Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ І РЕАЛІЗАЦІЯ МЕТОДУ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ДЛЯ ВИРІШЕННЯ МАРКЕТИНГОВИХ ЗАВДАНЬ

(тема)

Виконав:
студент 2 курсу, групи ІНФм-19-1
Кравець Р.А.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Кузьомін О.Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)
2020 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Кравцю Роману Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження і реалізація методу інтелектуального аналізу даних для вирішення маркетинговиїх завдань*

затверджена наказом по університету від « 23 » жовтня 2020 року № 1428Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 28 листопада 2020 р.

3. Вихідні дані до роботи *Математичні моделі кластеризації даних, методи графічного представлення результатів кластеризації, теоретичні відомості про методи кластеризації часових рядів*

4. Перелік питань, що потрібно опрацювати в роботі
1. Огляд методів многомірного аналізу даних з пропущеними значеннями
2. Огляд методів кластеризації
3. Математичні моделі кластеризації часових рядів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Представлення даних у вигляді множини

Візуалізація даних у вигляді середнього квадратичного відхилення

Результат кластеризації даних

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	02.10.2020	
2	Аналіз завдання, підбір літератури	01.11.20-05.11.20	
3	Аналіз літератури з досліджуваної проблеми	05.11.20-07.11.20	
4	Аналіз технічних засобів	08.11.20-10.11.20	
5	Розробка методу	10.11.20-25.11.20	
6	Програмна реалізація	25.11.20-29.11.20	
7	Оформлення пояснювальної записки	29.11.20-1.12.20	
8	Перевірка на плагіат	02.12.20	
9	Рецензування	03.12.20	
10	Підготовка презентації та доповіді	04.12.20	
11	Занесення роботи в електронний архів	06.12.20	
12	Попередній захист атестаційної роботи	07.12.20	

Дата видачі завдання __23_ __жовтня___ 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Кузьомін О. Я.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до атестаційної роботи: 79 с., 3 табл., 22 рис., 38 джерел.

GOOGLE ANALYTICS, НЕПОВНІ ДАНІ, КЛАСТЕРИЗАЦІЯ ЧАСОВИХ РЯДІВ, КЛАСТЕРНИЙ АНАЛІЗ, ОБРОБКА ТА АНАЛІЗ ДАНИХ, DATA MINING, СТОХАСТИЧНІ МЕТОДИ, ПРОГНОЗУВАННЯ.

Метою є дослідження нестохастичних методів прогнозування та доцільність їх використання в реальних маркетингових задачах. Вирішується задача кластеризації та прогнозування багатовимірних часових рядів.

Об’єктом дослідження є дані з системи Google Analytics, які представлені у вигляді часових рядів.

Робота присвячена дослідженню проблеми многомірного аналізу даних з пропущеними значеннями, що піддаються кластеризації для прогнозування.

Розглянуті методи позбавлення від пропусків в задачах кластеризації даних. Досліджено проблеми кластеризації даних, що містять пропущені значення та розгляд методів, які дозволяють розв’язати цю задачу. Проведено тестування та порівняння результатів кожного з методів.

У результаті роботи реалізовані методи кластеризації та прогнозування та проведено аналіз багатовимірних данних Google Analytics.

GOOGLE ANALYTICS, INCOMPLETE DATA, CLUSTER ANALYSIS OF TIME SERIES, CLUSTER ANALYSIS, DATA PROCESSING AND ANALYSIS, DATA MINING, STOCHASTIC METHODS, FORECASTING.

The aim is to study non-stochastic forecasting methods and the feasibility of their use in real marketing tasks. The problem of clustering and forecasting of multidimensional time series is solved.

The object of the study is data from the Google Analytics system, which is presented in the form of time series.

The work is devoted to the study of the problem of multidimensional analysis of data with omitted values that can be clustered for forecasting.

Methods of getting rid of gaps in data clustering problems are considered. The problems of clustering data that contain missing values and the methods that allow solving this problem are investigated. Testing and comparison of the results of each of the methods were performed.

As a result of implemented, the analysis of multidimensional data of Google Analytics is completed, the methods of clustering and forecasting are implemented.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GA – Google Analytics

CSV – comma-separated values, значення, розділені комою

КА – кластерний аналіз

НМ – нечітка множина

ВВП – Валовий внутрішній продукт

DTW – Алгоритм динамічної трансформації часової шкали

ВСТУП

На сьогоднішній день динамічний розвиток інформаційно-комунікаційних технологій надає нам глобальні можливості не тільки в пошуку і організації доступу до потрібної інформації, а також ведення ефективного бізнесу. Трансформація в цифровий бізнес має вирішальне значення для успіху будь-якої компанії. Більш глибоке і широке впровадження цифрових технологій в бізнес і економіку, так звана «цифрова насиченість», веде до кількісних поліпшень продуктивності, які в свою чергу можуть прискорити конкурентоспроможність і економічне зростання. Кожного року конкуренція у сфері digital тільки зростає, тому дуже важливо розуміти поведінку споживачів. Саме в умовах жорсткої конкуренції особливо важливого значення набуває інформація щодо вдосконалення та підвищення результативності бізнесу.

З кожним роком об’єм даних, які відстежує бізнес, збільшується і стали виникати нові маркетингові задачі. Все більший інтерес становить робота з даними і розв’язання проблем, що пов’язані з їх обробкою і подальшим аналізом. Найпопулярнішими напрямками досліджень для вирішення маркетингових задач наразі все частіше стають: Big Data, Data Mining, Machine Learning, постають питання добування даних їх глобальний і інтелектуальний аналіз. Серед таких актуальних завдань знаходять своє місце і поняття класифікації та кластеризації.

Задачі подібні, але основна відмінність полягає у тому, що кластеризація передбачає розбиття за умови початкової невизначеності щодо конкретних груп. Вона може мати критерії щодо кількості кінцевих кластерів, але не їх зміст, загалом, можна сказати, що це навчання без вчителя. Завдання кластеризації полягає в поділі досліджуваної множини об’єктів на групи «схожих» об’єктів, називаних кластерами.

Отже кластеризація становить інтерес, як спосіб попередньої обробки даних, для більш зручного їх подальшого аналізу. Отримавши необхідні

групи, а також їх центроїди можна продовжувати роботу вже з конкретними представниками, а не з усім набором даних. Однієї із цілей кластеризації є виявлення внутрішніх зв’язків між даними шляхом визначення кластерної структури. Розбивка спостережень на групи схожих об’єктів дозволяє спростити подальшу обробку даних і прийняття розв’язків, застосовуючи до кожного кластера свій метод аналізу.

Реалізація методів кластеризації дозволяє краще зрозуміти дані; провести стиснення, виділивши найбільш типових представників, за умов збитковості даних; виявлення новизни, шляхом виділення об’єктів, що не потратили до жодного з кластерів.

Кластеризація в Data Mining набуває цінність тоді, коли вона виступає одним з етапів аналізу даних, побудови закінченого аналітичного рішення. Аналітику часто легше виділити групи схожих об’єктів, вивчити їх особливості і побудувати для кожної групи окрему модель, ніж створювати одну загальну модель на всіх даних. Таким прийомом постійно користуються в маркетингу, виділяючи групи клієнтів, покупців, товарів і розробляючи для кожної з них окрему стратегію.

Актуальною проблемою сьогодні є робота з пропущеними значеннями даних в часових рядах. Існує множина методів, але вони не передбачають відсутності інформації. Це особливо актуально в умовах безкінечно зростаючого об’єму інформації та в вирішенні маркетингових задач.

Таким чином є доцільним розглянути варіанти рішення проблеми кластеризації багатовимірних даних з пропусками, існуючі методи та їх ефективність.

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ ДЛЯ АНАЛІЗУ ДАНИХ

Будь-які методи обробки даних так чи інакше використовуються для структурування та аналізу існуючої інформації. Для більш ефективного вирішення маркетингових завдань таких як просування товарів масового споживання на ринок має сенс сегментувати споживачів на групи за певними параметрами: стать, вік, сімейний стан, дохід сім'ї і так далі. Для цього існує набір математичних методів, які дозволяють встановити закономірності в даних – у випадку з аналізом споживачів такої закономірністю будуть характерні групи споживачів.

Дані, які можуть бути використані для аналізу, бувають чотирьох типів:

- чисельні дані;
- інтервальні дані;
- рангові;
- номінальні дані.

Всі дані, які підходять під один із цих типів, можуть бути проаналізовані за допомогою формальних методів.

Для того щоб працювали більшість методів, бажано мати більше 30 подій. Цієї кількості подій зазвичай досить для отримання інформації, що в даній вибірці спостерігається статистичний ефект. Однак для поділу на групи необхідно мати вже набагато більше число подій – приблизно 30, помножене на число груп. Наприклад, для більш-менш правильного поділу споживачів на 3 групи бажано мати більше 100 респондентів. Безсумнівно, для різних завдань і методів кількість подій може бути різним, і якусь інформацію можна отримувати вже з 10 подій, проте тут діє загальне правило статистики: чим даних більше, тим краще.

1.1 Існуючі методи аналізу та інтелектуальної обробки даних

1.1.1 Кластерний аналіз

Термін «кластерний аналіз» в дійсності включає в себе набір різних алгоритмів класифікації [1]. Загальне питання, що задається дослідниками в багатьох областях, полягає в тому, як розбити дані на групи з близькими за значеннями параметрів. Наприклад, при сегментації ринку можна кластеризувати споживачів за двома параметрами – ціни і якості. Припустимо, компанія – виробник автомобілів провела опитування споживачів, в якому задавала два питання: «За яку ціну Ви готові купити автомобіль?» і «Оцініть якість автомобіля *X* за 50-бальною шкалою». В результаті опитування була побудована діаграма розсіювання «ціна-якість», яка представлена на рис. 1.1.

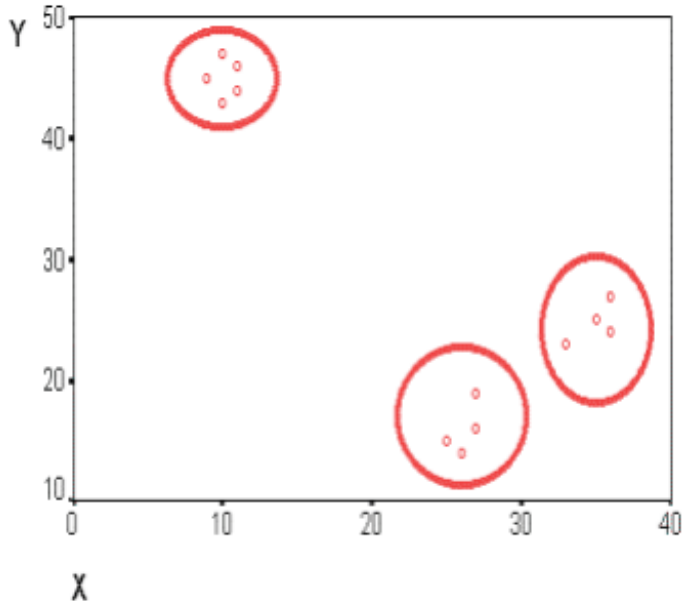


Рисунок 1.1 – Діаграма розсіювання «ціна-якість»

Володіючи цією інформацією, кожній групі споживачів можна запропонувати саме те, що необхідно саме цій групі, і за рахунок цього збільшити рівень продажів компанії. Зрозуміло, в реальному житті кластери, які помітні оком, зустрічаються нечасто, набагато частіше бувають ситуації,

коли всі результуючі параметри змішуються в одну «купу» як зображено на рисунку 1.2.

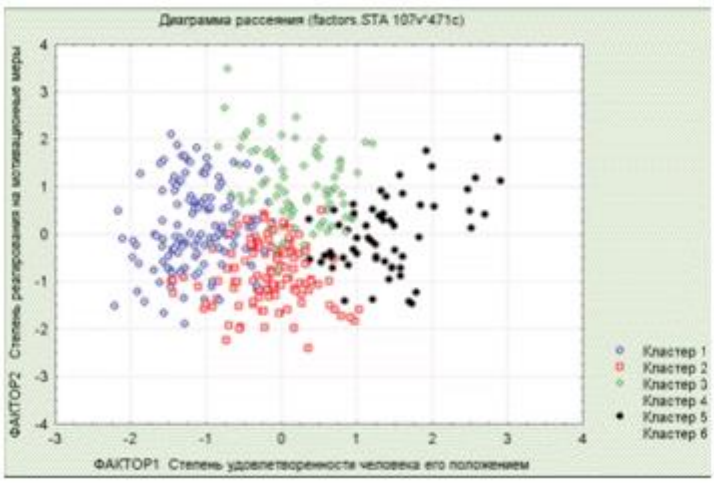


Рисунок 1.2 – Діаграма розсіювання зі змішаними результуючими параметрами

Особливо часто це зустрічається, коли аналізованих параметрів не два, а кілька десятків (кластерний аналіз не обмежує число аналізованих параметрів, тому можна розглядати всю проблему комплексно). Для проведення кластерного аналізу, крім збору даних, необхідно визначити дві речі: на скільки кластерів необхідно розділити дані і як визначити міру подібності в даних [2]. Наприклад, всі підприємства України можно кластеризувати за географічною ознакою на 10 кластерів. Тоді міра подібності буде визначатися комунікаційної близькістю підприємств один до одного.

1.1.2 Факторний аналіз

У разі наявності великої кількості параметрів (більше 100) має сенс згрупувати параметри і аналізувати вже не кожен параметр окремо, а групи параметрів як єдиний комплексний параметр (фактор). В основі факторного

аналізу лежить ідея про те, що за складними взаємозв’язками явно заданих ознак знаходиться відносно простіша структура, яка відображає найбільш істотні риси досліджуваного явища, а «зовнішні» ознаки є функціями прихованих загальних факторів, що визначають цю структуру. Наприклад, для аналізу структури економічного зростання України можна проаналізувати всі макроекономічні параметри, попередньо об’єднавши їх в групи. Одним з таких факторів буде ВВП. Об’єднання параметрів можна робити вручну, емпірично, як це зроблено з ВВП, а можна за допомогою методу факторного аналізу. Застосування факторного аналізу дозволяє, по-перше, зменшувати число розглянутих параметрів, по-друге – знаходити осмислені групи параметрів, кожна з яких буде одним самостійним параметром [3]. Специфікою цього методу є те, що при об’єднанні параметрів в фактори кожен фактор акумулює в собі загальні закономірності в усіх параметрах, відкидаючи особливості кожного параметра окремо.

1.1.3 Нейронні мережі

Початок нейронних мереж як інструменту аналізу даних було покладено в роботі Маккаллок і Питтса. У цій роботі пропонувалася модель штучного нейрона [4]. Передбачалося, що, моделюючи нейронну структуру мозку, можливо наблизитися до штучного інтелекту.

На той час відомо, що мозок людини складається з особливих біологічних клітин – нейронів, і здавалося, що побудова мереж з нейронів дозволить вирішувати складні завдання, які щодня вирішує мозок людини. З тих пір інтерес до нейронних мереж періодично зростав, що обумовлювалося новими розробками в цій області, і зараз нейронні мережі є одним з досить популярних інструментів аналізу даних. Нейронні мережі можуть бути застосовані практично до будь-якої області діяльності, що сильно приваблює багатьох дослідників.

Список завдань для нейронних мереж можна класифікувати в такий спосіб:

– класифікація образів. До відомих додатків відносяться розпізнавання букв, розпізнавання мови, класифікація сигналу електрокардіограми, класифікація клітин крові, забезпечення діяльності біометричних сканерів;

– кластеризація / категоризація. Кластеризація застосовується для вилучення знань, стиснення даних і дослідження властивостей даних;

– апроксимація функцій. Типовим прикладом є шумозаглушення при прийомі сигналу різної природи, незалежно від переданої інформації;

– передбачення / прогноз. Як приклад можна привести пророкування цін на фондовій біржі і прогноз погоди;

– оптимізація. Призначення штату працівників по ряду умінь і чинників є класичними прикладами завдань оптимізації;

– пам’ять, що адресується за змістом (асоціативна пам’ять). Асоціативна пам’ять доступна за вказівкою заданого змісту. Вміст пам’яті може бути викликано навіть по частковому входу або спотвореного змістом [5]. Асоціативна пам’ять може знайти застосування при створенні мультимедійних інформаційних баз даних;

– управління. Прикладом є оптимальне управління двигуном, рульове управління на кораблях, літаках.

Загальна схема аналізу даних за допомогою нейронних мереж складається з 5 етапів:

– вибір типології мережі. Існує 9 типів мереж, на цьому етапі підбирається найбільш відповідний під задачу тип мережі;

– експериментальний підбір пропускної здатності мережі. Після вибору типу необхідно підібрати структуру мережі (кількість нейронів, їх ваги, взаємозв’язку);

– експериментальний підбір параметрів навчання. Далі необхідно експериментально визначити параметри навчання: максимальний час навчання, кількість даних, максимально припустиму похибку;

– навчання мережі. За навчальною вибіркою проводиться навчання мережі. Передбачається, що навчальна вибірка містить в собі інформацію, яка характеризує дані в цілому;

– перевірка адекватності навчання. Проводиться аналіз отриманих результатів на даних, які не входили в навчальну вибірку. Здійснюється ручний контроль результатів роботи нейронної мережі.

1.1.4 Дерева рішень

Дерева рішень – це спосіб представлення правил в ієрархічній послідовної логічної структурі, який дозволяє співвіднести об’єкт або ситуацію на вході з одним або декількома вихідними (термінальними) вузлами [6]. Під правилом розуміється логічна конструкція, представлена у вигляді «якщо ... то».

