

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка
рівень вищої освіти – другий (магістерський)

Аналіз алгоритмів пошуку в структурі даних для створення зручного механізму
опитування користувачів мереж
(тема)

Виконав: студент 2 курсу, групи ІІЗм-18-2
Абраменко Р.О.
(прізвище, ініціали)

Спеціальності 121 –Інженерія програмного забезпечення
(код і повна назва спеціальності)

Освітньо-наукової програми
(тип програми)

Інженерія програмного
забезпечення
(повна назва освітньої програми)

Керівник доц. Ревенчук І.А.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри, проф. _____

З.В.Дудар

2020 р.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20 ____ р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Абраменко Роману Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз алгоритмів пошуку в структурі даних для створення зручного механізму опитування користувачів мереж

затверджена наказом університету від “___” _____ 20 ____ р № _____

2. Термін подання студентом роботи до екзаменаційної комісії _____

3. Вихідні дані до роботи електронні ресурси за обраною тематикою, вимоги до функціональності програми, методи кластеризації даних, пояснювальна записка.

4. Перелік питань, що потрібно опрацювати в роботі: мета роботи, аналіз проблемної галузі і постановка задачі, аналіз алгоритму кластеризації Хамелеон і розробка модифікації даного алгоритму, аналіз методів побудови зручного механізму опитування користувачів мереж, формування вимог до програмної системи та опис програмної системи.

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина			

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка*
1	Аналіз предметної галузі	15 січня 2020 р.	
2	Аналіз алгоритму кластеризації Хамелеон	10 лютого 2020 р.	
3	Розробка модифікації алгоритму Хамелеон	17 лютого 2020 р.	
4	Розробка програмної системи	23 березня 2020 р.	
5	Підготовка пояснювальної записки	30 березня 2020 р.	
6	Спецчастина	13 квітня 2020 р.	
7	Підготовка презентації та доповіді	04 травня 2020 р.	
8	Нормоконтроль, плагіат	11 травня 2020 р.	
9	Рецензування, отримання відзиву керівника, надання сканованих документів, архівування, заповнення анкети випускника	16 травня 2020 р.	
10	Оплата друку у видавництві, предзахист, допуск до захисту, пробне підключення	18 травня 2020 р.	

Дата видачі завдання _____ 2020 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Ревенчук І.А.
(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить: 71 с., 11 рис. 3 табл., 20 джерел.

АНАЛІЗ ДАНИХ, СТРУКТУРА ДАНИХ, АЛГОРИТМ ХАМЕЛЕОН, ЗВ'ЯЗНІСТЬ, КЛАСТЕРИЗАЦІЯ, МЕТОД К-НАЙБЛИЖЧИХ СУСІДІВ, ПОБУДОВА ГРАФА.

Об'єктом дослідження є кластеризація даних.

Метою роботи є аналіз алгоритмів пошуку в структурі даних для створення зручного механізму опитування користувачів мереж та побудова програмної системи за результатами аналізу.

Предметом дослідження є методи кластеризації для аналізу алгоритмів пошуку в структурі даних опитувань.

DATA ANALYSIS, DATA STRUCTURE, CHAMELEON ALGORITHM, CONNECTIVITY, CLUSTERING, K-NEAREST METHODS, BUILDING A GRAPH.

The object of the study is data clustering.

The aim of the work is to analyze the search algorithms in the data structure to create a convenient mechanism for interviewing network users and build a software system based on the results of the analysis.

The subject of the study are clustering methods for the analysis of search algorithms in the structure of survey data.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	7
1.1 Дослідження і аналіз методів інтелектуального аналізу даних	7
1.2 Основні підходи, які використовуються у кластерному аналізі	12
1.3 Вимоги до алгоритмів кластеризації	14
1.4 Актуальність кластерного аналізу та постановка задачі	16
2 АНАЛІЗ АЛГОРИТМУ КЛАСТЕРИЗАЦІЇ ХАМЕЛЕОН І РОЗРОБКА МОДИФІКАЦІЇ ДАНОГО АЛГОРИТМУ	17
2.1 Аналіз і опис основних етапів алгоритму Хамелеон	17
2.2 Аналіз, опис і модифікація етапу початкового поділу графа	20
3 АНАЛІЗ МЕХАНІЗМУ ОПИТУВАННЯ КОРИСТУВАЧІВ МЕРЕЖ	23
3.1 Сильні і слабкі сторони онлайн-опитувань	23
3.2 Аналіз існуючих форм опитувань	29
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	38
4.1 Функціональні вимоги	38
4.2 Нефункціональні вимоги	39
5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	40
5.1 UML проектування програмного забезпечення	40
5.2 Проектування архітектури програмного забезпечення	43
5.3 Проектування бази даних	45
5.4 Приклади найцікавіших алгоритмів та методів	48
5.5 Проектування UI / UX або іншого дизайну системи	51
ВИСНОВКИ	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	56
Додаток А Слайди презентації	59
Додаток Б Лістинг коду	67
Додаток В Апробація результатів роботи	69

ВСТУП

Аналіз даних стає все більш і більш значущим в нашому житті. У різних областях людської діяльності, постійно виникає необхідність вирішення завдань аналізу, прогнозу, діагностики, виявлення прихованих залежностей і підтримки прийняття оптимальних рішень. Актуальність цих завдань обумовлюється бурхливим зростанням обсягу інформації, розвитком технологій її збору, зберігання і організації в базах і сховищах даних, в результаті чого точні методи аналізу інформації та моделювання досліджуваних об'єктів найчастіше відстають від потреб реального життя. На даний момент є актуальною проблема розробки універсальних і надійних методів і підходів, придатних для обробки інформації в області опитувань користувачів. Методологія наукових досліджень може варіюватися, кожне дослідження засноване на даних, які повинні бути хорошої якості, і які потім аналізуються і інтерпретуються для отримання інформації. Відповідно до словника, опитування - це дослідження думок або досвіду групи людей, засноване на серії питань. Питання, які використовуються в опитуванні, як правило спрямовані на отримання конкретних даних від певної групи людей щодо їх переваг, думок, поведінки або фактичної інформації в залежності від мети дослідження.

Метою роботи є аналіз алгоритмів пошуку в структурі даних для створення зручного механізму опитування користувачів мереж та побудова програмної системи за результатами аналізу. Дана робота сконцентована на обробці великого обсягу даних, до яких застосовуються методи кластерного аналізу для визначення кластерів.

В роботі проведений аналіз предметної галузі, досліджені основні методи, підходи, вимоги до алгоритмів кластеризації даних, побудова модифікованого алгоритму Хамелеон та використання його для аналізу даних при опитуванні користувачів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Дослідження і аналіз методів інтелектуального аналізу даних

Інтелектуальний аналіз даних – це процес виявлення раніше необроблених невідомих, нетривіальних, практично корисних і доступних для інтерпретації знань, необхідних для прийняття рішень у різних галузях людської діяльності.

Методи Data Mining поділяються на статистичні (дескриптивний аналіз, кореляційний та регресивний аналіз, факторний аналіз, дисперсійний аналіз, компонентний аналіз, дискримінантний аналіз, аналіз тимчасових рядів) та кібернетичні (штучні нейронні мережі, еволюційне програмування, генетичні алгоритми, асоціативна пам'ять, нечітка логіка, дерева рішень, системи обробки експертних знань).

Кластерний аналіз (Data clustering) – це завдання розділити заданий зразок об'єктів на підмножини, що називаються кластерами, так, що кожен кластер складається з подібних об'єктів, а об'єкти різних кластерів значно відрізняються. Завдання кластеризації стосується статистичної обробки, а також до широкого класу завдань навчання без вчителя. Кластерний аналіз – це багатовимірна статистична процедура, яка збирає дані, що містять інформацію про вибірку об'єктів, а потім упорядковує об'єкти у відносно однорідні групи (кластеризація Q або техніка Q, що є самим кластерним аналізом) [1].

Кластерний аналіз – це спосіб групування багатовимірних об'єктів на основі результатів окремих спостережень за точками відповідного геометричного простору з подальшим виділенням груп як «згустків» цих точок (кластери, таксони). Скупчення з англійської означає "згусток", "гроно винограду", "скупчення зірок". Цей метод дослідження отримав розвиток в останні роки завдяки здатності комп'ютерів обробляти великі бази даних. Кластер – це група елементів, що характеризується загальною властивістю. Основна мета кластерного аналізу – знайти групи подібних об'єктів у вибірці [2].

Кластерний аналіз передбачає відбір компактних, віддалених груп об'єктів, знаходження «природного» розбиття сукупності в області об'єктного кластера. Він використовується, коли вихідні дані представлені як матриця близькості або відстані між об'єктами, або як точки у багатовимірному просторі [3]. Найпоширеніші дані другого роду, для яких кластерний аналіз орієнтований на відбір деяких геометрично видалених груп, усередині яких об'єкти розташовані близько [2].

Спектр використання кластерного аналізу дуже великий і тому існує величезна кількість різноманітних даних з різною структурою, смисловим навантаженням, значенням. Але багатофункціональність використання призвела до появи значної кількості розрізнених визначень, методів і підходів, які ускладнюють єдине застосування послідовного тлумачення кластерного аналізу.

Можна відзначити відповідні завдання кластерного аналізу:

- розробка типології чи класифікації;
- вивчення корисних концептуальних схем для групування об'єктів;
- генерація гіпотез на основі дослідження даних;
- тестування гіпотез або досліджень, щоб визначити, типи (групи) насправді є у наявних даних.

Незалежно від об'єкта дослідження, використання кластерного аналізу передбачає відповідні етапи:

- підбір колекцій з метою кластеризації;
- встановлення великої кількості змінних, відповідно до яких об'єкти у вибірці будуть оцінюватися;
- обчислення значень того чи іншого рівня подібності між об'єктами;
- використання методу кластерного аналізу з метою формування груп подібних об'єктів;
- контроль достовірності результатів.

Кластерний аналіз пред'являє наступні вимоги до даних:

- показники не потрібно співвідносити між собою;
- показники повинні бути безрозмірними;

- розподіл показників має бути близьким до нормального;
- показники необхідні для задоволення умов стійкості, що означає відсутність впливу на їх значення несподіваних факторів;
- вибірка має бути однорідна.

У випадку, якщо кластерному аналізу передують факторний аналіз, то в цьому випадку для вибірки не потрібне втручання – описані умови виконуються автоматично факторною операцією прогнозування (якщо Z-стандартизація виконується за відсутності негативних результатів для вибірки безпосередньо для кластерного аналізу, вона може спричинити зменшення чіткості поділу груп). В іншому випадку вибірку потрібно коригувати.

Мета кластеризації:

- представлення даних шляхом розкриття структури кластера. Поділ вибірки на категорії подібних об'єктів дає можливість полегшити подальшу обробку інформації, а також прийняття висновків, використовуючи власний спосіб розгляду будь-якого кластеру;
- стиснення даних. У випадку, якщо початкова вибірка є надмірно великою, в цьому випадку можна зменшити її, зберігаючи по одному звичайному представнику від будь-якого кластеру;
- виявлення новизни. Нехарактерні об'єкти, які неможливо додати до кластеру жодним чином.

У початковому випадку кількість кластерів, як правило, менше. У другому випадку важливіше гарантувати значний рівень подібності об'єктів всередині будь-якого кластеру, але кількість кластерів необмежена. У третьому випадку інтерес представляють поодинокі об'єкти, які не вписуються у будь-який з кластерів.

Абсолютно у всіх цих варіантах може використовуватися ієрархічна кластеризація, якщо великі кластери поділяються на найменші, а ті в свою чергу ще на менші. Подібні завдання називаються завданнями таксономії.

Результатом таксономії вважається деревоподібна ієрархічна структура. Наявність будь-якого предмета характеризується перерахуванням абсолютно всіх кластерів, до яких він належить, як правило, від великих до малих.

Кластеризація (навчання без вчителя) відрізняється систематизацією (навчання з учителем) тим, що мітки початкових об'єктів спочатку не були встановлені, але крім того множини може не бути.

Рішення завдання кластеризації є невизначеним, і тому є кілька причин:

- жодним чином не існує найкращого критерію якості кластеризації. Популярним є декілька евристичних критеріїв, а також декілька алгоритмів, які жодним чином не мають чітко сформульованого критерію, але виконують досить раціональну кластеризацію;
- число кластерів, як правило, невідомо заздалегідь і встановлюється відповідно з деяким суб'єктивним критерієм;
- результат кластеризації суттєво залежить від метрики, вибір якої, як принцип, також суб'єктивний і визначається фахівцем.

Завдання кластерного аналізу (або навчання без вчителя) полягає в наступному. Мається навчальна вибірка $X_\ell = \{x_1, \dots, x_\ell\} \in X$ і функція відстані між об'єктами $\rho(x, x')$. Потрібно розбити вибірку на непересічні підмножини, звані кластерами, так, щоб кожен кластер складався з об'єктів, близьких по метриці ρ , а об'єкти різних кластерів суттєво різнилися. При цьому кожному об'єкту $x_i \in X_\ell$ приписується мітка (число) кластера y_i . Алгоритм кластеризації – це функція $a: X \rightarrow Y$, яка будь-якому об'єкту $x \in X$ ставить у відповідність мітку кластера $y \in Y$. Множина міток Y в деяких випадках відомо заздалегідь, однак частіше ставиться завдання визначити оптимальне число кластерів, з точки зору того чи іншого критерію якості кластеризації [2].

