Міністерство освіти і науки України

Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук

Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«___» грудня 2021 р

Пояснювальна записка

до кваліфікаційної роботи
магістра

на тему: «**МЕТОД КЛАСТЕРИЗАЦІЇ ДАНИХ НА ОСНОВІ ІНФОРМАЦІЙНИХ КРИТЕРІЇВ ПРИ ОЦІНЮВАННІ РІВНЯ СОЦІАЛЬНО-ЕКОНОМІЧНОГО РОЗВИТКУ КРАЇН У КОНТЕКСТІ ЇХ ЦИФРОВОГО ПРОГРЕСУ**»

Захищено на засіданні
Атестаційної комісії № 38
протокол № __ від __.12.2021 р.
Оцінка _____ / _____
Голова Атестаційної комісії
д.т.н., професор
_____**МІНУХІН**
Сергій Володимирович

Виконав:

студент 6 курсу, групи КІ– 61
Галузь знань: 12 – Інформаційні технології
за спеціальністю 123 – Комп'ютерна інженерія.
Шевченко Дмитро Олександрович ___

Керівник:

д.т.н., професор кафедри теоретичної та прикладної системотехніки
Угрюмов Михайло Леонідович _____

Рецензент: д.т.н., професор кафедри безпеки інформаційних систем і технологій
Кузнецов Олександр Олександрович
_____

Харків – 2021

АНОТАЦІЯ

Кваліфікаційна робота складається зі вступу, восьми розділів, висновків, переліку використаних джерел і п'яти додатків. Загальний обсяг роботи становить 92 сторінки, з яких 62 сторінок основної частини з 34 рисунками та однією таблицею, 5 сторінок списку використаних джерел із 43-х найменувань, 6 додатків на 13 сторінках.

В наш час є актуальною проблема моніторингу (діагностики) стану інформаційних систем. Тому метою роботи було підвищення рівня стійкості до відмов систем шляхом зниження ймовірності некоректного визначення стану систем (помилки третього роду при кластерному аналізі стану систем) за допомогою ефективного методу кластеризації даних на основі інформаційних критеріїв в системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

При виконанні кваліфікаційної роботи були використані методи обробки та підготовки даних, математичні моделі і методи кластеризації та програмні засоби на основі мови Python, такі як scikit-learn, NumPy і інші.

У ході виконання роботи було отримано: огляд методів кластеризації та програмних засобів для рішення задачі кластеризації, методика обробки та підготовки вибірки і новий ефективний метод кластеризації на основі агентно-орієнтованого підходу.

Головним напрямком використання нового методу кластеризації є використання в системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

В майбутньому планується використання методів глибокого навчання для покращення якості методу кластеризації.

Ключові слова: МОНІТОРИНГ (ДІАГНОСТИКА) СТАНУ, МАШИННЕ НАВЧАННЯ, МЕТОД КЛАСТЕРИЗАЦІЇ, МІРИ ВНУТРІШНЬОКЛАСОВОЇ ВІДСТАНІ, АГЕНТНО-ОРІЄНТОВАНИЙ ПІДХІД

ABSTRACT

The qualification work consists of an introduction, eight chapters, conclusions, a list of references and five appendices. The total amount of the work is 92 pages, of which 62 pages of the main part that consists 34 figures and one table, 5 pages of the list of references that consists 43 titles, 6 appendices at 13 pages.

Nowadays, the problem of monitoring (diagnostics) of information systems is relevant. Therefore, the aim was to increase the level of resilience to system failures by reducing the likelihood of incorrect determination of systems (errors of the third kind in cluster analysis of systems) using an effective clustering method based on information criteria in monitoring systems of socio-economic progress in the context of their digital progress.

Methods of preprocessing data, mathematical models and methods for clustering and Python-based software such as scikit-learn, NumPy and others were used in the qualification work.

During the work the following were obtained: an overview of clustering methods and software for solving the clustering problem, a method of preprocessing data and a new effective clustering method based on an agent-oriented approach.

The main direction of using the new clustering method is using in monitoring systems of socio-economic progress in the context of their digital progress.

In the future, it is planned to use deep learning methods to improve the quality of the clustering method.

Keywords: CONDITION MONITORING (DIAGNOSTIC), MACHINE LEARNING, CLUSTERING METHOD, INTRA-CLASS DISTANCE MEASURES, AGENT-ORIENTED APPROACH

ЗМІСТ

ВСТУП

Однією з актуальних технічних проблем під час експлуатації об'єктів управління є проблема моніторингу (діагностики) стану інформаційних систем, що забезпечують ефективну роботу об’єктів управління, наприклад комп’ютерних мереж. Проблеми інформаційного забезпечення процесу моніторингу стану систем можливо вирішати на основі розробки програмного забезпечення для реалізації засобів IT технологій обробки різних типів даних, таких як табличні дані та потоки Big Data.

Однією з задач моніторингу систем являється кластерний аналіз, який реалізується за допомогою методів машинного навчання. Тому тема роботи - розробка методу кластеризації даних в системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу являється актуальною.

З появою нових обчислювальних технологій в теперішній час машинне навчання відрізняється від машинного навчання минулого. Воно започаткувалося з розпізнавання образів і теорії, що комп’ютери мають здатність до навчання, без програмування їх для виконання відповідних завдань. Вчені, які зацікавлені штучним інтелектом, бажали перевірити, чи є можливість навчання комп’ютерів, використовуючи дані. Ітераційний аспект для машинного навчання є досить важливим, так як, коли моделі навчаються на нових даних, вони розвивають здатність самостійно адаптуватися. Моделі навчаються, використовуючи попередні обчислення, для того щоб отримати надійні, повторювані рішення і результати. Це наука, яка не нова, але набула нових обертів.

Незважаючи на те, що багато алгоритмів машинного навчання були розроблені досить давно, можливість автоматично застосовувати складні математичні обчислення з’явилася нещодавно. Приведемо приклади

використання машинного навчання: алгоритми, які використовуються в самокерованих автомобілях; системи рекомендацій Amazon, Netflix; виявлення шахрайства; моніторинг соціально-економічного розвитку країн.

Відновлення інтересу до машинного навчання пояснюється тими ж факторами, які зробили інтелектуальний аналіз даних і байєсівський аналіз популярними, ніж будь-коли. Такі речі, як зростання обсягів та різноманітності доступних даних, дешевша й потужніша обчислювальна здатність та доступне зберігання даних. Це все значить, що можна швидко й автоматично розробляти моделі, які мають змогу аналізувати більші, складніші дані та надавати швидші та точніші результати – навіть у дуже великих масштабах.

Машинне навчання використовується в різних галузях та визнано важливою частиною роботи галузі. Розглянемо використання машинного навчання в різних галузях, таких як:

- Фінансові послуги. Банки та інші підприємства фінансової індустрії часто використовують технології машинного навчання для наступних ключових цілей, таких як виявлення важливої інформації в даних і запобігання шахрайству. Ці ідеї мають змогу визначити можливості інвестування та дати допомогу інвесторам у виборі моментів торгівлі. Інтелектуальний аналіз даних має змогу визначати клієнтів, які мають профілі високого ризику та використовувати спостереження, щоб виявити попереджувальні ознаки шахрайства.
- Уряд. Такі державні установи, як безпека громади та комунальні підприємства, потребують використання машинного навчання, так як у них є декілька джерел даних, які можна добути для отримання інформації. Моніторинг даних датчиків, наприклад, визначає шляхи для економії грошей та підвищення ефективності роботи. Машинне навчання має змогу мінімізувати викрадення особистих даних та дати допомогти для виявлення шахрайства.

- Охорона здоров'я. Машинне навчання є тенденцією, що швидко розвивається в галузі охорони здоров’я, при появленні переносних пристроїв і датчиків, які мають змогу отримати дані для моніторингу та оцінки здоров’я людини в режимі реального часу. Машинне навчання також має змогу допомогти медикам аналізувати дані, щоб визначити тенденції або ознаки, які можуть призвести до покращення діагностики та лікування.
- Роздрібна торгівля. Веб-сайти використовують машинне навчання для рекомендації товарів, які ймовірно можуть користувачам сподобатися за допомогою використання інформації про попередні покупки. Це досягається за допомогою використання машинного навчання для моніторингу та аналізу історії покупок. Роздрібні продавці використовують машинне навчання для збирання даних, їх обробки і аналізу. Це робиться для персоналізації покупок, оптимізації цін, реалізації маркетингової кампанії, планування товарів і для аналізу клієнтів.

В наш час сфера машинного навчання стрімко розвивається та досягла високого рівня у розвинених країнах. Але в Україні ця сфера погано розвинена та потребує вдосконалення. Різні галузі потребують впровадження методів машинного навчання для моніторингу систем, аналізу, обробки даних і тд.

Два поширених метода машинного навчання – це навчання з учителем (Supervised learning) і навчання без учителя (Unsupervised learning). Також існують інші методи машинного навчання, такі як напівконтрольоване навчання (Semisupervised learning) та навчання з підкріпленням (Reinforcement learning). Розглянемо такі методи, як:

- Навчання без учителя. Такий підхід навчання використовується для даних, які не мають позначок цільового класу. Мета навчання в тому, щоб вивчити дані та знайти певну структуру всередині. Навчання без учителя добре працює з економічними та транзакційними даними. Популярними методами являються відображення найближчих сусідів, кластеризація k-середніх і

розкладання сингулярних значень. Ці алгоритми також використовуються для сегментації текстових тем, рекомендацій елементів та визначення викидів у даних.

- Навчання з підкріпленням. Воно найчастіше використовується для ігор, робототехніки та навігації. За допомогою навчання з підкріпленням алгоритм шляхом проб і помилок виявляє, які дії приносять найбільшу винагороду. Цей тип навчання складається з трьох гловних компонентів: агента (учень або особа, яка приймає рішення), оточення (все, з чим агент взаємодіє) і дії (що може робити агент). Мета полягає в тому, щоб агент обрав дії, які максимізують очікувану винагороду за певний проміжок часу. Агент досягне мети набагато швидше, дотримуючись правильної політики.

Кластеризація є одним із найбільш поширених методів дослідницького аналізу даних, які використовуються для отримання інтуїції щодо структури даних. Метод визначається, як завдання ідентифікації підгруп у даних, щоб точки даних в одній підгрупі (кластері) були дуже схожими, тоді як точки даних в різних кластерах дуже відрізнялися.

Розглянувши основні положення про сферу машинного навчання та його різновиди можна дійти висновків, що на даний момент являється дуже *актуальною* тема введення машинного навчання до практичного використання у різних сферах діяльності та галузях.

В даній роботі розглядається новий метод кластеризації даних на основі інформаційних критеріїв в системах моніторингу при оцінюванні рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу з використанням агенто-орієнтованого підходу (або навчання з підкріпленням).

РОЗДІЛ 1.
АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ МАТЕМАТИЧНИХ МОДЕЛЕЙ, МЕТОДІВ І ЗАСОБІВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ПОБУДОВИ МЕТОДІВ КЛАСТЕРИЗАЦІЇ В СИСТЕМАХ МОНІТОРИНГУ РІВНЯ СОЦІАЛЬНО-ЕКОНОМІЧНОГО РОЗВИТКУ КРАЇН

1.1 Аналітичний огляд існуючих методів кластеризації даних

1.1.1 Метрики для розрахунку відстані між об’єктами

Визначимо відстань між об'єктами як метрику для кластерного аналізу. Зворотною величиною від відстані між об'єктами є міра близькості або подібності об'єктів. В літературі зустрічається і інший термін - метрика, який є синонімом до словосполучення відстань між об’єктами та визначає метод обчислення відстані. Існує більше 50 різних методів обчислення метрики, які представлені у багатьох виданнях присвячених кластерному аналізу. Найбільш поширений спосіб в разі кількісних ознак є евклідова відстань або евклідова метрика.

$$d(X_i, X_j) = \left(\sum_{k=1}^{z} (x_{ki} - x_{kj})^2 \right)^{\frac{1}{2}} \tag{1.1}$$

Також евклідову відстань називають узагальнена статична відстань Маньківського [1], де використовується інша величина в степенях двійки. Зазвичай ця величина позначається символом "p". Якщо p = 2 отримується звичайна Евклідову відстань. Нижче представлений вираз для узагальненої метрики Мінковського.

$$d(X_i, X_j) = \left(\sum_{k=1}^{z} \left| x_{ki} - x_{kj} \right|^p \right)^{\frac{1}{p}} \tag{1.2}$$

Використовуються також такі метрики, як Манхеттенська відстань (відстань міських кварталів або city-block), метрика "домінування" (Sup-метрику або Супремум-норма). Метрика Маньківського визначає широкий набір метрик, що включає найбільш популярні метрики. Також є інші методи обчислення відстані між об'єктами, які відрізняються від метрик Маньківського та вказують на більш гарні результати порівняно з ними. Найбільш відома – це відстань Махаланобіса [2, с. 301-310].

Відстань Махаланобіса — це відстань між точкою та розподілом. А не між двома різними точками. Фактично це багатоваріантний еквівалент евклідової відстані. Визначимо алгоритм розрахунку відстані Махаланобіса:

1. Трансформування стовпців в некорельовані змінні;
2. Масштабування стовпців до стану, коли дисперсія дорівнює 1;
3. Обчислення евклідової відстані.

Для обчислення цієї відстані використовується формула 1.3.

$$D^2 = (x - m)^T \cdot C^{-1} \cdot (x - m), \tag{1.3}$$

де D^2 – це квадрат відстані Махаланобіса;

x –- є вектором спостереження (рядок у наборі даних);

m – це вектор середніх значень незалежних змінних (середнє значення кожного стовпця);

C^{-1} – це обернена коваріаційна матриця незалежних змінних.

Для оцінки інформативності використовується відстань Кульбака-Лейблера [3]. Відстань Кульбака-Лейблера є несиметричною мірою інформаційної розбіжності двох розподілів ймовірності, метод добре зарекомендував себе в прикладній статистиці, механікі рідин і машинному

навчанні. І на відміну від розглянутих модифікацій відстань Кульбака-Лейблера дозволить оцінювати міру інформаційної відмінності [3].

В роботі [4] розглянуті наступні метрики для роботи з номінальними і порядковими атрибутами: порядкові - метрики Спірмена, Кендалла; номінальні - метрики Жаккар, Рассела-Рао, Бравайса, Юла. Як зазначено раніше, це далеко не всі існуючі метрики. Відповідно, якщо у дослідника крім кількісних атрибутів є атрибути інших типів, то виникає задача знайти метрику між різнотипними ознаками, виміряними в різних шкалах. В цьому випадку необхідно вирішити задачу приведення всіх різнотипних шкал до однієї загальної шкали [5].

1.1.2 Метрика для оцінки якості кластеризації

Правильне вимірювання якості алгоритмів кластеризації є ключовим моментом в даній задачі. Тому слід розглянути метрики для оцінки якості методів та моделей кластеризації. На даний час існує велика кількість метрик, тому розглянемо популярні метрики, які підтвердили свою використовуваність за довгі роки у задачах кластеризації:

- Метрика «Силует» (Silhouette Score)[6]. Метрика «Силует» та графік метрики «Силует» використовуються для вимірювання відстані між кластерами. Він відображає міру того, наскільки близько кожна точка в кластері знаходиться до точок сусідніх кластерів. Цей показник має діапазон [-1, 1] і є чудовим інструментом для візуальної перевірки подібності всередині кластерів і відмінностей між кластерами. За його допомогою можна з легкістю визначити наскільки гарно розділяється той чи інший кластер. Ця метрика розраховується за формулою 1.4.

$$s = \frac{b-a}{\max(a,b)}, \tag{1.4}$$

де a – це середня відстань між вибіркою та всіма іншими точками в тому ж кластері;

b – це середня відстань між вибіркою та всіма іншими точками в наступному найближчому кластері.

Типові графіки метрики «Силует» представляють мітку кластера на осі y, а фактичний показник силуету – на осі x. Розмір/товщина силуетів також пропорційні кількості зразків всередині цього кластера. Приведемо приклад такого графіка на рис 1.1.

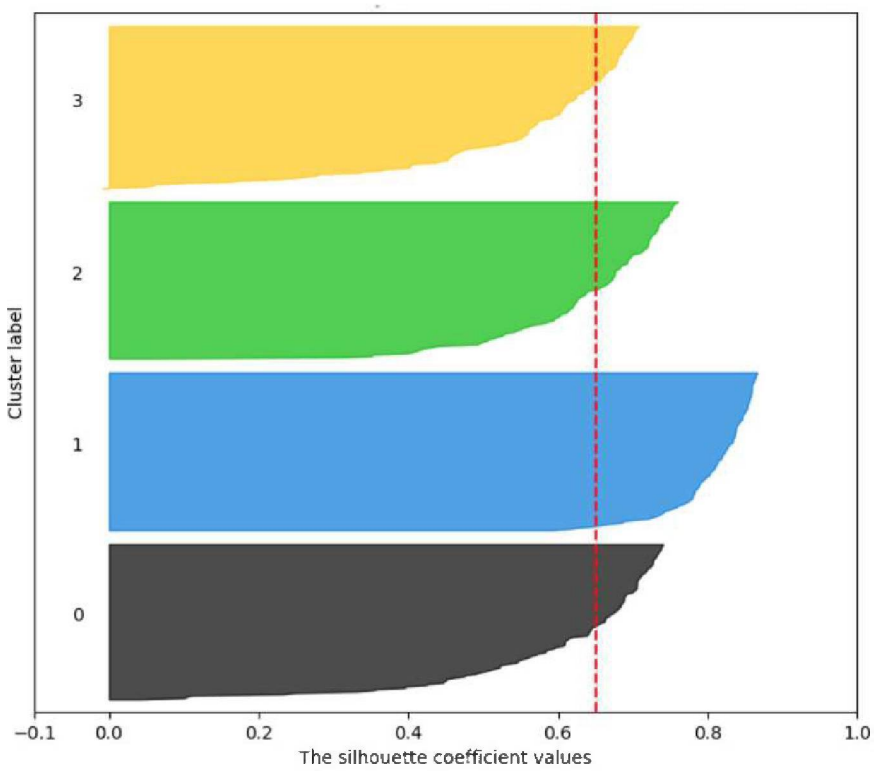

Рис. 1.1 — Приклад графіку метрики «Силует» [6]

Чим вищі коефіцієнти метрики «Силует» (чим ближче до +1), тим далі віддалені вибірки кластера від зразків сусідніх кластерів. Значення 0 вказує, що

вибірка знаходиться на кордоні прийняття рішень між двома сусідніми кластерами або дуже близько до неї. Натомість від'ємні значення вказують на те, що ці зразки, можливо, були призначені не тому кластеру. Усереднюючи коефіцієнти метрики «Силует», може бути отриманий глобальний показник метрики «Силует», який можна використовувати для опису ефективності усієї вибірки за допомогою одного значення.

- Індекс Ренда (Rand Index). Цей показник обчислює міру подібності між двома кластерами, враховуючи всі пари вибірок і підраховуючи пари, які призначені в однакових або різних кластерах у передбаченій і справжній кластерах. Ця метрика розраховується за формулою 1.5.

$$RI = \frac{NAP}{NP}, \tag{1.5}$$

де NAP – це кількість узгоджених пар;

NP – це загальна кількість пар.

RI може варіюватися від нуля до 1, де 1 визначає ідеальне розділення кластерів. Єдиний недолік індекса Ренда полягає в тому, що він припускає, що ми можемо знайти мітки кластерів реальної істини та використовувати їх для порівняння продуктивності нашої моделі, тому він набагато менш корисний, ніж оцінка метрики «Силует» для завдань чистого навчання без нагляду.

- Коригований індекс Ренда (Adjusted Rand Index)[7]. Цей показник – це індекс Ренда з поправкою на випадковість. Індекс Ренда обчислює міру подібності між двома кластеризаціями, враховуючи всі пари вибірок і підраховуючи пари, які відносяться до однакових або різних кластерів у передбачених і справжніх кластерах. Неочищений показник RI потім коригується на випадковість в оцінку ARI (Adjusted Rand Index) за формулою 1.6.

$$ARI = \frac{RI - ERI}{Max(RI) - ERI}, \quad (1.6)$$

де RI – індекс Ренда;

ERI – очікуваний індекс Ренда.

Коригований індекс Ренда, як і індекс Ренда, коливається від нуля до одиниці, причому нуль дорівнює випадковому позначенню, а одиниця — коли кластери ідентичні.

- Взаємна інформація (Mutual Information). Взаємна інформація є ще одним показником, який часто використовується для оцінки ефективності алгоритмів кластеризації. Це міра подібності двох міток з однаковими даними. Взаємну інформацію між кластерами U і V можна розрахувати за формулою 1.7.

$$MI(U,V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} log \frac{N|U_i \cap V_j|}{|U_i| * |V_j|}, \quad (1.7)$$

де |Ui| – кількість значень у кластері Ui;

|Vj| - кількість значень в кластері Vj.

Подібно до індексу Ренда, одним з основних недоліків цієї метрики є необхідність знати основні позначки істини для розподілу.

