Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(навчально-науковий інститут)
Факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістр
(бакалавра, магістра)

студента Півня Нікіти Костянтиновича
(ПІБ) і
академічної групи 126м-23-1
(шифр)
спеціальності 126 «Інформаційні системи та технології»
(код і назва спеціальності)
спеціалізації за освітньо-професійною (освітньо-науковою) програмою
126 Інформаційні системи та технології
(офіційна назва)
на тему «Дослідження методів машинного навчання без вчителя для кластеризації текстових даних українською мовою»
(назва за наказом ректора)

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	Доц. Каштан В.Ю.			
розділів:				

Рецензент	Проф. Лактіонов І.С.			

Нормоконтролер	Проф. Коротенко Г.М.			

Дніпро
2024

ЗАТВЕРДЖЕНО:
завідувач кафедри
інформаційних технологій та комп'ютерної інженерії
(повна назва)
_____ В.В. Гнатушенко
(підпис) (прізвище,ініціали)

«_____» _____ 2024 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістр

студенту Півню Н.К. **академічної групи** 126м-23-1
(прізвище та ініціали) (шифр)

спеціальності 126 «Інформаційні системи та технології»
спеціалізації за освітньою-професійною програмою
126 «Інформаційні системи та технології»
(офіційна назва)

на тему «Дослідження методів машинного навчання без вчителя для кластеризації текстових даних українською мовою»

затверджену наказом ректора НТУ «Дніпровська політехніка» від_____ № _____

Розділ	Зміст	Термін виконання
Розділ 1 Аналіз стану області дослідження	Описати проблему кластеризації текстових даних, її основні аспекти та завдання.	16.09.2024
Розділ 2 Огляд основних методів кластеризації текстових даних	Огляд основних методів кластеризації текстових даних.	14.10.2024
Розділ 3 Розробка інструменту дослідження методів машинного навчання без вчителя для кластеризації текстових даних українською мовою	Розробити програмний інструмент для дослідження методів машинного навчання без вчителя та провести експериментальні дослідження, що застосовуються до кластеризації текстових даних українською мовою.	30.11.2024

Завдання видано _____ В.Ю.Каштан
(підпис керівника) (прізвище, ініціали)
Дата видачі 05 вересня 2024 р.
Дата подання до екзаменаційної комісії _____р.
Прийнято до виконання _____ Н.К.Півень
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 86 с., 28 рис., 17 джерел, 2 додатки.

Об'єкт дослідження – текстові дані українською мовою, які потребують кластеризації для подальшої організації та аналізу.

Предмет дослідження – методи машинного навчання без вчителя, що використовуються для кластеризації текстових даних, зокрема алгоритми KMeans, DBSCAN, LDA та Single-Pass.

Мета дослідження – розробити інструмент та оцінити ефективність методів машинного навчання без вчителя для кластеризації текстових даних українською мовою, визначити їхні переваги та недоліки, а також запропонувати рекомендації щодо їх практичного застосування.

Наукова новизна полягає в систематичному аналізі та порівнянні методів кластеризації текстових даних українською мовою, що є малодослідженою темою в науковій спільноті. Це дослідження має на меті не лише виявлення найбільш ефективних методів кластеризації, але й розширення наукових знань у цій галузі, що має важливе значення для подальшого розвитку технологій обробки текстових даних та інтелектуального аналізу інформації.

Практичне значення проведеного дослідження полягає в тому, що розроблений програмний інструмент для кластеризації текстових даних українською мовою може бути використаний для вирішення широкого спектра задач у різних галузях, де важлива автоматична обробка текстів.

Ключові слова: ДИСТАНЦІЙНЕ ЗОНДУВАННЯ (ДЗЗ), СУПУТНИКОВИЙ ЗНІМОК, ГЕОІНФОРМАЦІЙНІ СИСТЕМИ (ГІС), ВУЛКАНІЧНА АКТИВНІСТЬ, SENTINEL-2, ЧАСОВІ РЯДИ

ABSTRACT

The explanatory statement has 86 p., 28 figures, 17 sources, and 2 appendixes.

Object of research: Sentinel-2 satellite images for monitoring volcanic activity in La Palma.

Subject of research: technology for analyzing the impact of volcanic activity on the marine environment and the surrounding ecosystem.

The purpose of the diploma project: development of information technology for monitoring volcanic activity in La Palma.

The qualification work is dedicated to the actual task of developing information technology for monitoring volcanic activity based on the processing and analysis of remote sensing data and geoinformation methods. An information technology was developed, with the help of which the impact of volcanic activity on the marine environment and the surrounding ecosystem was identified and analyzed.

Scientific research consists in the use of satellite images at various times, atmospheric correction, calculation of thermal indices, visualization of these indices and analysis of the obtained data. This provides an opportunity to monitor changes in temperature and other parameters of the volcano's surface, which can be useful for predicting volcanic eruptions and ensuring public safety.

The practical significance of the work is that the results of the study can be implemented in the processes of monitoring changes in the temperature and other parameters of the surface of the volcano, which can be useful for predicting volcanic eruptions and ensuring public safety.

Keywords: REMOTE SENSING (DZZ), SATELLITE IMAGE, GEO-INFORMATION SYSTEMS (GIS), VOLCANIC ACTIVITY, SENTINEL-2, TIME SERIES

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ГІС – геоінформаційні системи;

ДЗЗ – дистанційне зондування Землі;

SAR – радіолокатора з синтетичною апертурою;

ЦМР – цифрові моделі рельєфу;

ОТВ – Orfeo ToolBox;

ЕМ– електромагнітне випромінювання;

NIR – ближній інфрачервоний;

SWIR – короткий інфрачервоний;

MWIR – середньохвильовий інфрачервоний.

ВСТУП

Важливість категоризації (кластеризації) текстів зростає в умовах постійного збільшення обсягу доступної інформації. Сьогодні ми стикаємося з численними викликами, пов'язаними з обробкою великої кількості даних, які потребують систематизації. Наприклад, неможливо класифікувати вручну тисячі електронних листів, що надходять до наших поштових скриньок, так само як і неефективно впорядковувати великі масиви текстових документів, які використовуються в дослідженнях з інтелектуального аналізу даних. Внаслідок цього, дезорганізація документів та текстових даних стає серйозною проблемою, яку слід вирішити для підвищення продуктивності у виконанні завдань, що стосуються великих обсягів текстової інформації.

Завдяки останнім досягненням у галузі штучного інтелекту, неконтрольованого навчання та розвитку комп'ютерних технологій, з'явилася можливість автоматизації завдань, виконання яких вимагало б значних затрат часу з боку людини. Машинне навчання без вчителя, зокрема, пропонує різноманітні алгоритми та підходи, здатні ефективно організовувати текстові дані в кластери, виявляючи в них приховані структури та закономірності.

Дана кваліфікаційна робота присвячена дослідженню методів машинного навчання без вчителя для кластеризації текстових даних українською мовою. У рамках роботи будуть розглянуті алгоритми, такі як KMeans, DBSCAN, LDA та Single-Pass, їх порівняння та оцінка ефективності у контексті кластеризації текстової інформації. Це дослідження має на меті не лише виявлення найбільш ефективних методів кластеризації, але й розширення наукових знань у цій галузі, що має важливе значення для подальшого розвитку технологій обробки текстових даних та інтелектуального аналізу інформації.

Об'єкт дослідження – текстові дані українською мовою, які потребують кластеризації для подальшої організації та аналізу.

Предмет дослідження – методи машинного навчання без вчителя, що використовуються для кластеризації текстових даних, зокрема алгоритми KMeans, DBSCAN, LDA та Single-Pass.

Мета дослідження – розробити інструмент та оцінити ефективність методів машинного навчання без вчителя для кластеризації текстових даних українською мовою, визначити їхні переваги та недоліки, а також запропонувати рекомендації щодо їх практичного застосування.

Наукова новизна полягає в систематичному аналізі та порівнянні методів кластеризації текстових даних українською мовою, що є малодослідженою темою в науковій спільноті. Це дослідження має на меті не лише виявлення найбільш ефективних методів кластеризації, але й розширення наукових знань у цій галузі, що має важливе значення для подальшого розвитку технологій обробки текстових даних та інтелектуального аналізу інформації.

1 АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ

1.1 Опис проблеми кластеризації текстових даних

Кластеризація текстових даних є важливою задачею в галузі обробки природної мови (NLP) та машинного навчання, що полягає в автоматичному поділі текстових документів на групи (кластери) на основі схожості їх вмісту. У процесі кластеризації текст може бути класифікований у різні групи незалежно від його внутрішніх властивостей, що може призводити до неоднозначності результатів. Це, у свою чергу, створює потребу в розробці та введенні актуальних критеріїв якості кластеризації, щоб уникнути подібних проблем [2].

Аналіз великої кількості різнотипних даних породжує методологічну проблему вибору метрик для оцінювання якості кластеризації. Збільшення числа об'єктів, навіть якщо вони однотипні, може спричинити нерозрізнення відстаней між ними, що ускладнює процес кластеризації. Крім того, класичні методи зниження розмірності, які використовуються в кластерному аналізі, зазвичай орієнтовані на лінійні взаємозв'язки між змінними. Для виявлення складніших залежностей необхідно переходити до ядерних методів, що вимагає додаткових зусиль та ресурсів.

Однією з основних проблем, що виникають при кластеризації, є пошук глобального екстремуму функції критерію якості. Як правило, критерій якості є функцією, що залежить від великої кількості чинників, є нелінійною та має безліч локальних екстремумів. Для знаходження кластерів необхідно вирішити складну комбінаторну задачу пошуку оптимального варіанту класифікації. Якщо кількість груп заздалегідь невідома, це ускладнює перебірну задачу, що призводить до «комбінаторного вибуху» в умовах збільшення розмірності таблиць даних. Класичні алгоритми кластерного аналізу здійснюють спрямований пошук у невеликій підмножині простору рішень, що не гарантує знаходження строго оптимального рішення [2].

Для пошуку оптимальних рішень застосовують складніші методи, такі як генетичні алгоритми, нейронні мережі тощо. Існують експериментальні дослідження, що підтверджують переваги таких алгоритмів перед класичними. Однак навіть під час використання еволюційних методів виникають труднощі, пов’язані зі специфікою задачі кластерного аналізу, зокрема з труднощами інтерпретації використовуваних операторів рекомбінації та кросовера.

Крім того, результати кластеризації можуть варіюватися залежно від вибору початкових умов, порядку об'єктів та параметрів роботи алгоритмів. Для підвищення стійкості групувальних рішень пропонуються різноманітні способи, засновані на застосуванні ансамблів алгоритмів. Це може включати використання результатів групування, отриманих різними алгоритмами, або одним алгоритмом з різними параметрами налаштування.

Також існує проблема недостатності знань про об'єкт, що ускладнює створення моделі в важкоформалізованих галузях. У таких випадках застосування алгоритмів, що ґрунтуються на поданні класу як набору розподілених у просторі змінних, стає складним [3].

Не менш важливою є проблема подання результатів кластеризації. Для будь-якого алгоритму аналізу даних важливо, щоб його результати були зрозумілими та інтерпретованими. Для покращення інтерпретованості рішень можуть використовуватися логічні моделі, які застосовуються для розв'язання задач розпізнавання образів і прогнозування кількісних показників.

Таким чином, кластеризація текстових даних є складною, але важливою задачею в сучасному світі інформаційних технологій. Проблеми, описані вище, вимагають подальшого дослідження та розробки нових рішень, що можуть суттєво поліпшити якість кластеризації та забезпечити більш точні результати.

1.2 Завдання кластерного аналізу

Кластерний аналіз виконує кілька основних завдань, які є критично важливими для розуміння структури даних і формулювання коректних висновків на основі отриманих результатів.

Дослідження схем групування об'єктів – це завдання передбачає вивчення, як об'єкти можуть бути згруповані на основі їх характеристик. Кластеризація допомагає визначити, чи існують природні групи або підгрупи в даних, що можуть бути незрозумілими на перший погляд. Наприклад, у маркетингу кластерний аналіз може допомогти ідентифікувати сегменти споживачів з подібними уподобаннями або поведінкою, що може бути використано для цільового рекламного просування.

Вироблення гіпотез на базі досліджень даних – кластеризація є потужним інструментом для формування початкових гіпотез про структуру даних. На основі результатів кластерного аналізу дослідники можуть висувати припущення щодо того, чому певні об'єкти згруповані разом. Це може допомогти у подальших дослідженнях, коли потрібно буде підтвердити або спростувати ці гіпотези за допомогою додаткового аналізу чи експериментів.

Підтвердження гіпотез і досліджень даних – після проведення кластеризації результати можуть бути використані для перевірки, чи відповідають вони початковим уявленням про дані. Це важливий етап, оскільки він допомагає встановити надійність і коректність отриманих висновків. Наприклад, якщо дослідження виявило класи об'єктів, що мають схожі характеристики, це може підтвердити гіпотезу про певні закономірності у даних.

Визначення присутності груп усередині даних – це завдання дозволяє виявити раніше невідомі класи об'єктів у наборі даних. Кластерний аналіз може розкрити нові закономірності, які не були враховані на початку дослідження. Наприклад, у медицині кластеризація може допомогти

виявити нові підтипи захворювань на основі симптомів або генетичних даних, що може призвести до нових підходів у лікуванні.

1.3. Етапи кластерного аналізу

Застосування кластерного аналізу, незалежно від предмета вивчення, передбачає кілька важливих етапів, кожен з яких відіграє критичну роль у досягненні точних і значущих результатів

Першим етапом є формування вибірки, яка повинна бути репрезентативною для досліджуваної популяції. Це означає, що вибірка повинна відображати різноманітність об'єктів у популяції, щоб результати кластеризації були узагальненими та достовірними. Неправильна або неповна вибірка може призвести до спотворених результатів і неправильних висновків. Для цього важливо визначити, які саме об'єкти будуть включені у вибірку, та як їх кількість вплине на якість кластеризації.

Виділення простору ознак – другий етап. На цьому етапі необхідно визначити ключові характеристики (ознаки), які будуть використовуватися для кластеризації. Вибір ознак є критично важливим, оскільки саме вони визначають, як об'єкти будуть групуватися. Ознаки можуть бути кількісними (числовими) або якісними (категоріальними) і повинні мати сенс у контексті дослідження. Важливо також уникати включення надмірної кількості ознак, які не мають істотного впливу на результат, оскільки це може ускладнити аналіз і знизити якість кластеризації [1].

Вибір відповідної метрики для вимірювання подібності або відстані між об'єктами є критично важливим для успішного виконання кластеризації. Залежно від типу даних та вибраних ознак можуть використовуватися різні метрики, такі як евклідова відстань, манхеттенська відстань або косинусна подібність. Неправильний вибір метрики може призвести до помилкових кластерів і спотворення результатів, тому важливо адаптувати метрику до специфіки дослідження.

На цьому етапі відбувається безпосереднє застосування вибраного методу кластерного аналізу. Існує безліч алгоритмів кластеризації, і вибір конкретного методу залежить від даних і цілей дослідження. Наприклад, для виявлення круглих кластерів можна використовувати метод k-середніх, тоді як для виявлення ієрархічних структур підійде ієрархічна кластеризація. Важливо також провести налаштування параметрів алгоритму, якщо це потрібно, для досягнення кращих результатів.

Останнім етапом є перевірка якості кластеризації та оцінка її відповідності початковим очікуванням. Це може включати візуалізацію отриманих кластерів, використання статистичних показників, таких як силуетний коефіцієнт, або порівняння результатів з відомими класами об'єктів (якщо такі є). Це дозволяє зрозуміти, наскільки вдало було проведено кластеризацію і чи потрібно вносити корективи у вибір ознак, метрик або самих алгоритмів.

Для ефективної кластеризації існують дві ключові вимоги до даних:

Однорідність – усі об'єкти, що кластеризуються, повинні описуватися схожим набором характеристик. Це забезпечує, що різні об'єкти в одному кластері мають спільні риси, що сприяє їх коректному групуванню. Якщо об'єкти мають різнорідні ознаки, це може призвести до виникнення невдалих кластерів і ускладнити подальший аналіз.

Повнота – дані повинні бути представлені у достатньому обсязі, щоб забезпечити раціональне розв'язання задачі. Це означає, що обсяги даних повинні бути достатніми для того, щоб забезпечити адекватний рівень статистичної потужності та уникнути впливу випадкових коливань. Недостатня кількість даних може призвести до невірних кластерів і вивести з ладу процес кластеризації.

1.4 Цілі кластеризації для вироблення рекомендацій

Кластеризація є потужним інструментом у статистичному аналізі та машинному навчанні, і її цілі можуть варіюватися в залежності від конкретних застосувань. Основні цілі кластеризації включають [2]:

– розбиття вибірки на групи схожих об'єктів - кластеризація дозволяє розділити великий набір даних на менші групи (кластери), де об'єкти в кожному кластері мають схожі характеристики. Це значно спрощує розуміння структури даних і допомагає ухвалювати рішення. Наприклад, в бізнес-аналізі можна використовувати різні маркетингові стратегії для кожного кластера споживачів, що дозволяє персоналізувати підходи до клієнтів і підвищити ефективність кампаній;

– скорочення обсягу даних – кластеризація також сприяє зменшенню обсягу даних, оскільки дозволяє залишити по одному або кілька найтиповіших представників від кожного класу. Це важливо для збереження високого ступеня подібності об'єктів усередині кожного кластера, що полегшує подальший аналіз. Наприклад, у великих наборах даних про споживчі звички можна зберегти лише найхарактерніші профілі клієнтів, зменшуючи обсяг даних для обробки без втрати значущої інформації;

– виділення нетипових об'єктів, аномалій або викидів - кластеризація дозволяє виявити аномальні або нетипові об'єкти, які не вписуються в жоден з кластерів. Це може бути важливим для виявлення нових тенденцій або виявлення викидів, що можуть свідчити про помилки в даних чи нові, раніше не виявлені патерни. Наприклад, у фінансовому аналізі аномалії можуть вказувати на шахрайські транзакції, що вимагає подальшого розслідування;

– застосування ієрархічної кластеризації – ієрархічна кластеризація передбачає розподіл великих кластерів на менші, що дозволяє створювати багатоуровневу структуру класифікації. Результатом цього процесу є деревоподібна структура (дендрограма), де кожен об'єкт може бути представлений у різних кластерах, в залежності від рівня деталізації.

Це особливо корисно в біології для класифікації видів або у маркетингу для сегментації ринку на різні підгрупи.

1.5 Актуальність та значення кластеризації текстових даних

У сучасному світі обсяги текстових даних зростають з космічною швидкістю завдяки поширенню соціальних медіа, онлайн-коментарів, новин, блогів, наукових публікацій та інших джерел. Цей потік інформації створює необхідність у ефективних методах її обробки та аналізу. Кластеризація текстових даних є однією з основних технологій, що дозволяє структурувати ці дані, виявляти закономірності та формувати усвідомлені рішення на їх основі.

Структуризація інформації дозволяє організувати великі обсяги текстових даних у зрозумілі групи на основі подібності. Це важливо для полегшення доступу до інформації та її подальшого аналізу. Наприклад, кластеризація може використовуватися для організації новинних статей за темами, що допомагає журналістам швидше знаходити релевантні матеріали для своїх статей.

Виявлення закономірностей допомагає виявляти приховані патерни і тренди в текстових даних, які можуть бути неочевидними при ручному аналізі. Це особливо корисно в маркетингових дослідженнях, де аналіз споживчих відгуків може вказувати на нові тенденції або проблеми в продуктах.

Якісна кластеризація може допомогти компаніям виявляти нові ринки та оптимізувати свої пропозиції. Наприклад, аналіз відгуків клієнтів може вказувати на потреби, які ще не були враховані в бізнес-моделі, що дозволяє компанії адаптуватися до змін у споживчому попиті.

Кластеризація може бути використана для сегментації клієнтів на основі їх поведінки та вподобань, що дозволяє компаніям налаштовувати свої комунікаційні стратегії. Це призводить до більш персоналізованого

обслуговування клієнтів, що, в свою чергу, може підвищити лояльність і задоволеність споживачів.

1.6 Завдання для кваліфікаційної роботи

У даній кваліфікаційній роботі передбачено виконання ряду завдань. По-перше, необхідно провести огляд та аналіз методів кластеризації, зокрема вивчити основні принципи роботи алгоритмів KMeans, DBSCAN, LDA та Single-Pass. Для цього слід розглянути математичні основи кожного з методів, включаючи формули, алгоритми та їхні особливості. Наступним завданням є визначення переваг та недоліків цих алгоритмів, що дозволить виявити, у яких конкретних випадках рекомендовано використовувати кожен із методів.

Крім того, важливо провести експерименти для порівняння ефективності алгоритмів. Для цього потрібно вибрати відповідний набір даних для тестування, наприклад, текстові дані з українських джерел, та здійснити кластеризацію за допомогою всіх обраних алгоритмів на однакових наборах даних. Результати кластеризації слід оцінити за різними метриками, такими як внутрішня когерентність і силует, а також проаналізувати вплив викидів на результати.