Вирішенням завдання кластерного аналізу є розбиття, що задовольняє деяку умову оптимальності. Цей критерій може являти собою деякий функціонал, що виражає рівні бажаності різних сегментів і угруповань. Цей функціонал часто називають цільовою функцією. Завданням кластерного аналізу є задача оптимізації, тобто знаходження мінімуму цільової функції при деякому заданому наборі обмежень. Прикладом цільової функції може служити, зокрема, сума квадратів внутрішньогрупових відхилень за всіма кластерами [2].

Можна виділити наступні основні етапи кластерного аналізу.

Розвиток концепції нестійкої. Часто необхідно заздалегідь відібрати з вхідної множини змінних найбільш ефективну підсистему (в зарубіжній літературі ця процедура називається *feature selection*). Крім того, у певних завданнях доцільно трансформувати вихідні змінні так, щоб сформувати нові (*feature extraction*). Для цього, щоб виключити домінування нестабільних змінних, проводиться попереднє нормування вихідних змінних.

Встановлення методу обчислення відстані між об'єктами або групами об'єктів. Цей метод необхідний для відображення специфіки практичної задачі. Наприклад, у випадку безперервних змінних може бути задано Евклідову відстань. Щоб усунути ефект потужних лінійних кореляцій серед змінних, застосовують відстань Махаланобіса. Для номінальних змінних може використовуватися відстань Хеммінга. Для груп об'єктів також визначається спосіб знаходження відстані, наприклад, за принципом далекого сусіда, ближнього сусіда та інші. Принцип далекого сусіда доцільно, якщо про це є апіорна інформація, що таксони мають компакту сферичну форму. Принцип ближнього сусіда має сенс використовувати, якщо відомо, що таксони мають усі шанси бути витягнутої форми або концентрично розташовані.

Групування шаблонів. На цьому етапі триває створення груп об'єктів. Поділ на групи може бути суворим (створюється поділ початкової великої кількості об'єктів), а може бути і нечітким (розрахувати ступінь будь-якого об'єкта до груп). Існує величезна різноманітність алгоритмів угруповання.

Подання результатів. Необхідно отримати простий і інформативний опис отриманих кластерів. Часто для цілей такого опису обирають «типовий об'єкт» або визначають набір характеристик, усереднених за групою. Крім того, застосовується опис у вигляді набору таксонів. Під таксоном будемо розуміти частину простору змінних мінімального обсягу, що має певну задану форму і що містить точки відповідної групи.

Встановлення властивості набутого угруповання. Вже після закінчення кластеризації слід переконатись в тому, що сформовані групи насправді

відображають внутрішні закономірності, характерні для вирішення завдання, сприяють досягненню цілей аналізу, допомагають відкрити нові властивості. Існують також більш формальні способи перевірки якості, пов'язані з перебуванням ймовірності випадкового освіти груп, яку можна обчислити в рамках тієї чи іншої моделі розподілу (з перевіркою статистичних гіпотез про однорідність спостережень різних класів), з bootstrap методом, з обчисленням різних показників якості (внутрішньогрупового розподілення, Гудмана та Крускала та інші) [4].

1.2 Основні підходи, які використовуються у кластерному аналізі

На даний момент існує ряд підходів для вирішення завдань кластерного аналізу, заснованих на всіляких уявленнях про завдання, використання додаткової інформації, характерної для будь-якої предметної області. Коротко перерахуємо підходи, які часто використовуються. Слід зазначити, що описана нижче класифікація не вважається чіткою через те, що деякі методи мають усі шанси бути створенні як комбінація різних підходів:

- ймовірнісний підхід. Фактично очікується, що будь-який об'єкт генеральної сукупності належить одному з K класів, однак номери класів напряду не формуються. Об'єкти довільно вибираються із загальної сукупності; в результаті цього змінні, що описують об'єкти, є випадковими. Для кожного класу визначається розподіл ймовірностей заданого сімейства; параметри розподілу невідомі. Існує наявна вибірка досліджень, яка забезпечує реалізацію суміші розподілів;
- підхід, в якому використовується аналогія з центром ваги. Визначається вектор середніх значень показників для кожної групи і це інтерпретуються центром ваги групи. Застосовується критерій внутрішньогрупового розподілення, де k – координата центру ваги K -го кластеру за змінною

$X_j, j = 1, 2, \dots, n; k = 1, 2, \dots, K$. При заданому K оптимальне угруповання відповідає мінімальному значенню критерія;

- підхід, заснований на теорії графів. Більш звичним алгоритмом є алгоритм найкоротшого незамкнутого шляху. По-перше, будується мінімальне остове дерево графа, у якому вершини відповідають об'єктам, а ребра мають довжину, рівну відстані між відповідними об'єктами. Для утворення кластерів від побудованого дерева видаляють ребра найбільшої довжини;
- ієрархічний підхід. Цей напрямок все також містить відношення до теорії графів. Результати угруповання представлені у вигляді дерева угруповання. Алгоритми, засновані на цьому підході, можна розділити на агломеративні (які крок за кроком з'єднують найближчі групи чи об'єкти) та дивізимні (коли всі об'єкти належать до одного кластеру, який на подальших кроках ділиться на менші кластери, в результаті утворюється послідовність груп, що розщеплюються). Групувальні рішення передбачають вкладену ієрархію підгруп;
- підхід, заснований на понятті найближчого сусіда. Об'єднання виконується по черзі шляхом призначення об'єкта кластеру, в якому знаходиться найближчий об'єкт, за умови, що відстань до об'єкта не перевищує заданий поріг. Існують всілякі варіанти визначення відстані; при визначенні міри запобіжної близькості також можна передбачити розташування інших сусідніх точок;
- нечіткі алгоритми кластерного аналізу. Застосовуючи дані алгоритми, очікується, що будь-який кластер дає нечітко велику кількість об'єктів;
- підхід з використанням штучних нейронних мереж, заснований на аналогії з процесами, що відбуваються в біологічних нейронних системах. Відома величезна кількість алгоритмів даного виду. Звичайна архітектура забезпечує одношарову мережу, в якій будь-який нейрон відповідає якомусь кластеру. У процесі вивчення мережі відбувається ітеративна зміна передавальних ваг між вхідними та вихідними вузлами мережі; тим самим

здійснюється пошук оптимального значення критерію угруповання. Ці мережі дозволяють застосовувати паралельні методи обчислення;

- еволюційний підхід. Алгоритми такого роду будуються на аналогії з природною еволюцією. В них використовуються поняття популяції – набір всіляких угруповань (також їх називають хромосомами, за аналогією з відповідними біологічними об'єктами), та еволюційні оператори – процедури, що дозволяють отримати одну або декілька хромосом нащадків від однієї або декількох батьківських хромосом. Цими процедурами є: селекція, рекомбінація і мутація. Генетичний алгоритм здатний здійснювати пошук рішення, що доправляє глобальний мінімум критерієм якості угруповання.

Існують також інші підходи до вирішення завдання глобальної оптимізації, які можуть застосовуватися у кластерному аналізі.

1.3 Вимоги до алгоритмів кластеризації

Кластеризація – це багатообіцяюче поле для досліджень, де області потенційного застосування диктують додаткові вимоги:

- масштабованість. Багато алгоритмів кластеризації добре працюють на маленьких вибірках даних, які складаються з менш ніж 200 об'єктів; тим не менш, великі бази даних можуть містити мільйони об'єктів. Кластеризація вибірки з великої множини може призвести до необ'єктивних результатів. Необхідні добре масштабовані алгоритми;
- можливість роботи з характеристиками різних типів. Багато алгоритмів розроблені для кластеризації числових даних. Тим не менш, може бути затребувана робота з іншими типами даних: бінарних, категоріальних (номінальних) та порядкових даних або суміші цих типів даних;

- розпізнавання кластерів довільної форми. Багато алгоритмів кластеризації визначають кластери, ґрунтуючись на евклідовій та мангеттенській відстанях. Алгоритми, засновані на таких видах відстаней, мають тенденцію до знаходження кластерів сферичної форми з однаковою щільністю і однакового розміру. Однак кластер може бути будь-якої форми. Важливо розробити алгоритм для знаходження кластерів довільної форми;
- мінімальні вимоги до області дослідження для визначення вхідних параметрів. Багатьом алгоритмам кластеризації необхідні певні вхідні параметри (наприклад, такі, як кількість кластерів). Результати кластеризації можуть бути достатньо чутливі до вхідних параметрів. Часто параметри досить важко визначити, особливо якщо вхідна множина складається із багатовимірних об'єктів. Це не тільки вимагає втручання користувача, але і ускладнює контроль за якістю кластеризації;
- нечутливість до порядку записів, що входять. Деякі алгоритми кластеризації чутливі до порядку, у такому випадку для однієї і тієї ж множини, коли об'єкти представлені в різному порядку, будуть отримані зовсім різні класи;
- висока розмірність. База даних чи сховище даних може мати кілька вимірів чи атрибутів. Багато алгоритмів гарні для роботи з низькорозмірними даними;
- кластеризація на основі обмежень. Реальні програми можуть накладати на кластеризацію різні типи обмежень. Багатообіцяючим завданням є знаходження груп даних з хорошим проходженням кластеризації, яка відповідає висунутим обмеженням;
- простота використання та інтерпретації. Користувач очікує інтерпретовані, порівняні і корисні результати кластеризації;
- кількість переглядів бази даних. Наявної пам'яті повинно вистачати на обробку великих множин даних високих розмірностей [3, 5, 6].

1.4 Актуальність кластерного аналізу та постановка задачі

Незважаючи на велику кількість досліджень у галузі кластерного аналізу, у цій галузі існує ряд актуальних проблем. Перелічимо основні проблеми:

- правильність даних. Перевірка даних є важливою частиною будь-якого завдання по обробці даних. Якщо ваші дані з початку не є точними, ваші результати точно також не будуть точними. Ось чому потрібно перевірити ваші дані перед їх використанням;
- різноманітність даних. Проблема різноманітності даних в тому, що вони можуть мати різні типи даних, а також можуть мати синонімічний зміст. Наприклад, на питання закритого типу, що потребує відповіді «так/ні», користувач може відповісти «так» або «+», усе залежить від попереднього досвіду користувача.

На основі виконаного аналізу сформулюємо послідовні завдання для даної дипломної роботи:

- аналіз алгоритму кластеризації Хамелеон;
- реалізувати алгоритм мовою програмування JavaScript;
- реалізувати програмну систему опитувань користувачів мереж;
- використовуючи розроблену програму систему, провести кластеризацію;
- зробити висновки.

2 АНАЛІЗ АЛГОРИТМУ КЛАСТЕРИЗАЦІЇ ХАМЕЛЕОН І РОЗРОБКА МОДИФІКАЦІЇ ДАНОГО АЛГОРИТМУ

2.1 Аналіз і опис основних етапів алгоритму Хамелеон

Проблема поділу графа - це поділ вершин цього графа на p приблизно рівних частин таким чином, щоб число ребер між вершинами різних класів було найменшим. Ця задача використовується в багатьох різних галузях, охоплюючи паралельні наукові обчислення або планування завдань. Проблема поділу вважається NP - повною. Не в останню чергу, величезна кількість створених алгоритмів мають досить непогані результати поділу. Завдання поділу K-Way найчастіше вирішується рекурсивним чином навпіл. Останнім часом було помічено високоефективний метод для поділу графіків K-Way - багаторівнева рекурсивна бісекція. Основна конструкція багаторівневої рекурсивної навпіл є досить простою. На початку граф G стає грубішим до декількох сотень вершин, потім набутий зменшений графік ділиться навпіл, а потім цей поділ виконуються назад до початкового графа шляхом періодичного відновлення поділу.

Багаторівнева парадигма все ще має можливість використовуватися для побудови поділу K-Way безпосередньо на початкову кількість множин [5]. Множина огрубляється послідовно, як і в попередній схемі, але зараз огрублена множина розподіляється на k частин і поділ послідовно відновлюється до початкової множини. Існує ряд переваг у виконанні k -поділу. Для початку огрубіння потрібно опрацювати лише один раз, насправді це зменшує складність алгоритму та час виконання. По-друге, багаторівнева рекурсивна бісекція здатна працювати гірше, ніж поділ K-Way. Таким чином, метод досягнення негайного поділу K-Way має можливість виконати завдання краще. Насправді обчислити хороший поділ K-Way складніше, ніж виконати хорошу бісекцію. Саме з цієї причини більш популярні рішення задачі поділу K-Way виконуються з підтримкою рекурсивної бісекції [6].

На стадії огрубіння величина множини послідовно зменшується, на стадії вихідного розподілу виконується поділ K-Way зменшеної множини (6-Way в даному випадку), на стадії відновлення виконується проектування поділу на початковий графік.

Наприклад, найпростіший метод обчислити початковий поділ у контексті багаторівневого алгоритму - це огрубіння графа до k вершин. Не в останню чергу, на етапі вдосконалення потрібно вдосконалити поділ K-Way, що є складнішим, ніж рекурсивна бісекція. Навіть для 8-Way розподіл часу для запропонованої схеми досить великий, як показано на рисунку 2.1. Щоб покращити поділ K-Way для $k > 8$, час виконання стає занадто великий.

