# МАГІСТЕРСЬКА РОБОТА

МР. ІПм - 55.00.00.000 ПЗ

**Група ІПм-22-4**

**Вінер Ростислав**

**2024**

**Івано-Франківський національний технічний університет нафти і газу**

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

**Вінер Ростислав Анатолійович**

(прізвище, ім'я, по батькові)

УДК 004.942

(індекс)

# МАГІСТЕРСЬКА РОБОТА

**Інтелектуальні моделі, методи та алгоритми обробки багатовимірних даних**

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Вінер Р.А.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Чесановський Микола Станіславович, асистент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

**Допущено до захисту**

В.о. завідувача кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Рецензент

доц.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

**Івано-Франківський національний технічний університет нафти і газу**

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ:**
В.о. зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ 04 ” вересня 2023 р.

# ЗАВДАННЯ

## НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

**Вінеру Ростиславу Анатолійовичу**

(прізвище, ім’я, по-батькові)

**1. Тема магістерської роботи “ Інтелектуальні моделі, методи та алгоритми обробки багатовимірних даних ”**

керівник проекту (роботи) Чесановський Микола Станіславович, асистент

затверджені наказом закладу вищої освіти від “ 18 ” грудня 2023 р. № 738/7

**2. Строк подання студентом проекту (роботи)** 16 січня 2024 р.

**3. Вихідні дані до проекту (роботи)** Теоретичні концепції та формальні моделі побудови та функціонування інформаційних та програмних технологій інтелектуального аналізу даних

**4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)**

1. Аналіз предметної області інтелектуального аналізу даних
2. Методики аналізу та обробки багатовимірних даних
3. Моделі та методи інтелектуальної обробки багатовимірнимх даних
4. Реалізація інформаційної технології обробки багатовимірних даних хмарними засобами

**5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)**

1. Компоненти системи ІАД (рис. 1.1)
2. Data Mining як міждисциплінарна галузь (рис. 1.2)
3. Інформаційна ієрархія DIKW (рис. 1.4)
4. Веб-сервіс інтелектуального аналізу даних Weka4WS (рис. 1.6)
5. Приклад роботи бінарної логістичної регресії (рис. 1.13)

**6. Консультанти розділів проекту (роботи)**

| Розділ | Консультант | Підпис, дата |
|---|---|---|
| Нормоконтроль | доц., к.т.н. Вовк Р.Б. | |
| Перевірка на плагіат | доц., к.т.н. Вовк Р.Б. | |
| | | |

**7. Дата видачі завдання** 04 вересня 2023 р.

**Керівник** _____
(підпис)

**Завдання прийняв до виконання**_____
(підпис)

# КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назви етапів магістерської роботи | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Підбір і вивчення літератури по темі магістерської роботи | 01.10.2023 | виконано |
| 2 | Аналіз предметної області інтелектуального аналізу даних | 25.10.2023 | виконано |
| 3 | Методики аналізу та обробки багатовимірних даних | 10.11.2023 | виконано |
| 4 | Моделі та методи інтелектуальної обробки багатовимірниих даних | 22.11.2023 | виконано |
| 5 | Реалізація інформаційної технології обробки багатовимірних даних хмарними засобами | 01.12.2023 | виконано |
| 6 | Реалізація функціональності запропонованої інформаційної технології | 15.12.2023 | виконано |
| 7 | Затвердження пояснювальної записки роботи завідувачем кафедри | 15.01.2024 | виконано |

**Студент – магістр** _____
(підпис)

**Керівник роботи** _____
(підпис)

# АНОТАЦІЯ

**Магістерська робота:** 79 с., 40 рис., 1 табл., 50 джерел.

**Тема:** Інтелектуальні моделі, методи та алгоритми обробки багатовимірних даних.

**Об'єкт дослідження:** методи та процеси інтелектуального аналізу даних при застосуванні їх в прикладних задачах.

**Мета роботи:** дослідження моделей та методів обробки багатовимірних даних та розробці інформаційної технології хмарного сервісу обробки та візуалізації багатовимірних числових величин з метою отримання раніше невідомих залежностей, тенденцій, знань, тощо.

**Предмет дослідження:** інтелектуальні методи, моделі та алгоритми оцінювання, обробки та візуалізації багатовимірних даних.

**Результати дослідження:**

В роботі виконано дослідження та виборі ефективних методів обробки та аналізу багатовимірних даних, а також побудови архітектури інформаційної технології із можливістю оброблювати та аналізувати дані великих обсягів.

**Висновок**

Виконано проектування та розробку інформаційної технології хмарного сервісу обробки та візуалізації багатовимірних даних великого об'єму з метою отримання раніше невідомих закономірностей, залежностей та знань.

**ДАНІ, ІНФОРМАЦІЯ, ЗНАННЯ, МОДЕЛЬ DIKW, BIG DATA, DATA MINING, ІНТЕЛЕКТУАЛЬНІ МЕТОДИ, НЕСТРУКТУРОВАНА ІНФОРМАЦІЯ, БАГАТОВИМІРНІ ДАНІ**

## ABSTRACT

**Master Thesis:** 79 pp., 40 fig., 1 tab., 50 sources.

**Thesis Subject:** Intelligent models, methods and algorithms for multidimensional data processing

**Object of research:** methods and processes of intellectual data analysis when applying them in applied tasks.

**Research goal:** research of models and methods of processing multidimensional data and development of information technology of a cloud service for processing and visualization of multidimensional numerical values in order to obtain previously unknown dependencies, trends, knowledge, etc.

**Subject of research:** intelligent methods, models and algorithms for evaluation, processing and visualization of multidimensional data.

**The results:**

The research and selection of effective methods of processing and analysis of multidimensional data, as well as the construction of an information technology architecture with the ability to process and analyze large volumes of data, were carried out in the work.

**Conclusion**

The design and development of the information technology of the cloud service for the processing and visualization of large volumes of multidimensional data was carried out in order to obtain previously unknown regularities, dependencies and knowledge.



**DATA, INFORMATION, KNOWLEDGE, DIKW MODEL, BIG DATA, DATA MINING, INTELLIGENT METHODS, UNSTRUCTURED INFORMATION, MULTIDIMENSIONAL DATA**

**ЗМІСТ**

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ..... 8
ВСТУП ..... 9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ..... 13
1.1. Сутність концепції інтелектуального аналізу даних ..... 13
1.2. Сутність багатовимірних даних та їх опис ..... 17
1.3. Системи веб-сервісів для інтелектуального аналізу даних ..... 21
1.4. Методики аналізу та обробки багатовимірних даних ..... 31
Висновки до розділу ..... 36
РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ БАГАТОВИМІРНИМХ ДАНИХ ..... 37
2.1. Дослідження та опис методів інтелектуального аналізу даних ..... 37
2.2. Опис процесів організації та обробки багатовимірних даних ..... 41
2.3. Дослідження та визначення переваг фреймворків для обробки багатовимірних даних ..... 45
2.4. Розробка архітектурної схеми системи обробки багатовимірних даних ..... 51
Висновки до розділу ..... 59
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОБРОБКИ БАГАТОВИМІРНИХ ДАНИХ ХМАРНИМИ ЗАСОБАМИ ..... 60
3.1. Опис функціональних можливостей хмарної PaaS-платформи Heroku 60
3.2. Розробка архітектури інформаційної технології ..... 65
3.3. Розміщення програмного скрипту на хмарній платформі ..... 66
Висновки до розділу ..... 72
ВИСНОВКИ ..... 74
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ ..... 75

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ІАД – Інтелектуальний Аналіз Даних

BPM – Business Process Management

CRM – Customer Relationship Management

DX – Developer Experience

OLAP – Online Analytical Processing

OLTP – Online Transactional Protocol

REST – Representational State Transfer

RPC – Remote Procedure Call

LR - Logistic Regression - Логістична регресія

ROC – receiver operating characteristic - Крива помилок

# ВСТУП

**Актуальність теми.**

Функціонувати в нормальному суспільстві стає все важче і важче без наявності якихось великих даних у деяких сферах нашого життя. Багато змін, пов'язаних із зростанням обсягів даних, є настільки тривіальними, що ми їх ледве помічаємо.

Еволюційний процес розвитку людства, його хід і результати знаходять своє відображення в різноманітних фактах, відомостях, даних, інформації тощо. У різні історичні епохи ті відомості, що визнавалися доцільними для подальшого збереження, у доступний спосіб фіксувалися, завдяки чому могли бути використані не лише сучасниками, але й наступними поколіннями. Таким чином, у найбільш узагальненому сенсі людство постійно продукує, накопичує і використовує певний «інформаційний продукт». З плином часу та еволюційним поступом його тематика постійно диверсифікувалася, а обсяги безупинно зростали. Проте в останні десятиріччя в результаті стрімкого розвитку комп'ютерних технологій та їх проникнення чи не в усі сфери життя характер динаміки цього процесу кардинально змінився. Це знайшло відображення в тому, що сучасний етап розвитку цивілізації описується в термінах «інформаційна епоха», «інформаційне суспільство», «інформаційна економіка», «інформаційна революція» тощо.

Характерною особливістю сучасного «оцифрованого» життя стала поява Big Data – «великих даних» (багатовимірних даних), які наразі привертають до себе все більше уваги і стають предметом вивчення дедалі ширшого кола дослідників – від аналітиків даних до економістів, соціологів, маркетологів, медиків і т.д.

На перший погляд, судячи з назви, йдеться просто про значні за розміром інформаційні масиви. Однак великий обсяг – лише одна з особливостей феномену багатовимірних даних, у якому, з одного боку,

знайшли втілення комп’ютерно-інформаційні тренди останніх десятиріч; з іншого – він сам здатен впливати й реально трансформує існуючі уявлення та напрацьовані впродовж тривалого часу практики й моделі поведінки як окремих індивідів, так і складних організаційних структур.

Сучасний світ керується даними, що спонукає до розвитку наук, що пов’язані з обробкою даних. Особливо на дані покладається бізнес: від найдрібніших підприємств та магазинів до корпорацій, що здатні впливати на світ у глобальному сенсі. Процес аналізу даних стрімко розвивається на протязі більш, ніж 40 років, що, насамперед, зумовлено зростанням приватного капіталу та росту ролі даних у розвитку бізнесу.

Інструменти та методи аналізу великих даних користуються великим попитом завдяки використанню “Big Data” у бізнесі. Організації можуть знайти нові можливості та отримати нову інформацію для ефективного ведення свого бізнесу. Ці інструменти допомагають надавати змістовну інформацію для прийняття кращих бізнес-рішень.

Компанії можуть вдосконалювати свої стратегії, пам’ятаючи про орієнтацію на споживача. Аналітика великих даних ефективно допомагає операціям стати більш швидкими, оптимальними та максимально корисними для кінцевого користувача. Це допомагає покращити прибуток компанії в найбільш короткі терміни.

Інтелектуальний аналіз даних – Data Mining – переводить аналіз даних на новий рівень. Зростання Data Mining зумовлене не лише зростанням важливості даних у різних галузях, а й розвитком технологій, що дозволяють опрацьовувати великі об’єми даних.

Іншим важливим аспектом у керуванні бізнесом є CRM-системи. Це системи, що сприяють залученню та роботі з клієнтами. Це ще один потужний інструмент, тому доцільно було б його застосувати у тандемі з інтелектуальним аналізом даних.

**Мета магістерської роботи.**

**Мета дослідження** полягає у дослідження моделей та методів обробки багатовимірних даних та розробці інформаційної технології хмарного сервісу обробки та візуалізації багатовимірних числових величин з метою отримання раніше невідомих залежностей, тенденцій, знань, тощо.

**Об'єкт дослідження** - методи та процеси інтелектуального аналізу даних при застосуванні їх в прикладних задачах.

**Предмет дослідження** – інтелектуальні методи, моделі та алгоритми оцінювання, обробки та візуалізації багатовимірних даних.

Відповідно до мети роботи було сформовано наступні **задачі:**

- провести аналіз предметної області інтелектуального аналізу даних;

- дослідити програмні сервіси та інструменти для інтелектуального аналізу даних;

- навести методики аналізу та обробки багатовимірних даних;

- дослідити існуючі ефективні фреймворки для обробки багатовимірних даних;

- розробити архітектурної схеми системи обробки багатовимірних даних;

- розробити інформаційну технологію на основі хмарної PaaS-платформи.

**Методи дослідження.** В даній роботі використано методи аналізу, синтезу, системного аналізу, логічного узагальнення результатів, методи штучного інтелекту, діалектичний метод, абстрактно-логічний метод, метод порівняння та узагальнення.

**Наукова новизна отриманих результатів** полягає у дослідженні та виборі ефективних методів обробки та аналізу багатовимірних даних, а також побудови архітектури інформаційної технології із можливістю оброблювати та аналізувати дані великих обсягів.

**Практичне значення магістерської роботи** полягає в проектуванні та розробці інформаційної технології хмарного сервісу обробки та візуалізації

багатовимірних даних великого об’єму з метою отримання раніше невідомих закономірностей, залежностей та знань.

**Структура магістерської роботи.** Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 79 сторінок, і містить 40 рисунків, 1 таблицю, список використаних джерел із 50 найменувань.

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

## 1.1. Сутність концепції інтелектуального аналізу даних

Інтелектуальним аналізом даних (Data Mining) називають процес визначення нових, коректних та потенційно корисних знань на основі великих масивів даних. «Інтелектуальний аналіз даних» деякі дослідники вважають синонімом ще одного популярного терміна – виявлення знань у даних – “Knowledge Discovery in Databases” (KDD), на думку інших – інтелектуальний аналіз даних є лише важливим кроком у процесі виявлення знань.

Виявлені в результаті інтелектуального аналізу знання називають патерном (зразком). Тобто задача інтелектуального аналізу полягає в ефективному виявленні осмислених патернів з наявного масиву даних великого розміру.

Отримані знання мають бути цікавими.

Ознаками цікавих знань є

− несподіваність – отримані знання мають дивувати (бути нетривіальними) та нести нову інформацію;

− застосовність – нові знання мають бути придатними до застосування для досягнення поставлених цілей, або до формулювання на їхній основі нових корисних цілей.

Аналіз даних є природним результатом еволюції інформаційних технологій. Розвиток баз даних та технологій керування (менеджменту) потребує також розвитку технологій збору та зберігання даних, керування та аналізу даних, оскільки серйозною проблемою інформаційного суспільства залишається «суперечність між збільшенням обсягів інформації та зменшенням темпів зростання обсягів істинних знань».

Етапами інтелектуального аналізу є

1. Вивчення предметної області.

2. Збір даних.

3. Попередня обробка даних.

а) очищення даних – виключення «шумів» та суперечностей;

б) інтеграція даних – об’єднання даних з різних джерел в одному сховищі;

в) перетворення даних до підходящої форми (агрегація, стиснення, скорочення розмірності, дискретизація атрибутів, тощо).

4. Аналіз даних з метою виявлення патернів.

5. Інтерпретація знайдених патернів (візуалізація, відбір корисних патернів відповідно до функції корисності).

6. Використання нових знань.

Компонентами системи інтелектуального аналізу є (рис. 1.1):

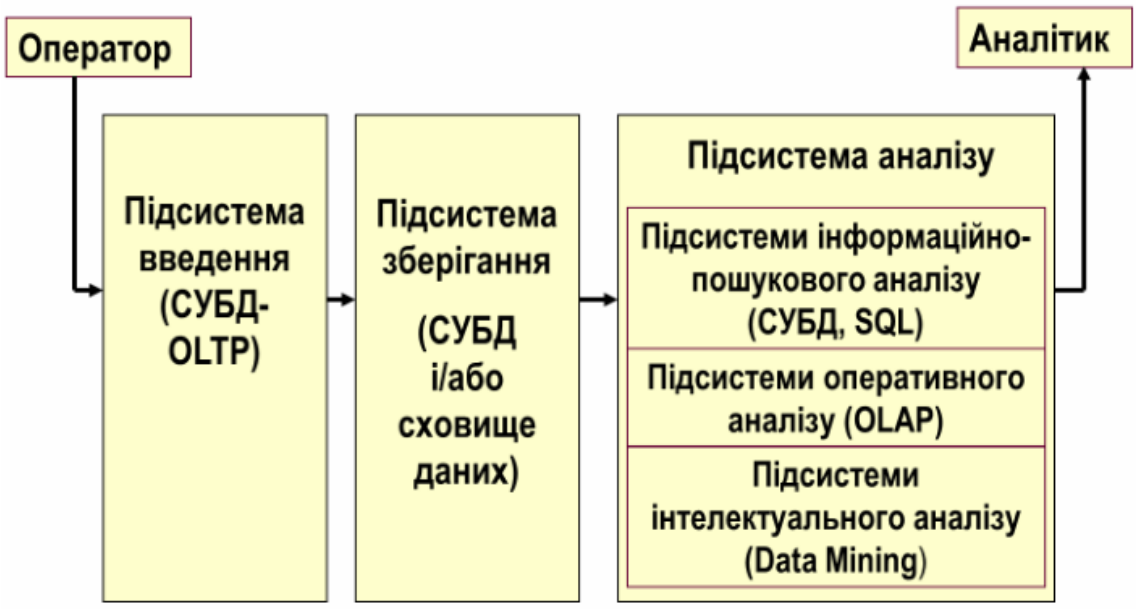



Рис. 1.1. Компоненти системи ІАД

1. База даних та OLTP – On-Line Transaction Processing – підсистема операційної (транзакційної) обробки даних, тобто СУБД, якими можуть бути реляційна база даних, сховище даних, транзакційна, об’ктно-орієнтована,

об’єктно-реляційна, просторова (Spatial databases), часова (Temporal databases), текстова або мультимедійна бази даних; всесвітня павутина, тощо.

2. Сервер бази даних або сховища даних, який відповідає за добування істотних даних на основі користувацького запиту.

3. База знань – знання про предметну область, за якими визначають шляхи пошуку та оцінюють корисність отриманих патернів.

4. Служба добування знань – містить набір функціональних модулів для здійснення власне процедур інтелектуального аналізу даних.

5. Модуль оцінки патернів – визначає міру корисності патернів.

6. Графічний користувацький інтерфейс.

Відповідно, основними типами задач аналізу даних є задачі опису та задачі прогнозування, тобто:

• Класифікація – процес знаходження моделей чи функцій, які описують та розрізняють класи для прогнозування класу довільно заданого об’єкта з відомими атрибутами на основі навчаючої вибірки;

• Кластеризація – виявлення ознак, за якими можна буде здійснювати класифікацію, шляхом групування “схожих” між собою об’єктів, генерування міток класів на основі відстаней між об’єктами;

• Регресія – встановлення залежностей неперервних результуючих змінних від вихідних;

• Асоціація – пошук закономірностей між декількома подіями, що відбуваються одночасно;

• Послідовність – пошук часових закономірностей між транзакціями;

• Прогнозування – оцінювання пропущених або майбутніх значень цільових чисельних показників;

• Виявлення відхилень або викидів;

• Оцінювання – передбачення неперервних значень ознаки;

• Аналіз зв’язків – пошук залежностей у наборі даних;

• Візуалізація – створення графічного образу аналізованих даних;

• Підведення підсумків – опис конкретних груп об’єктів з аналізованого набору;

• Еволюційний аналіз – опис та моделювання регулярностей та трендів для об’єктів, чия поведінка змінюється у часі.


Рис. 1.2. Data Mining як міждисциплінарна галузь

На рис. 1.2 наведено неповний перелік дисциплін, що вплинули і продовжують впливати на формування змісту та методів аналізу даних, а на рис. 1.3 наведено основні галузі аналізу даних:

• OLAP – On-Line Analitical Processing – технологія оперативної аналітичної обробки багатовимірних даних.

• Data mining – дослідження та виявлення “машиною” нових нетривіальних, необхідних для прийняття рішень, практично корисних, доступних для інтерпретації людиною знань, прихованих у “сирих” даних.

• Візуалізація – представлення багатовимірного розподілу даних на двовимірній площині, при якому відображено, принаймні якісно, основні закономірності вхідного розподілу – його кластерна структура, топологічні

особливості, внутрішні зв’язки між ознаками, інформація про розташування даних у вхідному просторі тощо.


Рис. 1.3. Основні галузі аналізу даних

## 1.2. Сутність багатовимірних даних та їх опис

Наразі досить розповсюдженою є концепція інформаційної ієрархії DIKW (data – information – knowledge – wisdom) (рис. 1.4). Попри те, що ця модель не уникла критики, вона дає уявлення про першооснову виникнення знань, ілюструючи поступове ускладнення пізнавального процесу.

