МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно- обчислювальних систем і управління

МІСЮРА Богдан Романович

Інтелектуальний модуль рекомендацій для персоналізації користувацького досвіду в електронній комерції/Intelligent recommendation module for personalizing user experience in e-commerce

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
Б. Р. Місюра
_____

Науковий керівник:
к.і.н., Г. В. Сапожник
_____

Кваліфікаційну роботу
допущено до захисту:

"___" _____ 20___ р.

Завідувач кафедри
_____ **М. П. Комар**

ТЕРНОПІЛЬ - 2024

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
«____» _____2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
МІСЮРІ Богдану Романовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Інтелектуальний модуль рекомендацій для персоналізації користувацького досвіду в електронній комерції/Intelligent recommendation module for personalizing user experience in e-commerce
керівник роботи Сапожник Григорій Вікторович, к.іст.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 12 грудня 2023 р. № 753.

2. Строк подання студентом закінченої кваліфікаційної роботи 15 травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
- Провести глибокий аналіз електронної комерції як предметної області.
- Вивчити принципи та методології сучасних рекомендаційних систем.
- Проаналізувати існуючі рішення у галузі рекомендаційних систем.
- Розробити алгоритмічне та інформаційне забезпечення рекомендаційного модуля.
- Провести детальний аналіз та моделювання використовуючи набір даних.
- Візуалізувати результати кластеризації та аналізу головних компонент.

5. Перелік графічного матеріалу в роботі:
- Архітектура рекомендаційного модуля
- Структура формування рекомендацій для персоналізації користувацького досвіду в електронній комерції
- графіки з результатами експериментальних досліджень.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2024 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2024 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2024 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2024 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2024 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 20.05.2024 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту у системі «Unicheck».	до 10.06.2024 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 14.06.2024 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.06. 2024 р.	

Студент_____ Б. Р. Місюра
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи_____ Г.В. Сапожник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Інтелектуальний модуль рекомендацій для персоналізації користувацького досвіду в електронній комерції» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 45 сторінок і містить 18 ілюстрацій, 6 таблиць, додатоки та 7 використаних джерел.

Метою роботи є розробка інтелектуального модуля рекомендацій, спрямованого на персоналізацію користувацького досвіду в електронній комерції.

Методами розроблення обрано метод аналізу (для дослідження існуючих підходів до рекомендаційних систем), метод синтезу (для поєднання переваг існуючих методів), методи моделювання (для представлення та дослідження процесів рекомендацій), метод порівняльного аналізу (для оцінювання адекватності моделі рекомендацій).

Внаслідок виконання роботи обґрунтовано раціональний підхід до розроблення моделей рекомендацій та розроблено програмний засіб, який дозволяє створювати і досліджувати моделі рекомендацій.

Результати дослідження можуть бути використані в науково-дослідних установах і підрозділах підприємств, що займаються розробленням моделей рекомендацій.

Ключові слова: РЕКОМЕНДАЦІЙНІ СИСТЕМИ, ПЕРСОНАЛІЗАЦІЯ, МАШИННЕ НАВЧАННЯ, ЕЛЕКТРОННА КОМЕРЦІЯ, ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ.

ANNOTATION

Qualification work on the topic «Intelligent recommendation module for personalizing user experience in e-commerce» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 45 pages and it contains 18 figures, 6 tables, an annex, and 7 sources.

The purpose of the work is to develop an intelligent recommendation module aimed at personalizing the user experience in e-commerce.

Research methods include analysis (to study existing approaches to recommendation systems), synthesis (to combine the advantages of existing methods), modeling (to represent and study recommendation processes), and comparative analysis (to evaluate the adequacy of the recommendation model).

As a result of the work, a rational approach to the development of recommendation models was substantiated, and a software tool was developed that allows creating and researching recommendation models.

The research results can be used in research institutions and enterprise departments involved in the development of recommendation models.

Keywords: RECOMMENDATION SYSTEMS, PERSONALIZATION, MACHINE LEARNING, E-COMMERCE, INTELLIGENT MODULE.

ЗМІСТ

ВСТУП

Актуальність дослідження полягає в тому, що в сучасному світі електронна комерція стає все більш важливою складовою економіки, а отже, підвищується значення персоналізованих рекомендацій для користувачів. Завдяки широкому доступу до Інтернету та зростаючому попиту на онлайн-покупки, підприємства електронної комерції активно шукають способи покращення користувацького досвіду. У цьому контексті рекомендаційні системи, здатні адаптуватися до індивідуальних потреб кожного користувача, стають ключовим інструментом для збільшення продажів та відтримання лояльності клієнтів.

Дослідження актуальне і з практичної точки зору, оскільки успішна реалізація інтелектуальних модулів рекомендацій може призвести до значного зростання конкурентоспроможності компаній в електронній комерції. Такі модулі дозволять підприємствам забезпечити клієнтам персоналізовані пропозиції, що відповідають їхнім індивідуальним потребам та вподобанням.

Крім того, актуальність дослідження підтверджується зростаючим інтересом до розробки та впровадження інноваційних технологій в електронній комерції. Штучний інтелект та машинне навчання вже знаходять широке застосування в різних галузях, і їхнє використання в рекомендаційних системах є логічним кроком для оптимізації процесу підбору товарів для користувачів.

Прагнення до підвищення ефективності та персоналізації в електронній комерції відображається в зростаючому обсязі досліджень та розробок у цій галузі. Широкий спектр методів та підходів, представлених у літературі, свідчить про активний інтерес до пошуку оптимальних стратегій підвищення ефективності рекомендаційних систем в електронній комерції.

Метою роботи є розробка інтелектуального модуля рекомендацій, спрямованого на персоналізацію користувацького досвіду в електронній комерції. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Провести глибокий аналіз електронної комерції як предметної області.
2. Вивчити принципи та методології сучасних рекомендаційних систем.
3. Проаналізувати існуючі рішення у галузі рекомендаційних систем.
4. Розробити алгоритмічне та інформаційне забезпечення рекомендаційного модуля.
5. Провести детальний аналіз та моделювання використовуючи набір даних.
6. Візуалізувати результати кластеризації та аналізу головних компонент.

Об'єктом дослідження є процес персоналізації користувацького досвіду в електронній комерції.

Предметом дослідження є методи рекомендаційних систем в контексті персоналізації користувацького досвіду в електронній комерції.

Методи дослідження включають аналіз існуючих рекомендаційних систем у сфері електронної комерції з метою вибору найбільш ефективних підходів для персоналізації користувацького досвіду. Це охоплює як кількісні методи, такі як аналіз даних та статистичні методи, так і якісні підходи, наприклад, огляд літератури та експертні оцінки. Використання цих методів дозволяє здійснити комплексний аналіз різноманітних аспектів рекомендаційних систем і вибрати оптимальний шлях для подальшого дослідження.

Практичне значення дослідження полягає в розробці і впровадженні ефективних інтелектуальних модулів рекомендацій для персоналізації користувацького досвіду в електронній комерції. Це дозволить підприємствам залучати більше клієнтів, підвищувати конверсію та збільшувати обсяги продажів за рахунок точних та релевантних рекомендацій продуктів або послуг. Крім того, вдосконалення рекомендаційних систем сприятиме

підвищенню задоволення клієнтів та збереженню їхньої лояльності, що має ключове значення для успішної діяльності в сучасному цифровому ринковому середовищі.

Структура та розмір роботи. Кваліфікаційна робота складається із вступу, трьох розділів, висновків і списку використаних джерел. Загальний обсяг роботи складає 45 сторінок комп'ютерного тексту, включаючи 18 рисунків і 6 таблиць. У списку використаних джерел зазначено 7 назв, які займають 2 сторінки.

Апробація результатів дослідження. Основні теоретичні положення роботи й практичні результати дослідження доповідалися й обговорювалися V Всеукраїнської мультидисциплінарної студентської наукової конференції «Розвиток сучасної науки: актуальні питання теорії та практики», яка відбулася 19 квітня 2024 року у місті Тернопіль, Україна.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Огляд електронної комерції як предметної області

Електронна комерція, знана також як е-комерція, почала свій розвиток у 1970-х роках з появи перших комп'ютерних мереж. ARPANET, предтеча сучасного Інтернету, та EDI (Electronic Data Interchange) були серед піонерів цифрової епохи, дозволяючи компаніям обмінюватися документами та проводити комерційні транзакції в електронному вигляді. Ці ранні технології заклали основу для подальшого розвитку е-комерції, яка значно трансформувалася з розвитком Інтернету та веб-технологій у кінці XX століття.

Протягом наступних десятиліть електронна комерція еволюціонувала у важливий сегмент світової економіки. Цей розвиток був підсиленний значним зростанням інтернет-технологій, що дозволило досягти глобальної доступності та зручності у торгівлі. Сучасний стан електронної комерції характеризується інтенсивною конкуренцією між платформами та магазинами, що спонукає їх до постійних інновацій та вдосконалення.

Одним із ключових аспектів сучасної е-комерції є використання мобільних технологій. Зі збільшенням кількості користувачів смартфонів та мобільного інтернету, магазини стикаються з необхідністю оптимізувати свої онлайн-платформи для мобільних пристроїв, забезпечуючи зручний і швидкий доступ до своїх товарів та послуг.

Інтеграція штучного інтелекту відкрила нові можливості для оптимізації комерційних процесів. Штучний інтелект використовується для аналізу великих обсягів даних про поведінку користувачів, їхні переваги та історію покупок, що дозволяє не тільки покращити процес вибору та покупки, але й значно персоналізувати користувацький досвід. Це створює можливості для

високоадаптивних рекомендаційних систем, які можуть пропонувати товари та послуги, що найкраще відповідають індивідуальним запитам та очікуванням клієнтів.

Така персоналізація не тільки сприяє зростанню продажів, але й підвищує лояльність клієнтів, роблячи їхній досвід покупки більш задовільним та комфортним.

Електронна комерція, яка за своєю суттю є цифровим проведенням торгівлі, охоплює широкий спектр бізнес-моделей, кожна з яких відіграє свою роль у формуванні сучасного ринку. Серед основних бізнес-моделей можна виділити B2B (business-to-business), B2C (business-to-consumer) і C2C (consumer-to-consumer), кожна з яких зазнала впливу технологічних інновацій і змінила свої оперативні стратегії відповідно до потреб цифрової епохи.

1. B2B моделі стали більш ефективними завдяки автоматизації процесів та використанню систем електронних закупівель, які дозволяють компаніям ефективно взаємодіяти між собою, мінімізуючи традиційні бар'єри.
2. B2C комерція використовує прямий зв'язок з кінцевим споживачем через онлайн платформи, де ключове значення має здатність системи рекомендацій адаптувати товарні пропозиції до потреб і вподобань користувачів.
3. C2C модель сприяла зростанню платформ, таких як eBay або Etsy, де користувачі можуть безпосередньо продавати товари один одному, створюючи децентралізовані ринкові майданчики.

Соціальні медіа трансформували пейзаж е-комерції, ставши потужним інструментом для прямих продажів. Бренди та малі підприємства використовують платформи соціальних медіа для непрямого маркетингу та безпосереднього залучення споживачів, що дозволяє їм більш ефективно взаємодіяти з клієнтурою і, відповідно, збільшувати конверсії продажів.

Технологія блокчейн пропонує революційний підхід до забезпечення безпеки транзакцій, дозволяючи здійснювати обмін данними і вартостями в надійний та незмінний спосіб, що має велике значення для збереження довіри споживачів.

Машинне навчання і аналітика великих даних стали основою для розробки алгоритмів, які можуть прогнозувати поведінку покупців, оптимізувати запаси та персоналізувати рекламні кампанії, використовуючи величезні обсяги зібраних даних для виявлення закономірностей і тенденцій покупок.

Інноваційні технології, такі як розширена реальність і голосові помічники, розширюють можливості взаємодії з товаром, дозволяючи споживачам віртуально "приміряти" товари або використовувати голос для управління покупками, що робить процес більш інтерактивним і персоналізованим.

Ці та інші тренди не тільки сприяють збільшенню ефективності електронної комерції, але й активно впливають на структуру споживчого ринку, змушуючи підприємства адаптувати нові стратегії та технології для виживання і успіху в умовах постійної конкуренції і змінних вимог споживачів.

1.2 Принципи та методології рекомендаційних систем

Принципи та методології рекомендаційних систем складають основу для створення інтелектуальних інструментів, що покликані персоналізувати досвід користувачів в електронній комерції. Рекомендаційні системи аналізують величезні обсяги даних про поведінку користувачів, їхні вподобання та історію взаємодій, щоб вибудовувати персоналізовані пропозиції продуктів або послуг. Цей підхід може значно підвищити ефективність маркетингових стратегій та сприяти підвищенню продажів.

Ключові принципи рекомендаційних систем:

1. Персоналізація: Рекомендаційні системи мають забезпечувати індивідуалізовані пропозиції для кожного користувача, виходячи з аналізу його минулих дій, переваг та інтеракцій на платформі.
2. Масштабованість: З огляду на великі обсяги користувачів та даних у системах е-комерції, рекомендаційні системи мають бути здатні масштабуватися для обробки збільшеного навантаження без втрати продуктивності.
3. Адаптивність: Системи повинні швидко реагувати на зміни у поведінці користувачів та ринкових умов, адаптуючи рекомендації в реальному часі.

Рекомендаційні системи можуть використовувати декілька підходів та алгоритмів, серед яких:

1. Колаборативна фільтрація: Цей метод базується на гіпотезі, що люди, які однаково оцінили один набір об'єктів, імовірно однаково оцінять інші об'єкти. Алгоритми колаборативної фільтрації можуть бути поділені на підходи засновані на пам'яті (користувач-користувач або елемент-елемент) та модельні (використання методів машинного навчання для прогнозування оцінок).
2. Контент-базована фільтрація: Цей підхід рекомендує об'єкти, схожі на ті, які користувач оцінив позитивно в минулому, використовуючи метадані об'єктів, такі як жанри фільмів або категорії товарів.
3. Гібридні системи: Комбінація колаборативної та контент-базованої фільтрації для подолання обмежень, інерентних кожному підходу, зокрема питання початкового холодного старту та обмеженої області охоплення.

Використання цих методологій дозволяє створювати більш точні та ефективні рекомендаційні системи, які можуть суттєво покращити користувацький досвід у сфері електронної комерції, забезпечуючи

користувачам товари та послуги, які найбільше відповідають їхнім інтересам та потребам.

1.3 Огляд існуючих рішень

Персоналізація користувацького досвіду в сфері електронної комерції є критично важливим аспектом, що сприяє підвищенню задоволення клієнтів і зростанню продажів. Розуміння індивідуальних переваг споживачів та їхніх поведінкових моделей дозволяє компаніям ефективніше звертатися до потреб кожного користувача. Інтелектуальні модулі рекомендацій, які використовують передові технології машинного навчання та штучного інтелекту, відіграють ключову роль у реалізації цих персоналізованих стратегій. Для ілюстрації цього, розглянемо п'ять досліджень, які демонструють різні підходи та методики у створенні та вдосконаленні інтелектуальних систем рекомендацій в контексті електронної комерції.

Таблиця 1.1 відображає назви вибраних досліджень, а також їхні переваги та недоліки на основі доступних даних.

Таблиця 1.1 - Огляд існуючих рішень

Назва дослідження	**Переваги**	**Недоліки**
"PCFinder: An Intelligent Product Recommendation Agent for E-Commerce" [1]	Ефективне інтегрування модулів пошуку та управління базами даних.	Обмеження дослідження у розширенні функціональності агентів.
"Personalized recommender systems in e-commerce and m-commerce: a comparative study" [2]	Порівняльний аналіз показує ключові відмінності та ефективність систем у різних контекстах.	Зосередження лише на двох типах комерції, може не враховувати інші потенційні області застосування.
"Improving the usability of an e-commerce web site through personalization" [3]	Покращення користувацького інтерфейсу шляхом персоналізації, що збільшує задоволеність користувача.	Можлива висока складність інтеграції та управління персоналізованими компонентами.
"Personalizing similar product recommendations in fashion e-commerce" [4]	Інноваційний підхід до персоналізації рекомендацій у модній індустрії, що підвищує релевантність продуктів.	Специфічність до однієї галузі може обмежувати універсальність методики.

"OntoCommerce: an ontology focused semantic framework for personalised product recommendation for user targeted e-commerce" [5]	Використання онтологій для семантичного розуміння користувацьких потреб і підвищення точності рекомендацій.	Потребує глибоких знань в області семантичних технологій та онтологій, що може ускладнити впровадження.

Наше дослідження відрізняється від представлених вище робіт за кількома ключовими параметрами. По-перше, ми зосереджуємося на глибокому інтегруванні поведінкових даних та контекстної інформації для створення динамічної моделі рекомендацій, що здатна адаптуватися до змін у поведінці споживачів в реальному часі. На відміну від статичних методів, які часто використовуються в традиційних системах, наш підхід використовує машинне навчання для аналізу та прогнозування потенційних інтересів користувача на основі його останніх активностей та взаємодій з продуктами. Це дозволяє забезпечити високу точність та релевантність рекомендацій.

По-друге, дане дослідження включає розробку алгоритмів для визначення впливу зовнішніх чинників, таких як сезонність, рекламні акції та соціальні впливи, на поведінку покупців. Ми розробили інтегровану систему, що враховує не тільки історичні дані покупок, але й актуальні тренди в соціальних медіа та веб-аналітиці. Цей комплексний підхід дозволяє не тільки адаптувати пропозиції до особистих уподобань користувачів, але й враховувати загальну ринкову ситуацію, що збільшує загальну ефективність рекомендаційної системи в електронній комерції.

1.4 Вибір перспективного шляху і постановка задачі дослідження

Актуальність нашого дослідження полягає у важливості персоналізації користувацького досвіду в умовах стрімко зростаючого ринку електронної комерції. Сучасний споживач знаходиться в центрі безлічі рекламних пропозицій, тому здатність точно і ефективно рекомендувати товари, які відповідають його потребам і перевагам, може значно збільшити лояльність

клієнтів та загальний обсяг продажів. Підвищення точності рекомендацій, заснованих на глибокому аналізі поведінкових патернів та індивідуальних характеристик користувачів, є ключовим чинником конкурентоспроможності на ринку.

Завдяки інтеграції передових технологій аналітики великих даних і машинного навчання, наша робота спрямована на створення інтелектуальної рекомендаційної системи, яка може адаптуватися до змін у споживацьких уподобаннях та динамічно реагувати на ринкові тенденції. Це не тільки сприяє підвищенню ефективності рекламних кампаній, але й забезпечує більшу задоволеність користувачів, що, у свою чергу, зміцнює позиції компанії на ринку. Таким чином, наше дослідження вносить важливий вклад у розвиток технологій для електронної комерції, акцентуючи на персоналізації як стратегічному ресурсі для досягнення бізнес-цілей.

Метою бакалаврської роботи є розробка інтелектуального модуля рекомендацій, спрямованого на персоналізацію користувацького досвіду в електронній комерції. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Провести глибокий аналіз електронної комерції як предметної області, включно з її історичним розвитком, сучасним станом та основними трендами.
2. Вивчити принципи та методології сучасних рекомендаційних систем, з акцентом на їхнє застосування в електронній комерції.
3. Проаналізувати існуючі рішення у галузі рекомендаційних систем, визначити їхні переваги та недоліки.
4. Розробити алгоритмічне та інформаційне забезпечення рекомендаційного модуля, зосередившись на архітектурі системи, виборі та оптимізації методів машинного навчання.

5. Провести детальний аналіз та моделювання використовуючи набір даних, зокрема за допомогою технік нормалізації даних, аналізу головних компонент та оптимізації кластеризації.
6. Візуалізувати результати кластеризації та аналізу головних компонент, аналізуючи їхню ефективність та потенційні впливи на користувацький досвід у контексті електронної комерції.

Завдання 1-3 мають бути розглянуті в теоретичних розділах роботи, а завдання 4-6 – у практичних розділах, що охоплюють розробку, оцінку та тестування алгоритму.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОГО МОДУЛЯ РЕКОМЕНДАЦІЙ

2.1 Архітектура рекомендаційного модуля

Розвиток цифрових технологій та зростаюча конкуренція на ринку електронної комерції вимагають від компаній знаходження нових способів залучення та утримання клієнтів. Одним із ключових інструментів у цьому процесі є персоналізація користувацького досвіду за допомогою інтелектуальних систем рекомендацій. Ці системи дозволяють підвищити задоволеність користувачів, забезпечуючи їм індивідуалізовані пропозиції та оптимізуючи користувацький інтерфейс на основі аналізу поведінки та переваг. Архітектура (рисунок 2.1) таких систем включає кілька ключових компонентів, які забезпечують збір, обробку та аналіз великих обсягів даних для формування точних та актуальних рекомендацій.