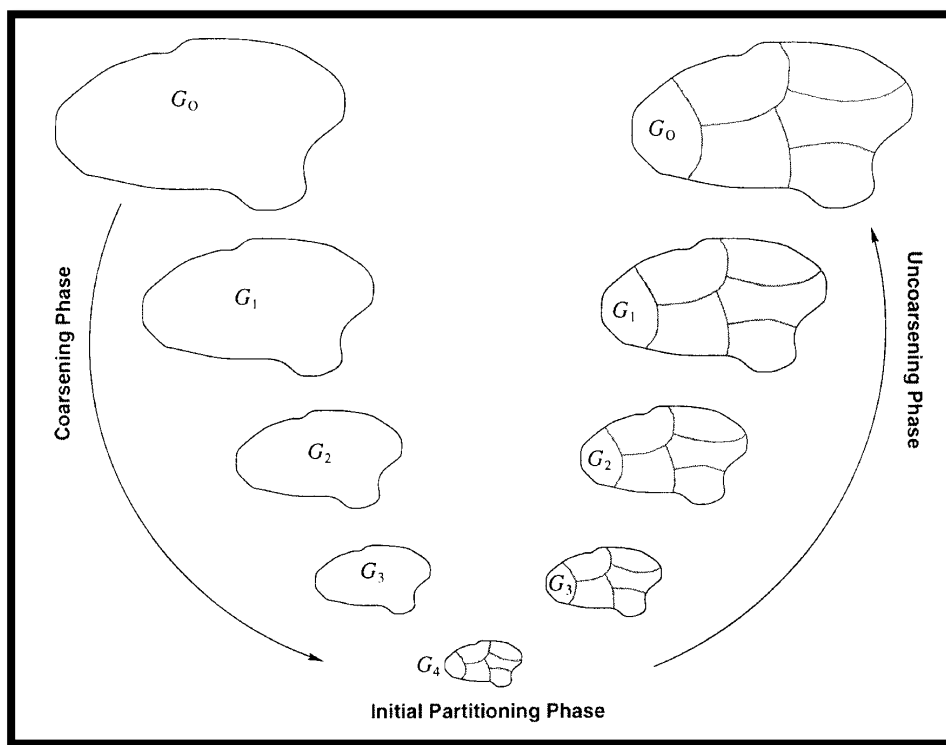


Рисунок 2.1 – Різні стадії алгоритму багаторівневого K-Way поділу

Хамелеон – це новий ієрархічний алгоритм, який долає обмеження існуючих алгоритмів кластеризації. Цей алгоритм оцінює динамічне моделювання в ієрархічній кластеризації. Основним фактором методу Хамелеона вважається той факт, що він забезпечує одночасно взаємозв'язок і близькість при визначенні

подібних пар кластерів. Просто це дозволяє подолати обмеження. Хамелеон користується новим підходом у визначенні ступеня взаємопов'язаності та близькості між парами кластерів. У представленому підході алгоритм сам обчислює внутрішні властивості кластерів, це означає, що вони не залежать від статичних моделей, встановлених користувачем, і мають усі шанси автоматично підлаштовуватися до внутрішніх характеристик об'єднаних кластерів.

Хамелеон знаходить кластери двофазним алгоритмом. На першому кроці Хамелеон використовує алгоритм розбиття графів для кластеризації великих чисел на досить малі підкласи. На другому кроці алгоритм використовується для пошуку природних кластерів за допомогою послідовного поєднання придбаних малих підкласів.

Хамелеон представляє об'єкти за підтримки графіка K найближчих сусідів. Таке подання даних у вигляді графа дозволяє змінювати масштаб великих розмірів даних. Будь-яка вершина у наданому графі представляє єдиний об'єкт даних. Між вершинами є ребро, якщо один об'єкт вважається одним із K найближчих сусідів другого об'єкта. Граф K найближчих сусідів має теорію, що радіус суміжності об'єкта визначає щільність регіону, в якому знаходиться цей об'єкт. Це дозволяє виявляти природні кластери.

Наступним кроком є побудова черги з поперемінно зменшеними гіперграфами. Для огрубіння графів є всі шанси використовувати певну кількість доступних алгоритмів. На будь-якому рівні огрубіння завершується, як тільки значення отриманого огрубіння графа зменшилося в 1,7 рази.

На третьому кроці поділ огрубіння графа K -Way виконується таким чином, щоб було досягнуто обмеження балансу та оптимізовано функцію поділу.

На четвертому кроці виконується відновлення графа. Поділ огрубіння графа проектується на наступний рівень вихідного графа та виконується алгоритм поліпшення поділу для поліпшення цільової функції, не порушуючи обмеження балансу.

На останній ітерації визначається показник схожості між будь-якою парою кластерів з урахуванням їх відносної зв'язаності та близькості. Це дозволяє

вибирати кластери для об'єднання, які добре пов'язані і досить близькі. Вибираючи кластери та виходячи з цих двох критеріїв, Хамелеон долає межі існуючих алгоритмів, що враховують взаємозв'язок чи близькість.

Отже, можна виділити наступні етапи:

- побудова графа. Граф має можливість бути симетричним або асиметричним. При побудові графа можуть використовуватися різні види відстаней;
- огрубіння графа. Огрубіння графа може бути виконано відповідними методами: Heavy Edge Matching (HEM), Random Matching (RM), Light Edge Matching (LEM);
- початковий поділ графа. Існує ряд підходів поділу графів: графічні, комбінаторні та спектральні методи. Більшість методів використовують ділення графа навпіл, так як алгоритми мають змогу працювати рекурсивно навпіл;
- відновлення графа та вдосконалення поділу графа. Для поліпшення поділу графів застосовуються відповідні алгоритми: Fiduccia-Mattheyses (FM), Kernighan–Lin (KL), Boundary KL, Boundary FM. Ці ж алгоритми мають усі шанси використовуватись на етапі поділу, приймаючи за початкове значення поділ огрубленого графа;
- об'єднання подібних класів для отримання остаточного розбиття.

2.2 Аналіз, опис і модифікація етапу початкового поділу графа

Рекурсивне ділення навпіл використовується для відстеження впровадження, що зменшує заповнення під час розкладання розрідженої матриці. Ці алгоритми зазвичай називають алгоритмами вкладених перерізів. Вкладені перерізи рекурсивно ділять граф майже на рівні половини, видаляючи вузли сепаратора, поки потрібна кількість поділів ще не отримана. Спосіб отримання вузлового

сепаратора полягає в отриманні ділення навпіл графа та обчислюванні вузлового сепаратора з реберного сепаратора. Вузли графа нумеруються так, що насправді на будь-якому рівні рекурсії вузли сепаратора нумеруються після вузлів в половині графа. Ефективність та складність алгоритму вкладених перерізів залежить від алгоритму розрахунку сепаратора. Загалом малі сепаратори призводять до меншого наповнення [7].

Для вирішення завдання розбиття графа можна рекурсивно використовувати метод бінарної поділу, при якому граф розділяється на дві рівні частини за першу ітерацію, тоді на другому кроці кожна з набутих частин все ще ділиться на дві частини тощо. У тому випадку, коли важлива кількість сегментів k не є ступенем двійки, будь-який поділ навпіл повинен бути реалізований у відповідному співвідношенні [8].

Процедура рекурсивного поділу навпіл працює для розбиття графа з n вершин на довільне число доменів k . Схема роботи методу поділу навпіл на 5 частин наведена на рисунку 2.2 і звучить так.

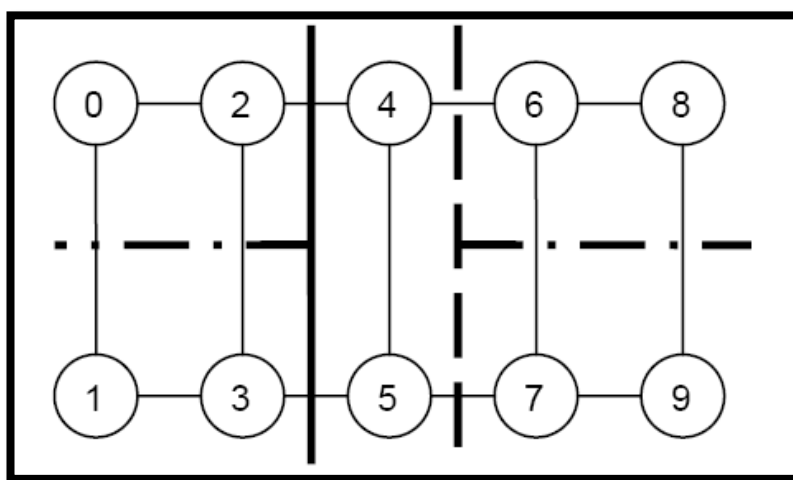


Рисунок 2.2 – Схема роботи методу розподілу навпіл

На початку граф слід розділити на дві частини у співвідношенні 2:3 (на схемі це безперервна лінія), після чого праву частину розбити у співвідношенні 1:3 (це пунктирна лінія в діаграмі), потім залишається розділити останню ліву і праву крайню підобласті у співвідношенні 1:1 (це пунктирна лінія з крапкою) [8].

На рисунку 2.3 показаний результат розбиття 100 вершин графа на 7 частин приблизно рівного розміру. У будь-якому з коренів дерева розрізів призначається кількість вершин відповідного підграфа, два числа, розділені символом риски, показують пропорцію, в якій вершини підграфа діляться на 2 частини [9].

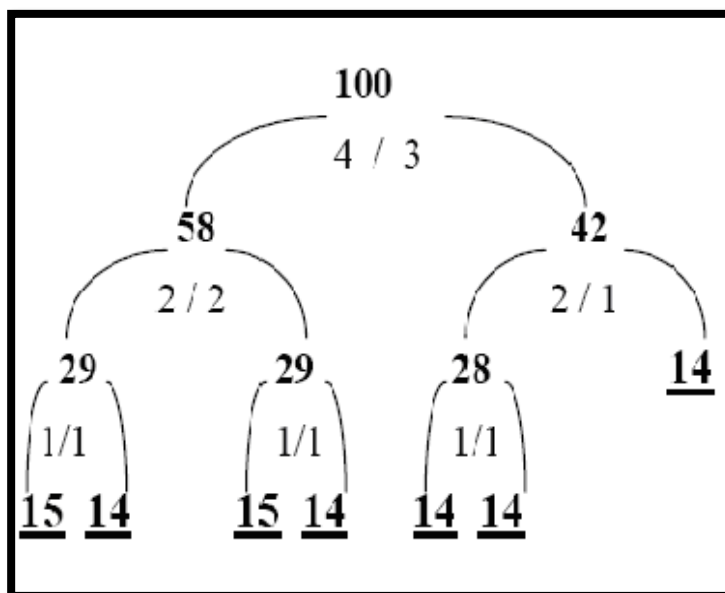


Рисунок 2.3 – Процедура рекурсивного розподілу множини на 7 частин

Метод працює досить швидко і вимагає невеликої кількості оперативної пам'яті. Однак отримане розбиття за якістю поступається більш складним та обчислювально трудомістким методам.

3 АНАЛІЗ МЕХАНІЗМУ ОПИТУВАННЯ КОРИСТУВАЧІВ МЕРЕЖ

Швидкий успіх веб-опитувань як нового способу збору даних з великих підгруп населення супроводжувався збільшенням числа онлайн-керуючих, кожен з яких характеризується різними цілями і функціями. Веб-опитування стають важливою частиною індустрії опитувань. За оцінками асоціації Esomar, в Сполучених Штатах (США) більше третини досліджень ринку в даний час проводиться за допомогою онлайн-опитувань. Веб-опитування представляють у багатьох випадках багатообіцяючі рішення, особливо при опитуванні населення з високим рівнем використання Інтернету; це пов'язано з відносно низькими витратами, швидкістю збору даних і простотою впровадження. Дослідження веб-опитувань - це багатоетапний процес з чітко визначеним протоколом на кожному етапі. Опитування - це «систематичний метод збору інформації від (вибірки) суб'єктів для побудови кількісних дескрипторів атрибутів більшої сукупності, членами якої є суб'єкти».