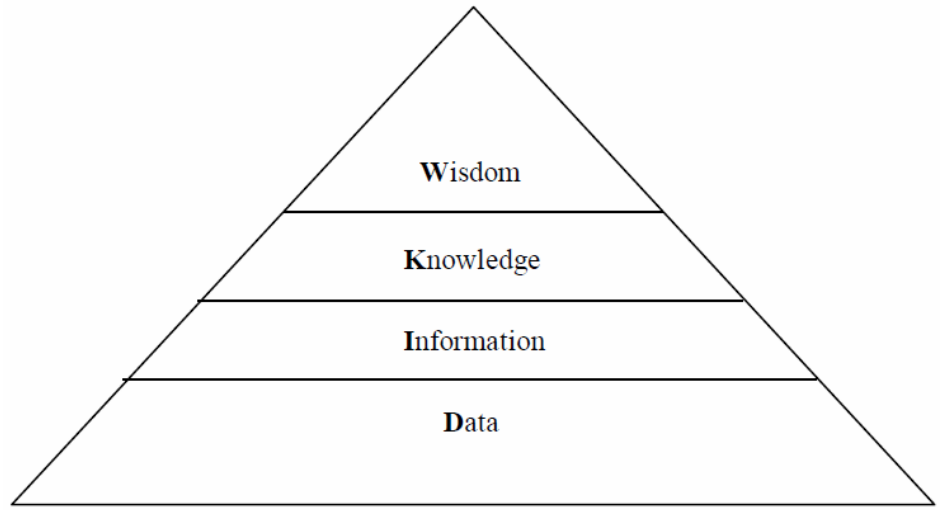



Рис. 1.4. Інформаційна ієрархія DIKW

Як видно з рисунка 1.4, в основі інформаційної піраміди перебувають дані (data), котрі можуть бути трансформовані в інформацію (information), з

якої, в свою чергу, можуть бути здобуті знання (knowledge) і, нарешті, останнім кроком є перетворення знань у мудрість (wisdom). Кожен вищий рівень ієрархії є більш довершеним, ніж попередній, завдяки додаванню певних властивостей до рівня нижчого порядку – дані перетворюються на інформацію завдяки додаванню контексту; знання додає опцію «як» (механізм використання); мудрість – «коли» (умови використання).

Саме поява певного контексту, у якому розуміють дані, і робить їх інформацією. Зрозуміло, що сам факт наявності таких даних є недостатнім для їх ефективного використання, оскільки спочатку слід осягнути, що ж означають ці дані, тобто яку інформацію вони в собі несуть. Для цього потрібно застосувати відповідні методи обробки даних.

Усі можливі методи обробки даних поділяються на природні й технічні. Природні – це методи, засновані на органах чуттів людини; технічні методи включають апаратні (обробка здійснюється за допомогою спеціальних пристроїв – телефонів, рентгенівських апаратів, мікроскопів тощо) та програмні (за допомогою комп'ютерного коду).

Без змістовного тлумачення (поза контекстом) дані так і залишаться просто зафіксованими сигналами. Наприклад, маючи результати лабораторного аналізу, неможливо зробити висновок, чи здоровою є людина або ж якого лікування вона потребує (для цього мають бути застосовані такі методи, як зір, читання, логічне мислення, аналіз).

Важливим для нашого дослідження є поділ отриманої з даних інформації на аналогову та цифрову. Перша отримується завдяки застосуванню природних методів обробки даних і сприймається виключно людиною. Вона має неперервний характер і поділяється відповідно до п'яти органів людського чуття на візуальну, аудіальну, тактильну, нюхову, смакову. Цифрова інформація є дискретною і сприймається обчислювальною технікою.

Таким чином, інформація – це «… смисловий продукт взаємодії даних та адекватних їм методів». З одних і тих же даних, застосовуючи різні методи обробки, можна отримати різну інформацію.

Термінологічний стандарт ISO/IEC 2382-1 «Information technology – Vocabulary» (перегляд ISO/IEC 2382:2015), оперуючи поняттями трьох нижчих рівнів інформаційної ієрархії DIKW, дає наступні визначення даних, інформації та знання:

- дані – це подання інформації в певному формалізованому вигляді, придатному для передачі, інтерпретації чи обробки;

- інформація (у процесах її обробки) – це будь-який факт, поняття чи значення, отримані з даних, а також контекст, вибраний зі знань, чи контекст, асоційований зі знаннями;

- знання – це організоване, інтегроване зібрання фактів та узагальнень.

Дані, інформація та знання можуть існувати як до, так і після процесу їх обробки.

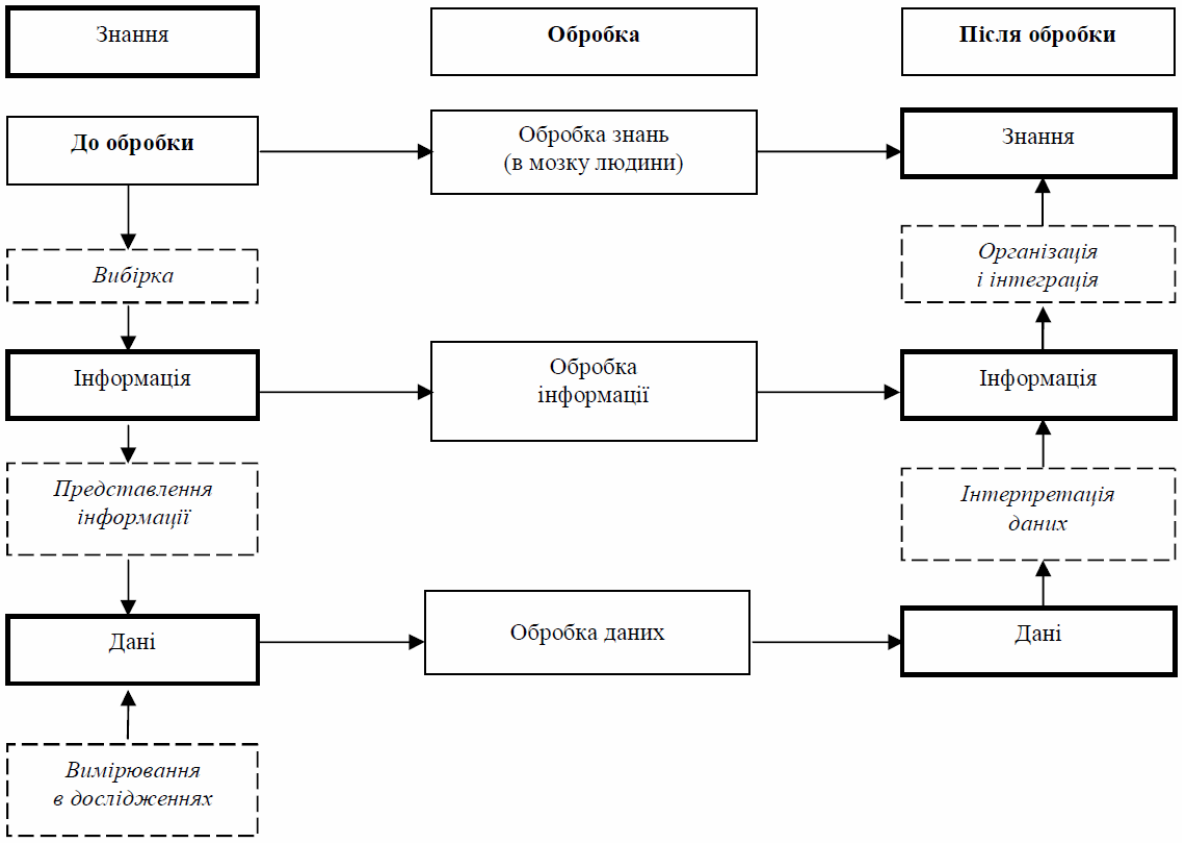



Рис. 1.5. Співвідношення між даними, інформацією та знаннями

Як видно з рисунка 1.5, дані можуть бути отримані двома шляхами: або за допомогою здійснення вимірювань в дослідженнях, або ж «низхідним» шляхом (у контексті моделі DIKW), виокремивши певну інформацію з уже наявних знань і певним чином її представивши. Так, в статистичній практиці розмежовуються поняття первинних (до процедури їх обробки) та статистичних (після обробки) даних.

Зауважимо, що обробка знань, на відміну від обробки даних та інформації, може здійснюватися лише завдяки свідомості людини. Зазвичай знання існують не в документах, а в індивідуальній чи колективній свідомості, з чого можна зробити висновок, що знання – «… це люди плюс інформація».

Розвиток саме технічних засобів надав людству нові можливості щодо фіксування, обробки, збереження та аналізу дедалі більших за обсягом і різноманітніших за природою і структурою масивів даних. З іншого боку, еволюція технологій спричинила і збільшення виробництва даних – їх нині продукують найрізноманітніші пристрої – від мобільних телефонів і подібних «дрібних» гаджетів до приладів рівня Слоанівського телескопа чи адронного колайдера.

«Цунамі» цифрових даних, що зародилося в середині 2000-х років отримало назву Big Data – «великі дані». Однією з особливостей феномену великих багатовимірних даних є те, що, належачи за своєю сутністю до предметної сфери фахівців з комп'ютерних технологій, цей концепт став не лише широко відомим в IT-спільноті, а й викликав велику зацікавленість серед представників бізнесу, мас-медіа та науковців.

Сьогодні термін Big Data вживається щонайменше у двох значеннях:

- як власне дані, що характеризуються великим обсягом і низкою інших властивостей;

- як технології роботи з такими даними.

Big Data визначають як надзвичайно великі набори даних, які можуть бути проаналізовані за допомогою комп'ютерів з метою виявлення певних

шаблонів (зразків, паттернів), тенденцій та зв’язків, особливо стосовно поведінки та взаємодії людей.

Наразі загальновизнаною є концепція великих даних згідно з якою основними їх характеристиками є обсяг (volume), швидкість (velocity), різноманітність (variety).

Коли йдеться про обсяги великих даних, дослідники зазвичай оперують такими незвичними для пересічного користувача одиницями виміру інформації, як терабайти, петабайти тощо. Відтак серед дефініцій Big Data зустрічаються такі, у яких йдеться про «незручно великі» обсяги: визначають Big Data як набори даних такого обсягу, що традиційні інструменти не здатні виконувати їх захват, управління та обробку за прийнятний для практики час.

Швидкість як характеристика великих даних означає, що дані стрімко накопичуються та потребують якомога швидшого реагування у вигляді обробки потоків даних у режимі реального часу.

Різноманітність Big Data полягає в тому, що це може бути як структурована, так і неструктурована інформація. До останньої належать текстові файли різноманітних документів, електронні листи, sms-повідомлення, аудіофайли, цифрові зображення, відеокліпи тощо.

Особливістю таких неструктурованих даних є те, що часто вони створюються користувачами Інтернету. Тому профілі користувачів у соціальних мережах, блоги, коментарі під статтями і новинами, пошукові запити тощо перетворюються на дані й за умови застосування відповідних методів аналізу можуть виявитися дуже інформативними з погляду наявності в них так званих «прихованих знань», котрі можуть бути отримані за допомогою інтелектуального аналізу даних – Data Mining.

## 1.3. Системи веб-сервісів для інтелектуального аналізу даних

Поступове збільшення обсягів інформації в сучасних інформаційних системах формує необхідність у розробці та впровадженні нових підходів до

аналізу великих обсягів даних (Великих даних, Big Data). Для аналізу Великих даних розробляються нові системи та методи інтелектуального аналізу даних, засновані на концепції розподілених та незалежних обчислень, у тому числі системи сервісів, що реалізовані у вигляді сервісно-орієнтованих архітектур (СОА) та Веб-СОА. У Веб-СОА стандартні методи аналізу даних представлені в вигляді окремих Веб-сервісів доступних для взаємодії з іншими сервісами чи Веб-агентами за допомогою стандартних протоколів мережі Інтернет.

Розглянемо поширені системи Веб-сервісів для інтелектуального аналізу даних.

Можна виділити наступні системи Веб-сервісів для інтелектуального аналізу даних: Weka4WS, Orange4WS, KNIME, MATLAB, ClowdFlows і DAME.

Доступ до Веб-сервісів, зазвичай, реалізують на базі протоколів Simple Object (Simple Object Access Protocol, SOAP) та Representational State (Representational State Transfer, REST). Призначення Веб-сервісів та варіанти взаємодії з ними описують на мові Web Service Definition Language (WSDL).

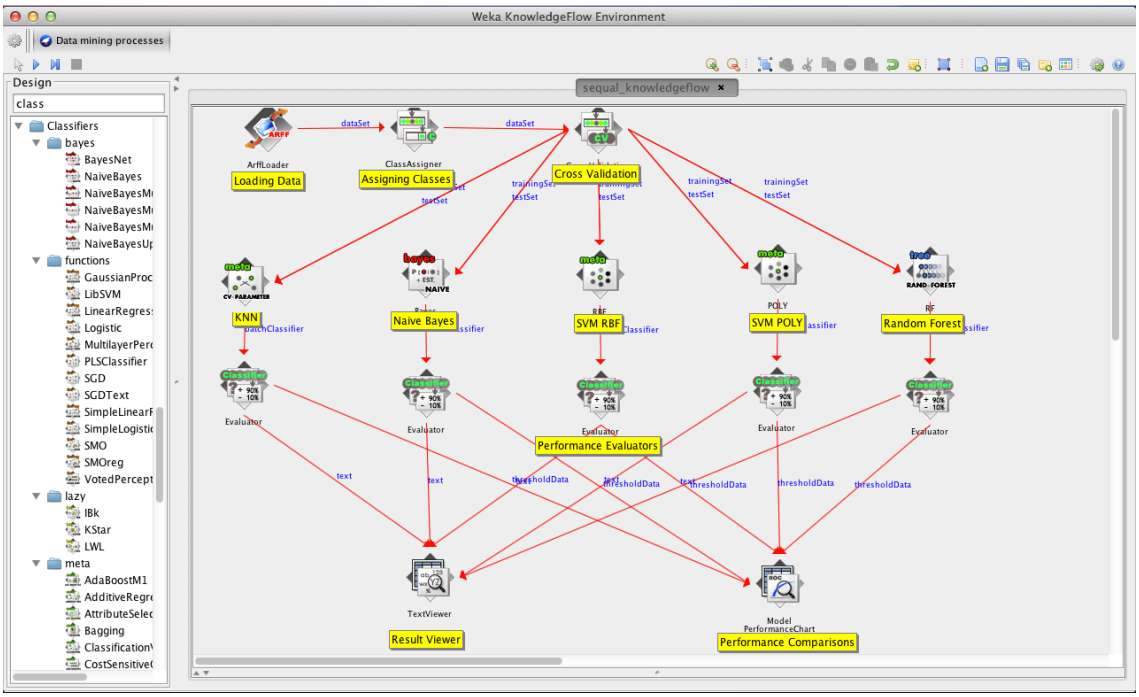



Рис. 1.6. Веб-сервіс інтелектуального аналізу даних Weka4WS

Система Weka4WS є розширенням пакету Weka, яке забезпечує виконання методів інтелектуального аналізу даних у вигляді Веб-сервісів на розподілених вузлах мережі Інтернет. Weka4WS побудована на базі Грід-сервісів за технологією віддаленого управління ресурсами (Web Services Resource Framework, WSRF) та пакету Globus для управління виконанням методів інтелектуального аналізу у вигляді потоків розрахунків (Workflows). Архітектура Weka4WS передбачає аналіз даних різними методами з фіксованими параметрами чи одним методом з різними параметрами на розподілених вузлах мережі Інтернет.

Фунціонал WEKA забезпечує як і попередню обробку даних так і достатню кількість операторів проте не дуже зрозумілий інтерфейс, на який скаржаться у відгуках та розбиття операторів на не інтуїтивні категорії йдуть у протиріччя із основними вимогами до інструментальних засобів побудови сценаріїв аналітики.

Система Orange4WS є розширенням пакету Orange. В порівнянні з Orange, Orange4WS включає в себе такі функції, як можливість застосування Веб-сервісів в якості компонентів потоків розрахунків.

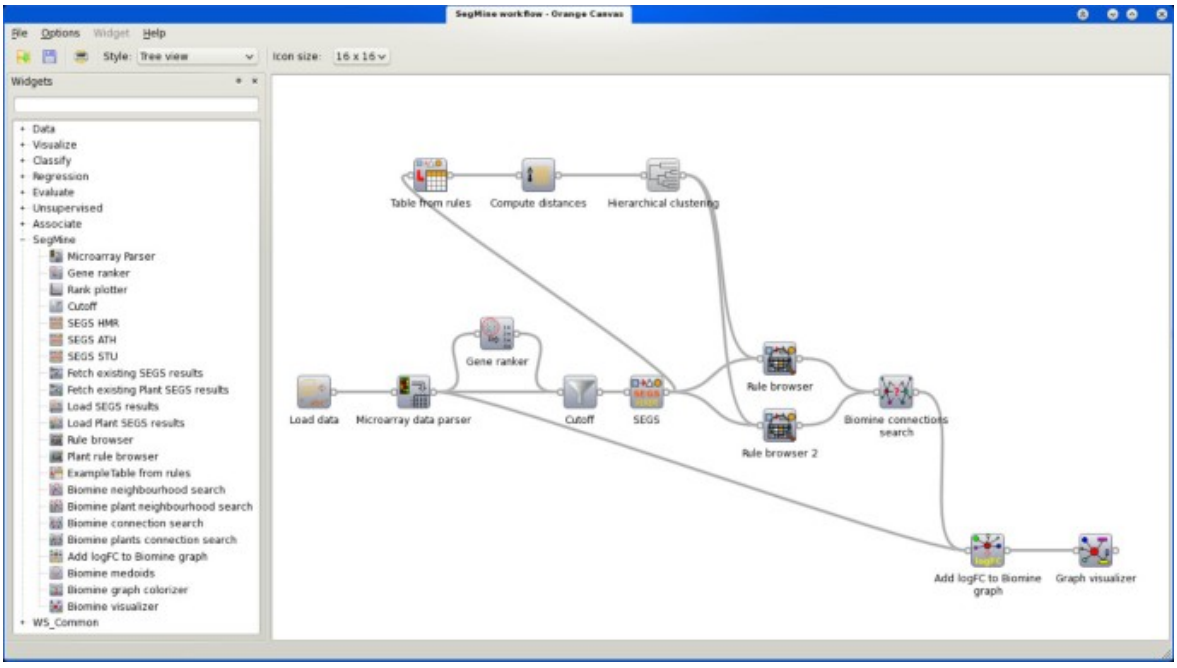



Рис. 1.7. Система інтелектуального аналізу Orange4WS

Потоки розрахунків мають вигляд онтологій, в яких описані типи та джерела даних, сервіси інтелектуального аналізу даних в абстрактному вигляді, зручному для автоматичного управління. В Orange4WS є можливість імпорту зовнішніх Веб-сервісів за їх WSDL-описом.

Orange - це візуалізація даних, машинне навчання та інструментарій вибору даних з відкритим кодом. У ньому представлено візуальне програмування, призначене для дослідницького аналізу даних та інтерактивної візуалізації даних.

Orange - це програмний пакет з відкритим кодом, випущений в рамках GPL. Версії до 3.0 включають основні компоненти в C++ із обгортками в Python, доступні на GitHub. Починаючи з версії 3.0, Orange використовує поширені бібліотеки з відкритим кодом Python для наукових обчислень, такі як numpy, sciру та scikit-learn, в той час як його графічний інтерфейс користувача працює в рамках платформи Qt між платформами.

Orange має наступну функціональність:

- віджети для введення даних, фільтрації даних, вибірки, імпутації, маніпулювання та вибору функції;

- віджети для загальної візуалізації (графічна скринька, гістограми, графік розсіювання) та багатоваріантну візуалізацію (мозаїчне відображення);

- перехресна валідація, процедури на основі вибірки, оцінка надійності та оцінка методів прогнозування;

- непідконтрольні алгоритми навчання кластеризації (k-засоби, ієрархічна кластеризація) та методи прогнозування даних (багатовимірне масштабування, аналіз основних компонентів, аналіз кореспонденції).

Приклад візуалізації даних за допомогою Orange представлено на рисунку 1.8.

Головним недоліком додатку є те, що програма підтримує цілий спектр віджетів, які потрібно завжди доставляти на вашу робочу машину (замість використання єдиної системи), а також відсутність розкладу заливок даних.