Ця діаграма (рисунок 2.1) компонентів ілюструє архітектуру інтелектуального модуля рекомендацій для персоналізації користувацького досвіду в електронній комерції. Модуль складається з трьох основних блоків: інтерфейс користувача, серверний бекенд та база даних.

1. Інтерфейс користувача:
    - Обробник запитів користувача: відповідає за прийом вхідних запитів від користувачів та передачу їх до серверного бекенду для обробки.
2. Серверний бекенд:
    - Обробка даних: цей компонент приймає дані від інтерфейсу користувача та готує їх до подальшої обробки. Він може включати завдання, такі як валідація даних, їх нормалізація чи вибірка необхідних даних для аналізу.

- Модель машинного навчання: застосовує алгоритми машинного навчання для аналізу оброблених даних, ідентифікації шаблонів поведінки користувачів та прогнозування потенційних інтересів.
- Двигун рекомендацій: використовує результати моделі машинного навчання для формування конкретних рекомендацій, які надсилаються назад до інтерфейсу користувача.

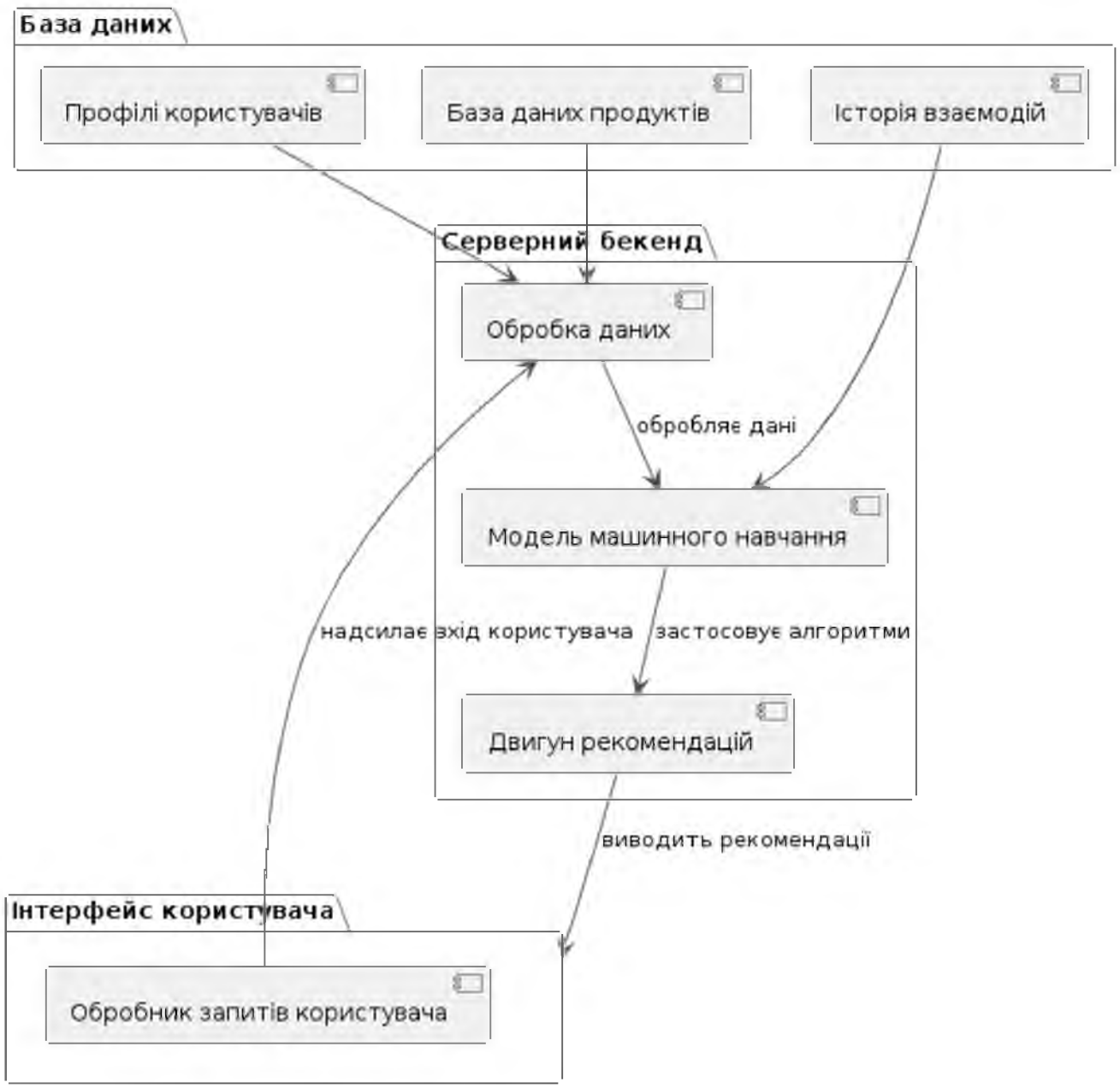


Рисунок 2.1 – Архітектура рекомендаційного модуля

3. База даних:
    - Профілі користувачів: зберігає інформацію про користувачів, їхні переваги, історію покупок та інші персональні дані.
    - База даних продуктів: містить інформацію про всі товари, доступні в електронній комерції, включаючи описи, ціни, категорії тощо.
    - Історія взаємодій: відстежує всі дії користувачів на платформі, включаючи перегляди сторінок, покупки та інші взаємодії, які можуть бути використані для покращення точності рекомендацій.

Взаємодія між цими компонентами відбувається через чітко визначені інтерфейси, де кожен компонент приймає вхідні дані від попереднього та передає результати до наступного, забезпечуючи плавний потік інформації та високу інтегрованість системи.

2.2 Методи машинного навчання для рекомендацій

2.2.1 Причини вибору кластеризпації для проєкту

Кластеризація є одним з основних методів навчання без учителя, що дозволяє систематизувати великі обсяги нерозмічених даних за допомогою групування об'єктів на основі подібності їхніх атрибутів. В контексті рекомендаційних систем, цей метод використовується для виявлення природних групувань або кластерів серед товарів чи користувачів, що може значно підвищити ефективність та точність рекомендацій.

Значення кластеризації у рекомендаційних системах:

1. Підвищення точності рекомендацій: Кластеризація дозволяє ідентифікувати групи користувачів зі схожими смаками та перевагами, що дозволяє рекомендувати товари, які сподобались іншим членам цих

груп. Це призводить до більш персоналізованого та влучного підходу до формування рекомендацій.

2. Ефективність обробки даних: Кластеризація зменшує обсяг даних, що обробляється, за рахунок агрегування користувачів та товарів у менші групи. Це спрощує модель, зменшує вимоги до обчислювальних ресурсів і покращує швидкість відгуку системи.
3. Зменшення проблеми розрідженості даних: Більшість рекомендаційних систем стикаються з викликом великої розрідженості матриць користувач-товар. Кластеризація допомагає управляти цією проблемою, об'єднуючи користувачів та товари з схожими характеристиками, що збільшує щільність даних в кожному кластері.
4. Виявлення нових зв'язків та інсайтів: Кластеризація може виявити неочевидні взаємозв'язки між продуктами та користувацькими поведінковими патернами, відкриваючи нові можливості для крос-продажу та упередженого маркетингу.
5. Адаптивність до змін у поведінці користувачів: Методи кластеризації можуть динамічно адаптуватися до змін у поведінці користувачів, швидко реагуючи на нові тренди та переваги без необхідності повного переучування моделі.

Для даного проєкту вибір певного методу кластеризації, такого як K-середніх або аналіз головних компонент, базується на специфічних вимогах до швидкості обробки, точності та можливостей інтерпретації результатів. Кожен з цих методів має свої переваги та обмеження, які враховуються під час рішення про їхнє застосування в контексті задачі.

Враховуючи наведені аргументи, кластеризація є стратегічним вибором для покращення персоналізації та ефективності рекомендаційних систем у сфері електронної комерції.

2.2.2 Аналіз головних компонент

Аналіз головних компонент (PCA) є статистичним методом, що використовується для зменшення розмірності даних зі збереженням якомога більшої кількості інформації. Цей метод широко застосовується в областях, де необхідно аналізувати великі набори даних, щоб виявити ключові структурні взаємозв'язки між змінними.

Основною ідеєю PCA є перетворення вихідного набору корелюючих змінних в новий набір некорелюючих змінних, які називаються головними компонентами. Ці компоненти впорядковані таким чином, що перша головна компонента має максимальну дисперсію (тобто, вона забезпечує найбільше "інформації" у вигляді дисперсії). Кожна наступна компонента, в свою чергу, має максимальну дисперсію при умові ортогональності до попередніх компонент.

1. Визначення коваріаційної матриці: Нехай X є $n \times p$ матрицею де n вказує кількість спостережень, а p - кількість змінних. Перший крок полягає в обчисленні коваріаційної матриці X, яка вимірює взаємозв'язки між змінними.
2. Обчислення власних значень і власних векторів: Коваріаційна матриця подається до процедури визначення власних значень та власних векторів. Власні вектори вказують напрямки нових осей, а власні значення відображають дисперсію, яку кожен власний вектор захоплює.
3. Сортування власних значень: Власні значення сортуються в порядку зменшення, і вибирається k найбільших для формування нового простору зменшеної розмірності.
4. Формування нового набору даних: Новий набір даних формується шляхом множення вихідних даних на власні вектори, що відповідають найбільшим власним значенням. Це перетворення результатує в нових осях, які є лінійно незалежними одна від одної.

В рекомендаційних системах для електронної комерції PCA може застосовуватися для виявлення основних факторів, які впливають на вибір споживачів, а також для зменшення розмірності наборів даних про поведінку користувачів, що, в свою чергу, може покращити ефективність та точність рекомендаційних алгоритмів. Зменшення кількості змінних допомагає уникнути проблеми перенавчання та підвищує обчислювальну ефективність алгоритмів, що важливо для обробки великих обсягів даних.

2.2.3 Алгоритм k-середніх

Алгоритм k-середніх є одним з найпопулярніших методів кластеризації, що використовується для поділу набору даних на кластери, в яких кожне спостереження належить до кластера з найближчим середнім значенням. Цей метод часто застосовується в аналізі даних для ідентифікації гомогенних груп у великому наборі даних.

Математично алгоритм k-середніх можна описати таким чином:

1. Ініціалізація: Оберіть k випадкових центроїдів з набору даних. Центроїди є середніми (центрами) кластерів.
2. Присвоєння: Кожне спостереження присвоюється до кластера, центроїд якого має найменшу відстань до цього спостереження. Відстань зазвичай вимірюється за допомогою евклідової метрики:

$$d(\mathbf{x}, \mathbf{c}) = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2}$$

де x є вектором спостереження, а c є вектором центроїда.

3. Оновлення центроїдів: Обчисліть нові центроїди шляхом визначення середнього значення всіх точок, присвоєних до кожного кластера.

4. Ітерація: Повторюйте кроки 2 і 3 до тих пір, поки центроїди не перестануть змінюватися або не буде досягнуто максимальної кількості ітерацій.

Основною метою алгоритму k-середніх є мінімізація суми квадратів відстаней між точками і їх найближчими центроїдами. Об'єктивна функція, яку алгоритм намагається мінімізувати, має вигляд:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|^2$$

де $\mathbf{x}(i)$ є спостереженнями, що належать до кластера j, $\mathbf{c}j$ є центроїдом кластера j, і nj є кількістю точок в кластері j.

У контексті персоналізації електронної комерції, алгоритм k-середніх може бути використаний для сегментації користувачів на основі їх покупкової поведінки або переглянутих продуктів. Такий підхід дозволяє виявляти групи користувачів з подібними інтересами або потребами, що в подальшому може бути використано для цілеспрямованих маркетингових кампаній або персоналізованих рекомендацій, підвищуючи ефективність взаємодії з клієнтом та його задоволення від користувацького досвіду.

2.3 Алгоритм рекомендації для персоналізації користувацького досвіду в електронній комерції

Завдяки поєднанню різних технологій та методів машинного навчання, таких як колаборативна фільтрація, контент-орієнтована фільтрація та гібридні системи, сучасні рекомендаційні системи можуть ефективно прогнозувати та визначати товари та послуги, які найбільш імовірно зацікавлять конкретного користувача. Ця персоналізація не тільки підвищує

задоволеність клієнтів, але й сприяє більшому взаємодійному та залученому досвіду покупок.

Алгоритм, який розроблено, включає кроки зі збору та аналізу даних, створення моделей для генерування рекомендацій, а також їх інтеграції та оцінки. Він призначений для того, щоб допомогти розробникам та маркетологам ефективно впроваджувати персоналізацію в їхні електронні комерційні рішення.

Рисунок 2.1 - Структура формування рекомендацій для персоналізації користувацького досвіду в електронній комерції

Кроки:

1. Збір даних про користувача:
    - Зберігайте історію переглядів, покупок та взаємодій користувача на сайті.

- Використовуйте форми для реєстрації, анкети або соціальні мережі для отримання інформації про вподобання та демографічні дані користувачів.

2. Обробка та аналіз даних:
    - Аналізуйте поведінкові та демографічні дані для виявлення шаблонів та переваг користувачів.
    - Класифікуйте користувачів за групами на основі їх інтересів та поведінки.
3. Розробка моделі рекомендації:
    - Використовуйте методи машинного навчання, такі як колаборативна фільтрація, контент-орієнтована фільтрація або гібридні моделі для створення персоналізованих рекомендацій.
    - Навчіть модель на основі історичних даних покупок та взаємодій користувачів.
4. Імплементація системи рекомендацій:
    - Інтегруйте систему рекомендацій на ваш сайт електронної комерції, щоб вона автоматично відображала персоналізовані пропозиції та продукти.
    - Використовуйте рекомендації для підвищення задоволеності користувачів, збільшення часу перебування на сайті та зростання продажів.
5. Оцінка та оптимізація:
    - Аналізуйте ефективність рекомендаційних систем через метрики такі як конверсія, виручка та задоволеність клієнтів.
    - Постійно удосконалюйте моделі, використовуючи зворотний зв’язок користувачів та оновлення даних.
6. Адаптація до нових трендів:
    - Враховуйте зміни на ринку та нові тренди у вподобаннях користувачів для внесення коректівів у систему рекомендацій.

- Регулярно оновлюйте систему для відповідності поточним вимогам ринку та технологіям.

Такий підхід дозволить створити ефективну систему рекомендацій, яка забезпечить персоналізацію взаємодії з користувачами, збільшить їх задоволеність та лояльність, а також підвищить конверсію та загальний обіг в електронній комерції.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз набору даних

Перед реалізацією запропонованого алгоритму здійснено аналіз набору даних, який використовується для розробки інтелектуального модуля рекомендацій у сфері електронної комерції. Набір даних, залучений до дослідження, включає економічні показники первинної харчової промисловості за країнами та регіонами за період від 1995 до 2015 років.

Датасет містить такі основні атрибути (див. таблицю 1 у додатку):

- **Рік (Year)**: рік, за який проведено заміри.
- **Країна (Country)** та **ISO3**: назва країни та її трьохбуквений код ISO.
- **Континент (Continent)** та **Регіон 1 (Region 1)**: континент та деталізований регіон країни.
- **Маржа GME (markup_GME)**: числовий індикатор, що відображає націнку в промисловості.
- **Alpha, Q, II, VA, LAB, K**: економічні показники, які включають вартість активів, інвестиції, валову додану вартість, робочу силу та капітал.
- **Субсидії (Subsidies)** та **Податки (Taxes)**: вартісні показники, пов'язані з економічними стимулами або обов'язковими платежами.

Під час аналізу виявлено відсутні значення у кількох ключових атрибутах, що потребують уваги перед подальшою обробкою даних. Зокрема, це стосується таких атрибутів, як ISO3, континент, регіон, а також економічні індикатори (див. рисунок 3.1).

Для кожної країни та регіону було обраховано статистичні показники такі, як середнє значення, медіана та стандартне відхилення для маржі GME, що дозволяє зрозуміти розподіл і варіативність даних в межах кожного географічного сегменту (див. таблицю 3.1).

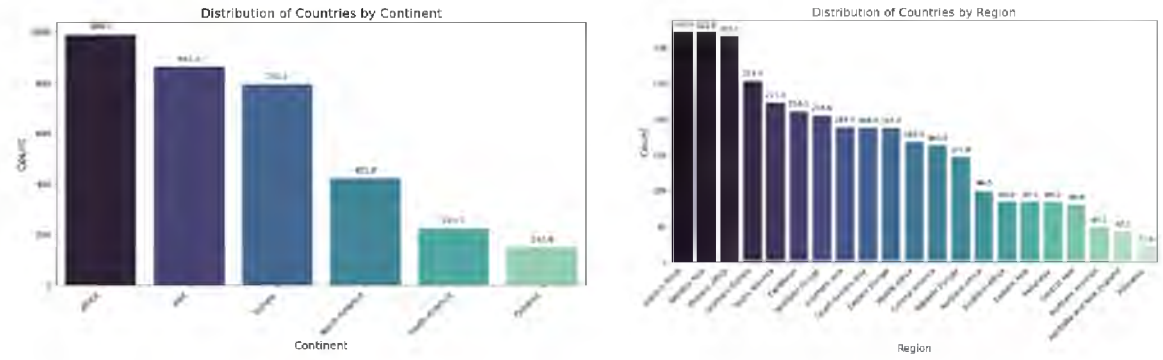
	data type	#missing	%missing	#unique	min	max	first value	second value	third value
year	float64	8200	0.023329	21	1995.000000	2015.000000	1995.000000	1996.000000	1997.000000
country	object	0	0.000000	252	nan	nan	Afghanistan	Afghanistan	Afghanistan
ISO3	object	8200	0.023329	170	nan	nan	AFG	AFG	AFG
continent	object	8200	0.023329	6	nan	nan	Asia	Asia	Asia
Region 1	object	8200	0.023329	21	nan	nan	Southern Asia	Southern Asia	Southern Asia
markup_GME	float64	0	0.000000	3434	0.000000	6.940018	2.334669	2.119724	1.838874
alpha	float64	8200	0.023329	389	0.599701	1.201182	1.027315	1.031692	1.013233
Q	float64	8200	0.023329	3433	4826.317990	1617937378.339795	172971.607477	163649.190338	164919.271031
II	float64	8200	0.023329	3433	3147.013870	599245357.848095	61941.565077	68570.464238	81166.926831
VA	float64	8200	0.023329	3433	-421634.179627	1030571726.000000	111030.042400	95078.726100	83752.344200
LAB	float64	8200	0.023329	3433	28.148732	841104400.000000	15577.660000	13309.470000	11686.020000
K	float64	8200	0.023329	3433	0.100000	52562820.000000	11581.997000	9926.351000	8750.423000
Subsidies	float64	8200	0.023329	3416	-37195830.000000	-0.101966	-469.821600	-426.695100	-376.079100
Taxes	float64	8200	0.023329	3421	0.184345	211042100.000000	2083.829000	1786.181200	1574.846300

Рисунок 3.1 – Аналіз набору даних

Таблиця 3.1 – Описова статистика

	year	markup_GME	alpha	Q	II	VA	LAB	K	Subsidies	Taxes
count	3433,00	3515,00	3433,00	3433,00	3433,00	3433,00	3433,00	3433,00	3433,00	3433,00
mean	2005,25	1,85	0,98	18638804,11	8108677,43	10530126,68	4112984,14	905153,84	-235833,18	369060,32
std	5,96	0,68	0,09	80193832,58	33184218,55	48865740,29	31930236,52	3910952,56	1433823,53	4913510,45
min	1995,00	0,00	0,60	4826,32	3147,01	-421634,18	28,15	0,10	-37195830,00	0,18
25%	2000,00	1,45	0,92	246901,96	114516,99	119674,81	15945,26	2698,66	-21566,00	887,53
50%	2005,00	1,76	0,98	1126333,85	599516,08	455271,31	61719,60	17299,86	-1316,92	4223,47
75%	2010,00	2,17	1,03	8857661,99	4276952,00	4396310,20	988849,20	135275,27	-173,25	24821,88
max	2015,00	6,94	1,20	1617937378,34	599245357,85	1030571726,00	841104400,00	52562820,00	-0,10	211042100,00

Дослідження розподілу країн за континентами та регіонами виконано з метою виявлення географічних особливостей набору даних. Використання кольорової палітри "mako" допомогло візуалізувати розподіл за континентами (рис. 3.2А) та регіонами (рис. 3.2Б), що надає інтуїтивно зрозуміле представлення кількісної присутності країн у різних частинах світу.



А Б

Рисунок 3.2 - Розподілу країн за континентами та регіонами

Оцінка зміни економічних показників (Q, II, VA, LAB, K) та націнки (markup_GME) протягом часу для кожної країни (рис. В.1-5, див додаток В) виявляє тенденції та взаємозв'язки між роками та показниками. Цей аналіз допомагає ідентифікувати закономірності, що можуть впливати на стратегії ведення бізнесу в первинному харчовому секторі.