Третім завданням є оптимізація параметрів алгоритмів, що передбачає визначення оптимальних значень для KMeans, налаштування параметрів щільності для DBSCAN (параметри eps та min_samples) та визначення кількості тем для LDA. Крім цього, потрібно оптимізувати параметри для Single-Pass.

Четвертим завданням є візуалізація результатів кластеризації. У цьому контексті рекомендується розробити графічні методи візуалізації результатів, включаючи діаграми розсіювання та теплові карти, а також порівняти візуалізації для різних алгоритмів для кращого розуміння структури даних.

На завершення роботи слід сформулювати висновки та рекомендації, підсумувавши результати дослідження для кожного з алгоритмів, визначивши, який з них є найефективнішим для кластеризації текстових даних в українському контексті, та запропонувавши рекомендації щодо практичного застосування алгоритмів кластеризації в різних бізнес-сценаріях і дослідженнях.

1.7 Висновки до першого розділу

У першому розділі було детально проаналізовано проблему кластеризації текстових даних, її основні аспекти та завдання. Кластеризація є ключовою задачею у сфері обробки природної мови (NLP), яка дозволяє автоматично групувати документи за схожістю вмісту, проте виникає низка проблем, пов’язаних з точністю та інтерпретованістю отриманих результатів.

Основними викликами в кластеризації текстових даних є:

– неоднозначність класифікації текстів через можливі неточності у виборі ознак та метрик для вимірювання схожості;

– методологічні проблеми вибору метрик якості кластеризації, зокрема при збільшенні кількості об’єктів та необхідності використання нелінійних підходів;

– проблеми оптимізації — складність пошуку глобального екстремуму через наявність численних локальних мінімумів;

– залежність результатів від початкових умов та параметрів алгоритму, що призводить до варіативності кінцевих кластерів;

– важливість інтерпретованості рішень, особливо при використанні алгоритмів машинного навчання.

Також було окреслено етапи кластерного аналізу, включаючи формування вибірки, визначення ознак та вибір метрики подібності. Значну

увагу приділено проблемам оцінки якості кластеризації, підбору оптимальних алгоритмів і параметрів, а також важливості відповідності початковим гіпотезам.

Таким чином, кластеризація текстових даних є складним процесом, який потребує вдосконалення методів для підвищення точності, стійкості та інтерпретованості отриманих результатів.

2 ОГЛЯД ОСНОВНИХ МЕТОДІВ КЛАСТЕРИЗАЦІЇ ТЕКСТОВИХ ДАНИХ

2.1 Підготовка даних

Для проведення кластеризації текстових даних необхідно спершу перетворити текстову інформацію у числовий формат, щоб подати її на вхід алгоритму кластеризації. Однак, сам процес перетворення даних і кластеризації не передбачає фільтрації нерелевантних даних. Це означає, що, якщо в набір даних включені нерелевантні елементи, їх слід видалити перед початком вилучення ознак. В іншому випадку ці дані можуть негативно вплинути на якість кластерів, що, в свою чергу, позначиться на точності категоризації загалом.

Процес кластеризації текстів має ряд етапів, спрямованих на очищення і підготовку даних для максимального поліпшення результатів кластеризації. Як зазначено у джерелі [1], для досягнення якісних кластерів необхідно підготувати текстові дані, зводячи їх до найбільш релевантної форми. Проте, важливо адаптувати підхід до очищення залежно від специфіки кожного набору даних. Наприклад, у даному проекті всі вхідні дані походять з конкретного корпоративного контексту, що передбачає наявність певних спільних характеристик у текстах.

Процес підготовки текстових даних включає кілька важливих етапів:

1. Токенізація є початковим етапом, який полягає у розбитті тексту на окремі елементи — токени. Кожне речення або абзац перетворюється на масив, де кожен елемент цього масиву є токеном (окреме слово або символ). Це основоположна операція, оскільки після поділу тексту на токени всі наступні етапи можуть бути виконані простіше і ефективніше, застосовуючи методи для обробки масивів даних.

2. Розділові знаки, такі як коми, крапки, знаки питання і т.д., не несуть значущої інформації для кластеризації текстів, тому їх необхідно видалити.

У більшості випадків розділові знаки не допомагають у визначенні тематики тексту і можуть лише створювати шум у даних.

3. Стоп-слова — це слова, які є дуже поширеними у мовленні, але не мають важливого значення для аналізу тексту. До таких слів належать прийменники, сполучники та інші частини мови, наприклад: "the", "for", "your", "a". Ці слова слід видалити з тексту, щоб фокусувати аналіз на тих елементах, які дійсно можуть впливати на кластеризацію.

4. Деякі слова можуть бути нерелевантними для конкретної задачі кластеризації, навіть якщо вони не належать до стоп-слів. Наприклад, у задачі категоризації листів за темами, цифри можуть бути нерелевантними і вважатися неконтекстними. Однак, у випадку, коли потрібно категоризувати листи за датами, ці ж цифри набувають релевантності. Тому важливо адаптувати процес очищення тексту під конкретну задачу.

5. Стеммінг — це процес перетворення слів до їх основної форми (кореня). Наприклад, в українській мові слова "працювати", "працює" і "працювали" можуть бути перетворені до форми "працю". Проте цей метод має певні недоліки, оскільки іноді отримані форми не є реальними коренями слів. Наприклад, слово "спільний" може бути скорочено до "спільн", що є некоректним і непридатним для подальшої обробки.

6. Лемматизація є складнішою альтернативою стеммінгу, оскільки враховує не лише морфологічну форму слова, але й його контекст. Це дозволяє перетворювати слова на їх базову форму, зберігаючи семантичну точність. Наприклад, слова "працював", "працює" і "працювати" будуть приведені до базової форми "працювати". Це робить лемматизацію більш точним і надійним підходом для попередньої обробки тексту, оскільки вона зберігає правильну форму слів, що є важливим для подальшого аналізу.

Ці етапи підготовки тексту допомагають створити більш чисті і структуровані дані для подальшої кластеризації. Врахування специфіки даного набору дозволяє максимально ефективно кластеризувати текстові дані, що підвищує якість отриманих результатів.

2.2 Виділення ознак

Більшість сучасних алгоритмів машинного навчання потребують числових даних для коректного виконання своїх завдань. Відповідно, необхідно перетворити початкові дані в числовий формат, тобто витягти ознаки та представити їх в іншому форматі. Цей процес називається виділенням ознак і є важливим у багатьох галузях машинного навчання, таких як обробка зображень, розпізнавання облич і, що найважливіше в цьому контексті, обробка тексту [3]. У наступних підрозділах розглядаються основні методи виділення ознак, що використовуються в цій роботі, а також пов'язані з ними дослідження. Серед цих підходів є як простіші методи, такі як словесні ознаки (bag of words) і tf-idf ваги, так і більш складні та сучасні, як word2vec та doc2vec.

2.2.1 Словесні ознаки

Метод словесних ознак перетворює текстові дані на числові, що відображають кількість входжень кожного слова в кожному документі. Процес працює наступним чином: спочатку створюється словник, що містить усі унікальні слова в наборі текстів, а потім підраховується кількість входжень кожного слова в кожному документі. Це дає так звану частотну матрицю термінів, де одна вісь містить слова зі словника, а інша — документи. Кожна клітинка матриці показує, скільки разів певне слово зустрічається в певному документі.

Простий приклад допоможе краще зрозуміти цей підхід:

– doc1: «Дозвольте мені пояснити підхід словесних ознак»;

– doc2: «Словесні ознаки – це метод векторизації»;

– doc3: «Словесні ознаки – не найкращий метод векторизації».

Як результат, словник містить наступні слова: [дозвольте, мені, пояснити, мішок, слів, підхід, це, метод, векторизації, не, найкращий].

В табл..2.1 наведено частотну матрицю, де рядки відповідають словам, а стовпці – документам.

Таблиця 2.1 – Частотна матриця

Слово	doc1	doc2	doc3
алгоритм	1	0	1
машинного	1	0	0
навчання	1	1	1
потребує	1	0	0
даних	1	1	1
для	1	0	0
тренування	1	0	0
машинне	0	1	0
вимагає	0	1	0
обробки	0	1	1
застосовується	0	0	1
в	0	0	1

Наступним кроком є побудова частотної матриці, що показує кореляцію між словами в словнику та кожним документом. Основною перевагою методу мішка слів є його простота і легкість впровадження. Однак він має кілька недоліків, зокрема: метод не враховує семантичного контексту (порядок і значення слів).

Часто виникає проблема розрідженості матриці, оскільки велика кількість ознак може мати нульові значення. Це призводить до низької ефективності під час обробки великих наборів даних [1].

2.2.2 TF-IDF

Хоча ми можемо використовувати частотну матрицю, отриману за допомогою мішка слів, для кластеризації, результати можуть бути

незадовільними, оскільки цей підхід не враховує значущість слів у контексті. Наприклад, слово "це" може мати високу частоту, але його важливість у контексті задачі є мінімальною.

Метрика tf-idf (term frequency-inverse document frequency) вирішує цю проблему, враховуючи як частоту входжень слова, так і його значущість. Вона обчислює значення для кожного слова в документі, перемножуючи частоту терміна (tf) на зворотну частоту документа (idf) [2]:

$$TF(w) = \frac{n_r(w)}{n_r}, \tag{2.1}$$

де $n_r(w)$ – кількість появ терміна w у документі; n_r – загальна кількість слів у документі.

Ось як це працює: Частота терміна (TF) обчислюється як кількість входжень слова в документі, поділена на загальну кількість слів у цьому документі. Це дозволяє нормалізувати частоту термінів для уникнення переваги довших документів.

Зворотна частота документа (IDF) оцінює важливість терміна, враховуючи рідкість його використання в наборі документів. Вона обчислюється як логарифм від кількості всіх документів, поділеної на кількість документів, що містять це слово. Поєднуючи ці два показники, алгоритм надає більші ваги рідкісним словам і менші — часто вживаним, роблячи їх більш значущими для подальшої обробки:

$$IDF(w) = \log\left(\frac{n_r}{n_r(w)}\right) \tag{2.2}$$

Метод tf-idf широко застосовується у виділенні ознак із текстових даних і з моменту своєї появи [4] зазнав численних варіацій, що покращують його ефективність у різних контекстах.

2.2.3 Word2Vec

Перш ніж розглянути технологію Word2Vec, важливо зрозуміти концепцію моделі skip-gram, запропонованої Томашем Міколовим. Ця модель намагається усунути обмеження деяких традиційних методів виділення ознак, таких як "bag of words" та TF-IDF, які враховують лише частотність слів і не встановлюють зв’язків між ними. Головною метою моделі skip-gram є побудова синтаксичних і семантичних зв’язків між словами, що дозволяє отримати краще представлення тексту під час виділення ознак. Метод word2vec працює завдяки використанню архітектури нейронної мережі, яка має лише три шари: вхідний шар, прихований шар та вихідний шар, як показано на рисунку 2.1.

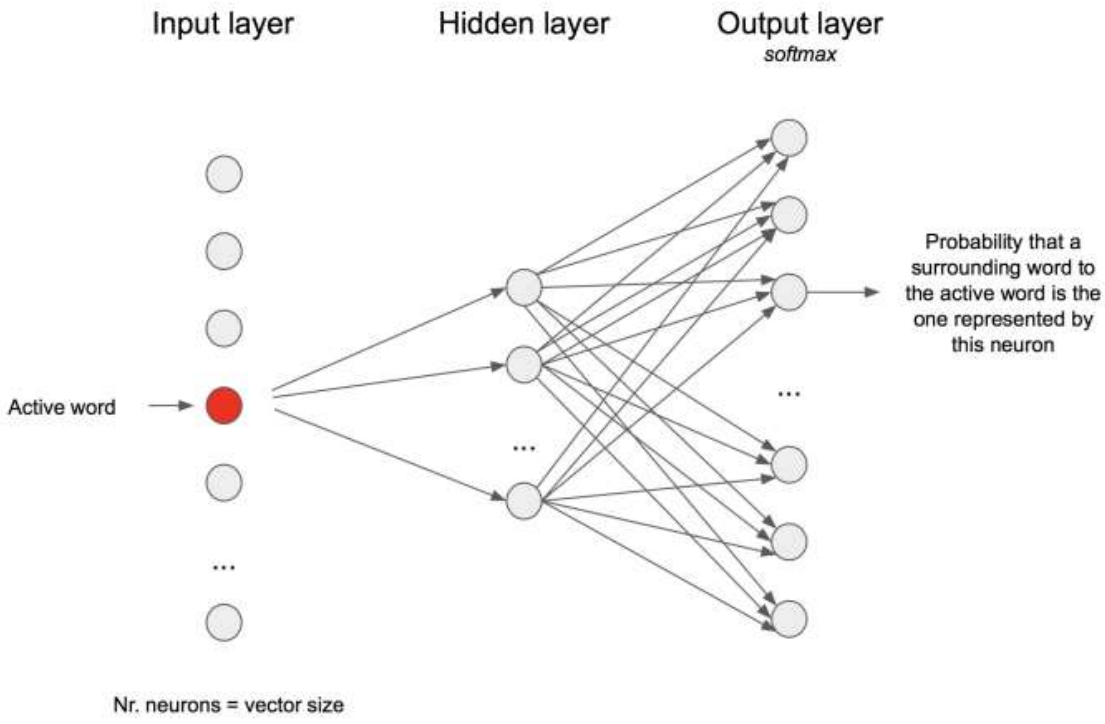


Рисунок 2.1 – Нейромережева схема для Word2vec

Технологія Word2Vec, опублікована в 2013 році тим же автором разом із кількома дослідниками Google [5], являє собою вдосконалення моделі skip-gram, зокрема у створенні векторів ознак і оптимізації часу навчання. Основний принцип роботи Word2Vec полягає в тому, що слова, які часто

зустрічаються поруч у тексті, повинні мати схожі векторні представлення. Цей підхід використовує нейронну мережу, яка складається з трьох шарів: вхідного, прихованого та вихідного.

Метою такої мережі є навчитися прогнозувати слова, які можуть з'явитися в оточенні певного слова у реченні. Після навчання моделі, вагові коефіцієнти прихованого шару стають векторами ознак для кожного слова, що дозволяє використовувати ці вектори для подальших завдань, таких як аналіз тексту чи кластеризація.

Метод Word2Vec виявляється особливо корисним в задачах аналізу настроїв завдяки здатності виявляти контекст [6], а також у задачах кластеризації та обробки великих корпусів текстів [1].

2.2.4 Doc2Vec

У 2014 році Міколов та його співавтори розширили метод Word2Vec, запропонувавши підхід Doc2Vec (рис.2.2), який спрямований на створення векторного представлення для цілих документів, а не окремих слів. Цей підхід також відомий як "вектор абзацу" (paragraph vector), і він дозволяє отримати більш узагальнене представлення тексту, що враховує як семантику, так і структуру документа.

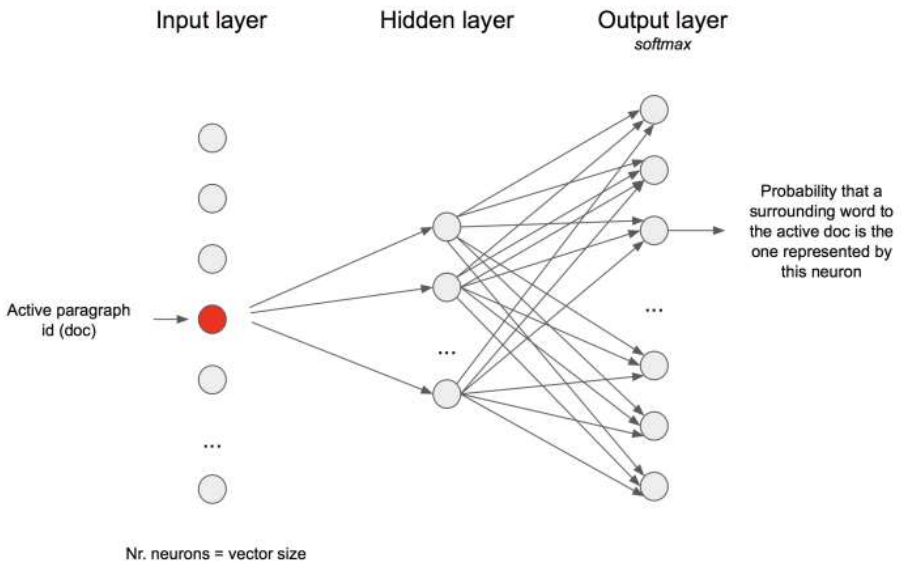


Рисунок 2.2 – Нейромережева схема для Doc2Vec

Основна ідея Doc2Vec полягає в тому, що замість створення векторів для кожного окремого слова, генерується один вектор для всього документа або абзацу. Це дає можливість подолати недоліки методів, таких як "мішок слів", які ігнорують семантичні зв’язки між словами і їхній порядок. Doc2Vec є ефективним для аналізу великих текстових корпусів і класифікації документів, оскільки забезпечує більш глибоке представлення контексту документа в порівнянні з попередніми методами.

2.3 Кластеризація

Кластеризація є важливою складовою сучасних технологій, що стосуються обробки даних [7] і широко застосовується в багатьох наукових та інженерних дисциплінах. Її використовують у статистичному аналізі даних, машинному навчанні, аналізі зображень, комп'ютерній графіці, біоінформатиці, а також для задач інформаційного пошуку (рис.2.3). В Роботі [8] зазначають, що кластеризація є однією з фундаментальних частин розуміння даних. Оскільки кластеризація оперує з даними без попередньо визначених шаблонів, вона відноситься до групи некерованих методів навчання.

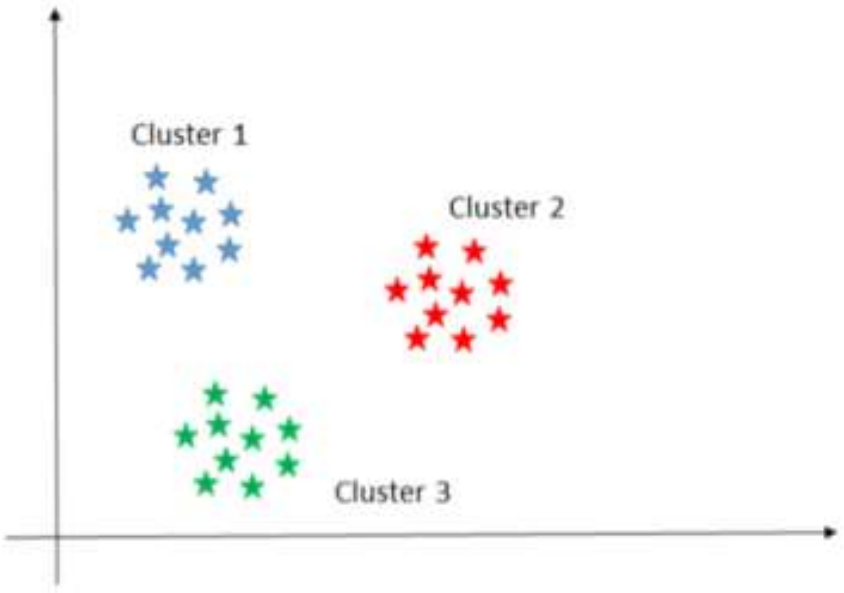


Рисунок 2.3 – Кластерний аналіз

Кластеризація — це процес, який розділяє дані на групи (кластери), в яких екземпляри, що належать до однієї групи, схожі між собою, тоді як екземпляри з різних груп відрізняються [8]. Кількість кластерів може варіюватися від одного, що представляє групу з усіма даними, до загальної кількості екземплярів у наборі даних. Таким чином, чим більше кластерів, тим більш схожими є об'єкти всередині кожного кластера.

Процес кластеризації вимагає метрики для класифікації об'єктів, тобто потрібна міра схожості для порівняння різних екземплярів. У кластеризації ця міра називається «міра відстані». Відстань між двома об'єктами служить мірою їхньої схожості. Залежно від характеру задачі, найпоширенішими мірами відстані є «евклідова відстань» та «косинусна відстань».

Однак зазначені міри відстані можуть бути застосовані лише до числових даних для обчислення відстані між об'єктами. Тому процес виділення ознак є критично важливим для кластеризації, оскільки він забезпечує перетворення даних, яке дозволяє обчислювати міри відстані. У рамках цього проекту для кластеризації використовувались евклідова та косинусна відстані, які будуть детальніше описані в наступних пунктах.

Підходи до кластеризації оцінюються на основі таких характеристик, як відокремленість кластерів, розмірність даних, щільність, форма та дисперсія. Одна з ключових властивостей — дисперсія, яка вказує на ступінь варіації всередині кластера. Менша дисперсія означає, що кластери є більш компактними та щільними, що робить їх легшими для аналізу та інтерпретації .

Існує кілька способів обчислення відстаней у задачах кластеризації, серед яких евклідова та косинусна відстані є найпоширенішими. У цьому проекті використовувалися саме ці два типи відстаней, і вони пояснюються детальніше нижче.

