

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)

РОЗРОБКА ТА ДОСЛІДЖЕННЯ МЕТОДУ КЛАСТЕРИЗАЦІЇ

ДЛЯ ПРОГНОЗУВАННЯ

(тема)

Виконав:

студент 2 курсу, групи ІНФМ-18-2

Вискребенцева С.О.

(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Інформатика

(повна назва освітньої програми)

Керівник Кобилін О.А.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

2019 р.

Путятін Є.П.

(прізвище, ініціали)

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)Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 20 ____ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУстудентові Вискребенцевій Світлані Олександрівні
(прізвище, ім'я, по батькові)1. Тема роботи Розробка та дослідження методу кластеризації для прогнозування

затверджена наказом по університету від « 21 » жовтня 2019 року № 1506Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 22 листопада 2019 р.3. Вихідні дані до роботи Математичні моделі кластеризації данихперелік використовуваних програмних засобів:теоретичні відомості про методи кластеризації даних з пропускамипрогнозування при обробці часових рядів

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд методів кластеризації даних2. Огляд методів обробки даних з пропусками3. Математична модель алгоритму нечіткої кластеризації FCM4. Математичні моделі кластеризації часових рядів5. Практична оцінка методів, що аналізуються

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Актуальність проблеми аналізу методів кластеризації даних для прогнозування, постановка задачі, тестові зображення

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 | 21.10.2019 | виконано |
| 2 | Аналіз завдання, підбір літератури | 22.10.19-24.10.19 | виконано |
| 3 | Аналіз літератури з досліджуваної проблеми | 25.10.19-31.10.19 | виконано |
| 4 | Аналіз технічних засобів | 01.11.19-04.11.19 | виконано |
| 5 | Розробка методу | 05.11.19-07.11.19 | виконано |
| 6 | Програмна реалізація | 07.11.19-10.11.19 | виконано |
| 7 | Оформлення пояснювальної записки | 11.11.19-18.11.19 | виконано |
| 8 | Перевірка на плагіат | 19.11.19 | виконано |
| 9 | Рецензування | 20.11.19 | виконано |
| 10 | Підготовка презентації та доповіді | 20.11.19 | виконано |
| 11 | Занесення роботи в електронний архів | 21.11.19 | виконано |
| 12 | Попередній захист атестаційної роботи | 22.11.19 | виконано |
| | | | |
| | | | |

Дата видачі завдання 21 жовтня 2019 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Кобилін О.А. _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до атестаційної роботи: 79 с., 3 табл., 34 рис., 40 джерел.

КЛАСТЕРНИЙ АНАЛІЗ, НЕПОВНІ ДАНІ, ОБРОБКА ТА АНАЛІЗ ДАНИХ, DATA MINING, FCM, МЕТОДИ ВІДНОВЛЕННЯ ДАНИХ, ЧАСОВІ РЯДИ, ПРОГНОЗУВАННЯ.

Об'єктом дослідження є часові ряди з пропущеними значеннями, що піддаються кластеризації для прогнозування.

Метою є розгляд існуючих методів позбавлення від пропусків в даних в задачах кластеризації та доцільність їх використання в реальних задачах. Вирішується задача кластеризації та прогнозування часових рядів.

Розглянуто найпоширеніші методи кластеризації даних та позбавлення від пропусків в задачах кластеризації даних. Дослідження проблеми кластеризації даних, що містять пропущені значення та розгляд методів, які дозволяють розв'язати цю задачу. Проведення експериментів та порівняння результатів кожного з методів, висновки про доцільність використання того чи іншого методу та побічні ефекти.

У результаті роботи проведено аналіз методів кластеризації та прогнозування часових рядів, що можуть містити пропуски.

CLUSTER ANALYSIS, INCOMPLETE DATA, DATA PROCESSING AND ANALYSIS, DATA MINING, FCM, DATA RECOVERY METHODS, TIME SERIES, FORECASTING.

The object of the study is time series with missing values, which are clustered for forecasting.

The goal is to consider existing methods for getting rid of data gaps in clustering problems and the expediency of their use in real-world problems. The task of clustering and forecasting time series is being solved.

The most common methods of data clustering and getting rid of gaps in data clustering problems are considered. Investigation of the problem of clustering data containing missing values, and consideration of methods that can solve this problem. Conducting experiments and comparing the results of each of the methods, conclusions about the advisability of using a particular method and side effects.

As a result of implemented, the analysis of clustering methods and forecasting of time series, which may contain omissions.

ЗМІСТ

| | |
|--|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів | 7 |
| Вступ..... | 8 |
| 1 Огляд основних методів кластеризації часових рядів та роботи з пропусками..... | 10 |
| 1.1 Існуючі методи кластеризації даних з пропусками..... | 12 |
| 1.2 Прості методи заповнення пропусків | 15 |
| 1.2.1 Виключення рядків з наявністю пропущених значень | 15 |
| 1.2.2 Кластеризація із заповненням пропусків вибірковими статистиками | 15 |
| 1.2.3 Заповнення пропусків з урахуванням структури зв'язків | 16 |
| 1.3 Складні методи заповнення пропусків | 17 |
| 1.3.1 Приклади глобальних методів | 17 |
| 1.3.2 Приклади локальних методів..... | 19 |
| 1.4 Методи кластеризації даних, алгоритм k-середніх | 20 |
| 1.4.1 Метрики й відстані..... | 25 |
| 1.4.2 Міри близькості..... | 27 |
| 1.4.3 Метод k-середніх..... | 29 |
| 1.5 Особливості кластеризації часових рядів..... | 31 |
| 1.6 Метод нечіткої кластеризації даних k-середніх (FCM) | 34 |
| 1.7 Постановка задачі дослідження..... | 38 |
| 2 Математичні моделі кластеризації та прогнозування часових рядів | 43 |
| 2.1 Алгоритм fuzzy k-means (c-means) | 43 |
| 2.2 Алгоритм динамічної трансформації часової шкали (DTW) | 45 |
| 2.3 Алгоритм DBSCAN | 48 |
| 2.4 Робота з пропусками з метою подальшого прогнозування..... | 50 |
| 3 експериментальні дослідження методів прогнозування і кластеризації часових рядів | 52 |
| 3.1 Розгляд предметної області | 52 |

| | | |
|-------|---|----|
| 3.2 | Практичні результати застосування простих методів обробки даних з пропусками..... | 53 |
| 3.2.1 | Опис даних, що використано для проведення аналізу методів..... | 53 |
| 3.2.2 | Результати застосування методів | 55 |
| 3.3 | Практичні результати прогнозування та кластеризації часових рядів | 64 |
| 3.3.1 | Прогнозування ряду та побудова SARIMA моделі | 66 |
| 3.3.2 | Кластеризація часового ряду | 71 |
| | Висновки | 74 |
| | Перелік джерел посилання | 75 |

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

КА – кластерний аналіз

ПО – предметна область

НМ – нечітка множина

DTW – dynamic time warping

ВСТУП

Все більший інтерес породжує сфера обробки і аналізу даних. Якщо раніше обчислювальні машини, а разом з ними і комп'ютерні науки розвивалися досить повільно і основний інтерес становив розвиток саме апаратної бази: збільшення пам'яті, як оперативної так і жорстких дисків, загальна швидкість обчислень та інше. Розвиток програмного забезпечення був обмежений цими характеристиками, необхідно було розв'язувати проблеми працездатності програми в таких умовах, думати про те як скоротити ресурси, що використовуються, зменшити розміри коду. Це ті задачі, які ставилися і вирішувались раніше і були тоді найбільш популярними.

З доволі швидким розвитком комп'ютерних технологій і наук пов'язаних з тим, задачі, що вирішуються, стали змінюватися. На сьогодні проблема з пам'яттю або обчислювальними ресурсами не стоїть так гостро, апаратна база стрімко розвивається і більше не становить основний інтерес. Ще однією рушійною силою є те, що комп'ютери стали все більш доступними і вже доволі тривалий час не малий вклад в розвиток роблять прості користувачі. Враховуючи це, збільшується кількість інформації, що потрапляє в комп'ютер і безпосередньо в Інтернет.

Зі збільшення об'ємів даних стали виникати нові задачі, все більший інтерес становить робота з даними і розв'язання проблем, що пов'язані з їх обробкою і подальшим аналізом. Оскільки можливості створення нового контенту мають всі бажаючі – обсяги даних непомірно зростають, а їх впорядкованість досить слабка. З'являється необхідність пошуку необхідних даних, в тому числі зображень, їх обробки, для маркетингових і статистичних досліджень, використанню у інших сферах, а також для простого користувача.

Найпопулярнішими напрямками досліджень 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.

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, що особливо актуально в умовах безкінечно зростаючого об'єму інформації. Такий підхід дозволяє: краще зрозуміти дані, шляхом використання для кожного кластеру найбільш підходящого алгоритму аналізу; провести стиснення, виділивши найбільш типових представників, за умов збитковості даних; виявлення новизни, шляхом виділення об'єктів, що не потрапили до жодного з кластерів.

Так, наприклад, однією з проблем кластеризації можна виділити роботу з пропущеними значеннями, а саме роботу з часовими рядами. Існує безліч методів, але вони не передбачають відсутності якоїсь кількості інформації. Але, як тільки ми виходимо за рамки тестових даних і переходимо до обробки реальних - стикаємося з проблемою з цією проблемою, адже в дійсності ідеальних даних не існує і всі вони містять шуми (некорисну інформацію, яка не вплине або навіть зашкодить результату), пропуски невідповідні формати та інше.

Таким чином є доцільним розглянути варіанти рішення проблеми кластеризації даних з пропусками, які існують методи та їх доцільність.

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ КЛАСТЕРИЗАЦІЇ ЧАСОВИХ РЯДІВ ТА РОБОТИ З ПРОПУСКАМИ

За тривалий час розвитку людства та, безпосередньо, комп'ютерної техніки було накопичено досить велику кількість інформації. Слід зазначити, що ті дані, які на сьогодні, є в нашому розпорядженні не завжди добре структуровані та підготовлені одразу. Найчастіше, та інформація, що доступна є досить хаотичною, містить шуми (тобто некорисну інформацію, яка, скоріше за все, нам не знадобиться), дефекти. Тому зараз актуальними питаннями є зберігання та обробка цих даних.

Зважаючи на сучасні потреби все більш актуальними та затребуваними є розробки в галузі аналізу накопичених даних, адже ми живемо в той час коли кількість інформації, якою ти володієш, вже менш значуща ніж її якість і можливість обробити, зробити висновки на цьому підґрунті.

Одне з місць в обробці таких даних займає кластеризація. 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 [1]. 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.

На рисунку 1.1 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 кожен 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.

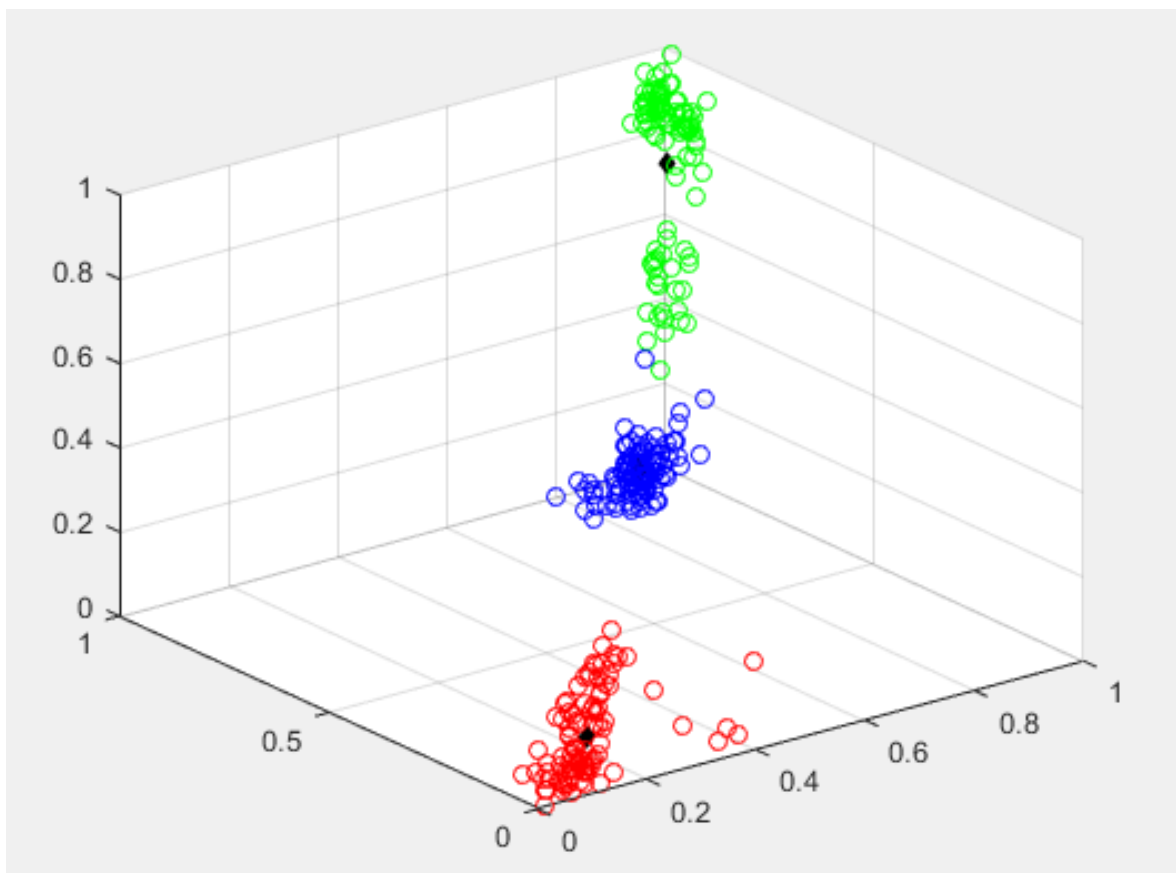


Рисунок 1.1 – Приклад результату кластеризації тестових даних у форматі 3D

Більш детально розглянемо приведений процес кластеризації [1]:

Позначимо множину об'єктів, що спостерігається, як $\tau = \{x_i\}_{i=1}^n$ і складається з набору атрибутів $x_i = \{t_1^i, t_2^i, \dots, t_m^i\}$, де t_i^y приймає значення із заданої множини T_v^i . Завдання кластеризації полягає в побудові множини

$C = \{c_v\}_{v=1}^k$ і відображення $F: \tau \rightarrow C$ заданої множини об'єктів на множину кластерів.

Кластер містить записи з τ схожі (за заданим критерієм) один на одного $x_i \in c_v, x_j \in c_v \Rightarrow d(x_i, x_j) < \varepsilon$,

де $d(x_i, x_j)$ – міра близькості між об'єктами (відстань), а ε - максимальне значення порога, що формує один кластер.

Сфера застосування може бути доволі широка, використання в сегментації зображень [2], прогнозування, аналіз текстів, оптимізація, машинне навчання. Якщо детальніше розглянути сфери застосування, можна побачити, що вони самі по собі є досить важливими і актуальними, сегментацію зображень можна використовувати для діагностики біологічного або біомедичного матеріалу, так у джерелі [3] наводяться основні етапи сегментації цитологічних зразків на основі біонічних пікселів. Важливість аналізу та комп'ютерної обробки текстової інформації наводиться в роботі [5, 6].

Як вже було зазначено, 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.

1.1 Існуючі методи кластеризації даних з пропусками

Отже, існують декілька можливих розв'язання задачі обробки даних з пропусками. Їх умовно можна поділити на такі, що націлені на попередню

підготовку даних і ті, що вимагають адаптації (модифікації) стандартних алгоритмів кластеризації для обробки безпосередньо даних з пропусками. Кожні з цих варіантів має свої недоліки та переваги, і не є універсальними для боротьби з цією проблемою.

При застосуванні першого варіанту позбавлення від пропусків також є декілька підходів. По-перше, при підготовці даних є варіант видалення наявних пропусків. В такому випадку можна позбутися зайвої інформації, яка перешкоджає якісному аналізу даних, однак існує суттєвий недолік, втрачається інформація, яка може бути корисною, і якщо таких пропусків велика кількість, то є ризик видалити занадто багато і зробити подальший аналіз неможливим, або неінформативним. В цьому випадку можна видалити не всі наявні дані, а лише ті за якими відбувається аналіз, або пропуски в яких є неприйнятними.

По-друге, відбувається заміна пропусків на їх оцінки, найчастіше це може бути середнє значення всієї вибірки, мода, медіана та інше і регресивні моделі, коли невідоме значення ознаки обчислюється за допомогою знайденої функції регресії за відомим ознаками. Широке поширення набув EM-алгоритм, який передбачає вірогідну модель побудови вибірок.

Другий варіант полягає в використанні алгоритмів, що вже існують для застосування у обробці повних даних та адаптацію їх до даних з пропусками. У цьому випадку нема потреби проводити попередню обробку, а саме відновлення, але алгоритми, що все існують, потребують модифікації. При такому підході не відбувається втрати даних і їх спотворення, на відміну від попереднього випадку, адже коли відсутня інформація заповнюється штучними даними це погіршує правильне отримання результатів. Однак до недоліків слід віднести необхідність розробки нових алгоритмів, або адаптацію наявних для вирішення такої задачі.

На рисунку 1.2 наведено графік існуючих алгоритмів заповнення пропусків в таблицях.

Отже, розглянувши основні методи та можливості обробки, а саме кластеризації даних з пропусками можна стверджувати, що на сьогодні не існує універсальної моделі, яка б розв'язувала цю проблему на будь-яких наборах даних. Слід обирати шляхи вирішення тієї чи іншої проблема попередньо проаналізувавши похідні дані і виходячи з них робити висновок про обраний шлях вирішення задачі.

Результати порівняння різних підходів показують, що при вирішенні практичних завдань апріорі не ясно, який з підходів виявиться найбільш прийнятним. Таким чином, створення нових підходів і алгоритмів кластеризації неповних даних є актуальним завданням.