Використання ящикових діаграм (boxplots) для порівняння економічних показників між регіонами та країнами (рис. 3.3) показує різноманітність розподілу показників, що є критично важливим для глибокого розуміння економічних процесів на різних територіях.

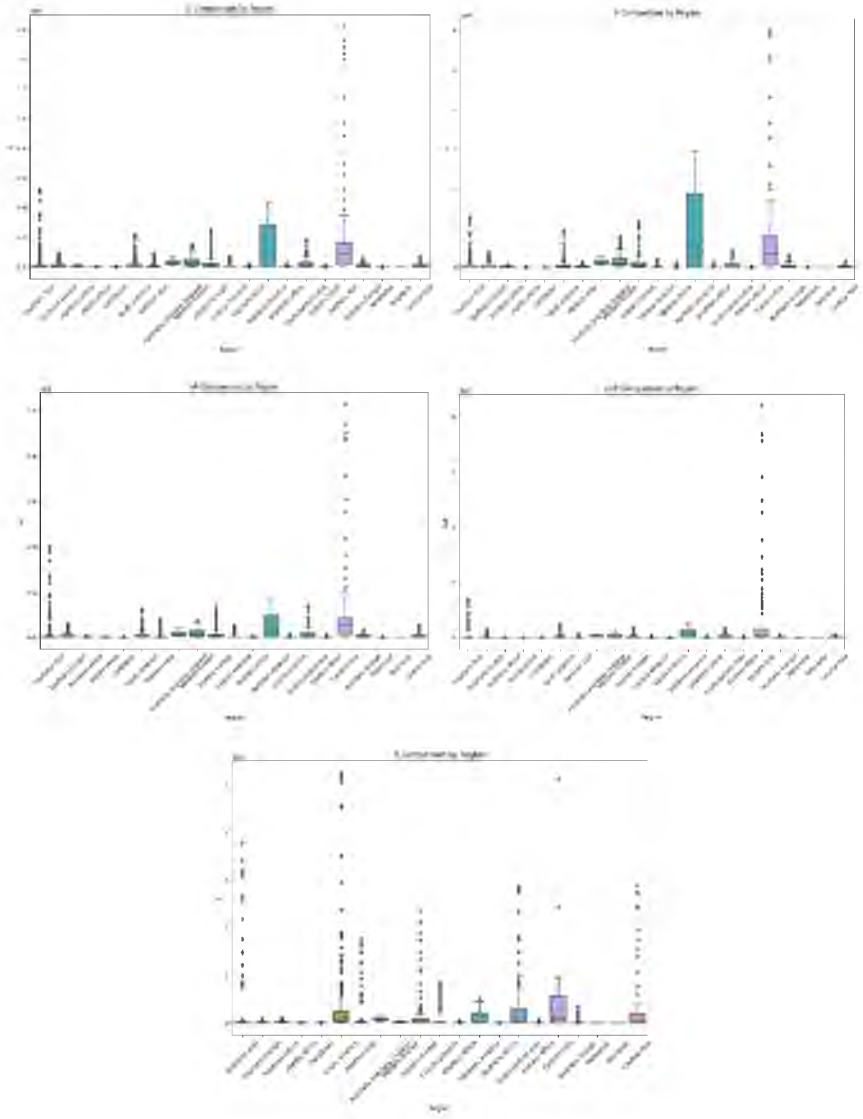


Рисунок 3.3 - boxplots

Детальний аналіз націнки (markup_GME) з використанням лінійних графіків (line plots) та ящикових діаграм дозволяє виявити як загальні тенденції зміни націнок за роками, так і відмінності між регіонами (рис. 3.4 та легенда див. рис.В.1 дод.В). Цей аналіз може вказувати на економічні виклики або можливості в конкретних географічних зонах.

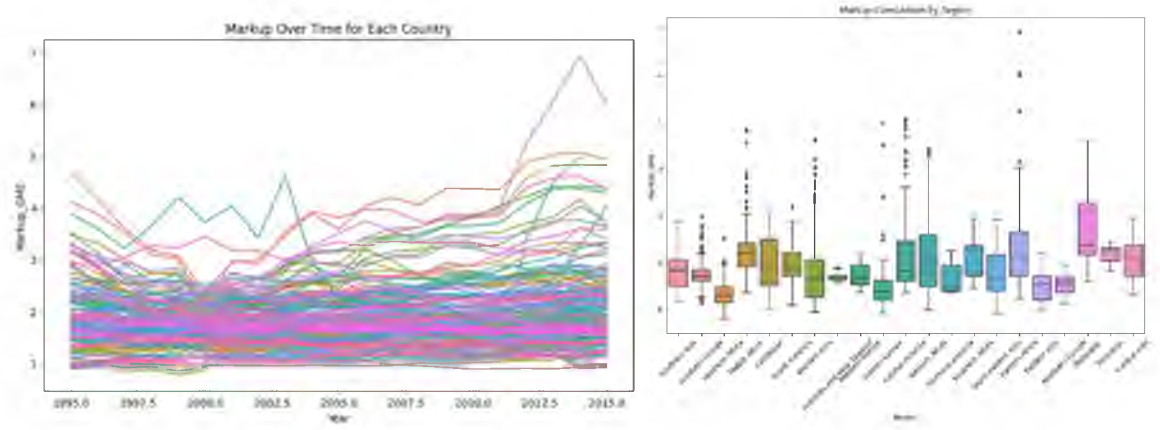


Рисунок 3.4 - Детальний аналіз націнки

Аналіз кореляцій та залежностей між субсидіями, податками та іншими економічними показниками за допомогою pairplots (рис. 3.5) і кореляційних матриць (рис. 3.6) надає цінну інформацію про взаємодію різних економічних факторів.

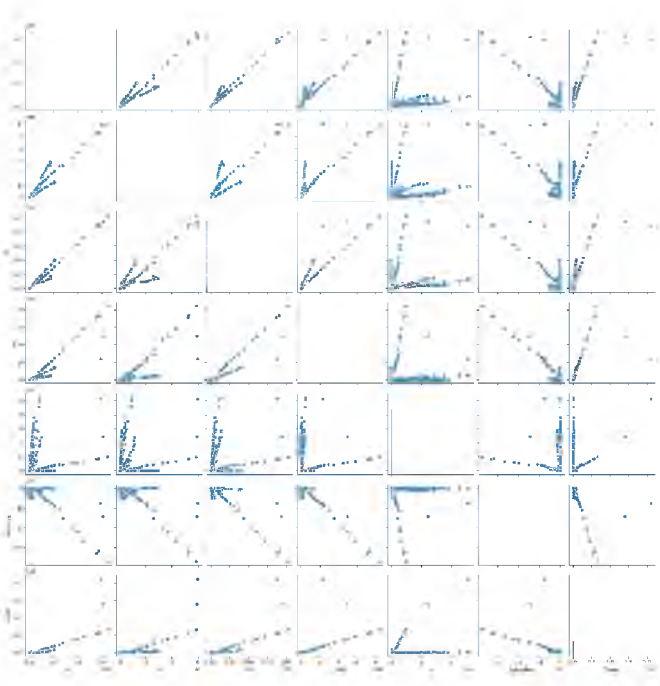


Рисунок 3.5 - pairplots

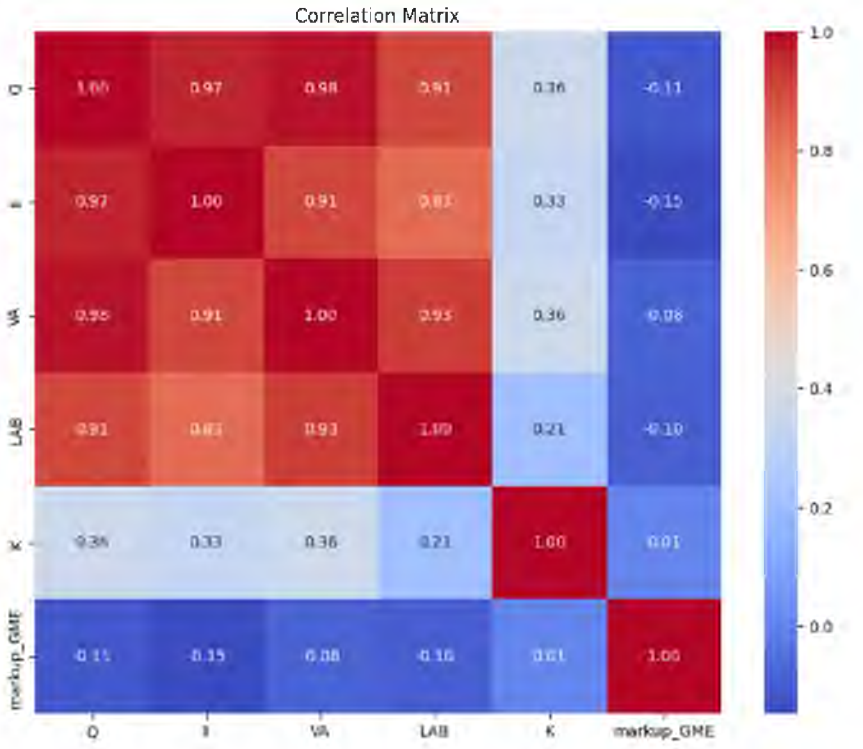


Рисунок 3.5 - Кореляційна матриця

Дослідження наявності викидів у економічних показниках і націнках через ящикові діаграми (рис. 3.6) є важливим для оцінки якості даних та виявлення аномальних значень, які можуть вказувати на помилки введення даних або нестандартні економічні умови.

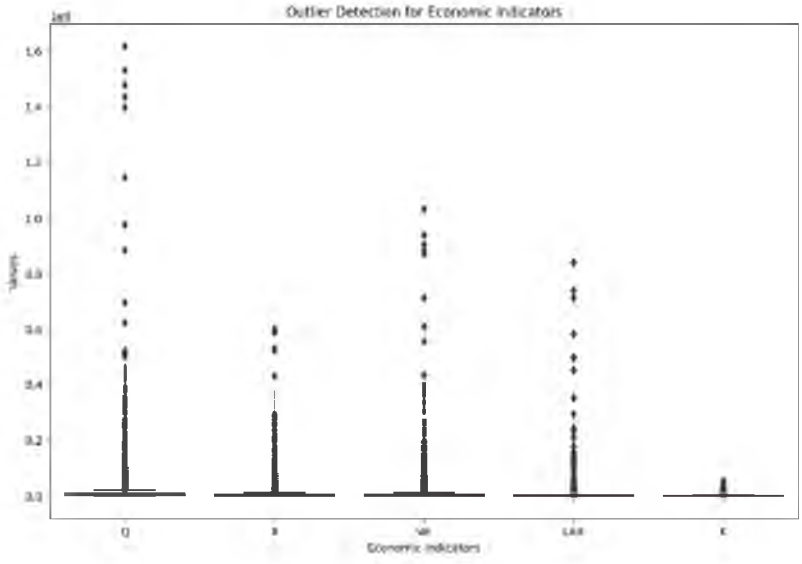


Рисунок 3.6 – Наявність викидів

Цей комплексний аналіз даних не лише виявляє ключові економічні індикатори та їхні тренди, але й допомагає зрозуміти структуру ринку первинних харчових продуктів, що є незамінним для формулювання стратегій розвитку та впровадження ефективних політичних рішень.

3.2 Моделювання

3.2.1 Нормалізація даних

У процесі підготовки даних для подальшого аналізу та моделювання особливе місце займає нормалізація даних. Нормалізація даних — це критичний крок у підготовці даних, який забезпечує єдині масштаби для всіх змінних, тим самим зменшуючи ризик зміщених результатів у моделях машинного навчання, що залежать від величини змінних.

Перш за все, з датасету були вибрані числові змінні (ознаки), такі як економічні індикатори та інші кількісні параметри. Необхідно зазначити, що некатегоріальні змінні, такі як 'year', 'country', 'ISO3', 'continent', 'Region 1', були виключені з аналізу через їх некількісний характер.

Використання модуля **SimpleImputer** зі стратегією 'mean' (середнє значення) дозволило здійснити імпутацію пропущених даних у числових колонках. Цей підхід є стандартним для вирішення проблеми відсутніх значень та дозволяє уникнути втрати цінної інформації.

Після імпутації, дані були нормалізовані за допомогою **StandardScaler**, який трансформує кожен атрибут в датасеті так, що його середнє значення буде рівне 0, а стандартне відхилення — 1. Цей метод є ефективним для вирівнювання масштабів різних змінних, особливо коли вони мають різні одиниці вимірювання або варіабельність.

Завершальний крок полягає у розділенні нормалізованих даних на тренувальну та тестову вибірки з використанням функції **train_test_split** із

бібліотеки **sklearn.model_selection**. Параметр **test_size=0.2** вказує, що 20% даних буде використано для тестування моделі, тоді як решта 80% даних слугуватимуть для тренування. Встановлення параметра **random_state** забезпечує відтворюваність результатів поділу.

3.2.2 Аналіз головних компонент (PCA)

Аналіз головних компонент (PCA) є статистичним методом, який застосовується для зменшення розмірності даних шляхом трансформації до нового набору змінних, які називаються головними компонентами. Ці компоненти вибираються таким чином, що вони послідовно максимізують дисперсію даних, що відображається на них, забезпечуючи тим самим збереження основної структури даних при зниженні їх складності.

У нашому дослідженні PCA використовувався для оптимізації аналізу великих датасетів. Початковий набір даних був трансформований використовуючи PCA, що дозволило ідентифікувати основні взаємозв'язки та патерни між змінними. Результати було візуалізовано на рисунку (див. Рис. 3.7), де показано відсоток збереженої дисперсії в залежності від кількості використаних головних компонент.

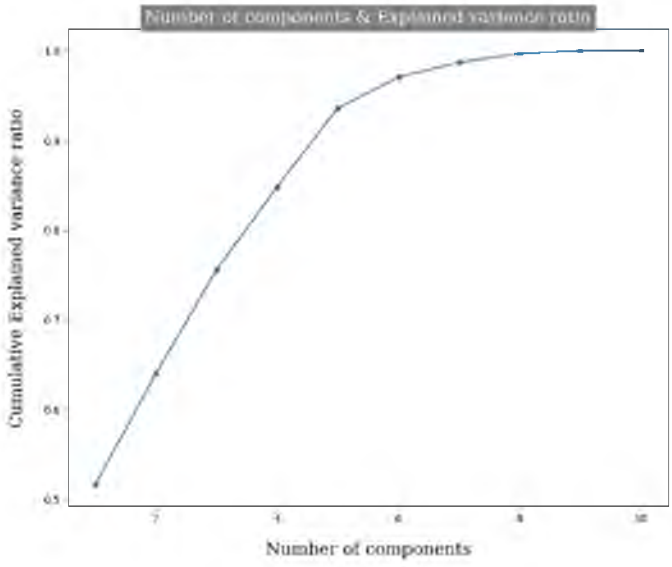


Рисунок 3.7 - Кількість компонентів і пояснений коефіцієнт дисперсії

Для визначення оптимальної кількості головних компонент було проведено аналіз кумулятивної поясненої дисперсії. Як видно з рисунка (див. Рис. 3.8), значення кумулятивної дисперсії наближається до плато, що вказує на те, що додавання додаткових компонент не призводить до значного збільшення поясненої дисперсії.

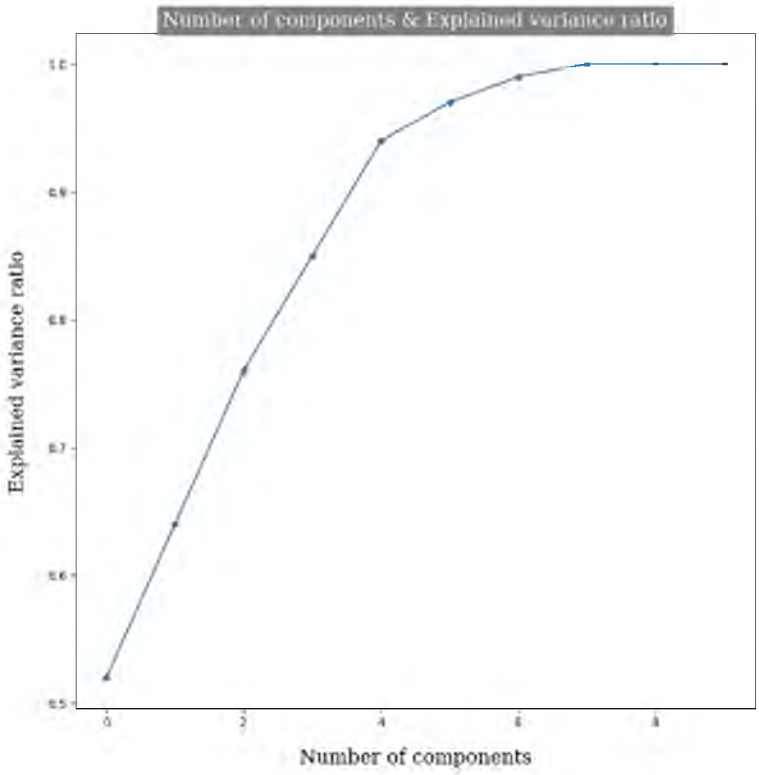


Рисунок 3.8 - Пояснене відношення дисперсії

Для ефективного використання методу аналізу головних компонент (PCA), було проведено оцінювання змін варіативності з різною кількістю компонентів. Специфічно, PCA застосовано з трьома, чотирма та п'ятьма головними компонентами для виявлення, як варіативність відхиляється між цими конфігураціями. Це дозволяє краще зрозуміти баланс між зменшенням розмірності даних та збереженням їх варіативності.

Застосування PCA з трьома компонентами показало, що вони пояснюють [51.61%, 12.33%, 11.61%] варіативності даних. Додавання четвертої та п'ятої компонент збільшило пояснену варіативність на [9.26%] і

[8.80%] відповідно. Це демонструє, що значуще збільшення пояснювальної спроможності PCA досягається за включенням трьох компонент, тоді як додаткові компоненти пропонують порівняно менше внеску.

Scree Plot, що візуалізує власні значення для кожної з головних компонент, надає важливу інформацію про те, коли додавання нових компонент призводить до зменшення приросту пояснювальної сили моделі. Графік показує, що значущий приріст варіативності спостерігається при переході від однієї до трьох компонент, але потім значення власних чисел різко знижуються, сигналізуючи про доцільність обмеження кількості компонент до трьох для оптимального аналізу (див. Рис. 3.9).

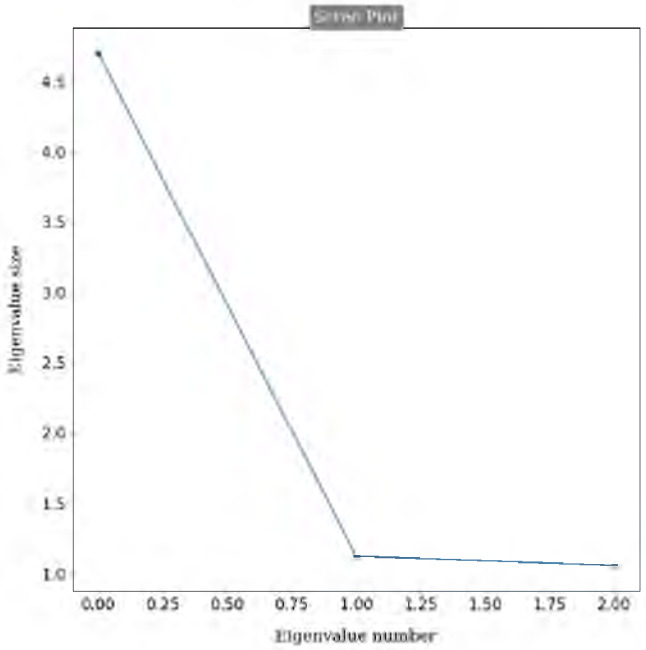


Рисунок 3.9 - Scree Plot

3.2.3 Оптимізація кластеризації з використанням PCA

Кластеризація є процесом у машинному навчанні та аналізі даних, який включає угруповання подібних точок даних на основі певних ознак або характеристик. Метою кластеризації є розподіл набору даних на підмножини, або кластери, таким чином, щоб точки даних усередині одного кластеру були

більш схожими одна на одну, ніж на ті, що знаходяться в інших кластерах. Такий підхід сприяє виявленню закономірностей, структур або природних групувань у даних, полегшуючи краще розуміння та аналіз складних наборів даних. Використовуються різні алгоритми кластеризації, такі як k-середніх, ієрархічна кластеризація та DBSCAN, для досягнення цього групування на основі різних критеріїв.

Застосування методу головних компонент (PCA) до набору даних із трьома компонентами призводить до трансформації набору даних із ознаками, позначеними як "feature1", "feature2" та "feature3". Статистика набору даних вказує на характеристики трансформованих ознак (див. Таблицю 1).