- Індекс Калінського-Харабаша [8]. Ця індекс також відомий як критерій відношення дисперсії. Оцінка визначається як відношення між дисперсією всередині кластера та дисперсією між кластерами. Ця метрика дозволяє оцінити продуктивність алгоритму кластеризації, без використання інформації про основні мітки істини. Чим вище індекс Калінського-Харабаша, тим краще продуктивність методу кластеризації даних. Ця метрика розраховується за формулою 1.8.

$$CH = \left[\frac{\sum_{k=1}^{K} n_k \|c_k - c\|^2}{K-1} \right] / \left[\frac{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \|d_i - c_k\|^2}{N-K} \right], \quad (1.8)$$

де n_k – це кількість точок k-го кластера;

c_k - центроїд k-го кластера;

c – глобальний центроїд;

N – загальна кількість точок даних.

Більше значення індексу Калінського-Харабаша означає, що кластери щільні та добре розділені, хоча немає прийнятного граничного значення. Зазвичай вибирається те рішення, яке дає пік або принаймні різкий згин на лінійній діаграмі індексів Калінського-Харабаша. З іншого боку, якщо лінія плавна (горизонтальна, висхідна чи низхідна), то немає причини віддавати перевагу одному рішенню перед іншими.

Також розглянемо метрики для оцінки якості моделі, які найчастіше використовується для задач класифікації. Ці метрики вимагають знання позначок істини для розподілу. Такими метриками є:

- Точність (Accuracy) – це найпростіший та інтуїтивно зрозумілий показник продуктивності класифікатора. Він розраховується по формулі 1.9, як відношення кількості правильно передбаченого класу до загальної кількості передбачень:

$$\text{Точність} = \frac{TP+TN}{T}, \quad (1.9)$$

де TP – кількість значень правильної класифікації позитивного класу;

TN – кількість значень правильної класифікації негативного класу;

T – загальна кількість передбачень.

Коли існує проблема дисбалансу класів, точність є неправильною метрикою для використання. Приклад: Нехай існує 2 класи: клас А становить

99% набору даних, а клас B – це решта 1%. Якщо прогнозувати клас A кожного випадку, буде досягнута точність 99%. За метрикою точності можна вважати, що модель працює чудово, але насправді модель працює дуже погано.

- Точність (Precision) і відклик (Recall). Загалом, існує компроміс між відкликом (відсоток дійсно позитивних випадків, які були класифіковані як такі) і точністю (відсоток позитивних класифікацій, які дійсно позитивні). Ці метрики можна розрахувати за формулами 1.10-1.11.

$$\text{Відклик} = \frac{TP}{TP+FN}, \tag{1.10}$$

де TP – кількість значень правильної класифікації позитивного класу;

FN – кількість значень неправильної класифікації негативного класу.

$$\text{Точність} = \frac{TP}{TP+FP}, \tag{1.11}$$

де TP – кількість значень правильної класифікації позитивного класу;

FP – кількість значень неправильної класифікації позитивного класу.

У ситуаціях, коли потрібно виявити екземпляри класу меншості, зазвичай більш важливою метрикою є відклик, ніж точність. Але, як сказано раніше, повинен бути компроміс між цими метриками.

- Метрика F. Ця метрика використовується в тих випадках, коли потрібно досягнути високого значення точності (Precision) і відклику (Recall) та отримати лише один показник якості. F1 можна розрахувати за формулою 1.12.

$$F1 = \frac{2*precision*recall}{precision+reca}, \tag{1.12}$$

де precision – точність;

recall – відклик.

Метрику F1 не можна коригувати в залежності від потреб класифікації, тому виник окремий випадок метрики, відомої як F-Beta, яка має, корегувальний параметер β. Цей параметер надає змогу корегувати значення відклику та точності в такій залежності, що чим більше значення β тим більша залежність метрики від відклику і тим менша від точності. Дана метрика розраховується за формулою 1.13.

$$F_\beta = (1 + \beta^2) * \frac{precision*recall}{\beta^2*precision+rec}, \quad (1.13)$$

де β – параметер корегування.

- Метрика ROC-AUC. Крива ROC-AUC є вимірюванням продуктивності для проблем класифікації, виявлення викидів при різних порогових налаштуваннях. ROC – це крива ймовірності, а AUC – ступінь або міра відокремленості. Ця метрика відображує, наскільки математична модель здатна розрізняти класи. Чим вища міра AUC, тим краще модель правильно прогнозує класи.

Крива ROC зображена на рис. 1.1, де відображена залежність TPR (True Positive Rate) від FPR (False Positive Rate). По осі ординат відображена міра TPR, по осі абсцис — FPR.

TPR – це частка правильних прогнозів у прогнозах позитивного класу. Ця міра розраховується за формулою 1.14.

$$TPR = \frac{TP}{TP+FN}, \quad (1.14)$$

де TP – кількість значень правильної класифікації позитивного класу;

FN – кількість значень неправильної класифікації негативного класу.

FPR – це частка неправильних прогнозів у прогнозах позитивного класу. Ця міра розраховується за формулою 1.15.

$$FPR = \frac{FP}{FP+T}, \tag{1.15}$$

де FP – кількість значень неправильної класифікації позитивного класу;

TN – кількість значень правильної класифікації негативного класу.

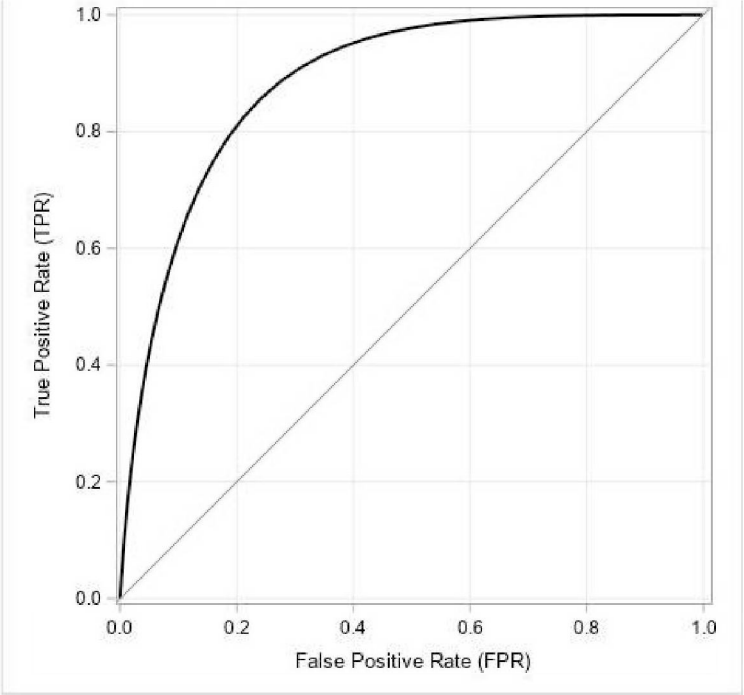

Рис. 1.2 — Крива ROC

Площа під кривою ROC визначає значення метрики ROC-AUC. Для відмінної моделі значення цієї метрики буде дорівнювати одиниці. Діагональна пряма, яка відображена на рис. 1.2 вказує на модель, яка не має змоги відрізнити позитивний клас від негативного, тобто не має можливості розділення класів.

Цю метрику слід використовувати, коли модель повинна працювати однаково добре як на позитивному, так і на негативному класі.

1.1.3 Математичні моделі і методи кластеризації даних

Будь-який метод кластеризації даних будує математичну модель за допомогою якої розділяє набір даних на кластери. Тому під словом метод кластеризації будемо вважати метод та його математичну модель. Розрізняють два види методів кластеризації даних: ієрархічні і неієрархічні [9-10]. Ієрархічні методи виконують послідовне поєднання елементів в кластери використовуючи міру їх близькості або ітеративне розділення сукупності елементів на більш маленькі кластери. В цьому випадку результат кластеризації являє собою ієрархічну структуру вкладених кластерів. Неієрархічні методи мають змогу знаходити й ідентифікувати "згущення" об'єктів в просторі змінних. Виділяють також кластеризацію "з навчанням", при якій передбачається, що кількість класів відома раніше та існує тренувальна вибірка – сукупність елементів, де відомо, до яких класів вони належать. Решта об'єктів класифікуються за мірою близькості до об'єктів з навчальної вибірки.

Головними методами ієрархічної кластеризації даних являються метод ближнього сусіда, метод повного зв'язку, агломеративний метод, метод середньої зв'язку Кінга і метод Уорда (в деяких виданнях метод Варда). Існують також центроїдні методи і методи, які використовують медіану. В роботі [11], описані негативні сторони використання цього типу методів. Неієрархічних методів більше, хоча вони використовують однакові принципи. Загалом, вони являються ітеративними методами розбиття вхідної сукупності. В процесі розподілу формуються нові кластери. Це виконується до моменту зупинки. Методи розрізняються між собою вибором початкових точок, правилом створення нових кластерів і правилом зупинки. Найбільш часто використовується алгоритм К-середніх [12; 13, стор. 125-134], який є досить швидким та ефективним. При використанні цього алгоритму розробник фіксує кількість кластерів в вихідному розбитті.

Розглянемо більш детальніше три методи кластеризації:

- Метод кластеризації k-середніх [14] - це метод векторного квантування, спочатку отриманий з обробки сигналів, який спрямований на розподіл n спостережень на k кластерів, в яких кожне спостереження належить до того кластера, де середнє значення найближче (центри кластера або центроїд кластера). Із цього слідує, що простір даних розділяється на комірки Вороного. Метод кластеризації k-середніх мінімізує дисперсії всередині кластеру (квадратичні відстані Евкліда), але не регулярні відстані Евкліда, що являється складною задачею Вебера, де середнє значення вибірки оптимізує квадратичні помилки, а геометрична медіана вибірки мінімізує евклідові відстані.

Проблема досить складна для обчислення, але евристичні алгоритми мають гарну ефективність та швидко сходяться до локального оптимального значення. Як правило вони є подібними до методу максимізації очікувань для сумішей розподілів Гауса з використанням послідовного підходу уточнення, що застосовується як для k-середніх, так і для моделювання суміші Гауса. Обидва методи використовують центроїди для моделювання вибірки, але кластеризація k-середніх, зазвичай, виявляє кластери порівнянної просторової протяжності, а механізм максимізації очікувань визначає кластери різної форми.

Визначимо алгоритм методу кластеризації k-середніх. Цей найвідоміший алгоритм використовує послідовну техніку уточнення. Через нескінченість його зачастіше називають "алгоритмом k-середніх". Також ще одною назвою може бути алгоритм Лойда, часто в інформатиці. Іноді цей метод називають "наївним k-середнім", оскільки існують більш швидші альтернативи. [15] Дано початковий набір k-середніх $m_1^{(1)},...,m_k^{(1)}$, алгоритм проходить шляхом виконання та чергування двох кроків [16]:

- Крок призначення: Призначте кожне спостереження кластеру з більш близьким середнім значенням вибірки, тобто з найменшим квадратом

відстані Евкліда. (З точки зору математики це означає розділити спостереження відповідно до діаграми Вороного, сформованої за допомогою середніх значень.

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\}, \tag{1.16}$$

де кожен x_p призначається рівно одному $S^{(t)}$, навіть якщо він міг бути призначений двом або більше з них.

- Крок оновлення: Розрахувати ще раз середні значення (центроїди) для елементів, призначених кожному кластеру.

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j \tag{1.17}$$

Алгоритм сходиться, коли центроїди більше не змінюються. Також алгоритм не дає гарантії знаходження оптимального. [17] Цей алгоритм більш часто визначається, як призначення об'єктів найближчому кластеру використовуючи міру відстані. При використанні іншої міри відстані, крім евклідової відстані, алгоритм може алгоритму гірше сходитися, або не сходитися взагалі.

- Агломеративний метод кластеризації. Він є найпоширенішим типом ієрархічної кластеризації, яка використовується для групування об’єктів у кластери на основі їх схожості. Він також відомий як AGNES (агломеративна вкладеність). Алгоритм починається з обробки кожного об’єкта як однотонного кластера. Далі, пари кластерів послідовно об’єднуються, поки всі кластери не будуть об’єднані в один великий кластер, що містить усі об’єкти. Результатом є представлення об’єктів на основі дерева, яке називається дендрограмою. Приклад дендрограми приведено на рис 1.3.

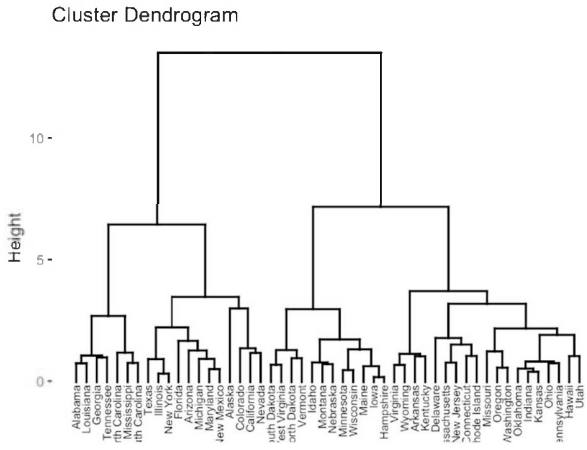

Рис. 1.3 – Дендрограма агломеративного методу кластеризації [18]

Вибираючи потрібну глибину дерева, можна розбити вибірку на потрібну кількість кластерів.

- Метод середнього зсуву [19] (Mean Shift) – це алгоритм кластеризації на основі навчання без вчителя, який має на меті виявити краплі в плавній щільності вибірок. Це алгоритм на основі центроїдів, який працює шляхом оновлення кандидатів для центроїдів, як середнє значення точок у певному регіоні. Ці кандидати потім фільтруються на етапі після обробки, щоб усунути дублікати, щоб сформувати остаточний набір центроїдів. Тому, на відміну від К-середніх, він не потребує самостійного вибору кількості кластерів.

На високому рівні Mean Shift працює наступним чином:

1. Створення розсувного вікна/кластеру для кожної точки даних;
2. Кожне з ковзних вікон зміщується до областей з більшою щільністю шляхом зміщення їх центру (центру ковзного вікна) до середнього значення точок даних у ковзному вікні. Цей крок повторюватиметься до тих пір, поки жоден зсув не приведе до більшої щільності (кількості точок у ковзному вікні);
3. Виділення розсувних вікон шляхом видалення вікон, що перекриваються. Коли кілька розсувних вікон перекриваються, вікно, що містить найбільшу кількість точок, зберігається, а інші видаляються.

4. Призначення точок даних розсувному вікну, в якому вони знаходяться.

Визначимо переваги та недоліки даного методу кластеризації. До переваг можна віднести, те що це простий кластерний метод, який дуже добре працює з даними сферичної форми. Крім того, він автоматично вибирає кількість кластерів на відміну від інших алгоритмів кластеризації. Також вихід середнього зсуву не залежить від ініціалізації, оскільки на початку кожна точка є кластером. До недоліків можна віднести, те що алгоритм не дуже масштабований, оскільки під час його виконання вимагає багаторазового пошуку найближчих сусідів. Він має складність $O(n^2)$. Крім того, ручний вибір пропускної здатності може бути нетривіальним, а вибір неправильного значення може призвести до поганих результатів.

Розглянувши більш детально деякі методи кластеризації перейдемо до методів вибору кількості кластерів. Один із найпоширеніших це метод ліктя.[20] Метод складається з побудови варіації, як функції кількості кластерів і вибору вигину кривої в якості кількості використовуваних кластерів. Той же метод можна використовувати для вибору кількості параметрів в інших моделях, керованих даними, наприклад кількості основних компонентів для опису набору даних.

Вибираючи серед ієрархічних та неієрархічних методів, треба враховувати такі моменти. Неієрархічні методи мають більш вищу стійкість до: викидів, неправильного вибору міри схожості, вибору мало значущих змінних для кластеризації та ін. Але розробник повинен перед початком фіксувати кількість кластерів на виході, правило зупинки та, якщо потрібно, початковий центр кластера. Остання умова сильно впливає на ефективність роботи методу кластеризації. Якщо немає можливості задати таку умову, рекомендується використання ієрархічних методів. Відзначимо істотну властивість для обох типів методів: кластеризація всіх елементів не завжди є вірним рішенням. Ймовірно, найбільш оптимально буде перед початком видалити викиди з

вибірки, а після цього продовжити кластеризації. Також можна не вказувати занадто високий критерій зупинки (наприклад можна робити зупинку, коли кластеризовано більше 90% спостережень).

Відзначимо, що вибір конкретного методу кластеризації, вимагає від аналітика гарного знайомства з природою і передумовами методів, в іншому випадку отримані результати будуть схожі на «середню температуру по лікарні». Для того щоб переконатися в тому, що обраний метод дійсно ефективний в даній області, як правило, застосовують наступну процедуру: розглядають кілька апріорі різних між собою груп і перемішують їх представників між собою випадковим чином. Потім проводять процедуру кластеризації з метою відновити вихідне розбиття на групи. Показником ефективності роботи методу буде частка збігів об'єктів в виявлених і вихідних групах.

Деякі автори приходять до висновку, що вибір метрики і методу кластеризації не є ключовим моментом в кластерному аналізі. Таке твердження можливо в наступних випадках. По-перше, воно стосується тільки неієрархічних методів. По-друге, в будь-якому випадку необхідно вибирати метрику таким чином, щоб вона не суперечила ідеї обраного методу об'єднання кластерів. Особливу увагу слід приділити вибору метрики в разі, якщо змінні є залежними. Адекватна метрика може бути вирішенням даної проблеми.

1.2 Аналітичний огляд існуючих програмних засобів для кластеризації даних

Розглянемо існуючі програмні засоби для задач кластеризації. Бібліотека scikit-learn в Python реалізує різні алгоритми, методи і математичні моделі машинного навчання, включаючи і кластеризацію даних. Виділимо 10 популярних алгоритмів кластеризації реалізованих у цій бібліотеці, таких як:

- Поширення спорідненості (Affinity Propagation);
- Агломеративна кластеризація (Agglomerative Clustering);
- BIRCH;
- DBSCAN;
- K-середніх (K-Means);
- Mini-Batch K-Means;
- Середній зсув (Mean Shift);
- OPTICS;
- Спектральна кластеризація (Spectral Clustering);
- Суміш гаусів (Mixture of Gaussians).

На рисунку 1.4 відображене графічне порівняння цих методів та відображення їх роботи на різних наборах даних. Розглянемо реалізацію деяких алгоритмів більш детально:

- *BIRCH.* Кластеризація BIRCH (BIRCH - скорочення від збалансоване ітеративне скорочення та кластеризація за допомогою ієрархій) передбачає побудову деревної структури, з якої витягуються центроїди кластера. Більш детально описано в роботі «BIRCH: An efficient data clustering method for large databases, 1996»[21].

Він реалізований за допомогою класу Birch, і основною конфігурацією для налаштування є гіперпараметри «threshold» та «n_clusters», останній з яких забезпечує оцінку кількості кластерів.

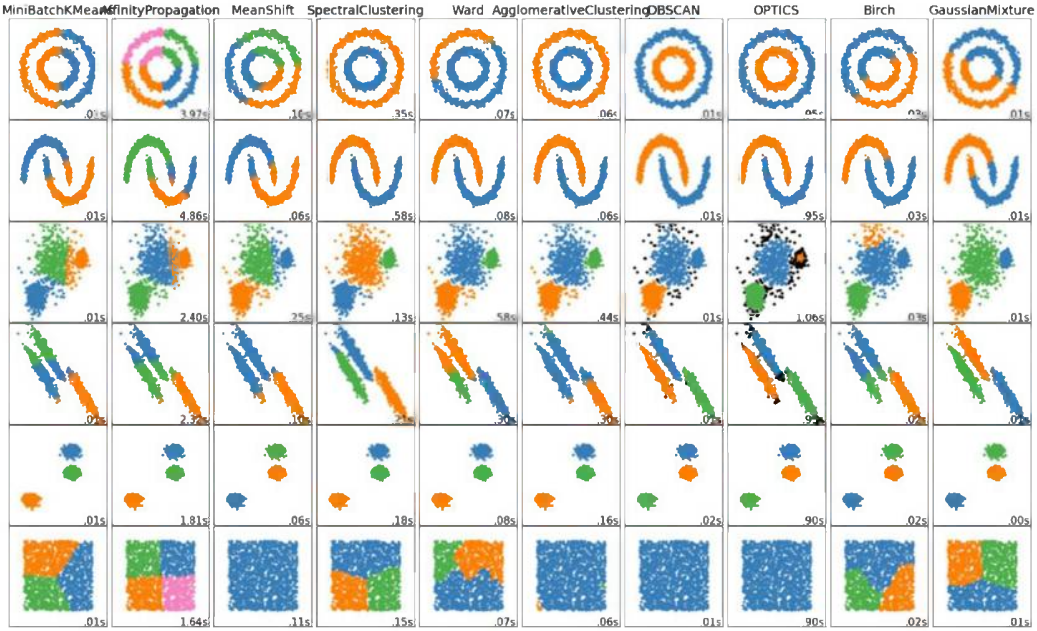

Рис. 1.4 - Порівняння алгоритмів кластеризації в scikit-learn [22]

Реалізація на мові Python [23] представлена в лістингу Г.1. У даному фрагменті коду виконується генерація вибірки, будується модель за алгоритмом BIRCH та виконується розбиття на кластери. Результат розбиття відображений на рисунку 1.5.