Евклідова відстань вимірює відстань між двома точками в просторі шляхом обчислення довжини прямої лінії, яка з'єднує ці дві точки. Для

двовимірного простору, наприклад, якщо є дві точки *p1* і *p2* з координатами *(x1 ,y1)* та *(x2 ,y2)*, евклідова відстань обчислюється за формулою [6]:

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.3}$$

Однак, у багатовимірних задачах, таких як текстова кластеризація, простір має значно більше вимірів. Це робить завдання n-вимірним. У загальному випадку, евклідова відстань між точками *p1* та *p2* з координатами у багатовимірному просторі обчислюється за наступною формулою [6]:

$$d(p_1, p_2) = \sqrt{\sum_{i=1}^{n} (p_{2,i} - p_{1,i})^2} \tag{2.4}$$

,

Де n – кількість вимірів,*p1,i* та *p2,i* – координати точок *p1* та *p2* в *i*-тому вимірі.

Косинусна відстань вимірює не пряму відстань між точками, як евклідова, а кут між векторами, які представляють ці точки у векторному просторі. Це особливо корисно при кластеризації тексту, де вектори слів використовуються для представлення документів або текстових даних.

Косинусна відстань визначається через косинус кута між двома векторами. Якщо є два вектори *A* та *B*, то косинус кута між ними обчислюється за допомогою скалярного добутку цих векторів, поділеного на добуток їхніх норм [10]:

$$\cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{2.5}$$

де:$A \cdot B$ – скалярний добуток векторів *A* і *B*,||A|| та ||B|| – норми (довжини) векторів *A* і *B*, θ – кут між векторами *A* і *B*.

Косинусна відстань вимірює подібність між векторами: якщо косинус кута близький до 1, то вектори майже паралельні (дуже подібні), а якщо косинус близький до 0, то вектори ортогональні (не подібні).

2.3.1 Метод K-середні

Перед тим, як пояснити, як працює алгоритм K-середніх, варто визначити його місце в категорії алгоритмів кластеризації. Алгоритми кластеризації в цій категорії створюють розділи у даних, які формують кластери. Ці типи алгоритмів особливо ефективні для великих наборів даних, але їхнім недоліком є необхідність попереднього визначення кількості кластерів [9].

З моменту свого створення алгоритм K-середніх став популярним і використовувався в різноманітних додатках для кластеризації. Він також зазнав багатьох змін, спрямованих на покращення його продуктивності. Одним із прикладів є варіація K-середніх, відома як K-середні++, яка буде використана в цьому проекті. Основна відмінність між цією варіацією та оригінальним алгоритмом полягає в методі вибору початкових позицій центроїдів. В оригінальному алгоритмі K-середніх ці позиції вибираються випадковим чином, тоді як у K-середніх++ використовується метод рандомізованого посіву, що підвищує продуктивність алгоритму. Цю реалізацію можна знайти в бібліотеці Python scikit-learn.

Алгоритм K-середніх використовується для кластеризації даних, і його робота полягає в тому, щоб знайти групи (кластери) у наборі даних. На початку алгоритму випадковим чином обираються k центроїдів — це центральні точки, які представляють кожен кластер. Центроїди розміщуються в просторі ознак, де представлені всі точки набору даних.

Далі, алгоритм виконує ітерації, доки не досягне збіжності. Це означає, що положення центроїдів перестануть змінюватися між ітераціями. Кожна ітерація складається з двох основних кроків:

Для кожної точки p_i з набору даних D знаходиться найближчий центроїд. Відстань обчислюється за допомогою обраної метрики (наприклад, евклідової відстані). Після цього точка p_i відноситься до того кластеру C_j, центроїд якого є найближчим.

Після того, як всі точки набору даних були віднесені до відповідних кластерів, для кожного кластеру C_j обчислюється новий центроїд. Це середнє значення всіх точок, що належать до цього кластеру. Центроїд оновлюється і переміщується в нове положення, яке відповідає обчисленому середньому значенню. Процес повторюється доти, доки центроїди перестануть змінювати свої координати, тобто алгоритм досягне збіжності.

З математичної точки зору, метою алгоритму є мінімізація суми квадратів помилок (SSE — Sum of Squared Errors). Ця функція вимірює загальну відстань між кожною точкою і центроїдом її кластеру. Функція SSE визначається рівнянням [6]:

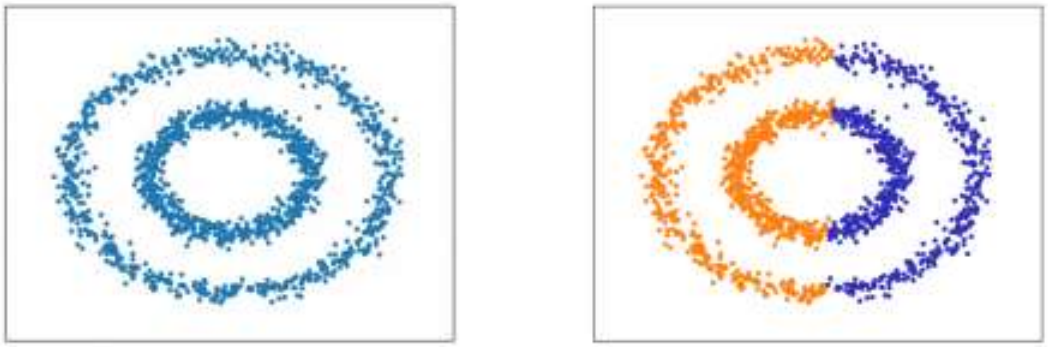
$$SSE(C) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - m_i\|^2 \tag{2.6}$$

де x_j – це точка з набору даних, C_i — це кластер і-го порядку (один із k кластерів), mi — це середнє значення точок у кластері Ci (його центроїд), $\|x_j - m_i\|^2$– це квадрат евклідової відстані між точкою x_j та центроїдом m_i.

Одним із недоліків алгоритму K-середніх є необхідність визначення кількості кластерів k, що може бути непростим завданням. У наступних розділах будуть описані методи для вибору правильного значення k. Інша проблема полягає в тому, що кластери, отримані за допомогою алгоритму K-середніх, завжди мають сферичну форму, що обмежує його застосування до даних із складнішою геометрією. Це обмеження алгоритму ілюструється на рисунку 2.4.

З рис.2.4 можемо помітити, що на зображенні є два окремих кластери, кожен з яких відповідає колу. Однак алгоритм K-середніх розіб'є ці два кола не правильно, а на дві рівні частини. Наприклад, частина кожного кола потрапить до першого кластеру, а інша частина — до другого. Це відбувається тому, що алгоритм K-середніх припускає, що всі кластери мають однакову дисперсію відносно центроїда, що не завжди відповідає реальності.

Ще один недолік алгоритму полягає в тому, що він не є детермінованим. Це означає, що початкові положення центроїдів обираються випадковим чином, що може призвести до різних результатів кластеризації кожного разу.



а) б)

Рисунок 2.4 Обмеження K-середніх: а)набір точок, що зображують два кола; б) Кластери, знайдені за допомогою K-середніх [6]

2.3.2 Метод DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) був розроблений Естером та ін. [10] з метою кластеризації даних, що містять шум. Цей алгоритм належить до категорії алгоритмів кластеризації на основі щільності, які намагаються виявити кластери у найбільш щільних областях простору ознак, а не просто розподілити простір на зони, як це роблять алгоритми, такі як K-середні.

DBSCAN отримує на вхід набір точок з даних, параметр Neighborhood (N) та параметр min_points. Параметр N визначає радіус, що оточує точку у наборі даних, тоді як min_points — це мінімальна кількість точок, необхідна для того, щоб точка вважалася основною. Під час виконання алгоритму кожна точка у наборі даних отримує одну з трьох міток: ядро, межа або шум. Основна точка — це точка, яка містить принаймні min_points у її околиці N. Гранична точка має принаймні одну основну точку у своєму оточенні, а шумова точка — це точка, яка не є ані ядром, ані межею (рисунок 2.5).

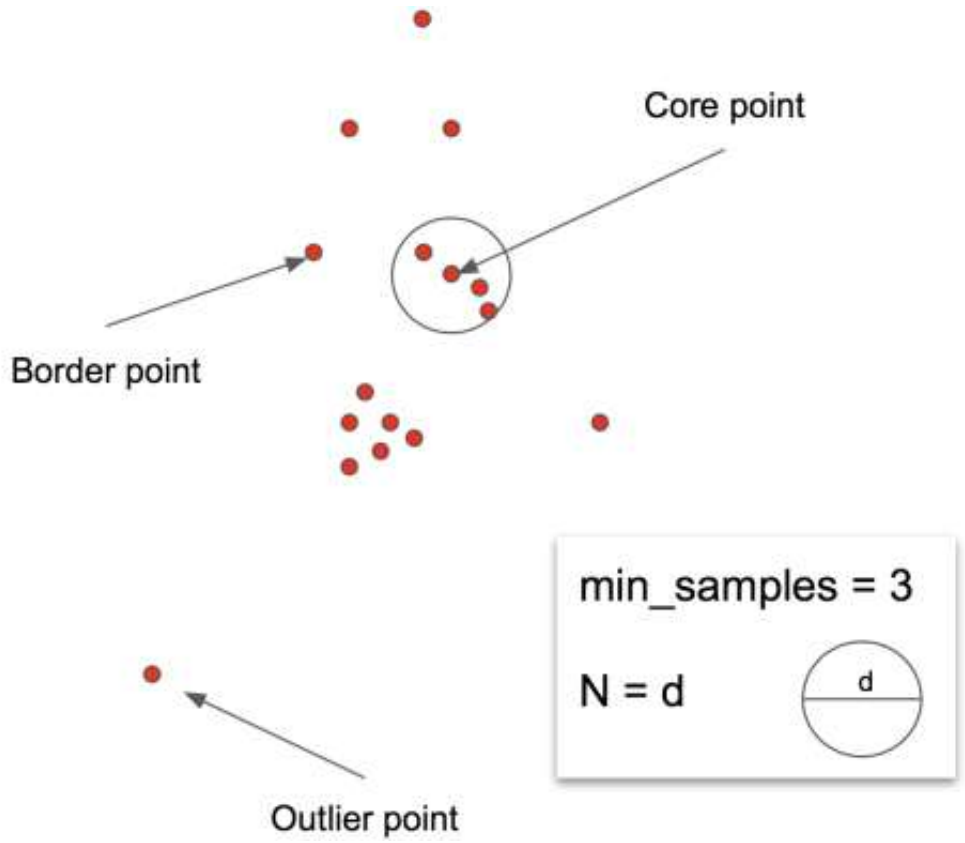


Рисунок 2.5 – Прив'язка точок DBSCAN [8]

Алгоритм працює, спочатку формуючи кластери з основних та граничних точок, а потім виконує пошук у глибину, починаючи з кожної основної точки. Цей процес повторюється, поки всі опорні точки не будуть віднесені до кластеру.

Переваги DBSCAN включають його здатність виявляти будь-яку кількість кластерів без попереднього визначення числа k, а також можливість виявляти та ігнорувати викиди. Проте одним із його головних недоліків є висока чутливість до параметра N. Якщо значення N занадто мале, розріджені кластери можуть бути визнані шумом, тоді як занадто велике N може призвести до об'єднання різних кластерів. Таким чином, для досягнення оптимальних результатів необхідно ретельно підбирати значення N.

Завдяки своїй популярності, особливо серед алгоритмів на основі щільності, DBSCAN став об'єктом багатьох досліджень. Як і K-середні, було

створено кілька варіацій цього алгоритму з метою покращення його продуктивності.

2.3.3 Агломеративна кластеризація

Агломеративна кластеризація є методом, що належить до зовсім іншої категорії кластеризації порівняно з попередніми згаданими методами. Це ієрархічний алгоритм кластеризації, що базується на поступовому об'єднанні даних (рис.2.6).

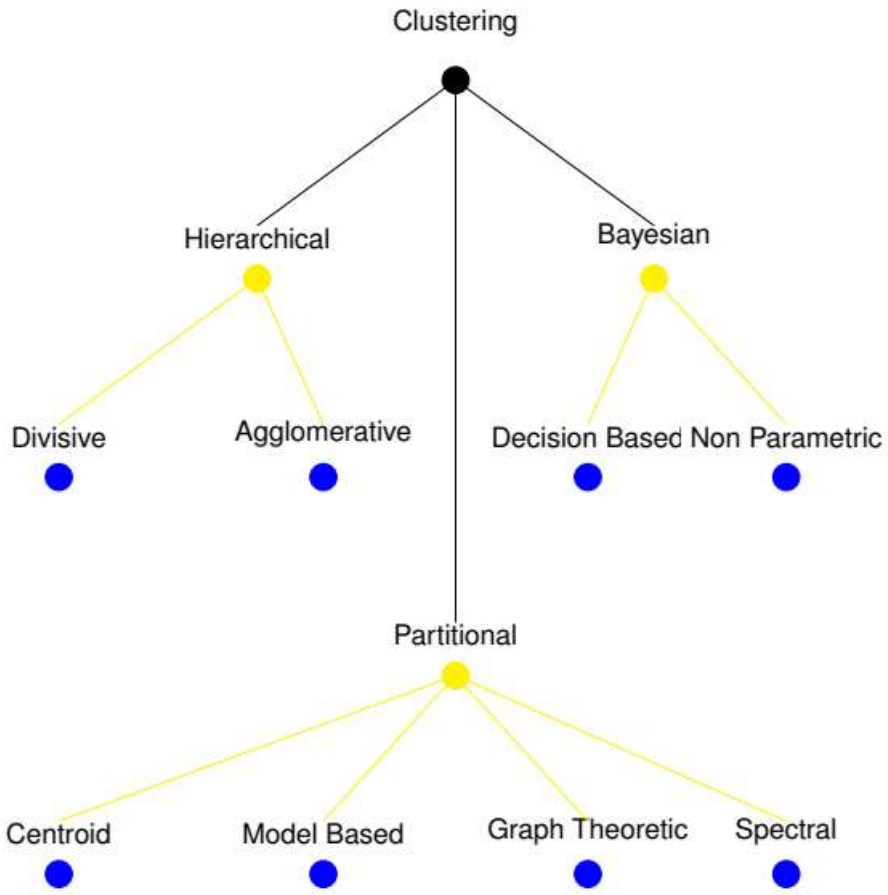


Рисунок 2.6 – Ієрархічна кластеризація [9]

Існує два основних підходи до застосування цієї категорії алгоритмів: зверху вниз: починаючи з одного кластера, що містить всі точки в просторі пошуку, цей підхід передбачає розбивання кластера до тих пір, поки кожна точка в наборі даних не стане окремим кластером. Вгору: починаючи з кількості кластерів, що дорівнює кількості точок у наборі даних, ми

об'єднуємо їх до тих пір, поки не досягнемо ситуації, коли залишиться лише один кластер, який містить всі точки.

Агломеративна кластеризація застосовує останній підхід, починаючи з множини окремих кластерів, а потім поступово об'єднуючи їх.

У результаті виконання алгоритму ієрархічної кластеризації формується ієрархічна структура, відома як дендограмма. Дендограма – це деревоподібна структура, де корінь представляє кластер, що містить усі екземпляри набору даних, а листя – це окремі екземпляри даних.

Існує кілька популярних методів для обчислення відстаней між кластерами (також відомих як критерії зв'язку), які використовуються в агломеративній кластеризації. Серед них можна виділити чотири основні методи: одиничний зв'язок, повний зв'язок, середній зв'язок та зв'язок Уорда. Кожен із цих методів має власний підхід до обчислення відстаней між наборами даних. Розглянемо ці методи детальніше.

Одиничний зв'язок (Single Linkage): цей метод визначає відстань між двома кластерами як мінімальну відстань між будь-якими двома точками, що належать до різних кластерів. Тобто обираються найближчі точки з двох кластерів A та B, і відстань між ними використовується для визначення схожості між кластерами. Формула для обчислення одиничного зв'язку має вигляд [11]:

$$d(A, B) = \min\{d(a_i, b_j)\}, \quad \forall a_i \in A,\, b_j \in B \tag{2.7}$$

Повний зв'язок (Complete Linkage): на відміну від одиничного зв'язку, цей метод обирає найбільш віддалені точки з двох кластерів. Тобто відстань між кластерами дорівнює найбільшій відстані між точками a_i з кластеру A та b_j з кластеру B. Формула для повного зв'язку [11]:

$$d(A, B) = \max\{d(a_i, b_j)\}, \quad \forall a_i \in A,\, b_j \in B \tag{2.8}$$

Середній зв'язок (Average Linkage): цей метод обчислює відстань як середнє значення всіх попарних відстаней між точками з двох кластерів *A* та *B*. Відстань між кластерами є середнім значенням відстаней між усіма точками з різних кластерів. Формула для середнього зв'язку [11]:

$$d(A,B) = \frac{1}{|A| \cdot |B|} \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} d(a_i, b_j) \tag{2.9}$$

де |A| та |B| — кількість точок у кластерах *A* і *B* відповідно.

Зв'язок Уорда (Ward's Linkage): цей метод мінімізує дисперсію всередині кластерів. Подібно до цільової функції K-середніх, він обчислює вартість об'єднання двох кластерів на основі суми квадратів різниць між точками та центром кластеру. Формула для обчислення вартості злиття двох кластерів [11]:

$$\text{merging\_cost} = \sum_{i \in A \cup B} \|x_i - m_{A \cup B}\|^2 - \sum_{i \in A} \|x_i - m_A\|^2 - \sum_{i \in B} \|x_i - m_B\|^2$$

(2.10)

2.3.4 Агломеративна кластеризація

Метод Single Pass (одноразове проходження) — це простий та швидкий алгоритм для кластеризації даних, особливо підходящий для великих наборів даних. Він працює на основі одного проходу через всі дані і кластеризує їх на основі певного порогу схожості. Основна ідея полягає в тому, щоб поступово створювати кластери під час обробки кожного нового зразка, не повертаючись до попередніх даних. Ось як працює цей метод:

– ініціалізується перший кластер, до якого додається перший об'єкт $x1$;

– для кожного наступного об'єкта xi вимірюється відстань або схожість між xi та всіма наявними кластерами. Відстань може бути виміряна за допомогою таких метрик, як евклідова відстань або косинусна

схожість.Якщо об'єкт xi достатньо близький (відповідає заданому порогу схожості) до якогось кластеру, він додається до цього кластеру. Якщо відстань перевищує поріг, створюється новий кластер, і об'єкт xi додається до нього.

Початково задається перший кластер $C1$, який містить перший елемент набору даних [10]:

$$C_1 = \{x_1\} \tag{2.11}$$

Для кожного нового елементу xi , обчислюється відстань до кожного наявного кластеру [10]:

$$d(x_i, C_j) = \min\{d(x_i, c) : c \in C_j\} \tag{2.12}$$

Метод Single Pass має перевагу в простоті та швидкості, оскільки він виконує кластеризацію за один прохід через дані. Однак його результат залежить від порядку подання даних і порогового значення, що може призвести до утворення неврівноважених або надто численних кластерів.

2.3.5 Кластеризація за зв'язками на графіку

Ієрархічна кластеризація, введена Джо Х. Уордом у 1963 році, є ще одним методом кластерного аналізу, відомим також як кластеризація за зв'язками графів. Цей метод дозволяє створювати ієрархії між кластерами, які можуть бути організовані в два типи: агломеративна і дивізіональна.

Агломеративна ієрархічна кластеризація починається зі створення набору різних кластерів для всього набору даних, а потім об'єднує їх попарно, поки не залишиться лише один кластер. Початковим набором кластерів може бути набір усіх спостережень. У цьому методі два кластери об'єднуються, якщо їхня відмінність є найменшою, що може бути обчислено різними способами, залежно від обраного критерію зв'язку.

У свою чергу, роздільна ієрархічна кластеризація є зворотним процесом: початково існує лише один кластер, який потім розбивається на

кілька. Для визначення, який кластер розбити, обчислюється середня несхожість кожного кластера, і обирається той, що має найбільшу різницю. Це робить роздільну кластеризацію більш обчислювально витратною, оскільки алгоритм K-середніх потрібно запускати на кожній ітерації.

Деякі з критеріїв зв'язності, що використовуються в ієрархічній кластеризації, включають:

– одиничний зв'язок: мінімізує відстань між спостереженнями в кожному кластері та його сусідами;

– повне зчеплення: мінімізує максимальну відстань між спостереженнями в кожному кластері та його сусідами;

– середній зв'язок: враховує середнє значення відстані між спостереженнями в двох кластерах;

– ворд-зв'язок: мінімізує загальну дисперсію в кожному кластері.

Однак одиничний зв'язок може призвести до неправильного об'єднання кластерів, оскільки він ґрунтується лише на найближчих спостереженнях, які можуть бути не достатніми для підтвердження близькості двох кластерів.

Кластеризація за зв'язками графів надає більше інформації порівняно з традиційними алгоритмами кластеризації. Використання дендрограми дозволяє візуалізувати, які кластери є найбільш схожими, а які — найбільш несхожими один з одним.

2.4 Методи валідації кластеризації

Одна з найзначніших відмінностей між неконтрольованим і контрольованим навчанням полягає у процесі валідації. У керованому навчанні ми відразу маємо доступ до правильної інформації, що дозволяє порівнювати результати моделі з правильними, а отже, робити висновки щодо її якості. Натомість, у неконтрольованому навчанні, зокрема в

кластеризації, відсутні попередні мітки, які допомагали б у порівнянні отриманих результатів із тими, що модель повинна була б створити.

