МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: Використання інфраструктури Oracle Big Data у задачах аналізу інформації

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи КБ-20-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»

_____ Татьяніна Назара Едуардовича
(підпис)

Науковий керівник _____ Редич Олександр Володимирович, к.е.н, доцент
(підпис)

Керівник освітньо-професійної програми _____ Омельчук Антон Анатолійович, к.т.н.
(підпис)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних технологій і систем протокол від «22» травня 2024 № 15

Завідувач кафедри _____ Філоненко Михайло Миколайович, к.ф.-м.н., доцент
(підпис)

Ірпінь – 2024

Реєстрація №28 27.05.2024 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 22.05.24 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 5 (відмінно) 06.06.2024 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ В.М. Франчук
(підпис)

Члени комісії: _____ М.М. Філоненко
(підпис)

_____ С.М. Грищенко
(підпис)

_____ В.О. Ніжегородцев
(підпис)

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	28		06.06.2024 Франчук В.М. Філоненко М.М.
2. Дотримання вимог щодо оформлення	від 0 до 10	9		Грищенко С.М.
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		Ніжегородцев В.О. Суховецька Л.П.
4. Рецензент	від 0 до 5	5		
РАЗОМ БАЛІВ	від 0 до 50	46		

Оцінювання кваліфікаційної роботи науковим керівником - 46 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалоюЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
92	A	5(відмінно)	06.06.2024	Франчук В.М. Філоненко М.М. Грищенко С.М. Ніжегородцев В.О. Суховецька Л.П.

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційні управляючі системи і технології (за галузями)»

Затверджую
Завідувач кафедри _____ Філоненко М.М.
«23» жовтня 2023

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Татьяніну Назару Едуардовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Використання Oracle Big Data у задачах аналізу інформації
Затверджена наказом ректора від «20» жовтня 2023 № 1411
2. Строк здачі закінченої роботи «20» травня 2024
3. Мета роботи проведення структурування Великих даних за допомогою Oracle Big Data та аналізу інформації інструментарієм машинного навчання
Об'єкт дослідження технології та інфраструктура великих даних (Big Data)
Предмет дослідження Oracle Big Data Lite та інструментарій машинного навчання як платформа аналізу інформації
Орієнтовна база дослідження інтернет ресурси із інформацією про населення країн планети Земля
4. Вихідні дані до роботи датасет для аналізу, інформаційне середовище бази практики, образи для інсталяції Oracle Big Data Lite
5. Перелік графічного матеріалу рисунки та графіки що пояснюють суть застосованих методів та засобів. Структурні схеми та таблиці, графіки результатів дослідження.
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ BIG DATA
1.1.Основні поняття Big Data
1.2. Інфраструктура розподілених обчислень Hadoop
1.3. Середовище Oracle Big Data Lite Virtual Machine
РОЗДІЛ 2. АНАЛІЗ ТА ОБРОБКА ДАНИХ МЕТОДАМИ ORACLE BIG DATA
2.1. Обробка датасету у середовищі Oracle Big Data
2.2. Статистичний аналіз отриманих струкутрованих даних
РОЗДІЛ 3. АНАЛІЗ ДАНИХ МЕТОДАМИ ML (Oracle ML, R, Colab тощо)
3.1.Опис методу машинного навчання (класифікація, дерева рішень, правила асоціацій тощо)
3.2. Обробка даних методами машинного навчання
3.3. Пояснення результатів і формування висновків

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційноїроботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.10.2023	15.10.2023
2	**І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	29.12.2023	29.12.2023
3	**ІІ розділ** (аналіз експериментальної частини про методики досліджень)	15.02.2024	15.02.2024
4	**ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	15.04.2024	15.04.2024
5	**Висновки**	01.05.2024	01.05.2024
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	06.05.2024	06.05.2024
7	Отримання відгуку керівника	28.05.2024	28.05.2024
8	Отримання зовнішньої рецензії	30.05.2024	30.05.2024

8. Дата видачі завдання «23» жовтня 2023

9. Керівник кваліфікаційної роботи _____ Редич О.В.
(підпис) (прізвище, ініціали)

10. Керівник освітньо-професійної програми _____ Омельчук А.А.
(підпис) (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти _____ Татьянін Н.Е.
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Використання інфраструктури Oracle Big Data у задачах аналізу інформації

Метою даної дипломної роботи є дослідження можливостей та переваг використання Big Data у сфері аналізу даних, дослідження процесів структурування Великих даних за допомогою Oracle Big Data та аналіз інформації інструментарієм машинного навчанн. Завданням роботи є розгляд основних підходів до роботи з великими обсягами даних, вивчення методів обробки, аналізу та візуалізації інформації в умовах великого обсягу даних.

Об'єктом дослідження є технології та інфраструктура великих даних (Big Data), процеси трансформації даних з використанням Big Data технологій, а також їхні можливості та перспективи в сучасному інформаційному суспільстві.

Предметом дослідження є Oracle Big Data Lite та інструментарій машинного навчання як платформа аналізу інформації, використання Big Data у різних сферах, таких як бізнес, наука, медицина, технології та інші.

Базою дослідження виступають теоретичні та практичні аспекти використання Big Data, аналіз сучасних підходів та розробок у цій галузі, застосування технологій Big Data на прикладі обробки інтернет ресурсів із інформацією про населення країн планети Земля.

За результатами дослідження інформації про населення планети сформульовані рекомендації щодо ефективного використання Big Data в задачах аналізу даних, а також визначені перспективи подальшого розвитку даного напрямку.

Одержані результати будуть використані для підвищення ефективності аналізу даних у різних сферах діяльності, а також для сприяння подальшому розвитку та впровадженню Big Data технологій.

Ключові слова: Big Data, аналіз даних, обробка даних, візуалізація даних, технології, методи, машинне навчання.

Кваліфікаційна робота містить 70 сторінок, 15 рисунків, список використаних джерел 42 найменувань, 6 додатків.

Рік виконання кваліфікаційної роботи: 2024.

Рік захисту роботи: 2024.

SUMARY

The use of Oracle Big Data infrastructure in information analysis tasks

The purpose of this thesis is to study the possibilities and advantages of using Big Data in the field of data analysis, to study the processes of structuring Big Data using Oracle Big Data and to analyze information using machine learning tools. The objective of the work is to consider the main approaches to working with large amounts of data, to study methods of processing, analyzing and visualizing information in the context of large amounts of data.

The object of the study is Big Data technologies and infrastructure, data transformation processes using Big Data technologies, as well as their capabilities and prospects in the modern information society.

The subject of the study is Oracle Big Data Lite and machine learning tools as a platform for information analysis, the use of Big Data in various fields such as business, science, medicine, technology, and others.

The research is based on theoretical and practical aspects of Big Data use, analysis of modern approaches and developments in this field, application of Big Data technologies on the example of processing Internet resources with information about the population of the countries of the planet Earth.

Based on the results of the study of information about the world's population, recommendations for the effective use of Big Data in data analysis tasks were formulated, and prospects for further development of this area were identified.

The obtained results will be used to improve the efficiency of data analysis in various fields of activity, as well as to promote the further development and implementation of Big Data technologies.

Keywords: Big Data, data analysis, data processing, data visualization, technologies, methods, machine learning.

The qualification work contains 70 pages, 15 figures, a list of used sources with 42 titles, and 6 appendixes.

Year of completion of the qualification work: 2024.

Year of defense of the work: 2024.

ЗМІСТ

ВСТУП 3
РОЗДІЛ 1 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ BIG DATA 6
1.1 Основні поняття Big Data 6
1.2. Інфраструктура розподілених обчислень Hadoop 13
1.3. Середовище Oracle Big Data Lite Virtual Machine 17
РОЗДІЛ 2 АНАЛІЗ ТА ОБРОБКА ДАНИХ МЕТОДАМИ ORACLE BIG DATA 23
2.1 Обробка датасету у середовищі Oracle Big Data Lite Virtual Machine 23
2.2 Статистичний аналіз отриманих струкутрованих даних 28
РОЗДІЛ 3 АНАЛІЗ ДАНИХ МЕТОДАМИ ML 36
3.1 Опис методу машинного навчання 36
3.2. Обробка даних методами машинного навчання 42
3.3. Пояснення результатів і формування висновків 50
ВИСНОВКИ 53
ДОДАТКИ 59
Додаток А 59
«Лістинг програмного коду. Зчитування та обробка даних» 59
Додаток Б 61
«Лістинг програмного коду. Побудова гістограми із бочковою діаграмою темпів зростання населення» 61
Додаток В 64
«Лістинг програмного коду. Побудова гістограми із бочковою діаграмою темпів зростання населення 64
Додаток Г 67
«Лістинг програмного коду. Побудова графіку тенденції зміни населення (1970-2024)» 67
Додаток Д 68
«Лістинг програмного коду. Побудова графіку тенденції зміни середнього приросту населення (1970-2024)» 68
Додаток Е 69
«Лістинг програмного коду. Побудова точкового графіку питомої ваги території - питома вага населення країни у 2024 році 69

ВСТУП

У сучасному світі, що характеризується високою динамікою змін та постійним зростанням обсягів даних, ефективне управління та аналіз інформації є критично важливими завданнями для підприємств у всіх сферах діяльності. Відповідно до цього, з'являються нові технології та підходи, спрямовані на оптимізацію обробки та аналізу великих обсягів даних, серед яких особливе місце займає інфраструктура Oracle Big Data.

Використання інфраструктури Oracle Big Data у задачах аналізу інформації відкриває широкі можливості для підприємств щодо ефективного використання своїх даних. Інтеграція різноманітних джерел даних, включаючи структуровані та неструктуровані дані, дозволяє отримувати повнішу картину діяльності підприємства та його середовища.

Розвиток інфраструктури Oracle Big Data сприяє збільшенню продуктивності та швидкості аналітичних операцій, що є ключовими факторами в умовах постійно зростаючого обсягу даних. Паралельна обробка даних, оптимізація запитів та використання розподіленої архітектури дозволяють зменшити час виконання складних аналітичних операцій і сприяють прийняттю швидких та обґрунтованих управлінських рішень.

Актуальність дослідження полягає в необхідності підприємств адаптуватися до умов сучасного ринкового середовища, де швидкість прийняття управлінських рішень та їх обґрунтованість визначають конкурентоспроможність. Враховуючи великий обсяг даних, який генеруються та накопичуються, інфраструктура Oracle Big Data надає потужний інструментарій для аналізу цих даних та виходу на новий рівень управління підприємством.

Основні проблеми великих даних включають обробку величезних обсягів інформації, різноманітність даних, конфіденційність та безпеку, а також якість даних. Великі обсяги інформації потребують потужної обчислювальної інфраструктури та ефективних алгоритмів обробки даних.

Багато провідних університетів і наукових інститутів по всьому світу активно досліджують великі дані. Серед них Массачусетський технологічний інститут (MIT), Стенфордський університет, Гарвардський університет та Оксфордський університет. Існує безліч наукових журналів, які публікують дослідження з великих даних, такі як "Journal of Big Data", "Big Data Research", "IEEE Transactions on Big Data", "Data Mining and Knowledge Discovery". Крім того, щорічно проводяться конференції, такі як "IEEE International Conference on Big Data" та "ACM SIGKDD Conference on Knowledge Discovery and Data Mining". Технологічні гіганти, такі як Google, Amazon, Microsoft, Facebook та IBM, застосовують великі дані для покращення своїх продуктів та послуг, а також для розробки нових технологій. Наприклад, Google використовує великі дані для покращення своїх алгоритмів пошуку та реклами, а Amazon - для рекомендаційних систем. У фінансовій сфері великі дані використовуються для аналізу ризиків, виявлення шахрайства, прогнозування ринкових трендів та покращення обслуговування клієнтів. Банки та фінансові установи, такі як JPMorgan Chase та Goldman Sachs, активно інвестують у технології великих даних.

Метою даної дипломної роботи є дослідження можливостей та переваг використання Big Data у сфері аналізу даних, дослідження процесів структурування Великих даних за допомогою Oracle Big Data та аналіз інформації інструментарієм машинного навчання.

Об'єктом дослідження є технології та інфраструктура великих даних (Big Data), процеси трансформації даних з використанням Big Data технологій, а також їхні можливості та перспективи в сучасному інформаційному суспільстві.

Предметом дослідження є Oracle Big Data Lite та інструментарій машинного навчання як платформа аналізу інформації, використання Big Data у різних сферах, таких як бізнес, наука, медицина, технології та інші.

Базою дослідження виступають теоретичні та практичні аспекти використання Big Data, аналіз сучасних підходів та розробок у цій галузі,

застосування технологій Big Data на прикладі обробки інтернет ресурсів із інформацією про населення країн планети Земля.

Завдання дослідження включають аналіз функціональних можливостей інфраструктури Oracle Big Data, визначення оптимальних стратегій і методик використання цієї інфраструктури для аналізу населення країн світу, тенденції їх змінювання, оцінку ефективності застосування даного інструментарію у практичних сценаріях.

Особистий внесок автора в даній дипломній роботі полягає у проведенні дослідження практичного застосування інфраструктури Oracle Big Data у задачах аналізу інформації. Автор детально розглянув технології та інструменти, що входять до складу цієї інфраструктури, та провів аналіз їхньої ефективності та можливостей застосування у різних сферах діяльності підприємств. Додатково до цього, автор використав методи машинного навчання та бібліотеки Python для дослідження населення світу.

Під час дослідження було побудовано гістограми та бочкові діаграми, точкову діаграми, графіки тенденції зміни населення, і проведено аналіз результатів. Результати аналізу даних свідчать про стійке зростання населення світу протягом розглянутого періоду. В результаті цього дослідження були сформульовані висновки та рекомендації щодо оптимального використання інфраструктури Oracle Big Data з метою підвищення конкурентоспроможності та ефективності підприємств у сучасних умовах.

Апробація розробки відбувалась на серверах ДПУ, де була розгорнута інфраструктура Oracle Big Data. Крім того, результати досліджень були опубліковані на IV Міжнародній науково-практичній інтернет-конференції "ЦИФРОВА ТРАНСФОРМАЦІЯ ФІНАНСОВОЇ СИСТЕМИ УКРАЇНИ ТА КРАЇН V-4 В УМОВАХ ЄВРОІНТЕГРАЦІЇ", теза "ВИКОРИСТАННЯ ВЕЛИКИХ ДАНИХ У ФІНАНСАХ", (2023 р.) і на І Всеукраїнській конференції молодих вчених "Математичні методи та цифрові технології в економіці", теза "ВИКОРИСТАННЯ ВЕЛИКИХ ДАНИХ ДЛЯ РОЗВИТКУ ФІНАНСОВОЇ СФЕРИ", (2024 р.)

РОЗДІЛ 1 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ BIG DATA

1.1 Основні поняття Big Data

Великі дані, або BD, представляють собою великі масиви інформації, як структурованої, так і неструктурованої, розміри яких настільки великі, що традиційні методи їх обробки і аналізу, зазвичай засновані на бізнес-аналітиці та системах управління базами даних, виявляються неефективними.

Такі дані часто характеризуються трьома основними характеристиками, що відомі як "3V": великим обсягом (Volume), різноманіттям типів даних (Variety) та швидкістю їхнього збору та обробки (Velocity) (рис.1.1).



Рис. 1.1. Поняття Big Data

Великі дані застосовуються в різних сферах, включаючи медицину, енергетику, фінанси та інші, з метою оптимізації операцій, покращення обслуговування клієнтів, розробки персоналізованих маркетингових стратегій та підвищення ефективності бізнесу в цілому[10].

Об'єм даних у контексті Big Data є однією з його ключових характеристик. Величезні обсяги даних створюються та накопичуються з надзвичайною

швидкістю з різних джерел, таких як сенсори інтернету речей, мобільні додатки, соціальні мережі, онлайн-транзакції та багато інших. Важливою особливістю цих даних є їх різноманітність та надходження у реальному часі. Обсяги даних можуть бути настільки великими, що стандартні інструменти та технології для їх зберігання, обробки та аналізу можуть бути неефективними[2]. Такі обсяги даних часто вимагають використання розподілених систем зберігання та обробки даних, паралельних обчислень та інших спеціалізованих інструментів. Об'єм даних в BD постійно зростає, що вимагає постійного вдосконалення технологій та методів обробки та аналізу даних для ефективного використання цієї інформації у різних сферах діяльності, включаючи бізнес, науку, медицину та інші галузі.

Швидкість у контексті BD визначається як здатність системи обробляти та аналізувати великі обсяги даних у реальному часі або наближено до нього[2]. Важливість швидкості в обробці даних полягає в тому, що в деяких випадках значення може втратити актуальність протягом короткого періоду часу. Наприклад, у фінансовому секторі швидкість обробки даних може бути критичною для прийняття управлінських рішень в реальному часі на основі останніх ринкових трендів та ситуації.