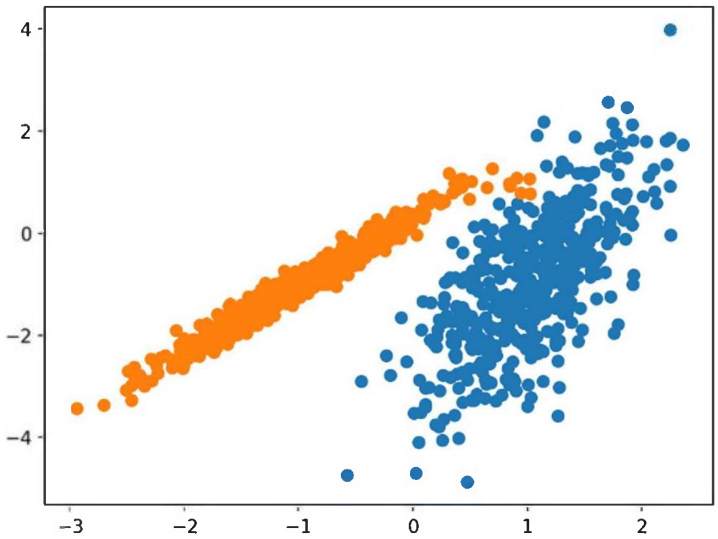

Рис. 1.5 – Алгоритм BIRCH [23]

- *DBSCAN.* Розглянемо ще один досить потужний алгоритм під назвою DBSCAN.[22] Алгоритм розглядає кластери , як ділянки високої щільності , розділених районах з низькою щільністю. Завдяки цьому досить загальному погляду кластери, знайдені DBSCAN, можуть мати будь-яку форму, на відміну від k-середніх, що передбачає, що кластери мають опуклу форму. Центральним компонентом DBSCAN є концепція основних зразків , які є зразками, що знаходяться в зонах високої щільності. Отже, кластер - це сукупність основних зразків, кожна близька одна до одної (вимірювана за допомогою певної міри відстані), і набір непрофільних зразків, які знаходяться близько до основного зразка (але самі не є основними зразками).

Кластер - це набір основних зразків, який може бути побудований шляхом рекурсивного відбору основного зразка, пошуку всіх його сусідів, які є основними зразками, пошуку всіх *їх* сусідів, що є основними зразками тощо. Кластер також має набір непрофільних зразків, які є зразками, які є сусідами основного зразка в кластері, але самі не є основними зразками. Інтуїтивно ці зразки знаходяться на краях кластера.

Реалізація [24] прикладу використання алгоритму DBSCAN представлена в лістингу Д.1. В цьому прикладі генерується та нормалізується вибірка, будується модель алгоритму, розбивається вибірка на кластери, розраховуються метрики та будується графік. Графік показано на рисунку 1.6. Там колір позначає приналежність до кластера, а великі кола позначають зразки ядра, знайдені алгоритмом. Менші кола - це непрофільні зразки, які все ще є частиною кластера. Більше того, викиди позначені чорними крапками.

Реалізації та більш детальну інформацію про інші методи з бібліотеки skicit-learn можна знайти на офіційному сайті бібліотеки [22] та інших обзорах цих методів [23].

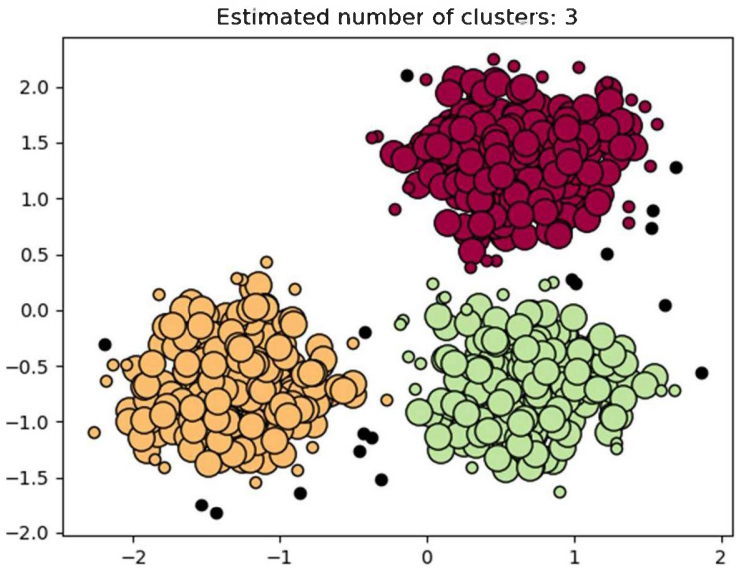

Рис. 1.6 – Приклад результату алгоритму DBSCAN [24]

Проаналізувавши бібліотеку skicit-learn для Python перейдемо до розгляду інших програмних засобів для кластеризації даних. Розглянемо Open Source Clustering Software [25]. Воно включає бібліотеку кластеризації C, програму графічного інтерфейсу Cluster 3.0, модуль розширення Python, модуль доступний для Perl.

Бібліотека кластеризації C [25] складається з колекції процедур кластеризації, які можна використовувати для аналізу даних про експресію генів. Підпрограми для ієрархічної (парних простих, повного, середнього і центроїдного зв'язку) кластеризації, кластеризації k -середніх і k -медіан та 2D карт на прямокутній сітці.

Програма графічного інтерфейсу Cluster 3.0, доступна для Windows, Mac OS X та Linux / Unix, забезпечує найпростіший доступ до методів кластеризації в бібліотеці кластеризації C. Cluster 3.0 - це вдосконалена версія оригінальної програми Cluster / TreeView, розроблена Майклом Айзеном з лабораторії Berkeley . Також доступна версія командного рядка Cluster 3.0.

Для користувачів Python розроблений модуль розширення Python [26], який надає доступ до процедур у бібліотеці кластеризації C. Подібний модуль доступний для Perl. Процедури також можна викликати безпосередньо з інших програм C за допомогою вихідного коду.

Розглянемо ще один програмний пакет для кластеризації під назвою CLUTO [26]. CLUTO - це програмний пакет для кластеризації мало- та багатовимірних наборів даних та для аналізу характеристик різних кластерів. CLUTO добре підходить для кластеризації наборів даних, що виникають у багатьох різноманітних областях застосування, включаючи пошук інформації, транзакції покупців клієнтів, Інтернет, ГІС, науку та біологію. Дистрибутив CLUTO складається як з окремих програм, так і з бібліотеки, за допомогою якої прикладна програма може отримувати безпосередній доступ до різних алгоритмів кластеризації та аналізу, реалізованих у CLUTO. Відзначимо такі особливості, як [26]: кілька класів алгоритмів кластеризації: на основі розділів, агломерацій та розділення графів; кілька функцій подібності / відстані: Евклідова відстань, косинус, коефіцієнт кореляції, розширений Жакард, визначений користувачем; численні нові критерійні функції кластеризації та агломеративні схеми злиття; традиційні агломеративні схеми злиття: одинарне, повне посилання, UPGMA; широкі можливості візуалізації кластера та параметри виводу: приписка, SVG, gif, xfig тощо; кілька методів для ефективного узагальнення кластерів: найбільш описові та чіткі розміри, кліки та найчастіші набори предметів; може масштабуватися до дуже великих наборів даних, що містять сотні тисяч об’єктів і десятки тисяч розмірів.

1.3 Постановка задачі дослідження

Об'єктом дослідження даної кваліфікаційної роботи являються процеси моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Предметом дослідження даної кваліфікаційної роботи являються методи і моделі оцінки рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу на основі впровадження програмних засобів в системи моніторингу.

Мета даної кваліфікаційної роботи – підвищення рівня стійкості до відмов систем шляхом зниження ймовірності некоректного визначення стану систем (помилки третього роду при кластерному аналізі стану систем) за допомогою ефективного методу кластеризації даних на основі інформаційних критеріїв в системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Для досягнення мети потрібно виконати наступний перелік сформованих задач, таких як: визначення постановки задачі кластеризації даних; аналітичний огляд існуючих методів кластеризації даних; аналітичний огляд існуючих програмних засобів для кластеризації даних; формалізація уявлення вхідних даних; підготовка та обробка вхідних даних для покращення ефективності методу; опис методу кластеризації даних; опис обчислювального алгоритму вирішення задачі; порівняння ефективності різних методів кластеризації даних; огляд та аналіз отриманих результатів.

Отже дана робота присвячена розробці ефективного методу кластеризації даних на основі інформаційних критеріїв в системах моніторингу використовуючи програмні засоби, який має високі показники точності та може використовуватися для моніторингу систем різних галузей та напрямків.

Тому перейдемо до визначення постановки задачі кластеризації даних. Нехай задано загальну вибірку вхідних даних в такому вигляді [27]:

$$\tilde{X} = \left(x_{0j} : x_{ij} \in X\right),\ j = \overline{1,m}, i = \overline{1,n} \quad , \tag{1.18}$$

де $x_{0j} \in (x_{01}, x_{02}, \ldots, x_{0m})$ – вектор цільових ознак;

X – матриця вхідних ознак.

Задана метрика подібноті між об'єктами $\rho(x,x')$. Треба розділити вибірку на підмножини, які не пересікаються. Такі множини називаються кластерами, де кожен кластер складається з елементів, схожих за метрикою, та об'єкти різних кластерів сильно відрізняються. Кожному об'єкту $x_i \in X^m$ задається номер кластера y_i.

Алгоритм кластеризації даних [27] - це функція $a: X \to Y$, яка для любого елемента $x \in X$ визначає номер кластера $y \in Y$. Множина Y іноді відома перед початком задачі, але часто ставиться задача отримати оптимальну кількість кластерів, в залежності від визначеного критерію якості кластеризації.

Рішення задачі кластеризації даних принципово неточно. Цьому є кілька причин [27]:

- Немає точно найкращого критерію якості моделі кластеризації. Відомо множину евристичних критеріїв, а також множину алгоритмів, які не мають точно вираженого критерію, але виконують досить якісну кластеризацію. Всі вони можуть давати різні результати.
- Як правило, число кластерів невідомо перед початком і встановлюється в залежності від міркувань експерта.
- Є залежність результату кластеризації від метрики. Як правило вибір якої, також суб'єктивно визначається експертом.

Висновки до розділу 1

В підрозділі 1.1.1, проаналізувавши науково-технічну літературу та інформацію в мережі Інтернет були розглянуті метрики для знаходження відстані, метрики для оцінки якості кластеризації та проведено аналітичний огляд методів та моделей кластеризації. Також були визначені недоліки та переваги використання тих, чи інших метрик та математичних методів/моделей кластеризації.

В підрозділі 1.1.2 були розглянуті програмні засоби для вирішення задач кластеризації, які мають відкритий доступ та які можна застосовувати для прикладних задач.

В підрозділі 1.1.3 була визначена постановка задачі дослідження. Були описані предмет, об'єкт, мета та задачі, які потрібно розробити для досягнення мети дослідження. Також була сформульована постановка задачі кластеризації та визначено чому завдання кластеризації принципово неоднозначно.

РОЗДІЛ 2.
ОПИС ВХІДНИХ ДАНИХ, ЇХ ОБРОБКИ ТА МЕТОДУ КЛАСТЕРИЗАЦІЇ ДАНИХ НА ОСНОВІ ІНФОРМАЦІЙНИХ КРИТЕРІЇВ ТА АГЕНТНО-ОРІЄНТОВАНОГО ПІДХОДУ

2.1 Формалізація уявлення вхідних даних

Для застосування тих и інших методів, математичних моделей або алгоритмів кластеризації потрібно мати уявлення про вхідні та вихідні дані.

Розглянемо в якому виді повинні бути представлені дані для коректної роботи кластеризації даних. Найчастіше вибірки представляють у вигляді таблиці та використовують розширення файлів csv. В цих файлах кожен стовпець вибірки розділений розділювачем типу коми, точки з комою та інших. Для різних програмних засобів виділяють різні формати вхідних даних, але представлення у самому програмному засобі найчастіше однакове.

Нехай було завантажено вибірку у програмний засіб. Вона представлена у вигляді таблиці чи матриці. Розмірність вибірок може бути досить різною, від декількох одиниць/десятків до мільярдів елементів і більше. Дані діляться на числові та категоріальні. Різні методи, математичні моделі та алгоритми по різному працюють з цими даними та відповідно видають різні показники ефективності.

Так, наприклад, стандартний алгоритм k-середніх не застосовується безпосередньо до категоріальних даних з різних причин. Зразок простору для категоріальних даних є дискретним і не має природного походження. Евклідова функція відстані на такому просторі насправді не має значення.

Існує різновид k-засобів, відомий як k-режими, представлений у роботі [28] Чжесюе Хуангом, що підходить для категоріальних даних. У статті Хуанга [28] також є розділ про "k-прототипи", який застосовується до даних із поєднанням категоріальних та числових ознак. Він використовує міру відстані,

яка поєднує відстань Хеммінга для категоріальних ознак та евклідову відстань для числових ознак.

Дещо більше про категоріальні дані. Алгоритми кластеризації, які базуються на відстані мають змогу обробляти категоріальні дані. Потрібно лише вибрати відповідну функцію відстані, таку як відстань Гоуера, яка поєднує атрибути за необхідністю в єдину відстань. Тоді ви можете запустити ієрархічну кластеризацію, DBSCAN, OPTICS та багато іншого. Вибір функції відстані має високий вплив на результати кластеризації.

Згадаємо визначення кластеризації. Кластеризація даних - це завдання згрупувати набір елементів таким чином, щоб елементи однієї групи (які називаються кластерами) були більш схожими (в тому чи іншому значенні) між собою, ніж з іншими групами (кластерами). Отже, для кластеризації потрібна якісна подібність , тому алгоритм знає, коли об'єкти «більш схожі», ніж інші.

Ось чому в багатьох алгоритмах використовується певна форма відстані: ближче = більше схоже. Це дуже інтуїтивний спосіб визначити подібність. З безперервними змінними це досить складно, щоб правильно нормалізувати дані. Для багатовимірних даних люди іноді застосовують PCA. Якщо масштабування / зважування трохи не відповідає, результати все одно можуть бути гарними. Подібним чином, якщо у даних є невелика помилка, це лише незначно впливає на відстань.

На жаль, це не переноситься на дискретні чи категоріальні змінні. Використовується безліч підходів, таких як одноразове кодування (кожна категорія стає своїм атрибутом), двійкове кодування (перша категорія 00, друга 01, третя 10, четверта 11) що ефективно відображає ваші дані в просторі, де можливо було б використовувати k-засоби та інше методи кластеризації. Але ці підходи дуже крихкі. Вони, як правило, працюють, якщо є лише двійкові категорії та вони не дуже часто змінюються. Але проблема в тому, що існує низька дискримінація. Може бути ситуація, коли існує 0 об'єктів на відстані 0

(це були б дублікати) та сотні об'єктів на відстані 2. Тож, який би алгоритм не був використовувано, йому доведеться об'єднати всі ці об'єкти одночасно, оскільки вони мають абсолютно однакову схожість.

Отже використовувати категоріальні дані у кластеризації вибірок дуже складно. Підсумуємо *визначення вхідних даних для задач кластеризації. Вхідні дані* – це вибірка представлена у вигляді таблиці чи матриці (стовпці – змінні, рядки - приклади), яка має числові, категоріальні або найчастіше змішані дані відображені у будь-якому форматі файлу з розділювачами, який може зчитати програмний засіб.

Після проведення процесу кластеризації даних буде отримано вихідні дані. Так як, програмних засобів багато, то у кожного засобу є свої особливості. Виділимо найчастішу модель відображення вихідних даних. Дані представлені у вигляді пари «індекс» - «номер кластеру». Тобто прикладу з індексом «і» буде визначено номер кластеру, який відповідає індексу «і» з результуючої вибірки. Вихідні дані можуть бути записані у будь-який формат файлу, який підтримує програмний засіб.

Підсумуємо *визначення вихідних даних для задач кластеризації. Вихідні дані* – це вибірка представлена у вигляді матриці, таблиці або мапи, де зберігаються пари «індекс» - «номер кластеру». Ця вибірка може бути записана у будь-який формат файлу, який підтримує програмний засіб.

В якості вхідних даних буде використана вибірка рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу. Вибірка відображає соціально-економічний розвиток країн за 2012-2020 роки. Вибірка представляє в сумі 37 змінних стану. За допомогою вибірки потрібно оцінити рівень соціально-економічного розвитку за допомогою кластеризації.

Розглянемо більш детальніше поля вибірки та опишемо їх:

1. EGI_XXXX: індекс електронного уряду (E-Government Index) за «XXXX» рік. Множина значень для позначки «XXXX» включає наступні роки: 2012, 2014, 2016, 2018, 2020;

2. EPI_XXXX: індекс електронної участі (E-Participation Index) за «XXXX» рік. Множина значень для позначки «XXXX» включає наступні роки: 2012, 2014, 2016, 2018, 2020;
3. OSI_XXXX: індекс онлайн-послуг (Online Service Index) за «XXXX» рік. Множина значень для позначки «XXXX» включає наступні роки: 2012, 2014, 2016, 2018, 2020;
4. HCI_XXXX: індекс людського капіталу (Human Capital Index) за «XXXX» рік. Множина значень для позначки «XXXX» включає наступні роки: 2012, 2014, 2016, 2018, 2020;
5. TII_XXXX: індекс телекомунікаційної інфраструктури (Telecommunication Infrastructure Index) за «XXXX» рік. Множина значень для позначки «XXXX» включає наступні роки: 2012, 2014, 2016, 2018, 2020;
6. NRI_19 : індекс готовності до мережі (Networked Readiness Index) за 2019 рік;
7. ICT_XX : індекс розвитку інформаційно-комунікаційних технологій (The ICT Development Index) за «20XX» рік. Множина значень для позначки «XX» включає наступні значення: 12, 13, 15, 16, 17;
8. SPI_XX : індекс соціального прогресу (Social Progress Index) за «20XX» рік. Множина значень для позначки «XX» включає наступні значення: 14, 15, 16, 17, 18, 20;
9. Cluster: категоріальна змінна, яка вказує на приналежність до певного кластеру. Представлена у вигляді строки чотирьох видів, таких як «High income», «Low income», «Lower middle income», «Upper middle income». Ця змінна стану являється цільовою для даної задачі.

2.2 Підготовка та обробка вхідних даних

Важливою частиною у будь-якій задачі машинного є підготовка та обробка вхідних даних. Це робиться через те, що зазвичай вхідні дані представлені у такому вигляді, що не можуть використовуватися у математичному методі або моделі. До етапу підготовки та обробки даних входять наступні кроки:

- *Конвертація типів змінних.* Іноді при зчитуванні даних можуть виникати помилки. Прикладом може слугувати ситуація, коли змінна типу числа, зчитується як рядок;
- *Видалення непотрібних змінних стану, вибір корисних змінних стану.* Для покращення математичних методів та моделей потрібно вірно обрати змінні стану, які при їх використанні давали би гарні показники оцінки якості. Для визначення корисності змінних стану, використовують матрицю кореляції, дерева рішень та інші методи;
- *Видалення повторюваних або невідповідних спостережень.* Через похибки при вимірюванні або збору даних іноді виникають повторювані або невідповідні спостереження, які потрібно видалити для коректної роботи математичних моделей та методів;
- *Робота з пропущеними значеннями.* Як згадувалося раніше, при зборі інформації можуть виникати помилки, або непередбачувані ситуації при яких частина даних не буде записана у вибірку. Загальними методами для роботи з пропущеними значеннями є видалення рядків, видалення змінних стану, заповнення змінних на основі середнього значення або медіани. Також існують варіанти заповнення пропущених значень за допомогою методів машинного навчання, таких як лінійна регресія, випадковій ліс та інші;

- *Трансформація змінних стану.* Зазвичай математичні методи та моделі працюють краще на даних, які приведені до нормального розподілу. Тому змінні стану з розподілом, який відрізняється від нормального трансформують. Для трансформації можуть бути використані такі методи трансформації, як логарифмічний, поліноміальний, Бокса-Кокса та інші;
- *Нормалізація або стандартизація змінних стану.* Використання нормалізованих або стандартизованих даних, тобто приведених числових даних з широкого інтервалу значень до більш вузького інтервалу дозволяє методу швидше сходитися при навчанні та покращує ефективність методу;
- *Трансформація категоріальних змінних.* Якщо у вибірці існують категоріальні змінні, то потрібно ці змінні перевести у числові шляхом використання різних методів кодування. Це потрібно через те, що математичні методи та моделі вміють працювати тільки з числовими даними;
- *Робота з викидами (аномальними значеннями).* Часто у вибірках присутні викиди або аномальні значення, вплив яких, погіршує роботу математичних методів та моделей. Через це виникає потреба в очищенні наборів даних від викидів. Для цих потреб використовуються спеціальні методи, такі як IQR, ізоляційний ліс, місцевий фактор викиду (Local Outlier Factor), однокласовий SVM (One Class Support Vector Machines), надійна коваріація (Robust Covariance) та інші;
- *Зменшення розмірності.* Зазвичай методи зменшення розмірності використовуються для візуалізації даних вибірки. У більш рідких випадках ці методи використовуються для зменшення розмірності

даних для тренування математичних моделей та методів. Найбільш відомими традиційними методами є PCA (Principal Component Analysis), T-SNE (t-distributed Stochastic Neighbor Embedding), SVM (Support Vector Machines). Із методів глибокого навчання для зменшення розмірності популярними є автоенкодери.