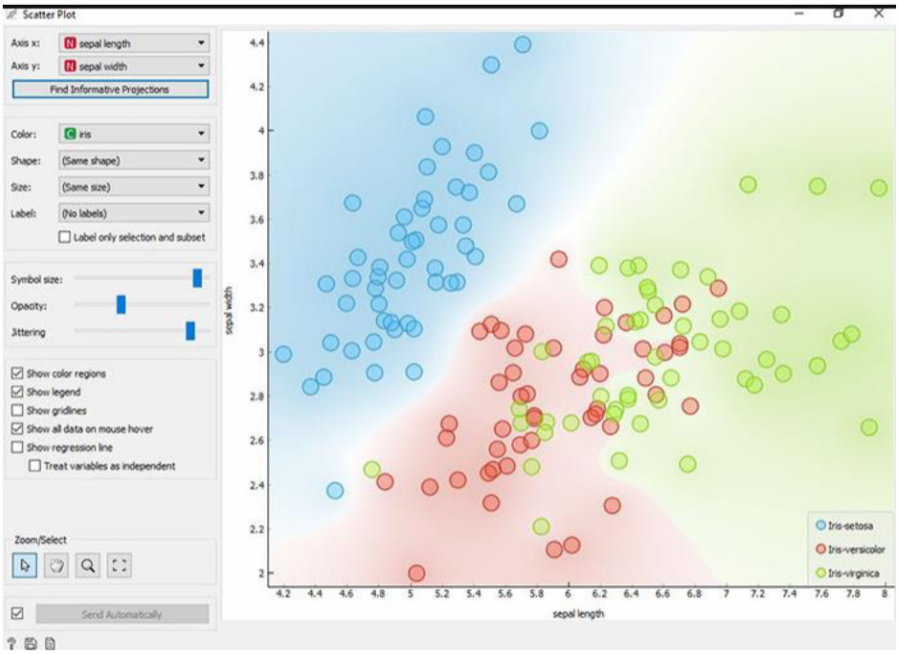



Рис. 1.8. Візуалізація даних за допомогою Orange

Система KNIME також заснована на архітектурі Веб-сервісів. Клієнтські вузли представлені у вигляді Веб-сервісів, які компонуються в потоки розрахунків. Потоки розрахунків включають вузли для попередньої обробки даних, побудови та візуалізації аналітичних моделей.

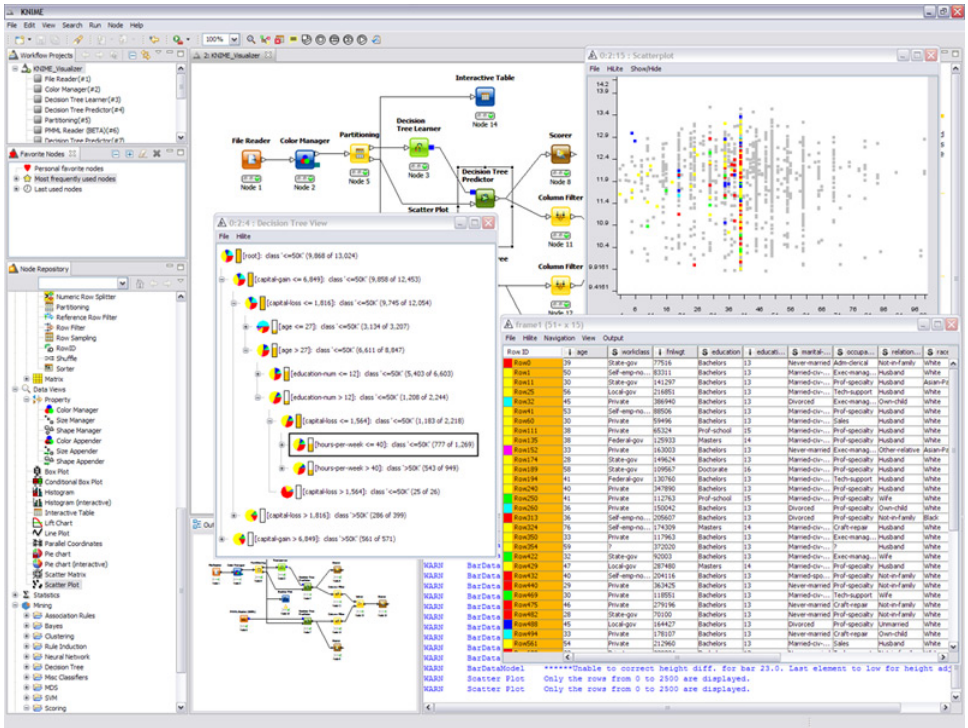


Рис. 1.9. Система KNIME

Система KNIME — це кросплатформене програмне забезпечення для побудови сценаріїв аналітики великих даних з частково відкритим вихідним кодом, розроблена та підтримувана однойменною компанією. Початковою метою було створення модульної, високо масштабованої та відкритої платформи інтелектуальної обробки даних не орієнтуючись на якусь конкретну область застосування. Програмне забезпечення має як безкоштовну версію, так і платну. Мова програмування на якій реалізована KNIME — Java.

KNIME може працювати із наступними джерелами даних:

- формат Xlsx;
- формат ARFF;
- формат XML;
- формат JSON;
- формат SQL Server;
- формат MongoDB.

В MATLAB реалізовані два типи взаємодії з Веб-сервісами – REST та SOAP. Для аналізу великих обсягів даних існує можливість створення Веб-сервісів та завантаження їх у хмарні системи.

Окрім розглянутих систем Weka4WS, Orange4WS та KNIME, розроблених у вигляді локального програмного забезпечення з графічним інтерфейсом, існує дві системи, які потребують встановлення браузера та наявності доступу до мережі Інтернет: ClowdFlows та DAME.

Система ClowdFlows розроблена у вигляді Веб-додатку для управління потоками розрахунків, які виконуються у хмарному середовищі. Система має відкритий програмний код для створення та спільного виконання потоків розрахунків в задачах інтерактивного машинного навчання та інтелектуального аналізу даних. ClowdFlows підтримує взаємодію з Веб-сервісами за протоколом SOAP.

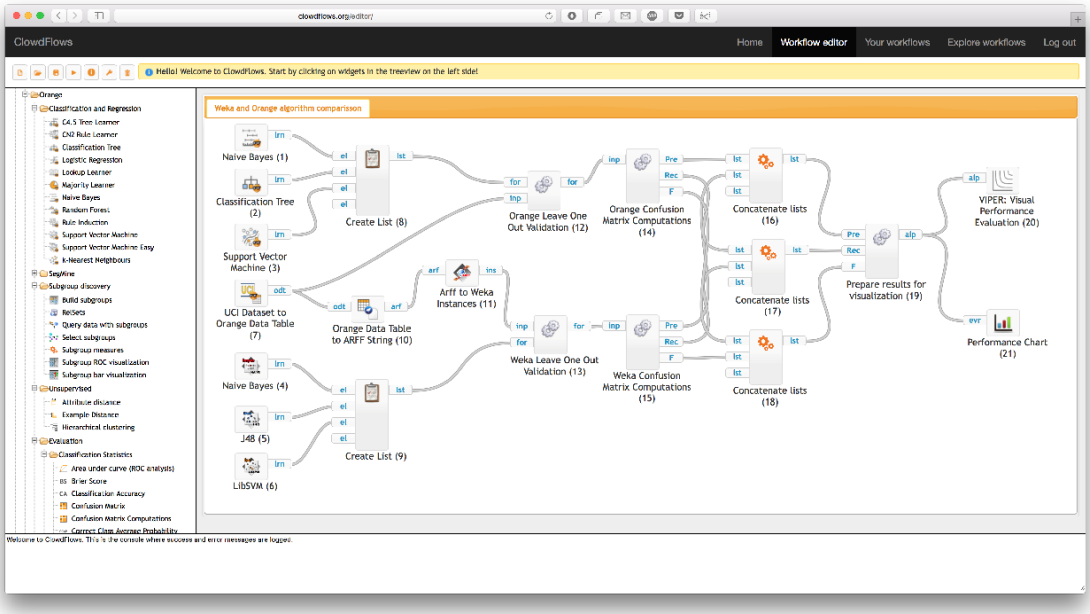



Рис. 1.10. Система ClowdFlows

Система DAME створена у вигляді пакету Веб-додатків для взаємодії з розподіленими середовищами. DAME забезпечує зручний доступ до великих обсягів даних у хмарних системах та включає Веб-сервіси для обробки та аналізу даних.

Alteryx - американська компанія з комп'ютерного програмного забезпечення. Для роботи із великими даними компанія пропонує чотири основні модулі — Connect, Promote, Server та Desinger. Саме останній із компонентів продукції компанії використовується для аналітики великих даних у простій та зручній візуальній формі за допомогою drag and drop редактора.

Програмне забезпечення надає оператори для роботи із різними типами структурованих файлів, як csv або excel, та бази даних Microsoft SQL Server та Oracle.

Для попередньої обробки даних в програмному забезпечені наявно дев’ятнадцять операторів, проте відсутня можливість для нормалізації вхідної інформації.

Серед алгоритмів та методів, які застосовані для аналізу великих даних були використані наступні:

- лінійна регресія;

- логістична регресія;

- наївний баєсів класифікатор;

- метод Random Forest;

- метод опорних векторів.

Для візуалізації у Alteryx Desinger використовується розвинута система генерації звітів, що дозволяє із широкого спектру заготовлених шаблонів візуалізувати інформацію багатьма способами та зберігати у різних форматів від png до pdf.

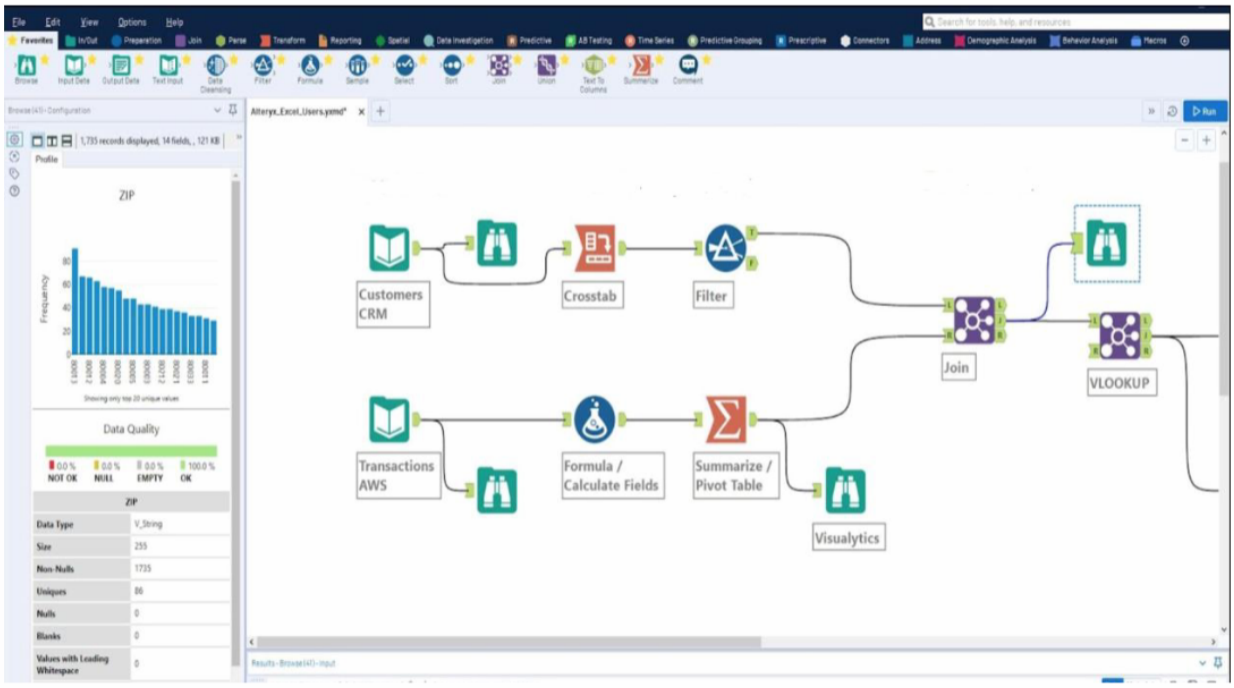



Рис. 1.11. Функціонал Alteryx Desinger

Основним недоліком програмного забезпечення Alteryx Desinger можна назвати те, що воно не є кросплатформеним, а запускається лише на операційній системі Windows. Більш того, Alteryx Desinger — це проект із закритим вихідним кодом, з чого випливає що подивитися як реалізовані алгоритми для аналізу та обробки даних немає можливості. А отже неможливо сказати із якою ефективністю працює система.

Порівняємо системи за зручністю їх застосування.

Порівняльний аналіз. Для порівняння систем Веб-сервісів були вибрані наступні критерії:

1. SOAP і REST (C1) – підтримка популярних протоколів доступу до Веб-сервісів. Майже всі системи підтримують SOAP, у системі DAME реалізована взаємодія за протоколом REST. Лише в MATLAB реалізовані обидва типи протоколів.

2.Кросплатформеність (C2) – можливість встановлення на операційних системах Windows, Linux і Mac. Усі системи є кросплатформеними.

3.Хмарні обчислення (C3) – підтримка хмарних обчислень та Грід-обчислень при вирішенні складних і трудомістких задач інтелектуального аналізу даних. Лише 3 системи Weka4WS, MATLAB й ClowdFlows підтримують хмарні та Грід-обчислення.

4.Аналітична універсальність (C4) – можливість застосовування кількох різних методів інтелектуального аналізу даних в одному потоці розрахунків. Стандартні три групи методів інтелектуального аналізу реалізовані в системах Weka4WS, Orange4WS, KNIME та MATLAB.

5.Потоки розрахунків (C5) – можливість виконання потоків розрахунків, збереження результатів та проведення експериментів. Потоки розрахунків реалізовані в чотирьох системах Weka4WS, Orange4WS, KNIME та ClowdFlows.

6.Імпорт Веб-сервісів (C6) – підтримка імпорту сторонніх Веб-сервісів з мережі Інтернет.

7.Відкритий програмний код (C7) – можливість розширення та поліпшення коду. Чотири системи Weka4WS, Orange4WS, KNIME і ClowdFlows мають відкритий програмний код.

8.Веб-інтерфейс (C8) – можливість роботи у Веб-браузерах. Дві системи ClowdFlows та DAME реалізовані у вигляді Веб-додатків, проте вимагають підключення до мережі Інтернет.

9.Веб-збереження (С9) – можливість збереження даних у Веб-сховищі, що дозволяє виконувати різні експерименти з даними без повторного завантаження. Така особливість наявна лише в DAME.

У таблиці 1.1 наведені результати порівняння систем Веб-сервісів для інтелектуального аналізу даних.

Таблиця 1.1.

Результати порівняння систем Веб-сервісів для інтелектуального аналізу даних

| | C1 | | C2 | C3 | C4 | | | | C5 | C6 | C7 | C8 | C9 | Σ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *SOAP* | *REST* | *Крос-платформеність* | *Хмарні обчислення* | *Класифікація* | *Кластеризація* | *Зменшення розмірності* | *Універсальність системи* | *Потоки розрахунків* | *Імпорт Веб-сервісів* | *Відкритий програмний код* | *Веб-інтерфейс* | *Веб-збереження* | |
| **Weka4WS** | + | – | + | + | + | + | + | + | + | – | + | – | – | **9** |
| **Orange4WS** | + | – | + | – | + | + | + | + | + | + | + | – | – | **9** |
| **KNIME** | + | – | + | – | + | + | + | + | + | + | + | – | – | **9** |
| **MATLAB** | + | + | + | + | + | + | + | + | – | + | – | – | – | **9** |
| **ClowdFlows** | + | – | + | + | + | + | – | – | + | + | + | + | – | **9** |
| **DAME** | – | + | + | – | + | + | – | – | – | – | – | + | + | **6** |
| | **5** | **2** | **6** | **3** | **6** | **6** | **4** | **4** | **4** | **4** | **4** | **3** | **1** | |

Отже, результатів порівняльного аналізу слідує, що розглянуті системи веб-сервісів для інтелектуального аналізу даних поки не відповідають усім критеріям зручності застосування, так, у системах Weka4WS, Orange4WS, KNIME і ClowdFlows реалізовані 9 з 13 пунктів порівняльного аналізу та у системі DAME лише 6 з 13 пунктів (табл. 1.1). У системах Orange4WS та KNIME відсутня підтримка хмарних обчислень, яка полегшує та пришвидшує роботу з великими обсягами даних. Основною перевагою системи Weka4WS порівняно з системою ClowdFlows є універсальність системи за кількістю реалізованих методів інтелектуального аналізу даних, а порівняно з системою MATLAB – підтримка виконання потоків розрахунків та наявність відкритого програмного коду, у свою чергу головною перевагою

цих систем порівняно з системою Weka4WS є можливість імпорту зовнішніх Веб-сервісів.

Отже, після порівнянь систем веб-сервісів, найбільш перспективною для вирішення комплексних задач з аналізу великих багатовимірних даних є система Weka4WS, обчислювальні можливості якої обмежуються лише кількістю.

### 1.4. Методики аналізу та обробки багатовимірних даних

Під аналізом багатовимірних даних розуміється як аналіз масивів даних в рамках можливостей персонального комп'ютера, так і в рамках можливостей систем керування базами даних, при цьому як в першому, так і в другому випадку при формуванні статистики і візуалізації виникають певні труднощі, які полягають в необхідності забезпечення скоординованої роботи комп'ютерних програм на десятках, сотнях або навіть тисячах серверів.

До основних способів аналізу багатовимірних масивів інформації відносять такі:

– глибинний аналіз, класифікація даних. Ці методики прийшли з технологій роботи зі звичайною структурованою інформацією в невеликих масивах. Однак в нових умовах використовуються вдосконалені математичні алгоритми, засновані на досягненнях в цифровій сфері;

– краудсорсінг. В основі цієї технології можливість отримувати і обробляти потоки в мільярди байт з багатьох джерел. Кінцеве число «постачальників» не обмежується нічим. Хіба тільки потужністю системи;

– спліт-тестування. З масиву вибираються кілька елементів, які порівнюються між собою по черзі «до» і «після» зміни. A\B тести допомагають визначити, які чинники мають найбільший вплив на елементи. Наприклад, за допомогою спліт-тестування можна провести величезну кількість ітерацій поступово наближаючись до достовірного результату;

– прогнозування. Аналітики намагаються заздалегідь задати системі ті чи інші параметри і в подальшій перевіряти поведінку об’єкта на основі надходження великих масивів інформації;

– машинне навчання. Штучний інтелект в перспективі здатний поглинати і обробляти великі обсяги несистематизованих даних, згодом використовуючи їх для самостійного навчання;

– аналіз мережевої активності. Методики big data використовуються для дослідження соцмереж, взаємовідносин між власниками аккаунтів, груп, спільнотами. На основі цього створюються цільові аудиторії за інтересами, геолокації, віком і іншим метрик.

*Нечітка кластеризація*

В області аналізу даних нечітке моделювання часто дозволяє отримувати більш адекватні результати в порівнянні з результатами, які ґрунтуються на використанні традиційних аналітичних моделей і алгоритмів.

Нечітка множина є сукупністю елементів довільної природи, щодо яких не можна з повною певністю стверджувати – чи належить той чи інший елемент розглянутої сукупності даній множині чи ні.

Взаємозв’язок між кластерним аналізом і теорією нечітких множин засновано на тій обставині, що при вирішенні задач структуризації складних систем більшість об’єктів виявляються розмитими за своєю природою. Ця розмитість полягає в тому, що перехід від приналежності до неналежності елементів до даних класах швидше поступовий, ніж стрибкоподібною.

Вимога знаходження однозначної кластеризації елементів досліджуваної проблемної області є досить грубим і жорстким, особливо при вирішенні погано або слабо структурованих задач системного аналізу. Методи нечіткої кластеризації послаблюють цю вимогу. Ослаблення вимоги здійснюється за рахунок введення в розгляд нечітких кластерів і відповідних їм функцій приналежності, які приймають значення з інтервалу [0, 1]. На

рисунку 1.112 зображена нечітка кластеризація з різними ступеням приналежності спостережень.

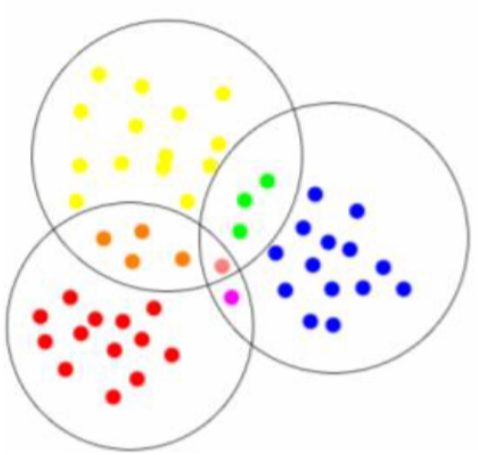


Рис. 1.12. Нечітка кластеризації

Для будь-якої міри схожості величина приналежності спостереження кластеру залежить від схожості об’єкта і прототипу цього кластера. В разі якщо мірою подібності є відстань, то величина приналежності об’єкта обернено пропорційна його відстані до центроїда кластера. Сума приналежності спостереження кластерам в будь-який момент часу повинна бути дорівнює 1.

