Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



Звіт до розрахунково-графічної роботи

з дисципліни “Основи смарт технологій та систем”  
на тему:

**«Life Expectancy (WHO)»**

**Виконала:** студентка ОІ-32

**Дребот Соломія**

**Прийняв:** асистент кафедри АСУ

**Бадзь Вікторія**

Львів – 2024

**Зміст**

[Вступ 3](#_Toc181460336)

[1. Вибір датасету та опис бізнес-проблеми 4](#_Toc181460337)

[1.1. Контекст 4](#_Toc181460338)

[1.2. Цей набір даних має на меті відповісти на такі ключові питання: 5](#_Toc181460339)

[1.3. Назви колонок та їх розшифровування 5](#_Toc181460340)

[2. Аналіз даних 7](#_Toc181460341)

[3. Опрацювання даних 13](#_Toc181460342)

[3.1. Видалення рядочків, які мають більше ніж 5 відсутніх значень 13](#_Toc181460343)

[3.2. Видалення 10-ти колонок, які мають інформацію лише про одну країну за 2013 рік 13](#_Toc181460344)

[3.3. Опрацювання числових колонок 14](#_Toc181460345)

[3.4. Опрацювання аномалій 20](#_Toc181460346)

[3.5. Перетворення категоріальних значень в числові 22](#_Toc181460347)

[3.6. Видалення не потрібних колонок 22](#_Toc181460348)

[3.7. Мінімізація даних 23](#_Toc181460349)

[4. Формування поділу датасету 24](#_Toc181460350)

[5. Вибір метрик для валідації 25](#_Toc181460351)

[6. Тренування моделі 26](#_Toc181460352)

[7. Вибір найкращої моделі 27](#_Toc181460353)

[8. Підбір гіперпараметрів 29](#_Toc181460354)

[9. Аналіз важливості ознак 31](#_Toc181460355)

[10. Створення скриптів для розгортання та використання моделі 32](#_Toc181460356)

[10.1. Скрипт для тренування моделі: 32](#_Toc181460357)

[10.2. Скрипт для використання моделі на нових даних: 34](#_Toc181460358)

[Висновки 37](#_Toc181460359)

[Додаток 38](#_Toc181460360)

Вступ

**Тема:** прогнозування та аналіз даних за допомогою регресійних моделей

**Мета:** освоїти методи регресійного аналізу для прогнозування та аналізу різноманітних даних. Набути навички у визначенні відповідних регресійних моделей для конкретних даних, здійснювати їх налаштування та оцінку. Навчитися аналізувати результати регресійних моделей, перевіряти їх на адекватність та виявляти можливі проблеми, такі як перенавчання, недонавчання та вплив викидів.

1. Вибір датасету та опис бізнес-проблеми

Для виконання розрахунково-графічної роботи я вибрала датасет «Life Expectancy (WHO)» на сайті Kaggle[1].

# Контекст

Хоча в минулому було проведено багато досліджень щодо факторів, які впливають на тривалість життя, враховуючи демографічні змінні, структуру доходів та рівень смертності, вони не враховували вплив імунізації та індексу людського розвитку. Було виявлено, що вплив імунізації та індексу людського розвитку в минулому не враховувався. Крім того, деякі з минулих досліджень були проведені з урахуванням множинної лінійної регресії на основі набору даних за один рік для всіх країн. Таким чином, це дає мотивацію для вирішення обох факторів, зазначених раніше, шляхом формулювання регресійної моделі на основі моделі змішаних ефектів та множинної лінійної регресії з урахуванням даних за період з 2000 по 2015 рік для всіх країн. Також будуть розглянуті такі важливі види імунізації, як вакцинація проти гепатиту В, поліомієліту та дифтерії. Коротше кажучи, це дослідження буде зосереджене на факторах імунізації, факторах смертності, економічних факторах, соціальних факторах та інших факторах, пов'язаних зі здоров'ям, а також на інших факторах, пов'язаних зі здоров'ям. Оскільки спостереження в цьому наборі даних базуються на різних країнах, країні буде легше визначити фактор, який сприяє зниженню очікуваної тривалості життя. Це допоможе підказати країні, на яку сферу слід звернути особливу увагу, щоб ефективно підвищити очікувану тривалість життя населення. Середня абсолютна відсоткова похибка (MAPE), також відома як середнє абсолютне відсоткове відхилення (MAPD), є оціночною метрикою для регресійних задач. Ідея цієї метрики полягає в тому, щоб бути чутливою до відносних помилок. Наприклад, вона не змінюється при глобальному масштабуванні цільової змінної.

Проект покладається на точність даних. Репозиторій даних Глобальної обсерваторії охорони здоров'я (ГООЗ) Всесвітньої організації охорони здоров'я (ВООЗ) відстежує стан здоров'я, а також багато інших пов'язаних з ним факторів для всіх країн. Набори даних є загальнодоступними з метою аналізу даних про стан здоров'я. Набір даних про очікувану тривалість життя, фактори здоров'я для 193 країн був зібраний з того ж веб-сайту репозиторію даних ВООЗ, а відповідні економічні дані були зібрані з веб-сайту Організації Об'єднаних Націй. З усіх категорій факторів, пов'язаних зі здоров'ям, було обрано лише ті критичні фактори, які є більш репрезентативними. Було помічено, що за останні 15 років у секторі охорони здоров'я відбувся величезний розвиток, що призвів до покращення показників смертності населення, особливо в країнах, що розвиваються, порівняно з останніми 30 роками. Тому в цьому проекті ми розглянули дані за 2000-2015 роки для 193 країн для подальшого аналізу. Окремі файли даних були об'єднані в єдиний набір даних. При первинному візуальному огляді даних було виявлено деякі відсутні значення. Оскільки дані були отримані від ВООЗ, ми не виявили явних помилок. Відсутні дані були оброблені в програмному забезпеченні R за допомогою команди Missmap. Результат показав, що більшість відсутніх даних стосувалися чисельності населення, гепатиту В та ВВП. Відсутні дані стосувалися менш відомих країн, таких як Вануату, Тонга, Того, Кабо-Верде тощо. Знайти всі дані для цих країн було складно, і тому було вирішено виключити ці країни з остаточного набору даних моделі. Остаточний об'єднаний файл (остаточний набір даних) складається з 22 стовпців і 2938 рядків, що означає 20 прогнозних змінних. Усі змінні прогнозування було поділено на кілька широких категорій: фактори, пов'язані з імунізацією, фактори смертності, економічні фактори та соціальні фактори.

# Цей набір даних має на меті відповісти на такі ключові питання:

Чи справді різні фактори прогнозування, які були обрані спочатку, впливають на очікувану тривалість життя? Які прогнозні змінні насправді впливають на очікувану тривалість життя?

Чи повинна країна з нижчим значенням очікуваної тривалості життя (<65 років) збільшувати свої витрати на охорону здоров'я для того, щоб підвищити середню тривалість життя?

Як рівень дитячої та дорослої смертності впливає на очікувану тривалість життя?

Чи має очікувана тривалість життя позитивну чи негативну кореляцію зі звичками харчування, способом життя, фізичними вправами, курінням, вживанням алкоголю тощо?

Як впливає шкільна освіта на тривалість життя людини?

Чи має очікувана тривалість життя позитивний чи негативний зв'язок із вживанням алкоголю?

Чи в густонаселених країнах очікувана тривалість життя нижча?

Який вплив на очікувану тривалість життя має рівень охоплення вакцинацією?

# Назви колонок та їх розшифровування

1) Country: Країна.

2) Year: Рік.

3) Status: Статус країни, розвинена вона чи та, що розвивається.

4) Life Expectancy: Тривалість життя в роках.

5) Adult Mortality: Коефіцієнти смертності дорослого населення обох статей (ймовірність померти у віці від 15 до 60 років на 1000 населення).

6) Infant Deaths: Кількість дитячих смертей на 1000 населення.

7) Alcohol: Зареєстроване споживання алкоголю на душу населення (15+) (у літрах чистого спирту).

8) percentage expenditure: Видатки на охорону здоров'я у відсотках від валового внутрішнього продукту на душу населення (%).

9) Hepatitis B: Охоплення імунізацією проти гепатиту В (HepB) серед дітей віком 1 рік (%).

10) Measles: Кір - кількість зареєстрованих випадків на 1000 населення.

11) BMI: Середній індекс маси тіла всього населення.

12) under-five deaths: Кількість смертей дітей віком до п'яти років на 1000 населення.

13) Polio: Охоплення щепленнями проти поліомієліту (Pol3) серед дітей віком 1 рік (%).

14) Total expenditure: Загальні державні видатки на охорону здоров'я у відсотках від загальних державних видатків (%).

15) Diphtheria: Охоплення щепленнями АКДП (дифтерійно-правцевим анатоксином) та кашлюком (АКДП3) серед дітей віком 1 рік (%).

16) HIV/AIDS: Смертність на 1 000 живонароджених ВІЛ/СНІД (0-4 роки).

17) GDP: Валовий внутрішній продукт на душу населення (у доларах США).

18) Population: Кількість населення в країні.

19) thinness 10-19 years: Поширеність худорлявості серед дітей та підлітків віком від 10 до 19 років (%).

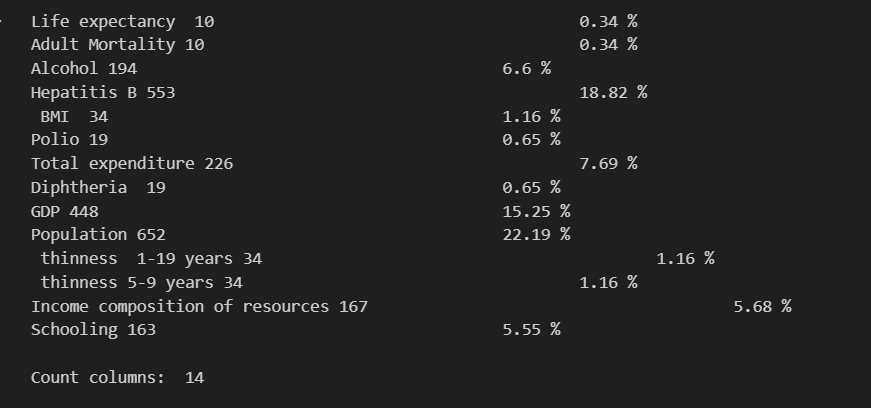
20) thinness 5-9 years: Поширеність худорлявості серед дітей віком від 5 до 9 років (%).

21) Income composition: Індекс людського розвитку в розрізі структури ресурсів за доходами (індекс змінюється від 0 до 1).

22) Schooling: Кількість років навчання (років).

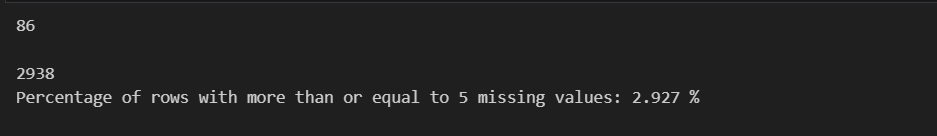
1. Аналіз даних

Спочатку я вирішила виясни кількість відсутніх значень у колонках:



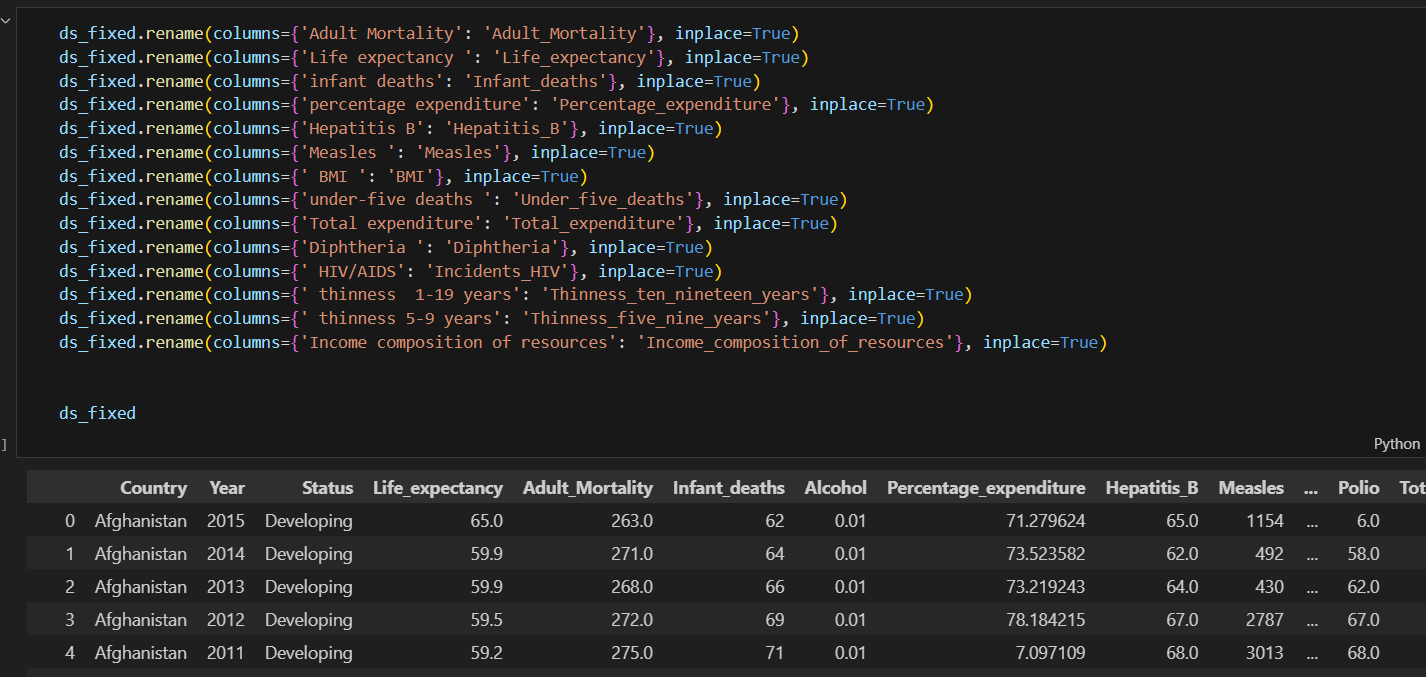
*Рис. 1 Кількість відсутніх значень у колонках.*