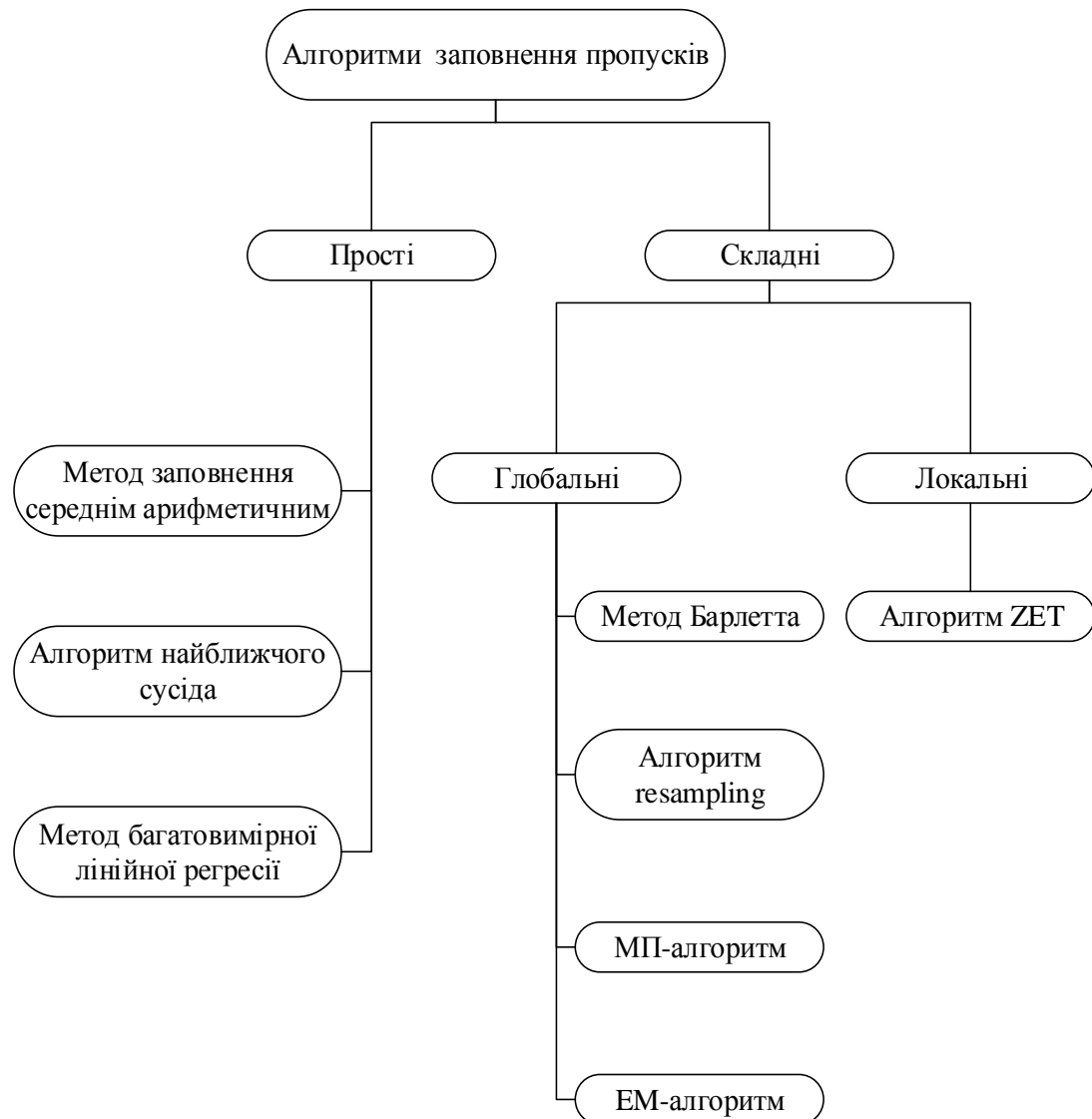


Рисунок 1.2 – Класифікація алгоритмів заповнення пропусків

1.2 Прості методи заповнення пропусків

1.2.1 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.

1.2.2 Кластеризація із заповненням пропусків вибірковими статистиками

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

підхід, який полягає у припущенні, 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.

1.2.3 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.

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.

Ефективність використання даних методів позбавлення від пропусків в даних для подальшого їх використання в задачах кластеризації розглянуто в роботі [7].

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.

1.3 Складні методи заповнення пропусків

Як представлено на рисунку 1.2 складні методи поділяються на локальні та глобальні. Детальний розгляд методів наводиться в роботі [6].

1.3.1 Приклади глобальних методів

Метод Бардетта – це метод, що базується на рівняннях лінійної регресії. Містить декілька кроків, які дозволяють відновити пропущені значення. Спочатку всі пропуски заповнюються певними значеннями, у якості яких може слугувати те саме середнє значення, для обраних даних. Далі, враховуючи початкове значення, обране на попередньому кроці, будується регресійна модель. Ця модель надалі використовується для прогнозування нового значення для пропуску, що оброблявся.

Наступний метод – Resampling. Він полягає в тому, що на місця пропусків поміщаються довільні дані з тих, що наявні в наборі. Оскільки алгоритм є ітеративним, на кожній з ітерацій будується регресійна модель (отримуються рівняння лінійної регресії). Всі отримані рівняння аналізуються і підсумкове використовується для остаточного прогнозування з метою заповнення пропусків.

Перевагою методу є можливість отримати більш точного прогнозу за рахунок ітераційності зазначеного процесу.

МП-алгоритми - методи, що засновані на оцінках максимальної правдоподібності. Їх суть полягає у побудові моделі породження пропусків, згодом робляться висновки на підґрунті функцій правдоподібності. Дані методи висувають більш слабкі умови до початкових даних, що забезпечується можливістю врахувати специфіку конкретної області. Сам факт необхідності будувати модель пропусків можна віднести до недоліків, як фактор, що ускладнює метод.

ЕМ-алгоритм, назва кого є аббревіатурою двох кроків, з яких він складається [1]:

Е-крок (expectation)

$$g_{i,j}^0 = g_{i,j};$$

$$g_{i,j} = \frac{\omega_i p_i(x_j)}{\sum_{v=1}^k \omega_v p_v(x_j)}, i = 1, \dots, k, j = 1, \dots, n. \quad (1.1)$$

$$\delta = \max \left\{ \left| g_{i,j}^0 - g_{i,j} \right| \right\}. \quad (1.2)$$

Таким чином, на цьому кроці, за допомогою регресії, відбувається оцінка значень, що можуть бути використані для заповнення пропусків.

М-крок (maximization)

$$\sum_{j=1}^n g_{i,j} \ln p_i(x_j) \rightarrow \max_{\Theta} , i = 1, \dots, k, \quad (1.3)$$

$$\omega_i = \frac{1}{n} \sum_{j=1}^n g_{i,j}, i = 1, \dots, k. \quad (1.4)$$

На цьому кроці розраховується нова коваріаційна матриця, враховуючи попередні заповнені пропуски.

Кроки Е та М виконуються доти, доки коваріаційна матриця не припинить змінюватися.

1.3.2 Приклади локальних методів

До локальних методів відновлення даних розглянемо ZET алгоритм, він у своїй роботі використовує лише необхідне частину даних, що обробляються.

ZET алгоритм базується на таких гіпотезах [8] як:

1) Гіпотеза збитковості, що полягає в припущенні про наявність збиткових даних в реальних таблицях, в яких рядки можуть бути схожі між собою, а стовпці знаходитися у відносинах залежності. Якщо такої збитковості не проявляється – алгоритм втрачає свою актуальність.

2) Гіпотеза локальної компактності. Припущення полягає у тому, що для прогнозування деякого об'єкта і-го рядка j-го стовпця необхідно використовувати лише певну частину даних, що складається з елементів рядків, що схожі на і-й рядок, та елементів стовпців, що схожі на j-й стовпець.

3) Гіпотеза лінійних залежностей, полягає у припущенні про лінійність зв'язків між стовпцями та строками. Для кожного окремого пропуску є тільки певна кількість схожих рядків та стовпців.

Отже, на початку, для кожного пропуску, відбувається пропуск необхідних рядків та стовпців, враховуючи наведені умови. Потім використовується лінійна регресія задля прогнозування значень пропущених даних.

1.4 Методи кластеризації даних, алгоритм k-середніх

Виходячи з [9], можна сказати, що методи кластеризації поділяються на ієрархічні та алгоритми розподілу (неієрархічні) алгоритми.

В процесі ієрархічної кластеризації відбувається злиття і поділ кластерів під час побудови дерева вкладених кластерів (дендрограми). На рисунку 1.4 наведено результат ієрархічної кластеризації та приклад дендрограми. Ієрархічні методи також поділяються на:

- агломеративні, на початку алгоритму кожен елемент – це окремий кластер, подальші кроки кластери поєднуються в один. Таким чином суть метода полягає в об'єднанні кластерів, зменшенні їх кількості;
- дивизивні (роз'єднувальні), що передбачають один кластер на початку і подальше його розділення на більшу кількість кластерів.

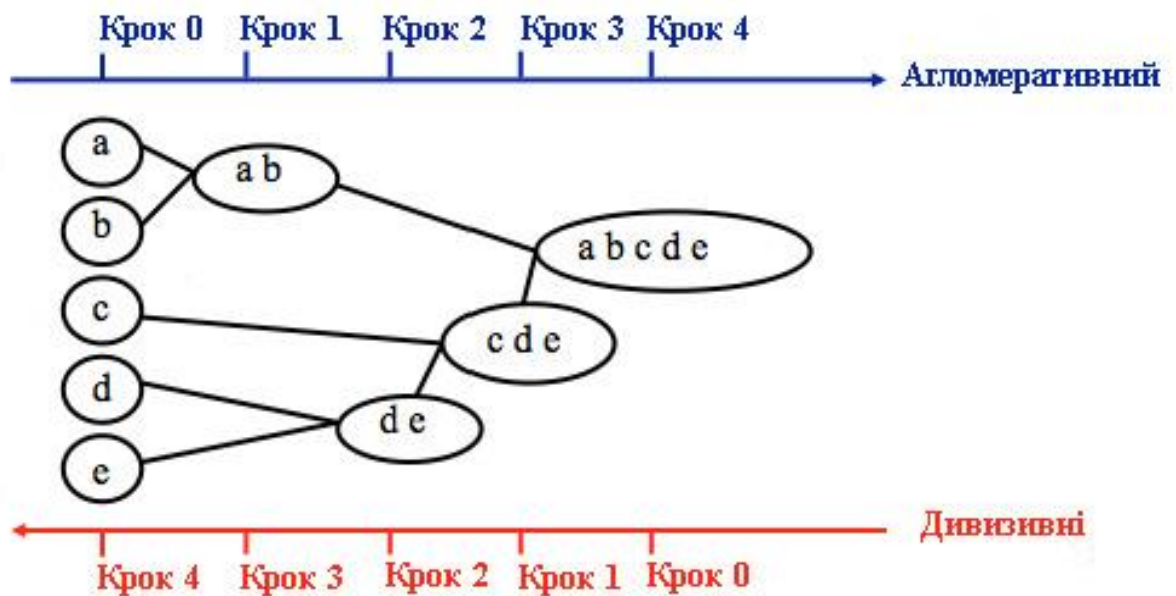


Рисунок 1.3 – Ієрархічні методи кластерного аналізу

При використанні агломеративних ієрархічних алгоритмів можна використовувати одне з правил об'єднання двох кластерів [9]:

- Метод ближнього сусіда (одиначний зв'язок)

Даний метод полягає у об'єднанні двох найбільш близьких об'єктів серед різних кластерів, близькість визначається схожістю об'єктів у матриці збігів. На кожному з кроків продовжується пошук найбільш схожих сусідів і це відбувається до тих пір, поки матриця збігів не буде вичерпана.

До переваг методу можна віднести його простоту серед подібних методів, а також можливість виділяти кластери форми будь якої складності. Однак, метод містить і недоліки, він потребує критерій порогового рівня подібності за яким буде припинено об'єднання, та, як побічний ефект може утворитися один загальний кластер, що не відповідає цілям кластеризації.

$$\rho_{\min}(S_l, S_m) = \min_{x_i \in S_l; x_j \in S_m} \rho(x_i, x_j). \quad (1.5)$$

- Метод найбільш віддаленого сусіда (повний зв'язок)

Метод полягає у визначенні відстані між кластерами за найбільшою відстанню між будь-якими двома об'єктами в різних кластерах. Метод, на відміну від попереднього, не варто використовувати, якщо кластери мають подовжену форму.

$$\rho_{\min}(S_l, S_m) = \max_{x_i \in S_l; x_j \in S_m} \rho(x_i, x_j). \quad (1.6)$$

- Незважений центроїдний метод

За даним методом, у якості відстань між двома кластерами, визначається відстань між їх центрами ваги

$$\rho_{gc}(S_l, S_m) = \rho(\overline{x_j}, \overline{x_m}). \quad (1.7)$$

де $\overline{x_j}, \overline{x_m}$ - вектори середніх S_l та S_m відповідно.

- Зважений центроїдний метод

Метод аналогічний попередньому, але передбачає розрахунок з урахуванням того, що при визначенні розмірів кластерів використовуються ваги. Тому, у випадках, коли відомо про відмінності у розмірах кластерів бажаніше використовувати саме цей метод, у порівнянні з попереднім.

– Метод неваженого попарного середнього

За даним методом відстань між кластерами розраховується як середня відстань між усіма парами об'єктів в цих кластерах.

$$\bar{\rho}(S_l, S_m) = \frac{1}{n_l n_m} \sum_{x_i \in S_l} \sum_{x_j \in S_m} \rho(x_i, x_j). \quad (1.8)$$

Метод добре працює при використанні у роботі з кластерами протяжної форми та у випадках коли відомо про різні розміри кластерів.

– Метод зваженого попарного середнього

Метод аналогічний попередньому, відрізняється тим, що при розрахунку розмір кластерів береться за ваговий коефіцієнт. Так само може бути використаний у випадках наявності кластерів різних розмірів, підходить для такої ситуації краще ніж попередній метод.

– Метод Варда (Ward's method)

На відміну від попередніх методів даний метод, для оцінки відстаней між кластерами використовує методи дисперсійного аналізу. В цьому випадку відстань між кластерами визначається як приріст суми квадратів відстаней об'єктів до центрів кластерів. Це досить ефективний метод, але за результатами його використання можна отримати кластери малого розміру.

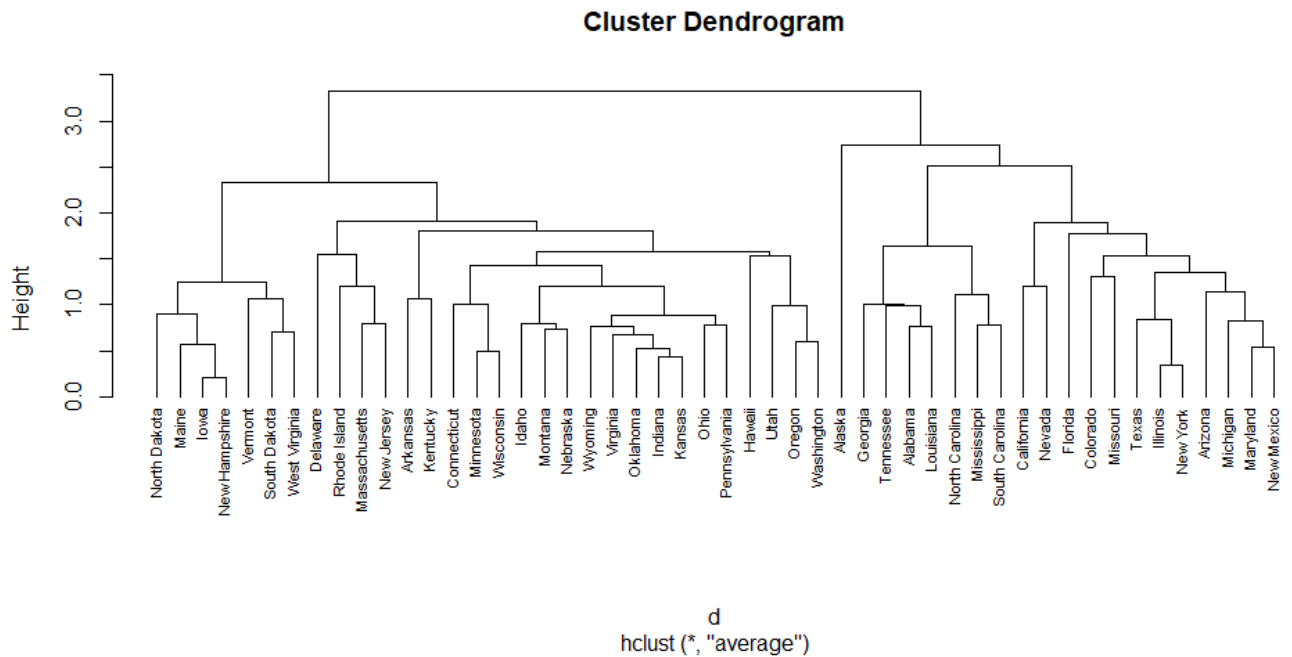


Рисунок 1.4 – Приклад дендрограми

Неієрархічні методи відрізняються тим, що потребують наявності умови зупинки і кількості кластерів. Це ітеративний процес поділу на кластери до тих пір поки не буде виконана умова зупинки. При великій кількості спостережень неієрархічні методи КА більш зручні оскільки у такому випадку дендограми стають перевантаженими і втрачають наочність.

Існує два підходи до визначення границь кластерів, при використанні неієрархічних алгоритмів. За першим кластери визначаються в місцях найбільшого скупчення точок, другий підхід полягає у мінімізації міри відмінності об'єктів.

Сам алгоритм розбиття, в загальному випадку, уявляє собою ітеративний процес з певним переліком кроків:

- 1) На першому кроці обирається/задається певна кількість k кластерів, які необхідно отримати із вхідних даних за результатом КА.
- 2) На першій ітерації, випадковим чином, обираються k центри кластерів, у якості яких слугують початкові записи.
- 3) Далі, для кожного запису визначається найближчий до нього центр кластера.

4) Наступним кроком виконується обчислення нових центроїдів (центрів тяжіння). Після цього центри кластерів, з попередніх кроків, замінюються на розраховані центроїди

Останні два кроки виконуються ітераційно до тих пір, поки не буде виконано умову зупинки, у якості якої може бути перетин заданої максимальної кількості ітерацій, або похибка менше заданої ε , коли границі кластерів і розташування центроїдів перестають змінюватися.

Найбільш популярним методом неієрархічної кластеризації є метод найменших квадратів:

$$\sum_{j=1}^k \sum_{i=1}^{n_j} |x_i - s_j|^2 \rightarrow \min. \quad (1.9)$$

і його чисельна реалізація називається методом k-середніх.