3.1 Сильні і слабкі сторони онлайн-опитувань

Онлайн-опитування мають численні сильні і слабкі сторони. З сильних сторін можна виділити:

- глобальне охоплення. Вчені прогнозують, що в 2020 році у всьому світі буде 4,5 мільярда Інтернет-користувачів. Global Reach стверджує, що 26 відсотків користувачів є носіями англійської мови, а 74 відсотки - ні. Рівень використання Інтернету найвищий в промислово розвинених країнах, тому в деяких регіонах потенціал для онлайн-опитувань вище. Вчені стверджують, що коли велика частина суспільства має доступ до Інтернету, то зникає основний недолік використання онлайн-опитувань - відсутність

репрезентативності. Інтернет стане ще більш цінним інструментом для отримання інформації від респондентів, що живуть в різних частинах країни або по всьому світу;

- звернення бізнесу до бізнесу і бізнесу до споживача. Онлайн-опитування можуть застосовуватися як в умовах «бізнес-бізнес» (B-to-B), так і «бізнес-споживач» (B-to-C). Література приділяла недостатню увагу онлайн-опитувань B-to-B і фокусувалася на опитуваннях B-to-C. Проте B-to-B пропонує великі можливості. Як зазначає видання Pointer, управління взаємовідносинами з клієнтами (CRM) і маркетингові дослідження (MR) стають ближчими один до одного. У міру подальшого розвитку CRM, можливості для онлайн-досліджень B-to-B будуть рости. Моніторинг в реальному часі, електронні замовлення та електронна відповідність надають нові можливості для дослідників;
- гнучкість. Онлайн-опитування досить гнучкі. Вони можуть проводитися в декількох форматах: електронна пошта з вбудованим опитуванням; електронна пошта з посиланням на URL опитування; відвідування веб-сайту користувачем, якого потім запрошують взяти участь в опитуванні і т.д. Опитування можуть бути у вигляді простого тексту або HTML сторінок. Крім того, їх можна легко адаптувати до демографії клієнтів, мови, досвіду покупок і т.д., маючи кілька версій опитувань;
- швидкість і терміни. Онлайн-опитування можуть проводитися ефективним за часом способом, зводячи до мінімуму час, необхідний для проведення опитування та збору даних. Вчені прийшли до висновку, що швидкість і глобальне охоплення Інтернету забезпечують доступ в реальному часі для взаємодії з географічно різними групами респондентів і інформаційними серверами. Ширококутний доступ до Інтернету також полегшує передачу мультимедійного контенту завдяки швидкості завантаження, що збільшує обсяг і багатство онлайн-опитувань. Ці фактори призвели до появи інноваційних інтернет-технологій таких, як онлайн-фокус-групи, чати і дошки оголошень;

- технологічна інновація. Онлайн-опитування пройшли довгий шлях від простих текстових опитувань по електронній пошті 1980-х років до технологій, доступних сьогодні. Респонденти можуть натиснути URL-адресу, відправлену по електронній пошті, і перейти в багатофункціональний веб-інструмент опитування, який є директивним і потужним, або відповідає безпосередньо на опитування по електронній пошті, вводячи відповіді відповідно до інструкцій. Вчені прийшли до висновку, що Інтернет-технологія не тільки забезпечує більш широкий контроль і гнучкість по відношенню до того, де представлені об'єкти або інформація, але і забезпечуючи таким чином більш складні відображення, ніж це було можливо з паперовими опитуваннями;
- зручність. Онлайн-опитування забезпечують зручність декількома способами. Респонденти можуть відповісти в зручний для себе час. Вони можуть зайняти стільки часу, скільки їм потрібно, щоб відповісти на окремі питання. Деякі онлайн-опитування дозволяють респондентам починати, а потім повертатися до питання, на якому вони зупинилися раніше. Як відзначають вчені, замість того, щоб відповідати в незручний час через телефонного опитування, респондент може пройти онлайн-опитування, коли він або вона вважатиме це зручним;
- простота введення і аналізу даних. Для респондентів відносно не складно заповнити онлайн-опитування. Для компаній, які проводять онлайн-опитування, значна частина адміністративного часу відправлення та отримання анкет, а також введення даних значно знижується;
- різноманітності питань. Веб-опитування можуть включати дихотомічні питання, питання з декількома варіантами відповідей, шкали, питання в мультимедійному форматі, питання як з однією відповіддю, так і з декількома відповідями, і навіть відкриті питання;
- низька вартість адміністрування. Витрати на дослідження можна розділити на дві категорії: підготовка і адміністрування. Що стосується витрат на підготовку, до недавнього часу онлайн опитування могли бути дорогими

через технологічні вимоги. Сьогодні завдяки наявності передового програмного забезпечення для опитувань і спеціалізованих онлайн-форм для розробки опитувань, витрати на підготовку набагато нижче. Що стосується адміністрування опитувань, онлайн-опитування автоматично поміщаються в базу даних, а потім зводяться в таблиці і аналізуються скоординованим, інтегрованим чином, що значно знижує витрати. А оскільки опитування проводяться самостійно і не вимагають поштових витрат або проведення опитувань, затрати також знижуються;

- легкість спостереження. Через низькі витрати на розсилку електронних листів і простоти проведення опитувань в Інтернеті, компанії з більшою ймовірністю будуть відправляти наступні нагадування, і вони можуть робити це по всьому світу, щоб підвищити частоту відповідей на опитування;
- контрольована вибірка. Зразки для онлайн-опитувань можуть бути отримані декількома способами. За допомогою своїх власних баз даних компанії можуть розробляти списки розсилок для своїх клієнтів. Такий підхід дозволяє компаніям не тільки отримувати відгуки клієнтів, але і покращувати їх відносини, показуючи, що вони зацікавлені в думках клієнтів. Компанії також можуть працювати з фірмами з дослідження ринку та отримати доступ до демографічно збалансованим панелям - B-to-B і / або B-to-C;
- контроль порядку відповідей. На відміну від поштових опитувань, онлайн-опитування можуть вимагати, щоб респондент відповідав на питання в порядку, заданому розробником дослідження, а також забороняв респонденту дивитися вперед на наступні питання. Це знижує упередженість дослідження. Однак, вчені звернули увагу, що при проведенні опитування необхідно надати можливість користувачу дізнатися, на скільки ще питань необхідно відповісти;
- контроль завершення відповідей. Онлайн-опитування можуть бути побудовані так, що респондент повинен відповісти на питання, перш ніж

перейти до наступного питання або завершити опитування;

- перейти до можливостей. Онлайн-опитування можуть бути побудовані таким чином, щоб респонденти відповідали тільки на ті питання, які відносяться саме до них, таким чином, адаптуючи опитування;
- знання характеристик респондента і не респондента. Коли компанії використовують свої власні бази даних або онлайн-панелі досліджень ринку, вони отримують вигоду двома способами. По-перше, вони вже знають демографію потенційних респондентів і можуть привласнити їм унікальний ідентифікатор; їм не потрібно запрошувати цю інформацію кожен раз, коли вони проводять опитування. По-друге, оскільки характеристики всіх членів вибірки відомі, можна порівняти демографію респондентів з демографією не респондентів. Це допомагає перевірити результати опитування або попередити компанію про розбіжності.

Якщо не вжити правильного рішення, онлайн-опитування також мають ці потенційні недоліки:

- сприйняття як небажане опитування. Небажане опитування є великою проблемою. Вчені кажуть, що 692 мільйони з 909 мільйонів сканованих повідомлень електронної пошти (76 відсотків), відправлених її клієнтам із США, були помічені як спам. В результаті багато респондентів стикаються з труднощами при проведенні відмінностей між законним опитуванням і спам-повідомленням: «Навіть якщо електронний лист приходить з надійного джерела, малоімовірно, що деякі клієнти натиснуть на посилання, щоб перейти їх на веб-сайт;
- перекіс атрибутів Інтернет-населення. До недавнього часу користувачі Інтернету і електронної пошти не були по-справжньому репрезентативними для населення всього світу. Вчені говорять, що різниця між внутрішнім і онлайн-населенням швидко скорочується і може бути незначною в найближчому майбутньому;
- питання по підбору і впровадженню зразків. Досліджувані несхвально ставляться до деяких методів відбору відомостей. Сильно розкритиковані

методи відбору зразків - це електронна розсилка. Бланкова розсилка часто нагадує спам, коли повідомлення відправляються величезній кількості потенційних респондентів;

- брак досвіду онлайн-опитувань. Незважаючи на те, що Інтернет-популяція стає все більш представницькою, все ж можуть виникнути труднощі з опитуваннями через відсутність обізнаності можливих респондентів;
- технологічне різноманітність. Проблеми можуть виникати через моніторів різних розмірів і налаштувань, з різними операційними системами і одним з багатьох поколінь веб-браузерів. Питання і їх відповіді, які здаються акуратно вирівняними на одному моніторі, можуть бути перекручені і заплутані на іншому моніторі;
- неоднозначні питання. Оскільки онлайн-опитування проводяться самостійно, відповіді та інструкції повинні бути гранично чіткими. Якщо немає, то деякі люди можуть бути розчаровані і покинути опитування, не припиняючи все опитування. Тому при розробці веб-опитувань необхідно обов'язково використовувати технологію для поліпшення методів збору даних, не перевантажуючи респондента;
- безособовість. Як і у випадку з поштовими опитуваннями, в онлайн-опитуваннях зазвичай відсутній людський контакт. Це може обмежити можливість поглибленого дослідження;
- питання конфіденційності. Питання конфіденційності респондентів залишаються важливими. Проблеми діляться на дві категорії: безпека передачі і як будуть використовуватися дані. Стандартні опитування по електронній пошті не мають високого рівня безпеки. Повідомлення можуть бути перехоплені. Крім того, багато респондентів задаються питанням, чи будуть їхні відповіді розглядатися конфіденційно, і чи буде їх контактна інформація продаватися іншим фірмам. Вчені узагальнюють проблему конфіденційності, відзначаючи, що дослідні онлайн-компанії повинні переконати учасників погодитися на опитування і обмін особистими даними. Отже, галузь потребує самореалізації та встановлення жорстких

стандартів конфіденційності; в іншому випадку буде втручання уряду. Що стосується безпеки, багато респондентів не наважуються відкрити вкладення електронної пошти, побоюючись, що вони можуть бути заражені вірусом.

3.2 Аналіз існуючих форм опитувань

У Інтернеті є безліч інструментів онлайн-опитування. Так веб-сайт Capterra - який допомагає бізнес-компаніям визначити правильне програмне забезпечення для своїх організацій - перелічує майже 200 безкоштовних та комерційних веб-інструментів опитування. Дві найпопулярніші програми опитування - Google Forms та SurveyMonkey. Ці веб-інструменти дозволяють створювати опитування, форми та тести, використовуючи різні типи питань, вони задовольняють різні потреби та випадки.

3.2.1 Ціна

Google Forms - це 100% безкоштовно для всіх, хто має обліковий запис Google. Ви можете використовувати його для складання стільки опитувань, скільки вам потрібно, задавати стільки питань, скільки вам потрібно, і збирати відповіді від стільки людей, скільки готових взяти участь у вашому опитуванні, і все, не плативши ні копійки.

У SurveyMonkey також є базовий план, але він набагато скромніший. Хоча ви і можете створювати необмежену кількість опитувань, але ви можете задавати максимум 10 запитань за опитування, і можете бачити максимум 100 відповідей. Щоб задавати необмежену кількість запитань та відповідей за допомогою

SurveyMonkey, вам доведеться оновити обліковий запис до преміального плану. Преміальні плани SurveyMonkey починаються від \$ 32 / місяць для індивідуальних планів та \$ 25 / місяць для командних планів (мінімум три користувача).

Тож якщо ціна є ключовим фактором, і якщо ви думаєте, що вам потрібно буде задати більше 10 питань або очікуєте отримати більше 100 відповідей, Google Forms - це кращий інструмент для вас. Але якщо ціна не є однією з ваших головних міркувань, SurveyMonkey має багато функцій, яких ви не знайдете в Google Forms.

3.2.2 Дизайн та налаштування

Google Forms насправді мають лише чотири варіанти налаштувань:

- колір шаблону, який змінює колір заголовка та полів;
- колір фону;
- фото як зображення заголовка;
- шрифт. У вас є лише чотири варіанти на вибір, жоден з яких не виглядає надто професійним.

Як підсумок Google Forms не надає вам багато місця в плані налаштування дизайну.

Базовий план SurveyMonkey також досить обмежений, що стосується варіантів налаштування: ви можете вибрати дев'ять заздалегідь розроблених тем і відкоригувати макет опитування. Але за допомогою преміального плану ви можете завантажити логотип, створити спеціальний шаблон, вибрати один з десятків шрифтів, змінити кольори та розміри шрифту або додати фонове зображення.

І хоча Google Forms завжди відображає свій брендинг у ваших опитуваннях, індивідуальний план SurveyMonkey Premier (\$ 99 / місяць) та план Premier Team (\$ 75 / місяць) дають вам можливість видалити брендинг SurveyMonkey.

Ви можете створити багатосторінкові опитування в обох додатках. Але в Google Forms запитання на одній сторінці відображаються у довгому списку, який

потрібно прокрутити. У SurveyMonkey ви можете ввімкнути можливість переключення питань.

Якщо налаштування дизайну опитування для вас не дуже важливо, Google Forms має бути достатнім. Але якщо вам потрібно створити фірмове опитування або хочете детальніше налаштувати дизайн, вам необхідно інвестувати в преміальний план SurveyMonkey.

3.2.3. Параметри співпраці

Google Forms і SurveyMonkey дозволяють створювати власні шаблони опитування, якими ви можете поділитися зі своєю командою. Але в іншому SurveyMonkey перемагає.

Користувачі SurveyMonkey, які підписалися на командний план, отримують ряд функцій співпраці. Ви можете встановити детальні дозволи, щоб надати різним користувачам можливість переглядати, редагувати або коментувати опитування. Члени команди можуть залишати коментарі як до попереднього перегляду, так і до результатів опитування, а також ви можете надсилати сповіщення членам команди, щоб вони повідомили, що ви внесли зміни або залишили коментар.

Співпраця в Google Forms менш гнучка. Ви можете додати співробітників, але єдиний дозвіл, який ви можете встановити - це редагувати. Не існує варіанту перегляду, і система не підтримує коментарів. Вона також не надсилає сповіщення про внесення змін, і на відміну від деяких інших інструментів Google, Google Forms не зберігає історію змін.

Якщо ви співпрацюєте з невеликою командою, інструменти співпраці Google Forms - це, мабуть, все, що вам потрібно. Але якщо ви співпрацюєте з великою командою або зовнішніми партнерами, або якщо у вас є складні робочі процеси з перегляду та затвердження - план команди SurveyMonkey є кращим вибором.

3.2.4 Типи питань, правила та оцінка

SurveyMonkey пропонує більше варіантів, ніж Google Forms, що стосується типів питань. На додаток до основних типів питань, які ви знайдете в обох інструментах - множинний вибір, коротка відповідь, випадające меню та прапорці - SurveyMonkey пропонує повзунки, поля прийому платежів та матриці.

Звичайно, більшість сучасних типів запитань SurveyMonkey, включаючи завантаження файлів, які доступні безкоштовно в Google Forms, доступні лише з преміальним планом.

