[i],
277             "Lower_CI": conf_int.iloc[i, 0],
278             "Upper_CI": conf_int.iloc[i, 1],
279             "ARIMA_order": order,
280             "AIC": final_model.aic,
281             "ADF_p_value": adf_p_value,
282             "Test_period": "2020-2024",
283             "MAE": mae,
284             "RMSE": rmse,
285             "MAPE": mape
286         })
287
288     return pd.DataFrame(results)
289
290 forecast_per_capita = make_forecast(
291     per_capita,
292     "CO2 per capita",
293     "CO2 per capita"
294 )

```

```

215 forecast_total = make_forecast(
216     total,
217     "CO2_total",
218     "CO2 total"
219 )
220
221
222 final_forecast = pd.concat(
223     [forecast_per_capita, forecast_total],
224     ignore_index=True
225 )
226
227 final_forecast.to_excel(
228     "co2_arima_forecast_2025_2028_test_2020_2024.xlsx",
229     index=False
230 )
231
232 print(final_forecast)

```

### 1.3) Наступний код виконує прогнозування часових рядів за допомогою методу Exponential Smoothing

```

1 import warnings
2 warnings.filterwarnings("ignore")
3
4 import pandas as pd
5 import numpy as np
6 import matplotlib.pyplot as plt
7 from statsmodels.tsa.holtwinters import ExponentialSmoothing
8
9 def load_data(path, value_name):
10     df = pd.read_csv(path)
11
12     df = df.rename(columns={
13         df.columns[0]: "Country",
14         df.columns[2]: "Year",
15         df.columns[3]: value_name
16     })
17
18     df["Year"] = pd.to_numeric(df["Year"], errors="coerce")
19     df[value_name] = pd.to_numeric(df[value_name], errors="coerce")
20
21     df = df[
22         (df["Year"] >= 2000) &
23         (df["Year"] <= 2024)
24     ]
25
26     return df
27
28 per_capita = load_data(
29     "/data/notebook_files/co2_emissions_per_capita.csv",
30     "CO2_per_capita"
31 )
32
33 total = load_data(
34     "/data/notebook_files/annual_co2_emissions.csv",
35     "CO2_total"
36 )
37
38 def calculate_metrics(actual, forecast):
39     df = pd.DataFrame({
40         "actual": pd.Series(actual).reset_index(drop=True),
41         "forecast": pd.Series(forecast).reset_index(drop=True)
42     }).dropna()
43
44     errors = df["actual"] - df["forecast"]
45
46     mae = np.mean(np.abs(errors))
47     rmse = np.sqrt(np.mean(errors ** 2))
48
49     nonzero = df["actual"] != 0
50     mape = np.mean(
51         np.abs(errors[nonzero] / df["actual"][nonzero])
52     ) * 100
53
54     return mae, rmse, mape
55
56 def make_exp_smoothing_forecast(data, indicator_col, indicator_name):
57     results = []
58     forecast_years = [2025, 2026, 2027, 2028]
59
60     for country in data["Country"].unique():
61
62         country_data = data[data["Country"] == country].sort_values("Year")
63
64         series = country_data.set_index("Year")[indicator_col]
65         series = series.astype(float).dropna()
66
67         if len(series) < 15:
68             continue
69
70         train = series[series.index <= 2019]
71         test = series[(series.index >= 2020) & (series.index <= 2024)]
72
73         if len(train) < 10 or len(test) == 0:
74             continue
75
76         try:
77
78             train_model = ExponentialSmoothing(
79                 train,
80                 trend="add",
81                 seasonal=None,
82                 damped_trend=False
83             ).fit(optimized=True)

```

```

84
85     test_forecast = train_model.forecast(steps=len(test))
86
87     mae, rmse, mape = calculate_metrics(
88         test.values,
89         test_forecast.values
90     )
91
92     final_model = ExponentialSmoothing(
93         series,
