НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Факультет інформатики та обчислювальної техніки
(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління
(назва кафедри)

"На правах рукопису"
УДК 519.68; 681.513.7;
612.8.001.57; 007.51/.52

«До захисту допущено»
Завідувач кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“ ____ ” _____ 20 *18* р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за спеціальністю *122 Комп'ютерні науки та інформаційні технології*
(код та назва спеціальності)

спеціалізацією *Інформаційні управляючі системи та технології*
(код та назва спеціалізації)

на тему: *Рекомендаційні системи щодо уподобань користувача соціальних мереж з врахуванням його профілю та психотипу*

Виконала: студентка *VI* курсу групи ІС-61м
(шифр групи)

Купцова Ірина Володимирівна _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник *к.ф.-м.н., доц. Гавриленко О.В.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант *к.т.н., доц. Жданова О.Г.* _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018

Магістерська дисертація на тему: « Рекомендаційні системи щодо уподобань користувача соціальних мереж з врахуванням його профілю та психотипу

Виконала: Купцова І. В., ІС-61м

Науковий керівник: Гавриленко О.В.

РЕФЕРАТ

Магістерська дисертація:149 с., 41 рис., 38 табл., 2 додатки, 101 джерело.

Актуальність теми. Кількість доступної користувачу інформації настільки велика, що важко виділити щось конкретне та необхідне шляхом звичайного перегляду. Тому системи, які допомагають аналізувати дані та орієнтують в них, представляють велику цінність.

Внутрішньоресурсні рекомендації є звичною функцією соціальних мереж, але вони використовують лише власний контент для обробки. Також мають місце рекомендаційні системи інакших структур, які не враховують соціальну складову користувача, а отже використовують вузький спектр інформації для формування рекомендацій. Подібні системи обмежені або в інформації про об’єкти рекомендацій, або в даних про користувача, що не дозволяє створити повноцінні та задовільні пропозиції.

У зв’язку з прагненням вирішити обидві проблеми однієї області, актуальною є розробка рекомендаційної системи на основі соціальних мереж, які допомагають у персоніфікації користувача та складанні його психотипу за допомогою його профілю.

Зв’язок роботи з науковими програмами, планами, темами

Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Інтелектуальний аналіз даних» (№УДК 519.68; 681.513.7; 612.8.001.57; 007.51/.52).

Мета роботи і задачі дослідження. Мета дисертаційної роботи – збільшення релевантності персоніфікованих рекомендацій. Для цього необхідно виконати такі задачі:

- охарактеризувати існуючі методи визначення рекомендацій та здійснити їх порівняльний аналіз;
- формалізувати задачу складання персональних рекомендацій;з
- реалізувати та проаналізувати обрані алгоритми надання рекомендацій;
- запропонувати метод підвищення релевантності рекомендацій;

- розробити програмну реалізацію розробленого методу;
- виконати аналіз отриманих результатів.

Об'єкт дослідження: процес надання персоніфікованих рекомендацій.

Предмет дослідження: методи аналізу персоніфікованих даних та надання рекомендацій на їх основі.

Методи дослідження, застосовані у даній роботі, базуються на методах машинного навчання та експертної оцінки.

Наукова новизна отриманих результатів. Розроблено підхід до розв'язання задачі кластеризації та класифікації наборів даних категоріального типу та надання рекомендацій шляхом удосконалення алгоритму кластеризації k-середніх, а також досліджено та вдосконалено метод попереднього аналізу вхідної вибірки.

Апробація результатів. Результати досліджень були апробовані на:

- 4-й міжнародній науково-практична конференція "Актуальні питання сучасної науки", м. Київ;
- науково-практичній конференції "Інформатика та обчислювальна техніка ІОТ-2018", м. Київ;
- VI конкурс стартапів Sikorsky Challenge, 11-12 жовтня 2017 року;
- наглядова рада Укроборонпрому, березень 2018 року.
-

Публікації. За матеріалами дисертації було опубліковано 4 наукові роботи:

- стаття в збірнику "Управління проектами, системний аналіз та логістика", Серія "Технічні науки" (ISSN: 2309-8635);
- тези доповіді на 8-й міжнародній науково-технічній конференції "Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління", м. Харків;
- тези доповіді на 4-й міжнародній науково-практичній конференції "Актуальні питання сучасної науки", м. Київ;

- тези доповіді на науково-практичній конференції “Інформатика та обчислювальна техніка ІОТ-2018”, м. Київ.

РЕКОМЕНДАЦІЙНІ СИСТЕМИ, КЛАСТЕРИЗАЦІЯ, КЛАСИФІКАЦІЯ ДАНИХ, АНАЛІЗ СОЦІАЛЬНИХ МЕРЕЖ, ПОБУДОВА ПСИХОТИПУ, МАШИННЕ НАВЧАННЯ

ABSTRACT

Master dissertation: 149 p., 41 fig., 38 tables., 2 appendixes, 101 sources.

Actuality. Volume of information available to user is so great that it is difficult to distinguish something specific and necessary through a regular review. Therefore, systems that help analyze and orientate data are of great value.

Recommendations based on internal resource are common feature of social networks, but it uses only its own content for processing. There are also recommended systems of other structures that do not take into account the social component of the user and, therefore, use a narrow range of information to formulate recommendations. Such systems are limited either in the information about the objects of the recommendations or in the user data, which does not allow to create complete and satisfactory offers.

In connection with the desire to solve both problems of one area, it is relevant to develop a system based on social networks that help in personalizing the user and compiling his psychotype by his profile.

Relationship of work with scientific programs, plans, themes

The work was carried out at the Department of Automated Systems for Information Processing and Management of the National Technical University of Ukraine "Igor Sikorsky Kyiv Politechnic Institute" within the framework of the theme "Intellectual Data Analysis".

Goal and tasks of research. The goal of the dissertation is to increase the relevance of personified recommendations. To do this it is needed to accomplish the following tasks:

- to characterize the existing methods for defining recommendations and perform their comparative analysis;
- to formalize the task of providing personal recommendations;
- to implement and analyze selected algorithms for providing recommendations;
- to propose a method of increasing recommendations relevance;
- to develop a software implementation of the developed method;
- to perform results analysis.

Object of research: process of providing personalized recommendations.

Subject of research: methods of personified data analysis and recommendations provision on their basis.

The research methods used in this paper are based on machine learning and expert assessment methods.

Scientific novelty of the obtained results. The approach to solving the problem of clustering and categorical type data sets classification and providing recommendations by improving the k-means clustering algorithm has been developed, and the method of preliminary analysis of the input sample has been researched and improved.

Test results. The results of the research were tested on:

- 4th international scientific and practical conference "Actual problems of modern science", Kyiv;
- scientific-practical conference "Informatics and Computing IOT-2018", Kyiv;
- VI Sikorsky Challenge Startup Competition, October 11-12, 2017;
- Supervisory Board of Ukroboronprom, March 2018.

Publications. On the materials of the dissertation was published 4 scientific works:

- - article in the collection "Project Management, System Analysis and Logistics", Series "Technical Sciences" (ISSN: 2309-8635);
- abstract of the report at the 8th International Scientific and Technical Conference "Modern Directions in the Development of Information and Communication Technologies and Control Tools", Kharkiv;
- Abstracts of the report at the 4th International Scientific and Practical Conference "Actual Issues of Modern Science", Kyiv;
- Abstract of the report at the scientific-practical conference "Informatics and Computing IOT-2018", Kyiv.

RECOMMENDER SYSTEMS, CLUSTERING, CLASSIFICATION OF DATA, ANALYSIS OF SOCIAL NETWORKS, CONSTRUCTION OF PSYCHOTYPE, MACHINE LEARNING

ЗМІСТ

ВСТУП

Кількість доступної користувачу інформації настільки велика, що важко виділити щось конкретне та необхідне шляхом звичайного перегляду. Тому системи, які допомагають аналізувати дані та орієнтують в них, представляють велику цінність.

Наукове дослідження, проведене Haubl and Murray в 2003 році показало, що на сайтах, які використовують персоналізовані товарні рекомендації, потенційним покупцям на третину простіше знайти продукти, в результаті чого, кількість повторних візитів, а як наслідок і продажів, збільшилася [44].

Аналіз соціальних даних стрімко набирає популярність у всьому світі. З цим пов'язаний феномен соціалізації персональних даних: стали публічно доступними факти біографії, переписка, щоденники, фото-, відео-, аудиоматеріали тощо. Таким чином, соціальні мережі є унікальним джерелом даних про особисте життя та інтереси реальних людей. Це відкриває безпрецедентні можливості для вирішення дослідницьких та бізнес- задач (багато з яких неможливо було вирішувати ефективно через недостатність даних), а також створення допоміжних сервісів та застосувань для користувачів соціальних мереж [79].

Інтерес до питання класифікації даних досі залишається на високому рівні, існує багато застосувань, які надають користувачам персональні рекомендації. Прикладами таких систем є: застосування, яке надає поради у виборі книжок, CD та інших продуктів на Amazon.com; радить фільми на MovieLens; а також новини в VERSIFI Technologies [62].

Проте, не дивлячись на всі ці успіхи, поточне покоління рекомендаційних систем як і раніше вимагає подальшого удосконалення. Ці поліпшення включають в себе ефективні методи для визначення рекомендацій, аналіз та прогнозування поведінки користувачів.

Внутрішньоресурсні рекомендації є звичною функцією соціальних мереж, але вони використовують лише власний контент для обробки. Також мають місце рекомендаційні системи інакших структур, які не враховують соціальну складову

користувача, а отже використовують вузький спектр інформації для формування рекомендацій. Подібні системи обмежені або в інформації про об’єкти рекомендацій, або в даних про користувача, що не дозволяє створити повноцінні та задовільні пропозиції.

Існуючі рекомендаційні системи орієнтовані на аналіз діяльності близьких користувачу соціальних кіл, які визначаються однаковою геолокацією, участю в схожих спільнотах, спільними друзями тощо. За рахунок встановлення сімейних та робочих контактів за допомогою соціальних мереж, даний метод може не відтворювати реальну картину. Частість спілкування між користувачами може бути викликана вимушеними чинниками. Тож, при побудові рекомендацій, важливо враховувати приховані аспекти портрету користувача – “психотип”, який може визначатися за допомогою аналізу його контенту.

У зв’язку з прагненням вирішити обидві проблеми однієї області, актуальною є розробка рекомендаційної системи на основі соціальних мереж, які допомагають у персоніфікації користувача та складанні його психотипу за допомогою аналізу соціального профілю.

1 ОГЛЯД РІШЕНЬ ЩОДО НАДАННЯ РЕКОМЕНДАЦІЙ

Створення рекомендаційних систем – відносно новий напрямок розвитку програмного забезпечення. Свій початок він взяв у 1990-х роках. Однак серйозним поштовхом до розвитку цієї теми був конкурс Netflix Prize, організований в 2006 році компанією Netflix, лідером прокату DVD на американському ринку [42]. Учасники мали як можна краще передбачити, яку оцінку поставить користувач певному фільму. Якість передбачень вимірювалася за допомогою метрики RMSE (середньоквадратичне відхилення). Netflix мала алгоритм, який передбачав оцінки користувачів з якістю 0.9514 за метрикою RMSE. Задачею було покращити цей показник хоча б на 10%, а переможець отримав би мільйон доларів. В результаті, методи побудови рекомендаційних систем почали стрімко розвиватися [35].

Пошукова система є корисною у випадку, коли користувач має специфічні потреби, та використовує ключові слова, які можуть описати ці потреби. Рекомендаційна система слугує у випадках, коли невідомі конкретні потреби користувача, а також немає ключових слів для опису цих потреб. Тож, рекомендаційна система – це зручна альтернатива пошуковим алгоритмам, так як дозволяє знайти неочевидні об’єкти [74,32]. Її місією є:

- допомога користувачу в пошуку цікавих йому об’єктів;
- допомога постачальникам в представленні їх послуг підходящим користувачам;
- допомога веб-ресурсам в залученні користувачів.

Прикладами відомих рекомендаційних систем є:

- Amazon [53] – один з лідерів області. Рекомендує книги та інші товари, основуючись на тому що користувач купував, переглядав, які рейтинги ставив тощо;
- Netflix [80] – надання послуг оренди фільмів (в минулому), зараз рекомендує потокове відео. Надання рекомендацій на основі історії переглядів та надання оцінок;

- Last.fm [71] – система рекомендації музики. Для надання рекомендацій використовує рейтинги власних користувачів, а також “зовнішні” дані про музику – автор, стиль, дата, теги тощо;
- Pandora [82] – система рекомендації музики. Для надання рекомендацій використовує “зміст” музичної композицій, використовуючи дані Music Genome Project, в якому професійні музиканти аналізують треки за декількома сотнями атрибутів;
- Google, Yahoo [100,64] – пошукові системи, які намагаються передбачити наскільки даний документ релевантний даному запиту, який рейтинг користувач надасть даному продукту.

Рекомендаційні системи є дуже потужним складником в комерційних проектах. Таким чином [91]:

- 2/3 орендованих фільмів з Netflix обираються користувачами з індивідуальних рекомендацій;
- 38% найбільш популярних новин сервісу Google News були рекомендовані;
- 35% продажів з Amazon – рекомендовані товари.

Соціальні мережі часто рекомендують користувачам потенційних друзів, музику, події та інший контент в залежності від кількості соціальних зв’язків. Найвідоміші приклади подібних ресурсів: Facebook [61], Instagram [68], Twitter [96] тощо. Цей тип рекомендацій має дещо інші цілі, ніж рекомендації предметів комерції. Продажі компаній залежать від порад системи щодо вибору певного продукту, збільшення числа соціальних зв’язків тільки підвищує можливості користувача всередині мережі. Це, в свою чергу, збільшує саму мережу, так як вона сильно залежить від кількості користувачів та їх задоволеності. Так, рекомендаційна система, укріпляє положення соціальної мережі на арені подібних ресурсів. Така форма рекомендацій базується радше на структурних відносинах, ніж на рейтингових даних. Таким чином, характер базових алгоритмів змінюється.

Користувачі соціальних мереж часто зважають на думку своїх друзів стосовно музики, відео, подій тощо. Однак, якщо користувач також пов'язаний зі

спеціалізованими спільнотами, для створення рекомендацій справедливо буде використати і ці дані. Таким чином, на процес створення рекомендацій впливає структура мережі та користувачі або спільноти зі схожими інтересами, а також особисті дані користувача.

Не завжди об’єкти, зазначених вподобань в соціальних мережах, відповідають дійсному настрою та характеру користувача. Для виявлення більш точних точок вподобань, доречно використовувати засоби аналізу його психотипу. Об’єднавши цю інформацію з отриманою раніше, можна отримати більш точний портрет кінцевого споживача.

Так, тільки за допомогою аналізу фотографій можна скласти певний висновок, наприклад, за допомогою використання розпізнавання обличь та емоцій, або визначення домінуючих кольорів набору світлин [2].

Ефективність роботи рекомендаційної системи в значній мірі залежить від об’єму та якості даних про активність користувача. Тож взаємодія користувача з соціальною мережею за допомогою форм зворотного зв’язку є важливою складовою[44, 52].

Для оцінки ефективності рекомендаційних систем та готових алгоритмів можна застосовувати різні набори метрик [20]:

- метрики точності прогнозу (MAE, MSE, RMSE);
- метрики прийняття рішень (ROC, Precision/recall);
- метрики оцінки ранжування результатів (Spearman Rank Coefficient, Fraction of concordant pairs);
- бізнес-метрики (збільшення продаж, конверсії).

1.1 Аналіз проблеми збору даних з соціальних мереж

При роботі з соціальними даними необхідно приймати до уваги такі фактори, як нестабільність якості користувацького контенту (спам та фальшиві облікові записи), проблеми з забезпеченням приватності особистих даних при збереженні та обробці, а також часті оновлення користувацької моделі та функціоналу. Все це

вимагає постійного удосконалення алгоритмів вирішення аналітичних та бізнес-задач.

Обробка соціальних даних вимагає також розробки відповідних рішень, що дозволяють враховувати їх розмірність. Так, наприклад, база даних соціальної мережі Facebook на сьогоднішній день містить більше 1 мільярда користувацьких облікових записів та більше 100 мільярдів зв'язків між ними. Кожен день користувачі додають більше 200 мільйонів фотографій та залишають більше 2 мільярдів коментарів до різноманітних об'єктів мережі. На сьогоднішній день більшість існуючих алгоритмів, які дозволяють ефективно вирішувати актуальні задачі, не спроможні обробляти дані подібної розмірності за прийнятний час. У зв'язку з цим, виникає необхідність в нових рішеннях, які дозволяють розподілену обробку і зберігання даних без суттєвої втрати якості результатів [73].

Оскільки сценарії користування інтерфейсів соціальних мереж не передбачають автоматичного збору даних множини користувачів з метою побудови соціального графа, то виникає ряд проблем:

- приватність даних – часто доступ до даних користувачів дозволений тільки для зареєстрованих та авторизованих учасників мережі, що вимагає підтримки емуляції користувацької сесії за допомогою спеціальних облікових записів;
- слабка структурованість даних – в багатьох випадках програмні інтерфейси (API) соціальних мереж мають обмежений функціонал, що вимагає вилучення необхідних даних за допомогою спеціальних алгоритмів та побудови їх структурованого представлення, зручного для подальшої обробки;
- розмірність даних обумовлює необхідність в паралельному методі збору даних, а також в методах отримання репрезентативної вибірки користувачів соціальної мережі.

Під час заповнення свого профіля в соціальній мережі користувачі часто помилково або спеціально не заповнюють деякі поля або надають неправдиву

інформацію про факти своєї біографії, інтереси та вподобання. Крім того, користувацький профіль часто обмежений набором базових атрибутів, що є недостатнім для рішення багатьох задач, що передбачають персоналізацію результатів [63].

1.2 Аналіз методів надання рекомендацій

Перший етап розвитку рекомендаційних систем ознаменувався появою алгоритмів колаборативної фільтрації. В 1990-х роках з появою перших цифрових магнітофонів з’явилася потреба передбачати вподобання користувача в фільмах та передачах. Так як на той час не існувало добре структурованих описів для всіх передач та потужність користувацьких пристроїв не була достатньою для важких обчислень, основою системи рекомендацій став алгоритм колаборативної фільтрації [51].

В таких системах рекомендації користувачу визначаються на основі оцінок інших користувачів, опис об’єкту рекомендації не використовується. Тобто, якщо два різні користувачі зазвичай обирають одні й ті самі об’єкти, а потім один користувач змінює свої вподобання, інший отримує рекомендації подібно першому [34].

Спосіб розгляду подібних відношень, які засновані на подібності та несхожості зображений на рисунку 1.1 [33].

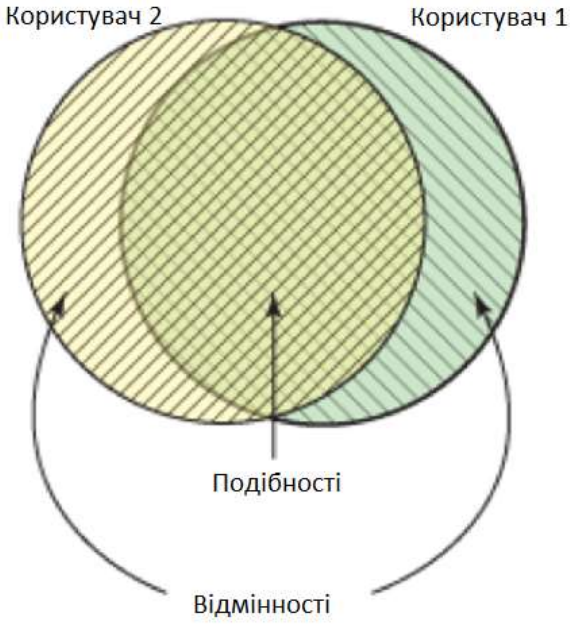

Рисунок 1.1 – Розгляд відносин для формулювання рекомендацій

Подібності визначають за якими ознаками варто групувати користувачів, які мають схожі інтереси. Відмінності – можливості, які можуть бути використані для створення рекомендацій.

Алгоритм колаборативної фільтрації має ряд проблем [18, 15]:

- проблема холодного старту;
- розрідженість даних;
- проблема масштабованості.

Проблема холодного старту постає при появі нового користувача або об’єкта – невідомо що порадити користувачу без історії вподобань або як змусити систему рекомендувати щойно створену подію. Для того, щоб визначити смаки користувача, системі доводиться проводити анкетування під час реєстрації, тим самим створюючи нову проблему – настирливість. Більшість рекомендаційних сервісів передбачають деяку активність, вони змушують користувача оцінювати десятки подій, фільмів, товарів, аналізують час, витрачений на перегляд певного матеріалу. Але подібні дані не є точними та не можуть повністю охопити потреби та вподобання користувача.

Ще одна перепона – розрідженість даних. Попит на рекомендації зазвичай перевищує наявність оцінок в системі. Як правило, користувачі воліють не надавати оцінки, а отримувати їх, не брати участь у наповненні бази даних, а користуватися

нею. В результаті чого, матриця «подія-користувач» є великою та розрідженою. Користувацька неактивність також, в свою чергу, є чинником виникнення проблеми холодного старту. В будь-якому випадку, необхідна певна критична маса користувацьких оцінок. Наприклад, якщо в системі, яка рекомендує події, частина об'єктів оцінюється малою кількістю користувачів, дані об'єкти будуть рекомендуватися рідко, навіть при високих оцінках. Тобто, якщо в базі кількість експертів буде малою в порівнянні з кількістю об'єктів, то прогнози будуть неточними. Ця проблема є особливо гострою для нових, нещодавно виниклих об'єктів оцінювання.

Проблема масштабованості носить технічний характер та пов'язана зі складністю обчислень при роботі з великими обсягами даних. При великій кількості користувачів та подій, алгоритм колаборативної фільтрації стає занадто важким для розрахунків. Розробники систем цієї області зазначають, що всі сучасні алгоритми колаборативної фільтрації були розроблені на невеликих базах даних. Наприклад, MovieLens працює з 35 000 клієнтами та 3000 товарами, а EachMovie з базою, яка складається з 4000 користувачів та 1600 товарами. Дороговартісні обчислення доречно виконувати в офлайн-режимі, але традиційна поклієнтська фільтрація в такому режимі практично не функціонує, а виконувати обчислення в режимі реального часу важко. Це можливо тільки при зменшенні кількості чинників створення рекомендацій, що в свою чергу зменшує їх якість. Інакше, обслуговування подібної системи стає невиправдано затратним.

Можна виділити ще одну проблему алгоритму колаборативної фільтрації – рекомендація принципово нових об'єктів. Багато діючих систем ігнорують такі можливості. Наприклад, якщо користувач обере для відвідування декілька концертів класичної музики, то система вже точно не порадить йому рок-фести. Фільтрація занадто міцно базується на схожості об'єктів. З цією причини рекомендації бувають або занадто загальними (всі події однієї тематики), або занадто вузькими (виступи однієї й тої самої групи). Намагаючись розв'язати дану проблему, деякі системи, наприклад, Daily-Leaner, ігнорують дуже подібні до вподобаних користувачем об'єкти. І це знову таки, може погіршити якість рекомендацій.

Не дивлячись на те, що рішення із застосуванням алгоритму колаборативної фільтрації не працювало в реальному часу та іноді неправильно визначало інтереси користувача, простота підходу швидко зробила його популярним в своїй області. Але в середині 2000-х якість рекомендацій досягла деякої границі і, в результаті, класичний метод колаборативної фільтрації як основний метод рекомендацій зайшов у глухий кут [51, 34, 85].

Другий етап розвитку рекомендаційних систем розпочався з винайденням алгоритму фільтрації по контенту. В процесі розвитку нових підходів до рішення задачі на гребені хвилі з'явилися нові компанії, зокрема, Jinni та Aprico, які зрозуміли, що якісна інформація про рекомендуємий контент могла би слугувати серйозною основою для підлаштування під інтереси користувачів. Ця нова хвиля досліджень використовувала байєсовську класифікацію, яка дозволяла зрозуміти та оцінити чому користувач віддає перевагу тому чи іншому контенту [51].

Основні кроки при використанні даного підходу [34]:

- аналіз контенту об'єктів;
- визначення набору його критеріїв (жанри, теги, слова);
- визначення критеріїв, які подобаються користувачу;
- співставлення отриманих даних та отриманих рекомендацій.

Цей підхід має такий ряд проблем:

- масштабованість;
- невисока точність;
- виправданий тільки якщо каталог подій відносно малий або збільшується поступово;
- обмеженість у зв'язку з властивостями об'єктів;
- схожість різних об'єктів;
- вузькі рекомендації.

Проблема схожості різних об'єктів полягає в тому, що два різні об'єкти, які представлені однаковим набором властивостей, неможливо розрізнити.

Якщо система рекомендує тільки ті об’єкти, характеристики яких співпадають зі смаком користувача, то це означає, що він отримає рекомендації тільки таких об’єктів, які схожі з вподобаними раніше – отримає вузькі рекомендації.

До переваг алгоритму можна віднести більш теплий «холодний старт», для отримання рекомендацій необхідні тільки дані про користувача та об’єкт. Дуже часто подібні системи використовують перед колаборативною фільтрацією, коли оцінок користувача ще недостатньо.

Фільтрація по контенту унікально характеризує кожного користувача, але колаборативна фільтрація все-одно має певні переваги. По-перше, колаборативна фільтрація буде ефективною, коли важко проаналізувати та чітко означити об’єкт рекомендації. По-друге, колаборативна фільтрація має можливість надавання нетривіальних рекомендацій, які будуть відповідати не вподобаному користувачем контенту, а його профілю [85].

Третій етап розвитку рекомендаційних систем алгоритми гібридної фільтрації, яка комбінує методи колаборативної фільтрації та фільтрації по контенту, яка дозволяє оминути недоліки попередніх наведених методів. Способи поєднання цих методів можуть різнитися [62]:

- окрема реалізація методів та поєднання результатів;
- включення деяких характеристик фільтрації по контенту в колаборативну;
- включення деяких характеристик колаборативної фільтрації в контентну;
- побудова загальної уніфікованої моделі, яка включає характеристики обох методів.

Для реалізації методів колаборативної фільтрації можливе використання таких алгоритмів:

- кластеризація користувачів;
- user-based (алгоритм, заснований на аналізі користувачів);
- item-based (алгоритм, заснований на аналізі об’єктів);
- SVD (Singular value decomposition – сингулярний розклад матриці).

Метод кластеризації користувачів передбачає визначення умовної міри схожості користувачів, об’єднання їх в кластери таким чином, щоб схожі користувачі опинилися в одному кластері. Оцінка користувача буде передбачатися як середня оцінка по кластеру для даного об’єкта.

Кластеризацію можна проводити за алгоритмом k-середніх, ціллю якого є розділення m спостережень на k кластерів, при чому кожне спостереження відноситься до того кластера, до центру якого воно ближче. В якості мірі близькості зазвичай використовується Евклідова відстань. При використанні такого методу, кількість кластерів невідома та обирається дослідником заздалегідь, а якість кластеризації залежить від першочергового розбиття [17].

Крім проблем, зазначених при загальному огляді колаборативних алгоритмів, можна додати наступні недоліки:

- не враховується специфіка кожного користувача – всі користувачі діляться на класи (шаблони);
- якщо в кластері ніхто не оцінював об’єкт, то отримати рекомендацію буде неможливо.

Алгоритми user-based (створення рекомендацій, заснованих на користувачах) та item-based (створення рекомендацій, заснованих на об’єктах) схожі за своєю концепцією. В обох випадках, спочатку необхідно знайти наскільки користувачі/об’єкти в базі даних схожі на аналізуємого користувача/об’єкт, потім за оцінками інших користувачів/об’єктів передбачити вибір даного користувача/об’єкта, з більшою вагою враховуючи тих користувачів/об’єкти, які більше схожі на даний. Для виявлення коефіцієнта схожості рахується коефіцієнт кореляції Пірсона.

Прогалинами цих алгоритмів є:

- холодний старт;
- відсутність рекомендацій для нових користувачів/об’єктів;
- тривіальність рекомендацій.

Також до недоліків вже описаних алгоритмів можна додати велику ресурсоємність обчислень – для того, щоб створювати рекомендації необхідно тримати в пам’яті всі оцінки всіх користувачів.

Алгоритм SVD (Singular Value Decomposicion – сингулярний розклад матриці) позбавлений цих мінусів. Маємо матрицю, яка складається з оцінок, які користувачі присвоїли об’єктам. За допомогою її сингулярного розкладу отримаємо рекомендації. Так, використовуючи історію користувачів, можна виявити приховані ознаки об’єктів [35].

При використанні алгоритмів контентної фільтрації застосовують:

- наївний байєсьвський класифікатор;
- алгоритми кластеризації (наприклад, метод k-середніх);
- дерев прийняття рішень.

Наївний байєсовський класифікатор – алгоритм класифікації, заснований на теоремі Байєса для розрахунку ймовірностей. Назва «наївний» походить від наївного припущення, що всі змінні, які розглядаються, незалежні одна від одної. В дійсності, це не завжди так, але не практиці цей алгоритм все ж знаходить застосування [7].

Перевагами даного методу є [1]:

- класифікація, в тому числі багатокласова, виконується швидко;
- коли припущення про незалежність виконується, класифікатор перевершує інші алгоритми, наприклад, як логістична регресія, вимагаючи при цьому менший об’єм даних для навчання;
- алгоритм краще працює з категоріальними ознаками, ніж з неперервними.

Недоліками даного методу є:

- якщо в тестовому наборі даних присутнє деяке значення категорійного типу, яке не зустрічалося в навчаючому наборі даних, модель присвоїть нульову вірогідність цьому значенню та не зможе зробити прогноз;
- хоч алгоритм є гарним класифікатором, значення прогнозованих вірогідностей не завжди є достатньо точним.

– припущення про незалежність ознак не завжди коректне. В реальності набори повністю незалежних ознак зустрічаються дуже рідко.

В рекомендаційних системах також можливе застосування алгоритмів, які базуються на деревах прийняття рішень. Головною перевагою застосування подібних алгоритмів є дешевизна їх побудови та використання, а також висока швидкість класифікації невідомих об’єктів. Також вони можуть бути використані для створення набору правил, які легко інтерпретувати, зберігаючи при цьому точність в порівнянні з іншими методами, які застосовуються в рекомендаційних системах [92]. Прикладами подібних алгоритмів є Random Forest та ID3.

Random Forest, або алгоритм випадкового лісу, є популярним методом розв’язання задач машинного навчання. Цей алгоритм є відносно молодим – кінцеве його представлення відбулося в 2001 році. Метод передбачає собою створення множини дерев прийняття рішень та усереднення результату їх передбачень. Важливим є фактор випадковості при створенні кожного дерева. Зрозуміло, якщо ми створимо багато однакових дерев, результат їх усереднення набуде рішення одного дерева [16].

Нехай ми намагаємося побудувати дерево рішень для моделі, яка має 10 атрибутів. І нехай в першому вузлі найкращим варіантом буде використати змінну №1 (наприклад, у випадку жадібного алгоритму), але, на справді, оптимальніше буде використовувати змінну №4. Саме в такій ситуації можна якнайкраще сформулювати гасло алгоритму Random Forest: якщо ми не можемо побудувати одне оптимальне дерево, то побудуємо 1000 неоптимальних дерев (кожне неоптимальне по-своєму), які будуть перекривати недоліки одне одного. Дерева можуть використовувати будь-які змінні з випадковою кількістю атрибутів. Кожне з них буде неоптимальним, можливо, неефективним, але разом вони утворюватимуть працюючий ліс, який дозволить правильно рекомендувати події користувачу [76].

До переваг методу випадкового лісу можна віднести:

– висока швидкість навчання (побудови дерев);

– ефективність для обробки великих об’ємів даних;

– висока точність.