Метод дерев рішень може допомогти при прийнятті складного рішення, на яке впливають десятки параметрів. Дерева рішень широко застосовуються в багатьох областях діяльності:

– банківська справа. Оцінка кредитоспроможності клієнтів банку при видачі кредитів;

– промисловість. Контроль за якістю продукції (виявлення дефектів), випробування без руйнувань (наприклад, перевірка якості зварювання);

– медицина. Діагностика різних захворювань;

– молекулярна біологія. Аналіз будови амінокислот;

– консалтинг. Компанія McKinsey використовує дерева рішень (issue tree, термін McKinsey) для консультацій своїх клієнтів.

Це далеко не повний список областей, де можна використовувати дерева рішень. Ще не досліджені багато потенційних областей застосування цього інструменту.

1.1.5 Регресійний аналіз

Основною метою регресійного аналізу є визначення наявності та характеру зв’язку між змінними (в найпростішому випадку будується залежність *y(x)* виходячи з приблизної форми кривої) [7]. Кілька років тому американський Інститут стратегічного планування провів дослідження «Маркетингова стратегія і рівень прибутку», в якому розглядався вплив найбільш значущих змінних на рівень прибутку компанії. З’ясувалося, що графік залежності рентабельності від частки ринку виглядає наступним чином (рис. 1.3).

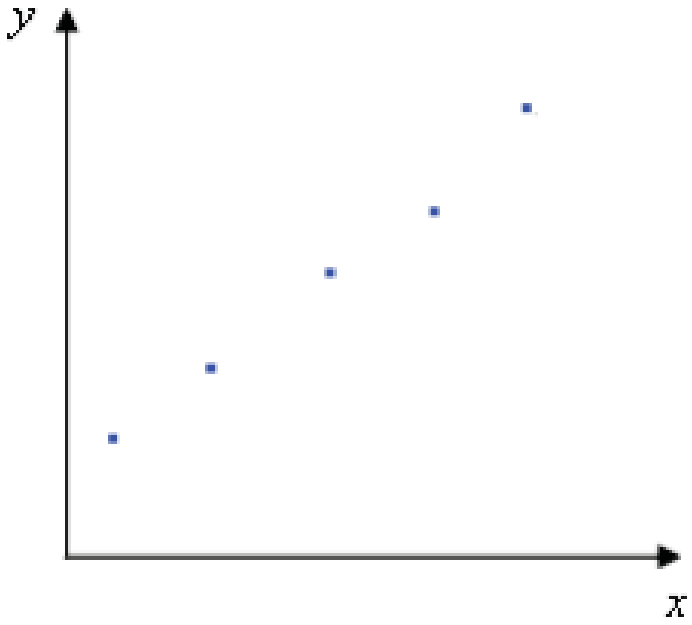


Рисунок 1.3 – Графік залежності рентабельності від частки ринку

Неозброєним поглядом видно, що це пряма, однак її точні параметри допомагає встановити регресійний аналіз.

1.1.6 Дискримінантний аналіз

Дискримінантний аналіз – це інструмент статистики, який використовується для прийняття рішення про те, які змінні виникають при поділі наборів даних [8]. Наприклад, якийсь дослідник в галузі освіти вирішує дослідити, які змінні відносять випускника середньої школи до однієї з трьох категорій: (1) поступає в коледж, (2) що поступає в професійну школу або (3) відмовляється від подальшої освіти або професійної підготовки. Для цієї мети дослідник може зібрати дані про різні змінні, пов'язані з учнями школи: стать, вік, успішність, матеріальний стан сім'ї. Після випуску більшість учнів має потрапити в одну з названих категорій. Потім можна використовувати дискримінантний аналіз для визначення того, які змінні дають найкраще передбачення вибору учнями подальшого шляху. Наприклад, можна математично визначити, що учні з низькою успішністю і низьким достатком в сім'ї швидше за всіх потрапляють в 3-ю категорію. Ще приклад: є дані про клієнтів / споживачів, яких можна розділити по групах (які вчинили повторну покупку – не скоїть повторну покупку; покупці марки А – покупці марки В – покупці марки С; високі ризики неповернення кредиту – низькі ризики неповернення кредиту), також є додаткова інформація про клієнтів / споживачів. Дискримінантний аналіз дозволяє з'ясувати, чи дійсно групи розрізняються між собою, і якщо так, то яким чином (які змінні вносять найбільший внесок в існуючі відмінності) [9].

1.1.7 Кореляційний аналіз

Кореляційний аналіз дозволяє судити про те, наскільки схоже поводяться різні змінні. У найзагальнішому вигляді прийняття гіпотези про наявність кореляції означає, що зміна значення змінної *А* станеться одночасно з пропорційним зміною значення *Б*: якщо обидві змінні зростають,

то кореляція позитивна; якщо одна змінна зростає, а друга зменшується – кореляція негативна [10]. При вивченні кореляцій намагаються встановити, чи існує якийсь зв'язок між двома показниками в одній вибірці, або між двома різними вибірками, і якщо цей зв'язок існує, то чи супроводжується збільшення одного показника зростанням (позитивна кореляція) або зменшенням (негативна кореляція) іншого.

1.2 Основні засоби кластерного аналізу

1.2.1 Постатовка задачі кластеризації

Кластеризація – це процес розбиття заданої вибірки об'єктів (спостережень) на підмножини (як правило, непересічні), які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися [11]. Вона може бути корисною на початкових етапах проведення аналізу.

Можна виділити наступні основні завдання кластерного аналізу:

– розробка типології або класифікації;

– дослідження корисних концептуальних схем групування об'єктів;

– породження гіпотез на основі дослідження даних;

– перевірка гіпотез для визначення, чи дійсно типи (групи), виділені тим чи іншим способом, присутні в наявних даних.

Звичайною задачею кластерного аналізу є розбивка на групи всієї множини об'єктів $Q = \{q_j\}_{j=1}^{n}$, де n – їх загальне число [12].

Формальна постановка задачі кластеризації здійснюється в такий спосіб. Визначається множина об'єктів даних

$$Q = Q = \{q_j\}_{j=1}^{n} = \{q_1, q_2, ..., q_n\}. \quad (1.1)$$

Кожний об’єкт q_j може мати набір атрибутів:

$$x_j = (x_{j1}, x_{j2}, ..., x_{jm}). \quad (1.2)$$

Прикладом такої множини об’єктів може бути, наприклад, множина аварій протягом певного часу на промислових об’єктах, або об’єктах мобільного зв’язку, або електростанціях, кожна з яких характеризується набором показників (атрибутів) тривалості, категорії наслідків, ступеня пожежної і техногенної безпеки, зв’язку з часом доби, пори року, завантаження об’єкта тощо.

Розв’язком задачі кластеризації є множина сформованих кластерів

$$C = \{C_k\}_{k=1}^{g} = \{c_1, c2, ..., q_g\}, \quad (1.3)$$

де c_k – кластер, що містить схожі об’єкти із множини Q:

$$c_k = \{q_i, q_j \mid d_{i,j} < \sigma\}. \quad (1.4)$$

В цьому виразі $d_{i,j} = d(x_i, x_y)$ – міра близькості між об’єктами q_i, q_j, яка визначається з урахуванням наборів (векторів) їх атрибутів x_i, x_j [13].

Величина σ визначає порогове значення для міри близькості між об’єктами.

Усі алгоритми (методи) кластеризації в цілому розділяють на ієрархічні й неієрархічні алгоритми.

Ієрархічні алгоритми кластерного аналізу у свою чергу розділяють на агломеративні (що збирають) й дивізимні (що розділяють) [14].

На рисунку 1.4 наведено результат кластеризації, на якому можна побачити, що було сформовано три кластери (виділені різними кольорами), що характеризують найбільш наближені між собою точки.

Таким чином, в результаті отримуємо кластери, які характеризують свою групу і надають нам можливість працювати не з величезною кількістю різної інформації, а з одним представником, за яким згодом можна буде зробити висновок про весь кластер і кожну його складову.

Отже, на початку проведення аналізу дуже корисним буде провести кластеризацію, яка надалі спростить розрахунки. Набагато легше виділити групи схожих об’єктів, вивчити їх особливості і побудувати для кожної групи окрему модель, ніж створювати одну загальну модель на всіх даних.

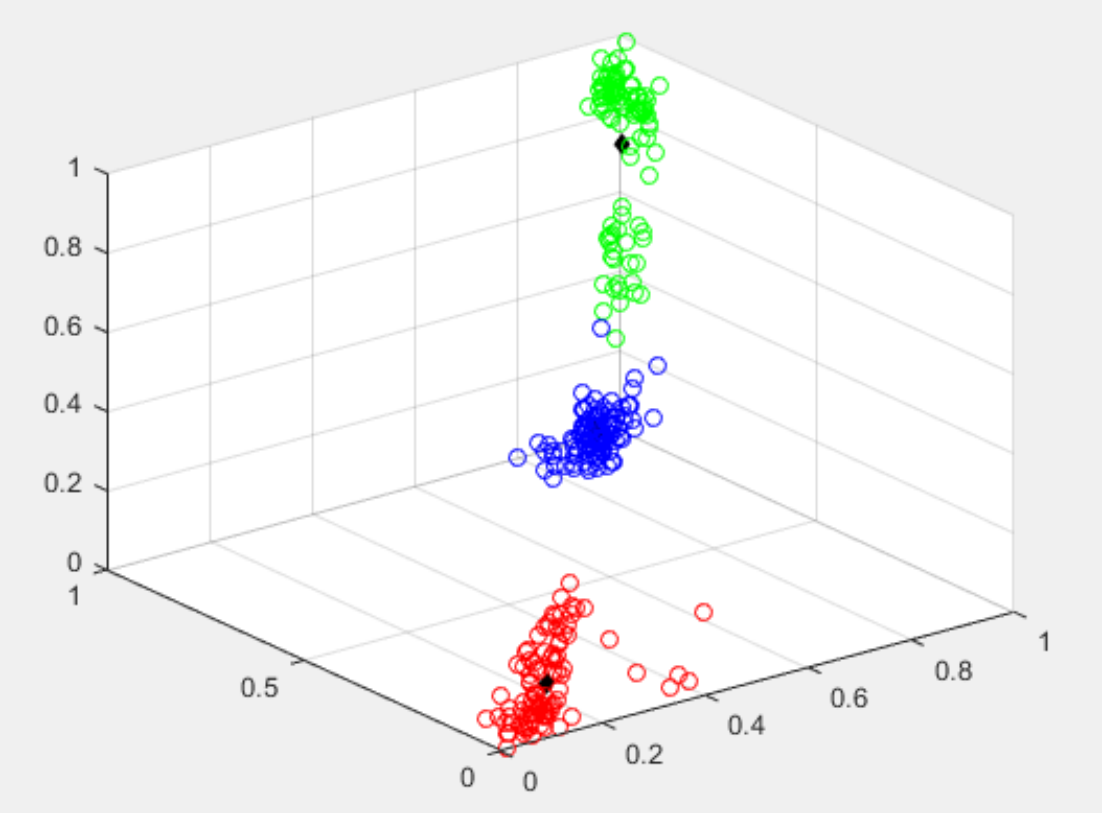


Рисунок 1.4 – Приклад результату кластеризації тестових даних у форматі 3D

1.2.2 Основні поняття, які використовуються в задачах кластеризації

До основних засобів, які використовуються в задачах кластеризації, відносяться відстані, метрики і міри близькості.

Ці засоби, іноді суттєво вирізняючись між собою, використовуються в різноманітних алгоритмах (методах) кластеризації, яких загалом відомо декілька сотень.

1.2.3 Метрики й відстані

Метрика – це функція на парах елементів множини. Метрики належать до основних засобів формування альтернатив класів у задачах розпізнавання і кластеризації, де оцінки приналежності до класу обчислюються однозначним способом [15].

Ці функції (метрики) використовуються для обчислення відстаней між об'єктами q_i, q_j і у термінах метричних просторів і є відстанями між об'єктами цих просторів. Ці обчислені відстані є мірами близькості цих об'єктів.

Якщо набори атрибутів $x_j = (x_{j1}, x_{j2}, ..., x_{jm})$. можна представити у вигляді m-вимірних числових векторів, які є елементами m-вимірного метричного простору дійсних чисел, то кожна змінна з набору x_j може приймати значення із множини дійсних чисел, що дає можливість застосувати відповідні метрики метричного простору дійсних чисел для групування (кластеризації) об'єктів [16].

Інакше кажучи, множина об'єктів $Q = \left\{ q_j \right\}_{j=1}^{n}$ повністю визначається множиною даних X_Q, яка є підмножиною m-вимірного простору дійсних чисел:

$$X_Q = \{x_j\}_{j=1}^{n} \subseteq R^m, \tag{1.5}$$

$$x_j = (x_{j1}, x_{j2}, ..., x_{jm}), j = \overline{1,n}. \tag{1.6}$$

Для множини даних X_Q ожна визначити m-вимірний вектор середніх значень точок даних $\bar{x}$ і коваріаційну матрицю S розмірності $m \times m$, які використовуються в розв'язаннях задач кластеризації:

$$\bar{x} = (\bar{x}_k)_{k=1}^{m} = (\bar{x}_{1,} \bar{x}_1, ..., \bar{x}_{jm}), \tag{1.7}$$

$$\bar{x}_k = \frac{1}{2} \sum_{j=1}^{n} x_{jk}, \tag{1.8}$$

$$S = \left\| \frac{1}{n-1} \sum_{j=1}^{n} (x_j - \bar{x})(x_j - x)^T \right\|. \tag{1.9}$$

Ненегативне значення $d_{i,j} = d(x_i, x_y) d_{i,j} = d(x_i, x_j)$ називається відстанню між елементами $x_i, x_j \, x_i, x_j$, якщо виконуються наступні чотири умови:

$$\forall x_{i,} x_y : d(x_i, x_j) > 0, \tag{1.10}$$

$$x_i = x_j \Leftrightarrow d(x_i, x_j), \tag{1.11}$$

$$d(x_i, x_j) = d(x_j, x_i), \tag{1.12}$$

$$d(x_i, x_j) \le d(x_j, x_i) + d(x_i, x_j). \quad (1.13)$$

Тобто:

а) $d(x_i, x_j) > 0$, для всіх $x_i, x_j\, x_i, x_j$ – відстань завжди невід'ємна;

б) $d(x_i, x_j) = 0, d(x_i, x_j) = 0$ тоді й тільки тоді, коли $x_i = x_j\, x_i = x_j$– об'єкти співпадають;

в) $d(x_i, x_j) = d(x_j, x_i)$, $d(x_i, x_j) = d(x_j, x_i)$ – відстань від об'єкта q_i до об'єкта q_j завжди дорівнює зворотній відстані від об'єкта q_j до об'єкта q_i;

г) $d(x_i, x_j) \le d(x_i, x_j) + d(x_i, x_j),$ – відстань від об'єкта q_i до об'єкта q_j завжди менша або дорівнює сумі відстаней між ними та третім об'єктом.

Якщо відстань $d(x_i, x_j)$ менше деякого значення σ, то приймають рішення, що елементи близькі і їх містять в один кластер [17].

Велика кількість алгоритмів розв'язання задачі кластеризації використовують у якості формату вхідних даних матрицю відмінності (відстаней) *D* [7]. Рядки й стовпці матриці відповідають елементам множини X_Q. Елементами матриці є значення $d(x_i, x_j)$ у рядку *i* і стовпці *j*. Очевидно, що на головній діагоналі значення будуть дорівнювати нулю як приведено в таблиці 1.1

Таблиця 1.1 – Зразок матриці відстаней

x_j	Елемент	1	2	3	4	5
0,471666	1	0	1,662236	1,285075	0,924313	1,52347
-1,19057	2	1,662236	0	0,377161	2,586548	0,138766
-0,81341	3	1,285075	0,377161	0	2,209388	0,238394
1,395979	4	0,924313	2,586548	2,209388	0	2,447782
-1,0518	5	1,52347	0,138766	0,238394	2,447782	0

Іноді для підготовки вхідних даних використовують більш спеціальні методи, наприклад, з використанням карт самоорганізації [18].

Для розглянутої множини об’єктів із простору R^m при кластеризації найчастіше використовують наступні відстані.

Евклідова відстань (Euclidian Distance):

$$d_{E1}(x_{i,}x_j) = \sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2}. \quad (1.14)$$

Цю відстань часто підносять до квадрата, щоб додати більше ваги більш віддаленим одне від одного об’єктам, тобто

$$d_{E2}(x_j x_i) = \sum_{k=1}^{m}(x_{ik} - x_{jk})^2. \quad (1.15)$$

Відстань за Хемінгом:

$$d_H(x_i x_j) = \sum_{k=1}^{m}\left|x_{ik} - x_{jk}\right|. \quad (1.16)$$

Ця відстань є просто середньою різниць по координатах. У більшості випадків дана міра відстані приводить до таких же результатів, як і для звичайної відстані Евкліда, однак для неї вплив окремих більших різниць (викидів) зменшується, тому що вони не підносяться до квадрата).

Відстань Чебишева:

$$d_\infty(x_i x_j) = \max_{k=1,m}\left|x_{ik} - x_{jk}\right|. \quad (1.17)$$

Ця відстань може виявитися корисною, коли бажають визначити два об’єкти як «різні», якщо вони різняться по якій-небудь одній координаті (яким-небудь одним виміром).

Відстань Махаланобіса (Mahalanobis Distance):

$$d_M(x_{i,}x_j) = \sqrt{(x_i - x_j)^{S-1}(x_i - x_j)^T}. \quad (1.18)$$

Дана міра відстані використовує коваріаційну матрицю, яка зображена на рисунку 1.5. Але вона погано працює, якщо коваріаційна матриця вираховується на всій множині вхідних даних.