94         trend="add",
95         seasonal=None,
96         damped_trend=False
97     ).fit(optimized=True)
98
99     fitted_values = final_model.fittedvalues
100     forecast = final_model.forecast(steps=4)
101     residuals = series - fitted_values
102
103     plt.figure(figsize=(10, 5))
104
105     plt.plot(
106         series.index,
107         series.values,
108         marker="o",
109         label="Actual data"
110     )
111
112     plt.plot(
113         series.index,
114         fitted_values.values,
115         linestyle="--",
116         label="Fitted values"
117     )
118
119     plt.plot(
120         forecast_years,
121         forecast.values,
122         marker="o",
123         linestyle="--",
124         label="Forecast"
125     )
126
127     plt.title(f"{country}: Exponential Smoothing Forecast")
128     plt.xlabel("Year")
129     plt.ylabel(indicator_name)
130     plt.legend()
131     plt.grid(True)
132     plt.show()
133
134     plt.figure(figsize=(6, 6))
135
136     plt.scatter(
137         series.values,
138         fitted_values.values
139     )
140
141     min_value = min(series.min(), fitted_values.min())
142     max_value = max(series.max(), fitted_values.max())
143
144     plt.plot(
145         [min_value, max_value],
146         [min_value, max_value],
147         linestyle="--"
148     )
149
150     plt.title(f"{country}: Actual vs Fitted")
151     plt.xlabel("Actual values")
152     plt.ylabel("Fitted values")
153     plt.grid(True)
154     plt.show()
155
156     plt.figure(figsize=(10, 4))
157
158     plt.plot(
159         series.index,
160         residuals.values,
161         marker="o"
162     )
163
164     plt.axhline(y=0, linestyle="--")
165
166     plt.title(f"{country}: Residual Plot")
167     plt.xlabel("Year")
168     plt.ylabel("Residuals")
169     plt.grid(True)
170     plt.show()
171
172     for i, year in enumerate(forecast_years):
173         results.append(
174             {
175                 "Country": country,
176                 "Indicator": indicator_name,
177                 "Year": year,
178                 "Forecast": forecast.iloc[i],
179                 "Method": "Holt Exponential Smoothing",
180                 "Trend": "Additive",
181                 "Alpha": final_model.params["smoothing_level"],
182                 "Beta": final_model.params["smoothing_trend"],
183                 "Test_period": "2020-2024",
184                 "MAE": mae,
185                 "RMSE": rmse,
186                 "MAPE": mape
187             }
188         )
189
190     except Exception as e:
191         print(f"Error for {country}, {indicator_name}: {e}")
192         continue

```

```

191
192     return pd.DataFrame(results)
193
194 forecast_per_capita_es = make_exp_smoothing_forecast(
195     per_capita,
196     "CO2_per_capita",
197     "CO2 per capita"
198 )
199
200 forecast_total_es = make_exp_smoothing_forecast(
201     total,
202     "CO2_total",
203     "CO2 total"
204 )
205
206 final_es_forecast = pd.concat(
207     [forecast_per_capita_es, forecast_total_es],
208     ignore_index=True
209 )
210
211 final_es_forecast.to_excel(
212     "co2_exponential_smoothing_forecast_2025_2028_test_2020_2024.xlsx",
213     index=False
214 )
215
216 print(final_es_forecast)

```

## ДОДАТОК 2

У цьому додатку наведено таблиці результатів

### 2.1) Результати кластеризації європейських країн методами K-Means та

#### Complete Linkage, координати центрів кластерів

Country	CO2_capita_mean_z	CO2_total_mean_z	cluster_kmeans	cluster_complete_mapped	silhouette_kmeans	silhouette_complete