Недоліки алгоритму:

– побудована модель займає велику кількість пам’яті. Якщо ми будуємо ансамбль з K дерев на основі навчаючої вибірки розміром N, то вимоги до пам’яті складають O(K*N);
– навчена модель працює повільніше інших алгоритмів (якщо модель містить 100 дерев, треба пройти по всім, щоб отримати результат);
– може створювати занадто складні конструкції, які недостатньо повно представляють дані.

Алгоритм ID3 використовує рекурсивне розбиття підмножин у вузлах дерева по одному з обраних атрибутів. Для обрання атрибуту, за яким буде виконуватися розщеплення використовується критерій збільшення інформації або зменшення ентропії.

Перевагами даного методу є [29, 11]:

– простота інтерпретації класифікації. Алгоритм здатний не тільки класифікувати об’єкт, але й надати пояснення даного вибору в термінах предметної області;
– алгоритм синтезу дерева має складність, лінійну довжині вибірки;
– є гнучким до змін.

Недоліки алгоритму:

– жадібність. Локально оптимальний вибір предикату не обов’язково оптимальний глобально. У випадку невдалого вибору алгоритм не може повернутися на рівень вверх та змінити рішення;
– чим далі вершина u розташована від кореня дерева, тим менша довжина вибірки U, за якою приймається рішення про гілкування у даній вершині. І тим менш статистично надійним стає вибір предиката.

Алгоритми асоціативних правил зосереджені на пошуку закономірностей, за їх допомогою також можливе створення рекомендаційних систем. Ефективність подібних методів для виявлення персоналізованих рішень очевидна, однак, не

зважаючи на чітку застосовність алгоритмів у цілях рекомендаційних систем, вони не здобули популярності в даній сфері. Головною причиною цьому є схожість з методами колаборативної фільтрації, причому застосування асоціативних правил є менш гнучким рішенням [92].

Основним алгоритмом, який застосовується для отримання асоціативних правил є алгоритм Apriori. Він призначений для пошуку всіх частих множин ознак. Він є порівневим, використовує стратегію пошуку в ширину та виконується знизу-вверх. Основна особливість алгоритму – властивість антимонотонності – підтримка будь-якого набору елементів не може перевищувати мінімальної підтримки будь-якої з її підмножин. Завдяки цій властивості алгоритм не є жадібним та дозволяє обробляти великі масиви інформації за малий час. Одночасно, даний метод достатньо вимогливий до пам’яті та часу генерації елементних наборів [56, 2].

В рекомендаційній системі також можливе використання мурашиного алгоритму. Ціль метода – отримати прості правила виду: якщо «умова», то «наслідок». Передбачається, що всі атрибути категоріальні, тобто терми представлені у вигляді Атрибут = Значення. Алгоритм послідовно формує впорядкований список правил. Обчислення починається з порожньої множини правил, а після формування першого, всі тестові одиниці даних, які покриті цим правилом, видаляються з тестового набору.

В своїй роботі метод використовує направлений граф, де кожному атрибуту співставляється стільки вершин, скільки можливих значень він приймає в тестовому наборі. Відповідно, припускається, що перед початком роботи алгоритма, з тестового набору виділені множини атрибутів та можливих значень, а також ймовірних класів [38].

До переваг методу можна віднести можливість використання в динамічних застосуваннях (адаптується до змін).

Недоліками алгоритму є [47]:

– збіжність гарантується, але час збіжності не визначений;

- сильна залежність від налаштовуваних параметрів, які підбираються ґрунтуючись тільки на експериментах;
- теоретичний аналіз ускладнений (в результаті послідовності випадкових рішень, розподіл ймовірностей змінюється при ітераціях);
- дослідження є скоріш експериментальним, аніж теоретичним.

1.3 Висновки до розділу

В результаті вивчення літератури за темою, визначені основні недоліки оглянутих методів рекомендаційних систем, а саме:

- процес збору та якість аналізуємих вибірок;
- “холодний” старт;
- можливість виникнення несподіваних закономірностей в навчаючій вибірці, які не підтверджуються в тестовій;
- особливості роботи з даними категоріального типу з великим набором змінних;
- проблема незаповненості даних та їх аномальності (наприклад, користувач надає некоректні або несумісні особисті дані щодо місця проживання);
- визначення важливості атрибутів та їх впливу на рекомендаційну систему.

Одною з домінуючих тенденцій розвитку соціальних мереж як соціокультурного феномену є більш глибоке розуміння особливостей соціальної поведінки людини, і, як наслідок, створення нових засобів для самовираження, а також обміну інформацією та досвідом. Розумно очкувати подальшого розширення користувацької моделі та функціоналу різноманітних соціальних мереж, що призведе до появи нових типів даних у вигляді об’єктів та зв’язків соціального графа та, як наслідок, можливості ефективніше розв’язувати задачі, пов’язані з обробкою персональної інформації [59,4].

Огляд наявних рішень також показав, що немає універсального методу, який охоплював би роботу з багатокритеріальними даними, надаючи персональні рекомендації, які спираються на психотип користувача. Подібні системи

використовують досить вузьке коло даних для формулювання порад щодо вподобань користувачем об’єктів. Часто вони є надто нав’язливими при анкетуванні, а отримані рекомендації є очевидними. Для присвоєння контенту певної категорії використовуються ручні методи, що обмежує шанси рекомендацій даного об’єкту з огляду на суб’єктивність експерта та велике різноманіття тегів і способів означення інформації. Таким чином, можна стверджувати, що методи, які ґрунтуються на даних про об’єкти знаходяться в програшній ситуації. Для втримання високих рейтингів ресурсу, що використовує рекомендаційні системи, також важливо надавати нові, неочікувані комбінації об’єктів для попередження втомленості користувача. Тож, необхідно, дослідити та знайти способи покращення розглянутих підходів та методів, де ключовими факторами є якість та складність визначення рекомендацій.

2 МОДЕЛІ ТА МЕТОДИ РОЗВ'ЯЗАННЯ ЗАДАЧ НАДАННЯ РЕКОМЕНДАЦІЙ

Як було визначено в попередньому розділі, вибірка даних, що аналізується є суттєвим фактором, що впливає на якість надання рекомендацій. Тому, розділи 2.1, 2.2 будуть присвячені аналізу вхідної вибірки.

Розділи 2.3 – 2.8 стосуватимуться опису методів надання рекомендацій.

2.1 Визначення домінантних кольорів на фотографії

Маємо задачу визначення домінантних кольорів на фотографії задля отримання додаткового параметру для персоніфікації користувача.

2.1.1 *Усереднення кольору*

Очевидним варіантом вирішення проблеми є усереднення всіх кольорів, що містяться на зображенні. Але це не призведе до правдивих результатів. Можлива ситуація, коли відсутній на фотографії R/G/B колір буде результатом усереднення всіх пікселів. В результаті, неправильна гама кольорів буде визначена домінуючою.

2.1.2 *Частота появи кольорів*

Вибір кольорів, які зустрічаються найчастіше. Проблеми цього способу виявляються при обробці фотографій на які накладений фільтр або на засвічені/затемнені фотографії. В такому випадку, наприклад, при засвіченій фотографії, домінантний колір буде білий за рахунок надлишкової освітленості.

2.1.3 *Ієрархічна кластеризація*

Ієрархічна кластеризація створює ієрархію груп шляхом постійного злиття двох найбільш схожих груп. Кожна з цих груп спочатку складається з одного вузла. В кожній новій ітерації метод обчислює дистанцію між всіма парами груп, а найбільш

схожі – об’єднує. Цей процес повторюється до тих пір, поки не сформується одна група [94].

Оскільки зв'язок між кожною парою вузлів має бути обчислений декілька разів алгоритм є дуже повільним для великих масивів даних.

2.2 Визначення кількості обличь на фотографії

Нехай існує зображення. Необхідно визначити кількість обличь, що зображено на ньому.

2.2.1 *Порівняння шаблонів*

Метод полягає у виділенні областей обличчя на зображенні та наступному порівнянні цих областей для двох різних зображень. Кожна область, що співпала збільшує міру подібності зображень. Для порівняння використовуються алгоритми попіксельного порівняння.

Недоліком цього методу є необхідність у великих обсягах ресурсів як для зберігання ділянок, так і для порівняння. Фотографії мають бути виконані у строго встановлених умовах: не допускається помітних змін ракурсу, освітлення, емоціонального окрасу тощо [49].

2.2.2 *Лінійний дискримінантний аналіз*

Даний метод використовує таку проекцію простору зображень на простір ознак, яка мінімізує внутрішньокласову та максимізує міжкласову відстань в просторі ознак. В даних методах передбачається, що класи лінійно розділені.

Матриця W для проектування простору зображень на простір ознак обирається за наступною умовою:

$$W_{opt} = arg\max_{W} \frac{W^T S_B W}{W^T S_W W}, \tag{2.1}$$

де S_B – матриця міжкласової дисперсії;

S_W – матриця внутрішньокласової дисперсії.

Може існувати до $(c-1)$ векторів, які утворюють базис простору ознак, де c – загальна кількість класів. За допомогою цих векторів простір зображень переводиться в простір ознак.

Оскільки робота з матрицею $S_W \in R^{nxn}$ важка через її розмірність, використовується зменшення розмірності за допомогою метода головних компонент, після чого обчислення здійснюються в просторі меншої розмірності:

$$W_{fld} = arg \max_W \frac{W^T W_{pca}^T S_B W_{pca} W}{W^T W_{pca}^T S_W W_{cpa} W}, \tag{2.2}$$

де W_{cpa} – матриця для проектування в простір меншої розмірності (простір головних компонент).

Зазвичай тренувальний набір містить зображення обличь за деяких базових умов освітленості, на основі яких за допомогою лінійних комбінацій можна отримати будь-які інші умови освітленості. Цей метод дає високу точність розпізнавання для широкого діапазону умов освітленості та різних виразів обличчя. Однак, залишаються невирішеними питання застосування методу для великих наборів даних та можливість роботи алгоритму в ситуаціях, коли в тренувальній вибірці для деяких обличь існує зображення тільки в одних умовах освітленості [12].

2.3 Методи обробки категоріальних ознак

2.3.1 *Випадкова нумерація ознак*

Метод випадкової нумерації ознак створює нелогічні залежності та змушує алгоритми враховувати впорядкований перелік значень ознак, що не має чіткої логіки. Подібні алгоритми є нестійкими відносно різних способів нумерації ознак та показують низьку якість обробки.

2.3.2 *One-hot кодування*

Для кодуємої категоріальної ознаки створюється q нових ознак, де q - кількість категорій. Кожна i-та нова ознака – бінарна категорійна ознака i-ї категорії.

Такий метод збільшує розмірність вхідної матриці та забезпечує її сильну розрідженість.

2.4 Метод колаборативної фільтрації

2.4.1 *Постановка задачі*

Нехай u - користувачі, що є в Системі, $u = \overline{1,n}$;

i – об’єкти, що оцінюються, $i = \overline{1,m}$;

R - матриця з оцінками, де r_{ui} – оцінка користувача u об’єкту i.

Нехай деякі користувачі оцінили деякі об’єкти. І нехай оцінка – натуральне число від 1 до N. Тоді всі оцінки можна відобразити у вигляді матриці, приклад якої зображено на рисунку 2.1.

Користувачі \ Об'єкти	1	2	…	i	…	m
1	5	3		1	2	
2		2				4
…			5			
u	3	4		2	1	
…					4	
n			3	2		
a	3	5		?	1	

Рисунок 2.1 – Матриця оцінок

Нехай існує користувач a. Задача – передбачити яку оцінку поставив би користувач *а і-му* об’єкту.

2.4.2 *Опис методу*

Колаборативна фільтрація – метод, який дає прогнози відносно інтересів користувача по зібраній інформації про смаки множини користувачів. Його основне припущення – ті, хто погоджувався в минулому, схильні погоджуватися в майбутньому.

Варто відмітити, що ці прогнози індивідуальні, хоч використовувана інформація зібрана від багатьох учасників. Тим самим даний метод відрізняється від більш простого підходу, який дає усереднену оцінку для кожного об’єкта.

Системи колаборативної фільтрації зазвичай застосовують двоступеневу схему:

1) знаходять тих, хто розділяє оціночні судження активного (прогнозованого) користувача;
2) використовують оцінки користувачів, що мислять подібно активному користувачу для обчислення прогнозу.

2.4.3 *Опис алгоритму*

Розглянемо тільки користувача *a* та тих користувачів, які оцінили об’єкт *i.*

Алгоритм включає в себе три кроки:

1) для кожного користувача *u* вичислимо, наскільки його інтереси співпадають з інтересами користувача *a*;
2) після цього оберемо множину користувачів, найбільш близьких до *a*;
3) передбачимо оцінку на основі оцінок об’єкта *i* “сусідами” з попереднього кроку.

Детальний опис алгоритму наведено далі.

Перший крок. Кожному користувачу в матриці *R* відповідає один рядок. Тож, треба обчислити близькість векторів-рядків користувачів.

Для обчислення міри близькості між векторами візьмемо коефіцієнт кореляції Пірсона:

$$sim(u,a) = \frac{\sum_{i \in I}(r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i \in I}(r_{a,i} - \overline{r_a})^2 \cdot (r_{u,i} - \overline{r_u})^2}}, \quad (2.3)$$

де *sim(u, a)* – міра близькості (схожості) користувачів *a* та *u*;

$r_{u,\,i}$ – значення матриці R;

u – номер рядка;

i – номер стовпчика.

Якщо користувач не вказав оцінку для якогось об'єкта, відповідне значення матриці рівне 0.

I – множина об'єктів, які оцінив як користувач *a* так і користувач *u*. $\overline{r_a}$, r_u – середні оцінки користувачів *a* та *u* відповідно.

В чисельнику підраховується добуток відхилень оцінок двох користувачів від середніх значень для одного об'єкта. Знаменник необхідний для того, щоб дана величина приймала значення з відрізка *[-1, 1]*. Чим сильніше співпадають інтереси, тим ближче значення близькості до *I*. Якщо коефіцієнт Пірсона від'ємний, то інтереси користувачів протилежні.

Другий крок. Тепер необхідно обрати множину *K* найбільш схожих на *a* користувачів. Можна обрати всіх користувачів. Але, так як користувачів з несхожими або несильно пересічними інтересами достатньо багато, то вони будуть негативно впливати на точність передбачення оцінки для об'єкта *i*. Крім того, кількість користувачів впливає на об'єм обчислень на третьому кроці.

Одне з можливих рішень – встановити поріг міри близькості, обчисленої на першому кроці. Користувачі з мірою близькості, що перевищує поріг, ввійдуть в множину *K*. Інші – ні. Але частіше за все обирається ціла константа *k*. Потім всі користувачі сортуються за спаданням міри близькості. І в множину *K* входить *k* користувачів, найбільш близьких до *a*.

Третій крок. Маючи множину *K* близьких користувачів необхідно обчислити оцінку, яку поставив би користувач *a* об'єкту *i*. Варто нагадати, що оцінюються тільки ті користувачі, що оцінили об'єкт і.

Необхідна оцінка обчислюється за наступною формулою:

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u \in K}(r_{u,i} - \overline{r_u}) \times sim(a,u)}{\sum_{u \in K}|sim(a,u)|}, \quad (2.4)$$

де $p_{a,i}$ – передбачувана оцінка користувача *a* для об'єкта *i*.

За основу береться його середня оцінка $\overline{r_a}$, а потім додається середнє відхилення оцінки інших користувачів з множини *K* для об'єкта і від їх середньої оцінки. Чим ближче користувач *u* до користувача *a* (згідно з мірою близькості $sim(u,a)$, обчисленою на першому кроці), тим сильніше його вклад в передбачення оцінки.

Таким чином, описаний алгоритм передбачає оцінки для об'єктів, які поточний користувач ще не оцінив. Для того, щоб запропонувати рекомендацію для даного користувача, достатньо передбачити оцінки для всіх неоцінених об'єктів та обрати об'єкти з найбільшою передбаченою оцінкою [81, 99, 10].

Складність алгоритму – $O(mn)$.

2.4.4 *Недоліки*

Недоліками та слабкими місцями даного методу є [57, 19]:

- тривіальність наданих рекомендацій (пропонуються найбільш популярні об'єкти);
- не враховуються інтереси окремих користувачів;
- проблема холодного старту (неможливо надати рекомендацію користувачу, який не має "історії" оцінювання);
- розрідженість даних, так як користувачі немає точно визначених оцінок для всіх пар "користувач-об'єкт";
- слабкість точності наданих рекомендацій для користувачів з унікальними характеристиками.

2.5 Алгоритм наївних мереж Байєса

2.5.1 *Опис ідеї*

Теорема Байєса – одна з основних теорем теорії ймовірностей, яка визначає вірогідність настання події в умовах, коли на основі спостережень відома лише деяка часткова інформація про події [13].

Умовна ймовірність події *x* за умови події *y* позначається *p(x|y)*. Згідно з теорією ймовірностей:

$$p(x|y) = \frac{p(x,y)}{p(y)}, \tag{2.5}$$

де $p(x, y)$ - спільна ймовірність подій x, y;

$p(x)$, $p(y)$ – апріорні ймовірності кожної події окремо.

Таким чином, спільну ймовірність можна виразити двома способами:

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x). \tag{2.6}$$

За теоремою Байєса, умовна ймовірність $p(x|y)$ визначається наступним виразом:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}, \tag{2.7}$$

де *y* – вхідні дані (відома інформація);

x – параметри моделі, які ми хочемо обучити;

$p(x|y)$ – апостеріорна ймовірність – розподілення ймовірностей параметрів моделі після того, як були враховані вхідні дані;

$p(y|x)$ - правдоподіність – ймовірність даного значення за умови зафіксованих параметрів моделі.

Ціль класифікації полягає в тому, щоб зрозуміти до якого класу належить об’єкт, тому шукаємо не ймовірність, а найбільш вірогідний клас. Байєсовський класифікатор використовує оцінку апостеріорного максимуму для визначення найбільш вірогідного класу:

$$x_{map} = arg \max_{x \in X} \frac{P\left(\frac{y}{x}\right)P(x)}{P(y)}. \quad (2.8)$$

Тобто нам треба розрахувати ймовірність для всіх класів та обрати той клас, який має найбільшу ймовірність [27].

2.5.2 *Опис алгоритму*

Маємо навчальний набір даних з переліком ознак та цільову змінну з визначеними значеннями для кожного набору даних [14].

Алгоритм використання наївних мереж Байєса полягає в наступному:

Крок 1. Перетворити наявний набір даних в частотну таблицю.

Крок 2. Побудувати матрицю правдоподібності, розрахувавши відповідні ймовірності (апріорні).

Крок 3. За допомогою теореми Байєса розрахувати апостеріорну вірогідність для кожного класу. Клас з найбільшою апостеріорною вірогідністю буде результатом прогнозу.

Складність алгоритму – $O(nl)$, де n – кількість об’єктів, l – кількість атрибутів.

2.5.3 *Недоліки*

В даному методі можна виділити наступні недоліки [78]:

- якщо в тестовому наборі даних є деяке значення категорійної ознаки, яке не зустрічалось раніше в навчальній виборці, тоді модель присвоїть нульову ймовірність цьому значенню та не зможе виконати прогнозування;
- припущення про незалежність ознак. В реальності набори повністю незалежних ознак зустрічаються рідко.

2.6 Алгоритм k-середніх

2.6.1 *Опис ідеї*

Метод k-середніх – метод кластерного аналізу, ціллю якого є розділення m спостережень на k кластерів, при цьому кожне спостереження відноситься до того кластера, до центру якого воно найближче всього [70].

Задача визначення міри близькості об’єктів описана в розділі 2.6.2.

Так, розглянемо ряд спостережень $(x_1, x_2, \dots, x_n) \in R^n$.

Метод k-середніх розділяє n спостережень на k кластерів $(k \leq n)$

$C = \{C_1, C, \dots, C_k\}$, щоб мінімізувати сумарне квадратичне відхилення точок кластерів від центроїдів цих кластерів:

$$\min \left[\sum_{i=1}^{k} \sum_{x_j \in C_i} \left\| x_j - \mu_{\text{i}} \right\|^2 \right], \tag{2.9}$$

де $x_j \in R^n$;

$\mu_i \in R^n$;

μ_i – центроїд для кластера C_i.

2.6.2 *Визначення подібності між об’єктами*

Задля визначення подібності між двома об’єктами a та b необхідно представити ці обєкти у вигляді двох векторів x_a та x_b та обчислити косинусну подібність цих векторів [91]:

$$\cos(x_a, x_b) = \frac{x_a^T x_b}{||x_a|| x_b||}. \tag{2.9}$$

У задачі надання рекомендацій ця міра може бути використана для обчислення подібності користувачів, прийнявши користувача u як як вектор $x_u \in R^{|I|}$, де $x_{ui} = r_{ui}$, якщо користувач оцінив об'єкт i, та 0 в іншому випадку. Подібність між вибором двох користувачів u та v може бути обчислена як:

$$CV(u,v) = \cos(x_u, x_v) = \frac{\sum_{i\epsilon\ uv} r_{ui} r_{vi}}{\sqrt{\sum_{i\epsilon\ u} r_{ui}^2 \sum_{j\epsilon I_v} r_{vj}^2}}, \quad (2.10)$$

де I_{uv} – об'єкти, оцінені і u, і v.

Недолік цієї міри полягає в тому, що він не враховує дисперсії оцінок, наданих користувачами u, v.

Це вирішується впровадженням методу визначення подібності об'єктів шляхом використання кореляції Пірсона, яка визначається як:

$$PC(u,v) = \frac{\sum_{i\epsilon\ uv}(r_{ui} - \overline{r_u})(r_{vi} - \overline{r_v})}{\sqrt{\sum_{i\epsilon\ uv}(r_{ui} - \overline{r_u})^2 \sum_{i\epsilon\ uv}(r_{vi} - \overline{r_v})^2}}. \quad (2.11)$$

Відмінності в шкалах оцінки окремих користувачів часто виражені більше, ніж відмінності в рейтингах окремих об'єктів. Тому, під час обчислення подібності вподобань буде більш доцільним порівнювати рейтинги, що орієнтовані на середнє значення їх користувача, замість рейтингу об'єкта.

Скорегована косинусна оцінка (AC) – модифікація кореляції Пірсона, яка порівнює рейтинги, орієнтуючись на користувача [91]:

$$AC(i,j) = \frac{\sum_{u\epsilon\ ij}(r_{ui} - \overline{r_u})(r_{uj} - \overline{r_u})}{\sqrt{\sum_{u\epsilon\ ij}(r_{ui} - \overline{r_u})^2 \sum_{u\epsilon U_{ij}}(r_{uj} - \overline{r_u})^2}}. \quad (2.12)$$

2.6.3 *Опис алгоритму*

Алгоритм надання рекомендацій за даним методом наведено нижче [23, 13].

Якщо міра близькості до центроїда визначена, то розбиття обєктів на кластери зводиться до визначення цих кластерів. Число кластерів k визначається завчасно.

Розглянемо початковий набір k центроїдів $\mu_1, \mu_2, \dots, \mu_k$ в кластерах $C_1, C_2, \dots, C_k$.

Крок 0. Центроїди обираються випадково.

Крок 1. Для кожного елемента j визначити міру схожості до об'єкта i.

Крок 2. Обрати множину елементів S, яка є найбільш близькою до об'єкта i. Кожен об'єкт можна віднести тільки до одного кластера, навіть якщо його можна віднести до двох або більше кластерів.

Крок 3. Центроїд кожного i-го кластера переобчислюється за правилом: $\mu_j = \frac{1}{C_j}\sum_{x_j \in C_i} x_j$.

Крок 4. Якщо $\mu_i^t = \mu_i^{t+1}$ (значення μ_i не відрізняється від значення отриманого на попередньому кроці), закінчити обчислення. Інакше, перейти до кроку 1.

Складність однієї ітерації алгоритму - $O(kln)$, де l – кількість атрибутів.

2.6.4 *Недоліки*

Недоліками даного методу є [55, 14]:

- необхідність завчасного визначення кількості кластерів;
- чутливість до вибору початкових центрів кластера;
- чутливий до шумів та викидів;
- не гарантується досягнення глобального оптимуму, а тільки локального;
- класична реалізація не дозволяє роботу з категоріальними ознаками.

2.7 Алгоритм ID3

2.7.1 *Опис використовуваних метрик*

Ентропія $H(S)$ – міра невизначеності множини даних S [67]:

$$H(S) = \sum_{x \in X} -p(x) \log_2 p(x), \quad (2.13)$$

де S – поточна множина даних;

X – набір класів в S;

$p(x)$ – пропорція кількості елементів, які потрапили до класу x до кількості елементів в множині S.

В даному методі ентропія розраховується для кожного залишкового атрибуту. Атрибут з найменшою ентропією використовується для розбиття множини S.

Кількість інформації $IG(A, S)$ – міра різниці ентропії до та після розбиття множини S за атрибутом A:

$$IG(A, S) = H(S) - \sum_{t \in T} p(t) H(t), \quad (2.14)$$

де $H(S)$ – ентропія множини *S*;

T – підмножини, утворені шляхом розщеплення множини *S* за атрибутом *A* такі, що $S = \bigcup_{t \in T} t$;

$p(t)$ – пропорція кількості елементів, які містяться в t до кількості елементів в S;

$H(t)$ – ентропія підмножини t.

2.7.2 *Опис алгоритму*

Алгоритм працює з початковим набором S, прийнявши його за корінь.

Алгоритм працює рекурсивно, розбиваючи по обраній ознаці в кожному вузлі множину на підмножини, починаючи з кореня дерева. Для проведення розбиття обчислюється ентропія $H(S)$ (або кількість інформації $IG(S)$).

Крок 1. Обчислити ентропію кожного атрибуту множини *S*.

Крок 2. Розбити S на підмножини використовуючи атрибути з найменшою ентропією $H(S)$ (або найбільшою кількістю інформації $IG(S)$).

Крок 3. Побудувати дерево рішень з обраним атрибутом.

Крок 4. Рекурсивно розбити підмножини, використовуючи атрибути, що залишилося.

Складність алгоритму – $O(l * n \log_2 n)$, де l – кількість атрибутів [95].

2.7.3 *Недоліки*

Недоліками даного методу є [26]:

- жадібність. Локальний оптимум предиката *в* не являється глобально оптимальним. У випадку вибору неоптимального предиката алгоритм не здатний повернутися на рівень вверх та здійснити заміну невдалого предикату;
- чим далі вершина v розташована від дерева, тим менша довжина підвибірки S, по якій доводиться приймати рішення про гілкування в вершині v. Тим менш статистично надійним стає вибір предиката *в*;
- алгоритм здатний до перенавчання – як правило, він занадто ускладнює структуру дерева. Узагальнююча здібність алгоритму (якість класифікації нових об’єктів) відносно невелика.

Основна причина недоліків – неоптимальність жадібної стратегії нарощування дерев.

2.8 Алгоритм SVD

2.8.1 *Визначення SVD*

Сингулярний розклад (SVD - Singular Value Decomposition) є зручним методом при роботі з матрицями. Сингулярний розклад показує геометричну структуру матриці та дозволяє наглядно представити наявні дані [101].

Теорема. Для будь-якої дійсної (nxn) матриці A існують дві дійсні ортогональні матриці (nxn) U та V такі, що

$$U^T AV = \Lambda. \tag{2.15}$$

Більш того, можна обрати *U* та *V* так, щоб діагональні елементи Λ мали вигляд:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r \geq \lambda_{r+1} = \cdots = \lambda_n = 0, \tag{2.16}$$

де *r* – ранг матриці *A*.

Зокрема, якщо *A* невироджена, то

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = 0. \tag{2.17}$$

Кінець теореми.

Індекс *r* елемента λ_r – фактична розмірність власного простору матриці *A*.

Стовпці матриць *U* та *V* називають відповідно лівими та правими сингулярними векторами, а значення діагоналі матриці Λ називають сингулярними значеннями.

Нехай *A* – (mxn)-матриця та їй у відповідність поставлений лінійний оператор, який також називається *A*. Формулу сингулярного розкладу:

$A = U \Lambda V^T$ можна переформулювати в геометричних термінах. Лінійний оператор , який відображає елементи простору R^m в елементи простору R^n представимо у вигляді послідовно виконуваних лінійних операцій обертання, розтягнення та обертання. Число ненульових елементів на діагоналі матриці Λ є фактична розмірність матриці *A*. Тому компоненти сингулярного розкладу наглядно показують геометричні зміни при відображенні лінійним оператором *A* множини векторів з одного векторного простору в інший.

2.8.2 *Наївний алгоритм SVD*

Розглянемо наближений лінійний опис матриці $A=\{a_{ij}\}$ вигляду

$$a_{ij} = \sum_{k=1}^{r} u_{ik}\lambda_k v_{kj} + c_{ij}, \quad (2.18)$$

де i=1, …, m;

j = 1, …, n.

Значення $u_{ik}, \lambda_k, v_{kj}$ для даного значення k знайдені з умови мінімуму виразу

$$\varepsilon^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}^2, \quad (2.19)$$

при обмеженнях нормування

$$\sum_{j=1}^{n} u_{ik}^2 = \sum_{i=1}^{m} v_{kj}^2 = 1 \quad (2.20)$$

та впордякування

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r \geq \cdots \geq 0. \quad (2.21)$$

Вирази (2.19) – (2.21) запишемо в матричному вигляді:

$$A = U\Lambda V^T + C, \tag{2.22}$$

$$\varepsilon^2 = tr(CC^T) = \|C\|^2, \tag{2.23}$$

$$U^T U = VV^T = I, \tag{2.24}$$

де матриці $U = \{u_{kj}\}$;

$\Lambda = \text{diag}\{\lambda_k\}$;

$\text{V} = \{v_{ik}\}$.

Якщо значення *r* достатньо велике, то $C = 0$. Так буде заздалегідь при $r \geq \min\{m, n\}$. Мінімальне значення *r*, при якому виконується рівність $A = U\Lambda V^T$, рівне рангу матриці A.

Алгоритм знаходження сингулярного розкладу полягає в наступному. Знайдемо послідовно вектори u_k, v_k та сингулярні числа λ_k для *k=1,..,r*. В якості цих векторів беруться нормовані значення векторів a_k та b_k відповідно:

$$u_k = \frac{a_k}{\|a_k\|}, v_k = \frac{b_k}{\|b_k\|}. \tag{2.25}$$

Вектори a_k та b_k знаходяться як границі послідовності векторів $\{a_k\}$ та $\{b_k\}$, відповідно $a_k = \lim(a_{k_i})$ та $b_k = \lim(b_{k_i})$.

Сингулярне число λ_k знаходиться як добуток норм векторів:

$$\lambda_k = \|a_k\| \cdot \|b_k\|. \tag{2.26}$$

Процедура знаходження векторів u_k, v_k починається з вибору найбільшого за нормою рядка b_{1_1} матриці A.

Для *k=1* формули знаходження векторів a_{1_i}, b_{1_i} мають вигляд:

$$a_{1_i} = \frac{Ab_{1_i}^T}{b_{1_i}b_{1_i}^T}, \tag{2.27}$$

$$b_{1_{i+1}} = \frac{a_{1_i}^T A}{a_{1_i}^T a_{1_i}}. \tag{2.28}$$

Для обчислення векторів u_k, v_k при *k=2,…,r* використовується вищеприведена формула з тою різницею, що матриця *A* заміняється на скореговану на *k-му* кроці матрицю $A_{k+1} = A_k - u_k\lambda_k v_k$. [8]

2.8.3 *Недоліки*

До недоліків даного методу можна віднести необхідність у попередньому перетворенні даних, а також висока трудомісткість розкладу для великих об'ємів даних. SVD-розклад не єдиний [63,52].

Сингулярний розклад матриці неможливо використовувати окремо як самостійний алгоритм, а лише в цілях декомпозиції матриці.