Швидкість обробки даних у BD залежить від різних факторів, таких як архітектура системи, потужність обчислювальних ресурсів, оптимізація алгоритмів обробки та ефективність використання паралельних обчислень. Враховуючи великий обсяг даних, який часто супроводжується великою кількістю операцій з обробки, важливо, щоб система була здатна оперативно та ефективно виконувати запити та аналізувати дані.

Однією з технічних стратегій для забезпечення швидкості обробки даних у BD є використання розподіленої архітектури, де обчислювальні завдання розподіляються між різними вузлами в мережі, що дозволяє паралельно обробляти дані та підвищує швидкість роботи системи. Крім того, оптимізація алгоритмів та використання спеціалізованих технологій, таких як кешування та індексація, також можуть сприяти підвищенню швидкості обробки даних у BD.

Різноманітність у контексті BD визначається як широкий спектр типів та форматів даних, що включаються в аналіз та обробку. Це можуть бути як структуровані дані, що мають чітку організацію та формат, такі як таблиці баз даних з числовими значеннями або категоріями, так і неструктуровані дані, які не мають чіткої організації, наприклад, тексти, зображення, аудіо та відео. У Big Data також можуть входити напівструктуровані дані, які містять елементи структурованих та неструктурованих даних, такі як документи формату JSON або XML[27].

Різноманітність даних у BD також охоплює різні джерела та домені дані, з яких вони походять. Це можуть бути дані, зібрані з сенсорів Інтернету речей, дані від користувачів соціальних мереж, дані транзакцій бізнесу, дані супутникових зображень та інше[2]. Крім того, дані можуть бути зібрані з різних галузей, таких як медицина, фінанси, наука, виробництво та інші, що вносять свою власну специфіку та характеристики у дані.

У зв'язку з різноманітністю даних у BD виникає необхідність у розробці та використанні спеціалізованих методів та інструментів для їх аналізу, обробки та інтерпретації. Це може включати розробку алгоритмів машинного навчання та штучного інтелекту, які можуть ефективно працювати з різноманітними типами даних, а також розробку систем зберігання та обробки даних, що можуть масштабуватися та працювати з різноманітними форматами даних.

Правдивість даних у контексті BD визначається їхньою достовірністю та відповідністю реальному стану. Це включає якість даних, їх актуальність та правильність. Оскільки BD охоплює різноманітні джерела та типи даних, забезпечення правдивості може бути складною задачею.

Важливо проводити перевірку даних на наявність помилок, дублікатів та інших невірностей, а також визначати їхню джерело та методи збору. Для цього можуть використовуватися різноманітні техніки, включаючи автоматизовані алгоритми перевірки, аналіз метаданих та валідацію даних під час їх збору та обробки.

Забезпечення правдивості даних в BD є важливим для забезпечення достовірності аналітичних висновків та прийняття управлінських рішень на основі цих даних[5]. Недостовірні або неточні дані можуть призвести до неправильних висновків та втрати можливостей для підприємства.

Зокрема, у випадку великих обсягів даних, які генеруються автоматично або збираються з різних джерел, важливо забезпечити механізми автоматичної перевірки та корекції помилок. Також необхідно мати систему моніторингу якості даних, щоб оперативно реагувати на виявлені проблеми та підтримувати високу правдивість даних в усіх аспектах їхнього життєвого циклу.

У контексті BD, пошук зв'язків означає виявлення взаємозв'язків, патернів та кореляцій між різними наборами даних. Цей процес вимагає використання різноманітних алгоритмів аналізу даних, статистичних методів та інструментів машинного навчання для виявлення складних залежностей, які можуть бути прихованими в масиві інформації. Пошук зв'язків може включати в себе виявлення спільних характеристик між різними наборами даних, встановлення причинно-наслідкових зв'язків, а також визначення часових або просторових зв'язків між подіями або явищами. Цей процес може допомогти виявити нові знання, зробити прогнози та прийняти обґрунтовані рішення на основі виявлених зв'язків у великих обсягах даних. Пошук зв'язків може бути використаний у різних сферах, включаючи бізнес, науку, медицину, фінанси та інші галузі, де аналіз великих обсягів даних може призвести до виявлення цінних знань та можливостей[4]. Враховуючи різноманітність даних в BD, пошук зв'язків є важливим етапом аналізу, який може допомогти витягнути корисну інформацію з великого масиву неструктурованих даних і використати її для прийняття управлінських рішень та розробки нових стратегій.

Аналітика в контексті BD відіграє ключову роль у виявленні шаблонів, трендів та зв'язків у великих обсягах даних. Завдяки потужним алгоритмам та інструментам аналізу, які застосовуються до BD, можливо виявляти приховані закономірності та нові інсайти, що не завжди можуть бути виявлені за допомогою традиційних методів аналізу.

Одним з ключових аспектів аналітики в BD є здатність працювати з різноманітними типами даних, включаючи структуровані, наприклад, числові дані, та неструктуровані, такі як тексти, зображення або відео. Аналітичні моделі та алгоритми можуть бути адаптовані для роботи з будь-якими форматами даних, що відкриває широкі можливості для виявлення нових знань та розуміння ситуації на бізнес-рівні.

Ще однією важливою функцією аналітики в BD є здатність до роботи з великими обсягами даних в реальному часі. Швидкість обробки та аналізу даних є критичною, оскільки великі обсяги даних поступають з різних джерел у реальному часі, таких як сенсори Інтернету речей або потокові дані з соціальних мереж[12].

Нарешті, аналітика в BD може бути застосована у різних галузях, від фінансів та маркетингу до науки та медицини. Величезний обсяг даних, які доступні для аналізу, дозволяє розробляти інноваційні рішення та прогнозувати поведінку або події з високою точністю.

Зберігання є ключовим аспектом управління BD і вимагає відповідних стратегій та технологій. Одним з основних викликів є необхідність забезпечити ефективне та безпечне зберігання великих обсягів даних[**7**]. Велика кількість даних вимагає потужних систем зберігання з високою пропускною здатністю та масштабованістю. Розподілені системи зберігання, такі як Hadoop Distributed File System (HDFS) або Apache Cassandra, часто використовуються для зберігання Big Data, оскільки вони можуть легко масштабуватися та працювати з великими обсягами даних.

Крім того, обробка та зберігання BD можуть відбуватися в різних середовищах, включаючи локальні сервери, хмарні платформи або гібридні комбінації обох. Важливо забезпечити безпеку даних під час їх зберігання, включаючи шифрування, контроль доступу та моніторинг вразливостей. Для ефективного управління зберіганням даних важливо також використовувати стратегії реплікації для запобігання втрати даних через випадкові збої апаратного забезпечення або інші непередбачувані ситуації.

Крім того, зберігання BD також може включати архівування даних для подальшого використання або аналізу. Це може бути важливою стратегією для довгострокового зберігання великих обсягів даних, які більше не активно використовуються, але можуть бути корисними для майбутніх аналітичних потреб. Такі архівні системи повинні бути ефективними та масштабованими, а також забезпечувати можливість швидкого доступу до даних у разі потреби[14].

Обробка великих обсягів даних є важливим етапом в роботі з BD. Цей процес включає в себе ряд технік та методів для обробки, очищення, агрегації та аналізу великих обсягів даних з різних джерел. Одним із ключових викликів обробки BD є необхідність забезпечити швидку та ефективну обробку даних у реальному часі. Для цього використовуються різноманітні технології та інструменти, такі як розподілені системи обробки даних, паралельні обчислення та потужні алгоритми.

Однією з основних стратегій обробки BD є розподілений підхід до обробки даних, коли обробка великих обсягів даних розподіляється між різними вузлами великих кластерів. Це дозволяє розподілено обробляти великі обсяги даних, що дозволяє досягти високої швидкодії та масштабованості обробки.

Крім того, для ефективної обробки BD використовуються алгоритми паралельних обчислень, які дозволяють розділити обробку даних на окремі частини, які обробляються паралельно на різних обчислювальних ресурсах. Це дозволяє ефективно використовувати потужності розподілених обчислювальних систем для швидкої обробки великих обсягів даних.

Крім того, важливим аспектом обробки BD є можливість проведення аналізу даних у реальному часі, що вимагає використання спеціалізованих технологій та алгоритмів[5]. Для цього використовуються потужні системи аналітики даних, які забезпечують швидку та ефективну обробку потокових даних та виведення результата аналізу в реальному часі.

Таким чином, обробка великих обсягів даних в BD вимагає використання

різноманітних технологій та методів для забезпечення швидкої, ефективної та масштабованої обробки даних з різних джерел.

Забезпечення безпеки в контексті BD є складною та критичною задачею, оскільки великі обсяги даних створюють нові виклики для захисту інформації. Важливо забезпечити конфіденційність, цілісність та доступність даних усім авторизованим користувачам, у той час як уникнення несанкціонованого доступу, крадіжки даних та атак є надзвичайно важливими завданнями.

Одним із викликів в області безпеки BD є нестабільність джерела даних, оскільки вони можуть бути незворотніми та ненадійними. Це може призвести до втрати даних або пошкодження інформації, що становить серйозну загрозу для безпеки даних.

Для забезпечення безпеки даних в BD використовуються різноманітні технології та методи, такі як шифрування даних, автентифікація та авторизація користувачів, моніторинг активності користувачів та виявлення вторгнень, резервне копіювання та відновлення даних.

Паралельно з розвитком технологій безпеки, виникають нові загрози та вразливості, які вимагають постійного вдосконалення заходів безпеки. Такі виклики, як соціальна інженерія, атаки з використанням штучного інтелекту та розподілені атаки, стають все більш складними та важкодоступними для виявлення та запобігання.

У зв'язку з цим, важливою є роль постійного моніторингу та аналізу безпекових подій для виявлення потенційних загроз та реагування на них у найкоротший термін[20].Тільки завдяки комплексному підходу до забезпечення безпеки можна забезпечити ефективний захист великих обсягів даних в контексті Big Data.

Скалабельність в контексті BD є ключовою властивістю, що визначає можливість системи адаптуватися до зростаючих обсягів даних та вимог щодо їх обробки, зберігання та аналізу. Великі обсяги даних, які зазвичай характеризуються в BD, вимагають потужних обчислювальних ресурсів для їх ефективної обробки. Скалабельність означає, що система може збільшувати

свою продуктивність шляхом додавання нових обчислювальних ресурсів або за допомогою паралельних обчислень.

У межах скалабельної системи, розширення обсягу даних не призводить до відчутних затримок або перевантаження системи. Для досягнення скалабельності використовуються різноманітні підходи, включаючи горизонтальне та вертикальне масштабування систем, розподілені обчислення, а також використання хмарних технологій.

Скалабельність є необхідною характеристикою для успішного впровадження та ефективного використання систем BD у практичних сценаріях. Вона дозволяє підприємствам забезпечити стабільну та ефективну роботу з великими обсягами даних навіть при їх зростанні у майбутньому. Таким чином, скалабельність виступає як важливий фактор успіху у впровадженні Big Data технологій та платформ.

1.2. Інфраструктура розподілених обчислень Hadoop

Apache Hadoop є безкоштовною програмною платформою та фреймворком для організації розподіленого зберігання та обробки обсягів великих даних за допомогою моделі програмування MapReduce (рис.1.2).

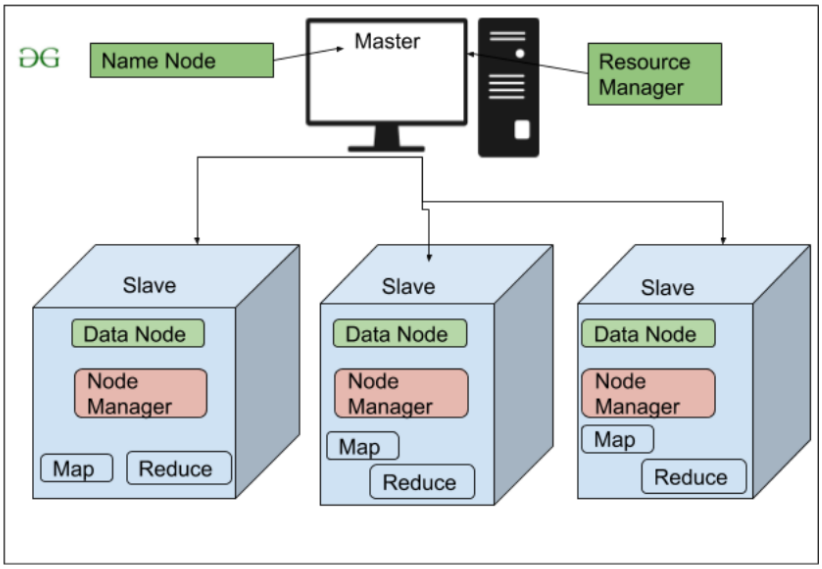


Рис. 1.2. Архітектура Apache Hadoop

Основні компоненти Apache Hadoop включають:

- Hadoop Common, який включає набір бібліотек та утиліт, необхідних для інших модулів Hadoop.

- Hadoop Distributed File System (HDFS), розподілена файлова система, що зберігає дані на звичайних серверах та забезпечує високу загальну пропускну здатність у кластері в цілому.

- Hadoop YARN, платформа, що відповідає за управління обчислювальними ресурсами у кластерах та їх використання для користувацьких завдань.

- Hadoop MapReduce, реалізація моделі програмування MapReduce для опрацювання великих обсягів даних.

Hadoop розділяє файли на великі блоки та розподіляє їх між вузлами кластера, після чого він надсилає код для обробки даних на ці вузли паралельно. Цей підхід використовує принцип локальності даних, що означає, що вузли опрацьовують лише дані, які знаходяться у їхній безпосередній доступності. Це дозволяє ефективно та швидко обробляти дані, що є перевагою порівняно з традиційними архітектурами суперкомп'ютерів.

Hadoop Common, як компонент Apache Hadoop Framework, включає загальні інструменти та бібліотеки, які підтримують інші модулі Hadoop. Цей компонент є необхідною складовою фреймворку Apache Hadoop, спільно з Hadoop Distributed File System (HDFS), Hadoop YARN та Hadoop MapReduce.

Hadoop Common дозволяє розробникам спілкуватися з інфраструктурою Hadoop, надаючи доступ до відповідних бібліотек та інструментів. Він також включає необхідні Java Archive (JAR) файли та скрипти, необхідні для запуску Hadoop.

Цей компонент є критично важливим для функціонування Hadoop, оскільки він забезпечує необхідну функціональність для виконання розподілених обчислень на великих обсягах даних в середовищі Hadoop[6].

Hadoop Distributed File System (HDFS) є розподіленою файловою системою, яка опрацьовує великі набори даних на стандартному обладнанні. Вона використовується для розширення масштабів кластера Apache Hadoop до

сотень або навіть тисяч вузлів. HDFS є одним із ключових компонентів Apache Hadoop, поряд з MapReduce та YARN.

Основні мети HDFS включають управління обробкою великих обсягів даних та виявлення помилок. Вона забезпечує швидке виявлення помилок та ефективне сканування, використовуючи стандартне обладнання. Це дозволяє підтримувати ефективність роботи з обладнанням при роботі з великими обсягами даних, зменшуючи мережевий трафік та збільшуючи швидкість обробки.

Важливо розрізняти HDFS від Apache HBase, що є системою управління базами даних, орієнтованою на стовпці, яка працює поверх HDFS. Apache HBase може краще відповідати потребам у реальному часі завдяки своєму двигуну обробки в пам'яті.

Hadoop YARN (Ще Одний Переговорщик Ресурсів) був впроваджений у версії Hadoop 2.0 для вирішення проблем, що виникали з традиційним MapReduce. YARN ефективно розділяє управління ресурсами MapReduce від обробки даних.

Основна концепція YARN полягає в розділенні функціональності управління ресурсами та планування/моніторингу робіт на окремі демони. Головна ідея полягає у наявності глобального ResourceManager (RM) та використанні ApplicationMaster (AM) для кожного застосування.

YARN не лише усуває проблему з одним джоб-трекером, але також дозволяє Hadoop підтримувати більше моделей та парадигм програмування. Також, YARN надає можливість різним двигунам обробки даних, таким як обробка графів, інтерактивна обробка, потокова обробка та пакетна обробка, працювати з даними, збереженими в HDFS (Hadoop Distributed File System), що збільшує ефективність системи.

Основними компонентами архітектури YARN є:

- Клієнт, який подає завдання map-reduce.

- Resource Manager, що відповідає за призначення та управління ресурсами серед усіх застосувань.

- Node Manager, який доглядає за окремими вузлами в кластері Hadoop та керує застосуваннями та робочим процесом на цих вузлах.