Таким чином, в загальному випадку завданням нечіткої кластеризації є знаходження нечіткого розбиття або нечіткого покриття безлічі елементів досліджуваної сукупності, які утворюють структуру нечітких кластерів, присутніх в розглянутих даних. Це завдання зводиться до знаходження ступенів належності елементів шуканим нечітким кластерам, які в сукупності і визначають нечітке розбиття або нечітке покриття вихідної множини розглянутих елементів.

Основні ідеї алгоритму для вирішення завдання нечіткої кластеризації мають назву нечітких с-середніх (FCM). Поряд з традиційним імовірнісним підходом до нечіткої кластеризації, коли кожен об’єкт з певною ймовірністю належить до кожного з кластерів, існує ймовірнісна підхід до кластерного аналізу. Можливісна кластеризації також розглядає нечіткі кластери і

відповідні їм функції приналежності, які беруть значення з інтервалу [0, 1]. Різниця полягає в тому, що імовірнісна кластеризація має на увазі наявність суворого обмеження, що сума приналежності об’єкта до всіх кластерів дорівнює 1, а можливісна кластер ний аналіз не має на увазі подібного обмеження.

Перевага можливісного кластерного аналізу над імовірнісним полягає в тому, що об’єкти, які мають низький рівень подібності з будь-яким з кластерів, будуть мати значення приналежності близьке нулю для всіх кластерів, в той час як імовірнісний нечіткий кластерний аналіз буде явно віддавати перевагу одному або декільком кластерам (хоча всі вони повинні бути досить погані).

Алгоритм FCM має ітеративний характер послідовного поліпшення деякого нечіткого розбиття, яке задається користувачем або формується автоматично за деяким евристичним правилом. На кожній з ітерацій рекурентно перераховуються значення функцій приналежності об’єктів нечітким кластерам і їх типові представники (центроїди).

*Модель бінарної логістичної регресії*

У математичній статистиці логістична регресія є широко використовуваною статистичною моделлю, яка використовує логістичну функцію для моделювання залежності вихідної змінної від набору вхідних в разі, коли перша є бінарної.

Це різновид множинної регресії, загальне призначення якої полягає в аналізі зв’язку між декількома незалежними змінними і залежною змінною.

Регресія в загальному вигляді застосовується, коли вхідні і вихідна змінні безперервні. А логістична регресія кращим чином підходить, коли вихідна змінна приймає тільки два значення.

Значення факторів в моделях бінарного вибору повинні бути виміряні в кількісній шкалі. Також в моделі бінарного вибору можна включати в якості факторів категоріальні змінні. Для моделювання ймовірності дихотомічної

залежної змінної підбирають спеціальну монотонно зростаючу функцію, яка може приймати значення в межах від 0 до 1.

Є спеціальна функції в моделях бінарного вибору зазвичай використовують:

– логістичну функцію;

– функцію стандартного нормального розподілу.

За допомогою методу бінарної логістичної регресії (рис. 1.13) можна досліджувати залежність дихотомічних змінних (бінарних, що мають лише два можливих значення) від незалежних змінних, що мають будь-який вид шкал.

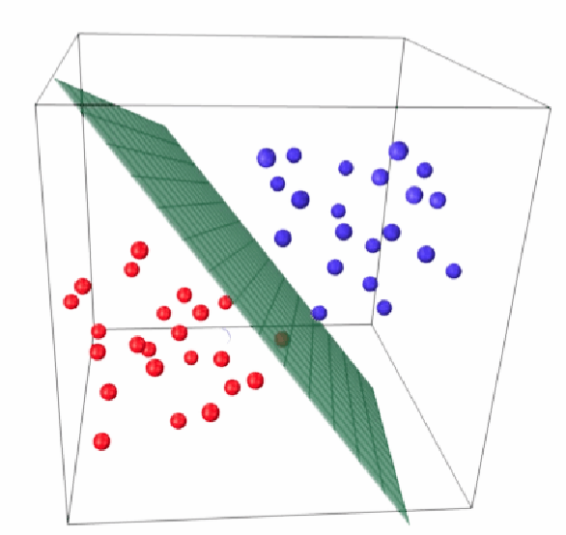


Рис. 1.13. Приклад роботи бінарної логістичної регресії

Як правило, у випадку з дихотомічними змінними мова йде про деяку подію, яка може відбутися або не відбутися; бінарна логістична регресія в такому випадку розраховує ймовірність настання події в залежності від значень незалежних змінних.

Імовірність настання події для деякого випадку розраховується за формулою

$$p = \frac{1}{1 + e^{-z}},$$

де $z= b1*X1+b2*X2+…+bn*Xn$ + a, $X_1$ – значення незалежних змінних; b1 – коефіцієнти, розрахунок яких є задачею бінарної логістичної регресії; a – деяка константа.

Якщо для p вийде значення менше 0,5, то можна припустити, що подія не настане; в іншому випадку передбачається настання події.

За допомогою логістичної регресії прогнозується ймовірність відгуку для залежної змінної від включених в модель незалежних змінних. На основі прогнозних значень ймовірності можна зробити класифікацію всіх спостережень на дві групи. Окремим аналізом при побудові моделі логістичної регресії є аналіз ROC-кривих (Receiver Operator Characteristic). ROC-аналіз дозволяє вибрати оптимальне значення порогового значення ймовірності для класифікації. ROC-крива – крива, яка використовується для представлення результатів бінарної класифікації та оцінки ефективності класифікації.

## Висновки до розділу

В даному розділі проведено аналіз предметної області застосування інтелектуального аналізу, описано сутність багатовимірних та великих даних та наведені основі програмні інструменти виконання інтелектуального аналізу даних. Також наведений перелік критеріїв, які необхідно враховувати при побудові інструментальних засобів аналітики сценаріїв великих даних та проведений аналіз існуючих програмних засобів. Були виділені основні переваги та недоліки.

# РОЗДІЛ 2. МОДЕЛІ ТА МЕТОДИ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ БАГАТОВИМІРНИМХ ДАНИХ

## 2.1. Дослідження та опис методів інтелектуального аналізу даних

Алгоритми класичної математичної статистики довгий час, як основні, підтримували концепцію усереднення з вибірки, що зводиться до операцій над фіктивними величинами (типу середньої температури аудиторій в усіх приміщеннях університету, середньої висоти будинку міста).

Становлення, розвиток Data Mining обумовлені низкою чинників, основними серед яких є: покращення програмного забезпечення управлінських процесів; удосконалення технологій зберігання і накопичення даних; можливість акумулювання великої кількості інформації в спеціальних базах даних; вдосконалення алгоритмів обробки інформації.

Зазначене обумовило створення інтелектуалізованих інформаційних систем, які спрямовані на підтримку управлінських заходів підприємств та сприяло розвитку систем, в які закладені спеціальні алгоритми прогнозування та планування господарської діяльності підприємства (в т.ч. і рівня конкурентоспроможності).

Практична реалізація методів ІАД відбувається на основі концепції шаблонів (паттернів). Шаблони являють собою неочевидні та несподівані закономірності та властивості в аних (складові прихованих даних), відображають елементи багатосторонніх відносин між даними, що обрані для опису економічного процесу. Пошук шаблонів реалізується методами емпіричних припущень про структуру вибірки, значень показників. Використання ІАД полягає у нетривалості обробки та пошуку даних.

До того ж сирі дані, в процесі їх технологічної обробки формують комплексні масиви корисної інформації. Отже, ці методи та аналітично –

програмні системи, що їх використовують, формуються на основі використання сучасних технологічних підходів щодо збирання, нагромадження та моніторингу інформації, перетворення її на знання (Knowledge), що є зрозуміле і доступне для користувача.

До методів та алгоритмів ІАД належать наступні: ШНМ, дерева рішень, символьні правила, лінійна регресія, методи к-найближчого сусіда, опорних векторів, байєсові мережі, кластерного аналізу, методи асоціативних правил тощо. Розглянемо деякі з них.

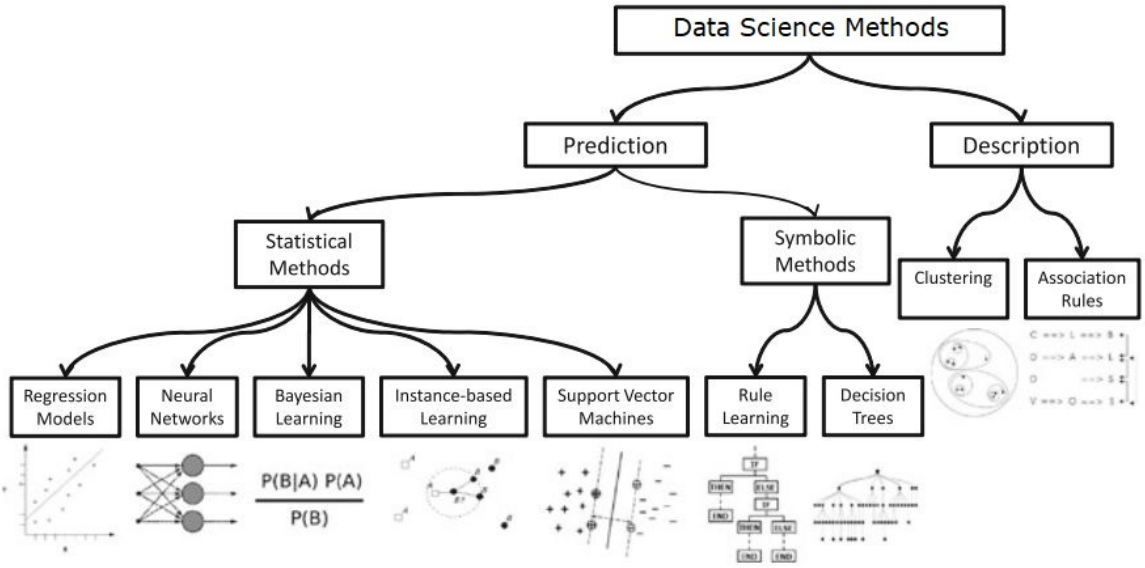



Рис. 2.1. Основні методи інтелектуального аналізу даних

**Класифікація (Classification).** Це найпростіша і найпоширеніша задача ІАД. В результаті розв’язання виявляються певні ознаки, що характеризують досліджувані групи об'єктів, поділяючи їх на класи. Якщо класів – два, то це бінарна класифікація. Для вирішення задач класифікації застосовують методи: к-ближнього сусіда (к-Nearest Neighbor); байєсових мереж (Bayesian Networks); індукції дерев рішень; нейронних мереж (neural networks).

**Кластеризація (Clustering).** Кластеризація є логічним продовженням ідеї класифікації. Це є складніша задача. Особливість кластеризації полягає у

тому, що класи об'єктів в даній задачі є не визначеними і нам потрібно розбити об'єкти на групи. Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу завдань навчання без вчителя. Головна відмінність кластеризації від класифікації полягає в тому, що перелік груп чітко не заданий і визначається в процесі роботи алгоритму.

Застосування кластерного аналізу в загальному виді зводиться до наступних етапів:

1. Ідентифікація вибірки об'єктів для кластеризації.

2. Визначення множини змінних, за якими планується проводити оцінку об'єктів у вибірці. При необхідності – нормалізація значень змінних.

3. Обчислення значень тієї або іншої міри схожості між об'єктами.

4. Застосування одного з методів кластерного аналізу для створення груп подібних об'єктів - кластерів).

5. Перевірка вірогідності результатів кластерного розв'язку.

Після одержання й аналізу результатів можливе коригування обраної метрики й методу кластеризації до одержання оптимального результату.

**Асоціація (Associations).** У процесі розв’язання задачі пошуку асоціативних правил відшукуються закономірності міжзв'язаними подіями в датасеті. Тут пошук не виявлених закономірностей виконується не на основі характеристик об'єкта, що аналізується, а між кількома подіями, які відбуваються одночасно. Найвідоміший алгоритм розв’язку задачі пошуку асоціативних правил – алгоритм Apriori.

**Послідовність (Sequence),** або послідовна асоціація (sequential association). Послідовність дає змогу знайти тимчасові закономірності між транзакціями. Задача послідовності подібна до асоціації, але її метою є встановлення закономірностей між подіями які пов'язаними між собою в часі тобто, що відбуваються з деяким певним інтервалом у часі.

**Прогнозування (Forecasting).** В задачах прогнозування оцінюються пропущені або можливі майбутні значення цільових числових показників. У завданнях прогнозування нам доступна лише інформація за попередні

періоди, проте в задачах класифікації послідовностей нам зазвичай доступна інформація з обох сторін від розглянутого періоду. Традиційні РНМ не дозволяють використовувати таку інформацію. Схема двобічної мережі LSTM представлена на рис. 2.2. У такій мережі вхідна послідовність обробляється з двох кінців одночасно, а виходи нейронів додаються, множаться або конкатенуються.

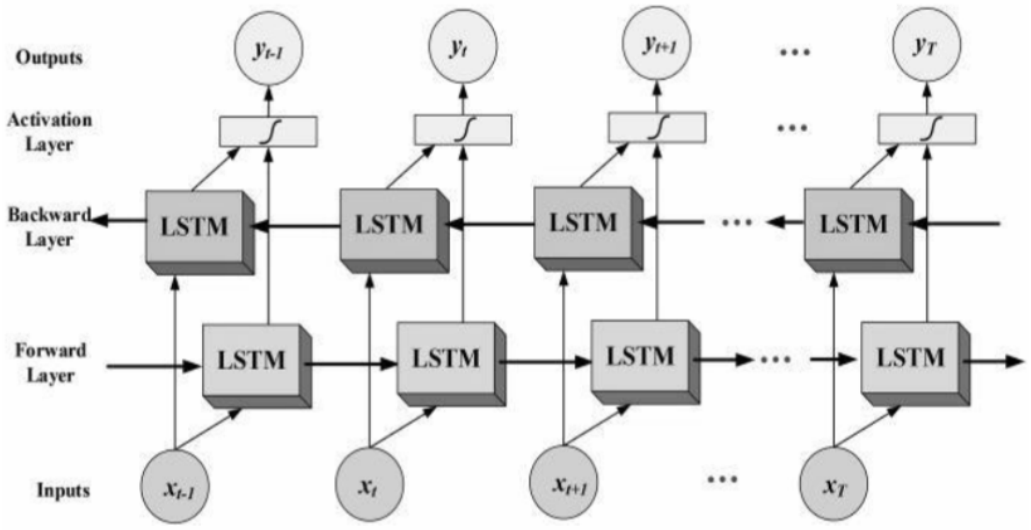



Рис. 2.2. Архітектура двобічної мережі LSTM

Для розв’язання таких задач широко застосовуються методи математичної статистики, нейроні мережі тощо.

**Візуалізація (Visualization, Graph Mining).** Для розв’язання задачі візуалізації використовуються графічні методи, що показують наявність закономірностей у даних, це можуть бути даних в 2D- і 3D-вимірах.

**Підведення підсумків (Summarization)** – це задача, основна мета якої полягає в описі певних груп об'єктів, що аналізуються.

**Прогностичне моделювання (Predictive Modeling).** Друга стадія ІАД – прогностичне моделювання – використовує результати роботи першої стадії і охоплює такі дії:

• прогнозування невідомих значень (outcome prediction);

• прогнозування розвитку процесів (forecasting).

Закономірності, отримані на цій стадії, формуються від часткового до загального. У результаті ми одержуємо деяке загальне знання про деякий клас об'єктів на підставі дослідження окремих представників цього класу.

Прогностичне моделювання, навпаки, дедуктивне. Закономірності, отримані на цій стадії, формуються від загального до часткового. Тут ми одержуємо нове знання про деякий об'єкт або ж групу об'єктів на підставі:

• знання класу, до якого належать досліджувані об'єкти;

• знання загального правила, що діє в межах цього класу об'єктів.

**Аналіз виключень (forensic analysis).** На третій стадії ІАД аналізуються виключення або аномалії, виявлені у знайдених закономірностях. Дія, що виконується це виявлення певних відхилень (deviation detection). Даний етап аналізу виключень часто застосовується на стадії як очищення даних.

## 2.2. Опис процесів організації та обробки багатовимірних даних

Цінність великих «сирих» даних визначається нашою здатністю вилучати з них «сенс», корисний за змістом і зручний за формою. Практика вимагає виділяти цінний екстракт швидко, використовуючи «свіжі» дані. Коли сукупність доступних даних охоплює екстремальне широкий спектр інформації, фірма (організація) може виконувати багато оперативних функцій автоматизовано, майже повністю на основі багатовимірних даних. Отже, треба будувати замкне ний комп'ютеризований цикл технологій – від збору даних до кінцевого застосування результатів (рішень, керування). «Непрозорі» й не-комп'ютерні процедури виносяться за межі «оперативного» циклу керування. (За штабами фірми залишаються функції нагляду (супервізія) та вищий рівень керування.) Виконання аналітичного завдання завершується видачею моделі або результату в формі, придатній для кінцевого застосування. (Вживають термін «actionable outputs».) Такий результат може використовуватися протягом певного періоду, коли

виконується «короткий» цикл аналітики (для керування використовують «свіжі» дані звуженої номенклатури). Схема циклів життя великих багатовимірних даних зображена на рис. 2.3.

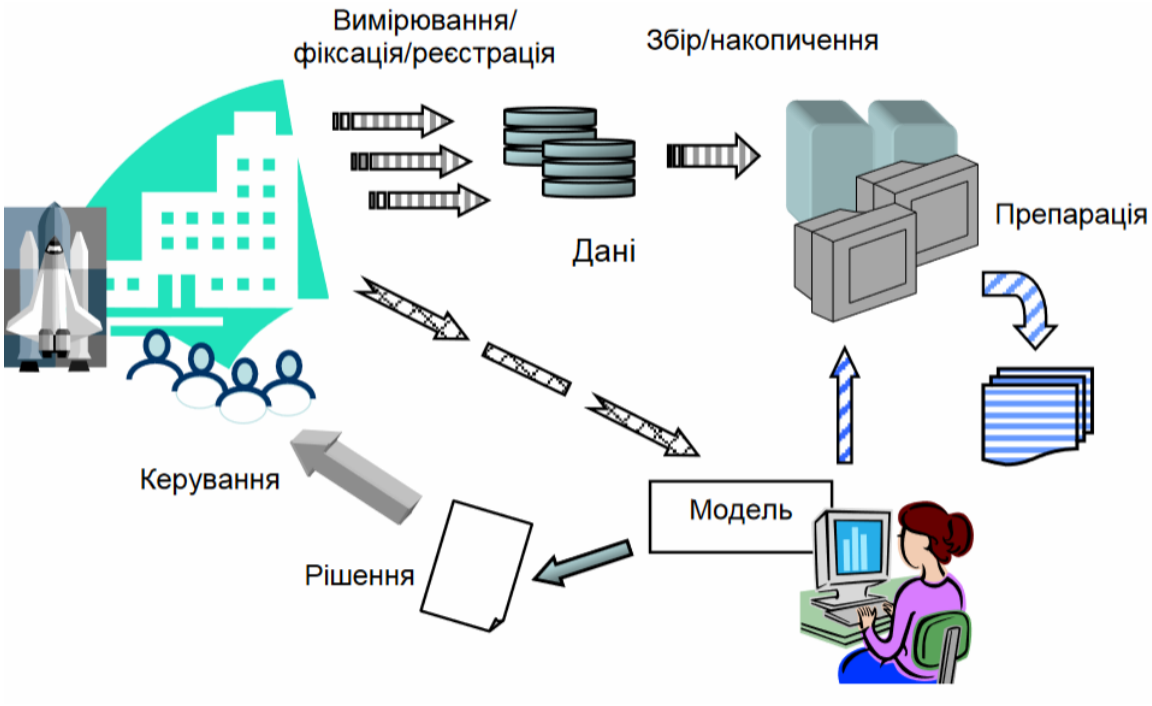



Рис. 2.3. Цикл використання багатовимірних даних

Оскільки аналітика використовує переважно статистичні методи, дані мають складатися з списку випадків (прикладів), що характеризують однотипні об'єкти або той самий об'єкт у варіабельних умовах. Випадки можуть трактуватися як екземпляри популяції, прецеденти, транзакції, цикли та періоди функціонування. (Існують дані, де поняття випадків та прикладів не збігаються). Більшість традиційних методів аналізу потребують, щоб дані всіх випадків складалися з єдиного набору атрибутів і збиралися за єдиною схемою вимірювання. Більшість класичних методів й процедур аналізу даних розраховані на зручно форматовані дані (зазвичай – у формі таблиці), що вміщуються в пам'яті комп'ютера. Натомість ВД наповнені переважно «сирими», різнорідними, неузгодженими, невпорядкованими та неструктурованими даними. Інформація щодо певного випадку може знаходитися у різних файлах і сховищах. Іноді доводиться розглядати як

«випадок» не тільки вектор чисел, а й цілий образ, текст, структуру і т. д. В деяких даних неясно, як розрізнити і виділити окремі випадки.