Після цього скільки рядочків мають більше ніж 5 відсутніх значень у різних колонках:



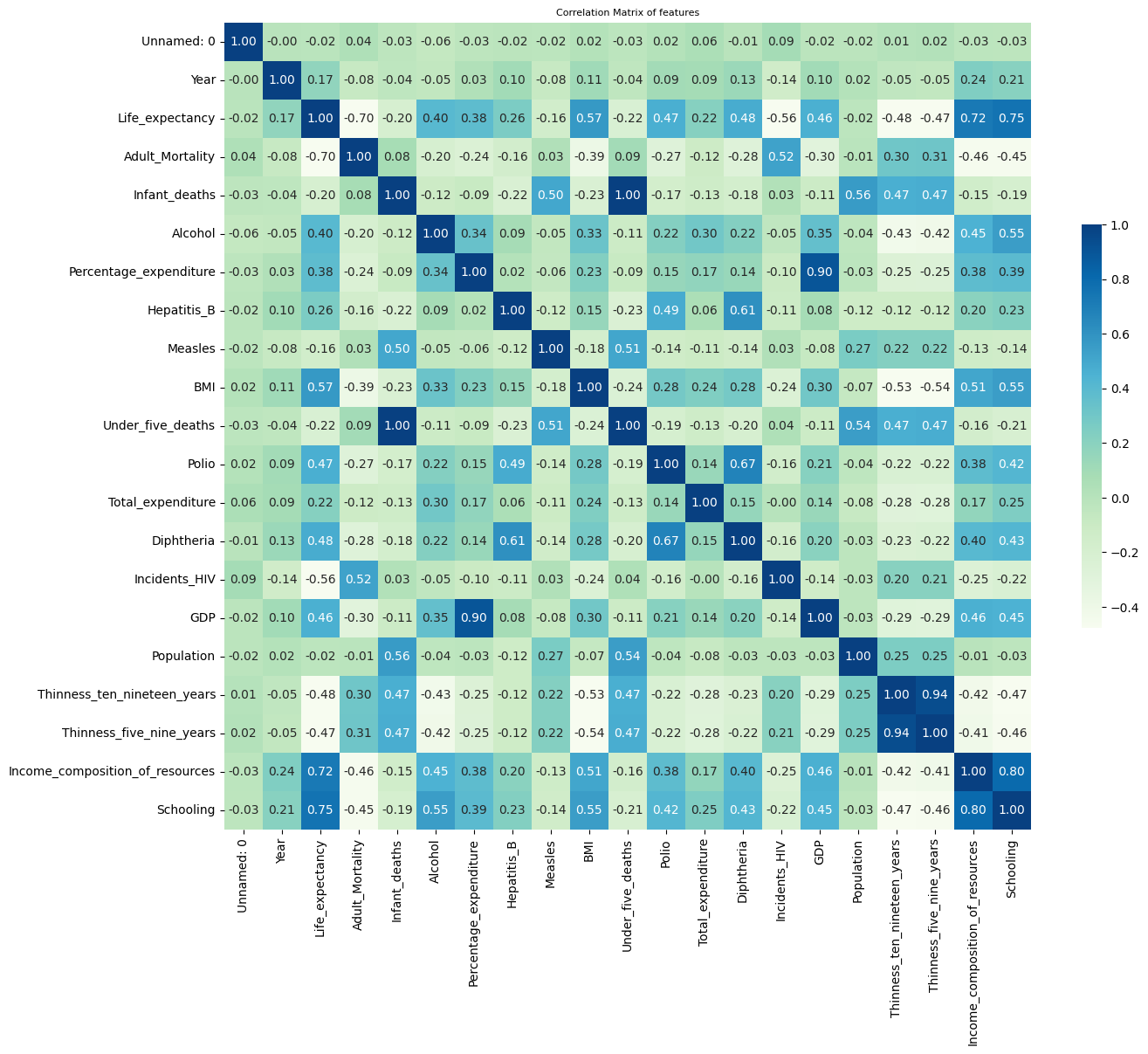
*Рис. 2 Кількість рядочків, в яких 5-ти різних колонках.*

Після цього я вирішила трохи поміняти назви колонок, тому що застосування цього датасету зі старими назвами було проблематичним, через їх некоректне написання.



*Рис. 3 Переписання колонок та їх вивід.*

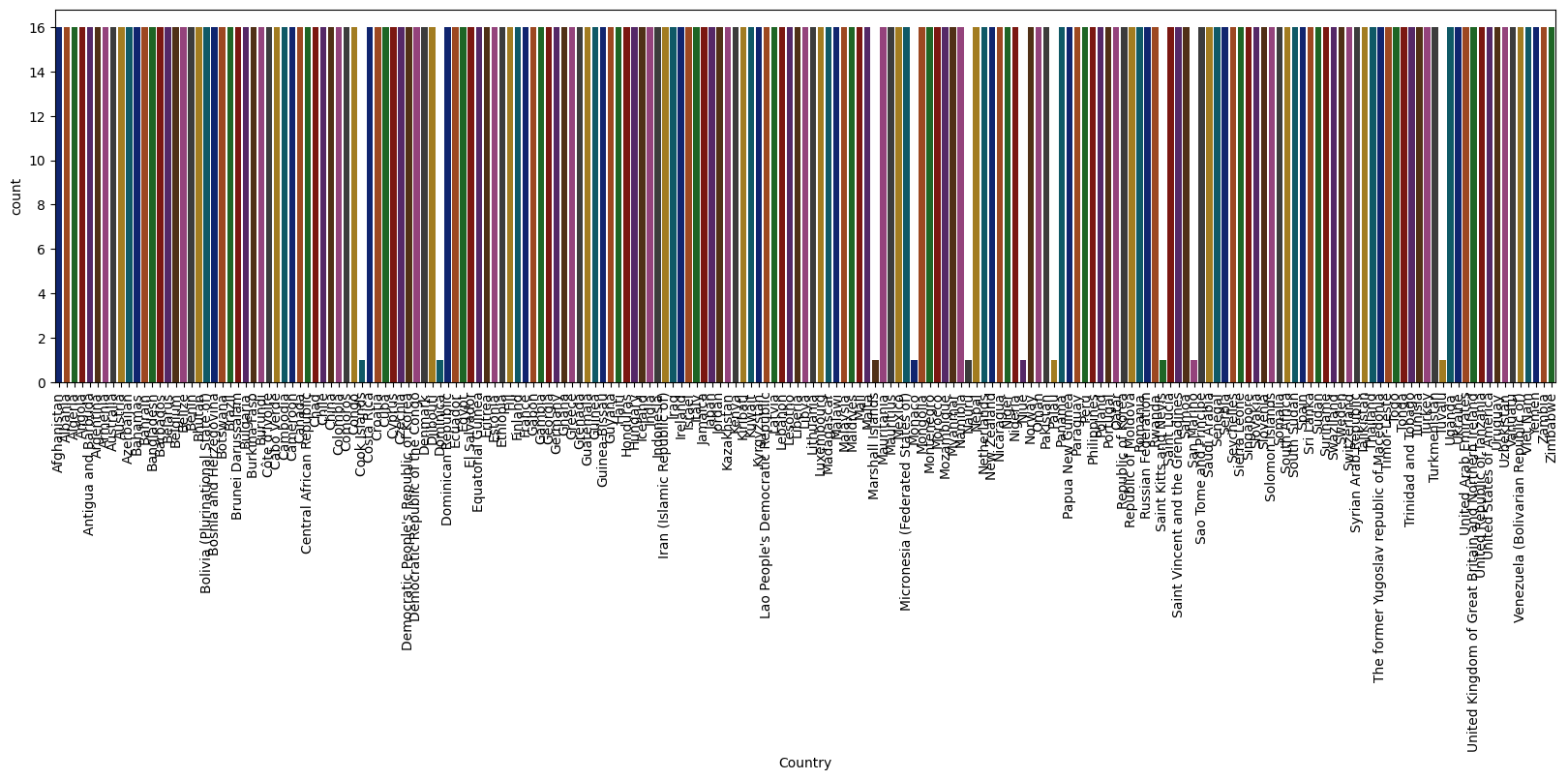
Вивела до всіх числових даних таблицю кореляції:



*Рис. 4 Матриця кореляції.*

Звідси можна замітити, що колонки «Under\_five\_deaths» та «Infant\_deaths» напряму залежать один від одного, тому в майбутньому я видалю колонку «Infant\_deaths», тому що вона входить в «Under\_five\_deaths».

Після я досліджувала скільки країн має інформацію про 15 років спостережень:



*Рис. 5 Графік, який виводить країни та за скільки років про них є інформація.*

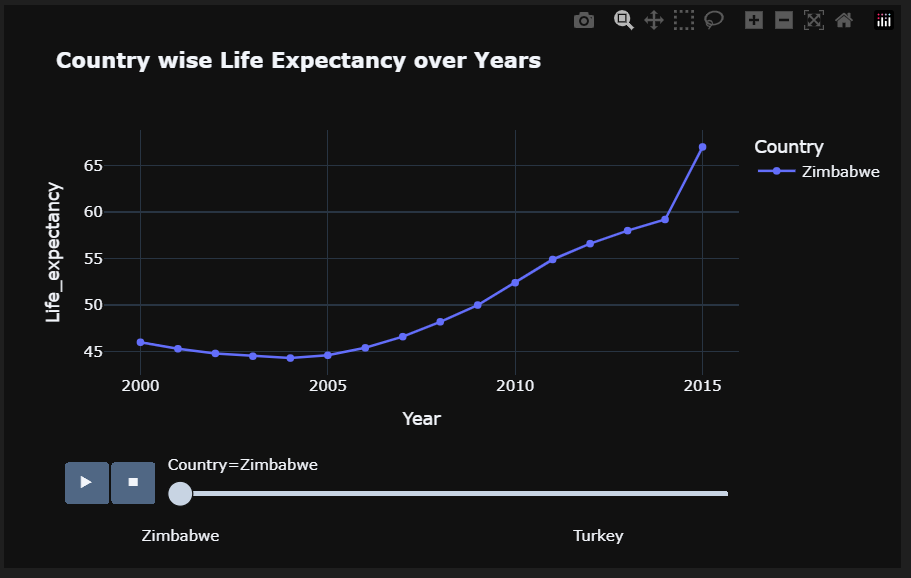
Я замітила, що є 10 країн, про яких відомо лише за один рік:



*Рис. 6 10 країн, про яких відомо лише за один рік.*

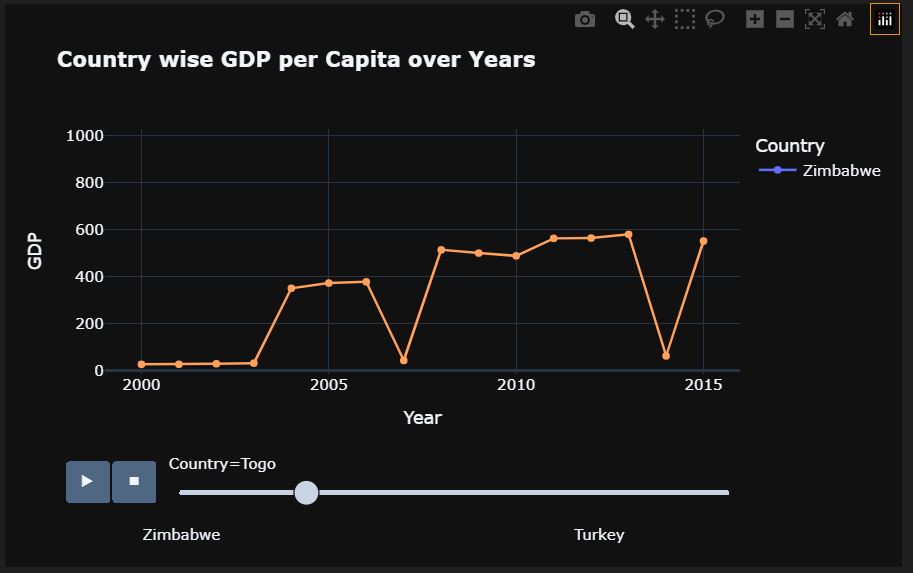
Оскільки для цих десяти країн є інформація лише за один рік (2013), я незабаром видалю їх.