В даній кваліфікаційній роботі проведена підготовка та обробка вхідних даних використовуючи пункти вище. Для виконання дій з вхідними даними використовувалась мова Python та допоміжні бібліотеки, такі як:

- Pandas [29] – це швидкий, потужний, гнучкий і простий у використанні інструмент аналізу та маніпулювання даними з відкритим вихідним кодом, побудований на основі мови програмування Python;
- Numpy [30] – це основний пакет для наукових обчислень з Python;
- Scikit-learn [31] – це модуль Python для машинного навчання;
- Matplotlib [32] – це повна бібліотека для створення статичних, анімованих та інтерактивних візуалізацій в Python;
- Seaborn [33] – це бібліотека візуалізації даних Python на основі matplotlib. Вона забезпечує високорівневий інтерфейс для малювання привабливої та інформативної статистичної графіки;
- SciPy [34] – це бібліотека, яка надає алгоритми для оптимізації, інтегрування, інтерполяції, задач на власні значення, алгебраїчних рівнянь, диференціальних рівнянь, статистики та багатьох інших класів задач.

Розглянемо кроки обробки даних для вибірки рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Першим кроком було завантаження даних з csv файлу, перевірка правильності завантаженої інформації та видалення непотрібних змінних стану.

Після цього кроку була отримана вибірка для подальшої обробки. Ця вибірка представлена на рис. 2.1.

Country Name	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018	OSI_2018	HCI_2018	TII_2018	...	ICT_16	ICT_15	ICT_13	ICT_12	SPI_20	SPI_18	SPI_17	SPI_
Albania	73.99	84.52	84.12	80.01	57.85	65.19	75.84	73.61	78.77	43.18	...	4.90	4.73	4.62	4.42	75.41	71.77	71.65	70.
Algeria	51.73	15.48	27.65	69.66	57.87	42.27	20.22	21.53	66.40	38.89	...	4.32	3.71	4.46	4.22	69.92	66.83	66.83	65.
Argentina	82.79	85.71	84.71	91.00	72.65	73.35	62.36	75.00	85.79	59.27	...	6.68	6.40	6.62	6.38	80.66	74.98	74.84	74.
Armenia	71.36	75.00	70.00	78.72	65.36	59.44	56.74	56.25	75.47	46.60	...	5.56	5.32	5.64	5.55	76.46	70.87	70.53	69.
Australia	94.32	96.43	94.71	100.00	88.25	90.53	98.31	97.22	100.00	74.36	...	8.08	8.29	8.23	8.14	91.29	88.32	88.23	87.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
United States of America	92.97	100.00	94.71	92.39	91.82	87.69	98.31	98.61	88.83	75.64	...	8.13	8.19	7.78	7.75	85.71	84.78	85.27	86.
Uruguay	85.00	85.71	84.12	85.14	85.74	78.58	91.57	89.89	77.19	69.67	...	6.75	6.70	7.05	6.80	82.99	79.40	NaN	N.
Viet Nam	66.67	70.24	65.29	67.79	66.94	59.31	69.10	73.61	65.43	38.90	...	4.18	4.28	4.48	4.39	68.85	NaN	NaN	N.
Zambia	42.42	30.95	25.88	67.45	33.94	41.11	39.89	47.92	56.89	18.53	...	2.19	2.04	2.68	2.63	55.34	NaN	NaN	N.
Zimbabwe	50.19	45.24	52.35	61.35	36.88	36.92	27.53	32.64	56.68	21.44	...	2.85	2.90	3.12	2.99	52.26	45.26	43.76	42.

Рис. 2.1 – Завантажена вибірка

Так як цільова змінна категоріальна і представлена у вигляді рядка. Потрібно провести кодування для того, щоб перевести цю змінну у числовий формат. Для цього використовувалась спеціальна функція із бібліотеки scikit-learn для кодування категоріальних змінних. В результаті якої цільова змінна була переведена у формат числової з діапазоном значень від 0 до 3.

Наступним кроком було визначити кореляцію змінних стану. Було побудовано дві матриці кореляції: загальна та зі збільшення найбільш корельованих змінних стану. Вони зображені на рис. 2.2 та 2.3 відповідно.

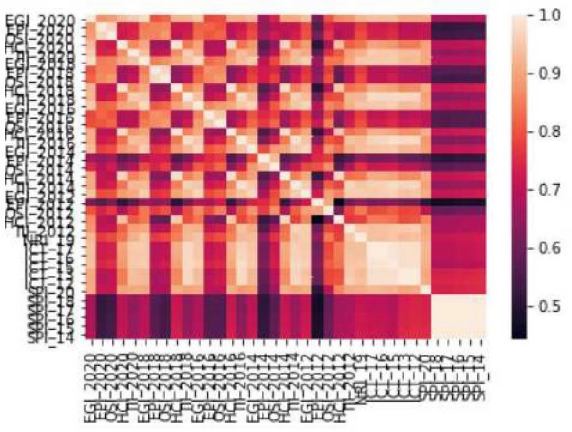

Рис. 2.2 – Загальна кореляція змінних стану

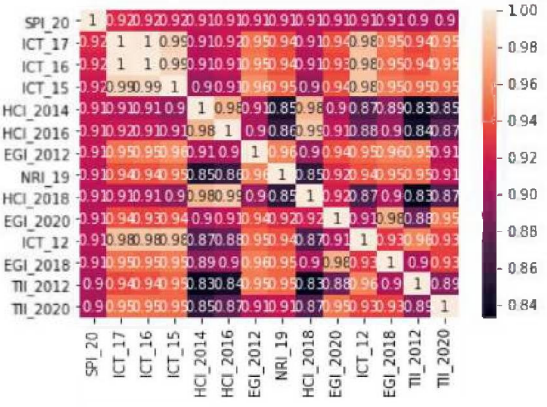

Рис. 2.3 – Кореляція змінних стану зі збільшенням найбільш корельованих

Не важко бачити, що усі змінні стану достатньо сильно корелюють між собою та гарно підходять для використання їх у якості вхідних даних для методу кластеризації.

Якщо подивитися на рис. 2.1, то можна побачити, що у рядках присутні значення «NaN», що вказують на наявність пропущених значень у вибірці. Тому була побудована таблиця, яка відображає в яких змінних стану є пропущені значення та їх відсоток до всіх значень. Таблиця представлена на рис. 2.4.

	Total	Percent
SPI_14	13	11.304348
SPI_15	13	11.304348
SPI_16	13	11.304348
SPI_17	13	11.304348
SPI_18	8	6.956522
ICT_12	1	0.869565
ICT_13	1	0.869565

Рис. 2.4 – Таблиця пропущених значень

Була висунута гіпотеза, що пропущенні значення для змінних стану ICT_12 та ICT_13 знаходяться в одному рядку. Перевіривши гіпотезу, було виявлено, що насправді ці пропущені значення були в одному рядку, тому було прийняте рішення заповнити ці значення використовуючи методи машинного

навчання. В якості методів для заповнення було обрано випадковий ліс та лінійну регресію.

Для тренування було видалені змінні стану з пропущеними значеннями, за винятком ICT_12 та ICT_13. Ці змінні стану використовувались, як цільові та в них був видалений лише рядок з пропущеними значеннями. На цих даних були натреновані моделі випадкового лісу та лінійної регресії. Якість цих моделей було перевірено за допомогою середньої квадратичної похибки (mean squared error) та виявлено, що модель лінійної регресії має кращі показники. Тому для заповнення пропущених значень було обрана модель лінійної регресії.

Пропущені дані у змінних стану ICT_12 та ICT_13 були заповнені, а інші змінні стану з пропущеними значеннями були видалені з вибірки. В наслідок цього була отримана вибірка без пропущених значень.

Наступним кроком була перевірка розподілів змінних стану та трансформація їх для приведення до нормального розподілу. Для перевірки нормальності розподілу була використана функція, яка знаходить скошеність (перекіс) для кожної змінної стану та побудовані ящики з вусами. Для даних перед обробкою на цьому кроці були побудовані ящики з вусами, які зображені на рис. 2.5.

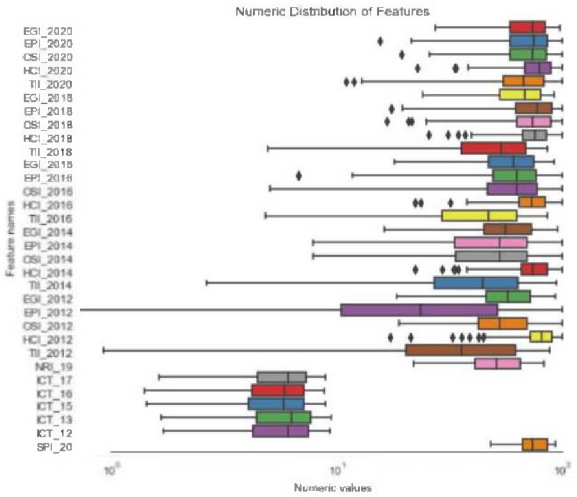

Рис. 2.5 – Ящики з вусами для змінних стану

Була визначена скошеність більш ніж на 0.5 для змінної стану EPI_2012 та застосована трансформація Бокса-Кокса, яка знизила перекіс до -0.05, що визначається, як майже нормальний розподіл.

Також після цього кроку були побудовані гістограми для всіх змінних стану. Вони представлені на рис 2.6.

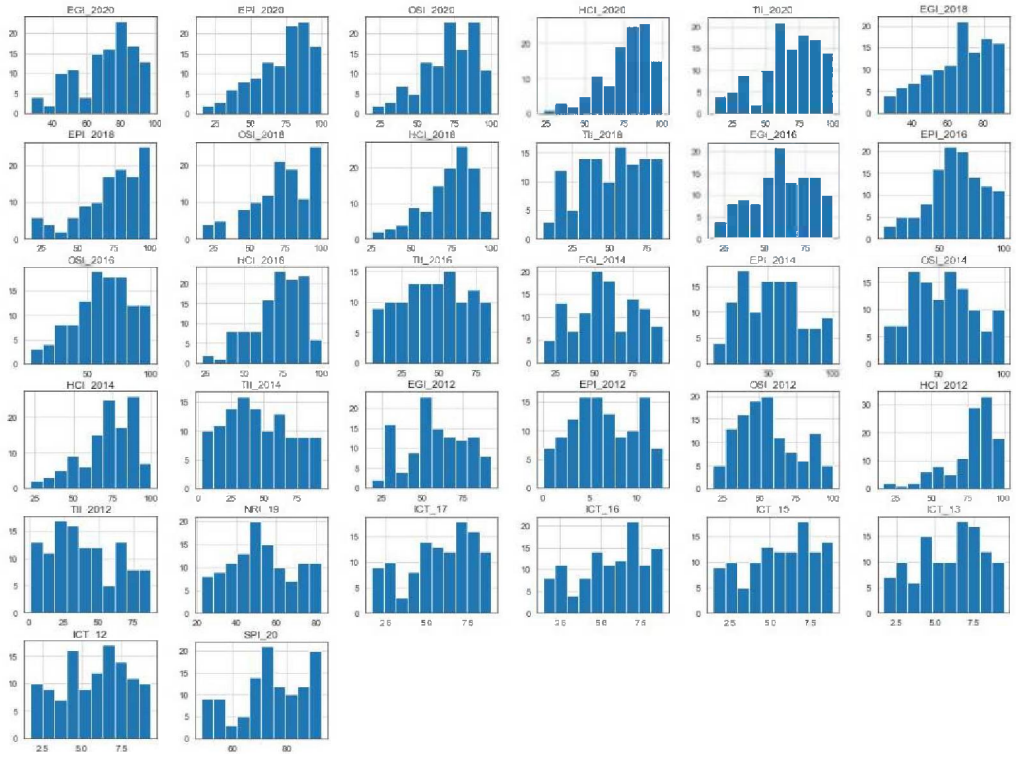

Рис. 2.6 – Гістограми змінних стану

На гістограмах можна бачити, що майже всі дані мають нормальний розподіл на скільки це можливо при використанні трансформацій.

Наступним кроком являється стандартизація та нормалізація вхідної вибірки. Для візуалізації результатів стандартизації та нормалізації даних в даній роботі будуть використовуватися два метода зменшення розмірності, такі як PCA (Principal Component Analysis) та T-SNE (t-distributed Stochastic Neighbor Embedding). Зменшення розмірності буде проведено до розмірності два задля

того, щоб можна було відобразити результати на двовимірному графіку. Перед тим, як переходити до наступного шагу потрібно більш детально розглянути представлені методи зменшення розмірності:

- PCA [35, 36] (Principal Component Analysis), він же метод головних компонент є основою багатоваріантного аналізу даних на основі проекційних методів. Найважливішим використанням PCA є представлення багатоваріантної таблиці даних у вигляді меншого набору змінних (сумарних індексів) для спостереження за тенденціями, стрибками, кластерами та викидами. Цей огляд може виявити зв'язки між спостереженнями та змінними, а також між змінними.

  PCA сходить до Коші, але вперше був сформульований у статистиці Пірсоном, який описав аналіз як пошук «ліній і площин, найбільш близьких до систем точок у просторі» [37].

  PCA є дуже гнучким інструментом і дозволяє аналізувати набори даних, які можуть містити, наприклад, мультиколінеарність, відсутні значення, категоріальні дані та неточні вимірювання. Мета полягає в тому, щоб витягти важливу інформацію з даних і виразити цю інформацію у вигляді набору підсумкових індексів, які називаються головними компонентами.

  Статистично PCA знаходить лінії, площини та гіперплощини в K-вимірному просторі, які якомога краще апроксимують дані в сенсі найменших квадратів. Лінія або площина, яка є апроксимацією найменших квадратів набору точок даних, робить дисперсію координат на прямій або площині якомога більшою.
- T-SNE [38] (t-Distributed Stochastic Neighbor Embedding) є нелінійною технікою з навчанням без вчителя, яка в основному використовується для дослідження даних та візуалізації даних великої розмірності. t-SNE дає

можливість перевірити, як дані впорядковані у просторі високої вимірності.

Алгоритм t-SNE обчислює міру подібності між парами екземплярів у просторі високої вимірності та в просторі з низькою вимірністю. Потім він намагається оптимізувати ці дві міри подібності за допомогою функції вартості.

t-SNE можна використовувати для високорозмірних даних, а потім вихідні дані цих розмірів стануть вхідними даними для іншої моделі класифікації.

Крім того, t-SNE можна використовувати для дослідження, вивчення або оцінки сегментації. t-SNE часто може показувати чітке розділення в даних. Це можна використовувати до використання моделі сегментації для вибору номера кластера або після, щоб оцінити, чи дійсно ваші сегменти витримують. Однак t-SNE не є підходом до кластеризації, оскільки він не зберігає вхідні дані і значення можуть часто змінюватися між запусками, тому він призначений виключно для дослідження.

Розглянувши методи зменшення розмірності, можна переходити до наступного кроку – стандартизації даних. В цій роботі використовувалися три варіанта стандартизації даних за допомогою бібліотеки scikit-learn:

1. Стандартизація з використанням надійного скалеру (Robust Scaler). Він дає змогу масштабувати об’єкти за допомогою статистичних даних, стійких до викидів. Цей скалер видаляє медіану та масштабує дані відповідно до квантильного діапазону (за замовчуванням IQR: міжквартильний діапазон). IQR – це діапазон між 1-м квартилем (25-й квантиль) і 3-м квартилем (75-й квантиль).

Приведемо на рис. 2.7 короткі характеристики про дані, такі як середнє, дисперсію, мінімум, максимум та квантилі.

	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018	OSI_2018	HCI_2018	TII_2018
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	-1.642355	-1.074476	-0.821370	-0.452619	-2.561217	-1.823519	-4.634957	-0.697280	-0.699022	-1.448592
std	8.522131	5.966719	4.950490	2.611162	39.326203	10.682450	21.626518	8.492294	5.129921	11.198990
min	-23.024463	-17.549757	-14.675018	-9.282658	-100.214823	-27.143563	-59.550000	-24.013158	-16.257528	-25.341712
25%	-7.637756	-4.915229	-3.988720	-1.795012	-22.153598	-9.497776	-15.170000	-4.539895	-2.881972	-9.350615
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	4.788732	3.157895	3.110171	1.679767	27.756892	7.199778	13.200000	6.296390	3.094724	8.208955
max	11.658018	7.371370	7.177527	3.528670	58.807734	14.405733	23.030000	11.129386	8.100943	16.894475

Рис. 2.7 – Короткі характеристики вибірки після використання надійного скалеру

Як було зазначено раніше, медіана була видалена і це зазначено на рис. 2.6 в рядку «50%». Також побудуємо ящики з вусами для кожної змінної стану. Представимо їх на рис. 2.8.

Для повного розуміння вигляду змінних стану після трансформації, були використані методи зменшення розмірності та побудовані графіки на яких відображена вибірка у двовимірному просторі. Графіки з використанням PCA та T-SNE відображені на рисунках 2.9 і 2.10 відповідно.

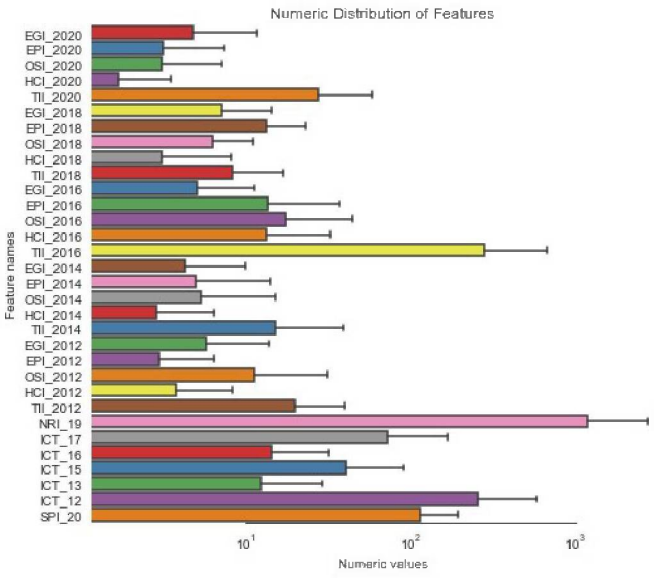

Рис. 2.8 – Ящики з вусами після використання надійного скалеру

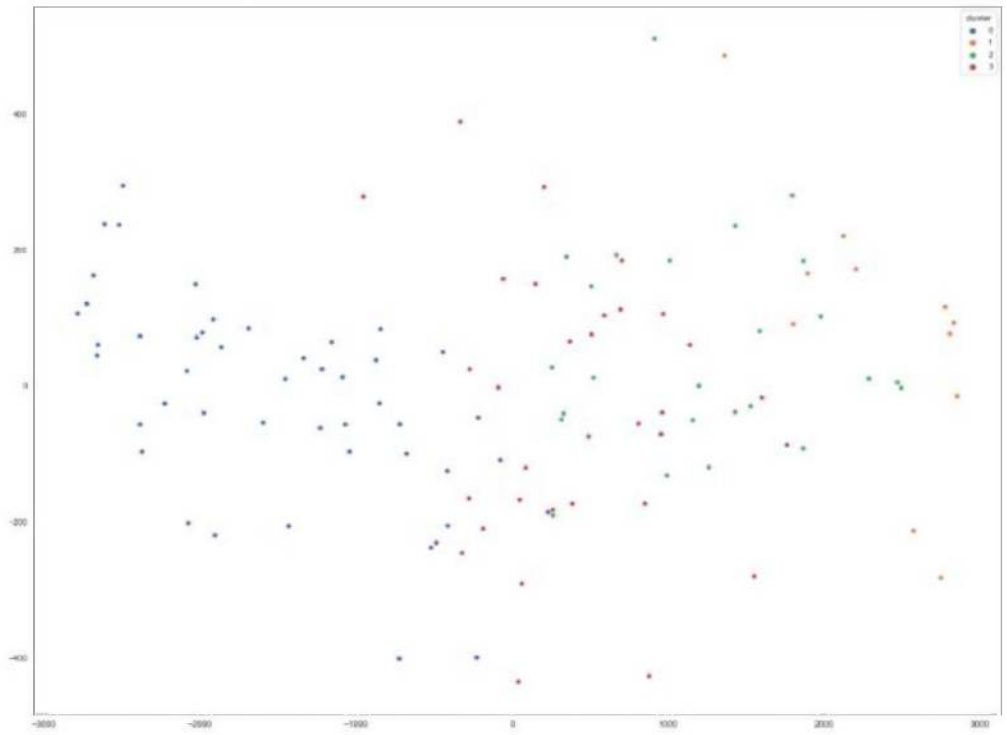

Рис. 2.9 – Відображення вибірки за допомогою PCA

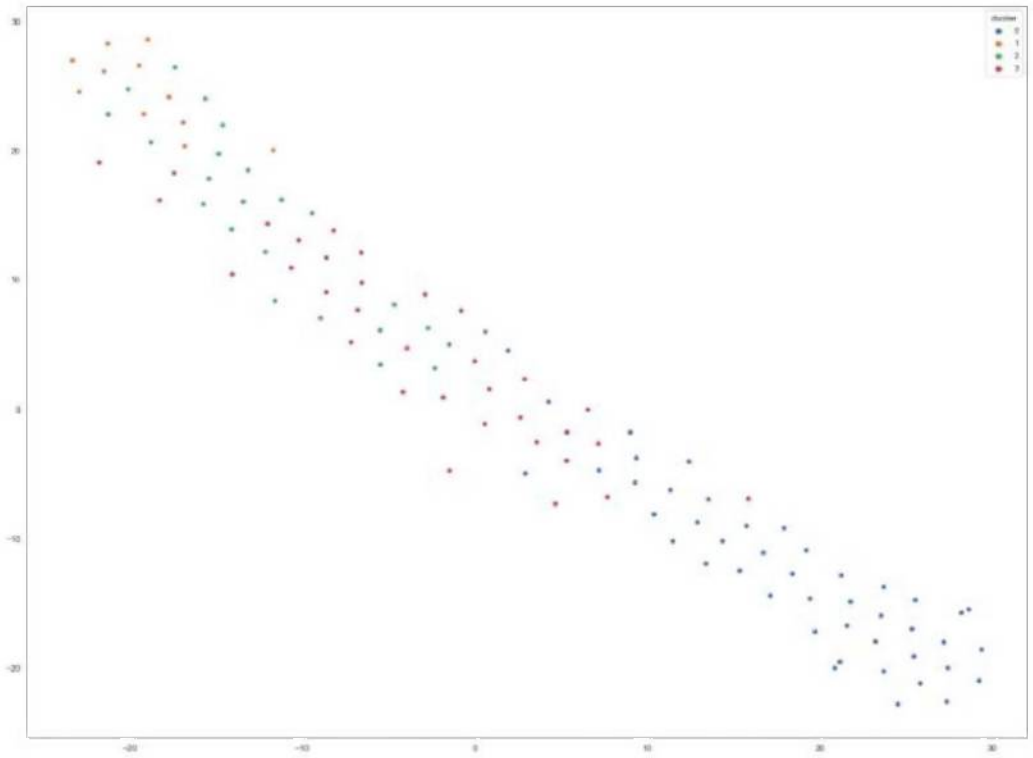

Рис. 2.10 – Відображення вибірки за допомогою T-SNE

На даних графіках різними кольорами точок відображена приналежність точки до певного типу кластера.