2.9 Алгоритм Random Forest

2.9.1 *Формальна постановка задачі*

В якості вхідних даних використовуються пари об'єктів $\{(x_i, y_i)\}_{i=1}^N$, де x_i – ознаковий опис об'єкта, а y_i – мітка класу (значення з дискретної множини, потужність якої дорівнює числу класів в задачі). Вимагається побудувати композицію вирішальних дерев, які мають наступні властивості [28]:

- вирішальна композиція з високою точністю класифікує нові об'єкти z_i, мітка класу яких невідома;
- за допомогою композиції отримано достатньо стійке рішення поставленої задачі, тобто при невеликій зміні параметрів побудованого рішення не відбувається серйозного зниження якості роботи.

2.9.2 *Опис алгоритму*

Нехай маємо вибірку X розміру N.

Алгоритм Random Forest полягає в наступному [66]:

Крок 1. Для кожного $n = 1, \dots, N$:

Крок 1.1. Згенерувати вибірку X_n наступним чином:

Крок 1.1.1. Повторити M разів, згенерувавши M підвибірок $X_1, \dots, X_M$:

Крок 1.1.1.1. Рівномірно взяти з вибірки N об'єктів з поверненням. Тобто, вибрати N разів випадковий об'єкт вибірки(вважаємо, що кожен елемент "дістається" з однаковою ймовірністю $\frac{1}{N}$), причому кожен раз обирається з усіх початкових N об'єктів. Позначити цю вибірку як X_i.

Крок 2.1. Побудувати вирішальне дерево b_n по вибірці X_n:

- по заданому критерію обрати кращу ознаку, виконати розбиття в дереві по ній і так для всієї вибірки;

- дерево будується, поки в кожному листку не більше, ніж n_{min} об'єктів або поки не досягнемо необхідної висоти;

- при кожному розбитті спочатку обрати m випадкових ознак з n початкових; здійснювати пошук оптимального розбиття вибірки тільки через них.

Кінцевий класифікатор:

$$a(x) = \frac{1}{N}\sum_{i=1}^{N} b_i(x). \qquad (2.29)$$

Простими словами – для задачі класифікації обрати рішення голосуванням по більшості.

Для задач класифікації рекомендується брати $m = \sqrt{n}$, а кожне дерево будувати до тих пір, поки в кожному листі не залишиться по одному об'єкту.

Складність алгоритму для побудови одного дерева – $O(mn \log_2 n)$, де m – кількість атрибутів, що мають бути прийняті для побудови дерева.

Складність алгоритму для побудови лісу дерев – $O(kmn \log_2 n)$, де k – кількість дерев.

2.9.3 *Недоліки*

Недоліками даного методу є [89, 72]:

- великий розмір отримуваних моделей. Необхідно O(K) пам'яті для зберігання моделі;
- якість отриманої композиції значно погіршується, якщо на вхід алгоритму подається мало розмічених даних, оскільки дерева не можуть якісно виявити приховані в даних закономірності;
- по мірі наближення до листків розбиття в вузлах стає все менше статистично обґрунтованим через малу кількість об'єктів, що розглядаються;
- в процесі збільшення числа дерев, що використовуються, зменшується відхилення, але підвищується дисперсія. При цьому також зростає гнучкість моделі, що дозволяє їй налаштуватися на викиди в даних та сприяє зниженню узагальнюючої здібності фінальної композиції на стадії тестування;
- практично не підтримується робота з категоріальними ознаками (як представленими у вигляді рядків, так і у вигляді цілих чисел). У випадку з представленням категоріальної ознаки у вигляді цілих чисел, рівні 1 та 2 вважатимуться більш близькими ніж 1 та 10, що може протирічити логікі;
- повільно обробляються пропущені значення в тренувальній виборці та ще гірше в тестовій:
    - в навчальній вибірці пропуски в ознаках числового типу заповнюється медіаною (більш стабільною оцінкою в порівнянні з середнім значенням), в категоріальних – модою;
    - в начальній виборці проходять запуски алгоритму з різними варіантами заповнення та визначається найкращий кандидат-заміна для пропуску.

2.10 Висновки до розділу

В цьому розділі був сформований формальний опис рекомендаційних алгоритмів кластеризації та класифікації даних.

Для кожного алгоритму було наведено опис ідеї методу та основні кроки його реалізації, а також було виділено їх недоліки.

В результаті аналізу всіх наведених алгоритмів було побудовано діаграму Ісікави, що вказує на основні недоліки розглянутих методологій. Діаграма Ісікави наведена на рисунку 2.2.

Рисунок 2.2 – Основні недоліки розглянутих алгоритмів кластеризації та класифікації

Основними проблемами, що необхідно вирішити є:

- робота з категоріальними ознаками;
- висока різноманітність даних;
- поява локальних оптимумів через випадкову ініціалізацію початкових центроїдів;
- необхідність завчасного вибору кількості кластерів;
- проблема холодного старту;
- зашумленість даних та викиди.

3 РОЗРОБКА МЕТОДОЛОГІЇ РОЗВ'ЯЗАННЯ ЗАДАЧ З НАДАННЯ РЕКОМЕНДАЦІЙ

3.1 Запропонований метод

3.1.1 *Означення*

Об'єкти – елементарна група даних, з якими оперує алгоритм кластеризації. Кожному об'єкту відповідає вектор атрибутів:

$$x_i = \{x_{i_1}, \dots, x_{i_l}\}. \tag{3.1}$$

Кількість атрибутів визначають розмірність простору атрибутів.

Відстань – $d(x_i, x_j)$ між об'єктами x_i та x_j – результат застосування обраної метрики.

Медоїд – об'єкт множини даних або кластера, для якого середня відстань до інших об'єктів мінімальна (тобто найближча до центру кластера точка) [24]:

$$x^m = arg \min_{X \in \{x_1, \dots, x_n\}} \sum_{i=1}^{n} d(X, x_i). \tag{3.2}$$

Ентропія $H(X)$ – міра невизначеності множини даних X:

$$H(X) = \sum_{k \in K} -p(k)\, log_2\, p(k), \tag{3.3}$$

де X – поточна множина об'єктів, $X = \{x_i\}$;

K – набір класів в X;

$p(k)$ – пропорція кількості елементів, які потрапили до класу K, до кількості елементів в множині X.

Кількість інформації $IG(a, X)$ – міра різниці ентропії до та після розбиття множини X за атрибутом a:

$$IG(a, X) = H(X) - \sum_{t \in T} p(t) H(t), \tag{3.4}$$

де $H(X)$ – ентропія множини X;

T – підмножини, утворені шляхом розщеплення множини X за атрибутом a такі, що $X = \bigcup_{t \in T} t$;

$p(t)$ – відношення кількості елементів, що містяться в t до кількості елементів в X;

$H(t)$ – ентропія підмножини t.

3.1.2 *Математична постановка задачі*

Постановка задачі кластеризації пов'язана з визначенням метричного простору [22] (X, d), де $d: X \times X \to R$ – метрика, X – множина об'єктів кластеризації така, що $X = \{X_i\}, i = \overline{1,k}$.

Кластеризація об'єктів множини X представляє собою відображення вигляду:

$$f: X_i \to c_i, i = \overline{1,k}, \tag{3.5}$$

де $C = \{c_i\}$;

k– кількість кластерів.

Для кожного кластера c_i з множиною елементів $X_i \subseteq X$ можна визначити медоїд x_i^m, і для нього виконується:

$$\varphi_x(c_i) = \sum_{x \in X} d\left(x_{ij}, x_i^m\right), \tag{3.6}$$

$$1 \le i \le k, 1 \le j \le p_i,$$

де $d(x_{ij}, x_i^m)$ – відстань від точки x_{ij} до медоїда x_i^m;

p_i – потужність множини X_i.

При цьому повинна виконуватись умова:

$$0 \le d(x, x_i^m) \le l, \tag{3.7}$$

де l – кількість атрибутів, що відповідає x.

Необхідно знайти такі $x_i^m, i = \overline{1,k}$, що

$$\sum_{i=1}^{k} d(x_{ij}, x_i^m) \to min. \tag{3.8}$$

3.1.3 *Попередня обробка вибірки*

Одна з ідей методу полягає у модифікації вибірки до вигляду, який дозволить вилучити та проігнорувати слабковпливові параметри. Це дозволить зменшити час обробки даних, а також підвищити точність класифікації. Приклад наявних слабковпливових атрибутів в вибірці даних наведено на рисунку 3.1.

	1		a			l
x_1			β			
			γ			
x_i			δ			
			ε			
x_n			θ			

Рисунок 3.1 – Приклад різних рівноймовірних значень атрибуту у вхідній вибірці даних

Задля зменшення часу обробки даних, а також підвищення точності класифікації, вибірка підлягає модифікації до вигляду, який дозволить вилучити та проігнорувати слабковпливові параметри.

В даному методі пропонується вилучати параметри, що задовольняють наступному виразу [65, 69]:

$$0.983\, log_2 n \leq H(a) \leq log_2 n \qquad (3.9)$$

Тобто видаляються параметри, що наповнені різними рівноймовірними даними та слабко впливають на результат надання рекомендацій.

В розділі 5.4 доведено, що для вибірок, об’єкти яких містять не менше 7 атрибутів, видалення атрибута, що задовольняє нерівності (3.9) не призводить до погіршень точності класифікації.

3.1.4 *Метрики визначення відстаней*

Вибірки з категоріальними даними є проблемою для багатьох алгоритмів машинного навчання: для роботи алгоритму, для кожної пари об’єктів необхідно виміряти відстань між ними – степінь схожості.

Класичні метрики, такі як відстань Мінковського, Манхетенська відстань, Евклідова відстань [54, 93] недоречні для використання у подібній задачі, так як некоректно відображають степінь близькості об’єктів. Для вирішення цієї проблеми пропонується використовувати метрику Хемінга [60].

За нею відстань між двома об’єктами x_i та x_j, які містять l категоріальних атрибутів визначається так:

$$d(x_i, x_j) = \sum_{a=1}^{l} \delta(x_{i_a}, x_{j_a}); \tag{3.10}$$

$$\delta(x_i, x_j) = \begin{cases} 0, x_{i_a} \neq x_{j_a} \\ 1, x_{i_a} = x_{j_a} \end{cases}, \tag{3.11}$$

де x_{i_a}, x_{j_a}– значення атрибута (категорії) a в об’єктах x_i та x_j відповідно.

Чим більша кількість невідповідностей категоріальних значень між x_i та x_j, тим більше відрізняються два об’єкти.

Схема визначення відстані між об’єктами наведена на рисунку 3.2.

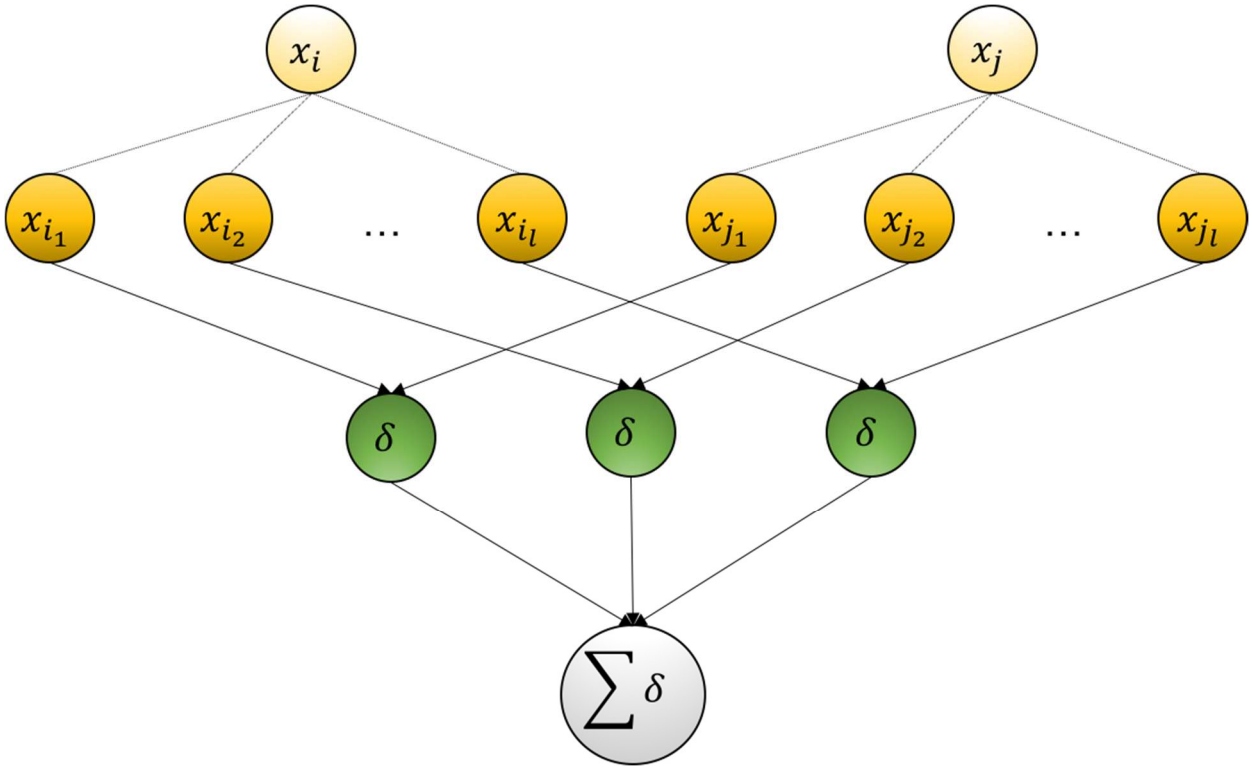

Рисунок 3.2 – Схема визначення відстані між об’єктами

3.1.5 *Визначення початкової кількості кластерів*

Задля визначення оптимальної кількості кластерів для розбиття вхідної вибірки пропонується використовувати:

- метод ліктя [87];
- метод спрощеної оцінки силуету [81];
- їх комбінацію.

Використовуючи метод ліктя необхідно побудувати функцію, що відображає зміну цільової функції (3.8) внутрішньокластериних варіацій даних в залежності від кількості кластерів. Число кластерів є найбільш підходящим (оптимальним) для задачі в тій позиції, де відмічена найбільша зміна цієї функції – в лікті графіка.

Опис методу спрощеної оцінки силуету наведений у розділі 5.3.1.

Використання метода ліктя потребує додаткового аналізу даних користувачем, що знижує поняття “автоматизації” процесу кластеризації, а метод спрощеної оцінки силуету змушує виконувати додаткові обчислення. Також результати, сформовані за використання метода ліктя не завжди відповідають найвищій оцінці якості розбиття.

На рисунку 3.3 наведено графік функції, побудованої за методом ліктя для визначення кількості кластерів. Використовуючи тільки його, важко визначити оптимальну для даної вибірки кількість кластерів.

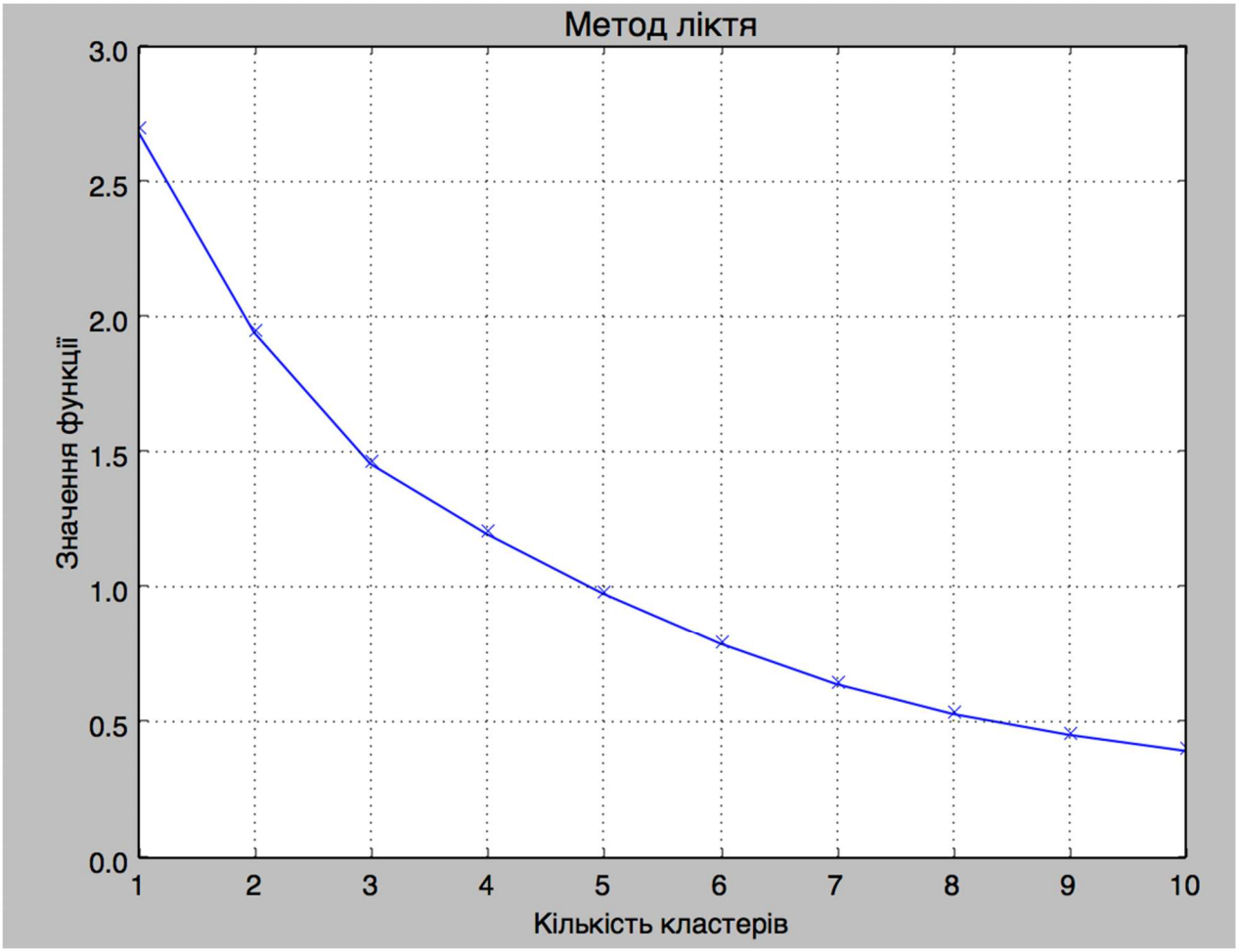

Рисунок 3.3 – Визначення кількості кластерів за методом ліктя

Тому, пропонується поєднувати ці два методи для більш точного та швидшого визначення кількості кластерів – спочатку обирати ймовірні оптимальні варіанти розбиття (кількість кластерів) за допомогою метода ліктя, а потім обирати з цих варіантів найбільш підходящий за методом спрощеної оцінки силуету.

Оптимальною кількістю кластерів для запропонованого варіанту вибірки даних буде 4.

3.1.6 *Ініціалізація початкових медоїдів*

Ідея алгоритму ініціалізації початкових медоїдів полягає в тому, щоб обрати нові об’єкти, які знаходяться якнайдалі від існуючих медоїдів.

Алгоритм ініціалізації початкових точок:

Крок 1. Обрати перший медоїд x_1^m випадковим чином з X;

Крок 2. для кожного $j: j = 1, \dots, k$ виконати:

Крок 2.1. для всіх x_i:

Крок 2.1.1. якщо $j = 1$: обрахувати $d_i = d(x_1, x_1^m)$;

Крок 2.1.2. інакше: обрахувати $d_i = \min\left(d(x_1, x_1^m), \dots, d\left(x_i, x_j^m\right)\right)$;

Крок 2.2. обрати такий наступний медоїд $x_j^m = x_i: \max(d_i)$.

Чим більша величина d_i, тим далі знаходиться i-й елемент від усіх існуючих медоїдів, що зменшує кількість ітерацій для перевизначення кластерів та допоможе уникнути проблеми виникнення локальних мінімумів при повністю випадковому виборі початкових медоїдів, як пропонується в класичних методах.

На рисунку 3.4 наведено схематичний опис кроку ініціалізації початкових медоїдів.

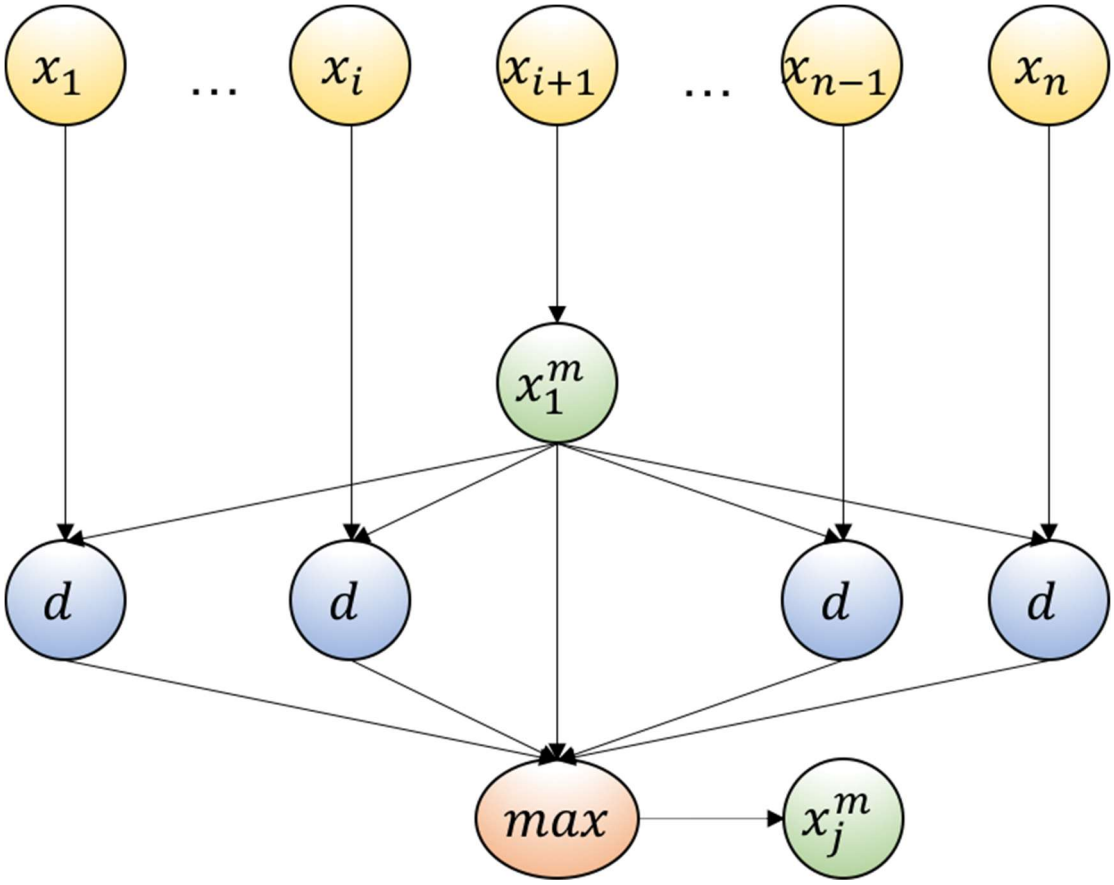

Рисунок 3.4 – Схема кроку ініціалізації медоїдів

На рисунку 3.5. наведена легенда схеми рисунка 3.4..

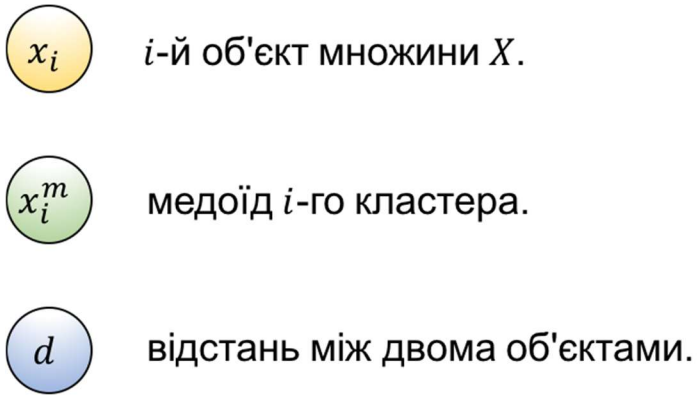

Рисунок 3.5 – Легенда схеми

3.1.7 *Визначення кластерів*

Задля розподілення об’єктів в кластери, необхідно скористатися метрикою Хемінга, описаною в п.3.1.5.

Алгоритм розподілення об’єктів в кластери:

Крок 1. для всіх x_i:

Крок 1.1. приписати $x_i \in c_j \Leftrightarrow j = \arg\min_k d(x_j, x_k^m)$.

Схема розподілення об’єктів по кластерам наведена на рисунку 3.6.

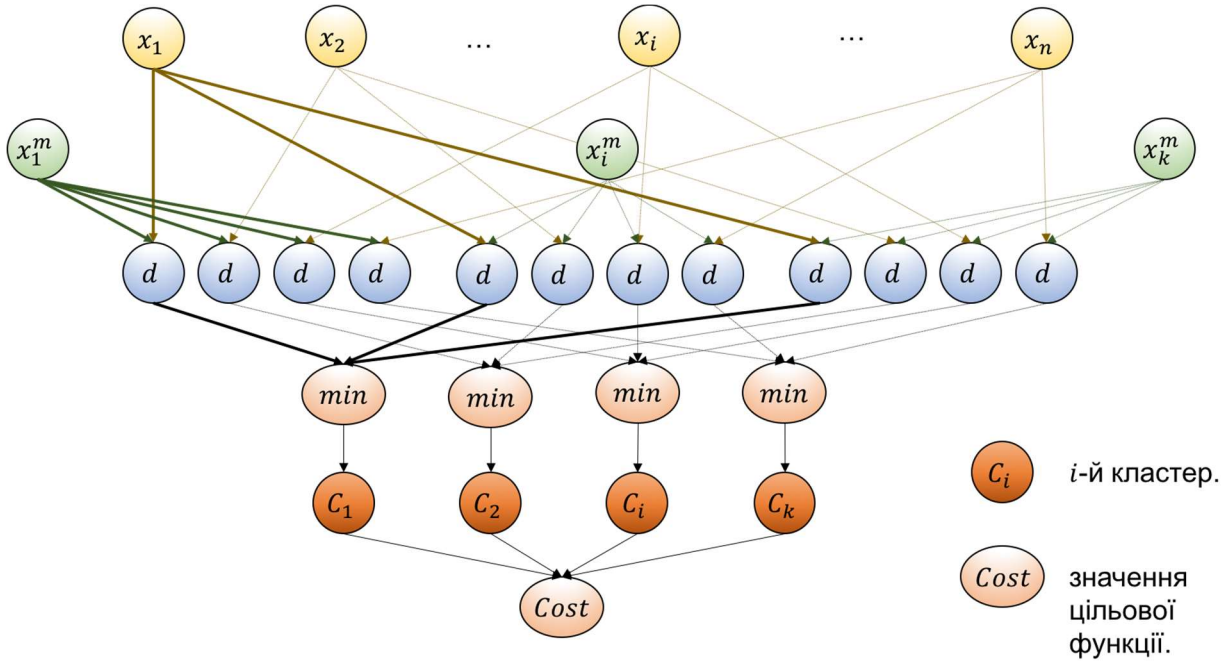

Рисунок 3.6 – Схема розподілення об’єктів по кластерам

3.1.8 *Модифікований алгоритм*

Блок-схема модифікованого алгоритму [21] наведена на рисунку 3.7.

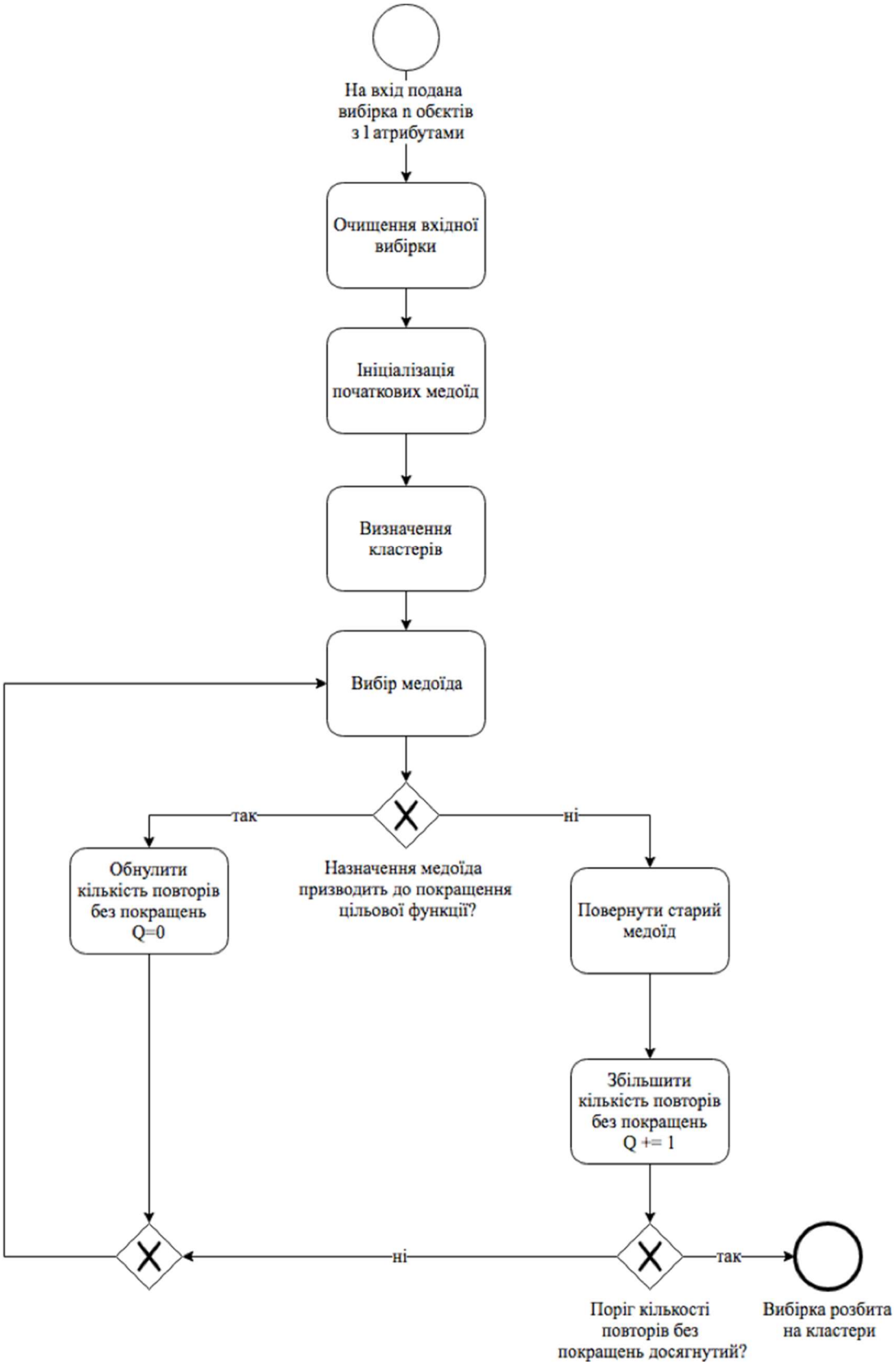

Рисунок 3.7 – Схема модифікованого алгоритму

Модифікований алгоритм полягає в наступному:

1) очистити вхідну вибірку (описано в розділі 3.1.3);
2) ініціалізувати початкові медоїди (описано в розділі 3.1.6);
3) визначити кластери (описано в розділі 3.1.7);
4) для кожного кластера c_j:
   - 4.1) визначити випадкову точку x_i в кластері $c_j: x_i \neq c_j$ як новий медоїд $x_i'^m$;
   - 4.2) перевизначити кластери відповідно до нового медоїда (описано в розділі 3.1.7);
   - 4.3) якщо $\sum_{i=1}^{n}\sum_{j=1}^{p_i} d(x_{ij}, x_i^m) < \sum_{i=1}^{n}\sum_{j=1}^{p_i} d(x_{ij}, x_i'^m)$ – повернути останній медоїд та перевизначити кластери відповідно до поверненого медоїда (описано в розділі 3.1.7);
   - 4.4) якщо результат не покращувався вже Q кроків – закінчити розгляд поточного кластера c_i.