YARN отримав популярність завдяки таким особливостям:

- Масштабованість, яка дозволяє розширювати та керувати тисячами вузлів і кластерів Hadoop.

- Сумісність, яка підтримує існуючі застосування map-reduce без перебоїв, зробивши його сумісним з Hadoop 1.0.

- Використання кластера, оскільки YARN підтримує динамічне використання кластера в Hadoop, це дозволяє оптимізувати використання кластера.

- Багатокористувацькість, яка дозволяє доступ до кількох двигунів, надаючи організаціям перевагу багатокористувацькості.

MapReduce у Hadoop є програмною базою та моделлю програмування, яка застосовується для опрацювання обширних обсягів даних. Робота програми включає два основних етапи: Map і Reduce.

На етапі Map дані розділяються та відображаються, а етап Reduce передбачає зменшення та об'єднання даних. Hadoop може виконувати програми MapReduce, що написані на різних мовах програмування, таких як Java, Ruby, Python і C++.

Програми MapReduce у хмарних обчисленнях є паралельними за своєю природою, що робить їх ефективними для обробки великих обсягів даних за допомогою кількох машин у кластері.

Процес виконання програми MapReduce складається з чотирьох основних фаз: розділення, відображення, перетасування та зменшення. Кожному розбиттю даних відповідає завдання Map, яке виконує функцію відображення для кожного запису.

Наприклад, MapReduce може використовуватись для підрахунку кількості кожного слова в наборі даних. У вихідних даних слова підраховуються та групуються за їх входженнями.

1.3. Середовище Oracle Big Data Lite Virtual Machine

Віртуальна машина в контексті Oracle Big Data Lite Virtual Machine є віртуальним середовищем, яке надає можливість запускати та тестувати різноманітні технології та інструменти BD на одному комп'ютері без необхідності встановлення та конфігурування окремих компонентів. Ця віртуальна машина побудована на базі відкритих технологій та містить у собі широкий спектр компонентів, таких як Hadoop, Apache Spark, Hive, HBase та інші, які інтегровані та готові до використання для обробки та аналізу великих обсягів даних.

Віртуальна машина дозволяє користувачам ефективно вивчати та експериментувати з різноманітними аспектами BD, надаючи готове середовище для розробки та тестування програм, аналізу даних та візуалізації результатів. Завдяки віртуальному середовищу, користувачам не потрібно витрачати час на складне конфігурування та налаштування окремих компонентів, що дозволяє швидше перейти до практичних завдань та досліджень у галузі BD[32].

Одним із основних переваг використання віртуальної машини є можливість створення ізольованого середовища для тестування та розробки, що дозволяє уникнути конфліктів та проблем, пов'язаних зі встановленням різних версій та компонентів на основній операційній системі. Також, використання віртуальної машини спрощує процес підтримки та резервного копіювання, оскільки весь необхідний функціонал та дані маються в одному комплекті, що легко переноситься та відновлюється.

Hadoop є відкритою програмною платформою для обробки та зберігання великих обсягів даних у розподіленому середовищі. В основі Hadoop лежить ряд компонентів, які спільно створюють ефективну інфраструктуру для роботи з BD. Один з цих ключових компонентів - Hadoop Distributed File System (HDFS) - відповідає за розподіл та зберігання даних на великій кількості вузлів в кластері.

Крім HDFS, важливим компонентом Hadoop є MapReduce, який

забезпечує розподілену обробку даних на вузлах кластера. MapReduce дозволяє ефективно обробляти великі обсяги даних шляхом розділення задач на менші підзадачі, які обробляються паралельно на різних вузлах кластера, а потім результати об'єднуються.

Ще одним важливим аспектом Hadoop є його масштабованість. Система може легко масштабуватися з додаванням нових вузлів у кластер, що дозволяє обробляти ще більші обсяги даних без значного збільшення часу обробки. Це робить Hadoop ефективним інструментом для обробки даних у великих проектах та організаціях, де потрібно обробляти та аналізувати великі обсяги даних.

Важливо також відзначити, що Hadoop – це відкрита технологія, що означає, що вона доступна для всіх, хто бажає використовувати її для своїх потреб. Це сприяє швидкому розвитку та інноваціям у сфері обробки великих обсягів даних, оскільки спільнота розробників постійно вносить удосконалення та нововведення у систему.

Apache Spark є потужним фреймворком для обробки та аналізу великих обсягів даних у реальному часі. Він відрізняється високою швидкістю обробки та можливістю роботи з різноманітними типами даних, включаючи структуровані та неструктуровані дані. Spark надає розширені можливості для паралельної обробки даних на великій кількості вузлів, що дозволяє ефективно виконувати складні аналітичні завдання у реальному часі.

Однією з основних переваг Spark є його універсальність та гнучкість. Він підтримує різноманітні джерела даних, включаючи Hadoop Distributed File System (HDFS), Apache Cassandra, Apache HBase та інші. Крім того, Spark має різноманітні API, що дозволяють працювати з даними у різних мовах програмування, таких як Scala, Java, Python та R, що робить його дуже привабливим для різноманітних аналітичних та розробницьких завдань.

Ще однією важливою особливістю Apache Spark є його набір інструментів для розподіленого машинного навчання та глибокого аналізу даних[41]. Spark MLlib - це бібліотека машинного навчання, яка містить

різноманітні алгоритми для класифікації, регресії, кластеризації та інших завдань. Крім того, Spark дозволяє інтегрувати зовнішні бібліотеки для глибокого навчання, такі як TensorFlow або Keras, що розширює його можливості у сфері аналізу даних.

Узагальнюючи, Apache Spark відкриває широкі можливості для обробки та аналізу великих обсягів даних у реальному часі. Його висока швидкість, універсальність та багатофункціональність роблять його одним з найпопулярніших інструментів у сфері BD та аналітики.

У контексті систем обробки великих обсягів даних, Hive та HBase є ключовими компонентами для роботи з різними типами даних у розподіленому середовищі. Hive, розроблений Apache Software Foundation, надає можливість виконувати структуровані запити до даних у стилі, схожому на SQL, що дозволяє користувачам аналізувати великі обсяги даних, збережених у Hadoop Distributed File System (HDFS), з використанням звичайних SQL-підобних виразів. Hive дозволяє створювати таблиці, які відображаються на дані, збережені у HDFS, і використовувати їх для виконання запитів, що спрощує аналіз даних у розподіленому середовищі. З іншого боку, HBase - це розподілена система керування базами даних (NoSQL), яка забезпечує можливість зберігання та обробки великих обсягів неструктурованих даних у реальному часі. HBase використовує модель даних ключ-значення та забезпечує горизонтальне масштабування, що дозволяє обробляти великі обсяги даних у реальному часі. Він забезпечує високу доступність та надійність шляхом реплікації даних на різних вузлах у кластері. Обидва ці інструменти - Hive та HBase - інтегруються з екосистемою Hadoop та доповнюють її можливості у роботі з різними типами даних у великих обсягах.

Oracle NoSQL Database – це розподілена система управління базами даних, розроблена для ефективного зберігання та обробки великих обсягів неструктурованих даних у реальному часі. Вона базується на ключ-значення моделі даних, де кожен запис має унікальний ключ та відповідне значення. Ця база даних призначена для високошвидкісного доступу до даних та обробки

запитів, що робить її ідеальним рішенням для сучасних додатків, які вимагають швидкої реакції на запити користувачів або подій.

Основною особливістю Oracle NoSQL Database є горизонтальне масштабування, що дозволяє легко розширювати систему шляхом додавання нових вузлів до кластера. Це дозволяє системі зберігати та обробляти великі обсяги даних, а також забезпечує високу доступність та надійність.

Іншою важливою особливістю Oracle NoSQL Database є підтримка ACID (Atomicity, Consistency, Isolation, Durability) властивостей, що гарантує консистентність та надійність транзакцій. Це робить її придатною для використання в сферах, де важливо забезпечити цілісність даних та виконання бізнес-логіки.

Багатофункціональність Oracle NoSQL Database також включає в себе можливість використання різних типів консистентності даних, реплікацію даних для забезпечення високої доступності та паралельну обробку запитів для підвищення продуктивності системи.

У сукупності ці особливості роблять Oracle NoSQL Database потужним та ефективним рішенням для реалізації сучасних додатків, які потребують обробки великих обсягів неструктурованих даних у реальному часі з високою доступністю та надійністю.

Cloudera Manager – це інструмент управління та моніторингу, призначений для керування різними аспектами Hadoop-систем. Він забезпечує адміністраторам можливість керувати конфігураціями, моніторити стан та продуктивність різних компонентів Hadoop, а також автоматизувати рутинні адміністративні завдання. Cloudera Manager надає графічний інтерфейс, що спрощує процес управління Hadoop-системами, дозволяючи адміністраторам швидко та ефективно реагувати на зміни в середовищі. Він також надає інструменти для моніторингу відмовостійкості та безпеки системи, що допомагає забезпечити надійну та безпечну роботу Hadoop-систем. Cloudera Manager інтегрується з різними компонентами Hadoop, такими як HDFS, YARN, MapReduce, Hive, HBase та інші, що дозволяє адміністраторам

здійснювати повний контроль над всією інфраструктурою. Враховуючи складність та масштабність Hadoop-систем, Cloudera Manager стає незамінним інструментом для ефективного управління та моніторингу цих систем, допомагаючи забезпечити їх стабільну та надійну роботу.

Data Visualization є процесом перетворення даних у візуальні форми, такі як графіки, діаграми, картографічні зображення та інші графічні представлення, що допомагають розуміти великі обсяги даних шляхом їх візуального аналізу. Цей підхід дозволяє виявляти закономірності, тренди, аномалії та взаємозв'язки між даними, які можуть бути важко або неможливо виявити за допомогою традиційних методів аналізу[9]. Data Visualization використовує різноманітні інструменти та техніки для створення ефективних та інформативних візуальних представлень даних, які можуть бути корисними для прийняття управлінських рішень, розробки стратегій та виявлення нових можливостей. Цей процес може включати в себе використання різних типів графіків, від дуже простих до складних інтерактивних візуалізацій, залежно від потреб користувачів та характеристик даних.

Data Visualization є ключовим інструментом для розуміння складних взаємозв'язків між даними та виявлення важливих відношень, які можуть залишитися непоміченими без відповідного візуального аналізу. Це дозволяє аналітикам та дослідникам отримувати нові інсайти та зробити інформовані висновки на основі великого обсягу даних.

Один з важливих аспектів Data Visualization - це можливість відображення інформації в реальному часі, що дозволяє оперативно реагувати на зміни в даних та приймати швидкі та обгрунтовані рішення. Це особливо важливо в умовах швидко змінюючогося середовища, такого як фінансові ринки або мережі соціальних медіа.

Крім того, Data Visualization дозволяє ефективно спілкуватися результатами аналізу даних з іншими учасниками проекту чи зацікавленими сторонами. Візуальне представлення даних часто є більш зрозумілим та привабливим способом передачі інформації, ніж числові або текстові дані.

Необхідно також враховувати етичні аспекти використання Data Visualization, зокрема, необхідність захисту конфіденційності та приватності даних під час їх візуалізації. Це включає в себе дотримання відповідних правил і стандартів щодо обробки та зберігання особистої інформації[34].

Загалом, Data Visualization відіграє важливу роль у сучасному світі, допомагаючи зробити складні дані більш доступними та зрозумілими, а також сприяючи прийняттю обгрунтованих рішень на основі аналізу даних.

Інтеграція з Oracle Database в середовищі Oracle Big Data Lite VM є значущим аспектом, який розширює можливості аналізу та обробки даних. Ця інтеграція відкриває нові перспективи для користувачів у використанні великих обсягів даних, збережених у Oracle Database, у поєднанні з потужними інструментами Hadoop та Apache Spark, що входять до складу Oracle Big Data Lite VM.

Завдяки інтеграції з Oracle Database, користувачі можуть легко взаємодіяти з даними, які зберігаються у структурованому форматі, використовуючи звичні SQL-запити та інструменти аналізу, що надає ця база даних. При цьому вони також мають можливість використовувати потужні можливості Hadoop та Spark для обробки великих обсягів неструктурованих даних, що доповнюють аналітичні можливості Oracle Database.

Ця інтеграція дозволяє користувачам з легкістю обробляти та аналізувати різноманітні дані, незалежно від їх формату та походження, використовуючи оптимальні інструменти та технології для кожного конкретного завдання. Крім того, вона спрощує процес інтеграції та обміну даними між різними системами, що дозволяє підприємствам ефективно використовувати свої ресурси та здійснювати інформаційну аналітику на високому рівні.

Такий підхід до інтеграції з Oracle Database в середовищі Oracle Big Data Lite VM сприяє створенню гнучкої та масштабованої аналітичної інфраструктури, яка відповідає вимогам сучасного бізнесу та дозволяє використовувати всі переваги обробки та аналізу великих обсягів даних.

РОЗДІЛ 2 АНАЛІЗ ТА ОБРОБКА ДАНИХ МЕТОДАМИ ORACLE BIG DATA

2.1 Обробка датасету у середовищі Oracle Big Data Lite Virtual Machine

Вікно середовища Oracle Big Data Lite Virtual Machine представлено на рисунку 2.1.

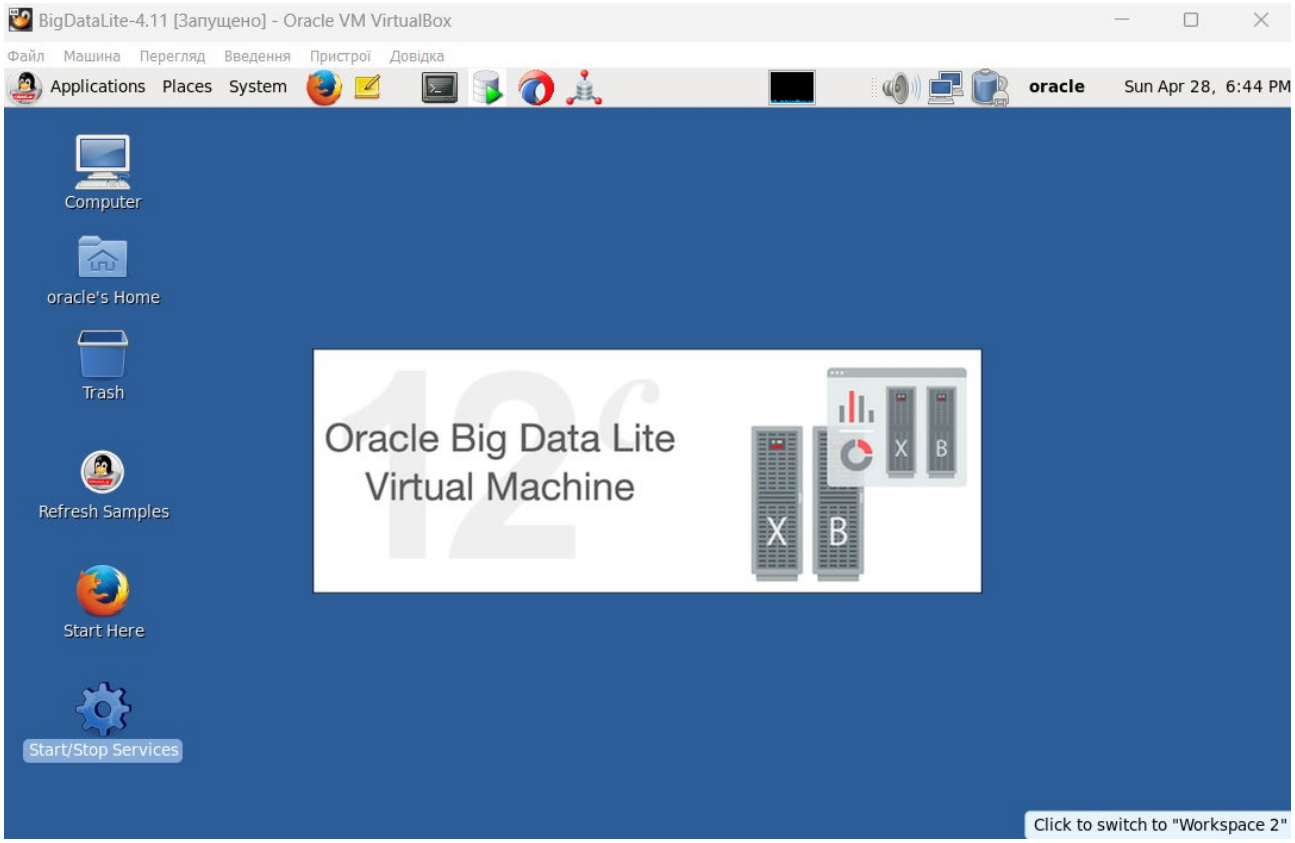


Рис. 2.1. Вікно середовища Oracle Big Data Lite Virtual Machine

Для дослідження Oracle Big Data Lite Virtual Machine було обрано датасет про населення країн світу та тенденцію їх зміни, який можна знайти на сервісі Worldometer.