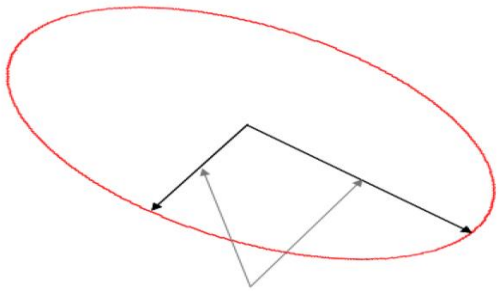


Рисунок 1.5 – Вектори коваріаційної матриці і еліпс множини рівновіддалених від центра точок в сенсі відстані Махаланобіса

Дана міра відстані припускає незалежність між випадковими змінними, що передбачає відстань в ортогональному просторі. В практичних додатках ці змінні не є незалежними.

У багатьох випадках замість відстані як міри близькості використовується значення косинуса кута між двома векторами

$$\cos\varphi = \frac{x_i \bullet x_y}{|x_i| \bullet |x_y|}. \quad (1.19)$$

Для визначення відстаней між кластерами часто використовується розрахунок відстаней до об’єктів або елементів, які входять в ці кластери. Відстань до всіх елементів бази можна оцінити шляхом обчислення відстані до одного з елементів. Це значно скорочує час процедур порівняння, тому що внутрішні відстані можна обчислити заздалегідь на попередньому етапі.

Найбільш застосовуваними для розрахунку відстаней між групами (кластерами) є відстані за принципом «близького сусіда», за принципом «далекого сусіда» і відстань між «центрами ваги» груп (кластерів) [19].

1.2.4 Міри близькості

Відстань $d(x_i, x_j)$ є мірою близькості об’єктів. Але існують і інші міри близькості об’єктів.

Для визначення подібності об’єктів у теорії кластеризації й розпізнавання використовується також більш загальне поняття величини міри близькості $r(q_i, q_j)$ об’єктів.

Приклад міри близькості:

$$r(q_i, q_j) = \frac{v(q_{i,} q_j)}{v_0}, \tag{1.20}$$

де $v(q_i, q_j)$ – число однакових ознак (атрибутів) у порівнюваних об’єктів q_i, q_j а величина $v_0\, v_0 -$ загальне число ознак (атрибутів) або число відповідностей ознак.

Умови, яким відповідають міри близькості, аналогічні розглянутим раніше для відстаней:

$$\forall q_{i,} q_y : r(q_i, q_j) > 0, \tag{1.21}$$

$$q_i = q_j \Leftrightarrow r(q_i, q_j), \tag{1.22}$$

$$r(q_i, q_j) = r(q_j, q_i), \tag{1.23}$$

$$r(q_i, q_j) \le r(q_j, q_i) + r(q_i, q_j), \tag{1.24}$$

$$r(q_i, q_j) \le r(q_j, q_i) + r(q_i, q_j). \tag{1.25}$$

Для міри близькості, на відміну від відстані, не завжди потрібне обов'язкове виконання умови. Ця умова обов'язкова лише для тих мір близькості, які є відстанями.

При виконанні нерівності $v(q_i, q_j) < \sigma$ об'єкти із множини Q розглядаються як близькі й містяться в один кластер. Інакше об'єкти містяться в різні кластери.

Було розглянуто міри близькості, які є відстанями: Евклідова відстань, відстань за Хемінгом, відстань Чебишева, відстань Махаланобіса, тощо.

1.3 Кластеризація часових рядів та її особливості

Задача кластеризації полягає в розбитті вихідної сукупності аналізованих об'єктів на окремих групах (кластерах) таким чином, щоб різниця між об'єктами всередині групи була мінімальними, а різниця між групами максимальних.

Подібно кластерному аналізу інших об'єктів, якість кластеризації тимчасових рядів визначається вибором наступних важливих елементів:

– міра відстані, що описує ступінь відмінностей між рядами;

– алгоритм розбиття рядів на кластери.

Існує велика кількість алгоритмів для виконання кластеризації, але найчастіше (по крайней мере, для відносно невеликих обсягів даних) на практиці застосовують ієрархічну кластеризацію, метод k-середніх і розбиття по медоідам (Partition Around Medoid, PAM). При цьому виділяють наступні три підходи для обчислення заходів відстані між тимчасовими рядами:

За вихідними даними (raw data-based approach): для обчислення міри відстані використовують вихідні значення аналізованих часових рядів (часовий або частотний областях). Реєстрація значень порівнюваних рядів, як правило, виконуватися через однакові проміжки часу, проте довжина цих рядів не обов'язково повинна бути однаковою.

За описовим ознаками (feature-based approach): в рамках цього підходу спочатку виконується зниження розмірності шляхом вкладення (embedding) аналізованих часових рядів в простір, утворене їх описовими ознаками. Обчислення відстані між окремими рядами далі виконується за значеннями цих ознак.

За результатами підгонки моделей: в рамках цього підходу робиться припущення, що аналізовані тимчасові ряди були породжені процесом, який можна апроксимувати моделлю з певним набором параметрів. Схожими вважаються ряди з близькими значеннями оцінених параметрів (наприклад, коефіцієнтів) такої моделі. Крім параметрів моделі для розрахунку відстані між окремими рядами можуть використовуватися також залишки – або в початковому вигляді, або після зниження їх розмірності як описаний вище другого підходу.

Оскільки кластерний аналіз належить до методів навчання без вчителя (unsuprevised learning), то практично неможливо зробити об'єктивний висновок про «правильності» одержуваних з його допомогою рішень. Як правило, на практиці вибирають той результат, який «має сенс» з точки зору розв'язуваної задачі. Проте, існує цілий ряд метрик, які намагаються описати якість кластеризації кількісно. Це дає можливість порівняти різні рішення і

вибрати найбільш «оптимальне» з них, в зв’язку з чим розрахунок подібних метрик якості часто є однією зі стадій кластерного аналізу.

За аналогією з центроїдами або медоідамі, одержуваними при кластеризації інших об’єктів, корисним результатом кластеризації часових рядів є виділення так званих «Прототипів» (prototypes), тобто найбільш представницьких рядів для кожної зі знайдених груп. Візуалізація прототипів допомагає наочно уявити найбільш типову форму часових рядів в кожному кластері. Крім того, прототипи можна використовувати для класифікації, тобто для віднесення нових часових рядів до тієї чи іншої групи. Вибір способу обчислення прототипів тісно пов’язаний з вибором міри відстані і алгоритму кластеризації. Існує кілька таких способів, однак найчастіше використовується або просте усереднення всіх рядів в кластері по кожній часовій позначці, або вибір такого тимчасового ряду з кластера, який максимально близький (відповідно до обраної міру відстані) до всіх інших рядах в цьому кластері.

Дані часового ряду є однієї з найпоширеніших форм даних, з якими зустрічаються у великій різноманітності сценаріїв, таких як фондові ринки, дані датчика, контроль відмови, контроль стану машини, екологічні застосування або медичні дані. Проблема кластеризації знаходить численне застосування в областях часового ряду, таке як визначення груп об’єктів з подібними тенденціями. Кластеризація часового ряду має численні застосування в різноманітних проблемних областях.

Дані часового ряду перебувають у межах класу контекстних подань даних. Багато видів даних, таких як дані часового ряду, дискретні ряди й просторові дані знаходяться у цьому класі.

Працюючи з часовими рядами можна зіткнутися з типовими складнощами data science, такими, як велика розмірність вхідних даних, наявність шуму та пропущені дані. Розглядаючи, безпосередньо, кластеризацію часових рядів слід звернути увагу на додаткові незручності:

– ряди можуть містити різну кількість відліків;

– більше ступенів свободи для визначення схожості одного об’єкта на інший;

– при виборі метрик та статистик слід звертати уваг на локальну залежність даних.

Важливою задачею, що вирішується при роботі з часовим рядом є визначення близькості, що буде використовуватись при кластеризації:

– близькість за часом. Полягає у тому, що необхідно знайти особливі точки і інтервали, що відповідають одне одному у часі, повна відповідність не вимагається, головне загальна схожість;

– близькість за формою. Полягає у знаходженні однакових характерні особливості, які можуть бути рознесені за часом або розтягнуті та таке інше;

– близькість за структурою. Полягає у знаходженні послідовностей з однаковим законом змінювання.

Часові ряді розглядаються і аналізуються з метою:

– визначення природи ряду;

– прогнозування майбутніх значень ряду.

Як вже зазначалося дані можуть бути в неідеальній, для аналізу, формі. Часові ряди також володіють недоліками, містять аномальні значення, що вимагає проведення попередньої обробки і згладжуванні ряду. Якщо цього не зробити аномальні дані можуть призвести до спотворення результатів, що будуть отримані.

1.3.1 Атрибути часових рядів

Часові дані містять два види атрибутів:

– контекстний атрибут;

– поведінковий атрибут.

Для випадку даних часового ряду контекстний атрибут відповідає виміру часу. Ці атрибути забезпечують контрольні точки, у яких виміряються

поведінкові значення. На рисунку 1.6 приведено порівняння часових рядів на підставі конкретних і поведінкових атрибутів. Мітки часу могли відповідати фактичним тимчасовим вартостям, у яких виміряються точки даних, або вони могли відповідати індексам, у яких виміряються ці значення.

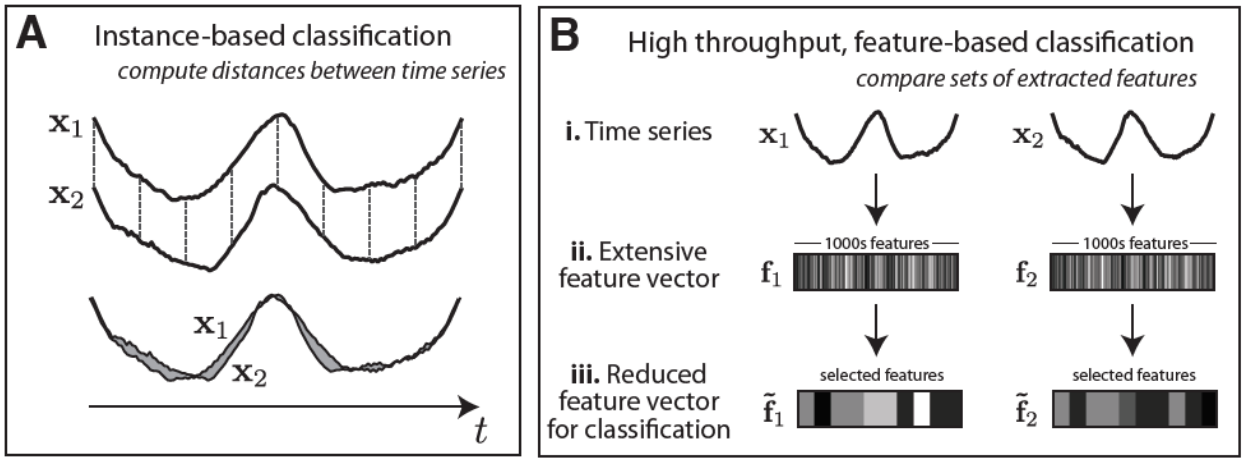


Рисунок 1.6 – Порівняння часових рядів на підставі контекстних і поведінкових атрибутів

Поведінковий атрибут може відповідати будь-якому виду поведінки, яка виміряється в контрольній точці. Деякі приклади включають значення біржевих даних, виміру датчика, такі як температура, інші медичні часові ряди тощо.

1.3.2 Вибір міри близькості кластеризації часових рядів

Визначення кластерів часового ряду надзвичайно складно через труднощі при визначенні близькості через різний часовий ряд, який може масштабуватися й перекладатися по-іншому й на часових й на поведінкових розмірах. Тому поняття близькості є дуже важливим для кластеризації даних часового ряду. Зверніть увагу, що, як тільки міра близькості була визначена для даних часового ряду, її можна розглядати як абстрактний об’єкт, на якім

може використовуватися множина заснованих на подобі методів, таких як спектральні методи або методи поділу.

Дані часового ряду дозволяють різноманітні формулювання для процесу кластеризації, залежно від того, чи кластеризируються ряди на основі їх кореляцій онлайн, або чи кластеризируються вони на основі їх форм [19]. Перший звичайно виконується з підходом онлайн на основі маленького вікна минулої історії, тоді як останній звичайно виконується з офлайновим підходом до всього ряду. На рисунку 1.7 зображені часові ряди в медицині.

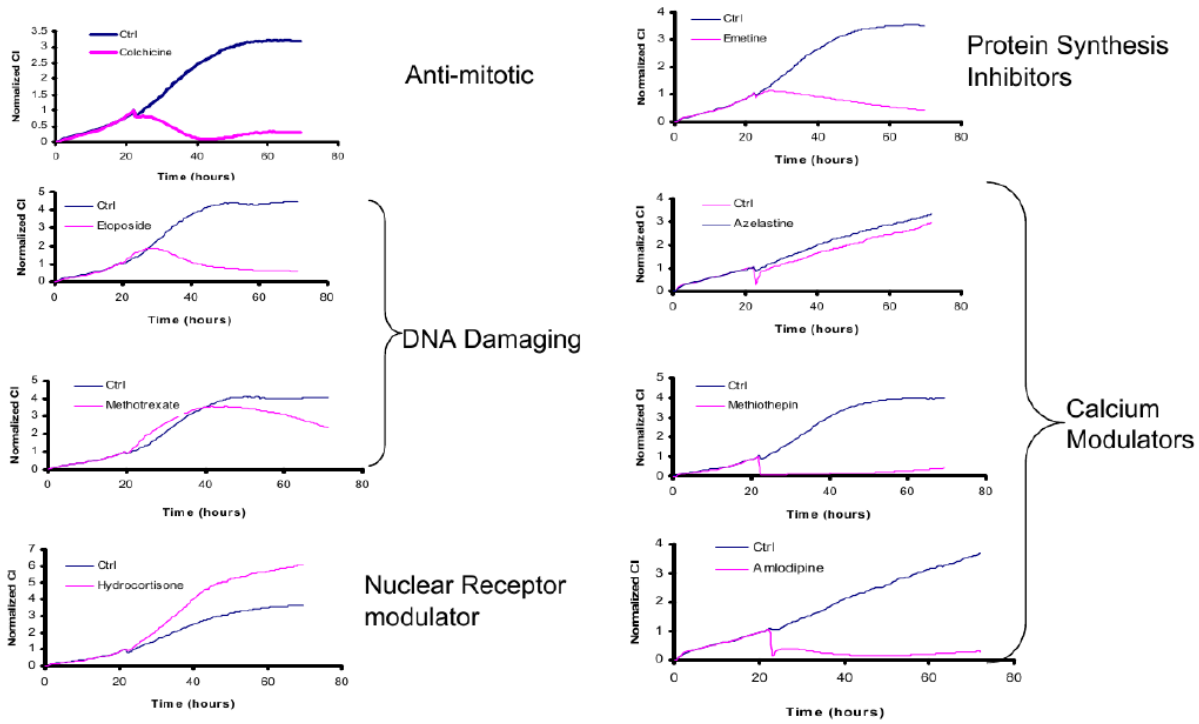


Рисунок 1.7 – Часові ряди в медицині

Однак деякі певні властивості, які є частиною природи даних часового ряду – такі, як висока розмірність, присутність шуму й висока кореляція – ставлять унікальні проблеми перед розробкою ефективних алгоритмів кластеризації. У часовому ряді надто важливо вирішити, яка близькість важлива для кластеризації:

– близькість у часі;

– близькість у формі;

– близькість у зміні даних.

Відповідно повинні бути обрані алгоритм, що відповідає кластеризації, й відповідна міра близькості. Наприклад, Евклідова відстань відображає подобу в часі, у той час як міра динамічної трансформації часової шкали відображає подобу у формі. Інші підходи, наприклад, засновані на моделі прихованих марківських процесів, застосовуються, коли має значення близькість в зміні часових даних [20].

Значна різниця між кластеризацією даних часового ряду й кластеризацією простих об’єктів в Евклидовому просторі - те, що часовий ряд, який буде кластеризуватися, не завжди може мати рівні довжини послідовностей. Коли усі часові ряди мають рівну довжину, можуть бути застосовані стандартні методи кластеризації шляхом представлення кожного часового ряду як вектора й використання традиційної нормованої відстані. З таким підходом вчасно може бути використана тільки близькість у часі, у той час як близькість у формі й близькість в зміні ігноруються.

1.3.3 Кластеризація із заповненням пропусків вибірковими статистиками

Дані не завжди приходять у зручному для обробки, а саме кластеризації, вигляді. Часто виникають ситуації, коли частина інформації відсутня. Це можуть бути пропущені відповіді у анкетуванні, коли різні люди не відповіли на деякі запитання. Можлива ситуація втрати або пошкодження частини отриманої інформації, збій у роботі датчиків, помилки в роботі програми. І такі ситуації це не виключення, а найчастіше саме так і трапляється. Тому виникає необхідність перейти до даних, які не мають пропусків, для подальшої роботи з ними.

Підхід, що передбачає заповнення невідомих вибірковими статистиками (середнє, медіана, тощо), спирається на, так званий, « наївний»

підхід, який полягає у припущені, що взаємозв’язок між змінними в даному наборі даних відсутній.

В даному випадку, до недоліків слід віднести той факт, що після заповнення розподілення даних буде відрізнятися від похідних значень, що деякі дані були замінені на «штучні», безперечно призведе до спотворення результатів. Також можна віднести зменшення дисперсії.

Але попри це, на даних з пропусками приблизно 30% можна сказати, що такий вид обробки пропусків дає хороші результати, адже результати кластеризації подібні до еталонних.

Так, якщо говорити про задачі кластеризації, то практичні результати показують, що за 30% пропусків можна досягти припустимих результатів кластеризації. Але, все ж слід враховувати втрату повної достовірності і точності такого аналізу.