Параметр Q задається вручну, експериментальним шляхом визначено, що при значенні $Q = \lfloor 2{,}6k \rfloor$ досягається оптимальний результат. Обґрунтування значення Q наведено в розділі 5.5.

Схема перевизначення медоїдів в кластері наведена на рисунку 3.8.

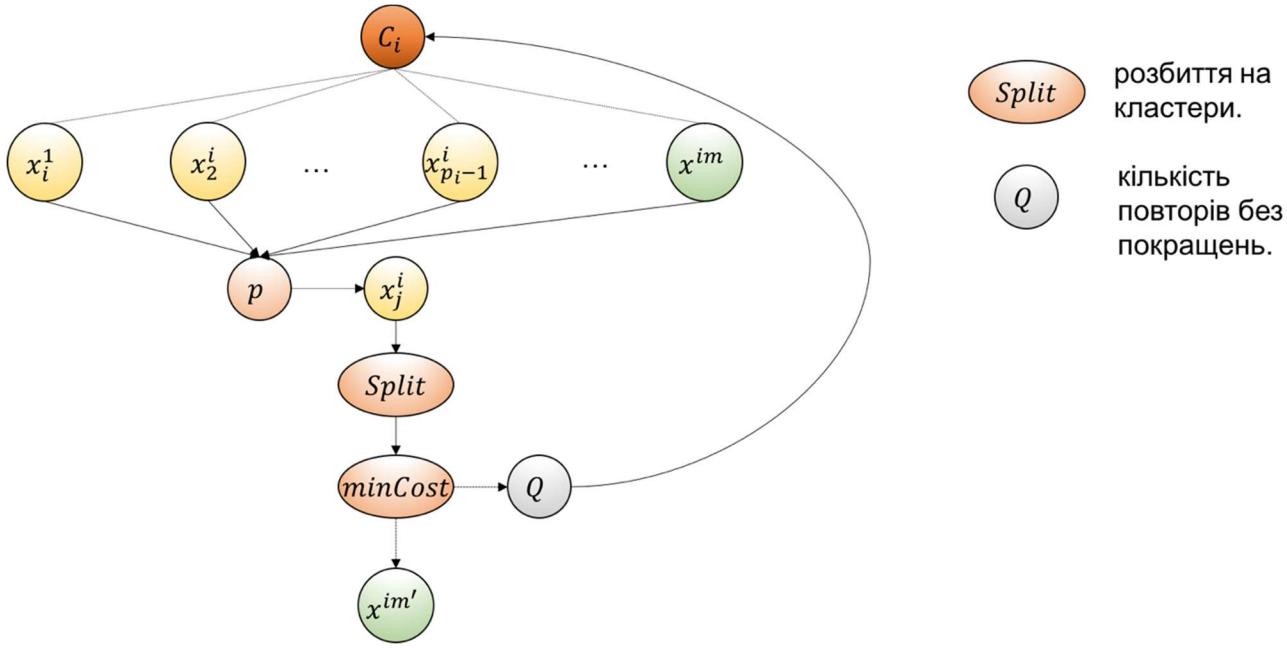

Рисунок 3.8 – Схема перевизначення медоїдів в кластері

Складність однієї ітерації модифікованого алгоритму – $O(kln)$.

Складність однієї ітерації алгоритму, що перебирає всі вершини для перевизначення нового медоїду – $O(kln^2)$.

3.2 Визначення домінантних кольорів на фотографії

3.2.1 *Опис методу визначення домінантних кольорів на фотографії*

Задля вирішення питання визначення домінантних кольорів на фотографії використовано модифікований метод кластеризації, що описаний в розділі 3.1, з використанням в якості метрики Евклідової відстані:

$$d(x_i, x_j) = \sqrt{\sum_{a=1}^{l}(x_{i_a}, x_{j_a})}, \quad (3.12)$$

де x_i, x_j – об’єкти, між якими проводиться пошук відстані;

l – кількість атрибутів;

x_{i_a}, x_{j_a} – a-й атрибут x_i, x_j об’єктів відповідно.

3.2.2 *Класифікація кольору*

RGB (абревіатура англійських слів “red”, “green”, “blue” - червоний, зелений, синій) – адитивна кольорова модель, яка описує спосіб кодування кольору.

Адитивною вона називається через те, що кольори отримуються шляхом додавання (від англ. “addition”) до чорного кольору.

Саму модель можна представити у вигляді тривимірного кубу. Кожна з координат представляються у вигляді одного октету, значення якого для зручності позначаються числами від 0 до 255 включно, до 0 – мінімальна, а 255 – максимальна інтенсивність.

Так як модифікований алгоритм призначений для роботи з категоріальними атрибутами, необхідно провести класифікацію визначених центрів кластерів (домінантних кольорів) до загальної категоріальної структури.

Таким чином, за допомогою моделі RGB можна класифікувати кольори, визначені розділі 3.2.1, обраховуючи мінімальну відстань від заданої точки до

координат, вказаних в таблиці 3.1. Найближча координата і буде визначати клас поточної точки.

Таблиця 3.1 – Координати класів кольорів

Колір	Координати (RGB)
Чорний	(0;0;0)
Червоний	(255;0;0)
Зелений	(0;255;0)
Синій	(0;0;255)
Жовтий	(255;255;0)
Фуксія	(255;0;255)
Морська хвиля	(0;255;255)
Білий	(255;255;255)

3.3 Визначення кількості обличь на фотографії

Для розпізнавання присутності обличчя на фотографії використовується метод Віоли-Джонса. Якщо шаблони відповідають конкретним областям на зображенні, вважається, що обличчя виявлене.

Якщо на фотографії присутні декілька обличь, то застосувавши шаблони напряму, їх не буде виявлено. Для пошуку образів невеликих розмірів на фотографії використовується метод ковзаючого вікна. В середині цього вікна знаходяться каскади Хаара. Після кожного проходження вікно збільшується, щоб знайти обличчя більшого масштабу.

3.3.1 *Інтегральне представлення зображення*

Інтегральне представлення зображення – це матриця, яка по розмірам співпадає з вихідним зображенням [75]. В кожному елементі зберігається сума інтенсивностей всіх елементів, які знаходяться вище або лівіше від x, y.

Елементи матриці зображення розраховуються наступним чином:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \tag{3.13}$$

де $ii(x, y)$ – інтегральний образ;

$i(x, y)$ – оригінальне зображення, приклад якого наведено на рисунку 3.9.

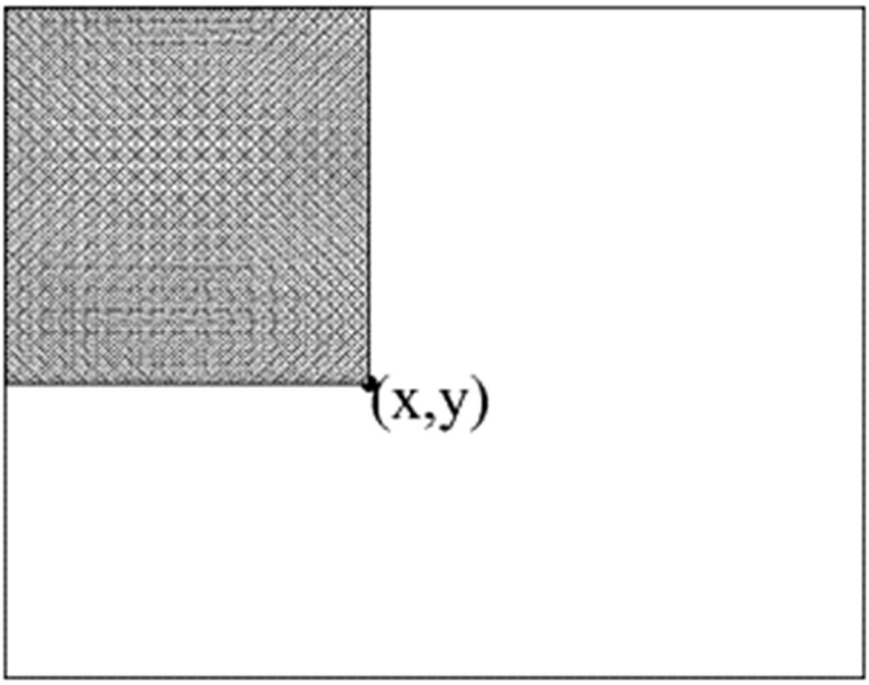

Рисунок 3.9 – Значення інтегрального зображення в точці (x,y) – сума всіх пікселів, що знаходяться вище та зліва

Кожен елемент матриці L(x,y) представляє суму пікселів в прямокутнику від (0,0) до (x,y). Тобто значення кожного пікселя (x,y) дорівнює сумі значень всіх пікселів, які знаходяться вище або лівіше від даного пікселя (x,y). Розрахунок матриці займає лінійний час, пропорційний числу пікселів в зображенні, тому інтегральне зображення вираховується за один прогін.

Розрахунок матриці можливо проводити за наступними формулами:

$$s(x, y) = s(x, y - 1) + i(x, y), \tag{3.14}$$

$$ii(x,y) = ii(x-1,y) + s(x,y), \tag{3.15}$$

де $s(x,y)$ – кумулятивна сума рядка;

$s(x,-1) = 0$;

$ii(-1,y) = 0$

За допомогою такої інтегральної матриці можна визначити суму пікселів довільного прямокутника довільної площі.

Нехай в прямокутнику ABCD, що зображений на рисунку 3.10 нас цікавить об’єкт D.

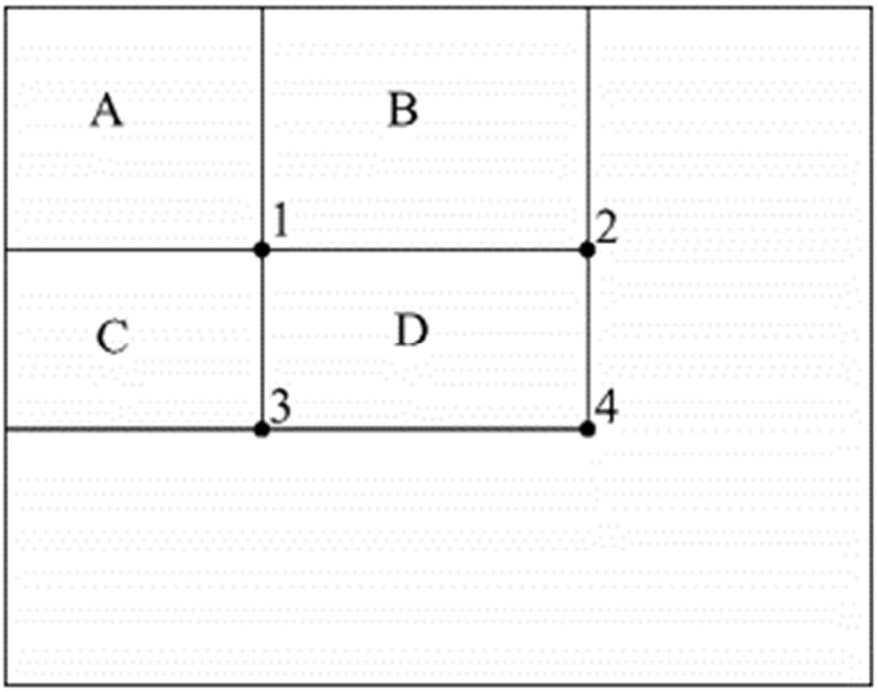

Рисунок 3.10 – Обчислення суми пікселів

Суму пікселів прямокутника D можна визначити чотирма обчисленнями масиву. Значення інтегрального зображення в точці 1 – сума пікселів в прямокутнику A. Значення в точці 2 – (A+B), точці 3 – (A+C), в точці 4 – (A+B+C+D). Сума в D може бути обчислена як 4+1-(2+3).

3.3.2 *Ознаки Хаара*

Ознака – відображення f: X=>D_f, де D_f – множина допустимих значень ознаки. Якщо задані ознаки f_1, …, f_n, то вектор ознак x = ($f_1(x_1)$, …, $f_n(x)$) називається ознаковим описом об'єкта $x \in X$. Ознакові описи допустимо ототожнювати з самими об'єктами. При цьому, множина X = D_{f1}* …* D_{fn} називається ознаковим простором.

Ознаки діляться на наступні типи в залежності від множини D_f:

- бінарна ознака: $D_f = \{0,1\}$;
- номінальна ознака: D_f – кінцева множина;
- порядкова ознака: D_f – кінцева впорядкована множина;
- кількісна ознака: D_f – множина дійсних чисел.

В стандартному методі Віоли-Джонса використовуються прямокутні ознаки – примітиви Хаара, приклад яких зображено на рисунку 3.11.

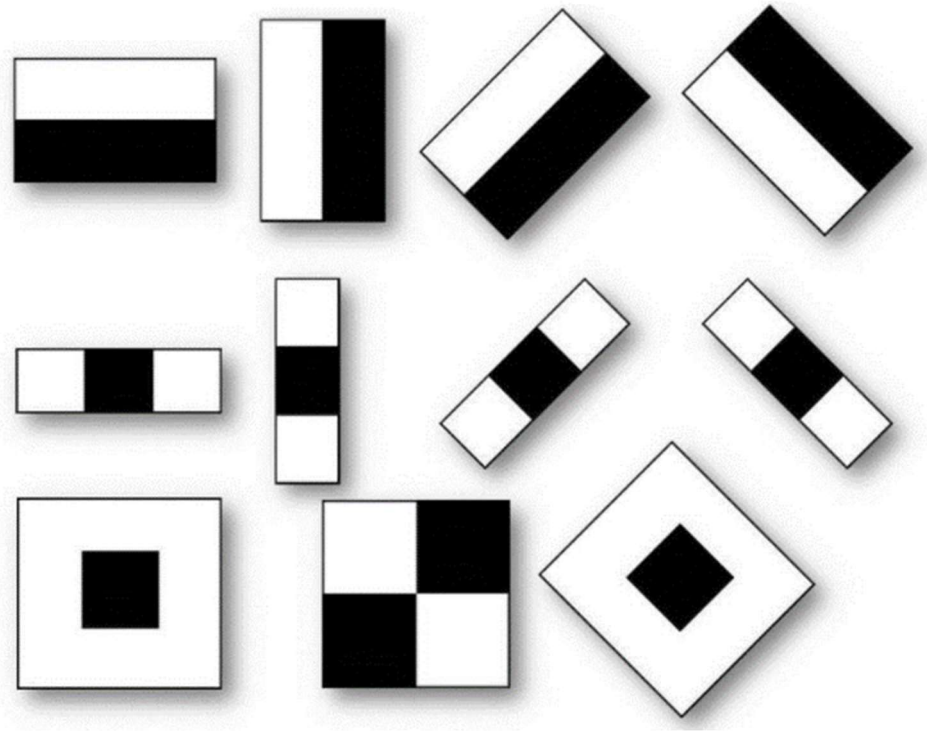

Рисунок 3.11 – Примітиви Хаара

Обчислювальним значенням такої ознаки буде:

$$F = X - Y, \tag{3.5}$$

де X – сума значень яскравості точок, які закриваються світлою частиною ознаки;

Y – сума значень яскравості точок, які закриваються темною частиною ознаки.

Для їх обчислення використовується поняття інтегрального зображення, яке розглянуте в розділі 3.3.1.

Ознаки Хаара дають точкове значення перепаду яскравості по вісі X та Y відповідно [25].

3.3.3 *Алгоритм*

Алгоритм сканування вікна з ознаками виглядає наступним чином:

Крок 1. Обрати вікно сканування у вибраному зображенні, обрати ознаки, які будуть використовуватися.

Крок 2. Вікно сканування послідовно рухати по зображенню з кроком в 1 піксель.

Крок 3. При скануванні зображення, в кожному вікні обчислюється близько 200 000 варіантів розташування ознак за рахунок зміни масштабу ознак та їх положення в вікні сканування (для зображення розмірами 24x24 пікселі).

Крок 4. Сканування відбувається послідовно для різних масштабів. (Масштабується не зображення, а вікно сканування).

Крок 5. Всі знайдені ознаки потрапляють в класифікатор, який виносить висновок.

3.4 Висновки до розділу

В даному розділі було сформульовано математичну постановку задачі та представлено модифікований метод кластеризації даних, що дозволяє обробляти категоріальні атрибути.

Для побудови модифікованого алгоритму було вирішено такі задачі:

- обрано та обґрунтовано метрику для роботи з категоріальними змінними;
- запропоновано метод попередньої обробки вхідної вибірки даних;
- запропоновано метод ініціалізації початкових медоїд;

- запропоновано метод перевизначення кластерів;
- запропоновано метод визначення нових медоїд;
- описано методологію визначення домінантних кольорів на фотографії;
- описано методологію розпізнавання обличь на фотографії.

4 ОПИС ПРОГРАМНОГО ПРОДУКТУ

4.1 Засоби розробки

При створенні програмного продукту були використані: мова програмування Python та система керування базами даних PostgreSQL.

Python – інтерпретована об’єктно-орієнтована мова програмування високого рівня з строгою динамічною типізацією [88].

Однією з переваг Python є її багатоплатформеність та масштабованість, тобто, вона працює на багатьох платформах. Крім того, вона має гармонійну архітектуру мови та має ряд достоїнств, а саме [30,88,46]:

- інтерпретованість, що дозволяє спростити відладку програм;
- вбудовані структури даних;
- простий та зручний синтаксис;
- велика кількість бібліотек;
- високий рівень документованості;
- динамічна типізація – тип змінної визначається під час її виконання;
- підтримка процедурного, функціонального та об’єктного стилів програмування;
- кросплатформеність – працює на великій кількості операційних систем Linux ,Unix, OS X, Windows та інші;
- стандартний дистрибутив має велику кількість корисних модулів;
- зручний для розв’язання математичних проблем;
- відкритий код;
- підтримка модульності – можливість використання власних модулів в інших програмах;
- автоматичне керування пам’яттю – немає потреби хвилюватися про розподіл або звільнення пам’яті;

– типи зв'язані з об'єктами, а не зі змінними; це означає, що змінній може бути призначене значення будь-якого типу, і що, наприклад, масив може містити об'єкти різних типів; традиційні мови не надають такої можливості.

Python розширювана мова: знання С дозволяє додавати нові функції, що вбудовуються, або модулі для виконання критичних операцій з максимальною швидкістю або написання інтерфейсу до комерційних бібліотек, доступним тільки у двійковій формі. Інтерпретатор мови Python може бути вбудований у програму, написану на С, і використовувати його як розширення або командну мову для цієї програми. Python використовується для розробки програм і дозволяє провести розробку набагато швидше, ніж традиційні мови типу С, С++ або Java.

Ці та інші особливості Python роблять розгортання застосувань швидким. Один недолік Python, у порівнянні з найбільш традиційними мовами, полягає в тому, що це не цілком компільована мова; замість цього, вона частково транслює програму до внутрішньої форми байт-коду, і цей байт-код виконується інтерпретатором Python. Однак, у перспективі – сучасні комп'ютери мають так багато невикористовуваного обчислювального потенціалу, що для 90% застосувань швидкодія зв'язана з вибором мови. Java теж компілюється в байт-код, але в даний час працює повільніше ніж Python у більшості випадків. Крім того, дуже просто об'єднати Python з модулями, написаними на С або С++, які можна використовувати, щоб збільшити швидкість роботи програм на критичних ділянках [88,46].

PostgreSQL – вільно розповсюджувана об’єктно-реляційна система управління базами даних (далі - СКБД). PostgreSQL є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).

Сервер PostgreSQL написаний на мові С. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп'ютері і скопіювати в деякий каталог [83,84].

4.2 Вхідні дані

Вхідними даними є:

а) інформація про користувача;

1) ім’я;
2) стать;
3) вік;
4) місто проживання;
5) використовувані операційні системи для мобільних додатків;
6) попередні місця роботи - компанії;
7) кількість місяців, відпрацьованих в компанії;
8) мови, якими володіє користувач;
9) фотографія профіля;
10) статус відносин;
11) релігія;
12) фотографії (з подальшою їх обробкою);
    - домінуючі кольори на зображенні;
    - кількість осіб, які зображені на знімках;

б) інформація про вакансії;

1) компанія;
2) посада;
3) посилання на зовнішній ресурс.

4.3 Вихідні дані

Вихідними даними є перелік позицій та компаній, що рекомендовані конкретному користувачу.

4.4 Варіанти використання

В сервісі виділяються такі ролі:

- відвідувач;
- адміністратор.

На рисунку 4.1 зображена діаграма варіантів використання сервісу з пошуку роботи.

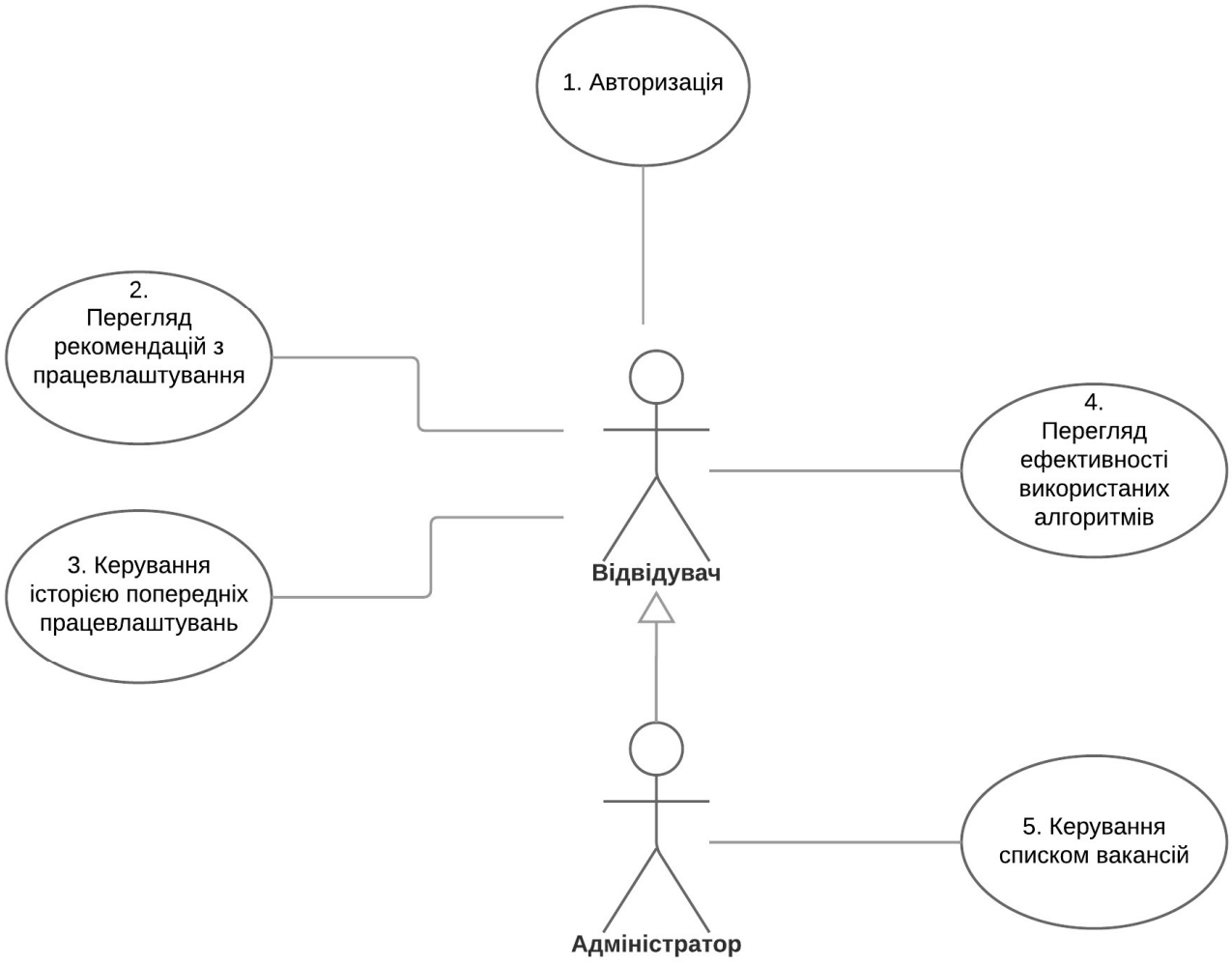

Рисунок 4.1 – Діаграма варіантів використання сервісу з пошуку роботи

Опис варіантів використання сервісу з пошуку роботи наведений в таблиці 4.1.

Таблиця 4.1 – Опис варіантів використання

№	Варіант використання	Опис
1	Авторизація	Користувач має змогу ввійти в сервіс за допомогою проходження процедури авторизації через Facebook.
2	Перегляд рекомендацій з працевлаштування	Користувач має змогу отримати рекомендації з найбільш підходящих місць роботи, обравши напрямок (роль в команді), який його цікавить.
3	Керування історією попередніх працевлаштувань	Користувач має змогу вводити попередні місця роботи (компанії) та кількість місяців, що пропацювали там, а також деактивувати неправильні записи.
4	Перегляд ефективності використаних алгоритмів	Користувач має змогу переглянути порівняльну ефективність алгоритмів кластеризації та класифікації даних, що висвітлені в дисертації.
5	Керування списком вакансій	Користувач має можливість створювати нові вакансії, вказуючи компанію, позицію та посилання на корисну інформацію про місце роботи, а також деактивувати створені записи.

4.5 Опис бізнес-процесу надання рекомендацій щодо пошуку роботи

На рисунку 4.2 наведено бізнес-процес надання користувачу рекомендацій щодо пошуку роботи, його опис наведено в таблиці 4.1.

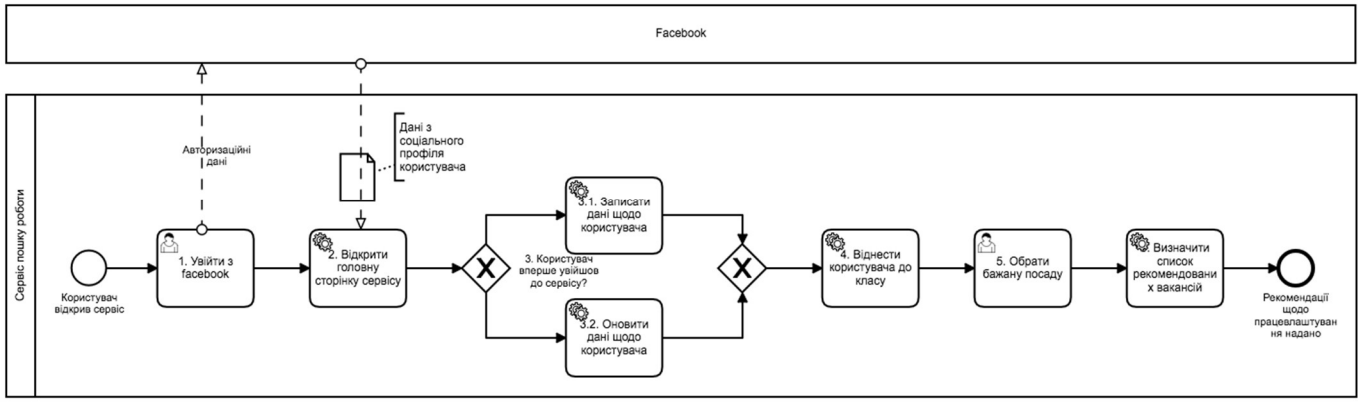

Рисунок 4.2 – Бізнес-процес надання рекомендацій з пошуку роботи

Таблиця 4.2 – Опис бізнес-процесу надання рекомендацій з пошуку роботи

№	Дія	Опис
1.	Увійти з Facebook	Користувач вводить дані свого облікового запису до форми авторизації. Сервіс надсилає ці дані до Facebook.
2.	Відкрити головну сторінку сервісу	Facebook надсилає в сервіс дані соціального профіля користувача, який пройшов авторизацію.
3.	Користувач вперше увійшов до сервісу?	Якщо виявлено, що користувач вперше користується сервісом, перейти до кроку 3.1. Інакше, перейти до кроку 3.2.
3.1.	Записати дані щодо користувача	Сервіс записує дані щодо користувача до бази даних. Перейти до кроку 4.
3.2.	Оновити дані щодо користувача	Сервіс оновлює дані щодо користувача в базі даних. Перейти до кроку 4.

№	Дія	Опис
4.	Віднести користувача до класу	Сервіс відносить користувача до одного з класів, визначених попередньою кластеризацією. Класифікація відбувається шляхом віднесення користувача до класу, відстань до медоїду якого від характеристик поточного користувача є мінімальною.
5.	Обрати бажану посаду	Користувач обирає бажану для пошуку посаду з випадаючого списку.
6.	Визначити список рекомендованих вакансій	Сервіс визначає найбільш підходящі вакансії для користувача шляхом виділення найбільш поширених компаній для класу, до якого відноситься поточний користувач.

У випадку, коли в сервісі присутньо більше 15% оновлених даних, проводиться перекластеризація користувачів для збереження актуальності рекомендацій.

4.6 Опис бази даних

Схема бази даних зображена на рисунку 4.3.

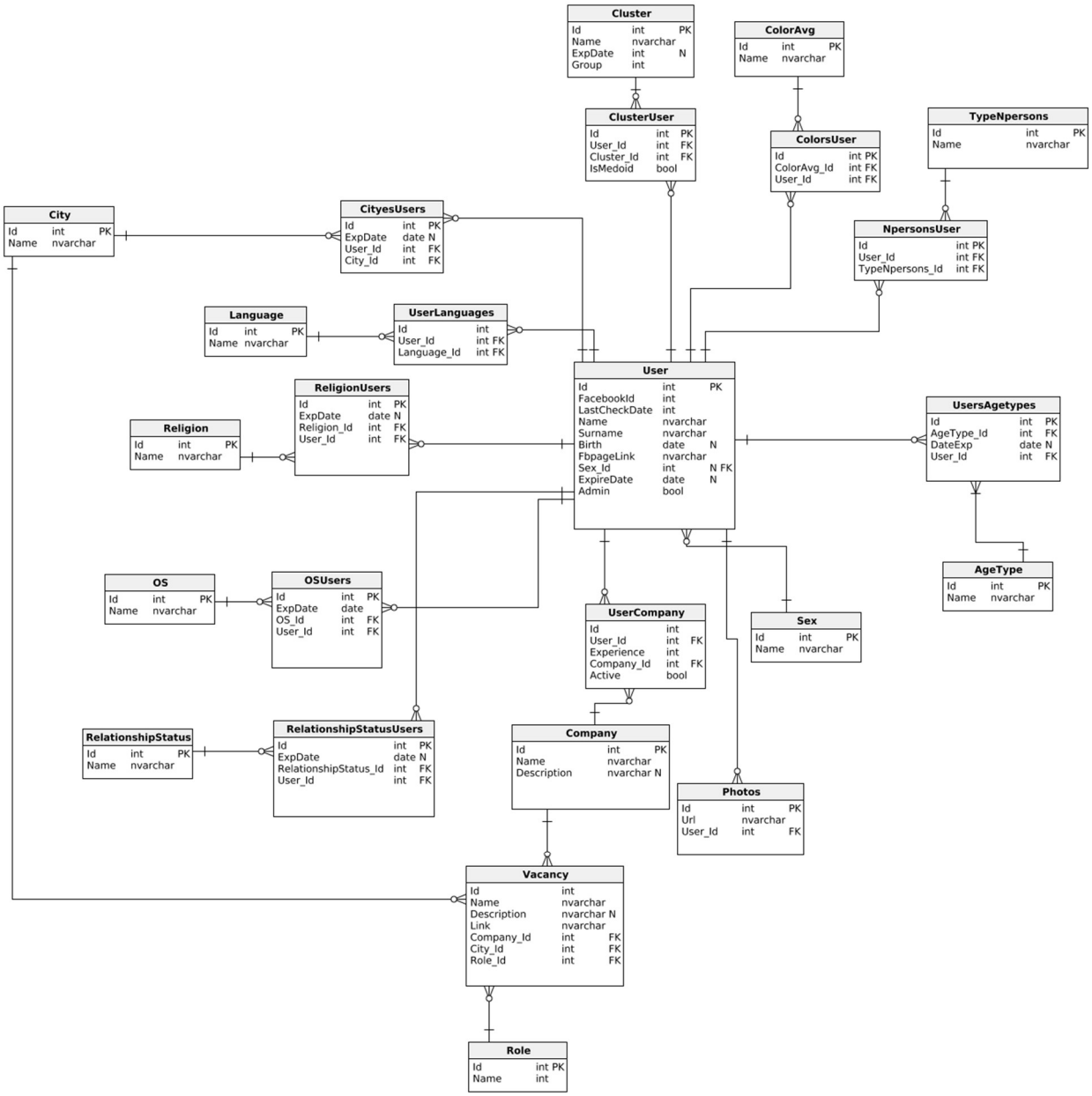

Рисунок 4.3 – Схема бази даних

Опис таблиць діаграми класів наведено в таблицях 4.3– 4.27.

