МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження методів кластеризації для виділення клієнтських груп

Виконав:
Студент 2 курсу групи ІПЗм-21-2
Тесленко Д.М.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення

Тип програми Освітньо-наукова
Керівник проф. Смеляков С.В.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. Кафедри _____ проф. Дудар З.В.
2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної Інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Програмна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«___»_____202_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Тесленка Дениса Максимовича
(прізвище, ім'я, по батькові)

1. Тема роботи: «Дослідження методів кластеризації для виділення клієнтських груп»

затверджена наказом університету від « 03 » квітня 2023 р. № 83Стз

2. Термін подання студентом роботи до екзаменаційної комісії 12 травня 2023 р.

3. Вихідні дані до роботи встановлений календарний план роботи, методичні вказівки до оформлення пояснювальної записки, алгоритми кластеризації інтелектуального аналізу даних.

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної галузі, огляд наявних математичних моделей, опис базових алгоритмів, опис метрик якості, створення плану для подальшого дослідження теми.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	26.01.2023	виконано
2	Здійснення огляду математичних моделей	28.01.2023	виконано
3	Вибір методів кластеризації для дослідження	05.02.2023	виконано
4	Вибір методів оцінки якості кластеризації	17.02.2023	виконано
5	Розробка плану подальшого дослідження	25.02.2023	виконано
6	Проведення експерименту	03.03.2023	виконано
7	Підготовка пояснювальної записки	30.04.2023	виконано
8	Захист роботи	15.05.2023	виконано

Дата видачі завдання __ _____ 202 р.

Студент _____
(підпис)

Керівник роботи _____ _____проф. Смеляков С.В.____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до роботи, 74 сторінки, 27 рисунки, 5 таблиць, 36 джерел.

АНАЛІЗ ДАНИХ, АНАЛІЗ ЕКОНОМІЧНИХ АНОМАЛІЙ, КЛАСТЕРНИЙ АНАЛІЗ, КЛІЄНТСЬКІ ГРУПИ, МАРКЕТИНГ, МЕТРИКИ ЯКОСТІ, НАБІР ДАНИХ, ПРИЙНЯТТЯ РІШЕНЬ.

Об’єктом дослідження є методи кластеризації клієнтів на окремі групи.

Метою роботи є дослідження алгоритмів кластеризації та їх порівняння задля визначення оптимальних і неоптимальних з них.

Результатом кваліфікаційної роботи є звіт з повним аналізом та порівнянням існуючих алгоритмів і використаних даних.

DATA ANALYSIS, ECONOMICS ANOMALIES ANALYSIS, CLUSTER ANALYSIS, CLUSTER GROUPS, MARKETING, QUALITY METRICS, DATASET, DECISION MAKING.

The object of the study is methods of clustering clients into separate groups.

The purpose of the work is to study clustering algorithms and compare them to determine the optimal and non-optimal ones.

The result of the qualification work is a report with a complete analysis and comparison of existing algorithms and used data.

Умови публікації пояснювальної записки

Я, Тесленко Денис Максимович, студент групи ІПЗм-21-2, здобувач вищої освіти на другому (магістерському) рівні, кафедра програмної інженерії, заявляю: моя кваліфікаційна робота на тему «Дослідження методів кластеризації для виділення клієнтських груп», що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути

опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Якщо є великий масив даних, то найбільш ефективний спосіб зрозуміти, що з ними робити та як аналізувати – розсортувати їх у групі для первинного аналізу. Групувати дані можна за допомогою сегментації за критеріями (наприклад, цінові та вікові групи) або кластеризації (математичні алгоритми самі виявляють «зв’язуючий» критерій або ознаку, який об’єднує дані). Основна відмінність кластеризації полягає в тому, що алгоритми виявляють і об'єднують параметри з схожими рисами з первинного масиву даних.

Маркетинг і продаж – один з напрямків застосування кластерного аналізу. Зокрема для прогнозування майбутнього ведення покупця – персоналізації та таргетування. Кластерний аналіз використовує математичну модель для виявлення групи схожих за поведінкою та властивостями клієнтів, що базується на мінімізації відмінності серед покупців у групі.

Кластерний аналіз – багатомірна статистична процедура, що виконує збір даних, що містить інформацію про вибір об'єктів, а потім упорядковує об'єкти в порівняльно однорідних групах.

Для будь-якого бізнесу кампанія, як маркетингова інвестиція, повинна бути спрямована на конкретну цільову групу.

Стандартний тип даних в датасетах, по яким кластеризуються клієнти, зазвичай виглядає наступним чином:

– базова інформація про клієнта – профіль / ідентифікатор клієнта, місце розташування і ціна покупки;

– інформація про продукти – сегмент, бренд, ієрархія продукту, розмір і т.д.;

– інформація про транзакції – проданий обсяг, деталі рахунку, дані, час та ідентифікатор продукту.

Більш глибоке розуміння клієнтських сегментів досягається шляхом розробки багатовимірних моделей кластерів на основі ключових бізнес-показників, таких як покупки, частота покупок, заказані товари або зміна ціни. Актуальність

результатів кластеризації для бізнес-осіб дозволяє, приймаючи рішення, виявити проблемні кластери, які вимагають продавці, використовувати більше ресурсів для досягнення цілого результату. Потім можна зосередити свої маркетингові та операційні зусилля на правильних кластерах, щоб забезпечити оптимальне використання ресурсів.

Хоча можливості прогнозування, запропоновані кластеризацією, можуть трансформувати результати таргетованого маркетингу, кластеризація показує найбільшу ефективність при використанні разом з іншими рішеннями для роздрібної аналітики. Цінність кластеризації продуктів особливо видна в дуже розрізненому наборі даних. На додаток до підвищення рентабельності маркетингових інвестицій (ROMI) з точки зору прибутковості клієнтів, кластеризація продуктів може допомогти рітейлерам таргетувати та активізувати клієнтів із категорій з невисокою купівельною здатністю.

Предметною галуззю даного дослідження є галузь клієнтської кластеризації. Ця тематика дає можливість детально дослідити різні підходи та алгоритми для вирішення даного завдання.

1. ОПИС ПРОБЛЕМНОЇ ГАЛУЗІ

1.1 Аналіз предметної області

Сьогодні компанії прагнуть отримати корпоративну стійкість, яка стає важливою частиною успішного розвитку бізнесу [1]. Однак без ефективної комунікації з клієнтами це важко зробити [2].

Сегментація ринку – це те, що допомагає компаніям справлятися з цим аспектом і точніше, цілеспрямованіше, ефективніше та вигідніше орієнтуватися на групи споживачів [3]. Сегментацію ринку можна визначити як процес поділу ринку на групи потенційних клієнтів, які мають схожу купівельну поведінку, потреби та особливості [4], виявлення яких, очевидно, вимагає ретельного вивчення ринку та додаткових витрат для організацій. Якщо бізнес не кластеризує існуючих клієнтів, це може призвести до неправильної стратегії таргетування та рекламної моделі [5], [6]. Існують різні методи сегментації, такі як RFM (Recency, Frequency, Monetary), матриця BCG (Boston Consulting Group) або аналіз ABC-XYZ [7]–[9], які будуть детальніше описані далі [10]. Існують різні підходи до визначення правильних стратегій націлювання для кожного клієнта [11], які включають використання нейронних мереж [12]–[14], класифікації [15]–[16] та алгоритмів кластеризації [17]. Алгоритми кластеризації стали цікавими для маркетингу, оскільки вони здатні автоматизувати процес сегментації ринку та розділити задані дані про поведінку клієнтів на однорідні групи. Однак навіть тут рівень оптимізації може залежати від бюджету та/або обраної стратегії (наприклад, кількості груп клієнтів, які компанія може собі дозволити або хоче націлити відповідно до своєї політики) [18].

Кластерний аналіз – ефективний інструмент, здатний покращити клієнтський досвід і разом з тим зекономити маркетинговий бюджет. Розглянемо як це працює і як кластеризація застосовується на практиці.

Для початку опишемо до яких саме аспектів маркетингових кампаній застосовується кластеризація[2]:

– аналіз цін: кластеризація є вихідною точкою для більш глибокого аналізу цін, щоб отримати сайти та збільшити обсяги продажів на основі прогнозованих

змін у структурі (шаблоні) закупок за відношенням до змін цін у кожному ідентифікованому кластері.

– аналіз аномалій: можна виявити неочевидні закономірності та аномалії у наведенні покупців.

– аналіз частоти покупок: дозволяє сформувати кластери покупців, які стали купувати частіше або рідше в конкретному проміжку часу.

– аналіз часу покупок: кластеризація часу покупок протягом дня або протягом тижня або в різні сезони може виявити періоди максимальної та мінімальної завантаження для оптимізації логістики та перерозподілу трафіку.

– аналітика дистрибуції: дистриб'ютори також можуть отримати вигоду від кластеризації продуктів, оскільки це допомагає їм ідентифікувати товари, які можна зв'язати разом, щоб уникнути багатократних поїздок і оптимізувати транспортні ресурси.

– прогнозовані сайти: надання кластеризації продуктів може дати рітейлерам можливості прогнозування, дозволяючи їм скласти нового клієнта з уже існуючими кластерами продуктів на основі визначених атрибутів клієнта, таких як бізнес-категорія, місцезнаходження та пропоновані послуги.

– аналіз просування: групування схожих продуктів на основі кластеризації товарів може допомогти роздрібним продавцям ідентифікувати набори продуктів, щоб підвищити продаж і збільшити кількість товарів, заказаних конкретним покупцем, на основі виявлених сходів у виборі.

У випадку кластеризації клієнтських груп існують 3 основні способи розбиття за характеристиками. Розглянемо їх.

ABC-XYZ. Основна ідея розділити клієнтів за загальним вкладом у вашій прибуток та за динамічним зростанням показників. ABC відповідає за вклад у прибуток, XYZ відповідає за стабільність прибутку (див. рис. 1).

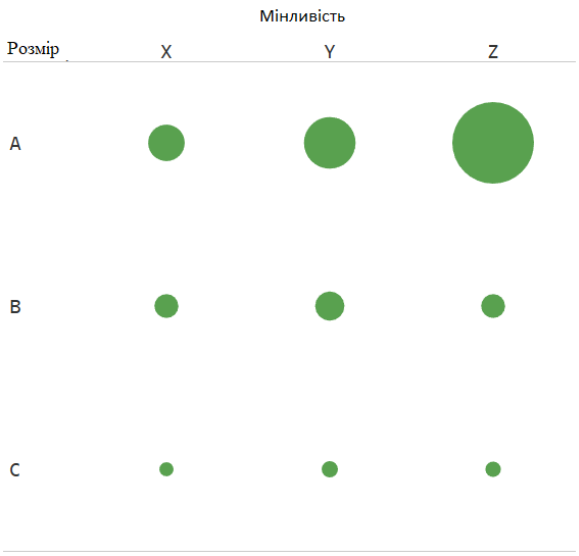

Рисунок 1 – ABC-XYZ розбиття

Це формує 9 сегментів:

– AX - сама більша і зі стабільною виручкою
– AZ - більше, але рідко роблять придбання, прибуток нестабільний
– CX - найдрібніші контриб'ютори, але зі стабільною виручкою
– CZ - дрібні покупці, придбання відбуваються рідко

У сегменті A визначають клієнтів, які формують 80% виручки, у сегменті B, хто дає ще 15% і в сегменті C, хто дає 5%. У сегменті X – найменша мінливість виручки (можна взяти 33 проценти), Z – найвища варіативність (відповідно верхньому 33 процентам). Під мінливістю розуміється величина дисперсії виручки.

Що дає цей аналіз: він дозволяє розділити ваших клієнтів на групи за ступенем важливості для вашого бізнесу. Клієнти з групи AX, AY, AZ найбільші і їм повинні приділяти їм більше всього уваги. Клієнти групи BX, BY вимагають додаткового уваги, їх можна розвивати. Увагу до груп в інших категоріях можна знизити. Особливо добре, якщо вдається виділити спільноти між клієнтами в різних сегментах, що дозволить вам націлити зусилля на залучення потрібних клієнтів.

RFM (Останній час покупки-Частота-Гроші). Основна ідея розділити клієнтів за 3-ма характеристиками: як давно відбулося останнє придбання клієнта (давність), як часто він купує товари (частота), який обсяг виручки він створив (гроші). У цілому підхід згадує ABS-XYZ, але кілька під іншим кутом.

В рамках цього підходу розділяються клієнти за групами Recency (див. рис. 2), наприклад:

– 0-30 днів;
– 31-60 днів;
– 61-90 днів;
– 90+.

По кількості покупок, наприклад:

– більше 15;
– 10-14;
– 5-9;
– 0-4.

По обсягу виручки:

– 1000+;
– 400-1000;
– 0-399.

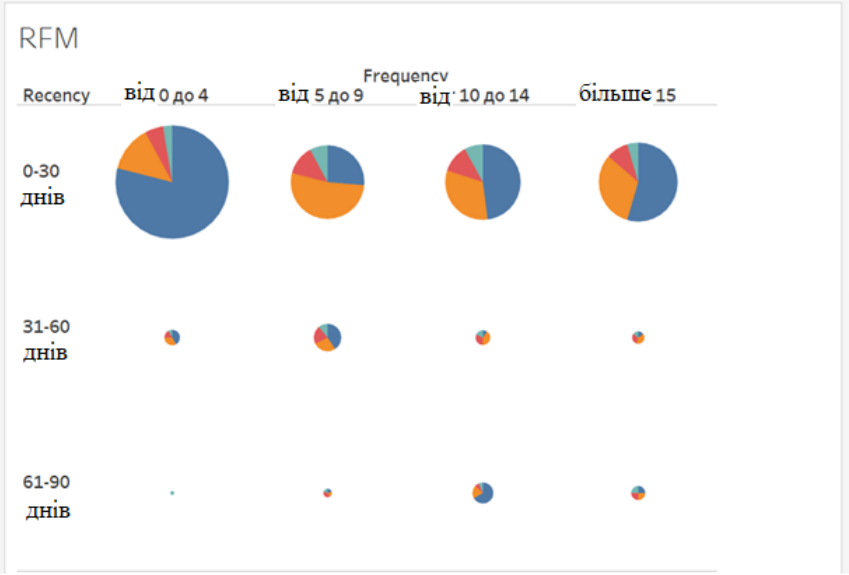

Рисунок 2 – RFM розбиття

Зрозуміло, що для кожного конкретного продукту, додатка або товару потрібно встановити свої межі.

У підсумку можна розділити клієнтів на безліч груп, кожен з яких характеризує клієнта за ступенем важливості для бізнес-стратегії.

Матриця BCG. Основна ідея полягає в тому, щоб розділити клієнтів за категоріями обсягу виручки та темпів зростання виручки. Такий підхід дозволяє визначити, хто великий і наскільки швидко росте. Всі клієнти розкладаються на 4 квадрати (див. рис. 3):

– зірки – найбільші клієнти з високими темпами зростання виручки. Це клієнти, кому потрібно приділити більшу увагу. Це сильна точка росту;

– дійні корови – великі клієнти, з низькими або негативними темпами виручки. Ці клієнти будуть формувати ядро поточної виручки. Проґавите корів і втратите бізнес;

– темні коники – поки дрібні клієнти, але з великим темпом зростання. Це група клієнтів, на кого треба звернути увагу, так як вони можуть вирости до зірки або дійних корів;

– собаки – дрібні клієнти з низькими або негативними темпами зростання. Це клієнти, яким можна не приділяти найбільшу увагу і застосовувати до них масові методи обслуговування, для скорочення витримок.