У контексті кластеризації існують два основні способи оцінки моделі. Якщо нам відомо, які результати повинні бути у моделі, ми можемо запозичити деякі концепції з керованого навчання. Такі підходи називають зовнішніми, оскільки модель оцінюється за допомогою зовнішньої інформації. Однак, зазвичай, при використанні методів неконтрольованого навчання ми не маємо такої інформації і лише прагнемо знайти закономірності в даних. Тому нам необхідно використовувати метрики, які не залежать від зовнішніх даних, а здатні оцінити якість моделі. У випадку кластеризації це стосується якості створених розділів (кластерів) з точки зору компактності та відокремленості.

Метрика валідації, що використовувалася в цьому проекті, є внутрішньою і позначається коефіцієнтом силуету. Силуетний коефіцієнт дозволяє виміряти, наскільки добре визначені кластери, і його можна обчислити двома способами, як це пояснюється у документації до бібліотеки scikit-learn [11]:

- середня відстань між екземпляром та всіма іншими у тому ж кластері:

$$a(i) = \frac{1}{|C(i)| - 1} \sum_{j \in C(i), j \neq i} d(i, j), \tag{2.13}$$

де d(i,j) — це відстань між точками i і j (зазвичай використовується евклідова відстань), |C(i)| — це кількість точок у кластері C(i),сума береться по всіх точках j, які належать до кластера C(i), окрім самої точки i;

- середня відстань між екземпляром та всіма іншими у найближчому кластері [12]:

$$b(i) = \frac{1}{|C_{\text{next}}(i)|} \sum_{j \in C_{\text{next}}(i)} d(i, j), \tag{2.14}$$

де d(i,j) — це відстань між точками *i* і *j*,||C_{nex}t (i)| — це кількість точок у найближчому кластері C_{next} (i);

- коефіцієнт силуету оцінює, наскільки добре дані розділені на кластери. Нехай *a(i)* — середня відстань від об'єкта *i* до всіх інших об'єктів у тому ж кластері.

Нехай *b(i)* — середня відстань від об'єкта *i* до всіх об'єктів у найближчому кластері [12]:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}. \tag{2.15}$$

Метрика силуету є важливим інструментом для оцінки якості кластеризації, оскільки дозволяє виміряти баланс між компактністю кластерів (наскільки щільно розташовані точки всередині кожного кластера) і їхньою відокремленістю (як далеко вони знаходяться від інших кластерів).

2.4 Висновки до другого розділу

У другому розділі було проведено огляд основних методів кластеризації текстових даних, що є важливими інструментами в аналізі інформації та обробці природної мови. Кластеризація текстових даних дозволяє групувати схожі документи, що сприяє кращому розумінню великих обсягів інформації та виявленню прихованих структур у даних.

Розглянуто кілька основних підходів до кластеризації текстів, включаючи K-середні, ієрархічну кластеризацію, методи на основі зв'язків, а також алгоритми на основі глибокого навчання. Кожен із цих методів має свої переваги та недоліки, що робить їх більш чи менш придатними для різних типів текстових даних.

Аналіз критеріїв зв'язності та методів відстані, які використовуються в кластеризації, виявив, що вибір відповідного критерію може суттєво вплинути на якість кластеризації. Розуміння цих критеріїв є критично важливим для досягнення оптимальних результатів.

Результати кластеризації текстових даних мають широкий спектр застосувань, включаючи тематичне моделювання, інформаційний пошук, автоматичне привласнення категорій та аналітику настроїв. Це підкреслює важливість кластеризації в сучасному світі даних.

Одним з основних викликів у кластеризації текстових даних є визначення оптимальної кількості кластерів, що може бути складним завданням. Крім того, специфіка текстових даних, таких як векторизація та обробка природної мови, потребує ретельного підходу для досягнення точності та релевантності результатів.

Використання нових технологій, таких як глибоке навчання та обробка природної мови, відкриває нові горизонти для покращення алгоритмів кластеризації. Інтеграція цих технологій може призвести до більш точних та адаптивних методів кластеризації текстових даних.

3 РОЗРОБКА ІНСТРУМЕНТУ ДОСЛІДЖЕННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ БЕЗ ВЧИТЕЛЯ ДЛЯ КЛАСТЕРИЗАЦІЇ ТЕКСТОВИХ ДАНИХ УКРАЇНСЬКОЮ МОВОЮ

3.1 Інтерпритація природної мови

Розуміння природної мови (Natural Language Understanding, NLU) — це процес вилучення смислу з тексту, що є важливою складовою частиною обробки природної мови (Natural Language Processing, NLP) і є досить складним завданням. Для ефективної обробки природної мови потрібно не лише розуміти окремі слова та речення, але й враховувати контекст усього тексту (рис.3.1). Це одна з причин, чому в обробці природної мови використовується більше, ніж просто словник у вигляді великої бази даних.

Рисунок 3.1 – Категорії лінгвістики

Згідно з дослідженнями, розуміння природної мови базується на різних мовних категоріях як показано на рис.3.1. Фонологічний аналіз

вивчає вимову слів (фонетика) та структуру звуків (фонологія). Наприклад, у слові "кот" фонеми 'к', 'о', 'т' утворюють звукову структуру слова.

Морфологічний аналіз стосується структури та утворення слів з морфем. Наприклад, слово "грати" складається з кореня "грат-" і суфікса "-и".

Лексичний аналіз вивчає значення слів та їхні відношення.

Синтаксичний аналіз оцінює структуру речень відповідно до формальних правил граматики, формуючи дерево синтаксичного аналізу. Наприклад, у реченні "Собака стрибнула через паркан" структура виглядає так:

- собака (підмет);
- стрибнула (дієслово);
- через (прийменник);
- паркан (додаток).

Семантичний аналіз зосереджений на значеннях речень. Цей процес візуалізовано на рисунку 3.2.

Граматичний процес словотворення, що виражає числові, відмінкові, родові або модальні характеристики слова, називається флексією. Канонічна форма відмінюваного слова, або лема, в комбінації з її формою відомою як лексема.

Рисунок 3.2 – Двофазний процес розуміння природної мови

3.2 Попередня обробка текстових даних

Попередня обробка тексту є одним із найважливіших етапів при використанні методів машинного навчання для роботи з текстовими даними. Основна мета цього етапу полягає у перетворенні неструктурованого тексту у формат, який є зрозумілим та інтерпретованим для алгоритмів машинного навчання. Це включає в себе видалення непотрібних елементів, нормалізацію та структурування даних для подальшої аналітики та кластеризації.

Опишемо основні кроки попередньої обробки тексту. Текстові документи часто містять непотрібну інформацію, таку як спеціальні символи, пунктуація, цифри, дати, HTML-теги тощо. Видалення цих елементів дозволяє уникнути плутанини під час аналізу та зосередитись на змістовних аспектах тексту. Наприклад, дати або номери телефонів, що не мають суттєвої значущості у кластеризації, можуть бути видалені або перетворені у відповідний формат.

Стоп-слова — це поширені слова, які зазвичай не несуть змістовного навантаження в контексті аналізу тексту (наприклад, "і", "але", "бо", "який"). Видалення стоп-слів є стандартною процедурою у більшості задач з аналізу тексту, оскільки вони не сприяють виявленню важливих патернів у даних і можуть заплутати модель. Видалення цих слів допомагає:

– зменшити обсяг даних для обробки;

– підвищити продуктивність алгоритмів за рахунок виключення непотрібних елементів;

– поліпшити точність моделей, які сприймають тільки змістовні слова.

Якщо текст містить 20-30% стоп-слів, їх видалення може зменшити розмір даних на 40-50%, що суттєво оптимізує процес обробки.

Нормалізація тексту включає такі процеси, як приведення всіх слів до нижнього регістру (маленьких літер), видалення зайвих пробілів і виправлення помилок друку. Ця процедура дозволяє уникнути проблем, пов'язаних з різними варіантами написання одного і того ж слова.

Стемінг — це процес скорочення слів до їх базової форми (основи або кореня), що дозволяє об'єднати різні форми слова ("працювати", "працював", "працює" — після стемінгу можуть бути скорочені до "прац".). Лематизація виконує подібну задачу, але з урахуванням граматичних правил мови. Обидва методи дозволяють зменшити кількість варіацій одного слова, що підвищує ефективність кластеризації та аналізу тексту.

Токенізація — це процес розбиття тексту на окремі слова або фрази, які називаються токенами. Токени є основними одиницями аналізу тексту, які передаються на всі подальші етапи обробки. Важливо, що токенізація повинна враховувати не тільки розділові знаки, а й контекстні особливості мови.

Після того як текст пройшов через попередні кроки обробки, його потрібно перетворити у числову форму, яку зможе обробляти алгоритм. Існують кілька підходів до цього перетворення:

– Bag of Words (BOW) — модель, що представляє текст як набір слів без урахування їхнього порядку. Кожне слово перетворюється у вектор, де кожен елемент відповідає частоті слова у тексті;

– TF-IDF (Term Frequency-Inverse Document Frequency) — це метод, що враховує не тільки частоту слова в документі, але й те, як часто це слово зустрічається у всіх інших документах, що дозволяє визначити важливість слова для певного документа;

– Word Embeddings (векторні представлення слів) — метод, який перетворює слова в багатовимірні вектори на основі їх семантичної

схожості. Популярні методи для створення векторних представлень слів включають Word2Vec, GloVe та FastText.

На рис.3.3 наведено структурну схему етапів попередньої обробки текстових даних, а на рис.3.4 фрагмент програмної реалізації цих етапів.

Рисунок 3.3 – Структурна схема етапів попередньої обробки текстових даних

Опишемо схему на рис.3.1. Текстові документи - вхідний текстовий документ. Токенізація – розбиття тексту на окремі слова або токени. Видалення розділових знаків – усунення пунктуації, яка не несе смислового навантаження. Видалення цифр – видалення числових символів, якщо вони не потрібні для аналізу. Видалення стоп-слів – видалення часто вживаних слів (наприклад, прийменників, сполучників), які не додають змістовної

інформації. Видалення складних слів – йдеться про видалення рідковживаних або складних для аналізу слів.

```
import nltk
import pymorphy2

# Завантажуємо стоп-слова для української мови
nltk.download('stopwords')
from nltk.corpus import stopwords

# Створюємо набір стоп-слів для української мови
ukrainian_stop_words = set(stopwords.words('ukrainian'))

# Ініціалізуємо лематизатор для української мови
morph = pymorphy2.MorphAnalyzer(lang='uk')

# Приклад тексту українською мовою
text = "Це приклад попередньої обробки тексту, який включає видалення стоп-слів та лематизацію."

# Токенізація тексту (розбиття на слова)
words = text.split()

# Видалення стоп-слів та лематизація
filtered_text = [morph.parse(word)[0].normal_form for word in words if word.lower() not in ukrainian_stop_words]

# Виведення результату
print(filtered_text)
```

Рисунок 3.4 – Фрагмент коду програмної реалізації етапу попередньої обробки текстових даних

3.3 Методика кластеризації текстових даних

Кластеризація текстових документів є однією з найфундаментальніших задач інтелектуального аналізу текстів. Цей процес може бути корисним для таких задач, як пошук інформації, вилучення тем, організація документів та підтримка перегляду. Основна мета кластеризації полягає в тому, щоб згрупувати текстові документи в категорії (кластери) на основі їхньої схожості, таким чином, щоб тексти всередині одного кластера були більш схожими, ніж тексти між різними кластерами. Методика кластеризації може бути застосована на різних рівнях деталізації тексту, таких як рівень документу, абзацу, речення або терміну.

Кластеризація текстових даних є викликом через їх неструктурованість. Текстові дані не відповідають жорстким структурам, як це відбувається зі структурованими даними (наприклад, у базах даних). Для того щоб комп'ютери могли працювати з такими даними, було запроваджено обробку природної мови (Natural Language Processing, NLP), що включає методи для перетворення тексту в машинно-читабельний формат.

Запропонована методика кластеризації текстових даних представлена на рис.3.5 і складається з п’яти етапів.

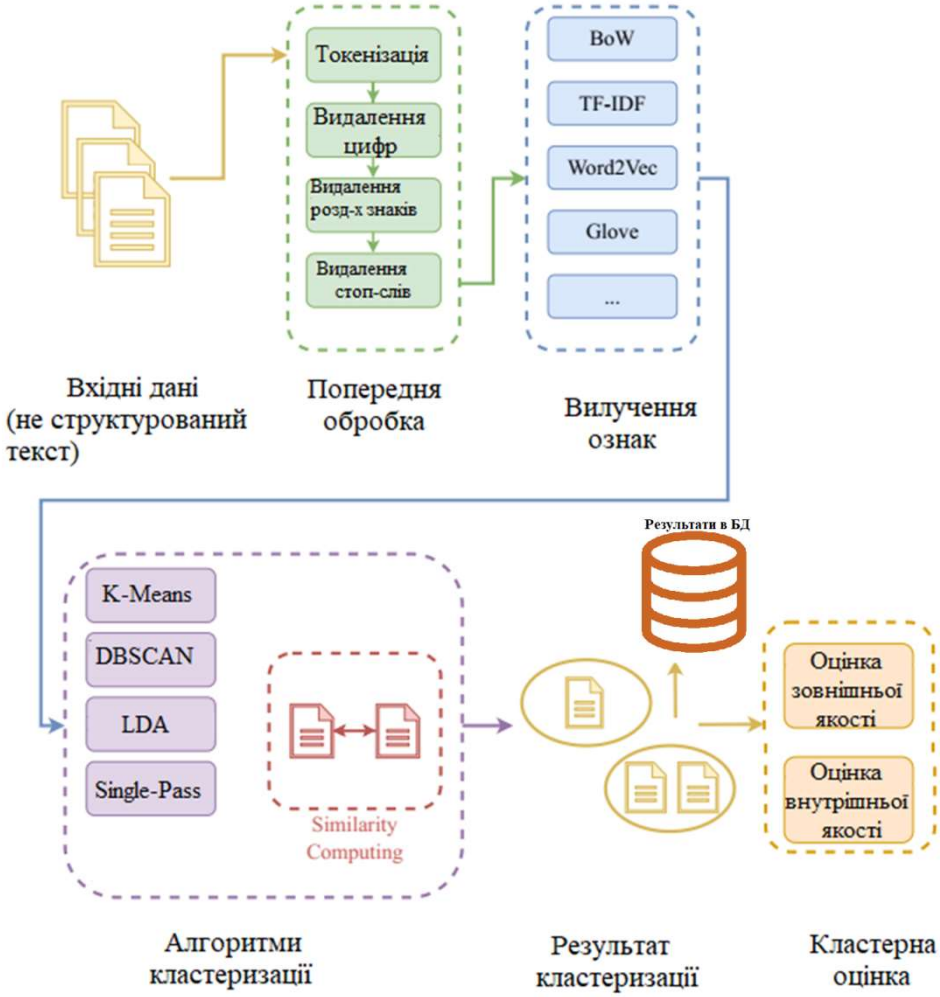


Рисунок 3.5 – Методика кластеризації текстових даних

Етап 1 – збір та попередня обробка даних. Важливим першим кроком є збір відповідного набору даних. Текстові дані повинні бути попередньо

оброблені для підвищення їх якості, що впливає на точність і ефективність подальшого аналізу. До ключових методів попередньої обробки належать нормалізація тексту, видалення стоп-слів, стемінг та лематизація.

Етап 2 – представлення текстових даних. Для того щоб можна було застосовувати алгоритми кластеризації, текстові дані повинні бути представлені у числовому вигляді. Однією з найпростіших моделей є модель Bag-of-Words (BoW), яка перетворює текст у вектори термінів. Інші популярні підходи включають TF-IDF, який враховує вагу термінів залежно від їх частоти в текстах, та вбудовування слів (word embeddings), які зберігають семантичні зв'язки між словами.

Етап 3 – визначення схожості текстів. Для кластеризації важливо визначити міру схожості між текстами. Найпоширенішою метрикою є косинусна подібність, яка вимірює кут між векторами документів і не залежить від їх довжини. Інші варіанти включають евклідову відстань та Якову схожість.

Етап 4 – Алгоритми кластеризації. В роботі запропоновано дослідити методи кластеризації без вчителя: K-Means, DBSCAN, LDA та Single-Pass. На цьому етапі набір документів розподіляється на k кластерів на основі їхньої подібності, мінімізуючи середню відстань між точками всередині кластера та його центроїдом.

Етап 5 – Оцінка якості кластеризації. Після проведення кластеризації важливо оцінити її якість. Для цього використовуються такі метрики, як індекс Сілуета, коефіцієнт Rand, та інші міри якості, що оцінюють внутрішню когерентність кластерів та їхню відокремленість.

3.3 Опис програмного інструменту

Розроблений програмний інструмент є графічним додатком, призначеним для кластеризації текстових даних. Він забезпечує

інтерактивний інтерфейс, який дозволяє користувачеві завантажувати текстові файли, вибирати різні методи кластеризації та візуалізувати результати аналізу. Інструмент реалізовано з використанням бібліотек Python, таких як sklearn [13] для алгоритмів машинного навчання, tkinter [15] для графічного інтерфейсу, matplotlib [16] для візуалізації даних та nltk [17] для обробки природної мови.

Архітектура програми складається з кількох ключових компонентів. Перший компонент – це графічний інтерфейс (GUI), в якому використано бібліотеку tkinter для створення інтуїтивно зрозумілого графічного інтерфейсу. Користувач може взаємодіяти з додатком через вікна, кнопки та текстові поля. Інтерфейс забезпечує простий процес завантаження текстових файлів, вибору алгоритмів кластеризації та перегляду результатів.

Другий компонент – обробка тексту, що включає функції для завантаження, обробки та векторизації тексту. Процес обробки передбачає токенізацію, яка дозволяє розподілити текст на окремі слова або фрази, а також фільтрацію стоп-слів, що включає видалення слів, які не несуть змістовного навантаження (наприклад, "і", "в", "на"). Векторизація тексту здійснюється за допомогою TfidfVectorizer, що дозволяє оцінити важливість слів у контексті документів.

Третій компонент – це алгоритми кластеризації, серед яких реалізовано кілька методів, таких як KMeans, DBSCAN, LDA (Latent Dirichlet Allocation) та Single-Pass. Кожен алгоритм виконує свою обробку даних і генерує результати, які згодом використовуються для візуалізації. KMeans — це алгоритм, що групує дані на основі відстані між ними; DBSCAN класифікує точки на основі щільності; LDA виявляє тематичні структури у великому обсязі текстових даних, а Single-Pass — простий метод кластеризації, який обробляє дані в один прохід.

Четвертий компонент — візуалізація результатів. Візуалізація кластерів здійснюється за допомогою методів з бібліотеки matplotlib, які дозволяють графічно відобразити результати кластеризації. Це допомагає користувачеві зрозуміти структуру кластерів та їхні взаємозв'язки, що є важливим аспектом для подальшого аналізу даних.

Останнім компонентом є база даних, для збереження вхідних текстових даних і результатів кластеризації. Реалізація інтеграції з базою даних дозволяє зберігати та відновлювати інформацію, що забезпечує ефективність роботи з великими обсягами даних.

Структура проекту організована таким чином, що в ній виділяється кілька ключових папок як показано на рис.3.6.

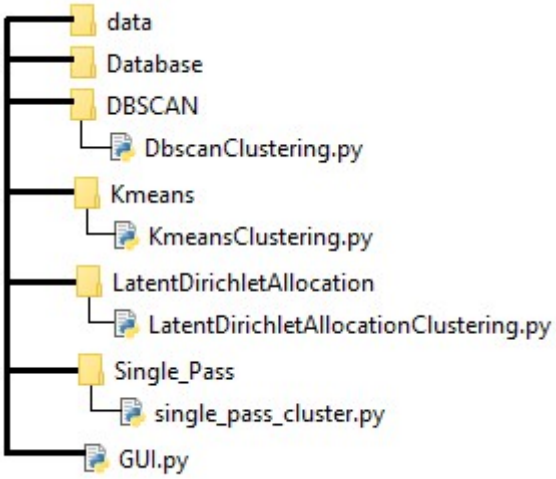


Рисунок 3.6 – Структура програмного інструменту

Папка data містить приклади тексту для аналізу та список стоп-слів, які використовуються для обробки текстових даних. Папки K-Means, DBSCAN, LDA та Single-Pass містять програмний код реалізованих методів кластеризації, що дозволяє структурувати код відповідно до обраних алгоритмів. Нарешті, файл GUI.py містить код для графічного інтерфейсу, що забезпечує взаємодію користувача з додатком, включаючи функції

завантаження файлів, вибору алгоритму кластеризації та візуалізації результатів.

Діаграми варіантів використання (Use Case Diagrams) є важливим елементом моделювання системи, який дозволяє візуалізувати взаємодію між користувачами (акторів) і системою (рис.3.7). Вони допомагають зрозуміти вимоги до системи, визначити основні функціональні можливості та проаналізувати, як різні користувачі можуть взаємодіяти з програмним інструментом.