Алгоритм k –середніх полягає у тому, що на початку обирається k довільних центри далі, за цими центрами решта множини розбивається на групи. На наступному кроці обчислюються нові центри для отриманих кластерів таким чином, щоб [10] квадрат евклідової відстані від елемента кластера до його центроїда був меншим ніж відстань до центроїдів решти кластерів.

У якості міри близькості зазвичай використовується Евклідова відстань:

$$\rho(x, y) = \|x - y\| = \sqrt{\sum_{p=1}^n (x_p - y_p)^2}. \quad (1.10)$$

де $x, y \in R^n$.

1.4.1 Метрики й відстані

Для формування кластерів в задачах КА використовуються метрики, які уявляють собою функцію, що визначає відстань між точками і класами в просторі R^p (метричний простір).

У роботі [11] наводяться визначення відстані і метрики. Так, $d_{i,j} = d(x_i, x_j)$ називається відстанню, якщо воно задовольняє наступним умовам:

- Невід'ємність: для всіх x_i та x_j має виконуватись умова $d(x_i, x_j) > 0$

$$\forall x_i, x_j : d(x_i, x_j) > 0. \quad (1.11)$$

- Ідентичність: $d(x_i, x_j) = 0$, тоді й тільки тоді, коли $x_i = x_j$ тобто об'єкти співпадають

$$x_i = x_j \Leftrightarrow d(x_i, x_j) = 0. \quad (1.12)$$

- Симетричність: умова виконується тоді, коли відстань від x_i до x_j дорівнює відстані від x_j до x_i

$$d(x_i, x_j) = d(x_j, x_i). \quad (1.13)$$

Для будь-яких трьох p -мірних точок x_i , x_j та x_l відстань $d(x_i, x_j)$ називається метрикою, якщо виконується додаткова умова:

- Нерівність трикутника: відстань від x_i до x_j завжди менше або дорівнює сумі відстаней від кожної з них до x_l .

$$d(x_i, x_j) \leq d(x_i, x_l) + d(x_l, x_j). \quad (1.14)$$

Після розрахунку відстані робиться висновок чи належить об'єкти одному кластеру таким чином, що при $d(x_i, x_j) < \sigma$ об'єкти розміщуються в один кластер, а у протилежному випадку об'єкти різні і будуть належати до різних кластерів.

У базовому алгоритмі нечітких k-середніх відстань між об'єктом і центром кластера розраховується через стандартну норму Евкліда., що наведено у формулі (1.6), яка є найбільш використовуваною для таких розрахунків, але існують і інші міри, що підбираються в залежності від даних, з якими необхідно працювати. Наприклад, Евклідова відстань використовується для роботи з кількісними даними, для них же можна застосувати Манхеттенську метрику (1.15) або відстань Махаланобіса (1.16) [13]:

$$d_{ik}^{(1)} = \sum_{j=1}^N |x_{ij} - x_{kj}|. \quad (1.15)$$

$$d_{ik}^M = (x_{ij} - x_{kj})^T W^{-1} (x_{ij} - x_{kj}). \quad (1.16)$$

де W – коваріаційна матриця вибірки $X = \{X_1, X_2, \dots, X_n\}$.

Норма Махаланобіса дозволяє виділяти кластери у вигляді гіпереліпсоїдів, вісі яких можуть бути орієнтовані в довільних напрямках.

Для номінальних (якісних) даних використовується міра подібності Хеммінга (1.17) та міра подібності Роджерса-Танімото (1.18) [13]:

$$\mu_{ij}^H = \frac{n_{ik}}{N}. \quad (1.17)$$

де n_{ik} – це число співпадаючих ознак у зразків X_i та X_k .

$$\mu_{ij}^{R-T} = n_{ik}^N (n_i' + n_k' - n_{ik}''). \quad (1.18)$$

де n_i' , n_k' - загальне число одиничних ознак у зразків X_i та X_k відповідно; n_{ik}'' - число одиничних ознак, що збігаються, у зразків X_i та X_k .

Для змішаних типів ознак можна використовувати відстань Журавльова [13]:

$$d_{ik} = \sum_{j=1}^N I_{ik}^j. \quad (1.19)$$

$$\text{де } I_{ik}^j = \begin{cases} 1, \text{ якщо } |x_{ij} - x_{kj}| < \varepsilon \\ 0, \text{ у іншому випадку} \end{cases}$$

Для наочного розгляду деяких наведених метрик на рисунку 1.6 наведено ізолінії деяких норм.

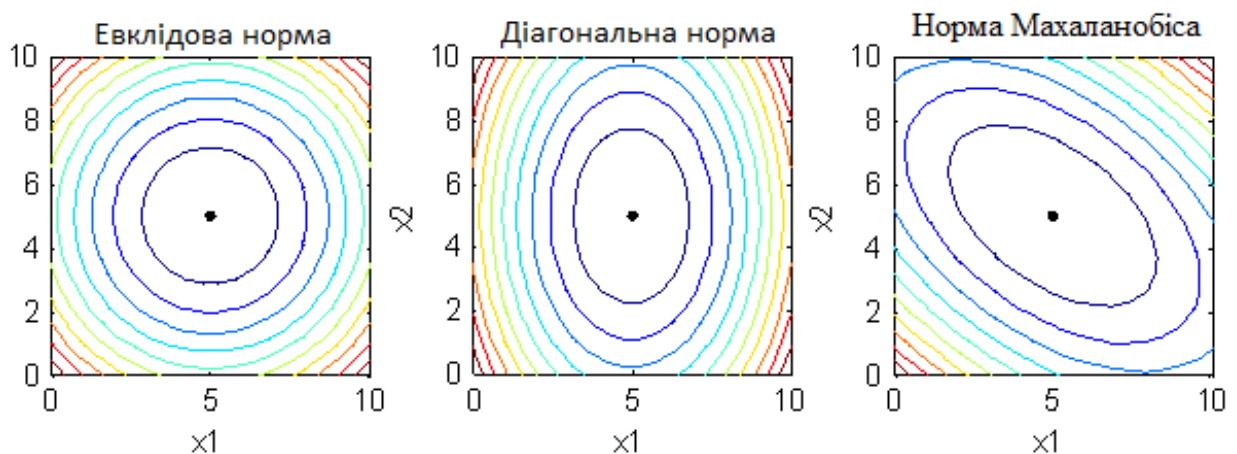


Рисунок 1.5 - Ізолінії різних норм

1.4.2 Міри близькості

Відстань $d(x_i, x_j)$ є різновидом міри близькості, однак існує ряд додаткових мір, які також можуть представляти інтерес.

Прикладом міри близькості є:

$$r(q_i, q_j) = \frac{v(q_i, q_j)}{v_0}. \quad (1.20)$$

де $v(q_i, q_j)$ – кількість однакових атрибутів у об'єктах q_i, q_j , що порівнюються; v_0 – загальне число атрибутів.

До мір близькості висуваються такі ж умови, як і розглянуті раніше, для відстаней (1.11-1.14), але на відміну від них, у даному разі виконання умови 1.14 не є обов'язковим.

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

Після розрахунку робиться висновок: при $r(q_i, q_j) < \sigma$ об'єкти розміщуються в один кластер, а у протилежному випадку об'єкти різні і будуть належати до різних кластерів.

Існує ряд асоціативних мір близькості:

Міра близькості Жаккара – це бінарна міра, що заснована на використанні інформації про мнодину загальних символів. Цю міру можна назвати першою відомою мірою близькості.

$$r_J(q_i, q_j) = \frac{c}{a+b+c}. \quad (1.21)$$

Міра близькості Сьоренсена – це статистичний коефіцієнт.

$$r_S(q_i, q_j) = \frac{2c}{a+b}. \quad (1.22)$$

де a і b число видів в зразках a і b , відповідно; c — число спільних для обох зразків видів.

Міра близькості Сокала-Сніта:

$$r_{SS}(q_i, q_j) = \frac{c}{2(a+b)+c}. \quad (1.23)$$

Міри близькості Дейка:

$$r_{SS}(q_i, q_j) = \frac{c}{2(a+b)+c}; \quad (1.24)$$

$$r_{D_2}(q_i, q_j) = \frac{c - \min(a, b)}{c + \min(a, b)}. \quad (1.25)$$

Міра близькості Кульчинського:

$$r_K(q_i, q_j) = \frac{a+b}{2ab}. \quad (1.26)$$

Міра незгоди Танімото:

$$r_T(q_i, q_j) = \frac{a+b}{a+b+c}. \quad (1.27)$$

Міра близькості Браун-Бланке:

$$r_B(q_i, q_j) = \frac{c}{\max(a, b)}. \quad (1.28)$$

1.4.3 Метод k-середніх

Метод k-середніх розділяє m спостережень на k груп (кластерів), щоб мінімізувати сумарне квадратичне відхилення точок кластерів від центроїдів кластерів:

$$\min \left[\sum_{i=1}^k \sum_{x^{(j)} \in S_i} \|x^{(j)} - \mu_i\|^2 \right]. \quad (1.29)$$

де $x^{(j)} \in R^n, \mu_i \in R^n$; μ_i – центроїд для кластера S_i .

Також, при використанні алгоритму слід звернути увагу на те, що k-means володіє низкою обмежень, таких як:

$$u_{ij}^{(l)} \in \{0,1\}; \sum_{j=1}^c u_{ij} = 1; 0 < \sum_{j=1}^d u_{ij} < d, \text{ тобто кожен вектор даних може належати}$$

лише одному кластеру і кожен кластер повинен містити не менше одного елемента і не більше загальної кількості елементів.

Використання алгоритму k-means для кластеризації супроводжується рядом проблем, а саме:

- необхідність заздалегідь визначити кількість результуючих кластерів, що не завжди може бути зручним, адже загалом необхідно експериментальним шляхом підбирати прийнятну, для даного набору даних, кількість кластерів, і для кожного оптимальна кількість буде своя. Існують додаткові методи, які дозволяють обчислювати цей параметр перед, безпосередньо, початком КА;

- алгоритм дуже чутливий до вибору початкових центрів кластерів, оскільки на першому кроці центри кластерів обираються випадковим чином – це призводить до зростання ймовірності похибки і можливості результатів відмінних між собою при повторному (багаторазовому) запуску алгоритму.

Також складнощі викликають ситуації приналежності об'єкта однаково до декількох кластерів або до жодного зі сформованих.

Попри недоліки k-means можна назвати досить простим алгоритмом, добре придатним для розуміння загальні процесів кластеризації і хорошою основою для побудови, на його базі, розширених версій, що покликані вирішувати більш вузько направлені задачі.

Одною з таких модифікацій цього методу можна вважати fuzzy k-means або c-means [14], який відрізняється тим, що кожен елемент кластеру належить до нього з певною ймовірністю, і його не обов'язково можна чітко віднести до одного з кластерів, тобто групи можуть перетинатися. Що загалом вирішує проблему приналежності об'єкта однаково до декількох кластерів або до жодного зі сформованих, оскільки визначає саме ступінь цієї приналежності.

Неієрархічні методи більш стійкі до шумів, неправильної метрики та наявності незначимих параметрів у порівнянні з ієрархічними методами, які виграють, у випадках з невизначеною кількістю кластерів, ітерацій, або умови зупинки. Також ієрархічні методи дозволяють більш детально вивчити структуру даних.

1.5 Особливості кластеризації часових рядів

Часовий ряд – це орієнтована у часі або хронологічна послідовність за предметною областю, що становить інтерес. Це спосіб представлення статистичних даних, з якими найчастіше може зіткнутися аналітик. Данні часових рядів використовуються в найрізноманітніших сферах людської діяльності, таких як фондові ринки, дані датчика, контроль стану машини, екологічні застосування або медичні дані.

Наприклад [14], на рисунку 1.7 показана ринкова прибутковість казначейських цінних паперів США з 10-річним постійним терміном погашення з квітня 1953 року по грудень 2006 року. Цей графік називається графіком часового ряду.

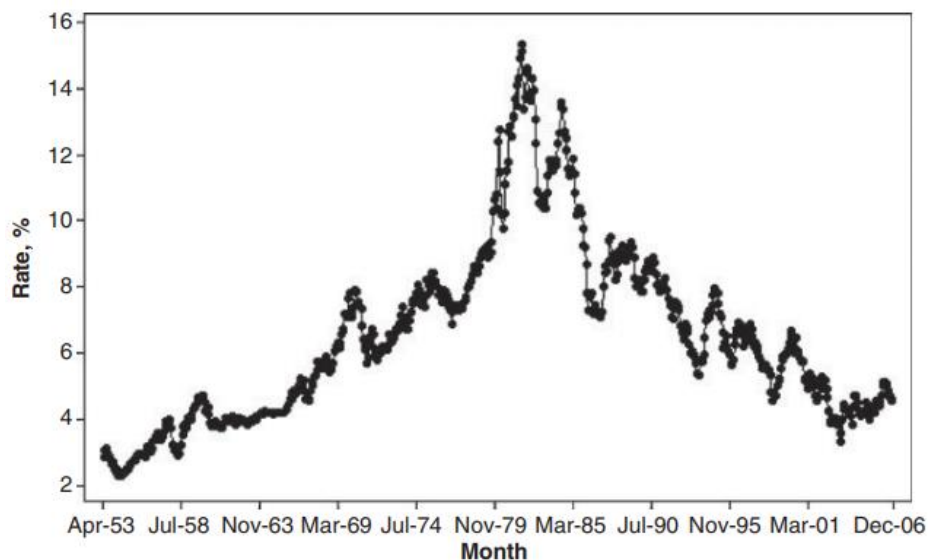


Рисунок 1.6 – Приклад графіку часового ряду

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 увагу 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 ряди 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.

Існує два види аномальних явищ:

- явища першого роду, що викликані технічними збоями і можуть бути виявлені і усунуті;

– до явищ другого роду відносяся ті, що мають епізодичний характер і не можуть бути усунуті.

Для того, щоб виявити такі значення використовується критерій Ірвіна, за ним аномальною точкою вважається Y_t , що відстає від попередньої точки Y_{t-1} на величину, більшу середньоквадратичного відхилення:

$$\lambda_i = \frac{|Y_t - Y_{t-1}|}{\sigma} \quad (1.30)$$

де λ_i - критерій Ірвіна, а σ - середньоквадратичне відхилення:

$$\sigma = \sqrt{\frac{\sum_{t=1}^n (Y_t - \bar{Y})^2}{n-1}} \quad (1.31)$$

Розрахувавши λ_i робиться висновок про аномальність точки, яка вважається такою у разі виконання умови: $\lambda_i > \lambda_{таб}$, де $\lambda_{таб}$ - це табличне значення критерію Ірвіна, частину якої наведено у таблиці 1.2.

Таблиця 1.1 – Табличні значення критерію Ірвіна

| | | | | |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| λ_1 | λ_3 | λ_5 | λ_7 | λ_9 |
| | 1,443582 | 1,443582 | 1,443582 | 1,443582 |
| $\lambda_{12} \lambda_{12}$ | $\lambda_{14} \lambda_{14}$ | $\lambda_{16} \lambda_{16}$ | $\lambda_{18} \lambda_{19}$ | $\lambda_{20} \lambda_{20}$ |
| 1,443582 | 1,443582 | 1,443582 | 1,443582 | 1,443582 |

Також існують випадкові коливання часового ряду, впоратися з ними може допомогти згладжування [16], яке має декілька методів:

Проста (середньоарифметична) ковзаюча середня:

$$Y_t = \frac{\sum_{i=t-p}^{t+p} Y_i}{2p+1}; p < t < n-p \quad (1.32)$$

Зважена (середньозважена) ковзаюча середня:

$$Y_t = \frac{\sum_{i=t-p}^{t+p} \rho_i Y_i}{\sum_{i=t-p}^{t+p} \rho_i}; p < t < n - p. \quad (1.33)$$

Среднехронологічна:

$$Y_t = \frac{\frac{Y_{t-T/2}}{2} \sum_{i=t-T/2+1}^{t+T/2-1} Y_i + \frac{Y_{t+T/2}}{2}}{T}; \frac{T}{2} < t < n - \frac{T}{2}. \quad (1.34)$$

Зазвичай T має значення 4 – квартали, або 12 – місяців, що дорівнює року.

Експоненційне згладжування:

$$Y_t = \frac{\sum_{i=1}^t \rho Y_i}{\sum_{i=1}^t \rho_i}. \quad (1.35)$$

Дане згладжування для даної точки враховує значення всіх попередніх точок.

1.6 Метод нечіткої кластеризації даних k-середніх (FCM)

Традиційні принципи кластерного аналізу припускають, що виділяються групи, де кожен об'єкт може належати тільки до однієї групи. Обмеженість такого підходу часто призводить до аналітичної невизначеності, і тому розумною альтернативою є інтерпретація компонентів систем як нечітких об'єктів. У роботі [17] розглядається застосування нечіткої

кластеризації для діагностики хвороби Альцгеймера, і робиться наголос, що чітка логіка дозволяє лише визначити, хворий пацієнт або ні, тоді як використання нечіткої логіки потенційно дає змогу оцінити «на скільки» хворий.

Цей алгоритм є узагальненням алгоритму k-means, в разі, якщо кластери є нечіткими множинами, і, елемент може належати різним кластерам з різним ступенем ймовірності. Дозволяє розбити наявну множину елементів потужністю N на задане число HM k .

Як все зазначалося основною відмінністю даного методу від попереднього є можливість приналежності елемента даних до декількох кластерів із певною вірогідністю, що визначається за допомогою функції належності. Функція належності – це поняття із задач, пов'язаних з нечіткими множинами, вона дозволяє визначити ступінь з яким елемент належить тій чи іншій множині (у даному випадку кластеру). Існує безліч методів для визначення цієї функції, такі як експертна оцінка, параметричній ідентифікації нечітких моделей, також продовжують пропонуватися і нові методи, наприклад [18] детальніше визначає ці поняття. Функція належності може приймати значення на інтервалі $[0, 1]$ у порівнянні з класичною логікою, яка однозначно визначає 0 або 1. Використання такої логіки значно розширяє можливості застосування її у вирішенні різноманітних задач (наприклад, кластеризації).

Алгоритм fuzzy k-means [19]:

Для того щоб вирішити певні проблеми під час кластеризації, такі як кількість кластерів і початкові центри кластерів, було введено термін для цільової функції FKM. Кластеризація X на c кластерів за цим алгоритмом повинна мінімізувати таку цільову функцію:

$$f[U, V] = \sum_{i=1}^n \sum_{k=1}^c u_{ik} d_{ik} + \gamma \sum_{i=1}^n \sum_{k=1}^c u_{ik} \log u_{ik}; \quad (1.36)$$

за умови

$$\sum_{k=1}^c u_{ik} = 1, u_{ik} \in (0,1], 1 \leq i \leq n, 1 \leq k \leq c. \quad (1.37)$$

де U – матриця розбиття n на c ;

V – матриця розбиття c на m , що містить центри кластерів;

d_{ik} – міра відмінності між k -м центром кластера та i -м об'єктом.