1.3.4 Виключення рядків з наявністю пропущених значень

Це метод, що легко реалізувати, але він може призвести до суттєвої втрати важливих даних. Його можна використовувати лише тоді, коли пропуски в даних розміщені випадковим чином і їх доволі мало, щоб вплинути на кінцевий результат.

Даний метод з видалення всіх рядків, які містять пропуски, є найгіршим варіантом. Він можливий лише у випадках коли вибірка містить мінімальну кількість пропусків, або тоді коли було попередньо проведено інший вид обробки і відбувається видалення залишків пустих значень.

Такий підхід не дає коректних результатів, якщо кількість пропусків більше ніж 30%, за таких даних не вдається побудувати кластері побідні еталонним. Однак, слід зазначити, що у разі, коли кількість пропусків знаходиться близько 10% формування кластерів відбувається досить схоже на необхідний результат.

Але все ж, даний метод не підходить для аналізу даних зі значною кількістю пропусків, особливо якщо важлива велика точність розрахунків і певність результатів. Це найпростіший метод, який може бути використано у разі крайньої необхідності, де потреба швидкості отримання результатів перевищує їх точність.

1.3.5 Заповнення пропусків з урахуванням структури зв’язків

Попередній метод передбачав відсутність зав’язків між параметрами, це так званий «наївний» метод. Альтернативою йому можна назвати метод, що враховує зв’язки між параметрами. Наприклад між такими параметрами медичної оцінки стану людини, як trestbps (артеріальний тиск у стані спокою) та віком пацієнта можна встановити кореляційний взаємозв’язок. Цей факт може допомогти відновити пропущені дані в параметрі, використовуючи рівняння простої регресії [21].

Як і попередній метод потребує випадкове розподілення пропусків і залежить від правильно обраного методу регресійного аналізу.

Серед простих методів боротьби з пропущеними значеннями при кластеризації, шляхом їх попередньої обробки, найкращим вважається метод боротьби з пропусками з урахуванням взаємозв’язків між полями, але не завжди він дає переваги при використанні, не всі набори можуть мати очевидні зв’язки тому значних переваг, у порівнянні з попереднім методом заміни на середні значення, не можна добитися.

Отже, за результатами можна сказати, що у разі необхідності позбавитися від незначної кількості пропусків, можна використовувати метод їх видалення, у випадках, якщо пропусків близько 30% цей варіант розглядати не варто, слід звернути увагу на два інші методи, що є більш складними, але при цьому збільшують точність відновлення, та якість

подальшої кластеризації. Адже аналізуючи отримані графіки і таблиці було зроблено висновок, про подібність результатів до еталонних значень.

1.4 Постановка задачі дослідження

Говорячи про актуальність і затребуваність даної теми слід дослідити способи застосування даних методів та розробки в даній сфері, зацікавленість серед дослідників. Зважаючи на сучасні потреби все більш актуальними і затребуваними є розробки в галузі аналізу накопичених даних, адже ми живемо в той час коли кількість інформації, якою ти володієш, вже менш значима ніж її якість і можливість обробити, зробити висновки на цьому підґрунті. Можна сказати, що класичні методи кластеризації є досить широкою темою для досліджень і модифікації для вирішення різноманітних задач та різних типів даних.

Отже кластеризація становить інтерес для попередньої обробки даних, з метою подальшого прогнозування та більш зручного їх подальшого аналізу. Отримавши необхідні групи і відповідні їм центроїди можна робити подальші висновки та прогнози вже з конкретними представниками, а не з усім набором даних, що особливо актуально в умовах безкінечно зростаючого об’єму інформації. Такий підхід дозволяє: більш усвідомлено обробляти дані, шляхом використання для кожного кластеру найбільш підходящого алгоритму аналізу; виявлення новизни, шляхом виділення об’єктів, що не потрапили до жодного з кластерів; провести стиснення, виділивши найбільш типових представників, за умов збитковості даних.

Алгортитм дій наступний:

Крок 1. Встановити трекінговий код на сайт.

Крок 2. Зібрати необхідні дані та сформувати датасет.

Крок 3. Вигрузити дані в систему візуальної аналітики.

Крок 4. Візуалізувати дані як множину на числовій осі.

Крок 5. Визначити середнє квадратичне відхилення.

Крок 6. Провести кластеризацію даних методом *k*-means.

Метою є дослідження нестохастичних методів прогнозування та доцільність їх використання в реальних маркетингових задачах. Вирішується задача кластеризації та прогнозування багатовимірних часових рядів. Аналіз та оцінка адаптованих методів кластеризації для вирішення подібного роду задач, та висновок про їх переваги перед методами попереднього позбавлення від пропусків пере процесом кластеризації.

Об’єктом дослідження є дані з системи Google Analytics, які представлені у вигляді часових рядів

Розбиття множини на групи подібних об’єктів – це дуже потужний механізм підготовки даних до подальшого аналізу, але існує проблема обробки даних, що не є повними, тобто мають пропуски.

Предмет дослідження – це методи підготовки та обробки вхідних даних, що містять пропущені значення, для їх подальшого прогнозування, аналізу та використання в задачах кластеризації; розгляд адаптованого класичного методу кластеризації для вирішення проблеми неповних даних.

Завданням даної роботи є аналіз переваг і недоліків кожного з методів, спрямованих на відновлення даних, для визначення доцільності використання в задачах кластеризації при вирішенні маркетингових завдань. Більш детально були розглянуті методами нечіткої кластеризації FCM, метод видалення всіх рядків, що містять пропуски, заповненням пропусків вибірковими статистиками, заповнення пропусків з урахуванням структури зв'язків.

Для вирішення маркетингової задачі необхідно:

– встановити трекінговий код Google Analytics на сайті;

– зібрати, структурувати та вигрузити необхідні дані;

– реалізувати методи кластерного аналізу.

2 МАТЕМАТИЧНІ МОДЕЛІ МЕТОДІВ КЛАСТЕРИЗАЦІЇ БАГАТОВИМІРНИХ ЧАСОВИХ РЯДІВ ДЛЯ ВИРІШЕННЯ МАРКЕТИНГОВИХ ЗАДАЧ

2.1 Методи кластеризації одновимірних часових рядів для інтелектуального аналізу

Дані часових рядів є однією з найпоширеніших форм даних, що зустрічаються у широкому спектрі такі, як фондові ринки, дані датчиків, моніторинг несправностей, моніторинг стану машини, застосування навколишнього середовища або медичні дані. Проблема кластеризації знаходить численні програми в діапазон часових рядів, такий як визначення груп сутностей зі схожими тенденціями. Кластеризація часових рядів має численні програми в різних проблемних областях:

– фінансові ринки: На фінансових ринках значення запасів представляють часові ряди, які постійно змінюються з часом. Кластеризація таких часових рядів може дати численні уявлення в тренди базових даних;

– медичні дані: Різні види медичних даних, такі як показники ЕЕГ, мають форму часу серії. Кластеризація таких часових рядів може дати розуміння загальних форм в даних. Ці загальні форми можуть бути пов’язані з різними видами захворювань;

– застосування наук про Землю: численні програми в галузі науки про Землю, такі як температура або тиску, відповідають рядам, які можна видобувати, щоб визначити часті тенденції в даних. Вони можуть дати уявлення про загальні кліматичні тенденції в даних;

– моніторинг стану машини: численні форми машин створюють дані датчиків, які надають безперервне уявлення про стани цих об’єктів. Вони можуть бути використані для того, щоб подати ідею основних тенденцій;

– просторово-часові дані: Дані траєкторії можна вважати формою багатовимірних часових рядів дані, в яких координати X та координати Y об’єктів відповідають безперервно різні серії.

Тенденції цих серій можна використовувати для того, щоб визначити важливе кластери траєкторій у даних. Дані часових рядів входять до класу контекстних подань даних. Багато видів даних такі як дані часових рядів, дискретні послідовності та просторові дані потрапляють до цього класу. Контекстні дані містять два типи атрибутів:

– контекстний атрибут: для даних часових рядів це відповідає часовому виміру. Ці атрибути забезпечують контрольні точки, в яких вимірюються поведінкові значення. мітки часу можуть відповідати фактичним значенням часу, за яких вимірюються точки даних, або вони могли б відповідати індексам, за якими ці значення вимірюються;

– атрибут поведінки: це може відповідати будь-якій поведінці, яка вимірюється контрольна точка.

Деякі приклади включають значення цінних паперів, вимірювання датчиків, такі як температура та інші медичні часові ряди.

Визначення кластерів часових рядів є надзвичайно складним через складність у визначенні подібності між різними часовими рядами, які можна масштабувати та перекладати по-різному як за часовим, так і за поведінковим вимірами. Тому поняття подібності є дуже важливий для кластеризації даних часових рядів, і в цій главі буде присвячений розділ проблемі вимірювань подібності часових рядів. Зверніть увагу, що як тільки показник подібності визначено для даних часових рядів, його можна розглядати як абстрактний об’єкт, на якому застосовуються різноманітні методи такі як спектральні методи або методи розділення.

2.1.1 Алгоритм fuzzy *k*-means (*c*-means)

Нечітка кластеризація – це клас алгоритмів кластерного аналізу, в яких розподіл точок даних для кластеризації є не «чітким» («0 або 1», «так або ні»), а «нечітким» (в тому ж значенні, що й у нечіткій логіці) [22].

Запропонована множина алгоритмів нечіткої кластеризації, заснованих на мінімізації цільової функції *E*. Знаходження матриці нечіткої розбивки з мінімальним значення критерію *E* становить собою задачу нелінійної оптимізації, яка може бути вирішена різними методами. Найбільш відомий і часто застосовуваний метод розв'язку цієї задачі алгоритм нечітких *k*-середніх (*k*-means), в основу якого покладений метод невизначених множників Лагранжа.

Алгоритм Fuzzy *c*-means є узагальненням алгоритму *k*-means. Основна відмінність алгоритму – кластери представляються нечіткими множинами. Кожний об'єкт належить кластеру з різним ступенем приналежності.

Нечітка самоорганізація (FCM – Fuzzy Classifier Means) відрізняється від детермінованого методу *k*-means тим, що для кожного з об'єктів множини додатково оцінюються значення його функції приналежності до кожного із кластерів. Це дозволяє більш ретельно враховувати відмінності між кластерами, особливо для граничних точок сформованих груп. Крім того, традиційний *k*-means не може використовуватися у випадку кластерів, що перекриваються. При такій постановці задачі класифікації доцільно використовувати метод FCM [23].

Позначимо μ_{ij} – значення матричної функції приналежності об'єкта x_i кластеру X_j з центром (центроїдом) m_j:

$$\mu_{ij} = \mu(x_i, m_j). \tag{2.1}$$

Також позначимо міру близькості об'єкта x_i до центру m_j:

$$r_{ij} = r(x_i, m_j). \tag{2.2}$$

Метод FCM мінімізує нелінійну цільову функцію:

$$E_d = \sum_{i=1}^{n}\sum_{j=1}^{k} \mu_{ij}^{\ d}\ r_{ij}^{\ 2} \to \min, d \geq 1, \tag{2.3}$$

де n – потужність (кількість елементів) початкової множини X_Q;

k – кількість кластерів;

d – ваговий коефіцієнт (типове значення d=1.5).

Для матричної функції приналежності μ виконуються обмеження

$$\mu_{ij} = \mu(x_i, m_j) \,|\, \forall_i = 1, n : \sum_{j=1}^{k} \mu_{ij} = 1, 0 \leq \mu_{ij} \leq 1. \tag{2.4}$$

Це означає, що для кожної точки сумарне значення функції приналежності $\mu(x_i, m_j)$ по множині кластерів дорівнює 1.

Матрична функція приналежності об’єкта μ може бути визначена різними способами, наприклад, на підставі прийняття за умову нормального закону $f_{ij} = \frac{1}{\sigma}\varphi(\frac{r(x_i, m_j)}{\sigma})$ статистичного розподілу мір близькості $r(x_i, m_j)$:

$$\mu_{i,j} = \frac{f_{ij}}{\sum_{j=1}^{k} f_{ij}}. \tag{2.5}$$

Ініціалізацію початкових значень центроїдів кластерів m_j у методі FCM звичайно виконують випадковим способом.

Алгоритм Fuzzy *c*-means дозволяє знайти локальний оптимум, тому виконання алгоритму з різних початкових точок може привести до різних результатів [23]. На рисунку 2.1 зображена форма кластерів в алгоритмі Fuzzy *m*-means в тривимірному просторі.

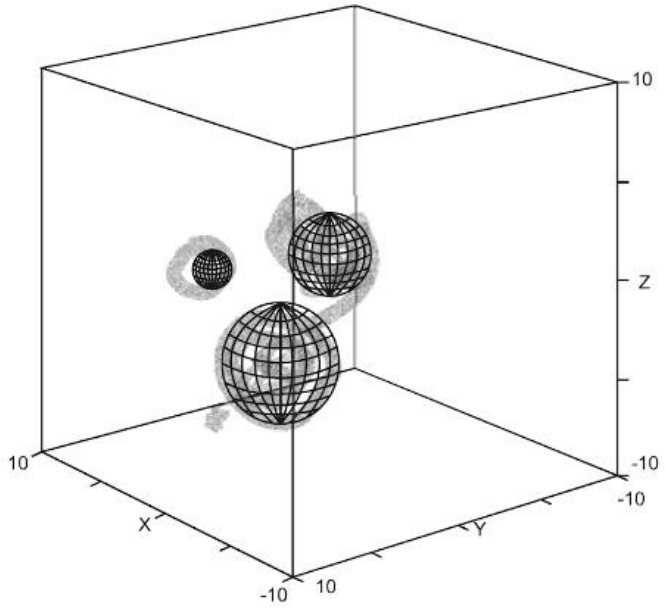


Рисунок 2.1 – Форма кластерів в алгоритмі Fuzzy *C*-Means в тривимірному просторі

На другому етапі для розв’язання задачі кластеризації значення матричної функції приналежності $\mu_{ij}\mu_{ij}$ й центроїдів m_j оновлюють ітеративно.

Якщо прийняти номер поточного кластеру (для якого визначаються нові значення μ_{ij} і m_j) за j а номер довільного кластеру за l, то ітеративні значення

$$m_j = \frac{\sum_{i=1}^{n} \mu_{ij}{}^{d} x_i}{\sum_{i=1}^{n} \mu_{ij}{}^{d}}. \qquad (2.6)$$

Ітерації методу виконують доти, поки відмінність функції приналежності по всіх кластерах між поточною й попередньою ітерацією стане незначущою, тобто буде виконано:

$$\triangle(\mu_{ij(h)}, \mu_{ij(h+1)}) < \varepsilon, \tag{2.7}$$

де $\triangle(\mu_{ij(h)}, \mu_{ij(h+1)})$ – деяка міра відмінності матриці приналежності на кроці h на кроці h+1;

$\triangle(\mu_{ij(h)}, \mu_{ij(h+1)})\mu_{ij(h)}$ – значення матриці приналежності на кроці h і на кроці h+1;

ε – апріорно задана припустима погрішність.

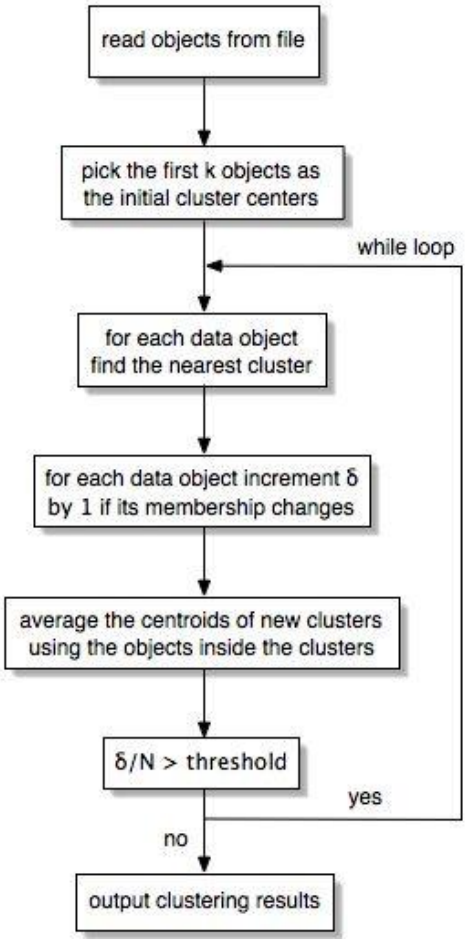
2.1.2 Алгоритм *k*-means

Найбільш простий, але в той же час досить неточний метод кластеризації в класичній реалізації. Він розбиває множина елементів векторного простору на заздалегідь відоме число кластерів *k*. Дія алгоритму таке, що він прагне мінімізувати середньоквадратичне відхилення на точках кожного кластера. Основна ідея полягає в тому, що на кожній ітерації переобчислюють центр мас для кожного кластера, отриманого на попередньому кроці, потім вектори розбиваються на кластери знову відповідно до того, який з нових центрів виявився ближчим за обраною метриці. Алгоритм завершується, коли на якийсь ітерації не відбувається зміни кластерів [24].