Населення Землі змінюється внаслідок різних факторів, включаючи народжуваність, смертність та міграцію. Ось декілька ключових фактів:

Поточне населення Землі в 2024 році становить приблизно 8.1 мільярда людей[40]. Населення світу зростає зі швидкістю близько 0.91% на рік у 2024 році (що більше, ніж 0.88% у 2023 році, але менше, ніж 0.98% у 2020 році, і

1.06% у 2019 році)[1]. Поточне збільшення населення оцінюється приблизно на 73 мільйони людей на рік. Прогнози ООН вказують, що населення світу може зрости до близько 8.5 мільярдів у 2030 році та 9.7 мільярдів у 2050 році, досягнувши піку близько 10.4 мільярдів людей протягом 2080-х років.

За останні кілька століть населення Землі значно зросло. Наприклад, з 1800 року, коли населення Землі становило близько 1 мільярда, воно зросло до 8 мільярдів у 2024 році. Швидкість зростання населення світу досягла піку у 1962 та 1963 роках з річним темпом зростання 2.2%; але з того часу швидкість зростання населення світу зменшилася вдвічі.

Ці дані структуровані в файлі excel, який представлено на рисунку 2.2.

Country	Population(2024	Annual Change	Population Density (people per km²)	Growth Rate
Albania	2842321	-0.35 %	98.8702	0.9957
Algeria	44903225	1.57 %	18.8531	1.0164
American Sa	44273	-0.81 %	222.4774	0.9831
Angola	35588987	3.08 %	28.5466	1.0315
Anguilla	15857	0.26 %	174.2527	1.0066
Antigua and	93763	0.57 %	212.1335	1.0058
Armenia	2780469	-0.09 %	93.4831	0.9962
Aruba	106445	-0.16 %	591.3611	0.9991
Austria	8939617	0.22 %	106.5877	1.002
Azerbaijan	10358074	0.53 %	119.6082	1.0044
Bahamas	409984	0.64 %	29.4043	1.0051
Bahrain	1472233	0.90 %	1924.4876	1.0061
Bangladesh	171186372	1.03 %	1160.035	1.0108
Barbados	281635	0.13 %	654.9651	1.0015

Рис. 2.2. Частина даних про населення Землі

Після чого ці дані було конвертовано в форматі csv(рис.2.3).

```
Country;Population(2024);Annual Change;Population Density (people per km²);Growth Rate
Albania;2842321;-0.35 %;98.8702;0.9957
Algeria;44903225;1.57 %;18.8531;1.0164
American Samoa;44273;-0.81 %;222.4774;0.9831
Angola;35588987;3.08 %;28.5466;1.0315
Anguilla;15857;0.26 %;174.2527;1.0066
Antigua and Barbuda;93763;0.57 %;212.1335;1.0058
Armenia;2780469;-0.09 %;93.4831;0.9962
Aruba;106445;-0.16 %;591.3611;0.9991
Austria;8939617;0.22 %;106.5877;1.002
Azerbaijan;10358074;0.53 %;119.6082;1.0044
Bahamas;409984;0.64 %;29.4043;1.0051
Bahrain;1472233;0.90 %;1924.4876;1.0061
Bangladesh;171186372;1.03 %;1160.035;1.0108
Barbados;281635;0.13 %;654.9651;1.0015
```

Рис. 2.3. Частина даних в форматі csv

У середовищі Oracle Big Data Lite Virtual Machine (BDLVM) можна завантажити дані до Apache Hadoop за допомогою різних інструментів і технологій, таких як Apache Sqoop або просто копіюванням файлів на Hadoop Distributed File System (HDFS).

Перехід до середовища Hadoop: для цього потрібно відкрити термінал на BDLVM і перейти до середовища Hadoop, наприклад, введіть команду для входу до Hadoop:

$ ssh hduser@localhost

Завантаження даних на HDFS: потрібно ввести команду Hadoop `hadoop fs -copyFromLocal`, щоб скопіювати дані з локальної файлової системи на Hadoop Distributed File System (HDFS). Наприклад:

$ hadoop fs -copyFromLocal /local/path/to/your/data /hdfs/path/for/your/data

Завантаження даних представлено на рисунку 2.4.

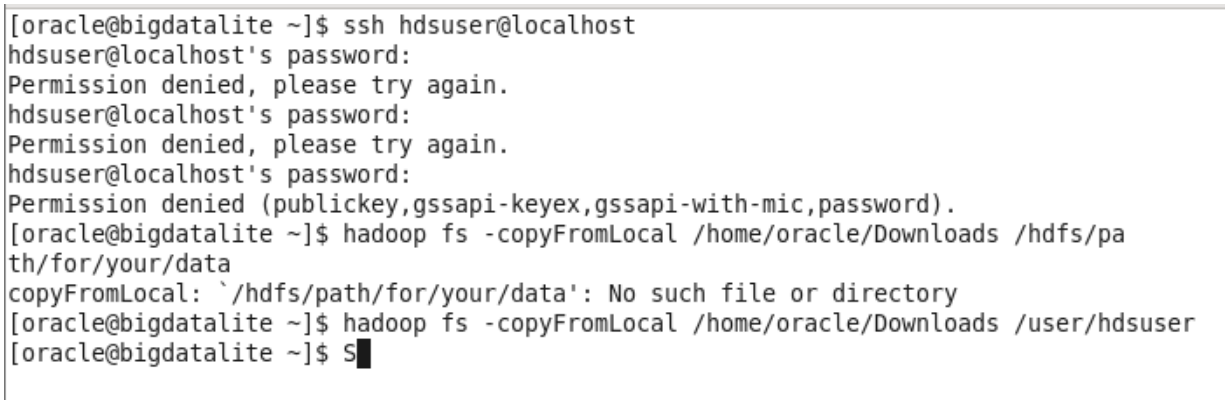


Рис. 2.4. Процес завантаження даних в середовище Apache Hadoop

Оскільки дані збережені в Apache Hadoop у форматі CSV та розташовані за шляхом `/user/hdsuser`, для їх перегляду за допомогою Apache Hive, потрібно виконати наступні кроки:

1. Створення таблиці Hive: спершу потрібно створити таблицю Hive, яка відображатиме ваші дані CSV:

```
CREATE TABLE my_csv_table (
   column1 STRING,
   column2 INT,
   column3 FLOAT
)
```

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

STORED AS TEXTFILE

LOCATION '/user/hdsuser';

У цьому прикладі `column1`, `column2`, `column3` – це назви стовпців CSV файлу, а типи даних (`STRING`, `INT`, `FLOAT`) повинні відповідати типам даних у стовпцях.

2. Виконання запиту: тепер, коли таблиця створена, можна виконати запити, щоб переглядати дані.

SELECT * FROM my_csv_table LIMIT 10;

Цей запит поверне перші 10 рядків з таблиці.

Виконання цих команд представлено на рисунку 2.5 – 2.8.

*create.sql

```
CREATE TABLE population_data (
    Country STRING,
    Population_2024 INT,
    Annual_Change FLOAT,
    Population_Density FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

Рис. 2.5. Sql-запит на створення таблиці

```
[oracle@bigdatalite ~]$ hive -f home/oracle/create.sql

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-1.1.0-cdh5.13.1.jar!/hive-log4j.p
perties
OK
Time taken: 5.3 seconds
WARN: The method class org.apache.commons.logging.impl.SLF4JLogFactory#release() was invoked.
WARN: Please see http://www.slf4j.org/codes.html#release for an explanation.
```

Рис.2.6. Створення таблиці з даними

*create.sql