Процедура мінімізації, чергуючи член матриці U і центр кластера матриці V , може бути застосована, наступним чином:

$$u_{ik} = \frac{\exp\left(\frac{-d_{ik}}{\gamma}\right)}{\sum_{s=1}^c \exp\left(\frac{-d_{is}}{\gamma}\right)}; \quad (1.38)$$

$$v_k = \frac{\sum_{i=1}^n u_{ik} x_i}{\sum_{i=1}^n u_{ik}}. \quad (1.39)$$

Задачею FCM кластеризації є пошук матриці належності розмірності $c \times l$:

$$M = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1l} \\ m_{21} & m_{22} & \dots & m_{2l} \\ & & \dots & \\ m_{c1} & m_{c2} & \dots & m_{cl} \end{bmatrix}$$

де m_{ij} – ступінь приналежності j -го елемента i -му кластеру.

При цьому матриця M має задовольняти умовам:

$$- m_{ij} \in [0,1], i = \overline{1,c}; j = \overline{1,l};$$

- кожен об'єкт має бути розподілений між усіма

кластерами $\sum_{i=1}^c m_{ij} = 1, j = \overline{1,l};$

- жоден з кластерів не має містити всі елементи, а також лишатися пустим $0 < \sum_{j=1}^l m_{ij} < l, i = \overline{1, c}$;

Для оцінки якості розбиття використовується критерій розкиду:

$$J = \sum_{i=1}^c \sum_{j=1}^l (m_{ij})^w d(v_i, x_j). \quad (1.40)$$

де $d(v_i, x_j)$ – Евклідова відстань між j -м об'єктом $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})$ та i -м центром кластера $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$, що наведена у формулі (1.10).

$w \in (1, \infty)$ – експонентна вага, що визначає нечіткість, розмитість кластерів.

Також визначається матриця координат центрів кластерів розмірністю $c \times n$:

$$V = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \dots & \dots & \dots & \dots \\ v_{c1} & v_{c2} & \dots & v_{cn} \end{bmatrix}.$$

Елементи цієї матриці визначаються за формулою:

$$v_{ik} = \frac{\sum_{j=1}^l (m_{ij})^w x_{jk}}{\sum_{j=1}^l (m_{ij})^w}, k = \overline{1, n(v)}. \quad (1.41)$$

Застосування та актуальність нечіткої кластеризації наводиться в статті [20]. Головні переваги методу — його простота та швидкість виконання. Даний алгоритм в адаптованому вигляді, для багатомірних даних,

наводиться в [21], також наводяться експерименти і порівняльний аналіз з класичним методом FCM.

1.7 Постановка задачі дослідження

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.

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: біль 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.

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.

Метою роботи можна назвати розгляд існуючих методів позбавлення від пропусків в даних в задачах кластеризації та доцільність їх використання і реальних задачах. 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.

Завданням даної роботи є аналіз переваг та недоліків кожного з методів, що направлені на відновлення даних, для визначення доцільності використання їх в задачах кластеризації та виділення найбільш придатного до застосування, порівняння їх між собою, оцінка результативності за наслідками порівняння кластеризації відновлених даних з результатами кластеризації еталонних. Особлива увага приділяється методам нечіткої кластеризації FCM, метод видалення всіх рядків, що містять пропуски, заповнення пропусків вибілковими статистиками, заповнення пропусків з урахуванням структури зв'язків. Наукова новизна – дослідження проблеми кластеризації даних, що містять пропущені значення та розгляд методів, які дозволяють розв'язати цю задачу. Проведення експериментів та порівняння результатів кожного з методів, висновки про доцільність використання того чи іншого методу та побічні ефекти.

Практична значущість роботи полягає у визначенні можливості використання в реальних задачах, що зазвичай не є ідеальними і з великою ймовірністю міститимуть пусті значення, методів обробки даних для використання їх в задачах кластеризації.

Говорячи про актуальність і затребуваність даної теми слід дослідити способи застосування даних методів та розробки в даній сфері, зацікавленість серед дослідників. Можна сказати, що класичні методи кластеризації є досить широкою темою для досліджень і модифікації для вирішення різноманітних задач та різних типів даних. В роботі [22] розглядається підходи до кластеризації відео на основі аналізу багатовимірних часових рядів. Розглядаючи цю роботу можна зрозуміти, що КА може бути використано навіть в таких складних задачах, як обробка відео. Оскільки відеоданні є неструктурованими – це ускладнює загалом відеоаналіз, робить його складною науковою галуззю, що саме по собі висвітлює розробки більш цінними і значущими. Також робиться наголос на тому, що кластеризація у даному випадку та її продуктивність знаходиться в прямій залежності від обраної функції.

Проблеми обробки та кластеризації відео так само розглядаються в роботі [23, 24]. В ній піднімаються проблеми створення єдиної структури, що здатна відображати і передавати історії високого рівня. Не зважаючи на те, що сучасні методи здатні впоратись з цими нюансами, залишаються питання потреби досить високошвидкісних процедур.

Так, в роботі [25] наводиться гібридний алгоритм нечіткої кластеризації, він дозволяє обробляти біомедичні дані, що є досить важливим завданням, складні задачі обробки відеопотоків, обробка інформації з датчиків IoT, що стрімко набуває популярності та таке інше. Важливими перевагами є обчислювальна простота і можливість використовувати для різних об'ємів даних, як великих так і маленьких. Крім FCM кластеризації [26] також використовуються on-line нейро-фаззі систему для вирішення задач послідовного нечіткого кластерування даних.

Говорячи про обробку саме часових рядів, можна зробити висновок, що це питання, наразі, є досить актуальним, пошук пропусків та власне кластеризація володіють своїми складнощами. Розробляються методи адаптивної кластеризації [27] для вирішення задачі обробки коротких часових рядів з нерівномірно розподіленими спостереженнями.

Піднімається питання неможливості застосування евклідової метрики, або використання стохастичних критеріїв для визначення відстані між вибірками, наприклад для реалізації, що наведена на рисунку 1.7.

Що доводить досить широкі можливості методів КА які потребують модифікацій задля більш ефективного використання у різних випадках для різних даних.

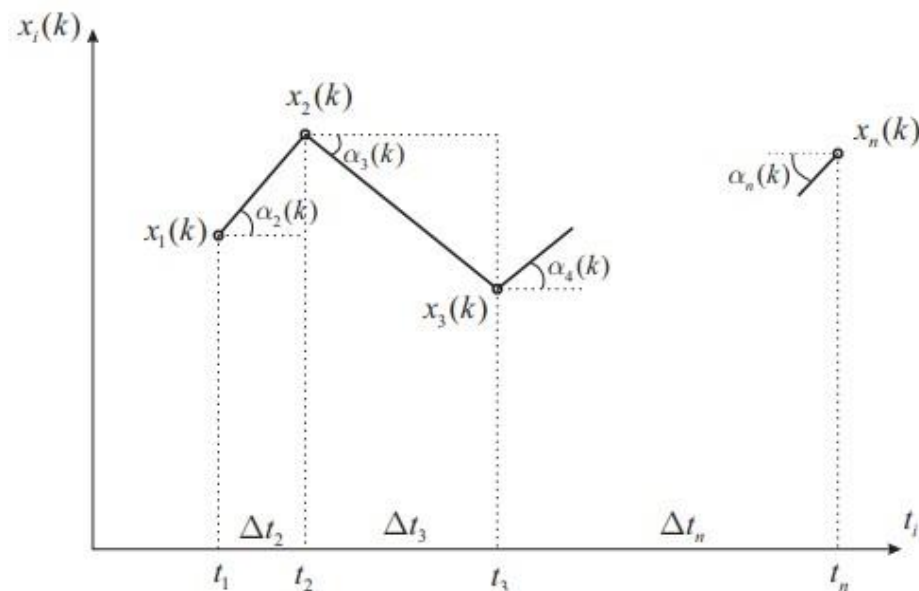


Рисунок 1.7 – Приклад часових рядів з неоднорідним тактом квантування.

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. Відсутні дані створюють багато труднощів, оскільки більшість процедур аналізу даних не були призначені для них [28].

Зважаючи на те, що в реальних даних велика ймовірність отримати таку ситуацію, коли інформація, що аналізується не повна, можна сказати, що відмовитися від кластеризації за цією причиною – не вихід, тому така задача потребує вирішення і є досить актуальною.

Отже існують декілька можливих рішень проблеми обробки даних з пропусками, які наведено на рисунку 1.2. Їх умовно можна поділити на такі, що націлені на попередню підготовку даних і ті, що вимагають адаптації (модифікації) стандартних алгоритмів кластеризації для обробки безпосередньо даних з пропусками. Кожен з цих варіантів має свої недоліки та переваги, і не є універсальними для рішення цієї проблеми.

Однак не зважаючи на наявність уже існуючих методів заповнення пропущених значень, ця тема також продовжує свій розвиток у модифікаціях методів задля отримання найкращих результатів у якомога більшій кількості

випадків. Так у роботі [28] розглядається розробка алгоритму заповнення пропущених даних на основі Адаліни. В даній роботі використовується непромережені системи, з паралельно працюючими адаптивними лінійними асоціаторами. Також наводиться схема (рисунок 1.8), на якій наочно можна побачити, як відбувається позбавлення від пропусків за допомогою однієї з реалізацій.

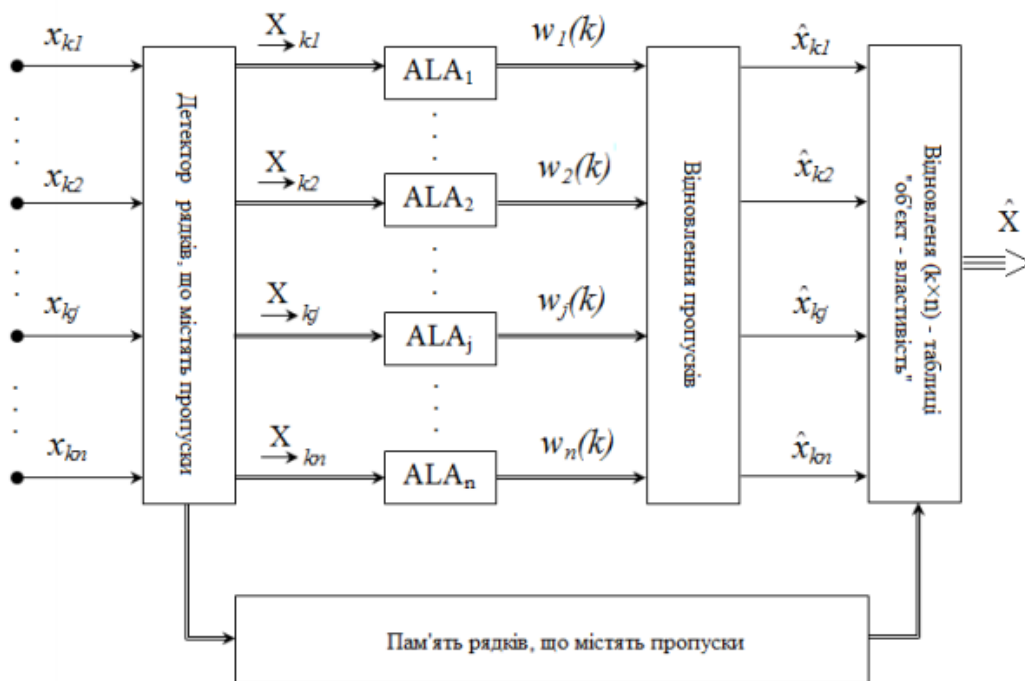


Рисунок 1.8 – Схема, що відображує один з варіантів відновлення пропусків в таблицях даних

2 МАТЕМАТИЧНІ МОДЕЛІ КЛАСТЕРИЗАЦІЇ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Для роботи з часовими рядами, а саме кластеризації, існує ряд можливих алгоритмів:

- k-середніх і його варіанти;
- ієрархічні алгоритми;
- засновані на щільності методи кластеризації, такі як DBSCAN.

Також важливим етапом обробки часових рядів є визначення багатомірний він чи одномірний.

2.1 Алгоритм fuzzy k-means (c-means)

Нечітка кластеризація складається з таких кроків [30, 31]:

Крок 1: Спочатку встановлюються параметри алгоритму, які включають c-кількість кластерів; m – експоненціальна вага; ε – параметр завершення алгоритму, для яких мають виконуватися умови: $c \in [2, N)$, $(m > 0)$ та $(\varepsilon > 0)$.

Можна зазначити, що m , строго більше 1, що вказує на показник ступеня членства, який використовується в критерії відповідності.

Слід зазначити, що $m \rightarrow 1$ дає більш чітку кластеризацію, тоді як $m \rightarrow Inf$ призводить до повної нечіткості. Значення, занадто близькі до 1, можуть привести до повільної конвергенції. Далі відзначимо, що навіть значення за замовчуванням $m = 2$ може привести до повної нечіткості, тобто членства $u(i, v) = 1 / k$. В цьому випадку рекомендується вибрати менше значення m .

Крок 2: Провести, випадковим чином, генерацію матриці нечіткого розбиття $\mu_{ij}^{(0)} \sim U(0, 1)$ та $let t = 1$.

Крок 3: Розрахувати центри кластерів (v_j), за допомогою формули:

$$v_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m}, (j = \overline{1, c}). \quad (2.1)$$

Крок 4: Отримати відстані від отриманих центрів (v_j) до об'єктів X .

$$D_{ij} = \sqrt{\|x_i - v_j\|^2}; i = \overline{1, N}; j = \overline{1, c}. \quad (2.2)$$

Крок 5: Оновити елементи матриці нечіткого розбиття μ_{ij} для всіх i, j , якщо $D_{ij} > 0$.

$$\mu_{ij} = \left(\sum_{k=1}^c \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}} \right)^{-1}, (i = 1, 2, \dots, N; j = 1, 2, \dots, C). \quad (2.3)$$

Якщо $D_{ij} = 0$, тоді $\mu_{ij} = \begin{cases} 1, \text{якщо } j = i \\ 0, \text{якщо } j \neq i \end{cases}$

Крок 6: Перевірити умову $\|\mu^{(t)} - \mu^{(t-1)}\|$.

Якщо $\|\mu^{(t)} - \mu^{(t-1)}\| < \varepsilon$, тобто різниця між матрицею нечіткого розбиття поточної і попередньої ітерації менше ніж заданий на початку розмір похибки ε . Якщо умова виконується, то відбувається зупинка алгоритму і на даному етапі будуть отримані кінцеві результати кластеризації. Якщо ж умова не виконується, то алгоритм знов починає розрахунки, переходячи на крок 3.

Проаналізувавши алгоритм можна сказати, що на першому ж кроці необхідно визначити кількість результуючих кластерів. Це важливий етап, і

як було зазначено раніше, саме він може викликати найбільше труднощів і неправильний вибір здатен негативно вплинути на результат.

Цю задачу можна вирішити емпіричним шляхом, тобто експериментально для заданого набору даних підібрати кількість кластерів, що найбільш підходить у даній конкретній ситуації. Наприклад, у разі нечіткої кластеризації, можна зробити висновок про необхідність зменшити кількість кластерів, якщо чітко видно, що одна зі сформованих груп повністю, або значною частиною, складається з елементів іншої/інших груп. Але існують методи, які дозволяють визначити цей параметр автоматично.

2.2 Алгоритм динамічної трансформації часової шкали (DTW)

Алгоритм DTW визначає оптимальну послідовність трансформації часу між двома часовими рядами та дозволяє знайти оптимальну відстань між ними.

Алгоритм динамічної трансформації часового ряду запропоновано як альтернативу евклідової метрики для обробки часових рядів. Адже говорячи про класичний спосіб розрахунку (евклідову метрику) слід зазначити, що вона володіє значним недоліком при рішенні таких задач. А саме, якщо один з двох рядів, що мають бути визначені, як однакові, буде зміщено у часі, тоді за евклідовою метрикою ряди можуть виявитися різними. Отже, для боротьби з цим недоліком було запроваджено метод DTW.

Різниця між Евклідовою метрикою і метрикою в методі DTW наведено на рисунку 2.1.

Модифікацією алгоритму DTW є алгоритм soft-DTW, при якому DTW відстань є:

$$DTW_{\gamma}(X_i, X_j) = -\gamma \log \sum_{k=1}^K e^{\left(\frac{d(w_k)}{K \cdot \gamma}\right)}. \quad (2.4)$$

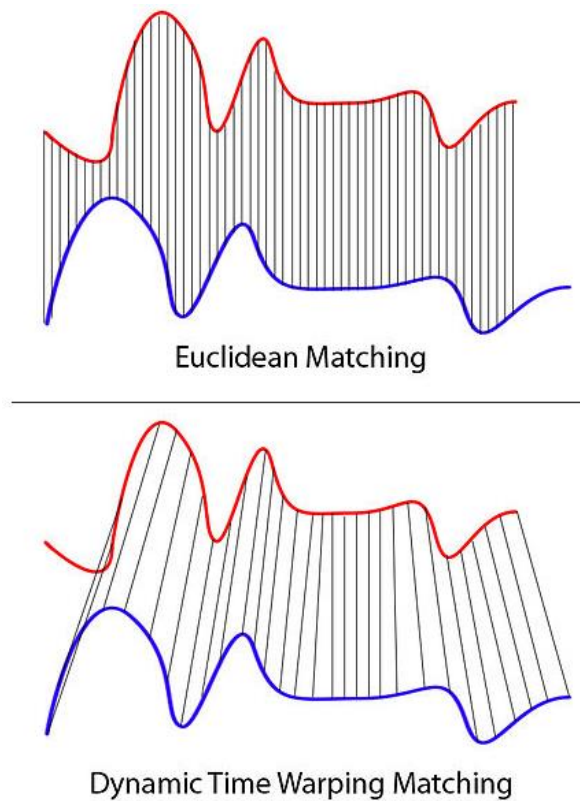


Рисунок 2.1 – Порівняння Евклідової метрики і DTW

Алгоритм складається з таких кроків [32]:

Крок 1: будується матриця відстаней $D = \{d_{i,j}\}$.