Таблиця 3.2 - Статистика набору даних після застосування PCA

	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
feature1	2812.0	0.00	2.17	-0.83	-0.48	-0.39	-0.18	51.38
feature2	2812.0	0.00	1.06	-2.16	-0.75	-0.07	0.62	5.86
feature3	2812.0	0.00	1.03	-3.95	-0.66	-0.05	0.66	3.96

Для визначення оптимальної кількості кластерів була створена серія оцінок результатів кластеризації для кількостей кластерів від 3 до 19. Підсумкова таблиця оцінок (див. Таблицю 3.3) надає інформацію про показники, що оцінюють кластери на основі когезії, розділення та компактності.

Таблиця 3.3 - Оцінка результатів кластеризації

metric	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
Silhouette score	0.353	0.143	0.289	0.299	0.310	0.314	0.734
Calinski Harabasz score	2119.259	75.315	2035.050	2088.014	2100.828	2141.340	2280.909
Davies Bouldin	0.794	0.103	0.528	0.802	0.820	0.841	0.877

На основі аналізу цих метрик було визначено, що оптимальним варіантом є кластеризація з 3 кластерами. Підтвердження цьому надає високий рівень показника Silhouette (0.734), Calinski Harabasz (2100.044) та Davies Bouldin (0.528), що вказує на гарну якість кластеризації (див. Рисунок 3.10).

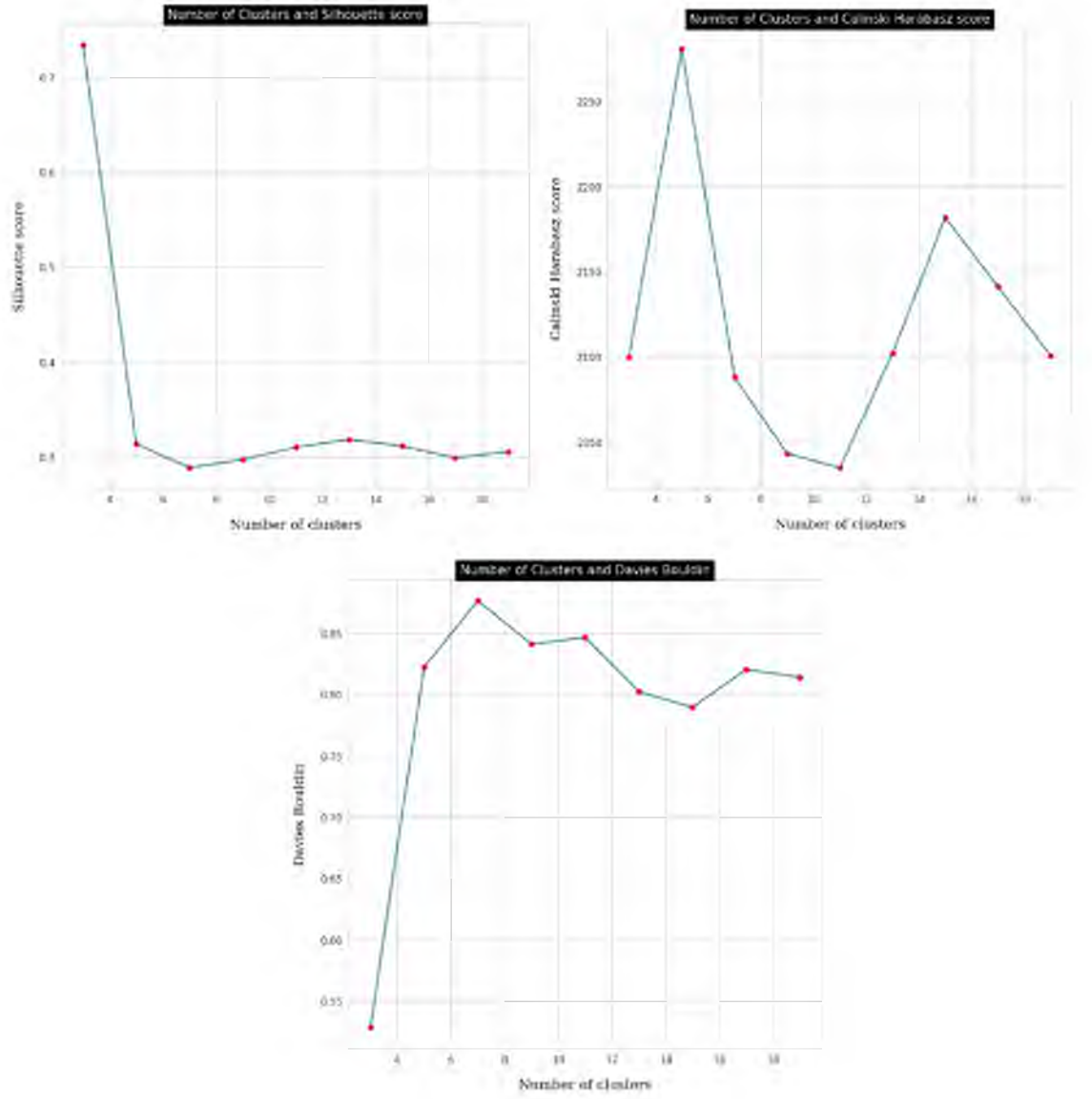


Рисунок 3.10 - Зв’язок кількості кластерів з метриками оцінки

На рисунку 3.10 представлено графік зв'язку кількості кластерів з Silhouette score, Calinski Harabasz score та Davies Bouldin.

Виділена точка на графіку (елемент згину), визначена за допомогою методу KneeLocator, вказує на 5 кластерів як оптимальне число для кластеризації, однак на основі порівняння показників було вирішено, що 3-кластерний розподіл є кращим. Це демонструє, що, незважаючи на розгляд меншої кількості кластерів, якість кластеризації вища, що робить рішення про 3 кластери більш виправданим на основі оцінювальних метрик.

В контексті аналізу кластеризації виявлено, що рішення на основі п'яти кластерів, рекомендоване методом ліктя k-means, показує нижчу якість кластеризації порівняно з трьохкластерною моделлю. Підтвердженням цього служать метрики оцінки кластеризації: коефіцієнт силуета (Silhouette score) становить 0.3137, показник Calinski-Harabasz (Calinski Harabasz score) — 2280.91, та індекс Девіса-Болдуїна (Davies Bouldin score) — 0.8225. Ці показники вказують на меншу когезію та більшу розрізненість кластерів у випадку п'яти кластерів, в порівнянні з моделлю, що використовує три кластери, де зазначені метрики виявилися вищими. Таким чином, на підставі аналітичної оцінки, трьохкластерна модель демонструє вищу якість угруповання та, відповідно, вважається більш переважною для даного набору даних.

3.3 Візуалізація кластеризації та аналіз головних компонент

В рамках аналізу кластеризації на основі методу головних компонент (PCA) було проведено розподіл об'єктів на три кластери. Застосовуючи PCA, визначено три головні компоненти даних, які представлені як 'feature1', 'feature2', та 'feature3'. Це забезпечило зниження розмірності без значної втрати інформації. Результати кластеризації представлені у вигляді таблиці, де кожен об'єкт має присвоєний кластер (Label) (див. Таблицю 3.4).

Таблицю 3.4 - Результати кластеризації

	feature1	feature2	feature3	Label
0	-0.287223	0.669641	1.948777	0
1	-0.389691	0.449140	0.113761	0
2	0.020784	-0.219476	-0.628232	0
3	-0.584256	-0.243742	-0.938992	0
4	-0.529695	-0.032530	-1.530217	0
...	...	...	...	...
2807	-0.440420	-1.048868	0.009167	0

	feature1	feature2	feature3	Label
2808	-0.468473	-0.221972	0.056182	0
2809	-0.353852	0.846739	-1.573552	0
2810	-0.392388	-0.620174	1.159123	0
2811	-0.635796	-0.241695	-1.019932	0

Для візуалізації розподілу кластерів використано розсіяні діаграми. На Рисунку 3.11 зображено взаємозв'язок між 'feature1' та 'feature2' з виділеними центроїдами кластерів, позначеними червоним кольором. Аналогічно, Рисунки 3.12 та 3.13 ілюструють відносини між 'feature1' та 'feature3', та 'feature2' та 'feature3' відповідно. Кожен рисунок надає уявлення про розподіл та розміщення кластерів у багатовимірному просторі.

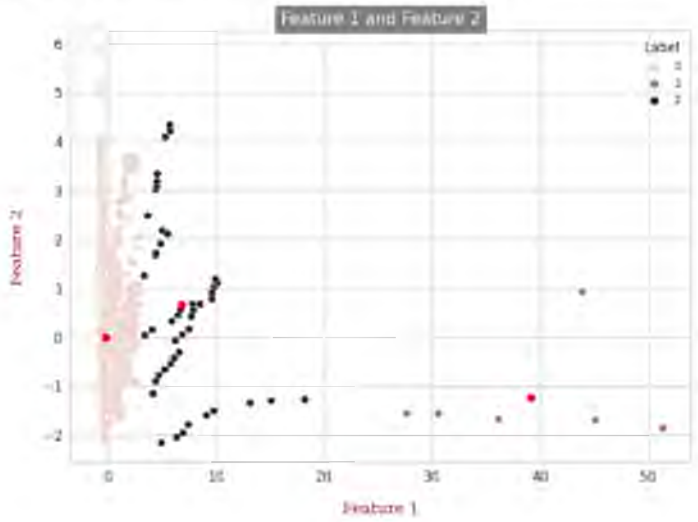


Рисунок 3.11 - Взаємозв'язок між 'feature1' та 'feature2'

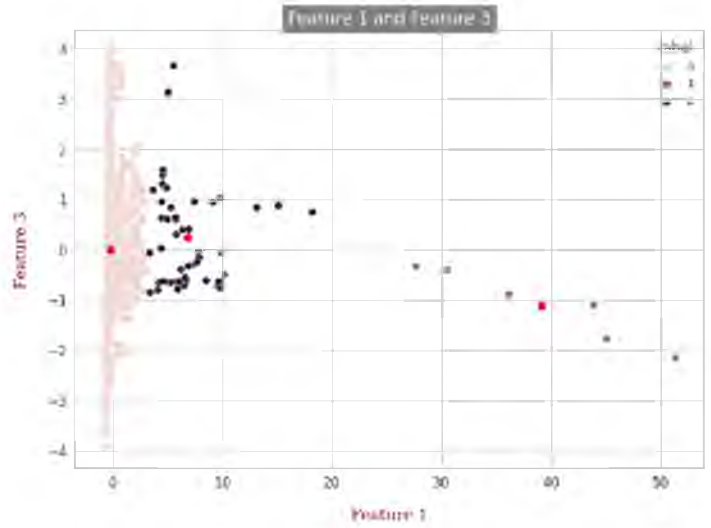


Рисунок 3.12 - Взаємозв'язок між 'feature1' та 'feature3'

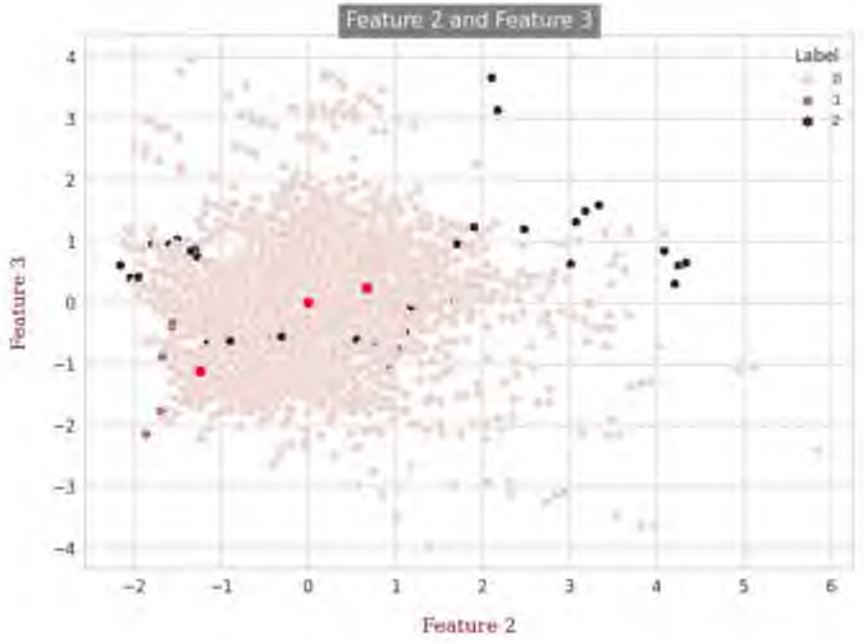


Рисунок 3.13 - Взаємозв'язок між 'feature2' та 'feature3'

Аналізуючи кластеризацію на навчальному наборі даних, використовуючи PCA, було досягнуто високих показників оцінки якості кластеризації, таких як силуетний коефіцієнт (0.9592), показник Calinski-Harabasz (791.97), та індекс Davies-Bouldin (0.0238) (див. Таблицю 3.5).

Таблицю 3.5 - Результати оцінки якості кластеризації

	count	mean	std	min	25%	50%	75%	max
feature1	703.0	-1.010729e-17	2.781244	-0.713535	-0.556857	-0.483441	-0.192552	51.607863
feature2	703.0	-1.516094e-17	1.232860	-16.404724	-0.215790	0.053620	0.281350	17.818928
feature3	703.0	-2.526823e-18	1.066404	-6.858568	-0.591567	0.001271	0.586172	9.858239

Кореляційний аналіз даних (рис. 3.14), представлений у вигляді матриці кореляцій (див. Таблицю Кореляції), виявив різноманітні ступені зв'язків між числовими змінними в наборі даних. Спостерігається, що показник 'Q' має високу позитивну кореляцію з показниками 'II' (r = 0.966) та 'VA' (r = 0.985), та від'ємну кореляцію з 'Subsidies' (r = -0.767), що може свідчити про значну взаємозалежність цих економічних індикаторів.

Аналіз розподілу за мітками кластеризації виявив, що існує позитивний зв'язок між роком і кластером (r = 0.097), що може вказувати на тенденцію зміни характеристик даних у часі. Водночас 'markup_GME' має незначний від'ємний зв'язок із присвоєнням кластерів (r = -0.039), що може вказувати на менш виражену залежність між цією змінною та розподілом на кластери.

	year	markup_GME	alpha	Q	II	VA	\
year	1.000000	0.103738	-0.069204	0.107212	0.108731	0.102109	
markup_GME	0.103738	1.000000	0.095119	-0.112215	-0.146814	-0.084456	
alpha	-0.069204	0.095119	1.000000	-0.025592	-0.011447	-0.034226	
Q	0.107212	-0.112215	-0.025592	1.000000	0.966564	0.984722	
II	0.108731	-0.146814	-0.011447	0.966564	1.000000	0.907144	
VA	0.102109	-0.084456	-0.034226	0.984722	0.907144	1.000000	
LAB	0.061988	-0.097336	-0.048983	0.906937	0.829082	0.925358	
K	0.112161	0.007960	-0.008913	0.355696	0.329433	0.360020	
Subsidies	-0.069750	0.127406	0.026744	-0.767293	-0.707192	-0.778961	
Taxes	0.055573	-0.059989	-0.052952	0.764315	0.716720	0.767604	
Label	0.096812	-0.039208	-0.030147	0.562082	0.577520	0.530247	

	LAB	K	Subsidies	Taxes	Label
year	0.061988	0.112161	-0.069750	0.055573	0.096812
markup_GME	-0.097336	0.007960	0.127406	-0.059989	-0.039208
alpha	-0.048983	-0.008913	0.026744	-0.052952	-0.030147
Q	0.906937	0.355696	-0.767293	0.764315	0.562082
II	0.829082	0.329433	-0.707192	0.716720	0.577520
VA	0.925358	0.360020	-0.778961	0.767604	0.530247
LAB	1.000000	0.205765	-0.901828	0.670318	0.382730
K	0.205765	1.000000	-0.134253	0.275362	0.628125
Subsidies	-0.901828	-0.134253	1.000000	-0.570584	-0.237406
Taxes	0.670318	0.275362	-0.570584	1.000000	0.227265
Label	0.382730	0.628125	-0.237406	0.227265	1.000000

Рисунок 3.14 – Кореляційна матриця

Для візуального представлення залежностей між вибраними характеристиками були створені розсіяні діаграми. На рисунку 3.15 Q та II відображено залежність між загальним обсягом продукції (Q) та інвестиціями (II) з розміткою за кластерами, що показує як розподіляються економічні

показники по різних кластерах. Ці візуалізації можуть бути корисними для ідентифікації груп країн зі схожими економічними профілями або для виявлення викидів та аномалій.

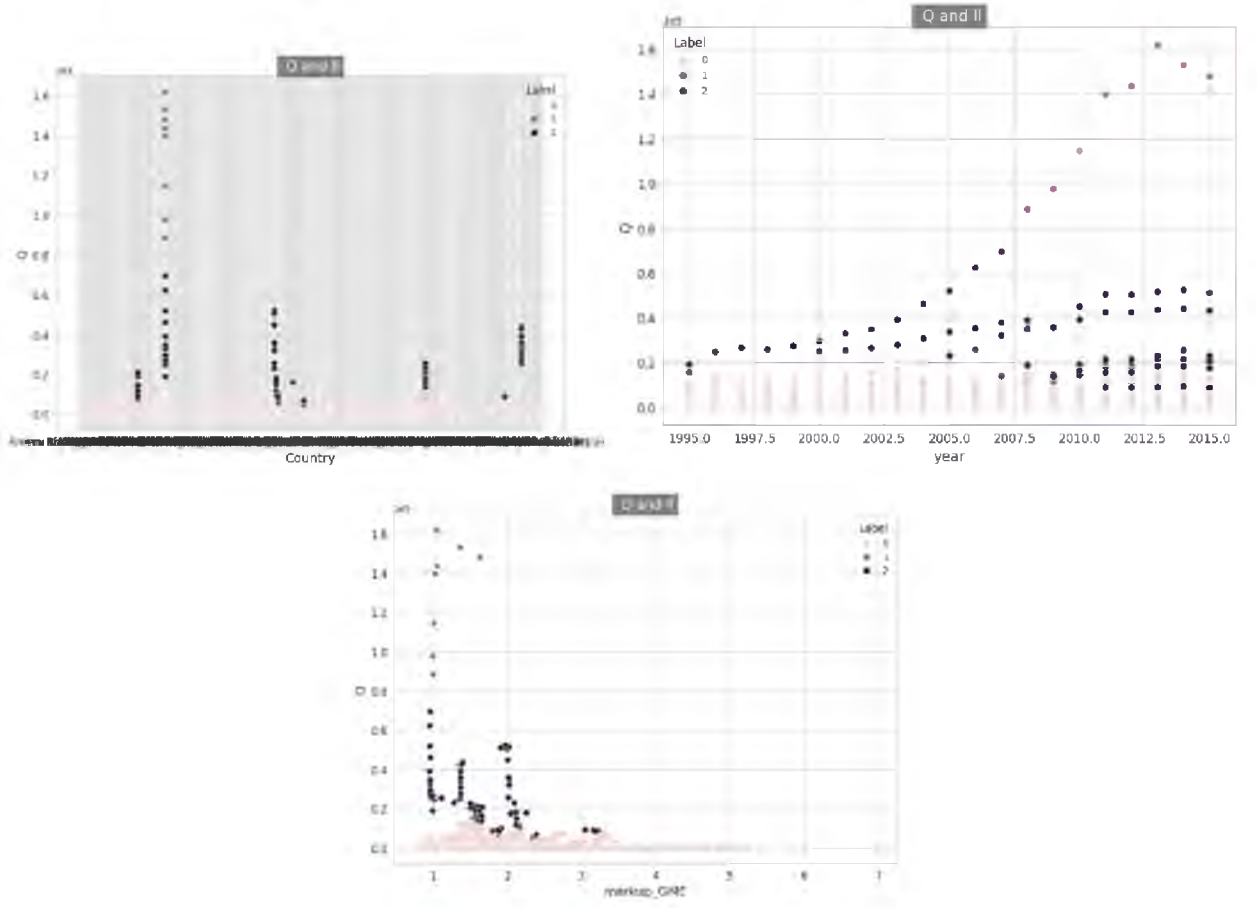


Рисунок 3.15 – Візуалізація кластерів

Графік розподілу кластерів (Рисунок 3.16) демонструє кількісні характеристики обраної моделі кластеризації, де кожен кластер позначено унікальним кольором для зручності сприйняття. Особливо зазначається, що кластер, представлений кольором 'cadetblue', має найбільшу кількість об'єктів, що може свідчити про переважання певного типу економічних характеристик серед даних.

