COMPUTER SYSTEMS AND NETWORKS

Vol. 5, No. 1, 2023

DOI: https://doi.org/10.23939/csn2023.01.137

УДК 004.021; 004.383

СПЕЦІАЛІЗОВАНА ПРОГРАМНА ПЛАТФОРМА ДЛЯ АНАЛІЗУ ІНФОРМАЦІЇ В СХОВИЩАХ ДАНИХ

О. С. Харченко, Ю. С. Клушин

Національний університет "Львівська політехніка",
кафедра електронних обчислювальних машин
E-mails: oleksiikharchenkodev@gmail.com, Yurii.S.Klushyn@lpnu.ua

Наведено дизайн, висвітлено розроблення та оцінювання спеціалізованої програми для аналізу, розроблення агрегацій даних і візуалізації великих обсягів даних. Основна мета цієї програми – спростити оброблення даних, прискорити їх аналіз і полегшити написання коду для задач із великими обсягами даних. Для цього використано машинне навчання, а також два репозиторії.

Програма містить зручний і зрозумілий інтерфейс, сервери, що обробляють різні типи запитів від користувачів і передають їх у базу даних, а також саму базу даних з двома репозиторіями.

Методологія дослідження, використана в цьому дослідженні, передбачає ретельний аналіз наявних програм і методів вирішення проблем із великими обсягами даних. Цей аналіз вплинув на розроблення основних функцій програми. Потім було здійснено ретельне тестування та оцінювання цих функцій. Виконано дослідження користувачів для оцінювання ефективності програм з машинним навчанням порівняно з програмами, які працюють без нього, а також порівняння швидкості реалізацій розробки програм та обробки даних.

Результати дослідження показують, що такий підхід пришвидшив розроблення програм та оброблення даних, зробив їх якіснішими та точнішими. У дослідженні зроблено висновок, що платформа має значний потенціал для підвищення продуктивності великих компаній і що зі збільшенням кількості даних і технологій без використання цього розроблення програм із такою логікою буде абсолютно неефективним.

Ключові слова: сховище DWH; django; react; машинне навчання.

Вступ

У наш час розроблення додатків істотно залежить від оброблення інформації. Цього досягають за допомогою реалізації різних типів розроблення баз даних, таких як реляційні та нереляційні. Розроблення додатків із розширеною логікою обробки даних забезпечує швидшу взаємодію з клієнтом, меншу ймовірність помилок, легший аналіз цих даних, подальшу візуалізацію та їх міграцію в інші додатки чи системи. В результаті зростає кількість даних, їх доступність у майбутньому забезпечує точніший аналіз тієї чи іншої сфери бізнесу чи життєдіяльності. Така інформація про клієнта надає необмежений потенціал її власнику, саме так збирають дані про ринки чи інші сфери.

Обробка та аналіз великих даних давно стали невід'ємною частиною IT-сектору, який називається великими даними, і дав роботу таким фахівцям, як інженери з даних, інженери з якості даних, аналітики даних тощо. Великі дані не обробляють за допомогою бібліотек або фреймворків за замовчуванням, для цього використовують інші інструменти, тому це досі окремий тренд у програмуванні.

Ось чому дані стають дедалі важливішими в сучасному світі, а їх оброблення з кожним днем дедалі складнішим та значущішим.

Аналіз останніх досліджень та публікацій

1. Великі дані

Великі дані – це надзвичайно великі набори даних, які неможливо обробляти, керувати чи аналізувати за допомогою традиційних методів обробки даних. Великим дані зазвичай притаманні певні характеристики.

Обсяг: великі дані стосуються надзвичайно великих наборів даних, які можуть варіюватися від терабайтів до петабайтів або навіть ексабайтів.

Швидкість: великі дані генеруються з високою швидкістю, часто в реальному або майже реальному часі, що ускладнює керування та обробку.

Різноманітність: великі дані мають багато різних форм, зокрема структуровані, напівструктуровані та неструктуровані дані, як-от текст, зображення, відео та дані датчиків.

Правдивість: якість великих даних часто невизначена, вони містять неточності, невідповідності, відсутні дані.

Великі дані генеруються з різних джерел, ураховуючи соціальні мережі, пристрої Інтернету речей (IoT), наукові дослідження та бізнес-транзакції. Аналіз великих даних може допомогти компаніям і організаціям отримати цінну інформацію про поведінку клієнтів, ринкові тенденції та операційну ефективність.

Основна мета великих даних – отримати цінну інформацію та знання з великих, складних і різноманітних наборів даних, з якими традиційні методи оброблення даних не можуть працювати. Статті та знання, отримані з великих даних, можна використовувати для різних цілей, наприклад:

Бізнес-аналітика: великі дані можна використовувати, щоб одержати інформацію про поведінку, уподобання та тенденції клієнтів, що дає змогу компаніям приймати рішення на основі даних і оптимізувати свою діяльність.

Дослідження та розробки: великі дані можна використовувати в наукових дослідженнях для виявлення закономірностей, кореляцій та ідей, які можуть допомогти дослідникам розробляти нові продукти, послуги та методи лікування.

Виявлення шахрайства: великі дані можна використовувати у фінансах і банківській справі для виявлення шахрайських транзакцій і шаблонів, які вказують на шахрайську діяльність.

Прогностична аналітика: великі дані можна використовувати для створення прогнозних моделей, щоб прогнозувати майбутні тенденції, закономірності та результати на підставі історичних даних.

Персоналізація: великі дані можна використовувати для створення персоналізованого досвіду для клієнтів, наприклад, персоналізованих рекомендацій, пропозицій і вмісту.

Основна мета великих даних – отримувати цінну інформацію та знання, які можна використовувати для стимулювання зростання бізнесу, інновацій і конкурентоспроможності.