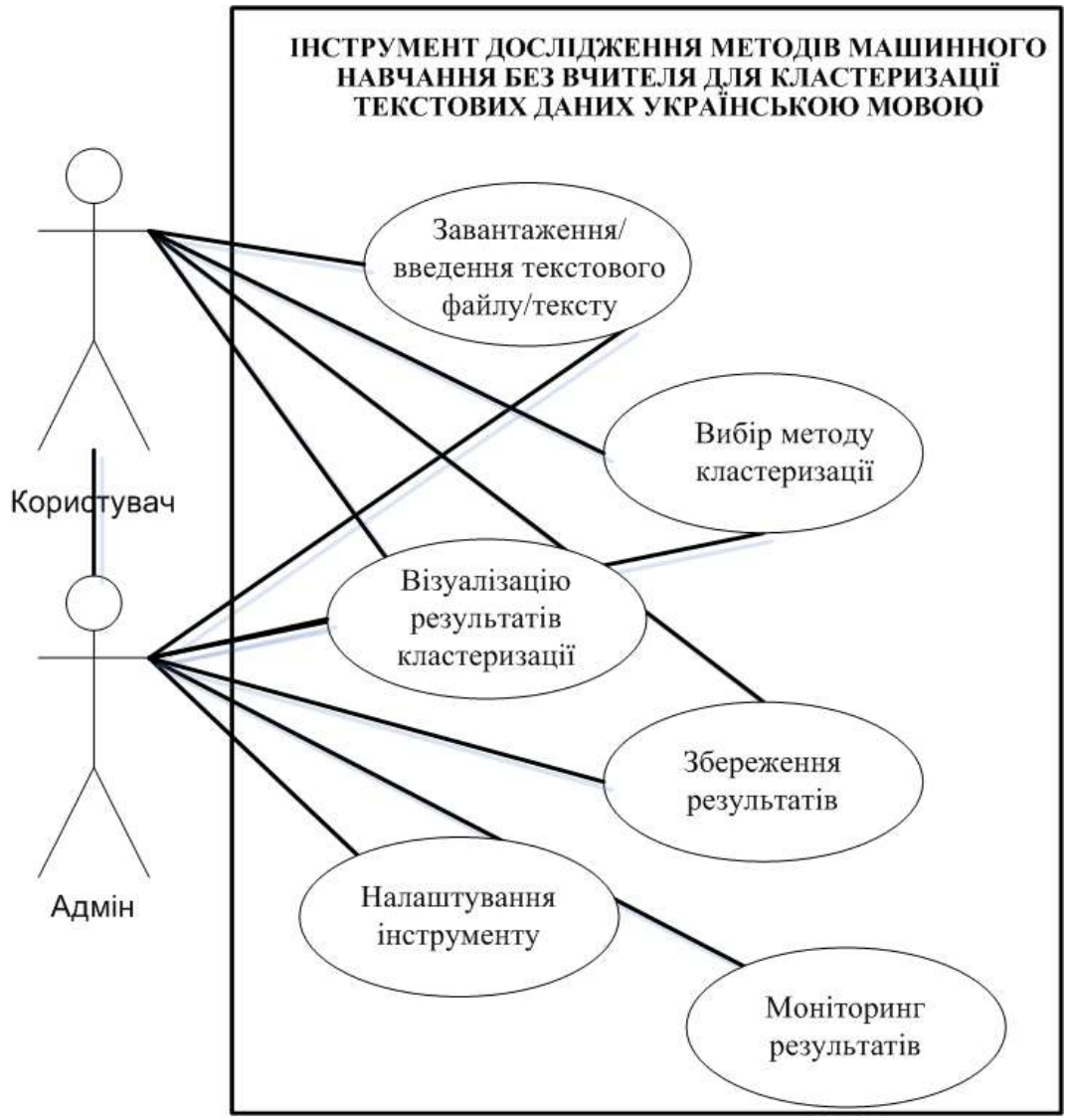


Рисунок 3.7 – Діаграми варіантів використання програмного інструменту

Основними компонентами діаграми варіантів використання є актори, варіанти використання та взаємозв'язки. Актори – це зовнішні сутності, які взаємодіють із системою. В даному контексті акторами є користувачі програми, які можуть бути як досвідченими аналітиками, так і новими користувачами. Кожен актор представляє різний тип користувача, який може виконувати певні дії в системі. Варіанти використання – це функції або дії, які користувачі можуть виконувати в системі. У нашому випадку варіанти використання включають завантаження текстового файлу, вибір методу кластеризації, візуалізацію результатів кластеризації, збереження результатів у базі даних, налаштування інструменту та моніторинг результатів.

Діаграма варіантів використання для розробленого програмного інструменту може виглядати так: користувач (актор 1) може виконувати такі дії, як завантаження текстового файлу, натиснувши кнопку "Завантажити"; вибір алгоритму кластеризації з доступного списку (KMeans, DBSCAN, LDA, Single-Pass); візуалізацію результатів кластеризації у графічному вигляді, натиснувши кнопку "Візуалізувати"; а також збереження результатів кластеризації у базі даних через відповідний інтерфейс. Адміністратор (актор 2) може налаштовувати параметри інструменту для оптимізації його функцій та відслідковувати результати та стан кластеризації.

Переваги використання діаграм варіантів використання є численними. По-перше, вони надають чітке уявлення про вимоги до системи, що допомагає розробникам і замовникам зрозуміти, як система повинна працювати. По-друге, аналіз діаграм може виявити потенційні пропуски в вимогах або функціональності, що дозволяє їх виправити на ранніх етапах розробки. По-третє, діаграми сприяють покращенню комунікації між розробниками, аналітиками та замовниками, забезпечуючи спільне розуміння системи. По-четверте, варіанти використання слугують основою

для розробки тестових сценаріїв, що дозволяє перевірити правильність реалізації системи.

3.3 Інтеграція бази даних

Для забезпечення збереження вхідних текстових даних і результатів кластеризації, в нашій системі використовуватиметься база даних SQLite. Це дозволить ефективно зберігати дані, надавати до них легкий доступ для подальшої обробки та аналізу, а також забезпечувати масштабованість для великих наборів даних.

Для забезпечення коректної роботи з даними, будуть створені дві основні таблиці. Texts – зберігає текстові документи, введені користувачем або завантажені з файлів. ClusteringResults – зберігає результати кластеризації кожного тексту, включаючи метод, використаний для кластеризації, та ідентифікатор кластеру.

Структура розроблених таблиць представлено в табл.31 – табл. 3.3.

Таблиця 3.1 – Структура даних «Texts»

Поле	Тип	Примітка
Id (PRIMARY KEY)	INTEGER	Унікальний ідентифікатор для кожного текстового документа
text_content	TEXT	Сам текст, введений користувачем або завантажений з файлу

Таблиця 3.2 – Структура даних «ClusteringResults»

Поле	Тип	Примітка
Id (PRIMARY KEY)	INTEGER	Унікальний ідентифікатор для результатів кластеризації
text_id	INTEGER	Ідентифікатор тексту з таблиці Texts, для якого збережено результат кластеризації
method	TEXT	Назва методу кластеризації, наприклад, KMeans, DBSCAN тощо
cluster_id	INTEGER	Ідентифікатор кластеру, до якого віднесено текст.
file_name	TEXT	Назва файлу, якщо текст був завантажений з файлу. Якщо текст введений користувачем, значення може бути порожнім

Таблиця 3.3 – Структура даних «Metrics»

Поле	Тип	Примітка
Id (PRIMARY KEY)	INTEGER	Унікальний ідентифікатор кожної метрики
clustering_result_id	TEXT	Ідентифікатор результату кластеризації з таблиці ClusteringResults, для якого розраховано метрики.
metric_name	TEXT	Назва метрики, наприклад, silhouette_score або davies_bouldin_score.
metric_value	REAL	Значення метрики для відповідного результату кластеризації.

На рисунку 3.8 наведено фрагмент коду створення таблиць

```
CREATE TABLE IF NOT EXISTS Texts (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    text_content TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS ClusteringResults (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    text_id INTEGER NOT NULL,
    method TEXT NOT NULL,
    cluster_id INTEGER NOT NULL,
    file_name TEXT,
    FOREIGN KEY (text_id) REFERENCES Texts(id)
);
```

Рисунок 3.8 – Фрагмент створення SQL-запити для створення таблиць

Таблиця ClusteringResults має зовнішній ключ text_id, який посилається на поле id у таблиці Texts (рис.3.9). Це дозволяє зв'язувати кожен результат кластеризації з відповідним текстом, що було введено або завантажено в систему. Таблиця Metrics має зовнішній ключ на ClusteringResults (clustering_result_id), що вказує на конкретний результат кластеризації, для якого розраховано метрику.

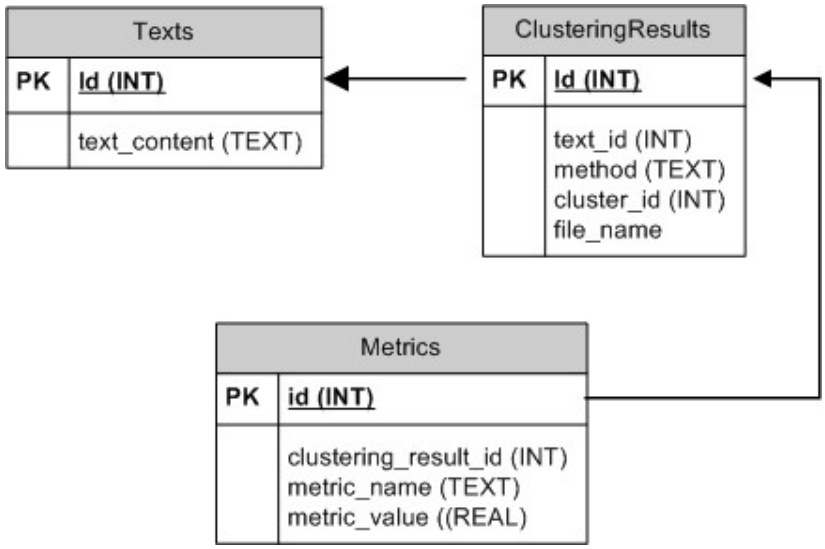


Рисунок 3.9 – Зв’язки БД

3.4 Розробка графічного інтерфейсу програмного інструменту для кластеризації текстів

Графічний інтерфейс для кластеризації текстів на Python розроблено з використанням бібліотеки Tkinter [15], яка є вбудованим засобом для створення простих інтерфейсів. Основна мета інтерфейсу — спростити взаємодію користувача з процесом кластеризації текстів, що включає завантаження текстових файлів, вибір методу кластеризації та відображення результатів. Водночас, забезпечується зберігання текстів та результатів кластеризації в базі даних для подальшого використання і аналізу [13,14].

Програма починається зі створення класу TextClusteringApp, який містить усі основні компоненти і функціонал інтерфейсу. Після ініціалізації вікна програми з використанням методу root.title задається його розмір та заголовок. Лістінг програми наведено у додатку А.

Для роботи з базою даних SQLite також на етапі ініціалізації встановлюється підключення до бази. Використовується метод create_connection, який перевіряє, чи може програма підключитися до бази даних clustering_results.db. Якщо підключення успішне, створюються необхідні таблиці для збереження текстів та результатів кластеризації через метод create_tables. Таблиці зберігають тексти та результати кластеризації, що дає змогу здійснювати пошук за методом або кластером у майбутньому.

Користувачеві надається два варіанти введення тексту: або ввести його вручну у текстове поле, або завантажити з текстового файлу за допомогою кнопки "Завантажити файл". Завантаження файлу здійснюється за допомогою функції upload_file, яка використовує діалог вибору файлу з бібліотеки filedialog. Після вибору файлу його вміст читається та вставляється у текстове поле для подальшої обробки.

Тексти, що вводяться або завантажуються, автоматично зберігаються в базі даних через функцію save_text, яка повертає унікальний ідентифікатор тексту (ID). Це дозволяє в майбутньому зберігати та пов’язувати результати кластеризації з конкретним текстом.

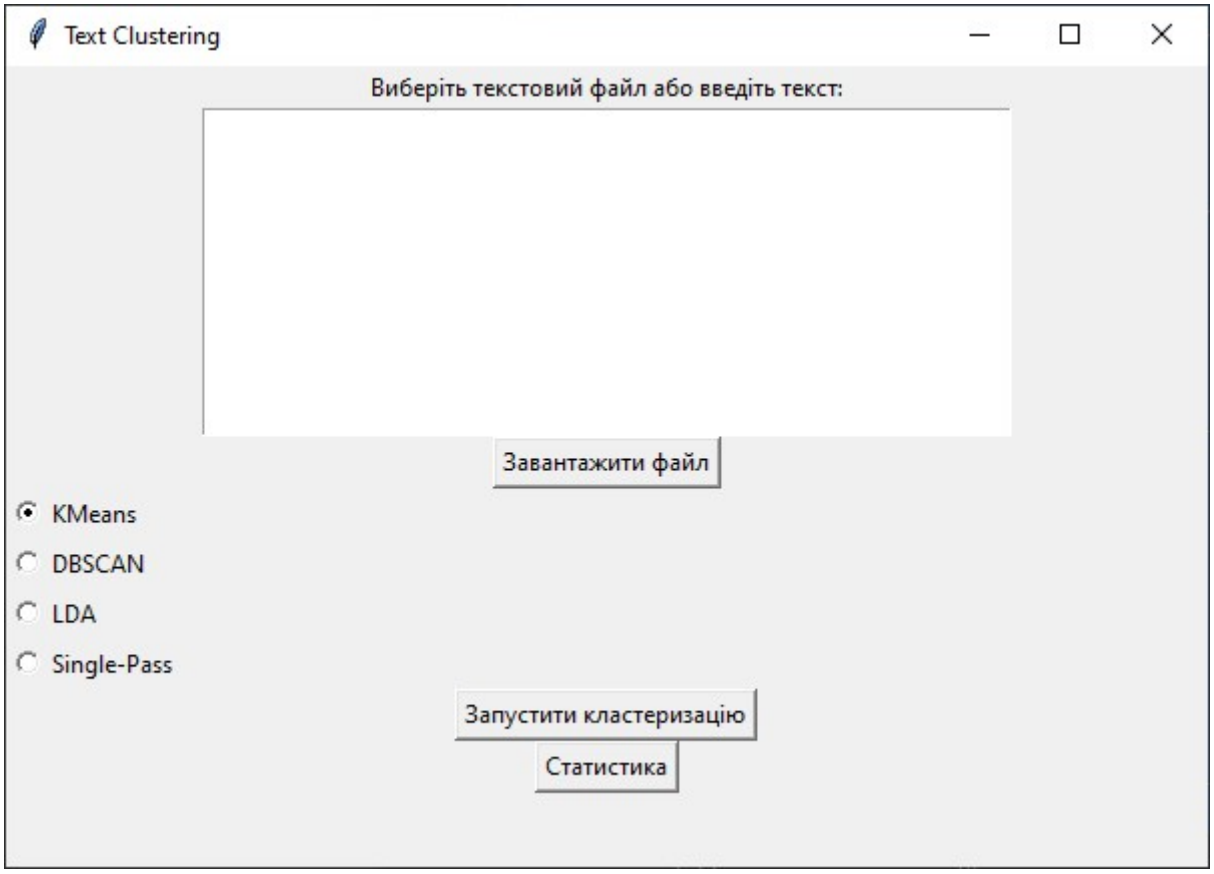


Рисунок 3.10 – Графічний інтерфейс програмного інструменту

Графічний інтерфейс, який представлено на рис.3.10 надає користувачеві можливість вибрати один з кількох методів кластеризації, які реалізовані у програмі:

– KMeans — алгоритм кластеризації, який формує певну кількість кластерів, мінімізуючи відстань між точками всередині кластеру;

– DBSCAN — алгоритм для виявлення кластерів на основі щільності точок;

– LDA (Latent Dirichlet Allocation) — метод тематичного моделювання, який групує тексти за темами;

– Single-Pass — простий метод кластеризації на основі проходження одного разу по всіх документах.

Кожен з цих методів можна вибрати за допомогою радіокнопок, і обраний метод зберігається у змінній cluster_method.

Після вибору методу кластеризації користувач може запустити процес за допомогою кнопки "Запустити кластеризацію". Функція cluster_text обробляє введені або завантажені тексти, виконує їх попередню обробку, яка включає токенізацію та видалення стоп-слів, і далі застосовує вибраний метод кластеризації.

Попередня обробка текстів виконується через функцію preprocess_text, яка використовує бібліотеку nltk для токенізації та фільтрації стоп-слів. Стоп-слова завантажуються заздалегідь з файлу через метод load_stopwords. Це дозволяє підготувати текст для подальшого використання у моделях кластеризації.

Тексти векторизуються за допомогою методу TfidfVectorizer з бібліотеки sklearn. Векторизація необхідна для того, щоб перетворити текст у числове представлення, з яким можуть працювати алгоритми кластеризації.

Результати кластеризації відображаються користувачеві у вигляді візуалізації кластерів на основі даних, отриманих після застосування обраного методу. Для цього використовуються методи, наприклад, kmeans_clustering, dbscan_clustering, lda_clustering. Ці методи не лише виконують кластеризацію, але й оцінюють якість кластеризації за допомогою таких метрик, як коефіцієнт силуету (silhouette_score) або індекс Девіса-Боулдінга (davies_bouldin_score). Також передбачено відображення результатів на двовимірній площині за допомогою бібліотеки matplotlib.

Окрім цього, результати зберігаються в базу даних. Для цього використовується функція save_clustering_result, яка перевіряє, чи вже існує результат для конкретного тексту і методу, і якщо такого немає, зберігає нові результати кластеризації у відповідну таблицю бази даних.

Однією з важливих функцій графічного інтерфейсу є можливість перегляду статистики проведених кластеризацій. Для цього передбачена кнопка "Статистика" (рис. Б1), яка відкриває нове вікно і показує таблицю з агрегованими результатами за кожним методом кластеризації. Статистика відображає метод, ідентифікатор кластера, кількість текстів у кожному кластері та перелік текстових файлів або введених текстів.

Дані для цієї таблиці витягуються з бази даних за допомогою SQL-запиту, а для відображення використовується компонент ttk.Treeview, який дозволяє створювати таблиці у Tkinter з можливістю прокручування.

3.5 Експериментальні дослідження

3.5.1 Кластеризація тексту за допомогою методів без навчання

Для аналізу та кластеризації текстових даних використовємо методи без навчання: K-means, DBSCAN, LDA та Single Pass. На першому експерименті ми використовуємо бібліотеку sklearn.cluster для реалізації вище описаних методів. Завантажуємо файл у форматі .txt і запускаємо кожен метод, при цьому ми не застосовуємо попередню обробку до текстових даних.

Результат методу K-means наведено на рис.3.11. В результаті отримали три кластери. Кластер 0: Теми: завжди, україні, осінь, золота, тепла. Цей кластер зосереджений на сезонних змінах і особливостях осені в Україні. Кластер 1: Теми: світі, інформаційних, триває, технологій,

розвиток. Кластер зосереджений на технологіях, їх розвитку і їх впливі на світ. Тут присутні тексти, що пов'язані з інформаційними технологіями.

Кластер 2: Теми: та, європи, українська, своїми, відома. Цей кластер містить інформацію про українську культуру, мову та традиції, а також упомінання про інновації.

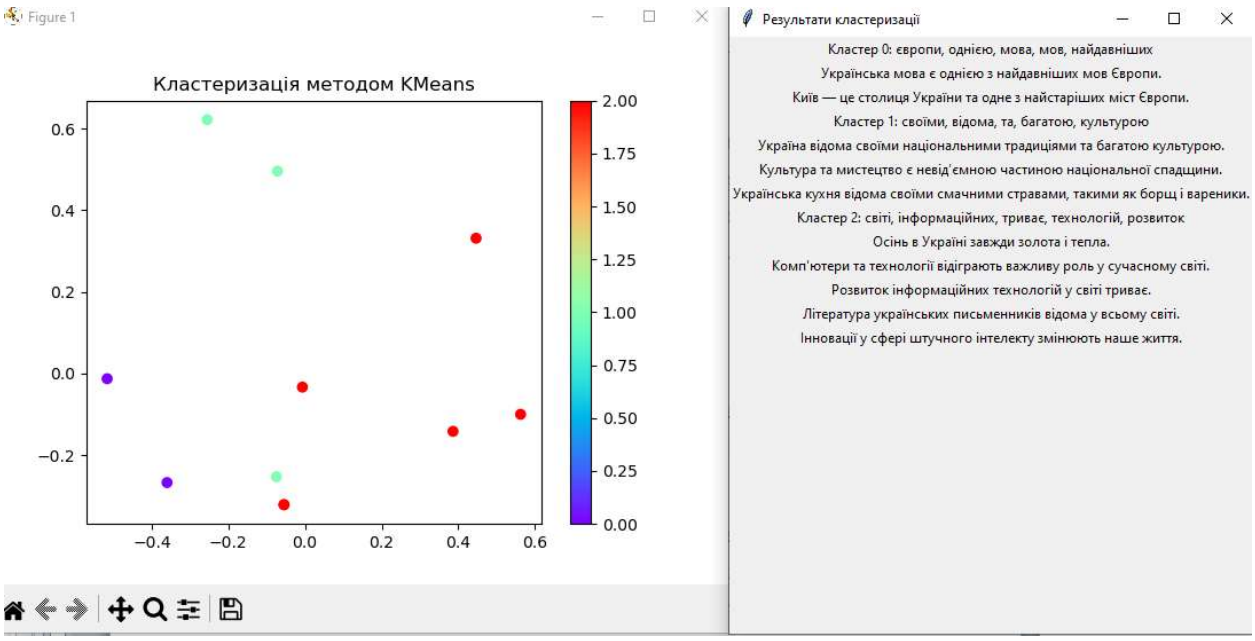


Рисунок 3.11 – Результат кластеризації тексту методом K-means

Результат методу DBSCAN наведено на рис.3.12. В результаті отримали два кластери.