Далі я досліджувала популяцію кожної країни за всі роки:



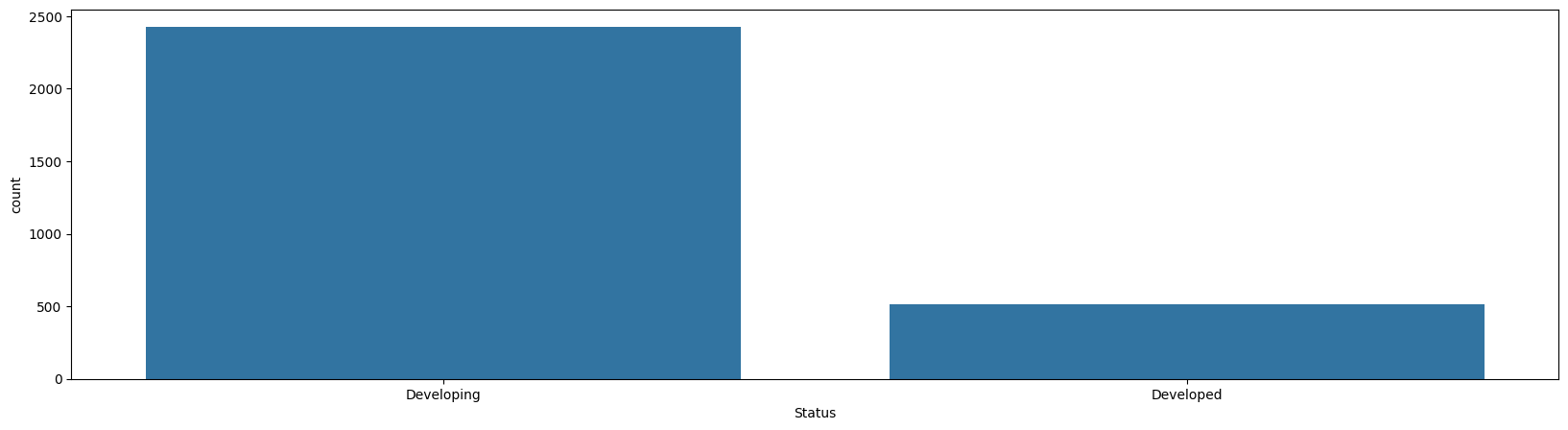
*Рис. 7 Інформація про популяцію за всі роки в країні.*

Також досліджувала валовий внутрішні продукт на душу населення за всі роки в країні.



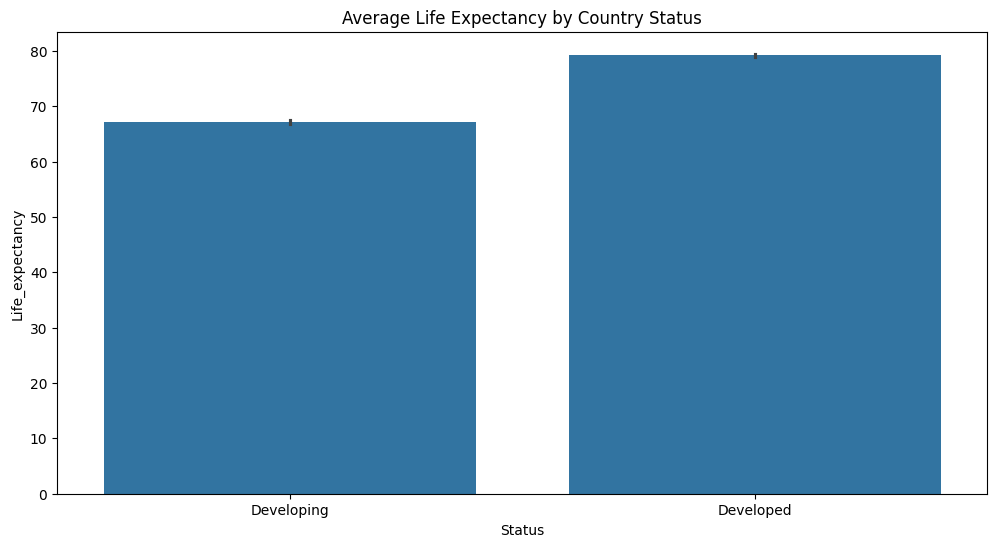
*Рис. 8 Інформація про валовий внутрішні продукт на душу населення в країні за всі роки.*

Також цікаво було дізнатися кількість розвинених країн та тих, що розвиваються:



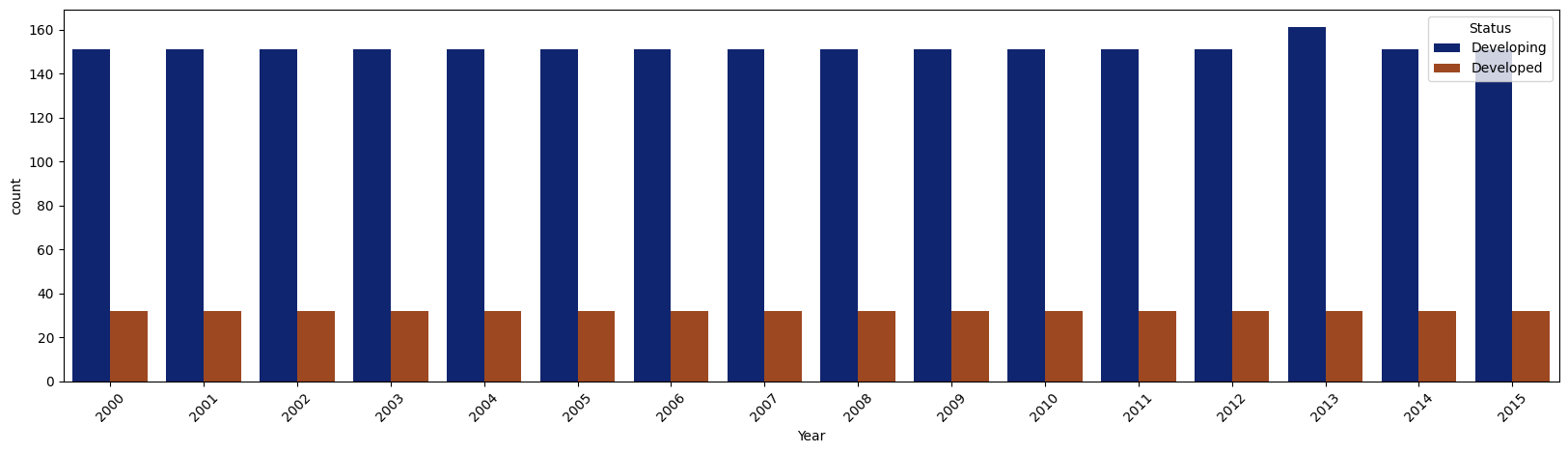
*Рис. 9 Графік кількості розвинених країн та тих, що розвиваються.*

Також чи впливає розвиненість країни на середню тривалість життя:



*Рис. 10 Графік середньої тривалості життя в залежності від того чи країна розвинена чи та, що розвивається.*

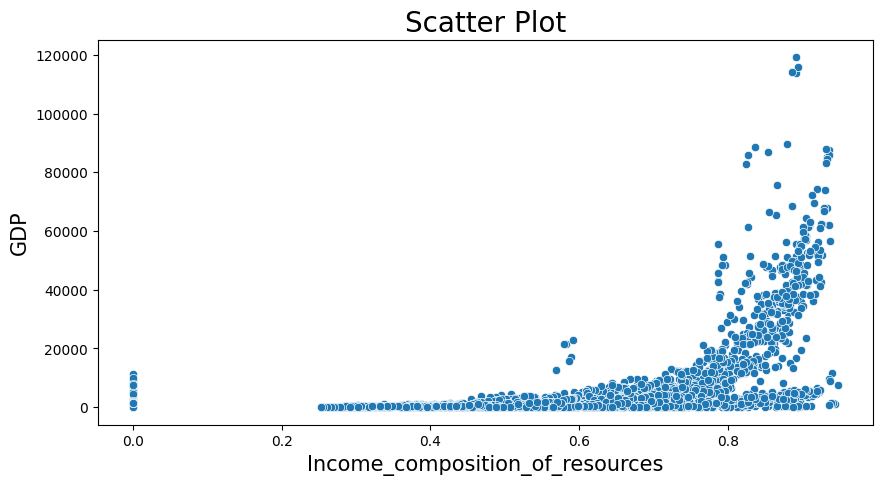
Також хотіла подивитися чи є якісь країни, які протягом якихось рокі змінили свій статус з тої, що розвивається до розвиненої:



*Рис. 11 Графік розвинених країн з 2000 до 2015 років.*

На графіку видно, що лише у 2013 році кількість слаборозвинених країн зросла, і саме ці країни я вилучу, щоб вони не впливали негативно на модель.

Також чи вирішила подивитися, чи впливає індекс доходів у структурі ресурсів на ВВП на душу населення.

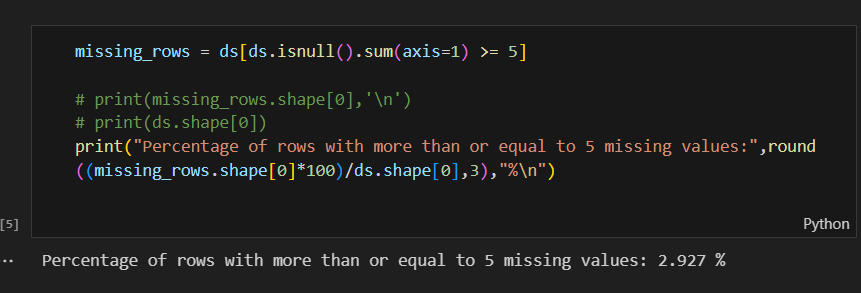


*Рис. 12*

1. Опрацювання даних

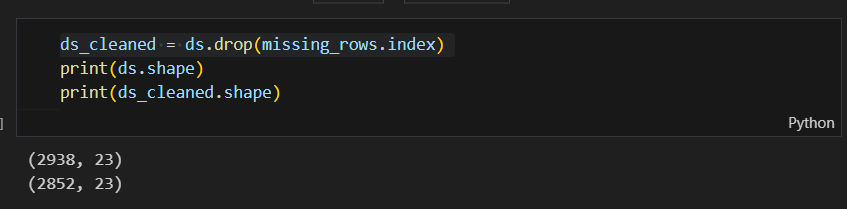
# Видалення рядочків, які мають більше ніж 5 відсутніх значень

Спочатку я вирішила видалити всі рядочків, які мають більше ніж 5 відсутніх значень, адже якщо потім замінювати їх різними методами, то вони будуть спотворенні. Для визначення кількості цих відсутніх колонок я запустила цей код та получила такий результат:



*Рис. 13 Виведення проценту відсутніх значень, яких більше 5-ти.*

З результату видно, що їх кількість не більше 3%, тому їх видалення не повинно вплинути на результати моделі, тому я їх видалила:



*Рис. 14 Кількість значень в датасеті до і після видалення рядочків.*

# Видалення 10-ти колонок, які мають інформацію лише про одну країну за 2013 рік

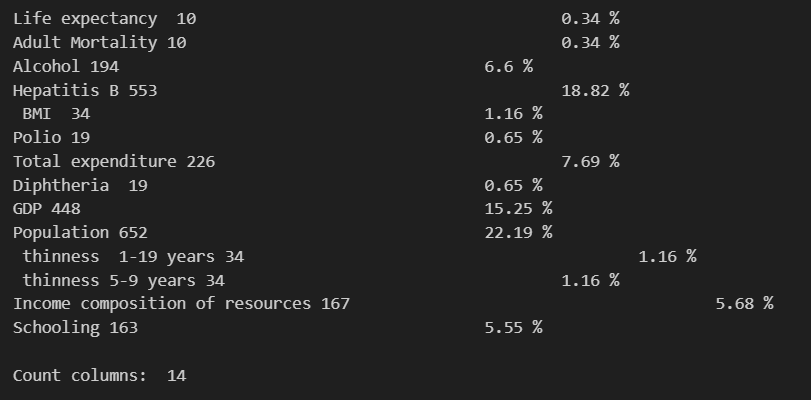
З попереднього розділу я дослідила, що є 10 рядочків, які позначають 10 різних країн за 2013 рік, з цього аналізу я вирішила їх також видалити, адже вони можуть не правильно вплинути на модель, через це вона може видавати більш хибні данні.



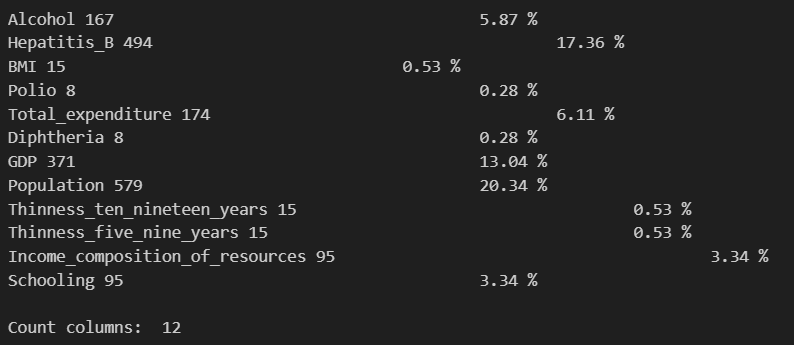
*Рис. 15 Кількість значень в датасеті до і після видалення рядочків.*

# Опрацювання числових колонок

Після проведення цих маніпуляцій кількість колонок, в яких були відсутні значення зменшилася з 14 до 12, що показує користь цих дій.