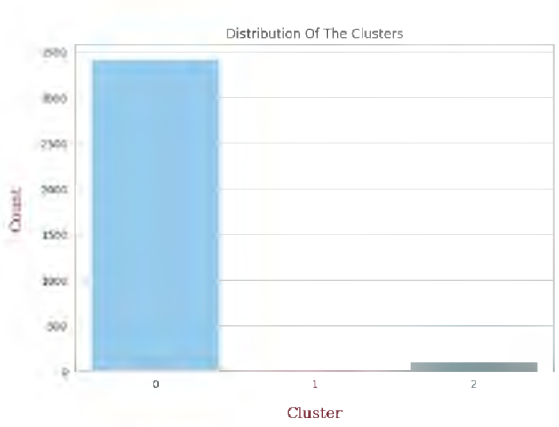


Рисунок 3.16 – Графік розподілу кластерів

Загалом отримані результати демонструють значущість кластеризації у визначенні закономірностей та структури економічних даних, відкриваючи нові можливості для глибшого аналізу та інтерпретації кластерів.

ВИСНОВКИ

У рамках аналізу предметної області електронної комерції та рекомендаційних систем стало очевидним, що персоналізація користувацького досвіду відіграє ключову роль у сучасній цифровій економіці. Зростання конкуренції та зміни у споживчих уподобаннях вимагають від компаній постійного удосконалення та впровадження нових технологій для забезпечення високоякісного обслуговування клієнтів. Огляд існуючих досліджень підкреслив різноманітність підходів та методів у розробці рекомендаційних систем, відмічаючи як їхні переваги, так і недоліки.

Отже, незважаючи на значний прогрес у цій галузі, є певний потенціал для подальшого вдосконалення та інновацій. Зокрема, важливими напрямками є розвиток гібридних систем, які поєднують в собі переваги різних методів, а також дослідження можливостей використання новітніх технологій, таких як штучний інтелект та аналіз великих даних, для покращення точності та ефективності рекомендаційних систем. Відповідно, подальша робота у цьому напрямку може значно підвищити конкурентоспроможність підприємств у сфері електронної комерції та забезпечити задоволення та лояльність клієнтів.

У даній роботі розглянуто архітектуру рекомендаційного модуля для персоналізації користувацького досвіду в електронній комерції. Ця архітектура включає інтерфейс користувача, серверний бекенд та базу даних, кожен з яких відіграє важливу роль у забезпеченні ефективності та точності рекомендаційних систем. Взаємодія між компонентами забезпечує плавний потік даних та їх обробку з метою надання користувачам індивідуалізованих пропозицій, що сприяє підвищенню задоволення користувачів та покращенню результативності електронної комерції.

Розглянуто два основних методи машинного навчання для рекомендаційних систем: аналіз головних компонент (PCA) та алгоритм кластеризації k-середніх. Обидва ці методи є ефективними інструментами для

підвищення ефективності та точності рекомендацій, а також для покращення користувацького досвіду в електронній комерції. Використання PCA дозволяє зменшити розмірність даних та виявити ключові фактори, що впливають на поведінку користувачів, тоді як алгоритм кластеризації k-середніх дозволяє групувати користувачів або товари за подібністю, що сприяє персоналізації та підвищенню задоволення клієнтів. Враховуючи унікальні особливості кожного методу, їхнє використання у поєднанні може забезпечити більшу ефективність та точність рекомендаційних систем у сфері електронної комерції.

Розроблено алгоритм рекомендації для персоналізації користувацького досвіду в електронній комерції. Цей алгоритм включає кроки зі збору, аналізу та обробки даних, розробки моделей рекомендацій, їх імплементації на сайті електронної комерції, а також оцінки та оптимізації їх ефективності. Завдяки такому підходу компанії можуть покращити взаємодію з користувачами, забезпечити їм персоналізовані рекомендації, що підвищить їхню задоволеність та лояльність, а також сприятиме збільшенню обороту та конверсії у сфері електронної комерції.

Аналіз кластеризації на основі методу головних компонент та використання PCA виявив високі показники оцінки якості кластеризації на навчальному наборі даних. Силуетний коефіцієнт становить 0.9592, що свідчить про хорошу компактність кластерів, та великий відстані між ними. Показник Calinski-Harabasz склав 791.97, що вказує на чітку роздільну здатність між кластерами та їхню однорідність. Індекс Davies-Bouldin набув значення 0.0238, що свідчить про мінімальну взаємну схожість кластерів. Такі високі показники підтверджують ефективність використання PCA для зниження розмірності даних та кращого розподілу об'єктів за групами, що відкриває нові можливості для дослідження та аналізу великих обсягів інформації.

Крім того, кореляційний аналіз показав різноманітні ступені зв'язків між числовими змінними в наборі даних. Висока позитивна кореляція показника 'Q' з показниками 'II' та 'VA' підтверджує взаємозалежність між цими економічними показниками. Зв'язок між роком та кластером, який виявився позитивним, може вказувати на тенденцію зміни характеристик даних у часі. Ці висновки створюють базу для подальшого вивчення та розуміння динаміки економічних процесів, що лежать в основі розподілу об'єктів за кластерами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Xiao, B., Aïmeur, E., & Fernandez, J. M. (2003). PCFinder: An intelligent product recommendation agent for e-commerce. Retrieved from https://www.researchgate.net/publication/4021831_PCFinder_An_intelligent_product_recommendation_agent_for_e-commerce
2. Zenebe, A., Ozok, A., & Norcio, A. F. (2005). Personalized recommender systems in e-commerce and m-commerce: a comparative study. Retrieved from https://www.researchgate.net/publication/228364913_Personalized_recommender_systems_in_e-commerce_and_m-commerce_A_comparative_study
3. Abbattista, F., Degemmis, M., Licchelli, O., et al. (2002). Improving the usability of an e-commerce web site through personalization. Retrieved from https://www.academia.edu/30838088/Improving_the_usability_of_an_e-commerce_web_site_through_personalization
4. Agarwal, P., Vempati, S., & Borar, S. (2018). Personalizing similar product recommendations in fashion e-commerce. arXiv preprint arXiv:1806.11371. Retrieved from https://arxiv.org/abs/1806.11371
5. Deepak, G., & Kasaraneni, D. (2019). OntoCommerce: an ontology focused semantic framework for personalised product recommendation for user targeted e-commerce. International Journal of Computer Aided Engineering and Technology, 11(6), 725-738. doi:10.1504/IJCAET.2019.100445
6. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.
7. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні

науки» спеціальності 122 «Комп’ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

ДОДАТОК А

КОД ДЛЯ РЕАЛІЗАЦІЇ

```
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:03:00.457636Z","iopub.execute_input":"2024-02-28T06:03:00.458036Z","iopub.status.idle":"2024-02-28T06:03:17.002921Z","shell.execute_reply.started":"2024-02-28T06:03:00.458005Z","shell.execute_reply":"2024-02-28T06:03:17.001587Z"}}
!pip install kneed
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:03:20.73124Z","iopub.execute_input":"2024-02-28T06:03:20.731693Z","iopub.status.idle":"2024-02-28T06:03:37.50811Z","shell.execute_reply.started":"2024-02-28T06:03:20.731655Z","shell.execute_reply":"2024-02-28T06:03:37.506733Z"}}
!pip install basemap
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:03:37.510369Z","iopub.execute_input":"2024-02-28T06:03:37.510757Z","iopub.status.idle":"2024-02-28T06:03:53.210987Z","shell.execute_reply.started":"2024-02-28T06:03:37.510723Z","shell.execute_reply":"2024-02-28T06:03:53.209909Z"}}
!pip install pycountry
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:04:25.822362Z","iopub.execute_input":"2024-02-28T06:04:25.822891Z","iopub.status.idle":"2024-02-28T06:04:42.747539Z","shell.execute_reply.started":"2024-02-28T06:04:25.822853Z","shell.execute_reply":"2024-02-28T06:04:42.746475Z"}}
!pip install pycountry_convert
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:04:51.938525Z","iopub.execute_input":"2024-02-28T06:04:51.939054Z","iopub.status.idle":"2024-02-28T06:05:06.881509Z","shell.execute_reply.started":"2024-02-28T06:04:51.939013Z","shell.execute_reply":"2024-02-28T06:05:06.880306Z"}}
!pip install plotly
# %% [markdown]
# # <div style="color:yellow;display:inline-block;border-radius:5px;background-color:#007BA7;font-family:Nexa;overflow:hidden"><p
style="padding:15px;color:yellow;overflow:hidden;font-size:95%;letter-spacing:0.5px;margin:0"><b> </b>Import Libraries</p></div>
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:05:12.906336Z","iopub.execute_input":"2024-02-28T06:05:12.906803Z","iopub.status.idle":"2024-02-28T06:05:15.084244Z","shell.execute_reply.started":"2024-02-28T06:05:12.906765Z","shell.execute_reply":"2024-02-28T06:05:15.083072Z"}}
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns
import pycountry_convert as pc
import pycountry
from mpl_toolkits.basemap import Basemap
from geopy.geocoders import Nominatim
import warnings
from warnings import filterwarnings
warnings.filterwarnings('ignore')

# %% [markdown]
# # <div style="color:yellow;display:inline-block;border-radius:5px;background-color:#007BA7;font-family:Nexa;overflow:hidden"><p
style="padding:15px;color:yellow;overflow:hidden;font-size:95%;letter-spacing:0.5px;margin:0"><b> </b>Load Dataset</p></div>
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:05:21.938325Z","iopub.execute_input":"2024-02-28T06:05:21.938984Z","iopub.status.idle":"2024-02-28T06:05:23.918354Z","shell.execute_reply.started":"2024-02-28T06:05:21.938948Z","shell.execute_reply":"2024-02-28T06:05:23.917125Z"}}
# load dataset
df = pd.read_excel('/kaggle/input/global-markup-estimates-primary-foods-industry/Global Markup Estimates for the Primary Foods Industry/Global Markups Primary Food.xlsx')

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:05:27.684226Z","iopub.execute_input":"2024-02-28T06:05:27.685539Z","iopub.status.idle":"2024-02-28T06:05:27.809648Z","shell.execute_reply.started":"2024-02-28T06:05:27.685488Z","shell.execute_reply":"2024-02-28T06:05:27.808682Z"}}
df.head().style.set_properties(**{"background-color": "#457B9D","color":"#A8DADC","border": "1.5px solid Yellow"})
# %% [markdown]
# # <div style="color:yellow;display:inline-block;border-radius:5px;background-color:#007BA7;font-family:Nexa;overflow:hidden"><p
style="padding:15px;color:yellow;overflow:hidden;font-size:95%;letter-spacing:0.5px;margin:0"><b> </b>Summary Table</p></div>
#
# The summary table provides information on the percentage of missing values, unique values, as well as the minimum and maximum values for each variable. Additionally, examining the first three values in each column can offer initial insights into the dataset as a whole.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:05:44.521403Z","iopub.execute_input":"2024-02-28T06:05:44.522087Z","iopub.status.idle":"2024-02-28T06:05:44.531509Z","shell.execute_reply.started":"2024-02-28T06:05:44.522051Z","shell.execute_reply":"2024-02-28T06:05:44.530326Z"}}
# summary table function
def summary(df):
    print(f'data shape: {df.shape}')
    summ = pd.DataFrame(df.dtypes, columns=['data type'])
    summ['#missing'] = df.isnull().sum().values * 100
    summ['%missing'] = df.isnull().sum().values / len(df)
    summ['#unique'] = df.nunique().values
    desc = pd.DataFrame(df.describe(include='all').transpose())
    summ['min'] = desc['min'].values
    summ['max'] = desc['max'].values
    summ['first value'] = df.loc[0].values
    summ['second value'] = df.loc[1].values
    summ['third value'] = df.loc[2].values
    return summ
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:05:49.824519Z","iopub.execute_input":"2024-02-28T06:05:49.824996Z","iopub.status.idle":"2024-02-28T06:05:49.964002Z","shell.execute_reply.started":"2024-02-28T06:05:49.824963Z","shell.execute_reply":"2024-02-28T06:05:49.962502Z"}}
summary(df).style.background_gradient(cmap='YlOrBr')
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:05:54.35029Z","iopub.execute_input":"2024-02-28T06:05:54.351206Z","iopub.status.idle":"2024-02-28T06:05:54.367184Z","shell.execute_reply.started":"2024-02-28T06:05:54.351166Z","shell.execute_reply":"2024-02-28T06:05:54.365617Z"}}
df.duplicated().sum()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:05:57.772404Z","iopub.execute_input":"2024-02-28T06:05:57.773947Z","iopub.status.idle":"2024-02-28T06:05:57.781058Z","shell.execute_reply.started":"2024-02-28T06:05:57.773864Z","shell.execute_reply":"2024-02-28T06:05:57.779899Z"}}
list(df.columns)
# %% [markdown]
# # <div style="color:yellow;display:inline-block;border-radius:5px;background-color:#007BA7;font-family:Nexa;overflow:hidden"><p
style="padding:15px;color:yellow;overflow:hidden;font-size:95%;letter-spacing:0.5px;margin:0"><b> </b>Descriptive Statistics</p></div>
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:06:03.478334Z","iopub.execute_input":"2024-02-28T06:06:03.478723Z","iopub.status.idle":"2024-02-28T06:06:03.529213Z","shell.execute_reply.started":"2024-02-28T06:06:03.478696Z","shell.execute_reply":"2024-02-28T06:06:03.527985Z"}}
df.describe().style.background_gradient(cmap='inferno')
# %% [markdown]
# <font size="+2" color='#059c99'><b> ✨ Missing Data </b></font>
#
# Objective: Check for missing values and decide on appropriate handling strategies.
#
# ### Explanation:
#
# * The code prints the number of missing values for each column in the dataset.
# * Depending on the extent of missing data, you may need to decide whether to impute missing values, remove rows, or apply other strategies.
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:06:08.854264Z","iopub.execute_input":"2024-02-28T06:06:08.854649Z","iopub.status.idle":"2024-02-28T06:06:08.900365Z","shell.execute_reply.started":"2024-02-28T06:06:08.854621Z","shell.execute_reply":"2024-02-28T06:06:08.899135Z"}}
# Calculate markup_GME statistics by country
country_markup_stats = df.groupby('country')['markup_GME'].agg(['mean', 'median', 'std']).reset_index()
# Calculate markup_GME statistics by region
region_markup_stats = df.groupby('Region 1')['markup_GME'].agg(['mean', 'median', 'std']).reset_index()
# Calculate markup_GME statistics by continent
continent_markup_stats = df.groupby('continent')['markup_GME'].agg(['mean', 'median', 'std']).reset_index()
# Display the results
print("Markup_GME Statistics by Country:")
print(country_markup_stats)
print("\nMarkup_GME Statistics by Region:")
print(region_markup_stats)
print("\nMarkup_GME Statistics by Continent:")
print(continent_markup_stats)

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:06:23.427654Z","iopub.execute_input":"2024-02-28T06:06:23.428104Z","iopub.status.idle":"2024-02-28T06:06:23.45543Z","shell.execute_reply.started":"2024-02-28T06:06:23.428073Z","shell.execute_reply":"2024-02-28T06:06:23.454042Z"}}
# Check for missing values and decide on appropriate handling strategies.
missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values)
# Handle missing values
```

```
# Finally, show the processed dataframe
print("\nProcessed DataFrame:")
print(df.head())
# %% [markdown]
# # <div style="color:yellow;display:inline-block;border-radius:5px;background-color:#007BA7;font-family:Nexa;overflow:hidden"><p
style="padding:15px;color:yellow;overflow:hidden;font-size:95%;letter-spacing:0.5px;margin:0"><b> </b>Exploratory Data Analysis</p></div>
#
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Distribution of Countries by Continent</b></font>
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:06:35.082941Z","iopub.execute_input":"2024-02-28T06:06:35.082941Z","iopub.status.idle":"2024-02-
28T06:06:35.420176Z","shell.execute_reply.started":"2024-02-28T06:06:35.082899Z","shell.execute_reply":"2024-02-28T06:06:35.419234Z"}}
# Set a color palette for the countplot
palette = "mako"
# Countplot of population distribution by continent
plt.figure(figsize=(12, 6))
ax = sns.countplot(x='continent', data=df, order=df['continent'].value_counts().index, palette=palette)
# Display values on top of the bars
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='center', xytext=(0, 10), textcoords='offset points')
# Title and labels
plt.title('Distribution of Countries by Continent', fontsize=16)
plt.xlabel('Continent', fontsize=14)
plt.ylabel('Count', fontsize=14)
# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha="right")
# Remove grid lines
ax.yaxis.grid(False)
# Show the plot
plt.show()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:40:54.326823Z","iopub.execute_input":"2024-02-28T06:40:54.327338Z","iopub.status.idle":"2024-02-
28T06:40:54.873637Z","shell.execute_reply.started":"2024-02-28T06:40:54.327301Z","shell.execute_reply":"2024-02-28T06:40:54.872466Z"}}
# Set a color palette for the countplot
palette = "mako"
# Countplot of population distribution by continent
plt.figure(figsize=(12, 6))
ax = sns.countplot(x='Region 1', data=df, order=df['Region 1'].value_counts().index, palette=palette)
# Display values on top of the bars
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()), ha='center', va='center', xytext=(0, 10), textcoords='offset points')
# Title and labels
plt.title('Distribution of Countries by Region', fontsize=16)
plt.xlabel('Region', fontsize=14)
plt.ylabel('Count', fontsize=14)
# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha="right")
# Remove grid lines
ax.yaxis.grid(False)
# Show the plot
plt.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Temporal Trends</b></font>
#
# Objective: Check how economic indicators (Q, II, VA, LAB, K) and markup change over the years for each country. Identify any patterns or trends in the data over time.
#
# ### Explanation:
#
# * The code generates line plots for each economic indicator over the years, with a separate line for each country.
# * These visualizations help identify trends, patterns, and variations in economic indicators across different countries.
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:41:20.685231Z","iopub.execute_input":"2024-02-28T06:41:20.685742Z","iopub.status.idle":"2024-02-
28T06:41:42.656616Z","shell.execute_reply.started":"2024-02-28T06:41:20.685706Z","shell.execute_reply":"2024-02-28T06:41:42.654451Z"}}
# Check how economic indicators (Q, II, VA, LAB, K) and markup change over the years for each country.
# Create a line plot for economic indicators over the years
economic_indicators = ['Q', 'II', 'VA', 'LAB', 'K']
for indicator in economic_indicators:
    plt.figure(figsize=(10, 6))
    sns.lineplot(x='year', y=indicator, hue='country', data=df)
    plt.title(f'{indicator} Over Time for Each Country')
    plt.xlabel('Year')
    plt.ylabel(indicator)
    plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
    plt.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Regional and Continental Comparison</b></font>
#
# Objective: Explore and compare economic indicators between countries, regions, and continents. Look for variations and similarities in economic performance.
#
# ### Explanation:
#
# * The code creates boxplots to compare economic indicators by region. Each box represents the distribution of values for an economic indicator in a specific region.
# * This analysis allows you to observe how economic indicators vary across regions and continents.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:41:51.708183Z","iopub.execute_input":"2024-02-28T06:41:51.708644Z","iopub.status.idle":"2024-02-
28T06:41:54.753778Z","shell.execute_reply.started":"2024-02-28T06:41:51.708607Z","shell.execute_reply":"2024-02-28T06:41:54.752467Z"}}
# Boxplot for economic indicators by region
for indicator in economic_indicators:
    plt.figure(figsize=(12, 8))
    sns.boxplot(x='Region 1', y=indicator, data=df)
    plt.title(f'{indicator} Comparison by Region')
    plt.xlabel('Region')
    plt.ylabel(indicator)
    plt.xticks(rotation=45)
    plt.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Markup Analysis</b></font>
#
# Objective: Analyze the markup_GME variable to understand how markup changes over time and across countries. Identify any outliers or patterns in markup values.
#
# ### Explanation:
#
# * The code generates line plots for markup_GME over the years, with a separate line for each country.
# * Additionally, boxplots are created to compare markup values by region.
# * This analysis helps in understanding how markup values evolve and if there are any countries or regions with significantly different markup patterns.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:42:05.648728Z","iopub.execute_input":"2024-02-28T06:42:05.649225Z","iopub.status.idle":"2024-02-
28T06:42:10.796274Z","shell.execute_reply.started":"2024-02-28T06:42:05.649192Z","shell.execute_reply":"2024-02-28T06:42:10.795009Z"}}
# Analyze the markup_GME variable to understand how markup changes over time and across countries.
# Line plot for markup over the years
plt.figure(figsize=(10, 6))
sns.lineplot(x='year', y='markup_GME', hue='country', data=df)
plt.title('Markup Over Time for Each Country')
plt.xlabel('Year')
plt.ylabel('Markup_GME')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
# Boxplot for markup by region
plt.figure(figsize=(12, 8))
sns.boxplot(x='Region 1', y='markup  GME', data=df)
```

```
plt.xlabel('Region')
plt.ylabel('Markup_GME')
plt.xticks(rotation=45)
plt.show()

# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Subsidies and Taxes</b></font>
#
# Objective: Analyze the relationship between subsidies, taxes, and other economic indicators. Evaluate how these financial aspects impact economic variables.
#
# ### Explanation:
#
# * The code creates a pairplot for subsidies, taxes, and economic indicators.
# * Pairplots visually represent relationships between variables, helping to identify patterns and correlations.
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:42:26.289204Z","iopub.execute_input":"2024-02-28T06:42:26.289707Z","iopub.status.idle":"2024-02-28T06:45:26.209091Z","shell.execute_reply.started":"2024-02-28T06:42:26.289661Z","shell.execute_reply":"2024-02-28T06:45:26.207759Z"}}
# Pairplot for subsidies, taxes, and economic indicators
subset_cols = economic_indicators + ['Subsidies', 'Taxes']
sns.pairplot(df[subset_cols])
plt.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Correlation Analysis</b></font>
#
# Objective: Check for correlations between different economic indicators and markup. Identify which factors are strongly correlated or inversely correlated.
#
# ### Explanation:
#
# * The code generates a heatmap of the correlation matrix between economic indicators and markup_GME.
# * Strong correlations (positive or negative) are visually represented, providing insights into how variables are related.
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:45:37.608265Z","iopub.execute_input":"2024-02-28T06:45:37.608825Z","iopub.status.idle":"2024-02-28T06:45:38.040542Z","shell.execute_reply.started":"2024-02-28T06:45:37.608786Z","shell.execute_reply":"2024-02-28T06:45:38.039213Z"}}
# Check for correlations between different economic indicators and markup.
# Heatmap for correlation matrix
correlation_matrix = df[economic_indicators + ['markup_GME']].corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Outlier Detection</b></font>
#
# Objective: Look for any outliers in economic indicators or markup values. Investigate potential reasons for outliers.
#
# ### Explanation:
#
# * The code creates boxplots to identify outliers in economic indicators and markup_GME.
# * Outliers may indicate anomalies or extreme values that may require further investigation.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:45:45.481308Z","iopub.execute_input":"2024-02-28T06:45:45.481788Z","iopub.status.idle":"2024-02-28T06:45:46.00101Z","shell.execute_reply.started":"2024-02-28T06:45:45.481756Z","shell.execute_reply":"2024-02-28T06:45:46.000087Z"}}
# Look for any outliers in economic indicators or markup values.
# Boxplot for economic indicators
plt.figure(figsize=(12, 8))
sns.boxplot(data=df[economic_indicators])
plt.title('Outlier Detection for Economic Indicators')
plt.xlabel('Economic Indicators')
plt.ylabel('Values')
plt.show()
# Boxplot for markup
plt.figure(figsize=(8, 6))
sns.boxplot(x='markup_GME', data=df)
plt.title('Outlier Detection for Markup')
plt.xlabel('Markup_GME')
plt.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Country-Specific Analysis</b></font>
#
# Objective: Conduct an analysis for specific countries or regions of interest. Identify factors that contribute to economic performance.
#
# ### Explanation:
#
# * The code selects a specific country (in this case, Afghanistan) and generates line plots for economic indicators over the years.
# * This analysis allows for a focused examination of a particular country's economic trends.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:45:54.284139Z","iopub.execute_input":"2024-02-28T06:45:54.284712Z","iopub.status.idle":"2024-02-28T06:45:54.649221Z","shell.execute_reply.started":"2024-02-28T06:45:54.284665Z","shell.execute_reply":"2024-02-28T06:45:54.647671Z"}}
# Conduct an analysis for specific countries or regions of interest.
# Choose a country for analysis
selected_country = 'Afghanistan'
selected_df = df[df['country'] == selected_country]
# Line plot for economic indicators over the years for the selected country
plt.figure(figsize=(10, 6))
sns.lineplot(x='year', y=economic_indicators[0], label=economic_indicators[0], data=selected_df)
sns.lineplot(x='year', y=economic_indicators[1], label=economic_indicators[1], data=selected_df)
# Repeat for other economic indicators
plt.title(f'Economic Indicators Over Time for {selected_country}')
plt.xlabel('Year')
plt.ylabel('Values')
plt.legend()
plt.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Economic Growth Trend Comparison</b></font>
#
# Objective: Compare the economic growth trend (measured by the variable Q) across continents.
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:46:08.200645Z","iopub.execute_input":"2024-02-28T06:46:08.201136Z","iopub.status.idle":"2024-02-28T06:46:12.536188Z","shell.execute_reply.started":"2024-02-28T06:46:08.201104Z","shell.execute_reply":"2024-02-28T06:46:12.534825Z"}}
# Create a line plot for economic growth (Q) over the years for each continent
plt.figure(figsize=(12, 8))
sns.lineplot(x='year', y='Q', hue='continent', data=df, palette='viridis')
plt.title('Economic Growth Trend Comparison Across Continents')
plt.xlabel('Year')
plt.ylabel('Economic Growth (Q)')
plt.legend(title='Continent', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()

# %% [markdown]
# ### Explanation:
#
# * This analysis visualizes the economic growth trend (variable Q) over the years for each continent using a line plot.
# * The plot allows for a quick comparison of economic growth trajectories across continents.
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Markup vs. Subsidies and Taxes</b></font>
#
# Objective: Explore the relationship between markup_GME and subsidies/taxes using a scatter plot.
#
```