Таблиця 2.1 – Приклад матриці відстаней у методі DTW

| | y_1 | y_2 | y_3 | y_4 | y_5 | y_6 | y_7 |
|-------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| x_1 | Δ_{11} | Δ_{12} | Δ_{13} | Δ_{14} | Δ_{15} | Δ_{16} | Δ_{17} |
| x_2 | Δ_{21} | Δ_{22} | Δ_{23} | Δ_{24} | Δ_{25} | Δ_{26} | Δ_{27} |
| x_3 | Δ_{31} | Δ_{32} | Δ_{33} | Δ_{34} | Δ_{35} | Δ_{36} | Δ_{37} |
| x_4 | Δ_{41} | Δ_{42} | Δ_{43} | Δ_{44} | Δ_{45} | Δ_{46} | Δ_{47} |
| x_5 | Δ_{51} | Δ_{52} | Δ_{53} | Δ_{54} | Δ_{55} | Δ_{56} | Δ_{57} |

Зазвичай використовується евклідова відстань (1.10), або різниця за координатами, для одного виміру це становить:

$$d_{i,j}(x_i, x_j) = |x_i - x_j|. \quad (2.5)$$

Кожен елемент матриці відповідає відстані між точками x_i, x_j .

Крок 2: будуємо матрицю трансформації $D_{DTW} = \{r_{i,j}\}$, елементи якої розраховуються за формулою (2.6):

$$r_{i,j} = d_{i,j} + \min(D_{i-1,j}, D_{i-1,j-1}, D_{i,j-1}). \quad (2.6)$$

Крок 3: побудова оптимального шляху трансформації і DTW відстань.

Шлях трансформації W встановлює відповідність між X_i і X_j , та визначається за формулою:

$$W = \{w_k\}_{k=1}^K, \quad w_k = (i, j)_k, \quad \max(n_i, n_j) \leq K < (n_i + n_j). \quad (2.7)$$

де K – це довжина шляху.

Також, шлях трансформації має задовольняти умовам:

- має містити всі точки обох часових рядів

$$w_1 = (1,1), \quad w_K = (n_i, n_j). \quad (2.8)$$

- безперервність. Шлях трансформації пересувається на один крок за один раз

$$\forall w_k = (w_i, w_j), w_{k+1} = (w_{i+1}, w_{j+1}): w_i - w_{i+1} \leq 1, w_j - w_{j+1} \leq 1. \quad (2.9)$$

- монотонність. Обмеження гарантує, що повернення до попередньої точки не відбудеться, тобто індекси не зменшуватимуться.

DTW відстань (вартість шляху) між двома послідовностями розраховується за допомогою формули:

$$DTW(X_i, X_j) = \min \left(\frac{\sum_{k=1}^K d(w_k)}{K} \right). \quad (2.10)$$

Загалом методи DTW показали значну ефективність при кластеризації часових рядів. Але, попри це, при великій кількості даних він може призвести до збільшення навантаження. Тому в [33] розглядається можливість розробити ітеративну процедуру DTW, що здатна скоротити часові послідовності.

2.3 Алгоритм DBSCAN

DBSCAN – це алгоритм, що засновано на щільності. Кластеризація за ним відбувається таким чином, що до однієї групи потрапляють точки, які розташовані близько одна до одної. Точки ж, які стоять досить далеко від найближчого сусіда вважаються викидами. Цей алгоритм досить часто використовується для кластеризації даних, в якості альтернативи алгоритму k-means.

При роботі з алгоритмом DBSCAN [34] використовуються такі поняття:

Околиця:

$$E(x_j) = \{x_i \in X_Q | d(x_i, x_j) \leq \sigma\}. \quad (2.11)$$

Кореневим об'єктом, називається об'єкт околиця якого містить не менше m (ступінь) об'єктів.

$$x_j \in X_Q | \text{card}E(x_j) \geq m. \quad (2.12)$$

Об'єкт x_i є безпосередньо щільно-досяжний з об'єкта x_j , якщо x_j - це кореневий об'єкт, а x_i знаходиться на відстані не більшій ніж σ від точки x_j .

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

$$\{x_p \in X_Q \mid p = \overline{1, k}, \quad x_1 \equiv x_j \quad x_2 \equiv x_i\}; \quad (2.13)$$

$$\forall p = \overline{1, k-1}: \{x_{p+1} \in E(x_p) \mid \text{card}E(x_p) \geq m\}. \quad (2.14)$$

Шумом називається множина точок, які не належать до жодного кластеру.

Розглядаючи алгоритм слід зазначити, що точка може бути в одному з трьох станів:

- не переглянута;
- помічена як та, що не є точкою жодного з кластерів;
- та, що належить деякому кластеру.

Алгоритм DBSCAN складається за таких етапів:

Крок 1: Для кожної околиці σ кожної точки $x_i \in X_Q$ знайти точки та кореневі об'єкти x_j , які мають більше ніж m сусідів.

Крок 2: Для ядрових точок (кореневих об'єктів) сусідніх графів визначити компоненти зв'язності.

Крок 3: Для кожної не ядрової точки знайти кластер, який знаходиться в околиці σ . Якщо такий кластер знайдено помістити точку в нього, якщо ж ні тоді задана точка є шумом.

Алгоритм завершується тоді, коли всі точки переглянуті, інакше обирається наступна точка, що ще не була переглянута.

На рисунку 2.1 [35] наведено приклад, за яким видно формування кластерів при застосуванні методів, що використовують щільність

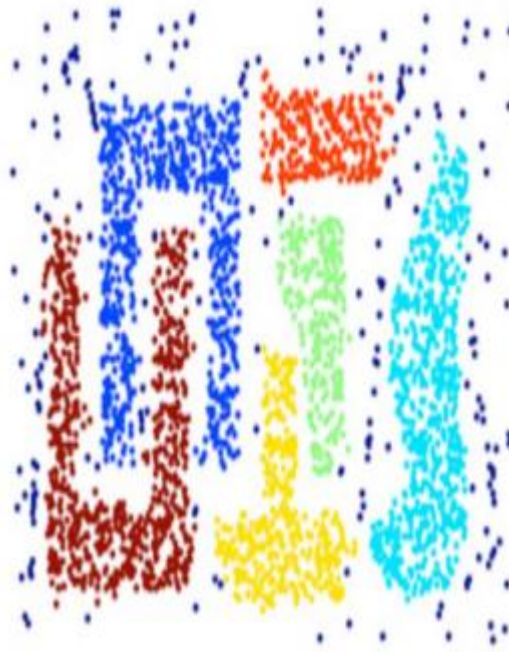


Рисунок 2.2 - Приклад кластеризації на основі щільності

До переваг алгоритму слід віднести те, що немає необхідності на початку визначати кількість кластерів, на відміну від *k-means*, де це значення є параметром функції. Це досить суттєво, як все було сказано раніше, необхідність на початку визначення кількості кластерів ускладнює процес кластеризації, робить його менш зручним, адже природнім було б те, що б кількість кластерів визначалася у залежності від насправді існуючих груп за їх відмінностями.

Однак, DBSCAN вимагає такі параметри, як максимальну відстань $\sigma = d_{max}(x_i, x_j)$ між сусідніми точками (околиця об'єкта), і мінімальне число точок в околиці - m .

2.4 Робота з пропусками з метою подальшого прогнозування

Відновлення пустих значень вирішує задачу заповнення пропусків в таблицях і часових рядах такими значеннями, що найбільш точно підходять і мають мінімальну похибку.

В роботі [36] розглядається важливість і складність нечіткої кластеризації в реальних умовах, які передбачають вміст пропущених даних, коли похідна таблиця має пусті клітинки, задача може ускладнюватися ще і тим, що кількість цих значень невідома і хаотична. Таким чином, вирішення цієї проблеми є актуальним і продовжує розглядатися, попри наявність напрацювань у даній сфері.

Нечітка кластеризація досить часто використовується для вирішення проблем пропусків, шляхом адаптування класичного алгоритму, але існують і інші методи, що дозволяють розв'язати цю задачу. Так, наприклад, в роботі [37] розглядається нейронна мережа, яка здатна впоратися з відсутніми значеннями в режимі реального часу з високою швидкістю, при цьому маючи просту реалізацію.

Підходи на основі математичного апарату обчислювального інтелекту, а, насамперед, нейромережах [38], для вирішення такої задачі, вважаються досить ефективними.

Перспективною є сфера вивчення методів кластеризації з метою прогнозування. Алгоритм полягає в тому, що необхідно на перших кроках відновити, одним з можливих методів, пропущені дані, після чого зробити прогноз (передбачення часових рядів, на основі наявної ділянки, з відновленими заздалегідь даними). Далі провести кластеризацію розширеної таким чином області даних.

Вирішення такої задачі може застосовуватися і убути необхідним у багатьох сферах, таких як наука, економіка, соціальна сфера, виробництво та таке інше. З розвитком ІТ, а також збільшенням кількості даних, що зберігаються, зростає популярність і розвиток методів прогнозування. Вони потребують досить детального вивчення похідних наборів даних, а також, безпосередньо методів, що будуть використовуватися для аналізу.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННІ МЕТОДІВ ПРОГНОЗУВАННЯ І КЛАСТЕРИЗАЦІЇ ЧАСОВИХ РЯДІВ

3.1 Розгляд предметної області

В роботі розглядатиметься таблиця, що містить медичну інформацію. Ця предметна область була обрана на сам перед тому, що на сьогоднішній день, обробка саме медичної інформації є доволі актуальною та важливою задачею. В контексті кластеризації даних з пропусками така ПО є досить доречною, адже в ній є велика ймовірність пустих, незаповнених полів, відсутність певної кількості важливої інформації. Та попри це обробка такого роду даних дуже важлива і необхідна задача, а її достовірність і точність – це беззаперечна необхідність.

Отже, можна зазначити, що ймовірність втрати певної частини інформації в таких ситуаціях не виключена, що зумовлено значним поспіхом, заповненням анкети самим пацієнтом, який не надав значущості якимось питанням та багато інших причин. При цьому, попри дані фактори, необхідно проаналізувати отриману інформації якомога точніше і отримати результат максимально приближений до еталонного, того, що можна отримати при повних даних.

Які цілі можуть переслідуватися в кластеризації медичної інформації?
Насамперед це:

- групування пацієнтів за захворюваннями, пошук спільного та відмінного в межах групи, що дозволить покращити якість лікування;
- оцінка критеріїв, що впливають на ту чи іншу групу, здатна покращити розуміння причин захворювань;
- можливість оцінити шкідливі звички, кому вони притаманні, як впливають на здоров'я;
- регіональні відмінності, або відмінності відносно статі, соціальних груп та іншого.

Кластеризація використовується в безлічі різних областей діяльності людини. У медицині кластеризація симптомів і захворювань помітно полегшує аналіз і вид лікування. Вірний розгляд кластерів симптомів шизофренії, параної і інших захворювань є визначальним при веденні успішної терапії [39].

В роботі [40] розглядається аналіз і обробка мікроскопічних зображень крові, що є складною але важливою задачею. Було використано метод сегментації кольорів.

Всі ці дослідження та висновки можуть допомогти визначити найбільш ефективні методи лікування та допоможуть лікарю в його подальшій роботі. Також, в майбутньому, такий аналіз може допомогти у розвитку штучного інтелекту і його застосуванні в медицині, адже формалізація даних дозволяє вирішувати складні питання без навчання і отримання знань від експертів (що є непростою задачею). А організація пацієнтів в групи дає можливість, як лікарю, так і, надалі, штучному інтелекту працювати за даними швидше і ефективніше.

3.2 Практичні результати застосування простих методів обробки даних з пропусками

3.2.1 Опис даних, що використано для проведення аналізу методів

Перевірка методів кластеризації, а саме можливості отримання допустимих результатів кластеризації в умовах неповних даних проводитиметься на прикладі таблиці з інформацією про хвороби серця пацієнтів.

Ця таблиця містить в собі 303 рядки та 8 стовпців, перші 15 рядків наведено на рисунку 3.1.

| 1 | age | sex | cp | trestbps | fbs | restecg | thalach | exang |
|----|-----|-----|----|----------|-----|---------|---------|-------|
| 2 | 63 | 1 | 3 | 145 | 1 | 0 | 150 | 0 |
| 3 | 37 | 1 | 2 | 130 | 0 | 1 | 187 | 0 |
| 4 | 41 | 0 | 1 | 130 | 0 | 0 | 172 | 0 |
| 5 | 56 | 1 | 1 | 120 | 0 | 1 | 178 | 0 |
| 6 | 57 | 0 | 0 | 120 | 0 | 1 | 163 | 1 |
| 7 | 57 | 1 | 0 | 140 | 0 | 1 | 148 | 0 |
| 8 | 56 | 0 | 1 | 140 | 0 | 0 | 153 | 0 |
| 9 | 44 | 1 | 1 | 120 | 0 | 1 | 173 | 0 |
| 10 | 52 | 1 | 2 | 172 | 1 | 1 | 162 | 0 |
| 11 | 57 | 1 | 2 | 150 | 0 | 1 | 174 | 0 |
| 12 | 54 | 1 | 0 | 140 | 0 | 1 | 160 | 0 |
| 13 | 48 | 0 | 2 | 130 | 0 | 1 | 139 | 0 |
| 14 | 49 | 1 | 1 | 130 | 0 | 1 | 171 | 0 |
| 15 | 64 | 1 | 3 | 110 | 0 | 0 | 144 | 1 |

Рисунок 3.1 – Приклад таблиці heart

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

Таблиця 3.1 – Похідні дані для дослідження

| № | Назва стовпчика | Опис |
|---|-----------------|--|
| 1 | age | Вік, в роках |
| 2 | sex | Стать (1 = чоловік; 0 = жінка) |
| 3 | cp | Тип болю в грудях |
| 4 | trestbps | Артеріальний тиск у стані спокою (в мм рт.ст. при надходженні в госпіталь) |
| 5 | fbs | Рівень цукру в крові натще > 120 мг / дл (1 = істина; 0 = брехня) |
| 6 | restecg | Результати електрокардіографії в спокої |
| 7 | thalach | Досягнуто максимальний пульс |
| 8 | exang | Розвинути стенокардію, викликану фізичним навантаженням (1 = так; 0 = ні) |

3.2.2 Результати застосування методів

Для початку проведемо кластеризацію похідної таблиці, яка не містить пропущених даних. Для кластеризації обрано метод – fuzzy k-means і мову програмування R.

Виконаємо побудову "нечітких" кластерів з використанням функції `fanny()` з пакету `cluster`:

```
res.fanny <- fanny(data, k = 3, memb.exp = 1.3,
                  metric = "euclidean", stand = TRUE, maxit = 500)
print(res.fanny$membership,3)
```

- обрано 3 кластери ($k = 3$);
- `memb.exp` обрано таким чином, що `memb.exp -> 1` дає більш чітку кластеризацію, тоді як `memb.exp -> Inf` призводить до повної нечіткості;
- `metric` - рядок символів, який вказує метрику, яка буде використовуватися для розрахунку відмінностей між спостереженнями. Варіанти «euclidean», «manhattan» і «SqEuclidean». Евклідові відстані - це корінь суми квадратів різниць, манхеттенські відстані – сума абсолютних різниць, а «SqEuclidean», квадрат евклідових відстаней – сума квадратів різниць;
- `stand` – параметр логічного типу, якщо `true`, вимірювання в x стандартизуються перед обчисленням відмінностей. Вимірювання стандартизовані для кожної змінної (стовпчик) шляхом вирахування середнього значення змінної і ділення на середнє абсолютне відхилення змінної;
- `maxit` – максимальна кількість ітерацій, в нашому випадку – 500;
- перші, середні та останні 10 рядків, отримані за результатами кластеризації, наведено на рисунку 3.1.

```

      [,1]  [,2]  [,3]
[1,] 0.503 0.2476 0.2492
[2,] 0.203 0.7080 0.0889
[3,] 0.322 0.5473 0.1304
[4,] 0.239 0.6528 0.1079
[5,] 0.323 0.2860 0.3913
[6,] 0.343 0.3160 0.3407
[7,] 0.526 0.2766 0.1974
[8,] 0.138 0.7956 0.0665
[9,] 0.466 0.2874 0.2469
[10,] 0.403 0.4520 0.1449
...
[145,] 0.409 0.2480 0.3427
[146,] 0.449 0.1752 0.3762
[147,] 0.278 0.6023 0.1195
[148,] 0.479 0.3619 0.1593
[149,] 0.154 0.7817 0.0641
[150,] 0.205 0.6947 0.1006
[151,] 0.389 0.1626 0.4487
[152,] 0.366 0.2239 0.4102
[153,] 0.483 0.2632 0.2538
[154,] 0.556 0.2238 0.2203
...
[294,] 0.502 0.2211 0.2764
[295,] 0.210 0.3137 0.4762
[296,] 0.189 0.0882 0.7227
[297,] 0.280 0.1620 0.5580
[298,] 0.385 0.1915 0.4240
[299,] 0.265 0.1431 0.5919
[300,] 0.285 0.5300 0.1853
[301,] 0.439 0.2076 0.3530
[302,] 0.142 0.0932 0.7650
[303,] 0.468 0.3800 0.1517

```

Рисунок 3.2 – Результат застосування функції fanny

В результаті виводиться матриця коефіцієнтів приналежності, максимальний з яких визначає цільовий кластер.

Для оцінки міри нечіткості, отриманої класифікації, використовується коефіцієнт поділу Dunn (3.1):

$$F_k = \sum_{i=1}^n \sum_{r=1}^k \frac{\mu_{ir}^2}{k}. \quad (3.1)$$

Цей коефіцієнт приймає значення 1 в разі чіткої кластеризації:

```

dunn_coeff      normalized
0.4203087      0.1304630

```

В даному випадку $F_k = 0.42$, а його нормована версія, що змінюється від 0 до 1 – 0.13.