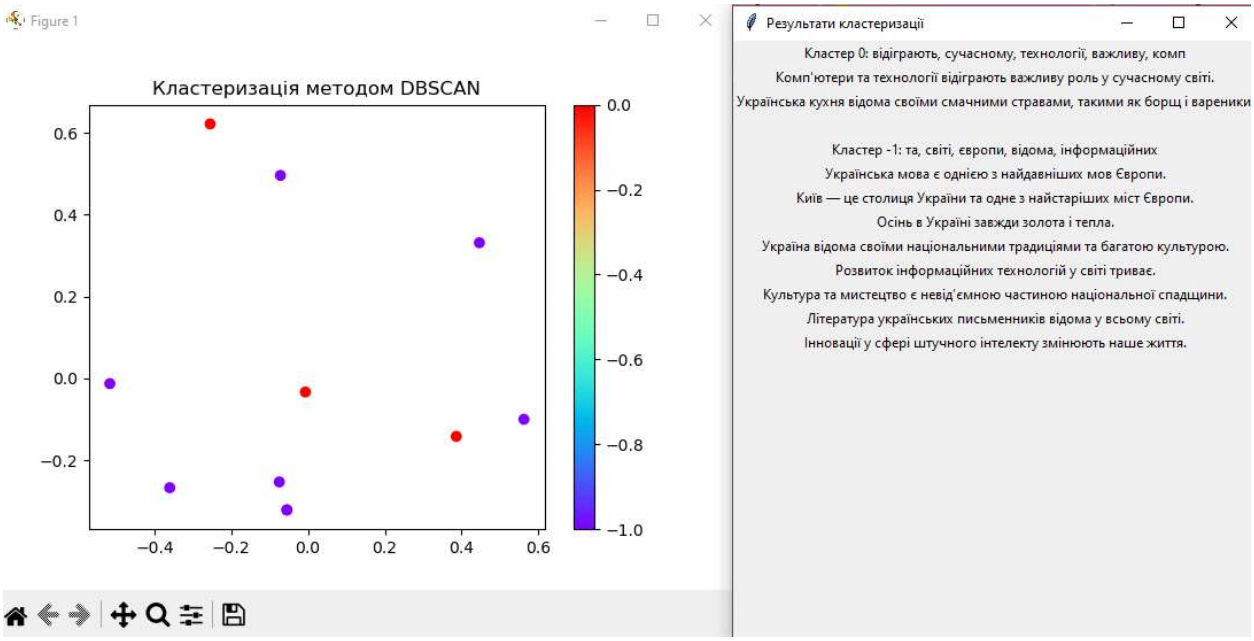


Рисунок 3.12 – Результат кластеризації тексту методом DBSCAN

Кластер 0: Теми: відіграють, сучасному, технології, важливу, комп'ютери. Цей кластер зосереджений на технологіях і їхньому впливі на сучасний світ, а також на українській кухні. Він поєднує в собі інформацію про важливість технологій і культурні аспекти, але може бути поліпшений, якщо зосередитися на більш однорідній тематиці. Можливо, слід розглянути можливість розподілу текстів на два окремі кластери: один для технологій, інший для кулінарії. Кластер -1: Теми: та, світі, європи, відома, інформаційних. Цей кластер містить широкий спектр текстів, які охоплюють теми української мови, культури, традицій та розвитку технологій. Це вказує на те, що кластер має змішану природу, і деякі тексти, можливо, не зовсім підходять за змістом до інших. Рекомендується переглянути тексти та визначити підкластери, щоб краще відобразити відмінності між культурною і технологічною інформацією.

Результат методу LDA наведено на рис.3.13. В результаті отримали три кластери. Кластер 0: Теми: світі, та, інформаційних, триває, технологій. Цей кластер зосереджений на темах, пов'язаних із технологіями, їхнім розвитком і впливом на сучасний світ. Тексти відображають інформаційні та культурні аспекти технологій, вказуючи на важливість IT-сфери в

Україні. Це дозволяє зробити висновок, що кластер є однорідним, але може бути корисно зосередитися на відмінностях між технологіями та їх впливом на суспільство. Кластер 1: Теми: завжди, осінь, тепла, україні, золота. У цьому кластері зосереджені тексти, що описують сезонні зміни в Україні та культурні аспекти, пов'язані з українською кухнею. Це вказує на те, що кластер має певний фокус на культурні традиції та природні явища, що є важливими для української ідентичності. Слід зазначити, що кластер є однорідним і містить тексти, що добре взаємодоповнюють одне одного. Кластер 2: Теми: відома, та, письменників, всьому, українських. Цей кластер фокусується на українській культурі, традиціях та літературі. Він відображає важливість культурної спадщини та її вплив на українську ідентичність. Тексти демонструють багатство української мови та культури, що робить кластер також однорідним.

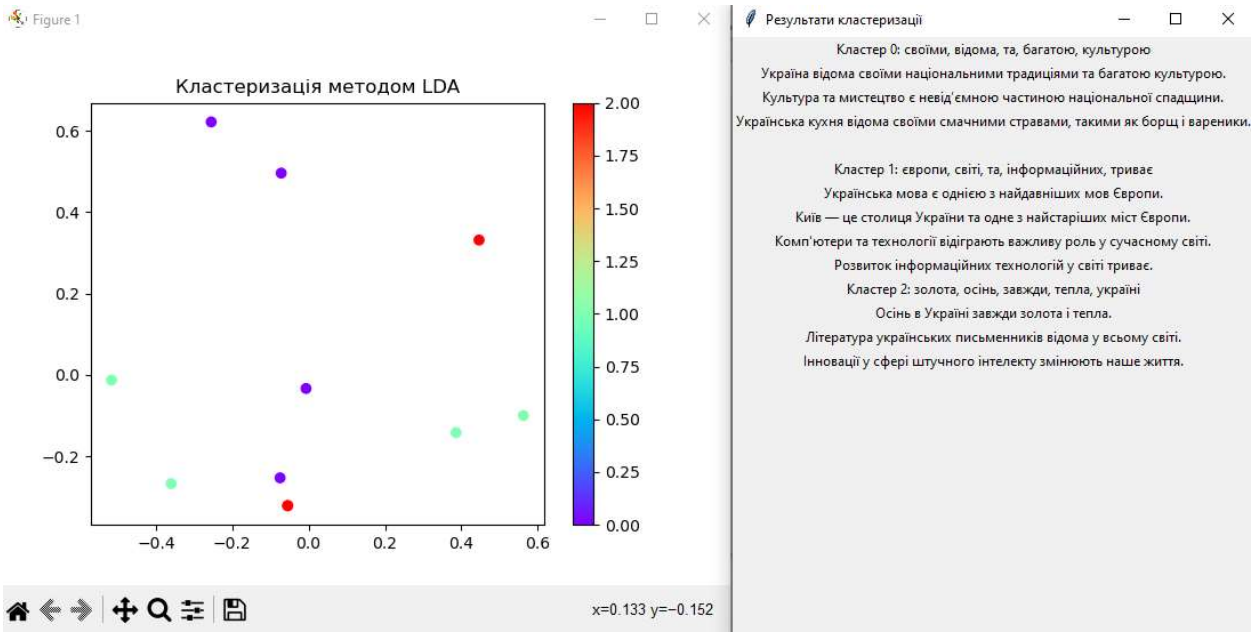


Рисунок 3.13 – Результат кластеризації тексту методом LDA

Результат методу Single Pass наведено на рис.3.14. В результаті отримали три кластери. Кластер 0: Теми: найдавніших, мова, однією, мов, європи. Цей кластер, в основному, фокусується на українській мові, її

історичному значенні та статусі в Європі. Проте наявність другого речення, яке стосується технологій, вказує на розширення тематики, що може призвести до зменшення однорідності кластера. Для покращення результатів доцільно було б розділити цей кластер на окремі групи для більш чіткого відображення різних аспектів — мовного та технологічного. Кластер 1: Теми: культура, національної, мистецтво, спадщини, частиною. Цей кластер зосереджується на культурній спадщині України, підкреслюючи важливість мистецтва у формуванні національної ідентичності. Кластер є однорідним, містить одне речення, яке чітко відображає цю тему. Це показує, що метод Single Pass може забезпечити точні результати, коли дані добре структуровані. Кластер 2: Теми: відома, світі, своїми, та, інформаційних. Цей кластер охоплює різноманітні аспекти, пов’язані з українською культурою, традиціями, природою та технологіями. Теми варіюються від культурних до технологічних, що може свідчити про широту охоплення, але також вказує на можливу неоднорідність. Кластер містить різні тексти, які підкреслюють як історичні, так і сучасні елементи української ідентичності.

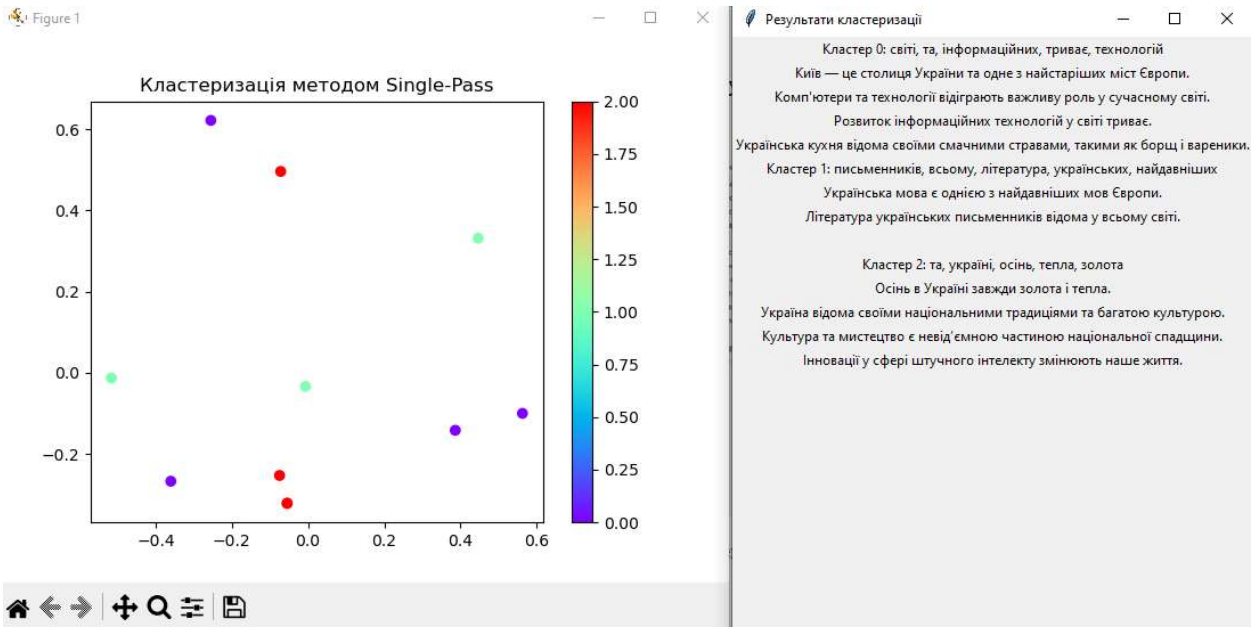


Рисунок 3.14 – Результат кластеризації тексту методом Single Pass

Отримані результати показують, що методи без навчання можуть бути ефективними для кластеризації українських текстових даних. Кожен метод має свої сильні сторони: DBSCAN добре виявляє неявні структури, K-means забезпечує чітке розмежування, Single Pass дозволяє обробляти великі набори даних без значних витрат ресурсів, а LDA дозволяє глибше зрозуміти теми в текстах.

3.5.2 Кластеризація текстових даних після попередньої обробки

Тепер проведемо дослідження методів без навчання: K-means, DBSCAN, LDA та Single Pass для кластеризації українських текстових даних з поперередньою обробкою і створення назв кластерів. Запропоновано дванадцять тем: "Наука", 1: "Політика", 2: "Економіка", 3: "Технології", 4: "Кухня", 5: "Медицина", 6: "Освіта", 7: "Спорт", 8: "Культура", 9: "Погода", 10: "Історія", 11: "Глобальні тренди". Завантажуємо файл стоп словами українською мовою.

Запускаємо програму , завантажуємо файл або можна ввести текст у текстовому полі як показано на рис.3.15.

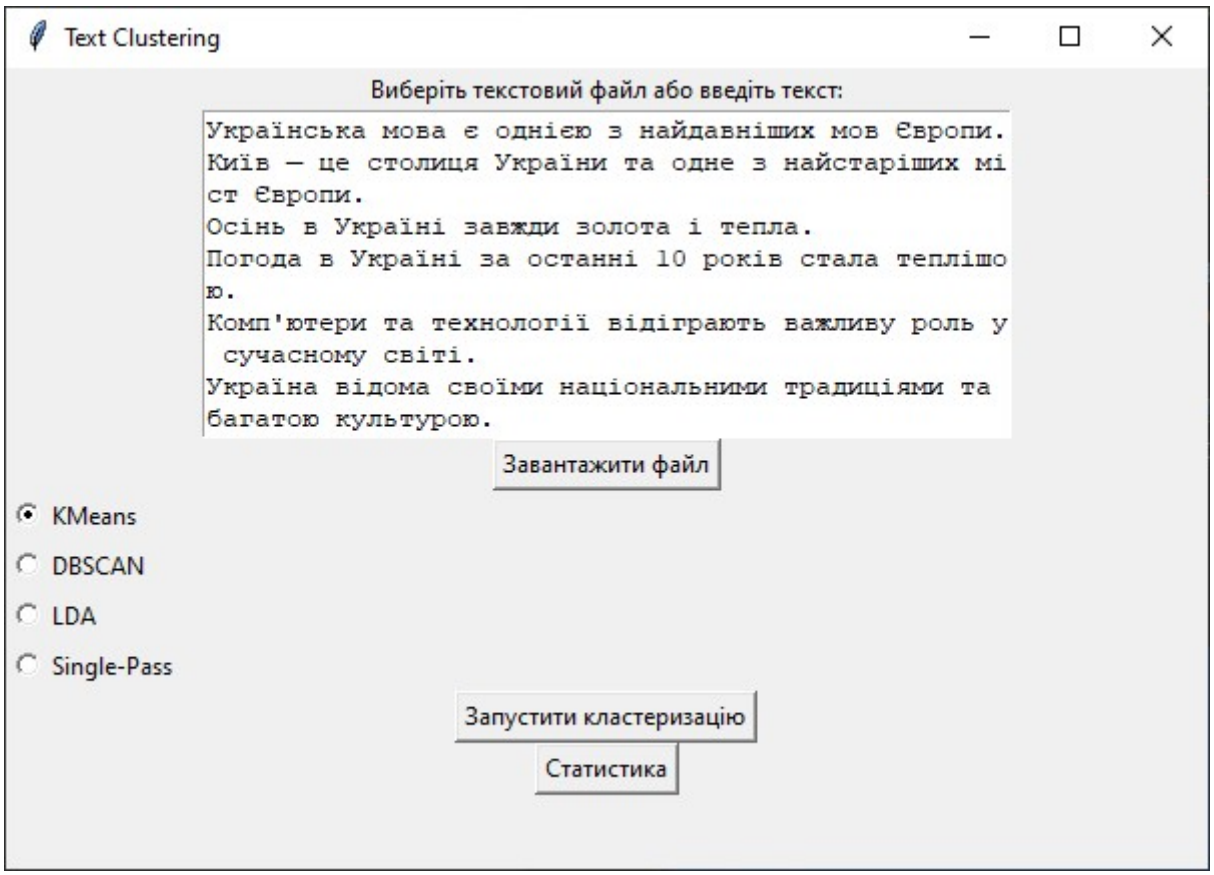


Рисунок 3.15 – Програмний інструмент кластеризації тексту

Обираємо метод K-means і натискаємо на кнопку «Запустити кластеризацію» і отримуємо результат на рис.3.16. За змочуванням метод поділяє на п’ять кластерів текст.

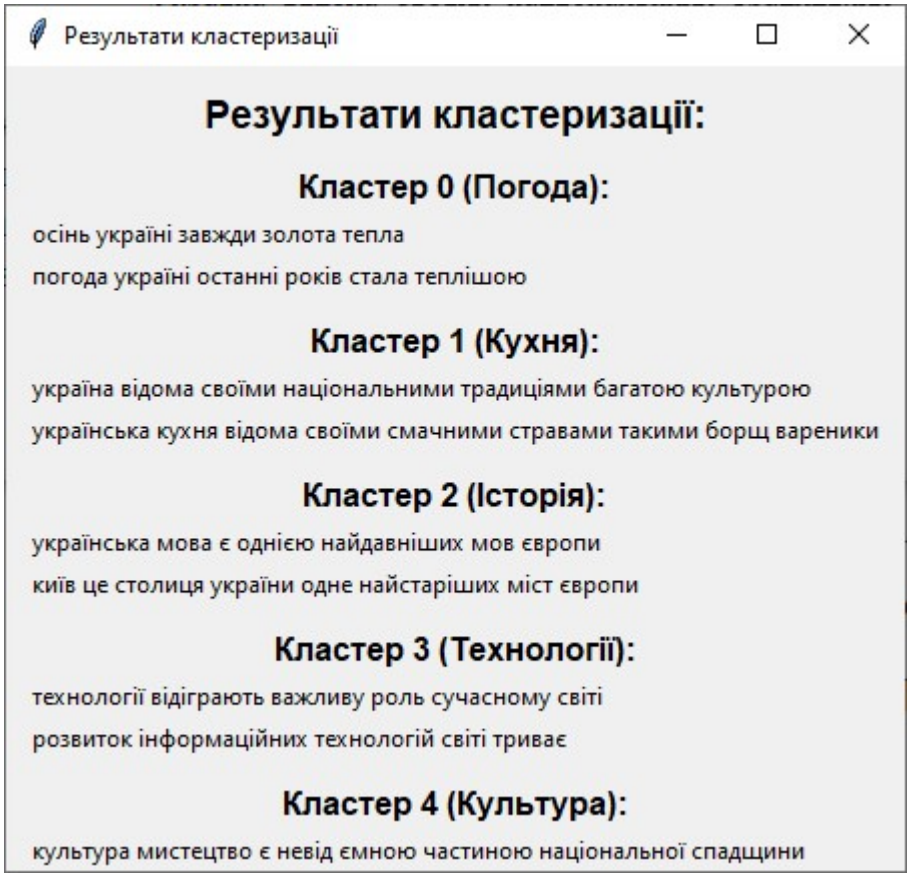


Рисунок 3.16 – Результат кластеризації тексту методом K-means з попередньою обробкою

Результати кластеризації методом KMeans демонструють цікаві тематичні групи серед українських текстових даних. Загальна кількість кластерів становить п'ять.

Кластер 0 фокусується на темі погоди. У цьому кластері містяться документи, що розглядають сезонні зміни в Україні, зокрема, такі як "осінь україні завжди золота тепла" та "погода україні останні років стала теплішою". Цей кластер свідчить про зацікавленість у кліматичних умовах та їх впливі на повсякденне життя. Кластер 1 зосереджений на темах кухні та культури. Документи, які належать до цього кластеру, підкреслюють важливість національних традицій, таких як "україна відома своїми національними традиціями багатою культурою" та "українська кухня відома своїми смачними стравами такими борщ вареники". Цей кластер акцентує

увагу на гастрономії та культурній ідентичності України. Кластер 2 присвячений історії. У ньому розміщені документи, що підкреслюють історичну цінність української мови та її місце в Європі. Цей кластер може бути важливим для дослідження історії України та її культурної спадщини. Кластер 3 зосереджується на технологіях. Документи цього кластеру, такі як "технології відіграють важливу роль сучасному світі" та "розвиток інформаційних технологій світі триває", акцентують увагу на технологічному розвитку та його впливі на суспільство. Тема важливості технологій в сучасному світі є актуальною та цікавою для широкого кола дослідників. Кластер 4 акцентує увагу на культурі. Єдиний документ у цьому кластері говорить про важливість культури та мистецтва в національній спадщині України: "культура мистецтво є невід ємною частиною національної спадщини". Це вказує на те, що культура займає важливе місце у свідомості суспільства.

На основі проведеного аналізу можна зробити кілька висновків. По-перше, результати кластеризації показують, що документи можна поділити на кілька тематичних груп, що охоплюють важливі аспекти українського суспільства, такі як погода, культура, історія та технології. По-друге, відзначається, що деякі кластери мають більше документів (кластер 0 та 1), тоді як інші містять лише один документ (кластер 4). Це може вказувати на необхідність додаткового збору даних для більш чіткої кластеризації.