Belgium	1,258204838	-0,393591742	1	1	0,703091751	0,779650883
Czechia	1,654649865	-0,37883588	1	1	0,61947356	0,707809846
Denmark	0,124214273	-0,718873067	1	3	0,258043616	-0,04090676
Finland	1,163197495	-0,673775553	1	1	0,730117997	0,763852842
Greece	0,249123175	-0,500858603	1	3	0,419107175	-0,250336704
Norway	0,635263887	-0,71773431	1	1	0,653669276	0,525890129
Netherlands	1,080506272	-0,126606166	1	1	0,617921775	0,686688231
Ireland	0,869415126	-0,728906584	1	1	0,715054124	0,685004147
Poland	0,444554779	0,660406687	2	2	0,649176947	0,162435049
Italy	-0,349112074	1,093340486	2	2	0,14290752	0,631314476
France	-0,99759141	0,841088663	2	2	0,48626945	0,508584952
Spain	-0,637688289	0,494973599	2	2	0,467489047	0,469342967
Ukraine	-0,968501032	0,372021162	2	2	0,227031548	0,28717636
United Kingdom	-0,102361538	1,371713976	2	2	0,548389482	0,548389482
Portugal	-1,227963242	-0,664287976	3	3	0,808738404	0,597283884
Croatia	-1,428476864	-0,833989911	3	3	0,77182733	0,569711603
Hungary	-1,227532801	-0,6773802	3	3	0,811716998	0,601043531
Slovakia	-0,318670355	-0,747610644	3	3	0,933965666	0,426828706
Sweden	-1,349664723	-0,699284842	3	3	0,800693342	0,582248958
Germany	1,128432617	3,028190903	4	4	0	0

cluster	cx	cy	radius
1	0,879322	-0,5299	1,039907
2	-0,43512	0,805591	1,141572
3	-1,11046	-0,72451	1,042128
4	1,128433	3,028191	0,25

cluster	cx	cy	radius
1	1,110206	-0,50324	0,808476
2	-0,43512	0,805591	1,141572
3	-0,73985	-0,69176	1,257231
4	1,128433	3,028191	0,25

### 2.2) Результати прогнозування викидів CO2 методом ARIMA на 2025-2028 роки

Country	Indicator	Year	Forecast	Lower CI	Upper CI	ARIMA_order	AIC	ADF_p_value	Test_period	MAE	RMSE	MAPE	
1	Belgium	CO2 per capita	2025	6,968976	6,381286	7,558465	(0, 1, 2)	19,7493	0,7681656	2020-2024	0,84316	0,91967	11,27455
2	Belgium	CO2 per capita	2026	6,835255	6,182863	7,487647	(0, 1, 2)	19,7493	0,7681656	2020-2024	0,84316	0,91967	11,27455
3	Belgium	CO2 per capita	2027	6,835255	5,790031	7,890479	(0, 1, 2)	19,7493	0,7681656	2020-2024	0,84316	0,91967	11,27455
4	Belgium	CO2 per capita	2028	6,835255	5,480911	8,17741	(0, 1, 2)	19,7493	0,7681656	2020-2024	0,84316	0,91967	11,27455
5	Croatia	CO2 per capita	2025	4,866457	4,55866	5,174253	(1, 1, 2)	-9,56023	0,6845596	2020-2024	0,1679	0,1907	3,667015
6	Croatia	CO2 per capita	2026	4,781137	4,293926	5,358113	(1, 1, 2)	-9,56023	0,6845596	2020-2024	0,1679	0,1907	3,667015
7	Croatia	CO2 per capita	2027	4,866456	4,106166	5,627606	(1, 1, 2)	-9,56023	0,6845596	2020-2024	0,1679	0,1907	3,667015
8	Croatia	CO2 per capita	2028	4,781139	3,877668	5,855113	(1, 1, 2)	-9,56023	0,6845596	2020-2024	0,1679	0,1907	3,667015
9	Czechia	CO2 per capita	2025	6,929124	6,105116	7,753131	(0, 1, 1)	30,8403	0,995511	2020-2024	1,06794	1,33595	19,98585
10	Czechia	CO2 per capita	2026	6,929124	5,426284	8,431963	(0, 1, 1)	30,8403	0,995511	2020-2024	1,06794	1,33595	19,98585
11	Czechia	CO2 per capita	2027	6,929124	4,970025	8,888222	(0, 1, 1)	30,8403	0,995511	2020-2024	1,06794	1,33595	19,98585
12	Czechia	CO2 per capita	2028	6,929124	4,601549	9,256698	(0, 1, 1)	30,8403	0,995511	2020-2024	1,06794	1,33595	19,98585
13	Denmark	CO2 per capita	2025	5,061237	4,156441	5,966033	(0, 1, 3)	42,2576	0,9497359	2020-2024	0,18556	0,23783	3,777768
14	Denmark	CO2 per capita	2026	5,277882	4,133312	6,422452	(0, 1, 3)	42,2576	0,9497359	2020-2024	0,18556	0,23783	3,777768
15	Denmark	CO2 per capita	2027	5,245783	3,193357	6,572209	(0, 1, 3)	42,2576	0,9497359	2020-2024	0,18556	0,23783	3,777768
16	Denmark	CO2 per capita	2028	5,245783	3,130325	7,17854	(0, 1, 3)	42,2576	0,9497359	2020-2024	0,18556	0,23783	3,777768
17	Finland	CO2 per capita	2025	5,300579	4,844254	5,896033	(0, 1, 0)	71,7412	0,9895968	2020-2024	1,45686	1,58573	24,70546