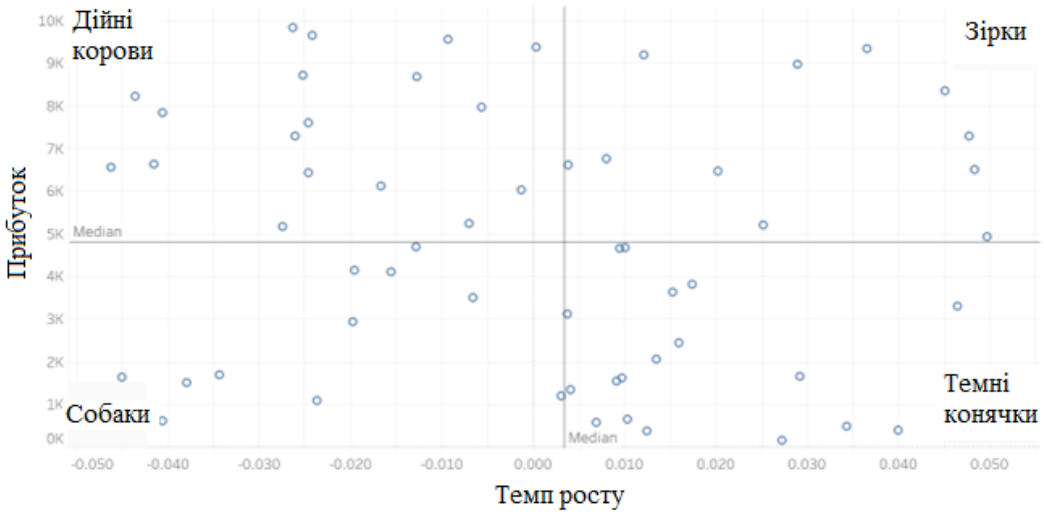

Рисунок 3 – Матриця BCG

Перевага всіх евристичних методів – відносна простота реалізації та можливість розділити своїх клієнтів на зрозумілі з точки зору бізнесу групи.

Недоліки в тому, що ми використовуємо лише кілька властивостей клієнтів, для їх опису та виключаємо з розгляду інші фактори. До того ж, частіше за все клієнти знаходяться в сегментах тимчасово, змінюють положення, а встановити реальну спільноту всередині сегмента виявляється складно.

Незалежно від обраних характеристик та критеріїв для вирішення проблеми кластеризації можна використати велику кількість різноманітних алгоритмів [19]. Усі вони категоріально поділяються за принципами своєї роботи або формою вихідних кластерів чи способом їх подання. Так можна виділити, зокрема, ієрархічні та пласкі [20].

Ієрархічні алгоритми (так звані алгоритми таксономії) будують не одне розбиття набору на кластери, що не перетинаються, а систему вкладених розбиттів. Іншими словами, на виході ми отримуємо дерево, що складається з кластерів, коренем якого є вибірка в цілому, а листям – найбільш дрібні кластери [21]. Власне алгоритм ієрархічної кластеризації і є головним представником цього підходу.

На відміну від нього, пласкі алгоритми, такі як k-means чи MeanShift будують одне розбиття об’єктів на кластери.

Також алгоритми поділяються на чіткі та нечіткі, що означає вид вихідних даних кластеризації. Чіткі алгоритми (DBSCAN, k-means) кожен об'єкт ставлять у відповідність одному кластеру, таким чином кожен об'єкт може належати тільки одному кластеру. Алгоритми, що відносяться до нечітких (c-means) кожному об'єкту ставлять декілька значень, що показують ступінь відношення об'єкта до кожного з кластерів. Це означає, що кожен об'єкт відноситься до кожного кластеру з деякою ймовірністю.

Іншою відмінністю у алгоритмах може бути форма кластерів. У той час як деякі алгоритми (OPTICS, DBSCAN) виділяють кластери будь-якої форми, інші, такі як MeanShift, будують кластери тільки радіально.

Це означає, що кожен із алгоритмів має свої переваги, недоліки, особливості використання та можуть потенційно бути порівняними на конкретній задачі виділення клієнтських груп.

1.2 Постановка задачі

З огляду на інформацію наведену у попередньому пункті можна сформулювати задачу даної роботи – проведення дослідження, яке порівнює ефективність застосування різних алгоритмів кластеризації для виділення клієнтських груп.

Для визначення ефективності треба визначити набір характеристик, за якими можна провести порівняння:

- якість кластеризації;
- якість фільтрації викидів;
- можливість автоматизації визначення оптимальної кількості кластерів;
- можливість виділення кластерів різної щільності.

Окремо треба зупинитися на понятті якості кластеризації. Якість кластеризації може визначатися декількома способами (формулами), які будуть детальніше розглянуті у наступному розділі:

- Silhouette index;
- Calinski-Harabasz index;
- Davies-Bouldin index.

Під час вибору алгоритмів кластеризації було вирішено обрати алгоритми що відносяться до різних категорій алгоритмів. Таким чином було обрано 5 алгоритмів, які є часто-застосовними при вирішення задач кластеризації:

- k-means;
- BIRCH;
- Agglomerative clustering (агломераційна кластеризація);
- DBSCAN;
- OPTICS.

Аби виконати поставлену задачу розділимо її на набір підзадач:

- здійснити ознайомлення з математичним представленням та принципами роботи обраних алгоритмів;

– обрати та описати датасет, на основі якого буде проводитися дослідження застосовності обраних алгоритмів;

– програмно реалізувати усі алгоритми;

– побудувати план експерименту дослідження, за яким можна буде провести порівняння;

– провести дослідження, зробити усі релевантні розрахунки згідно плану;

– формалізувати результати дослідження.

Варто зазначити, що підготовчими кроками вважаються усі, що розміщені до плану експерименту дослідження.

Таким чином, у кінці роботи планується отримати вичерпну порівняльну характеристику алгоритмів кластеризації та визначити оптимальні і неоптимальні.

2 МАТЕМАТИЧНЕ ПРЕДСТАВЛЕННЯ

2.1 Огляд математичних моделей

Існує кілька підходів до кластеризації. Не розглядаючи повний список, зосередимося на тих, які будуть використані у рамках даної роботи. Кожен підхід краще всього підходить для певного розподілу даних [21].

Кластеризація на основі центроїда організовує дані в неієрархічних кластерах. K-means – це найбільш часто використовуваний алгоритм кластеризації на основі центроїдів (див. рис. 4). Алгоритми на основі центроїдів ефективні, але чутливі до початкових умов і викидів.

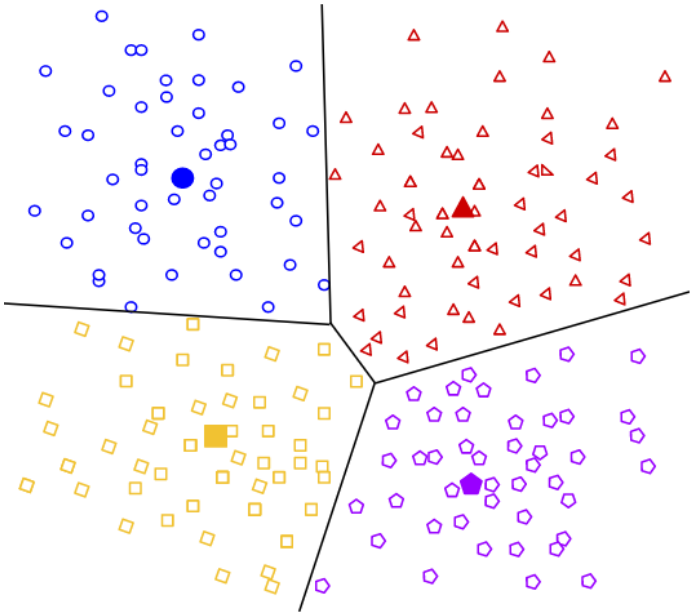

Рисунок 4 – Центроїдна кластеризація

Кластеризація на основі щільності об'єднує області високої щільності прикладів у кластери (див. рис. 5). Це дозволяє використовувати розподіл довільної форми, якщо можна з'єднати щільні області. Ці алгоритми мають труднощі з даними різної щільності та високої розмірності. Крім того, по задуму ці алгоритми не відносять до кластерів викиди.

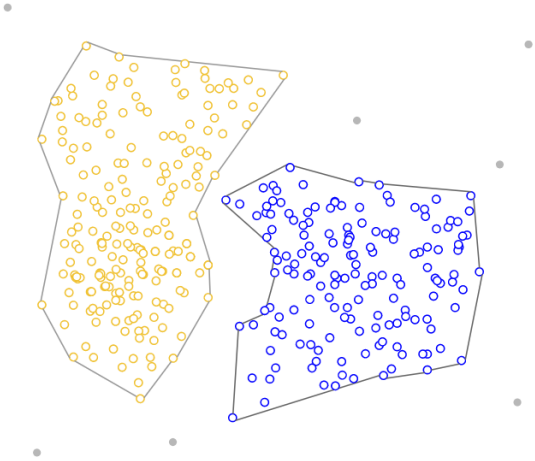

Рисунок 5 – Щільнісна кластеризація

Кластеризація на основі розподілу передбачає, що дані складаються з розподілу, таких як розподіл Гауса. На рисунку 6 алгоритм на основі розподілу об'єднує дані в три розподілу Гауса. По мірі збільшення відстані від центру розподілу зменшується ймовірність того, що точка належить розподілу. Шари показують зменшення ймовірності. Якщо невідомий тип розподілу даних, треба використовувати інший алгоритм.

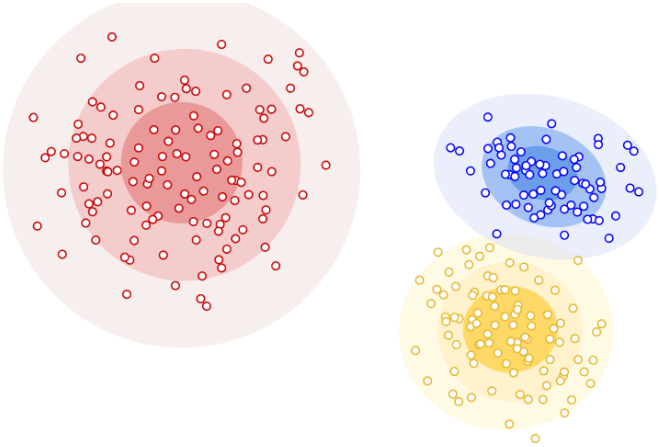

Рисунок 6 – Кластеризація на основі розподілу

Ієрархічна кластеризація створює дерево кластерів (див. рис. 7). Логічно, що ієрархічна кластеризація добре підходить для ієрархічних даних. Крім того, ще однією перевагою є те, що будь-яку кількість кластерів можна вибрати, розрізавши дерево на потрібному рівні.

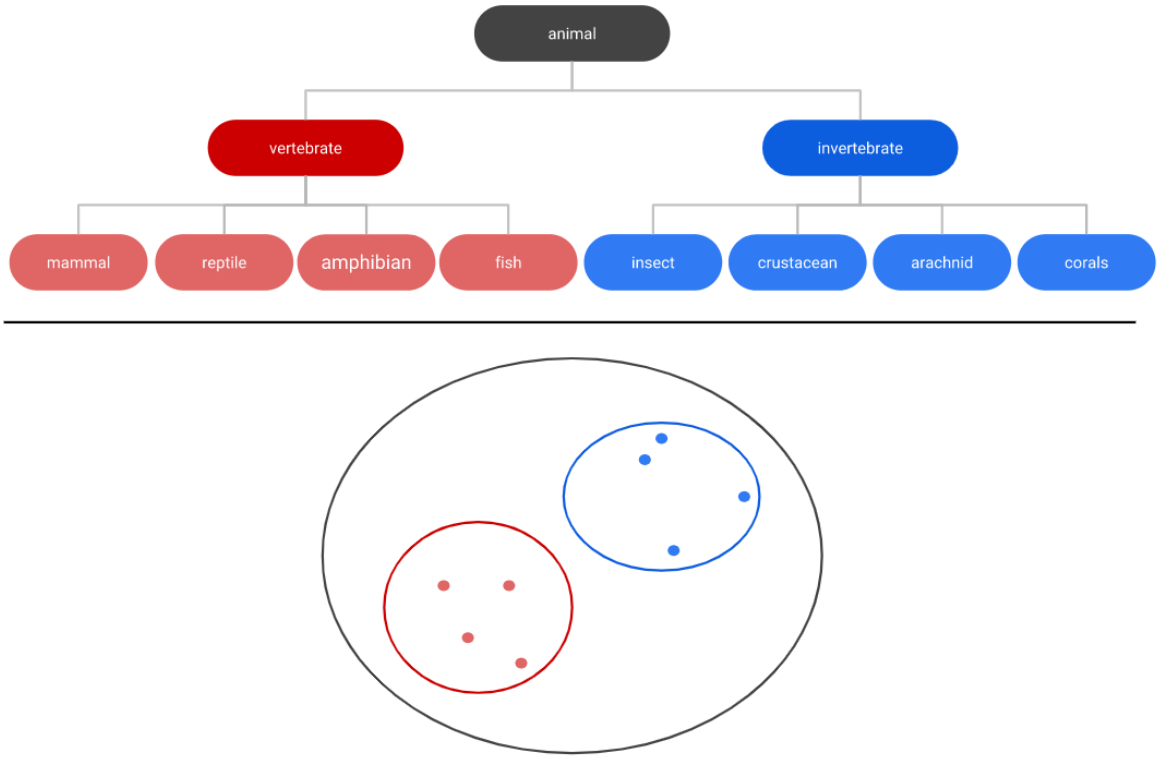

Рисунок 7 – Ієрархічна кластеризація

Кожен алгоритм кластеризації оцінює схожість об’єктів та їх потенційне відношення до одного кластеру за допомогою скалярного значення відстані між точками у багатовимірному просторі. Існує декілька алгоритмів розрахунку відстані. Розглянемо основні.

Евклідова відстань. Найбільш проста функції відстані. Представляє собою геометричне розташування в багатомірному просторі [22]:

$$p(x, x^1) = \sqrt{\sum_{i}^{n} (x_i - x_i^1)^2}$$

Відстань міських кварталів (манхеттенська відстань). Ця відстань є середньою різницею по координатах. У більшості випадків ця міра відстані призводить до таких же результатів, як і для звичайної відстані Евкліда. Однак для цього заходу вплив окремих великих різниць (викидів) зменшується (бо вони не зводяться у квадрат). Формула для розрахунку манхеттенської відстані:

$$p(x, x^1) = \sum_{i}^{n} |x_i - x_i^1|$$

Говерівська відстань, яка була описана Говером у 1971 році [23]. Однією з її переваг є можливість оперувати зі змішаними даними (наприклад, якісні та кількісні ознаки) і може бути описана як середнє арифметичне часткових відмінностей між зразками. Говерівська відстань завжди представляє число від 0, що вказує на повну ідентичність, до 1, що означає, що зразки максимально відрізняються. Метрики для кожного типу даних описані нижче:

– кількісні: манхеттенська відстань, нормована за діапазоном;

– номінальні: змінні k категорій спочатку зводяться до k бінарних колонок після чого застосовується відстань Дайса.

Формула для розрахунку виглядає наступним чином [24]:

$$D = \frac{1}{p}\sum_{j=1}^{p} s_j(x_1, x_2),$$

де p – кількість ознак датасету, а s_j – відстань Дайса або манхеттенська відстань, в залежності від типу даних.

Тепер розглянемо детальніше алгоритми кластеризації, що були обрані у рамках даної роботи.