```
SELECT * FROM population_data;
```

Рис. 2.7. Sql-запит на виведення даних

Ethiopia	126,527,060	2.55%	1271.0
Japan	123,294,513	-0.53%	3381.0
Philippines	117,337,368	1.54%	3941.0
Egypt	112,716,598	1.56%	1131.0
DR Congo	102,262,808	3.29%	451.0
Vietnam	98,858,950	0.68%	3191.0
Iran	89,172,767	0.70%	551.0
Turkey	85,816,199	0.56%	1121.0
Germany	83,294,633	-0.09%	2391.0
Thailand	71,801,279	0.15%	1411.0
United Kingdom	67,736,802	0.34%	2801.0
Tanzania	67,438,106	2.96%	761.0
France	64,756,584	0.20%	1181.0
South Africa	60,414,495	0.87%	501.0
Italy	58,870,763	-0.28%	2001.0
Kenya	55,100,587	1.99%	971.0
Myanmar	54,577,997	0.74%	841.0
Colombia	52,085,168	0.41%	471.0
South Korea	51,784,059	-0.06%	5331.0
Uganda	48,582,334	2.82%	2431.0
Sudan	48,109,006	2.63%	271.0
Spain	47,519,628	-0.08%	951.0
Argentina	45,773,884	0.58%	171.0
Algeria	45,606,481	1.57%	191.0
Iraq	45,504,560	2.27%	1051.0
Afghanistan	42,239,854	2.70%	651.0

Рис. 2.8. Результат виведення даних

Дані про населення Землі, які були зібрані, структуровані та завантажені в середовище Apache Hadoop, піддаються подальшій обробці та аналізу для використання в задачах машинного навчання. Після їх інтеграції в систему обробки даних, виконуються різні методи і техніки обробки даних для підготовки їх до моделювання.

Першим етапом обробки даних є виконання операцій перегляду, фільтрації та очищення даних. Це включає видалення дублікатів, обробку відсутніх значень, нормалізацію даних тощо. Наприклад, дані про населення можуть містити різні формати або помилки, які потрібно виправити перед подальшим аналізом.

Далі виконуються етапи агрегації та групування для отримання статистичних даних або побудови підсумкових звітів. Наприклад, обчислюють середнє значення народжуваності за роками або визначають регіони з найвищою та найнижчою смертністю.

Після цього переходять до етапу візуалізації даних, де використовуються різноманітні графіки та діаграми для наглядного представлення результатів

аналізу. Це допомагає зрозуміти патерни та відношення між різними змінними та факторами населення.

Після попереднього аналізу та візуалізації даних переходять до застосування методів машинного навчання для розв'язання конкретних завдань. Наприклад, побудову моделей прогнозування темпів зростання населення на основі історичних даних чи класифікацію регіонів за їхнім рівнем розвитку на основі різних показників населення.

Загалом, використання даних про населення Землі для машинного навчання сприяє розумінню демографічних тенденцій, прогнозуванню майбутніх змін та розробці стратегій управління ресурсами та соціальними програмами.

2.2 Статистичний аналіз отриманих струкутрованих даних

У рамках статистичного аналізу отриманих структурованих даних проводиться описова статистика, що дозволяє отримати загальне уявлення про основні характеристики даних. Наприклад, для числових змінних можуть бути розраховані такі параметри, як середнє значення, медіана, стандартне відхилення та діапазон варіації. Наприклад, середнє значення числової змінної може складати 50, медіана - 45, стандартне відхилення - 10, а діапазон варіації - від 20 до 80. Для категоріальних змінних проводиться розрахунок частоти входжень кожної категорії, а також може бути визначена мода - значення змінної, що найчастіше зустрічається в наборі даних. Наприклад, якщо у нашому наборі даних категоріальної змінної "стать" категорія "чоловіки" зустрічається 60 разів, а категорія "жінки" - 40 разів, то мода дорівнює "чоловіки". Описова статистика допомагає отримати уявлення про основні характеристики даних і визначити їх розподіл та тенденції.

Графічний аналіз в середовищі Oracle Big Data відіграє важливу роль у візуалізації та розумінні структурованих даних. Використання графіків дозволяє аналізувати розподіл даних, виявляти взаємозв'язки та тренди, а також

виявляти аномалії чи випадки, що виходять за рамки очікувань.

Наприклад, за допомогою візуалізацій можна побудувати гістограми для розуміння розподілу числових даних, таких як витрати або прибуток. Також можна побудувати діаграми розсіювання для вивчення взаємозв'язків між двома або більше змінними, наприклад, між витратами і доходами.

Крім того, графіки дозволяють візуалізувати результати статистичного аналізу, такого як кореляційний аналіз або аналіз варіації. Наприклад, можна побудувати графік лінійної регресії для відображення залежності між залежною та незалежною змінною.

Для великих обсягів даних, що характеризують Oracle Big Data, важливо використовувати ефективні та швидкі алгоритми для візуалізації, щоб забезпечити продуктивність та ефективність аналізу. Такі інструменти, як Apache Zeppelin або Jupyter Notebook, можуть бути використані для побудови графіків у великому масштабі даних.

В результаті графічний аналіз в середовищі Oracle Big Data допомагає зрозуміти дані, виявити тенденції та залежності, а також прийняти обґрунтовані управлінські та стратегічні рішення на основі даних.

У контексті Oracle Big Data, перевірка на нормальність даних є важливим етапом статистичного аналізу, який дозволяє встановити, чи відповідає розподіл даних нормальному розподілу. Це важливо для багатьох статистичних методів, що базуються на припущенні про нормальність даних, таких як t-тести, аналіз варіації (ANOVA) та інші.

Для проведення перевірки на нормальність в Oracle Big Data можна використати різні статистичні тести, такі як тест Шапіро-Уілка або тест Колмогорова-Смірнова. Ці тести оцінюють, наскільки добре розподіл даних відповідає теоретичному нормальному розподілу. Наприклад, тест Шапіро-Уілка перевіряє гіпотезу про те, що дані мають нормальний розподіл. Якщо p-значення, отримане в результаті цього тесту, менше встановленого рівня значущості (наприклад, 0.05), то гіпотеза про нормальність даних відхиляється[15].

Важливо враховувати, що перевірка на нормальність є статистичним тестом, який має свої обмеження. Наприклад, великі вибірки можуть показати статистичну значущість, навіть якщо різниця між спостереженим розподілом та нормальним розподілом є незначною з практичної точки зору. Також варто пам'ятати, що результати перевірки на нормальність можуть залежати від обраного методу тестування та величини вибірки. Тому важливо проводити аналіз з урахуванням цих обмежень і здійснювати відповідні корективи у випадку необхідності.

У середовищі Oracle Big Data можна виконати кореляційний аналіз для встановлення зв'язків між числовими змінними у структурованих даних. Для цього можна використати різноманітні інструменти та платформи, доступні у складі середовища, такі як Apache Spark, Apache Hive або інші[36]. Кореляційний аналіз дозволяє розрахувати коефіцієнти кореляції між парами змінних та визначити ступінь їх взаємозв'язку. Наприклад, можна визначити, чи існує статистично значима кореляція між обсягом продажів товарів та витратами на рекламу, щоб з'ясувати, наскільки ефективно використані рекламні кошти у відношенні до доходу. Для виконання кореляційного аналізу можна використовувати SQL-запити у Hive, які дозволяють розраховувати коефіцієнти кореляції за допомогою вбудованих функцій або використовувати Spark для виконання розрахунків у розподіленому середовищі. Такий аналіз допомагає розкрити залежності між різними змінними та виявити потенційні впливи, що можуть бути корисні для прийняття управлінських рішень та розробки стратегій.

Аналіз варіації (ANOVA) є статистичним методом, що використовується для порівняння середніх значень між трьома чи більше групами. У середовищі Oracle Big Data ANOVA може бути застосований до структурованих даних для виявлення статистично значущих різниць між групами. Для проведення ANOVA у Oracle Big Data можна скористатися вбудованими функціями та інструментами для обробки даних, такими як Apache Hive, Spark MLlib або Python бібліотеки, такі як pandas та scipy.stats[24].

Під час аналізу варіації важливо враховувати деякі аспекти, такі як відповідність умовам ANOVA, розподіл даних та вибір підходящої моделі. Крім того, проведення ANOVA може включати оцінку ефекту різниці між групами та інтерпретацію отриманих результатів.

Враховуючи обмеження обсягу та обробки даних у середовищі Oracle Big Data, проведення аналізу варіації може вимагати використання оптимізованих підходів до обробки даних, таких як збереження даних у відповідному форматі та використання розподілених обчислень для підвищення швидкодії обробки. Таким чином, ANOVA може бути корисним інструментом для порівняння групових різниць у великих обсягах даних, що зберігаються та оброблюються у середовищі Oracle Big Data.

У середовищі Oracle Big Data можна проводити регресійний аналіз для моделювання залежності між змінними та прогнозування значень цільової змінної на основі інших змінних. Для цього можна використовувати різні інструменти та технології, такі як Apache Spark MLlib, Python з бібліотеками Scikit-learn або власні розробки на мовах програмування, таких як Java або Scala.

Першим кроком у регресійному аналізі є підготовка даних, включаючи вибір змінних для аналізу та розбиття даних на тренувальний та тестовий набори для оцінки ефективності моделі. Потім проводиться вибір моделі регресії, такої як проста лінійна регресія або більш складні моделі, наприклад, поліноміальна регресія чи регресія з використанням нейронних мереж.

Після побудови моделі проводиться аналіз її ефективності за допомогою різних метрик, таких як середньо-квадратична помилка (MSE), коефіцієнт детермінації (R-squared) та інші. Важливо також провести аналіз важливості змінних та оцінити їх вплив на цільову змінну[11].

Завершальним кроком є використання побудованої моделі для прогнозування значень цільової змінної для нових даних. Для цього модель може бути використана у різних аналітичних або додаткових застосунках, які інтегровані з Oracle Big Data, для отримання прогнозів та прийняття рішень на

основі цих прогнозів.

У середовищі Oracle Big Data можна використовувати кластерний аналіз для виявлення схожих груп об'єктів у великих обсягах даних. Цей метод аналізу дозволяє розділити дані на кластери на основі їх схожості за допомогою різних алгоритмів, таких як k-середніх, ієрархічна кластеризація або методи згуртування. Після формування кластерів можна провести аналіз кожного кластера для виявлення його характеристик та особливостей. Наприклад, можна визначити ключові фактори, що відрізняють кожен кластер, або провести класифікацію нових об'єктів, що потрапляють у вже сформовані кластери. Кластерний аналіз може бути корисним для виявлення груп споживачів зі схожими вподобаннями, географічних областей зі схожими характеристиками або виявлення аномалій у великих масивах даних. Цей метод є важливим інструментом у сфері обробки та аналізу даних, особливо коли маємо справу з великими обсягами структурованих даних, які потребують класифікації або групування для подальшого використання[29].

Дискримінантний аналіз - це статистичний метод, що використовується для визначення відмінностей між двома або більше групами об'єктів на основі їх характеристик[39]. У контексті Oracle Big Data, дискримінантний аналіз може бути використаний для класифікації об'єктів у великих наборах даних за допомогою існуючих характеристик.

Для проведення дискримінантного аналізу у середовищі Oracle Big Data можна використовувати спеціалізовані алгоритми та програмні бібліотеки, які надають зручний інтерфейс для обробки та аналізу даних. Наприклад, можна використовувати статистичний пакет R або Python з бібліотеками, такими як scikit-learn або StatsModels.

Перед проведенням дискримінантного аналізу, необхідно визначити групи об'єктів та вибрати характеристики (змінні), за якими буде проводитися класифікація. Потім проводяться розрахунки для визначення, наскільки добре вибрані характеристики розділяють групи, та які саме характеристики найбільше впливають на класифікацію.

Результатом дискримінантного аналізу може бути модель, яка дозволяє класифікувати нові об'єкти на основі їх характеристик. Ця модель може бути використана для прогнозування або прийняття рішень на основі вхідних даних.

Під часовим рядом в середовищі Oracle Big Data розуміється послідовність даних, зібраних в різні моменти часу. Ці дані можуть бути зібрані з різних джерел, таких як датчики, журнали подій, або інші джерела, і можуть включати інформацію про певні параметри або показники протягом певного періоду часу. Обробка часових рядів у середовищі Oracle Big Data включає в себе різноманітні завдання, такі як прогнозування майбутніх значень, виявлення трендів та сезонності, аналіз змін у часі та ідентифікацію аномалій.

Для аналізу часових рядів у середовищі Oracle Big Data можна використовувати різні інструменти та техніки. Наприклад, можна використовувати Apache Spark для обробки великих обсягів даних та виконання розрахунків у реальному часі. Також можна використовувати Hadoop для зберігання та обробки великих обсягів часових рядів у розподіленому середовищі. Для виявлення трендів та сезонності можна використовувати різноманітні статистичні методи, такі як експоненціальне згладжування, аналіз ARIMA або сезонний декомпозиційний аналіз. Також можна використовувати машинне навчання для прогнозування майбутніх значень часових рядів на основі історичних даних.

Загалом, обробка часових рядів у середовищі Oracle Big Data відкриває широкі можливості для аналізу та використання часової інформації для прийняття рішень у різних сферах, таких як фінанси, маркетинг, виробництво та інші.

Багатовимірний аналіз у контексті Oracle Big Data відображається у застосуванні методів статистичного аналізу, спрямованих на розуміння взаємозв'язків та патернів у даних з великою кількістю змінних. Даний аналіз може включати методи зведеної статистики, такі як кореляційний аналіз, щоб визначити ступінь зв'язку між різними змінними у великому наборі даних. Крім того, можуть використовуватися методи факторного аналізу або аналізу

головних компонент для виявлення основних факторів, що впливають на змінні у датасеті.

Oracle Big Data надає інструменти для виконання багатовимірного аналізу, такі як Apache Spark MLlib або бібліотеки Python для машинного навчання, які дозволяють застосовувати різноманітні алгоритми аналізу даних. Ці інструменти дозволяють робити складні обчислення та моделювання на великих обсягах даних, що забезпечується розподіленою обробкою даних у середовищі Hadoop або Spark.

Oracle Big Data також надає можливості для інтеграції з іншими популярними інструментами та технологіями, такими як Apache Hadoop, Apache Spark, та іншими. Це дозволяє розширити функціональність та ефективність аналітичних процесів, використовуючи інструменти, які найкращим чином відповідають конкретним потребам та завданням.

Багатовимірний аналіз даних у середовищі Oracle Big Data дозволяє виявляти складні залежності та взаємозв'язки між різними параметрами, що допомагає зрозуміти природу даних та зробити обґрунтовані висновки. Це особливо важливо для прийняття управлінських рішень на основі аналізу великих обсягів даних.

Підтримка розподіленої обробки даних у середовищі Hadoop або Spark дозволяє швидко та ефективно працювати з великими обсягами даних, забезпечуючи високу продуктивність та масштабованість. Це дозволяє виконувати складні обчислення та аналізувати великі набори даних без затримок та перешкод.

Використання бібліотек Python для машинного навчання також дозволяє застосовувати різноманітні алгоритми для розв'язання завдань аналізу даних, включаючи класифікацію, кластеризацію, прогнозування та інші. Це робить Oracle Big Data універсальним інструментом для розв'язання різноманітних завдань аналізу та обробки даних.

Багатовимірний аналіз даних у середовищі Oracle Big Data може бути застосований у різних сферах, включаючи фінанси, маркетинг, медицину,

телекомунікації та інші. Наприклад, в фінансовій сфері багатовимірний аналіз може допомогти виявити фактори, що впливають на прибутковість портфеля інвестицій, а в медичній сфері – виявити взаємозв'язки між клінічними показниками та розвитком захворювань. Такий аналіз дозволяє виявляти складні взаємозв'язки та патерни у великих обсягах даних, що сприяє ухваленню обґрунтованих управлінських рішень.

РОЗДІЛ 3 АНАЛІЗ ДАНИХ МЕТОДАМИ ML

3.1 Опис методу машинного навчання

Великий обсяг даних, що становить виклик для аналізу та витягування корисної інформації, є характеристикою сучасного інформаційного світу. У контексті аналізу даних про населення Землі, використання методів машинного навчання виявляється як потужний інструмент для виявлення складних закономірностей та залежностей. Методи машинного навчання дозволяють автоматизувати процес аналізу даних, що включає у себе виявлення взаємозв'язків між різними атрибутами населення, прогнозування та класифікацію[4].

У цьому контексті, використання методів машинного навчання вимагає комплексного підходу до обробки даних, вибору моделей та їх навчання. Збір та підготовка даних є першими етапами, які передбачають обробку та очищення даних від шуму та аномалій, а також конвертацію даних у формат, придатний для подальшого використання. Вибір відповідних ознак та моделей також має велике значення, оскільки від цього залежить якість аналізу та прогнозів.

Навчання моделей машинного навчання вимагає використання великої кількості даних для виявлення закономірностей та встановлення оптимальних параметрів. Після завершення процесу навчання, моделі проходять валідацію на тестовому наборі даних для перевірки їхньої ефективності та генералізації на нових даних.

Оцінка результатів аналізу, виявлення важливих факторів та їхнє інтерпретація є ключовими кроками у використанні методів машинного навчання для аналізу даних про населення Землі. Подальша робота з моделями включає в себе впровадження їх у реальне середовище та постійний моніторинг їхньої продуктивності з метою забезпечення актуальності та точності аналізу.

Збір даних для аналізу населення Землі – це складний процес, який вимагає використання різноманітних джерел і методів. Великі дані, що