Таблиця 4.3 – Опис таблиць бази даних

Назва таблиці	Опис
User	Користувачі
Company	Компанії
Vacancy	Вакансії
Sex	Стать

Назва таблиці	Опис
UsersAgeTypes	Таблиця характеристик Користувач-Вікова група
AgeType	Вікова група
UserCompany	Таблиця характеристик Користувач-Компанія
NpersonsUser	Таблиця характеристик Користувач – Група соціалізованості
TypeNpersons	Група соціалізованості (кількість осіб на фотографії)
ColorsUser	Таблиця характеристик Користувач – Кольоротип
ColorAvg	Кольоротип
Photos	Фотографії користувача
Role	Посади
UserLanguages	Таблиця характеристик Користувач – Мова
Language	Мова
CitiesUsers	Таблиця характеристик Користувач – Місто
City	Місто
ReligionUsers	Таблиця характеристик Користувач – Релігія
Religion	Релігія
OSUsers	Таблиця характеристик Користувач – Використовувана операційна система
OS	Використовувана операційна система
RelationshipStatusUsers	Таблиця характеристик Користувач – Статус відносин
RelationshipStatus	Статус відносин
Cluster	Кластери даних
ClusterUser	Таблиця характеристик Користувач-Кластер

Таблиця 4.4 – Опис таблиці User

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Facebookid	Ідентифікатор користувача з Facebook	int			
LastCheckDate	Дата останнього оновлення	int			
Name	Ім’я	nvarchar			
Surname	По батькові	nvarchar			
Birth	Дата народження	date			+
FbPageLink	Посилання на сторінку Facebook	nvarchar			
Sex_Id	Стать	int		+	+
ExpireDate	Дата набуття неактуальності	date			+
Admin	Право адміністування	bool			

Таблиця 4.5 – Опис таблиці Company

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва компанії	nvarchar			
Description	Опис	nvarchar			+

Таблиця 4.6 – Опис таблиці Vacancy

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва вакансії	nvarchar			
Description	Опис	nvarchar			+
Link	Посилання	nvarchar			
Company_Id	Ідентифікатор компанії	int		+	
City_Id	Ідентифікатор міста	int		+	
Role_Id	Ідентифікатор посади	int		+	

Таблиця 4.7 – Опис таблиці Sex

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва статевої принележності	nvarchar			

Таблиця 4.8 – Опис таблиці UsersAgeTypes

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
AgeType_Id	Ідентифікатор вікової групи	int		+	
DateExp	Дата набуття неактуальності	date			+
User_Id	Ідентифікатор користувача	int		+	

Таблиця 4.9 – Опис таблиці AgeType

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва вікової групи	nvarchar			

Таблиця 4.10 – Опис таблиці NpersonsUser

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
User_Id	Ідентифікатор користувача	int		+	
TypeNpersons_Id	Ідентифікатор групи соціалізованості	int		+	

Таблиця 4.11 – Опис таблиці TypeNpersons

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва групи соціалізованості	varchar			

Таблиця 4.12 – Опис таблиці ColorsUser

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
ColorAvg_Id	Ідентифікатор кольоротипу	int		+	
User_Id	Ідентифікатор користувача	int			

Таблиця 4.13 – Опис таблиці ColorAvg

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва кольоротипу	nvarchar			

Таблиця 4.14 – Опис таблиці Photos

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Url	Посилання на фотографію	nvarchar			
User_Id	Ідентифікатор користувача	int		+	

Таблиця 4.15 – Опис таблиці Role

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва посади	nvarchar			

Таблиця 4.16 – Опис таблиці UserCompany

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
User_Id	Ідентифікатор користувача	int		+	
Expierence	Час роботи (в місяцях)	int			
Company_Id	Ідентифікатор компанії	int		+	

Назва поля	Опис	Тип поля	PK	FK	Null
Active	Актуальність запису	bool			

Таблиця 4.17 – Опис таблиці UserLanguages

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
User_Id	Ідентифікатор користувача	int		+	
Language_Id	Ідентифікатор мови	int		+	

Таблиця 4.18 – Опис таблиці Language

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва мови	nvarchar			

Таблиця 4.19 – Опис таблиці CitiesUsers

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
ExpDate	Дата набуття неактуальності	date			+
User_Id	Ідентифікатор користувача	int		+	
City_Id	Ідентифікатор міста	int		+	

Таблиця 4.20 – Опис таблиці City

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва міста	nvarchar			

Таблиця 4.21 – Опис таблиці ReligionUsers

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
ExpDate	Дата набуття запису неактуальності	date			+
User_Id	Ідентифікатор користувача	int		+	
Religion_Id	Ідентифікатор релігії	int		+	

Таблиця 4.22 – Опис таблиці Religion

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва релігії	nvarchar			

Таблиця 4.23 – Опис таблиці OSUsers

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
ExpDate	Дата набуття запису неактуальності	int			+
User_Id	Ідентифікатор користувача	int		+	

Назва поля	Опис	Тип поля	PK	FK	Null
OS_Id	Ідентифікатор операційної системи	int		+	

Таблиця 4.24 – Опис таблиці RelationshipStatusUsers

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
ExpDate	Дата набуття запису неактуальності	date			+
RelationshipStatus_Id	Ідентифікатор статусу відносин	int		+	
User_Id	Ідентифікатор користувача	int		+	

Таблиця 4.25 – Опис таблиці RelationshipStatus

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва статусу відносин	varchar			

Таблиця 4.26 – Опис таблиці Cluster

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
Name	Назва кластера	nvarchar			

Назва поля	Опис	Тип поля	PK	FK	Null
ExpDate	Дата набуття запису неактуальності	int			+
Group	Номер групи	int			

Таблиця 4.27 – Опис таблиці ClusterUser

Назва поля	Опис	Тип поля	PK	FK	Null
Id	Ідентифікатор	int	+		
User_Id	Ідентифікатор користувача	int		+	
Cluster_Id	Ідентифікатор кластера	int		+	
IsMedoid	Чи є даний користувач медоїдом	bool			

4.7 Керівництво користувача

Для роботи з програмним забезпеченням, користувач має мати обліковий запис у соціальній мережі Facebook [61].

4.7.1 *Авторизація*

Для авторизації необхідно натиснути на кнопку “Увійти з Facebook”. Сторінка аторизації зображена на рисунку 4.4.

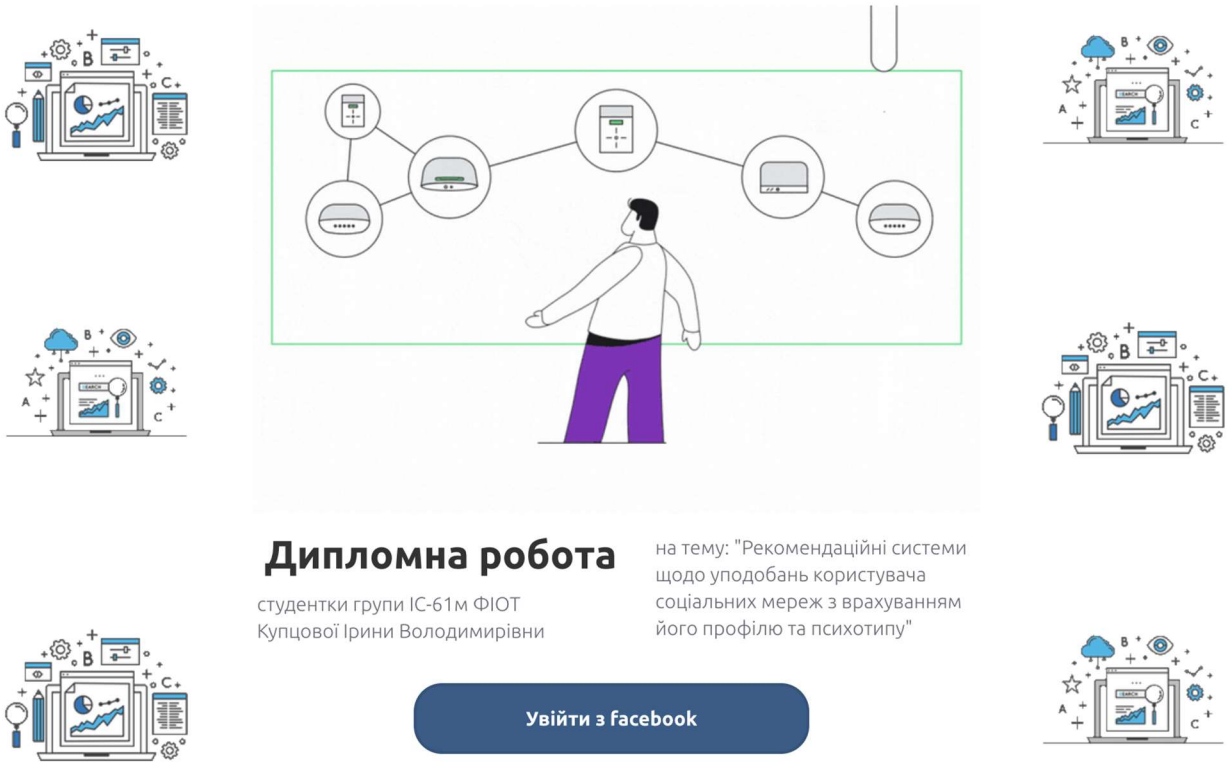

Рисунок 4.4 – Сторінка авторизації

Далі необхідно вказати свої авторизаційні дані для входу в Facebook у формі, що зображена на рисунку 4.5 та погодитися зі збором даних даним застосуванням у формі, що зображена на рисунку 4.6.

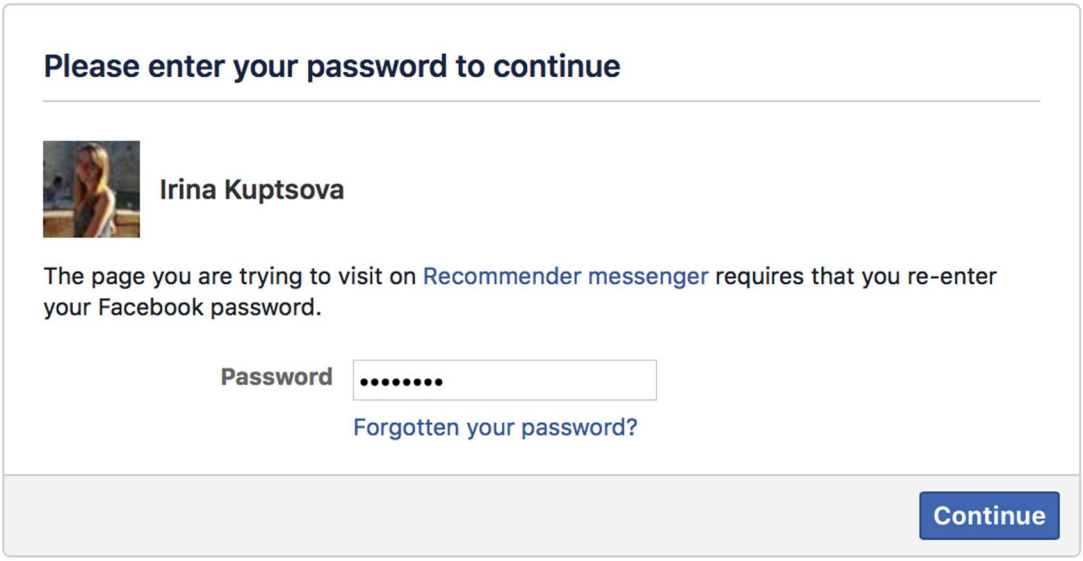

Рисунок 4.5 – Сторінка авторизації в Facebook

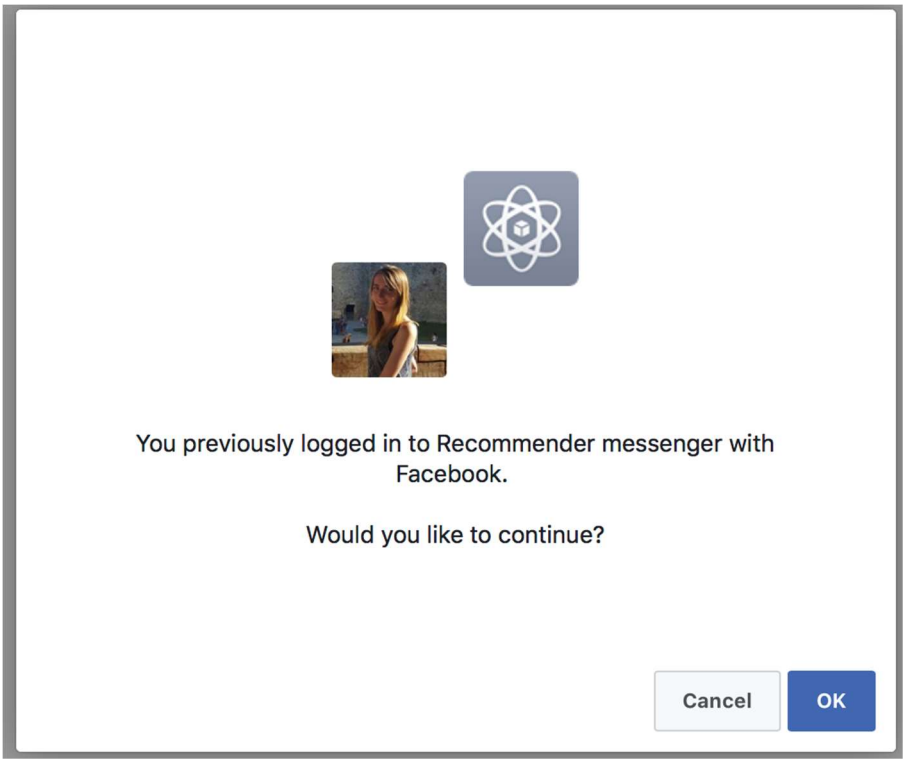

Рисунок 4.6 – Сторінка погодження зі збором даних

4.7.2 *Пошук роботи*

Після авторизації буде відкрита сторінка пошуку роботи, що зображена на рисунку 4.7. Після вибору бажаної посади з відповідного випадаючого списку буде виведено список компаній, які рекомендовано для працевлаштування з огляду на дані соціального профілю користувача.

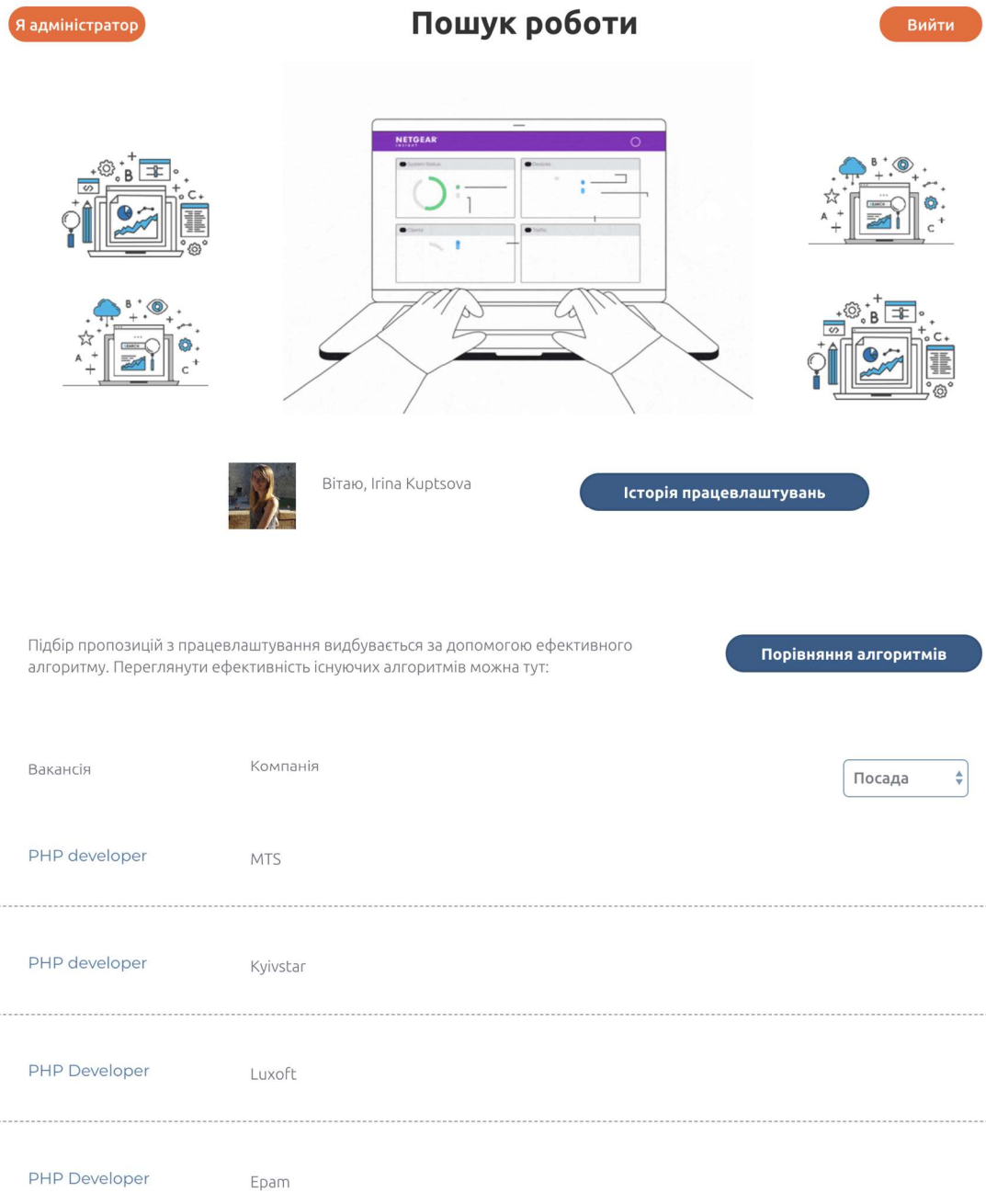

Рисунок 4.7 – Сторінка пошуку роботи

Для перегляду ефективності використовуваних алгоритмів, необхідно натиснути кнопку “Порівняння алгоритмів”.

Для введення історії попередніх працевлаштувань, необхідно натиснути кнопку “Історія працевлаштувань”.

У разі, якщо користувач має права адміністрування, для введення списку вакансій, необхідно натиснути кнопку “Я адміністратор”.

Для виходу з облікового запису, необхідно натиснути кнопку “Вийти”.

4.7.3 *Порівняння алгоритмів*

На сторінці порівняння алгоритмів, що зображена на рисунку 4.8, вказано графіки залежності оцінки розбиття SWC (метод оцінювання вказаний в розділі 5.3.1) від кількості кластерів для методу k-середніх та модифікованого методу та гістограми, що показують точність класифікації (метод оцінювання вказаний в розділі 5.3.2), здійснених окремими методами для різних наборів даних (опис наборів даних вказаний в розділі 5.6.1).

Для повернення на головну сторінку необхідно натиснути кнопку “Назад”.

Рисунок 4.8 – Сторінка порівняння алгоритмів

4.7.4 *Історія працевлаштувань*

Для вказання історії попередніх працевлаштувань необхідно обрати компанію з випадаючого списку на сторінці “Історія працевлаштувань”, що зображена на рисунку 4.9, якщо у списку відсутня конкретна назва компанії, необхідно обрати елемент “Інша”, у текстове поле “Досвід” необхідно ввести кількість місяців, відпрацьованих у вказаній компанії та натиснути кнопку “Додати”.

Для відміни доданого запису, необхідно натиснути кнопку “Деактивувати”.

Для повернення на головну сторінку необхідно натиснути кнопку “Назад”.

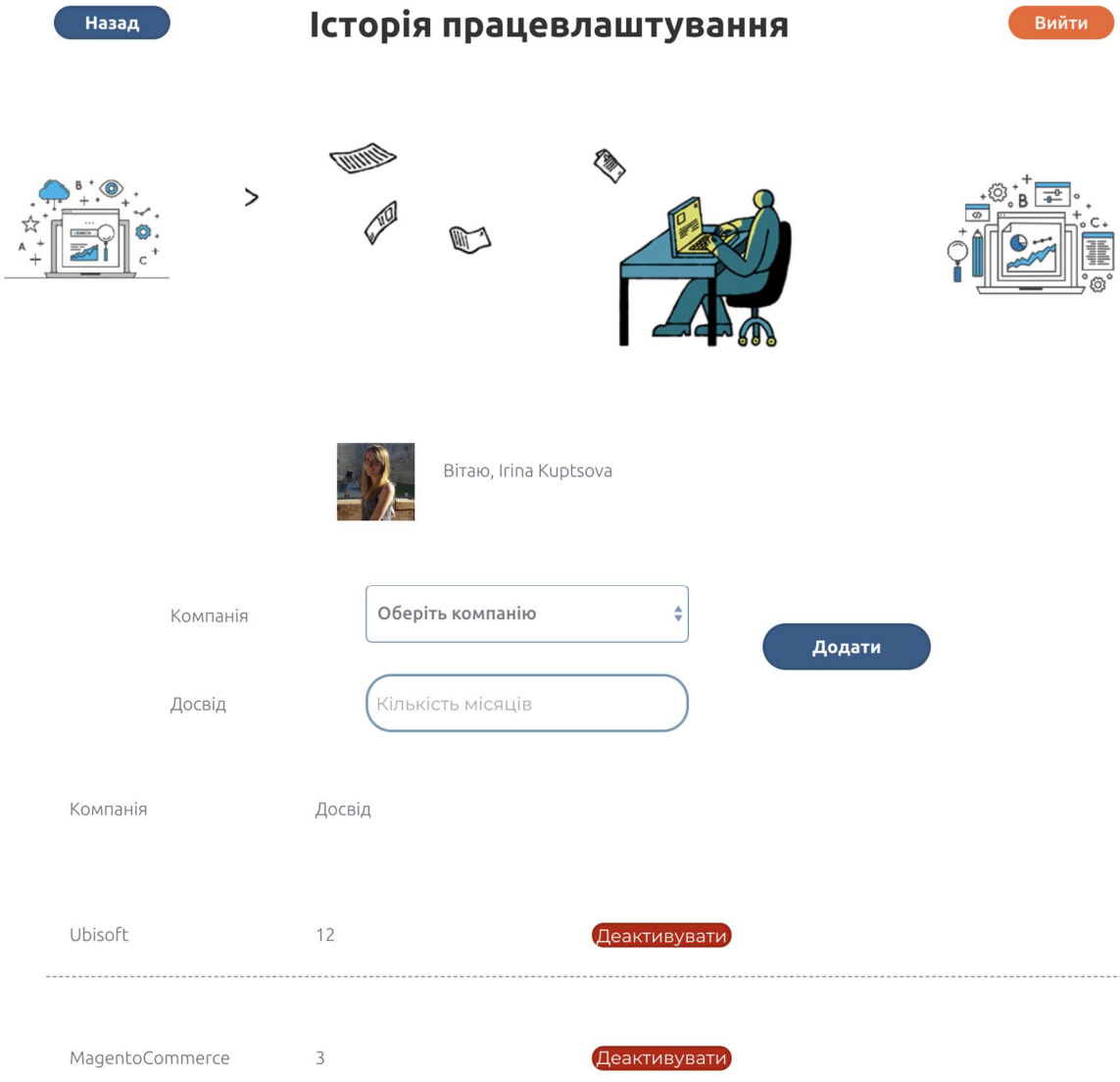

Рисунок 4.9 – Сторінка вказання історії працевлаштувань

4.7.5 *Адміністрування*

На сторінці адміністрування, що зображена на рисунку 4.10, користувач має можливість вказати нові вакансії, що пропонуються для рекомендацій. Можна вказати назву компанії, пропоновану посаду та посилання на зовнішній ресурс. Для деактивації вакансії необхідно натиснути кнопку “Деактивувати”.

Для повернення на головну сторінку необхідно натиснути кнопку “Назад”.

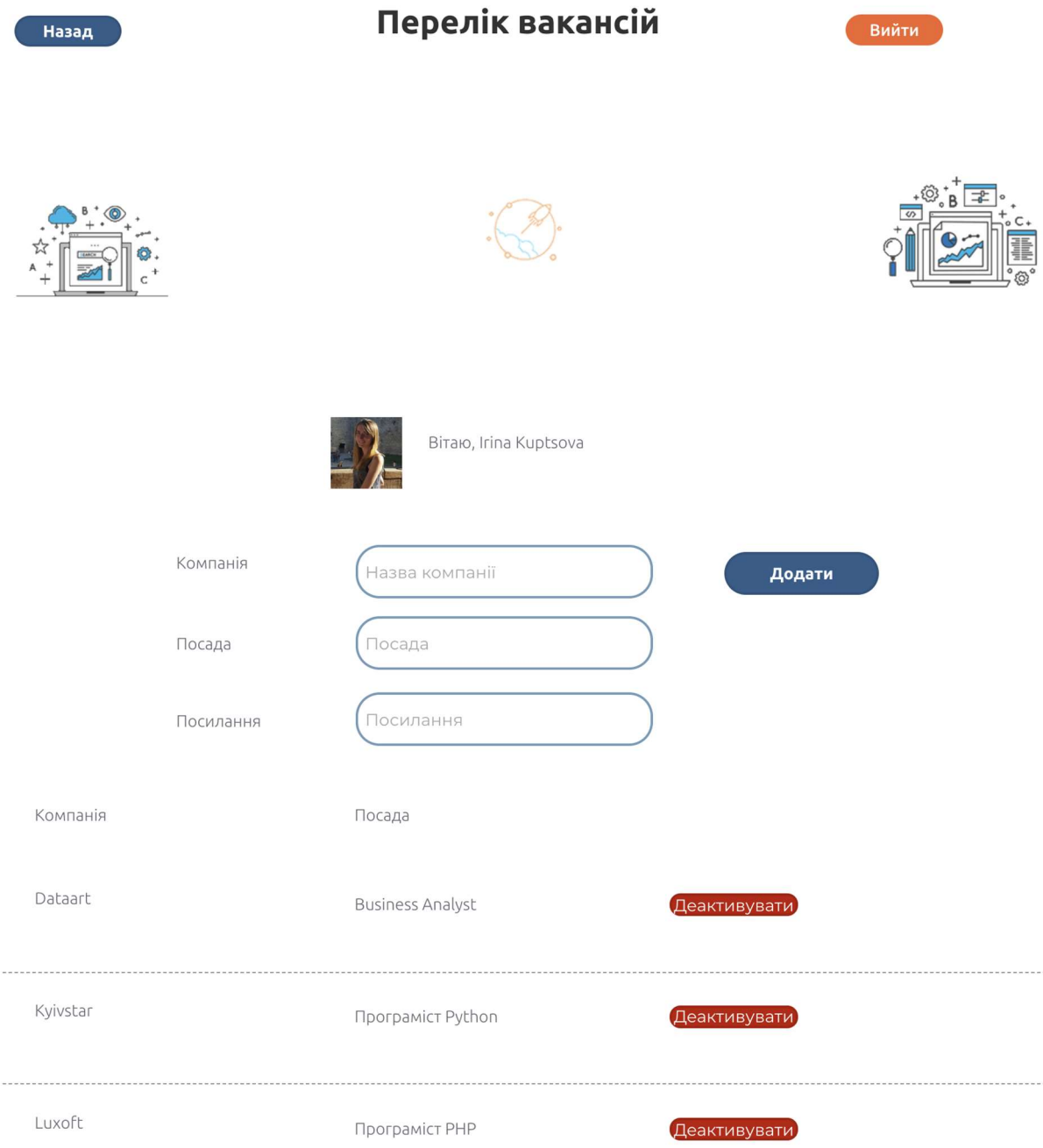

Рисунок 4.10 – Сторінка адміністрування

4.8 Висновки до розділу

В даному розділі було обґрунтовано вибір засобів розробки, наведено склад вхідних та вихідних даних, наведено діаграму прецедентів, описано базу даних, що використовується в створеному застосуванні, та бізнес-процес знаходження

рекомендацій з вибору роботи користувачем, а також наведено керівництво користувача.

Сервіс має види користувачів – відвідувач та адміністратор, який має всі права відвідувача, а також можливості управління деякими параметрами роботи сервісу.

Було розроблено зручний інтерфейс для роботи з сервісом для будь-якої системної ролі, а також створена можливість авторизації за допомогою соціальної мережі Facebook з ціллю пришвидшення процесу реєстрації, що має бути привабливим фактором для розвитку сервісу, а також засобом збору інформації про користувача.

5 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

5.1 Визначення домінантних кольорів на фотографії

На рисунках 5.1-5.3 показана робота модифікованого алгоритму для вибору домінуючих кольорів фотографії.