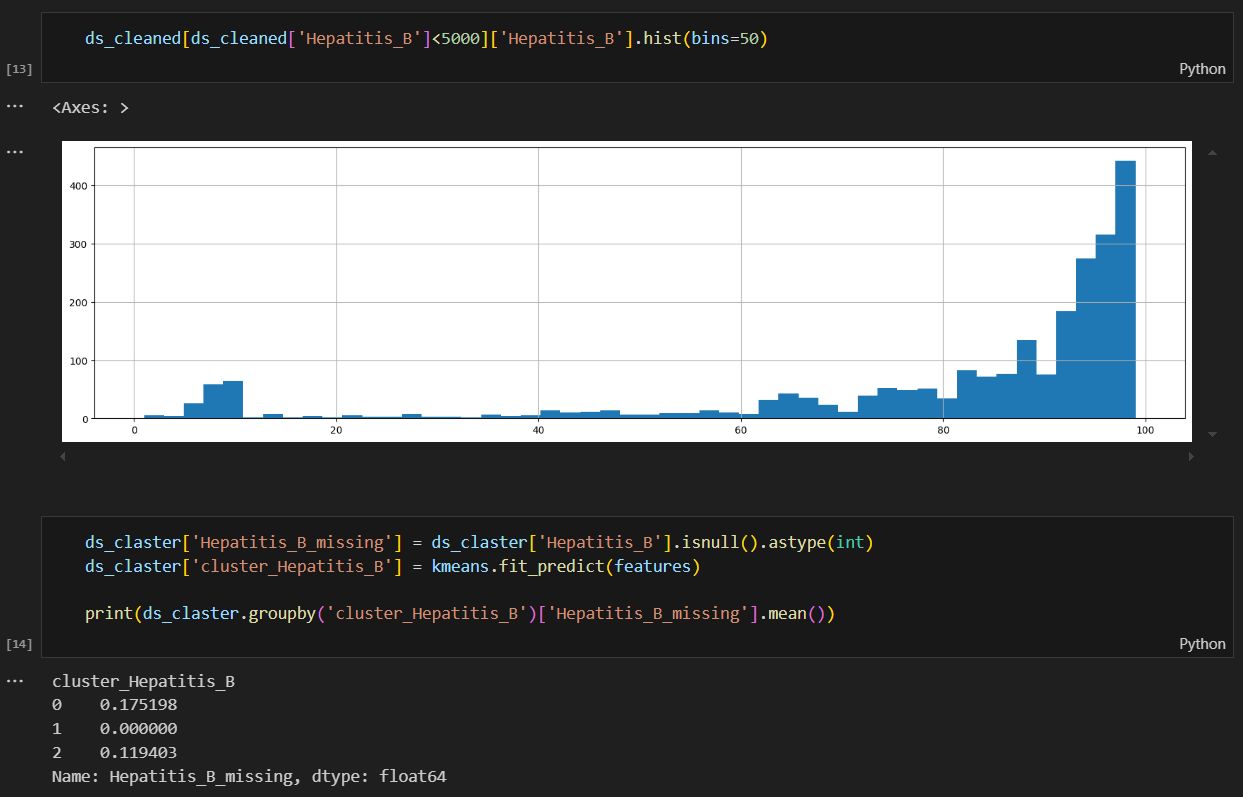


*Рис. 16 Кількість відсутніх значень в колонках до пророблених дій.*



*Рис. 17 Кількість відсутніх значень в колонках після пророблених дій.*

Для визначення якого методу застосувати, я спочатку виводила графік колонки і одразу ж після застосовувала метод кластеризації для визначення чи дані були відсутні випадково чи ні.



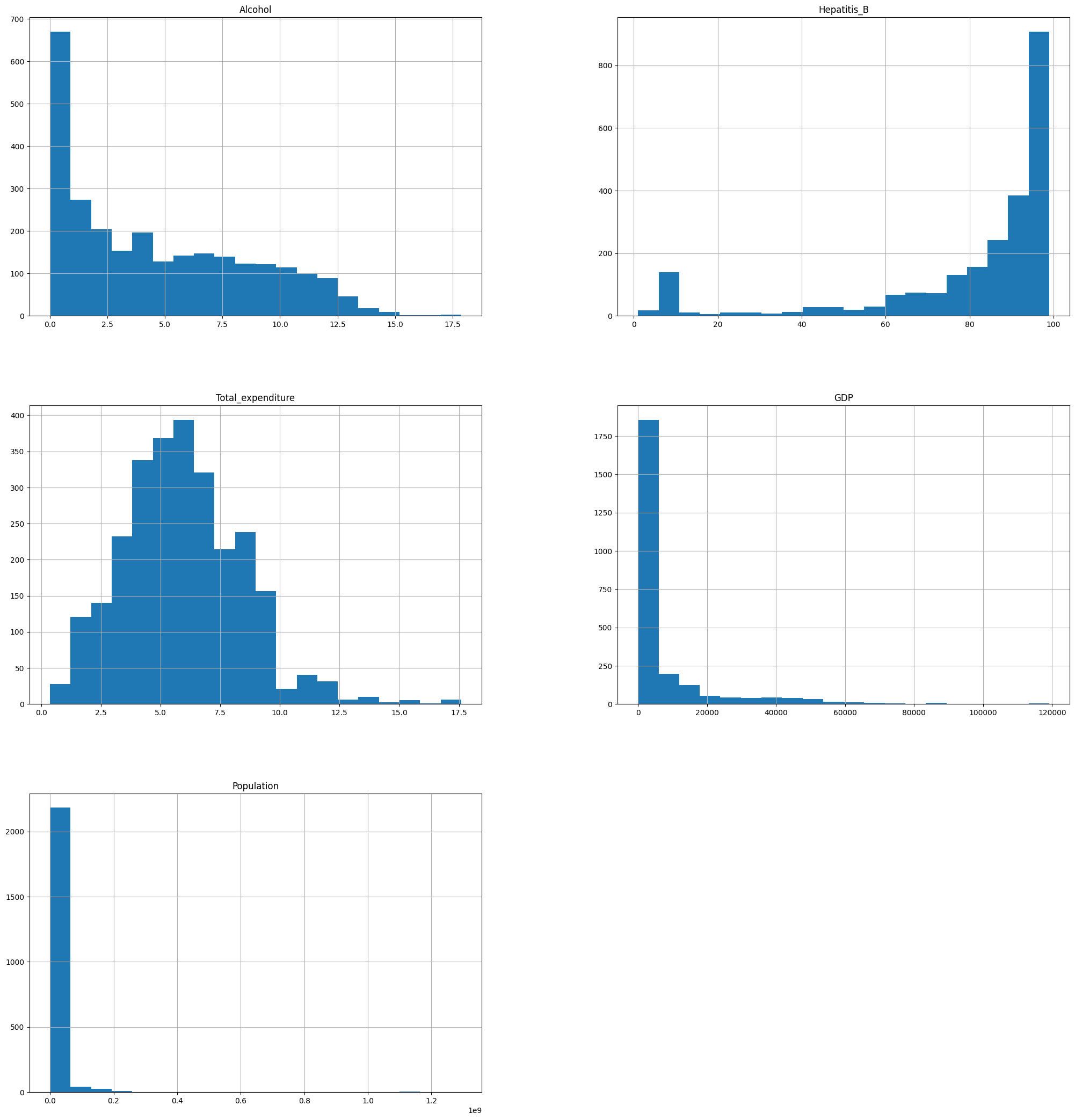
*Рис. 18 Приклад виведення графіку т застосування методу кластеризації на колонці «Hepatitis\_B».*

В результаті аналізу всіх колонок з відсутніми значеннями я вирішила їх поділити та застосувати відповідні методи:

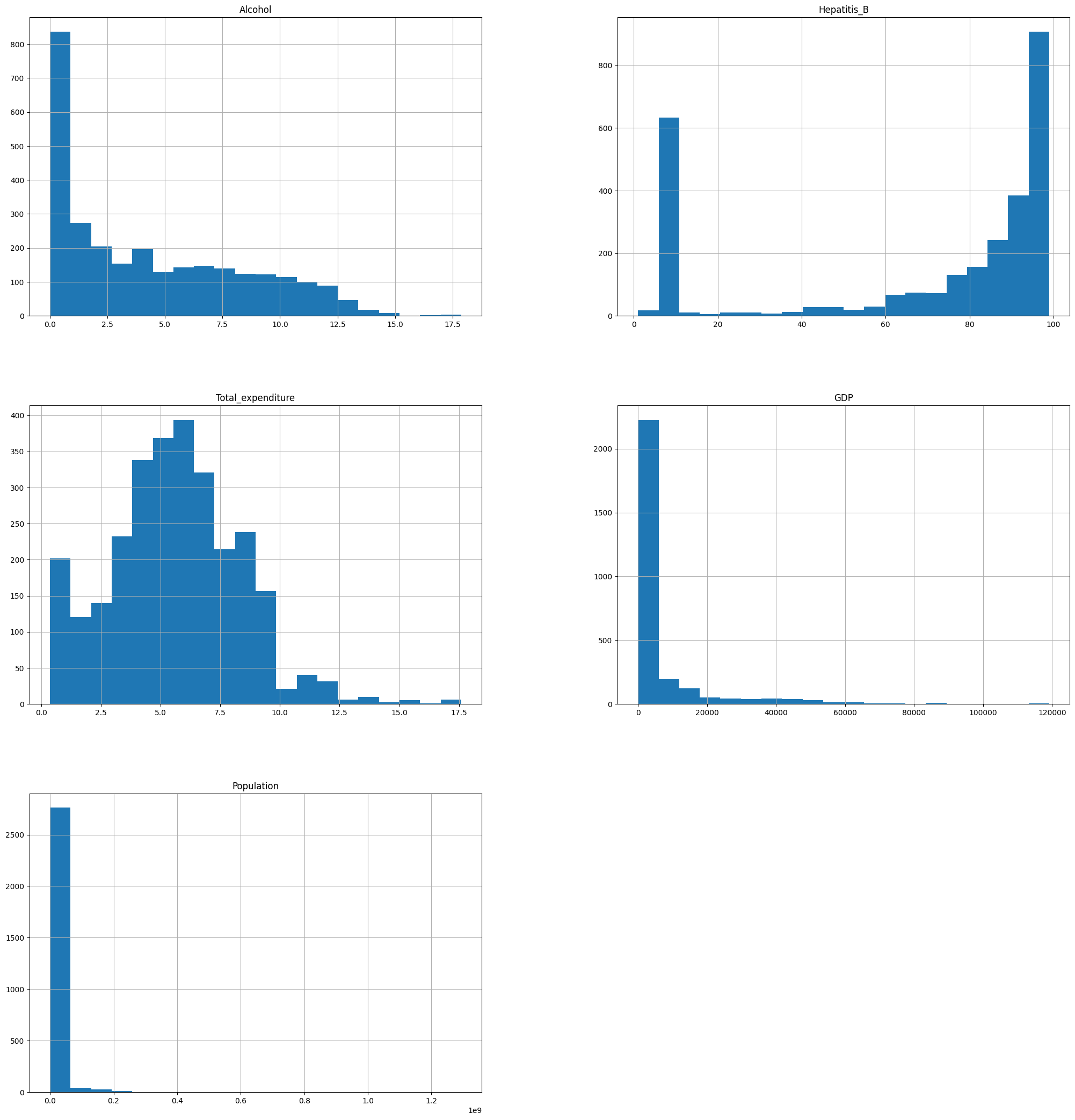
Для стовпчиків: Alcohol, Hepatitis\_B, Total\_expenditure, GDP, Population – я буду використовувати persentline імплікації.

Для стовпчиків: BMI, Polio, Diphtheria, Thinness\_ten\_nineteen\_years, Thinness\_five\_nine\_years, Income\_composition\_of\_resources, Schooling – я використаю середню/медіанну імплікацію.