2.1.1 K-means

Кластеризація k-means – це алгоритм навчання без вчителя, який групує набір даних без міток у різні кластери [25]. Тут k визначає кількість заздалегідь визначених кластерів, які необхідно створити в процесі, наприклад, якщо k=2, буде два кластери, а для k=3 буде три кластери і так далі.

Це ітераційний алгоритм, який розділяє набір даних без міток на k різних кластерів таким чином, що кожен набір даних належить лише одній групі зі схожими властивостями.

Це дозволяє нам кластеризувати дані в різні групи та є зручним способом самостійного виявлення категорій груп у немаркованому наборі даних без необхідності навчання.

K-means – це алгоритм на основі центроїда, де кожен кластер пов'язаний із центроїдом. Основною метою цього алгоритму є мінімізація суми відстаней між точкою даних і відповідними кластерами.

Алгоритм приймає набір даних без міток як вхідні дані, ділить набір даних на k-кількість кластерів і повторює процес, доки не знайде найкращі кластери.

Алгоритм кластеризації k-середніх в основному виконує два завдання:

– визначає найкраще значення для k центральних точок або центроїдів за допомогою ітераційного процесу.

– призначає кожну точку даних найближчому k-центру. Ті точки даних, які знаходяться поблизу певного k-центру, створюють кластер.

Таким чином, кожен кластер має точки даних з деякими спільними рисами, і він знаходиться далеко від інших кластерів.

На діаграмі нижче показана робота алгоритму кластеризації K-середніх (див. рис. 8):

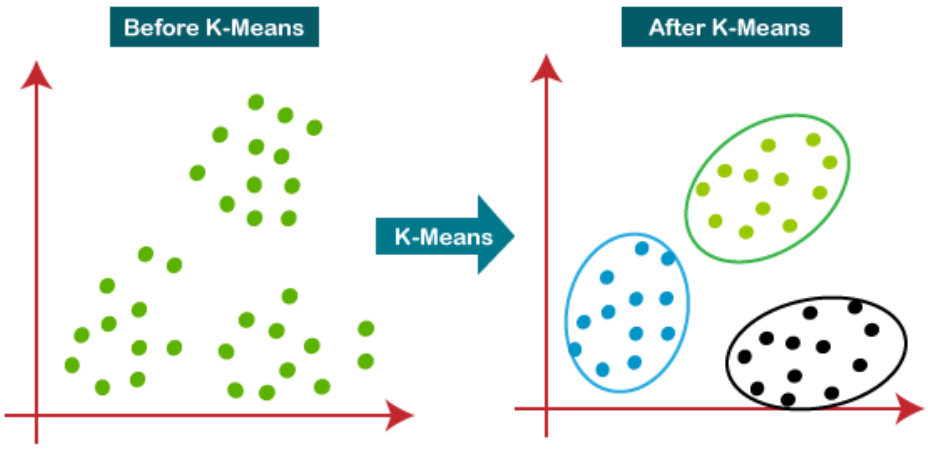

Рисунок 8 – K-means кластеризація

Алгоритм дуже чутливий до стартового розташування центроїдів і може не завжди давати стабільні результати.

2.1.2 BIRCH

Алгоритми кластеризації, такі як кластеризація k-means, не виконують кластеризацію дуже ефективно, і важко обробляти великі набори даних з обмеженою кількістю ресурсів (наприклад, пам’ять або повільніший ЦП). Отже, звичайні алгоритми кластеризації погано масштабуються з точки зору часу роботи та якості, оскільки розмір набору даних збільшується. Саме тут на допомогу приходить кластеризація BIRCH [26]. Це алгоритм кластеризації, який може кластеризувати великі набори даних, спочатку генеруючи невеликий і компактний піднабір великого набору даних, який зберігає якомога більше інформації. Цей менший піднабір потім кластеризується замість кластеризації більшого набору даних.

BIRCH часто використовується для доповнення інших алгоритмів кластеризації шляхом створення піднаборів набору даних, які тепер може використовувати інший алгоритм кластеризації. Однак BIRCH має один істотний недолік – він може обробляти лише метричні атрибути. Метричний атрибут – це будь-який атрибут, значення якого можна представити в евклідовому просторі, тобто категоріальні атрибути не повинні бути присутніми. BIRCH оперує двома важливими термінами: Clustering Feature (CF) і CF-Tree – дерево кластеризації: BIRCH узагальнює великі набори даних у менші щільні області, які називаються Clustering Feature (CF). Формально запис функції кластеризації визначається як упорядкована трійка (N, LS, SS), де «N» – це кількість точок даних у кластері, «LS» – лінійна сума точок даних, а «SS» – квадратна сума точок даних у кластері. Запис CF може складатися з інших записів CF. Дерево CF: Дерево CF є фактичним компактним представленням. Дерево CF – це дерево, у якому кожен листовий вузол містить підкластер. Кожен запис у дереві CF містить покажчик на дочірній вузол і запис CF, що складається із суми записів CF у дочірніх вузлах. У кожному листовому вузлі є максимальна кількість записів. Це максимальне число називається порогом.

Параметрами алгоритму BIRCH є:

– threshold – максимальна кількість точок даних, яку може містити субкластер у листовому вузлі дерева CF.

– branching_factor – цей параметр визначає максимальну кількість підкластерів CF у кожному вузлі (внутрішньому вузлі).

– n_clusters – кількість кластерів, які будуть повернуті після завершення всього алгоритму BIRCH, тобто кількість кластерів після останнього кроку кластеризації. Якщо значення не встановлено, останній крок кластеризації не виконується, і повертаються проміжні кластери.

На основі вказаних параметрів на кожному етапі приймаються рішення про розбиття на підкластери за схемою, зображеною на рисунку 9.

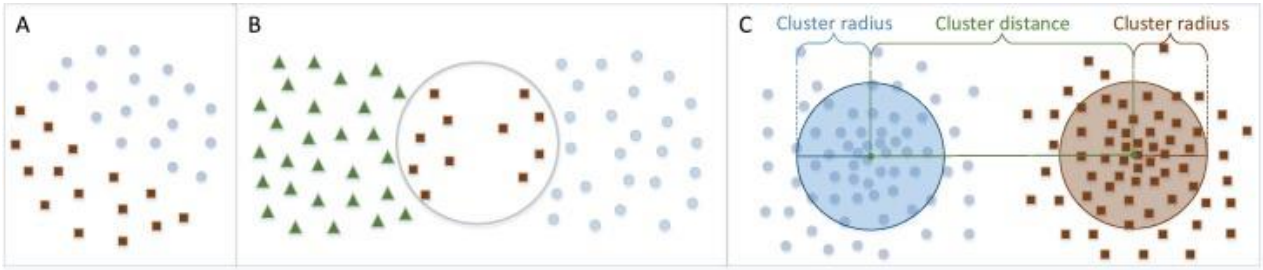

Рисунок 9 – Принцип роботи BIRCH

На кожній ітерації розбиття кластер розбивається таким чином, щоб у отриманих підкластерах максимізувалася кластерна відстань та мінімізувався радіус кластеру.

2.1.3 Agglomerative Clustering

Агломераційна кластеризація, також відома як ієрархічний кластерний аналіз, – це алгоритм, який групує подібні об’єкти в групи, які називаються кластерами [27]. Кінцева точка – це набір кластерів, де кожен кластер відрізняється від одного кластера, а об’єкти в кожному кластері загалом схожі один на одного.

Агломераційну кластеризацію можна виконати за допомогою матриці відстані або необроблених даних. Коли надаються вихідні дані, програмне забезпечення автоматично обчислює матрицю відстані у фоновому режимі. Матриця відстані нижче показує відстань (наприклад, евклідову) між шістьма об’єктами (див. рис. 10).

B	16				
C	47	37			
D	72	57	40		
E	77	65	30	31	
F	79	66	35	23	10
	A	B	C	D	E

Рисунок 10 – Матриця відстаней

Агломераційна кластеризація починається з розгляду кожного спостереження як окремого кластера. Потім він кілька разів виконує наступні два кроки:

– визначає два кластери, які знаходяться найближче один до одного, і

– об’єднує два найбільш схожі кластери.

Цей ітераційний процес триває, доки всі кластери не будуть об’єднані. Основним результатом агломераційної кластеризації є дендрограма, яка показує ієрархічний зв’язок між кластерами (див. рис. 11):

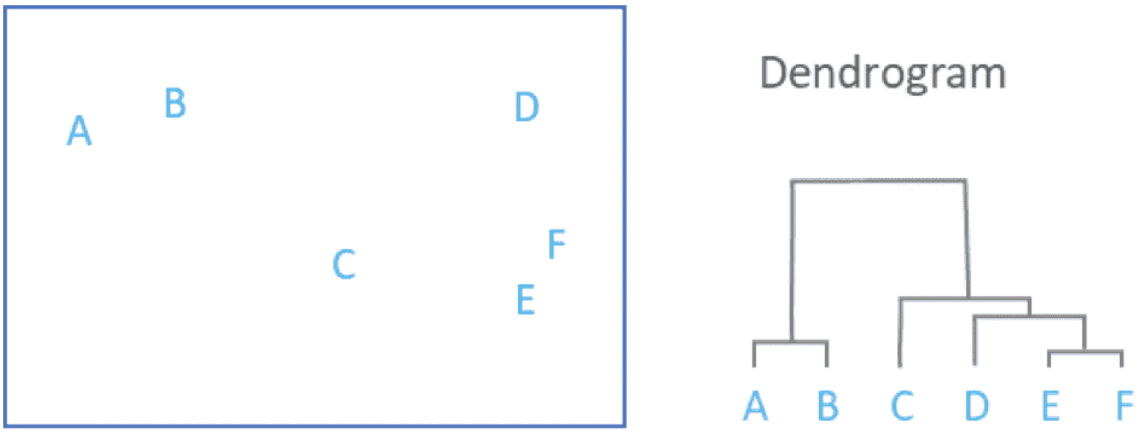

Рисунок 11 – Дендрограма

Таким чином, у нас є можливість створення будь-якої кількості кластерів, відсікаючи дендрограму на необхідній висоті.

2.1.4 DBSCAN

DBSCAN означає просторову кластеризацію додатків із шумом на основі щільності [28]. Він здатний знаходити кластери довільної форми та кластери з шумом (тобто викиди).

Основна ідея DBSCAN полягає в тому, що точка належить кластеру, якщо вона близька до багатьох точок цього кластера.

Існує два ключові параметри DBSCAN:

– eps: відстань, що визначає околиці. Дві точки вважаються сусідніми, якщо відстань між ними менше або дорівнює eps.

– minPts: мінімальна кількість точок даних для визначення кластера.

Завдяки цим параметрам усі точки класифікуються як основні точки, точки прикордонні або викиди:

– основна точка: точка вважається основною, якщо в її оточенні з радіусом eps є принаймні minPts кількості точок (включаючи саму точку).

– прикордонна точка: точка є прикордонною, якщо вона знаходиться у оточенні основної точки, а в її оточенні кількість точок менше minPts.

– викид: точка вважається викидом, якщо вона не є основною точкою і недоступна з будь-якої основної точки.

Ці моменти можна краще пояснити за допомогою візуалізації (див. рис. 12).

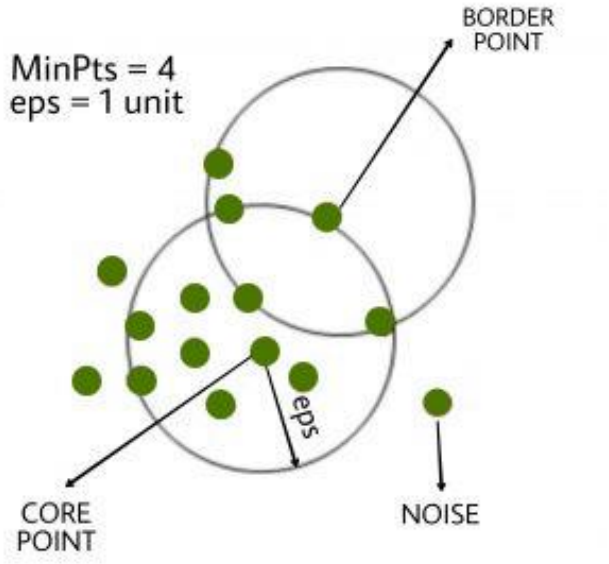

Рисунок 12 – DBSCAN кластеризація

У цьому випадку minPts дорівнює 4. Червоні точки є основними, оскільки в їх оточенні є принаймні 4 точки з радіусом eps. Ця область на малюнку показана колами. Жовті точки є прикордонними, тому що до них можна дістатися з основної точки та мають менше 4 точок у своєму околиці. Досяжність означає перебування в зоні навколо основної точки. Точки B і C мають дві точки (включно з самою точкою) у своєму околиці (тобто навколишній території з радіусом eps). Нарешті, N є викидом, оскільки він не є основною точкою і не може бути досягнутий з основної точки. Тепер можна описати те, як працює алгоритм.

Визначаються minPts і eps. Початкову точку вибирають випадковим чином, її околиці визначають за допомогою радіуса eps. Якщо в околицях є принаймні minPts кількість точок, ця точка позначається як основна і починається формування кластера. Якщо ні, точка позначається як шум. Після початку формування кластера (скажімо, кластер A), усі точки в околицях початкової точки стають частиною

кластера А. Якщо ці нові точки також є основними точками, точки, які знаходяться поблизу них, також додаються до кластер А.

Наступним кроком є випадковий вибір іншої точки серед точок, які не були відвідані на попередніх кроках. Потім застосовується та сама процедура.

Цей процес завершується, коли всі точки відвідані.

Відстань між точками визначається за допомогою методу вимірювання відстані, як у алгоритмі k-середніх. Найпоширенішим методом є евклідова відстань.

Застосовуючи ці кроки, алгоритм DBSCAN може знайти області високої щільності та відокремити їх від областей низької щільності.

Кластер включає основні точки, які є сусідами (тобто доступними одна від одної), і всі прикордонні точки цих основних точок. Необхідною умовою для формування кластера є наявність принаймні однієї центральної точки. Хоча це дуже малоймовірно, ми можемо мати кластер лише з однією основною точкою та її прикордонними точками.

2.1.5 OPTICS

OPTICS (Ordering Points to Identify Cluster Structure) є схожим на DBSCAN алгоритмом, який має трохи інший процес [29]. Він створює діаграму досяжності, яка потім використовується для виділення кластерів. І хоча все ще є вхідні дані, а саме максимальний епсилон, він здебільшого вводиться лише якщо ви хочете спробувати прискорити час обчислення. Інші параметри не мають такого великого впливу, як їхні аналоги в інших алгоритмах кластеризації, і набагато легше використовувати значення за замовчуванням.

Щоб зрозуміти, як він створює діаграму досяжності, треба висвітлити кілька визначень. До концепцій DBSCAN, додається ще кілька визначень:

– відстань ядра - мінімальний епсилон, щоб зробити окрему точку основною точкою, враховуючи кінцевий параметр minPts.

– відстань досяжності - відстань досяжності об'єкта p відносно іншого об'єкта q є найменшою відстанню від q, якщо q є основним об'єктом. Вона також не може бути меншою за відстань осердя q (див. рис. 13).