Рисунок 5.1 – Результати роботи модифікованого алгоритму для визначення домінантних кольорів на фотографії

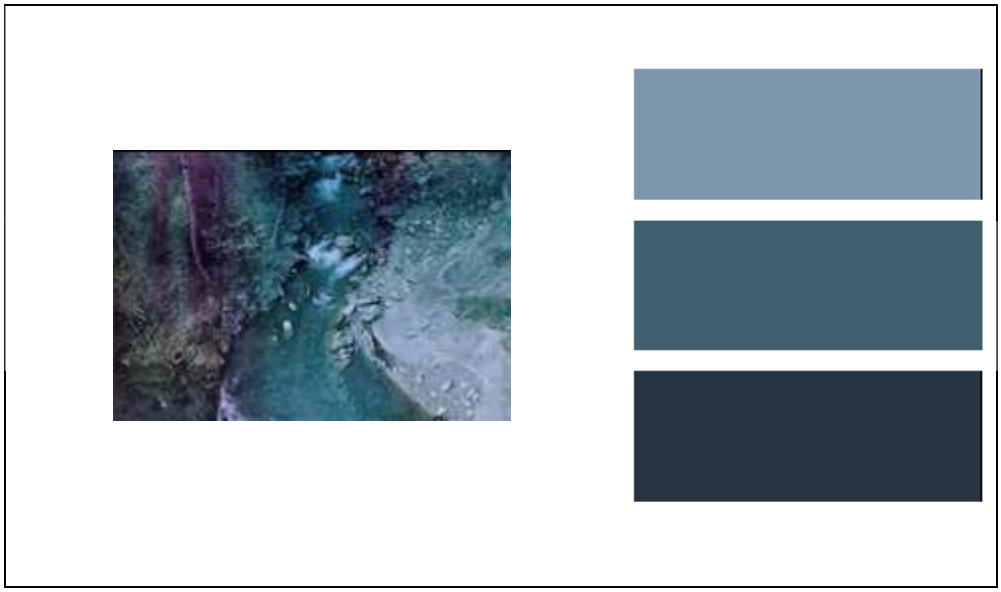

Рисунок 5.2 – Результати роботи модифікованого алгоритму для визначення домінантних кольорів на фотографії

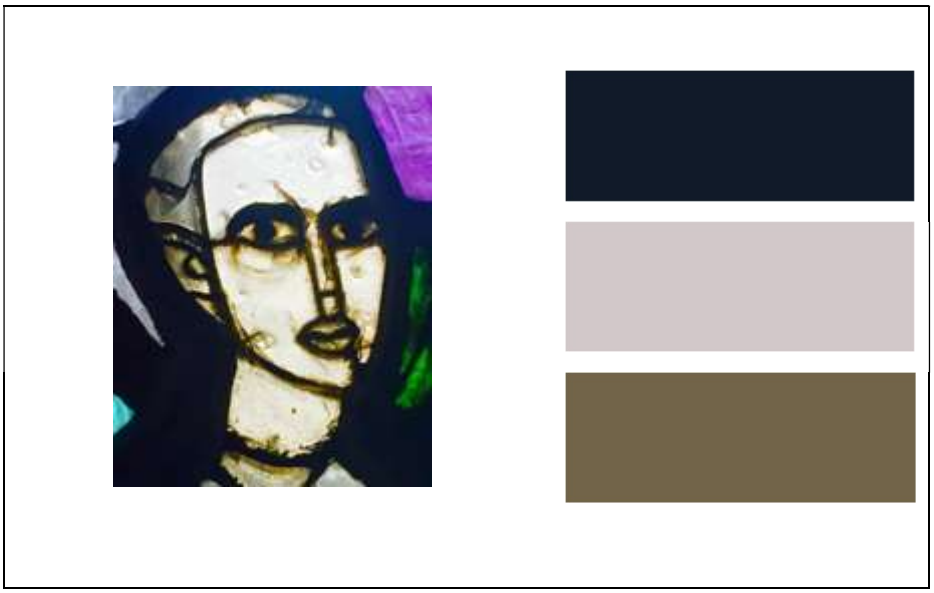

Рисунок 5.3 – Результати роботи модифікованого алгоритму для визначення домінантних кольорів на фотографії

5.2 Визначення кількості обличь на фотографії

На рисунку 5.4 представлені результати визначення обличь на фотографії за допомогою каскадів Хаара.

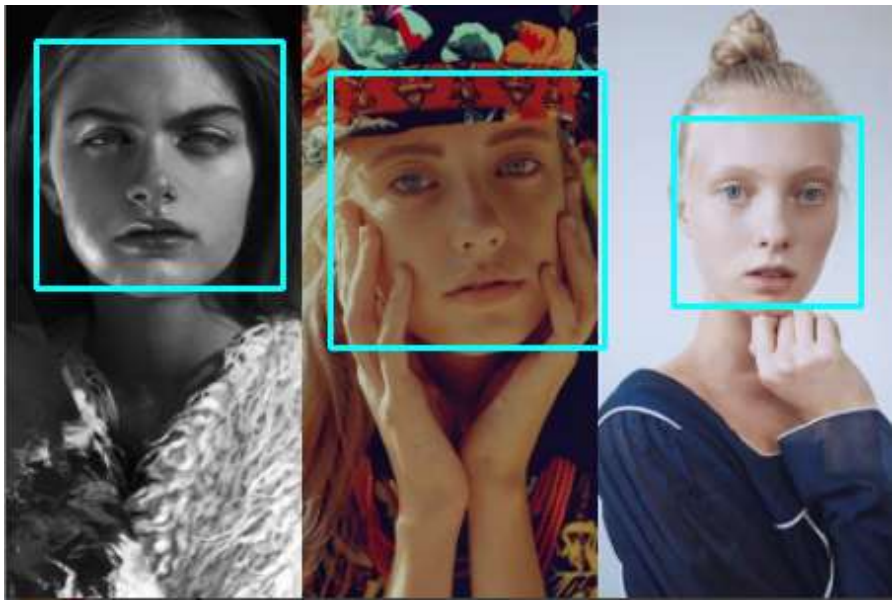

Рисунок 5.4 – Результати визначення обличь на фотографії за допомогою каскадів Хаара

5.3 Методи оцінювання

5.3.1 *Визначення якості розбиття на кластери*

Для оцінки результатів кластеризації пропонується використовувати метод спрощеного індексу оцінки силуету (Simplified Silhoutte index) [37].

Силует кожного кластера визначається наступним чином: припустимо, об'єкт x_i належить кластера c_r. Визначимо відстань від цього об'єкта до медоїду того ж самого кластера c_r через a_{ri}. Тепер визначимо середню відстань від x_i до медоїдів інших кластерів $c_t, t \neq r$ через d_{ti}. Покладемо $b_{ri} = \min\limits_{t \neq r} d_{ti}$.

Сенс цієї величини можна визначити як міру несхожості окремого елемента з елементами найближчого кластера. Таким чином, силует кожного окремого елемента визначається наступним чином:

$$S_{x_i} = \frac{b_{ri} - a_{ri}}{\max(a_{ri}, b_{ri})}. \quad (5.1)$$

Знаменник введено задля нормалізації значень. Очевидно, що високе значення показника S_{xi} характеризує собою кращу приналежність елемента x_i кластеру c_r. Оцінка для всієї кластерної структури досягається усередненням показника по всім елементам:

$$SWC = \frac{1}{N} \sum_{i=1}^{n} S_{x_i}. \quad (5.2)$$

Найкраще розбиття характеризується максимальним SWC, що досягається коли відстань всередині кластера a_{ri} мала, а відстань між елементами сусідніх кластерів b_{ri} велика.

5.3.2 *Метод оцінки класифікаційних алгоритмів*

Для виміру точності роботи класифікаційних алгоритмів обрана наступна метрика:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F}, \quad (5.3)$$

де TP – істинно-позитивне рішення;

TN – істинно-негативне рішення;

FP – помилково-позитивне рішення;

FN – помилково-негативне рішення, що сформовані з матриці контигентності, що наведена в таблиці 5.1.

Таблиця 5.1 – Матриця контигентності для класу

i-й клас		Визначена алгоритмом мітка класу	
		Об’єкти належать класу	Об’єкти не належать класу
Істинна мітка класу	Об’єкти належать класу	TP	FP
	Об’єкти не належать класу	FN	TN

5.4 Визначення кількості атрибутів для аналізу вибірки

Для визначення мінімальної розмірності набору атрибутів, що дозволяє видалення одного з них без втрати якості розбиття було проведено випробування для вибірки даних, сформованої для програмного продукту - Psychotypes Data Set (містить 9 категоріальних змінних).

Так, було виконано 16 прогонів алгоритму, описаного в розділі 3.1.8, для вибірки даних, включаючи та відкидаючи атрибут, що задовольняє нерівності (3.9) для різних випадкових наборів атрибутів потужністю l.

Результати прогону вказані в таблиці 5.2 та на рисунку 5.5.

Відношення r визначається як:

$$r = \frac{P_a}{P_f}, \tag{5.4}$$

де P_a – точність класифікації об’єктів для вибірки з видаленим атрибутом a, що задовольняє нерівність (3.9);

P_f – точність класифікації об’єктів для вибірки без видалення атрибутів.

Точність класифікації об’єктів для всіх кластерів визначається як середнє значення точності для кожного кластера:

$$P_f = \frac{\sum_{i=1}^{k} Accuracy(i)}{k}, \quad (5.5)$$

де $Accuracy(i)$ – значення точності для і-го кластера, обчислене за формулою (5.3);

k – кількість кластерів.

Значення P_a визначається аналогічно.

Таблиця 5.2 – Залежність точності кластеризації від кількісті атрибутів, що піддаються аналізу

Кількість атрибутів, l / Номер набору (ряд)	2	3	4	5	6	7	8	9
1	0,762	0,86	0,955	0,975	0,981	0,987	0,989	0,992
2	0,779	0,801	0,883	0,934	0,942	0,951	0,974	0,983
3	0,762	0,785	0,861	0,912	0,924	0,944	0,963	0,988
4	0,775	0,853	0,893	0,912	0,927	0,942	0,965	0,981
5	0,771	0,793	0,912	0,953	0,963	0,978	0,983	0,99
6	0,725	0,834	0,925	0,962	0,977	0,985	0,992	0,997
7	0,826	0,864	0,935	0,974	0,981	0,984	0,988	0,991
8	0,894	0,923	0,935	0,953	0,967	0,972	0,983	0,992
9	0,835	0,889	0,893	0,904	0,925	0,965	0,985	0,991
10	0,875	0,894	0,902	0,912	0,922	0,934	0,947	0,964
11	0,894	0,903	0,923	0,934	0,945	0,956	0,958	0,972
12	0,846	0,867	0,889	0,901	0,921	0,948	0,952	0,962
13	0,901	0,920	0,935	0,942	0,952	0,972	0,992	0,998

Кількість атрибутів, l / Номер набору (ряд)	2	3	4	5	6	7	8	9
14	0,864	0,885	0,875	0,899	0,912	0,936	0,954	0,972
15	0,892	0,904	0,920	0,942	0,956	0,972	0,983	0,996
16	0,821	0,876	0,892	0,904	0,944	0,985	0,989	0,992

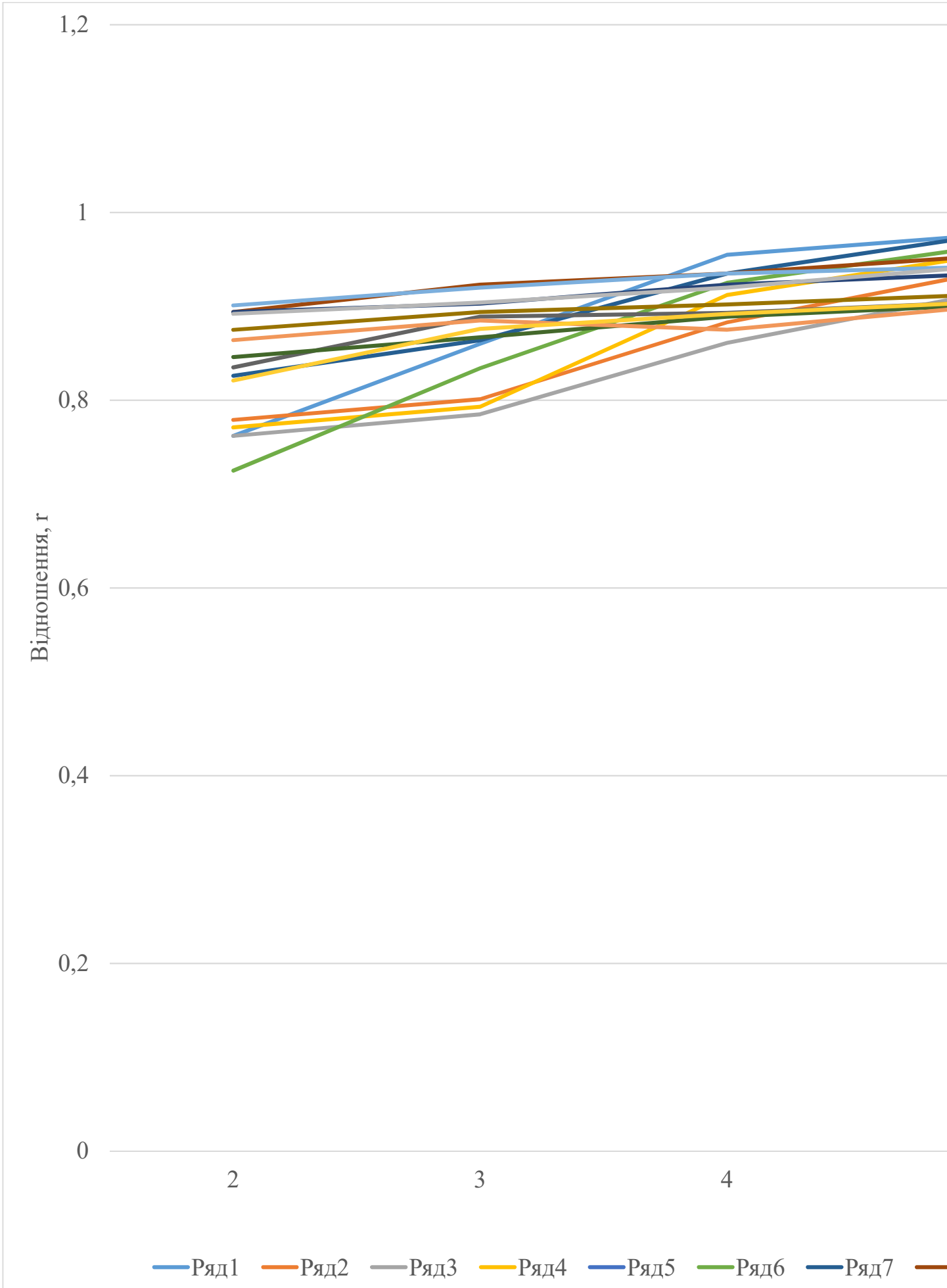
1,2
1
0,8
0,6
0,4
0,2
0
Відношення, г
2
3
4
Ряд1
Ряд2
Ряд3
Ряд4
Ряд5
Ряд6
Ряд7

Рисунок 5.5 – Залежність точності кластеризації від кількості атрибутів, що піддаються аналізу

Так, з наведених даних видно, що при наборі даних, що містить не менше 7 атрибутів, видалення одного, що задовольняє нерівності (3.9) не призводить до значних втрат класифікації, а тому відкидання подібних атрибутів з вхідної вибірки є доцільним.

5.5 Визначення порогового значення кількості прогонів алгоритму кластеризації без покращень цільової функції

Вибірки, що були використані для визначення порогового значення Q наведено в таблиці 5.3 [98, 90].

Таблиця 5.3 – Вибірки, що були використані

Скорочення	Назва вибірки
S1	Chicago Entree Data Set
S2	Daily and sport activities Data Set
S3	Sponge Data Set
S4	Psychotypes Data Set
S5	Plants Data Set
S6	Water treatment flowers Data Set
S7	Million Song Dataset Challenge Data Set
S8	Scholarly Paper Recommendation Data Set

В таблиці 5.6 та на рисунку 5.4 описано залежність якості розбиття набору даних від значення Q, поясненого в розділі 3.1.8.

Таблиця 5.4 – Залежність якості розбиття від значення Q

Q	Вибірка даних							
	S1	S2	S3	S4	S5	S6	S7	S8
0,2k	0,541	0,672	0,490	0,531	0,550	0,458	0,610	0,582
0,4k	0,552	0,680	0,512	0,552	0,572	0,475	0,645	0,612

Q	Вибірка даних							
	S1	S2	S3	S4	S5	S6	S7	S8
0,6k	0,567	0,683	0,512	0,573	0,593	0,497	0,650	0,642
0,8k	0,585	0,685	0,550	0,592	0,612	0,515	0,674	0,659
1,0k	0,618	0,729	0,562	0,610	0,622	0,534	0,695	0,688
1,2k	0,619	0,735	0,574	0,62	0,643	0,559	0,711	0,729
1,4k	0,625	0,748	0,582	0,653	0,665	0,575	0,735	0,765
1,6k	0,658	0,753	0,590	0,688	0,678	0,589	0,758	0,789
1,8k	0,693	0,762	0,605	0,707	0,683	0,602	0,773	0,801
2,0k	0,723	0,772	0,617	0,720	0,702	0,615	0,798	0,843
2,2k	0,738	0,785	0,623	0,729	0,712	0,617	0,811	0,864
2,4k	0,747	0,809	0,623	0,735	0,719	0,619	0,825	0,863
2,6k	0,750	0,812	0,623	0,735	0,719	0,619	0,825	0,864
2,8k	0,753	0,812	0,621	0,735	0,719	0,618	0,825	0,863
3,0k	0,750	0,810	0,623	0,735	0,719	0,620	0,825	0,863
3,2k	0,750	0,812	0,623	0,735	0,719	0,618	0,825	0,863
3,4k	0,750	0,812	0,623	0,735	0,719	0,618	0,825	0,863

В таблиці 5.5 наведено значення якості розбиття вибірок даних (S1 Opt – S8 Opt) для ситуації прогону алгоритму по всім об’єктам вибірки з метою перевизначення медоїдів.

Таблиця 5.5 – Значення якості розбиття для ситуації повного прогону

S1	S2	S3	S4	S5	S6	S7	S8
0,75	0,812	0,623	0,735	0,719	0,619	0,825	0,862

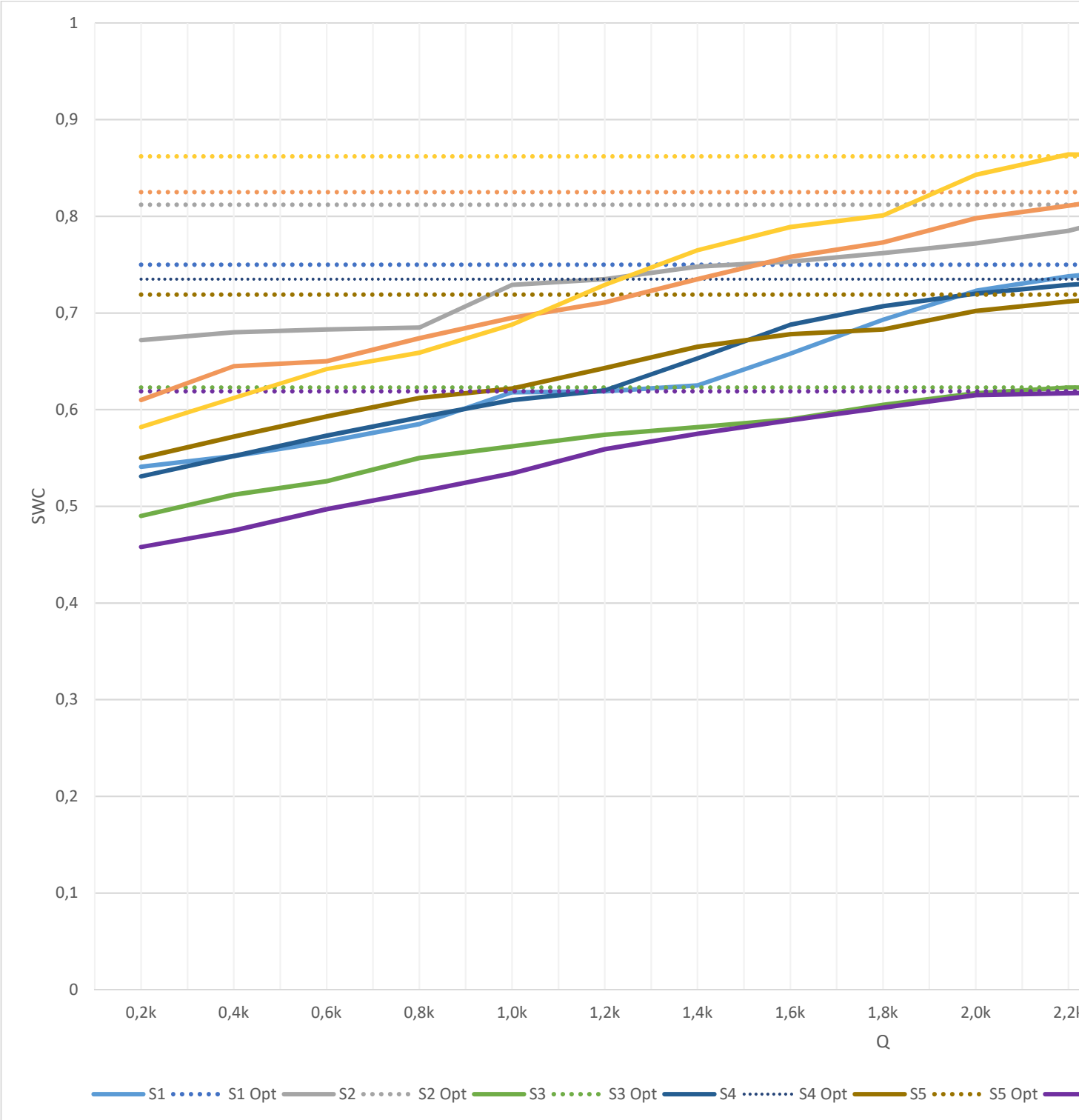

Рисунок 5.6 – Залежність оцінки якості розбиття SWC від значення Q

Як видно з наведених даних, після значення $Q = 2{,}6k$ оцінка SWC досягає однакового значення для тих же наборів даних за умови повного перебору всіх об’єктів набору, з чого можна зробити висновок про можливість використання порогового значення кількості прогонів алгоритму кластеризації без покращень цільової функції.

5.6 Порівняння алгоритмів надання рекомендацій

5.6.1 *Вибірки, що використовувалися для аналізу*

Для проведення експериментів використовувалися наступні вибірки [98]:

1) University Data Set;
2) Diabetes Data Set;
3) Automobile Data Set;
4) Molecular Data Set;
5) Plants Data Set;
6) Water Treatment Flowers Data Set;
7) Psychotypes Data Set (вибірка, сформована самостійно для роботи представленого програмного продукту).

5.6.2 *Результати кластеризації*

Для проведення експериментів з кластеризації даних було використано вибірки Plants Data Set, Water Treatment Flowers Data Set та Psychotypes Data Set.

Результати застосування методу індексу спрощеної оцінки силуету з використанням модифікованого методу до набору даних Psychotypes Data Set наведено на рисунку 5.7. За допомогою цих даних можна визначити оптимальну кількість кластерів для розбиття запропонованої вибірки даних.

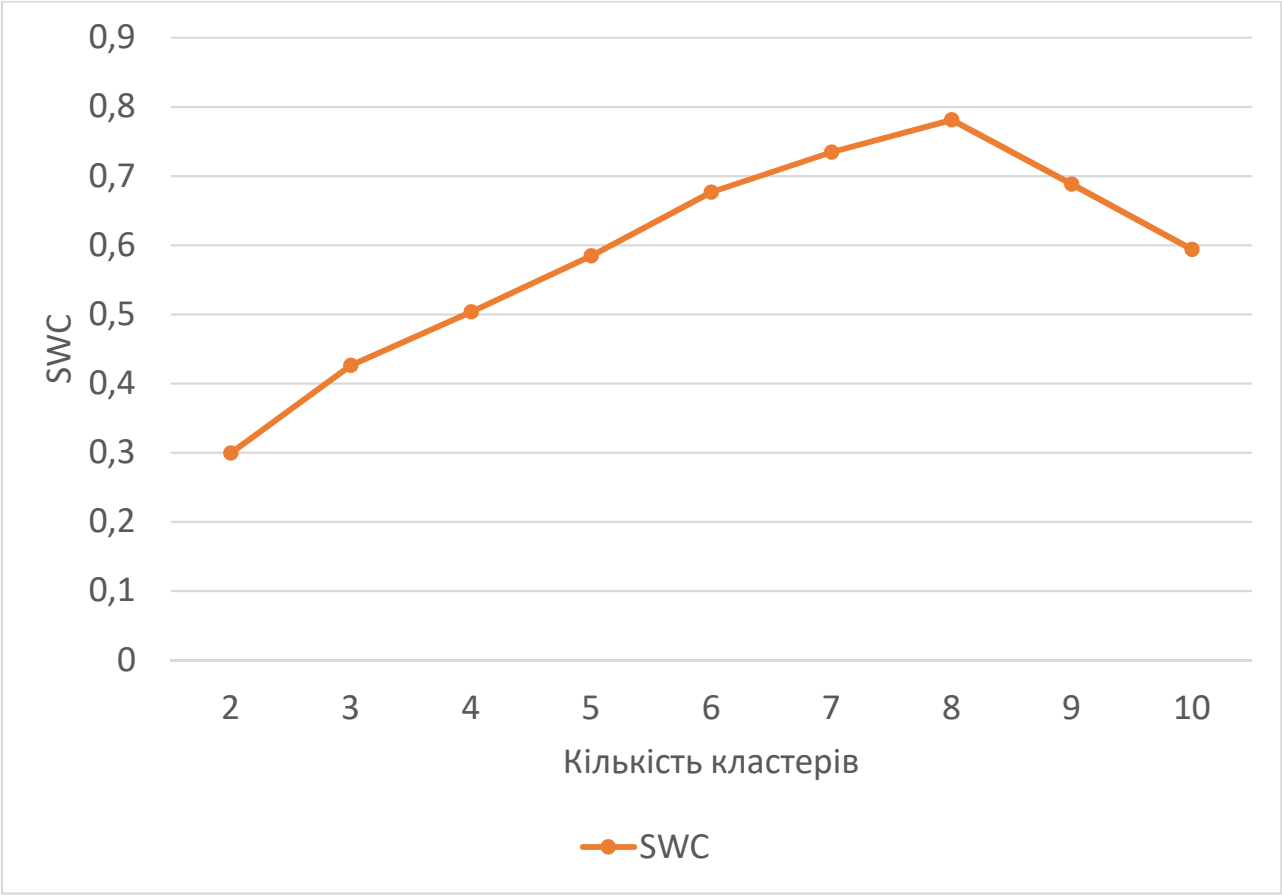

Рисунок 5.7 – Залежність індексу спрощеної оцінки силуету від кількості кластерів для вибірки даних Psychotypes Data Set

Згідно з отриманими результатами – оптимальною кількістю кластерів для даного набору даних є 8, саме за цієї умови досягається найякісніше розбиття.

В таблиці 5.6 та на рисунку 5.8 наведено порівняння оцінки якості кластеризації після застосування методу k-середніх та модифікованого методу до вибірки Psychotypes Data Set.

Таблиця 5.6 – Порівняння якості розбиття вибірки Psychotypes Data Set

Кількість кластерів	SWC (k-середніх)	SWC (Модифікований метод)
2	0,24623	0,29958
3	0,40562	0,42648
4	0,43065	0,50362
5	0,51669	0,58465

Кількість кластерів	SWC (k-середніх)	SWC (Модифікований метод)
6	0,57042	0,67675
7	0,66193	0,73461
8	0,69394	0,78120
9	0,61023	0,68809
10	0,59042	0,59385

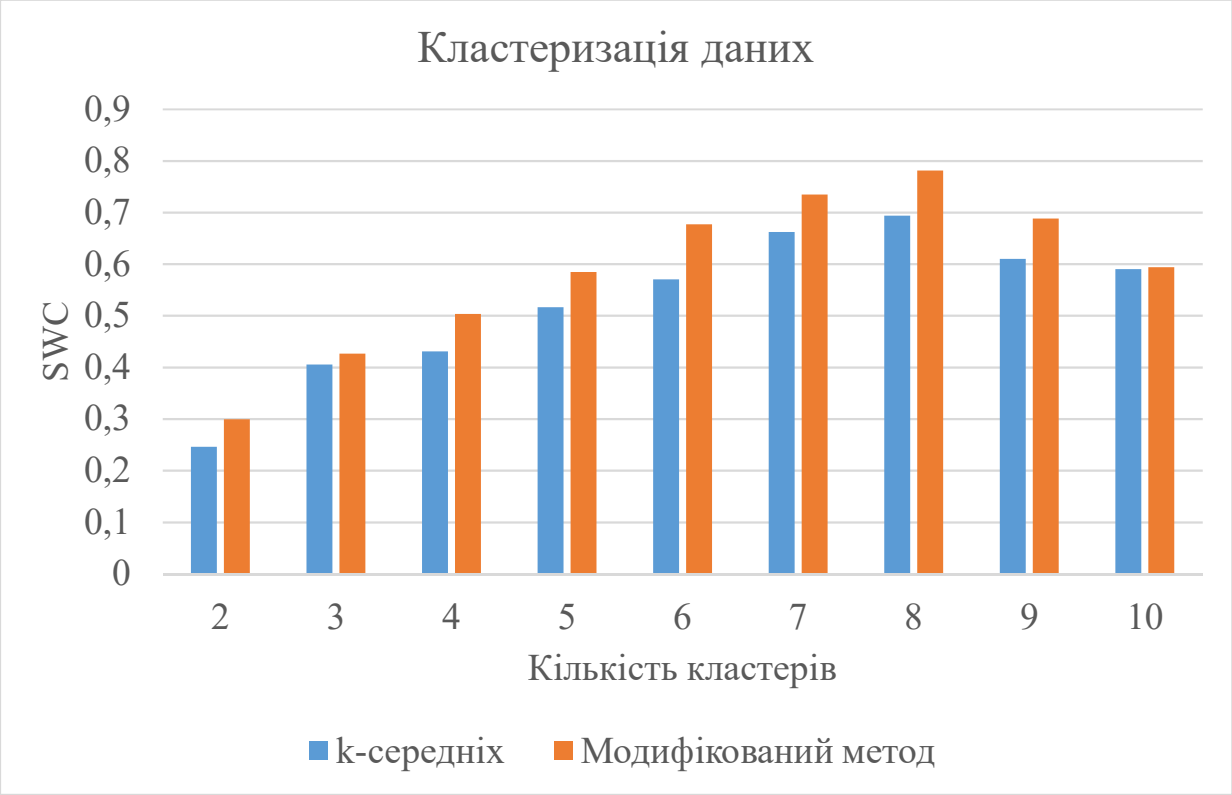

Рисунок 5.8 – Порівняння якості розбиття вибірки Psychotypes Data Set

В таблицях 5.7-5.8 та на рисунках 5.9-5.10 наведено результати експериментів над вибірками Plants Data Set та Water Treatment Flowers Data Set відповідно.