Процес аналітики включає два етапи:

1) доставка та компіляція даних (пошук, добір, фільтрація, агрегація, комплектування, інтеграція, зменшення розмірності, синхронізація, переформатування);

2) власне глибокий аналіз підготовлених даних.

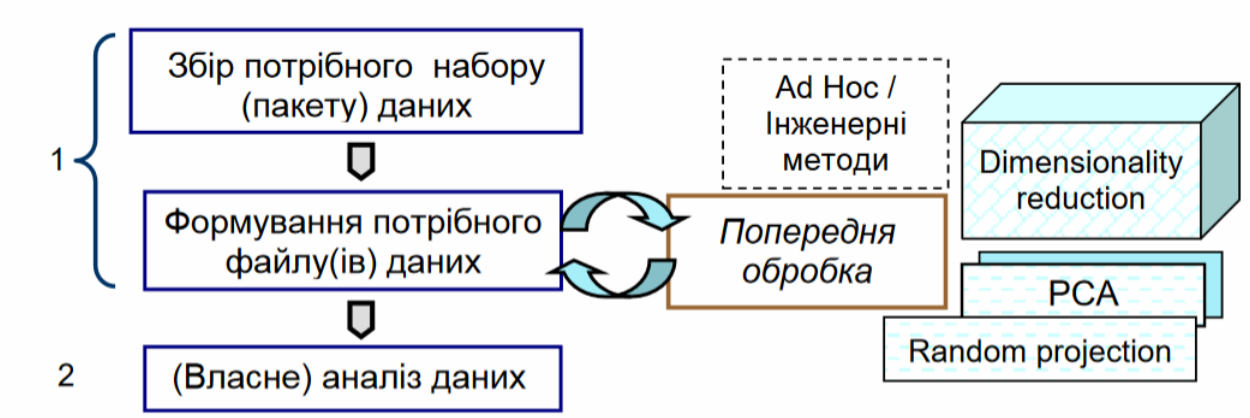



Рис. 2.4. Схема процесу аналітики багатовимірних даних

Ланцюг проходження завдання аналітики даних показано на рис. 2.4. Етап глибокого аналізу даних у свою чергу може складатися з ланцюга завдань. Попередня обробка може залучати методи, які традиційно розглядалися як методи власне аналізу даних (аналіз головних компонент, random projection і т. д.).

Одна з тенденцій аналітики багатовимірних даних – перенесення аналітичних засобів в програмне забезпечення баз даних, аби виконувати значну частину роботи в місцях зберігання, без передачі даних на сервер аналітика.

Зокрема, фірма SAS у співпраці з Oracle та Teradata інтегрує свою аналітики в програмне забезпечення баз даних. Виконання аналітики прямо на платформі баз даних дає можливість розосередити, розподілити виконання задачі й використати паралелізм. (Такий режим може бути вимушеним у

зв'язку з захистом даних.) Але такий режим далеко не завжди прийнятний з огляду на розмаїття методів аналізу, інструментарію й мов програмування. До того ж, масовану ітеративну переробку (з багаторазовим скануванням активної порції даних) зазвичай ефективніше виконувати, маючи ректифікований файл в локальній пам'яті комп'ютера аналітика. В деяких платформах та інструментах застосовується режим in-Memory Analytics, коли «гарячі» дані утримуються в пам'яті RAM (не переміщуються на диск).

Одна з передових сучасних програмних платформ аналізу даних (яка підтримує увесь цикл аналізу) – Apache Hadoop and MapReduce. Багатий комплект методів і програм аналізу даних для вказаної платформи надається відкритим середовищем Apache Mahout та Apache Spark.

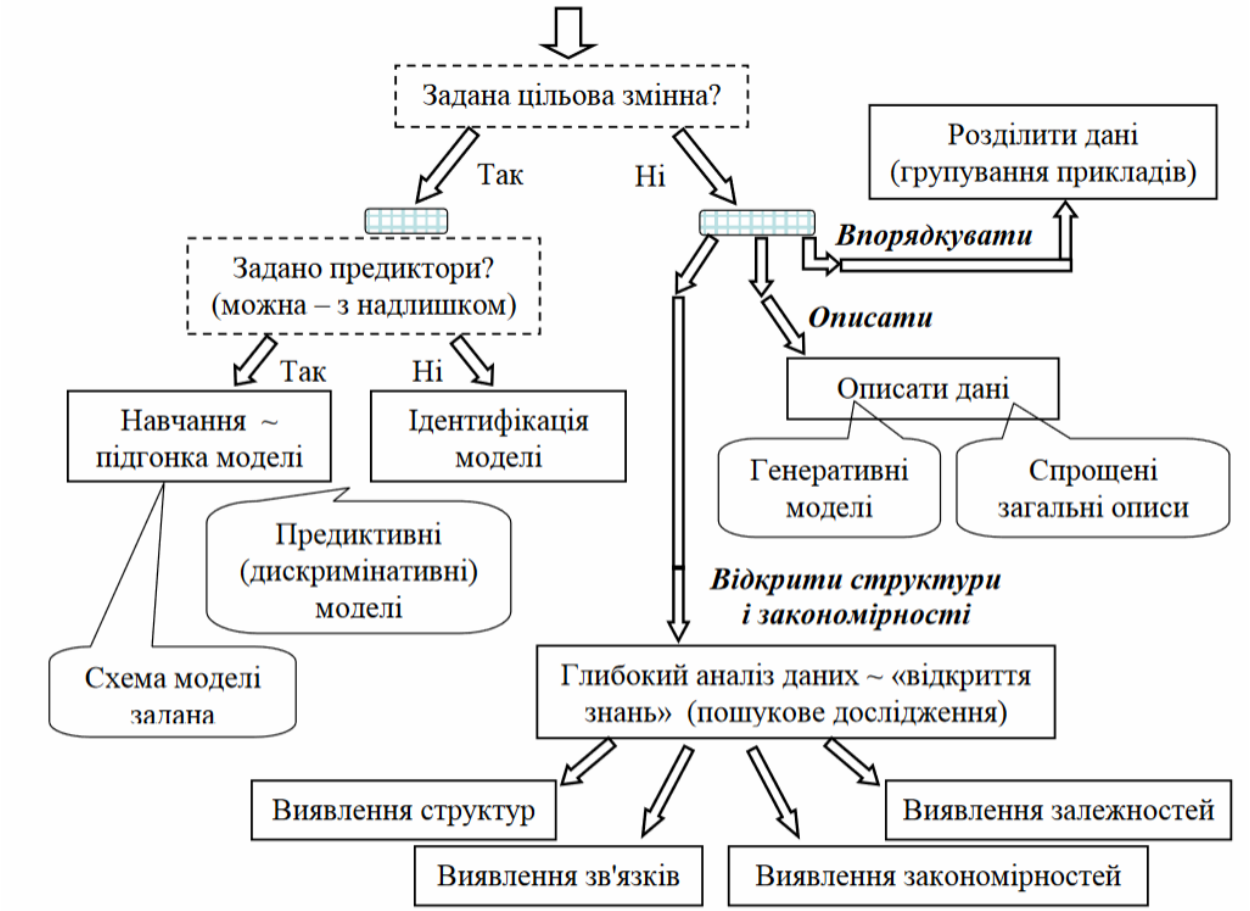



Рис. 2.5. Типи задач та результатів аналітики багатовимірних даних

Взагалі, можна виділити наступні роди завдань з повномасштабним використанням багатовимірних даних:

1) розширені режими традиційного пошуку інформації;

2) «інтелектуальний» пошук потрібної інформації (скомпонованих фактів, записів, фрагментів файлів);

3) масована проміжна переробка даних (чи краще сказати – «відпрацювання») однотипною процедурою за один-два проходи через масу даних (mining, concentration);

4) індукція моделі об'єкту (джерела), звідки взято дані;

5) екстракція знань з даних (відкриття структур і закономірностей).

На рис. 2.5 запропоновано один з варіантів систематизації аналітики багатовимірних даних за родами задач та типами результатів.

## 2.3. Дослідження та визначення переваг фреймворків для обробки багатовимірних даних

У сфері бізнесу є три основні цілі застосування OLAP-технологій: аналіз даних, планування бюджету, фінансова консолідація. Багатовимірна модель даних, можливість проводити аналіз великого обсягу даних та швидке оброблення запитів роблять OLAP-технології безальтернативним механізмом для аналізу продажів, маркетингових кампаній та інших завдань з великою кількістю вихідних даних. OLAP-куб містить дані та інформацію про виміри (агрегати). Зараз представлено величезну кількість різноманітних OLAP-систем. Розроблено декілька класифікацій продуктів цього типу: наприклад, класифікація за способом зберігання даних, за місцезнаходженням OLAP-машини, за ступенем готовності до застосування

В даний час існує близько декілька десятків різновидів фреймворків з обробки багатовимірних даних. Одним з найбільш широко використовуваних варіантів є класичний, а також один з найкращих в наш час, фреймворк – Hadoop.

Hadoop чудово підходить для надійних, масштабованих, розподілених обчислень. Однак, він також може використовуватися для зберігання файлів

загального призначення. Він може зберігати та обробляти петабайти даних. Це рішення складається з трьох основних компонентів:

- файлова система HDFS, відповідальна за зберігання даних у кластері Hadoop;

- система MapReduce, призначена для обробки великих обсягів даних у кластері;

- YARN, ядро, яке обробляє управління ресурсами.

Hadoop може зберігати та обробляти багато петабайтів інформації, тоді як для найшвидших процесів у Hadoop потрібно лише кілька секунд. Він також забороняє будь-які редагування даних, які вже зберігаються в системі HDFS під час обробки.

Наступним розглянемо фреймворк Apache Spark. Це фреймворк з відкритим кодом, створений як більш досконале рішення, порівняно з Apache Hadoop.

Початковий фреймворк був чітко побудований для роботи з Big Data. Основна відмінність цих двох рішень - це модель пошуку даних. Hadoop зберігає дані на жорсткому диску разом з кожним кроком алгоритму MapReduce. У той час як Spark здійснює всі операції, використовуючи пам'ять з випадковим доступом. Завдяки цьому Spark демонструє швидку продуктивність і дозволяє обробляти масивні потоки даних. Функціональні стовпи та основні особливості Spark - це висока продуктивність та безвідмовна безпека.

Він має п'ять компонентів: ядро та чотири бібліотеки, що оптимізують взаємодію з Big Data. Spark SQL - одна з чотирьох виділених базових бібліотек, яка використовується для структурованої обробки даних.

У випадку Apache Spark вам потрібно оптимізувати код вручну, оскільки в ньому немає жодного процесу автоматичної оптимізації коду. Це перетворюється на недолік, коли всі інші технології та платформи рухаються до автоматизації. Apache Spark не має власної системи управління файлами.

Він залежить від деяких інших платформ, таких як Hadoop або інших хмарних платформ.

І останнім розглянемо такий фреймворк, як Apache Flink. Це система потокової передачі даних, спрямована на забезпечення можливостей для розподілених обчислень по потоках даних. Трактуючи пакетні процеси як особливий випадок потокової передачі даних, Flink є ефективним як пакетною, так обробкою в режимі реального часу.

Flink добре підходить для розробки програм на основі подій. Ви можете ввести в нього контрольні точки, щоб зберегти прогрес у разі відмови під час обробки. Flink також має зв'язок із популярним засобом візуалізації даних Zeppelin.

MapReduce - модель програмування та пов'язана з нею реалізація для обробки та генерації великих наборів даних з паралельним розподіленим алгоритмом на кластері. Алгоритм MapReduce корисний для обробки величезної кількості даних паралельно, надійним та ефективним способом у кластерних середовищах. Він розділяє вхідні завдання на більш дрібні та керовані підзадачі для їх виконання паралельно.

Алгоритм MapReduce працює, розбиваючи процес на 3 фази (рис. 2.5):

- Map фаза;
- Sort & Shuffle фаза;
- Reduce фаза.

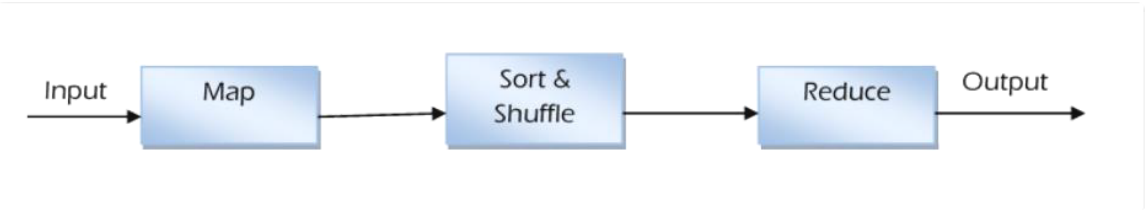



Рис. 2.5. Фази MapReduce

У MapReduce для кожної фази є пара ключ-значення як для входу, так і виходу. MapReduce завжди очікуватиме введення у вигляді пар ключ & значення від шарів HDFS (якщо розглядати Hadoop, рис. 2.6).

Як тільки обробка MapReduce завершиться, вона знову видасть результат поверх HDFS у вигляді пари (Key, Value).

| Phase | Input | Output |
|---|---|---|
| Mapper | (K,V) | (K,V) |
| Shuffle & Sort | (K,V) | (K, list(V)) |
| Reducer | (K,list(V)) | (K,V) |

Рис. 2.6. Введення та виведення фаз MapReduce

Функція map - перший крок в алгоритмі MapReduce. Вона приймає вхідні значення та розділяє їх на менші підзадачі, а потім паралельно виконує необхідні обчислення для кожної підзадачі.

Фаза map виконує наступні два кроки:

- розщеплення - приймає вхідний набір даних і ділить на менші набори;

- картографування - приймає менші набори підданих як вхідні дані та виконує необхідні дії або обчислення для кожної підмножини даних.

Вихід функції Map - це набір пар ключів і значень як <Ключ, Значення>. Приклад роботи map функції можна побачити на рис. 2.7.

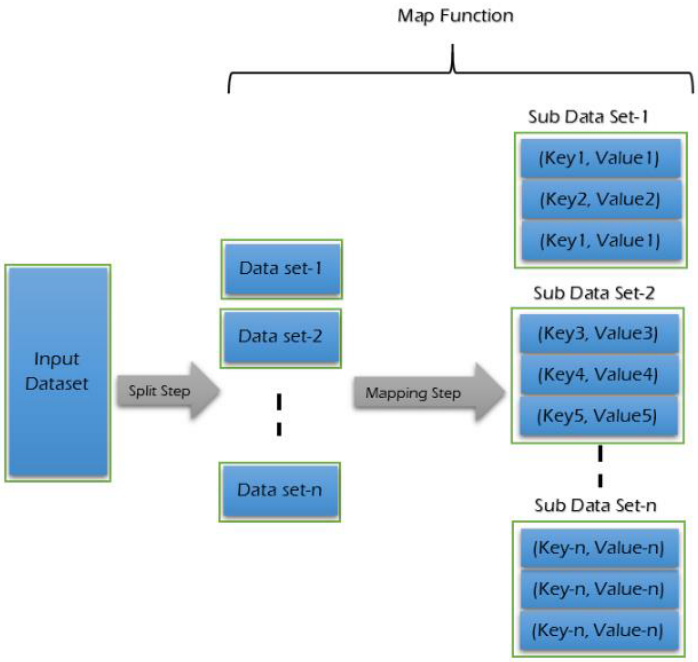



Рис. 2.7. Приклад роботи map функції

Sort & Shuffle - це другий крок в алгоритмі MapReduce. Вихід Mapper буде прийнятий як вхід для сортування та переміщення. Переміщення - це групування даних з різних вузлів на основі ключа. Це логічна фаза.

Він виконує наступні два кроки:

- об'єднання - поєднує всі пари ключових значень, які мають однакові ключі та повертає <Ключ, Список <Значення>>;

- сортування - відбирає результат з об'єднання та сортує всі пари ключових значень за допомогою ключів. Цей крок також повертає <Ключ, Список <Значення>> вихід, але з відсортованими парами ключ-значення.

Нарешті, функція shuffle повертає список <Ключ, Список <Значення>> відсортованих пар до фази reducer. Приклад фази Sort & Shuffle можна побачити на рис. 2.8.

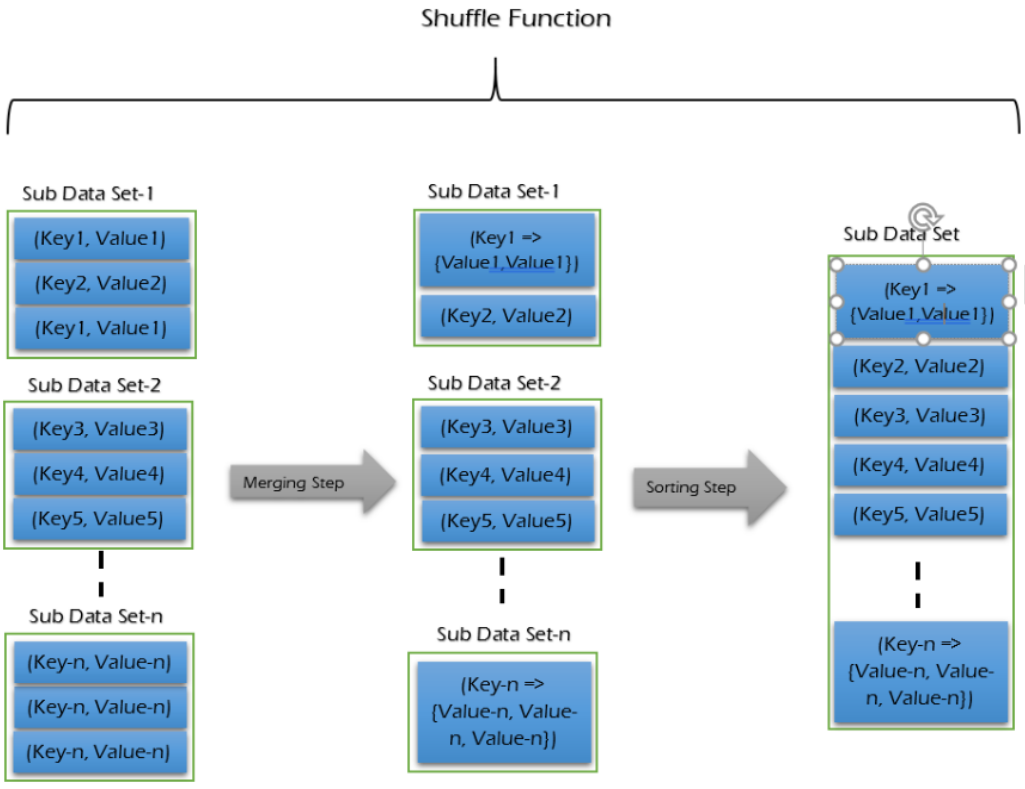



Рис. 2.8. Приклад роботи shuffle функції

Фаза reduce є завершальним кроком в алгоритмі MapReduce.

Вона бере список <Ключ, список <Значення>> відсортованих пар з функції Shuffle та виконує операцію reduce. Після завершення фази reduce кластер збирає дані для формування відповідного результату і відправляє їх назад на сервер Hadoop. Приклад роботи фази reduce можна побачити на рис. 2.9.