18	Finland	CO2 per capita	2026	5,300579	4,243914	6,182444	(0, 1, 0)	71,7412	0,9895968	2020-2024	1,45686	1,58573	24,70546
19	Finland	CO2 per capita	2027	5,300579	3,728233	7,328325	(0, 1, 0)	71,7412	0,9895968	2020-2024	1,45686	1,58573	24,70546
20	Finland	CO2 per capita	2028	5,300579	3,130325	8,812737	(0, 1, 0)	71,7412	0,9895968	2020-2024	1,45686	1,58573	24,70546
21	France	CO2 per capita	2025	3,489352	3,291557	4,269547	(0, 2, 1)	6,75666	0,9805963	2020-2024	0,22867	0,27782	5,510154
22	France	CO2 per capita	2026	3,729335	3,123318	4,335353	(0, 2, 1)	6,75666	0,9805963	2020-2024	0,22867	0,27782	5,510154
23	France	CO2 per capita	2027	3,609319	2,852959	4,365679	(0, 2, 1)	6,75666	0,9805963	2020-2024	0,22867	0,27782	5,510154
24	France	CO2 per capita	2028	3,489352	2,599901	4,378054	(0, 2, 1)	6,75666	0,9805963	2020-2024	0,22867	0,27782	5,510154
25	Germany	CO2 per capita	2025	6,616703	5,844254	7,389152	(0, 1, 2)	29,4757	0,9977795	2020-2024	0,52365	0,56225	7,180106
26	Germany	CO2 per capita	2026	6,56578	5,500795	7,630765	(0, 1, 2)	29,4757	0,9977795	2020-2024	0,52365	0,56225	7,180106
27	Germany	CO2 per capita	2027	6,56578	5,194717	8,068443	(0, 1, 2)	29,4757	0,9977795	2020-2024	0,52365	0,56225	7,180106
28	Germany	CO2 per capita	2028	6,56578	4,945461	8,186099	(0, 1, 2)	29,4757	0,9977795	2020-2024	0,52365	0,56225	7,180106
29	Greece	CO2 per capita	2025	5,183155	4,42445	5,94186	(0, 2, 1)	26,4299	0,9572951	2020-2024	0,38244	0,45625	7,228607
30	Greece	CO2 per capita	2026	5,055618	3,839095	6,27214	(0, 2, 1)	26,4299	0,9572951	2020-2024	0,38244	0,45625	7,228607
31	Greece	CO2 per capita	2027	4,928081	3,258712	6,597449	(0, 2, 1)	26,4299	0,9572951	2020-2024	0,38244	0,45625	7,228607
32	Greece	CO2 per capita	2028	4,800544	2,662755	6,938332	(0, 2, 1)	26,4299	0,9572951	2020-2024	0,38244	0,45625	7,228607
33	Hungary	CO2 per capita	2025	4,135669	3,678469	4,592869	(0, 1, 0)	24,2428	0,9246314	2020-2024	0,47453	0,58712	11,06261
34	Hungary	CO2 per capita	2026	4,135669	3,443776	4,927562	(0, 1, 0)	24,2428	0,9246314	2020-2024	0,47453	0,58712	11,06261
35	Hungary	CO2 per capita	2027	4,135669	3,22127	5,050638	(0, 1, 0)	24,2428	0,9246314	2020-2024	0,47453	0,58712	11,06261
36	Hungary	CO2 per capita	2028	4,135669	2,913121	5,613121	(0, 1, 0)	24,2428	0,9246314	2020-2024	0,47453	0,58712	11,06261
37	Ireland	CO2 per capita	2025	6,107125	5,30113	6,913121	(0, 2, 1)	30,6357	0,9138737	2020-2024	0,23786	0,2642	3,361886
38	Ireland	CO2 per capita	2026	5,875427	4,712979	7,037874	(0, 2, 1)	30,6357	0,9138737	2020-2024	0,23786	0,2642	3,361886
39	Ireland	CO2 per capita	2027	5,643728	4,192875	7,094581	(0, 2, 1)	30,6357	0,9138737	2020-2024	0,23786	0,2642	3,361886
40	Ireland	CO2 per capita	2028	5,412029	3,70595	7,118038	(0, 2, 1)	30,6357	0,9138737	2020-2024	0,23786	0,2642	3,361886
41	Italy	CO2 per capita	2025	5,087886	4,426366	5,749406	(0, 1, 0)	17,9747	0,9029826	2020-2024	0,32771	0,40865	6,377063
42	Italy	CO2 per capita	2026	5,087886	4,152355	6,023417	(0, 1, 0)	17,9747	0,9029826	2020-2024	0,32771	0,40865	6,377063
43	Italy	CO2 per capita	2027	5,087886	3,942099	6,236373	(0, 1, 0)	17,9747	0,9029826	2020-2024	0,32771	0,40865	6,377063
44	Italy	CO2 per capita	2028	5,087886	3,64846	6,410926	(0, 1, 0)	17,9747	0,9029826	2020-2024	0,32771	0,40865	6,377063
45	Netherlands	CO2 per capita	2025	5,5604	4,921005	6,200694	(0, 1, 2)	23,7061	0,9965087	2020-2024	1,20779	1,3606	17,95008
46	Netherlands	CO2 per capita	2026	5,423438	4,581305	6,265572	(0, 1, 2)	23,7061	0,9965087	2020-2024	1,20779	1,3606	17,95008