2. Стандартизація з використанням стандартного скалеру (Standart Scaler). Він стандартизує ознаки, видаляючи середнє значення та масштабуючи одиницю дисперсії. Стандартизація вибірки x розраховується як:

$$z = \frac{x-u}{s}, \tag{2.1}$$

де u – це середнє значення для навчальних вибірок;

s - стандартне відхилення навчальних вибірок.

Приведемо коротку інформацію, графіки ящиків з вусами та двовимірні графіки стандартизованої вибірки, як було зроблено для попереднього методу стандартизації. Коротка інформація про стандартизовані дані, графіки ящиків з вусами, двовимірний графік з використанням PCA та двовимірний графік з використанням T-SNE зображено на рисунках 2.11-2.14 відповідно.

	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018
count	1.150000e+02	1.150000e+02	1.150000e+02	1.150000e+02	1.150000e-02	1.150000e-02	1.150000e+02
mean	7.839140e-16	-3.050700e-16	-4.827057e-16	-2.187622e-15	6.101400e-16	2.239754e-16	-9.654113e-17
std	1.004376e+00	1.004376e+00	1.004376e+00	1.004376e+00	1.004376e-00	1.004376e-00	1.004376e+00
min	-2.519990e+00	-2.773280e+00	-2.810687e+00	-3.396450e+00	-2.494036e-00	-2.380620e-00	-2.550358e+00
25%	-7.065886e-01	-6.465131e-01	-6.426054e-01	-5.163477e-01	-5.003820e-01	-7.215426e-01	-4.892673e-01
50%	1.935598e-01	1.808664e-01	1.666430e-01	1.740987e-01	6.541251e-02	1.714494e-01	2.152561e-01
75%	7.579363e-01	7.124341e-01	7.976478e-01	8.202162e-01	7.743131e-01	8.483809e-01	8.282891e-01
max	1.567517e+00	1.421687e+00	1.622850e+00	1.531392e+00	1.567340e-00	1.525893e-00	1.284813e+00

Рис. 2.11 – Короткі характеристики вибірки після використання стандартного скалеру

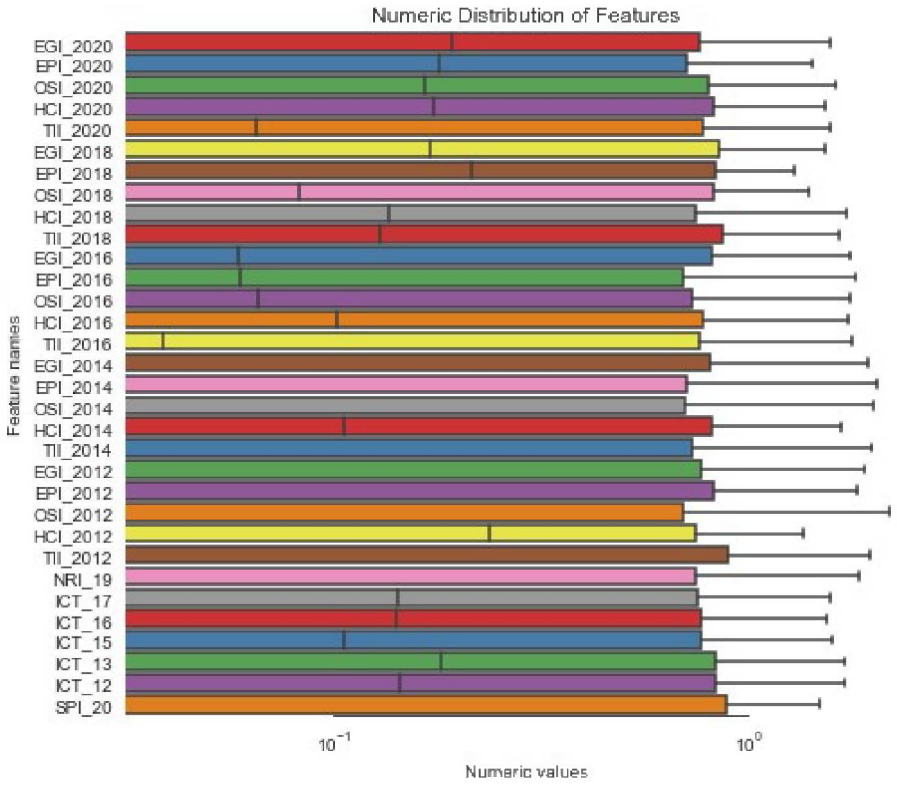

Рис. 2.12 – Ящики з вусами після використання стандартного скалеру

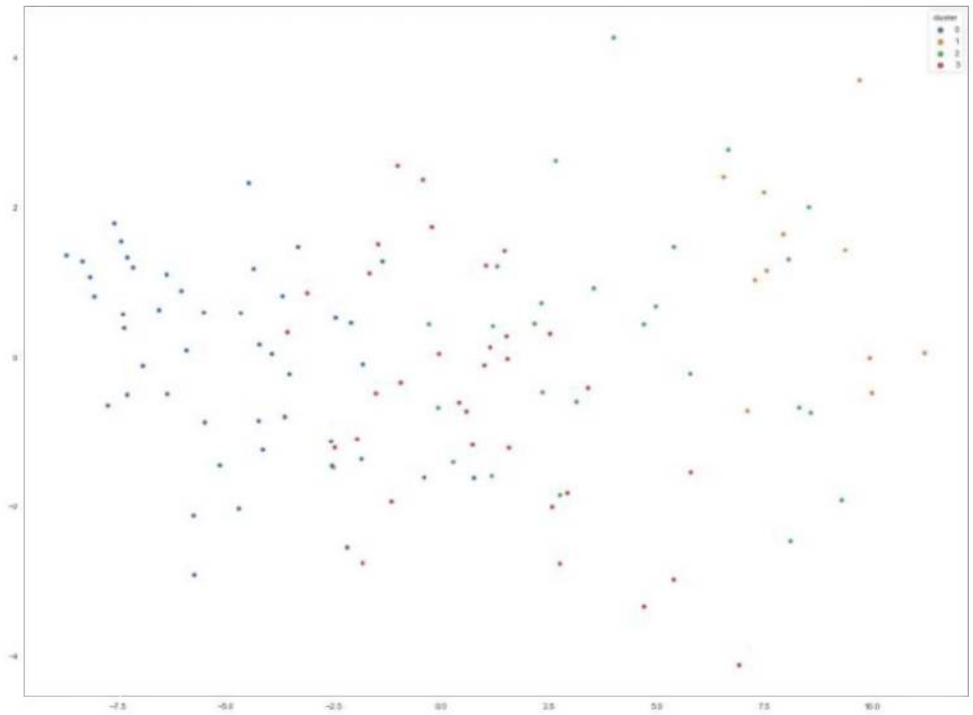

Рис. 2.13 – Відображення вибірки за допомогою PCA

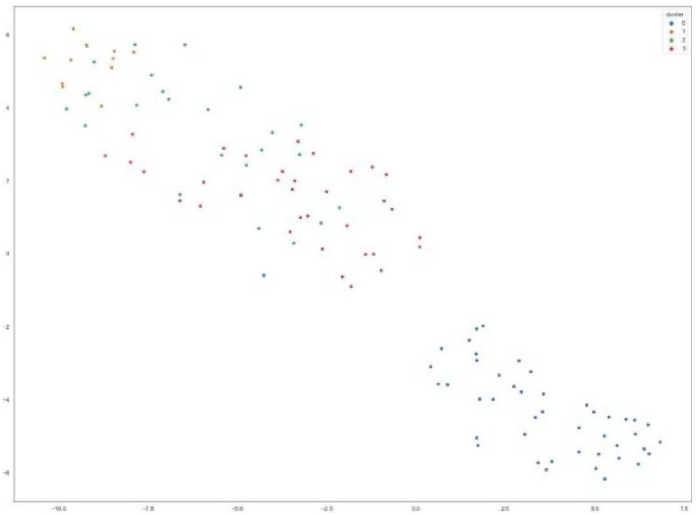

Рис. 2.14 – Відображення вибірки за допомогою T-SNE

3. Стандартизація з використанням мінімум-максимум скалера (MinMaxScaler). Він масштабує та перекладає кожну ознаку окремо так, щоб вона знаходилась у заданому діапазоні на навчальному наборі, наприклад між нулем і одиницею.

В даній роботі проводиться масштабування до діапазонів від 0 до 1 та від -3 до 3. Масштабування від -3 до 3 показало гірші результати ніж інше, тому приведемо результати масштабування тільки для діапазону від 0 до 1. Приведемо короткі характеристики даних, графіки ящиків з вусами та двовимірні графіки масштабованого набору даних.

	EGI_2020	EPI_2020	OSI_2020	HCI_2020	TII_2020	EGI_2018	EPI_2018	OSI_2018	HCI_2018
count	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000	115.000000
mean	0.616510	0.661097	0.633960	0.689237	0.614087	0.609398	0.664992	0.663466	0.638731
std	0.243719	0.239424	0.226541	0.203817	0.247300	0.257103	0.261886	0.241653	0.210601
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.443645	0.506981	0.489018	0.584455	0.490881	0.424695	0.537418	0.554122	0.549113
50%	0.663864	0.704212	0.671547	0.724566	0.630193	0.653286	0.721119	0.683307	0.667428
75%	0.801938	0.830928	0.813873	0.855682	0.804739	0.826569	0.880964	0.862474	0.794477
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Рис. 2.15 - Короткі характеристики вибірки після використання мінімум-максимум скалеру

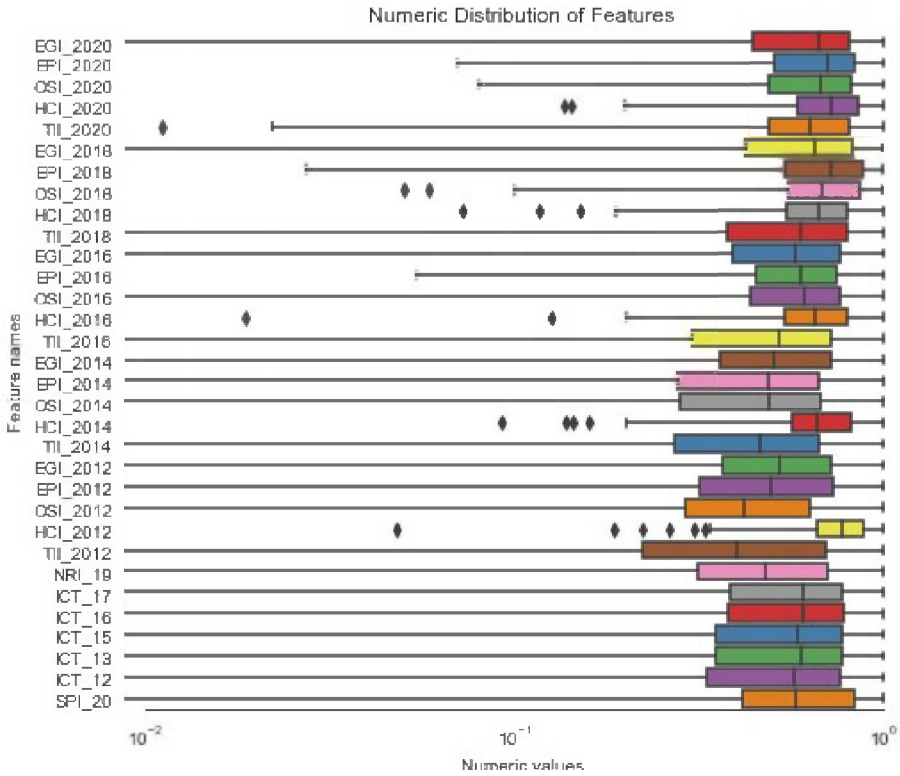

Рис. 2.16 – Ящики з вусами після використання мінімум-максимум скалеру

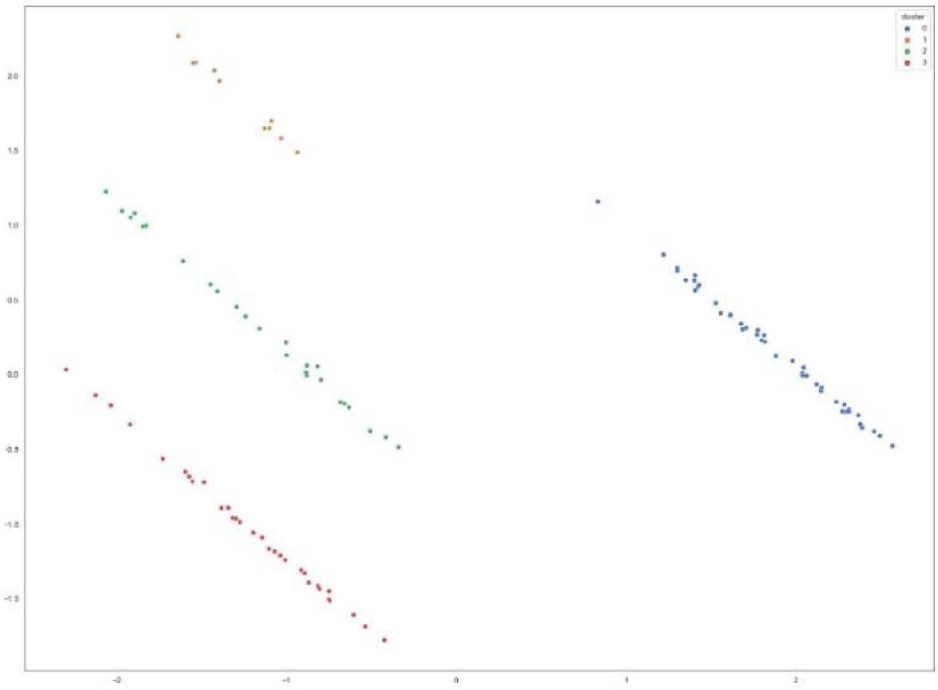

Рис. 2.17 – Відображення вибірки за допомогою PCA

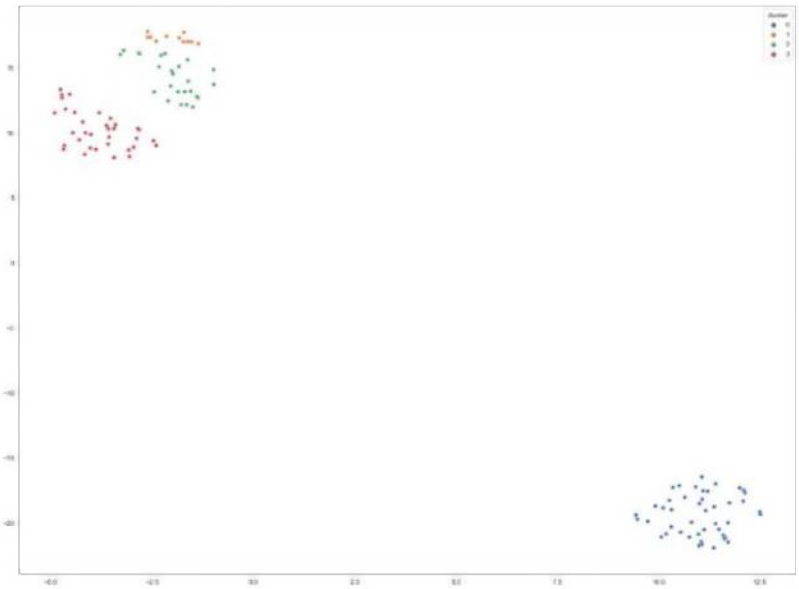

Рис. 2.18 – Відображення вибірки за допомогою T-SNE

В результаті стандартизації вибірки було отримано чотири вибірки, такі як:

- Вибірка, яка стандартизована за допомогою надійного скалера;
- Вибірка, яка стандартизована за допомогою стандартного скалера;
- Вибірка, яка стандартизована за допомогою мінімум-максимум скалера в діапазоні від -3 до 3;
- Вибірка, яка стандартизована за допомогою мінімум-максимум скалера в діапазоні від 0 до 1.

Як видно із графіків ящиків з вусами в деяких змінних стану є наявність викидів (аномальних значень). Для задачі кластеризації є дуже важливим, щоб у вибірці буле мінімальна кількість викидів або їх відсутність. Це пояснюється тим, що викиди можуть дуже сильно змінювати центри кластерів, тим самим погіршуючи якість кластеризації. Тому для фільтрації вибірок від викидів були застосовані наступні методи:

- IQR (interquartile range) метод. Цей метод використовується для визначення викидів наступним чином. Спершу розраховується міжквартильний

діапазон для даних, після цього цей показник перемножується на константне значення 1.5 (константа, яка використовується для розпізнавання викидів), яке будемо позначати як «Т». Потім будується діапазон значень для визначення викидів наступним чином. Нижня границя діапазону розраховується, як перший квартиль – Т. Тоді як верхня границя – третій квартиль + Т. Числа які виходять за межі цього діапазону вважаються викидами.

- Ізоляційний ліс (Isolation forest). Він являється ефективним методом для виявлення викидів (аномалій) в наборах даних. Цей метод заснований на бінарних деревах рішень. Основний принцип ізоляційного лісу полягає в тому, що викиди є нечисленними і далекими від решти спостережень. Щоб побудувати дерево, алгоритм випадковим чином вибирає об'єкт із простору ознак і випадкове розділене значення в діапазоні між максимумами та мінімумами.

Після застосування цих методів виявлення викидів, були побудовані двовимірні графіки вибірок з використанням PCA та T-SNE. Ці графіки були проаналізовані експертом та визначені реальні викиди в вибірках. Реальні викиди були видалені з вибірок.

2.3 Метод кластеризації даних на основі інформаційних критеріїв та агентно-орієнтованого підходу в системах моніторингу соціально-економічного розвитку країн

Розглянемо базовий алгоритм методу нечіткої кластеризації с-середніх [39]. Метод нечіткої кластеризації с-середніх дозволяє виробляти нечітку класифікацію (розбивку) елементів вихідної множини потужністю N на задане число підмножин – «класів» [40]. Метод нечіткої кластеризації с-середніх відрізняється від його попередника методу k-середніх методом визначення приналежності об'єкта до можливого класу.

Алгоритм с-середніх [41] включає послідовність операцій:

1. На першому етапі задаються випадковим чином k центрів кластерів c_j, j = 1, ..., k;
2. Далі розраховується матриця приналежності елементів до кластерів w_{ij}. На цьому етапі є різні варіації способу визначення відстані між можливим центром і елементом підмножини. При нормального розподілу даних, матриці приналежності буде виглядати наступним чином:

$$w_{ij} = \frac{\aleph(d(x_i, c_j) \mid \mu = 0, \sigma_j)}{\sum_{i=1}^{P_i} \aleph(d(x_i, c_j) \mid \mu = 0, c_j)}, \tag{2.2}$$

де x_i – і-й елемент множини, i = 1, ..., P_j;

c_j – центр j-ого кластера;

$d(x_i, c_j)$ – відстань між точками x_i і c_j;

$\aleph(d(x_i, c_j) \mid \mu = 0, \sigma_j)$ – щільність ймовірності нормального розподілу в точці $d(x_i, c_j)$.

3. Наступним кроком необхідно перемістити центри кластерів, враховуючи матрицю приналежності:

$$c_j \leftarrow \frac{\sum_{i=1}^{P_j} w_{ij} x_i}{\sum_{i=1}^{P_j} w_{ij}}, \tag{2.3}$$

4. Потім необхідно розрахувати функцію втрат (на цьому етапі можна скористатися принципом максимальної правдоподібності). У разі нормального розподілу функція втрат дорівнює:

$$Loss\_function = \sum_{j=1}^{k} \sum_{i=1}^{P_j} d(x_i c_j)^2 w_{ij} \to \min \tag{2.4}$$

5. У разі якщо значення функції втрат не зменшилось, треба перейти до пункту 2. Так само тут може бути функція зупинки, відповідно до якої вихід може наступити раніше, при досягненні якоїсь умови (наприклад, досягнення певної кількості ітерацій).