На першому етапі центроїди кластерів вибираються випадково або за певним правилом (наприклад, вибрати центроїди, максимізує початкові відстані між кластерами). Відносимо спостереження до тих кластерам, чиє середнє (центр ваги) до них найближче [25]. Кожне спостереження належить

тільки до одного кластеру, навіть якщо його можна віднести до двох і більше кластерам. Потім центр ваги кожного *i*-го кластера переобчислюють за таким правилом:

Таким чином, алгоритм *k*-середніх полягає в перерозрахунку на кожному кроці центроїда для кожного кластера, отриманого на попередньому кроці. На рисунку 2.2 зображена реалізація алгоритму *k*-means.




```
N:  number of data objects
K:  number of clusters

objects[N]:  array of data objects
clusters[K]:  array of cluster centers
membership[N]:  array of object memberships


kmeans_clustering( )
1   while  δ/N > threshold
2      δ ← 0
3      for i ← 0 to N-1
4         for j ← 0 to K-1
5            distance ← | objects[i] - clusters[j] |
6            if distance < dmin
7               dmin ← distance
8               n ← j
9         if membership[i] ≠ n
10           δ ← δ + 1
11           membership[i] ← n
12        new_clusters[n] ← new_clusters[n] + objects[i]
13        new_cluster_size[n] ← new_cluster_size[n] + 1
14     for j ← 0 to K-1
15        clusters[j][*] ← new_clusters[j][*] / new_cluster_size[j]
16        new_clusters[j][*] ← 0
17        new_cluster_size[j] ← 0
```

Рисунок 2.2 – Реалізація алгоритму *k*-means

Перевагою алгоритму є швидкість і простота реалізації.

Проблеми алгоритму *k*-means:

– необхідно заздалегідь знати кількість кластерів. Мною було запропоновано метод визначення кількості кластерів, який грунтувався на знаходженні кластерів, розподілених по якомусь закону (в моєму випадку все зводилося до нормального закону). Після цього виконувався класичний алгоритм *k*-means, який давав більш точні результати;

– алгоритм дуже чутливий до вибору початкових центрів кластерів. Класичний варіант працює за принципом випадкового вибору кластерів, що дуже часто є джерелом похибки. Як варіант вирішення, необхідно проводити дослідження об'єкта для більш точного визначення центрів початкових кластерів;

– алгоритм не справляється із завданням, коли об'єкт належить до різних кластерів в рівній мірі або не належить жодному.

2.1.3 Алгоритм DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise, щільнісний алгоритм просторової кластеризації з присутністю шуму) - популярний алгоритм кластеризації, який використовується в аналізі даних в якості однієї з замін методу k-середніх [26].

В алгоритмі DBSCAN прийняті наступні визначення:

а) окіл $E(x_j)\,E\left(x_j\right)$ об'єкта x_j – підмножина точок x_i, відстань яких до x_j не перевищує σ:

$$E(x_j)=\{x_i \in X_Q \mid d(x_i,x_j)\le \sigma; \tag{2.8}$$

б) кореневим об'єктом або ядровим об'єктом ступеня m називається об'єкт x_j, σ - окіл якого містить не менше m об'єктів:

$$x_j \in X_Q \,|\, cardE(x_j) \geq m; \tag{2.9}$$

в) об'єкт x_i є безпосередньо щільно-досяжний з об'єкта x_j, якщо він входить в околицю $x_i \in E(x_j)$, тобто знаходиться на відстані не більшій ніж σ σ від точки x_j, і x_j – кореневий об'єкт.

$$x_i \in E(x_j) \,|\, cardE(x_j) \geq m; \tag{2.10}$$

г) об'єкт x_i є щільно-досяжний з об'єкта x_j, якщо існують

—

$$\{x_p \in X_Q \,|\, p = \overline{1,k}; x_1 \equiv x_j; x_2 \equiv x_i\} \tag{2.11}$$

такі, що утворюють шлях з безпосередньо щільно-досяжних точок від x_i до x_i, тобто кожна точка x_{p+1} x_{p+1} є безпосередньо щільно-досяжною з точки x_p:

$$\forall_p = \overline{1,k-1}: \{x_{p+1} \in E(x_p) \,|\, cardE(x_p) \geq m\}. \tag{2.12}$$

Всі точки шляху повинні бути ядровими, можливо за виключенням x_j – інакше шляху не буде;

д) всі точки, не досяжні з будь-якої іншої точки, є викидами.

Отже, ядрова точка x_j утворює «кластер» разом з усіма точками досяжними з неї. Кожен кластер містить принаймні одну ядрову точку. Не-ядрові точки можуть бути частиною кластера, тоді вони утворюють «ребро» кластера, оскільни з них не будуть досяжні інші точки. На рисунку 2.3 зображена досяжність об'єктів в методі DBSCAN.

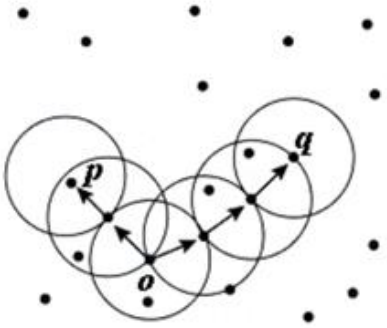


Рисунок 2.3 – Досяжність об’єктів в методі DBSCAN

Алгоритм DBSCAN можна описати наступними кроками:

1. Знайти точки у кожному σ-околі кожної точки $x_i \in X_Q$ та визначити ядрові точки x_j, у яких більше ніж m сусідів.

2. Знайти компоненти зв’язності для ядрових точок на графі сусідів, виключивши з розгляду точки, які не є ядровими.

3. Приєднати кожну не ядрову точку до найближчого кластера, за умови, що кластер знаходиться в σ-околі, інакше помітити її як шумову (викид).

4. При виконанні кластеризації важливо, скільки в результаті повинно бути побудовано кластерів. На рисунку 2.4. зображено хід кластеризації методом DBSCAN. Припускається, що кластеризація повинна виявити природні локальні згущення об’єктів. Тому число кластерів є параметром, який часто суттєво ускладнює вид алгоритму, якщо припускається невідомим, й суттєво впливає на якість результату, якщо припускається відомим.

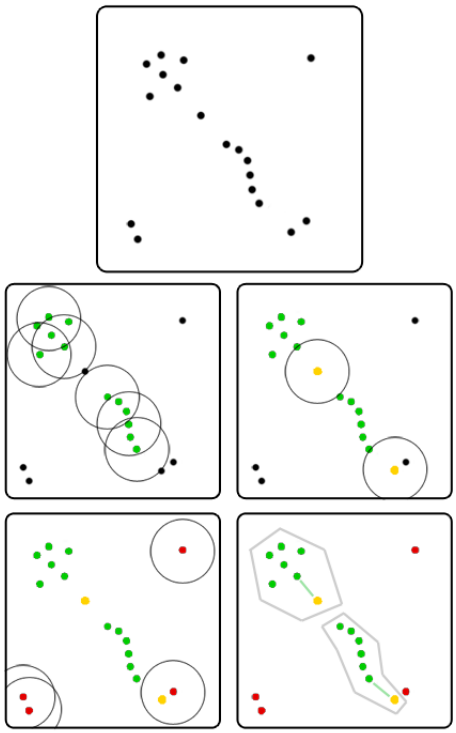


Рисунок 2.4 – Хід кластеризації методом DBSCAN

Головною перевагою алгоритма DBSCAN є те, що він не вимагає попередніх припущень про кількість кластерів і знаходить кластери довільної форми, але потрібно прийняти два параметри – відповідно максимальну відстань $\sigma = d_{\max}(x_i, x_j)$ між сусідніми точками, яка називається околицею об’єкта, і мінімальне число точок в околиці (кількість сусідів) m, коли можна говорити, що ці екземпляри даних утворюють один кластер.

DBSCAN дозволяє працювати з великими базами даних і є стійким до викидів, але погано працює з початковими множинами з великим перепадом щільностей і дає неоднозначні результати для крайових точок.

2.1.4 Алгоритм просіювання

Для такого групування уводиться парне порівняння об’єктів q_i, q_j з множини $Q = \{q_j\}_{j=1}^{n}$ й для кожного об’єкта q_i формується кортеж (група) «схожих» об’єктів:

$$c_j = \{q_p \mid p \neq j\}. \quad (2.13)$$

Далі формується оцінка $r(c_l q_j)$ «розкиду» елементів кортежу відносно q_j. Множина Q впорядковується, при умові

$$r(c_l q_i) \leq r(c_l q_j) \Rightarrow q_i \leq q_j, \quad (2.14)$$

тобто об'єкти ранжуються за розкидом інших об'єктів відносно них.

Кластеризація за принципом решета: перший об'єкт із відсортованого списку разом зі своїм кортежем формує перший кластер, після чого елементи кортежу видаляються з вибірки; перший з об'єктів, що залишилися, і його кортеж формують другий кластер; елементи вибірки, які не були вилучені раніше, видаляються; процедура класифікації триває, поки вся вибірка або задана її частка не буде класифікована. Така класифікація приводить до перекриття (пересічні кластери), тому що кластери можуть перетинатися через те, що спочатку всі елементи беруть участь в упорядкуванні вибірки, й порядок після їхнього виключення не переглядається.

Варіантом обробки, не пов'язаної з конкретним видом розподілів класів, є алгоритм Уішарта, побудований на основі правила найближчих сусідів, коли кортеж формується з фіксованого числа найближчих елементів n_j, а в якості оцінки $r(c_l q_j)$ використовується відстань $d(q_j n_j)$ від елемента q_j до самого дальнього з n_j його найближчих сусідів.

Параметрами сімейства алгоритмів просівання є: спосіб формування кортежу, міра подібності на множині об'єктів, функція $r(c_l q_j)$ оцінки розкиду.

Для реалізації просіювання звичайно вихідні числові вектори сортуються за зростанням і будується матриця мір близькості. Структура

матриці залежить від вибраного методу визначення відстані між кластерами, які ми хочемо отримати (за центрами ваг; за найближчими сусідами; за найбільш віддаленими об’єктами) і від порогового значення відстані.

2.1.5 Алгоритм Гюстафсона-Кесселя

Модифікація методу *k*-середніх, де враховуються кореляційні залежності кластера

$$p_{ij} = \frac{\sum_{i,j=1}^{n} (c_i - \overline{c_l})(c_j - \overline{c_l})}{\sqrt{\sum_{i=1}^{n} (c_i - \overline{c_l})^2} \sqrt{\sum_{j=1}^{n} (c_j - \overline{c_l})^2}}. \quad (2.15)$$

Як міри близькості в алгоритмі Гюстафсона-Кесселя використовується відстань Махаланобіса [27].

Завдяки цьому кластери замість сферичної форми стають еліпсоїдами, що дозволяє більш якісно проводити розбивку, у випадку, якщо елементи витягнуті уздовж яких-небудь напрямків, інакше кажучи, якщо існують стійкі комбінації, що визначають кластер, то краще розглядати елементи на приналежність до кластера уздовж цих комбінацій.

2.1.6 Алгоритм FOREL

Одна з модифікацій *k*-середніх. Відмінність полягає в тому, що під близькістю елементів у кластері розуміється покриття їх сферою заданого радіуса.

Схема роботи алгоритму полягає в наступному – береться центр кластера (на першому кроці це довільний елемент), і до кластера

приписуються всі елементи, які віддалені від центру не більше ніж на задану відстань $d_{\max}$. Потім перераховується центр, у якості якого береться середня точка отриманого кластера, і заново перераховуються елементи кластера. Так триває доти, поки центр не стабілізується [28].

2.2 Методи кластеризації багатовимірних часових рядів для інтелектуального аналізу даних

2.2.1 Кластеризація на основі онлайн кореляції

Методи кластеризації на основі онлайн-кореляції тісно пов’язані з проблемою прогнозування. Такі методи, як правило, засновані на групуванні потоків на основі їх кореляції між собою в минулому вікні історії. Таким чином, функція подібності між різними серіями використовує функцію внутрішньопотокової регресії для того, щоб вловити кореляції між різними потоками. Ці методи, як правило, засновані на вікнах безпосередньої історії. Зокрема, вікно довжиною p використовується для регресійного аналізу, і різні серії кластеризовані на основі цих тенденції. Деякі найпоширеніші методи сегментування таких потоків визначають функції подібності на основі регресії у попередньому вікні історії. Зверніть увагу, що два потоки в одному кластері не повинні позитивно корелювати. Насправді також можна вважати, що два потоки з досконалою негативною кореляцією належать до одного кластера, якщо передбачуваність між різними потоками висока. Це досить часто трапляється у багатьох реальних сценаріях, коли деякі потоки можна майже ідеально передбачити від інших.

Метод селективних м’язів призначений для визначення k найкращих представників поточного часового ряду, які можуть бути використані для прогнозування інших серій. Цей підхід можна вважати версією k-медоїдної кластеризації для онлайн-кластеризації часових рядів на основі передбачуваності. Одним із важливих аспектів оригінального підходу

Selective Muscles є те, що він був розроблений для пошуку *k* найкращих представників, які передбачають один конкретний потік у даних [29]. З іншого боку, при неконтрольованій кластеризації кореляції бажано визначити найкращий набір представників, які можуть передбачити всі потоки даних. Однак ці дві проблеми в принципі майже абсолютно однакові, оскільки один і той же підхід у селективних м'язах може використовуватися з агрегованою функцією для різних часових рядів.

Підхід використовує жадібний метод для того, щоб вибрати *k*-представників для оптимізації передбачуваності інших потоків. Підхід до відбору представників такий. У кожній ітерації потік включається в репрезентативний набір, який оптимізує помилку оцінки. Згодом наступний потік, який відбирається, базується на максимізації сукупного впливу на помилку оцінки, враховуючи потоки, які вже були вибрані. У кожній ітерації потік додається до набору представників, для яких зростаючий вплив на оцінку помилка якомога більша. Однак техніка в оптимізована для того, щоб вибрати *k* потоків, які оптимізують конкретну залежну змінну. Графічне представлення використовується для моделювання залежностей між потоками з використанням структури зв'язку. Жадібний алгоритм використовується для того, щоб вибрати оптимальний набір представників. Було показано, що підхід має межу наближення *(e - 1) / e* щодо оптимального вибору представників. Метод виконує кластеризацію безпосередньо на потоках, визначаючи функцію подібності на основі регресії. Іншими словами, два потоки вважаються подібними, якщо можливо передбачити один потік від іншого. В іншому випадку для групування потоків використовується безпосереднє вікно історії.

Загалом, для проблеми кластеризації не потрібно моделювати витрати. Спрощена версія алгоритму, в якій всі витрати встановлюються однаковими, наведена на рисунку 2.5.

```
Algorithm CorrelationCluster(Time Series Streams: [1...n]
    NumberOfStreams: k;
begin
  J = Randomly sampled set of k time series streams;
  At the next time stamp do
  repeat
    Add a stream to J, which leads to
      maximum decrease in regression error;
    Drop the stream from J which leads to
      least increase of regression error;
  until(J did not change in last iteration)
end
```

Рисунок 2.5 – Динамічна підтримка представників кластеру

Подібність між двома потоками *i* та *j* дорівнює помилці регресії при прогнозуванні потоку *j* з потоку *i* з використанням будь-якої лінійної моделі. Зауважимо, що функція подібності між потоками *i* та *j* не є симетричною, оскільки помилка передбачення потоку і з потоку j відрізняється від помилки передбачення потоку *j* з потоку *i*. Ці методи для кластеризації потоків на основі онлайн-кореляції дуже корисні у багатьох додатках, оскільки можна вибрати невеликі підмножини потоків, з яких можна ефективно передбачити всі інші потоки. Ряд інших методів, які не обов'язково безпосередньо пов'язані м'язи відбирають представників із вихідних потоків даних. Такі методи зазвичай використовуються для вибору датчика.

Алгоритми вибору датчиків природно пов'язані з кластеризацією кореляцій [30]. Це пов'язано з тим, що такі методи зазвичай вибирають репрезентативний набір потоків, який можна використовувати для передбачення інших потоків у даних. Представницькі потоки використовуються як проксі для збору потоків з усіх різних потоків. Такий підхід застосовується з метою економії енергії. Це, природно, створює кластери, в яких кожен потік належить кластеру, що містить певного представника. За останні роки було розроблено ряд методів для визначення кореляції між множинними потоки в режимі реального часу. Методи, що використовуються для статистичних вимірювань, щоб знайти кореляційні зв'язки та прогнозувати поведінку базового потоку даних. Роботи з

запропонованих методів для вибору датчика з використанням специфічних знань про зв’язок та зворотного зв’язку щодо корисності відповідно. Методи вибору спостереження пропонуються в тому випадку, коли важливість набору датчиків може бути попередньо змодельована як субмодульна функція. Метод використовує лінійну регресію для того, щоб визначити співвідношення базових даних та використовувати їх для прогнозування. Методика пропонує загальні методи моніторингу для визначення статистичних кореляцій даних у режимі реального часу. Алгоритми кластеризації кореляцій у часових рядах також можуть бути використані для моніторингу ключових змін у тенденціях кореляції в базовому потоці. Це тому, що коли кореляція між різні потоки змінюються з часом, це призводить до зміни членства потоків у різних кластерах даних.

2.2.2 Алгоритм динамічної трансформації часової шкали (DTW)

Алгоритм дозволяє знайти оптимальну відповідність між часовими послідовностями.

Використання евклідової відстані має істотний недолік: якщо два часових ряди однакові, але один з них незначно зміщений у часі (уздовж осі часу), то евклідова метрика може порахувати, що ряди відрізняються один від одного як зображено на рисунку 2.6. DTW-алгоритм був введений для того, щоб подолати цей недолік і надати наочне вимірювання відстані між рядами, не звертаючи увагу як на глобальні, так і на локальні зрушення на часовій шкалі.