28T06:46:32.353091Z","shell.execute_reply.started":"2024-02-28T06:46:31.675193Z","shell.execute_reply":"2024-02-28T06:46:32.351779Z"}}
```
# Create a scatter plot to explore the relationship between markup and subsidies/taxes
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Subsidies', y='Taxes', hue='markup_GME', data=df, palette='coolwarm', size='markup_GME')
plt.title('Markup vs. Subsidies and Taxes')
plt.xlabel('Subsidies')
plt.ylabel('Taxes')
plt.legend(title='Markup_GME', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()

# %% [markdown]
# ### Explanation:
#
# * This analysis visualizes the relationship between markup_GME and Subsidies/Taxes using a scatter plot.
# * The size and color of the markers indicate the intensity of markup, providing a comprehensive view of the relationships.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:46:47.476564Z","iopub.execute_input":"2024-02-28T06:46:47.476979Z","iopub.status.idle":"2024-02-28T06:46:50.40812Z","shell.execute_reply.started":"2024-02-28T06:46:47.476948Z","shell.execute_reply":"2024-02-28T06:46:50.406901Z"}}
import plotly.express as px
# Choropleth map for a specific year (let's say 1995)
fig = px.choropleth(df,
                    locations="ISO3",
                    color="Q",  # Choose the column you want to visualize
                    hover_name="country",
                    title="Economic Growth (Q) Distribution in 1995",
                    color_continuous_scale=px.colors.sequential.Plasma)
fig.show()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:47:08.769509Z","iopub.execute_input":"2024-02-28T06:47:08.770048Z","iopub.status.idle":"2024-02-28T06:47:08.875617Z","shell.execute_reply.started":"2024-02-28T06:47:08.770008Z","shell.execute_reply":"2024-02-28T06:47:08.873975Z"}}
# Choropleth map for a specific year (let's say 1995)
fig = px.choropleth(df,
                    locations="ISO3",
                    color="II",  # Choose the column you want to visualize
                    hover_name="country",
                    title="Economic Growth (II) Distribution in 1995",
                    color_continuous_scale=px.colors.sequential.Viridis)
fig.show()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:47:20.786207Z","iopub.execute_input":"2024-02-28T06:47:20.786768Z","iopub.status.idle":"2024-02-28T06:47:20.894426Z","shell.execute_reply.started":"2024-02-28T06:47:20.786725Z","shell.execute_reply":"2024-02-28T06:47:20.8933Z"}}
# Choropleth map for a specific year (let's say 1995)
fig = px.choropleth(df,
                    locations="ISO3",
                    color="LAB",  # Choose the column you want to visualize
                    hover_name="country",
                    title="Economic Growth (LAB) Distribution in 1995",
                    color_continuous_scale=px.colors.sequential.Magma)
fig.show()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:47:29.187935Z","iopub.execute_input":"2024-02-28T06:47:29.188403Z","iopub.status.idle":"2024-02-28T06:47:29.289295Z","shell.execute_reply.started":"2024-02-28T06:47:29.18837Z","shell.execute_reply":"2024-02-28T06:47:29.28737Z"}}
# Choropleth map for a specific year (let's say 1995)
fig = px.choropleth(df,
                    locations="ISO3",
                    color="VA",  # Choose the column you want to visualize
                    hover_name="country",
                    title="Economic Growth (VA) Distribution in 1995",
                    color_continuous_scale=px.colors.sequential.Cividis)
fig.show()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:47:42.925166Z","iopub.execute_input":"2024-02-28T06:47:42.925654Z","iopub.status.idle":"2024-02-28T06:47:43.03143Z","shell.execute_reply.started":"2024-02-28T06:47:42.925623Z","shell.execute_reply":"2024-02-28T06:47:43.02992Z"}}
# Choropleth map for a specific year (let's say 1995)
fig = px.choropleth(df,
                    locations="ISO3",
                    color="K",  # Choose the column you want to visualize
                    hover_name="country",
                    title="Economic Growth (K) Distribution in 1995",
                    color_continuous_scale=px.colors.sequential.Inferno)
fig.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Markup Outliers Bubble Chart</b></font>
#
# Objective: Identify and visualize outliers in markup_GME using a bubble chart.
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:47:57.927763Z","iopub.execute_input":"2024-02-28T06:47:57.928317Z","iopub.status.idle":"2024-02-28T06:48:05.970392Z","shell.execute_reply.started":"2024-02-28T06:47:57.928276Z","shell.execute_reply":"2024-02-28T06:48:05.968888Z"}}
# Identify and visualize outliers in markup using a bubble chart
plt.figure(figsize=(12, 8))
sns.scatterplot(x='year', y='markup_GME', hue='country', size='markup_GME', data=df, palette='Set2')
plt.title('Markup Outliers Bubble Chart')
plt.xlabel('Year')
plt.ylabel('Markup_GME')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
# %% [markdown]
# ### Explanation:
#
# * This analysis creates a bubble chart to identify outliers in markup_GME over the years.
# * The size of the bubbles represents the intensity of markup, and the color distinguishes different countries.
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Visualized the countries with the highest and lowest values for each feature using both pie charts</b></font>
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:48:17.763515Z","iopub.execute_input":"2024-02-28T06:48:17.764273Z","iopub.status.idle":"2024-02-28T06:48:20.863902Z","shell.execute_reply.started":"2024-02-28T06:48:17.764219Z","shell.execute_reply":"2024-02-28T06:48:20.862636Z"}}
# List of features
features = ['Q', 'II', 'LAB', 'K']
# Iterate over features
for feature in features:
    # Find countries with the highest and lowest values for the chosen feature
    highest_countries = df.nlargest(4, feature)['country']
    lowest_countries = df.nsmallest(4, feature)['country']
    # Plot pie charts for the highest values
    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    df_high = df[df['country'].isin(highest_countries)]
    plt.pie(df_high[feature], labels=df_high['country'], autopct='%1.1f%%', startangle=90)
    plt.title(f'Top 5 Countries with Highest {feature}')
    # Plot pie charts for the lowest values
    plt.subplot(1, 2, 2)
    df_low = df[df['country'].isin(lowest_countries)]
    plt.pie(df_low[feature], labels=df_low['country'], autopct='%1.1f%%', startangle=90)
    plt.title(f'Top 5 Countries with Lowest {feature}')
    plt.tight_layout()
    plt.show()

# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Exploring Variation in Multiple Features </b></font>
#
```

%% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:49:22.523224Z","iopub.execute_input":"2024-02-28T06:49:22.52377Z","iopub.status.idle":"2024-02-28T06:49:22.933911Z","shell.execute_reply.started":"2024-02-28T06:49:22.523728Z","shell.execute_reply":"2024-02-28T06:49:22.932712Z"}}

```
```

```
features = ['Q', 'II', 'VA', 'LAB', 'K', 'Subsidies', 'Taxes']
# Set up a multi-feature box plot
plt.figure(figsize=(14, 8))
sns.boxplot(data=df[features], palette="Set3")
plt.title('Variation in Multiple Features by Box Plots')
plt.xlabel('Features')
plt.ylabel('Values')
plt.show()
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Assessing Feature Importance: Quantifying Variability through Variance and Standard Deviation</b></font>
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:49:31.640647Z","iopub.execute_input":"2024-02-28T06:49:31.641274Z","iopub.status.idle":"2024-02-28T06:49:32.203763Z","shell.execute_reply.started":"2024-02-28T06:49:31.641242Z","shell.execute_reply":"2024-02-28T06:49:32.202886Z"}}
# List of features to assess
features = ['Q', 'II', 'VA', 'LAB', 'K', 'Subsidies', 'Taxes']
# Calculate variance and standard deviation for each feature
variance_values = df[features].var()
std_deviation_values = df[features].std()
# Print variance and standard deviation values
print("Variance values:")
print(variance_values)
print("\nStandard Deviation values:")
print(std_deviation_values)
# Plotting
plt.figure(figsize=(12, 6))
# Bar plot for Variance
plt.subplot(1, 2, 1)
variance_values.plot(kind='bar', color='skyblue')
plt.title('Variance of Features')
plt.ylabel('Variance')
# Bar plot for Standard Deviation
plt.subplot(1, 2, 2)
std_deviation_values.plot(kind='bar', color='lightcoral')
plt.title('Standard Deviation of Features')
plt.ylabel('Standard Deviation')
plt.tight_layout()
plt.show()
# %% [markdown]
# ### Explanation
#
# This data represents the feature importance assessment using variance and standard deviation. For the feature "Q," the variance is approximately 6.43e+15, and the standard deviation is around 8.02e+07. Similarly, for features "II," "VA," "LAB," "K," "Subsidies," and "Taxes," the corresponding variance and standard deviation values are provided. These statistics quantify the variability in each feature, offering insights into their importance in the analysis.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:49:42.067481Z","iopub.execute_input":"2024-02-28T06:49:42.068046Z","iopub.status.idle":"2024-02-28T06:49:42.087273Z","shell.execute_reply.started":"2024-02-28T06:49:42.068009Z","shell.execute_reply":"2024-02-28T06:49:42.086033Z"}}
# Exclude non-numeric columns from correlation calculation
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
df_numeric = df[numeric_columns]
# Calculate the correlation matrix
corr = df_numeric.corr()
# Print the correlation matrix
print("Correlation Matrix:")
print(corr)
# %% [markdown]
# ### Explanation
#
# This correlation matrix reveals the relationships between different variables. Each cell represents the correlation coefficient between two variables, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). A coefficient close to 0 suggests a weak correlation.
#
# For example, there is a strong positive correlation (0.966) between variables Q and II, while there is a strong negative correlation (-0.902) between LAB and Subsidies. Understanding these correlations helps to identify patterns and dependencies among the variables in the analysis.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:49:48.277669Z","iopub.execute_input":"2024-02-28T06:49:48.278172Z","iopub.status.idle":"2024-02-28T06:49:49.100397Z","shell.execute_reply.started":"2024-02-28T06:49:48.278138Z","shell.execute_reply":"2024-02-28T06:49:49.099224Z"}}
f, ax = plt.subplots(figsize=(15, 15))
plt.title('\n    Correlations between Features', pad=20, size=30)
ax = sns.heatmap(corr, annot=True,
            fmt=".2f",
            vmin=-1, vmax=1,
            center=0, square=True, linewidths=.5,cmap="crest",
            cbar_kws={"shrink": 0.8})
# %% [markdown]
# # <div style="color:yellow;display:inline-block;border-radius:5px;background-color:#007BA7;font-family:Nexa;overflow:hidden"><p
style="padding:15px;color:yellow;overflow:hidden;font-size:95%;letter-spacing:0.5px;margin:0"><b> </b>Modeling </p></div>
#
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ We normalize the data using StandardScaler:</b></font>
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:50:31.407633Z","iopub.execute_input":"2024-02-28T06:50:31.408052Z","iopub.status.idle":"2024-02-28T06:50:31.493236Z","shell.execute_reply.started":"2024-02-28T06:50:31.408022Z","shell.execute_reply":"2024-02-28T06:50:31.491918Z"}}
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.impute import SimpleImputer
import matplotlib.pyplot as plt
import seaborn as sns
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:50:42.478478Z","iopub.execute_input":"2024-02-28T06:50:42.478987Z","iopub.status.idle":"2024-02-28T06:50:42.497217Z","shell.execute_reply.started":"2024-02-28T06:50:42.478947Z","shell.execute_reply":"2024-02-28T06:50:42.495689Z"}}
# Assuming your features are all columns except non-numeric ones like 'year', 'country', 'ISO3', 'continent', 'Region 1'
features = df.select_dtypes(include=['float64', 'int64']).columns
X = df[features]
# Impute missing values
imputer = SimpleImputer(strategy='mean')  # You can use other strategies like 'median', 'most_frequent'
X_imputed = imputer.fit_transform(X)
# Normalize the data using StandardScaler
scaler = StandardScaler()
X_normalized = scaler.fit_transform(X_imputed)
# Split the data into training and testing sets
X_train, X_test = train_test_split(X_normalized, test_size=0.2, random_state=42)

# %% [markdown]
# # <div style="color:yellow;display:inline-block;border-radius:5px;background-color:#007BA7;font-family:Nexa;overflow:hidden"><p
style="padding:15px;color:yellow;overflow:hidden;font-size:95%;letter-spacing:0.5px;margin:0"><b> </b>Principal Component Analysis</p></div>
#
#
# <font size="+1" color='#059c99'><b> ✨ Perform dimensionality reduction using PCA (Principal Component Analysis)</b></font>
#
# Utilize PCA (Principal Component Analysis) for dimensionality reduction. PCA identifies key patterns and reduces the dataset's complexity by transforming variables into a smaller set of uncorrelated components. This aids in retaining essential information while simplifying the data for more efficient analysis and visualization.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:50:53.932914Z","iopub.execute_input":"2024-02-28T06:50:53.933477Z","iopub.status.idle":"2024-02-28T06:50:53.969884Z","shell.execute_reply.started":"2024-02-28T06:50:53.933436Z","shell.execute_reply":"2024-02-28T06:50:53.967983Z"}}
# Perform dimensionality reduction using PCA
pca = PCA()
X_pca = pca.fit_transform(X_train)
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:50:59.690336Z","iopub.execute_input":"2024-02-28T06:50:59.690821Z","iopub.status.idle":"2024-02-28T06:51:00.012828Z","shell.execute_reply.started":"2024-02-28T06:50:59.690788Z","shell.execute_reply":"2024-02-28T06:51:00.011606Z"}}
# Calculate explained variance ratio
explained_ratio = pca.explained_variance_ratio
```

```
plt.figure(figsize=(10, 8))
font1 = {'family': 'serif', 'color': 'black', 'weight': 'normal', 'size': 16}
plt.title('Number of components & Explained variance ratio', backgroundcolor='grey', color='white', fontdict=font1)
ax = sns.lineplot(x=np.arange(1, len(explained_ratio) + 1), y=np.cumsum(explained_ratio))
ax = sns.scatterplot(x=np.arange(1, len(explained_ratio) + 1), y=np.cumsum(explained_ratio))
ax.set_xlabel('Number of components', fontdict=font1, labelpad=16)
ax.set_ylabel('Cumulative Explained variance ratio', fontdict=font1, labelpad=16)
plt.show()
# %% [markdown]
# ### Explanation
#
# Assess the number of components and their cumulative explained variance ratio using PCA. This graphical representation illustrates how much of the dataset's variance is retained as the number of components increases. The plot helps in determining an optimal balance between dimensionality reduction and retaining significant information.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:51:10.620792Z","iopub.execute_input":"2024-02-28T06:51:10.621506Z","iopub.status.idle":"2024-02-28T06:51:10.626495Z","shell.execute_reply.started":"2024-02-28T06:51:10.621471Z","shell.execute_reply":"2024-02-28T06:51:10.625167Z"}}
explained_ratio = np.cumsum(np.round(pca.explained_variance_ratio_,2))
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:51:14.243666Z","iopub.execute_input":"2024-02-28T06:51:14.244124Z","iopub.status.idle":"2024-02-28T06:51:14.625634Z","shell.execute_reply.started":"2024-02-28T06:51:14.244093Z","shell.execute_reply":"2024-02-28T06:51:14.624298Z"}}
#Number of components & Explained variance ratio for dataset without potential outliers
plt.figure(figsize=(10,10))
plt.title('Number of components & Explained variance ratio',backgroundcolor='grey',color='white',fontdict=font1)
ax=sns.lineplot( x=np.arange(0,len(explained_ratio)) , y=explained_ratio)
ax=sns.scatterplot(x=np.arange(0,len(explained_ratio)) , y=explained_ratio)
ax.set_xlabel('Number of components',fontdict=font1,labelpad=16)
ax.set_ylabel('Explained variance ratio',fontdict=font1,labelpad=16)
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Based on the variance ratio deviations, PCA was performed with 3 and 4 components and 5.</b></font>
#
# ### Explanation
#
# PCA was applied with different component configurations, specifically with 3, 4, and 5 components. This exploration aims to assess how the variance ratio deviates across these configurations, providing insights into the trade-off between dimensionality reduction and preserving data variability.
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:51:30.449057Z","iopub.execute_input":"2024-02-28T06:51:30.449506Z","iopub.status.idle":"2024-02-28T06:51:30.524772Z","shell.execute_reply.started":"2024-02-28T06:51:30.449459Z","shell.execute_reply":"2024-02-28T06:51:30.522513Z"}}
for i in range(3,6):
    pca = PCA(n_components=i)
    pca.fit(X_train)
    explained_Variance = pca.explained_variance_ratio
    print(f"For {i} as n_components, Explained Variance equals to {explained_Variance}")
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:51:34.629338Z","iopub.execute_input":"2024-02-28T06:51:34.629794Z","iopub.status.idle":"2024-02-28T06:51:34.656572Z","shell.execute_reply.started":"2024-02-28T06:51:34.629758Z","shell.execute_reply":"2024-02-28T06:51:34.654833Z"}}
pca = PCA(n_components=3)
pca.fit(X_train)
explained_Variance = pca.explained_variance_ratio_
# %% [markdown]
# <font size="+1" color='#059c99'><b> ✨ Scree Plot for entire dataset</b></font>
#
# The Scree Plot visualizes the eigenvalues of principal components in the entire dataset. It helps identify the point at which adding more components provides diminishing returns in explaining the variance. This aids in determining an appropriate number of principal components to retain for efficient dimensionality reduction.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:51:41.730409Z","iopub.execute_input":"2024-02-28T06:51:41.730953Z","iopub.status.idle":"2024-02-28T06:51:42.082008Z","shell.execute_reply.started":"2024-02-28T06:51:41.730911Z","shell.execute_reply":"2024-02-28T06:51:42.080619Z"}}
plt.figure(figsize=(10,10))
plt.title('Scree Plot',backgroundcolor='grey',color='white',fontdict=font1)
plt.plot(pca.explained_variance_, marker='o')
plt.xlabel("Eigenvalue number",fontdict=font1,labelpad=16)
plt.ylabel("Eigenvalue size",fontdict=font1,labelpad=16)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
# %% [markdown]
# ### Explanation
#
# Eigenvalue number refers to the order or position of an eigenvalue in a set. Eigenvalues represent the magnitude of variability captured by each principal component in a dataset. Eigenvalue size indicates the magnitude or strength of the variability explained by a specific eigenvalue. Larger eigenvalues correspond to principal components that capture more significant portions of the overall variance in the data.
# %% [markdown]
# # <div style="color:yellow;display:inline-block;border-radius:5px;background-color:#007BA7;font-family:Nexa;overflow:hidden"><p
style="padding:15px;color:yellow;overflow:hidden;font-size:95%;letter-spacing:0.5px;margin:0"><b> </b>Clustering</p></div>
#
# **Clustering** is a technique in machine learning and data analysis that involves grouping similar data points together based on certain features or characteristics. The goal of clustering is to partition a dataset into subsets, or clusters, so that data points within the same cluster are more similar to each other than to those in other clusters. This process helps identify patterns, structures, or natural groupings within the data, facilitating better understanding and analysis of complex datasets. Various clustering algorithms, such as k-means, hierarchical clustering, and DBSCAN, are used to achieve this grouping based on different criteria.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:51:49.689818Z","iopub.execute_input":"2024-02-28T06:51:49.691332Z","iopub.status.idle":"2024-02-28T06:51:49.698484Z","shell.execute_reply.started":"2024-02-28T06:51:49.69128Z","shell.execute_reply":"2024-02-28T06:51:49.697052Z"}}
def Clustering_AfterDmR(k,data) :
    kmeans = KMeans(n_clusters=k ,**kmeans_set)
    kmeans.fit(data)
    dict = {'n_clusters' : k ,  'Silhouette score' : silhouette_score(data ,kmeans.labels_ ) ,
            'Calinski Harabasz score' :  calinski_harabasz_score(data, kmeans.labels_),
             'Davies Bouldin' : davies_bouldin_score(data, kmeans.labels_)}
    return (dict)
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:51:54.189164Z","iopub.execute_input":"2024-02-28T06:51:54.190637Z","iopub.status.idle":"2024-02-28T06:51:54.197383Z","shell.execute_reply.started":"2024-02-28T06:51:54.190568Z","shell.execute_reply":"2024-02-28T06:51:54.196028Z"}}
kmeans_set = {"init":"k-means++" , "n_init" : 10 , "max_iter" : 300 , "random_state" :222}
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:51:58.820186Z","iopub.execute_input":"2024-02-28T06:51:58.820696Z","iopub.status.idle":"2024-02-28T06:51:59.331689Z","shell.execute_reply.started":"2024-02-28T06:51:58.820662Z","shell.execute_reply":"2024-02-28T06:51:59.330362Z"}}
from yellowbrick.cluster import KElbowVisualizer
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score , calinski_harabasz_score , davies_bouldin_score
from kneed import KneeLocator
# %% [markdown]
# Principal Component Analysis (PCA) is applied to the dataset with three components, resulting in a transformed dataset with features labeled as "feature1," "feature2," and "feature3." The dataset statistics indicate the characteristics of the transformed features:
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T06:58:29.465458Z","iopub.execute_input":"2024-02-28T06:58:29.466056Z","iopub.status.idle":"2024-02-28T06:58:29.591722Z","shell.execute_reply.started":"2024-02-28T06:58:29.466016Z","shell.execute_reply":"2024-02-28T06:58:29.589937Z"}}
pca = PCA(n_components=3)
pca.fit(X_train)
explained_Variance = pca.explained_variance_ratio
PCA_ds = pd.DataFrame(pca.transform(X_train), columns=(["feature1","feature2", "feature3"]))
PCA_ds.describe().T
# %% [markdown]
# These statistics provide insights into the distribution and variation of the transformed features after applying PCA, helping to understand the impact of dimensionality reduction on the dataset.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:00:08.080171Z","iopub.execute_input":"2024-02-28T07:00:08.080812Z","iopub.status.idle":"2024-02-28T07:00:11.234226Z","shell.execute_reply.started":"2024-02-28T07:00:08.080761Z","shell.execute_reply":"2024-02-28T07:00:11.232772Z"}}
scores_df_list = [] # List to store DataFrames
for k in range(3, 21, 2):
    clustering_result = Clustering_AfterDmR(k, PCA_ds)
    df_result = pd.DataFrame.from_dict(clustering_result, orient='index').T
    scores_df_list.append(df_result)
# Concatenate the list of DataFrames into a single DataFrame
scores_df = pd.concat(scores_df_list, ignore_index=True)
```

```
scores_df.describe().T
# %% [markdown]
# These metrics provide insights into the performance and characteristics of the clustering algorithm, including the number of clusters formed, silhouette scores indicating cluster cohesion, Calinski Harabasz scores measuring cluster separation, and Davies Bouldin scores assessing cluster compactness.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:00:20.098973Z","iopub.execute_input":"2024-02-28T07:00:20.099411Z","iopub.status.idle":"2024-02-28T07:00:21.034135Z","shell.execute_reply.started":"2024-02-28T07:00:20.099372Z","shell.execute_reply":"2024-02-28T07:00:21.032833Z"}}
for col in scores_df.drop(columns=['n_clusters']):
    plt.figure(figsize=(10,10))
    plt.title('Number of Clusters and '+col,backgroundcolor='black',color='white',fontsize=15 )
    plt.xticks(fontsize=12)
    plt.yticks(fontsize=12)
    plt.xlabel('Number of clusters',fontdict=font1,labelpad=16)
    plt.ylabel(col,fontdict=font1,labelpad=16)
    plt.plot(range(3,21,2),scores_df[col],marker='o',markerfacecolor='red')
    plt.show()
# %% [markdown]
# After evaluating clustering performance metrics, I decided that 3 clusters would be the optimal choice. This decision is based on the obtained scores, suggesting that a division into three distinct groups provides a balance between capturing meaningful patterns in the data and avoiding unnecessary complexity
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:00:32.501028Z","iopub.execute_input":"2024-02-28T07:00:32.501455Z","iopub.status.idle":"2024-02-28T07:00:32.791021Z","shell.execute_reply.started":"2024-02-28T07:00:32.501424Z","shell.execute_reply":"2024-02-28T07:00:32.789345Z"}}
kmeans3 = KMeans(n_clusters=3 , **kmeans_set)
kmeans3.fit(PCA_ds)
print(f'Silhouette score: {silhouette_score(PCA_ds ,kmeans3.labels_)},\nCalinski Harabasz score: {calinski_harabasz_score(PCA_ds, kmeans3.labels_)},\nDavies Bouldin: {davies_bouldin_score(PCA_ds, kmeans3.labels_)}')
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:00:36.754918Z","iopub.execute_input":"2024-02-28T07:00:36.755395Z","iopub.status.idle":"2024-02-28T07:00:38.625713Z","shell.execute_reply.started":"2024-02-28T07:00:36.755357Z","shell.execute_reply":"2024-02-28T07:00:38.624611Z"}}
List = []
for k in range(1,21):
    kmeans = KMeans(n_clusters=k,**kmeans_set)
    kmeans.fit(PCA_ds)
    List.append(kmeans.inertia_)
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:00:42.886002Z","iopub.execute_input":"2024-02-28T07:00:42.886466Z","iopub.status.idle":"2024-02-28T07:00:42.893261Z","shell.execute_reply.started":"2024-02-28T07:00:42.88643Z","shell.execute_reply":"2024-02-28T07:00:42.891473Z"}}
font2 = {'family': 'serif', 'color': 'darkred', 'weight': 'normal', 'size': 14}
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:00:46.018786Z","iopub.execute_input":"2024-02-28T07:00:46.020402Z","iopub.status.idle":"2024-02-28T07:00:46.393223Z","shell.execute_reply.started":"2024-02-28T07:00:46.020335Z","shell.execute_reply":"2024-02-28T07:00:46.391995Z"}}
plt.figure(figsize=(10,7))
plt.title('Number of Clusters and Inertia',backgroundcolor='gray',color='white',fontsize=15 )
plt.plot(range(1,21),List)
plt.xticks(range(1,21),fontsize=12)
plt.yticks(fontsize=12)
plt.xlabel('Number of clusters',fontdict=font2,labelpad=16)
plt.ylabel('Inertia',fontdict=font2,labelpad=16)
plt.show()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:01:32.665847Z","iopub.execute_input":"2024-02-28T07:01:32.666397Z","iopub.status.idle":"2024-02-28T07:01:32.678527Z","shell.execute_reply.started":"2024-02-28T07:01:32.666335Z","shell.execute_reply":"2024-02-28T07:01:32.677309Z"}}
k1 = KneeLocator(range(1,21),List,curve='convex',direction='decreasing')
k1.elbow
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:01:36.530734Z","iopub.execute_input":"2024-02-28T07:01:36.531136Z","iopub.status.idle":"2024-02-28T07:01:36.815227Z","shell.execute_reply.started":"2024-02-28T07:01:36.531109Z","shell.execute_reply":"2024-02-28T07:01:36.813996Z"}}
kmeans = KMeans(n_clusters=5 , **kmeans_set)
kmeans.fit(PCA_ds)
print(f'Silhouette score: {silhouette_score(PCA_ds ,kmeans.labels_)},\nCalinski Harabasz score: {calinski_harabasz_score(PCA_ds, kmeans.labels_)},\nDavies Bouldin: {davies_bouldin_score(PCA_ds, kmeans.labels_)}')
# %% [markdown]
# In contrast to the 5-cluster solution suggested by the k-elbow method, the performance scores for the 3-cluster solution demonstrate superiority. This indicates that, despite considering a smaller number of clusters, the quality of the clustering is higher, making the 3-cluster solution a more favorable choice based on the evaluation metrics.
#
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:01:42.333091Z","iopub.execute_input":"2024-02-28T07:01:42.333808Z","iopub.status.idle":"2024-02-28T07:01:42.341001Z","shell.execute_reply.started":"2024-02-28T07:01:42.333761Z","shell.execute_reply":"2024-02-28T07:01:42.339651Z"}}
PCA_ds = PCA_ds.copy()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:01:46.276152Z","iopub.execute_input":"2024-02-28T07:01:46.276691Z","iopub.status.idle":"2024-02-28T07:01:46.299682Z","shell.execute_reply.started":"2024-02-28T07:01:46.276657Z","shell.execute_reply":"2024-02-28T07:01:46.297642Z"}}
PCA_ds['Label'] = kmeans3.labels_
PCA_ds
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:01:50.860802Z","iopub.execute_input":"2024-02-28T07:01:50.861642Z","iopub.status.idle":"2024-02-28T07:01:50.870996Z","shell.execute_reply.started":"2024-02-28T07:01:50.861595Z","shell.execute_reply":"2024-02-28T07:01:50.869196Z"}}
centroids = kmeans3.cluster_centers_
centroids
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:01:54.69723Z","iopub.execute_input":"2024-02-28T07:01:54.697818Z","iopub.status.idle":"2024-02-28T07:01:55.314342Z","shell.execute_reply.started":"2024-02-28T07:01:54.697779Z","shell.execute_reply":"2024-02-28T07:01:55.313256Z"}}
plt.figure(figsize=(10, 7))
plt.title('Feature 1 and Feature 2', backgroundcolor='gray', color='white', fontsize=15)
sns.scatterplot(data=PCA_ds, x='feature1', y='feature2', hue='Label')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)
plt.xlabel('Feature 1', fontdict=font2, labelpad=16)
plt.ylabel('Feature 2', fontdict=font2, labelpad=16)
plt.show()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:01.546146Z","iopub.execute_input":"2024-02-28T07:02:01.54688Z","iopub.status.idle":"2024-02-28T07:02:02.198987Z","shell.execute_reply.started":"2024-02-28T07:02:01.546829Z","shell.execute_reply":"2024-02-28T07:02:02.197237Z"}}
plt.figure(figsize=(10, 7))
plt.title('Feature 1 and Feature 3', backgroundcolor='gray', color='white', fontsize=15)
sns.scatterplot(data=PCA_ds, x='feature1', y='feature3', hue='Label')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.scatter(centroids[:, 0], centroids[:, 2], c='red', s=50)
plt.xlabel('Feature 1', fontdict=font2, labelpad=16)
plt.ylabel('Feature 3', fontdict=font2, labelpad=16)
plt.show()
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:15.920728Z","iopub.execute_input":"2024-02-28T07:02:15.92121Z","iopub.status.idle":"2024-02-28T07:02:16.633974Z","shell.execute_reply.started":"2024-02-28T07:02:15.921176Z","shell.execute_reply":"2024-02-28T07:02:16.629871Z"}}
plt.figure(figsize=(10, 7))
plt.title('Feature 2 and Feature 3', backgroundcolor='gray', color='white', fontsize=15)
sns.scatterplot(data=PCA_ds, x='feature2', y='feature3', hue='Label')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.scatter(centroids[:, 1], centroids[:, 2], c='red', s=50)
plt.xlabel('Feature 2', fontdict=font2, labelpad=16)
plt.ylabel('Feature 3', fontdict=font2, labelpad=16)
plt.show()
# %% [markdown]
# Now, let's implement clustering and Principal Component Analysis (PCA) on the X_test dataset. This involves grouping similar data points together and reducing dimensionality for efficient analysis and visualization.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:29.001418Z","iopub.execute_input":"2024-02-28T07:02:29.001876Z","iopub.status.idle":"2024-02-28T07:02:29.017565Z","shell.execute_reply.started":"2024-02-28T07:02:29.001844Z","shell.execute_reply":"2024-02-28T07:02:29.015823Z"}}
pca_test = PCA(n_components=3)
pca_test.fit(X_test)
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:32.466155Z","iopub.execute_input":"2024-02-28T07:02:32.466964Z","iopub.status.idle":"2024-02-28T07:02:32.496843Z","shell.execute_reply.started":"2024-02-28T07:02:32.466926Z","shell.execute_reply":"2024-02-28T07:02:32.495669Z"}}
explained_variance = pca_test.explained_variance_ratio_
PCA_test = pd.DataFrame(pca_test.transform(X_test), columns=(["feature1","feature2", "feature3"]))
PCA_test.describe().T
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:36.444007Z","iopub.execute_input":"2024-02-28T07:02:36.445657Z","iopub.status.idle":"2024-02-28T07:02:36.502192Z","shell.execute_reply.started":"2024-02-28T07:02:36.445607Z","shell.execute_reply":"2024-02-28T07:02:36.50067Z"}}
```

```
kmeans_T.fit(PCA_test)
print(f'Silhouette score: {silhouette_score(PCA_test,kmeans_T.labels_)},\nCalinski Harabasz score: {calinski_harabasz_score(PCA_test, kmeans_T.labels_)},\nDavies Bouldin: {davies_bouldin_score(PCA_test, kmeans_T.labels_)}')
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:40.075639Z","iopub.execute_input":"2024-02-28T07:02:40.076043Z","iopub.status.idle":"2024-02-28T07:02:40.082187Z","shell.execute_reply.started":"2024-02-28T07:02:40.076013Z","shell.execute_reply":"2024-02-28T07:02:40.081013Z"}}
centroids_T = kmeans_T.cluster_centers_
PCA_test['Label'] = kmeans_T.labels_
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:44.308494Z","iopub.execute_input":"2024-02-28T07:02:44.309013Z","iopub.status.idle":"2024-02-28T07:02:44.762907Z","shell.execute_reply.started":"2024-02-28T07:02:44.308979Z","shell.execute_reply":"2024-02-28T07:02:44.761653Z"}}
plt.figure(figsize=(10, 7))
plt.title('Feature 1 and Feature 2', backgroundcolor='gray', color='white', fontsize=15)
sns.scatterplot(data=PCA_test, x='feature1', y='feature2', hue='Label')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.scatter(centroids_T[:, 0], centroids_T[:, 1], c='red', s=50)
plt.xlabel('Feature 1', fontdict=font2, labelpad=16)
plt.ylabel('Feature 2', fontdict=font2, labelpad=16)
plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:52.428615Z","iopub.execute_input":"2024-02-28T07:02:52.429155Z","iopub.status.idle":"2024-02-28T07:02:52.916043Z","shell.execute_reply.started":"2024-02-28T07:02:52.429116Z","shell.execute_reply":"2024-02-28T07:02:52.914749Z"}}
plt.figure(figsize=(10, 7))
plt.title('Feature 1 and Feature 3', backgroundcolor='gray', color='white', fontsize=15)
sns.scatterplot(data=PCA_test, x='feature1', y='feature3', hue='Label')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.scatter(centroids_T[:, 0], centroids_T[:, 2], c='red', s=50)
plt.xlabel('Feature 1', fontdict=font2, labelpad=16)
plt.ylabel('Feature 3', fontdict=font2, labelpad=16)
plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:02:58.747249Z","iopub.execute_input":"2024-02-28T07:02:58.747672Z","iopub.status.idle":"2024-02-28T07:02:59.217775Z","shell.execute_reply.started":"2024-02-28T07:02:58.747642Z","shell.execute_reply":"2024-02-28T07:02:59.216641Z"}}
plt.figure(figsize=(10, 7))
plt.title('Feature 2 and Feature 3', backgroundcolor='gray', color='white', fontsize=15)
sns.scatterplot(data=PCA_test, x='feature2', y='feature3', hue='Label')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.scatter(centroids_T[:, 1], centroids_T[:, 2], c='red', s=50)
plt.xlabel('Feature 2', fontdict=font2, labelpad=16)
plt.ylabel('Feature 3', fontdict=font2, labelpad=16)
plt.show()
# %% [markdown]
# The entire dataset is eligible for clustering analysis. Applying clustering to the entire dataset involves grouping data points based on similarities or patterns, providing insights into the overall structure of the data.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:06.532685Z","iopub.execute_input":"2024-02-28T07:03:06.533116Z","iopub.status.idle":"2024-02-28T07:03:06.546832Z","shell.execute_reply.started":"2024-02-28T07:03:06.533083Z","shell.execute_reply":"2024-02-28T07:03:06.545648Z"}}
from sklearn.preprocessing import StandardScaler
# Select only numeric columns
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
# Create a DataFrame with only numeric columns
numeric_df = df[numeric_columns]
# Apply StandardScaler to numeric columns
scaler = StandardScaler()
scaled_features = scaler.fit_transform(numeric_df)

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:10.556532Z","iopub.execute_input":"2024-02-28T07:03:10.557997Z","iopub.status.idle":"2024-02-28T07:03:11.786491Z","shell.execute_reply.started":"2024-02-28T07:03:10.557937Z","shell.execute_reply":"2024-02-28T07:03:11.784647Z"}}
from sklearn.impute import SimpleImputer
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
# Assuming scaled_features is your data with NaN values
# Create an imputer instance
imputer = SimpleImputer(strategy='mean')
# Impute missing values
scaled_features_imputed = imputer.fit_transform(scaled_features)
# Now, you can apply KMeans
kmeans = KMeans(n_clusters=3, **kmeans_set)
kmeans.fit(scaled_features_imputed)

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:15.84161Z","iopub.execute_input":"2024-02-28T07:03:15.842008Z","iopub.status.idle":"2024-02-28T07:03:15.850267Z","shell.execute_reply.started":"2024-02-28T07:03:15.841978Z","shell.execute_reply":"2024-02-28T07:03:15.849131Z"}}
centroids = kmeans.cluster_centers_
centroids
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:20.33558Z","iopub.execute_input":"2024-02-28T07:03:20.336139Z","iopub.status.idle":"2024-02-28T07:03:20.343638Z","shell.execute_reply.started":"2024-02-28T07:03:20.336094Z","shell.execute_reply":"2024-02-28T07:03:20.342371Z"}}
df['Label'] = kmeans.labels_
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:23.921791Z","iopub.execute_input":"2024-02-28T07:03:23.922255Z","iopub.status.idle":"2024-02-28T07:03:23.955067Z","shell.execute_reply.started":"2024-02-28T07:03:23.922214Z","shell.execute_reply":"2024-02-28T07:03:23.953534Z"}}
df
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:28.602138Z","iopub.execute_input":"2024-02-28T07:03:28.602659Z","iopub.status.idle":"2024-02-28T07:03:28.622415Z","shell.execute_reply.started":"2024-02-28T07:03:28.602624Z","shell.execute_reply":"2024-02-28T07:03:28.620631Z"}}
# Select only numeric columns
numeric_columns = df.select_dtypes(include='number')
# Calculate correlation matrix
corr = numeric_columns.corr()
# Display the correlation matrix
print(corr)

# %% [markdown]
# Furthermore, I created scatter plots to visualize relationships between some features based on their correlation. The clusters can also be observed on the scatter plots based on the 'Label' column.
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:33.358256Z","iopub.execute_input":"2024-02-28T07:03:33.358777Z","iopub.status.idle":"2024-02-28T07:03:33.368796Z","shell.execute_reply.started":"2024-02-28T07:03:33.35874Z","shell.execute_reply":"2024-02-28T07:03:33.366882Z"}}
df.columns
# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:42.001336Z","iopub.execute_input":"2024-02-28T07:03:42.001792Z","iopub.status.idle":"2024-02-28T07:03:44.372467Z","shell.execute_reply.started":"2024-02-28T07:03:42.001759Z","shell.execute_reply":"2024-02-28T07:03:44.370827Z"}}
plt.figure(figsize=(10, 7))
plt.title(' Q and II', backgroundcolor='gray', color='white', fontsize=15)
# Extract the numeric columns for the scatterplot
x_col = df.columns.get_loc('country')  # Choose the column for the x-axis
y_col = df.columns.get_loc('Q')  # Choose the column for the y-axis
# Plot the scatterplot
sns.scatterplot(data=df, x=df.iloc[:, x_col], y=df.iloc[:, y_col], hue=df['Label'])
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.xlabel('Country', fontsize=14)
plt.ylabel('Q', fontsize=14)
plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:49.782409Z","iopub.execute_input":"2024-02-28T07:03:49.782844Z","iopub.status.idle":"2024-02-28T07:03:50.458329Z","shell.execute_reply.started":"2024-02-28T07:03:49.782811Z","shell.execute_reply":"2024-02-28T07:03:50.456932Z"}}
plt.figure(figsize=(10, 7))
plt.title(' Q and II', backgroundcolor='gray', color='white', fontsize=15)
# Extract the numeric columns for the scatterplot
x_col = df.columns.get_loc('year')  # Choose the column for the x-axis
y_col = df.columns.get_loc('Q')  # Choose the column for the y-axis
# Plot the scatterplot
```

```
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.xlabel('year', fontsize=14)
plt.ylabel('Q', fontsize=14)
plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:03:57.116642","iopub.execute_input":"2024-02-28T07:03:57.117069Z","iopub.status.idle":"2024-02-28T07:03:57.772498Z","shell.execute_reply.started":"2024-02-28T07:03:57.117042","shell.execute_reply":"2024-02-28T07:03:57.771253Z"}}
plt.figure(figsize=(10, 7))
plt.title(' Q and II', backgroundcolor='gray', color='white', fontsize=15)
# Extract the numeric columns for the scatterplot
x_col = df.columns.get_loc('markup_CME')  # Choose the column for the x-axis
y_col = df.columns.get_loc('Q')  # Choose the column for the y-axis
# Plot the scatterplot
sns.scatterplot(data=df, x=df.iloc[:, x_col], y=df.iloc[:, y_col], hue=df['Label'])
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.xlabel('markup_CME', fontsize=14)
plt.ylabel('Q', fontsize=14)
plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2024-02-28T07:04:06.596342Z","iopub.execute_input":"2024-02-28T07:04:06.596968Z","iopub.status.idle":"2024-02-28T07:04:06.823632Z","shell.execute_reply.started":"2024-02-28T07:04:06.596932Z","shell.execute_reply":"2024-02-28T07:04:06.822462Z"}}
colors = ['lightskyblue' , 'lightpink' , 'cadetblue','lightskyblue' , 'lightpink' , 'cadetblue','lightskyblue']
pl = sns.countplot(x=df["Label"], palette= colors)
pl.set_title("Distribution Of The Clusters")
plt.xlabel('Cluster',fontdict=font2,labelpad=16)
plt.ylabel('Count' ,fontdict=font2,labelpad=16)
```