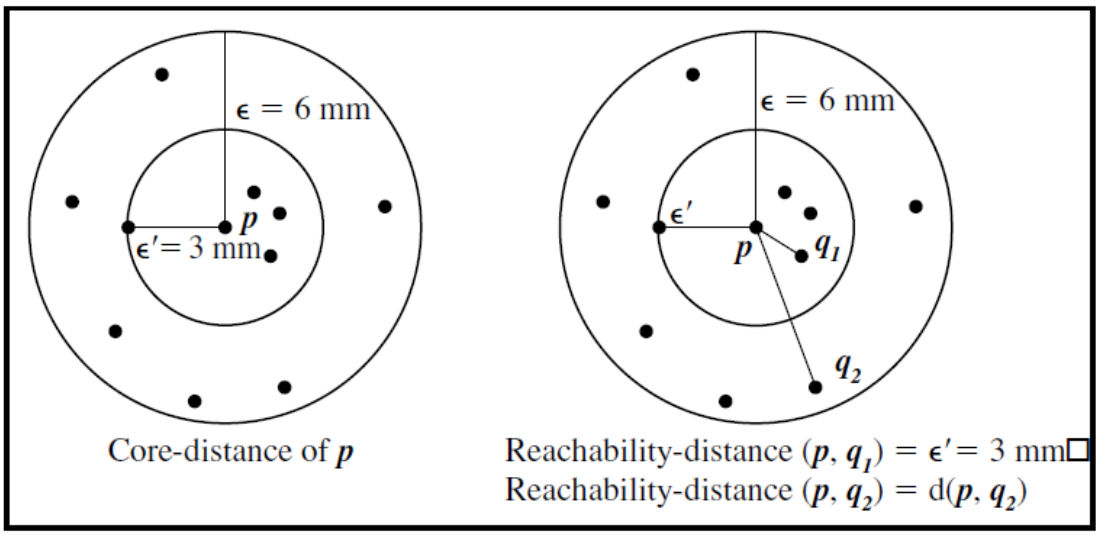

Рисунок 13 – OPTICS кластеризація

Незважаючи на те, що в цих розрахунках використовується параметр minPts, ідея полягає в тому, що він не матиме особливого впливу, оскільки всі відстані будуть масштабуватися приблизно з однаковою швидкістю.

Ми використаємо ці визначення для створення графіка досяжності, який потім використовуватиметься для вилучення кластерів. По-перше, ми починаємо з обчислення основних відстаней для всіх точок даних у наборі. Потім ми прокрутимо весь набір даних і оновимо відстані досяжності, обробляючи кожну точку лише один раз. Ми лише оновлюватимемо відстані досяжності для точок, які потребують покращення, але ще не оброблені. Це тому, що коли ми обробляємо точку, ми встановлюємо її порядок, а також відстань досяжності. Наступною точкою даних, обраною для обробки, буде точка, яка має найближчу відстань досяжності. Таким чином алгоритм утримує кластери поруч один з одним у вихідному порядку.

Наступним кроком буде вилучення фактичних міток кластера з графіка. Найпоширенішим способом зробити це є пошук «долин» на графіку,

використовуючи локальні мінімуми та максимуми. Залежно від обраного методу тут можуть бути використані ще кілька параметрів.

Нижче наведено порівняння деяких згенерованих зразків даних і отриманих оптичних міток і графіка досяжності (див. рис. 14). Кольорові крапки ідентифіковані як кластери, тоді як сірі представляють шум.

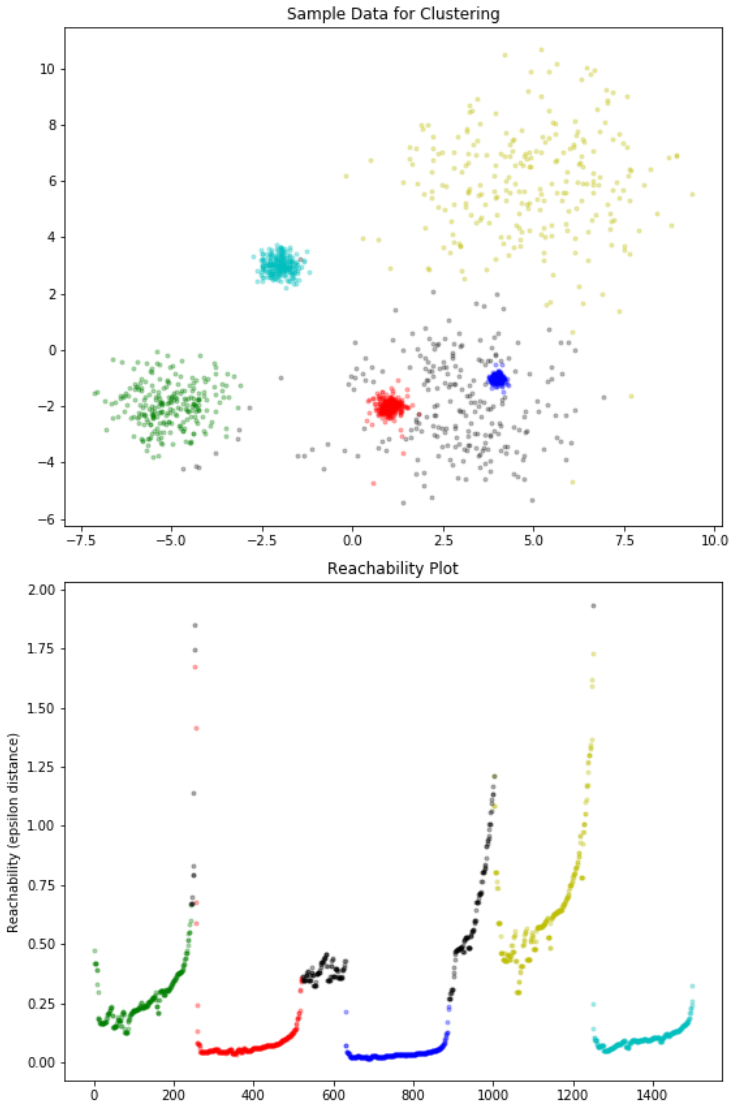

Рисунок 14 – Діаграма досяжності та щільності

Як можна побачити, відсікаючи діагаму досяжності горизонтальною лінією, ми можемо утворити кластери, фільтруючи зайвий шум.

2.2 Огляд алгоритмів розрахунку якості кластеризації

Проблему оцінки якості в задачі кластеризації важко розв'язати, як мінімум, з двох причин:

– теорема неможливості Клейнберга - немає оптимального алгоритму кластеризації;

– багато алгоритмів кластеризації не здатні визначити справжню кількість кластерів у даних. Найчастіше кількість кластерів подається на вхід алгоритму та підбирається декількома запусками алгоритму.

Тому можна лише оцінити якість кластеризації на певному датасеті за існуючими формулами оцінки якості кластеризації, що оцінюють правильність кластеризації на основі щільності, відстані кластерів та однозначності відношення об’єктів до певного кластера.

Прийнято виділяти дві групи методів оцінки якості кластеризації:

– зовнішні (англ. External) методи порівнюють результати кластеризації з заздалегідь відомим поділом на класи.

– внутрішні (англ. Internal) методи відображають якість кластеризації лише за інформацією даних.

Так як у рамках роботи ми не знаємо оптимальну кількість кластерів (клієнтських груп, що значно відрізняються), то використовуватимемо лише внутрішні методи.

2.2.1 Silhouette index

Силуетний аналіз відноситься до методу інтерпретації та перевірки узгодженості в кластерах даних [30]. Величина силуету є мірою того, наскільки об’єкт схожий на власний кластер (щільність) порівняно з іншими кластерами (відокремлення). Його можна використовувати для вивчення відстані розділення між отриманими кластерами. Графік силуету показує, наскільки близько кожна точка в одному кластері до точок у сусідніх кластерах, і, перевіряючи логічність

вибраної кількості кластерів і їх потенційні перекриття. Таким чином він забезпечує візуальну оцінку таких параметрів, як кількість кластерів.

Техніка перевірки силуету обчислює індекс силуету для кожного зразка, середній індекс силуету для кожного кластера та загальний середній індекс силуету для набору даних. Використовуючи підхід, кожен кластер може бути представлений індексом силуету, який базується на порівнянні його щільності та розділення. Якщо значення індексу силуете високе, об’єкт добре збігається з власним кластером і погано збігається з сусідніми кластерами. Коефіцієнт силуету розраховується з використанням середньої відстані між кластерами (a) та середньої відстані до найближчого кластера (b) для кожного зразка. Коефіцієнт силуету визначається як

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

де a(i) – середня відмінність і-го об’єкта від усіх інших об’єктів у тому ж кластері, а b(i) – середня відмінність і-го об'єкта з усіма об'єктами в найближчому кластері.

Діапазон значень силуету – [-1, 1]. Якщо значення силуету наближається до 1, зразок добре згрупований і вже призначений для дуже відповідного кластера. Якщо значення силуету приблизно дорівнює 0, зразок можна призначити іншому кластеру, найближчому до нього, і зразок знаходиться однаково далеко від обох кластерів. Це означає, що це вказує на те, що утворені кластери часто перекриваються, або щільність кластерів занадто низька для чіткої кластеризації. Якщо значення силуету близьке до –1, зразок неправильно класифікується та просто розміщений десь між кластерами, що означає незастосовність даної кластеризації.

2.2.2 Davies–Bouldin index

Для розуміння принципу роботи індексу валідності кластера, потрібно знати про поняття міжкластерної відстані d(a, b) між двома кластерами a, b та внутрішньокластерної відстані D(a) кластера a.

Міжкластерна відстань d(a, b) між двома кластерами a і b може визначатися як:

– відстань одного з’єднання: найближча відстань між двома об’єктами, що належать до a і b відповідно.

– повна відстань зв’язку: відстань між двома найбільш віддаленими об’єктами, що належать до a і b відповідно.

– середня відстань зв’язку: середня відстань між усіма об’єктами, що належать до a і b відповідно.

– відстань зв’язку центроїда: відстань між центроїдом двох кластерів a і b відповідно.

Внутрішньокластерна відстань D(a) кластера a може визначатися як:

– відстань зв’язку повного діаметра: відстань між двома найдальшими об’єктами, що належать кластеру a.

– середня відстань зв’язку діаметра: середня відстань між усіма об’єктами, що належать до кластера a.

– відстань зв’язку діаметра центроїда: подвоєна середня відстань між усіма об’єктами та центроїдом кластера a.

Індекс Дейвіса–Болдіна (ДБ), метрика для оцінки алгоритмів кластеризації, є внутрішньою схемою оцінки, у якій перевірка того, наскільки добре було виконано кластеризацію, виконується з використанням величин і характеристик, властивих набору даних [31].

Індекс Девіса-Болдіна для k кластерів визначається як:

$$DB(U) = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\{\frac{D(X_i)+D(X_j)}{d(X_i,X_j)}\}$$

Чим менше значення індексу ДБ, тим краще кластеризація. Проте у нього також є недолік. Гарне значення, отримане за допомогою цього методу, не означає найкращий пошук інформації.

2.2.3 Calinski-Harabasz Index

Індекс Калінскі-Харабаша (CH) (введений Калінскі та Харабашем у 1974 році) може бути використаний для оцінки моделі, коли базові мітки істинності невідомі, коли перевірка того, наскільки добре було зроблено кластеризацію, здійснюється з використанням кількостей і характеристик, властивих набір даних. Індекс CH (також відомий як критерій співвідношення дисперсії) є показником того, наскільки об'єкт схожий на власний кластер (згуртованість) порівняно з іншими кластерами (відокремленість) [32]. Тут згуртованість оцінюється на основі відстані від точок даних у кластері до його центроїда кластера, а розділення базується на відстані центроїдів кластера від глобального центроїда. Індекс CH має форму (a . відокремленість)/(b . згуртованість) , де a і b є ваговими коефіцієнтами.

Розрахунок індексу Калінскі-Харабаша:

Індекс CH для кількості K кластерів у наборі даних D = [d1, d2, d3, … dN] визначається як,

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

де n_k і c_k є номерами. точок і центроїда k-го кластера відповідно, c – глобальний центроїд, N – загальна кількість точок даних.

Більше значення індексу CH означає, що кластери щільні та добре розділені, хоча немає «прийнятного» граничного значення. Нам потрібно вибрати таке рішення, яке дає пік або принаймні різкий «лікоть» на лінійному графіку індексів CH. З іншого боку, якщо лінія плавна (горизонтальна, висхідна чи спадна), то немає підстав віддавати перевагу одному рішенню перед іншими (див. рис. 15).

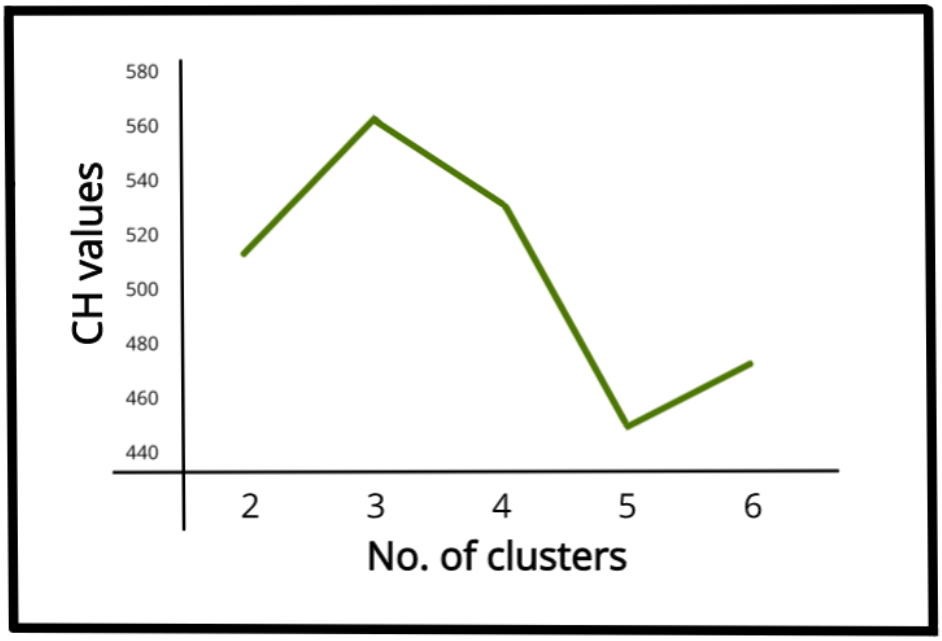

Рисунок 15 – Значення індексу при різній кількості кластерів

До переваг метода можна віднести наступне:

– оцінка вища, коли кластери щільні та добре розділені, що відповідає стандартній концепції кластера.

– оцінка обчислюється швидко.

Недоліками, навпроти, можна вважати те, що індекс Калінскі-Харабаша, як правило, вищий для опуклих кластерів, ніж інші концепції кластерів, наприклад кластери на основі щільності, як ті, що отримані за допомогою DBSCAN.

3 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

3.1 Огляд використаного набору даних

Для проведення експерименту необхідно було обрати датасет, який можна буде використати для тестування алгоритмів кластеризації. Даний датасет повинен був мати велику кількість записів, достатню кількість ознак бінарного та кількісного типу з різними діапазонами. Серед представлених у сервісі Kaggle датасетів за тематикою аналізу клієнтської поведінки було обрано набір даних «Датасет відвідувачів готелю» [33].

Даний датасет зібраний у проміжок часу з 2015 по 2018 рік у Лісабонському отелі.

Набір даних складається з 83 тисяч унікальних записів, 31 ознаки (значущих та незначущих для кластеризації), які будуть описані та продемонстровані далі. З більш детальною інформацією про датасет можна ознайомитися у відповідній статті [33].

Записами датасету є дані про відвідувачів отелю, їх запити та модель поведінки. Ознайомимося з даними детальніше.