Побудуємо діаграму. На рисунку 3.2 наведено ординаційну діаграму з результатами кластеризації.

```
fviz_cluster(res.fanny, frame.type = "norm", frame.level = 0.7)
```

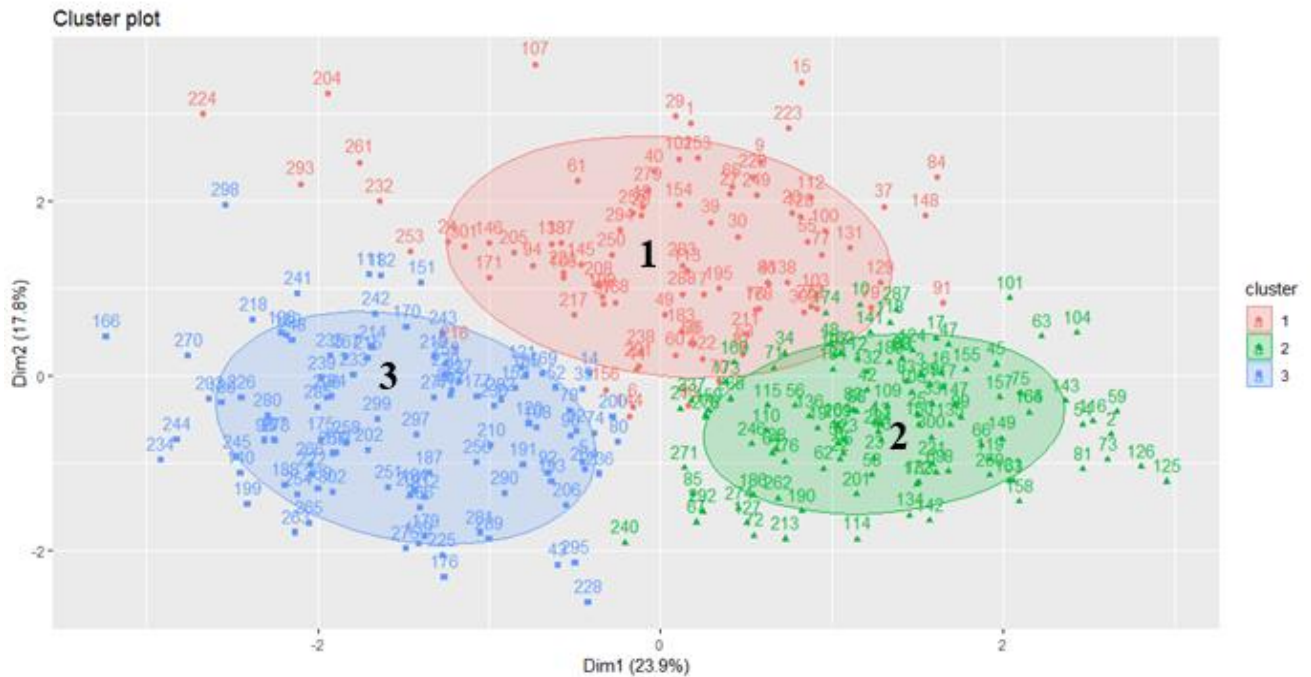


Рисунок 3.3 – Графік результатів нечіткої кластеризації

Для подальшого аналізу внесемо у похідні дані 30% пропусків випадковим чином.

Кластеризація з видаленням пропусків:

```
heartMissedNO <- subset(heartMissed, exang != "")
```

Перші, середні та останні 10 рядків, отримані за результатами кластеризації, наведено на рисунку 3.3

| | [,1] | [,2] | [,3] |
|-----|--------|--------|--------|
| 1 | 0.4942 | 0.2168 | 0.2890 |
| 2 | 0.1298 | 0.7809 | 0.0893 |
| 3 | 0.2238 | 0.6263 | 0.1499 |
| 7 | 0.4347 | 0.2923 | 0.2730 |
| 10 | 0.3306 | 0.4800 | 0.1894 |
| 14 | 0.3695 | 0.2501 | 0.3804 |
| 15 | 0.4660 | 0.2665 | 0.2675 |
| 16 | 0.1732 | 0.7063 | 0.1205 |
| 17 | 0.2529 | 0.5813 | 0.1659 |
| 18 | 0.4144 | 0.2278 | 0.3578 |
| ... | | | |
| 160 | 0.3868 | 0.3782 | 0.2349 |
| 162 | 0.2761 | 0.5348 | 0.1892 |
| 164 | 0.1312 | 0.7797 | 0.0891 |
| 171 | 0.4473 | 0.1848 | 0.3678 |
| 173 | 0.3781 | 0.3545 | 0.2674 |
| 177 | 0.3981 | 0.1947 | 0.4072 |
| 178 | 0.4136 | 0.3201 | 0.2663 |
| 183 | 0.4156 | 0.2709 | 0.3135 |
| 184 | 0.3178 | 0.4751 | 0.2072 |
| 185 | 0.3908 | 0.1834 | 0.4258 |
| ... | | | |
| 283 | 0.4502 | 0.2238 | 0.3260 |
| 288 | 0.4696 | 0.2597 | 0.2707 |
| 289 | 0.2670 | 0.1818 | 0.5512 |
| 293 | 0.4406 | 0.1686 | 0.3908 |
| 294 | 0.5121 | 0.1760 | 0.3119 |
| 297 | 0.3096 | 0.1419 | 0.5485 |
| 299 | 0.3098 | 0.1351 | 0.5551 |
| 300 | 0.2257 | 0.5778 | 0.1965 |
| 301 | 0.4582 | 0.1537 | 0.3882 |
| 302 | 0.2466 | 0.0995 | 0.6539 |

Рисунок 3.4 – Результат застосування функції fanny при видалених 30% даних

Коефіцієнт Dunn приймає значення:

| | |
|------------|------------|
| dunn_coeff | normalized |
| 0.4153731 | 0.1230597 |

На рисунку 3.4 наведено ординаційну діаграму з результатами кластеризації.

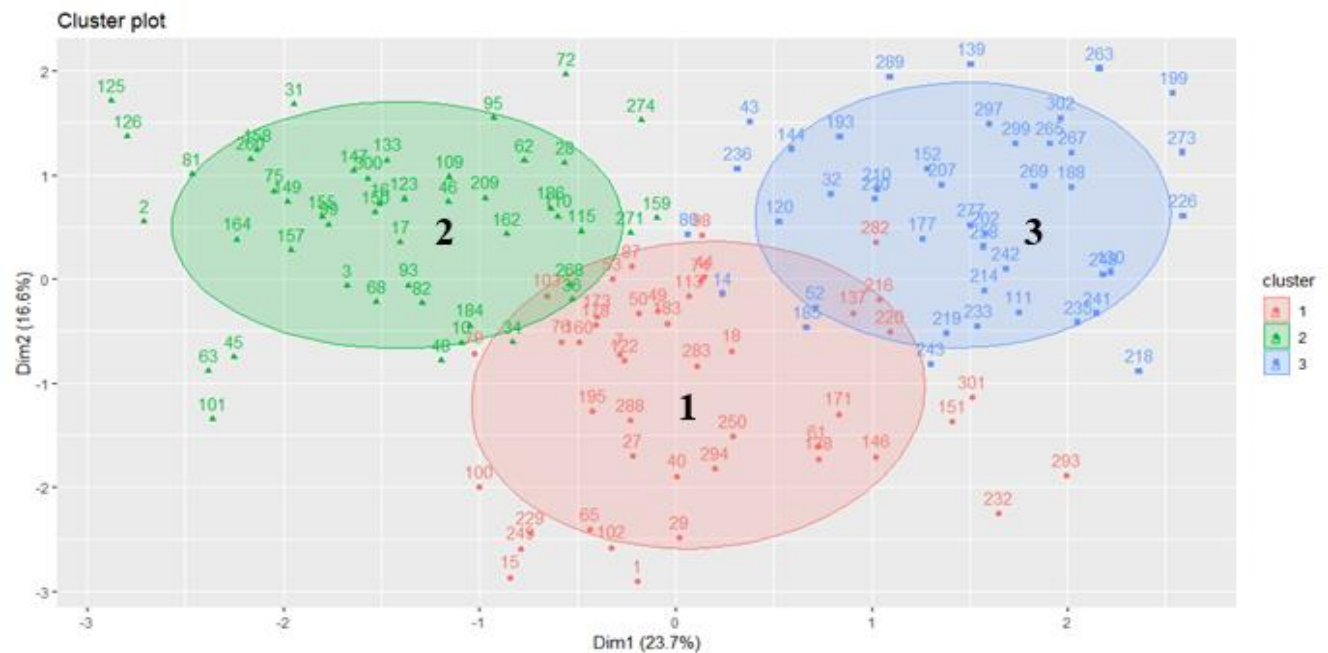


Рисунок 3.5 – Графік результатів нечіткої кластеризації при видалених 30% даних

Кластеризація із заповненням пропусків вибілковими статистиками:

Отже, введемо припущення, що взаємозв'язку між змінними, в даному випадку немає, тоді ефективним способом заповнення пропусків буде використання середніх значень, для цього оберемо медіану.

```
heartMissedProcess<-heartMissed
```

```
ind <- apply(heartMissedProcess, 1, function(x) sum(is.na(x))) > 0
```

```
heartMissedProcess[ind, 1:8]
```

```
pPmI <- preProcess(heartMissedProcess[, 1:8], method = 'medianImpute')
```

```
heartMissedProcess[, 1:8] <- predict(pPmI, heartMissedProcess[, 1:8])
```

```
(Imp.Med <- heartMissedProcess[ind, 1:8])
```

Перші, середні та останні 10 рядків, отримані за результатами кластеризації, наведено на рисунку 3.5

| | [,1] | [,2] | [,3] |
|--------|-------|-------|--------|
| [1,] | 0.413 | 0.263 | 0.3237 |
| [2,] | 0.206 | 0.660 | 0.1336 |
| [3,] | 0.303 | 0.501 | 0.1959 |
| [4,] | 0.253 | 0.593 | 0.1542 |
| [5,] | 0.342 | 0.270 | 0.3887 |
| [6,] | 0.354 | 0.276 | 0.3703 |
| [7,] | 0.433 | 0.281 | 0.2853 |
| [8,] | 0.158 | 0.739 | 0.1034 |
| [9,] | 0.204 | 0.682 | 0.1133 |
| [10,] | 0.337 | 0.459 | 0.2040 |
| ... | | | |
| [145,] | 0.362 | 0.304 | 0.3339 |
| [146,] | 0.414 | 0.181 | 0.4048 |
| [147,] | 0.269 | 0.555 | 0.1759 |
| [148,] | 0.358 | 0.416 | 0.2264 |
| [149,] | 0.159 | 0.744 | 0.0965 |
| [150,] | 0.205 | 0.657 | 0.1380 |
| [151,] | 0.389 | 0.168 | 0.4431 |
| [152,] | 0.350 | 0.273 | 0.3776 |
| [153,] | 0.396 | 0.341 | 0.2635 |
| [154,] | 0.411 | 0.314 | 0.2751 |
| ... | | | |
| [294,] | 0.432 | 0.231 | 0.3372 |
| [295,] | 0.282 | 0.304 | 0.4141 |
| [296,] | 0.309 | 0.122 | 0.5691 |
| [297,] | 0.334 | 0.173 | 0.4935 |
| [298,] | 0.370 | 0.212 | 0.4187 |
| [299,] | 0.327 | 0.160 | 0.5131 |
| [300,] | 0.285 | 0.489 | 0.2261 |
| [301,] | 0.394 | 0.209 | 0.3967 |
| [302,] | 0.265 | 0.132 | 0.6034 |
| [303,] | 0.423 | 0.300 | 0.2771 |

Рисунок 3.6 – Результат застосування функції fanny при заміні 30% значень на медіани

Коефіцієнт Dunn приймає значення:

| | |
|------------|------------|
| dunn_coeff | normalized |
| 0.38203172 | 0.07304758 |

На рисунку 3.6 наведено ординаційну діаграму з результатами кластеризації.



Рисунок 3.7 – Графік результатів нечіткої кластеризації при заміні 30% значень на медіани

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.

Приклад для заповнення поля age:

```
lm(age ~ trestbps, data = heartMissKor)
ageTres <- function(oP) {if (is.na(oP)) return(NA)
  else return(34.7253 + 0.1508 * oP)
}
heartMissKor[is.na(heartMissKor$age), 'age'] <-
  sapply(heartMissKor[is.na(heartMissKor$age), 'trestbps'], ageTres)
heartMissKor[ind, 10]
```

Перші, середні та останні 10 рядків, отримані за результатами кластеризації, наведено на рисунку 3.7.

| | [,1] | [,2] | [,3] |
|-----|-------|--------|--------|
| 1 | 0.472 | 0.2742 | 0.2542 |
| 2 | 0.190 | 0.7050 | 0.1051 |
| 3 | 0.292 | 0.5531 | 0.1552 |
| 4 | 0.392 | 0.3199 | 0.2886 |
| 5 | 0.346 | 0.2427 | 0.4117 |
| 6 | 0.337 | 0.2566 | 0.4063 |
| 7 | 0.429 | 0.3461 | 0.2249 |
| 8 | 0.159 | 0.7487 | 0.0926 |
| 9 | 0.401 | 0.3250 | 0.2744 |
| 10 | 0.236 | 0.6512 | 0.1132 |
| ... | | | |
| 145 | 0.386 | 0.3268 | 0.2872 |
| 146 | 0.394 | 0.1980 | 0.4083 |
| 147 | 0.273 | 0.5945 | 0.1325 |
| 148 | 0.315 | 0.5478 | 0.1371 |
| 149 | 0.143 | 0.7837 | 0.0728 |
| 150 | 0.194 | 0.6897 | 0.1162 |
| 151 | 0.330 | 0.1575 | 0.5122 |
| 152 | 0.359 | 0.2779 | 0.3628 |
| 153 | 0.411 | 0.2994 | 0.2894 |
| 154 | 0.414 | 0.2784 | 0.3075 |
| ... | | | |
| 294 | 0.424 | 0.2859 | 0.2899 |
| 295 | 0.245 | 0.2601 | 0.4951 |
| 296 | 0.218 | 0.0900 | 0.6924 |
| 297 | 0.301 | 0.1528 | 0.5466 |
| 298 | 0.399 | 0.1970 | 0.4038 |
| 299 | 0.284 | 0.1524 | 0.5633 |
| 300 | 0.298 | 0.5108 | 0.1914 |
| 301 | 0.438 | 0.2158 | 0.3461 |
| 302 | 0.190 | 0.1035 | 0.7063 |
| 303 | 0.424 | 0.2601 | 0.3158 |

Рисунок 3.8 – Результат застосування функції `fanny` при заміні 30% значень з урахуванням структури зв'язків

Коефіцієнт Dunn приймає значення:

| | |
|-------------------------|-------------------------|
| <code>dunn_coeff</code> | <code>normalized</code> |
| 0.4116371 | 0.1174556 |

На рисунку 3.8 наведено ординаційну діаграму з результатами кластеризації при заміні 30% значень з урахуванням структури зв'язків.

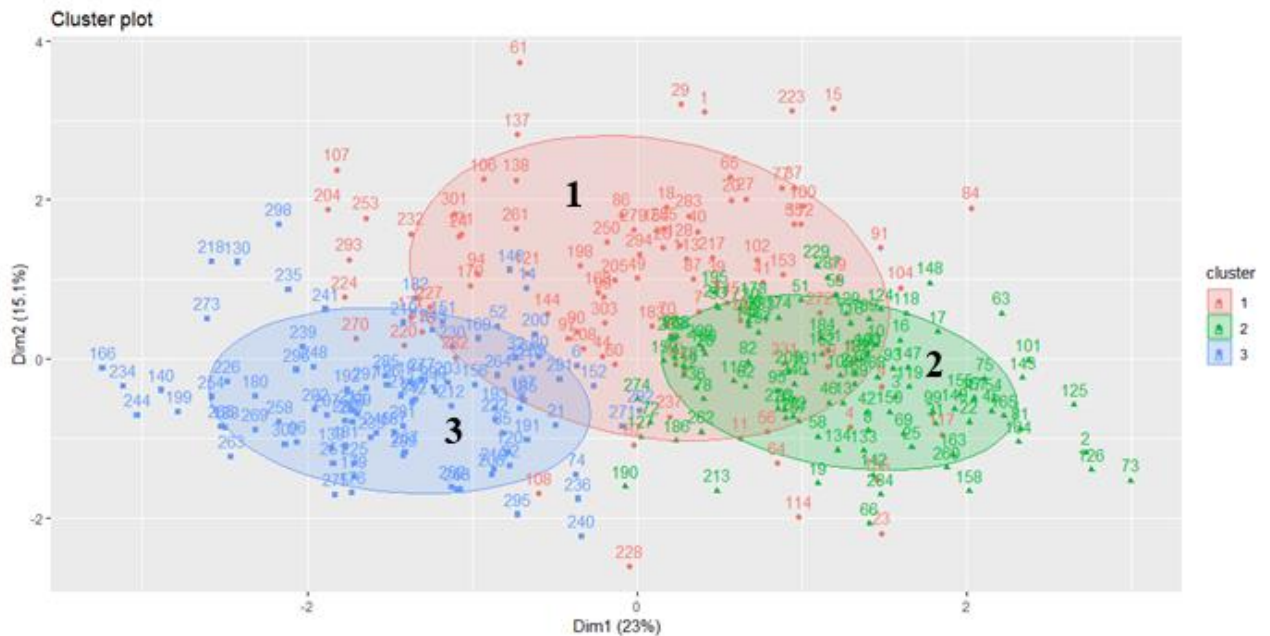


Рисунок 3.9 – Графік результатів нечіткої кластеризації при заміні 30% значень з урахуванням структури зв'язків

Для проведення кластеризації і отримання параметрів на кожному етапі використовувалася функція:

```

kmean <- function(data) {
  res.fanny <- fanny(data, k = 3, memb.exp = 1.3,
                    metric = "euclidean", stand = TRUE, maxit = 500)
  print(res.fanny$membership,3)
  res.fanny$coeff
  print( res.fanny$coeff)
  Dunn <- res.fanny$membership^2
  fviz_cluster(res.fanny, ellipse.type = "norm", ellipse.level = 0.7)
}

```

Розглядаючи використану функцію можна зазначити, що цільовою таблицею обрано *data* (у даному випадку *heart.csv*) і, початково обрано 3 кластери. *memb.exp* – число *r*, строго більше 1, що вказує на показник ступеня членства, який використовується в критерії відповідності.

Слід зазначити, що $r \rightarrow 1$ дає більш чітку кластеризацію, тоді як $r \rightarrow Inf$ призводить до повної нечіткості, значення, занадто близькі до 1, можуть привести до повільної конвергенції, навіть значення за замовчуванням $r = 2$ може привести до повної нечіткості, тобто членства $u(i, v) = 1 / k$. В цьому випадку рекомендується вибрати менший `memb.exp (= r)`.