Метод нечіткої кластеризації c-середніх так само має ряд істотних недоліків обмежують його застосування, наприклад якщо цільовий кластер має більш складну форму ніж просто m-мірна сфера, або у разі перетину кластерів і високою шуму даних, то метод нечіткої кластеризації не здатний якісно визначати приналежність елементів до того, чи іншого кластеру [42, 43].

В роботі [39] було запропоновано використовувати нову функцію втрат для оцінки якості кластеризації даних. А саме було запропоновано замінити формулу 2.4 на формулу 2.5.

$$H(X,M) = -\frac{1}{J\sum_{j=1}^{J} P_j} \sum_{j=1}^{J} \{P(M_j^{(t+1)}) \sum_{i=1}^{P_j} D_{KL}(x_i, M_j^{(t+1)})\} \to \min \tag{2.5}$$

Впровадження нової функції втрат дозволило усунути такі недоліки, як адаптацію методу до широкого розкиду відхилень змінних стану і кластерам зі

складною m-мірну формою. Також це дозволило методу врахувати інформативність змінних стану елементів, що підвищило якість кластеризації.

В даній роботі пропонується покращити якість методу [39] за рахунок впровадження агентно-орієнтованого підходу навчання. В науці таких підхід ще називається, як навчання з підкріпленням (reinforcement learning).

Навчання з підкріпленням — це навчання моделей машинного навчання приймати послідовність рішень. Агент вчиться досягати мети в невизначеному, потенційно складному середовищі. У навчанні з підкріпленням штучний інтелект стикається з ігровою ситуацією. Комп'ютер використовує метод проб і помилок, щоб знайти рішення проблеми. Щоб змусити машину робити те, що потрібно, штучний інтелект отримує або винагороди, або штрафи за дії, які він виконує. Його мета — максимізувати загальну винагороду.

В такому навчанні встановлюється політика винагород, тобто правила гри, але жодних підказок чи пропозицій щодо вирішення цієї гри не дається. Модель має з'ясувати, як виконати завдання, щоб максимізувати винагороду, починаючи від абсолютно випадкових випробувань і закінчуючи складною тактикою та надлюдськими навичками. Використовуючи можливості пошуку та багатьох випробувань, навчання з підкріпленням наразі є найефективнішим способом впроваджувати штучний інтелект машини.

Розглянемо новий метод кластеризації на основі агенто-орієнтованого підходу. Нехай дана вибірка даних $X = \{X_j\}$; $X_j = \{x_{jp}^0\}$. Тоді загальна кількість елементів у вибірці N буде розраховуватися за формулою 2.6.

$$N = \sum_{j=1}^{J^*} P_j^*, \tag{2.6}$$

де P_j^* кількість елементів в j-ому кластері;

J^* - кількість кластерів.

Визначимо міри внутрішньо-кластерної відстані:

$$d\left(x_{jp}^{0}, c_{j}^{0}\right) = \begin{cases} \sum_{m=1}^{M} \rho(x_m \mid f_i) \log 2[\frac{\rho(x_m \mid f_i)}{\rho(x_m)}] \\ -\sum_{x \in X} p(x) \log\left(q(x)\right) \end{cases} \quad (2.7)$$

У формулі 2.7, перший вираз позначає відносну ентропію Кульбака-Лейблера, а другий – крос-етропію.

Тоді середня міра внутрішньо-кластерної відстані:

$$M\left(c_{j}^{0}\right) = \frac{1}{P_j} \sum_{p=1}^{P_j} d(x_{jp}^{0}, c_{j}^{0}) \quad (2.8)$$

Відповідно функція втрат:

$$LF(X) = \frac{1}{J_t} \sum_{j=1}^{J_t} M(c_{j}^{0}) \quad (2.9)$$

Далі слід визначити додаткові вирази для розрахунку розподілу Коші, матриці приладдя з використанням розподілу Коші та вираз для корекції центрів кластерів. Ці вирази позначені на формулах 2.10 – 2.12 відповідно.

$$\rho\left(x_{jp}^{0}, c_{j}^{0}\right) = \frac{1}{\pi\eta\left[1+\frac{d\left(x_{jp}^{0}, c_{j}^{0}\right)}{\eta^2}\right]} \quad (2.10)$$

$$w_{jp} = \frac{\rho\left(x_{jp}^{0}, c_{j}^{0}\right)}{\sum_{p=1}^{P_j} \rho\left(x_{jp}^{0}, c_{j}^{0}\right)} \quad (2.11)$$

$$c_j^0 = \frac{\sum_{p=1}^{P_j} w_{jp} x_{jp}^0}{\sum_{P=1}^{P_j} w_{jp}} \quad (2.12)$$

Після того, як було визначено всі необхідні вирази та проведений опис алгоритму, визначимо алгоритм методу кластеризації на основі агенто-орієнтованого підходу:

1. Визначаємо: $J_t^{(n)} > J^*$; $P_t^{(n)} = int(\frac{N}{J_t^{(n)}})$; здійснюємо генерацію центрів кластерів випадковим чином $\{c_j^0\}$;
2. Використовуючи обрану міру внутрішньо-кластерної відстані $d_1(x_{jp}^0, c_j^0)$ вибираємо $P_t^{(n)}$ найближчих сусідів для кожного j-ого кластера;
3. Для кожного j-ого кластера, використовуючи $\{w_{jp}\}$ і $\{\rho(x_{jp}^0 | M_j)\}$, коригуємо центри кластерів $\{c_j^0\}$;
4. Для кожного j-ого кластера, використовуючи обрану міру внутрішньо-кластерної відстані $d(x_{jp}^0, c_j^0)$ обираємо $P_t^{(n)}$ найближчих сусідів. Видаляємо елементи, які дублюються: $P_t^{(n)} \to P_j^{(n)}$;
5. Обчислюємо для кожного j-ого кластера середню міру внутрішньо-кластерної відстані $M(c_j^0)$, а далі $LF(X)$ – для всіх кластерів;
6. Елітний відбір. Знаходимо кластер із найбільшим значенням $M(c_j^0)$, і видаляємо його;
7. Обновляємо $J_t^{(n+1)} = J_t^{(n)} - 1$; $P_t^{(n+1)} = int(\frac{N}{J_t^{(n+1)}})$;
8. Переходимо до пункту 2, якщо $J_t^{(n+1)} > 1$.

Використовуючи описаний вище алгоритм виконується навчання методу кластеризації за допомогою агентно-орієнтованого підходу. Це один з

небагатьох методів кластеризації, який використовує агентно-орієнтований підхід навчання.

Принцип проб та помилок дозволяє більш точно визначити приналежність елементів з вибірки даних до певних кластерів та дозволяє працювати з високорозмірними наборами даних. Завдяки такому підходу виявляється покращення стійкості до викидів.

Даний метод являється абсолютно новим та перевершує інші методи кластеризації у задачі моніторингу та оцінки рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу. Порівняльні результати представлені у наступному розділі, де представлена порівняльна таблиця з іншими популярними методами кластеризації, а також приведено результати кластеризації з використанням вибірок із різними типами стандартизації даних та різними методами боротьби з викидами.

Висновки до розділу 2

В підрозділі 2.1 були сформульовані визначення та описано уявлення про вхідні, вихідні дані. Також були визначені вхідні дані для даної роботи та описані її змінні.

В підрозділі 2.2 була проведена підготовка та обробка вхідних даних для подальшої обробки цих даних методом кластеризації. Були використані такі методи обробки, як видалення непотрібних змінних стану, трансформація категоріальних змінних стану, стандартизація даних, трансформація змінних стану для приведення до нормального розподілу. Також були застосовані методи боротьби з пропущеними значеннями, викидами (аномальними значеннями) та методи зменшення розмірності для візуалізації всієї вибірки на двовимірному графіку.

В підрозділі 2.3 був описаний метод кластеризації на основі с-середніх та модифікації його за допомогою впровадження ентропії Кульбака-Лейблера, як функції втрат. Після цього був представлений новий метод та алгоритм навчання на базі агентно-орієнтованого підходу та визначені основні вирази для розрахунків.

РОЗДІЛ 3.
РОЗРОБКА ЗАСОБІВ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ КЛАСТЕРИЗАЦІЇ ДАНИХ

3.1 Розробка програмного забезпечення для методу кластеризації даних

В якості мови програмування на якій базується розробка програмного забезпечення для нового методу кластеризації даних є мова Python. Вона інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Її високо рівневі вбудовані структури даних у поєднанні з динамічним типізацією та динамічним зв'язуванням роблять цю мову дуже привабливою для швидкої розробки додатків, а також для використання в якості мови сценаріїв або склеювання для з'єднання існуючих компонентів. Простий, легкий у засвоєнні синтаксис Python підкреслює читабельність і, отже, знижує витрати на обслуговування програми. Python підтримує модулі та пакунки, що сприяє модульності програм і повторному використанню коду. Інтерпретатор Python і велика стандартна бібліотека доступні у вихідній або двійковій формі безкоштовно для всіх основних платформ і можуть вільно поширюватися.

Існує велика кількість бібліотек на базі мови Python, які можна з легкістю використовувати для розробки алгоритмів та математичних методів для машинного навчання. Такими бібліотеками є NumPy, SciPy, scikit-learn, Pandas та інші. Також існують дуже зручні бібліотеки для візуалізації результатів досліджень, такі як matplotlib, seaborn, plotly.

Було розроблено головний файл для запуску тренування, розрахунку якості методу кластеризації та візуалізації отриманих результатів. Назвою цього файлу є «main.py». Цей файл являється головним способом до запуску програмного забезпечення для вирішення задачі кластеризації даних. Він

безпосередньо імпортує усі необхідні файли та функції для реалізації методу кластеризації та задач описаних вище.

Після старту програмного забезпечення, дані про статус тренування виводяться у термінал, з якого це програмне забезпечення було запущено. В терміналі відображаються такі дані, як кількість кластерів, загальна втрата, втрати для кожного з кластерів та кількість видалених кластерів. Ці показники дозволяють слідкувати за виконанням тренування методу кластеризації даних. Приклад виводу у термінал відображено на рис. 3.1.

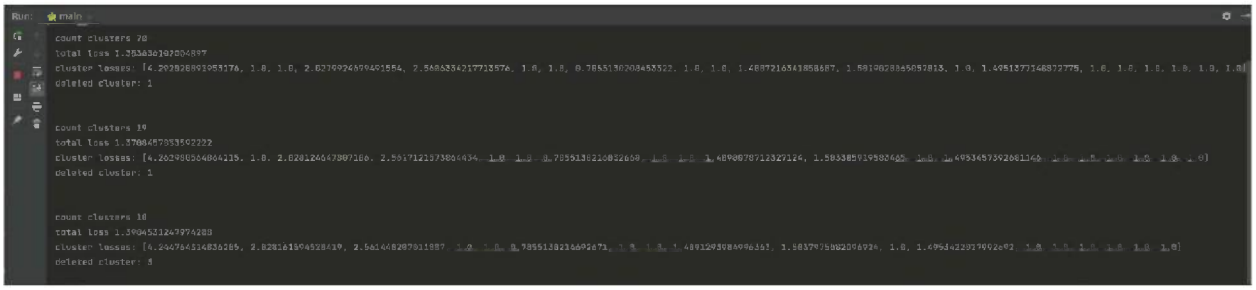

Рис. 3.1 – Приклад виводу даних про тренування у термінал

Для реалізації нового та ефективного алгоритму на основі інформаційних критеріїв та агентно-орієнтованого підходу було реалізовано безліч допоміжних функцій, які реалізують усю математичну частину для методу кластеризації даних. Для виконання обчислень була використана бібліотека NumPy та math. Так наприклад для розрахунку коваріаційної матриці використовуються функції «np_covariance_matrix» та інверсної – «np_inverse_covariance». Реалізації цих функцій приведена в лістингу Г.1.

Використовуючи мову Python не виникало труднощів у реалізації тих чи інших функцій, які потрібні для тренування та перевірки якості моделі кластеризації. Бібліотека scikit-learn дозволяє з легкості підраховувати оцінку якості методу кластеризації з використанням лише однієї стрічки коду, де викликається функція цієї бібліотеки та задаються потрібні параметри.

Для візуалізації результатів кластеризації, використовувалася бібліотека matplotlib за допомогою якої будувалися графіки кривої ROC, двовимірні та трьох вимірні графіки результатів кластеризації. Приведемо приклади результатів кластеризації, які видаються при закінченні навчання методу кластеризації даних. На рис. 3.2 зображено криву ROC, яка відображає загальну оцінку якості методу кластеризації та на рис 3.3 відображені криві ROC для кожного цільового класу.

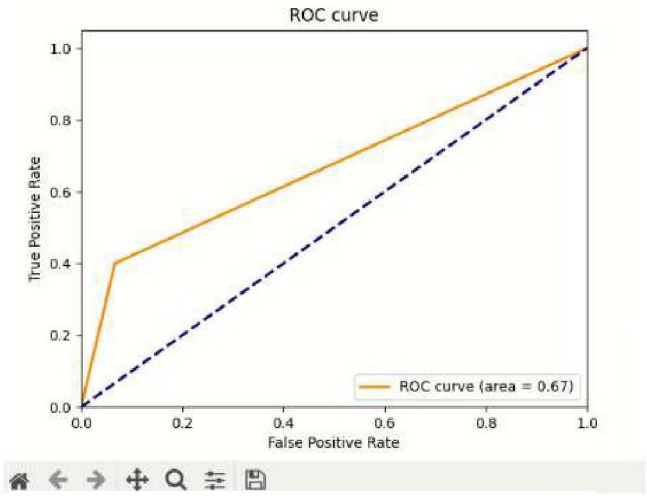

Рис. 3.2 – Приклад графіку кривої ROC

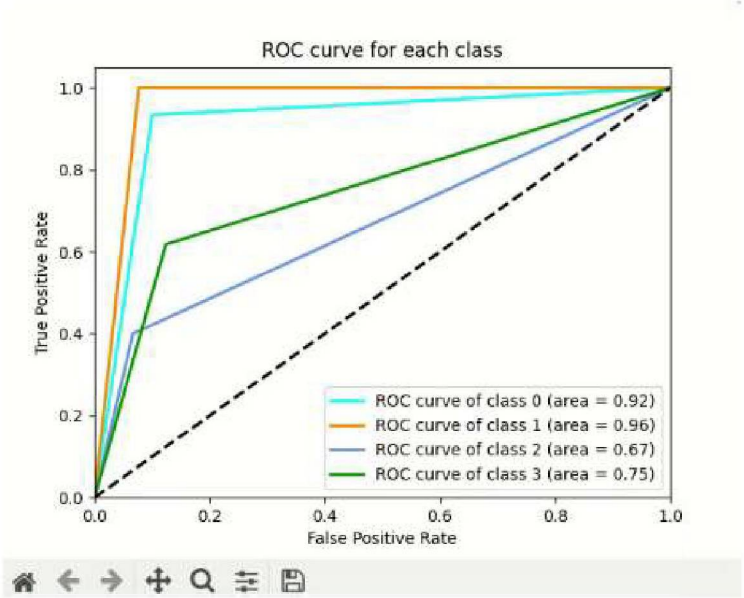

Рис. 3.3 – Приклад графіку кривої ROC для кожного цільового класу

Також приведемо приклади візуалізації результатів обчислень на двовимірних та трьох вимірних графіках. Ці графіки зображена на рисунках 3.4 і 3.5 відповідно.

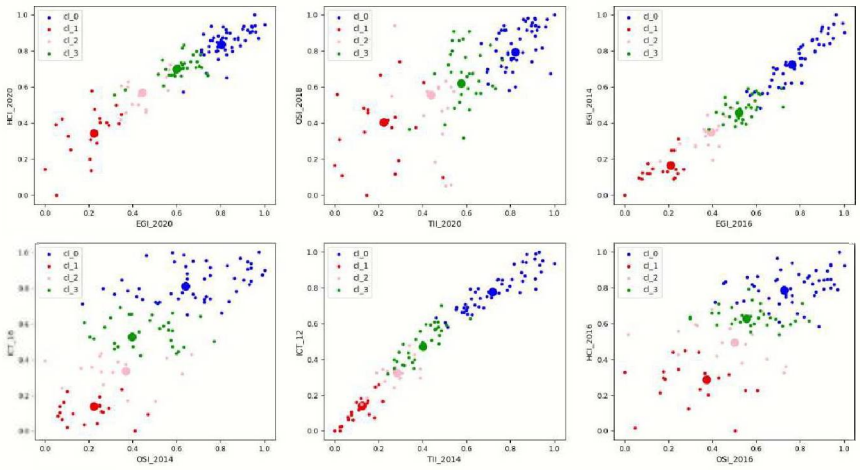

Рис. 3.4 – Приклад двовимірного графіку відображення результатів кластеризації

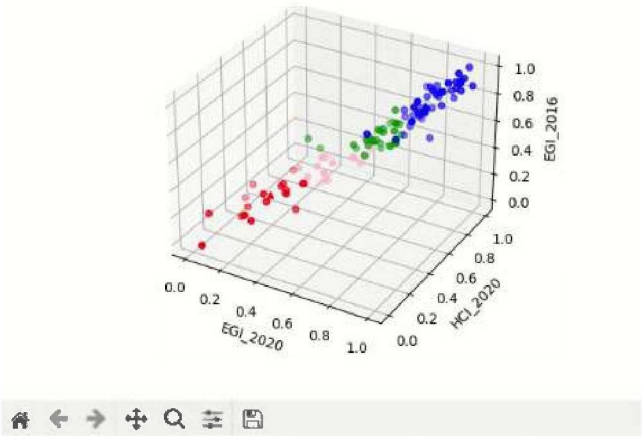

Рис. 3.5 – Приклад трьох вимірного графіку відображення результатів кластеризації

3.2 Порівняння ефективності методів кластеризації даних. Огляд та аналіз отриманих результатів.

Спочатку визначимо метрики для оцінювання якості методу кластеризації даних. Відповідно до підрозділу 2.1, вибірка містить цільову змінну «cluster», яка вказує на приналежність рядка з вибірки до певного кластеру. Тому для оцінки якості методів кластеризації будуть використовуватися метрики, які базуються на знанні про цільову змінну. Ці метрики описані у розділі 3. В даному розділі будуть використані наступні метрики: точність (accuracy) та ROC-AUC.

Для порівняння з новим методом кластеризації використаємо класичні методи кластеризації, такі як К-середніх, агломеративний метод та базовий с-середніх. В якості вхідних даних використаємо підготовлену та оброблену вибірку із мінімум-максимум скалером. Результати тестувань приведемо в таблиці 3.1.

Таблиця 3.1

Результати порівняння різних методів кластеризації

Метод кластеризації	*Точність*	*ROC-AUC*
К-середніх	0.924	0.942
Агломеративний метод	0.919	0.941
Базовий с-середніх	0.931	0.95
Новий метод	**0.957**	**0.974**

В таблиці 3.1 напівжирним виділені найкращі результати оцінки якості методів кластеризації. Не важко бачити, що новий метод кластеризації значно перевершує інші методи.

Так як при обробці та підготовці вхідних даних була отримана множина вибірок даних через використання різних методів стандартизації даних та

виявлення викидів. Тому в даному розділі буде відображена лише частина найкращих результатів.

Вибірка на якій новий метод кластеризації показав найкращий результат є вибірка з використанням мінімум-максимум скалера в діапазоні від 0 до 1. Було отримано 95.652 % точності та 0.9744 значення ROC-AUC. Результати розподілу кластерів зображено на рис. 3.6.

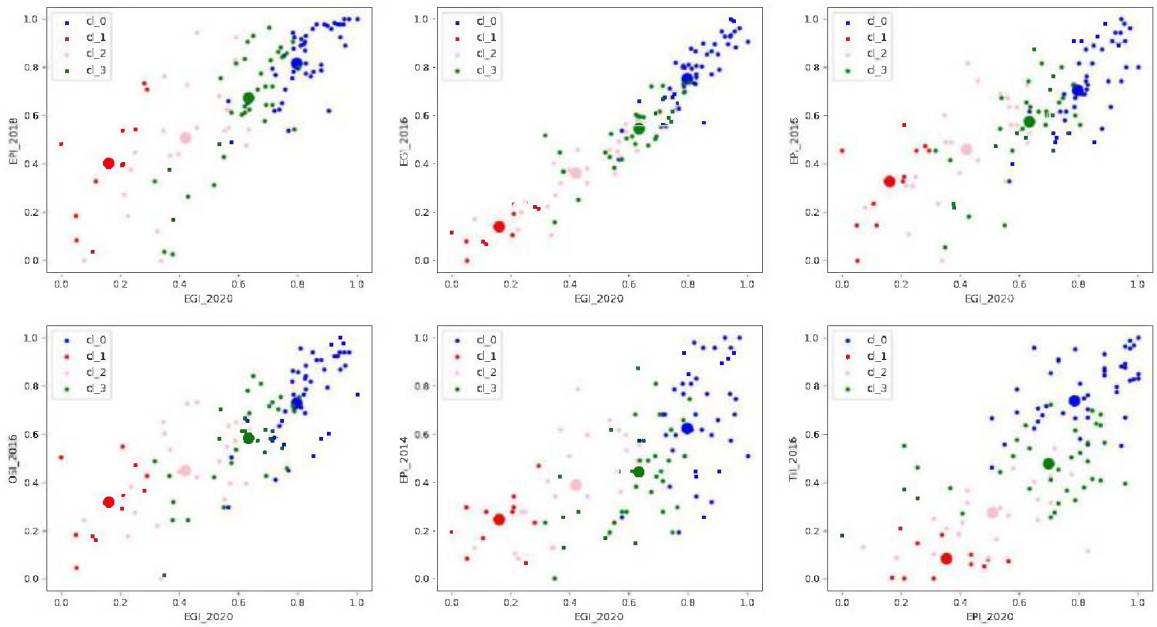

Рис. 3.6 – Результати розподілу кластерів у вибірці з використанням мінімум-максимум скалера в діапазоні від 0 до 1

Також була перевірена якість моделі побудованої на нестандартизованій вибірці з використанням метод боротьби з викидами. Ця модель кластеризації досягла 91.13% точності та 0.934 значення ROC-AUC. Результати розподілу кластерів зображено на рис. 3.7.