Результат методу DBSCAN наведено на рис.3.17. Результати кластеризації методом DBSCAN вказують на деякі особливості в структурі текстових даних українською мовою. Загальна кількість кластерів, згідно з отриманими мітками, становить три, а також є група шуму, позначена як -1. Кластер -1 представляє собою кластер шуму. Цей кластер вказує на те, що ці документи не знайшли достатньої кількості сусідів у порівнянні з іншими, щоб бути включеними в основні кластери. Однак, варто зазначити, що в

цьому кластері присутня інформація, що стосується історії та культури України, яка може бути важливою для подальшого аналізу.

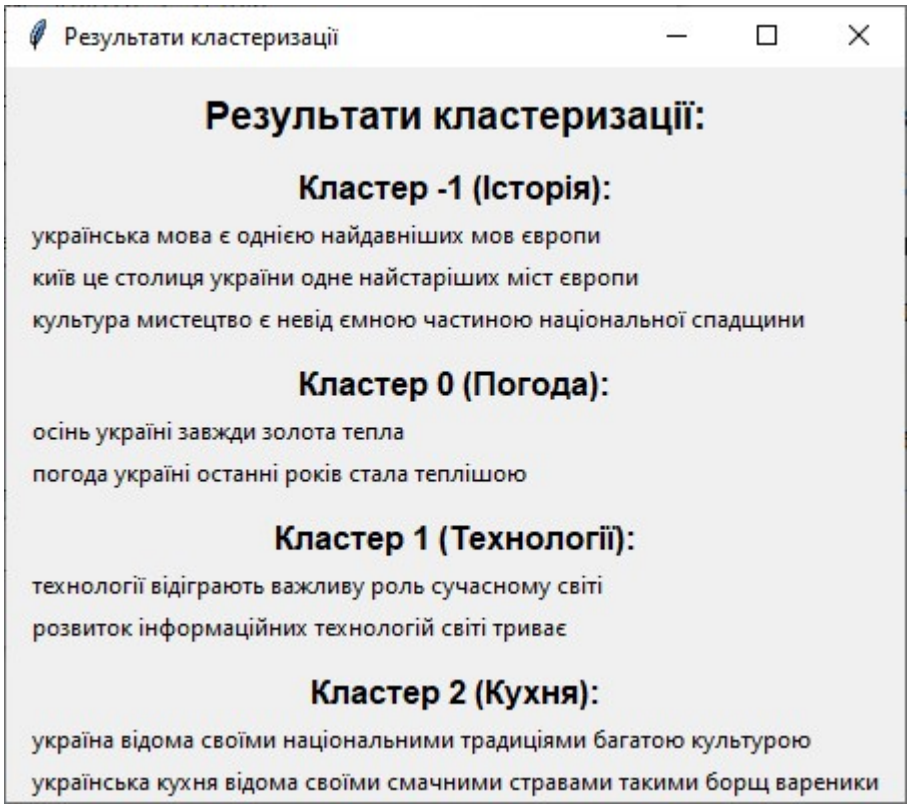


Рисунок 3.17 – Результат кластеризації тексту методом DBSCAN з попередньою обробкою

Кластер 0 фокусується на темах погоди. Він містить два документи: "осінь україні завжди золота тепла" та "погода україні останні років стала теплішою". Це свідчить про те, що в українських текстах спостерігається певна зацікавленість до кліматичних умов та сезонних змін. Кластер 1 присвячений технологіям. Документи, що входять до цього кластеру: "технології відіграють важливу роль сучасному світі" та "розвиток інформаційних технологій світі триває". Це вказує на актуальність теми технологічного розвитку в сучасному українському суспільстві. Кластер 2 зосереджується на культурі та гастрономії. У ньому містяться такі документи, як "україна відома своїми національними традиціями багатою культурою" та "українська кухня відома своїми смачними стравами такими

борщ вареники". Цей кластер підкреслює важливість культурних аспектів і національної ідентичності в українському суспільстві.

Результати кластеризації методом LDA наведено на рис.3.18 і свідчать про певні особливості у структурі текстових даних. Загальна кількість документів становить 9, з яких 4 класифіковані в чотири кластери. Кластер 0 присвячений історії і демонструє переважання тем, пов'язаних із культурою, гастрономією та історією України. Він відзначається високою концентрацією інформації про національні традиції та культурну спадщину, що може свідчити про важливість цих аспектів для українського суспільства.

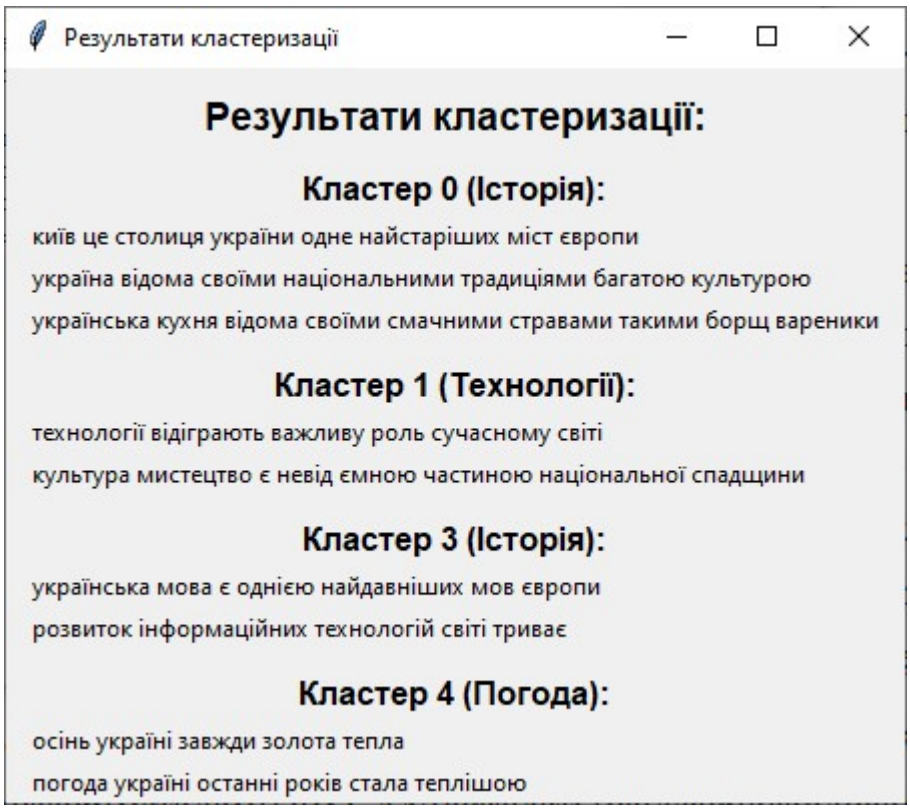


Рисунок 3.18 – Результат кластеризації тексту методом LDA з попередньою обробкою

Кластер 1 вказує на теми, пов'язані з технологічним прогресом і культурою. Поєднання цих двох тем підкреслює важливість розвитку

технологій у контексті збереження культурної ідентичності. Кластер 3 акцентує увагу на значенні української мови та розвитку інформаційних технологій. Він показує взаємозв'язок між мовною культурою та сучасними технологічними змінами, що може вказувати на намагання зберегти мовну ідентичність у швидко змінюваному світі. Кластер 4 охоплює теми, пов'язані з погодою та їх впливу на життя українців, що підкреслює важливість екологічних аспектів у текстах.

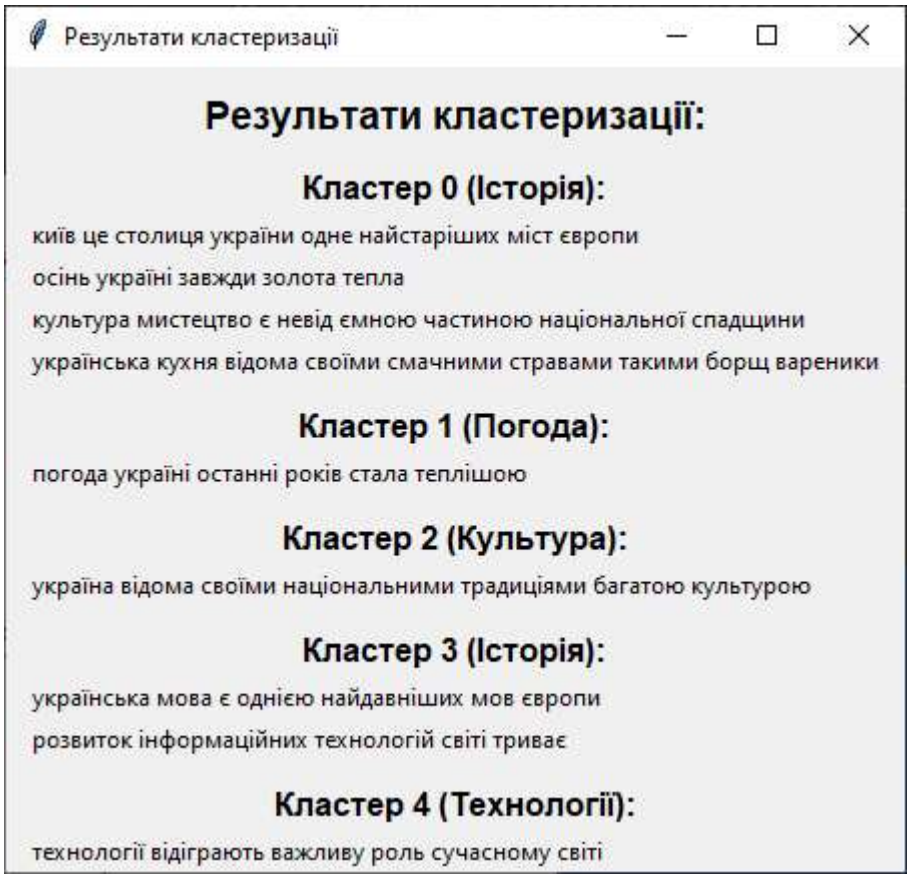


Рисунок 3.19 – Результат кластеризації тексту методом Single Pass з попередньою обробкою

Результати кластеризації методом Single-Pass наведено на рис.3.19. Кластеризація призвела до формування чотирьох основних кластерів. У Кластері 1 зібрані документи, які акцентують увагу на національних

традиціях і культурній спадщині України, зокрема фрази "україна відома своїми національними традиціями багатою культурою" та "культура мистецтво є невід ємною частиною національної спадщини". Кластер 2 зосереджений на українській кухні, зокрема на смачних стравах, таких як борщ і вареники. У Кластері 3 представлені документи, що торкаються мовної тематики, технологій та їх розвитку, зокрема "українська мова є однією найдавніших мов європи", а також "технології відіграють важливу роль сучасному світі". Кластер 4 містить інформацію про Київ як столицю України та зміни в погоді за останні роки. Класифікація за темами також показує, що Кластер 0 має акценти на історії (5 документів), кухні (4 документи), культури (1 документ) та погоди (1 документ). Кластер 1 зосереджується на погоді, Кластер 2 – на культурі, а Кластер 3 відображає технологічні та історичні аспекти.

Загалом, результати кластеризації методом Single-Pass свідчать про певні недоліки в класифікації, зокрема на невелику кількість відокремлених кластерів та слабку роздільну здатність. Це вказує на необхідність розгляду альтернативних методів кластеризації.

3.5.3 Метрики

В таблиці 3.4 наведено результати метрик для методів кластеризації без вчителя (див. рис. Б2 – Б5). Силуетний коефіцієнт показує, наскільки добре кожна точка належить до свого кластера, порівнюючи відстань між точкою та іншими точками в своєму кластері з відстанню до найближчого сусіднього кластера. Коефіцієнт варіюється від -1 до +1. Позитивне значення означає, що точка ближча до свого кластеру, ніж до іншого, що є хорошою ознакою. Негативні значення вказують на те, що точка може належати до іншого кластера.

Індекс Девіса-Боулдінга (Davies-Bouldin Index) визначає відношення внутрішньої розсіюваності (компактності) кожного кластера до відстаней між кластерами (роздільність). Чим менше значення, тим кращою є кластеризація. Ідеальне значення індексу — 0, що свідчить про ідеальне відокремлення кластерів. Вищі значення свідчать про більше перекриття між кластерами [11].

Inertia (Інерція) – це метрика, специфічна для методу KMeans. Інерція вимірює суму квадратів відстаней між кожною точкою та її центроїдом (центральною точкою кластеру). Чим менше значення інерції, тим краще кластери відповідають своїм центроїдам [11].

Таблиця 3.4 – Результати метрик

Метрики	K-means	DBSCAN	LDA	Single Pass
Силуетний коефіцієнт	0.06	0.06	-0.02	-0.01
Індекс Девіса-Боулдінга	1.10.	1.51	1.63	1.57
Inertia	1.42	-	-	-
Середня відстань до центроїда		1.39		0.31
Логарифмічна ймовірність	-	-	-144.30	-

Середня відстань до центроїда оцінює, наскільки далеко об'єкти в кластері знаходяться від центральної точки (центроїда). Вона допомагає визначити, наскільки щільно згруповані об'єкти у кластерах. Менші значення означають, що кластери є більш компактними, що зазвичай вважається добрим результатом кластеризації.

Логарифмічна ймовірність використовується для темної кластеризації (LDA) і оцінює, наскільки добре модель LDA пояснює дані. Логарифмічна ймовірність вимірює ймовірність даних відповідати моделі кластерів. Менш негативні значення вказують на краще пояснення даних моделлю. Більш негативні значення означають, що модель LDA погано відповідає даним, що може вказувати на слабке тематичне розбиття.

Аналізуючи значення з табл. 3.4 можна зробити висновки: Single-Pass показує найгірші результати за силуетним коефіцієнтом, що вказує на слабку структурованість кластерів, хоча один з показників індексу Девіса-Боулдінга (0.87) показує кращу роздільну здатність у порівнянні з іншими методами. KMeans демонструє трохи кращі результати з точки зору індексу Девіса-Боулдінга (1.10) та силуетного коефіцієнта (0.06), що вказує на більш збалансовану кластеризацію. DBSCAN показує дещо слабкі результати за індексом Девіса-Боулдінга (1.51) та силуетним коефіцієнтом (0.06), але забезпечує краще групування об'єктів з високою відстанню до сусідів. LDA має найгірший індекс Девіса-Боулдінга (1.63) та слабкий силуетний коефіцієнт (-0.02), що вказує на найслабші результати кластеризації серед представлених методів.

Загалом, найкращими результатами в цьому аналізі є метод KMeans, який показує відносно кращу структурованість кластерів, але є можливість вдосконалення за рахунок налаштування параметрів або використання інших алгоритмів.

На рис.3.20 – рис.3.24 наведено графіки кластеризації для методів KMeans, DBSCAN, LDA та Single-Pass. Перший компонент на осі X та другий компонент на осі Y показують розподіл кластерів у двовимірному просторі після зменшення розмірності.

На рис.3.20 було виділено 5 кластерів, позначених різними кольорами в легенді: "Кластер 0", "Кластер 1" тощо. Кожен кластер має відповідний

центроїд, позначений великими червоними хрестами. Відокремленість кластерів є помітною, особливо для кластера 0, який значно віддалений від інших у нижньому лівому куті графіка.

Кластери 1, 2, 3 і 4 розташовані вздовж осі, яка збігається з горизонтальною віссю (перша компонента). Кластер 0 чітко відокремлений як по першій, так і по другій компоненті, що свідчить про більшу відстань між ним та іншими кластерами.

З графіку оис.3.20 видно, що компоненти для кластерів 1, 2, 3 і 4 мають незначну варіацію по другій компоненті, тобто ці кластери розташовані близько один до одного на вертикальній осі. Це може свідчити про те, що точки в цих кластерах є схожими або не дуже добре розділеними у просторі ознак.

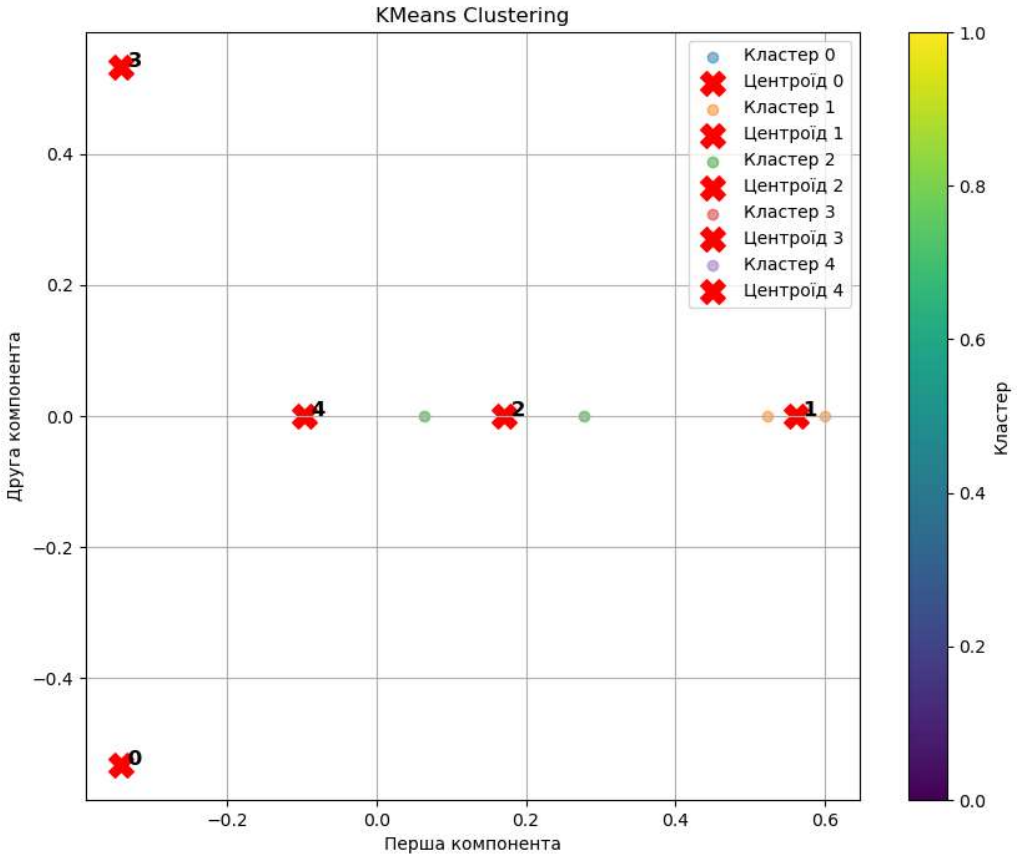


Рисунок 3.20 – Графік кластеризації методом K-Means

На рис.3.21 графік демонструє візуалізацію кластеризації на основі DBSCAN. У результатах спостерігаються три ідентифіковані кластери, при цьому кластер -1 позначає "шум" або точки, які не належать до жодного кластера. Центроїди (червоні хрести) позначені для кожного з кластерів. Існують три значущі кластери:

– Кластер 0 (червоні точки) — розташований ліворуч унизу;

– Кластер 1 (оранжеві точки) — розташований у центрі;

– Кластер 2 (червоні точки) — розташований праворуч. Усі інші точки (сині) віднесені до кластеру -1, що означає, що вони визнані шумом і не входять до жодного з виявлених кластерів.

Коефіцієнт силуета для цієї кластеризації дорівнює 0.06, що вказує на низький рівень відокремленості кластерів і можливі проблеми з якістю кластеризації. Значення коефіцієнта Девіса-Боулдінга становить 1.51, що свідчить про те, що деякі кластери мають перекриття або недостатньо добре відокремлені.

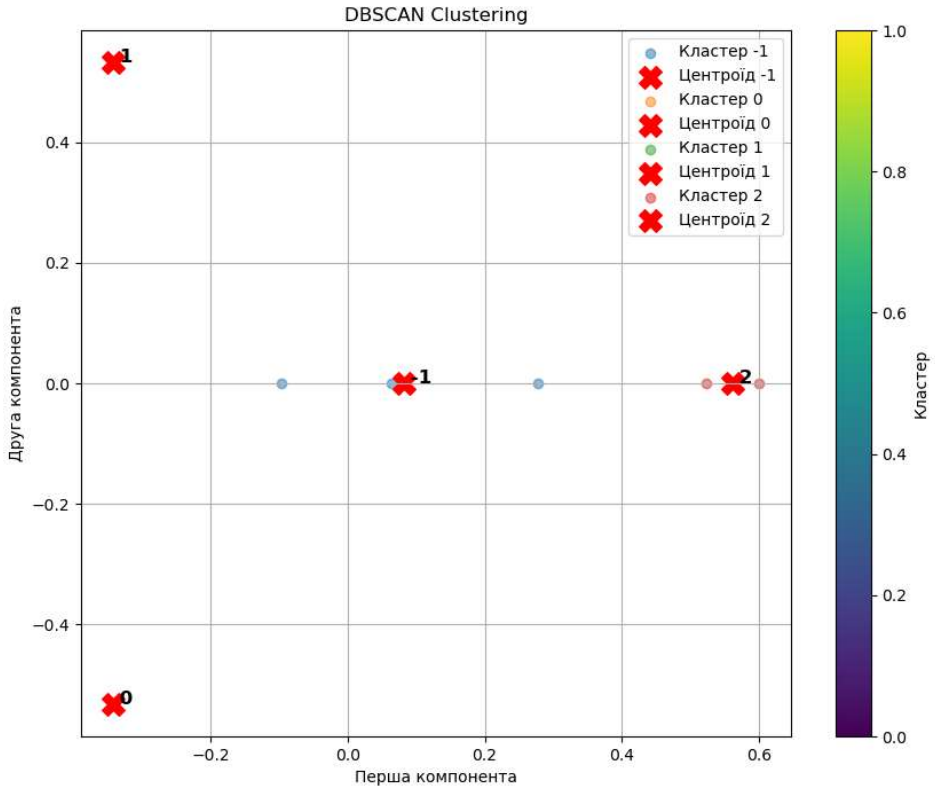


Рисунок 3.21 – Графік кластеризації методом DBSCAN

На рис.3.22 графік ілюструє вибір значення параметра eps (мінімальна відстань для формування кластерів). Є кілька помітних зламів на кривій, особливо в діапазоні між eps = 1 та eps = 6, що може вказувати на оптимальну величину eps, за якої спостерігається різка зміна в результатах кластеризації. Пропоноване оптимальне значення eps на основі графіку — приблизно 1.5, оскільки після цього значення на графіку спостерігається стрибок у відстанях, що може свідчити про наявність добре відокремлених кластерів при такому значенні.