стосуються населення, можуть бути зібрані з різноманітних джерел, таких як демографічні агентства, урядові бази даних, соціальні мережі, датчики IoT. Ці дані можуть містити інформацію про розподіл населення за віком, статтю, етнічною приналежністю, географічним розташуванням, освітою, здоров'ям та іншими характеристиками.

Для збору даних можуть використовуватися різні методи, такі як опитування, аналіз великих текстових наборів даних (наприклад, відгуки від користувачів), використання сенсорів для збору фізіологічних даних, обробка зображень тощо. Збір даних також може включати автоматичне зчитування інформації з веб-сайтів за допомогою веб-скрапінгу або використання API для доступу до даних з різних джерел.

Однією з ключових викликів у зборі даних є забезпечення їх якості та точності. Це може вимагати фільтрації та очищення даних від шуму, аномальних значень або неправильно введених даних. Також важливо забезпечити конфіденційність та безпеку зібраних даних, особливо коли вони містять особисту інформацію про населення[19].

Загалом, збір даних про населення Землі є складним та багатоаспектним процесом, який вимагає використання різноманітних джерел та методів з метою забезпечення точності, якості та конфіденційності даних.

Підготовка даних для аналізу населення Землі вимагає комплексного підходу, щоб забезпечити якість та достовірність отриманих результатів. Перший етап включає збір різноманітних даних, що охоплюють демографічні, економічні, географічні та соціокультурні аспекти населення. Ці дані можуть бути отримані з різних джерел, включаючи національні статистичні органи, міжнародні організації, дослідницькі установи та інші джерела.

Далі проводиться аналіз та обробка цих даних з метою виявлення пропущених значень, аномалій та інших неправильностей. Цей процес включає в себе заповнення пропусків, видалення відхилень та нормалізацію даних для забезпечення однорідності та стабільності аналізу[18].

Для підготовки даних також може здійснюватися вибір ознак, що

найбільш інформативні для подальшого аналізу. Цей процес вимагає експертного аналізу та врахування доменних знань для визначення значущих факторів, які впливають на характеристики населення.

Після вибору ознак виконується їх трансформація та кодування у формат, придатний для використання в алгоритмах машинного навчання[3]. Це може включати в себе перетворення категоріальних даних у числовий формат, шкалювання значень та інші операції для забезпечення оптимальної роботи моделей.

Крім того, підготовка даних передбачає розбиття набору даних на тренувальний та тестовий набори для оцінки ефективності моделі. Цей процес дозволяє перевірити роботу моделі на незалежних даних та визначити її загальну здатність до узагальнення.

Отже, підготовка даних для аналізу населення Землі є складним та багатоетапним процесом, який вимагає врахування різноманітних аспектів якості та достовірності даних для отримання достовірних та об'єктивних результатів.

При виборі ознак для аналізу даних BD про населення Землі важливо враховувати різноманітні аспекти, які відображають сукупність характеристик та особливостей людей. Перш за все, слід врахувати демографічні ознаки, такі як вік, стать, етнічна належність, релігійні переконання тощо. Ці дані є ключовими для розуміння структури населення та виявлення залежностей між різними групами.

Додатково, до ознак можуть входити економічні показники, такі як дохід, рівень безробіття, рівень освіти та інші фактори, які відображають економічне становище і соціальний розвиток населення. Аналіз цих ознак може допомогти виявити тенденції у розподілі ресурсів та можливостей між різними групами.

Крім того, важливими ознаками можуть бути географічні дані, такі як місце проживання, регіональні особливості, кліматичні умови тощо. Ці дані можуть впливати на різноманітні аспекти життя людей, від доступу до ресурсів до схильності до певних захворювань.

Загалом, вибір ознак для аналізу даних про населення Землі повинен бути комплексним і враховувати різноманітні аспекти соціального, економічного, демографічного та культурного життя людей, щоб надати повний і глибокий розуміння структури та динаміки населення.

При виборі моделі для аналізу даних BD про населення Землі, необхідно враховувати ряд факторів, які впливають на ефективність та точність аналітичного процесу. Перш за все, важливо врахувати природу даних та завдання, яке необхідно вирішити. Наприклад, якщо дані мають складну структуру та високу розмірність, може бути доцільно використовувати глибокі нейронні мережі або інші складні моделі здатні виявляти складні залежності між ознаками. У той же час, якщо дані мають просту структуру та невеликий обсяг, ефективним може бути використання менш складних моделей, таких як лінійна регресія або метод опорних векторів.

Другим важливим аспектом є обчислювальні вимоги моделі. При аналізі великих обсягів даних, ефективність та швидкість обчислень можуть стати вирішальними факторами. Тому необхідно враховувати складність обчислювальних операцій, які пов'язані з кожною моделлю, та доступні ресурси для їх виконання.

Крім того, варто врахувати можливість інтерпретації результатів моделі. Особливо в сферах, де важливо зрозуміти причинно-наслідкові зв'язки між ознаками та результатами, використання моделей, які можна легко інтерпретувати, може бути перевагою.

Загалом, вибір моделі для аналізу даних BD про населення Землі - це компромісний процес, що враховує різноманітні фактори, такі як природа даних, обчислювальні ресурси, точність та інтерпретація результатів.

Навчання моделі в контексті аналізу даних BD про населення Землі є складним процесом, що включає в себе кілька етапів. Перш за все, необхідно визначити відповідну модель машинного навчання, яка відображатиме основні закономірності у великому обсязі даних. Це може бути, наприклад, дерево рішень, нейронна мережа або ансамбль моделей, такий як випадковий ліс.

Після вибору моделі проводиться підготовка даних, що включає в себе очищення від аномалій, заповнення пропущених значень та перетворення категоріальних ознак у числовий формат. Наступним етапом є розділення даних на тренувальний та тестовий набори для оцінки продуктивності моделі.

Після підготовки даних розпочинається процес навчання моделі. Це включає в себе використання тренувального набору даних для налаштування параметрів моделі з метою мінімізації функції втрат та підвищення її точності[2]. Під час цього процесу модель використовує алгоритм оптимізації, такий як стохастичний градієнтний спуск, для адаптації ваг моделі[22].

Після завершення процесу навчання моделі використовується тестовий набір даних для оцінки її продуктивності. Це допомагає визначити, наскільки добре модель узагальнює знання на нових даних, які вона не бачила під час навчання. У випадку задовільного результату модель може бути вважена готовою до використання для аналізу реальних даних про населення Землі.

Валідація моделі є ключовим етапом у процесі розробки машинного навчання для аналізу даних big data про населення Землі. Цей процес передбачає ретельну оцінку продуктивності моделі на відокремленому наборі даних, що не використовувалися під час її навчання. Основною метою валідації є визначення здатності моделі узагальнювати знання на нових, реальних даних про населення.

Під час валідації моделі використовуються різні метрики оцінки її продуктивності, такі як точність, чутливість, специфічність та F-міра. Ці метрики дозволяють оцінити якість класифікації (у випадку задач класифікації) або точність прогнозів (у випадку задач регресії).

Під час валідації також проводиться аналіз відповідності моделі реальним вимогам та специфікаціям завдання. Це може включати оцінку того, наскільки добре модель вирішує конкретні проблеми або відповідає на конкретні питання, що ставляться перед аналізом даних про населення[35].

Додатково, важливим аспектом валідації є перевірка на наявність перенавчання (overfitting) або недонавчання (underfitting) моделі. Перенавчання

може виникнути, коли модель надто добре пристосовується до навчальних даних і втрачає здатність узагальнювати знання на нові дані, тоді як недонавчання відбувається, коли модель недостатньо складна для відтворення складних залежностей у даних[30].

Отже, валідація моделі є важливим етапом у процесі розробки машинного навчання, який допомагає забезпечити надійність, стабільність та ефективність моделі в аналізі даних про населення Землі.

Гіперпарамертне налаштування є важливою складовою процесу розробки моделей машинного навчання і спрямована на досягнення оптимальної продуктивності моделі шляхом налагодження значень гіперпараметрів. Гіперпараметри є параметрами, які не вивчаються під час навчання моделі і встановлюються перед початком процесу навчання. Їх значення визначаються експериментально або на основі попередніх знань про дані та тип моделі. Наприклад, для моделей глибокого навчання гіперпараметрами можуть бути кількість шарів, кількість нейронів у кожному шарі, швидкість навчання тощо.

Процес гіперпараметрного налашутвання зазвичай включає в себе використання алгоритмів оптимізації, таких як згортка, пошук по сітці або оптимізація з використанням алгоритмів оптимізації градієнта. Ці алгоритми дозволяють систематично оцінювати різні комбінації гіперпараметрів та визначати ту, яка найкраще відповідає потребам конкретної задачі[23].

Головною метою гіперпараметрної настройки є зменшення функції втрат моделі на відокремленому наборі валідаційних даних. Це дозволяє забезпечити оптимальну загальну продуктивність моделі та уникнути перенавчання або недонавчання. Для досягнення цієї мети може знадобитися проведення повного перебору гіперпараметрів або використання евристичних методів для швидкого знаходження прийнятних рішень.

Після завершення процесу гіперпараметрної настройки важливо оцінити продуктивність кінцевої моделі на тестовому наборі даних, щоб переконатися, що вона ефективно узагальнює знання на нових даних і має високу загальну продуктивність. Такий підхід допомагає підтвердити ефективність розробленої

моделі та забезпечити її використання в практичних застосуваннях.

Впровадження моделі машинного навчання для аналізу даних BD про населення Землі передбачає створення програмного забезпечення, яке дозволяє інтегрувати навчену модель у вже існуючі інформаційні системи. Цей процес включає в себе розробку програмного коду для взаємодії з моделлю, створення API (інтерфейсу програмування додатків) для забезпечення доступу до моделі з інших додатків або систем, а також розгортання моделі на потужних обчислювальних ресурсах, щоб забезпечити високу швидкість обробки даних. Впровадження моделі також включає процес тестування, валідації та моніторингу її продуктивності в реальному середовищі, з метою забезпечення надійності та ефективності її роботи[37].

3.2. Обробка даних методами машинного навчання

Обробка даних методами машинного навчання включає в себе кілька етапів, починаючи зі збору та очищення даних, його подальшу обробку та аналіз з метою виявлення закономірностей та побудови прогностичних моделей. До даних про населення країн у форматі Country; 2024 Population; Annual Change; Population Density (people per km²); Growth Rate, було додатково зібрано дані з кількістю населення країн попередніх років (1970, 1980, 1990, 2000, 2010 та 2020), тому обробка може бути здійснена наступним чином.

Спочатку, дані піддаються структуризації, щоб забезпечити їхню однорідність та відповідність формату. Після цього проводиться етап очищення, в ході якого видаляються аномальні значення та виправляються помилки вводу, код подано у Додатку А.

Після очищення дані готуються для подальшого аналізу. Це включає перетворення категоріальних змінних, таких як назви країн, в числовий формат за допомогою кодування даних, наприклад, методом one-hot encoding.

Після підготовки даних проводиться їхнє подальше аналіз методами машинного навчання. Можливість використання різних методів, таких як

класифікація, кластеризація або регресія, залежить від конкретних цілей дослідження.

На цьому етапі виявляються закономірності між змінними та визначаються впливові фактори. Наприклад, можливо, буде встановлено, що густота населення та темпи приросту населення корелюють з економічними показниками країн.

Крім того, можуть бути побудовані прогностичні моделі, які дозволяють передбачити майбутні тенденції населення з урахуванням історичних даних. Для цього можуть використовуватися методи регресії або часових рядів.

Нарешті, результати аналізу та побудовані моделі піддаються оцінці та валідації для перевірки їхньої точності та відповідності вихідним даним.

На основі результатів аналізу даних за допомогою методів машинного навчання була розроблена програма на платформі Google Colaboratory за допомогою мови програмування Python. Під час розробки програми спочатку було використано бібліотеки для обробки та аналізу даних, такі як pandas та NumPy, для завантаження та підготовки вхідних даних.

Після цього було використано відповідні бібліотеки машинного навчання, такі як scikit-learn та TensorFlow, для побудови та навчання моделей, які використовуються для аналізу даних про населення.

У програмі було реалізовано алгоритми обробки даних, включаючи очищення, кодування категоріальних змінних, а також побудову та навчання моделей. Для цього використовувалися класи та методи зазначених бібліотек, які дозволяли легко взаємодіяти з даними та реалізовувати необхідні операції.

Крім того, незважаючи на те, що програма вже включає основні функції та можливості, вона може доповнюватись та вдосконалюватись у майбутньому, адже платформа Google Colaboratory забезпечує легку модифікацію та розширення програм.

В результаті було отримано гістограму з бочковою діаграмою розподілу даних щодо темпів зростання населення Землі (рис 3.1), код подано у Додатку Б.

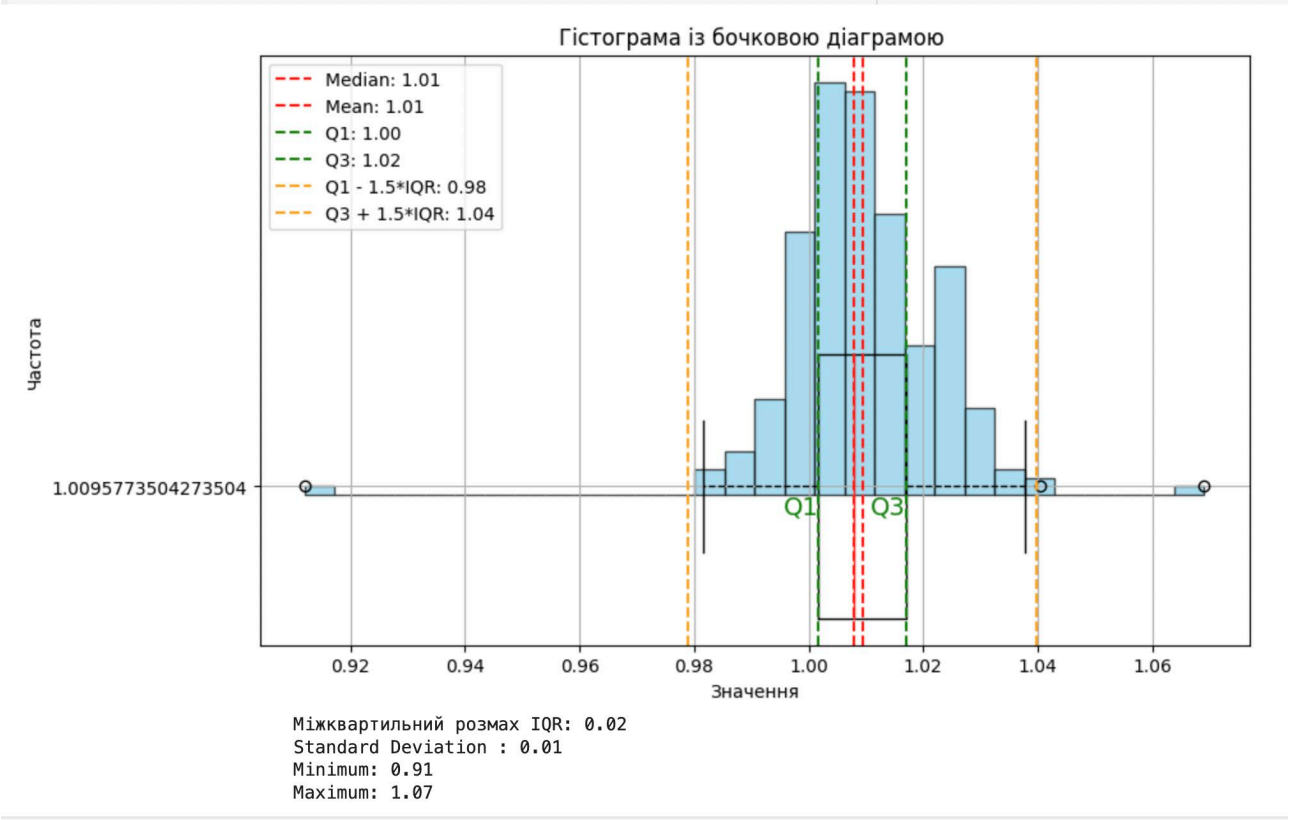


Рис. 3.1. Гістограма із бочковою діаграмою темпів зростання населення

З огляду на те, що гістограма відображає темп приросту населення з 1970 по 2024 рік, і всі значення зосереджені близько 1,01%, це означає, що за цей період середньорічний приріст населення був близький до 1,01%. Це характерно для тривалого періоду, де зміни в рівні народжуваності та смертності в різних країнах можуть значно змінюватися.

Ключові статистичні показники:

- Медіана: 1.01% - Це середній показник, який вказує на те, що половина країн має темпи зростання нижче 1.01%, а інша половина - вище.
- Середнє значення: 1.01% - Середній показник темпів зростання населення, що співпадає з медіаною, що вказує на симетричний розподіл.
- Q1 (перший квартиль): 1.00% - Показник, нижче якого знаходиться 25% даних.
- Q3 (третій квартиль): 1.02% - Показник, вище якого знаходиться 25% даних.
- IQR (міжквартильний розмах): 0.02% - Різниця між Q3 і Q1, що показує

діапазон середніх 50% даних.

- Мінімальне значення: 0.91%
- Максимальне значення: 1.07%
- Стандартне відхилення: 0.01%

На основі гістограми можна зробити висновки про темпи зростання населення для окремих країн. Детальніший розгляд приросту населення окремих країн виявляє різні демографічні тенденції. Країни з високим приростом населення (1.02% і вище), такі як Індія, Китай та Пакистан, мали стабільно високий рівень народжуваності протягом багатьох років, що відобразилося у значному прирості населення. Інші країни, як США та Індонезія, показують приріст близький до середнього значення (1.00%-1.02%) завдяки комбінації природного приросту та імміграції. На противагу цьому, країни з низьким приростом населення, такі як Японія та Німеччина, стикаються зі старінням населення і низькими рівнями народжуваності (менше 1.00%), що призводить до зменшення приросту або навіть від'ємних показників у деякі роки. Країни з дуже низьким приростом населення (менше 0.98%), такі як Україна та Болгарія, переживають економічні труднощі та демографічну кризу. Україна, зокрема, перебуває в стані війни, що значно впливає на демографічні показники через значну міграцію та втрати населення.

Також було побудовано гістограму з бочковою діаграмою приросту населення у 2024 році (Рис 3.2), код подано у Додатку В.

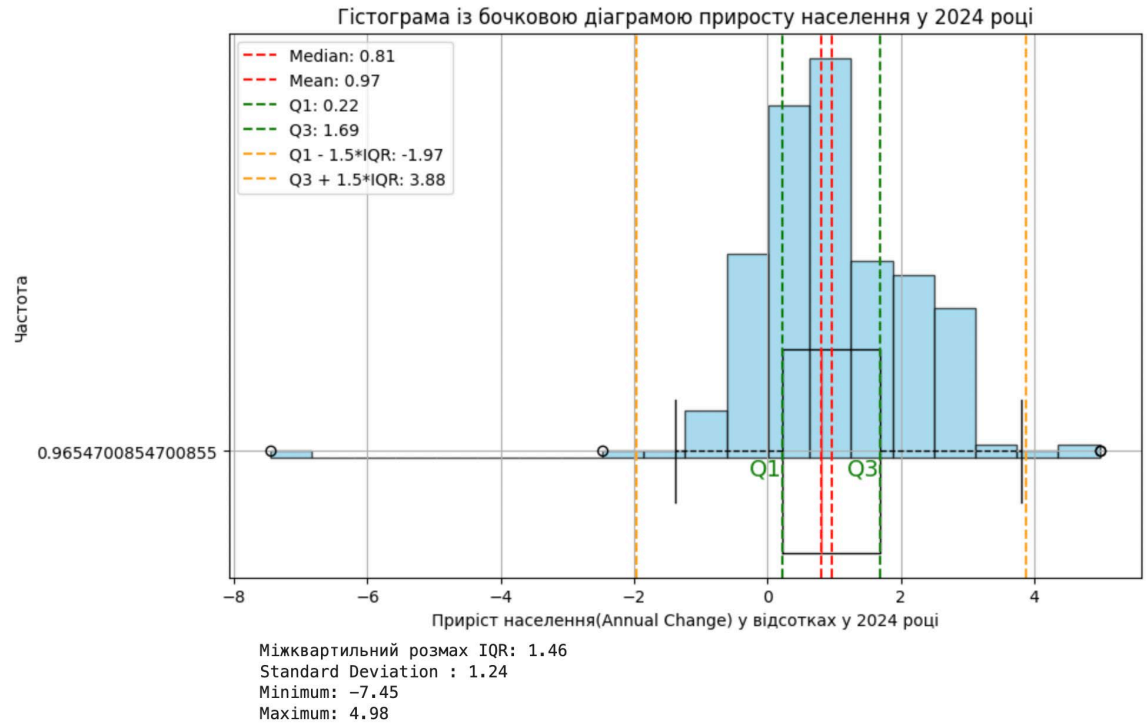


Рис. 3.2. Гістограма з бочковою діаграмою приросту населення у 2024 році

Гістограма з бочковою діаграмою надає загальний огляд приросту населення в різних країнах. Медіана приросту населення становить 0.81%, що вказує на те, що половина країн має приріст населення нижче цього значення, а інша половина – вище. Середнє значення приросту становить 0.97%, що трохи перевищує медіану. Це свідчить про наявність деяких країн з дуже високими показниками приросту, які впливають на середнє значення. Інтерквартильний розмах (IQR) становить 1.46%, що свідчить про значну варіабельність приросту населення між країнами. Перший квартиль (Q1) становить 0.22%, а третій квартиль (Q3) – 1.69%. Це означає, що 50% країн мають приріст населення в межах від 0.22% до 1.69%. Нижня межа (-1.97%) і верхня межа (3.88%) визначають діапазон нормальних значень приросту населення. Будь-які значення поза цими межами вважаються аномальними.

Розглянемо окремі країни з найвищими та найнижчими показниками приросту населення. Сирія демонструє значний приріст населення, з щорічною зміною на рівні 4.98%, що є одним з найвищих показників у світі. Це може бути пов'язано з поверненням біженців та відносною стабілізацією ситуації після