Обидва інструменти також мають варіанти рандомізації порядку, в якому респонденти бачать запитання, що корисно при використанні інструментів для складання вікторин. Але Google Forms пропонує лише загальний варіант рандомізації. SurveyMonkey пропонує більш детальні варіанти рандомізації, дозволяючи вам рандомізувати лише певну сторінку або певні питання. Знову ж таки, ця функція доступна лише у планах преміум.

Google Forms дозволяє встановлювати більш конкретні правила перевірки відповідей (тобто вимагати від користувачів відповіді певним чином) на основі числа, тексту, довжини або регулярного вираження. У відповідях, які не відповідають вашим правилам, відображатимуться налаштовані повідомлення про помилки. Це корисно для таких речей, як перевірка того, що введена адреса електронної пошти вірна, це забезпечуючи більш чисті дані.

Правила перевірки SurveyMonkey не допускають регулярних виразів, тому ви обмежуєтесь перевіркою форматування дати, точного форматування електронної адреси та форматування чисел. Це більш доступно, якщо ви не знаєте, як писати регулярні вирази, але це обмежує типи відповідей, які ви можете перевірити.

Якщо ви використовуєте додаток для створення вікторини і хочете негайно надати зворотній зв'язок, обидва додатки пропонують автоматичне підрахунок балів. Проте оцінка балів Google Forms є більш надійною, оскільки її можна використовувати для більш, ніж декількох запитань та випадających варіантів.

Питання з короткими відповідями також можна автоматично оцінити, встановивши діапазон чисел (наприклад, відповідь має бути від 1 до 10) або встановити текстові правила (наприклад, відповідь повинна містити слово "кисень").

Якщо вам потрібно зібрати платежі через вашу форму, SurveyMonkey пропонує пряму інтеграцію з Stripe, що дозволяє учасникам опитування здійснювати оплату. Google Форми не пропонують цю функцію, хоча ви можете включити посилання на PayPal або інший інструмент обробки платежів.

3.2.5 Шаблони

Google Forms пропонує лише 16 основних шаблонів, які допомагають швидко будувати такі речі, як додатки на роботу, форми замовлення та оцінки.

SurveyMonkey, з іншого боку, пропонує майже 200 шаблонів опитування - організованих за галузями та цілями - для всього, починаючи від опитування маркетингових досліджень до співбесіди з виїздом працівників. Кожен із цих шаблонів містить запитання, які перевіряються експертами SurveyMonkey, і запитання подаються у порядку, який максимально збільшує кількість та якість відповідей.

Безкоштовний план дає змогу отримати доступ до 40 коротких шаблонів опитування, тоді як повна бібліотека доступна користувачам з преміум планом.

Якщо ви хочете змішувати та узгоджувати, SurveyMonkey пропонує всім користувачам доступ до банку запитів із 1800+ попередньо написаних питань. Ці питання були написані та переглянуті методологами опитування SurveyMonkey та сертифіковані як методично обґрунтовані, що означає, що питання має надати точні результати з мінімальними упередженнями.

3.2.6 Розгалуження та умовна логіка

Умовна логіка в опитуваннях зводиться до тверджень if / then. Наприклад, "Якщо користувач обрав задоволений, то опитування закінчується і відображається сторінка подяки."

Google Forms дозволяє здійснити подібне розгалуження для запитань із вибором та випадками. Ви можете надіслати користувачів до іншого розділу чи запитання, виходячи з їх попередньої відповіді, або ви можете негайно закінчити опитування.

SurveyMonkey також пропонує умовну логіку, але лише щодо своїх планів преміум-класу. Це працює так само, як і Google Forms: надішліть респондента на інше запитання, сторінку чи кінець опитування залежно від того, як вони відповідають на певне запитання. Але SurveyMonkey пропонує умовну логіку щодо типів питань, яких немає у Google Forms, таких як прапорці та рейтинги.

3.2.7 Збір відповідей

Google Forms і SurveyMonkey дозволяють вам надсилати опитування електронною поштою, переглядати посилання або вбудувати опитування на свій веб-сайт.

Але SurveyMonkey пропонує ще кілька функцій, яких Google Forms не має. Ви можете змусити людей взяти опитування у Facebook Messenger, вставити опитування в мобільний додаток або навіть збирати відповіді в автономному режимі за допомогою програми SurveyMonkey Anywhere та завантажувати їх, коли ви знову в мережі. Якщо вам потрібна допомога, щоб отримати відповіді на ваші опитування, у SurveyMonkey також є продукт під назвою Audience, де ви платите їм за просування вашого опитування, щоб отримати більше відповідей.

3.2.8 Аналіз та звітність

Google Forms і SurveyMonkey пропонують різноманітні функції аналізу та візуалізації ваших даних. Вони показують індивідуальні відповіді, а також зведені дані у графічному форматі.

Google Forms відображає різні типи графіків залежно від типу запитання, яке ви задали.

SurveyMonkey, з іншого боку, дозволяє вам вибрати саме те, як ви хочете відображати підсумкові дані. У вільному плані ви можете вибрати один з восьми типів графіків, включаючи кругові діаграми, смугові та лінійні графіки, налаштувати кольори і навіть змінити мітки, показані на ваших графіках, для отриманих вами даних.

Звичайно, ви можете робити все це і за допомогою Google Forms, але це не так просто. Вам потрібно надіслати свої дані до Google Таблиць і вручну створити власні графіки. Тим не менш, наявність ваших даних у Google Таблицях також дає змогу створювати зведені таблиці та використовувати функцію «Огляд» для більш детального аналізу даних. Або ви можете надіслати свої дані в Google Data Studio для ще більшої візуалізації.

Безкоштовний план SurveyMonkey не дозволяє експортувати результати, тому для глибокого аналізу вам доведеться оновити свій план. Наявні інструменти звітності збільшують кількість та можливості на кожному рівні планів, що дає вам можливість фільтрувати відповіді, аналізувати відповіді з певної групи, та оцінювати статистичну значимість.

SurveyMonkey також може проаналізувати опитування - навіть до того, як ви отримаєте відповіді. Функція Genius оцінює ваше опитування на основі факторів, включаючи тривалість опитування, типи запитань та передбачуваний час для завершення. Загальна оцінка, показників завершеності та запропоновані сфери вдосконалення.

3.2.9 Експорт даних з Google Forms та SurveyMonkey

Google Forms - це окремий продукт, і ви можете переглядати всі відповіді безпосередньо в додатку: просто перейдіть на вкладку «Відповіді», щоб переглянути їх. Але якщо ви збираєте інформацію від багатьох людей, вам потрібно помістити ці дані в електронну таблицю, де вони готові до обробки та аналізу.

Ви можете зробити це автоматично, підключивши Google Forms до Google Sheets. Для цього необхідно замінити налаштування, щоб відповіді форми надсилалися до Google Sheets.

Google Sheets в свою чергу дозволяє експортувати файли у форматі .csv та .xlsx. Структура експортованого файлу складається з обов'язкового стовпця "відмітка часу" і стовпців питань, в рядках яких розташовуються дані відповідей користувачів.

SurveyMonkey також дозволяє експортувати дані опитувань в файлах формату .csv та .xlsx. Структура цих файлів містить такі дані:

- Respondent ID. Унікальний номер, пов'язаний з респондентом;
- Collector ID. Унікальний номер колекціонера;
- Start date. Часова позначка, коли респондент звертався до опитування;
- End date. Часова позначка, коли респондент подав або востаннє змінив опитування;
- IP address. IP-адреса респондента;
- Email Address. Доступно, якщо ви використовували колектор запрошень електронною поштою;
- First name;
- Last name;
- Question. Дані відповідей респондента.

Як ми можемо помітити дані обох платформ мають схожий формат для зберігання даних відповідей користувача, а саме "питання-відповідь". Незважаючи

на те, що SurveyMonkey містить більше додаткових даних, але ці дані є другорядними для нашої програмної системи і вони не як не вплинуть на її роботу.

Таким чином в майбутньому можна реалізувати функціонал, який дозволить зчитувати дані файлів .csv та .xlsx використовуючи csv та xlsx парсери даних.

3.2.10 Результат порівняння існуючих форм опитування

Якщо ціна викликає занепокоєння, вам краще скористатися Google Forms. Хоча SurveyMonkey має вільний план, він надзвичайно обмежений.

Якщо дизайн є ключовим фактором, SurveyMonkey - ваша краща ставка. Його опитування набагато більш налаштовані, і, чесно кажучи, вони виглядають більш професійними, ніж ті, які створені в Google Forms. SurveyMonkey також є більш сильним варіантом для просунутих користувачів, яким потрібен кількісний аналіз великої кількості даних, або для користувачів, які вважають, що вони можуть використовувати такі параметри, як шаблони та галузеві орієнтири.

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Функціональні вимоги

Функціональні вимоги описують сервіси, що надаються програмною системою, її поведінку в певних ситуаціях, реакцію на ті чи інші вхідні дані і дії, які система дозволить виконувати користувачам [10]. Іноді сюди додаються відомості про те, чого система робити не повинна.

Почнемо із вимог, що система зобов'язана мати. Особливо це стосується процедури уведення особистих даних. Система не повинна дозволяти користувачеві видаляти обов'язкові поля для заповнення. Також не можна визначити більше кластерів, ніж є елементів у вибірці та менш ніж одиниця. Значення значущості має бути у діапазоні від 0 до 100. Система повинна перевіряти поля введення на правильність заповненої інформації. Невірно заповнені поля даних можуть порушити підрахунок ймовірності надходження, що може спричинити за собою несподіваний невірний результат кластеризації даних. Користувач не повинен мати можливості змінювати дані, отримані з віддалених джерел.

Наступним кроком роздивимося вимоги, що система має робити. Система повинна мати особистий кабінет та можливості: реєстрації, авторизації, додавання особистих даних, редагування особистих даних, додавання, редагування та видалення опитування. Система має дати можливість користувачеві переглядати сформовані опитування, запускати їх у тестовому режимі. Користувач повинен мати змогу переглядати результати опитувань у графічному, табличному представленні та проводити кластерний аналіз даних.

4.2 Нефункціональні вимоги

Згідно специфікації вимог програмного забезпечення (IEEE 29148:2011) передбачені такі нефункціональні вимоги:

- вимоги до продуктивності. Програмна система повинна зберігати високу продуктивність при одночасному доступі декількох користувачів, не використовувати великий обсяг пам'яті, сервер бази даних повинен швидко обслуговувати запити;
- вимоги до збереження (даних). Усі дані мають бути збережені у базі даних Firebase Firestore та бути винесені до міграцій. Час зберігання необмежений. Дані, що зберігаються у пам'яті користувачів, мають бути видалені одразу ж після синхронізації із сервером;
- вимоги до якості програмного забезпечення. Програмна система повинна використовувати останні версії бібліотек, бути масштабованою та написаний код має відповідати принципам SOLID;
- вимоги до безпеки системи. Програмна система повинна запобігти атакам міжсайтового сценарію та конфіденційні дані користувачів повинні бути зашифровані;
- вимоги на інтелектуальну власність. Усі використані бібліотеки, алгоритми мають бути у вільному доступі (open source).

5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

5.1 UML проектування програмного забезпечення

Проектування програмного продукту було проведено з використанням діаграм мови UML. За допомогою цих діаграм було відображено роботу та набір функцій системи. Усього було побудовано три діаграми, за допомогою яких було розглянуто систему із усіх можливих сторін.

Першою створеною діаграмою була діаграма варіантів використання (див. рис. 5.1). Єдиним актором системи є користувач. Йому доступні такі можливості:

- реєстрація;
- авторизація;
- створення, редагування, проходження і видалення опитувань;
- перегляд результатів;
- проведення кластерного аналізу.

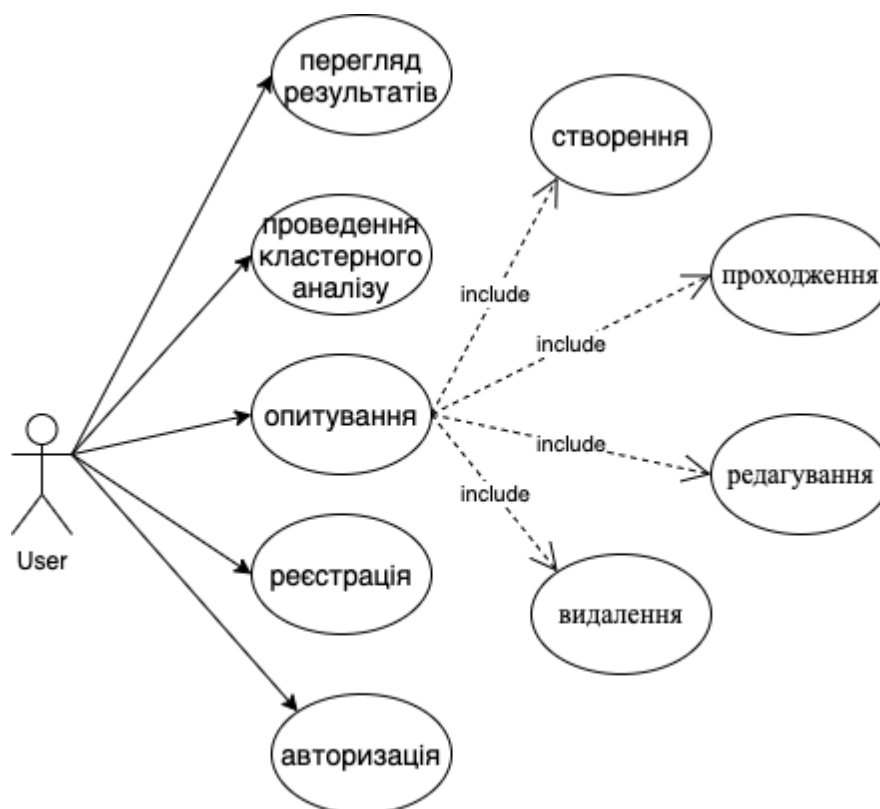


Рисунок 5.1 – Діаграма варіантів використання

Наступною є діаграма послідовності (див. рис. 5.2). Ця діаграма відображає взаємодії об'єктів впорядкованих за часом [11]. На ній продемонстровано у вигляді вертикальних ліній різні процеси або об'єкти, що існують водночас. Надіслані повідомлення зображуються у вигляді горизонтальних ліній в порядку відправлення. Діаграма відображує лише ті об'єкти, які безпосередньо беруть участь у взаємодії, при цьому ніякі статичні зв'язки з іншими об'єктами не вказуються.