Ще одною вибіркою для тестування була вибірка даних з використанням надійного скалера та методів боротьби з викидами. Було отримано 91.16%

точності та 0.935 значення ROC-AUC. Результати розподілу кластерів зображено на рис. 3.8.

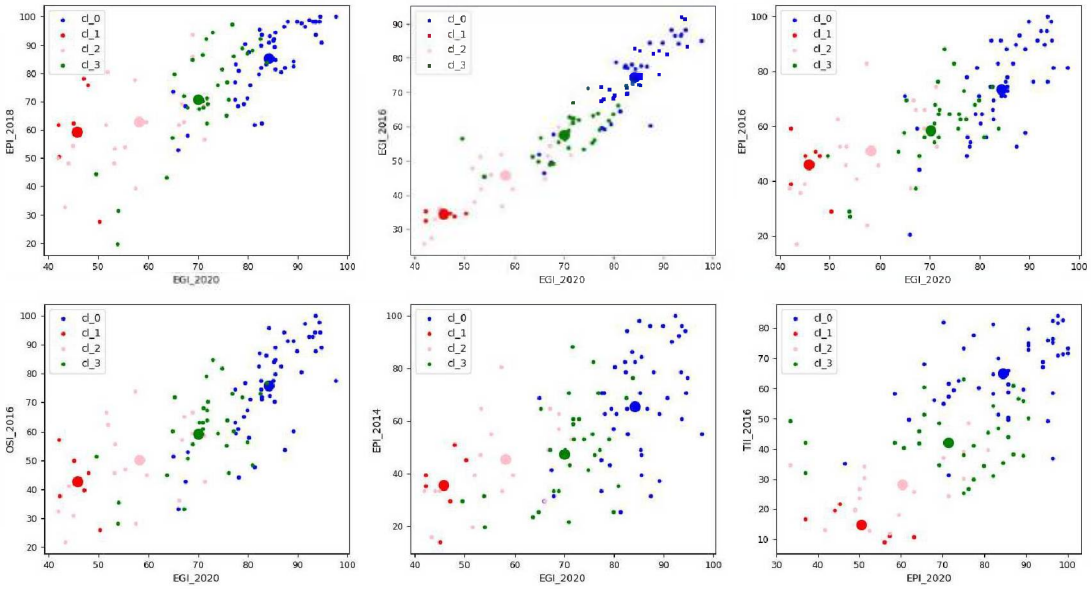

Рис. 3.7 - Результати розподілу кластерів у нестандартизованій вибірці з використанням методів боротьби з викидами

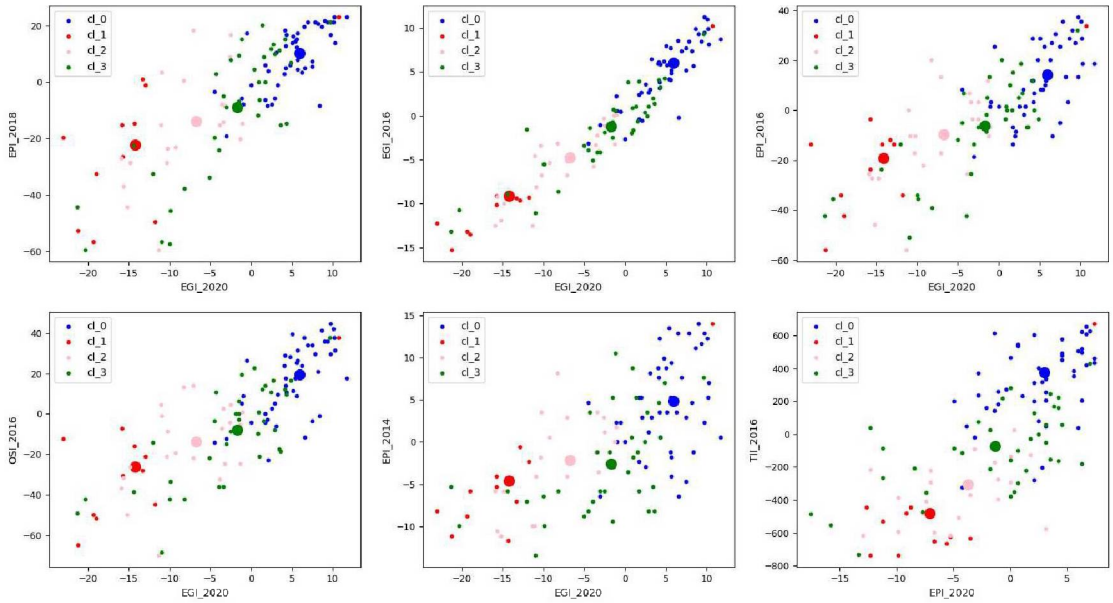

Рис. 3.8 - Результати розподілу кластерів у вибірці з використанням надійного скалера та методів боротьби з викидами

Для вибірки з використанням нормального скалера та методами боротьби з викидами були отримані кращі результати, ніж для попередніх двох. В даному випадку було отримано 93.01% точності та 0.946 значення ROC-AUC. Результати розподілу кластерів зображено на рис. 3.9.

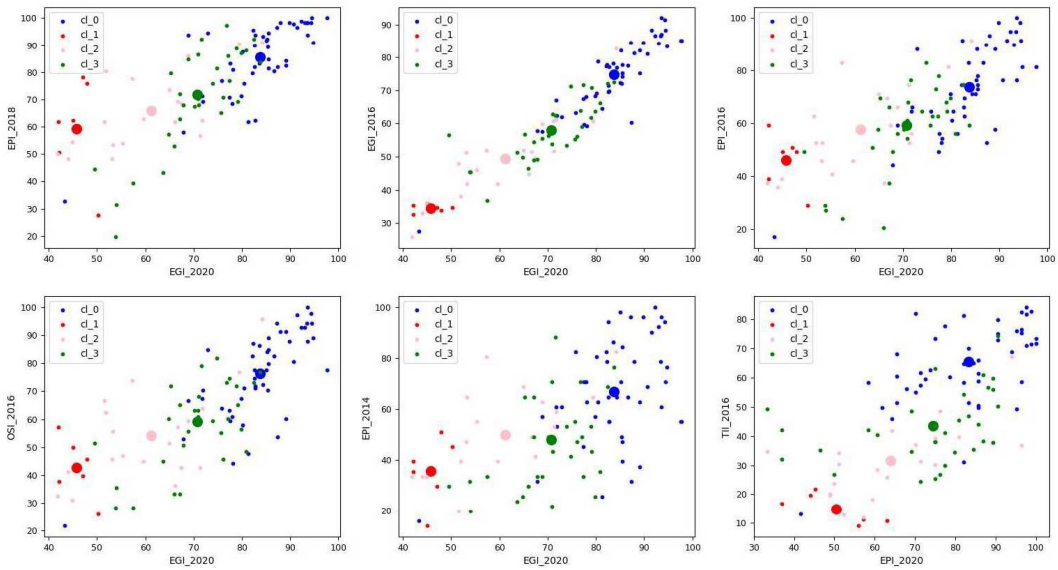

Рис. 3.9 - Результати розподілу кластерів у вибірці з використанням стандартного скалера та методів боротьби з викидами

Заключною вибіркою в цьому розділі є вибірка з використанням мінімум-максимум скалера в діапазоні від -3 до 3 та методів боротьби з викидами. Натренувавши метод кластеризації за допомогою цієї вибірки були отримані такі результати: 93.86% точності та 0.955 значення ROC-AUC. Результати розподілу кластерів зображено на рис. 3.10.

Проаналізувавши результати тренувань методу кластеризації на вибірках з різною обробкою та підготовкою даних, можна визначити, що мінімум-максимум скалер найкраще використовувати для даного методу. Цей вид стандартизації зменшує вплив викидів на тренування методу кластеризації та представляє дані у зручному вигляді для роботи методу. Інші методи

стандартизації також впливають на поліпшенні якості методу кластеризації, тому як без стандартизації даних, метод кластеризації працює гірше.

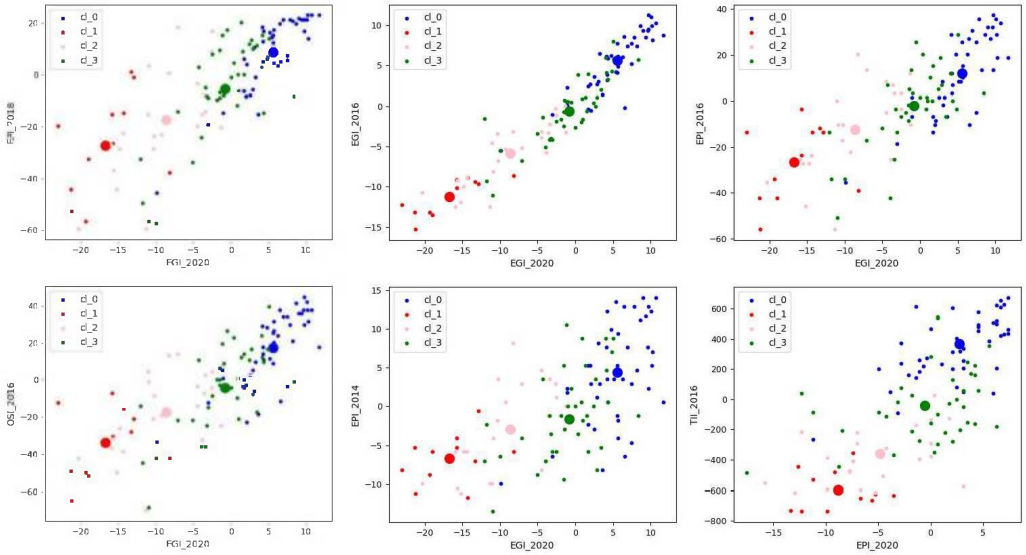

Рис. 3.10 - Результати розподілу кластерів у вибірці з використанням мінімального-максимального скалера та методів боротьби з викидами

Порівнюючи розроблений метод з іншими методами кластеризації, використовуючи вибірку на якій новий метод кластеризації показав найкращий результат, можна сказати, що новий метод значно перевершує інші методи кластеризації та є ефективним для задач моніторингу і оцінки рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Висновки до розділу 3

В підрозділі 3.1 був приведений опис основної мови програмування на якому базується програмне забезпечення та представлено пояснення про цей вибір. Також був проведений опис розробки програмного забезпечення та бібліотек, які для цього використовувалися. Були приведені приклади роботи програмного забезпечення та визначені дані, які виводяться у термінал при виконанні програмного забезпечення.

В підрозділі 3.2 були представлені результати роботи нового методу кластеризації даних на основі агентно-орієнтованого підходу. В результаті були зроблені висновки про вплив обробки та підготовки даних на якість методу кластеризації. Також було проведено порівняння цього методу з іншими методами кластеризації та підтверджено, що даний метод значно перевищує показники якості інших методів кластеризації.

ВИСНОВКИ

В даній кваліфікаційній роботі були досягнуті наступні результати:

- визначено постановку задачі даної кваліфікаційної роботи, включаючи мету, об'єкт та предмет дослідження;
- визначено постановку задачі кластеризації даних;
- проведено аналітичний огляд існуючих методів кластеризації даних, а також метрик для знаходження відстані між елементами вибірки даних та метрики для оцінювання якості методів кластеризації;
- проведено аналітичний огляд існуючих програмних засобів для кластеризації даних. В цей огляд включені програмні засоби, реалізовані на різних мовах програмування;
- приведена формалізація уявлення вхідних та вихідних даних. Для вхідної вибірки описані змінні стану та визначена цільова змінна стану;
- виконано опис методів обробки та підготовки вхідних даних та проведено підготовку та обробку вхідних даних для покращення ефективності методу кластеризації;
- розроблено новий метод кластеризації даних на основі агенто-орієнтованого підходу (або навчання з підкріпленням) та виконано його опис;
- виконано опис обчислювального алгоритму для нового методу кластеризації;
- проведено порівняння ефективності різних методів кластеризації даних та виконано огляд та аналіз отриманих результатів.

Головним результатом даної кваліфікаційної роботи був новий ефективний метод кластеризації даних на основі інформаційних критеріїв в системах моніторингу рівня соціально-економічного розвитку країн у контексті

їх цифрового прогресу для підвищення рівня стійкості до відмов систем шляхом зниження ймовірності некоректного визначення стану систем (помилки третього роду при кластерному аналізі стану систем). Новий метод кластеризації використовує агентно-орієнтований підхід для тренування, що в наш час є прогресивним підходом, який широко використовується за кордоном та тільки розвивається в Україні.

Головним напрямком використання нового методу кластеризації є використання в системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

Ефективність нового методу кластеризації була підвищена за рахунок правильного застосування методів обробки та підготовки вхідних даних. Було проведено тестування роботи методу на вибірках, які були оброблені та підготовлені різними методами боротьби з викидами та стандартизації даних. В результаті цього, найкращою вибіркою для роботи нового методу кластеризації була вибірка оброблена за допомогою мінімального-максимального скалера, який масштабував значення до діапазону від 0 до 1. Використовуючи цю вибірку, новий метод кластеризації досяг 95.652 % точності та 0.9744 значення ROC-AUC, що є дуже гарним результатом, так як набір економічних даних не є збалансованим по класам.

Аналізуючи роботу нового методу кластеризації в розділі 8, можна визначити, що метод кластеризації відображає стійкість до викидів, так як, використовуючи вибірку на якій не було застосовано методи для боротьби з викидами, було отримано досить високий результат якості кластеризації порівнюючи з вибірками, де методи боротьби з викидами були застосовані.

При проведені порівняння з іншими методами кластеризації даних, дослідження показало, що новий метод значно перевершує інші методи кластеризації. Для порівняння використовувалася найкраща оброблена та підготовлена вибірка вхідних даних.

В майбутньому планується вдосконалення розробленого методу кластеризації даних за рахунок використання глибокого навчання для зменшення розмірності вхідних даних. Скоріш за все буде використаний автоенкодер, який зменшить розмірність вхідних даних. Це дозволить підвищити якість методу кластеризації даних, через те, що зазвичай методи кластеризації більш якісно вирішують задачу кластеризації у мало-розмірному просторі значень.

Ще одною задачею для майбутнього розвитку є тестування методу кластеризації на різних вибірках та адаптація методу для різних вхідних даних. Це дозволить покращити роботу методу в цілому та зробити цей метод універсальним.

Отже в даній кваліфікаційній роботі були виконані усі сформовані задачі та досягнута мета, а саме підвищення рівня стійкості до відмов систем шляхом зниження ймовірності некоректного визначення стану систем (помилки третього роду при кластерному аналізі стану систем) за допомогою ефективного методу кластеризації даних на основі інформаційних критеріїв в системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мандель И.Д. Кластерный анализ. - М.: Финансы и статистика. 1988. - 176с.
2. Mahalanobis P.C. Analysis of race mixture in Bengal, J. Asiat. Soc. (India), Vol. 23, (1925), 301-310
3. Kullback, S., Leibler, R.A. On information and sufficiency. Annals of Mathematical Statistics. 22 (1): 79–86, 1951. DOI: 10.1214/aoms/1177729694.
4. Гайдышев И.П. Анализ и обработка данных. Специальный справочник. – СПб.: Издательство «Питер», 2001 г., 752 стр.
5. Загоруйко Н.Г., Елкина В.Н., Лбов Г.С. Алгоритмы обнаружения эмпирических закономерностей. – Новосибирск: Наука, 1985
6. P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20:53–65, 1987.
7. Santos J., Embrechts M. On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification. URL: http://dx.doi.org/10.1007/978-3-642-04277-5_18. Дата звернення: 25.12.2020.
8. Wang X., Xu Y. An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index. URL: http://dx.doi.org/10.1088/1757-899X/569/5/052024. Дата звернення: 12. 01. 2021.
9. Wishart D. "Exploiting the graphical user interface in statistical software: the next generation". Interface '98. Computing Science and Statistics, 30, 1998, 257-263
10. Benabdellah A., Benghabrit A., Bouhaddou I. A survey of clustering algorithms for an industrial context // Procedia Computer Science. 2019. Т. 148. С. 291-302.

11. Sneath, P.H.A. and Sokal, R.R., Numerical Taxonomy: The Principles and Practice of Numerical Classification, W.H. Freeman and Company, San Francisco (1973)

12. Good, I.J., 'Categorization of classification' In Mathematics and Computer Science in Biology and Medicine, HMSO, London, 115-125 (1965).

13. Wishart, D. (1986). Estimation of Missing Values and Diagnosis Using Hierarchical Classifications, Computational Statistics Quarterly, 2(1), p. 125-134

14. K-means clustering. URL: https://en.wikipedia.org/wiki/K-means_clustering Дата звернення: 29. 11. 2020.

15. Pelleg, Dan; Moore, Andrew (1999). "Accelerating exact k -means algorithms with geometric reasoning". Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '99. San Diego, California, United States: ACM Press: 277–281. doi:10.1145/312129.312248. ISBN 9781581131437. S2CID 13907420

16. MacKay, David (2003). "Chapter 20. An Example Inference Task: Clustering" (PDF). Information Theory, Inference and Learning Algorithms. Cambridge University Press. pp. 284–292. ISBN 978-0-521-64298-9. MR 2012999.

17. Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A k-Means Clustering Algorithm". Journal of the Royal Statistical Society, Series C. 28(1): 100–108. JSTOR 2346830.

18. Agglomerative Hierarchical Clustering - Datanovia. URL: https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/. Дата звернення: 24. 11. 2021.

19. Bepery C. и др. Improved Mean Shift Algorithm for Maximizing Clustering Accuracy. URL: http://dx.doi.org/10.38032/jea.2021.01.001. Дата звернення: 18. 11. 2021.

20. Elbow method (clustering) - Wikipedia. URL: https://ru.qaz.wiki/wiki/Elbow_method_(clustering). Дата звернення: 13. 11. 2021.
21. BIRCH: an efficient data clustering method for very large databases: ACM SIGMOD Record: Vol 25, No 2. URL: https://dl.acm.org/doi/10.1145/235968.233324. Дата звернення: 15. 11. 2021.
22. Clustering — scikit-learn 0.23.2 documentation. URL: https://scikit-learn.org/stable/modules/clustering.html. Дата звернення: 15. 11. 2021.
23. Brownlee J. Clustering Algorithms With Python. URL: https://machinelearningmastery.com/clustering-algorithms-with-python/. Дата звернення: 15. 11. 2021.
24. Demo of DBSCAN clustering algorithm. URL: https://scikit-learn.org/stable/auto_examples/cluster/plot_dbscan.html#sphx-glr-auto-examples-cluster-plot-dbscan-py. Дата звернення: 24. 11. 2021.
25. Open source Clustering software. URL: http://bonsai.hgc.jp/~mdehoon/software/cluster/index.html. Дата звернення: 12. 10. 2021.
26. CLUTO - Software for Clustering High-Dimensional Datasets | Karypis Lab. URL: http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview. Дата звернення: 12. 10. 2021.
27. Загальні положення про кластеризацію даних. URL: http://www.machinelearning.ru/wiki/index.php?title=Кластеризация. Дата звернення: 01. 11. 2020.
28. He Z. Approximation algorithms for K-modes clustering. URL: https://arxiv.org/ftp/cs/papers/0603/0603120.pdf. Дата звернення: 13.10.2021.
29. Pandas - Python Data Analysis Library. URL: https://pandas.pydata.org/. Дата звернення: 13.11.2021.

30. NumPy - The fundamental package for scientific computing with Python. URL: https://numpy.org/. Дата звернення: 13.11.2021.
31. scikit-learn: machine learning in Python — scikit-learn 1.0.1 documentation. URL: https://scikit-learn.org/stable/. Дата звернення: 13.11.2021.
32. Matplotlib — Visualization with Python. URL: https://matplotlib.org/. Дата звернення: 13.11.2021.
33. Seaborn: statistical data visualization — seaborn 0.11.2 documentation. URL: https://seaborn.pydata.org/. Дата звернення: 13.11.2021.
34. SciPy - Fundamental algorithms for scientific computing in Python. URL: https://scipy.org/. Дата звернення: 13.11.2021.
35. Tharwat A. Principal component analysis - a tutorial. URL: http://dx.doi.org/10.1504/IJAPR.2016.079733. Дата звернення: 24. 11. 2021.
36. What Is Principal Component Analysis (PCA) and How It Is Used? URL: https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used-507186. Дата звернення: 24. 11. 2021.
37. Jolliffe I., Jackson J. A User's Guide to Principal Components. // The Statistician. 1993. Т. 42. № 1. С. 76.
38. Cai T., Ma R. Theoretical Foundations of t-SNE for Visualizing High-Dimensional Clustered Data. URL: https://arxiv.org/abs/2105.07536. Дата звернення: 24. 11. 2021.
39. Бакуменко Н.С. Методи кластеризації даних на основі інформаційних критеріїв / Н.С. Бакуменко, В.В Донець, Д.О. Шевченко, О.О. Одинець, М.Л. Угрюмов // Науковий збірник «Праці міжнародної науково-технічної конференції «Комп'ютерне моделювання у наукоємних технологіях (КМНТ -2021)», (21-23 квітня 2021 р., м. Харків, Україна). – Харків: ХНУ, 2021. – 5 с.