тривалого військового конфлікту. Водночас Нігер показує приріст на рівні 3.80%, що є характерним для країн з високим рівнем народжуваності та молодим населенням. Інші країни з високим приростом населення включають Малі (3.10%) та Анголу (3.08%). У протилежному кінці спектру знаходяться країни з негативним приростом населення. Україна має найнижчий показник приросту населення (-7.45%), що пов'язано з війною, міграцією та демографічною кризою. Південна Корея демонструє приріст на рівні -0.06%, що свідчить про демографічні виклики, такі як низька народжуваність і старіння населення. Іспанія також має негативний приріст (-0.08%), що може бути пов'язано з економічними труднощами та низькою народжуваністю. Інші країни, такі як Саудівська Аравія, Малайзія та Гана, показують помірний приріст населення. Саудівська Аравія зазнає приросту населення на рівні 1.48%, що свідчить про стабільний демографічний розвиток та значну міграційну активність. Такий рівень приросту походить як із природженого приросту, так із міграційних потоків, що вказує на ефективність політик, спрямованих на підтримку економічного зростання та створення привабливих умов для мігрантів. У Малайзії приріст населення складає 1.09%, що є стандартним для країн з розвиненою та стабільною економікою. Це свідчить про помірну народжуваність, яка сприяє економічному розвитку та соціальній стабільності, відображаючи успішну інтеграцію демографічних процесів та економічного зростання. Гана має приріст населення на рівні 1.93%, що вказує на високу народжуваність та молоде населення. Це характерне для країн, що перебувають на стадії демографічного переходу.

Таким чином, дані приросту населення дозволяють оцінити демографічні тенденції та їх вплив на соціально-економічний розвиток різних країн. Зокрема, високий приріст населення може створювати додаткове навантаження на інфраструктуру та ресурси, тоді як скорочення населення може викликати проблеми зі старінням населення та скороченням робочої сили.

Після цього було побудовано графік тенденції населення за період 1970-2024 років (Рис 3.3), код подано у Додатку Г, та тенденції зміни середнього

приросту населення (Рис 3.4), код подано у Додатку Д, які показують, що світове населення демонструє стійке зростання.

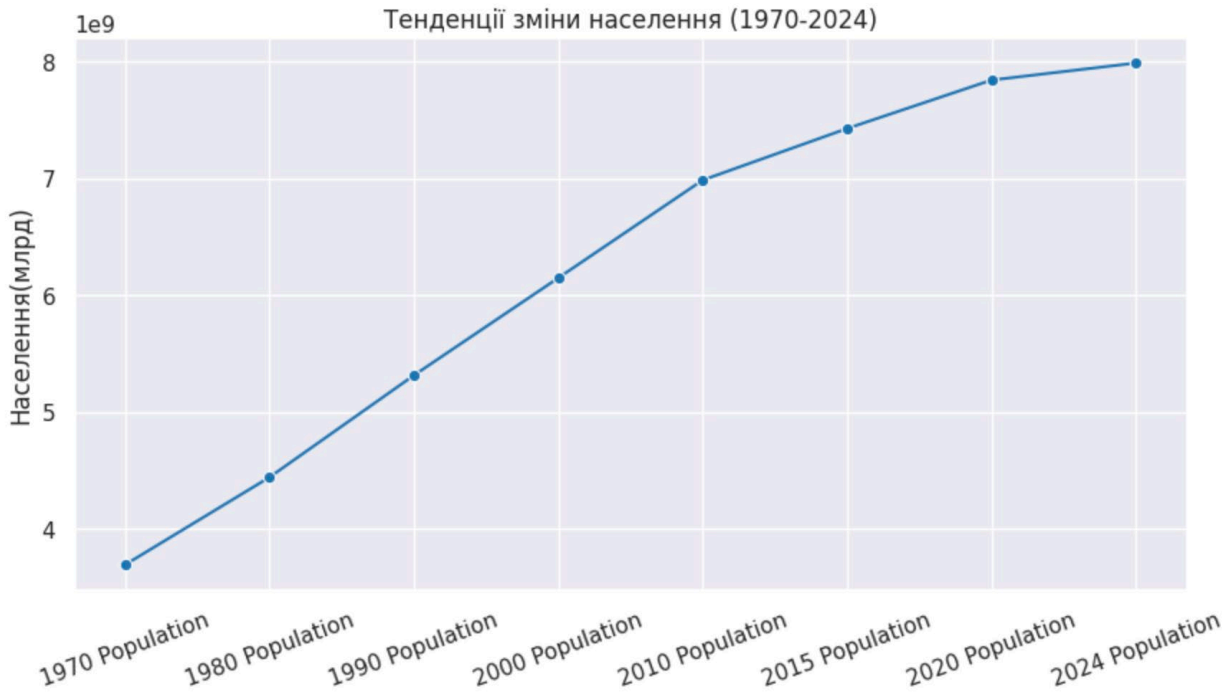


Рис. 3.3. Тенденції зміни населення (1970-2024)

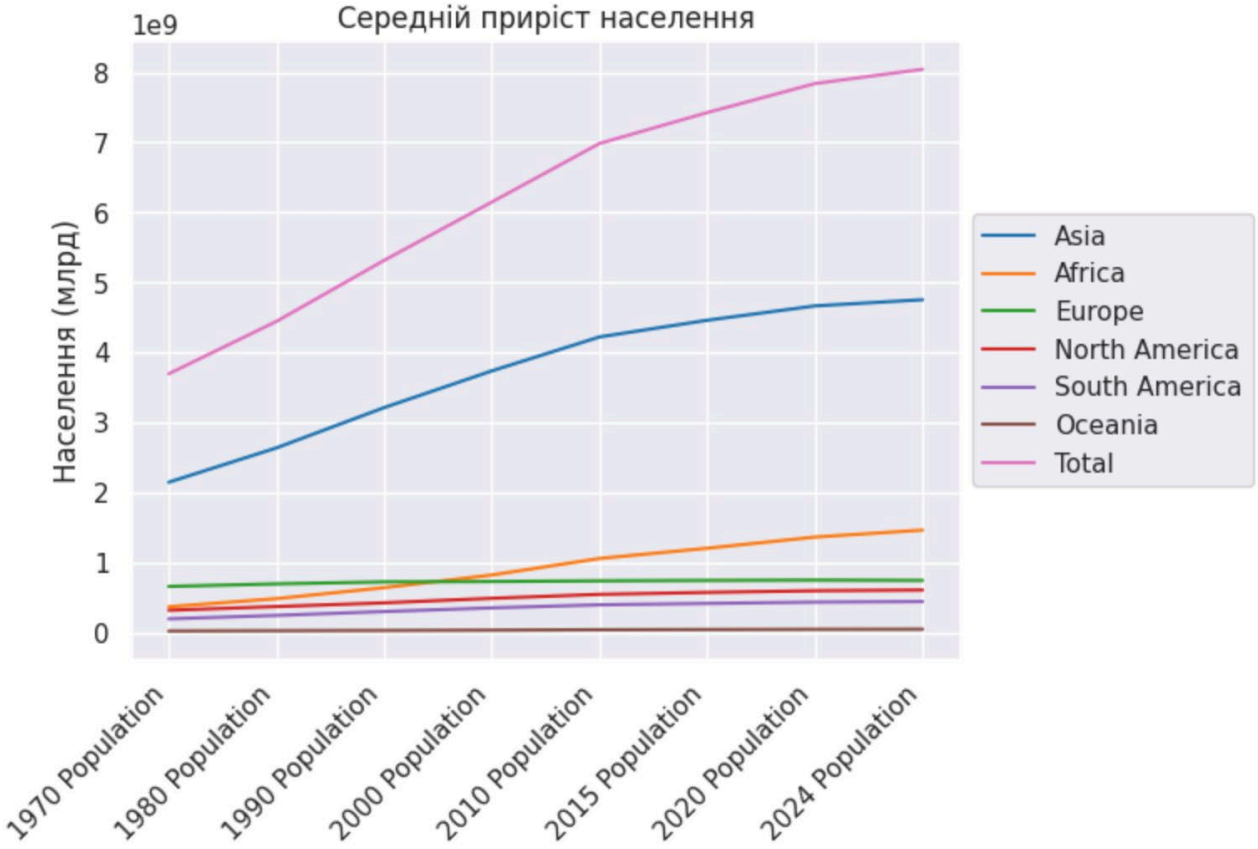


Рис 3.4. Тенденції зміни середнього приросту населення (1970-2024)

Перша тенденція вказує на постійне збільшення населення протягом зазначеного періоду. Графік чітко ілюструє значні демографічні зміни, які відбулися за останні десятиліття, та дозволяє робити висновки щодо майбутніх демографічних процесів. Друга тенденція показує що на кожному континенті населення з часом зростає, але найшвидше це відбувається з населенням Азії, після якої йде Африка.

Також було розроблено точкову діаграму питома вага території - питома вага населення країни у 2024 році (Рис 3.5), код подано у Додатку Е.

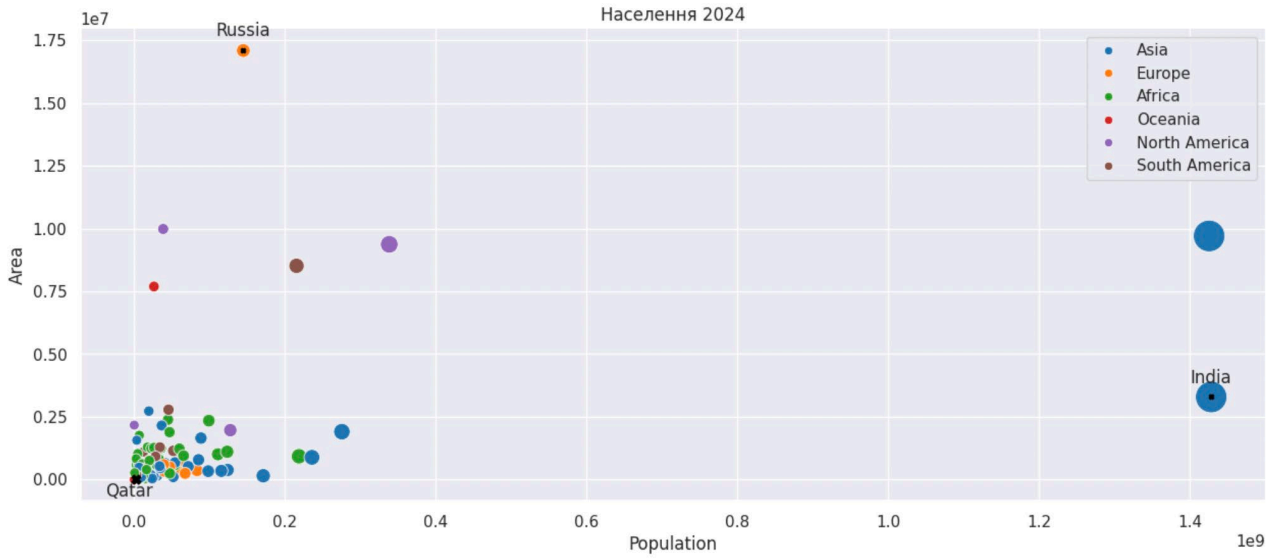


Рис. 3.5. Питома вага території - питома вага населення країни у 2024 році

Ця діаграма показує, що найбільше населення мають Індія та Китай, що відповідає поточним демографічним даним. Згідно з останніми оцінками, населення Індії становить близько 1.4 мільярда осіб, а Китаю — також близько 1.4 мільярда осіб. Наступними за чисельністю населення є Сполучені Штати Америки, Індонезія та Пакистан. Сполучені Штати мають населення близько 332 мільйонів осіб, Індонезія — 276 мільйонів, а Пакистан — близько 231 мільйона осіб.

Діаграма також демонструє співвідношення між населенням та площею країни, підкреслюючи такі країни, як Росія та Канада, які мають великі території, але відносно низьку щільність населення. Росія, найбільша країна за площею, має населення близько 146 мільйонів осіб, що значно менше

порівняно з її територією. Канада, друга за площею країна, має населення близько 38 мільйонів осіб.

Створена програма мовою програмування Python на платформі Google Colaboratory, яка виводить графік населення Землі та його тенденції на основі обробки даних про населення, є потужним інструментом для аналізу та візуалізації демографічних даних.

Завдяки використанню методів машинного навчання та бібліотек для обробки даних, програма забезпечує точний та швидкий аналіз великих обсягів інформації про населення. Шляхом побудови графіка населення Землі та його тенденцій програма дозволяє виявити ключові закономірності та зміни в динаміці населення країн.

Ця програма може бути корисною для дослідження та прийняття рішень у сферах економіки, демографії, медицини та соціології. Вона дозволяє аналізувати та передбачати тенденції у розвитку населення Землі на основі наявних даних, що важливо для планування соціально-економічних стратегій та ресурсного управління.

Узагальнюючи, створена програма є потужним інструментом для вивчення та аналізу динаміки населення Землі, що сприяє збільшенню розуміння та прийняттю обгрунтованих рішень в різних сферах глобального розвитку.

3.3. Пояснення результатів і формування висновків

Після обробки даних про населення Землі методами машинного навчання та аналізу результатів, можна зробити кілька висновків щодо динаміки та тенденцій у розподілі населення на глобальному рівні. Аналіз показав, що населення країн відрізняється за своєю кількістю, темпами приросту та густотою заселення. Зокрема, країни з великою кількістю населення, такі як Індія та Китай, мають найвищі загальні чисельність та щорічний приріст.

З іншого боку, країни з невеликою площею та великою густотою

населення, такі як Бангладеш та Нідерланди, можуть стикатися з проблемами перенаселення та екологічними проблемами. Динаміка зміни населення показує, що деякі країни, зокрема країни Африки, демонструють високі темпи приросту, що може мати значний вплив на глобальні демографічні та соціально-економічні тенденції у майбутньому.

Аналіз даних також вказує на регіональні різниці у динаміці населення, які можуть бути пов'язані з економічними, політичними та соціокультурними факторами. Наприклад, деякі розвинені країни, такі як Японія та деякі країни Європи, стикаються з проблемою старіння населення та зменшення приросту, що може мати важливі наслідки для економічного розвитку та соціального забезпечення.

Більш того, аналіз динаміки населення може бути корисним для прогнозування потреб у ресурсах, інфраструктурних проектів та розвитку соціальних послуг. Враховуючи тенденції зміни населення, країни можуть розробляти стратегії та політики, спрямовані на забезпечення сталого розвитку та поліпшення якості життя своїх громадян.

Додатково, аналіз динаміки населення може допомогти у прогнозуванні демографічних та соціально-економічних ризиків, таких як відтік робочої сили, нерівномірний розвиток регіонів та конфлікти через конкуренцію за ресурси. Враховуючи ці ризики, країни можуть приймати заздалегідь заходи для зменшення їхнього впливу та підтримки сталого розвитку.

Більш того, аналіз динаміки населення важливий для розроблення політик та програм з метою забезпечення сталого розвитку суспільства. Шляхом розуміння тенденцій у зміні розміру та складу населення можна планувати імміграційні політики, програми демографічної підтримки та соціально-економічні ініціативи. Аналізуючи динаміку населення, можна визначити потреби різних груп населення та розробляти програми, спрямовані на покращення їхнього благополуччя та якості життя. Такий підхід дозволяє країнам ефективно використовувати ресурси та планувати розвиток з урахуванням демографічних та соціально-економічних факторів.

Нарешті, аналіз динаміки населення може служити основою для розроблення міжнародних стратегій співпраці та розвитку, спрямованих на вирішення глобальних проблем, таких як бідність, голод, нерівність та кліматичні зміни. Враховуючи глобальні тенденції у розподілі населення, міжнародні організації та уряди можуть спільно розвивати стратегії для досягнення цілей сталого розвитку та забезпечення благополуччя всього людства.

ВИСНОВКИ

Під час проведення дослідження було встановлено, що інфраструктура Oracle Big Data відмінно справляється з обробкою великих обсягів даних та забезпечує швидкий доступ до необхідної інформації для аналізу. Вона дозволяє зберігати, обробляти та аналізувати дані з різних джерел, включаючи структуровані та неструктуровані дані.

Одним з ключових переваг використання інфраструктури Oracle Big Data є можливість використання різних інструментів та технологій для аналізу даних. Це включає в себе використання Hadoop, Apache Spark, Apache Hive та інших інструментів для обробки даних у великому масштабі.

Додатково, інфраструктура Oracle Big Data забезпечує високу надійність та масштабованість системи, що дозволяє ефективно працювати з великими обсягами даних та забезпечує стабільну роботу системи навіть у навантажених умовах.

У процесі аналізу даних важливо враховувати не лише технічні аспекти використання інфраструктури Oracle Big Data, а й методологічні підходи до аналізу даних. Це включає в себе визначення бізнес-задач, вибір відповідних методів аналізу, інтерпретацію результатів та формування рекомендацій для прийняття управлінських рішень.

Важливим етапом аналізу даних є їх візуалізація, яка дозволяє наглядно відображати результати та виявляти залежності між різними змінними. Для цього можна використовувати різноманітні графіки, діаграми та інші візуалізаційні засоби, що підвищує зрозумілість та доступність інформації для аналізу.

Нарешті, важливо підкреслити необхідність постійного моніторингу та оновлення інфраструктури Oracle Big Data, оскільки технології та методи аналізу даних постійно розвиваються. Це дозволить підтримувати високу ефективність та актуальність системи у процесі використання для аналізу інформації та прийняття управлінських рішень.

Крім вищезазначених висновків щодо використання інфраструктури Oracle Big Data у задачах аналізу інформації, важливо також врахувати значення написання програм мовою програмування Python на платформі Google Colaboratory для візуалізації результатів аналізу BD.

Програма, написана на мові програмування Python, забезпечує гнучкість та розширюваність у процесі візуалізації результатів аналізу даних. Python має багату екосистему бібліотек для візуалізації даних, таких як Matplotlib, Seaborn, Plotly, які надають широкі можливості для побудови різноманітних графіків та візуалізаційних ефектів. За допомогою цих бібліотек у ході цього дослідження було створено гістограми з бочковими діаграмами щодо темпів зростання населення Землі з 1970 по 2024 рік та приросту населення у 2024 році. Результати аналізу свідчать про те, що застосування гістограми з бочковою діаграмою дозволило отримати чітке уявлення про розподіл темпів зростання та приросту населення Землі.

Після цього було створено графік тенденції зміни населення, який свідчить про стійке зростання світового населення. Ця динаміка вказує на постійне збільшення чисельності населення протягом дослідженого періоду, що підтверджує прогнози, отримані за допомогою нашої моделі. Крім того, було розроблено точкову діаграму, яка враховує населення та площу кожної країни у 2024 році. Отримані результати аналізу підтверджують, що найбільше населення мають Індія та Китай, відображаючи поточні демографічні тенденції.

Написання програми мовою програмування Python для візуалізації результатів BD дозволяє легко та ефективно інтегрувати аналітичні дані з різних джерел та відображати їх у зручному та зрозумілому форматі для користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кількість населення Землі встановила новий рекорд. URL: https://telegraf.com.ua/ukr/nauka/2024-01-01/5825532-chislennost-naseleniya-zemli-pobila-novyy-rekord-skolko-lyudey-seychas-zhivet-na-planete (дата звернення 02.04.2024).
2. Машинне навчання: всебічний погляд на можливості і проблеми штучного інтелекту. URL: https://bizmag.com.ua/mashynne-navchannya/(дата звернення 12.04.2024).
3. Машинне навчання: основні поняття та алгоритми в data science. URL: https://goit.global/ua/articles/mashynne-navchannia-shcho-tse-take-iak-pratsiuie-i-dlia-choho-vykorystovuietsia/(дата звернення 12.04.2024).
4. Машинне навчання: що це таке, як працює і для чого використовується. URL: https://goit.global/ua/articles/mashynne-navchannia-shcho-tse-take-iak-pratsiuie-i-dlia-choho-vykorystovuietsia/ (дата звернення 12.04.2024).
5. Що таке big data?. URL: https://university.sigma.software/what-is-big-data/ (дата звернення: 21.03.2024).
6. Agrawal A. Prediction machines: the simple economics of artificial intelligence. 2018. 250 p.
7. Analysis of FIFA world cup data set. URL: https://indjst.org/articles/analysis-of-fifa-world-cup-data-set (date of access: 21.03.2024).
8. Ayres I. Super crunchers. New York : Random House Publishing Group, 2007.
9. Bergstrom C. T., West J. D. Calling bullshit: the art of scepticism in a data-driven world. Penguin Books, Limited, 2021.
10. Big Data: аналіз великих обсягів даних для прийняття рішень - IT рейтинг UA. URL: https://it-rating.ua/big-data-analiz-velikih-obsyagiv-danih-dlya-priynyattya-rishen (дата звернення: 21.03.2024).
11. Building A regression model for credit forecasting - fastercapital. FasterCapital. URL: https://fastercapital.com/topics/building-a-regression-model-for-credit-forecasting.html (date of access: 21.03.2024).

12. Chong D., Shi H. Big data analytics: a literature review. Journal of management analytics. 2015. Vol. 2, no. 3. P. 175–201. URL: https://doi.org/10.1080/23270012.2015.1082449 (date of access: 12.04.2024).
13. Cukier K. Learning with big data: the future of education. Houghton Mifflin Harcourt Publishing Company, 2014.
14. Data archiving. URL: https://success.outsystems.com/documentation/best_practices/architecture/data_archiving/ (date of access: 12.04.2024).
15. Descriptive statistics and normality tests for statistical data / A. Gupta et al. Annals of cardiac anaesthesia. 2019. Vol. 22, no. 1. P. 67. URL: https://doi.org/10.4103/aca.aca_157_18 (date of access: 20.04.2024).
16. Domingos P. The master algorithm: how the quest for the ultimate learning machine will remake our world. Basic Books, 2018. 352 p.
17. Gandomi A. H., Balusamy B., R N. A. Big data: concepts, technology, and architecture. Wiley & Sons, Limited, John, 2021. 400 p.
18. Gathering and analyzing population data - fastercapital. FasterCapital. URL: https://fastercapital.com/topics/gathering-and-analyzing-population-data.html/5 (date of access: 20.04.2024).
19. Guide to data collection for machine learning. AltexSoft. URL: https://www.altexsoft.com/blog/data-collection-machine-learning/ (date of access: 20.04.2024).
20. Gukanesh A. V., Kumar G. K., Saranya K. K. Twitter Data Analytics – Sentiment Analysis of An Election. International Journal of Trend in Scientific Research and Development. 2018. Volume-2, Issue-3. P. 1600–1603. URL: https://doi.org/10.31142/ijtsrd11457 (date of access: 07.05.2024).
21. Harford T. Data detective: ten easy rules to make sense of statistics. New York, USA : Penguin Publishing Group, 2021. 304 p.
22. How to prepare data before deploying a machine learning model? - geeksforgeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/how-to-

prepare-data-before-deploying-a-machine-learning-model/ (date of access: 21.03.2024).

23. Hyperparameter tuning a model (v2) - azure machine learning. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/azure/machine-learning/how-to-tune-hyperparameters?view=azureml-api-2 (date of access: 21.03.2024).
24. Kenton W. What is analysis of variance (ANOVA)?. Investopedia. URL: https://www.investopedia.com/terms/a/anova.asp (date of access: 20.04.2024).
25. Kumari R., Saini K. Data Mining and Predictive Analytics for Injection Molding: An Analysis. Indian Journal Of Science And Technology. 2023. Vol. 16, no. 45. P. 4131–4140. URL: https://doi.org/10.17485/ijst/v16i45.2567 (date of access: 21.03.2024).
26. Lindstrom M. Small data. Optimist, 2016.
27. Maximizing data processing speed with distributed computing. Custom software development company | MoldStud.com. URL: https://moldstud.com/articles/p-maximizing-data-processing-speed-with-distributed-computing (date of access: 12.04.2024).
28. Mayer-Schönberger V., Cukier K. Big data: a revolution that will transform how we live, work, and think. Eamon Dolan/Houghton Mifflin Harcourt, 2013. 256 p.
29. Oracle Help Center. URL: https://docs.oracle.com/en/database/oracle/machine-learning/oml4sql/21/dmcon/clustering.html (дата звернення: 12.04.2024).
30. Overfitting in machine learning: how to detect and avoid overfitting in computer vision?. The Complete Data Development Platform for AI | Encord. URL: https://encord.com/blog/overfitting-in-machine-learning/ (date of access: 12.04.2024).
31. Pauwels K. It's not the size of the data -- it's how you use it: smarter marketing with analytics and dashboards. AMACOM, 2014. 284 p.
32. Radi M. Big Data: Data Science. Independently Published, 2021.

33. Simon P. Too big to ignore: the business case for big data. Wiley & Sons, Limited, John, 2015.
34. The role of data visualization in descriptive analytics - fastercapital. FasterCapital. URL: https://fastercapital.com/topics/the-role-of-data-visualization-in-descriptive-analytics.html (date of access: 12.04.2024).
35. Training, validation, test split for machine learning datasets. The Complete Data Development Platform for AI | Encord. URL: https://encord.com/blog/train-val-test-split/ (date of access: 20.04.2024).
36. Turn data into insights with oracle big data. Oracle | Cloud Applications and Cloud Platform. URL: https://www.oracle.com/se/big-data/ (date of access: 12.04.2024).
37. Unlocking the power of machine learning: exploring open AI API for non-technical users. URL: https://impactum.mx/unlocking-the-power-of-machine-learning-exploring-open-ai-api-for-non-technical-users/ (date of access: 12.04.2024).
38. Véliz C. Privacy is power: why and how you should take back control of your data. Transworld Publishers Limited, 2021.
39. What is Discriminant analysis?. LinkedIn. URL: https://www.linkedin.com/pulse/what-discriminant-analysis-aionlinecourse-y0ykc/ (date of access: 12.04.2024).
40. Worldometer – світова статистика у реальному часі. URL: https://www.worldometers.info/uk/ (дата звернення 02.04.2024).
41. Wright S. J., Recht B. Optimization for Data Analysis. Cambridge University Press, 2021.
42. Zaharia M., Chambers B. Spark: the definitive guide: big data processing made simple. O'Reilly Media, 2018. 606 p.

ДОДАТКИ

Додаток А

«Лістинг програмного коду. Зчитування та обробка даних»