Зобразимо на даній діаграмі об'єкт кластеризації даних результатів опитувань. Користувач обирає вибірку (результати опитувань), заповнює критерії кластеризації у разі проходження перевірки правильності заповнення критеріїв кластеризації виконується алгоритм Хамелеон в іншому випадку демонструються інформаційні повідомлення помилок. Після виконання алгоритму повертається результат до екрану користувача у якості графу, табличних значень та значень.

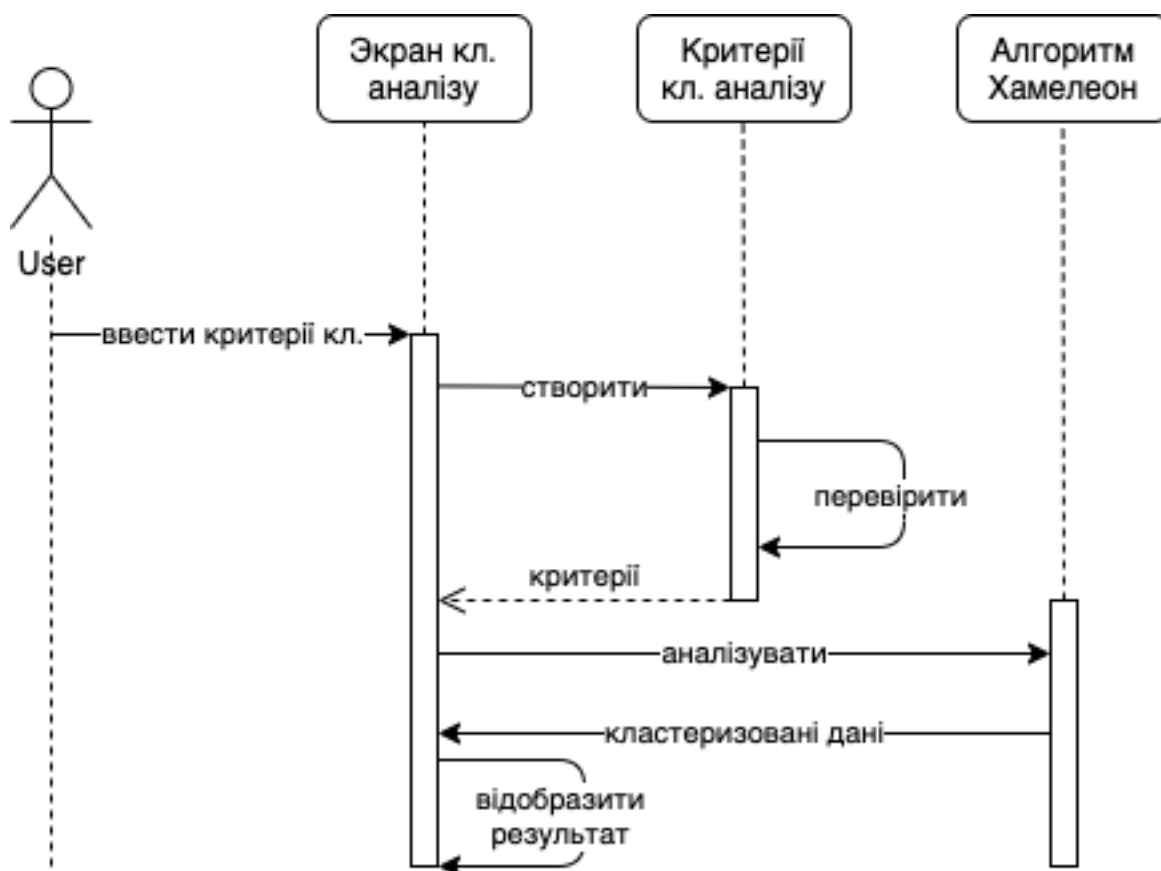


Рисунок 5.2 – Діаграма послідовності

Наступна діаграма, яку варто зобразити – діаграма активності системи. Такі діаграми використовуються при моделюванні бізнес – процесів, технологічних процесів, послідовних чи паралельних потоків дій у системі [12]. Діаграма активності в основному являє собою блок-схему для відображення потоку від однієї діяльності до іншої діяльності.

На рисунку 5.3 приведено таку діаграму для проектованої системи, яка демонструє активність, що була описана до діаграми послідовності.

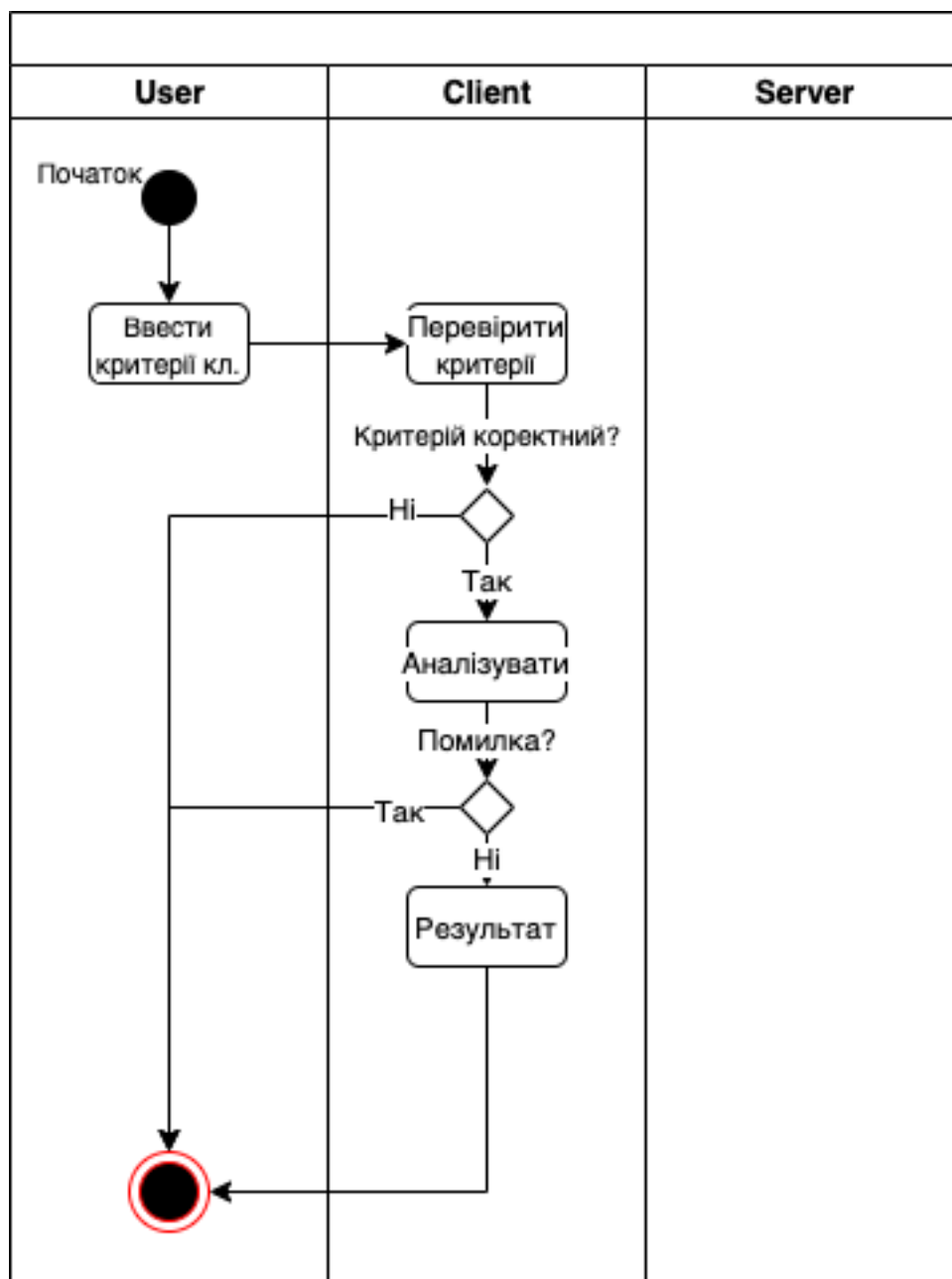


Рисунок 5.3 – Діаграма активності

5.2 Проектування архітектури програмного забезпечення

Для реалізації системи було прийнято рішення використовувати архітектуру клієнт-сервер. Ця архітектура є однією з архітектурних моделей програмного забезпечення, яка є домінуючою концепцією в створенні розподілених мережових додатків і забезпечує взаємодію та обмін даними між ними [13]. В якості основного архітектурного шаблону проектування був обраний MVC. Цей шаблон передбачає поділ даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти:

- вигляд – забезпечує взаємодію користувача з програмною системою. Компоненти уявлення надають дані користувачеві в зручному вигляді і відправляють дії контролера для управління даними;
- модель – визначає дані для програми (як правило, дані зберігаються в базі даних);
- контролер – керує компонентами, отримує сигнали у вигляді реакції на дії користувача (зміна положення курсора миші, натискання кнопки, ввід даних в текстове поле) і передає дані у модель.

Модель є центральним компонентом шаблону MVC, інкапсулює ядро даних і основний функціонал їхньої обробки, не залежить від процесу вводу чи виводу даних, відображає поведінку додатку, незалежну від інтерфейсу користувача. Вона стосується прямого керування даними, логікою та правилами додатку.

Вигляд може являти собою будь – яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації, наприклад основна сторінка та сторінка аналізу даних.

Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду. Він дозволяє структурувати код шляхом групування пов'язаних дій в окремий клас.

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через контролер зробить зміни до моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися. Більш детально роботу шаблону MVC приведено на рисунку 5.4.

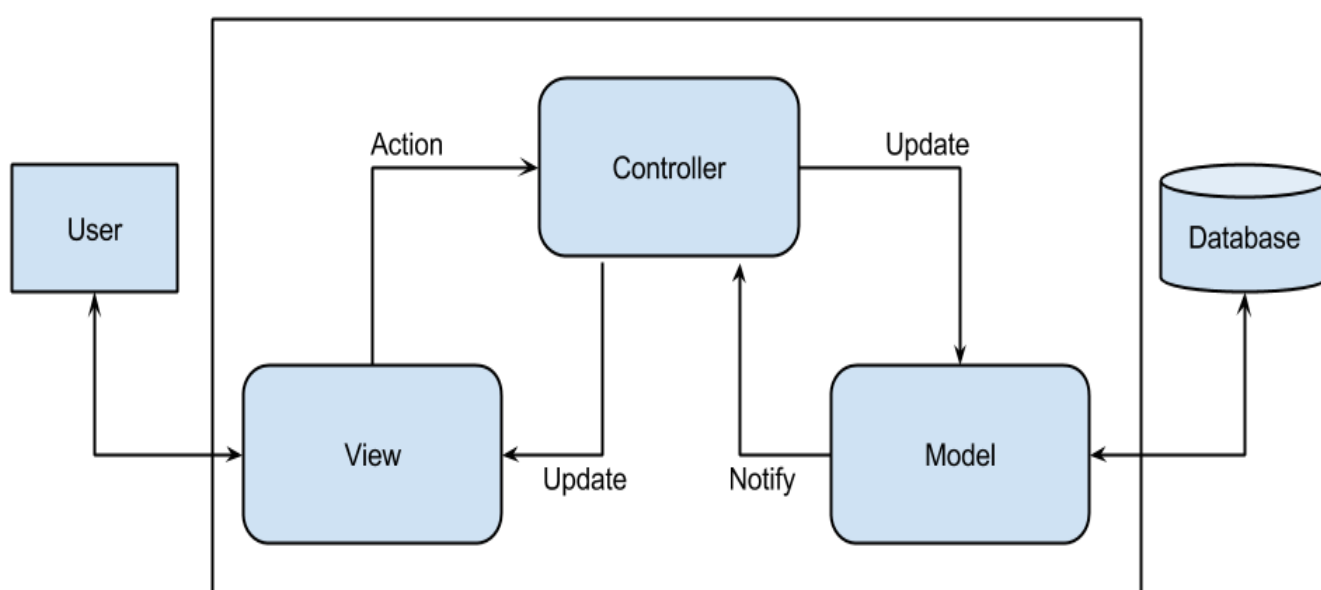


Рисунок 5.4 – Детальна схема обробки запиту користувача в шаблоні MVC

Підсумовуючи можна зазначити, що метою є реалізація гнучкого дизайну програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програми, а також надавати можливість повторного використання окремих компонентів програми.