Таблиця 5.7 – Порівняння якості розбиття вибірки Plants Data Set

Кількість кластерів	SWC (k-середніх)	SWC (Модифікований метод)
2	0,19761	0,20749
3	0,23456	0,23836
4	0,29875	0,31368
5	0,35671	0,39455
6	0,44172	0,48380
7	0,47123	0,48063
8	0,56231	0,58042
9	0,62967	0,67347
10	0,67530	0,71938
11	0,59358	0,62368
12	0,48532	0,50974

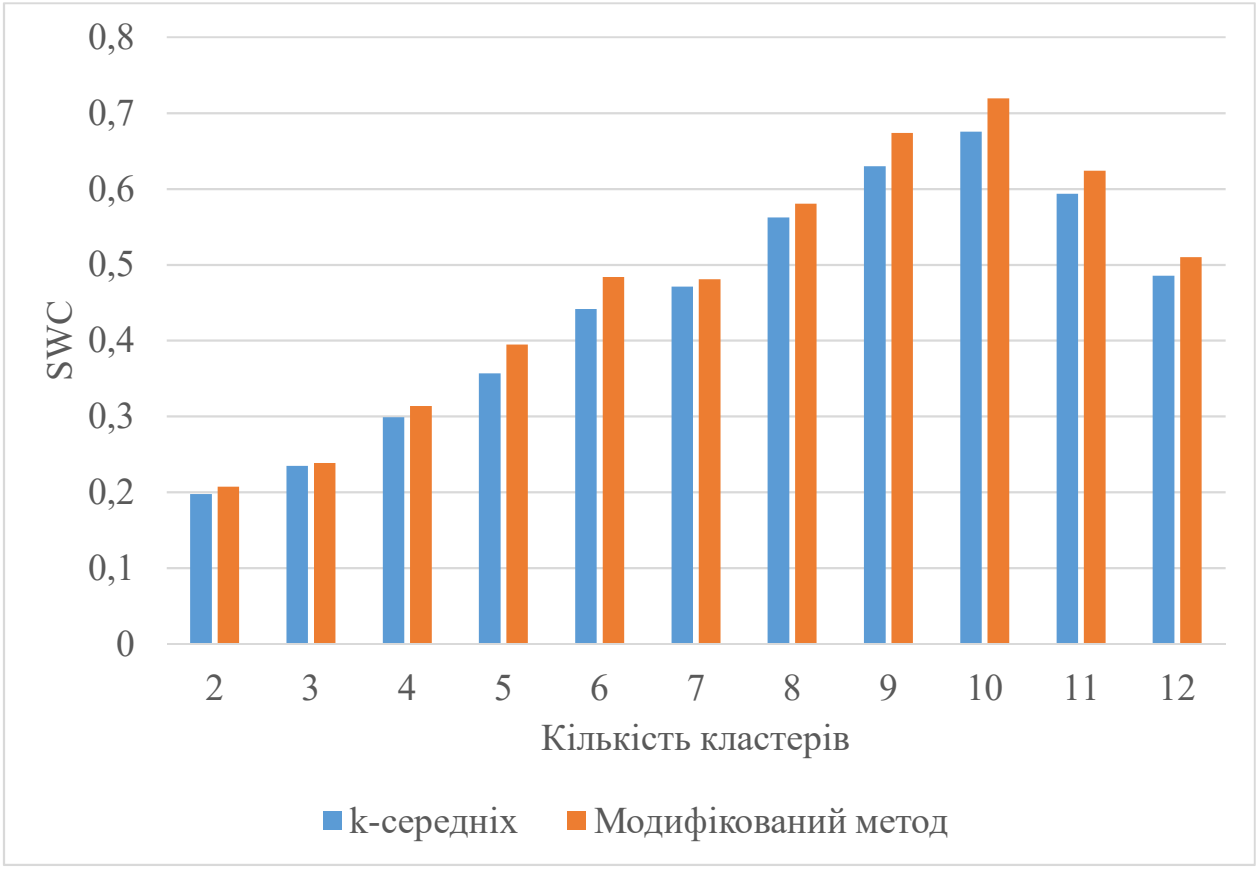

Рисунок 5.9 – Порівняння якості розбиття вибірки Plants Data Set

Таблиця 5.8 – Порівняння якості розбиття вибірки Water Treatment Flowers Data Set

Кількість кластерів	SWC (k-середніх)	SWC (Модифікований метод)
2	0,48823	0,50912
3	0,56172	0,59542
4	0,59022	0,61873
5	0,47095	0,49921
6	0,40156	0,44563

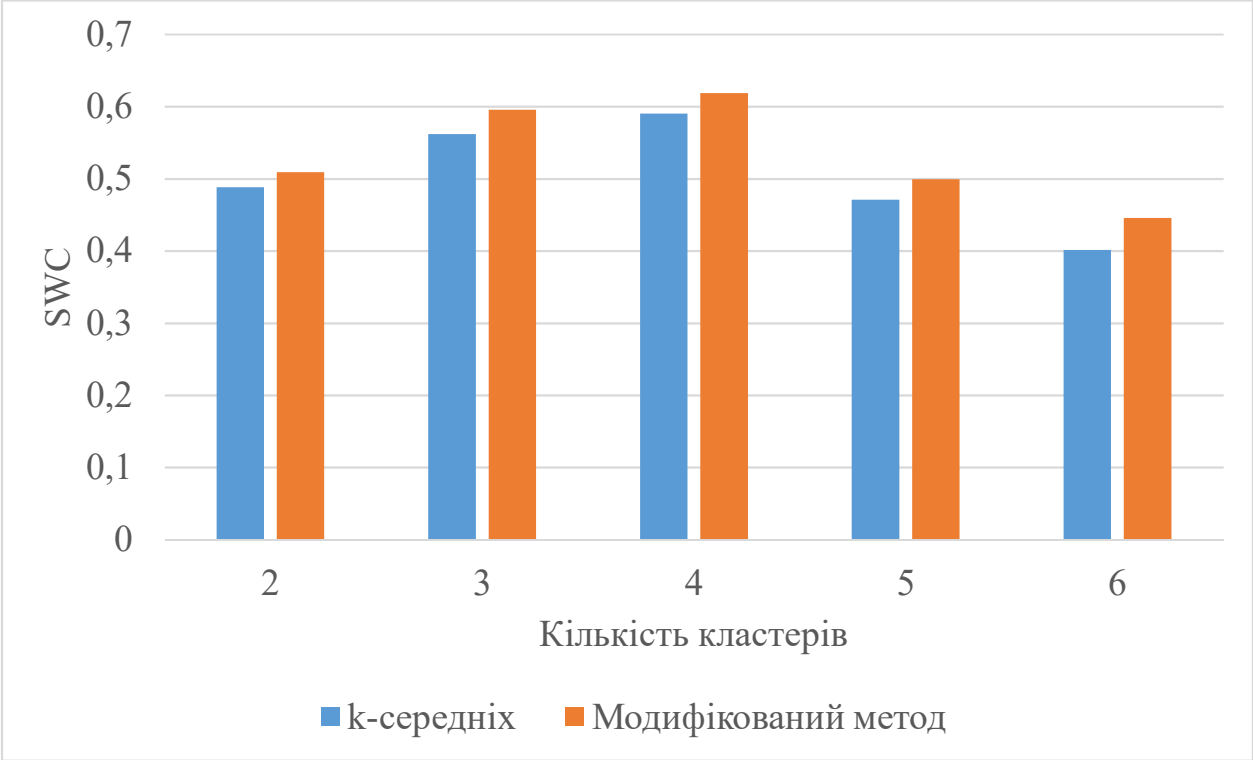

Рисунок 5.10 – Порівняння якості розбиття вибірки Water Treatment Flowers Data Set

5.6.3 *Результати класифікації*

Результати класифікації наведені в таблиці 5.11 та на рисунку 5.9. Показник ефективності *Accuracy* описаний в розділі 5.3.2.

Таблиця 5.9 – Порівняння класифікаційних алгоритмів для різних вибірок даних

Набір даних	*Accuracy*					
	КФ	НМБ	k-c	ID3	RF	Модифікований алгоритм
S1 (University Data Set)	0,75	0,89	0,84	0,79	0,83	0,87
S2 (Diabetes Data Set)	0,79	0,89	0,78	0,83	0,85	0,84
S3 (Automobile Data Set)	0,84	0,92	0,81	0,85	0,86	0,84
S4 (Molecular Data Set)	0,65	0,88	0,8	0,79	0,83	0,86
S5 (Psychotypes Data Set)	0,6	0,72	0,7	0,64	0,67	0,74

Примітки:

1. КФ – метод колаборативної фільтрації.
2. НМБ – метод наївних мереж Байєса.
3. k-c – метод k-середніх;
4. RF – метод Random Forest.

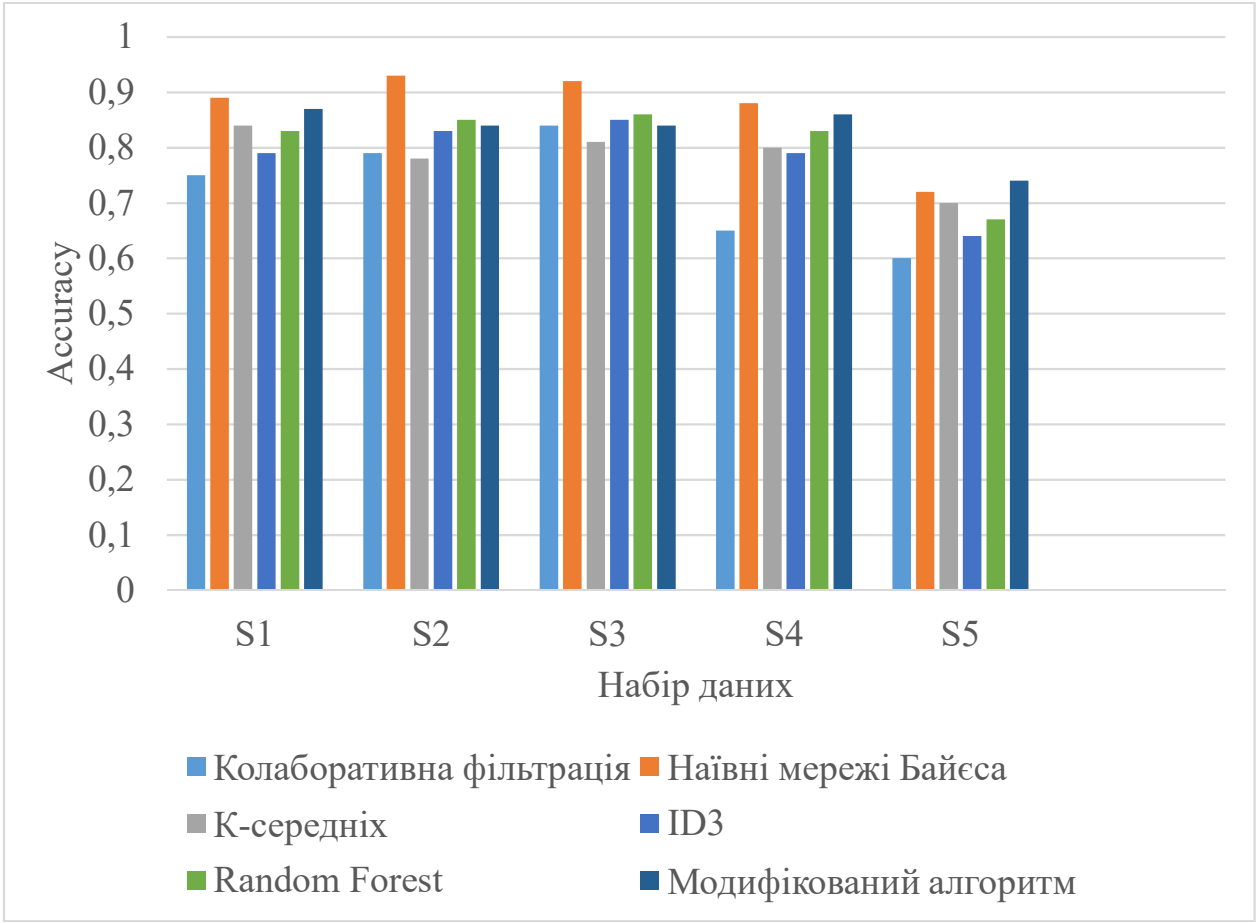

Рисунок 5.11 – Порівняння класифікаційних алгоритмів для різних вибірок даних

5.7 Висновки до розділу

В даному розділі було проведено ряд експериментів та отримано такі результати:

- обґрунтовано доцільність попередньої обробки вхідної вибірки за методом, описаним в розділі 3.1.3, та визначено мінімальну кількість атрибутів, при якій можна знехтувати одним, що відповідає нерівності (3.9) без втрати точності класифікації;
- визначення мінімальне значення кількості повторів без покращень Q, за якого не падає якість кластеризації;
- наведено порівняльні гістограми якості розбиття різних наборів даних методом k-середніх та модифікованим алгоритмом;
- наведено порівняльні гістограми точності класифікації різними алгоритмами (в тому числі, модифікованим алгоритмом).

Було визначено, що модифікований алгоритм кластеризації дозволяє досягти вищої якості розбиття, ніж метод k-середніх в середньому на 7%.

Для задачі класифікації було визначено, що доцільність вибору методу класифікації залежить від характеристик наборів даних. При побудові навчальної вибірки (розділення об’єктів на класи) за допомогою модифікованого методу кластеризації, можна досягти вищої точності використовуючи його і для класифікації наступних об’єктів. Однак, цей результат досягається до тих пір, поки навчальна вибірка значно перевищує тестову, в іншому випадку, набір даних вимагає перекластеризації.

6 РОЗРОБКА СТАРТАП-ПРОЕКТУ

6.1 Резюме стратапу

6.1.1 *Назва проекту*

Дана стартап-розробка називається WingX-Ray і являє собою мобільний комплекс для цифрового радіологічного неруйнівного контролю в авіації.

6.1.2 *Ідея проекту*

Дослідження в авіаційній галузі. Практичне рішення за напрямком діагностики неруйнівного контролю – контролю властивостей та параметрів об’єкта, не руйнуючи його та при якому не повинна бути порушена здатність об’єкта до використання та експлуатації.

Суть технології в побудові з серії шарів (рентгенівських знімків) 3D-моделі об’єкта дослідження.

6.1.3 *Партнери проекту*

Партнери та учасники проекту:

- Згуровський М. З. – ректор Національного технічного університету України «Київський політехнічний інститут», доктор технічних наук, голова наглядової ради “Укроборонпром”, академік НАН України, Академії педагогічних наук України;
- Малюкова І. Г. – керівник Стартап Школи “Sikorsky Challenge”, кандидат технічних наук;
- Мірошниченко С. І. – професор, доктор технічних наук, академік АН Вищої освіти України, генеральний директор Науково-виробничого об’єднання “Телеоптик”;
- Новіков Ю. Л. – керівник проекту, кандидат технічних наук;

6.1.4 *Технічні партнери проекту*

Напрямами розвитку та технічними партнерами проекту є:

а) авіація;

1) Boeing [137];

2) Progresstech [139];

3) Антонов [132];

4) Телеоптик;

б) військовий сектор;

1) Укроборонпром [135];

2) Артем [132];

в) медицина;

г) ветеринарія;

д) геологічні розвідки.

6.2 Технологічний аудит ідеї проекту

6.2.1 *Розвиток напрямку*

Починаючи з першого рентгенографічного приймача фірми “SwissRay”, де зображення формувалося чотирма матрицями приборів із зарядовим зв’язком (ПЗЗ), конструкції з багатьма сенсорами ввійшли в техніку цифрових приймачів. Використання багатьох сенсорів (сенсор-SA), які формують часткові (парціальні) зображення, дозволили отримати відносно недорогі та оптимізовані по параметрам рентгенографічні, мамографічні і рентгеноскопічні приймачі. Типова конструкція SA-приймача зображена на рисунку 6.1.

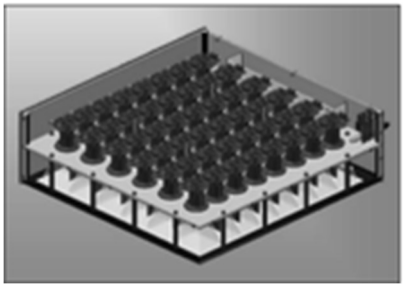

Рисунок 6.1 – Конструкція SA-приймача, пошарово, зверху вниз: поле сенсорів, об’єктиви, несуча панель, бленди, перетворювальний екран, дека

В той же час, доля SA-структур на світовому ринку відносно невелика – не більше 10%. Це зумовлено проявом в SA-структурах перешкод багатоканальності.

Приймачі в SA-структурах були розроблені в 90-х роках XX століття. До цього часу сформувалося три типи конструкцій рентгенівських приймачів.

Хронологічно першими були приймачі на надвеликих ПЗЗ-матрицях. На їх основі почався розвиток цифрових систем для загальної рентгенографії та систем дослідження органів грудної клітини. приймачі цього напряму відрізняються простотою, однак, мають великі габарити, масу, достатньо низьку чутливість. Останнє зумовлено складністю виготовлення світлосильних об’єктивів для ПЗЗ-матриць великих розмірів. Роздільна здатність приймачів першого покоління була невисока та складала 2.5 ... 3 пар ліній на мм (п.л./мм).

На фоні виділених недоліків першого покоління приймачів на ПЗЗ-матрицях, проект, що передбачав створення приймачів у вигляді плоских панелей (Flat Panel Detectors – FPD) здавався ідеальним рішенням. Таким чином, плоскі панелі стали другим направленням в побудові цифрових рентгенівських приймачів.

Розробка пласких панелей вимагала інвестицій на рівні $100 млн. доларів для кожного з проектів. Самі панелі мали роздільну здатність 3.5 ... 3.6 п.л./мм та високу квантову ефективність виявлення на низьких частотах. На жаль, обидва проекти мали серйозне внутрішнє протиріччя – невідповідність можливостей інтегральної технології (виготовлення сотень тисяч виробів у рік) та відносно малий об’єм замовлення. Це призвело до високої вартості панелі та низькому темпі розробок. До того ж, значні внутрішні шуми, наявність дефектних ділянок та схильність до

деградації ускладнили експлуатацію панелей. Незважаючи на відмічені недоліки, цифрові приймачі на основі пласких панелей, починаючи з 2005 року, є лідерами за продажами на світовому ринку [36].

6.2.2 *Порівняння з аналогами*

У 1997 році на виставці RSNA на стенді фірми Caresbuilt (США) були продемонстрована дослідна модель приймача для рентгенографії з полем 43х43 см та вдвічі більшою (в порівнянні з пласкими панелями) роздільною здатністю – 7 п.л./мм. сукупна цифрова матриця приймача містила 6kx6k пікселів. Сумарна площа кремнію в приймачі складала 60х55 мм, а товщина приймача – близько 10 см.

Рекордні параметри приймача стали можливі завдяки застосуванню 99-матричної SA-конструкції, розробленої співробітниками компанії “Телеоптік”. В приймачі вперше було використане автокалібруванню по ділянкам полів зору сенсорів, що перекриваються. В такій технології на зібраному зображенні присутні всі точки вихідного рентгенівського зображення, а границі парціальних зображень візуально не виявляються.

Рекордні характеристики демонстрував і мамографічний приймач фірми Bennett. Приймач складався з механічних щільно підігнаних волоконно-оптичних неоднорідних кабелів (фоконів), де великі торці примикали до перетворюючого екрану, а на менші торці були підклеєні матриці ПЗЗ. Наявність фоконів давала гарний оптичний зв’язок між перетворюючим екраном та матрицею ПЗЗ. Роздільна здатність складала 12 п. л./мм, що до сих пір є найкращим результатом для цифрових мамографічних приймачів.

Основними недоліками такої SA-конструкції були: втрата частини вихідного рентгенівського зображення на границях полів фоконів та відсутність перекриття полів матриць, що виключало автокалібрування. В результаті, через температурні дрейфи сенсорів, їх коефіцієнти передачі змінювалися. Як наслідок, на зображенні з’являлася перебивка у вигляді “шаховооїдошки”. Невдалість такої SA-конструкції

породила представлення про складність та неефективність алгоритмів зклеювання парціальних зображень у багатосенсорних структурах.

На практиці можливість ефективного приглушення перебивок багатоканальності підтверджено успішною експлуатацією більш ніж 2 тисяч цифрових багатосенсорних приймачів з калібруванням протягом останніх 10 років [36].

6.2.3 *Просторова модель та подавлення перебивок багатоканальності*

В 1996 році був запропонований підхід до побудови рентгено-телевізійної системи з високою роздільною здатністю. Він передбачає використання оптичного тракту на основі комбінації “екран-масив” оптичних каналів, що складаються з об’єктивів та сенсорів. Відмінністю запропонованого підходу є процедура відновлення зображення при об’єднанні парціальних зображень з виходів оптичних каналів, що забезпечує відсутність втрати інформації на границі полів зору окремих сенсорів усунення геометричних та яскравісних спотворень зображень в кожному з каналів.

Коректне рішення задачі повного відновлення зображення на основі парціальних зображень передбачає знання кореляційних зв’язків на границях парціальних полів зору. в підході, що розглядається, інформація про кореляційні зв’язки забезпечується шляхом перекриття парціальних полів зору. При цьому, кожне з зображень може бути використане для калібровки системи. При перекритті парціальних зображень створюються умови для автокалібрування, тобто, для зклеювання по інформації в зонах перекриття поточного зображення.

Приймачі, що мають автокалібровку, безперечно, виграють в стійкості до змін зовнішніх умов та дрейфів параметрів сенсорів.

Навпаки, коли в таких випадках конструкція приймача не дозволяє безпосередньо змінити зв’язки між сигналами на границі парціальних зображень, а, тим паче, коли на границях частина пікселів повного зображення втрачається, ситуація стає принципово іншою. Необхідно по тестовим зображенням отримувати інформацію про кореляцію парціальних зображень. За допомогою такої інформації

можливе відновлення повного зображення тільки при відсутності дрейфів параметрів сенсорів. В цілому, рішення задачі відновлення повного зображення по парціальним складовим в цьому випадку некоректне. При відсутності перекриття парціальних зображень існує ризик спотворення зображення в процесі склеювання [36] .

6.2.4 *Склеювання зображення по калібрувальним тестам*

Модель спотворень зображень в SA-системах може бути представлена двома послідовними ланками. Перша описує просторові дисторсії парціальних зображень $G_S\{X, Y, k\}$, а друга – поелементні ефекти, що пов'язані зі зміною інтенсивності сигналу яскравості $G_B\{X, Y, k, t, B\}$. Загальний вираз для оператора спотворення :

$$G\{X, Y, k\} = G_S\{X, Y, k\} G_B\{X, Y, k, t, B\}, \qquad (6.1)$$

де X, Y – просторові координати;

t – час;

k – номер оптичного каналу;

B - інтенсивність сигналу яскравості.

Задача відновлення зображення полягає в знаходженні зворотного перетворення, що описує усунення спотворення (6.1), що вносяться в процесі формування зображення:

$$B'\{X, Y, t\} = G^{-1}\{X, Y, k, t, B(X, Y, t, k)\}, \qquad (6.2)$$

де $B'\{X, Y, t\}$ - відновлене зображення.

При цьому, ціллю відновлення є отримання зображення, придатного для наступного аналізу оператором.

Спотворення зображень можна розділити на детерміновані в часі та ті, що мають залежність від часової координати. До першого типу відносяться геометричні, а також поелементні яскравісні спотворення, що описують світлові характеристики окремих пікселів оптичного каналу.

Просторові (геометричні) спотворення зумовлені наступними факторами: геометричною дисторсією об'єктива, неперпендикулярністю оптичної осі об'єктива та площини сенсора, геометричною неоднорідністю площини перетворювального

екрану (джерела світла). Для знаходження зворотного оператора $G_S^{-1}\{X, Y, k\}$. Було запропоновано використовувати метод апріорного моделювання на основі аналізу спотворень тест-об'єктів. В якості тест-об'єкта була використана лінійна сітка, причому кожен лінійний відрізок всередині тесту спостерігається як мінімум двома сусідніми сенсорами. Схема спотворення зображення клітини сітки зображена на рисунку 6.2.

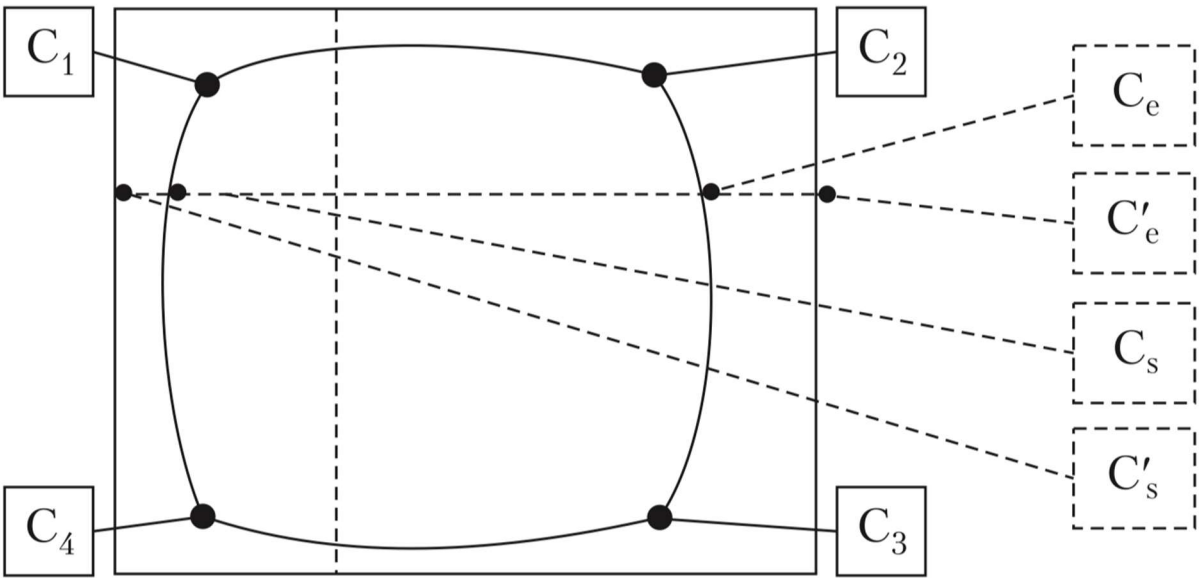

Рисунок 6.2 – Спотворення зображення клітини сітки

Процедура корекції полягає у відновленні вихідної прямокутної форми сітки.

Перетворення координат зображень (корекція) по горизонталі виконується відповідно зі співвідношенням:

$$C' = C_S' + C\frac{C_e' - C_S'}{C_e - C_S}, \tag{6.3}$$

де C - координати $X(Y)$ пікселів на вхідному зображенні;

C' - координати пікселя вихідного скоректованого зображення.

Аналогічно виконується корекція по вертикалі.

Якість відновлення геометричних спотворень зображення визначається точністю визначення координат елементів тест-об'єкта (ліній або точок сітки). Для підвищення точності відновлення оператора геометричних спотворень виконується поліноміальна апроксимація координат центрів ліній або точок сітки, обчислених за алгоритмом пошуку центра мас.

Другий тип детермінованих в часі спотворень зображень $G_B\{X, Y, k, t, B\}$ пов'язаний з їх яскравістю. Він зумовлений неідентичністю коефіцієнтів пропуску об'єктивів по полю, світлових характеристик сенсорів та постійною складовою коефіцієнтів посилення сигналу з виходу матриць.

Задача відновлення виду оператора $G_B\{X, Y, k, t, B\}$ вирішується на основі аналізу серій зображень чистого поля, при цьому тестові зображення являють собою зображення, отримані для різних значень рентгенівських доз при відсутності об'єктів в межах поля огляду.

Корекція яскравісних спотворень зображення виконується у відповідності з виразом:

$$B_{\text{ВИХ}}(X, Y) = \left(B_{\text{ВХ}}(X, Y) - C_{F1}(X, Y, n)\right) C_{F2}(X, Y, n) + B_{\text{ОП}}(n), \quad (6.4)$$

де C_{F1} - відступ n-го тесту;

C_{F2} - нахил лінії n-ого тесту;

n - номер тесту;

$B_{\text{ОП}}(n)$ - значення опорної яскравості для корекції n-го тесту;

$B_{\text{ВХ}}(X, Y)$ - вхідна яскравість;

$B_{\text{ВИХ}}(X, Y)$ - вихідна яскравість.

Для поля сенсорів процедура корекції реалізується наступним чином: за максимальною яскравістю на зображеннях тест-об'єктів обирається опорна точка. Задачею виконання перетворення яскравості є вирівнювання тестових зображень до яскравості опорної точки. Вхідна яскравість зображення кожного сенсору перераховується за формулою (6.4), утворюючи для кожного тесту рівнояскраве поле [36].

6.2.5 *Автокалібрування*

Описана в розділі 6.2.4 процедура яскравісної корекції зображень за калібрувальними тестами, є коректною за умови, що неідентичність пропускання об'єктивів по полю, світлові характеристики сенсорів та коефінієнти посилення

сигналу з виходу матриць не змінюються з часом. На жаль, на практиці це не виконується. В результаті яскравісна корекція з часом, а також при зміні температури сенсора погіршується. Доводиться заново формувати калібрувальні дані або використовувати більш ефективну обробку рентгенівських зображень (автокалібрування).

Вид оператора, що характеризує часову мінливість параметрів сигналів оптичних каналів, для поточного зображення можна відновити на основі методу апостеріорного моделювання при аналізі параметрів зображень сусідніх оптичних каналів в областях, що перекриваються. Параметри корекції спотворень обчислюються шляхом оцінок статистичних характеристик зображень в зонах перекриття.

Наявність зон перекриття дозволяє вирішити задачу відновлення єдиного зображення з відновлених парціальних зображень на виході оптичних каналів та усунення розривів в точках стикування парціальних зображень. Для цих цілей використовується локальна просторова фільтрація в межах областей, що перекриваються, при цьому враховуються особливості зорової системи оператора, який виконує наступний аналіз зображення. Яскравість в одних і тих самих точках в зонах перекриття, що спостерігається в різних каналах – прирівнюються, в результаті зображень в сусідніх каналах – підтягуються.

У відповідності з записаною вище процедурою будується обчислювальний процес. При цьому помилка в відновленні геометричних спотворень не перевищує ¼ пікселя, а похибка у відновленні яскравості не перевищує 0.1-0.5%, що робить границі зображень в сусідніх каналах візуально не спостережуваними. Обробка виконується з темпом у 5 мільйони пікселів у секунду. Час виконання корекції для одного рентгенівського зображення на серійному персональному комп’ютері складає 2-3 секунди [36].

6.3 Застосування результатів досліджень, проведених в дисертації, для побудови схеми розміщення сенсорів

Для аналізу щільності перебивок необхідно враховувати наступні особливості результатів роботи сенсорів:

- кількість гарячих пікселів (при високій, низькій яскравості та у темряві);
- кількість битих пікселів (при високій, низькій яскравості та у темряві);
- рівень шуму на довгих витримках;
- рівень шуму при використанні різної оптики.

Перелічені характеристики впливають на класифікацію сенсорів за впливом на спотворення зображень та дозволяють провести власний аналіз постачальних матеріалів, а також є етапом попередньої обробки вхідних даних для побудови схеми розміщення сенсорів в SA-приймачі.

На основі класифікаційних даних та загального плану розміщення сенсорів в SA-приймачі можливо змоделювати схему розташування сенсорів на несучій панелі шляхом переставляння та обертання сенсорів таким чином, щоб мінімізувати сумарні перебивки областей, що склеюються.

Задачі класифікації сенсорів та побудови схеми їх взаємного розташування вирішуються на основі досліджень, проведених в даній дисертаційній роботі, що описують методи кластеризації та класифікації даних.

6.4 Аналіз ринкових можливостей запуску стартап-проекту

6.4.1 *Поточний стан ринку продукту*

Проблематика ринку NDT (Nondestructive testing – неруйнуючий контроль) є наступною:

- неможливість якісної діагностики вузлів з композитних матеріалів та збірних приладів;
- неможливість якісної мобільної діагностики ключових вузлів повітряних суден в зонах конфліктів та стихійних лих;

- неможливість проведення динамічного контролю пристроїв в недоступних частинах фюзеляжу;
- високі утримання по доставці з повітряних суден на спеціалізовані точки проходження різних видів планового контролю та техобслуговування.

6.4.2 *Рішення*

Адаптувати для ринку NDT в авіації існуючий прототип, що дозволяє здійснювати:

- глибинне пошарове дослідження – томосинтез;
- дослідження вузлів з композитних матеріалів та складних механізмів у збірці;
- динамічне дослідження в реальному часі;
- отримання результатів в цифровому форматі з високою чіткістю.

Існуючий прототип повинен володіти наступними експлуатаційними перевагами:

- мобільність;
- можливість використання будь-яких джерел енергії;
- працеспроможність в складних кліматичних умовах;
- стійкість до зовнішніх факторів: удари, волога;
- модульна конструкція.

6.5 Розроблення ринкової стратегії

6.5.1 *Історія впровадження*

У 2004 році об’єм продажів приймачів по SA-технології склав 4%, а в 2012 році - близько 5% світового об’єму виробництва всіх видів цифрових приймачів.

Популярність багатосенсорних приймачів з автокалібровкою з кожним роком зростає в зв’язку з високими техніко-економічними та експлуатаційними характеристиками. На даному етапі однією з останніх розробок SA-приймача є SA-

приймач “ІОНА-Р-4343”, що має роздільну здатність 4.6 п.л./мм (найвищу з аналогів) та в середньому втричі меншу товщину – 22 см та масу – менше 25 кг. Завдяки цьому даний приймач має гарну сумісність зі штативами практично будь-яких рентгено-діагностичних комплексів.

6.5.2 *Ринкова стратегія*

Загальний об’єм ринку NDT в авіації – $608.25 млн.

Очікується щорічний ріст у розмірі 9.15%.

Об’єм ринку пристроїв NDT – $243.3 млн.

Найбільш перспективні сегменти ринку:

- Латинська Америка - $15.8 млн.;
- Середній Схід та Африка - $20.7 млн.

Оціночна потреба – 3000 пристроїв на рік.