Спочатку опрацюю колонки з методом persentline:

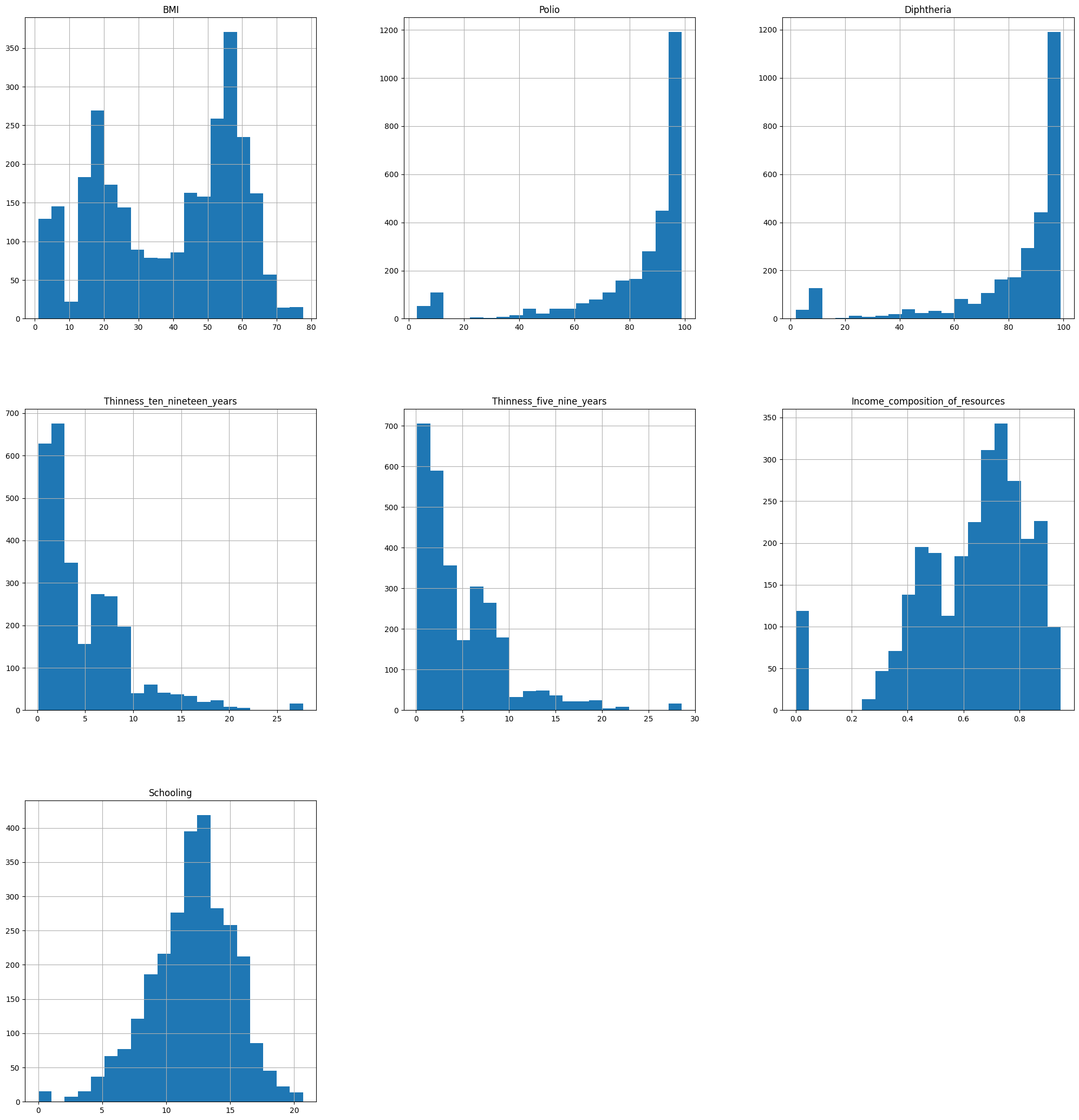


*Рис. 19 Колонки до застосування методу persentline.*

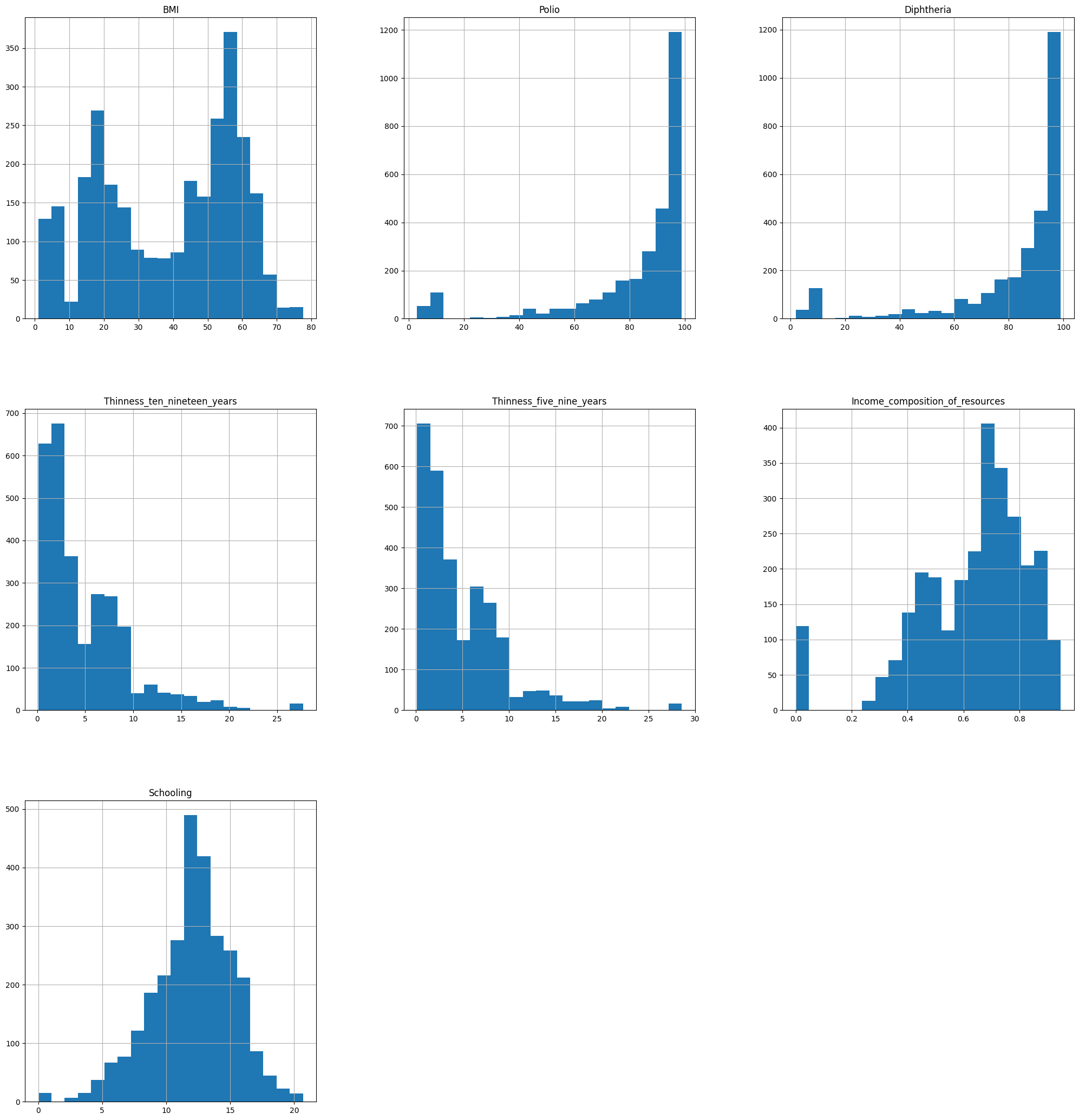


*Рис. 20 Колонки після застосування методу.*

Опрацювання колонок з методом заміни медіаною:

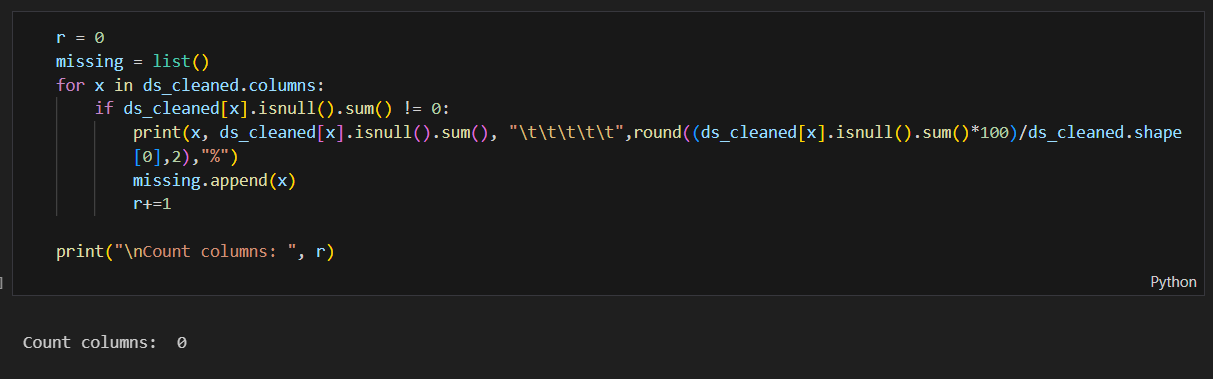


*Рис. 21 Колонки до застосування методу заміни медіаним значенням.*



*Рис. 22 Колонки після застосування даного методу.*

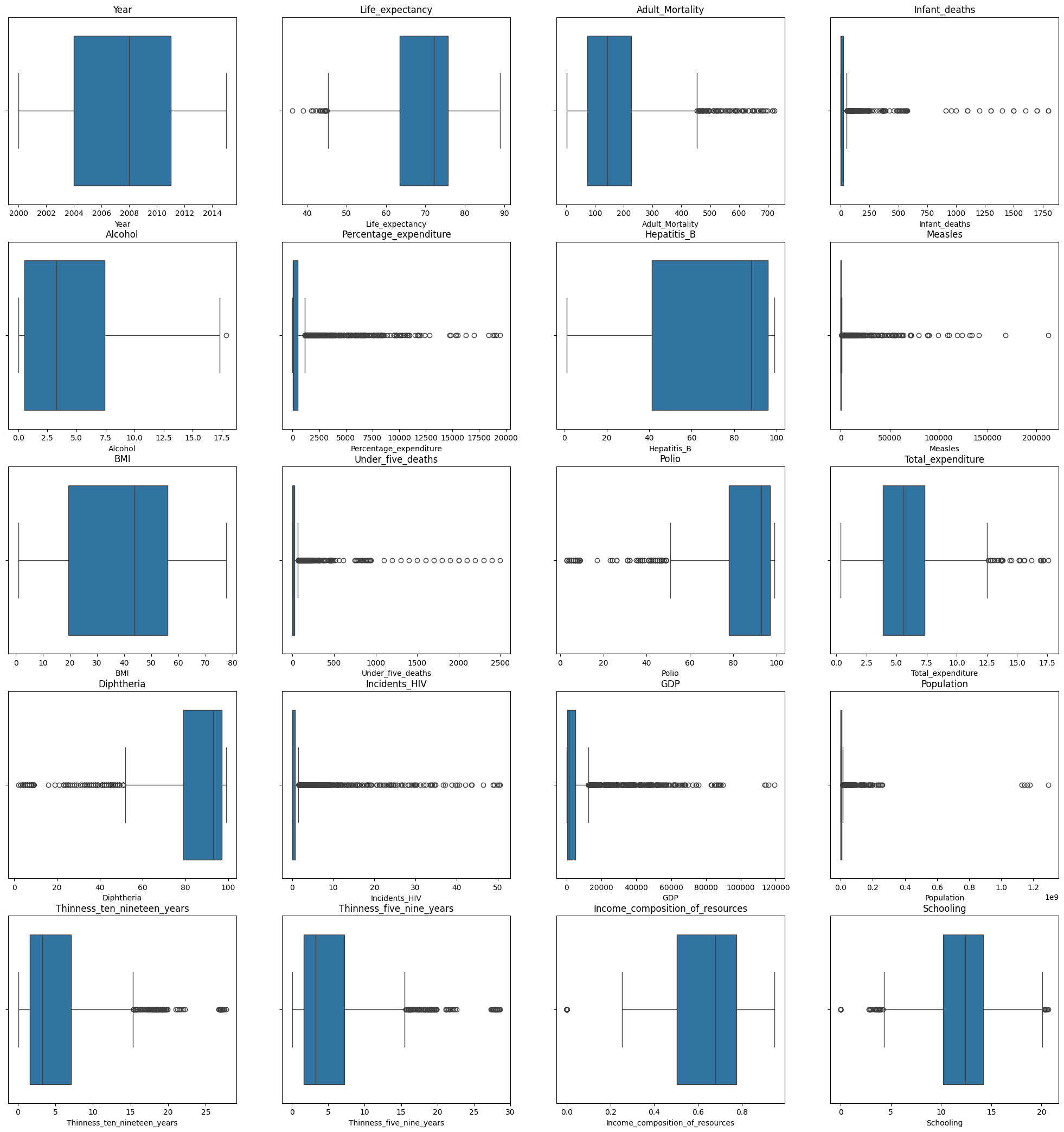
Після застосування цих методів кількість колонок, які мають відсутні значення зменшилася до 0.



*Рис. 23 Відображення колонок, в яких відсутні значення.*

# Опрацювання аномалій

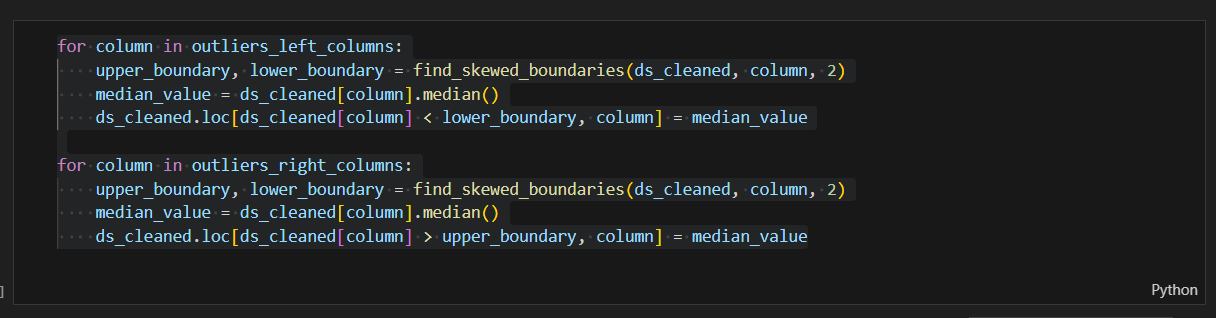
На роботу моделі можуть вплинути і аномалії, тому я вирішила дослідити чи вони є та в випадку їх присутності справитися з ними.