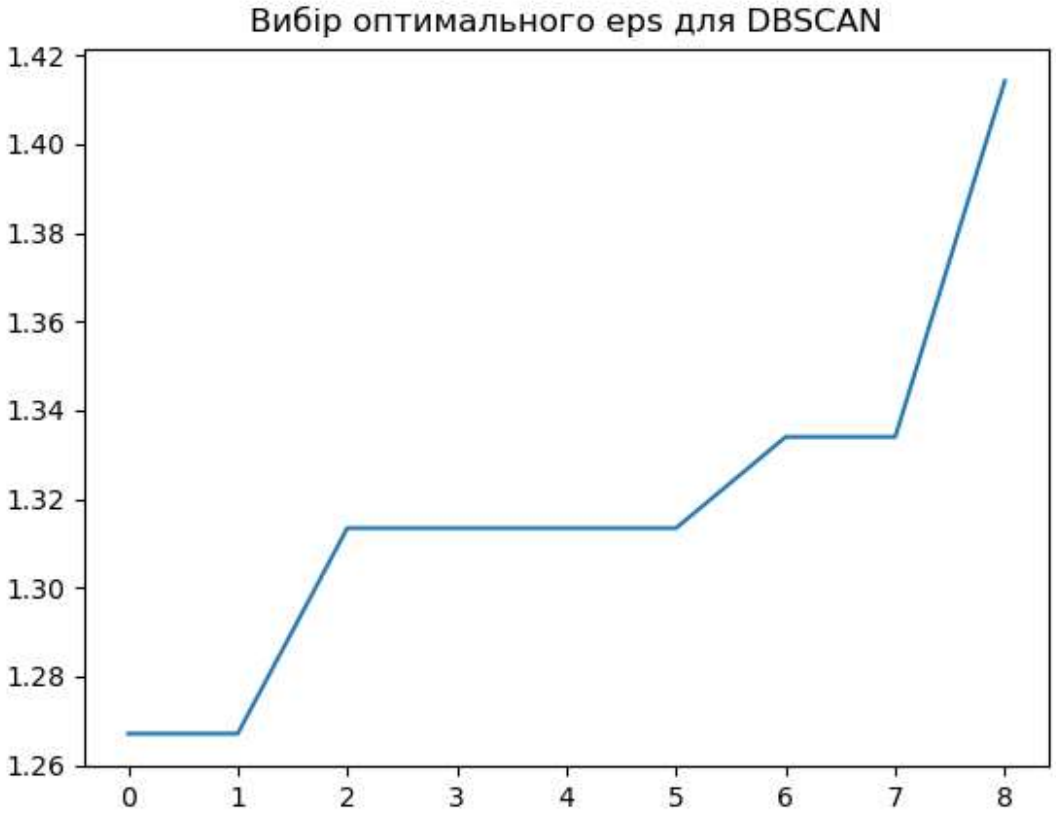


Рисунок 3.22 – Графік вибір значення параметра eps для DBSCAN

Графік на рис.3.23 для LDA Clustering виглядає більш структурованим, точки чітко розташовані навколо центроїдів, що свідчить про стабільніший розподіл даних по кластерам.

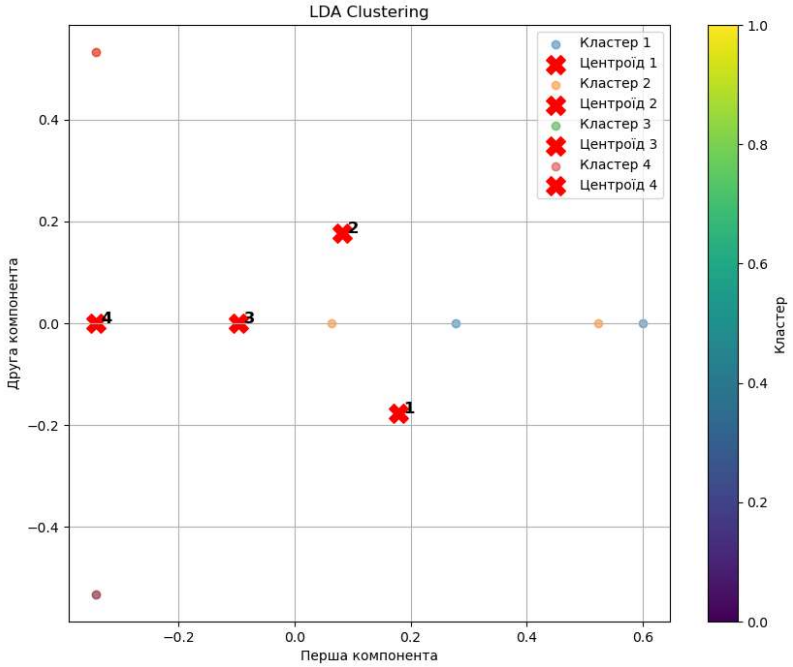


Рисунок 3.23 – Графік кластеризації методом LDA Clustering

Single-Pass Clustering (рис.3.24) має менш чіткі межі кластерів і деякі кластери здаються розмитими, що може пояснювати негативні значення силуетного коефіцієнта (-0.02 для першого запуску і -0.01 для другого) та високий індекс Девіса-Боулдінга (1.57 і 0.87).

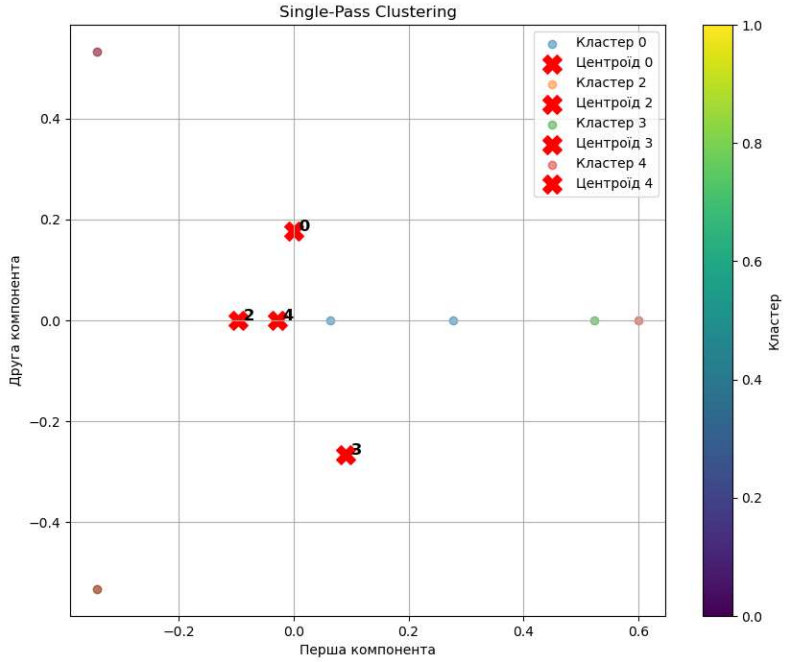


Рисунок 3.24 – Графік кластеризації методом Single-Pass Clustering

3.6 Висновки до третього розділу

Проведено комплексну розробку програмного інструменту для дослідження методів машинного навчання без вчителя, що застосовуються до кластеризації текстових даних українською мовою.

Розглянуто особливості інтерпретації української мови в контексті машинного навчання та визначено основні етапи попередньої обробки текстових даних. Було розроблено та реалізовано процес токенізації, нормалізації текстів, видалення стоп-слів та лемматизації, що дозволило підготувати дані для подальшої кластеризації. Було досліджено різні методи кластеризації без вчителя, такі як K-Means, DBSCAN, Latent Dirichlet Allocation (LDA) та Single-Pass. Кожен із методів має свої переваги та недоліки, які були ретельно проаналізовані під час експериментальних досліджень. Зокрема, методи K-Means та LDA продемонстрували кращі результати для тематичної кластеризації текстів.

Описано архітектуру розробленого інструменту для кластеризації тексту. Інструмент має модульну структуру, що дозволяє легко інтегрувати нові методи кластеризації, а також забезпечує можливість масштабування та модифікації. Здійснено інтеграцію бази даних для зберігання текстових даних, результатів кластеризації та іншої інформації, яка є важливою для дослідження. Це дозволяє зберігати дані для подальшого аналізу, а також відстежувати результати кластеризації за різними методами.

Розроблено зручний графічний інтерфейс, що дозволяє користувачеві виконувати кластеризацію текстів за допомогою кількох методів та отримувати візуалізовані результати. Інтерфейс також передбачає можливість інтеграції бази даних, що спрощує роботу з текстовими даними та аналіз результатів.

.

ВИСНОВКИ

У даній кваліфікаційній роботі на тему «Дослідження методів машинного навчання без вчителя для кластеризації текстових даних українською мовою» було проведено комплексне дослідження сучасних методів кластеризації тексту та розроблено програмний інструмент для їх практичного застосування.

У першому розділі виконано детальний аналіз стану вирішення проблеми кластеризації текстових даних, де було визначено основні завдання та етапи кластерного аналізу. Окремо розглянуто актуальність кластеризації для обробки українських текстових даних, а також сформульовано конкретні завдання для реалізації кваліфікаційної роботи.

Другий розділ присвячено огляду основних методів кластеризації текстових даних. Було досліджено ключові підходи до підготовки текстових даних, включно з виділенням ознак за допомогою TF-IDF, Word2Vec і Doc2Vec. Окремо проаналізовано популярні методи кластеризації, такі як K-середні, DBSCAN та агломеративна кластеризація. Важливу роль у цьому розділі відіграє аналіз методів валідації результатів кластеризації, що дозволило визначити їх ефективність.

У третьому розділі розроблено інструмент для дослідження методів кластеризації текстових даних українською мовою. Було реалізовано алгоритми кластеризації без вчителя та забезпечено інтеграцію бази даних для зберігання результатів. Графічний інтерфейс інструменту дозволяє користувачеві проводити кластеризацію за допомогою різних методів, отримуючи зрозумілі та наочні результати. Під час експериментальних досліджень використано кілька метрик, таких як силуетний коефіцієнт, однорідність та повнота, серед методів кластеризації найбільш ефективними виявились K-Means та LDA для кластеризації текстових даних за темами.

ПЕРЕЛІК ПОСИЛАНЬ

1. S. Akter, A. S. Asa, M. P. Uddin, M. D. Hossain, S. K. Roy, and M. I. Afjal. An extractive text summarization technique for bengali document(s) using k-means clustering algorithm. In 2017 IEEE International Conference on Imaging, Vision Pattern Recognition (icIVPR), pages 1–6, Feb 2017.

2. Sigit Adinugroho, Yuita Arum Sari, M. Ali Fauzi, and Putra Pandu Adikara. Optimizing K-means text document clustering using latent semantic indexing and pillar algorithm. 2017 5th International Symposium on Computational and Business Intelligence (ISCBI), pages 81–85, 2017..

3. Harsha S. Gowda, Mahamad Suhil, D. S. Guru, and Lavanya Narayana Raju. Semi-supervised text categorization using recursive k-means clustering. CoRR, abs/1706.07913, 2017.

4. Jasmine Irani, Nitin Pise, and Madhura Phatak. Clustering Techniques and the Similarity Measures used in Clustering: A Survey. International Journal of Computer Applications, 134(7):975–8887, 2016.

5. Hui Li and Qing Li. Forum topic detection based on hierarchical clustering. 2016 International Conference on Audio, Language and Image Processing (ICALIP), pages 529–533, 2016.

6. Andrea Morichetta, Enrico Bocchi, Hassan Metwalley, and Marco Mellia. CLUE: Clustering for mining web URLs. Proceedings of the 28th International Teletraffic Congress, ITC 2016, 1:286–294, 2017.

7. Raihannur Reztaputra and Masayu Leylia Khodra. Sentence structure-based summarization for Indonesian news articles. Proceedings - 2017 International Conference on Advanced Informatics: Concepts, Theory and Applications, ICAICTA 2017, pages 0–5, 2017.

8. Ma. Shiela C. Sapul, Than Htike Aung, and Rachsuda Jiamthapthaksin. Trending topic discovery of Twitter Tweets using clustering and topic modeling algorithms. In 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), pages 1–6. IEEE, jul 2017.

9. Krithi Shetty and Jagadish S. Kallimani. Automatic extractive text summarization using K-means clustering. 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), pages 1–9, 2017.

10. Aakanksha Sharaff, Hari Shrawgi, Priyank Arora, and Anshul Verma. Document Summarization by Agglomerative nested clustering approach. 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology, ICAECCT 2016, pages 187–191, 2017.

11. Lal, Chaman & Ahmed, Awais & Siyal, Reshma & Kumar, Suresh & Aftab, Shagufta & Jamali, Arshad. (2021). Text Clustering using K-MEAN. International Journal of Advanced Trends in Computer Science and Engineering.

12. Nasim, Zarmeen, and Sajjad Haider. (2020).Cluster analysis of urdu tweets. Journal of King Saud University-Computer and Information Sciences

13. Scikit-learn. 4.2. feature extraction. [Електронний ресурс] – Режим доступу до ресурсу: http://scikit-learn.org/stable/modules/feature_extraction.html, 2013. Accessed: 2018-05-15.

14. NumPy. [Електронний ресурс] – Режим доступу до ресурсу:https://numpy.org/

15. TensorFlow. [Електронний ресурс] – Режим доступу до ресурсу: https://www.tensorflow.org/

16. Matplotlib. [Електронний ресурс] – Режим доступу до ресурсу:https://matplotlib.org

17. Librosa. [Електронний ресурс] – Режим доступу до ресурсу:https://librosa.org/

Додаток А.
Програмний код (лістинг) компонентів застосунку

Фрагмент лістингу програми для реалізації методу K-means

```
# -*- coding: utf-8 -*-

import nltk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.cluster import KMeans

# Для роботи з українською мовою завантажуємо токенізатор та стоп-слова
nltk.download('punkt')



class KmeansClustering():
    def __init__(self, stopwords_path=None):
        self.stopwords = self.load_stopwords(stopwords_path)
        self.vectorizer = CountVectorizer()
        self.transformer = TfidfTransformer()

    def load_stopwords(self, stopwords=None):
        """
        Завантаження стоп-слів
        :param stopwords: шлях до файлу зі стоп-словами
        :return: список стоп-слів
        """
        if stopwords:
            with open(stopwords, 'r', encoding='utf-8') as f:
                return [line.strip() for line in f]
        else:
            return []

    def preprocess_data(self, corpus_path):
        """
        Попередня обробка тексту, кожен рядок — це один текст
        :param corpus_path: шлях до файлу з текстами
        :return: оброблений корпус
        """
        corpus = []
        with open(corpus_path, 'r', encoding='utf-8') as f:
```

```
        for line in f:
            tokens = nltk.word_tokenize(line.strip().lower())  # Токенізація тексту
            tokens = [word for word in tokens if word.isalpha() and word not in self.stopwords]
            corpus.append(' '.join(tokens))
    return corpus

def get_text_tfidf_matrix(self, corpus):
    """
    Отримати TF-IDF матрицю
    :param corpus: корпус текстів
    :return: TF-IDF матриця
    """
    tfidf = self.transformer.fit_transform(self.vectorizer.fit_transform(corpus))

    # Отримати ваги TF-IDF у вигляді матриці
    weights = tfidf.toarray()
    return weights

def kmeans(self, corpus_path, n_clusters=5):
    """
    Кластеризація текстів за допомогою KMeans
    :param corpus_path: шлях до корпусу (кожен рядок — окремий текст)
    :param n_clusters: кількість кластерів
    :return: {cluster_id1:[text_id1, text_id2]}
    """
    corpus = self.preprocess_data(corpus_path)
    weights = self.get_text_tfidf_matrix(corpus)

    clf = KMeans(n_clusters=n_clusters)

    y = clf.fit_predict(weights)

    # Результати кластеризації
    result = {}
    for text_idx, label_idx in enumerate(y):
        if label_idx not in result:
            result[label_idx] = [text_idx]
        else:
            result[label_idx].append(text_idx)

    # Виведення результатів кластеризації для користувача
    with open(corpus_path, 'r', encoding='utf-8') as f:
```

```
            texts = f.readlines()

        for cluster_id, text_indices in result.items():
            print(f"\nГрупа {cluster_id + 1}:")
            for idx in text_indices:
                print(f"- {texts[idx].strip()}")

        return result


if __name__ == '__main__':
    Kmeans = KmeansClustering(stopwords_path='../data/ukrainian_stopwords.txt')
    result = Kmeans.kmeans('../data/ukrainian_texts.txt', n_clusters=5)
    print(result)
```

Додаток Б.

Графічний матеріал

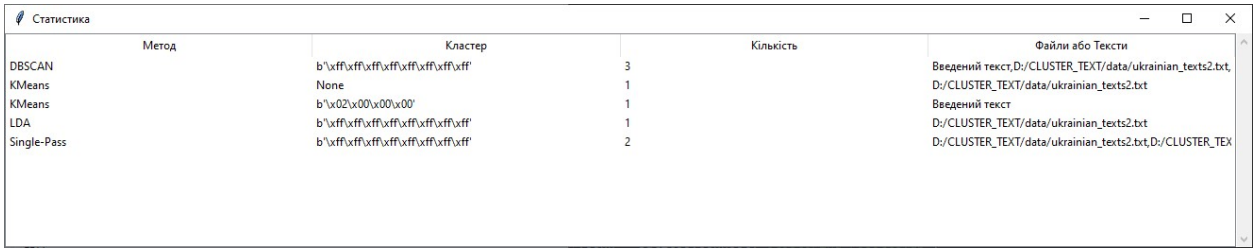


Рисунок Б.1 – Результат «Статистика»

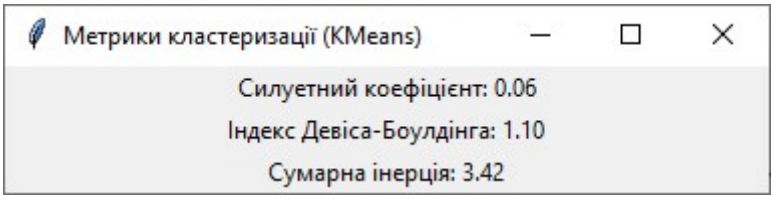


Рисунок Б.2 – Результат метрики кластеризації методом K-means

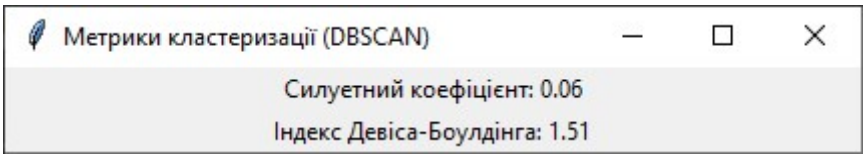


Рисунок Б.3 – Результат метрики кластеризації методом DBSCAN

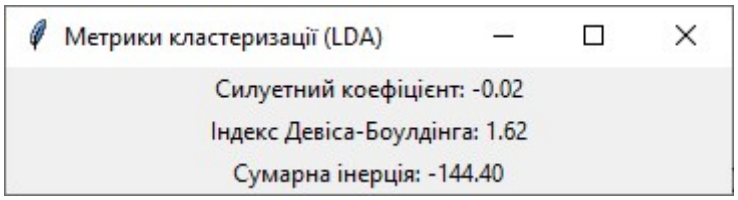


Рисунок Б.4 – Результат метрики кластеризації методом LDA

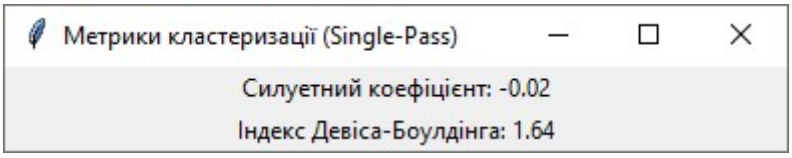


Рисунок Б.5 – Результат метрики кластеризації методом Single-Pass