6.5.3 *Огляд конкурентів*

Порівняння ключових конкурентів з розробленим рішенням наведено нижче у таблиці 6.1.

Таблиця 6.1 – Порівняння ключових конкурентів

Характеристики	WingX-Ray	GE DXR250U-W	VIDISCO FlashX Pro
Діагностика пошкоджень, що є невидимими при рентгенографії (наприклад, тріщини під кутом)	+	-	-
Виявлення глибини пошкоджень	+	-	-
Динамічне дослідження в режимі реального часу	+	-	-
Об’єм даних, що збирається для наступного аналізу	1 Gb	0.02 Gb	0.02 Gb

6.5.4 *Технологічні переваги*

Технологічними перевагами даного рішення є:

- модульна конструкція;
- висока ремонтоспроможність;
- висока роздільна здатність: до 100 мікрон;
- широкий діапазон напруг: 20-270 кВ;
- широкий спектр розмірів полів огляду: від 24x30 см до 43x60 см.

6.6 Прототипи продукту

На рисунку 6.3 зображена установка, що використовується для томосинтезу.

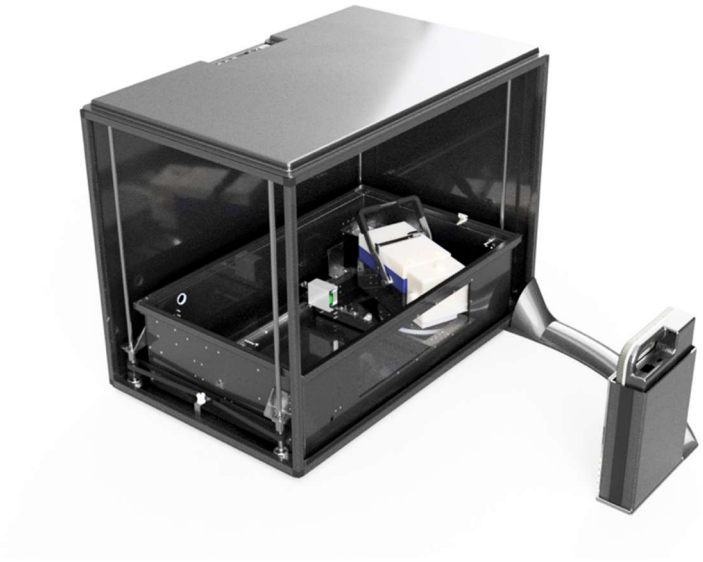

Рисунок 6.3 – Приклад установки, що використовується для томосинтезу

На рисунках 6.4-6.5 зображений процес виявлення пошкоджень лопасті пропелера за допомогою радіографії.

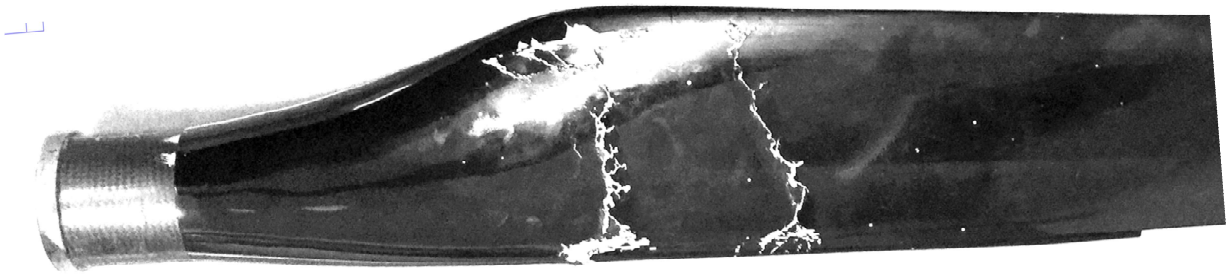

Рисунок 6.4 – Лопасть пропелера, що піддається аналізу

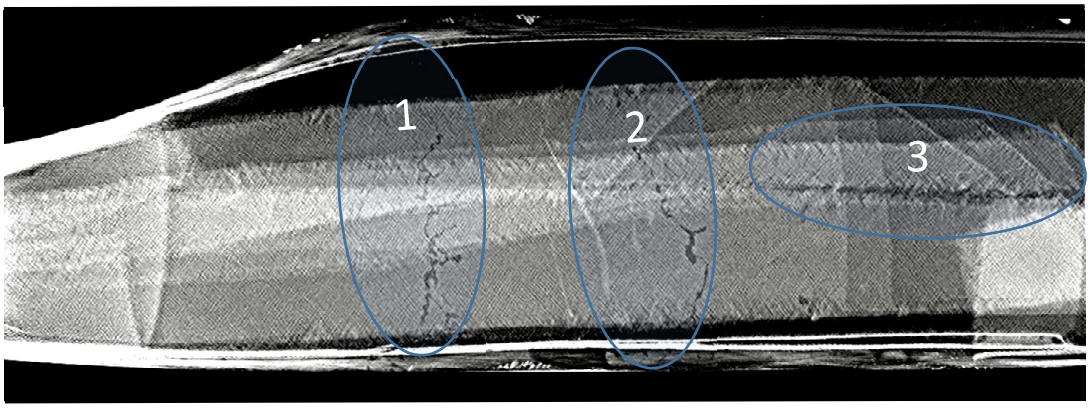

Рисунок 6.5 – Результат радіографічного аналізу

При проведенні радіографічного аналізу були досягнуті такі результати:

- роздільна здатність: 100х100 мікрон;
- час обробки зображення: 5 с.

На рисунку 6.6 зображено результат досліджень тієї ж лопасті пропелера за допомогою томосинтезу.

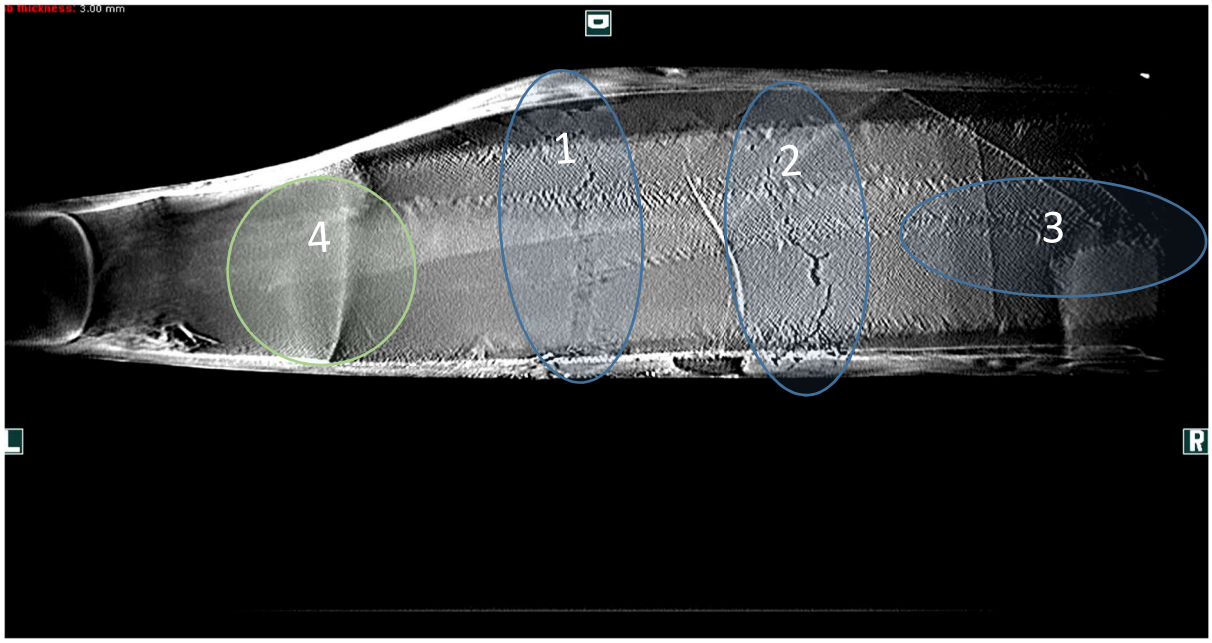

Рисунок 6.6 – Результат аналізу за допомогою томосинтезу

При проведенні аналізу за допомогою томосинтезу були досягнуті такі результати:

- кількість проекцій: 70;
- час обробки однієї проекції: 2.5 с;
- час створення 3-D моделі: 1-2 хв;
- об’ємна роздільна здатність: 250x250x1000 мікрон.

6.7 Побудова бізнес-моделі

6.7.1 *Бізнес-модель*

Модель: виробництво та продаж обладнання.

Канали збуту: канал продажів інвестора, мережі дистриб’юторів та дилерів.

Потенційні клієнти: аеропорти, аеродроми, інжинірингові компанії з обслуговування та техогляду, авіакомпанії.

6.7.2 *Прогноз розвитку бізнесу*

Ринкова вартість комплексу - $115 тис.

Прогноз продажів:

- 1-й рік – створення трьох приладів для введення у дослідну експлуатацію;
- 2-й рік – реалізація 10 приладів;
- 3-й рік та далі – 10-15 та більше приладів у рік.

Ринки:

- в перші роки – Європа, Африка, Південна Америка;
- в подальшому – США, Канада, Азія, тихоокеанський регіон.

Досягнення точки беззбитковості: 2 роки.

Термін окупності проекту: 4 роки.

6.7.3 *Досягнення*

Налаштовано виробництво та продаж мобільних стаціонарних приладів для томосинтезу в сферах медицини та ветеринарії (3-4%).

6.8 Висновки до розділу

Запропонований продукт WingX-Ray являє собою мобільний комплекс для здійснення цифрового радіологічного неруйнівного контролю. Даний продукт може бути використаний для широкого кола потреб у сферах медицини, ветеринарії, геології, авіації.

Було описано технологічні особливості роботи SA-прийомників та описано практичне застосування методів кластеризації та класифікації та використання результатів досліджень для контролю комплектуючих та побудови схем розміщення сенсорів на несучій панелі.

Проект зайняв 2 місце на VI Конкурсі Стартапів “Sikorsky Challenge” 11-12 жовтня 2017 року та був представлений на наглядовій раді “Укроборонпрому” у березні 2018 року.

Також було проведено бізнес-аналіз стартапу, означено його роль на ринку подібних технологій та побудовано прогноз розширення ринку збуту.

В додатку А знаходиться копія сертифікату, що засвідчує проходження навчання у Стартап Школі “Sikorsky Challenge”.

ВИСНОВКИ

Під час розробки магістерської дисертації було досліджено аналоги рекомендаційних сервісів та методів, що вирішують задачі кластеризації та класифікації даних.

Для підвищення якості кластеризації даних було модифіковано метод k-середніх. В результаті було досягнуто таких результатів:

- підвищення якості вхідної вибірки шляхом її попереднього аналізу;
- уникнення випадкової ініціалізації медоїдів, що призводить до виникнення локальних оптимумів;
- уникнення повного перебору об’єктів при перевизначення медоїдів;
- підвищення стійкості до шумів та аномальних значень.

В результаті проведення експериментів було виявлено, що модифікований метод дозволяє отримати кращі результати кластеризації (близько 6%) в порівнянні з методом k-середнім, а також є більш ефективним для застосування в задачах класифікації.

Отримані напрацювання були використані в стартап-проекті WingX-Ray для проектування SA-приймачів. Напрацювання в даній області дозволяють створювати портативні та безпечні прилади, що можуть використовуватися в місцях масового скопичення людей. В секторі медицини розробка дозволяє здійснювати мобільну оцінку пошкоджень, якісну діагностику на місцях, підвищувати рівень життя населення.

На основі класифікаційних даних та загального плану розміщення сенсорів в SA-приймачі можливо змоделювати схему розташування сенсорів на несучій панелі шляхом переставляння та обертання сенсорів таким чином, щоб мінімізувати сумарні перебивки областей, що склеюються.

Задачі класифікації сенсорів та побудови схеми їх взаємного розташування вирішуються на основі досліджень, проведених в даній дисертаційній роботі, що описують методи кластеризації та класифікації даних.

Результати роботи були оприлюднені на

- 4-й міжнародній науково-практична конференція “Актуальні питання сучасної науки”, м. Київ;
- науково-практичній конференції “Інформатика та обчислювальна техніка IOT-2018”, м. Київ, 23-24 квітня 2018 р.;
- VI конкурсі стартапів Sikorsky Challenge, 11-12 жовтня 2017 року;
- наглядовій рада Укроборонпрому, березень 2018 року;
- у збірнику “Управління проектами, системний аналіз та логістика”, Серія “Технічні науки”;
- на 8-й міжнародній науково-технічній конференції “Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління”, м. Харків, 26-27 квітня 2018 року.

ПЕРЕЛІК ПОСИЛАНЬ

1. 6 простых шагов для освоения наивного байесовского алгоритма [Електронний ресурс] / Режим доступу: http://datareview.info/article/6-prostyih-shagov-dlya-osvoeniya-naivnogo-bayesovskogo-algoritma-s-primerom-koda-na-python/.
2. Алгоритмы интеллектуального анализа данных [Електронний ресурс] / Режим доступу: https://tproger.ru/translations/top-10-data-mining-algorithms/
3. Александр Котов. Кластеризация данных [Текст] / Александр Котов // Москва, 2006 – с. 2–16.
4. Антон Коршунов. Анализ социальных сетей: методы и приложения [Текст] /Антон Коршунок, Іван Белобродов// Труди інституту системного програмування (електронний журнал) – 2007 – с.439-456.
5. Антонов [Електронний ресурс] // Режим доступу: http://www.antonov.com
6. Артем [Електронний ресурс] // Режим доступу: http://www.artem.ua/ru/
7. Баргесян А.А. Методы и модели анализа данных: OLAP і Data Mining [Текст] / Баргесян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. // Санкт-Петербург: БХВ-Петербург, 2004 р. – 331 с.
8. В.В. Стрижов. Информационное моделирование. Конспект лекций [Текст] – с.1-6.
9. Воронцов К.В. Лекции по алгоритмам кластеризации и многомерного шкалирования [Текст] – 2007 – с.2-17.
10. Воронцов К.В. Методы коллаборативной фильтрации и тематического моделирования [Текст] / Воронцов К.В. // 2010.
11. Вороцов К.В. Лекции по логическим алгоритмам классификации [Текст] / Воронцов К.В. // 2007 р. – 53 с.
12. Выделение и распознавание лиц [Електронний ресурс] //Режим доступу: http://wiki.technicalvision.ru/index.php/Выделение_и_распознавание_лиц#.D0.9C.D0.B5.D1.82.D0.BE.D0.B4_.D0.B3.D0.BB.D0.B0.D0.B2.D0.BD.D1.8B.D1.85_.D0.BA.D0.BE.D0.BC.D0.BF.D0.BE.D0.BD.D0.B5.D0.BD.D1.82.

13. Гавриленко О.В. Дослідження задачі оцінки об’єктів у комп’ютерних соціальних мережах за допомогою ймовірнісних алгоритмів [Текст] / Гавриленко О.В., Купцова І.В.// “Управління проектами, системний аналіз та логістики”. – Київ, 2018 . – Серія “Технічні науки”
14. Гавриленко О.В. Дослідження задачі оцінки об’єктів у комп’ютерних соціальних мережах за допомогою ймовірнісних алгоритмів [Текст]: матеріали 8-ї міжнародної науково-технічної конференції “Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління”, Харків, 26-27 квітня 2018 р. /Гавриленко О.В., Купцова І.В./. – 108 с. – с.67-68
15. Долгин А.Б. Экономика символического обмена [Текст] / Долгин А.Б. // Москва: Инфра-М, 2006 р. – 632 с.
16. Использование Random Forest (случайного леса) для предсказания цен акций [Електронний ресурс] / Режим доступу: http://rforfinance.ru/random-forest/
17. Кластеризация: метод k-средних [Електронний ресурс] / Режим доступу: http://statistica.ru/theory/klasterizatsiya-metod-k-srednikh/.
18. Колаборативная фильтрация [Електронний ресурс] / Режим доступу: https://ru.wikipedia.org/wiki/Коллаборативная _фильтрация.
19. Коллаборативная фильтрация [Електронний ресурс] // Режим доступу: http://www.machinelearning.ru/wiki/images/archive/9/95/20140413184117%21Voron-ML-CF.pdf
20. Корнілова А. Рекомендаційні системи. Огляд [Електронний ресурс] / Корнілова А. / Режим доступу: http://energyfirefox.blogspot.com/2013/12/blog-post.html.
21. Купцова І.В. Модифікація алгоритмів кластеризації за допомогою обчислення медоїдів кластерів та використання метрики Хемінга [Текст]: матеріали науково практичної конференції “Інформатика та обчислювальна техніка ІОТ-2018”, Київ: НТУУ “КПІ ім. Сікорського”, 23-24 квітня 2018 р. /Купцова І.В., Гавриленко О.В.

22. Купцова І.В. Модифікація алгоритму k-середніх для категоріальних змінних [Текст]: матеріали 4-ї міжнародної науково-практичної конференції “Актуальні питання сучасної науки”, Київ, 16-17 травня 2018 р./ Купцова І.В., Гавриленко О.В.
23. М. Т. Джонс. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы [Електронний ресурс] / М. Т. Джонс // Режим доступу: https://www.ibm.com/developerworks/ru/library/os-recommender1/index.html.
24. Машинное обучение [Електронний ресурс] // Режим доступу: http://www.liquisearch.com/what_is_medoid
25. Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц [Електронний ресурс] // Режим доступу: https://habrahabr.ru/post/133826/.
26. Методы построения деревьев решений в задачах классификации в Data Mining [Електронний ресурс] // Режим доступу: https://ami.nstu.ru/~vms/lecture/data_mining/trees.htm#_Toc123289774)
27. Наивный Байесовский классификатор [Електронний ресурс] // Режим досутупу: http://bazhenov.me/blog/2012/06/11/naive-bayes.html
28. Открытый курс машинного обучения. Тема 5: Композиции: беггинг, случайный лес [Електронний ресурс] // Режим досутупу: https://habrahabr.ru/company/ods/blog/324402/#1-begging
29. Паклин Н.Б. Бизнес-аналитика: от данных к знаниям [Текст] / Паклин Н.Б. // Санкт-Петербург: Питер. 2013 – 502 с.
30. Применение языка Python: преимущества и недостатки [Електронний ресурс] // Режим доступу: http://www.codingclub.net/Articles/Python/Primenenie_yuzika_Python_preimuschestva_i_nedostatki
31. Распознавание эмоций [Електронний ресурс] / Режим доступу: https://monocler.ru/predubezhdeniya-i-vospriyatie/
32. Рекомендательная система [Електронний ресурс] / Режим доступу: https://ru.wikipedia.org/wiki/Рекомендательная_система.

33. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы. [Електронний ресурс] / Режим доступу: https://www.ibm.com/developerworks/ru/library/os-recommender1/.
34. Рекомендательные системы: You can (not) advise [Електронний ресурс] / Режим доступу: https://habrahabr.ru/post/176549/.
35. Ройзнер М. Как работают рекомендательные системы. Лекция в Яндексе [Електронний ресурс] /Ройзнер М./ Режим доступу: https://habrahabr.ru/company/yandex/blog/241455/.
36. С. И. Мирошниченко. Цифровые приемники рентгеновских изображений [Текст] / С. И. Мирошниченко // Киев: издательство “Медицина Украины”, 2014. – 98 с.
37. Сивоголовко Е.В. Методы оценки качества четкой кластеризации [Текст] / Сивоголовко Е.В. // Компьютерные системы в образовании.–2011 – №4 – с.14-30.
38. Сипачев А. Классификация с использованием муравьиного алгоритма [Електронний ресурс] / Режим доступу: https://habrahabr.ru/post/221237/.
39. Социальные сети: современные тенденции и типы использования [Електронний ресурс] / Режим доступу: https://wciom.ru/fileadmin/file/monitoring/2010/99/2010_5(99)_16_Duzhnikova.pdf
40. Телеоптик [Електронний ресурс] // Режим доступу: http://teleoptic-ltd.com
41. Укроборонпром [Електронний ресурс] // Режим доступу: http://ukroboronprom.com.ua/uk/
42. Федоровський А.Н. Архитектура рекомендательной системы, работающей на основе неявных пользовательских оценок [Текст]: матеріали конференції «Russian Conference on Digital Libraries 2011», Вороніж, Росія, 10-22 жовтня 2011 р.: тези доповідей/ А.Н. Федоровський, В.К. Логачева/ [редкол.: Калініченко Л.А. (голова) та інші] – 218 с. – В надзаг.: Вороніж, Росія, 2011 р. – с. 53-59.

43. Фестиваль інноваційних проектів “Sikorsky Challenge” [Електронний ресурс] // Режим доступу: https://www.sikorskychallenge.com/festival/
44. Філіпов С.А. Организация больших обьемов данных в рекомендательных системах поддержки жизнеобеспечения, входящих в состав глобальных платформ электронной коммерции [Текст]: матеріали конференції «Data analytics and management in data intensive domains», Обнінск, Росія, 13-16 жовтня 2015 р.: тези доповідей / С.А. Філіпов, В.Н. Захаров, С.А. Ступников, Д.Ю. Ковалев / [редкол.: Леонід Калініченко (голова) та інші] – 298 с. – В надзаг.: Інститут проблем інформатики ФІЦ ІУ РАН, Москва – 119-124.
45. Цветная типология поведения [Електронний ресурс] / Режим доступу: https://optimizm.co.ua/stati/moya-komanda/cvetnaya-tipologiya-povedeniya
46. Что такое Python [Електронний ресурс] // Режим доступу: https://pythonz.net/promo/
47. Штовба С.Д. Муравьиные алгоритмы [Текст] / Штовба С.Д. // Exponenta Pro – 2003 р. - №4 – с.70-75
48. Эмоционально-экспрессивная окраска слов [Електронний ресурс] / Режим доступу:http://www.fio.ru/pravila/leksika/emotsionalno-ekspressivnaya-okraska-slov/
49. Ю. Лифшиц. Методы распознавания лиц [Текст] матеріали конференції Machine Learning – 2005 – с. 2-6.
50. A. Shekhawat. Ant Colony Optimization Algorithms: Introduction and Beyond [Текст]: матеріали Artificial Intelligence Seminar, Індія, Бомбей, 2009 р: тези доповідей / A. Shekhawat, P. Poddar, D. Boswal – В надзаг. Indian Institute of Technology Bombay.
51. Adolf Proidl. История систем рекомендаций [Електронний ресурс] / Adolf Proidl / Режим доступу: http://www.telemultimedia.ru/art.php?id=582.
52. Aggarwal C.C. Recommender Systems [Текст] / Aggarwal C.C.//Springer, 2016 p. – 493 c.
53. Amazon [Електронний ресурс] / Режим доступу: https://www.amazon.com/

54. Amel'kin S.A. Euclide-Mehalanobis Generalized Distance and Its Properties [Текст] / Amel'kin S.A, Zakharov A.V., and Khachumov V.M., // Inform. Tekhnol. Vych. Sist. .–2006.– № 4.– с. 40–44
55. Anomaly Detection: (Dis-)advatages of k-means clustering [Електронний ресурс] // Режим доступу: https://www.inovex.de/blog/disadvantages-of-k-means-clustering/
56. Apriori – масштабируемый алгоритм ассоциативных правил [Електронний ресурс] / Режим доступу: https://basegroup.ru/community/articles/apriori.
57. Biyun Hu. Data Sparsity: A Key Disadvantage of User-Based Collaborative Filtering? [Текст]: матеріали Asia-Pacific Conference Ap-Web 2012: Web Technologies and Applications, Берлін, 2012 /Biyun Hu, Zhoujun Li, Wenhan Chao // с.602-609
58. Boeing [Електронний ресурс] // Режим доступу: http://www.boeing.com
59. Boyd. D.M. and Ellison N.B. (2007) Social network sites: Definition, history, and scholarship, Journal of computer-mediated communacations, 13(1), article 11.
60. David J. Ketchen. The application of cluster analysis in Strategic Management Research: An analysis and critique [Текст] / David J. Ketchen, Jr; Christopher L. Shook. //Strategic Management Journal.–1996.–№**17** (6).–с.441–458.
61. Facebook [Електронний ресурс] / Режим доступу: https://www.facebook.com/
62. Gediminas Adomavicius. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions [Текст]: матеріали конференції «Artificial intelligence research and development», Нью-Джерсі, США, Червень 2005 р.: тези доповідей/ Gemidinas Adomavicius, Alexander Tuzhilin/ [редкол.: Teresa Alsinet (голова) та інші] - с. 734-749.
63. Gjorka M. et al. Practical recommendations on crawlinng inline social networks // Selected Areas in communications, IEEE Journal on. – 2011 – т.29 - №9 – с.1872-1892
64. Google [Електронний ресурс] / Режим доступу: https://www.google.com

65. Guilermo Santamaria-Bonfil. Measuring the Complexity of Continuous Distributions [Текст] / Guilermo Santamaria-Bonfil, Nelson Fernandez, Carlos Gershenon// Open access entropy, 2015 – с.1-21
66. Ho T.K. The random subspace method for constructing decision forests [Текст] / Ho T.K // Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 1998 – том 20, номер 8 – с.832-844.
67. ID3 Algorithm [Електронний ресурс] // Режим доступу: https://en.wikipedia.org/wiki/ID3_algorithm#Summary
68. Instagram [Електронний ресурс] / Режим доступу: https://www.instagram.com/?hl=ru
69. Jullien Carron. The information content of galaxy surveys [Текст] /Jullien Carron// DISS ETH. NO 20642, 2014 – с.164.
70. K-means Clustering [Електронний ресурс] // Режим доступу: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html
71. Last.fm [Електронний ресурс] / Режим доступу: https://www.last.fm
72. Leo Breiman. Random Forests [Текст] /Leo Breiman// Machine Learning. – 2001 – №45 – с. 5-32
73. Leskovec J., Faloursos C. Sampling from large graphs // Proceeding of the 12 ACM SIGKDD international conference on Knowledge discovery and data mining. – ACM, 2006. – с.631-633
74. Liang Xiang. Recommender system introduction [Електронний ресурс] / Liang Xiang/ Режим доступу: http://www.slideshare.net/xlvector/recommender-system-introduction-12551956.
75. M. J. Paul Viola. Robust Real-Time Face Detection [Текст] / M. J. Paul Viola // International Journal of Computer Vision – 2004 – с. 137-154.
76. M. Walker - Random Forests Algorithm [Електронний ресурс] / Режим доступу: http://www.datasciencecentral.com/profiles/blogs/random-forests-algorithm.
77. MovieLens [Електронний ресурс] / Режим доступу: https://movielens.org/

78. Naive Bayes Classifier [Електронний ресурс] // Режим доступу: http://www.statsoft.com/textbook/naive-bayes-classifier
79. Najork M., Wiener J.L. Breadth-first yields high-quality pages // Proceeding of the 10th international conference on World Wide Web. – ACM, 2001. – с. 114-118
80. Netflix [Електронний ресурс] / Режим доступу: https://www.netflix.com/ua/
81. P.Melwille. Recommender systems [Текст] / P.Melwille, V.Sindwani // Encyclopedia of Machine Learning, 2010.
82. Pandora [Електронний ресурс] / Режим доступу: http://www.pandora.com
83. PostgreSQL [Електронний ресурс] // Режим доступу: http://www.sai.msu.su/~megera/postgres/talks/what_is_postgresql.html
84. PostgreSQL [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/PostgreSQL
85. Prem Melvile Content-Boosted Collaborative Filtering for Improved Reccomendations [Текст]: матеріали конференції «AAAI Conference on Artificial Intelligence», Техас, США, 2002 р.: тези доповідей/ Prem Melvile, Raymond J.Mooney, Ramadass Nagarajan/ [редкол.: Teresa Alsinet (голова) та інші] – 218 с. – В надзаг.: Department of Computer Sciences, University of Texas – с. 187-192.
86. Progresstech Ukraine [Електронний ресурс] // Режим доступу: http://ptu.aero
87. Purnima Bholowalia. EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN [Текст] /Purnima Bholwalia, Arvind Kumar//International Journal of Computer Applications (0975-8887), 2014 - №9 – т.105 – с.17-24
88. Python [Електронний ресурс] // Режим доступу: https://uk.wikipedia.org/wiki/Python
89. Random Forest [Електронний ресурс] // Режим доступу: https://ru.wikipedia.org/wiki/Random_forest
90. Recommendation and Ratings Public Data Sets For Machine Learning [Електронний ресурс] // Режим доступу: https://gist.github.com/entaroadun/1653794

91. Recommender systems [Електронний ресурс] / Режим доступу: http://www.slideshare.net/T212/recommender-systems-1311490/43.
92. Ricci F. Recommender Systems Handbook [Текст] / Ricci F., Rocach L., Shapira B., Kantor P.B. // Лондон: Springer, 2011 p. – 845 c.
93. Sloane N.J.A. Minimal-Energy Clusters of Hard Spheres [Текст] / Sloane N.J.A, Hardin R.H., Duff T.S., Conway J.H // Discrete Comp. Geom. – 1995 – №14 – c.237-259
94. T. Segaran. Programming Collective Intalligence [Текст] / T. Segaran // США, Себастополь: O'Reilly Media, 2007 – c. 368.
95. T.Mitchell. Machine Learning [Текст] /T. Mitchell // McGraw Hill, 1997. – 254 c.
96. Twitter [Електронний ресурс] / Режим доступу: https://twitter.com/?lang=ru
97. U. Sinha. K-Means clustering [Електронний ресурс] / Режим доступу: http://aishack.in/tutorials/kmeans-clustering/
98. UCI. Machine Learning Repository [Електронний ресурс] // Режим доступу: http://archive.ics.uci.edu/ml/index.php
99. X.Su. A survey of Collaborative Filtering Techniques [Текст] / X.Su, T.M.Khoshgoftaar //Advances In Artificial Intelligence, 2009.
100. Yahoo [Електронний ресурс] / Режим доступу: https://www.yahoo.com
101. Yan-Bin Jia. Singular Value Decomposition [Текст] /Yan-Bin Jia// Com S 477/577 Notes. – 2017

Додаток А

Сертифікат навчання у Стартап Школі «Sikorsky Challenge»

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute "

Навчально-методичний комплекс «Інститут післядипломної освіти»
Training and Methodical Complex "Institute for Continuing Education"

«Be Next IT» — Відкрита інноваційна стартап школа-інкубатор (Ізраїль)
"Be Next IT" — Open Innovation Startup School-Incubator (Israel)

СЕРТИФІКАТ
CERTIFICATE

ТРС № 5012

засвідчує, що
this is to certify that

Купцова Ірина
Iryna Kuptsova

з 13 лютого по 20 квітня 2017 року
from February 13 to April 20, 2017

Пройшов(ла) навчання у Стартап Школі «Sikorsky Challenge»
за програмою
Took a course of training in the Startup School
"Sikorsky Challenge" under the program

«Вступ до інноваційного підприємництва та практика запуску стартапа»
"Introduction to Innovative Entrepreneurship and Practice of Launching a Startup"

Директор НМК «Інститут післядипломної освіти»
Director of TMC "Institute for Continuing Education"

І. Малюкова
I. Maliukova

Директор «Be Next IT»
Director of "Be Next IT"

І. Пеер
I.Peer

Додаток Б

Графічні матеріали

ПЛАКАТ 1 Блок-схема модифікованого алгоритму

ПЛАКАТ 2 Схематичний опис кроків модифікованого алгоритму

ПЛАКАТ 3 Опис бази даних

ПЛАКАТ 4 Залежність якості кластеризації від кількості атрибутів, що піддаються аналізу

ПЛАКАТ 5 Залежність якості кластеризації від значення кількості прогонів без покращень

ПЛАКАТ 6 Порівняльний аналіз алгоритмів кластеризації

ПЛАКАТ 7 Порівняльний аналіз алгоритмів класифікації

ПЛАКАТ 8 Порівняння радіологічного аналізу та аналізу за допомогою томосинтезу