40. Mykhaylo Ugryumov, Nina Bakumenko, Viktoriia Strilets. Application of the C-Means Fuzzy Clustering Method for the Patient's State Recognition Problems in the Medicine Monitoring Systems. Conference: Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Systems (COLINS-2019). Volume I: Main Conference Kharkiv, Ukraine, 10p., April 2019. URL: https://www.researchgate.net/publication/338819685 (Last accessed 15.02.2021).

41. Md. Abu Bakr Siddique, Rezoana Bente Arif, Mohammad Mahmudur Rahman Khan, Zahidun Ashrafi. Implementation of Fuzzy C-Means and Possibilistic C-Means Clustering Algorithms, Cluster Tendency Analysis and Cluster Validation. ArXiv e-Journal (ISSN 2331-8422). May 2019. URL: https://arxiv.org/abs/1809.08417v3 (Last accessed 27.02.2021).

42. Salar Askari. Fuzzy C-Means clustering algorithm for data with unequal cluster sizes and contaminated with noise and outliers: Review and development. Expert Systems with Applications, volume 165, August 2020. DOI: 10.1016/j.eswa.2020.113856. URL: https://doi.org/10.1016/j.eswa.2020.113856 (Last accessed: 27.02.2021).

43. Roland Winkler, Frank Klawonn, Rudolf Kruse. Problems of Fuzzy c-Means Clustering and Similar Algorithms with High Dimensional Data Sets. Conference: 34th Annual Conference GfKl 2010, 3rd German-Japanese Workshop. July 2010. DOI: 10.1007/978-3-642-24466-7-9. URL: https://www.researchgate.net/publication/225005676 (Last accessed: 20.02.2021).

ДОДАТКИ

Додаток А

Завдання на дипломну роботу

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) Магістр
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 –Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної та прикладної системотехніки
_____д.т.н., проф. Шматков С. І.

«____» _____20___року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Шевченка Дмитра Олександровича
(прізвище, ім'я, по батькові студента)

1. ТЕМА РОБОТИ «МЕТОД КЛАСТЕРИЗАЦІЇ ДАНИХ НА ОСНОВІ ІНФОРМАЦІЙНИХ КРИТЕРІЇВ ПРИ ОЦІНЮВАННІ РІВНЯ СОЦІАЛЬНО-ЕКОНОМІЧНОГО РОЗВИТКУ КРАЇН У КОНТЕКСТІ ЇХ ЦИФРОВОГО ПРОГРЕСУ»

керівник роботи **Угрюмов Михайло Леонідович, д.т.н, професор**
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "___"_____20__року №_____

2. Строк подання студентом роботи_____

3. Перелік питань, які потрібно розробити)

1. Визначення постановки задачі кластеризації даних.
2. Аналітичний огляд існуючих методів кластеризації даних.
3. Огляд існуючих програмних засобів для кластеризації даних.
4. Формалізація і уявлення вхідних даних.
5. Підготовка та обробка вхідних даних для покращення ефективності методу.
5. Опис методу кластеризації даних.
6. Опис обчислювального алгоритму вирішення задачі.
7. Опис обраного програмного забезпечення для вирішення задачі.
8. Порівняння ефективності різних методів кластеризації даних.
9. Огляд та аналіз отриманих результатів.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1.	Визначення постановки задачі кластеризації даних	15.10.20 – 01.11.20
2.	Аналітичний огляд існуючих методів кластеризації даних	01.11.20 – 01.01.21
3.	Огляд існуючих програмних засобів для кластеризації даних	01.01.21 – 01.02.21
4.	Формалізація і уявлення вхідних даних	02.02.21 – 01.03.21
5.	Підготовка та обробка вхідних даних для ефективності методу	02.03.21 – 15.04.21
6.	Опис методу кластеризації даних	19.03.21 – 16.04.21
7.	Опис обчислювального алгоритму вирішення задачі	17.04.21 – 17.05.21
8.	Опис обраного програмного забезпечення для вирішення задачі	28.04.21 – 19.05.21
9.	Порівняння ефективності різних методів кластеризації даних	20.05.21 – 30.06.21
10.	Огляд та аналіз отриманих результатів	30.06.21 – 12.07.21
11.	Створення презентації докладу	06.07.21 – 19.08.21
12.	Підготовка доповіді на конференцію	20.08.21 – 03.09.21
13.	Розробка пояснювальної записки дипломної роботи	21.09.21 – 05.10.21
14.	Представлення дипломного проекту керівнику та рецензенту	06.10.21 – 05.11.21
15.	Підготовка до захисту дипломної роботи	06.11.21 – 21.11.21

5. Дата видачі завдання_____

Студент Шевченко Д.О. _____
підпис

Керівник роботи Угрюмов М.Л. _____
підпис

Додаток Б

Затверджую

_____

«_____»_____2021 р.

Технічне завдання
на розробку програмного виробу
«Метод кластеризації даних на основі інформаційних критеріїв та агентно-орієнтованого підходу»

Назва розділу	Назва і зміст підрозділу
1. Введення	1.1. Назва виробу: метод кластеризації даних на основі інформаційних критеріїв та агентно-орієнтованого підходу. 1.2. Галузь застосування: економіка.
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – «Комп'ютерна інженерія» 2.2. Завдання на кваліфікаційну роботу магістра затверджено наказом ректора № 0210-05/1804 від 10.09.2021.
3. Призначення розробки	3.1. Мета розробки програмного виробу: підвищення рівня стійкості до відмов систем шляхом зниження ймовірності некоректного визначення стану систем (помилки третього роду при кластерному аналізі стану систем) за допомогою ефективного методу кластеризації даних на основі інформаційних критеріїв в системах моніторингу рівня соціально-економічного розвитку країн у контексті їх цифрового прогресу. 3.2. Призначення виробу: використання виробу для підвищення рівня стійкості до відмов систем шляхом зниження ймовірності некоректного визначення стану систем (помилки третього роду при кластерному аналізі стану систем).
4. Вихідні дані	У якості прототипу взяти метод кластеризації с-середніх з модифікацією у вигляді використання відносної ентропії Кульбака-Лейблера для розрахунку функції втрат описаний у науковому збірнику «Праці міжнародної науково-технічної конференції «Комп'ютерне моделювання у наукоємних технологіях

	(КМНТ -2021)», (21-23 квітня 2021 р., м. Харків, Україна). – Харків: ХНУ, 2021. – 5 с. // Бакуменко Н.С. Методи кластеризації даних на основі інформаційних критеріїв / Н.С. Бакуменко, В.В Донець, Д.О. Шевченко, О.О. Одинець, М.Л. Угрюмов
5. Технічні вимоги до виробу	5.1. Вимоги до функціональних характеристик: можливість кластеризувати елементи із вхідної вибірки на основі інформаційних критеріїв. 5.2. Вимоги до надійності: забезпечення працездатності методу кластеризації при подачі вхідних даних неправильного формату. 5.3. Вимоги до умов експлуатації: встановлення мови програмування Python та необхідних бібліотек для роботи методу кластеризації. 5.4. Вимоги до складу і параметрів технічних засобів: комп'ютер або ноутбук із 4 ГБ оперативної пам'яті та процесором не нижче Intel Core i3 9-го покоління . 5.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 8/10, підтримка Anaconda, Python 3. 5.6. Вимоги до маркування та упаковки: жорстка упаковка з пластмаси, маркування на українській та англійській мовах. 5.7. Вимоги до транспортування і зберігання: транспортування в упаковці будь-яким способом, температура транспортування/зберігання +5-+20 С°. 5.8. Спеціальні вимоги: відсутні.
6. Вимоги до документації.	Документацією до виробу «Метод кластеризації даних на основі інформаційних критеріїв та агентно-орієнтованого підходу» вважати: 1) Справжнє Технічне завдання на розробку виробу (представити як Додаток Б до пояснювальної записки до дипломної роботи). 2) Програму і методику випробувань розробленого виробу (представити як Додаток В до пояснювальної записки до дипломної роботи). 3) Опис виробу (представити у Розділі 2 до пояснювальної записки до дипломної роботи). 4) Фрагменти тексту програми (записати на диск з пояснювальною запискою до дипломної роботи).
7. Техніко-	Орієнтовна оцінка ефективності та якості виконуваної

<table>
<tr><td>економічні показники</td><td colspan="2">кластеризації даних: точність (accuracy) повинна бути вище 94.5%, метрика ROC-AUC повинна бути вище 0.95; Порівняння ефективності з іншими методами кластеризації представити у розділі 3.</td></tr>
<tr><td rowspan="11">8. Стадії і етапи розробки</td><td>Дата</td><td>Назва етапу</td></tr>
<tr><td>15.10.20 – 01.11.20</td><td>1. Визначення постановки задачі кластеризації даних;</td></tr>
<tr><td>01.11.20 – 01.01.21</td><td>2. Аналітичний огляд існуючих методів кластеризації даних;</td></tr>
<tr><td>01.01.21 – 01.02.21</td><td>3. Огляд існуючих програмних засобів для кластеризації даних;</td></tr>
<tr><td>02.02.21 – 01.03.21</td><td>4. Формалізація і уявлення вхідних даних;</td></tr>
<tr><td>02.03.21 – 15.04.21</td><td>5. Підготовка та обробка вхідних даних для ефективності методу;</td></tr>
<tr><td>19.03.21 – 16.04.21</td><td>6. Опис методу кластеризації даних;</td></tr>
<tr><td>17.04.21 – 17.05.21</td><td>7. Опис обчислювального алгоритму вирішення задачі.</td></tr>
<tr><td>28.04.21 – 19.05.21</td><td>8. Опис обраного програмного забезпечення для вирішення задачі.</td></tr>
<tr><td>20.05.21 – 30.06.21</td><td>9. Порівняння ефективності різних методів кластеризації даних.</td></tr>
<tr><td>30.06.21 – 12.07.21</td><td>10. Огляд та аналіз отриманих результатів.</td></tr>
<tr><td>9. Порядок контролю і приймання</td><td colspan="2">Загальні вимоги до приймання розроблюваного виробу:
1) Перевірка ходу розробки програмного виробу керівнику робіт виконувати 1 раз в 3 тижні.
2) Випробування виробу відповідно до Програми і методики випробувань провести на базі комп'ютерного класу або на базі приватного приміщення.
3) Захист розробленого виробу провести на засіданні атестаційної комісії.
4) Пояснювальну записку представити на паперових носіях в одному примірнику та в електронному вигляді.</td></tr>
</table>

Виконавець:
студент групи КІ-61

Шевченко Д.О.

_____

Замовник:
д. т. н., професор кафедри теоретичної та прикладної системотехніки

Угрюмов М. Л.

_____

Додаток В
Затверджую

_____
«_____»_____2021 р.

Програма і методика випробувань
виробу
«Метод кластеризації даних на основі інформаційних критеріїв та агенто-орієнтованого підходу»

1 Об'єкт випробувань

1.1 Найменування випробуваного програмного виробу: метод кластеризації даних на основі інформаційних критеріїв та агенто-орієнтованого підходу.

1.2 Область його застосування: економіка.

2. Мета випробувань

Перевірка працездатності виробу та правильності роботи методу кластеризації. Підтвердження характеристик розробленого виробу вимогам, які сформульовані в ТЗ.

3. Загальні положення

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії та перевірку даного виробу.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу або будь-якого приміщення в період роботи атестаційної комісії.

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї Програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до виробу

4.1. Вимоги до функціональних характеристик: можливість кластеризувати елементи із вхідної вибірки на основі інформаційних критеріїв.

4.2. Вимоги до надійності: забезпечення працездатності методу кластеризації при подачі вхідних даних неправильного формату.

4.3. Вимоги до умов експлуатації: встановлення мови програмування Python та необхідних бібліотек для роботи методу кластеризації.

4.4. Вимоги до складу і параметрів технічних засобів: комп'ютер або ноутбук із 4 ГБ оперативної пам'яті та процесором не нижче Intel Core i3 9-го покоління .

4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 8/10, підтримка Anaconda, Python 3.

4.6. Вимоги до маркування та упаковки: жорстка упаковка з пластмаси, маркування на українській та англійській мовах.

4.7. Вимоги до транспортування і зберігання: транспортування в упаковці будь-яким способом, температура транспортування/зберігання +5-+20 C^o.

4.8. Спеціальні вимоги: відсутні.

5. Вимоги до документації

Склад програмної документації, що подається на випробування, включає:

1) Справжнє Технічне завдання на розробку виробу (представити як Додаток Б до пояснювальної записки до дипломної роботи).

2) Програму і методику випробувань розробленого виробу (представити як Додаток В до пояснювальної записки до дипломної роботи).

3) Опис виробу (представити у Розділі 2 до пояснювальної записки до дипломної роботи).

4) Фрагменти тексту програми (записати на диск з пояснювальною запискою до дипломної роботи).

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Засоби випробувань представлено в комп'ютерному класі на яких встановлено наступні програмні засоби: Anaconda, Python 3.8. В програмному засобі Anaconda створено середовище, де встановлені усі необхідні бібліотеки для запуску, такі як numpy, scikit-learn, pandas, matplotlib, seaborn, scipy.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);
- власне випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1) Перевірку комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.

2) Виконання тесту:

2.1) Запустити термінал або команду строку.

2.2) Проглянути написаний код та перевірити на наявність помилок.

2.3) Перевірити наявність встановлених бібліотек шляхом запуску програми на даних, які згенеровані випадковим чином. Приклад виводу програмного забезпечення зображено на рис. В.1.

Рис. В.1 – Приклад виводу даних програмного забезпечення у термінал

3) Якість програмної документації перевіряється на відповідність до стандартів ISO/IEC 26514:2008.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1) Перевірку відповідності технічних характеристик програми вимогам технічного завдання.

2) Перевірку ступеня виконання функціональних вимог до програми.

3) Методику проведення перевірок:

3.1) Запустити термінал або команду строку.

3.2) Перевірити наявність встановлених бібліотек шляхом запуску програми на даних, які згенеровані випадковим чином. При відсутності бібліотек, втановити їх за допомогою Anaconda Navigator.

3.3) Порядок проведення випробувань:

- Запустити програму за допомогою терміналу з використанням прапорів, які вказують на шлях до вхідних даних та параметри методу кластеризації.
- Перевірити працездатність розробленого програмного продукту.
- Перевірити правильність виконання кластеризації даних. Приклад виводу результатів для перевірки зображено на рис. В.2.

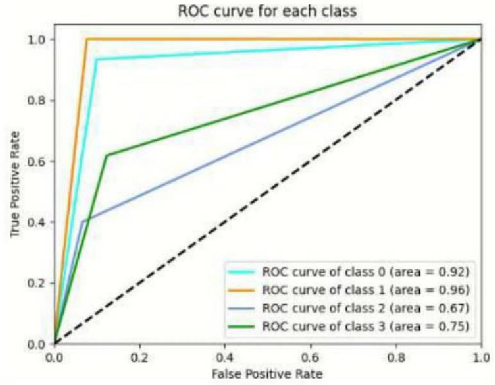

Рис. В.2 – Приклад виводу результатів для перевірки

- Перевірити правильність візуалізації результатів. Приклад візуалізації результатів зображено на рис В.3.

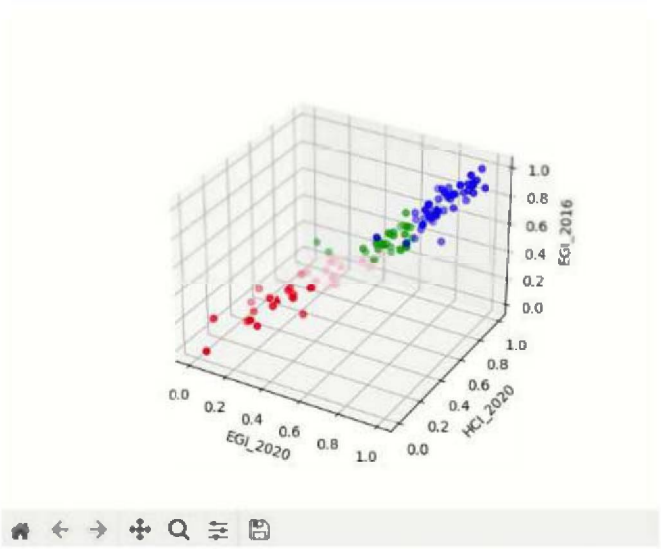

Рис. В.3 – Приклад візуалізації результатів

- Перевірити відповідність показників якості методу кластеризації до вказаних у розділі 3 пояснювальної записки до кваліфікаційної роботи магістра.

4) Якщо перевірки на першому та другому етапах виконано успішно, то виріб вважається таким, що пройшов випробування.

Виконавець:

студент групи КІ-61 Шевченко Д.О. _____

Додаток Г

Лістинг Г.1 – Приклад використання BIRCH.

```
# birch clustering
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import Birch
from matplotlib import pyplot
# define dataset
X, _ = make_classification(n_samples=1000, n_features=2,
n_informative=2, n_redundant=0, n_clusters_per_class=1,
random_state=4)
# define the model
model = Birch(threshold=0.01, n_clusters=2)
# fit the model
model.fit(X)
# assign a cluster to each example
yhat = model.predict(X)
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
     # get row indexes for samples with this cluster
     row_ix = where(yhat == cluster)
     # create scatter of these samples
     pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
pyplot.show()
```

Додаток Д

Лістинг Д.1 – Приклад використання DBSCAN.

```
import numpy as np
from sklearn.cluster import DBSCAN
from sklearn import metrics
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler

# Generate sample data
centers = [[1, 1], [-1, -1], [1, -1]]
X, labels_true = make_blobs(n_samples=750, centers=centers,
cluster_std=0.4,
                            random_state=0)
X = StandardScaler().fit_transform(X)

# Compute DBSCAN
db = DBSCAN(eps=0.3, min_samples=10).fit(X)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_

# Number of clusters in labels, ignoring noise if present.
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)
print('Estimated number of clusters: %d' % n_clusters_)
print('Estimated number of noise points: %d' % n_noise_)
print("Homogeneity: %0.3f" % metrics.homogeneity_score(labels_true,
labels))
print("Completeness: %0.3f" %
metrics.completeness_score(labels_true, labels))
print("V-measure: %0.3f" % metrics.v_measure_score(labels_true,
labels))
print("Adjusted Rand Index: %0.3f"
      % metrics.adjusted_rand_score(labels_true, labels))
print("Adjusted Mutual Information: %0.3f"
      % metrics.adjusted_mutual_info_score(labels_true, labels))
print("Silhouette Coefficient: %0.3f"
      % metrics.silhouette_score(X, labels))

# Plot result
import matplotlib.pyplot as plt
# Black removed and is used for noise instead.
unique_labels = set(labels)
colors = [plt.cm.Spectral(each)
          for each in np.linspace(0, 1, len(unique_labels))]
for k, col in zip(unique_labels, colors):
```

```
    if k == -1:
        # Black used for noise.
        col = [0, 0, 0, 1]
    class_member_mask = (labels == k)
    xy = X[class_member_mask & core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
             markeredgecolor='k', markersize=14)

    xy = X[class_member_mask & ~core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col),
             markeredgecolor='k', markersize=6)

plt.title('Estimated number of clusters: %d' % n_clusters_)
plt.show()
}
```

Додаток Е

Лістинг Е.1 – Реалізація розрахунку коваріаційної матриці.

```
import numpy as np
from scipy import linalg


ARRAY_LAMBDA = 0.001


def np_covariance_matrix(mat_entries, vec_cluster_indexes,
vec_cluster_center):
    """
    :param mat_entries: vector of entry vectors
    :param vec_cluster_indexes: vector of indexes selected entries
    :param vec_cluster_center: vector of cluster center coordinates
    :return: numpy covariance matrix
    """

    mat_cluster_entries = []
    if len(vec_cluster_indexes) != 0:
        for inx in vec_cluster_indexes:
            mat_cluster_entries.append(mat_entries[inx])
    else:
        mat_cluster_entries.append(vec_cluster_center)
        mat_cluster_entries.append(vec_cluster_center[:])
    if len(vec_cluster_indexes) == 1:
        mat_cluster_entries.append(vec_cluster_center)
    npmat_cluster_entries = np.array(mat_cluster_entries).T
    npmat_covariance = np.cov(npmat_cluster_entries)
    return npmat_covariance
def np_inverse_covariance(npmat_covariance):
    """
    :param npmat_covariance: numpy covariance matrix
    :return: numpy inverse covariance matrix
    """

    for i in range(len(npmat_covariance)):
        npmat_covariance[i][i] += ARRAY_LAMBDA
    return linalg.inv(npmat_covariance)
```