```
import tkinter as tk
!pip install datacleaner

#Ігнорувати попередження
import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
from scipy import stats
pd.set_option('display.float_format', lambda x: '%.3f' % x)

import datacleaner
from datacleaner import autoclean

%matplotlib inline
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
```

```
# Зчитування файлу
df = pd.read_csv("/content/world_population.csv")

# Перегляд даних
df.info()



df.dtypes
df.describe()

df.value_counts()

df.nunique()

# Очищення даних
df = autoclean(df)
df.head()

df.describe()

df.isnull().sum()
df.duplicated().sum()
```

Додаток Б

«Лістинг програмного коду. Побудова гістограми із бочковою діаграмою темпів зростання населення»

```
data = df['Growth Rate']


# Рахуємо медіану, квартилі, IQR, моду, середнє квадратичне відхилення, мінімум та максимум
median = np.median(data)
mean = np.mean(data)
q1 = np.percentile(data, 25)
q3 = np.percentile(data, 75)
iqr = q3 - q1
mode = stats.mode(data)
std_dev = np.std(data)
minimum = np.min(data)
maximum = np.max(data)


# Створюємо фігуру та підграфіки
fig, ax = plt.subplots(figsize=(10, 6))


# Гістограма
ax.hist(data, bins=30, color='skyblue', edgecolor='black', alpha=0.7)


# Бокс-плот
boxprops = dict(color='black')
whiskerprops = dict(color='black', linestyle='--')
medianprops = dict(color='red')
capprops = dict(color='black')
ax.boxplot(data, vert=False, positions=[np.mean(data)], widths=30,
```

```
    boxprops=boxprops, whiskerprops=whiskerprops, medianprops=medianprops, capprops=capprops)

# Позначаємо медіану, квартилі та моду на графіку
ax.axvline(median, color='r', linestyle='--', label=f'Median: {median:.2f}')
ax.axvline(mean, color='r', linestyle='--', label=f'Mean: {mean:.2f}')
ax.axvline(q1, color='g', linestyle='--', label=f'Q1: {q1:.2f}')
ax.axvline(q3, color='g', linestyle='--', label=f'Q3: {q3:.2f}')
ax.axvline(q1 - 1.5 * iqr, color='orange', linestyle='--', label=f'Q1 - 1.5*IQR: {q1 - 1.5 * iqr:.2f}')
ax.axvline(q3 + 1.5 * iqr, color='orange', linestyle='--', label=f'Q3 + 1.5*IQR: {q3 + 1.5 * iqr:.2f}')

# Додаємо мітки для квартилів Q1 та Q3 на бокс-плоті
ax.text(q1, 0.1, 'Q1', rotation=0, va='top', ha='right', color='g', fontsize=14)
ax.text(q3, 0.1, 'Q3', rotation=0, va='top', ha='right', color='g', fontsize=14)

# Додаємо легенду
ax.legend()

# Відображаємо графік
plt.title('Гістограма із бочковою діаграмою ')
plt.xlabel('Значення')
plt.ylabel('Частота')
plt.grid(True)
plt.show()

print(f"                Міжквартильний розмах IQR: {iqr:.2f}")
print(f"                Standard Deviation : {std_dev:.2f}")
```

```
print(f"                Minimum: {minimum:.2f}")
print(f"                Maximum: {maximum:.2f}")
```

Додаток В

«Лістинг програмного коду. Побудова гістограми із бочковою діаграмою темпів зростання населення

```
data = df['Annual Change']


# Рахуємо медіану, квартилі, IQR, моду, середнє квадратичне відхилення, мінімум та максимум
median = np.median(data)
mean = np.mean(data)
q1 = np.percentile(data, 25)
q3 = np.percentile(data, 75)
iqr = q3 - q1
mode = stats.mode(data)
std_dev = np.std(data)
minimum = np.min(data)
maximum = np.max(data)


# Створюємо фігуру та підграфіки
fig, ax = plt.subplots(figsize=(10, 6))


# Гістограма
ax.hist(data, bins=30, color='skyblue', edgecolor='black', alpha=0.7)


# Бокс-плот
boxprops = dict(color='black')
whiskerprops = dict(color='black', linestyle='--')
medianprops = dict(color='red')
capprops = dict(color='black')
ax.boxplot(data, vert=False, positions=[np.mean(data)], widths=30,
```

```
    boxprops=boxprops, whiskerprops=whiskerprops, medianprops=medianprops, capprops=capprops)

# Позначаємо медіану, квартилі та моду на графіку
ax.axvline(median, color='r', linestyle='--', label=f'Median: {median:.2f}')
ax.axvline(mean, color='r', linestyle='--', label=f'Mean: {mean:.2f}')
ax.axvline(q1, color='g', linestyle='--', label=f'Q1: {q1:.2f}')
ax.axvline(q3, color='g', linestyle='--', label=f'Q3: {q3:.2f}')
ax.axvline(q1 - 1.5 * iqr, color='orange', linestyle='--', label=f'Q1 - 1.5*IQR: {q1 - 1.5 * iqr:.2f}')
ax.axvline(q3 + 1.5 * iqr, color='orange', linestyle='--', label=f'Q3 + 1.5*IQR: {q3 + 1.5 * iqr:.2f}')

# Додаємо мітки для квартилів Q1 та Q3 на бокс-плоті
ax.text(q1, 0.1, 'Q1', rotation=0, va='top', ha='right', color='g', fontsize=14)
ax.text(q3, 0.1, 'Q3', rotation=0, va='top', ha='right', color='g', fontsize=14)

# Додаємо легенду
ax.legend()

# Відображаємо графік
plt.title('Гістограма із бочковою діаграмою приросту населення у 2024 році')
plt.xlabel('Приріст населення(Annual Change) у відсотках у 2024 році')
plt.ylabel('Частота')
plt.grid(True)
plt.show()

print(f"          Міжквартильний розмах IQR: {iqr:.2f}")
print(f"          Standard Deviation : {std_dev:.2f}")
```

```
print(f"                Minimum: {minimum:.2f}")
print(f"                Maximum: {maximum:.2f}")
```

Додаток Г

«Лістинг програмного коду. Побудова графіку тенденції зміни населення (1970-2024)»

```
plt.subplots(figsize=(10,5))
trend = df.iloc[:,5:13].sum().sort_values(ascending=True)

sns.lineplot(x=trend.index, y=trend.values, marker="o")
plt.xticks(rotation=20)
plt.ylabel("Населення(млрд)")
plt.title("Тенденції зміни населення (1970-2024)")
plt.show()
```

Додаток Д

«Лістинг програмного коду. Побудова графіку тенденції зміни середнього приросту населення (1970-2024)»

```
df_continents = df_continents.transpose()
df_continents



ax = df_continents.plot()

# Додавання заголовка, мітки осі x та мітки осі y
ax.set_title('Середній приріст населення')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45, horizontalalignment='right')
ax.set_ylabel('Населення (млрд)')

ax.legend(loc='center left', bbox_to_anchor=(1.0, 0.5))

plt.show()
```

Додаток Е

«Лістинг програмного коду. Побудова точкового графіку питомої ваги території - питома вага населення країни у 2024 році

```
# Стилізація графіку
plt.figure(figsize = (15, 6))
plt.title("Населення 2024")
plt.xlabel('Population', fontsize = 2)
plt.ylabel('Area', fontsize = 2)
sns.set_style('darkgrid')
sns.set_context('notebook', font_scale = 1)
sns.despine()

# Фільтрація значень 'Відсоток населення світу'
filtered_df = df[df['World Population Percentage'] <= 100]  # Включити тільки континенти

sns.scatterplot(data = filtered_df, y = 'Area (km²)', x = '2024 Population', hue = 'Continent',
                size = 'World Population Percentage', sizes = (50, 500))

# Знаходження точок
furthest_pointx = df.loc[df['2024 Population'].idxmax()] # Індія
furthest_pointy = df.loc[df['Area (km²)'].idxmax()] # росія
macau_point = df.loc[167] # Макау

# Додавання маркера та мітки
plt.scatter(furthest_pointx['2024 Population'], furthest_pointx['Area (km²)'],
            color = 'black', marker = 'X', s = 10)
```

```
plt.annotate(furthest_pointx['Country/Territory'], (furthest_pointx['2024 Population'], furthest_pointx['Area (km²)']),
             textcoords = 'offset points', xytext = (0,10), ha = 'center')
plt.scatter(furthest_pointy['2024 Population'], furthest_pointy['Area (km²)'],
            color = 'black', marker = 'X', s = 10)
plt.annotate(furthest_pointy['Country/Territory'], (furthest_pointy['2024 Population'], furthest_pointy['Area (km²)']),
             textcoords = 'offset points', xytext = (0,10), ha = 'center')
plt.scatter(macau_point['2024 Population'], macau_point['Area (km²)'],
            color = 'black', marker = 'X', s = 40)
plt.annotate(macau_point['Country/Territory'], (macau_point['2024 Population'], macau_point['Area (km²)']),
             textcoords = "offset points", xytext = (-5,-12), ha = 'center')

# Налаштування легенди для континентів
handles, labels = plt.gca().get_legend_handles_labels()
continent_labels = filtered_df['Continent'].unique()

filtered_handles = [h for h, label in zip(handles, labels) if label in continent_labels]
filtered_labels = [label for label in labels if label in continent_labels]

plt.gca().legend(filtered_handles, filtered_labels, loc = 'upper right', bbox_to_anchor = (1, 1))

plt.show()
```