Розглянемо два часових ряда – X_i довжиною n_i і X_j довжиною n_j:

$$X_i = \{x_i\}_{i=1}^{n_i}, \tag{2.16}$$

$$X_j = \{x_j\}_{i=j=1}^{n_i}. \quad (2.17)$$

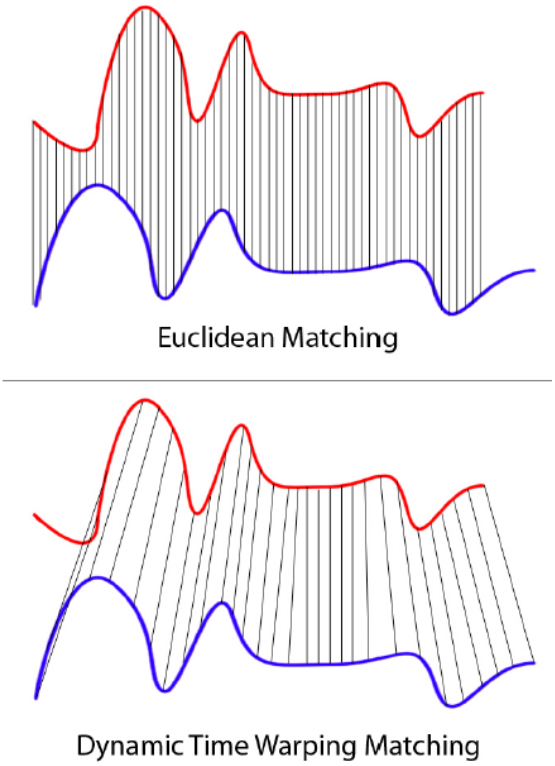


Рисунок 2.6 – Різниця між Евклідовою метрикою і метрикою в методі DTW

На першому етапі будуємо матрицю відстаней $D = \{d_{i,j}\}$ між елементами рядів порядку $n_i \times n_j, d_{i,j} \equiv_{\Delta_{ij}}$. Матриця відстаней в методів DTW зображена на рисунку 2.7.

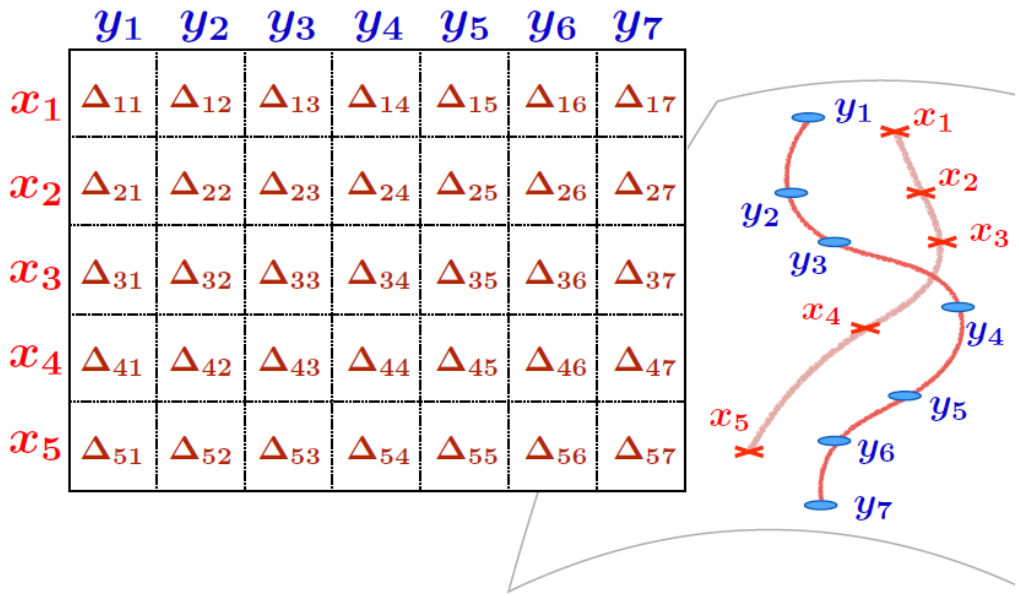


Рисунок 2.7 – Матриця відстаней в методі DTW

Зазвичай використовується евклідова відстань

$$d_{i,j}(x_i, x_j) = (x_i - x_j)^2 \tag{2.18}$$

або різниця по координатам. Для одного виміру це становить

$$d_{i,j}(x_i, x_j) = | x_i - x_j |. \tag{2.19}$$

Кожен елемент матриці відповідає відстані між точками $x_i x_i$.

На другому етапі будуємо матрицю трансформацій $D_{DTW} = \{r_{i,j}\}$, приклад якої зображений на рисунку 2.8.

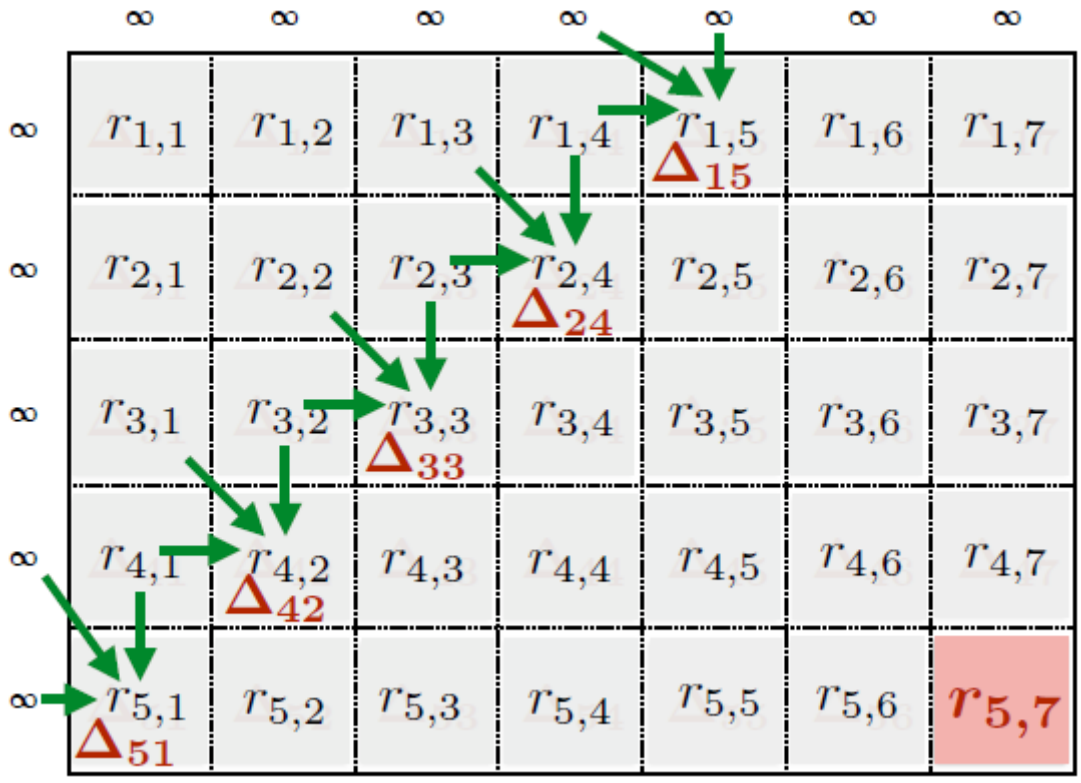


Рисунок 2.8 – Матриця трансформацій в методі DTW

Після заповнення матриці трансформації, ми переходимо до заключного етапу, який полягає в тому, щоб побудувати певний оптимальний шлях трансформації (деформації) і DTW відстань (вартість шляху) [31].

Шлях трансформації W – це набір суміжних елементів матриці D_{DTW}, який встановлює відповідність між X_i і X_j і мінімізує загальну відстань між ними:

$$W = \{W_k\}_{k=1}^{k}, W_k = (i,j)_k, \max(n_i n_j) \le K < (n_i + n_j), \qquad (2.20)$$

де K – довжина шляху.

Довжина k-го елемента W_k становить:

$$\forall W_k = (i,j)_k : d(W_k) = d_{i,j}(x_i, x_j). \qquad (2.21)$$

Шлях трансформації повинен відповідати таким обмежуючим умовам:

$$w_1 = (1,1), w_k = (n_i, n_j). \quad (2.22)$$

Це обмеження гарантує, що шлях трансформації містить всі точки обох часових рядів.

Безперервність (умова на довжину кроку):

$$\forall w_k = (w_i, w_j), w_{k+1} = (w_{i+1}, w_{j+1}) : w_i - w_{i+1} \le 1, w_j - w_{j+1} \le 1. \quad (2.23)$$

Це обмеження гарантує, що шлях трансформації пересувається на один крок за один раз. Тобто обидва індекси *i,j* можуть збільшитися лише на 1 на кожному кроці шляху.

Монотонність:

$$\forall w_k = (w_i, w_j), w_{k+1} = (w_{i-1}, w_{j-1}) : w_i - w_{i-1} \le 1, w_j - w_{j-1} \le 1. \quad (2.24)$$

Це обмеження гарантує, що шлях трансформації не повертатиметься назад до пройденої точки. Шлях трансформації зображенний на рисунку 2.9. Тобто обидва індекси *i* та *j* або залишаються незмінними, або збільшуються (але ніколи не зменшуються).

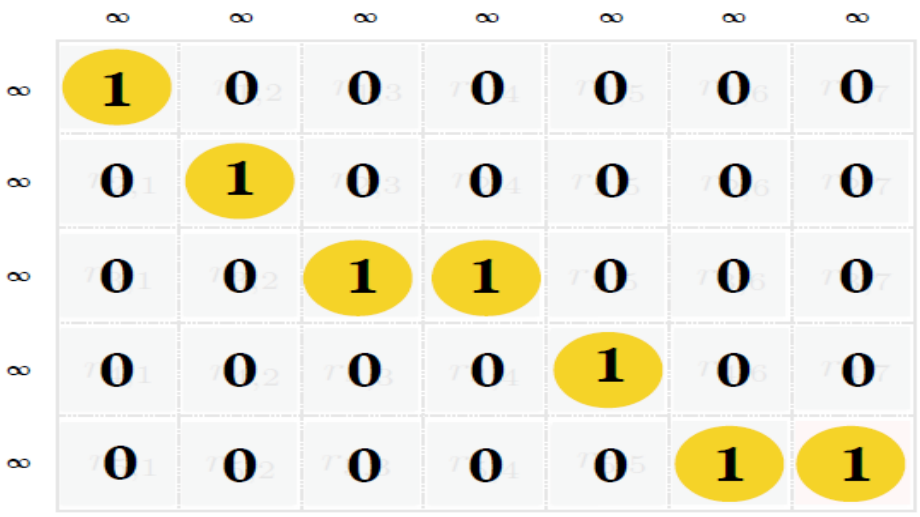


Рисунок 2.9 – Шлях трансформації в методі DTW

DTW відстань (вартість шляху) між двома послідовностями розраховується на основі оптимального шляху трансформації за допомогою формули:

$$DTW(X_i, X_j) = \min(\frac{\sum_{k=1}^{k} d(w_k)}{K}). \quad (2.25)$$

Хоча алгоритм успішно використовується в багатьох областях, він може видавати невірні результати, пов’язані з тим, що особлива точка (пік, западина, плато, точка перегину) одного ряду розташована трохи вище або нижче відповідної їй особливої точки іншого ряду [32].

Модифікацією алгоритма DTW є алгоритм soft-DTW, при якому DTW відстань (вартість шляху) є змінною, яка залежить від параметра згладжування γ:

$$DTW(X_i, X_j) = -\gamma \log \sum_{k=1}^{k} e(\frac{d(W_k)}{K \cdot \gamma}). \quad (2.26)$$

K у знаменнику використовується для урахування того, що шляхи трансформації можуть бути різної довжини.

Як і в одновимірному випадку, багатовимірний DTW дозволяє розтягуватися за віссю часу, відповідає всім елементам і широко використовується в області розпізнавання мови. DTW, на відміну від евклідової відстані, не вимагає, щоб два порівняні часові ряди були однакової довжини і не чутливі до місцевого зміщення часу. Однак DTW не є метрикою, оскільки вона не відповідає нерівності трикутника, і її часова складність становить *O (mn)*, що означає, що вона обчислювально дорогий для тривалих часових рядів і корисний лише для коротких, складаючи кілька тисяч балів. Більше того, DTW, як і євклідова відстань, було доведено чутливим до вмісту шуму дуже детальний опис обчислень та семантики DTW. Влахос та ін. застосовувати DTW до даних рукописного вводу. Перш ніж порівнювати дві траєкторії, вони перетворюються в інваріантний до повороту простір кута / довжини дуги, щоб видалити компоненти перекладу, обертання та масштабування. У новому просторі використовується техніка викривленого збігу, щоб компенсувати зміни форми. Цей підхід створює декілька роздільних здатностей порівняних часових рядів, огрублюючи їх і представляючи з меншою кількістю точок. Потім стандартний DTW запускається з найнижчою роздільною здатністю, а створений шлях обтікання передається на наступну вищу роздільну здатність. Нарешті шлях уточнений, і цей процес триває, поки не буде досягнуто початкове дозвіл часового ряду. FastDTW, будучи неоптимальним наближенням DTW і створюючи помилки до 19,2%, є швидшим, оскільки кількість комірок, які він оцінює, лінійно масштабується з тривалістю часового ряду. Було доведено підвищення точності та ефективності в діапазонах SakoeChiba та абстракції даних, що є двома іншими популярними наближеннями DTW. Однак не відомо про результати багатовимірних часових рядів, тому ефективність FastDTW при застосуванні до даних цього типу повинна бути перевірена.

2.2.3 Багатовимірний LCSS

Влахос та ін. запропонував дві неметричні функції відстані як продовження LCSS для багатовимірних часових рядів. Цей метод виявився стійким до шуму, особливо у порівнянні з DTW та ERP. LCSS не залежить від постійного відображення часових рядів; таким чином, підходи, що використовують або поширюють його, як правило, фокусуються на подібних частинах між досліджуваними послідовностями. LCSS, на відміну від евклідової відстані, не бере до уваги неперевершені елементи та відповідає лише подібним частинам [17]. Отже, це дозволяє траєкторіям розтягуватися по осі часу. DTW та Евклідова відстань намагається відповідати кожному елементу, тому вони більш чутливі до викидів. Однак при використанні LCSS часові ряди для порівняння повинні мати однакові частоти дискретизації.

2.2.4 Багатовимірна відстань редагування

У 2005 році Чень та співавт. запропонував EDR, редагувати відстань на реальній послідовності, щоб вирішити проблему порівняння реальних шумних траєкторій з точністю та надійністю, стверджуючи, що EDR є більш надійним та точним, ніж DTW, LCSS, ERP та евклідова відстань. EDR визначається як кількість операцій вставки, видалення або заміни для перетворення траєкторії *T1* в іншу *T2*. Зокрема, застосовуючи редагування відстані до послідовностей дійсних чисел, а не до рядків, як це було спочатку запропонований Левенштейном.

Елементи *T1* і *T2* повинні збігатися, якщо відстань між ними у всіх вимірах нижче порога ε, подібно до способу відстані LCSS. Таким чином, EDR вдається справлятися з галасливими багатовимірними часовими рядами,

не зазнаючи впливу викидів, і обробляти зміщення в осі часу, як відстань ERP.

2.2.5 Багатовимірна відповідність підпослідовності

Існує множина методів, запропонованих для багатовимірного узгодження підпослідовностей, які можуть бути використані для таких завдань видобутку даних, як кластеризація та класифікація. SPRING – це динамічне програмування на основі методу, який ідентифікує послідовності розвиваються числових потоків, які є найближчими до запиту в постійному просторі та лінійному часі в розмірі набору даних.

Коціфакос та ін. запровадили SMBGT, метод узгодження підпослідовностей, який дозволяє мати пробіли як у запиті, так і в цільових послідовностях і обмежує максимальну довжину збігу між двома. У своєму дослідженні вони застосовують SMBGT для побудови системи Query by-Humming, яка, подаючи пісню із запитом, що озвучує, отримує найкращі *K* найбільш подібні пісні в базі даних. Запропонована міра подібності, SMBGT, отримуючи запит *Q* і цільову послідовність *X*, знаходить підпослідовність *X*, яка найкраще відповідає *Q*. Експериментальна оцінка запропонованої міри подібності проводилася на двовимірних часових рядах нот довільної довжини. В експериментальній оцінці показано, що основною перевагою SMBGT перед порівняними методами узгодження послідовностей (SPRING, Edit distance та DTW) є те, що він краще справляється з високим рівнем шуму. У деяких додатках, таких як досліджувана проблема Query-by-Humming, це надзвичайно важливо.

3 ДОСЛІДЖЕННЯ МЕТОДУ ІНТЕЛЕКТУЛЬНОГО АНАЛІЗУ ДАНИХ ДЛЯ ВИРІШЕННЯ МАРКЕТИНГОВОЇ ЗАДАЧІ

3.1 Засоби сбору та аналізу даних

3.1.1 Google Analytics

Google Analytics — метричний сервіс, що дозволяє отримувати дані про кількість користувачів, аналізувати їх поведінку і дані про джерело, звідки відвідувачі переходять на ресурс [33].

Можливості Google Analytics:

– дозволяє відстежувати тренди;

– дає інформацію про джерело відвідувачів (з яких сторінок переходять) і яка їхня поведінка на сторінці;

– дає розуміння того, як конвертувати відвідувачів в покупців і клієнтів;

– відстежує, де і з якої причини користувачі залишають сторінку;

– показує, по яких запитах відвідувачі знаходять сайт;

– показує, який канал трафіку приносить більше доходу;

– дає список найпопулярніших сторінок сайту;

– дозволяє контролювати ефективність рекламних кампаній;

– дає розуміння того, який контент стимулює користувачів до дії або привертає увагу і багато іншого.

Google Analytics дозволяє розвивати ресурс, грунтуючись на отриманих даних від сервісу, а також дозволяє відстежувати ефективність проведеної рекламної кампанії.

Трекінговий javascript код був встановлений на веб-сайт. Кожного разу, коли користувач завантажує сторінку сайту, в його браузері виконується код відстеження. Під час першого візиту він записує в браузер відвідувача cookie-файл, який містить унікальний ідентифікатор користувача – Client ID.