47	Netherlands	CO2 per capita	2027	5,423438	4,040594	6,890233	(0, 1, 2)	23,7061	0,9965087	2020-2024	1,20779	1,3606	17,95008
48	Netherlands	CO2 per capita	2028	5,423438	3,60428	7,242597	(0, 1, 2)	23,7061	0,9965087	2020-2024	1,20779	1,3606	17,95008
49	Norway	CO2 per capita	2025	6,785593	6,131537	7,158029	(0, 1, 3)	-0,27374	1	2020-2024	0,45417	0,55104	6,472585
50	Norway	CO2 per capita	2026	6,543683	6,049575	7,037591	(0, 1, 3)	-0,27374	1	2020-2024	0,45417	0,55104	6,472585
51	Norway	CO2 per capita	2027	6,259663	5,665722	6,853824	(0, 1, 3)	-0,27374	1	2020-2024	0,45417	0,55104	6,472585
52	Norway	CO2 per capita	2028	6,259663	5,085661	7,130366	(0, 1, 3)	-0,27374	1	2020-2024	0,45417	0,55104	6,472585
53	Poland	CO2 per capita	2025	7,755798	7,144921	8,407145	(0, 1, 0)	22,5821	0,7046304	2020-2024	0,68339	0,83751	9,211806
54	Poland	CO2 per capita	2026	8,321013	7,502027	9,121799	(0, 1, 0)	22,5821	0,7046304	2020-2024	0,68339	0,83751	9,211806
55	Poland	CO2 per capita	2027	8,321013	7,502027	9,121799	(0, 1, 0)	22,5821	0,7046304	2020-2024	0,68339	0,83751	9,211806
56	Poland	CO2 per capita	2028	8,321013	7,502027	9,121799	(0, 1, 0)	22,5821	0,7046304	2020-2024	0,68339	0,83751	9,211806
57	Portugal	CO2 total	2025	85209040	71692475	1,01E+08	(0, 2, 0)	796,893	0,8759864</				

Country	Indicator	Year	Forecast	Method	Trend	Alpha	Beta	est_perio	MAE	RMSE	MAPE	
Portugal	CO2 per capita	2025	3,408907	2,79467	4,023145	(0, 1, 0)	14,4153	0,7319897	2020-2024	0,83481	0,86457	22,62058
Portugal	CO2 per capita	2026	3,408907	2,540244	4,277571	(0, 1, 0)	14,4153	0,7319897	2020-2024	0,83481	0,86457	22,62058
Portugal	CO2 per capita	2027	3,408907	2,345016	4,472799	(0, 1, 0)	14,4153	0,7319897	2020-2024	0,83481	0,86457	22,62058
Portugal	CO2 per capita	2028	3,408907	2,180432	4,637383	(0, 1, 0)	14,4153	0,7319897	2020-2024	0,83481	0,86457	22,62058
Slovakia	CO2 per capita	2025	5,282773	4,609548	5,955988	(0, 1, 0)	18,8166	0,9689491	2020-2024	0,55285	0,59571	9,87046
Slovakia	CO2 per capita	2026	5,282773	4,330689	6,234875	(0, 1, 0)	18,8166	0,9689491	2020-2024	0,55285	0,59571	9,87046
Slovakia	CO2 per capita	2027	5,282773	4,116713	6,448833	(0, 1, 0)	18,8166	0,9689491	2020-2024	0,55285	0,59571	9,87046
Slovakia	CO2 per capita	2028	5,282773	3,936233	6,629223	(0, 1, 0)	18,8166	0,9689491	2020-2024	0,55285	0,59571	9,87046
Spain	CO2 per capita	2025	4,655103	3,912482	5,397723	(0, 1, 1)	25,6021	0,3398069	2020-2024	0,5215	0,54969	11,38016
Spain	CO2 per capita	2026	4,655103	3,545132	5,856073	(0, 1, 1)	25,6021	0,3398069	2020-2024	0,5215	0,54969	11,38016
Spain	CO2 per capita	2027	4,655103	3,12763	6,182575	(0, 1, 1)	25,6021	0,3398069	2020-2024	0,5215	0,54969	11,38016
Spain	CO2 per capita	2028	4,655103	2,859548	6,450657	(0, 1, 1)	25,6021	0,3398069	2020-2024	0,5215	0,54969	11,38016
Sweden	CO2 per capita	2025	3,591654	3,104221	4,079087	(0, 1, 0)	3,3159	0,0536908	2020-2024	0,41137	0,42034	11,63593
Sweden	CO2 per capita	2026	3,591654	2,90232	4,298988	(0, 1, 0)	3,3159	0,0536908	2020-2024	0,41137	0,42034	11,63593
Sweden	CO2 per capita	2027	3,591654	2,747396	4,435913	(0, 1, 0)	3,3159	0,0536908	2020-2024	0,41137	0,42034	11,63593
Sweden	CO2 per capita	2028	3,591654	2,616789	4,56652	(0, 1, 0)	3,3159	0,0536908	2020-2024	0,41137	0,42034	11,63593
Ukraine	CO2 per capita	2025	3,763788	2,866331	4,632145	(0, 1, 0)	32,6159	0,9902725	2020-2024	0,87599	1,02022	23,50896
Ukraine	CO2 per capita	2026	3,763788	2,494593	5,032984	(0, 1, 0)	32,6159	0,9902725	2020-2024	0,87599	1,02022	23,50896
Ukraine	CO2 per capita	2027	3,763788	2,209347	5,318229	(0, 1, 0)	32,6159	0,9902725	2020-2024	0,87599	1,02022	23,50896
Ukraine	CO2 per capita	2028	3,763788	1,968785	5,558072	(0, 1, 0)	32,6159	0,9902725	2020-2024	0,87599	1,02022	23,50896