У датасеті представлені наступні ознаки:

– «Id» – ID користувача;

– «Nationality» – національність у форматі ISO 3155-3:2013;

– «Age» – вік користувача на момент отримання даних для датасету;

– «DaysSinceCreation» – кількість днів між моментом створення облікового запису користувача;

– «NameHash» – ім’я користувача, зашифроване у форматі SHA2-256 у строку задля не порушення анонімності;

– «DocIDHash» – зашифрований у форматі SHA2-256 наданий користувачем документ при реєстрації;

– «AverageLeadTime» – середня кількість днів між черговим бронюванням номеру у готелі та датою прибуття;

– «LodgingRevenue» – загальна сума, витрачена відвідувачем на проживання (в євро). Ця вартість включає витрати на номер, дитяче ліжечко та інші пов’язані витрати на проживання;

– «OtherRevenue» – загальна сума, витрачена замовником на інші витрати (в євро). Ця вартість включає їжу, напої, спа та інші витрати;

– «BookingsCanceled» – кількість бронювань, зроблені клієнтом, але згодом скасовані (клієнт повідомив готель, що він/вона не приїде зупинятися);

– «BookingsNoShowed» – кількість бронювань, які клієнт зробив, але згодом не з’явився (не скасував, але не зареєструвався, щоб залишитися в готелі);

– «BookingsCheckedIn» – кількість бронювань, за якими клієнт успішно відвідав готель;

– «PersonsNights» – загальна кількість осіб/ночей, які клієнт зупинився в готелі. Це значення розраховується шляхом підсумовування всіх забронюваних клієнтів за кількість осіб/ноч. Особа/ніч у кожному бронюванні є результатом множення кількості ночей проживання на суму дорослих і дітей;

– «RoomNights» – загальна кількість номерів/ночей, які клієнт провів у готелі (зареєстровані бронювання). Кімната/ніч – це множення кількості номерів у кожному бронюванні на кількість ночей у бронюванні;

– «DaysSinceLastStay» – кількість днів, що минула між останнім днем вилучення та датою останнього прибуття клієнта (зареєстрованого бронювання). Значення -1 означає, що клієнт ніколи не зупинявся в готелі;

– «DaysSinceFirstStay» – кількість днів, що минула між останнім днем вилучення та датою першого прибуття клієнта (зареєстрованого бронювання). Значення -1 означає, що клієнт ніколи не зупинявся в готелі;

– «DistributionChannel» – яким сервісом або способом клієнт зазвичай бронює кімнати;

– «MarketSegment» – поточний сегмент ринку замовника;

– «SRHighFloor» – зазначення, якщо клієнт зазвичай просить номер на верхньому поверсі (0: Ні, 1: Так);

– «SRLowFloor» – протилежне попередньому зазначення про номер на нижньому поверсі;

– «SRAccessibleRoom» – зазначення, якщо клієнт зазвичай просить доступну кімнату (0: Ні, 1: Так);

– «SRMediumFloor» – ознака, що клієнт зазвичай просить кімнату на середньому поверсі (0: Ні, 1: Так);

– «SRBathtub» – ознака, що клієнт зазвичай просить кімнату з джакузі (0: Ні, 1: Так);

– «SRShower» – ознака, що клієнт зазвичай просить кімнату з душовою кабиною (0: Ні, 1: Так);

– «SRCrib» – ознака, що клієнт зазвичай просить кімнату з дитячим ліжком (0: Ні, 1: Так);

– «SRKingSizeBed» – ознака, що клієнт зазвичай просить кімнату з великим двохспальним ліжком (0: Ні, 1: Так);

– «SRTwinBed» – ознака, що клієнт зазвичай просить кімнату з двохспальним ліжком (0: Ні, 1: Так);

– «SRNearElevator» – ознака, що клієнт зазвичай просить кімнату біля ліфту (0: Ні, 1: Так);

– «SRAwayFromElevator» – ознака, що клієнт зазвичай просить кімнату подалі від ліфта (0: Ні, 1: Так);

– «SRNoAlcoholInMiniBar» – ознака, що клієнт зазвичай просить кімнату без алкоголя у мінібарі (0: Ні, 1: Так);

– «SRQuietRoom» – ознака, що клієнт зазвичай просить кімнату подалі від галасу та шуму (0: Ні, 1: Так).

Завдяки функціональності сервісу Kaggle є можливість попереднього переглядання розподілу даних по кожній категорії (див. рис. 16):

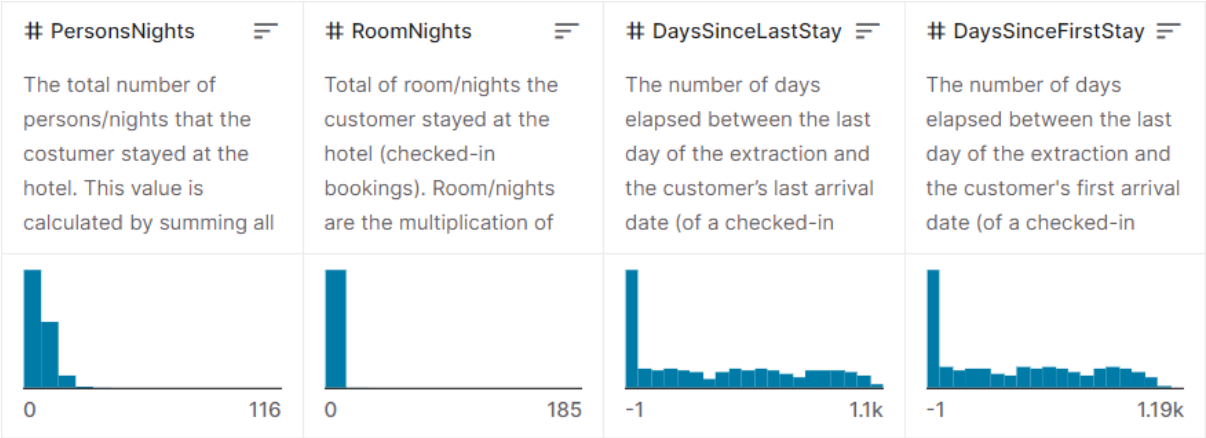

Рисунок 16 – Розподіл значень деяких ознак

Безпосередньо дані датасету у своєму вихідному вигляді виглядають наступним чином (див. рис. 17).

BookingsCheckedIn	PersonsNights	RoomNights	DaysSinceLastStay	DaysSinceFirstStay	DistributionChannel	MarketSegment
3	8	5	151	1074	Corporate	Corporate
1	10	5	1100	1100	Travel Agent/Operator	Travel Agent/Operator
0	0	0	-1	-1	Travel Agent/Operator	Travel Agent/Operator
1	10	5	1100	1100	Travel Agent/Operator	Travel Agent/Operator
0	0	0	-1	-1	Travel Agent/Operator	Travel Agent/Operator
1	4	2	1097	1097	Travel Agent/Operator	Other
0	0	0	-1	-1	Travel Agent/Operator	Other
1	10	5	1100	1100	Travel Agent/Operator	Other
0	0	0	-1	-1	Travel Agent/Operator	Other
1	6	3	1098	1098	Travel Agent/Operator	Travel Agent/Operator
0	0	0	-1	-1	Travel Agent/Operator	Travel Agent/Operator
1	10	5	1100	1100	Travel Agent/Operator	Travel Agent/Operator
0	0	0	-1	-1	Travel Agent/Operator	Travel Agent/Operator
1	8	4	1099	1099	Direct	Direct
0	0	0	-1	-1	Direct	Direct
1	6	3	1098	1098	Travel Agent/Operator	Travel Agent/Operator
1	3	3	1098	1098	Travel Agent/Operator	Travel Agent/Operator
0	0	0	-1	-1	Travel Agent/Operator	Travel Agent/Operator
1	8	4	1099	1099	Travel Agent/Operator	Travel Agent/Operator
0	0	0	-1	-1	Travel Agent/Operator	Travel Agent/Operator
1	4	2	1097	1097	Travel Agent/Operator	Other

Рисунок 17 – Вигляд даних у датасеті

Датасет представляє дані з можливістю RFM-моделі розбиття (давність, частота та грошове значення), яке було описано раніше та чудово підходить для задачі кластеризації завдяки своїй репрезентативності та об’єму та різноманітності даних.

3.2 Хід експерименту

У рамках дослідження було використано мову програмування Python, а саме пакет sklearn для алгоритмів кластеризації та пакет matplotlib для візуалізації графіків.

Першим завданням перед проведенням дослідження є підготовка датасету. Базовий датасет є представником змішаного типу, тобто має як чисельні так і категориальні поля. Такі алгоритми як k-means або BIRCH не застосовні до категоріальних даних, так як вони є дискретними і не представлені натуральними числами. Застосовуючи концепцію центроїдів неможливо порахувати середнє для категоріального поля.

Першою використаною стратегією для вирішення даної проблеми було застосування підходу One Hot Encoding [34], [35]. За цим підходом категоріальна характеристика перетворюється на набір із n бінарних характеристик, де n – кількість унікальних значень даної характеристики у межах датасету. Усі характеристики приймають значення 0 окрім однієї, яка відповідала категоріальному значенню семплу у базовому датасеті. Таким чином, застосувавши даний підхід на досліджуваному датасеті, маємо датасет, що складається виключно з чисельних даних. Окрім цього, з датасету були вилучені незначущі поля, такі як документи, ID та імена. Також для проведення експерименту, задля пришвидшення розрахунків було зменшено кількість семплів до 20000 з 83590 за допомогою випадкової вибірки. Ця вибірка не змінювалась при наступних експериментів з використанням інших методик.

Вигляд модифікації датасету, а саме даних, що використовувались для кластерного аналізу, для першого етапу експерименту із застосуванням підходу One Hot Encoding зображено на рисунку 18.

Nationalit	Nationality__WSM	Nationality__ZAF	Distributio	Distributio	Distributio	DistributionChannel__Travel_Agent	MarketSegment__Aviation	MarketSegment__Complementary	Marke
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	1	0	0	0	0	
0	0	0	1	0	0	0	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	1	0	0	0	0	
0	0	0	0	1	0	0	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	1	0	0	

Рисунок 18 – Вигляд модіфікованого датасету

При розробці маркетингових стратегій першим чином постає питання кількості кластерів, на які будуть розбиватися клієнти. Цей крок є одним із найважливіших, адже при замалій кількості кластерів є ризик охопити занадто багато несхожих за своєю поведінкою клієнтів. Це може спричинити недостатньо індивідуальний підхід, який не повністю враховує купівельну спроможність клієнта або не відповідає специфічним потребам. Таким чином, головним завданням є знаходження оптимальної кількості кластерів, при якій вони будуть мінімально між собою накладатися та бути щільними (групувати схожих за властивостями клієнтів). Є техніки, які дозволяють визначити оптимальну кількість кластерів при використанні будь-якого алгоритму кластеризації. На відміну від k-means, BIRCH та Агломераційної кластеризації, OPTICS та DBSCAN не приймають у якості параметру кількість кластерів, а розраховують їх оптимальну кількість самостійно. Тому для вибору оптимальної кількості кластерів було обрано алгоритм DBSCAN. Для його роботи необхідно задати два параметри – ε та min_samples. Другий параметр може бути встановлений рівним кількості характеристик, тому важливішим є визначення параметру ε. Було обрано алгоритм, який застосовує поняття K-Neighbors [36]. Ця техніка обчислює середню відстань між кожною точкою та її k найближчими сусідами, де k – це min_samples, який було обрано раніше. Потім середні значення k-відстаней відображаються в порядку зростання на графіку k-відстаней. Оптимальне значення для ε можна знайти в точці

максимальної кривини (тобто там, де графік має найбільший нахил). Ця точка представляє точку оптимізації, коли зменшення прибутку більше не варте додаткових витрат. Після застосування підходу на датасеті було побудовано наступний графік (див. рис. 19).

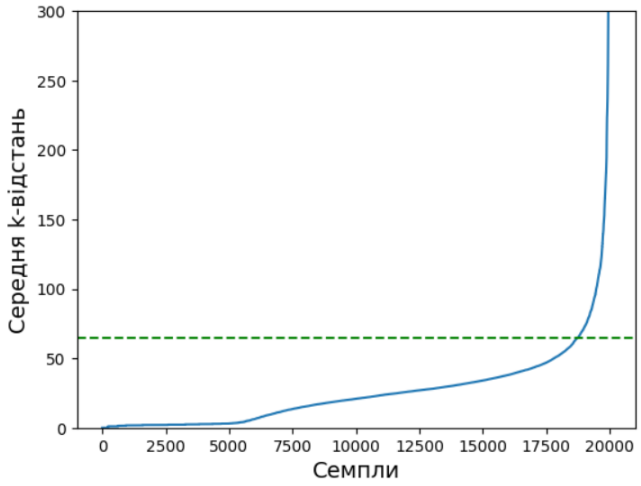

Рисунок 19 – Графік K-Neighbors

Виходячи з графіку, оптимальним значенням є є 65. Було підставлено отримані значення у якості параметрів DBSCAN та була отримана кількість кластерів рівна 4. Дане значення є цілком задовільним при визначенні кількості потенційно різних стратегій. Задля більш точного експерименту було проведено порівняння алгоритмів кластеризації при кількості кластерів рівної 3, 4, 5. Для алгоритму DBSCAN та OPTICS, які не приймають кількість кластерів у якості параметру, було запущено декілька кластеризацій з діапазоном є [35,80] та min_samples [28, 56].

Для кожного запуску кластеризації було знайдено індекс силуету – одну із метрик якості кластеризації без «ground truth» поміток (правильного в умовах датасету розділення на кластери). З отриманих результатів було обрано найкращі результати для кількості кластерів 3, 4, 5 за значеннням метрики. Для алгоритмів k-means, BIRCH та Агломераційної кластеризації було проведено по 3 запуски для кількості кластерів 3, 4 та 5. Для кожного результату кластеризації у експерименті було розраховано індекс Девіса-Болдіна (див. табл. 1).

Таблиця 1 – Значення індексу Девіса-Болдіна

Кількість кластерів	Алгоритм				
	K-means	BIRCH	Agglomerative clustering	DBSCAN	OPTICS
3	0.829	0.691	0.152	2.325	1.590
4	0.842	0.843	0.367	1.389	1.671
5	0.865	0.889	0.537	1.415	1.698

Як видно із отриманих результатів, DBSCAN та особливо OPTICS показують значно гірші результати через особливість своєї роботи та датасету. Дані датасету у багатовимірному просторі знаходяться достатньо щільно і не утворюють чітко виділені кластери. Через це, а також через застосований до датасету метод One Hot Encoding призводить до утворення одного великого кластера, який містить більшість семплів. Через це розділення датасету на 3, 4 або більше кластерів стає важким завданням для щільнісних алгоритмів. Тепер для кожного алгоритму знайдемо значення силуету (див. табл. 2).

Таблиця 2 – Значення індексу силуету

Кількість кластерів	Алгоритм				
	K-means	BIRCH	Agglome-rative clustering	DBSCAN	OPTICS
3	0.527	0.510	0.888	0.135	-0.457
4	0.439	0.424	0.705	-0.070	-0.540
5	0.426	0.404	0.593	-0.082	-0.558

Для повноти картини також знайдемо показники індексу Калінскі-Харабаша (див. табл. 3).