Завдяки cookie-файлам всі наступні заходи з того ж браузера будуть зараховані системою Google Analytics як повторні відвідування.

Щоб система Google Analytics могла співвідносити користувачів із трафіком, з кожним зверненням до системи надсилається унікальний ідентифікатор, пов'язаний із користувачем. Роль ідентифікатора може виконувати одиничний основний файл cookie з назвою _ga, що зберігає ідентифікатор клієнта Google Analytics. Разом з ідентифікатором клієнта можна також використовувати функцію User ID, щоб точніше визначати користувачів на всіх пристроях, де вони переглядають сайт або використовують додаток.

Для того, щоб інтегрувати Google Analytics на сайт потрібно першочергово перейти на сайт Google Analytics і зареєструватися. Далі створити акаунт:

– вибрати, що відстежувати: веб-сайт або мобільний додаток;

– придумати назву облікового запису, що відображає його зміст;

– ввести назву сайту;

– вказати URL сайту;

– вибрати відповідну галузь;

– вказати країну та часовий пояс (в якому велика частина аудиторії сайту) – вони будуть відображатися у звітах.

Після успішно створеного акаунта буде згенеровано код відстеження Google Analytics (рисунок 3.1), який потрібно встановити на кожній сторінці веб-сайту безпосередньо після тега <head>.

```
<!-- Global Site Tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=GA_TRACKING_ID"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'GA_TRACKING_ID');
</script>
```

Рисунок 3.1 – Трекіновий javascript код Google Analytics

Для установки Google Analytics на сайт можна використовувати кілька простих методів:

– використовувати безпосередньо скрипт Global Site Tag;

– скористатися диспетчером тегів;

– використовувати різні плагіни.

Наступним кроком буде настройка першого представлення. В першу чергу необхідно провести такі налаштування:

– задати валюту подання, яка буде відображатися в звітах;

– увімкнути фільтрацію ботів, щоб їх візити на сайт не спотворювали статистику по трафіку;

– включити відстеження пошукових запитів на сайті, якщо хочете знати, що шукають ваші відвідувачі.

3.1.2 Базові бібліотеки Python для аналізу даних

Бібліотека scikit-learn є однією з найпопулярніших платформ для повсякденного машинного навчання та науки про дані, оскільки вона побудована на Python, повнофункціональній мові програмування. Scikit learn був створений з урахуванням мислення програмного забезпечення. Його основний дизайн API полягає в тому, що він простий у використанні, але при цьому потужний, і при цьому зберігає гнучкість для дослідницьких робіт. Ця надійність робить його ідеальним для використання в будь-якому наскрізному проекті ML, починаючи від фази дослідження і закінчуючи розгортанням виробництва.

Scikit-learn поставляється з великою кількістю функцій. Ось декілька з них, які допоможуть вам зрозуміти поширення:

– алгоритми контрольованого навчання: згадайте будь-який керований алгоритм машинного навчання, про який ви могли чути, і існує дуже висока ймовірність того, що він є частиною scikit-learn. Починаючи з узагальнених

лінійних моделей (наприклад, лінійна регресія), підтримують векторні машини (SVM), дерева рішень до байєсівських методів – усі вони є частиною набору інструментів scikit-learn. Поширення алгоритмів машинного навчання є однією з головних причин широкого використання scikit-learn. Я почав використовувати scikit для вирішення навчальних завдань під наглядом, і рекомендував би це також людям, які вперше вивчають scikit / машинне навчання;

– перехресна перевірка: існують різні методи перевірки точності контрольованих моделей на невидимих даних за допомогою sklearn;

– алгоритми неконтрольованого навчання: Знову ж таки, в пропозиції є широке поширення алгоритмів машинного навчання – починаючи від кластеризації, факторного аналізу, аналізу основних компонентів до нейронованих нейронних мереж;

– вилучення функцій: Scikit-learn для вилучення об’єктів із зображень та тексту.

Пакет «pandas» – найважливіший інструмент, яким сьогодні – користуються вчені-аналітики та аналітики, що працюють у Python.

Pandas витягує дані з цього CSV у DataFrame – таблицю, в основному, а потім дозволяє робити такі речі:

– обчислювати статистику та відповідати на запитання щодо даних, наприклад:

1) Яке середнє, медіана, максимум або мінімум кожного стовпця?
2) Чи співпадає стовпець A зі стовпцем B?
3) Як виглядає розподіл даних у стовпці C?

– очищати дані, виконуючи такі дії, як видалення відсутніх значень та фільтрування рядків або стовпців за деякими критеріями;

– візуалізувати дані за допомогою Matplotlib. Побудувати графіки, лінії, гістограми тощо;

– зберігати очищені, перетворені дані назад у CSV, інший файл або базу даних.

Pandas побудований поверх пакету NumPy, тобто значна частина структури NumPy використовується або відтворюється в Pandas. Дані в pandas часто використовуються для статистичного аналізу в SciPy, побудови графіків функцій з Matplotlib та алгоритмів машинного навчання в Scikit-learn.

NumPy – це бібліотека мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих (і дуже швидких) математичних функцій для операцій з цими масивами.

Основним об'єктом NumPy є однорідний багатовимірний масив (в numpy називається numpy.ndarray). Це багатовимірний масив елементів (зазвичай чисел), одного типу [34].

Найбільш важливі атрибути об’єктів ndarray:

– ndarray.ndim – число вимірювань (частіше їх називають "осі") масиву;

– ndarray. – розміри масиву, його форма. Це кортеж натуральних чисел, що показує довжину масиву по кожній осі. Для матриці з n рядків і m стовпів, shape буде (*n*, *m*). Число елементів кортежу shape одно ndim;

– ndarray.size – кількість елементів масиву. Очевидно, дорівнює добутку всіх елементів атрибута shape;

– ndarray.dtype – об’єкт, що описує тип елементів масиву. Можна визначити dtype, використовуючи стандартні типи даних Python. NumPy тут надає цілий букет можливостей, як вбудованих, наприклад: bool_, character, int8, int16, int32, int64, float8, float16, float32, float64, complex64, object_, так і можливість визначити власні типи даних, в тому числі і складові;

– ndarray.itemsize – розмір кожного елемента масиву в байтах;

– ndarray.data – буфер, який містить фактичні елементи масиву. Зазвичай не потрібно використовувати цей атрибут, так як звертатися до елементів масиву найпростіше за допомогою індексів.

Matploitlib – це бібліотека Python, яка використовується для побудови графіків; ця бібліотека python надає API, орієнтований на заперечення, для інтеграції графіків у програми.

Matplotlib не є частиною стандартних бібліотек, яка встановлюється за замовчуванням, коли Python. Існує декілька наборів інструментів, які розширюють функціональність python matplotlib [35]. Деякі з них є окремими завантаженнями, інші можуть надсилатися з вихідним кодом matplotlib, але мають зовнішні залежності:

– базова карта: це набір інструментів для складання карт з різними проекціями карт, узбережжям та політичними межами;

– cartopy: це бібліотека картографування, що містить об’єктно-орієнтовані визначення проекції карти та можливості довільної трансформації точки, лінії, багатокутника та зображення;

– інструменти Excel: Matplotlib надає утиліти для обміну даними з Microsoft Excel;

– mlot3d: використовується для тривимірних графіків;

– natgrid: це інтерфейс до бібліотеки natgrid для нерегулярного сітчастого розташування даних з інтервалом.

Однією з найважливіших особливостей Matplotlib є його здатність добре грати з багатьма операційними системами та графічними інтерфейсами. Matplotlib підтримує десятки бекендів і типів виводу, а це означає, що ви можете розраховувати на його роботу незалежно від того, яку операційну систему ви використовуєте або який формат виводу ви бажаєте. Цей крос-платформенний підхід «все для кожного» був однією з найбільших сильних сторін Matplotlib. Це призвело до великої бази користувачів, а це, в свою чергу, призвело до активної бази розробників та потужних інструментів і повсюдного використання Matplotlib у науковому світі Python.

SciPy, наукова бібліотека для Python – це бібліотека з математикою, наукою та технікою, що має ліцензію BSD [36]. Бібліотека SciPy залежить від NumPy, що забезпечує зручне та швидке маніпулювання N-вимірними

масивами [37]. Основною причиною побудови бібліотеки SciPy є те, що вона повинна працювати з масивами NumPy. Він забезпечує безліч зручних та ефективних числових практик, таких як процедури для чисельної інтеграції та оптимізації. Це вступний підручник, який висвітлює основи SciPy та описує, як поводитися з різними модулями. SciPy містить різноманітні підпакети, які допомагають вирішити найпоширеніші проблеми, пов’язані з науковими обчисленнями [38].

Підпакети SciPy:

- введення / виведення файлів – scipy.io;
- спеціальна функція – scipy.special;
- операція лінійної алгебри – scipy.linalg;
- інтерполяція – scipy.interpolate;
- оптимізація та підгонка – scipy.optimize;
- статистика та випадкові числа – scipy.stats;
- числова інтеграція – scipy.integrate;
- швидкі перетворення Фур’є – scipy.fftpack;
- обробка сигналів – scipy.signal;
- маніпуляція із зображеннями – scipy.ndimage.

3.2 Платформа візуальної аналітики Tableau

Як провідний на ринку вибір сучасного бізнес-аналізу, платформа Tableau відома тим, що бере будь-які дані майже з будь-якої системи та швидко та легко перетворює їх на практичну інформацію.

Tableau була заснована в 2003 році в результаті проекту з інформатики в Стенфорді, який мав на меті покращити потік аналізу та зробити дані більш доступними для людей за допомогою візуалізації. Співзасновники Кріс Столте, Пат Ханрахан та Крістіан Шабо розробили та запатентували основоположну технологію Tableau VizQL, яка візуально виражає дані,

перекладаючи дії перетягування та перетягування в запити до даних через інтуїтивно зрозумілий інтерфейс.

Жива візуальна аналітика сприяє необмеженому дослідженню даних. Інтерактивні панелі інструментів допомагають швидко розкривати приховані ідеї. Tableau використовує природну здатність людей швидко розпізнавати візуальні тренди.

Tableau Desktop та Tableau Prep підтримуються в середовищах Windows та MacOS. Крім того, усі продукти Tableau працюють у віртуалізованих середовищах, коли вони налаштовані з відповідною базовою операційною системою Windows та мінімальними вимогами до обладнання. Ці віртуальні рішення включають середовища Citrix, Parallels та VMware.

Tableau підтримує можливість підключення до різноманітних даних, що зберігаються в самих різних місцях. На панелі «Підключення» перераховані найпоширеніші місця, до яких ви можете підключитися, або натисніть посилання «Більше», щоб побачити більше параметрів.

Після підключення до ваших даних Tableau робить наступне:

– відкриває новий аркуш. Це чистий аркуш, де ви можете створити свій перший вигляд;

– відображає джерело даних, до якого ви підключені. Якщо ви використовуєте кілька джерел даних, ви можете переглянути їх усі, перераховані тут;

– додає стовпці з джерела даних на панель даних ліворуч. Стовпці додаються як поля;

– автоматично призначає вашим даним типи даних (наприклад, дату, число, рядок тощо) та ролі (розмір чи міру).

Коли ви підключаєтеся до власних даних, можливо, вам доведеться виконати певну підготовчу роботу перед підключенням до них у Таблиці. Це пов’язано з тим, що Tableau робить припущення щодо ваших даних, щоб він міг правильно відображати їх для роботи.

На рисунках 3.2 та 3.3 зображені практичні приклади візуалізації даних в Tableau.

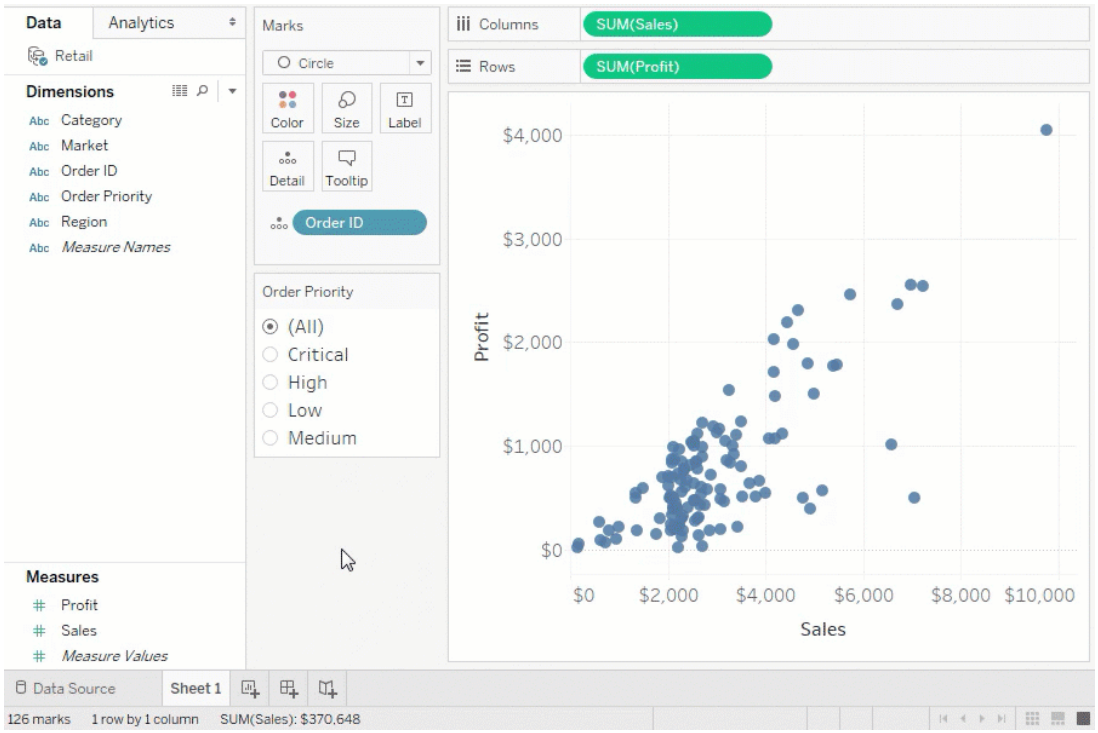


Рисунок 3.2 – Кореляційний аналіз в Tableau

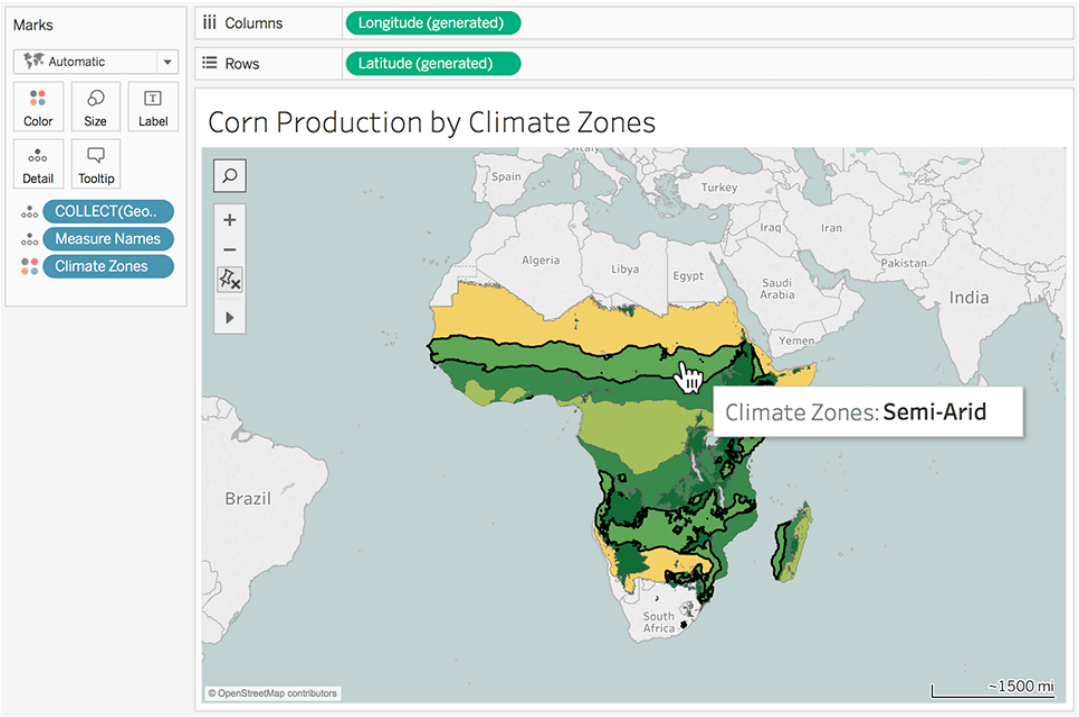


Рисунок 3.3 – Інтерактивні карти в Tableau

3.3 Маркетингова задача оцінки ефективності онлайн реклами

Розвиток маркетингових комунікацій за останні роки вийшов на якісно новий рівень. Сьогодні маркетолог володіє величезним об’ємом інформації і головна мета спеціаліста – зробити правильні висновки і спрямувати маркетингові зусилля на ті напрямки, які демонструють найвищу маркетингову ефективність.

Маркетингові дослідження відіграють вирішальну роль у формуванні стратегії розвитку як окремого підрозділу, так і компанії загалом. Аналіз та інтерпретація маркетингових даних є невід’ємними складовими того набору вмінь та навичок, які мають бути притаманними сучасному маркетологу. Сьогодні для засновника бізнесу вже замало просто налаштувати рекламну кампанію, запустити промо-акцію на радіо, або провести піар кампанію, пов’язану з випуском нового продукту. Керівник компанії аналізує кожну статтю маркетингових витрат і досягати максимальної їх ефективності – основна задача маркетингового підрозділу компанії.