United Kingdom	CO2 per capita	2025	4,154609	3,603304	4,705914	(0, 1, 2)	16,5558	0,333763	2020-2024	0,43145	0,48	9,409396
United Kingdom	CO2 per capita	2026	4,364012	3,56504	5,162984	(0, 1, 2)	16,5558	0,333763	2020-2024	0,43145	0,48	9,409396
United Kingdom	CO2 per capita	2027	4,364012	3,101363	5,714391	(0, 1, 2)	16,5558	0,333763	2020-2024	0,43145	0,48	9,409396
United Kingdom	CO2 per capita	2028	4,364012	2,629454	6,09857	(0, 1, 2)	16,5558	0,333763	2020-2024	0,43145	0,48	9,409396

## 2.3) Результати прогнозування викидів CO<sub>2</sub> методом Holt Exponential Smoothing на 2025-2028 роки

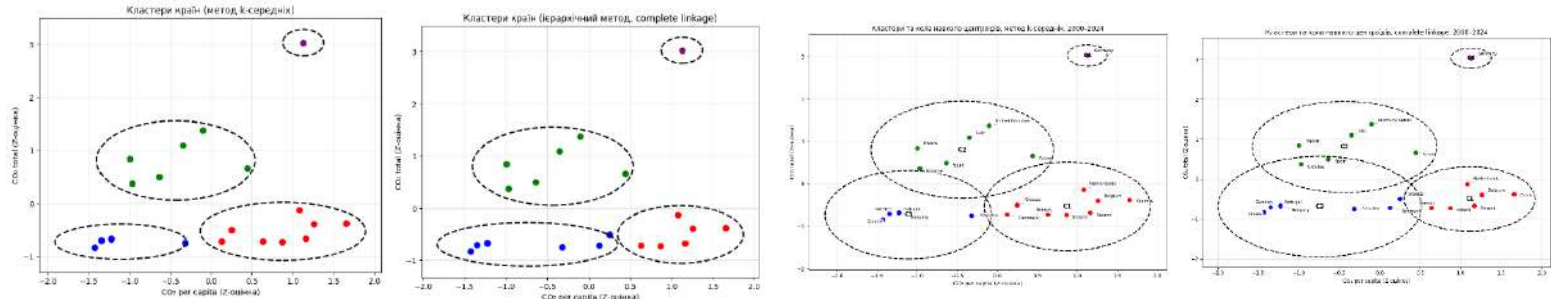
Country	Indicator	Year	Forecast	Method	Trend	Alpha	Beta	est_perio	MAE	RMSE	MAPE
Belgium	CO2 per c	2025	6,99974	Holt Expor Additive	0,62917	0	0	2020-202	0,37673	0,42525	5,01453
Belgium	CO2 per c	2026	6,78141	Holt Expor Additive	0,62917	0	0	2020-202	0,37673	0,42525	5,01453
Belgium	CO2 per c	2027	6,56309	Holt Expor Additive	0,62917	0	0	2020-202	0,37673	0,42525	5,01453
Belgium	CO2 per c	2028	6,34476	Holt Expor Additive	0,62917	0	0	2020-202	0,37673	0,42525	5,01453
Croatia	CO2 per c	2025	4,77397	Holt Expor Additive	1	0	0	2020-202	0,14473	0,16724	3,17634
Croatia	CO2 per c	2026	4,78322	Holt Expor Additive	1	0	0	2020-202	0,14473	0,16724	3,17634
Croatia	CO2 per c	2027	4,79246	Holt Expor Additive	1	0	0	2020-202	0,14473	0,16724	3,17634
Croatia	CO2 per c	2028	4,80171	Holt Expor Additive	1	0	0	2020-202	0,14473	0,16724	3,17634
Czechia	CO2 per c	2025	6,81959	Holt Expor Additive	1	0	0	2020-202	0,81471	1,03269	10,6807
Czechia	CO2 per c	2026	6,59521	Holt Expor Additive	1	0	0	2020-202	0,81471	1,03269	10,6807
Czechia	CO2 per c	2027	6,37083	Holt Expor Additive	1	0	0	2020-202	0,81471	1,03269	10,6807
Czechia	CO2 per c	2028	6,14646	Holt Expor Additive	1	0	0	2020-202	0,81471	1,03269	10,6807
Denmark	CO2 per c	2025	4,38304	Holt Expor Additive	0,55341	0	0	2020-202	0,37134	0,40816	7,67259
Denmark	CO2 per c	2026	4,13842	Holt Expor Additive	0,55341	0	0	2020-202	0,37134	0,40816	7,67259
Denmark	CO2 per c	2027	3,89379	Holt Expor Additive	0,55341	0	0	2020-202	0,37134	0,40816	7,67259
Denmark	CO2 per c	2028	3,64916	Holt Expor Additive	0,55341	0	0	2020-202	0,37134	0,40816	7,67259
Finland	CO2 per c	2025	5,28933	Holt Expor Additive	0,4112	0	0	2020-202	0,84148	0,90299	14,1828
Finland	CO2 per c	2026	5,00995	Holt Expor Additive	0,4112	0	0	2020-202	0,84148	0,90299	14,1828
Finland	CO2 per c	2027	4,73058	Holt Expor Additive	0,4112	0	0	2020-202	0,84148	0,90299	14,1828
Finland	CO2 per c	2028	4,45121	Holt Expor Additive	0,4112	0	0	2020-202	0,84148	0,90299	14,1828
France	CO2 per c	2025	3,95602	Holt Expor Additive	1,5E-08	0	0	2020-202	0,23133	0,28221	5,57599
France	CO2 per c	2026	3,82904	Holt Expor Additive	1,5E-08	0	0	2020-202	0,23133	0,28221	5,57599
France	CO2 per c	2027	3,70206	Holt Expor Additive	1,5E-08	0	0	2020-202	0,23133	0,28221	5,57599
France	CO2 per c	2028	3,57509	Holt Expor Additive	1,5E-08	0	0	2020-202	0,23133	0,28221	5,57599
Germany	CO2 per c	2025	6,45867	Holt Expor Additive	0,46409	0,35643	0	2020-202	1,28213	1,34536	17,5256
Germany	CO2 per c	2026	6,07871	Holt Expor Additive	0,46409	0,35643	0	2020-202	1,28213	1,34536	17,5256
Germany	CO2 per c	2027	5,69875	Holt Expor Additive	0,46409	0,35643	0	2020-202	1,28213	1,34536	17,5256
Germany	CO2 per c	2028	5,31878	Holt Expor Additive	0,46409	0,35643	0	2020-202	1,28213	1,34536	17,5256
Greece	CO2 per c	2025	5,13406	Holt Expor Additive	1	0	0	2020-202	0,36373	0,43744	6,87287
Hungary	CO2 per c	2026	4,00402	Holt Expor Additive	1	0	0	2020-202	0,37397	0,47403	8,74971