Таблиця 3 – Значення індексу Калінскі-Харабаша

Кількість кластерів	Алгоритм				
	K-means	BIRCH	Agglomerative clustering	DBSCAN	OPTICS
3	16651	14513	713	900	867
4	16365	14084	1007	791	720
5	15736	14255	794	611	600

Отримані результати показують значну перевагу k-means та BIRCH, хоча це є нормальним показником через специфічність роботи метрики, де результати роботи алгоритмів із концепцією центроїдів завжди показують кращі результати ніж інші алгоритми. Через це у подальшому даній метриці не приділялося багато уваги, зосередившись на індексі Девіса-Болдіна та індексу силуету, що давали більш репрезентативні значення та дозволяють оцінити якість отриманих кластеризацій.

Для покращення роботи алгоритмів кластеризації DBSCAN та OPTICS було вирішено використати відстань Говера замість Евклідової. Вищезгадані два алгоритми, так само, як Агломераційна кластеризація можуть приймати у якості вхідних даних матрицю досяжностей, у яких можна використати відстань для змішаних даних. Для цього базовий датасет зазнав наступних перетворень: поля «сегмент маркету», «країна», «канал розповсюдження» були помічені як категоріальні, так само як 13 бінарних полів «special requirements» – особливих вимог, які у датасеті мали бінарний тип даних.

Як і у першій модифікації, з датасету були вилучені незначущі поля, такі як документи, ID та імена. Усі відстані були автоматично нормалізовані у діапазоні [0,1] за допомогою використаної бібліотеки. Після вказаних перетворень датасет набув наступного вигляду (див. рис. 20).

	A	B	C	D	E	F	G	H	I	J
1	0	0,222581	0,333971	0,284171	0,322627	0,277949	0,323761	0,295337	0,350988	0,301594
2	0,222581	0	0,2133	0,071147	0,211031	0,120219	0,256409	0,140739	0,276828	0,086652
3	0,333971	0,2133	0	0,235742	0,057856	0,245095	0,103233	0,27542	0,122518	0,210939
4	0,284171	0,071147	0,235742	0	0,177886	0,130181	0,272044	0,118381	0,252758	0,081301
5	0,322627	0,211031	0,057856	0,177886	0	0,233751	0,094158	0,239118	0,074872	0,222283
6	0,277949	0,120219	0,245095	0,130181	0,233751	0	0,141862	0,118862	0,2156	0,134542
7	0,323761	0,256409	0,103233	0,272044	0,094158	0,141862	0	0,237984	0,073738	0,267661
8	0,295337	0,140739	0,27542	0,118381	0,239118	0,118862	0,237984	0	0,164246	0,167298
9	0,350988	0,276828	0,122518	0,252758	0,074872	0,2156	0,073738	0,164246	0	0,286946
10	0,301594	0,086652	0,210939	0,081301	0,222283	0,134542	0,267661	0,167298	0,286946	0
11	0,33624	0,215569	0,04878	0,238011	0,060125	0,247364	0,105502	0,277689	0,124787	0,162159
12	0,266805	0,05674	0,222676	0,017366	0,16482	0,117115	0,258977	0,101015	0,239692	0,095778
13	0,325463	0,213868	0,060692	0,17505	0,002836	0,233184	0,096994	0,241955	0,077708	0,22512
14	0,279018	0,158446	0,312218	0,160892	0,311084	0,173759	0,30995	0,20241	0,329235	0,163725

Рисунок 20 – Модифікація для другого етапу експерименту

Як і у першій частині експерименту, визначимо оптимальне значення є для алгоритму DBSCAN за допомогою алгоритму K-Neighbors. Отримуємо наступні результати (див. рис. 21).

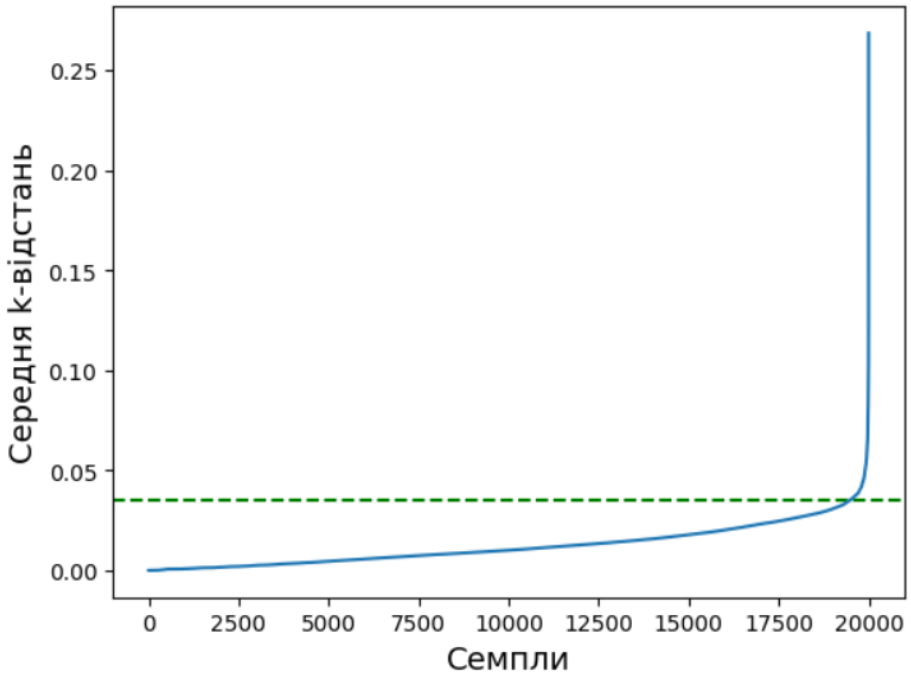

Рисунок 21 – Графік K-Neighbors для другого експерименту

Як і на першій модифікації датасету, було встановлено оптимальні значення є у околі значення 0.035 та min_samples для створення 3, 4 та 5 кластерів. Для

кожного результату кластеризації було визначено лише індекс силуету, так як індекс Девіса-Болдіна та Калінскі-Харабаша не розраховується із матрицею досяжності. Отримані результати зображено на таблиці 4.

Таблиця 4 – Значення індексу силуету для другого експерименту

Алгоритм	Кількість кластерів		
	3	4	5
Агломераційна кластеризація	0.888	0.705	0.593
Агломераційна кластеризація (Відстань Говера)	0.627	0.497	0.335
DBSCAN	0.135	-0.070	-0.082
DBSCAN (Відстань Говера)	0.311	0.202	0.155
OPTICS	-0.457	-0.540	-0.558
OPTICS (Відстань Говера)	0.145	0.107	0.073

Плюсом використання відстані Говера є те, що при її використанні є можливість задання вагових коефіцієнтів до кожної з харатеристик. Розділимо усі характеристики, наявні у датасеті на дві категорії – більш важливі та менш важливі. Виходячи із стандартної бізнес логіки, до менш важливих характеристик можна віднести країну походження клієнта та усі «Special requirements». Щоб не змінювати занадто сильно вхідні дані датасету, а просто перевірити застосовність даної можливості, для описаних вище характеристик встановимо значення 1, а для усіх інших – 2. Після проведення експерименту маємо наступні результати (див. табл. 5).

Таблиця 5 – Значення індексу силуету для другого експерименту

Алгоритм	Кількість кластерів		
	3	4	5
Агломераційна кластеризація	0.888	0.705	0.593
Агломераційна кластеризація (Відстань Говера з коефіцієнтами)	0.587	0.491	0.302
DBSCAN	0.135	-0.070	-0.082
DBSCAN (Відстань Говера з коефіцієнтами)	0.368	0.215	0.145
OPTICS	-0.457	-0.540	-0.558
OPTICS (Відстань Говера з коефіцієнтами)	0.129	0.112	0.086

Тепер, після проведення експерименту та отримання усіх можливих значень для кожної з варіацій датасету та використаних алгоритмів, можна за допомогою діаграм більш наглядно порівняти результати та зробити висновки.

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Для аналізу отриманих результатів відобразимо отримані раніше значення за допомогою графіків. Спочатку проаналізуємо значення коефіцієнту Девіса-Болдіна, отримані під час першої частини експерименту (див. рис. 22).

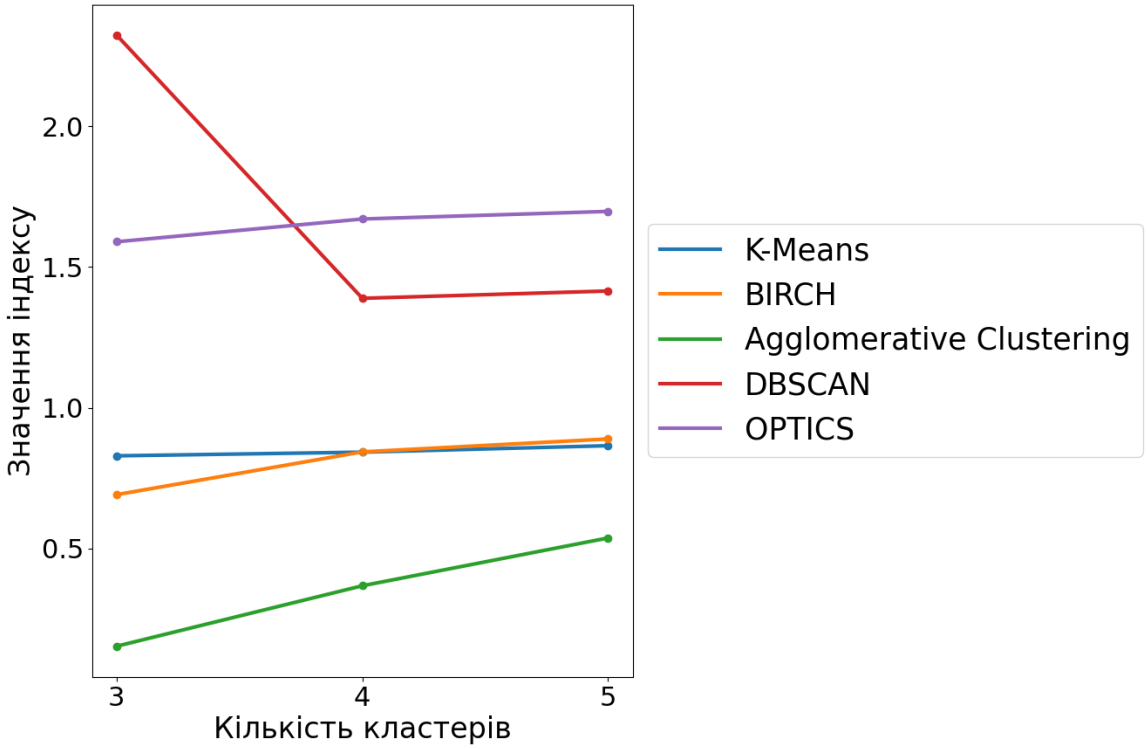

Рисунок 22 – Значення індексу Девіса-Болдіна

Беручи до уваги, що індекс Девіса-Болдіна показує відношення відстані між семплами та центроїдами до відстані між центроїдами різних кластерів, менше значення означає кращий результат. Отримані результати проілюстровані на графіку 1. На графіку видно, що найкращий результат показав алгоритм агломераційної ієрархічної кластеризації. Далі розташовані k-means та BIRCH. Останній показує дещо кращі результати при кількості кластерів меншій за 4, після чого вирівнюється з k-means. DBSCAN та особливо OPTICS показують значно гірші результати через особливість своєї роботи та датасету. Ці алгоритми тяжіють до виділення одного великого кластеру, що авжеж погано підходить для вирішення

поставленої задачі. На рисунку 23 зображено графік зі значеннями індекса силуету першої частини експерименту.

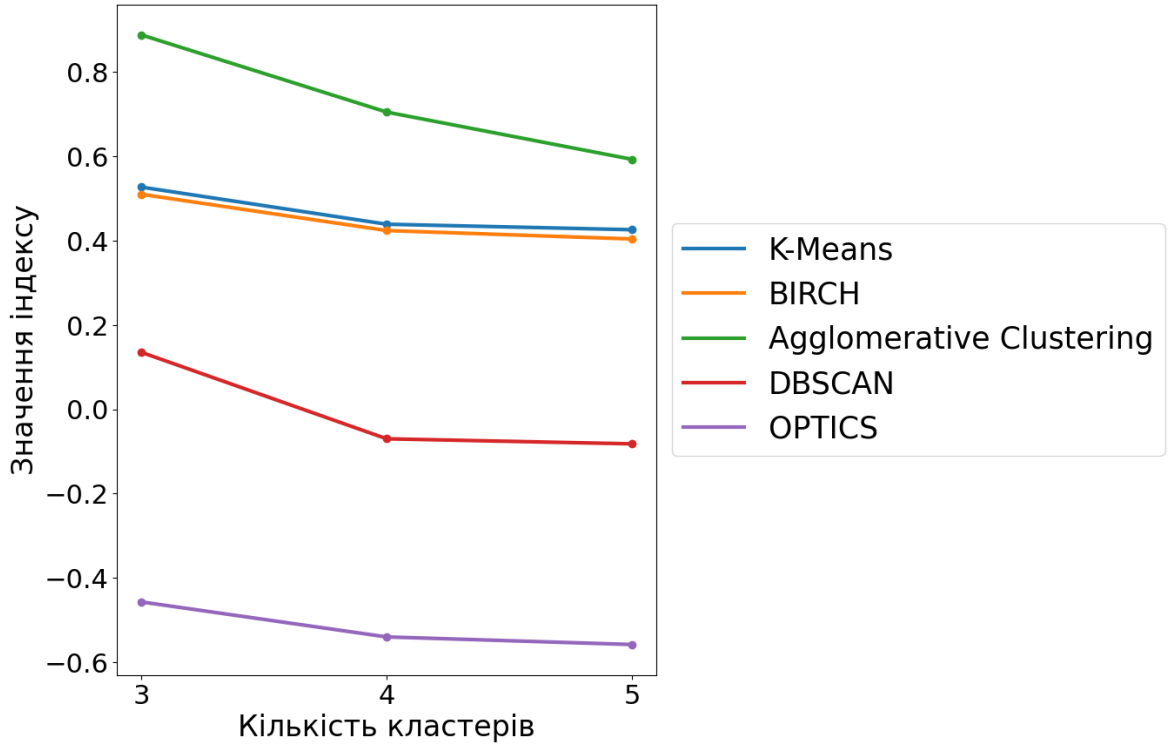

Рисунок 23 – Значення індексу силуету

На таблиці зображено розраховані значення силуету, що показує відношення відстаней між семплами одного кластеру до семплів інших кластерів. Отримуємо підтвердження отриманих висновків щодо неефективності щільнісних алгоритмів по відношення до інших. При цьому k-means та BIRCH показують майже ідентичні результати, як і з попередньою метрикою. Агломераційна кластеризація показує майже ідеальні результати при кількості кластерів, рівній 3, та в цілому у всіх замірах є вибором номер один.

Останньою метрикою, яка була розрахована для отриманих кластеризацій був індекс Калінскі-Харабаша. Його значення для кожного алгоритму наведені на рисунку 24.