Спеціалісти з маркетингу, при проведенні маркетингових досліджень, мають ставити правильні запитання та шукати на них відповіді серед масиву даних, якими вони володіють:

– Чи доцільними є витрати на кожен з маркетингових каналів?

– Як ми можемо покращити показник ефективності маркетингових витрат?

– Чи існують альтернативні маркетингові активності, які на даний момент не реалізовуються компанією?

– Чи задоволений наш цільовий споживач тією маркетинговою комунікацією, яку здійснює наша компанія?

Задля забезпечення стабільного розвитку компанії, маркетолог зобов’язаний забезпечити належний рівень комунікації з цільовим сегментом бізнесу щодо того як його продукт трансформує повсякденне життя та мислення кожного, хто його придбає. Саме тому спеціаліст з маркетингу

повинен аналізувати та прогнозувати маркетингову віддачу від коштів, витрачених на кожен з видів комунікацій.

Важливо зазначити, що сьогодні одним із основних інструментів маркетолога являється веб-сайт компанії, адже ми всі розуміємо: якщо ваш бізнес неможливо знайти в інтернеті – вашого бізнесу не існує. Комунікація з цільовімі споживачами онлайн набуває сьогодні неабиякого значення, особливо в умовах, коли більшість компаній зачиняються саме через втрачену, у порівнянні з конкурентами, можливість представити свій бізнес онлайн.

Маркетингову задачу, яку ми обрали для аналізу полягає в оцінці маркетингової ефективності такого каналу комунікації як онлайн реклама на платформі Google Ads. Іншими словами, який показник ROI (Return on Investment) для онлайн реклами спостерігається на даний момент і яким чином ми можемо його підвищити у наступних періодах.

До переліку основних показників, які допоможуть виявити залежність та спрогнозувати ROI в наступному періоді, увійшли:

– кількість показів рекламного оголошення;

– кількість кліків по рекламному оголошенню;

– рівень клікабельності рекламних оголошень (відношення кількості кліків до кількості показів рекламного оголошення);

– витрати на онлайн рекламу;

– дохід від рекламних кампаній Google Ads.

Результатом вирішення даної маркетингової задачі буде виявлення кореляції між показниками роботи рекламної кампанії на платформі Google Ads та доходу, отриманого в результаті проведення даної маркетингової активності.

3.4 Характеристика вхідного набору даних часових рядів, що використано для проведення аналізу

Таблиця містить в собі 61 рядкок та 7 стовпців, перші 19 рядків наведено на рисунку 3.4.

Day	Impressio	Clicks	CTR	Cost	Conv. value
9/2/2020	69,666	1,654	2.37%	3,655.73	19,458.32
9/7/2020	61,982	2,032	3.28%	4,075.44	9,511.18
9/8/2020	79,209	2,123	2.68%	4,725.83	44,522.05
9/11/2020	10,578	1,377	13.02%	2,672.94	27,005.70
9/12/2020	11,813	1,439	12.18%	3,569.49	24,161.66
9/13/2020	12,949	1,497	11.56%	3,474.39	6,581.57
9/14/2020	15,575	1,780	11.43%	4,766.09	100,338.35
9/15/2020	11,659	1,462	12.54%	3,712.22	9,823.92
9/19/2020	9,927	1,150	11.58%	2,360.54	14,455.67
9/21/2020	16,124	1,894	11.75%	5,241.14	16,610.67
9/23/2020	12,248	1,437	11.73%	3,058.28	6,322.09
9/25/2020	11,774	1,291	10.96%	2,974.90	37,210.33
9/26/2020	10,925	1,194	10.93%	2,452.84	45,409.16
9/27/2020	46,344	1,497	3.23%	3,034.22	11,860.10
9/28/2020	43,747	1,594	3.64%	2,969.98	39,051.96
9/30/2020	40,440	1,559	3.86%	3,102.69	100,183.52
10/2/2020	41,710	1,363	3.27%	2,885.48	46,215.10
10/5/2020	39,316	1,614	4.11%	3,653.79	25,102.39
10/6/2020	37,430	1,600	4.27%	3,543.95	27,267.57

Рисунок 3.4 – Приклад таблиці ad performance

З рисунку 3.4 можна побачити, що таблиця, яка розглядається, містить такі поля, що наведено в таблиці 3.1.

Таблиця 3.1 – Атрибути датасету

№	Назва стовпчика	Опис
1	day	Дата
2	impressions	Рекламні покази
3	clicks	Кліки на рекламне оголошення
4	ctr	Відношення кліків до показів
5	costs	Рекламні витрати
6	conv. value	Дохід

3.5 Практичні результати дослідження

Для початку завантажимо датасет в систему візуальної аналітики Tableau та визначемо атрибути стовбців. За замовчуванням система провела самплінг даних і тепер необхідно трансформувати та візуалізувати дані як множину на часовій осі. Результат трансформації даних зображений на рисунку 3.5.

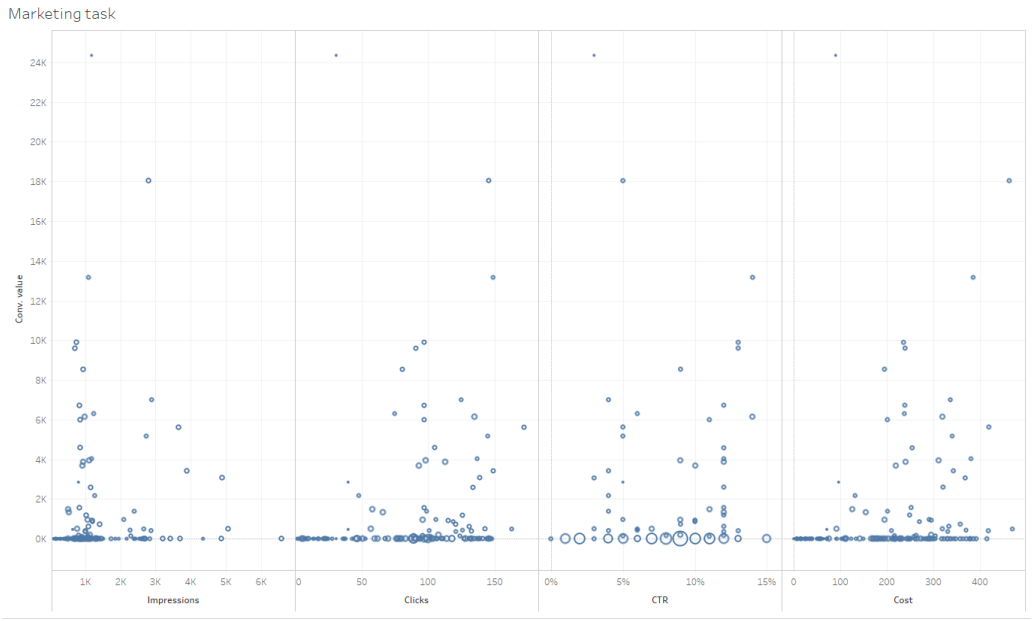


Рисунок 3.5 – Представлення даних у вигляді множини

Наступним кроком необхідно визначити середнє квадратичне відхилення. Результат перетворення даних зображений на рисунку 3.6.

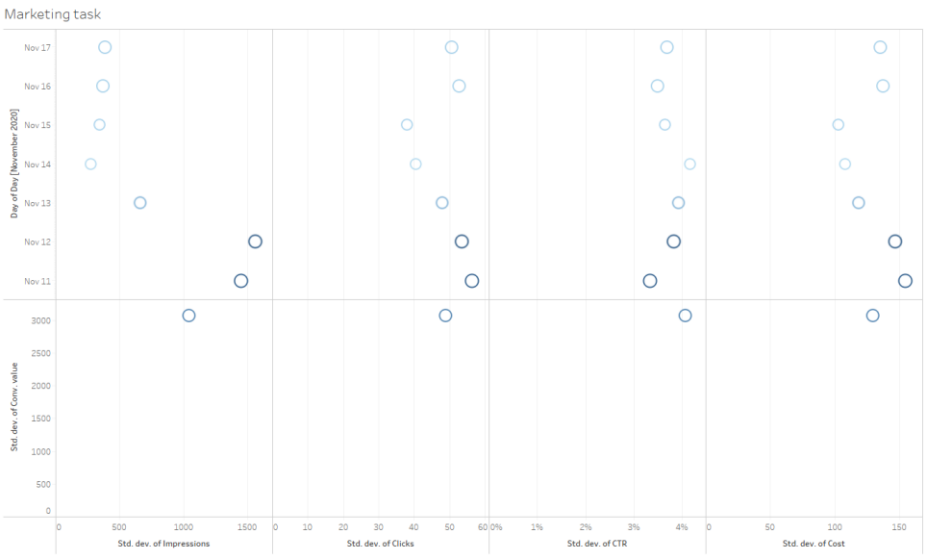


Рисунок 3.6 – Візуалізація даних у вигляді середнього квадратичного відхилення

Далі необхідно провести кластеризацію даних методом *k*-means. Кількість визначених кластерів – 6. Результат кластеризації даних зображений на рисунку 3.7.

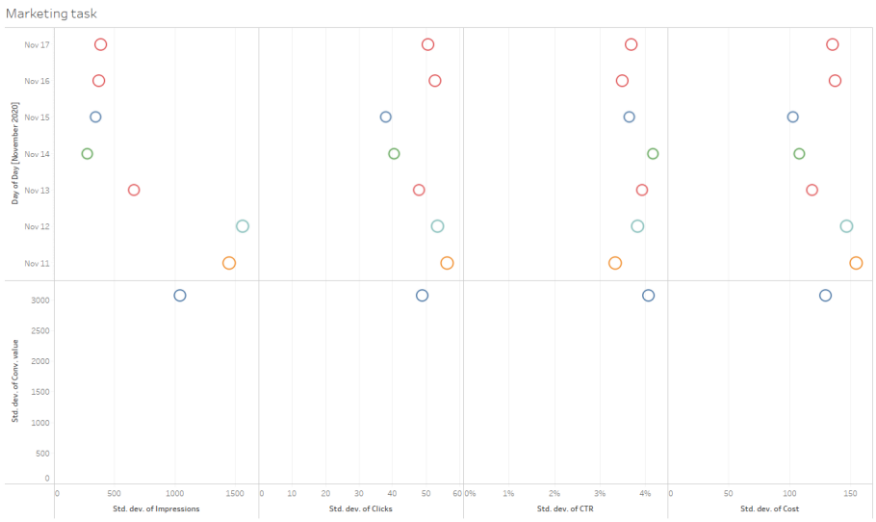


Рисунок 3.7 – Результат кластеризації даних

ВИСНОВКИ

В ході роботи було розглянуто методи кластеризації даних та часових рядів з попереднім прогнозуванням. Найбільш релевантними на даний час є методи кластеризації часових рядів. Кластеризація даних часового ряду, як і кластеризація для всіх типів даних, має мету створення кластерів з високою подобою всередені кластеру й низькою міжкластерною подобою. А саме, об'єкти, що належать тому ж кластеру, повинні показати високу подобу один одному, у той час як об'єкти, що належать різним кластерам, повинні показати низьку подобу, тобто високу відстань друг від друга. Реалізація методів кластеризації дозволяє краще зрозуміти дані; провести стиснення, виділивши найбільш типових представників, за умов збитковості даних; виявлення новизни, шляхом виділення об’єктів, що не потратили до жодного з кластерів.

Було розглянуто існуючі методи позбавлення від пропусків в даних в задачах кластеризації та доцільність їх використання і реальних задачах. Також розглянуто адаптований класичний метод кластеризації для вирішення проблеми неповних даних.

Для проведення кластеризації оброблених даних розглянуто метод кластеризації *k*-середніх.

За результатами проведеної роботи можна зробити висновок, що позбавляючись від пропусків, найгіршим варіантом є варіант видалення всіх рядків, які містять пропуски. Цей метод можливий лише у випадках коли вибірка містить мінімальну кількість пропусків, або тоді коли було попередньо проведено інший вид обробки і відбувається видалення залишків пустих значень. Найкращим вважається метод боротьби з пропусками з урахуванням взаємозв’язків між полями, але на даній вибірці він не значно перевершує метод заміни на середні значення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шумейко, А. А., & Сотник, С. Л. (2012). Интеллектуальный анализ данных (введение в Data Mining). *Днепропетровск: Белая ЕА*, *212*.

2. Вискребенцева С.О., Кобилін О.А. (2019) Методи сегментації зображень. Матеріали XXIII міжнародного молодіжного форуму. Радіоелектроніка та молодь у XXI столітті, 19-20.

3. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)* (pp. 665-670). IEEE.

4. Работягов, А. В., Ляшенко, В. В., & Кобылин, О. А. (2016). Сегментация сложных изображений цитологических препаратов.

5. Lyashenko, V., Mohammad, A., & Kobylin, O. (2015). Experiments with Fusion of Images with Use of Wavelet Transformation in Problems of the Text Information Analysis.

6. Деркач, О. І. (2016). Аналітична обробка текстової інформації за допомогою засобів кластеризації. *Young*, *34*(7).

7. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, *5*(57).

8. Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons.

9. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.

10. Perret, B., Chierchia, G., Cousty, J., Guimarães, S. J. F., Kenmochi, Y., & Najman, L. (2019). Higra: Hierarchical graph analysis. *SoftwareX*, *10*, 100335.

11. Steinley, D. (2006). K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, *59*(1), 1-34.

12. Ackermann, M. R. (2009). *Algorithms for the Bregman k-Median problem* (Doctoral dissertation, University of Paderborn).

13. Khachumov, M. V. (2012). Distances, metrics and cluster analysis. *Scientific and Technical Information Processing*, *39*(6), 310-316.

14. Huang, Z., & Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, *7*(4), 446-452.

15. Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons.

16. Zhang, J., Zhao, Z., Xue, Y., Chen, Z., Ma, X., & Zhou, Q. (2017). Time series analysis. *Handbook of Medical Statistics*, *269*.

17. Крашений, І. Е., Попов, А. О., Раміpез, Х., Горріз, Х. М., Крашеный, И. Э., Попов, А. А., ... & Горриз, Х. М. (2016). Використання методів кластеризації в системах нечіткого виводу для діагностики хвороби Альцгеймера на основі ПЕТ-зображень.

18. Штовба, С. Д. (2006). Побудова функцій належності нечітких множин за кластеризацією експериментальних даних. *Інформаційні технології та комп'ютерна інженерія*, (2), 92-95.

19. Xu, J., Han, J., Xiong, K., & Nie, F. (2016, July). Robust and Sparse Fuzzy K-Means Clustering. In *IJCAI* (pp. 2224-2230).

20. Gorshkov, Y., Kolodyazhniy, V., & Bodyanskiy, Y. (2009, June). New recursive learning algorithms for fuzzy Kohonen clustering network. In *Proc. 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems* (pp. 58-61).

21. Bodyanskiy, Y. V., Tyshchenko, O. K., & Mashtalir, S. V. (2019, June). Fuzzy Clustering High-Dimensional Data Using Information Weighting. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 385-395). Springer, Cham.

22. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. In *Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60-63). ACM.

23. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2017). Video shot boundary detection via sequential clustering. *International Journal* «*Information Theories and Applications*, *24*(1), 50-59.

24. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2018, August). Representative Based Clustering of Long Multivariate Sequences with Different Lengths. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 545-548). IEEE.

25. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)* (pp. 930-935). IEEE.

26. Бодянський, Є. В., Дейнеко, А. О., & Куценко, Я. В. (2016). Послідовне нечітке кластерування на основі нейро-фаззі підходу. *Радіоелектроніка, інформатика, управління*, (3 (38)).

27. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. *Information Technology and Management Science*, *19*(1), 23-28.

28. Schafer, J. L., & Graham, J. W. (2002). Missing data: our view of the state of the art. *Psychological methods*, *7*(2), 147.

29. Волкова, В. В., & Шафроненко, А. Ю. (2011). Нечітка кластеризація масивів даних з пропущеними значеннями. *Індуктивне моделювання складних систем*.

30. Kesemen, O., Tezel, Ö., & Özkul, E. (2016). Fuzzy c-means clustering algorithm for directional data (FCM4DD). *Expert systems with applications*, *58*, 76-82.

31. Женбинг, Х., Бодянский, Е. В., Тыщенко, А. К., & Ткачев, В. Н. (2017). Fuzzy Clustering Data Arrays with Omitted Observations.

32. Kate, R. J. (2016). Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery*, *30*(2), 283-312.

33. Hu, Z., Mashtalir, S. V., Tyshchenko, O. K., & Stolbovyi, M. I. (2018). Clustering matrix sequences based on the iterative dynamic time deformation procedure. *International Journal of Intelligent Systems and Applications*, *10*(7), 66-73.

34. Wang, D., Lu, X., & Rinaldo, A. (2017). DBSCAN: Optimal Rates For Density Based Clustering. *arXiv preprint arXiv:1706.03113*.

35. Tiwari, K. K., Raguvanshi, V., & Jain, A. (2016). DBSCAN: An Assessment of Density Based Clustering and It's Approaches.

36. Bodyanskiy, Y., Shafronenko, A., & Mashtalir, S. (2019, May). Online Robust Fuzzy Clustering of Data with Omissions Using Similarity Measure of Special Type. In *International Scientific Conference* «*Intellectual Systems of Decision Making and Problem of Computational Intelligence*» (pp. 637-646). Springer, Cham.

37. Mashtalir, S. V., Stolbovyi, M. I., & Yakovlev, S. V. (2019). Clustering Video Sequences by the Method of Harmonic k-Means. *Cybernetics and Systems Analysis*, *55*(2), 200-206.

38. Gautam, C., & Ravi, V. (2015). Data imputation via evolutionary computation, clustering and a neural network. *Neurocomputing*, *156*, 134-142.