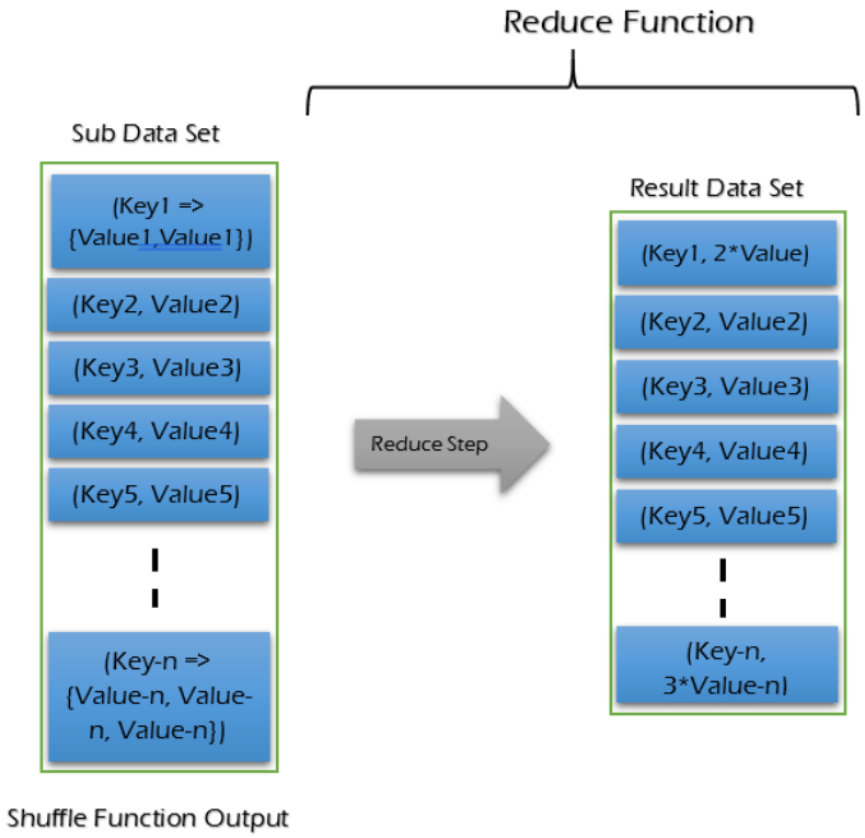



Рис. 2.9. Приклад роботи reduce функції

У Hadoop, з паралельним та розподіленим алгоритмом, MapReduce обробляє великі набори даних. Існують завдання, які нам потрібно виконати: Map і Reduce, MapReduce вимагає багато часу для виконання цих завдань, тим самим збільшуючи затримку. Дані розподіляються та обробляються через кластер в MapReduce, що збільшує час і зменшує швидкість обробки.

Як рішення цього обмеження, Hadoop Spark подолала цю проблему шляхом обробки даних в пам'яті. Обробка пам'яті відбувається швидше, оскільки не витрачається час на переміщення даних/процесів на диск і з нього. Spark в 100 разів швидша за MapReduce, оскільки вона обробляє все в

пам'яті. Однак обсяг оброблюваних даних також відрізняється: Hadoop MapReduce здатний працювати з набагато більшими наборами даних, ніж Spark. Hadoop MapReduce дозволяє паралельно обробляти величезну кількість даних. Він розбиває великий фрагмент на більш дрібні, які обробляються окремо на різних вузлах даних, і автоматично збирає результати по декількох вузлах, щоб повернути один результат. Якщо отриманий набір даних перевищує доступну оперативну пам’ять, Hadoop MapReduce може перевершити Spark.

Hadoop MapReduce є хорошим рішенням, якщо швидкість обробки не є критичною. Наприклад, якщо обробку даних можна проводити протягом нічних годин, є сенс розглянути можливість використання Hadoop MapReduce.

Переваги Apache Spark:

- швидка обробка даних. Обробка в пам'яті робить Spark швидше, ніж Hadoop MapReduce - до 100 разів для даних в оперативній пам'яті і до 10 разів для даних у дисках;

- ітеративна обробка. Якщо завдання полягає в обробці даних знову і знову - Spark перемагає Hadoop MapReduce.;

- обробка в режимі реального часу. Якщо бізнесу потрібна негайна інформація, тоді він повинен вибрати Spark та його обробку в пам'яті;

- обробка графіків. Обчислювальна модель Spark хороша для ітеративних обчислень, типових для обробки графіків.;

- приєднання до наборів даних.

## 2.4. Розробка архітектурної схеми системи обробки багатовимірних даних

Структурна схема призначена для відображення компонентів та взаємодії різних частин системи. Зазвичай така схема демонструє наявність підсистем у складі системи, а також інших компонентів, що забезпечують

управління та взаємодію між чисельними багаторівневими програмними модулями. Одним з найважливіших етапів проектування програмного продукту є розробка блок-схеми. Вона включає в себе сукупність компонентів, з яких складається система, взаємозв'язків між цими компонентами, додаткових об'єктів. Дана блок-схема дозволяє мати чітке уявлення про структуру системи, що відкриває можливості для модифікації такої у майбутньому та масштабуванні програмного продукту. Дана система спроектована з використанням принципів концепції багатошарової архітектури. Загалом, успішно застосовуються на практиці багато різних типів архітектури. Найчастіше використовується традиційна трирівнева система, яка розділяє додаток на три рівні.

Тут слід зазначити, що багаторівнева архітектура зазвичай представляє два недосконало пов'язаних поняття, n-шар та n-ярус. Шари та яруси часто називають терміном «шар», а термін «шар» іноді використовується разом із «ярусом Шар являє собою фізичний шар. Іншими словами, якщо говорити про трирівневу архітектуру, n-ярусна програма може бути розбита на сервер баз даних, веб-додаток на веб-сервері та рівень браузера користувача.

Шари представляють собою логічні рівні, тобто є рівень доступу до даних, рівень логіки бізнес процесів, рівень представлення, рівень обслуговування тощо. У цьому випадку логічний рівень не такий, як фізичний. Тому, як правило, рівень презентації включає контролер для обробки вводу даних і подання, що відображається у веб-браузері. Тобто вона поділяється на два фізичні рівні. Загальна структурна схема трирівневої архітектури зображена на рисунку 2.10.

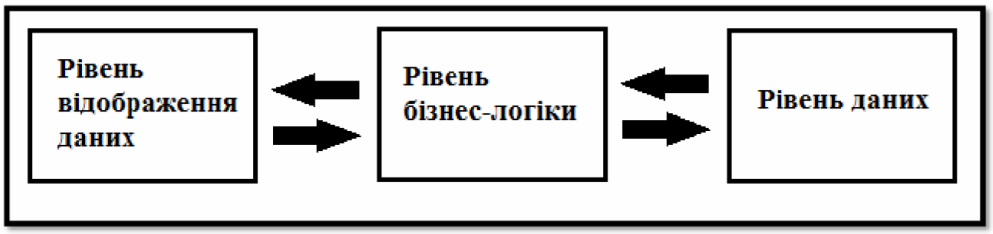



Рис. 2.10. Схема трирівневої архітектури

Презентаційний шар – це рівень прямої взаємодії з користувачем. Цей рівень включає компоненти інтерфейсу користувача та механізм прийому вводу від користувача. Для mpc asp.net цей рівень відображає представлення даних, усі компоненти, що входять до інтерфейсу користувача (стиль, статичний HTML, сторінка javascript), а також контекстні об'єкти моделі, контролера та запиту.

Бізнес рівень (рівень бізнес-логіки) – це набір компонентів, відповідальних за обробку даних, отриманих від рівня представлення, реалізацію всієї необхідної логіки програми, здійснення всіх обчислень, взаємодію з базою даних та передачу результатів обробки на рівень презентації.

Шар доступу до даних – це такий, що містить моделі, що описують використовувані об'єкти та конкретні класи, використовувані різними технологіями доступу до даних, такими як контекстні класи даних Entity Framework. Він також зберігає сховище для взаємодії шару бізнес-логіки з базою даних. Варто пам’ятати, що крайні рівні не можуть спілкуватися один з одним. Це означає, що рівень презентації не має прямого доступу до бази даних або рівня доступу до даних, а лише до рівня бізнес-логіки.

Зазвичай в системах не буває класичної трирівневої архітектури, оскільки не завжди вистачає саме трьох рівнів. Частіше використовується більша їх кількість. Концепція залишається незмінною, проте з додатковими правками, що необхідні для системи. Таким чином і з’являються додаткові модулі обробки даних. Загалом прикладні шари або ж логічні рівні розділяють згідно типів функціонування та іноді за рівнем абстракції. Також варто враховувати, що будь-яка система використовує додаткове програмне забезпечення або проміжне програмне забезпечення і даний проект не є винятком.

Система складається з декількох основних модулів або підсистем. Першим розглянемо підсистему взаємодії з даними. Тут існує два варіанти використання даної системи, проте вони обидва спрямовані на отримання

даних із зовнішньої системи або з просто ззовні системи та подальшої передачі їх у систему (рис 2.11).


Рис. 2.11. Підсистема взаємодії з даними

На вхід підсистеми надходять дані у певному вигляді, а саме у вигляді сформованого повідомлення із своєю структурою в service bus queue. Дана підсистема відповідає за зчитування даних з черги з подальшою їх обробкою та передачі на підсистему обробки даних. Варто зазначити, що на даному етапі також відбувається первинна обробка вхідних даних, їх валідація та формування із даних повідомлення для системи обробки даних, а також повернення результату про обробку назад у систему зберігання повідомлень.

Наступний рівень це рівень обробки даних в середині системи (рис 2.12). Дана підсистема містить у собі три основних модулі. Це модуль обробки даних, модуль запису даних в систему та модуль збереження результату. Перший модуль представлений бібліотеками класів, або ж плагінами, які містять у собі реалізацію певної бізнес логіки.

Даний модуль відповідає за попередню обробку даних. Всі процеси, що виконуються на цьому етапі відбуваються перед створенням або оновленням даних у системі, в той час, коли всі операції, що виконуються у модулі запису даних в систему навпаки відбуваються після збереження нових або після оновлення існуючих даних у системі.

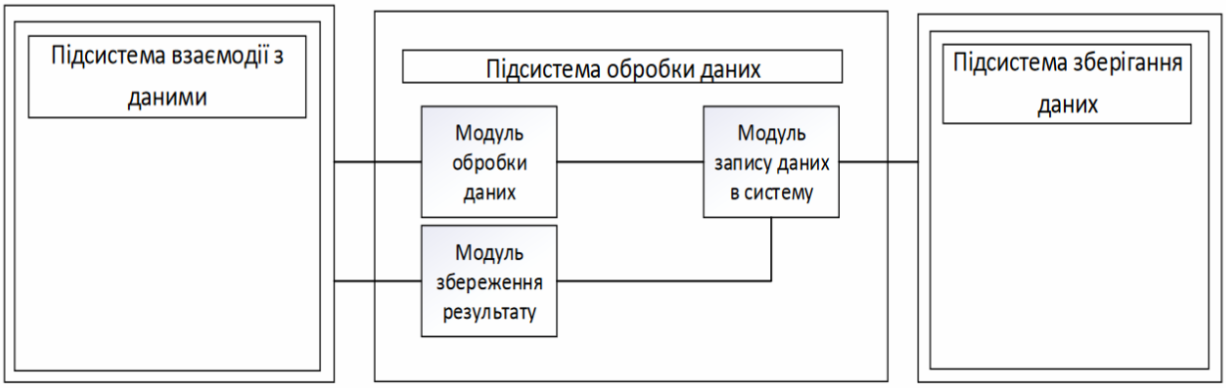



Рис. 2.12. Підсистема обробки багатовимірних даних

У даному модулі присутня логіка запису результатів обробки даних. Таким чином у системі створюється файл певного формату, який в собі містить корисну інформацію про всі зміни, що були створені в системі, а також передача результату обробки повідомлення з даними до підсистеми взаємодії з даними для подальшої обробки. Також даний модуль передає дані на подальшу обробку в підсистему зберігання даних безпосередньо у системі.

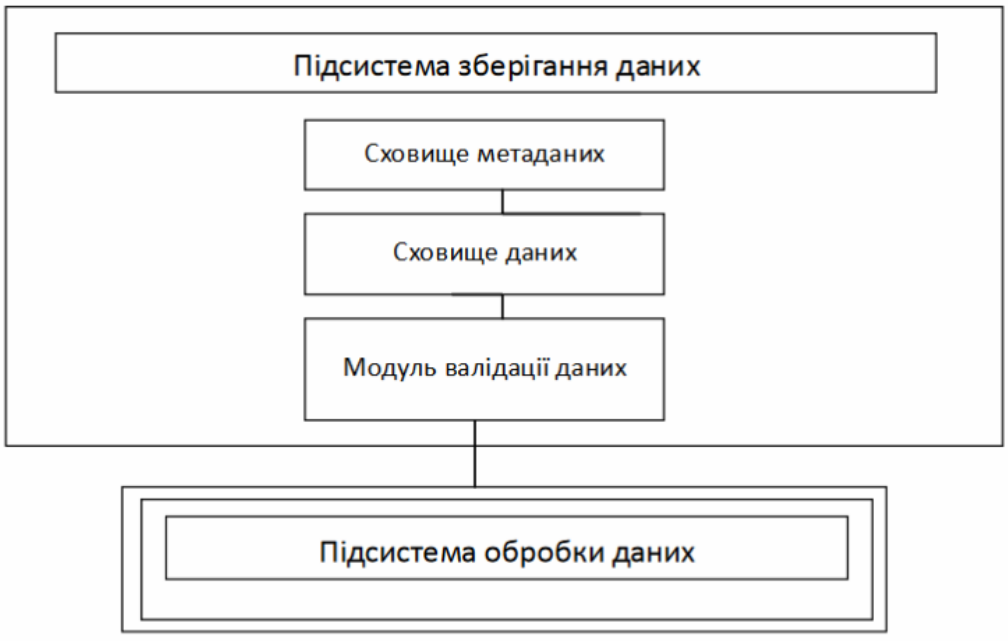



Рис. 2.13. Підсистема зберігання багатовимірних даних

На наступному рівні розміщені сховища зберігання даних та метаданих. Ці сховища є табличними, а інформація до них поступає після фінальної валідації, яка може бути пустою на даному етапі. Отже, як можна

побачити на рис. 2.13, ця підсистема складається з двох сховищ зберігання даних та модуля валідації.

Дві підсистеми, що залишились, логічно знаходяться на одному рівні. Ці підсистеми формування звітів (рис. 2.14) та відображення звітів (рис. 2.15).

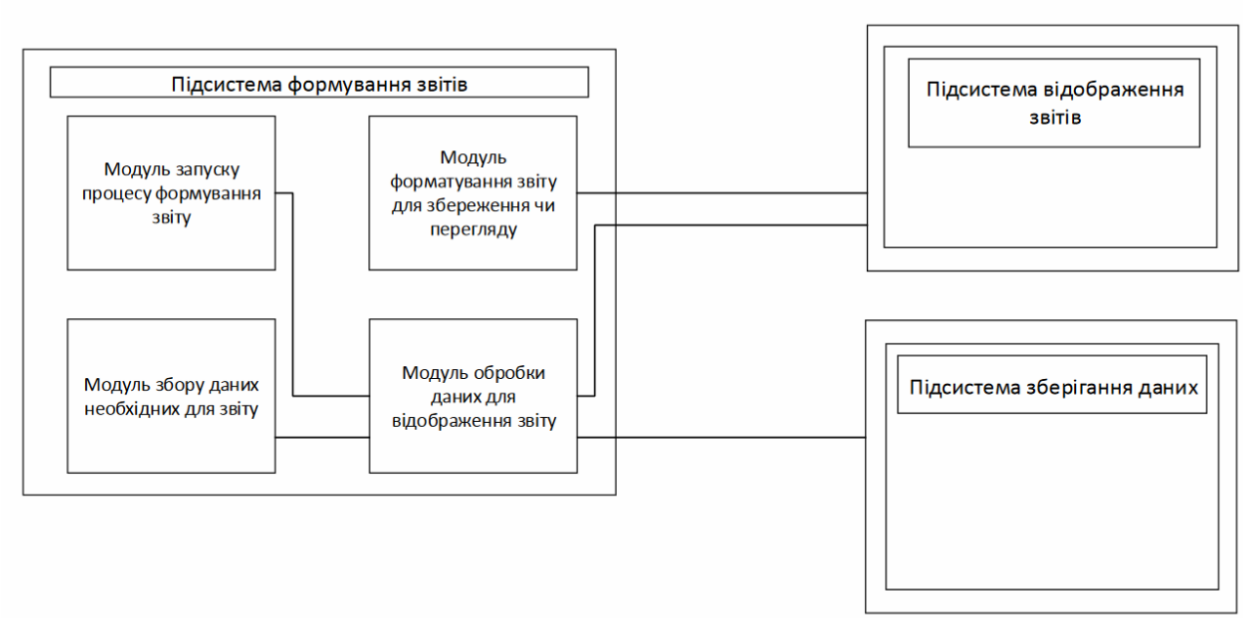



Рис. 2.14. Підсистема формування звітів

Більш детально розглянемо підсистему формування звітів. Вона складається з 4 основних модулів, серед яких: модуль запуску процесу формування звіту, модуль форматування звіту для збереження чи перегляду, модуль збору необхідних для звіту даних, модуль обробки даних для відображення звіту. На цьому рівні процес починається з модулю запуску формування звіту. Даний модуль може активуватись як періодично, за певним графіком, так і за вимогою користувача. У останнього варіанту також є два варіанти запуску.

Перший передбачає завантаження користувачем системи середовища для перегляду звіту, що і є тригером для запуску даного модуля. Наступним кроком є модуль обробки даних для відображення звіту. На даному етапі робиться запит до системи з відповідною вибіркою заздалегідь заданою

алгоритмом, а також можливо доповнений вибіркою фільтрів користувачем. Додатково даний модуль взаємодіє з модулем збору даних необхідних для звіту, до яких якраз належать налаштування користувача, обрані ним фільтри, мова тощо.

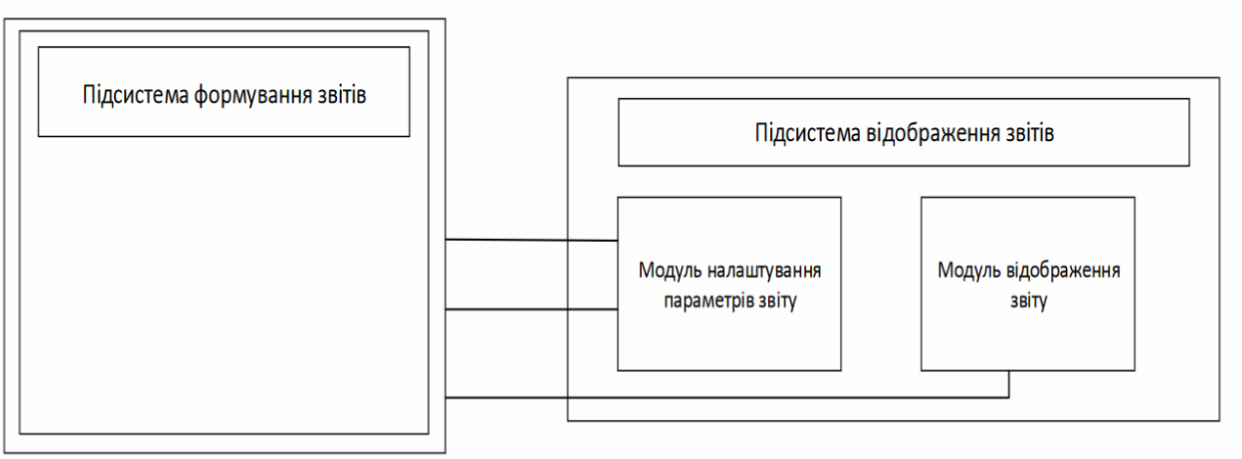



Рис. 2.15. Підсистема відображення звітів

Після обробки даних, сам звіт передається у підсистему відображення звітів, в якій, в свою чергу, наявний модуль налаштування параметрів звіту, в якому користувач має змогу обрати формат файлу для перегляду, а також обрати тип перегляду. Після опрацювання даним модулем даних про звіт, вони надходять до модуля формування звіту для перегляду чи завантаження. На даному етапі звіт повністю сформований, у заздалегідь обраному форматі та або автоматично завантажується, або відображається у інтерфейсі користувача.

Підсистема відображення звітів складається всього із двох модулів, перший модуль відповідає за можливість надання користувачеві додаткових фільтрів та можливостей для налаштування кінцевого вигляду звіту, а другий відповідає безпосередньо за відображення у інтерфейсі користувача.

Важливим модулем є модуль обробки даних. Блок-схема його роботи зображена на рисунку 2.16. На даному рисунку зображені основні компоненти алгоритму.