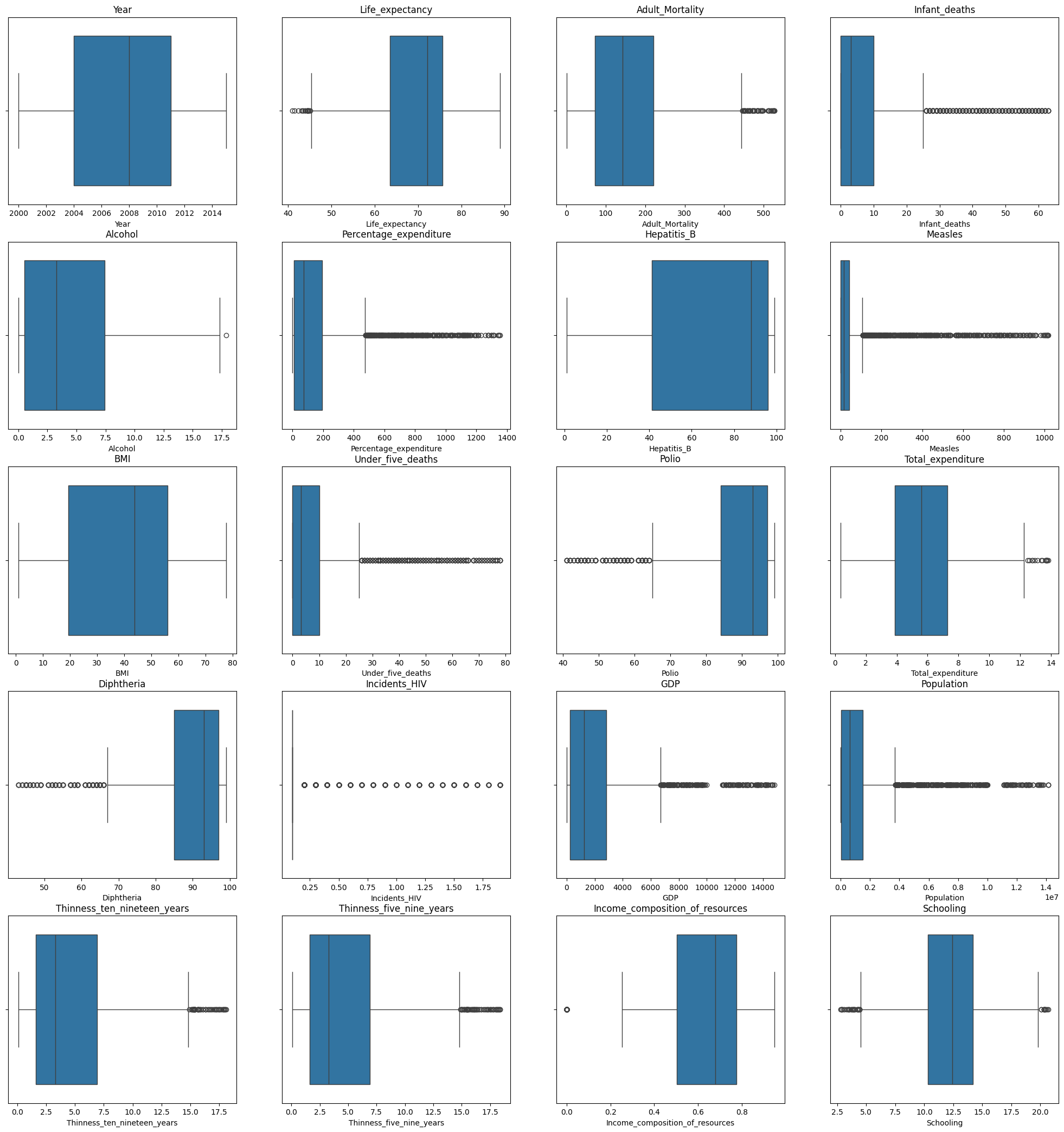
*Рис. 24 Виведення колонок та дослідження їх на аномальність.*

З цих графіків видно, що більшість колонок мають аномалії, тому для того, щоб з ними справитися, я вирішила їх не видаляти, а працювати з ними як зі значеннями, які відсутні, тому до них я застосувала метод заміни медіаним значення.

Але перед тим як застосовувати цей метод, я вирішила колонки, в яких присутні аномальні значення поділити на окремі частини, тобто до одного масиву заносити ті колонки, в яких аномальні значення зліва і до другої в яких аномальні значення з права, ну і в яких відсутні з обох боків заносити і туди і туди.



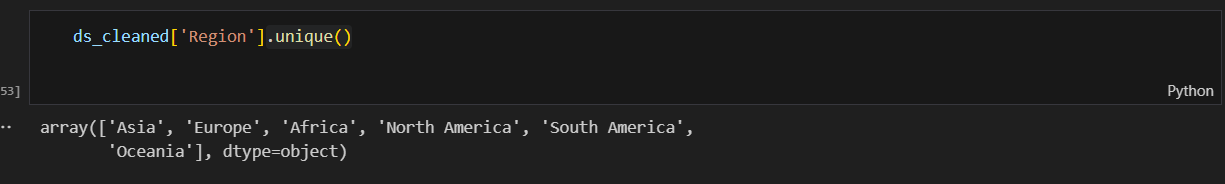
*Рис. 25 Опрацювання аномалій.*



*Рис. 26 Колонки після опрацювання аномалій.*

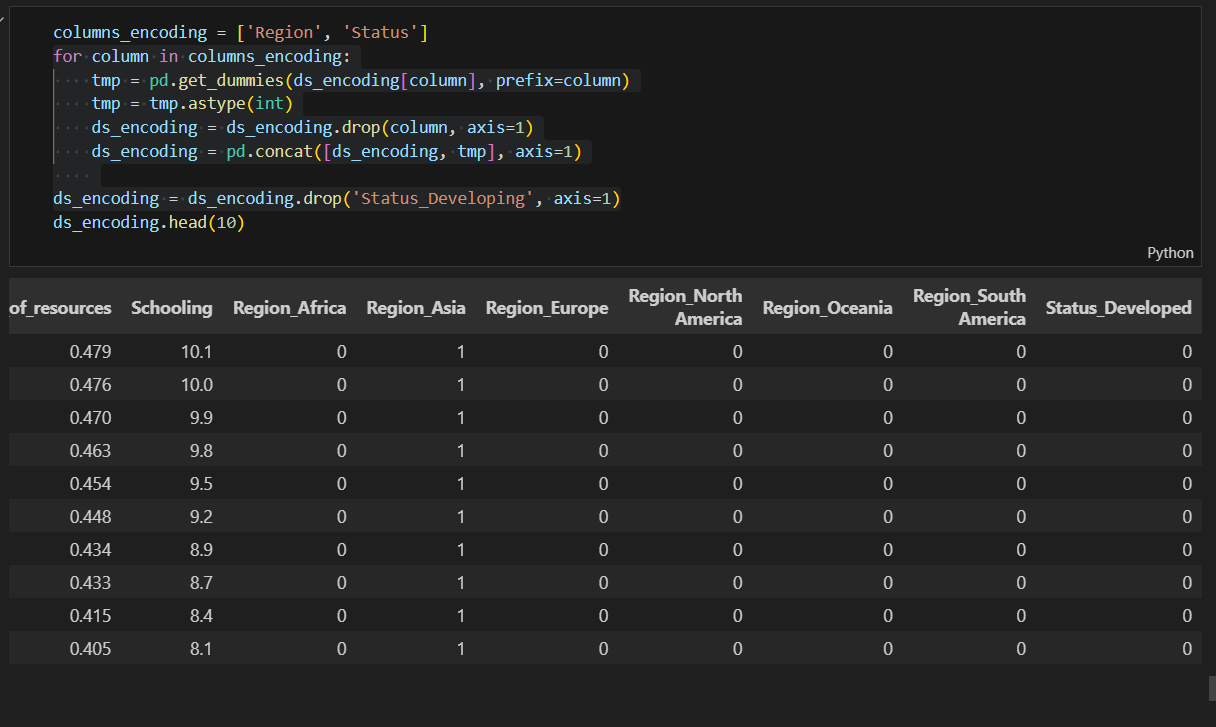
# Перетворення категоріальних значень в числові

Перед тим як перетворювати мої колонки з категоріальних в числові, у мене з однією виникла проблема, з колонкою країни. Застосування до неї більшості методів або погіршить роботу моделі (її продуктивність) або вона буде працювати не точно, через велику кількість країн, тому я вирішила їх класифікувати по регіонам розташування, що зменшило їх кількість та дає можливість застосувати методи перетворення даних в числові для кращої роботи моделі.



*Рис. 27 Всі регіони, на які були поділенні країни.*

Після категоризації колонки, я уже застосувала метод перетворення категоріальних значень в числові для двох моїх категоріальних колонок.

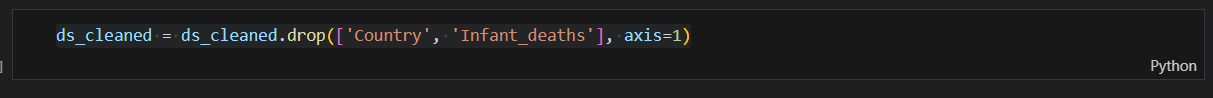


*Рис. 28 Створенні колонки після перетворення даних.*

# Видалення не потрібних колонок

В попередньому розділі при виведенні кореляції було видно, що колонки «Under\_five\_deaths» та «Infant\_deaths» напряму залежать один від одного, тому я вирішила видалити колонку «Infant\_deaths», тому що вона входить в колонку «Under\_five\_deaths».

Також я видаляю колонку з країнами, адже вона вже є не потрібна.

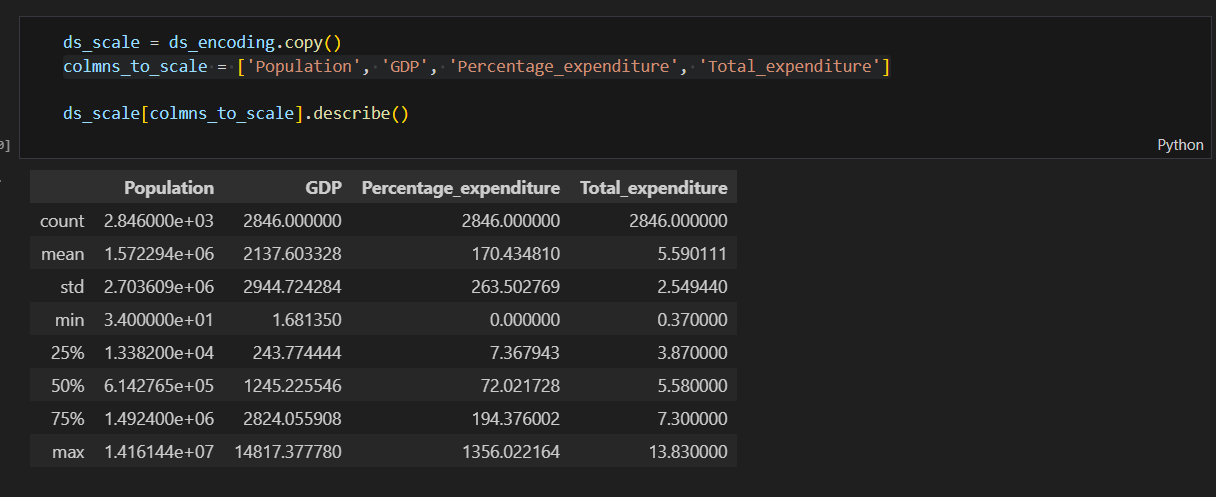


*Рис. 29 Видалення колонок.*

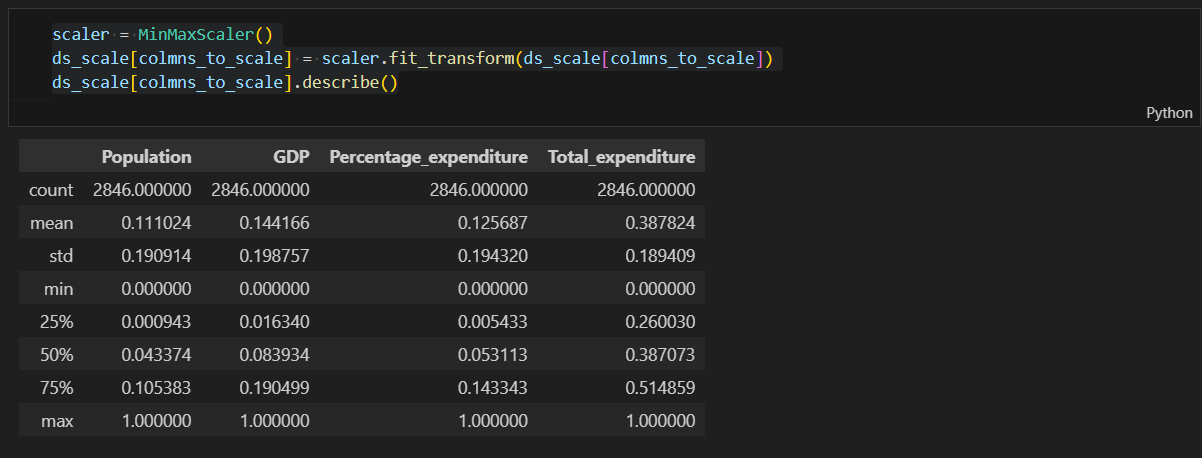
# Мінімізація даних

Для застосування методі регресії слід мінімізувати деякі дані для точної роботи моделі.