Hungary	CO2 per c	2027	3,93819	Holt Expor Additive	1	0	0	2020-202	0,37397	0,47403	8,74971
Hungary	CO2 per c	2028	3,87236	Holt Expor Additive	1	0	0	2020-202	0,37397	0,47403	8,74971
Ireland	CO2 per c	2025	6,09682	Holt Expor Additive	0,87576	0	0	2020-202	0,2393	0,26269	3,8937
Ireland	CO2 per c	2026	5,86165	Holt Expor Additive	0,87576	0	0	2020-202	0,2393	0,26269	3,8937
Ireland	CO2 per c	2027	5,62648	Holt Expor Additive	0,87576	0	0	2020-202	0,2393	0,26269	3,8937
Ireland	CO2 per c	2028	5,39131	Holt Expor Additive	0,87576	0	0	2020-202	0,2393	0,26269	3,8937
Italy	CO2 per c	2025	4,95832	Holt Expor Additive	0,99689	0	0	2020-202	0,28303	0,3214	5,27287
Italy	CO2 per c	2026	4,82866	Holt Expor Additive	0,99689	0	0	2020-202	0,28303	0,3214	5,27287
Italy	CO2 per c	2027	4,699	Holt Expor Additive	0,99689	0	0	2020-202	0,28303	0,3214	5,27287
Italy	CO2 per c	2028	4,56934	Holt Expor Additive	0,99689	0	0	2020-202	0,28303	0,3214	5,27287
Netherlands	CO2 per c	2025	5,81298	Holt Expor Additive	0,65105	0,27831	0	2020-202	1,28917	1,36009	18,8447
Netherlands	CO2 per c	2026	5,38024	Holt Expor Additive	0,65105	0,27831	0	2020-202	1,28917	1,36009	18,8447
Netherlands	CO2 per c	2027	4,9475	Holt Expor Additive	0,65105	0,27831	0	2020-202	1,28917	1,36009	18,8447
Netherlands	CO2 per c	2028	4,51476	Holt Expor Additive	0,65105	0,27831	0	2020-202	1,28917	1,36009	18,8447
Norway	CO2 per c	2025	6,52119	Holt Expor Additive	0,4323	0,4323	0	2020-202	0,47887	0,52421	6,73954
Norway	CO2 per c	2026	6,23473	Holt Expor Additive	0,4323	0,4323	0	2020-202	0,47887	0,52421	6,73954
Norway	CO2 per c	2027	5,94827	Holt Expor Additive	0,4323	0,4323	0	2020-202	0,47887	0,52421	6,73954
Norway	CO2 per c	2028	5,66181	Holt Expor Additive	0,4323	0,4323	0	2020-202	0,47887	0,52421	6,73954
Poland	CO2 per c	2025	7,02954	Holt Expor Additive	1	0	0	2020-202	0,62153	0,75078	8,33158
Poland	CO2 per c	2026	6,97897	Holt Expor Additive	1	0	0	2020-202	0,62153	0,75078	8,33158
Poland	CO2 per c	2027	6,92839	Holt Expor Additive	1	0	0	2020-202	0,62153	0,75078	8,33158
Poland	CO2 per c	2028	6,87782	Holt Expor Additive	1	0	0	2020-202	0,62153	0,75078	8,33158
Portugal	CO2 per c	2025	3,29211	Holt Expor Additive	0,92465	0	0	2020-202	0,6117	0,62174	16,4746
Portugal	CO2 per c	2026	3,16873	Holt Expor Additive	0,92465	0	0	2020-202	0,6117	0,62174	16,4746
Portugal	CO2 per c	2027	3,04536	Holt Expor Additive	0,92465	0	0	2020-202	0,6117	0,62174	16,4746
Portugal	CO2 per c	2028	2,92199	Holt Expor Additive	0,92465	0	0	2020-202	0,6117	0,62174	16,4746
Slovakia	CO2 per c	2025	5,33125	Holt Expor Additive	0,50769	0	0	2020-202	0,43152	0,44762	7,61517
Slovakia	CO2 per c	2026	5,22848	Holt Expor Additive							

## 2.4) Результати перевірки зміни кластерної належності країн на основі прогнозів, отриманих методом Exponential Smoothing

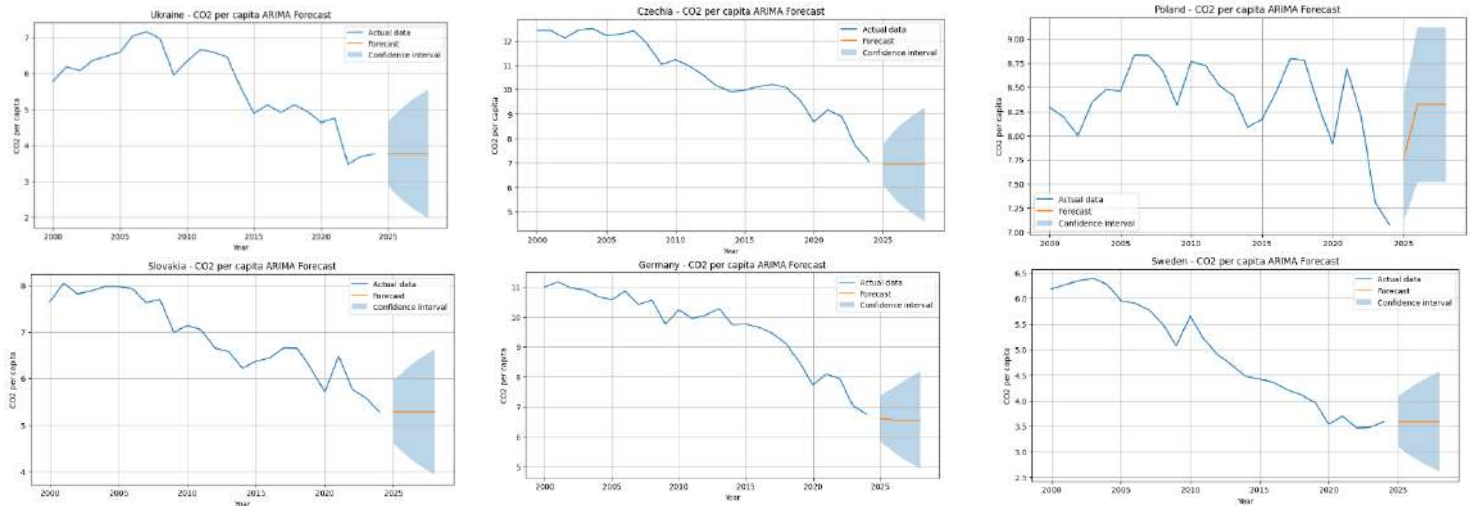
Country	CO2 capita mean	CO2 total mean	ES CO2 capita forecast mean	ES CO2 total forecast mean	ES CO2 capita forecast	ES CO2 total forecast	historical_cluster means	forecast_cluster means	min_distance	cluster_status	distance_to_cluster_1	distance_to_cluster_2	distance_to_cluster_3	distance_to_cluster_4
1 Belgium	1,258204838	-0,39359174	6,672251273	80747738,73	-0,436372993	-0,530410429	1	3	0,70147727	changed	1,315694959	1,336001782	0,701477275	3,88744904
2 Croatia	-1,42847686	-0,8398891	4,787838724	19129749,57	-1,428903896	-0,838515487	3	3	0,33823451	stable	2,328766	1,92111909	0,338234513	4,635880537
3 Czechia	1,654649865	-0,37883588	6,483021962	68138934,43	-0,536041157	-0,593457543	1	3	0,5891806	changed	1,416789447	1,40268383	0,589180597	3,98582619