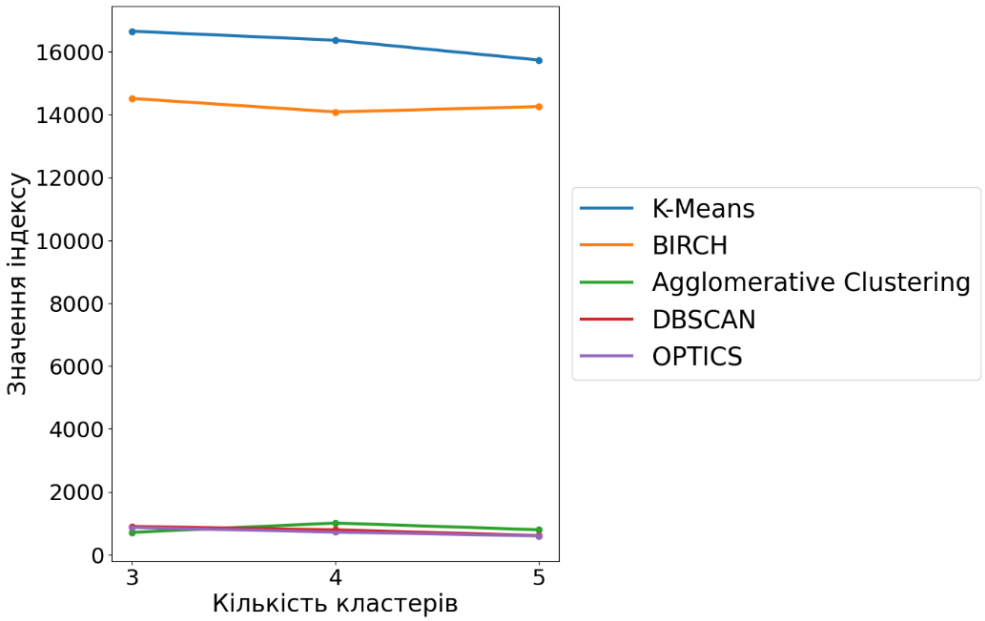

Рисунок 24 – Значення індексу Калінскі-Харабаша

Отримані результати є найменш репрезентативними через специфіку розрахунку метрики, у якій щільнісні та ієрахічні алгоритми завжди програють центроїдним алгоритмам. З даного графіку можна зробити єдиний висновок про невелику перевагу k-means над BIRCH, що підтверджує вигляд попередніх графіків.

Другою частиною експерименту було порівняння роботи трьох алгоритмів – агломераційної кластеризації, DBSCAN та OPTICS при використанні Евклідової та Говерівської відстані. Так як відстань Говера спеціально призначена для кластеризації датасетів змішаного типу, другий екстперимент було проведено до усіх алгоритмів, що будують матрицю досяжності семплів, а отже не залежать від алгоритму розрахунку відстані. У теорії, застосування цієї відстані мало позитивно вплинути на використані алгоритми.

Для того, щоб зобразити отримані табличні дані більш репрезентативно було побудовано стовпчасті діаграми, які показували кількісний виграш від використання іншої метрики відстані порівняно із експериментом з Евклідовою відстанню. У результаті маємо наступну діаграму (див. рис. 25):

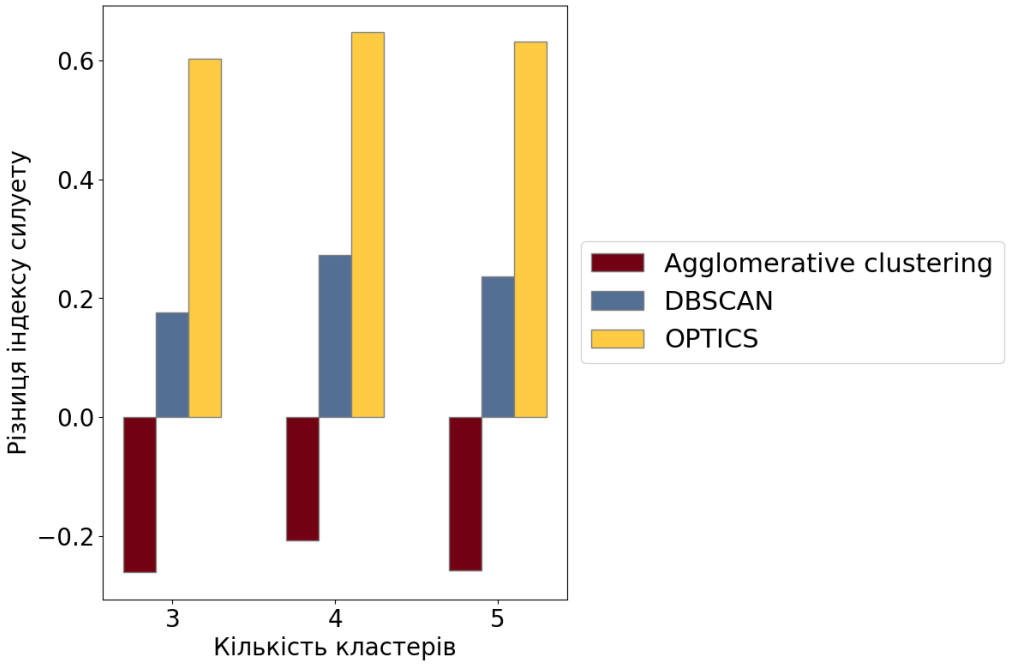

Рисунок 25 – Виграш від використання Говерівської відстані

Як можна побачити на діаграмі, всупереч очікуванням, агломераційна кластеризація показує дещо гірші результати (на 0.242 у середньому у абсолютних значеннях), порівняно із першим експериментом. Цілком ймовірно, що через метод One-Hot Encoding агломераційна кластеризація майже не використовує поля національності та специфічних запитів (Special requirements) та, зосереджуючись на основних ознаках клієнтів, краще відокремлює клієнтські групи. У будь-якому випадку, навіть отримані значення перевищують відповідні значення k-means та BIRCH, а отже алгомераційна кластеризація все ще показує найкращі результати.

DBSCAN, натомість, помітно покращує свої результати (на 0.229 у середньому у абсолютних значеннях), що доводить ефективність використання Говерівської відстані для щільнісних алгоритмів.

OPTICS, як алгоритм із найгіршими результатами у першому експерименті, що робили його зовсім незастосовним для вирішення поставленої задачі, отримав найбільший приріст значення, порівняно з іншими алгоритмами (на 0.626 у середньому у абсолютних значеннях).

Значення силуету, більшу за 0 (а при другому експерименті тепер усі алгоритми перетнули цю позначку), доводять можливість застосування

відповідного алгоритму. Тим не менш, це не впливає на розміщенні алгоритмів у списку, відсортованих за значенням силуету – за агломераційною кластеризацією йдуть k-means та BIRCH.

Тепер відобразимо на діаграмі отриманий виграш від використання Говерівської відстані з ваговими коефіцієнтами, що можуть допомогти утворити більш розділені кластери. Вагові коефіцієнти були змінені незначно, адже рішення про надання певних коефіцієнтів приймається відповідно до вимог бізнесу, тому і отримані результати відрізняються не дуже сильно, що можна побачити на рисунку 26.

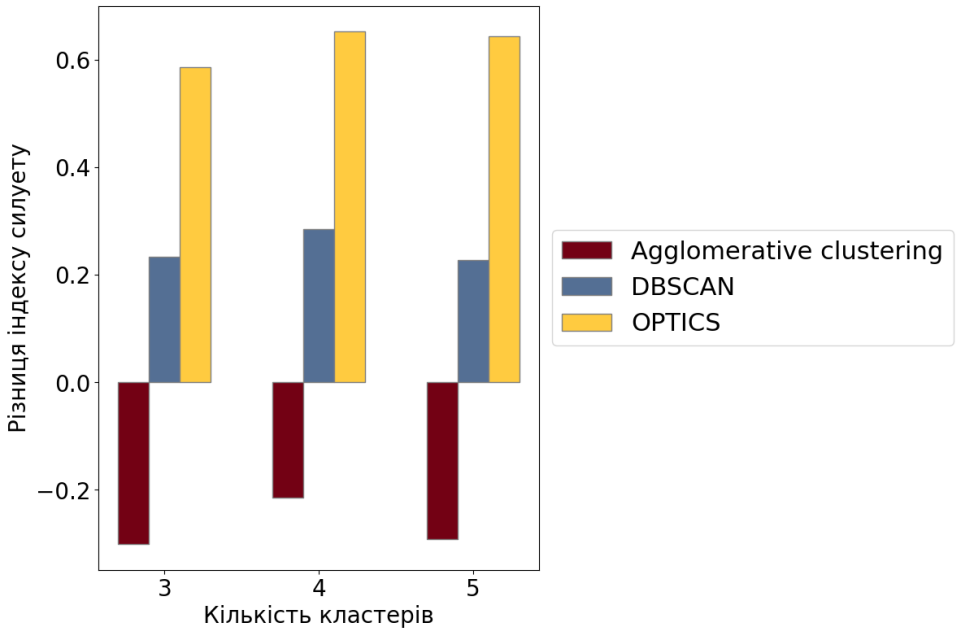

Рисунок 26 – Виграш від використання Говерівської відстані з ваговими коефіцієнтами

Як можна побачити, отримана діаграма майже не відрізняється від діаграми з рисунку 25. Маємо середній приріст значення силуету для DBSCAN на 0.249, OPTICS на 0.627 та спадання значення для агломераційної кластеризації на 0.268. Щоб мати повну картину та отримати краще візуальне представлення для отриманих результатів, зобразимо їх за допомогою теплової карти (див. рис. 27).

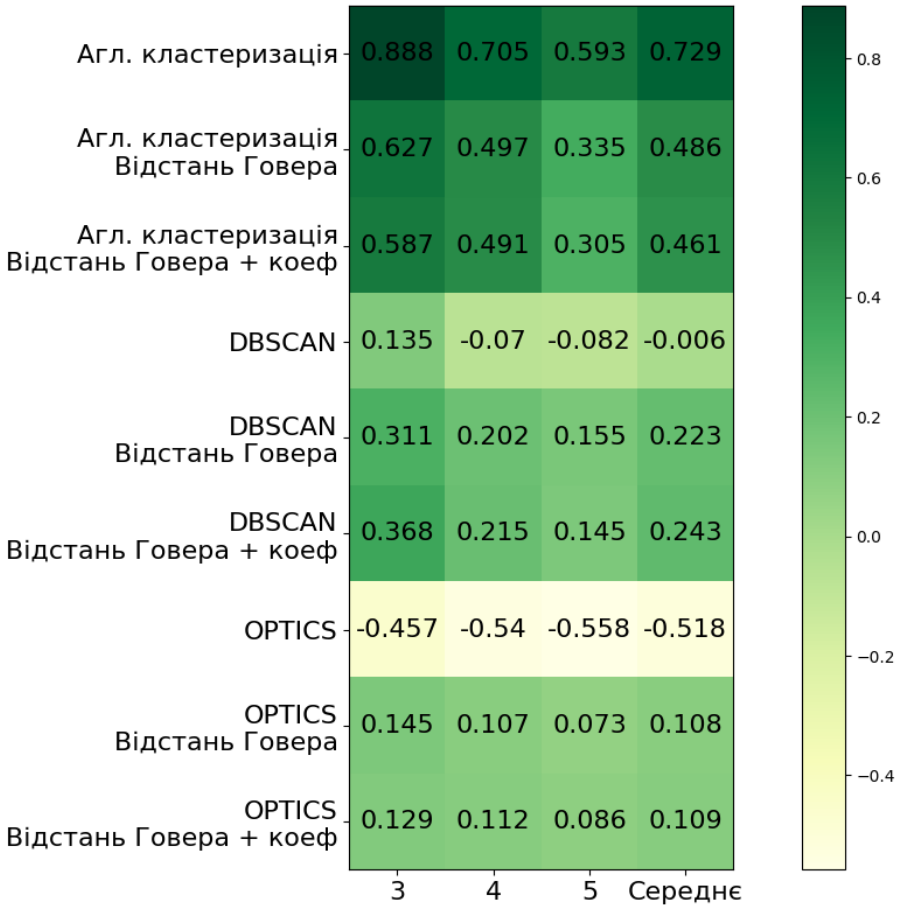

Рисунок 27 – Теплова карта трьох частин експерименту для алгоритмів

Як можна побачити на отриманій діаграмі, агломераційна кластеризація показує найкращі результати при першому експерименті та не отримує користі від використання відстані Говера, тим не менш залишаючись найкращим алгоритмом на обраному датасеті. DBSCAN має значно кращі значення у другому та особливо третьому експерименті. Алгоритм починає показувати конкурентоспроможні результати порівняно з k-means та BIRCH і може використовуватися для вирішення задачі виділення клієнтських груп. Алгоритм OPTICS отримує найбільший виграш при використанні відстані Говера (та мінімальний при використанні вагових коефіцієнтів), але все ще показує значення, недостатні, щоб бути альтернативою іншим методам, окрім того завжди забираючи приблизно у 5 разів більше часу ніж будь-який інший алгоритм на процес кластеризації.

5 ПРОГРАМНА РЕАЛІЗАЦІЯ

Під час створення програмного застосунку було використано мову програмування Python. Приклади основного створеного коду наведено нижче.

Імпорт необхідних бібліотек:

```
import csv
import numpy as np
import random
import pandas as pd
from collections import Counter
from sklearn.preprocessing import LabelBinarizer
```

Вичитування датасету та застосування підходу One-Hot Encoding для першої частини експерименту:

```
def column_one_hot_encoder(dataframe, field):
    jobs_encoder = LabelBinarizer()
    jobs_encoder.fit(dataframe[field]) # створюємо колекцію бінарних полів (розгорнуте представлення категоріального поля)
    transformed = jobs_encoder.transform(dataframe[field])
    ohe_df = pd.DataFrame(transformed, columns=[f'{field}__{item.replace(" ", "_").replace("/", "-")}' for item in jobs_encoder.classes_]) # створюємо датасет із отриманих полів
    return pd.concat([dataframe, ohe_df], axis=1).drop([field], axis=1) # створюємо результуючий датасет

def one_hot_encode(nrows_param):
    df = pd.read_csv("HotelCustomersDataset.csv", nrows=nrows_param) # зменшуємо кількість семплів датасету до необхідної кількості (20000 у нашому випадку)
    df = df.drop(['ID', 'NameHash', 'DocIDHash'], axis=1) # видаляємо незначущі поля
    df = column_one_hot_encoder(df, 'Nationality') # застосовуємо підхід до категоріальних полів
    df = column_one_hot_encoder(df, 'DistributionChannel')
    df = column_one_hot_encoder(df, 'MarketSegment')
    df.to_csv(f'HotelDataset_OneHotEncoded_{nrows_param}.csv', index=False) # повертаємо датасет у файл для подальшого аналізу
```

Робимо перетворення датасету для другого експерименту:

```
def binaryToCategorial(list):
    return ['yes' if int(i)==1 else 'no' for i in list]
```

```
def gower_transform(n_rows):
    with open(f'HotelDataset_{n_rows}.csv') as csv_file: # вичитуємо
датасет
        csv_reader = csv.reader(csv_file, delimiter=',')
        train_list = [row for row in csv_reader]
        headers = train_list[0] # відокремлюємо загловки
        train_list = train_list[1:]
        train_list = [row[:-13] + binaryToCategorial(row[-13:]) for row in
train_list] # перетворюємо числові значення 0/1 у ні/так для автоматичного
розпізнавання поля як категоріального
        write_to_file(f'HotelDataset_{n_rows}_gower.csv', headers,
train_list) # записуємо результуючий датасет у файл
```

Імпорт бібліотек для проведення експерименту:

```
import gower
from collections import Counter
from sklearn.neighbors import NearestNeighbors
from sklearn.cluster import DBSCAN, OPTICS, AgglomerativeClustering,
KMeans, Birch
import plotly.express as px
from sklearn.metrics import davies_bouldin_score, calinski_harabasz_score,
silhouette_score
from matplotlib import pyplot as plt
```