Спочатку я вибрала колонки, в яких великі значення, а тоді вже до них застосувала метод мінімізації:



*Рис. 30 Колонки до мінімізації.*



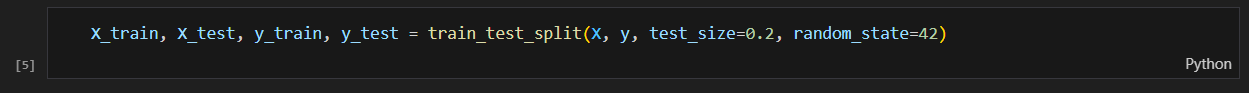
*Рис. 31 Колонки після мінімізації.*

Після цього я зберегла датасет в файлі для подальшого його використання.

1. Формування поділу датасету

Перед тим як ділити датасет на тренувальні та тестові набори, спочатку я визначила, що моєю цільовою зміною буде колонка, яка має назву «**Life\_expectancy**» – середня тривалість життя.

Після цього я поділила датасет на тренувальні та тестові набори, на тестові припадає 80% даних, на тренувальні відповідно 20%.



*Рис. 32 Ділення датасету на тренувальні та тестові.*

1. Вибір метрик для валідації

Для перевірки роботи моделі використовуються такі метрики:

**Root Mean Squared Error (RMSE).** Вимірює середню величину помилки передбачень моделі. Це середнє значення квадрату різниці між фактичними та передбаченими значеннями. Потрібно вибрати модель з найменшим RMSE для тестових даних, оскільки менше значення RMSE означає, що модель робить менш значні помилки в передбаченнях.

**Mean Absolute Error (MAE).** Вимірює середню величину абсолютної помилки між фактичними та передбаченими значеннями. Як і з RMSE, вибирайте модель з найменшим MAE для тестових даних.

**R-squared (R2).** Вимірює, яка частка варіації залежної змінної (цільової) пояснюється незалежними змінними (факторами). Значення варіюється від 0 до 1, де 1 означає, що модель ідеально описує дані. Потрібно вибрати модель з найвищим значенням R^2 для тестових даних, оскільки це вказує на кращу узгодженість між передбаченнями моделі та фактичними значеннями.

**Mean Squared Error (MSE).** Вимірює середню величину квадрату помилки між фактичними та передбаченими значеннями. На відміну від MAE, MSE більше штрафує великі відхилення, оскільки помилки підносяться до квадрату. Нижче значення MSE свідчить про те, що модель має кращу точність, оскільки це вказує на менші середні помилки передбачень. Для вибору моделі з найкращими характеристиками на тестових даних зазвичай також намагаються мінімізувати значення MSE, оскільки це вказує на більшу точність моделі у передбаченні цільової змінної.

Для валідації моделі я вибрала останню метрику **Mean Squared Error (MSE).**

1. Тренування моделі

Для тренування моделі я вибрала такі методи:

**XGBRegressor**: Потужна модель, заснована на градієнтному бустингу дерев рішень. Ефективна для складних даних, добре працює з нелінійними залежностями.

**LinearRegression**: Базова модель лінійної регресії, яка знаходить пряму, що мінімізує відстань між фактичними та передбаченими значеннями. Підходить для даних з лінійними зв’язками.

**Ridge**: Модель лінійної регресії з регуляризацією L2, яка зменшує значення коефіцієнтів для запобігання перенавчанню, корисна для даних з мультиколінеарністю.

**Lasso**: Модель лінійної регресії з регуляризацією L1, що обнуляє певні коефіцієнти, ефективна для відбору ознак у моделі.

**KNeighborsRegressor**: Алгоритм, що використовує схожі об’єкти для передбачень. Придатний для невеликих датасетів з локальними залежностями.

**RandomForestRegressor**: Багатодеревна модель, яка об’єднує кілька дерев рішень для покращення передбачень та зменшення ризику перенавчання.

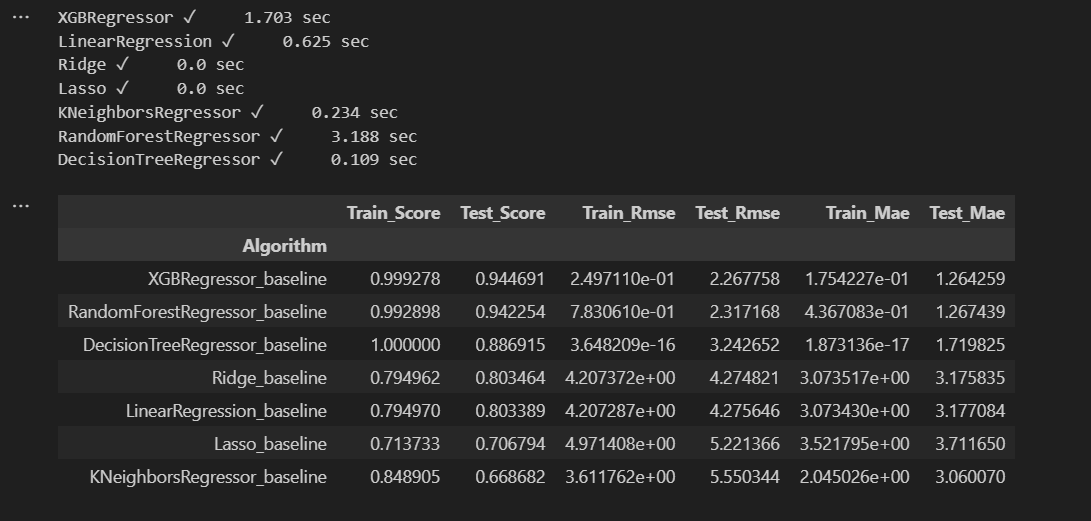
**DecisionTreeRegressor**: Алгоритм, що розбиває дані на групи для передбачення, схильний до перенавчання, але швидкий і зручний для інтерпретації.

1. Вибір найкращої моделі

Після тренування моделі використовуючи метрики для порівняння я виявила, що для моїх даних найкраще підходить метод «XGBRegressor».



*Рис. 33 Код для тренування моделей і виведення їх ефективності.*



*Рис. 34 Результат коду.*

Для вибору найкращої моделі були три такі: Score, RMSE та MAE.

Score вимірює, яка частина варіації залежної змінної пояснюється незалежними змінними, тому мені потрібно знайти найвищий тестовий бал.

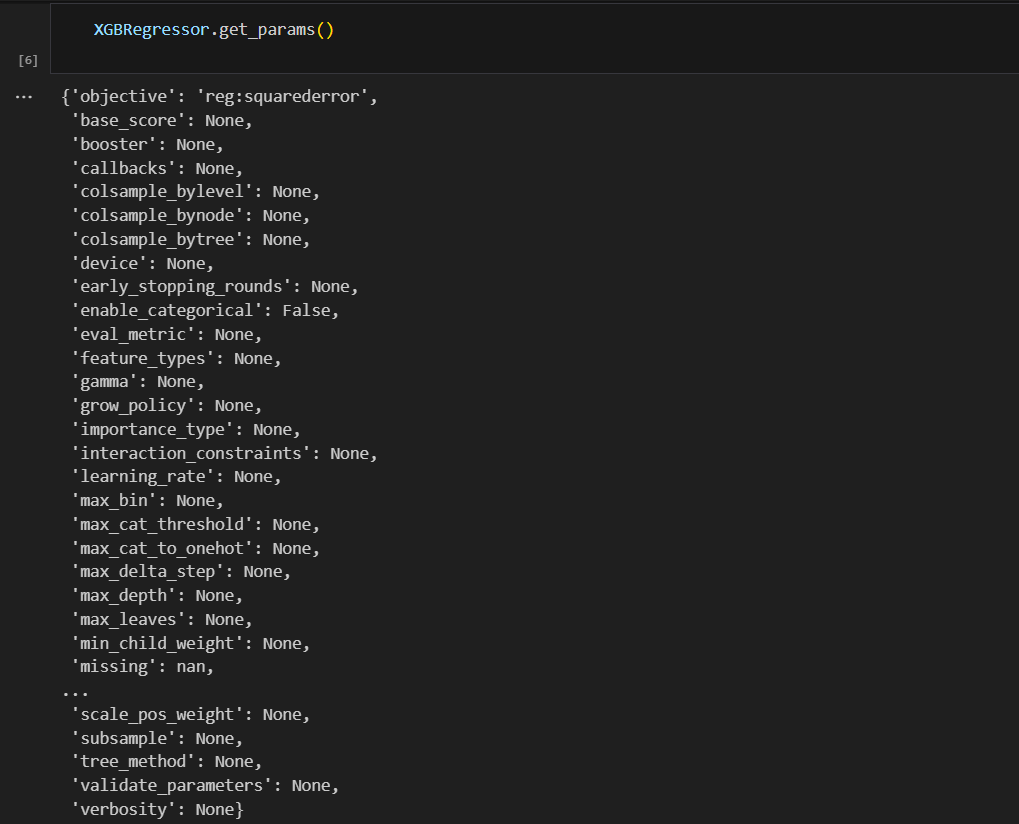
RMSE вимірює середню похибку прогнозів моделі. Це середнє квадратичне значення різниці між фактичним і прогнозованим значеннями, тому мені потрібно знайти найнижчий показник RMSE тесту.

MAE Вимірює середню абсолютну похибку між фактичними та передбачуваними значеннями, тому мені потрібно знайти найменше значення MAE.

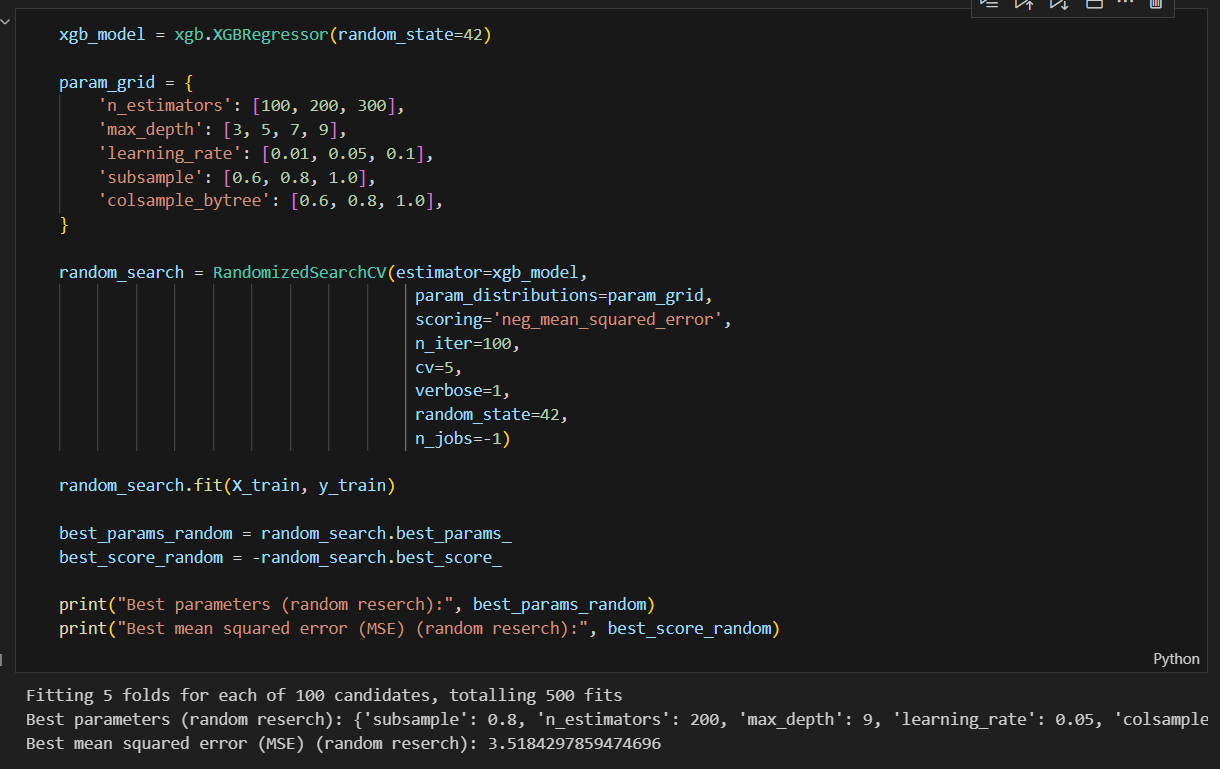
Отже, я обрала модель «XGBRegressor», оскільки вона відповідала всім моїм вимогам.