`metric` – рядок символів, який вказує метрику, яка буде використовуватися для розрахунку відмінностей між спостереженнями. Варіанти «euclidean», «manhattan» і «SqEuclidean». Евклідові відстані – це корінь суми квадратів різниць, манхеттенські відстані – сума абсолютних різниць, а «SqEuclidean», квадрат евклідових відстаней – сума квадратів різниць.

`stand` – параметр логічного типу, якщо `true`, вимірювання в x стандартизуються перед обчисленням відмінностей. Вимірювання стандартизовані для кожної змінної (стовпчика) шляхом вирахування середнього значення змінної і ділення на середнє абсолютне відхилення змінної.

`maxit` – максимальна кількість ітерацій.

3.3 Практичні результати прогнозування та кластеризації часових рядів

Для аналізу часових рядів, у якості тестового набору, оберемо дані затраченої електроенергії від електроприладів за період у 4 місяці. Графік цього часового ряду наведено на рисунку 3.10.

Appliances, energy use in Wh

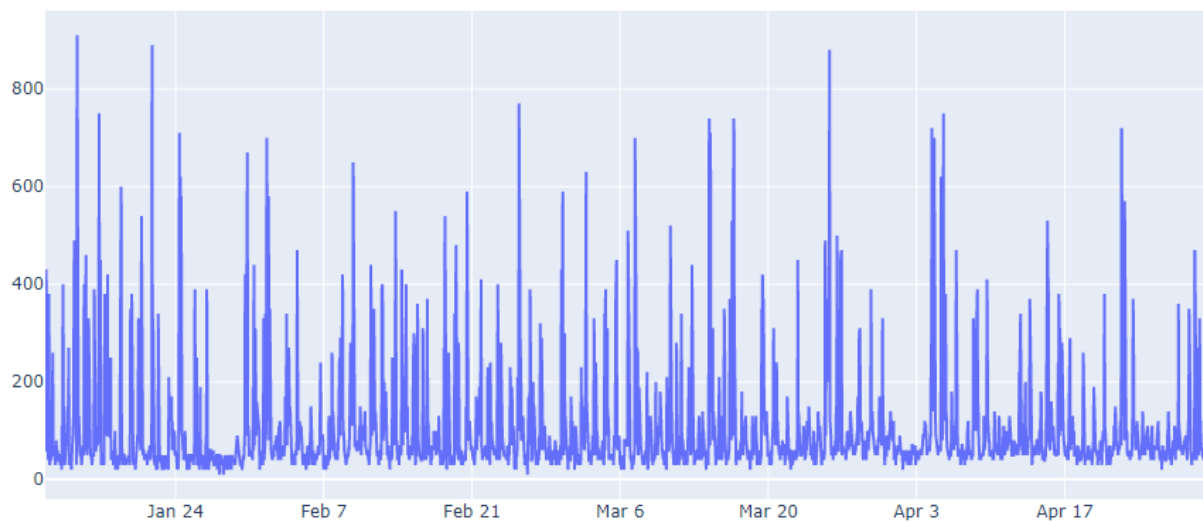


Рисунок 3.10 – Часовий ряд використаної, електронними приладами, енергії

Модель ряду можна представити як:

$$x = t + s + z, \quad (3.2)$$

де t – тренд, s - сезонність, z - похибка.

Модель ряду наведено на рисунку 3.11. З наведеного графіку видно, що випадкова величина z (random) має досить вагомий вплив на часовий ряд, що розглядається.

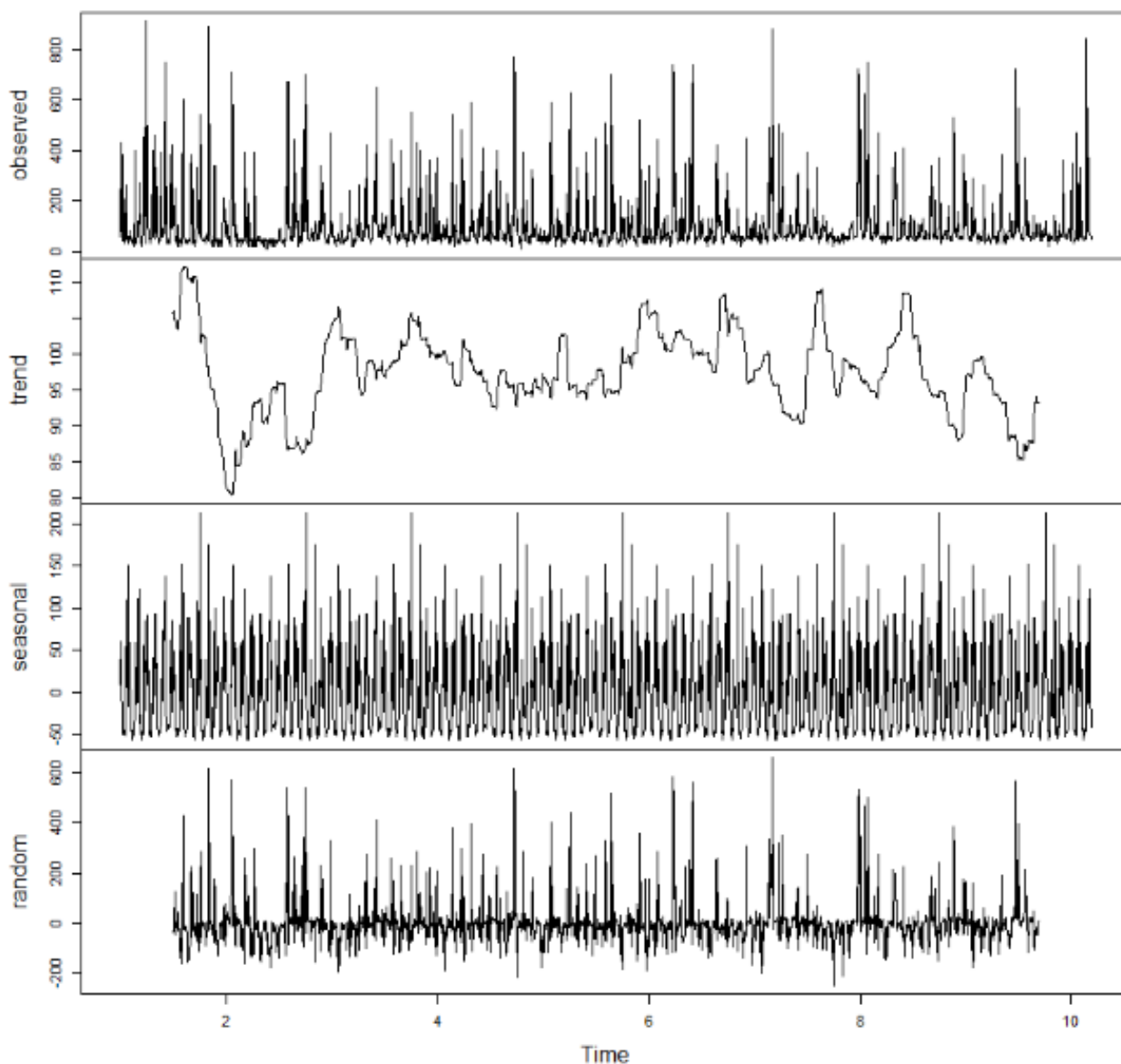


Рисунок 3.11 – Опис часового ряду, що розглядається

3.3.1 Прогнозування ряду та побудова SARIMA моделі

В процесі прогнозування, для того щоб побачити загальні тенденції з урахуванням наявних шумів, проведемо згляджування за днями та тижнями, за допомогою ковзної середньої, що наведено на рисунках 3.12, 3.13.

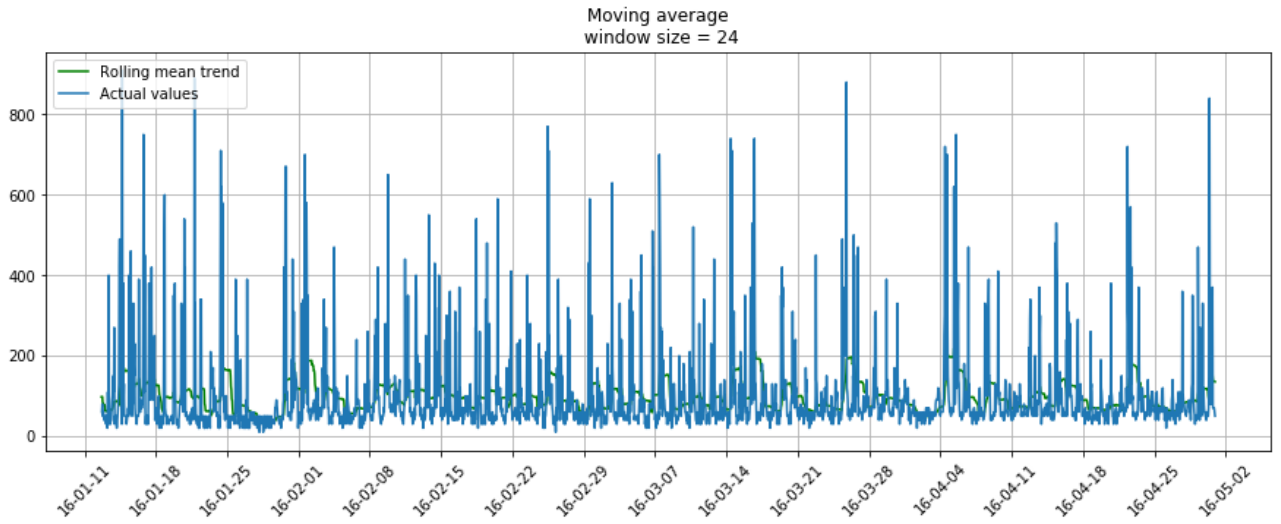


Рисунок 3.12 – Згладжування ряду за днями

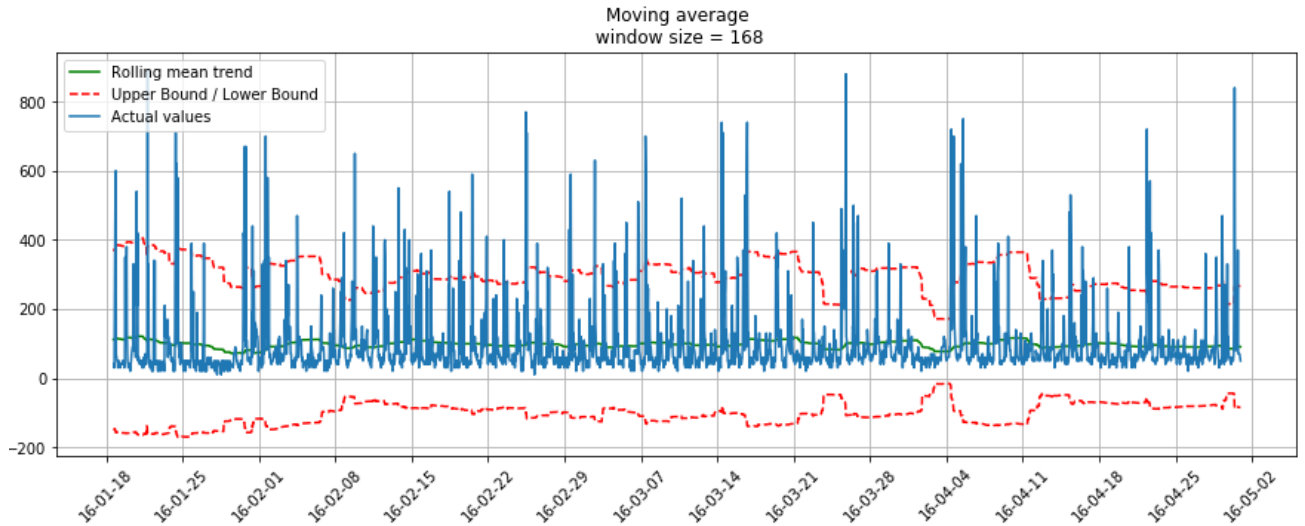


Рисунок 3.13 – Згладжування ряду за тижнями

Дані методи дозволяють прогнозувати на одну точку вперед, але використовуючи подвійне експоненційне згладжування можна побудувати прогноз на дві точки вперед. Такий графік наведено на рисунку 3.14.

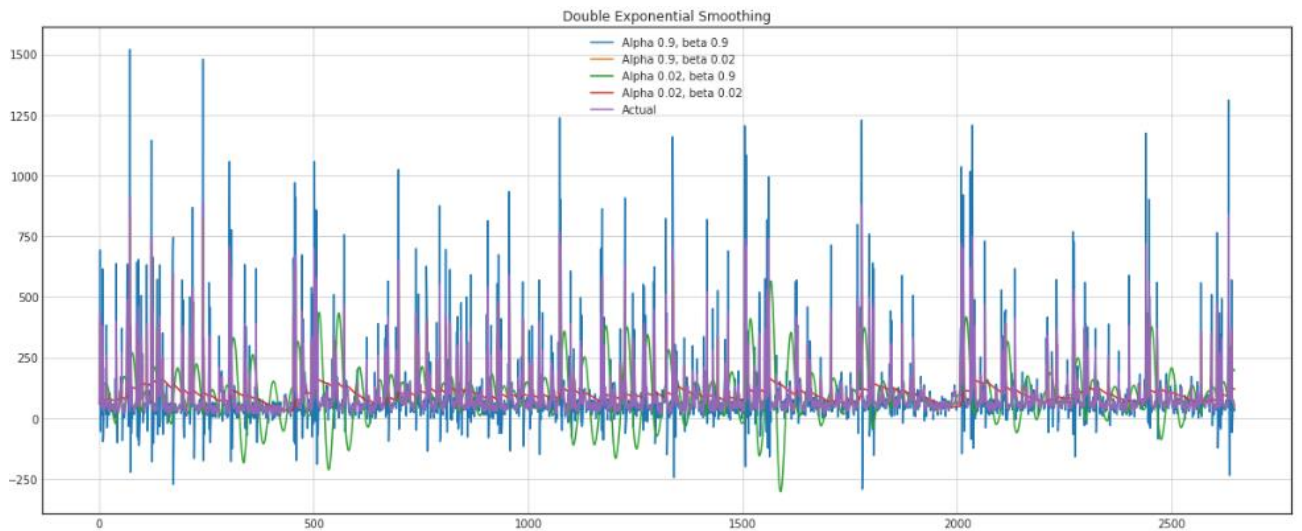


Рисунок 3.14 – Подвійне експоненційне згладжування ряду

Для подальшої обробки з метою отримати прогноз корисним буде використовувати стаціонарний ряд. Стаціонарним вважається ряд, що з часом не змінює своїх статичних значень (постійні математичне сподівання та дисперсія).

За рисунком 3.15 - 3.17 видно, що заданий ряд таким не є.

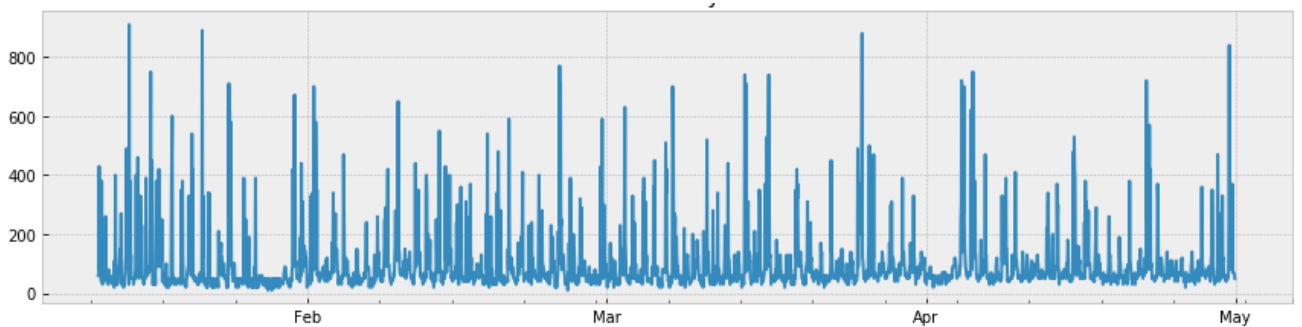


Рисунок 3.15 – Графіки аналізу часового ряду

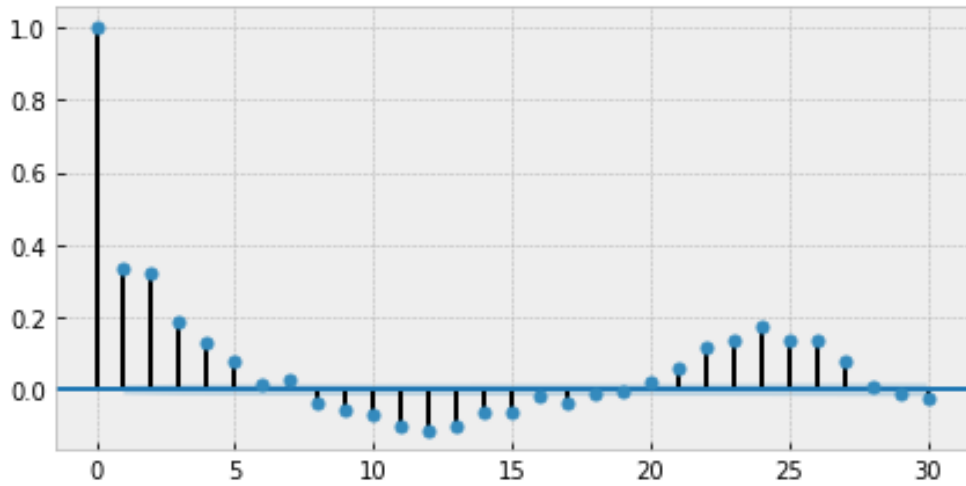


Рисунок 3.16 – Автокореляція часового ряду

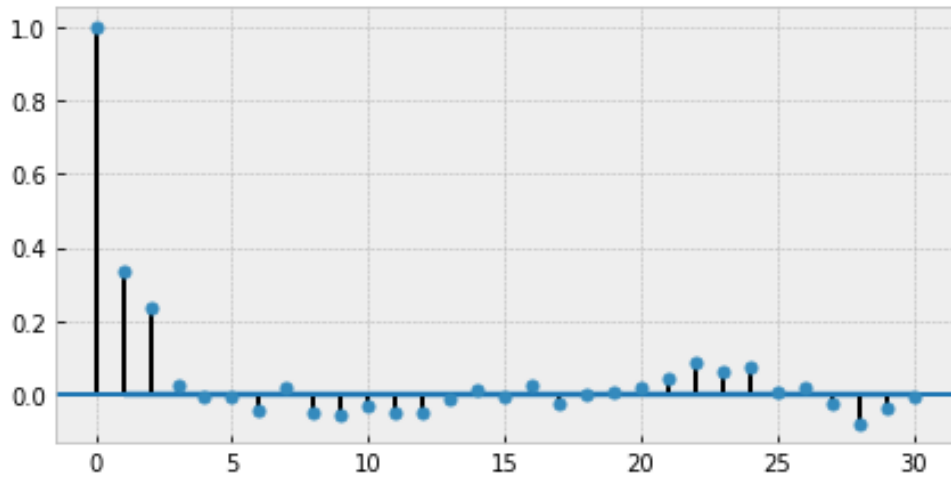


Рисунок 3.17 – Часткова автокореляція часового ряду

Після перетворень отриматемо ряд, що наведено на рисунку 3.18 – 3.20, з яким можна датлі працювати та будувати SARIMA модель.