Код для застосування методу K-Neighbors для знаходження оптимального епсилону:

```
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(X) # створюємо матрицю досяжності
distances, indices = nbrs.kneighbors(X)
distances = np.sort(distances, axis=0)
distances = distances[:,1] # отримуємо відсортований список відстаней
plt.plot(distances) # відображуємо на графіку
plt.axhline(y=65, color='g', linestyle='--') # будуємо паралельну лінію для
відображення точки найбільшої кривизни
plt.ylim([0, 300])
plt.grid()
plt.show()
```

Код для розрахунку метрик якості кластеризації для алгоритмів, що приймають кількість кластерів у якості параметру:

```
for cluster_num in [3,4,5]:
    model_complete = KMeans(n_clusters=cluster_num) # використовуємо одну із
наступних 3-х моделей
    model_complete = Birch(threshold=0.07, n_clusters=cluster_num)
    model_complete = AgglomerativeClustering(n_clusters=cluster_num,
linkage='complete')
    agglomerative = model_complete.fit_predict(X) # кластеризуємо датасет
```

```
    ag_index = silhouette_score(X, agglomerative) # розрахунок індексу
силуету
    print("silhouette_score:", ag_index)
    ag_index = calinski_harabasz_score(X, agglomerative) # розрахунок
індексу Калінскі-Харабаша
    print("calinski_harabasz_score:", ag_index)
    ag_index = davies_bouldin_score(X, agglomerative) # розрахунок індексу
Девіса-Болдіна
    print("davies_bouldin_score:", ag_index)
```

Код для визначення оптимальних значень епсилону та minPts для DBSCAN:

```
listi=[]
for eps in range(60,270,30): # перебираємо значення епсилону
    for min_s in range(20, 41, 10): # перебираємо значення minPts
        dbscan = DBSCAN (eps=eps, min_samples=min_s).fit_predict(X) #
отримуємо кластеризований датасет
        dbscan_clusters_ = len(set(dbscan)) - (1 if -1 in dbscan else 0) #
відсіюємо викиди
        print(DBSCAN eps   ',eps)
        if dbscan_clusters_ >= 3:
            db_index = silhouette_score(X, dbscan) # визначаємо значення
силуету для кластеризацій з кількістю кластерів, більшою за 2
            listi.append(Object(_eps=eps, _min_s=min_s, score=db_index,
num_clusters= dbscan_clusters_))
print('Result')
listi.sort(key=lambda x: x.score, reverse=True) # сортуємо отримані
кластеризації за значенням силуету
from collections import defaultdict
res = defaultdict(list)
print('--------------Final results---------------------')
for item in listi:
    if item.num_clusters not in res:
        res[item.num_clusters] = item # записуємо найкращий результат для
кожної кількості кластерів
for row in [res[3], res[4], res[5]]: # отримуємо вхідні параметри та
відповідні значення силуету для кожної кількості кластерів
    print('Res')
    print('Score', row.score)
    print('Cluster num', row.num_clusters)
    print('Mins', row._min_s)
    print('Eps', row._eps)
```

Код для розрахунку значень якісних метрик кластеризації для щільнісних алгоритмів:

```
dbscan = DBSCAN(eps=110, min_samples=40).fit_predict(X) # використовуємо
необхідний алгоритм
optics = OPTICS(eps=eps, min_samples=min_s).fit_predict(X)
dbscan_clusters_ = len(set(dbscan)) - (1 if -1 in dbscan else 0) #
фільтруємо викиди
```

```
print('DBSCAN clusters   ',dbscan_clusters_)
db_index = silhouette_score(X, dbscan) # знаходимо значення метрик
print("silhouette_score:", db_index)
db_index = calinski_harabasz_score(X, dbscan)
print("calinski_harabasz_score:", db_index)
db_index = davies_bouldin_score(X, dbscan)
print("davies_bouldin_score:", db_index)
```

Код для створення матриці досяжності з відстанню Говера:

```
def create_distance_matrix(n_rows):
    df = pd.read_csv(f"HotelDataset_{n_rows}_gower.csv") # вичитуємо
датасет
    X = np.asarray(df) # перетворюємо датасет у двовимірний масив
    weights = np.asarray([1] + [2] * 14 + [1] * 13) # встановлюємо або не
встановлємо вагові коефіцієнти
    #weights = np.asarray([1] * 28)
    categorial_properties = np.asarray([True] + [False] * 12 + [True] * 15)
# помічаємо категоріальні дані
    gower_matrix = gower.gower_matrix(data_x=X, weight=weights,
cat_features=categorial_properties) # будуємо матрицю досяжностей
    return gower_matrix
```

Використання отриманої матриці при роботі з алгоритмами:

```
dbscan = DBSCAN(eps=0.04, min_samples=20,
metric="precomputed").fit_predict(distance_matrix) # встановлюєсо, що
використовуємо матрицю досяжностей при роботі з DBSCAN
dbscan_clusters_ = len(set(dbscan)) - (1 if -1 in dbscan else 0) #
знаходимо викиди
print('DBSCAN clusters   ',dbscan_clusters_)
db_index = silhouette_score(distance_matrix, dbscan, metric='precomputed')
# встановлюєсо, що використовуємо матрицю досяжностей при роботі з індексом
силуету
print("Davies-Bouldin index DBSCAN:", db_index)
```

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було визначено проблему та проведено дослідження на тему виділення клієнтських груп.

Після проведення аналізу існуючих проблем предметної області, способів їх вирішення та існуючих алгоритмів у галузі кластерного аналізу, було обрано наступні рішення для порівняння із різних категорій: ієрархічні, щільнісні, центроїдні:

- k-means;
- BIRCH;
- Agglomerative clustering;
- DBSCAN;
- OPTICS.

Для визначення якості кластеризації, а саме розділення та ненакладання кластерів один на одного було обрано наступні метрики:

- Silhouette index (індекс силуету);
- Calinski-Harabasz index (індекс Калінскі-Харабаша);
- Davies–Bouldin index (індекс Девіса-Болдіна).

Задля кращого розуміння сутності алгоритмів було досліджено теоретичну та математичну складову кожного із алгоритмів а також їх складових:

- поняття відстані між об’єктами;
- поняття міжкластерної відстані;
- поняття внутрішньокластерної відстані.

Після цього було сплановано та проведено експеримент використовуючи досліджені методи та метрики. Було розроблено програмний застосунок мовою Python, за допомогою якого було проведено два експерименти з різною передобробкою датасетів для викоритання Евклідової та Говерівської відстані.

Усі експерименти проводилися на наборі даних «Датасет відвідувачів готелю». Датасет був достатньо репрезентативним для галузі клієнтської поведінки, маючи велику кількість ознак та записів, а також є реальною

інформацією про відвідувачів Лісабонського готелю, викладеною у вигляді статті та на порталі Kaggle.

Усі результати було записано та викладено у вигляді таблиць. У такому вигляді результати експерименту були проаналізовані, зображені у вигляді діаграм та були основою для висновків щодо застосовності кожного алгоритму.

Виходячи з вищезазначеного, можемо стверджувати виконання поставленого перед чинною роботою завдання – аналіз предметної галузі та проведення дослідження з побудовою необхідних програмних застосунків та проведення порівняльної характеристики алгоритмів кластеризації у задачі виділення клієнтських груп.

Також в рамках кваліфікаційної роботи, дослідження пройшло апробацію на Сьомій міжнародній науково-технічній конференції «Electrical, Electronic And Information Sciences“ IEEE Estream 2023, яка індексується в інформаційній базі SCOPUS, та готова до захисту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. A. Amran, S. K. Ooi, R. T. Mydin, and S. S. Devi, “The impact of business strategies on Online Sustainability Disclosures,” Business Strategy and the Environment, vol. 24, no. 6, pp. 551–564, 2013. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

2. L. Ližbetinová, P. Štarchoň, S. Lorincová, D. Weberová, and P. Průša, “Application of cluster analysis in marketing communications in small and medium-sized enterprises: An empirical study in the Slovak Republic,” Sustainability, vol. 11, no. 8, p. 2302, 2019.

3. P.S. Ray, H. Aiyappan, M.E. Elam, and T.W. Merritt, “Application of cluster analysis in marketing management,”The International Journal of Industrial Engineering: Theory, Applications and Practice, vol. 12, no. 2, pp. 127-133.

4. A. Weinstein, Handbook of Market Segmentation: Strategic Targeting for Business and technology firms, Third Edition. Hoboken, NJ: Taylor and Francis, 2013.

5. C. Fuchs and T. Gutmann, "How Technical Market Segmentation Can Help Build Products Your Customers Really Need," in IEEE Engineering Management Review, vol. 50, no. 1, pp. 17-19, 1 Firstquarter,march 2022, doi: 10.1109/EMR.2022.3140715.

6. B. E. K. Güzel, B. Mocan, B. Arslan, G. Polat and T. Kavuşan, "Demographic Targeting With Epsilon-greedy Exploration in Digital Advertising," 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, 2021, pp. 435-439, doi: 10.1109/UBMK52708.2021.9558951.

7. C. -W. Hsu, Y. -L. Chang, T. -S. Chen, T. -Y. Chang and Y. -D. Lin, "Who Donates on Line? Segmentation Analysis and Marketing Strategies Based on Machine Learning for Online Charitable Donations in Taiwan," in IEEE Access, vol. 9, pp. 52728-52740, 2021, doi: 10.1109/ACCESS.2021.3066713.

8. A. Fadrian and A. S. Arifin, "Study on 2G Termination in Indonesia using BCG Matrix," 2018 5th International Conference on Information Technology, Computer, and

Electrical Engineering (ICITACEE), Semarang, Indonesia, 2018, pp. 1-5, doi: 10.1109/ICITACEE.2018.8576904.

9. Z. Zenkova and T. Kabanova, "The ABC-XYZ analysis modified for data with outliers," 2018 4th International Conference on Logistics Operations Management (GOL), Le Havre, France, 2018, pp. 1-6, doi: 10.1109/GOL.2018.8378073.

10. J. Liao, A. Jantan, Y. Ruan and C. Zhou, "Multi-Behavior RFM Model Based on Improved SOM Neural Network Algorithm for Customer Segmentation," in IEEE Access, vol. 10, pp. 122501-122512, 2022, doi: 10.1109/ACCESS.2022.3223361.

11. R. Zhou et al., "A Hybrid Neural Network Architecture to Predict Online Advertising Click-Through Rate Behaviors in Social Networks," in IEEE Transactions on Network Science and Engineering, vol. 8, no. 4, pp. 3061-3072, 1 Oct.-Dec. 2021, doi: 10.1109/TNSE.2021.3102582.

12. K. Kim, E. Kwon and J. Park, "Deep User Segment Interest Network Modeling for Click-Through Rate Prediction of Online Advertising," in IEEE Access, vol. 9, pp. 9812-9821, 2021, doi: 10.1109/ACCESS.2021.3049827.

13. K. Smelyakov, A. Chupryna, O. Bohomolov and I. Ruban, "The Neural Network Technologies Effectiveness for Face Detection," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), 2020, pp. 201-205, doi: 10.1109/DSMP47368.2020.9204049.

14. K. Smelyakov, A. Chupryna, O. Bohomolov and N. Hunko, "The Neural Network Models Effectiveness for Face Detection and Face Recognition," 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2021, pp. 1-7, doi: 10.1109/eStream53087.2021.9431476.

15. T. C. Benton and D. J. Hand, "Segmentation into predictable classes," in IMA Journal of Management Mathematics, vol. 13, no. 4, pp. 245-259, Oct. 2002, doi: 10.1093/imaman/13.4.245.

16. P. Sarikprueck, W. -J. Lee, A. Kulvanitchaiyanunt, V. C. P. Chen and J. Rosenberger, "Novel Hybrid Market Price Forecasting Method With Data Clustering Techniques for EV Charging Station Application," in IEEE Transactions on Industry

Applications, vol. 51, no. 3, pp. 1987-1996, May-June 2015, doi: 10.1109/TIA.2014.2379936.

17. L. E. Ferro-Díez, N. M. Villegas, J. Díaz-Cely and S. G. Acosta, "Geo-Spatial Market Segmentation & Characterization Exploiting User Generated Text Through Transformers & Density-Based Clustering," in IEEE Access, vol. 9, pp. 55698-55713, 2021, doi: 10.1109/ACCESS.2021.3071620.

18. S. Dolnicar, R. Freitag, and M. Randle, “To segment or not to segment? an investigation of segmentation strategy success under varying market conditions,” Australasian Marketing Journal, vol. 13, no. 1, pp. 20–35, 2005.

19. M. Halkich and M. Vazirgiannis, “A data set oriented approach for clustering algorithm selection,” Principles of Data Mining and Knowledge Discovery, pp. 165–179, 2001.. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

20. D. Kukolj et al., "Comparison of algorithms for patent documents clusterization," 2012 Proceedings of the 35th International Convention MIPRO, Opatija, Croatia, 2012, pp. 995-997.

21. J. Yan, K. A. Linn, B. W. Powers, J. Zhu, S. H. Jain, J. L. Kowalski, and A. S. Navathe, “Applying machine learning algorithms to segment high-cost patient populations,” Journal of General Internal Medicine, vol. 34, no. 2, pp. 211–217, 2018.

22. R. S. Strichartz, The way of analysis. Sudbury, Canada: Jones and Barlett Publishers, 2000, pp. 355–357.

23. J. C. Gower, “A general coefficient of similarity and some of its properties,” Biometrics, vol. 27, no. 4, p. 857, 1971.

24. L. Kaufman and P. J. Rousseeuw, Finding groups in data an introduction to cluster analysis. Hoboken, NJ: Wiley-Interscience, 1990.

25. J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A K-means clustering algorithm,” Applied Statistics, vol. 28, no. 1, p. 100, 1979.

26. Nwadiugwu MC. Gene-Based Clustering Algorithms: Comparison Between Denclue, Fuzzy-C, and BIRCH. Bioinformatics and Biology Insights. 2020;14. doi:10.1177/1177932220909851

27. C. C. Aggarwal and C. K. Reddy, Data clustering: Algorithms and applications. Boca Raton, FL: Chapman and Hall/CRC, 2018, pp. 101–103.

28. E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “DBSCAN Revisited, revisited,” ACM Transactions on Database Systems, vol. 42, no. 3, pp. 1–21, 2017.

29. M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “Optics: Ordering Points To Identify the Clustering Structure,” ACM SIGMOD Record, vol. 28, no. 2, pp. 49–60, 1999.

30. P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” Journal of Computational and Applied Mathematics, vol. 20, pp. 53–65, 198.

31. D. L. Davies and D. W. Bouldin, “A cluster separation measure,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-1, no. 2, pp. 224–227, 1979.

32. S. Łukasik, P. A. Kowalski, M. Charytanowicz and P. Kulczycki, "Clustering using flower pollination algorithm and Calinski-Harabasz index," 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 2016, pp. 2724-2728, doi: 10.1109/CEC.2016.7744132.

33. N. Antonio, A. de Almeida, and L. Nunes, “A hotel's customers personal, behavioral, demographic, and geographic dataset from Lisbon, Portugal (2015–2018),” Data in Brief, vol. 33, p. 106583, 2020.

34. L. Yu, R. Zhou, R. Chen, and K. K. Lai, “Missing data preprocessing in credit classification: One-hot encoding or imputation?,” Emerging Markets Finance and Trade, vol. 58, no. 2, pp. 472–482, 2020.

35. M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," in IEEE Access, vol. 9, pp. 114381-114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

36. N. Rahmah and I. S. Sitanggang, “Determination of Optimal Epsilon (EPS) value on DBSCAN algorithm to clustering data on peatland hotspots in Sumatra,” IOP Conference Series: Earth and Environmental Science, vol. 31, p. 012012, 2016.