1. Підбір гіперпараметрів

Перед тим як підбирати гіперпараметри я вирішила спочатку ознайомитися, які з ними:

  
*Рис. 35 Виведення всіх параметрів методу.*

Після цього я вибрала декілька параметрів, які можуть покращити мою -модель, а також вирішила, що для їх пошуку я застосую метод випадкового пошуку.



*Рис. 36 Код для пошуку гіперпараметрів та виведення найкращих гіперпараметрів.*

У результаті найкращими гіперпаматерами виявилися:

param\_grid = {

    'n\_estimators': 200,

    'max\_depth': 9,

    'learning\_rate': 0.05,

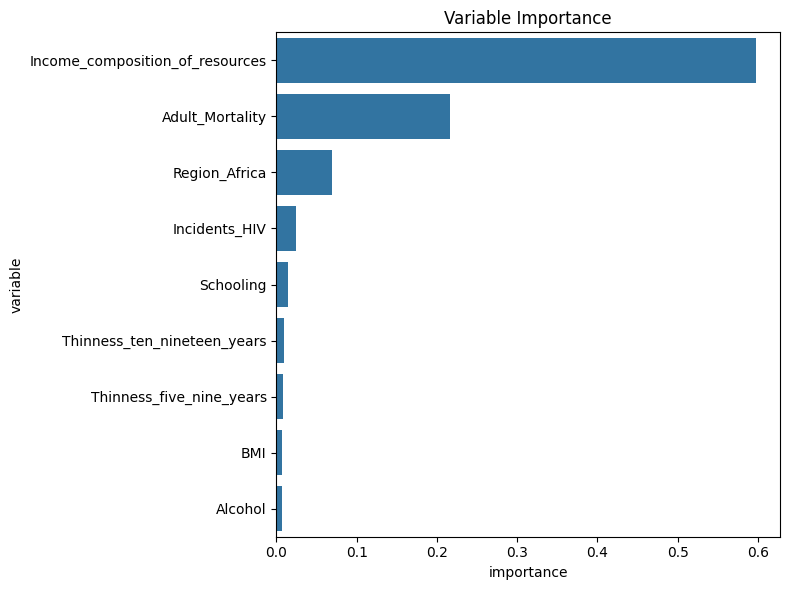
    'subsample': 0.8,

    'colsample\_bytree': 0.8,

}

1. Аналіз важливості ознак

Для визначення, які характери найбільше впливають на роботу моделі, я застосувала метод «Feature Selection». У результаті вияснилося, що ознауи такі як індекс людського розвитку, людська смертність та проживання в регіоні Африка найбільше впливають на модель.



*Рис. 37 Виведення 10-ти характеристик, які найбільше впливають на модель.*

1. Створення скриптів для розгортання та використання моделі

# Скрипт для тренування моделі:

def preprocess\_train\_data(ds: pd.DataFrame) -> pd.DataFrame:

    def impute\_na(df, variable, value):

        return df[variable].fillna(value)

    # remove all rows with more than 5 missing values

    missing\_rows = ds[ds.isnull().sum(axis=1) >= 5]

    ds = ds.drop(missing\_rows.index)

    # And now those 10 lines, of which only data for 2013 in 10 countries are known

    ds = ds[~ds['Country'].isin(columns.countries\_to\_remove)]

    percentile\_impute\_columns = dict()

    for column in columns.percentile\_impute\_columns:

        percentile\_impute\_columns[column] = ds[column].quantile(0.01)

        ds[column] = impute\_na(ds, column, percentile\_impute\_columns[column])

    median\_impute\_columns = dict()

    for column in columns.median\_impute\_columns:

        median\_impute\_columns[column] = ds[column].median()

        ds[column] = impute\_na(ds, column, median\_impute\_columns[column])

    def find\_skewed\_boundaries(df, variable, distance):

        IQR = df[variable].quantile(0.75) - df[variable].quantile(0.25)

        lower\_boundary = df[variable].quantile(0.25) - (IQR \* distance)

        upper\_boundary = df[variable].quantile(0.75) + (IQR \* distance)

        return upper\_boundary, lower\_boundary

    outliers\_left\_columns = dict()

    for column in columns.outliers\_left\_columns:

        upper\_boundary, lower\_boundary = find\_skewed\_boundaries(ds, column, 2)

        median\_value = ds[column].median()

        ds.loc[ds[column] < lower\_boundary, column] = median\_value

        outliers\_left\_columns[column] = median\_value

    outliers\_right\_columns = dict()

    for column in columns.outliers\_right\_columns:

        upper\_boundary, lower\_boundary = find\_skewed\_boundaries(ds, column, 2)

        median\_value = ds[column].median()

        ds.loc[ds[column] > upper\_boundary, column] = median\_value

        outliers\_right\_columns[column] = median\_value

    def creat\_Region\_col(df, mapping=columns.country\_to\_region):

        df["Region"] =df['Country'].map(columns.country\_to\_region)

    creat\_Region\_col(ds)

    ds = ds.drop(['Country', 'Infant\_deaths'], axis=1)

    columns\_encoding = dict()

    for column in columns.columns\_encoding:

        tmp = pd.get\_dummies(ds[column], prefix=column)

        tmp = tmp.astype(int)

        columns\_encoding = tmp

        ds = ds.drop(column, axis=1)

        ds = pd.concat([ds, columns\_encoding], axis=1)

    ds = ds.drop('Status\_Developing', axis=1)

    scaler = MinMaxScaler()

    ds[columns.colmns\_to\_scale] = scaler.fit\_transform(ds[columns.colmns\_to\_scale])

    param\_dict ={'percentile\_impute\_columns':percentile\_impute\_columns,

                  'median\_impute\_columns':median\_impute\_columns,

                  'outliers\_left\_columns':outliers\_left\_columns,

                  'outliers\_right\_columns':outliers\_right\_columns,

                  'columns\_encoding':columns\_encoding

    }

    with open('D:/3Kurs/1Sem/SS/rgr/pipeline/param\_dict.pickle', 'wb') as handle:

        pickle.dump(param\_dict, handle, protocol=pickle.HIGHEST\_PROTOCOL)

    ds.to\_csv('D:/3Kurs/1Sem/SS/rgr/data/train\_look.csv', index=False)

    return ds

import pandas as pd

import pickle

from prepricessing import preprocess\_train\_data

import columns

import xgboost as xgb

import hiperparameters

def train\_model(file\_name: str = 'train.csv', model\_name: str = 'XGBRegressor'):

    # loading data

    ds = pd.read\_csv('D:/3Kurs/1Sem/SS/rgr/data/' + file\_name)

    ds = preprocess\_train\_data(ds)

    X = ds[columns.X\_column]

    y = ds[columns.y\_column]

    XGBRegressor = xgb.XGBRegressor(\*\*hiperparameters.param\_grid)

    XGBRegressor.fit(X, y)

    with open(f'D:/3Kurs/1Sem/SS/rgr/models/XGBRegressor.pickle', 'wb') as handle:

        pickle.dump(XGBRegressor, handle, protocol=pickle.HIGHEST\_PROTOCOL)

# Скрипт для використання моделі на нових даних:

def preprocess\_testing\_data(ds: pd.DataFrame) -> pd.DataFrame:

    with open('D:/3Kurs/1Sem/SS/rgr/pipeline/param\_dict.pickle', 'rb') as handle:

        param\_dict = pickle.load(handle)

    def impute\_na(df, variable, value):

        return df[variable].fillna(value)

    # remove all rows with more than 5 missing values

    missing\_rows = ds[ds.isnull().sum(axis=1) >= 5]

    ds = ds.drop(missing\_rows.index)

    # And now those 10 lines, of which only data for 2013 in 10 countries are known

    ds = ds[~ds['Country'].isin(columns.countries\_to\_remove)]

    for column in columns.percentile\_impute\_columns:

        percentile\_impute\_columns = param\_dict['percentile\_impute\_columns'][column]

        ds[column] = impute\_na(ds, column, percentile\_impute\_columns)

    for column in columns.median\_impute\_columns:

        median\_impute\_columns = param\_dict['median\_impute\_columns'][column]

        ds[column] = impute\_na(ds, column, median\_impute\_columns)

    def find\_skewed\_boundaries(df, variable, distance):

        IQR = df[variable].quantile(0.75) - df[variable].quantile(0.25)

        lower\_boundary = df[variable].quantile(0.25) - (IQR \* distance)

        upper\_boundary = df[variable].quantile(0.75) + (IQR \* distance)

        return upper\_boundary, lower\_boundary

    for column in columns.outliers\_left\_columns:

        upper\_boundary, lower\_boundary = find\_skewed\_boundaries(ds, column, 2)

        median\_value = param\_dict['outliers\_left\_columns'][column]

        ds.loc[ds[column] < lower\_boundary, column] = median\_value

    for column in columns.outliers\_right\_columns:

        upper\_boundary, lower\_boundary = find\_skewed\_boundaries(ds, column, 2)

        median\_value = param\_dict['outliers\_right\_columns'][column]

        ds.loc[ds[column] > upper\_boundary, column] = median\_value

    def creat\_Region\_col(df, mapping=columns.country\_to\_region):

        df["Region"] =df['Country'].map(columns.country\_to\_region)

    creat\_Region\_col(ds)

    ds = ds.drop(['Country', 'Infant\_deaths'], axis=1)

    for column in columns.columns\_encoding:

        tmp = pd.get\_dummies(ds[column], prefix=column)

        tmp = tmp.astype(int)

        columns\_encoding = tmp

        ds = ds.drop(column, axis=1)

        ds = pd.concat([ds, columns\_encoding], axis=1)

    ds = ds.drop('Status\_Developing', axis=1)

    scaler = MinMaxScaler()

    ds[columns.colmns\_to\_scale] = scaler.fit\_transform(ds[columns.colmns\_to\_scale])

    return ds

import pandas as pd

import pickle

from prepricessing import preprocess\_testing\_data

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

import columns

def test\_model(file\_name: str = 'test.csv', model\_name: str = 'XGBRegressor'):

       # loading data

    ds = pd.read\_csv('D:/3Kurs/1Sem/SS/rgr/data/' + file\_name)

    ds = preprocess\_testing\_data(ds)

    X = ds[columns.X\_column]

    y = ds[columns.y\_column]

    with open(f'D:/3Kurs/1Sem/SS/rgr/models/{model\_name}.pickle', 'rb') as f:

        model = pickle.load(f)

    predictions = model.predict(X)

    pd.DataFrame(predictions).to\_csv('D:/3Kurs/1Sem/SS/rgr/data/predictions.csv', index=False)

    mae = mean\_absolute\_error(y, predictions)

    mse = mean\_squared\_error(y, predictions)

    rmse = mean\_squared\_error(y, predictions, squared=False)  # або np.sqrt(mse)

    r2 = r2\_score(y, predictions)

    # Виводимо результати

    print("Mean Absolute Error (MAE):", mae)

    print("Mean Squared Error (MSE):", mse)

    print("Root Mean Squared Error (RMSE):", rmse)

    print("R-squared (R2):", r2)

Висновки

У процесі виконання роботи було досягнуто таких результатів на кожному етапі проєкту:

1. **Аналіз даних**: Було проведено огляд даних та дослідження статистичних характеристик ознак, що дозволило визначити ключові чинники, які впливають на тривалість життя. Також було виявлено пропущені значення та аномалії, що в подальшому вимагало їх обробки для забезпечення якості даних.
2. **Попередня обробка даних**: На цьому етапі ми провели обробку пропущених значень (використовуючи медіанну та percentline імпутацію), аналіз та корекцію аномалій, а також масштабування числових ознак. Окрім цього, категоріальні змінні було закодовано, що забезпечило сумісність даних для моделювання. Попередня обробка виявилася складним етапом, оскільки вимагала ретельного аналізу та вибору оптимальних методів заповнення даних.
3. **Моделювання**: Було протестовано кілька регресійних моделей, включаючи XGBRegressor, Linear Regression, Ridge, Lasso, KNeighborsRegressor, RandomForestRegressor та DecisionTreeRegressor. Найкращі результати продемонструвала модель XGBRegressor, яка змогла врахувати як лінійні, так і нелінійні залежності між ознаками. Ця модель показала найвищу точність прогнозування з мінімальними помилками, у порівнянні з іншими моделями.
4. **Підбір гіперпараметрів**: Для XGBRegressor було проведено налаштування гіперпараметрів, що додатково покращило продуктивність моделі на тестових даних. Вибір оптимальних параметрів виявився трудомістким, оскільки потребував часу на тестування та порівняння різних комбінацій.

Виконання проекту дало мені важливий практичний досвід у роботі з реальними даними, а також покращило навички використання регресійних моделей у задачах прогнозування. Я навчився застосовувати методи обробки даних, розуміти та оцінювати вплив різних ознак, а також порівнювати моделі на основі метрик точності. Проект допоміг поглибити розуміння процесу підготовки даних, налаштування моделей та оцінки результатів, що є важливими складовими ефективного використання машинного навчання.

Загальні враження від виконання роботи позитивні, оскільки проєкт дав змогу на практиці зрозуміти, як аналітика та машинне навчання можуть сприяти вирішенню актуальних питань у сфері охорони здоров'я.

Додаток

Посилання на датасет, який був використаний на розрахунково-графічну роботу[1]:

<https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who/data>

Весь скрипт з дослідженнями, та кодами можна переглянути за посиланням:

<https://github.com/Rurararu?tab=repositories>