ДОДАТОК Б

ОЦІНКА ЗМІНИ ЕКОНОМІЧНИХ ПОКАЗНИКІВ (Q, II, VA, LAB, K) ТА НАЦІНКИ (MARKUP_GME) ПРОТЯГОМ ЧАСУ ДЛЯ КОЖНОЇ КРАЇНИ

Рисунок В.1 - Легенда

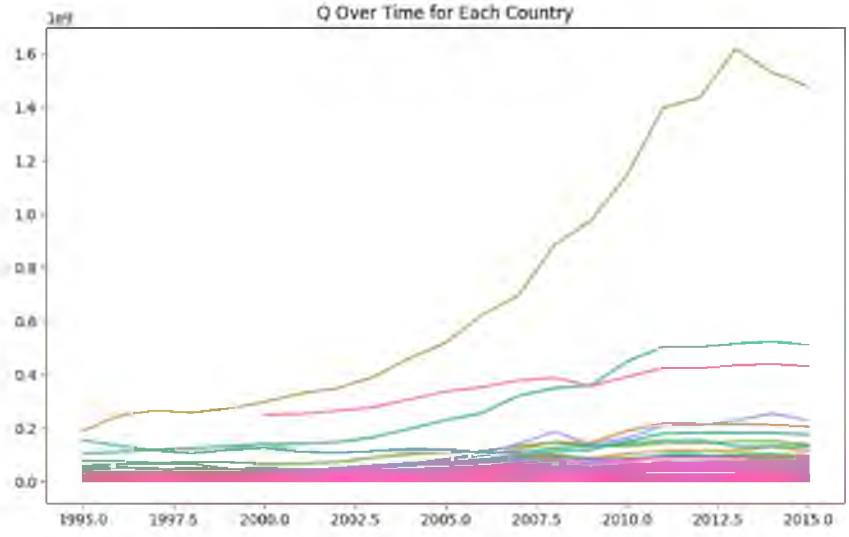


Рисунок В.2 - Q

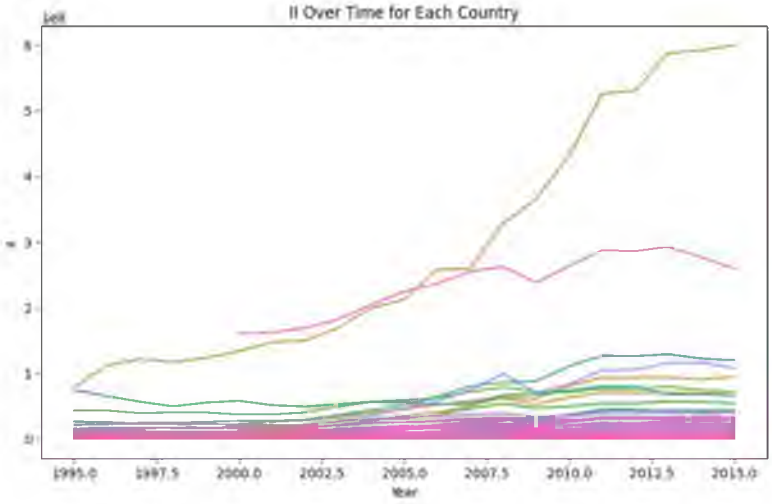


Рисунок В.3 - II

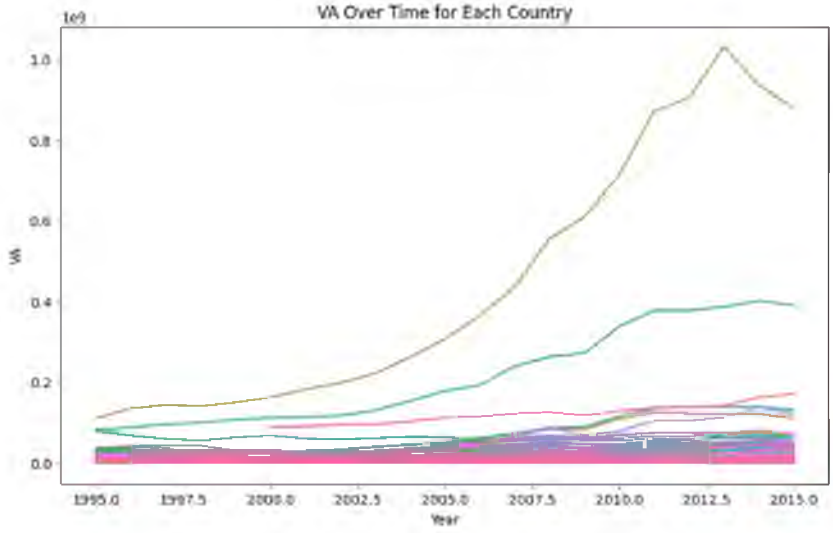


Рисунок B.4 - VA

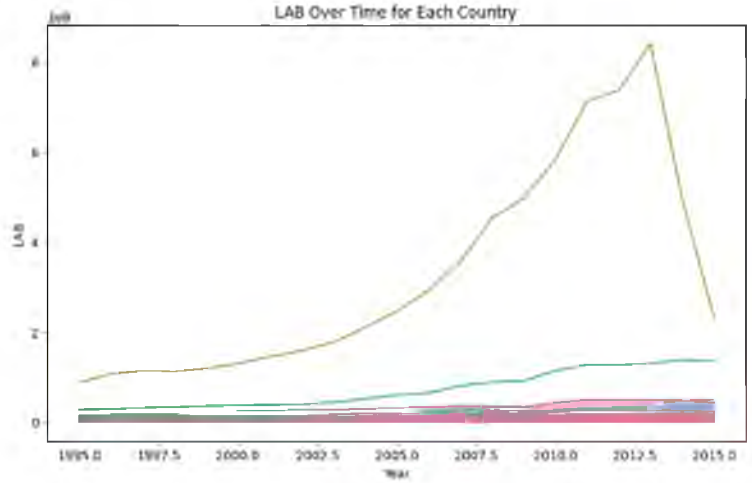


Рисунок B.5 - LAB

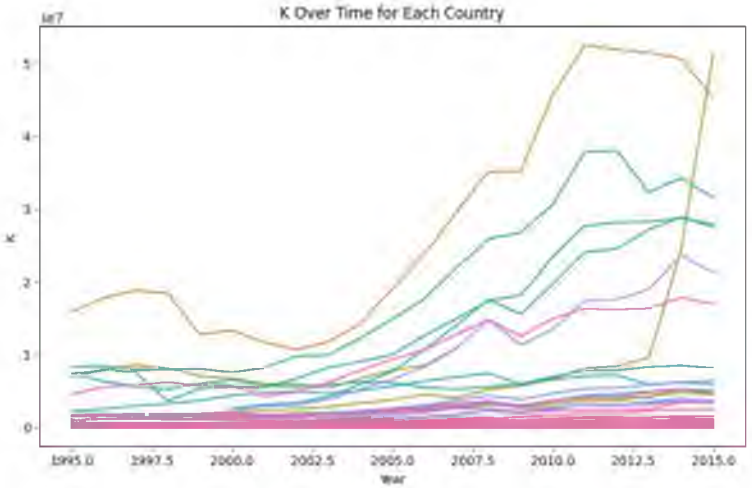
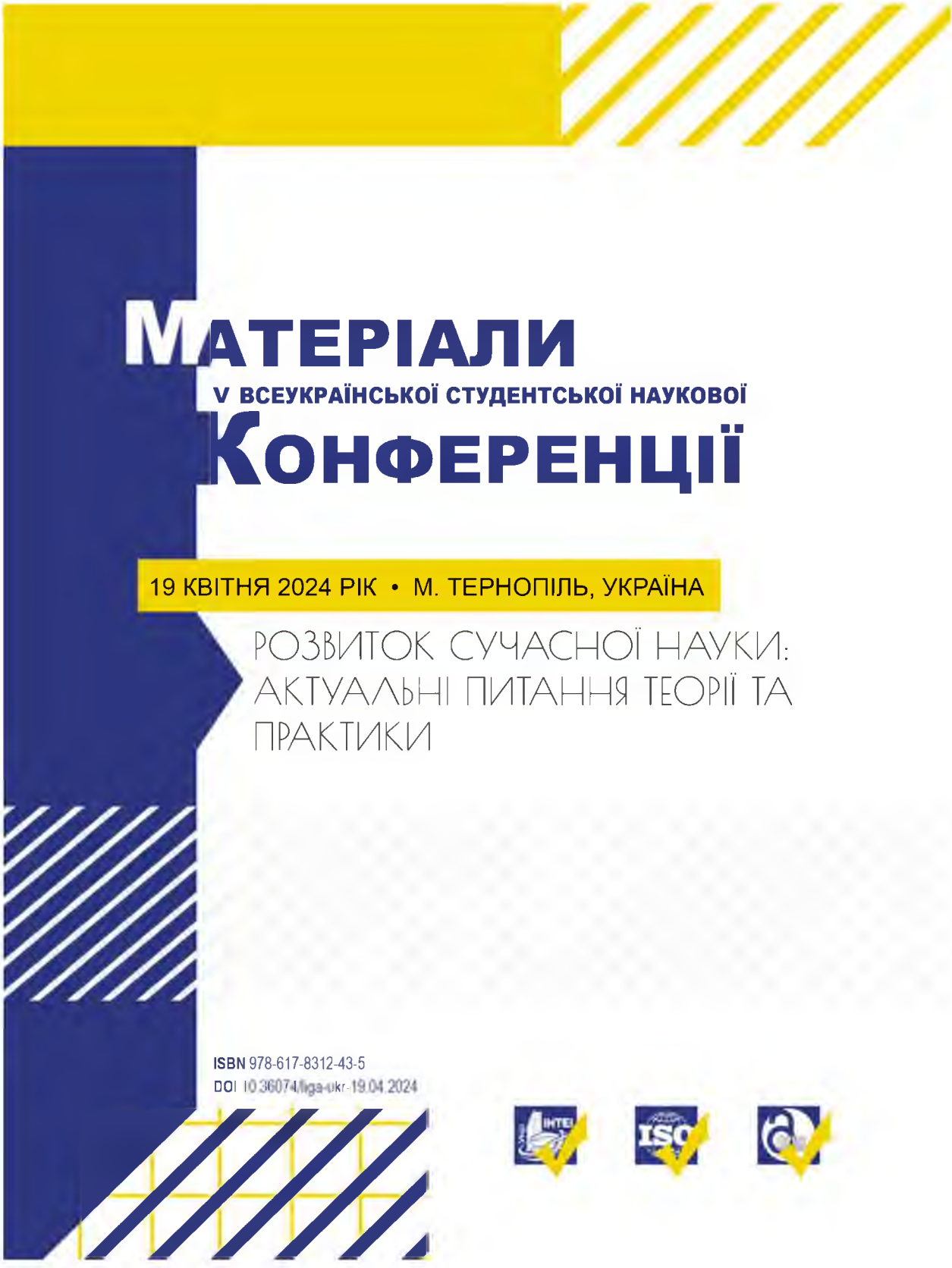


Рисунок B.6 – K

ДОДАТОК В

АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ



УДК 082:001
Р 64

Голова оргкомітету: Коренюк І.О.
Верстка: Зрада С.І.
Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 32 від 18.04.2024 року.

Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та інформаційному бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення №25 від 05.01.2024).

Р 64 **Розвиток сучасної науки: актуальні питання теорії та практики**: матеріали V Всеукраїнської студентської наукової конференції, м. Тернопіль, 19 квітня, 2024 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2024. — 246 с.

ISBN 978-617-8312-43-5
DOI 10.62732/liga-ukr-19.04.2024

Викладено матеріали учасників V Всеукраїнської мультидисциплінарної студентської наукової конференції «Розвиток сучасної науки: актуальні питання теорії та практики», яка відбулася 19 квітня 2024 року у місті Тернопіль, Україна.

УДК 082:001

ISBN 978-617-8312-43-5

СЕКЦІЯ 12.
ТЕХНОЛОГІЇ ЛЕГКОЇ ТА ДЕРЕВООБРОБНОЇ ПРОМИСЛОВОСТІ

СЕКЦІЯ 13.
ЕНЕРГЕТИКА ТА ЕНЕРГЕТИЧНЕ МАШИНОБУДУВАННЯ

СЕКЦІЯ 13.
КОМП'ЮТЕРНА ТА ПРОГРАМНА ІНЖЕНЕРІЯ

СЕКЦІЯ 14.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

Місюра Богдан Романович, здобувач вищої освіти факультету
комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Науковий керівник: Сапожник Григорій Вікторович, канд. іст. наук,
доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

АЛГОРИТМ РЕКОМЕНДАЦІЇ ДЛЯ ПЕРСОНАЛІЗАЦІЇ КОРИСТУВАЦЬКОГО ДОСВІДУ В ЕЛЕКТРОННІЙ КОМЕРЦІЇ

Завдяки поєднанню різних технологій та методів машинного навчання, таких як колаборативна фільтрація, контент-орієнтована фільтрація та гібридні системи, сучасні рекомендаційні системи можуть ефективно прогнозувати та визначати товари та послуги, які найбільш імовірно зацікавлять конкретного користувача. Ця персоналізація не тільки підвищує задоволеність клієнтів, але й сприяє більшому взаємодійному та залученому досвіду покупок.

Алгоритм (Рис.1), який розроблено, включає кроки зі збору та аналізу даних, створення моделей для генерування рекомендацій, а також їх інтеграції та оцінки. Він призначений для того, щоб допомогти розробникам та маркетологам ефективно впроваджувати персоналізацію в їхні електронні комерційні рішення.

Кроки:

1. Збір даних про користувача:

- Зберігайте історію переглядів, покупок та взаємодій користувача на сайті.
- Використовуйте форми для реєстрації, анкети або соціальні мережі для отримання інформації про вподобання та демографічні дані користувачів.

2. Обробка та аналіз даних:

- Аналізуйте поведінкові та демографічні дані для виявлення шаблонів та переваг користувачів.
- Класифікуйте користувачів за групами на основі їх інтересів та поведінки.

3. Розробка моделі рекомендації:

- Використовуйте методи машинного навчання, такі як колаборативна фільтрація, контент-орієнтована фільтрація або гібридні моделі для створення персоналізованих рекомендацій.
- Навчіть модель на основі історичних даних покупок та взаємодій користувачів.

4. Імплементація системи рекомендацій:

- Інтегруйте систему рекомендацій на ваш сайт електронної комерції, щоб вона автоматично відображала персоналізовані пропозиції та продукти.
- Використовуйте рекомендації для підвищення задоволеності користувачів, збільшення часу перебування на сайті та зростання продажів.

5. Оцінка та оптимізація:

- Аналізуйте ефективність рекомендаційних систем через метрики такі як конверсія, виручка та задоволеність клієнтів.
- Постійно удосконалюйте моделі, використовуючи зворотний зв'язок

користувачів та оновлення даних.

Рис. 1. **Структура формування рекомендацій для персоналізації користувацького досвіду в електронній комерції**

6. Адаптація до нових трендів:

• Враховуйте зміни на ринку та нові тренди у вподобаннях користувачів для внесення коректівів у систему рекомендацій.

• Регулярно оновлюйте систему для відповідності поточним вимогам ринку та технологіям.

Такий підхід дозволить створити ефективну систему рекомендацій, яка забезпечить персоналізацію взаємодії з користувачами, збільшить їх задоволеність та лояльність, а також підвищить конверсію та загальний обіг в електронній комерції.

Список використаних джерел:

1. Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. Expert Systems with Applications, 97, 205-227.