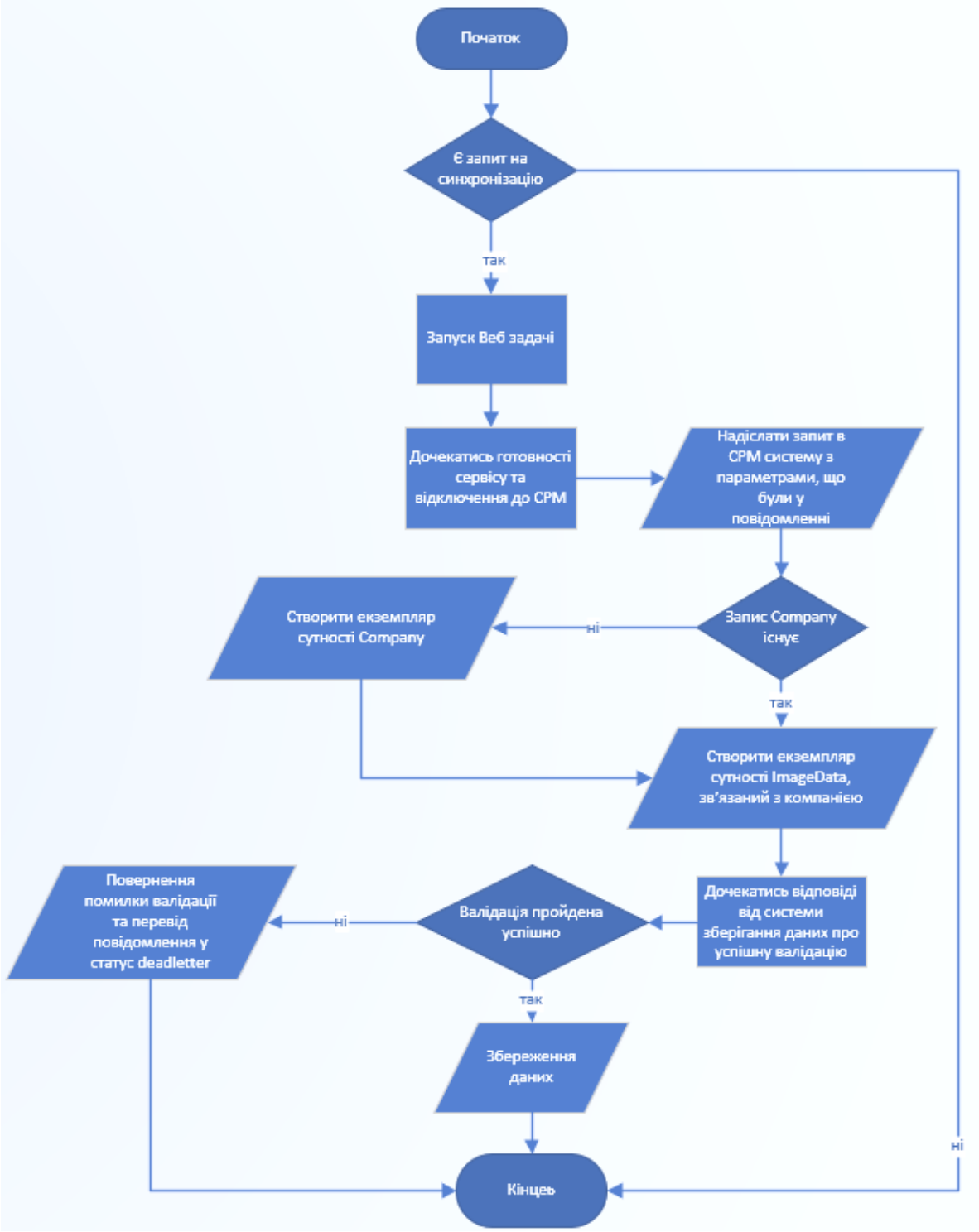



Рис. 2.16. Алгоритм модуля обробки даних

Після обробки вхідних даних, що записуються у вигляді повідомлення системою розпізнавання, іде обробка даних та виконуються запити до системи, які спрямовані на пошук сутності, до якої прив'язуються дані про обробці системою. Якщо екземпляра, вказаного у даних поки ще не існує в

системі, він створюється з можливістю заповнення необхідних для подальшої обробки атрибутів.

Якщо ж він наявний у системі, створюється додатково запис сутності, що зберігає в собі інформацію щодо оброблених зображень зовнішньою системою з необхідним набором даних. Аналогічний алгоритм відпрацьовую при обробці кожного нового набору даних та передачі відповідного повідомлення до черги з боку зовнішньої системи

## Висновки до розділу

В даному розділі виконано дослідження та опис методів інтелектуального аналізу даних, наведені процеси організації та обробки багатовимірних даних, описані фреймворки для обробки багатовимірних даних. Виконана розробка структурної схеми системи обробки багатовимірних даних на основі багаторівневої архітектури через достатньо велику кількість логічних модулів, що функціонально знаходяться на різних рівнях. Даний підхід дозволяє чітко розподілити обмін даними між рівнями та передбачає, що рівні можуть обмінюватись даними лише зі суміжними. Така архітектура системи забезпечує можливість легкого розгортання системи, а також відкриває можливості для масштабування системи за необхідності.

# РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОБРОБКИ БАГАТОВИМІРНИХ ДАНИХ ХМАРНИМИ ЗАСОБАМИ

## 3.1. Опис функціональних можливостей хмарної PaaS-платформи Heroku

Heroku – хмарна PaaS-платформа, що підтримує ряд мов програмування. З 2010 року є дочірньою компанією Salesforce.com. Heroku – одна з перших хмарних платформ, що з'явилась у 2007 році і, спочатку, підтримувала лише мову програмування Ruby, але, на даний момент, перелік мов, що підтримуються, також включають в себе мови Java, Node.js, Scala, Clojure, Python, Go, Ruby, PHP. На серверах системи використовуються операційні системи Debian та Ubuntu. Система розповсюджується як публічний хмарний сервіс за моделлю розгортання.

Heroku підтримує мови Ruby, Clojure, Node.js та системи керування базами даних, як Cloudant, Membase, MongoDB, Redis та PostgreSQL, як основну СКБД. Додатки, що працюють на Heroku, використовують також DNS-сервер Heroku. Для кожного окремого додатку виділяється декілька незалежних віртуальних процесів, що звуться «dynos». Ці процеси розподілені у віртуальній сітці, що складається з серверів. Heroku також підтримує систему керування версіями Git.

За своїми властивостями Heroku:

- має функції самообслуговування за вимогою – обирати підтримку для свого додатку, згідно з тарифом надання послуг, базу даних, що підходить найбільше для користувацького додатку, обирати розширення для додатку, а також налаштовувати інтеграцію додатку з Salesforce платформою.

- має універсальний доступ у Мережі – тобто, доступ до платформи може бути здійснений з будь-якого девайсу через веб-браузер або клієнт задля задання налаштувань проекту

- є еластичною – це значить, що користувач може у будь-який час отримати доступ до платформи, щоб змінити конфігурацію налаштувань на проекті, чи то звужуючи функціонал, чи то розширюючи.

Хмарний сервіси Heroku, пропонує кілька переваг для розробки паралельних і розподілених програм:

- **Масштабованість**. Хмарні служби забезпечують масштабовану платформу для розгортання паралельних і розподілених програм і керування ними. Ці служби дозволяють розробникам легко збільшувати або зменшувати обчислювальні ресурси, такі як сервери, сховища та мережі, у відповідь на зміну робочого навантаження.

- **Гнучкість**. Хмарні служби забезпечують гнучку платформу для розробки та розгортання паралельних і розподілених програм. Ці послуги пропонують широкий спектр обчислювальних ресурсів, інструментів і послуг, що дозволяє розробникам вибирати ті, які найкраще відповідають їхнім потребам.

- **Економічна ефективність**. Хмарні послуги можуть бути економічно ефективнішими, ніж традиційні локальні рішення для розробки паралельних і розподілених програм. Ці послуги зазвичай пропонують платіжну модель ціноутворення, що дозволяє розробникам платити лише за ті обчислювальні ресурси, які їм потрібні та які вони використовують.

- **Доступність**: хмарні служби забезпечують високу доступність для паралельних і розподілених програм із вбудованими функціями відмовостійкості та резервування. Ці служби, як правило, призначені для забезпечення того, щоб програми залишалися доступними та швидко реагували навіть у разі збою вузла або мережі.

- **Безпека**. Хмарні служби пропонують надійні функції безпеки для паралельних і розподілених програм. Ці служби зазвичай забезпечують шифрування, контроль доступу та функції моніторингу для забезпечення безпеки та цілісності даних.

- **Простий у використанні**: хмарні служби зазвичай прості у використанні, з інтуїтивно зрозумілими інтерфейсами та інструментами для розгортання та керування паралельними та розподіленими програмами.

Загалом, хмарні сервіси, такі як Heroku, забезпечують ідеальну платформу для розробки та розгортання паралельних і розподілених програм із багатьма перевагами перед традиційними локальними рішеннями.

Таким чином, з точки зору особи, що надає послуги, зважаючи на об’єднання ресурсів системи та непостійному характеру споживання з точки зору споживача, платформа дозволяє не лише умовно-безкоштовний хостинг, а і економити на споживчих апаратних потужностях. За рахунок процедур самообслуговування, дозволяє знижувати витрати на обслуговування.

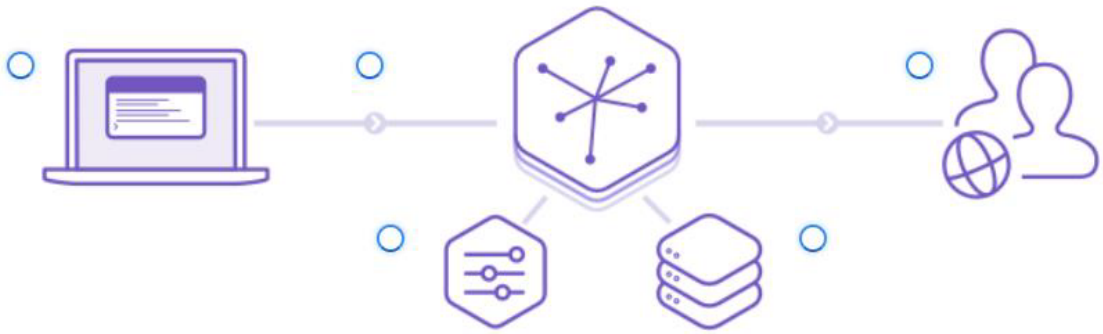


Рис. 3.1. Схема роботи сервісу Heroku

Процес роботи сервісу:

- користувач викладає код на платформу, за допомогою команд, що підтримуються сервісом;

- додаток запускається та працює на «dyno»-платформі;

- користувач робить запити, що обробляються сервером;

- користувач керує додатком через онлайн інтерфейс платформи;

- дані зберігаються на веб-базованих сервісах.

Розробник на Heroku отримує досвід у роботі з програмно-центрованим підходом для розміщення програм, який підтримує інтегрування з найбільш популярними інструментами та робочими процесами.

Отже, Heroku запускає додатки на «dynos» – онлайн смарт контейнерах з підходящим керованим середовищем. Розробник доставляє код, написаний на таких мовах програмування як Node.js, Ruby, Java, PHP, Python, Go, Scala, Clojure до вбудованої системи, що формує програму, готову для запуску. Мови та система, на яких працює додаток завжди оновлюються, тому програма завжди готова до роботи, незважаючи на патчі. Таке робоче середовище може підтримувати роботу додатку без втручання користувача.

Робочий процес Heroku створює логи з висхідних потоків програми, системних компонентів, сторонніх сервісів та відсилає їх до єдиного каналу. Платформа генерує три категорії логів для вашого додатку: програмні, системні та API-логи.

Завдяки підтримці Git, платформа Heroku може працювати як система контролю версій, оскільки кожна доставка коду зберігається окремо. Розробник може повернутися до попередньої версії програми за необхідності. Heroku також може працювати сумісно з контейнерами Docker та може завантажувати готові пакети коду через Docker.

Керування «dyno»-процесами, як частини робочого процесу Heroku, координує всі «dyno» частини додатку, що розробляється. Архітектура «dyno» розроблена таким чином, що утворює уявну сітку між кожним окремим процесом, тому при виході з ладу серверу – робоча частина програми на ньому одразу переміщається на інший робочий кластер абсолютно автоматично, без втручання розробника або служби підтримки. При керуванні шляхами, використовується HTTP протокол на основі запитів від додатку розробника до веб «dynos».

Heroku Developer Experience (DX) – додаток, що реалізовує підхід, централізований на додатку.

Це дозволяє розробнику неперервно займатись розробкою, не зважаючи на роботу серверів або інфраструктури. Доставка коду відбувається через популярні інструменти, такі як Git, або системи

безперервної інтеграції (CI-системи). Також, підтримується інтуїтивно зрозуміла дошка для розуміння результативності роботи програми.

Heroku DX пропонує користування інтуїтивно зрозумілої візуальної частини керування, що зветься дошкою. Тут відбувається вся організація програм, розміщених на платформі, а також надає доступ до метрик та даних, пов'язаних з роботою програм.

Працюючи з дошкою, розробник отримує такі дані, як швидкість відгуку, уведені дані, пам'ять, робота процесора, помилки тощо. Все це дає велику наглядність при аналізі роботи програми, що відображається завдяки інтуїтивному користувацькому інтерфейсу.


Рис. 3.2. Робота з OpEx, що демонструє завантеженість dyno-процесів та роботу запитів за хвилину

Отже, ця платформа підтримує вагомий перелік середовищ для популярних мов програмування на доволі вигідних умовах. Heroku реалізує PaaS архітектуру хмарного сервісу та має доволі легку взаємодію з користувачем та з іншими програмними середовищами.

## 3.2. Розробка архітектури інформаційної технології

Інформаційна технологія має реалізовувати прикладні випадки інтелектуального аналізу даних з імпортованого документа, що буде зчитаний CRM-системою та використовуватиме її, як контейнер для зберігання даних, виводитиме візуалізовані форми даних для більшої наглядності та повідомлятиме користувача про виконання аналізу даних.

Для безпосереднього інтелектуального аналізу даних використовуватиметься сценарій на мові Python, який буде приймати, як вхідний параметр масив даних з CRM-системи та, як вихідний параметр, відправлятиме оброблені дані знову в систему.

Для хостингу сценарію з реалізацією інтелектуального аналізу даних використовуватиметься платформа Heroku, що реалізуватиме взаємодію контейнерів зі сценарієм та контейнерів з платформи Salesforce. REST (Representational State Transfer) – підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів. Був описаний і популяризований 2000 року Роєм Філдінгом, одним із творців протоколу HTTP. В основі REST закладено принципи функціонування Всесвітньої павутини і, зокрема, можливості HTTP. Філдінг розробив REST паралельно з HTTP 1.1 базуючись на попередньому протоколі HTTP 1.0. Дані повинні передаватися у вигляді невеликої кількості стандартних форматів (наприклад, HTML, XML, JSON). Будь-який REST протокол (HTTP в тому числі) повинен підтримувати кешування, не повинен залежати від мережевого прошарку, не повинен зберігати інформації про стан між парами «запит-відповідь». Стверджується, що такий підхід забезпечує масштабовність системи і дозволяє їй еволюціонувати з новими вимогами.

Антиподом REST є підхід, заснований на виклику віддалених процедур (Remote Procedure Call, RPC). Підхід RPC дозволяє використовувати невелику кількість мережевих ресурсів з великою кількістю методів і складним протоколом. При підході REST кількість методів і складність

протоколу суворо обмежені, що призводить до того, що кількість окремих ресурсів має бути великою. REST – це архітектурний стиль для розподілених гіпертекстових систем.


Рис. 3.2. Архітектура інформаційної технології

Таким чином, може бути реалізовано повноцінний хмарний додаток, доступний для користувачів Salesforce, що реалізує інтелектуальний аналіз даних на прикладі будь-якого набору даних.

### 3.3. Розміщення програмного скрипту на хмарній платформі

Після встановлення потрібно скористатись командою «heroku login» через командну строку для безпосереднього входу у CLI:

```
$ heroku login
heroku: Press any key to open up the browser to login or q to exit
 ›   Warning: If browser does not open, visit
 ›   https://cli-auth.heroku.com/auth/browser/***
heroku: Waiting for login...
Logging in... done
Logged in as me@example.com
```

Рис. 3.3. Результат входу у CLI

Після приготування програми, слід викласти її у git-репозиторій. Оскільки Heroku підтримує git, то після клонування репозиторію, його можна викласти у середовищі на платформі Heroku.

```
$ git clone https://github.com/heroku/python-getting-started.git
$ cd python-getting-started
```

Рис. 3.4. Процес клонування репозиторію на платформу Heroku

Тепер середовище готове приймати код програми. Але спочатку треба створити новий додаток у dyno середовищі Heroku.

```
$ heroku create
Creating app... done, ⬢ serene-caverns-82714
https://serene-caverns-82714.herokuapp.com/ | https://git.heroku.com/serene-caverns-82714.git
```

Рис. 3.5. Створення додатку у Heroku

Після виконання цієї команди, середовище створить контейнер, що пов’язаний з git-репозиторієм та прийматиме усі зміни, що відображатимуться там. Тому середовище готове приймати код сценарію.

```
$ git push heroku main
Counting objects: 407, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (182/182), done.
Writing objects: 100% (407/407), 68.65 KiB | 68.65 MiB/s, done.
Total 407 (delta 199), reused 407 (delta 199)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Python app detected
remote: -----> Using Python version specified in runtime.txt
remote: -----> Installing python-3.10.1
remote: -----> Installing pip 20.2.4, setuptools 47.1.1 and wheel 0.36.2
```

Рис. 3.6. Вивантаження коду сценарію у репозиторій та у середовище

Після цього, код програми буде вивантажено у dyno середовище Heroku та готове до виконання. Для того, щоб скористатись сценарієм, треба звернутись до URL-адреси сценарію у середовищі Heroku. Ця адреса створюється одразу ж, як код програми буде доставлено.

Існує декілька способів інтегрувати Salesforce та Heroku, використовувати їх слід у різних випадках та при різних ситуаціях.

Техніка Heroku Connect використовується для зв'язування баз даних PotgreSQL, розміщеної на Heroku, та даних у Salesforce. Таким чином, виконується двосторонній зв'язок двох джерел даних.

Використовується при потребі оновлення даних у Salesforce через базу даних додатку на Heroku.

Salesforce Platform Events – викликає роботу Heroku при виникненні тих чи інших подій у Salesforce. Слід використовувати при проектуванні Salesforce додатків, що використовують події при роботі.

Виклики через Apex та тригери – ця техніка використовує мову розробки Apex у середовищі Salesforce. Дозволяє викликати REST API сервіси для оновлення даних у Salesforce. На відміну від Platform Events, дозволяє створювати кодові рішення у системі для реалізації подій, що відбуваються без відома користувача системи.

Виклики Salesforce REST API з Heroku-додатку – використовується для викликів додатку Salesforce з додатку Heroku, тобто реалізує зворотній зв'язок з платформою Salesforce. Це дозволяє отримувати інформацію про події або отримувати, безпосередньо, дані з CRM-платформи.

Реалізація зв'язку Salesforce та Heroku-додатку та результат виконання алгоритмів ІАД наведений нижче.