5.3 Проектування бази даних

Для зберігання і синхронізації даних була обрана хмарна база даних Firebase Firestore Database. Дана база даних відноситься до NoSQL БД.

NoSQL розшифровується як Not Only SQL і являє собою відносно новий підхід до дизайну бази даних [14]. Традиційна реляційна модель, використовує фіксовану схему і розбиває дані в таблиці. Проте, з великими наборами даних, коли дані занадто великі для одного сервера, виникає потреба в масштабуванні на декількох серверах. Це не дуже добре узгоджується з реляційною моделлю, тому що при запиті декількох таблиць, дані не завжди будуть доступні на цьому ж сервері.

Системи управління базами даних NoSQL надають механізм збереження та отримання даних, що організовані в інакший спосіб, ніж звичний реляційний підхід, хоча окремі NoSQL системи можуть надавати SQL-подібний синтаксис. Технологія NoSQL розроблена для забезпечення простоти архітектури бази даних та горизонтального масштабування. Ці якості досягаються за рахунок принципів організації структур даних (ключ-значення, граф, чи документ), які істотно відрізняються від RDBMS. Як наслідок, деякі операції доступу до даних виконуються швидше в NoSQL (наприклад такі, що передбачають складні операції реляційної алгебри), а деякі – в RDBMS. NoSQL бази даних використовують в індустрії високонавантажених веб-додатків реального часу, хоча багато NoSQL сховищ не гарантують цілісність даних, хоча забезпечують їх високу доступність та реплікацію.

Firebase Firestore Database - це хмарна база даних NoSQL. Дані синхронізуються в режимі реального часу та залишаються доступними, навіть якщо програма переходить в автономний режим. Дані зберігаються у форматі JSON (JavaScript Object Notation), який характеризується структурою дерева.

Програми Firebase залишаються чутливими і в режимі офлайн, оскільки пакет даних SDK Firestore Database SDK (Software Development Kit) зберігає дані на

локальному диску. Потім, після відновлення підключення, клієнтський пристрій отримує будь-які зміни, які він пропустив, синхронізуючи його з поточним станом сервера. Firebase пропонує також можливість налаштувати набір правил для доступу до бази даних. Основні правила вимагають, щоб користувачі входили в систему для доступу до даних, але їх можна переосмислити за допомогою Правил безпеки бази даних Firebase Firestore, гнучкої мови правил експресії. Це дозволяє більш чітко визначити, хто і яким чином може отримати доступ до кожного з документів бази даних.

База даних реального часу Firebase здатна керувати до 100 000 одночасних підключень. Важливо розуміти, що це не максимальна кількість користувачів, а кількість підключених користувачів одночасно. Для прикладу, додатків із 10 мільйонами користувачів, як правило, менше понад 100 тисяч одночасних користувачів. Однак можливо вийти за цю межу, реалізуючи кілька баз даних. Кількість відповідей, які може надіслати кожна база даних за секунду, становить приблизно 100 000, це означає, що це максимальна кількість одночасних дій читання на базі даних.

Документами в базі даних є: «Користувач» («User»), «Опитування» («Survey»), «Результат» («Result»). Всі документи зберігаються в своїй окремій колекції.

Документ «Користувач» призначений для зберігання особистих даних користувача програмною системою, містить таку структуру:

- Id - унікальний ідентифікатор;
- Email – електронна пошта.

Структура документа «Користувач» наведена у таблиці 5.1.

Таблиця 5.1 – Структура документа «Користувач»

Документ	Назва	Тип
User	Id	String
	Email	String

Документ «Опитування» призначений для зберігання опитувань, має таку структуру:

- Name – назва опитування;
- Description – опис опитування;
- Id - унікальний ідентифікатор;
- UserId - унікальний ідентифікатор користувача;
- Questions - список питань.

Структура документа «Опитування» наведена у таблиці 5.2.

Таблиця 5.2 – Структура документа «Опитування»

Документ	Назва	Тип
Survey	Id	String
	UserId	String
	Name	String
	Description	String
	Questions	Array

Документ «Результат» призначений для зберігання результатів проходження опитувань і містить таку структуру:

- Id - унікальний ідентифікатор;
- UserId - унікальний ідентифікатор користувача;
- Results - список відповідей.

Структура документа «Результат» наведена у таблиці 5.3.

Таблиця 5.3 – Структура документа «Результат»

Документ	Назва	Тип
Result	Id	String
	UserId	String
	Results	Array

5.4 Приклади найцікавіших алгоритмів та методів

Одним з найцікавіших алгоритмів системи є алгоритм Хамелеон (див. Додаток Б). Даний алгоритм служить для кластеризація даних опитувань. Нижче представлено фрагмент коду необхідний для початкової ініціалізації алгоритму Хамелеон.

```
class Chameleon {
  constructor(k, initNrOfClusters, resultNrOfClusters, points) {
    this.k = k;
    this.initNrOfClusters = initNrOfClusters;
    this.resultNrOfClusters = resultNrOfClusters;
    this.points = points;
    this.graph = [...points][...points];
    this.knnGraph = [...points][...points];
    this.clusters = [];
  }
}
```

Початковими даними є:

- `k` – це кількість сусідів для першої частини алгоритму (`k-nn` алгоритм);
- `initNrOfClusters` – це очікувана кількість кластерів після другої частини алгоритму;
- `resultNrOfClusters` – це очікувана кількість кластерів в кінці алгоритму;
- `points` – це список точок для кластеризації;
- `graph` – це матриця суміжності повного графа. Кожна вершина являє собою одну точку, кожен край представляє вагу зв'язку між двома точками ($1 / \text{відстань}$);
- `knnGraph` – це матриця суміжності `k-nn` графа;
- `clusters` – це список кластерів результатів.

Першим етапом кластеризації даних є реалізація таких методів: `initCompleteGraph`, `runKnn`, `initClusters`. Нижче представлено фрагменти коду цих методів.

```
initCompleteGraph() {
  for (let i = 0; i < this.points.length; i++) {
    for (let j = 0; j < this.points.length; j++) {
      if (i === j) {
        continue;
      }
    }
  }
}
```



```

    }
    if (this.graph[j][i] !== null) {
        this.graph[i][j] = this.graph[j][i];
        this.knnGraph[i][j] = this.graph[i][j];
        continue;
    }
    // weight = 1 / distance
    this.graph[i][j] = 1.0 /
CoordinatesCalculator.getDistance(this.points[i], this.points[j]);
    this.knnGraph[i][j] = this.graph[i][j];
}
}
}

```

Метод `initCompleteGraph` обчислює відстані між кожною парою точок і призначає вагу як $1/\text{відстань}$.

```

runKnn() {
    for (let i = 0; i < this.points.length; i++) {
        const weightsSorted = this.knnGraph[i].filter((item) =>
item).reverse();
        const minWeight = weightsSorted[k - 1];

        for (let j = 0; j < this.points.length; j++) {
            if (this.knnGraph[i][j] !== null && this.knnGraph[i][j] <
minWeight) {
                this.knnGraph[i][j] = null;
            }
        }

        for (let i = 0; i < this.points.length; i++) {
            for (let j = 0; j < this.points.length; j++) {
                if (this.knnGraph[i][j] == null && this.knnGraph[j][i] !== null)
{
                    this.knnGraph[i][j] = this.knnGraph[j][i];
                }
            }
        }
    }
}

```

Метод `runKnn` – це метод пошуку k найближчих сусідів для кожної точки. Він знаходить вагу свого k -го найближчого сусіда, який називається `minWeight`, і видаляє ці края, вага яких менший ніж `minWeight`.

```

initClusters() {
    const visitedPoints = new Array(this.points.length).fill(false);

    for (let i = 0; i < visitedPoints.length; i++) {
        if (!visitedPoints[i]) {
            const connectedPoints = runDfs(i, visitedPoints, []);
            const cluster = new Cluster();
            cluster.setPoints(connectedPoints);
            cluster.setGraph(createSubgraph(connectedPoints));
            clusters.add(cluster);
        }
    }
}

```

```

    }
}

```

Метод `initClusters` знаходить підключені компоненти в `knnGraph` за допомогою алгоритму DFS. Для кожного підключеного компонента він створює новий кластер.

Нижче представлено фрагмент коду алгоритму DFS.

```

runDfs(idx, visitedPoints, connectedPoints) {
    visitedPoints[idx] = true;
    connectedPoints.push(this.points[idx]);

    for (let i = 0; i < visitedPoints.length; i++) {
        if (!visitedPoints[i] && this.knnGraph[idx][i] != null) {
            this.runDfs(i, visitedPoints, connectedPoints);
        }
    }

    return connectedPoints;
}

```

Це рекурсивний алгоритм пошуку з'єднаних точок у `knnGraph`.

На другому етапі виконується пошук кластерів для розділу, додавання кластерів після їх розділу до списку усіх кластерів, та видалення старих кластерів з списку усіх кластерів. Нижче представлено фрагмент коду другого етапу.

```

while (this.clusters.length < this.initNrOfClusters) {
    const clusterToPartition = this.clusters.map()
        .max(Comparator.comparing((c) => c.getPoints().size()));
    const twoClusters = partitionCluster(clusterToPartition);
    this.clusters.push(twoClusters);
    this.clusters.pop();
}

```

На третьому етапі виконується пошук кластерів для об'єднання, додавання кластерів після їх об'єднання до списку усіх кластерів, та видалення старих кластерів з списку усіх кластерів. Нижче представлено фрагмент коду другого етапу.

```

while (this.clusters.length > this.resultNrOfClusters) {
    const twoClusters = findTwoClustersToConnect();
    const resultCluster = mergeTwoClusters(twoClusters[0], twoClusters[1]);
    this.clusters.add(resultCluster);
    this.clusters.removeAll(twoClusters);
}

```

Останнім кроком є повернення результату кластеризації.

5.5 Проектування UI / UX або іншого дизайну системи

Для початку необхідно розібратися з визначеннями UI/ UX дизайну.

UX – це User Experience (дослівно: «досвід користувача»). Тобто це те, який досвід / враження отримує користувач від роботи з вашим інтерфейсом [15]. Чи вдається йому досягти мети і на скільки просто або складно це зробити.

UI – це User Interface (дослівно «інтерфейс користувача»). Тобто це те, як виглядає інтерфейс і те, які фізичні характеристики набуває.

Головний екран системи побудови опитування зображений на рисунку 5.5.

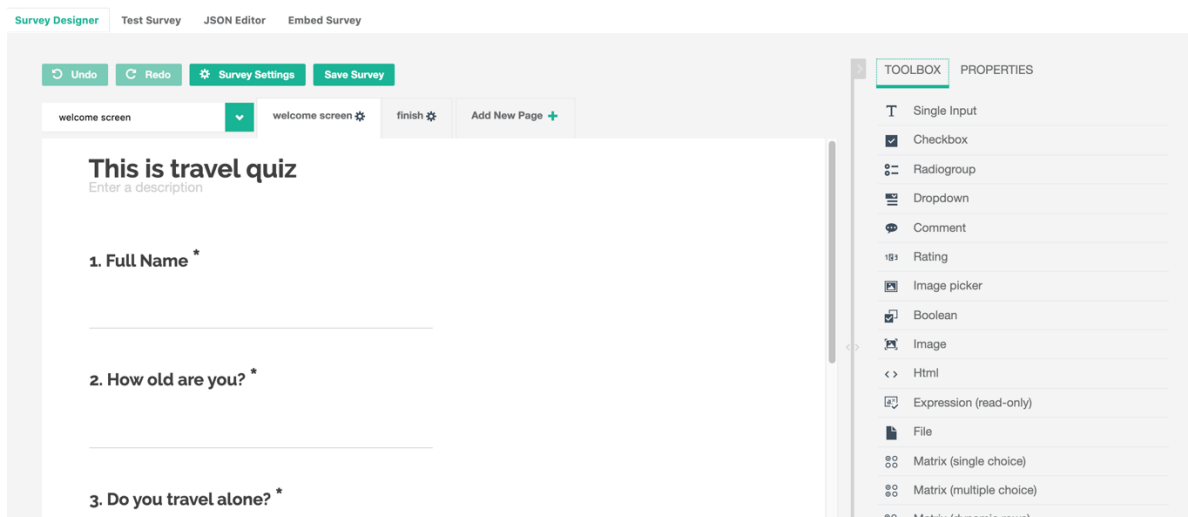


Рисунок 5.5 – Головний екран системи побудови опитування

Основними елементами на цьому екрані є:

- панель елементів. Панель елементів складається з двох вкладок: Toolbox і Properties. Toolbox - містить список елементів для побудова опитувань. Properties - містить список властивостей, якими володіє активний елемент, якщо елемент не вибрано, то містить список властивостей сторінки опитування;
- панель навігації. Панель навігації містить чотири режими. Survey designer - налаштування нового або редагування існуючого опитування та його

елементів. Test survey - тестування опитування. JSON editor - режим, що дозволяє змінити представлення опитування у JSON формат та працювати з ним. Embed survey - інструкція з розміщення опитування на вашому веб сайті;

- панель управління. Панель управління містить кнопки Undo, що дозволяє вернутись на попередній крок якщо він є, Redo, що дозволяє перейти на наступний крок, якщо він є, Survey setting – що відкриває панель Properties для даного опитування та Save survey, що зберігає опитування;
- область конструювання.

Survey дозволяє здійснити подібне розгалуження для запитань із вибором, за яким ви можете надіслати користувачів до іншого розділу чи запитання, виходячи з їх попередньої відповіді, або ви можете негайно закінчити опитування зображено на рисунку 5.6.