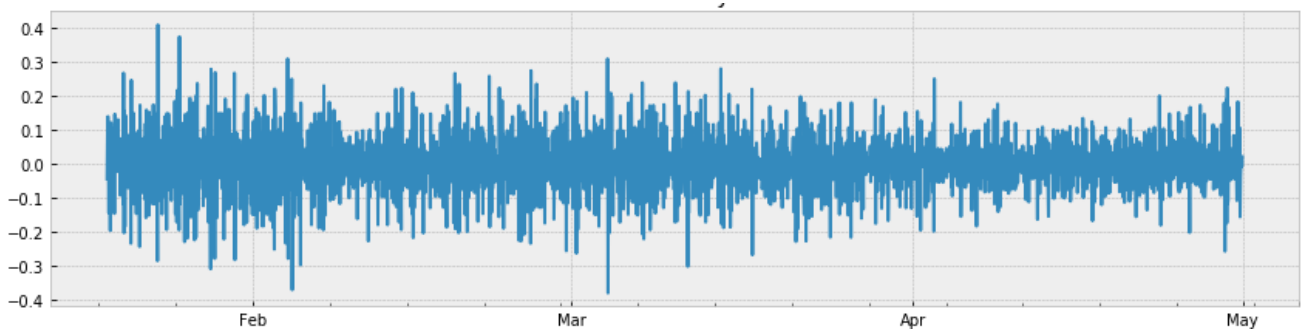


Рисунок 3.18 - Графіки аналізу часового ряду після перетворень

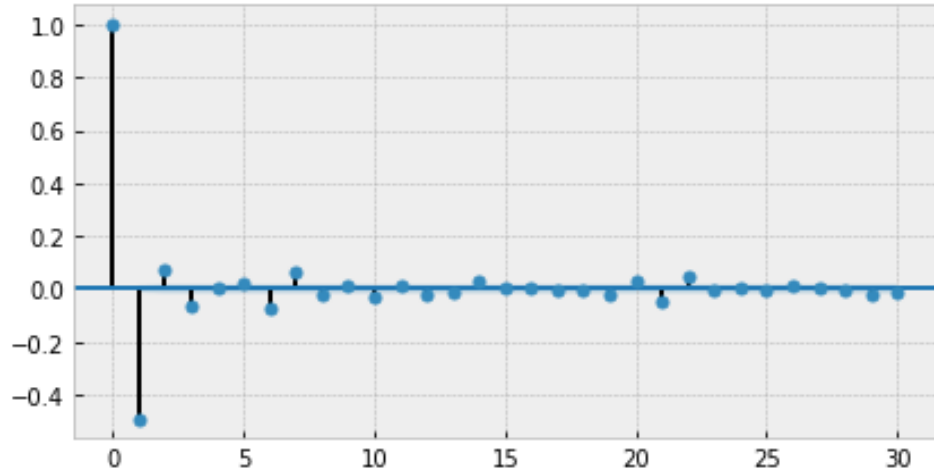


Рисунок 3.19 – Автокореляція часового ряду після перетворень

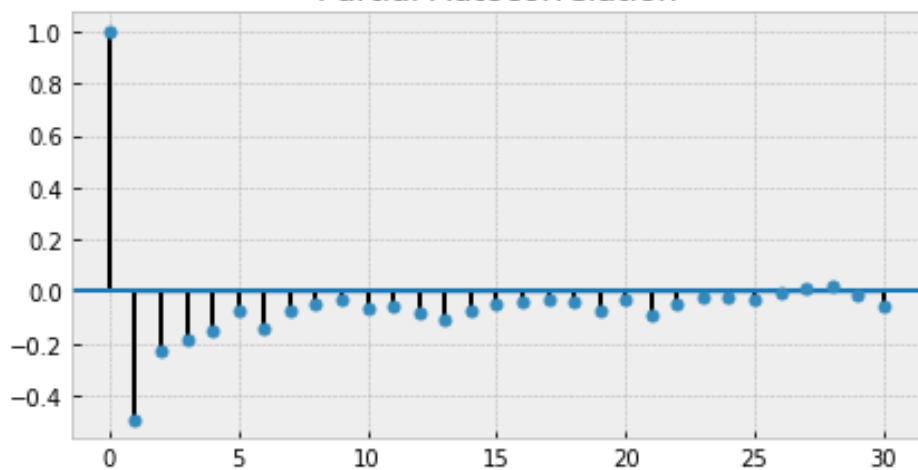


Рисунок 3.20 – Часткова автокореляція часового ряду після перетворень

Побудуємо прогноз по моделі, що вийшла. Модель наведено на рисунку 3.21.

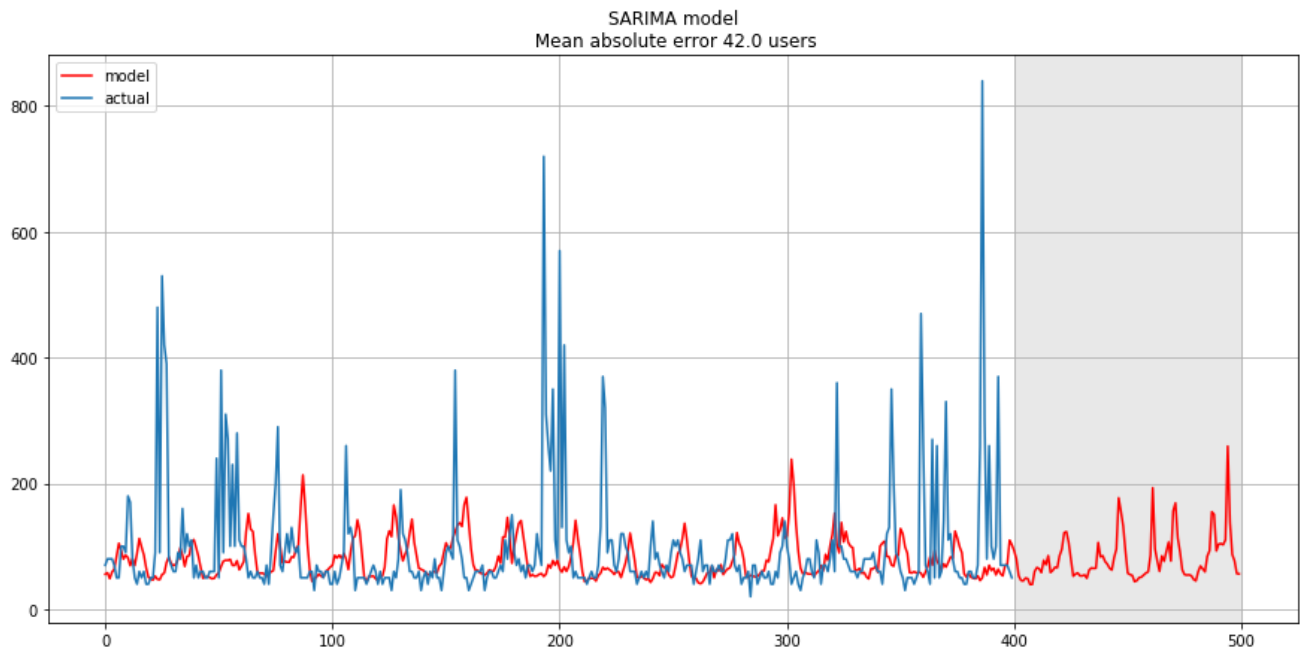


Рисунок 3.21 – Побудована SARIMA модель для похідного ряду

За результатами побудованої моделі можна сказати, що вона вийшла не ідеальною і було отримано не повністю достовірний прогноз, але з урахуванням викидів і шумів можна сказати, що модель не погано відтворила загальну тенденцію.

3.3.2 Кластеризація часового ряду

Наступним кроком проведемо кластеризацію часового ряду. Для цього використаємо: Euclidean k-means, DBA k-means та Soft-DTW k-means кластеризації часових рядів. На рисунках 3.22-3.24 наведено результати такої кластеризації відповідно.

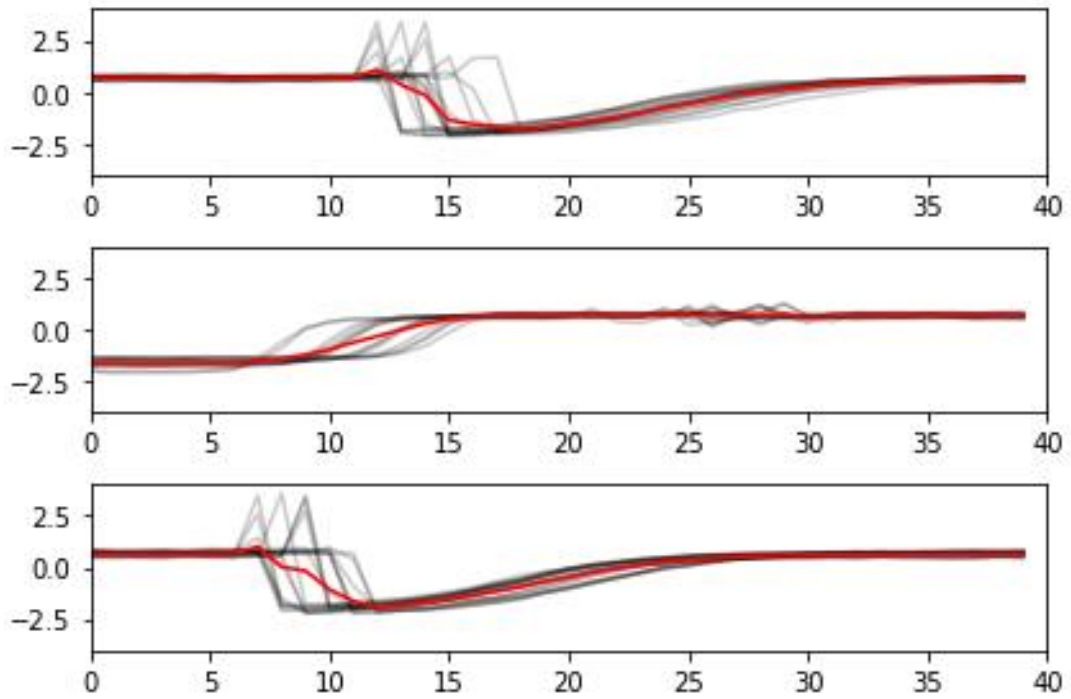


Рисунок 3.22 – Результат кластеризації часового ряду методом Euclidean k-means

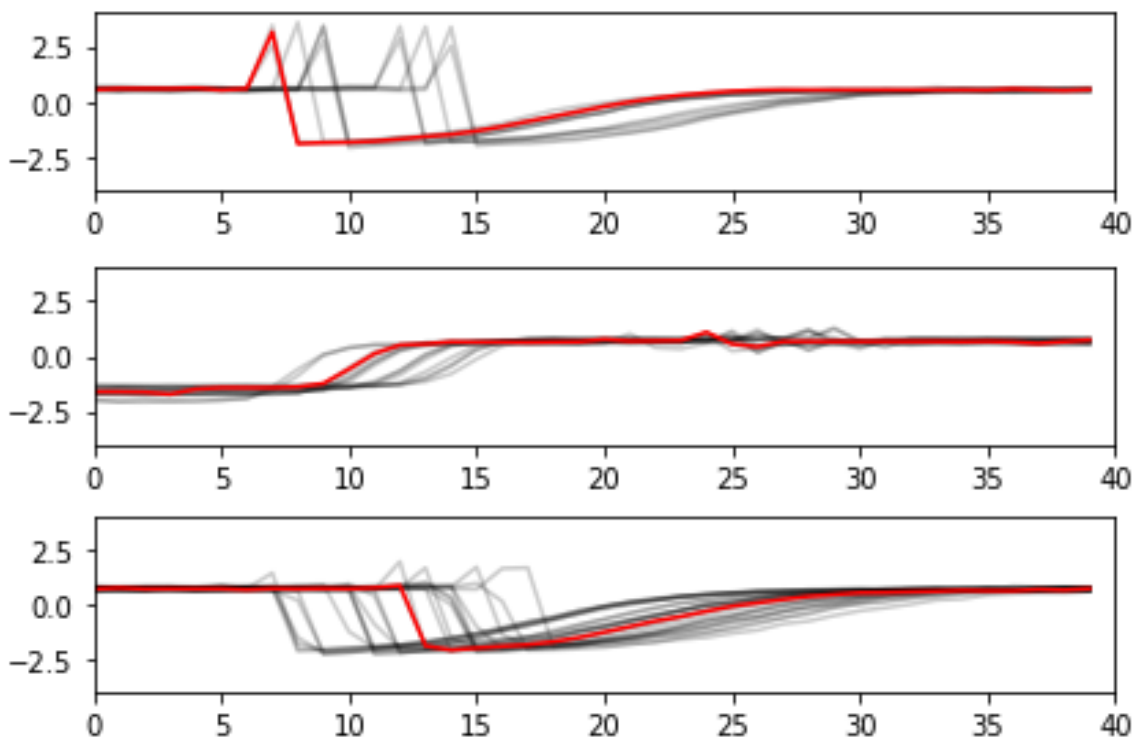


Рисунок 3.23 – Результат кластеризації часового ряду методом DBA k-means

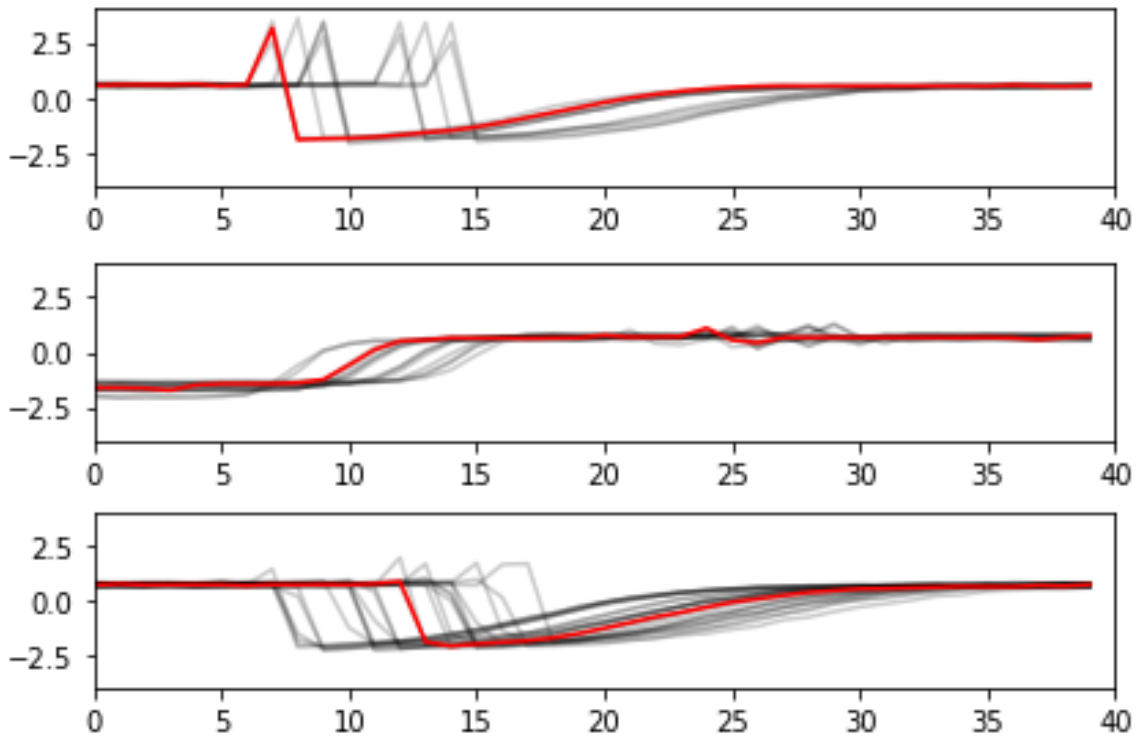


Рисунок 3.24 – Результат кластеризації часового ряду методом Soft-DTW k-means

ВИСНОВКИ

В ході роботи було розглянуто методи кластеризації даних та часових рядів з попереднім прогнозуванням. 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.

Якщо порівнювати таблиці, що є результатами методу fanny, то можна сказати, що обидва методи впоралися добре на даному наборі даних з урахуванням 30% пропусків. Якщо аналізувати графіки, то можна сказати, що є огріхи в кластеризації, відновлення відбулося не ідеально, тому слід враховувати втрату повної достовірності, при виборі одного з таких методів.

У якості обробки часових рядів було проведено аналіз ряду з метою його прогнозування, та отримано SARIMA модель, що наведено на рисунку 3.21. За цією моделлю можна зробити висновок, що вона побудована не бездоганно, але з урахуванням великої кількості викидів та значного впливу випадної величини, що відображено на рисунку 3.11, можна сказати, що отримана модель не погано відобразила загальну тенденцію ряду.

Також, було проведено КА та отримано результати кластеризації на рисунках 3.22, 3.24. Розглянуто декілька методів кластеризації і можна зробити висновок, що на цьому наборі вони дали досить схожі результати.

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., Кобылін О.А. (2019) Методи сегментації зображень. Матеріали ХХІІІ міжнародного молодіжного форуму. *Радіоелектроніка та молодь у ХХІ столітті, 19-20.*
3. Rabortiahov, 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 Radioelectronics, 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). Higr: 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., Рамірез, Х., Горріз, Х. М., Крашений, І. Э., Попов, А. А., ... & Горріз, 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 Radioelectronics, 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-фаззи 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.
39. Пеливан, М. А. (2015). Кластеризация методом наиболее удаленных соседей или методом полной связи. *Техногенной безопасности и устойчивого развития*, 23.
40. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.