Лістинг 3.1. Сценарій виконання алгоритму Бірча:

```
# birch clustering
from numpy import unique
```

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

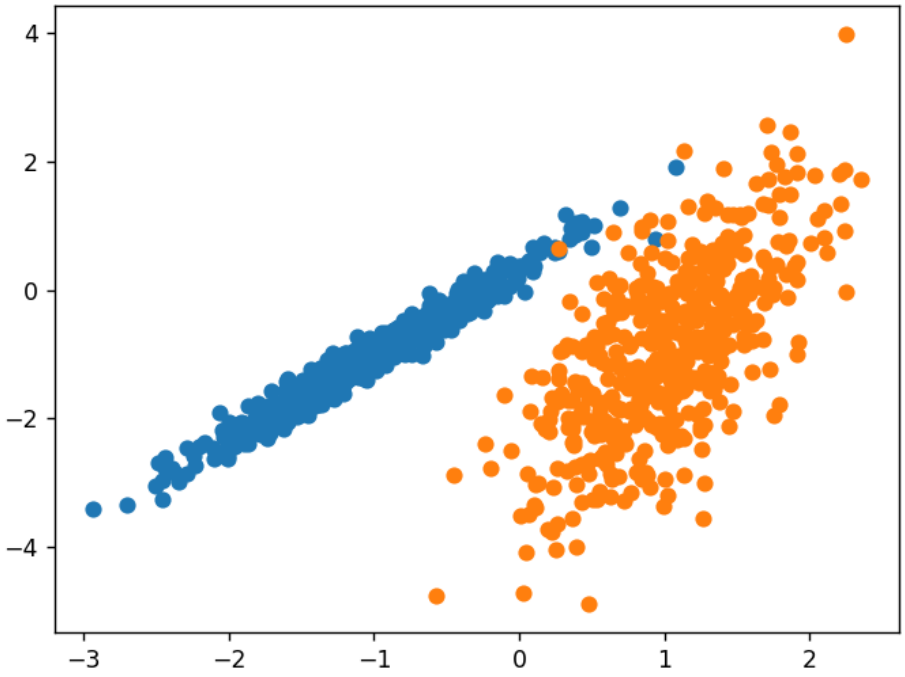



Рис. 3.7. Алгоритм кластеризації

Лістинг 3.2. Сценарій виконання алгоритму DBscan

```
# dbscan clustering
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import DBSCAN
from matplotlib import pyplot
# define dataset
     X, _ = make_classification(n_samples=1000, n_features=2,
n_informative=2, n_redundant=0, n_clusters_per_class=1,
random_state=4)
# define the model
model = DBSCAN(eps=0.30, min_samples=9)
# fit model and predict clusters
yhat = model.fit_predict(X)
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
# get row indexes for samples with this cluster
row_ix = where(yhat == cluster)
# create scatter of these samples
pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
     pyplot.show()
```

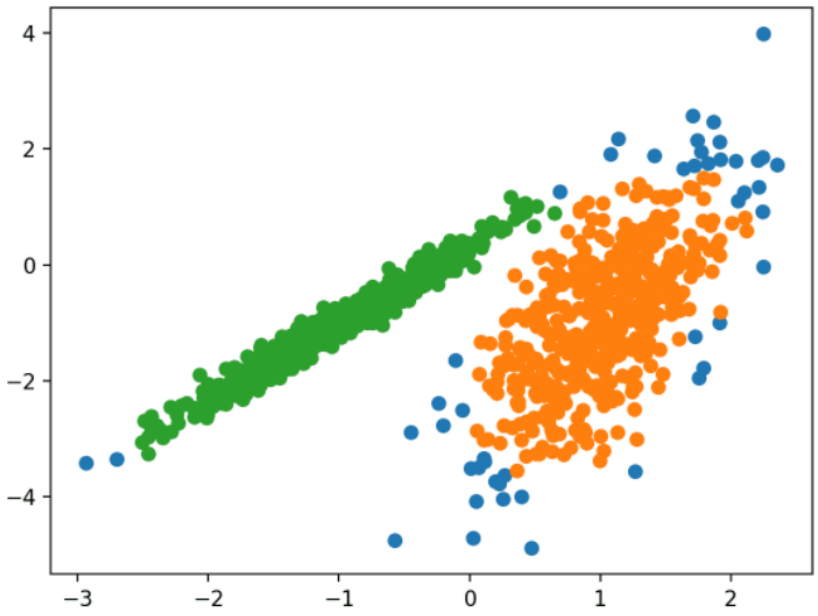



Рис. 3.8. Алгоритм DBscan

Лістинг 3.3. Сценарій виконання алгоритму k-means:

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

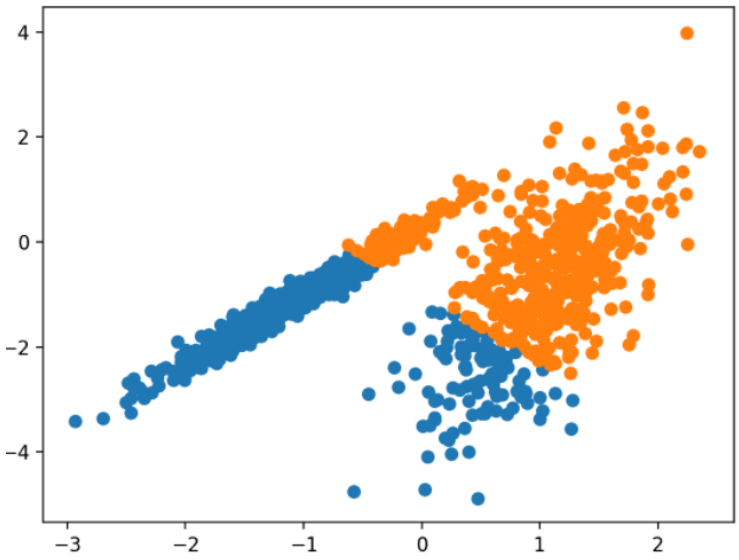
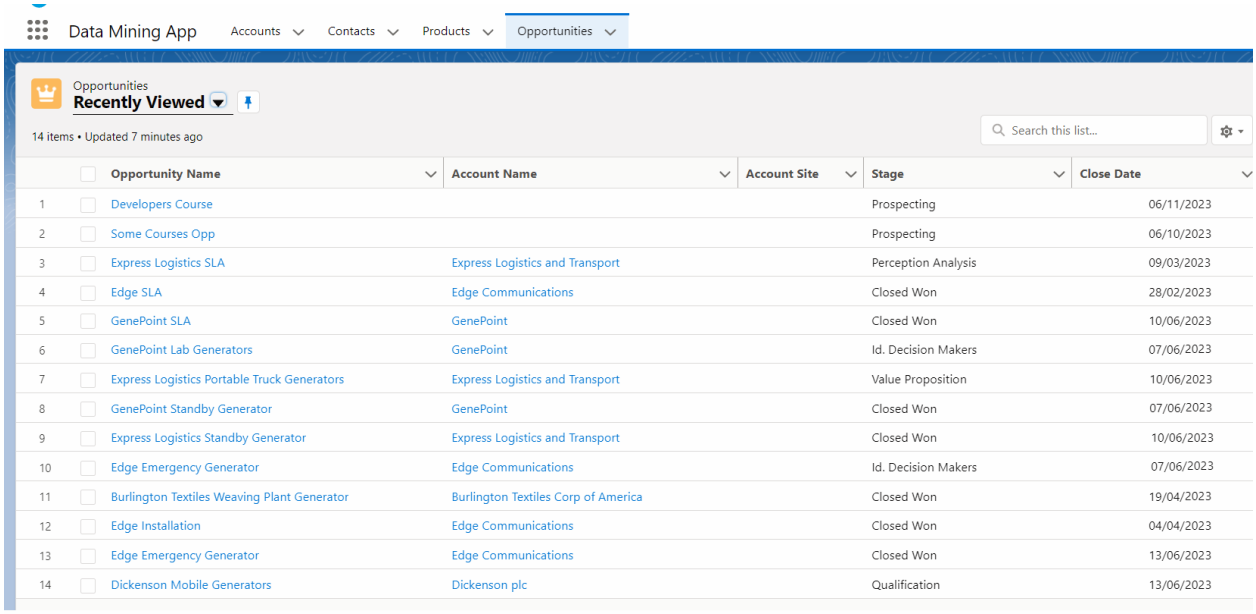



Рис. 3.9. Алгоритм k-means




| | Opportunity Name | Account Name | Account Site | Stage | Close Date |
|---|---|---|---|---|---|
| 1 | Developers Course | | | Prospecting | 06/11/2023 |
| 2 | Some Courses Opp | | | Prospecting | 06/10/2023 |
| 3 | Express Logistics SLA | Express Logistics and Transport | | Perception Analysis | 09/03/2023 |
| 4 | Edge SLA | Edge Communications | | Closed Won | 28/02/2023 |
| 5 | GenePoint SLA | GenePoint | | Closed Won | 10/06/2023 |
| 6 | GenePoint Lab Generators | GenePoint | | Id. Decision Makers | 07/06/2023 |
| 7 | Express Logistics Portable Truck Generators | Express Logistics and Transport | | Value Proposition | 10/06/2023 |
| 8 | GenePoint Standby Generator | GenePoint | | Closed Won | 07/06/2023 |
| 9 | Express Logistics Standby Generator | Express Logistics and Transport | | Closed Won | 10/06/2023 |
| 10 | Edge Emergency Generator | Edge Communications | | Id. Decision Makers | 07/06/2023 |
| 11 | Burlington Textiles Weaving Plant Generator | Burlington Textiles Corp of America | | Closed Won | 19/04/2023 |
| 12 | Edge Installation | Edge Communications | | Closed Won | 04/04/2023 |
| 13 | Edge Emergency Generator | Edge Communications | | Closed Won | 13/06/2023 |
| 14 | Dickenson Mobile Generators | Dickenson plc | | Qualification | 13/06/2023 |

Рис. 3.10. Представлення таблиці даних для обробки на Salesforce

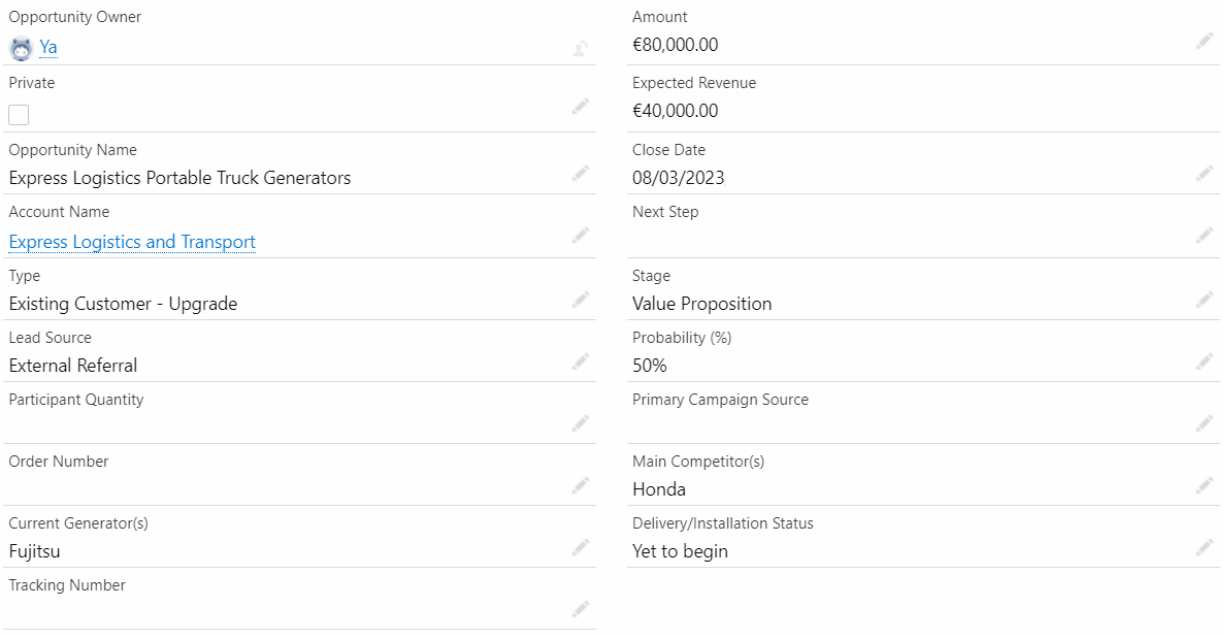



Рис. 3.11. Запис у представленні даних Salesforce

## Висновки до розділу

В даному розділі реалізована архітектура інформаційної технології яка здатна працювати незалежно від апаратної структури користувача, оскільки є хмарною. Реалізація CRM-технології у взаємодії з Data Mining задачами здатна суттєво позитивно вплинути на роботу компанії з даними з клієнтської

бази. Передача даних реалізується з використанням REST API архітектури, що забезпечує найбільш чіткий та найбільш злагоджений зв’язок між компонентами інформаційної технології.

# ВИСНОВКИ

В магістерській роботі розглянуто інтелектуальні моделі, методи та алгоритми обробки багатовимірних даних. Був проведений поглиблений аналіз існуючих рішень, предметної області, проаналізовано існуючі проблеми, а також визначили основні напрями руху проекту, в якому напрямку потрібно рухатись та якої мети досягти у процесі. Найбільш суттєвим недоліком сучасних рішень конкурентів є відсутність ефективної аналітики у системах зберігання багатовимірних даних. В процесі роботи було розроблено інформаційну технологію, що обслуговує CRM-систему з використанням методів Data Mining на віддаленій хмарній хостинг-платформі.

Переваги такого підходу до розробки полягає у тому, що розроблена система не залежить від апаратних можливостей користувача та розміщена повністю на хмарних сервісах, що потребує лише доступу до мережі. Переваги, що надає CRM-система, - це швидка обробка, візуалізація та сортування даних клієнтів, а також така робота з клієнтами, як інформування, залучення, надання послуг тощо. Важливою перевагою також є те, що такі системи здатні узгоджувати чітку та злагоджену роботу багатьох відділів на підприємстві.

Реалізація багатьох задач інтелектуального аналізу даних, математики та статистики доволі добре реалізована у високорівневій мові програмування Python, що підтримує багато бібліотек та пакетів. Також ця мова підтримується хостинг сервісом Heroku, що добре інтегрується з платформою Salesforce.

Завдяки розробленій системі користувачі системи Salesforce можуть робити пошук раніше невідомих даних, зв’язків, закономірностей для більш детальної обробки інформації. Реалізація CRM-додатку у взаємодії з Data Mining задачами здатна суттєво вплинути на роботу компанії з багатовимірними даними з клієнтської бази.

# ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ


1. Балабанов О.С. Аналітика великих даних: принципи, напрямки і задачі (огляд). //Проблеми програмуванння. – №2. – 2019. – С.47-68. – URL: https://doi.org/10.15407/pp2019.02.047

2. Інтелектуальний аналіз даних [Електронний ресурс]. – Режим доступу : http://mirznanii.com/a/308854/ntelektualniy-analz-danikh

3. Інтелектуальний аналіз даних: підручник /О.І.Черняк, П.В. Захарченко. – К.: Знання, 2014.

4. Інтелектуальний аналіз даних: практикум / Фісун М. Т., Кравець І. О., Казмірчук П. П., Ніколенко С. Г.: «Новий Світ - 2000», 2020. – 162с.

5. UpGrad 12 Most Useful Data Mining Applications of 2021 [Електронний ресурс] – Режим доступу : https://www.upgrad.com/blog/12-most-useful-data-mining-applications-of-2020/

6. Інтелектуальний аналіз даних [Електронний ресурс]. – Режим доступу : https://univerfiles.com/1168907/Інтелектуальний-аналіз-даних/

7. A Tutorial on Clustering Algorithms, available at: http://home.dei.polimi.it//Clustering/tutorial_html/kmeans.htm  access date December 05, 2014).

8. Medvedev Viktor. Cloud Technologies: A New Level for Big Data Mining / Viktor Medvedev, Olga Kurasova // Computer Communications and Networks. – Springer, 2016. – P. 55-67.

9. Talia D. The Weka4WS framework for distributed data mining in service-oriented Grids / D. Talia, P. Trunfio, O. Verta // Concurrency and Computation. - Vol. 20, No. 16. - Wiley, 2008. - P. 1933-1951.

10. Sholle D. What is Information? The Flow of Bits and the Control of Chaos / D.Sholle // Democracy and new media. – Cambridge, Mass.: MIT Press. – 2003. – P. 343–364. [Електронний ресурс]. – Режим доступу: http://web.mit.edu/commforum/papers/sholle.html.

11. Ackoff R. L. From Data to Wisdom / R. L. Ackoff // Journal of Applies Systems Analysis. – Vol. 16. – 1989. – P. 3–9.

12. DIKW [Електронний ресурс] // Вікіпедія: вільна енцикл. – Електрон. дані. – Режим доступу: https://uk.wikipedia.org/wiki/DIKW.

13. Brescia M. DAME: A Distributed Data Mining and Exploration Framework Within the Virtual Observatory / M. Brescia et al. // Remote Instrumentation for eScience and Related Aspects. - Springer, 2012. - P. 267-284.

14. UpGrad Most Common Examples of Data Mining [Електронний ресурс] – Режим доступу : https://www.upgrad.com/blog/most-common-examples-of-data-mining/

15. UpGrad Data Mining Techniques: Types of Data, Methods, Applications [Електронний ресурс] – Режим доступу : https://www.upgrad.com/blog/data-mining-techniques/

16. UpGrad Classification in Data Mining Explained: Types, Classifiers & Applications 2021 [Електронний ресурс] – Режим доступу : https://www.upgrad.com/blog/classification-in-data-mining/

17. Gula M. Matlab Adapter – Online Access to Matlab/Simulink Based on REST Web Services / M. Gula , K. Zakova // Intelligent Systems in Cybernet. and Automat. Theory. - Vol. 348. - Springer, 2015. - P. 199-206.

18. Kranjc J. ClowdFlows: A Cloud Based Scientific Workflow Platform / J. Kranjc, V. Podpecan, N. Lavrac // Lecture Notes in Computer Science. - Vol. 7524. - Springer, 2012. - P. 816-819.

19. Steh, Ju. V., Fajsal, M. E. Sardih, Lobur, M. V., Dombrova, M.S. and Arcibasov, V. E. - 2010), “Development and study of clustering algorithms for large collections of documents” Zbirnik naukovih prats IPPME im.G.E. Puhova NAN Ukrayini, Kiev, Ukraine, no. 58, pp. 283–290

20. Mansmann, S., Neumuth, T., & Scholl, M. H. (2007, September). OLAP technology for business process intelligence: Challenges and solutions. In International Conference on Data Warehousing and Knowledge Discovery - [Електронний ресурс] – Режим доступу: https://www.researchgate.net/

publication/220802446_OLAP_Technology_for_Business_Process_Intelligence_Challenges_and_Solutions

21. UpGrad Cluster Analysis in Data Mining: Applications, Methods & Requirements [Електронний ресурс] – Режим доступу : https://www.upgrad.com/blog/cluster-analysis-data-mining/

22. UpGrad Regression in Data Mining: Different Types of Regression Techniques [Електронний ресурс] – Режим доступу : https://www.upgrad.com/blog/regression-in-data-mining/

23. Podpecan V. Orange4WS Environment for Service-Oriented Data Mining / Vid Podpecan, Monika Zemenova, Nada Lavrac // The Computer Journal. – Vol. 55, No. 1. – Oxford Press, 2012. – P. 82-98.

24. Data migration [Електронний ресурс] : Режим доступу: https://docs.microsoft.com/dynamics365/admin/manage-configuration-data

25. System Степаненко О.П. Інтелектуальні системи підтримки управління діяльністю організації І О.П. Степаненко // Культура народів Причорноморья. - 2008. - № 140. - С. 119-122.

26. Верес О. М. Класифікація методів аналізу великих даних І О. М. Верес, Р. М. Оливко // Вісник Національного університету “Львівська політехніка” - 2017. - Випуск 872. - С.84-92.

27. Багаторівнева архітектура програмного додатку [Електронний ресурс] : Режим доступу: https://metanit.com/sharp/mvc5/23.5.php

28. Berthold M. KNIME: The Konstanz Information Miner / Michael R. Berthold et al. // Data Analysis, Machine Learning and Applications. - Springer, 2008. - P. 319-326.

29. Методи інтелектуального аналізу даних [Електронний ресурс]. – Режим доступу : http://buklib.net/books/24506/

30. Інтелектуальний аналіз геоданних [Електронний ресурс] – Режим доступу : https://cad.kpi.ua/attachments/093_2016d_Mahas.pdf

31. Zadeh, L. A. (2015). Fuzzy logic—a personal perspective. Fuzzy sets and systems, 281, 4-20.

32. Вискребенцева С.О., Кобилін О.А. (2019) Методи сегментації зображень. Матеріали XXIII міжнародного молодіжного форуму. Радіоелектроніка та молодь у XXI столітті, 19-20.

33. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In 2018 14th International Conference (TCSET) (pp. 665-670). IEEE.

34. Деркач, О. І. (2016). Аналітична обробка текстової інформації за допомогою засобів кластеризації. Young, 34(7).

35. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. Системи управління, навігації та зв'язку. Збірник наукових праць, 5(57).

36. Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

37. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323.

38. Terrasoft Що таке CRM-система та як вона працює? [Електронний ресурс] – Режим доступу : https://www.terrasoft.ua/page/definition-crm

39. Dou UA Що таке Salesforce система і чим вона цікава для досвідчених розробників? [Електронний ресурс] – Режим доступу : https://dou.ua/lenta/articles/what-salesforce-is/

40. Dou UA 5 кроків для початку кар'єри Salesforce-розробника [Електронний ресурс] – Режим доступу: https://dou.ua/forums/topic/35620/ ?from=tgj&utm_source=telegram&utm_medium=social

41. Trailhead Salesforce [Електронний ресурс] – Режим доступу: https://trailhead.salesforce.com/

42. Salesforce Forum [Електронний ресурс] – Режим доступу: https://salesforce.in.ua/

43. Heroku [Електронний ресурс] – Режим доступу: https://www.heroku.com/

44. Steinley, D. (2006). K means clustering: a half century synthesis. British Journal of Mathematical and Statistical Psychology, 59(1), 1-34.

45. Ackermann, M. R. (2009). Algorithms for the Bregman k-Median problem (Doctoral dissertation, University of Paderborn).

46. Khachumov, M. V. (2012). Distances, metrics and cluster analysis. Scientific and Technical Information Processing, 39(6), 310-316.

47. Huang, Z., & Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. IEEE Transactions on Fuzzy Systems, 7(4), 446-452.

48. Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). Introduction to time series analysis and forecasting. John Wiley & Sons.

49. Zhang, J., Zhao, Z., Xue, Y., Chen, Z., Ma, X., & Zhou, Q. (2017). Time series analysis. Handbook of Medical Statistics, 269.

50. Bodyanskiy, Y. V., Tyshchenko, O. K., & Mashtalir, S. V. (2019, June). Fuzzy Clustering High-Dimensional Data Using Information Weighting. In International Conference on Artificial Intelligence and Soft Computing (pp. 385-395). Springer, Cham.