The image shows a configuration window for survey logic. At the top, a tab labeled 'Logic' is selected. Below it, the 'Visible If' section contains a text input with 'Expression is empty' and a green upward arrow. Underneath are two buttons: 'Build' (highlighted in green) and 'Edit'. Below these buttons is a dropdown menu showing 'alone' with a green downward arrow, followed by a vertical dashed line and another dropdown menu showing 'equals' with a green downward arrow. Below this is a section titled 'Please enter/select the value' with a toggle switch. The toggle has 'No' on the left and 'Yes' on the right, with the switch currently positioned towards 'Yes'. Below the toggle is a 'Remove' button. Further down is an 'Add condition' button. At the bottom, there are two more sections: 'Enable If' and 'Required If'. Both have text inputs with 'Expression is empty' and a green downward arrow.

Рисунок 5.6 – Функціонал розгалуження для запитань

Survey, як і Google Forms, дозволяє встановлювати конкретні правила перевірки відповідей на питання, контролюючи правильність уведених даних. У відповідях, які не відповідають вашим правилам, відображатимуться налаштовані повідомлення про помилки.

Велика кількість типів питань, які ви знайдете в обох розглянутих інструментах: доповнюється коментарем, рейтингом та HTML розміткою. Кожен елемент має гнучкі налаштування а саме користувач має змогу: змінити положення підпису, задати значення за замовчуванням, ім'я, опис, вказати правильну відповідь, додати власноруч елементи відповідей або вказати дані з Інтернету.

Дана система дозволяє вбудувати опитування на свій веб-сайт або переглядати його у системі.

Survey пропонує функції аналізу та візуалізації ваших даних. Вони показують індивідуальні відповіді, а також зведені дані у графічному форматі. Сторінка візуалізації даних, зображена на рисунку 5.7, містить різні типи графіків залежно від типу запитання, але користувач у даній системі також має змогу самостійно змінювати представлення даних. Він може обирати з таких типів відображення: смугові та лінійні графіки, таблиця, гістограма, кругова діаграма, каліброва діаграма та діаграма розкидання. Користувач має змогу змінювати положення елементів на сторінці для більш зручного відображення.

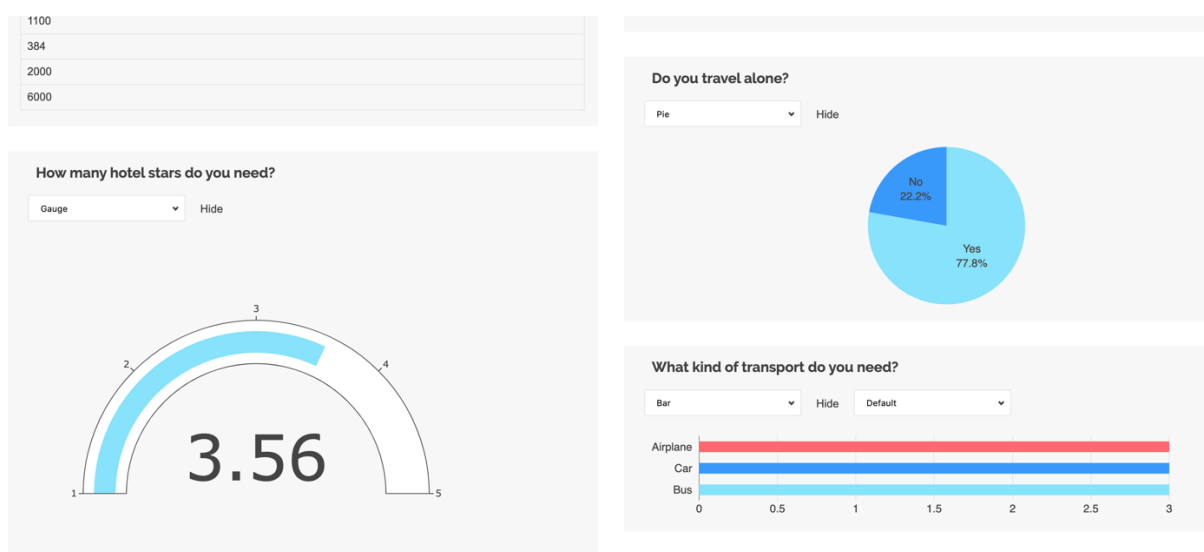


Рисунок 5.7 – Сторінка візуалізації даних

Сторінка аналізу даних зображено на рисунку 5.8. На даній сторінці виконується кластеризація даних. Кластеризація виконується за допомогою алгоритму хамелеон, який саме у таких випадках має значно більші шанси для вірного розбиття отриманих даних на кластери.

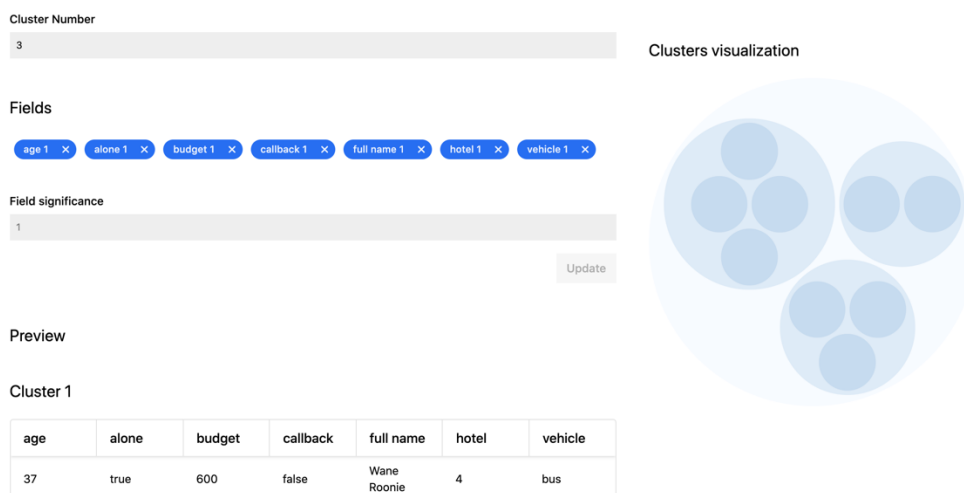


Рисунок 5.8 – Сторінка аналізу даних

На даній сторінці є можливість обрати кількість кластерів, увівши значення у поле «Cluster Number», допустимими значення для цього поля є діапазон від одного до кількості елементів у вибірці.

На даній сторінці присутні такі елементи:

- числове поле введення «Cluster Number». Дане поле дозволяє визначити, на скільки кластерів будуть поділені дані;
- перелік полів. Перелік усіх полів у даній вибірці. Кожне поле має числовий аргумент значущості, значення якого за замовчуванням дорівнює одиниці. Даний аргумент використовується при проведенні кластерного аналізу і враховується для формування кластерів;
- числове поле введення «Field significance». Дане поле дозволяє визначити значущість виборного поля з переліку полів;
- табличне представлення кластерів;
- візуальне відображення кластерів.

ВИСНОВКИ

В роботі було проведено аналіз алгоритмів пошуку в структурі даних для створення зручного механізму опитування користувачів мереж та побудова програмної системи за результатами аналізу.

Було приведено вирішення завдання кластеризації великих обсягів даних, а також вибірок з різними статистичними характеристиками та побудовано зручний механізм опитування користувачів мереж. Рішення цього завдання полягає у застосуванні модифікації алгоритму Хамелеон з тією комбінацією методів і алгоритмів на кожному етапі алгоритму, яка максимально підходить для вхідних вибірок на основі статистичних характеристик вибірки. Розроблена модифікація методу дозволяє вирішити завдання кластеризації на підставі індивідуального підходу до кожної вибірки і в той же час є універсальним для великої кількості різних вибірок. Створені моделі дозволяють скоротити час на кластеризацію, уніфікувати технологію кластеризації, у ряді випадків підвищити якість оцінювання результатів, ґрунтуючись на кількох методах. Всі ці результати мають важливе наукове і практичне значення в області кластеризації даних.

У роботі проведено дослідження та аналіз методів, засобів і технологій, що застосовуються під час роботи з даними, для кластеризації і класифікації.

У процесі дослідження було виявлено, що на даний момент існує дуже велика кількість методів кластеризації. Існуючі методи вельми різноманітні, мають велику кількість обмежень, що накладаються на вибірки для обробки. Наведена класифікація методів і відображені основні характеристики підходів, переваги і недоліки. Розглянуті алгоритми Хамелеон як найбільш актуальні при вирішенні завдання кластеризації великих обсягів лінійно нероздільних даних. Зроблено висновок, що метод Хамелеон може бути модифіковано, завдяки чому підвищиться швидкість кластеризації даних та алгоритм стане більш універсальним. Усі ці дослідження дозволили побудувати зручний механізм опитування користувачів мереж.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ляховець А.В. Экспериментальные результаты исследования качества кластеризации разнообразных наборов данных с помощью модифицированного алгоритма Хамелеон.//Вісник запорізького національного університету. 2011 №2. С. 86-73.
2. Ляховець А.В. Характеристики выборки данных для выбора k при построении графа k-ближайших соседей.//Тези доповіді VI Міжнародної науково-практичної конференції «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій». 2012. С. 168-169.
3. Ляховець А.В. Характеристики выборки данных для выбора k при построении графа k-ближайших соседей.//Тези доповіді XXII Міжнародної науково-технічної конференції «Проблемы информатики и моделирования ПИМ 2012». 2012. С. 59.
4. Ляховець А.В. Кластеризация с помощью нейронной сети Кохонена и модифицированного алгоритма иерархической кластеризации Хамелеон в различных предметных областях.//Науково-технічний журнал «Регистрация, хранение и обработка данных». 2013. С. 53.
5. S. Sumathi. Fundamentals of relational database management systems – Берлін: Springer-Verlag, 2007. – 359 с.
6. Dr.K.Thangadurai, M.Uma, Dr.M.Punithavalli. A Study On Rough Clustering. - Global Journal of Computer Science and Technology Vol. 10 Issue 5 Ver. 1.0 July 2010
7. Gan, Guojun, Chaoqun Ma, and Jianhong Wu, Data Clustering: Theory, Algorithms, and Applications, ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA, 2007.
8. Nitin Agarwal and [Huan Liu](#). "Modeling and Data Mining in Blogosphere". Synthesis Lectures on Data Mining and Knowledge Discovery No. 1, Morgan and Claypool Publishers, Robert Grossman (Editor), August 2009. [Morgan and Claypool webiste](#); [Amazon website](#).

9. F. Sha, G. Gardarin Gradual clustering algorithms for metric spaces In Proceedings: Proc. 15èmes Journées Bases de Données Avancées, *BDA'99* (October 1999).

10. Функциональные и нефункциональные требования [Электронный ресурс] URL: <https://studfiles.net/preview/1848692/page:16/> (дата звернення: 10.05.2020).

11. Хассан Г. UML-проектирование систем реального времени параллельных и распределенных приложений / Гома Хассан. – Москва: Пер. с англ. - М.: ДМК Пресс, 2011. – 704 с.

12. Леоненков А. Самоучитель UML. Эффективный инструмент моделирования информационных систем / Александр Леоненков. – Петербург: БХВ-Петербург, 2001. – 304 с.

13. Overview of Model-View-Controller (MVC) [Электронный ресурс] URL: <http://www.patricksoftwareblog.com/tag/mvc/> (дата звернення: 10.05.2020).

14. NoSQL Databases Explained [Электронный ресурс] URL: <https://www.mongodb.com/nosql-explained> (дата звернення: 10.05.2020).

15. Бородаев Д. В. Веб-сайт как объект графического дизайна / Д. В. Бородаев. – Х. : “Септима ЛТД”, 2006. – 288 с.

16. Technopreneurship in Ukraine. How to Boost Entrepreneurial Competence Development in the Ukrainian IT Industry (The Ukrainian IT Educational System: Basic Facts and Urgent Needs. A. Mendes, Z.Dudar, V. Kauk, T. Shatovska, I. Revenchuk, A. Chupryna, D. Fedasyuk, V. Yakovina, I. Lyutak) edited by Hans Lundberg.// - Linnaeus University Copycenter, 2016.- 160p. ISBN: 978-91-88357-68-7

17. Bohdan Sus, Iona Revenchuk, Nataliia Tmienova, Vira Vialkova Development of Virtual Laboratory Works for Technical and Computer Sciences.- 25th International Conference on Information and Software Technologies, ICIST 2019.- Vilnius, Lithuania, October 10–12, 2019.- pp 383-394

18. І.А. Ревенчук, К.В. Перцьова, О.І. Маренич. Програмна реалізація кластеризації пошукових запитів.- Біоніка інтелекту.-№.-2019.-С.7-14

19. І.А. Ревенчук. Математична модель агрегації даних в соціальних медіа.- Сб. наук. праць "Математичне та комп'ютерне моделювання. Серія: Техн.

Науки" за Міжнар. Наук. Конф. "Питання оптимізації обчислень (ПОО-XLIV).- Кам'янець-Подільський.- 2017.- С. 197-203

20. Bondarenko M. F., Dudar Z. V., Revenchuk I.A. Information Systems and Technologies Used in Distance Form of Education at the University.- Informational and Communication Technologies Technologies – Theory and Practice: Proceedings of the International Scientific Conference ICTMC-2010 Devoted to the 80th Anniversary of I.V. Prangishvili. Nova Science Publishers. Series: Computer Science, Technology and Applications. - 2012.- P.485-490. ISBN: 978-1-61470-050-0