4 Denmark	0,124214273	-0,71887307	4,016102082	25051555,13	-1,835382013	-0,808905007	1	1	0,72981642	changed	2,729003886	2,137133555	0,72981642	4,848453587
5 Finland	1,163197495	-0,67377555	4,870266742	25901530,42	-1,385488585	-0,804654922	1	3	0,2864663	changed	2,281415765	1,869785572	0,286466295	4,583721952
6 France	-0,99759141	0,84108866	3,76555143	249138434,3	-1,967348479	0,311584348	2	3	1,34452523	changed	2,968438	1,609899652	1,344525235	4,118714821
8 Germany	1,128432617	3,0281909	5,888728999	512310574,1	-0,849058671	1,627509591	4	2	0,92027094	changed	2,764363519	0,92027094	2,36650185	2,42329939
9 Greece	0,249123175	-0,5008586	4,869111236	47868100,87	-1,386097197	-0,694816682	1	3	0,27723044	changed	2,271414051	1,776397086	0,277230445	4,492621247
10 Hungary	-1,2275328	-0,6773802	3,97110262	38031155,22	-1,859083488	-0,744003823	3	3	0,74887564	stable	2,746762695	2,104501151	0,748875635	4,811933658
11 Ireland	0,869415126	-0,72890658	5,744064632	31964723,24	-0,925254223	-0,774337471	1	1	0,1917928	changed	1,82105619	1,654209212	0,191792797	4,321672324
12 Italy	-0,34911207	1,09334049	4,763829573	286640043,6	-1,441549653	0,499101591	2	2	1,05206612	stable	2,538756434	1,052066117	1,267614443	3,605704039
13 Netherlands	1,080506272	-0,12660617	5,163868153	98375344,86	-1,230847037	-0,442268075	1	3	0,30684452	changed	2,111987633	1,479979262	0,306844524	4,196461129
14 Norway	0,635263887	-0,71773431	6,091499936	34649045,12	-0,742258055	-0,760915203	1	1	0,36999883	changed	1,637953146	1,596332301	0,369998832	4,225731732
15 Poland	0,444554779	0,66040669	6,953680082	269341156,4	-0,288142824	0,412602913	2	2	0,41957209	stable	1,500427033	0,41957209	1,40329454	2,974556525
16 Portugal	-1,22796324	-0,66428798	3,107048021	3458455,1	-2,314186003	-0,766868445	3	3	1,20446944	stable	3,202287874	2,450210155	1,204469436	5,123875333
17 Slovakia	-0,31867036	-0,74761064	5,177095215	28678840,57	-1,223880268	-0,79076769	3	3	0,13135365	stable	2,119318841	1,78059216	0,131353651	4,485289361
18 Spain	-0,63768829	0,4949736	4,29273744	212236995,1	-1,689676583	0,127068101	2	3	1,02989149	changed	2,651670633	1,426293717	1,029891489	4,044533716
19 Sweden	-1,34966472	-0,69928484	2,94583835	33798494,86	-2,399096062	-0,765168163	3	3	1,28927569	stable	3,28684899	2,514855652	1,289275693	5,180060964
20 Ukraine	-0,96850103	0,37202116	3,542158945	117965564,8	-2,085010583	-0,34431217	2	1	0,04608635	changed	2,970136171	2,011076063	1,046086353	4,658325255
21 United Kingdom	-0,10236154	1,37171398	3,87241046	276784774,4	-1,911065215	0,449822829	2	2	1,42127598	changed	2,957382703	1,518221049	1,421275984	3,985790888

### ДОДАТОК 3

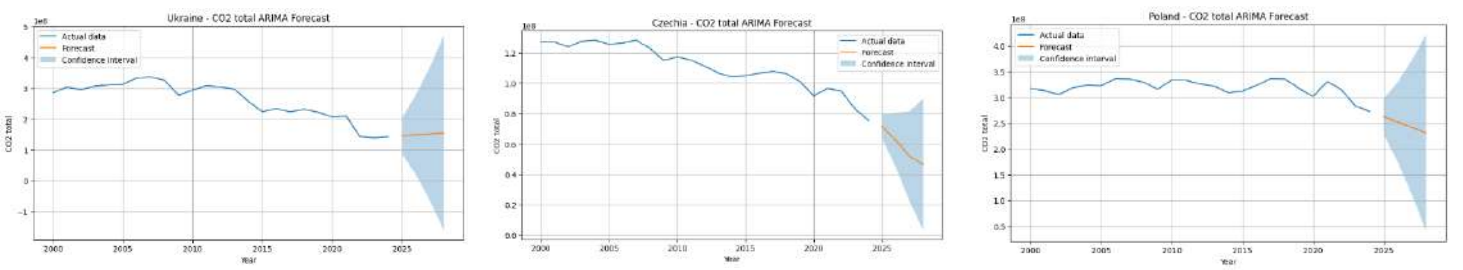
## 3.1) Результати кластеризації країн за показниками викидів CO<sub>2</sub>

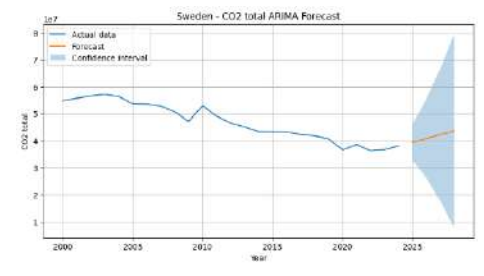
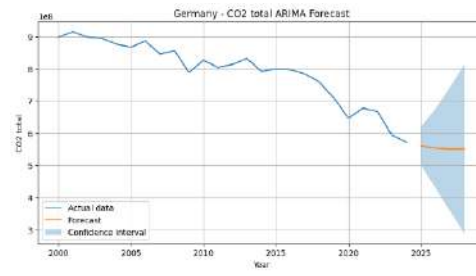
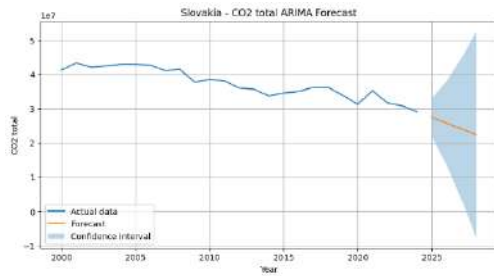


## 3.2.1) Прогнозування CO<sub>2</sub> per capita методом ARIMA для України, Чехії, Польщі, Словаччини, Німеччини та Швеції

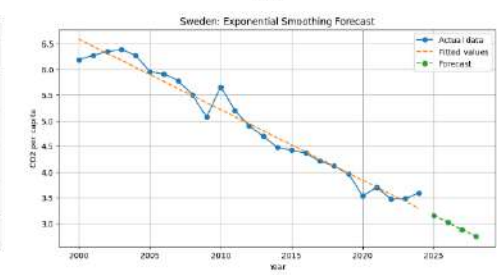
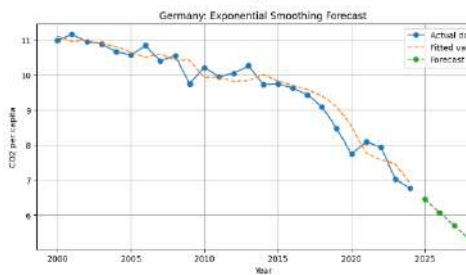
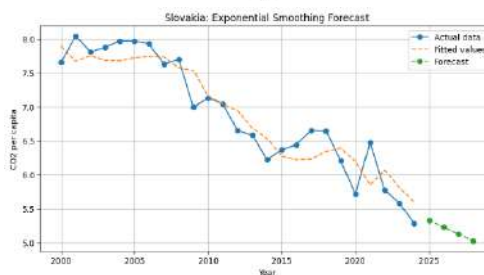
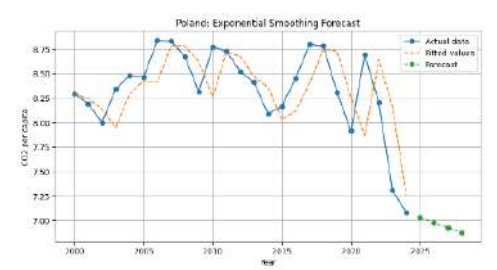
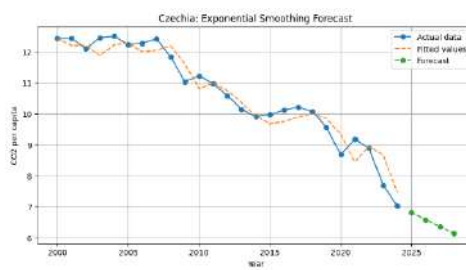
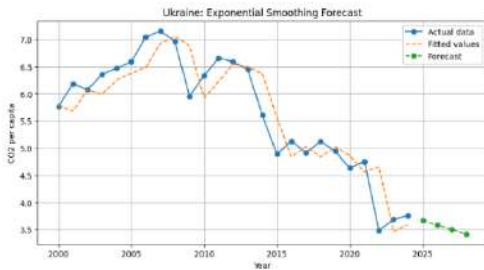


## 3.2.2) Прогнозування CO<sub>2</sub> total методом ARIMA для України, Чехії, Польщі, Словаччини, Німеччини та Швеції





### 3.3.1) Прогнозування CO<sub>2</sub> per capita методом Exponential Smoothing для України, Чехії, Польщі, Словаччини, Німеччини та Швеції



### 3.3.2) Прогнозування CO<sub>2</sub> total методом Exponential Smoothing для України, Чехії, Польщі, Словаччини, Німеччини та Швеції