Для оброблення та аналізу великих даних потрібні спеціальні інструменти та технології, такі як розподілені обчислювальні системи, бази даних NoSQL та алгоритми машинного навчання. Ці інструменти дають можливістьу швидше обробляти й аналізувати великі набори даних, що дає змогу витягувати значущу інформацію та цінність із великих даних.

Великі дані пропонують різні методи обробки для роботи з великими обсягами даних. Нижче наведено деякі з поширених методів обробки:

1. Пакетна обробка. Передбачає обробку великих обсягів даних через регулярні проміжки часу, наприклад щодня або щотижня. Пакетна обробка корисна під час обробки великих наборів даних, які не потребують обробки в реальному часі.

2. Обробка в режимі реального часу: обробка в режимі реального часу передбачає обробку даних, які генеруються у режимі реального часу. Обробка в режимі реального часу корисна під час обробки даних, які потребують негайної дії чи відповіді, наприклад виявлення шахрайства або аналіз фондового ринку.

3. Потокова обробка – це метод обробки в реальному часі, який передбачає обробку даних невеликими безперервними потоками. Потокова обробка корисна під час обробки великих обсягів даних, які генеруються у режимі реального часу та потребують негайного аналізу та відповіді.

4. MapReduce: MapReduce – це структура розподіленого обчислення, яку застосовують для паралельної обробки великих наборів даних на кількох машинах. MapReduce зазвичай використовується для пакетної обробки та може обробляти великі набори даних, які не поміщаються в пам'ять однієї машини.

5. Обробка в пам'яті: обробка в пам'яті передбачає обробку даних у пам'яті, а не на диску. Це корисно для обробки великих наборів даних, які потребують пришвидшення обробки.

6. Обробка графіків: обробка графіків передбачає обробку даних у форматі графіків, таких як соціальні мережі, системи рекомендацій або виявлення шахрайства. Обробка графіків корисна для аналізу зв'язків і шаблонів у великих наборах даних. Великі дані пропонують різні методи обробки для роботи з великими обсягами даних. Кожен метод обробки має переваги та недоліки і використовується на основі конкретних потреб програми чи проєкту.

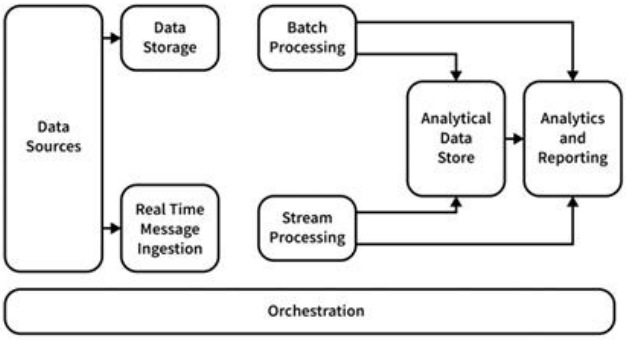

Рис. 1. Приклад проєкту з архітектурою великих даних

Рис. 2. Основні функції великих даних

2. Машинне навчання

Машинне навчання – це галузь інформатики та штучного інтелекту, яка зосереджується на розробленні алгоритмів і статистичних моделей, які дають змогу комп'ютерним системам навчатися на основі даних і підвищувати свою продуктивність у виконанні певного завдання без явного про-

грамування. Метою машинного навчання є розроблення моделей, які можуть аналізувати та вивчати шаблони з великих наборів даних, а також використовувати ці знання для прогнозування або виконання дій на основі нових даних.

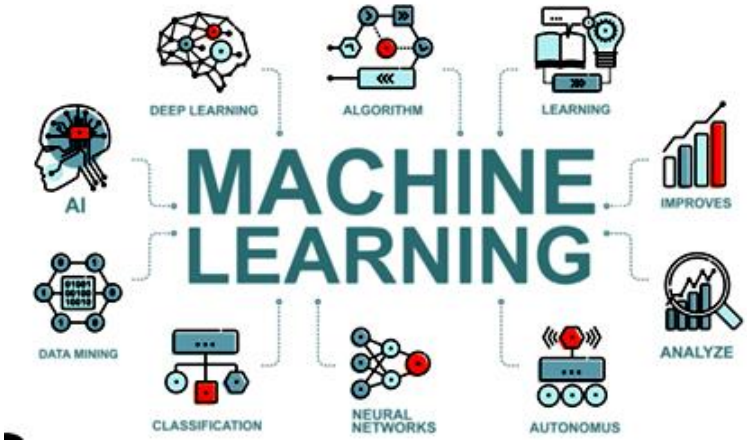

Рис. 3. Сфери використання машинного навчання

Машинне навчання відіграє вирішальну роль в аналізі та обробці великих даних. У зв'язку зі стрімким зростанням обсягу даних за останні роки традиційних методів обробки даних стало недостатньо для обробки великих наборів даних. Алгоритми машинного навчання мають можливість автоматично аналізувати та отримувати інформацію з великих і складних наборів даних, що робить їх важливим інструментом у аналітиці великих даних.

Однією з ключових переваг машинного навчання великих даних є його здатність навчатися на основі великих обсягів даних для підвищення точності прогнозів або класифікацій. Це особливо важливо у сценаріях великих даних, де обсяг, швидкість і різноманітність даних можуть зробити традиційні статистичні методи непрактичними.

Алгоритми машинного навчання можна використовувати для різноманітних завдань аналізу великих даних, таких як очищення даних, вибір функцій, розпізнавання образів і виявлення аномалій. Ці алгоритми можна навчити на великих наборах даних для розпізнавання шаблонів і зв'язків, які неочевидні для розпізнавання за допомогою традиційних методів аналізу даних.

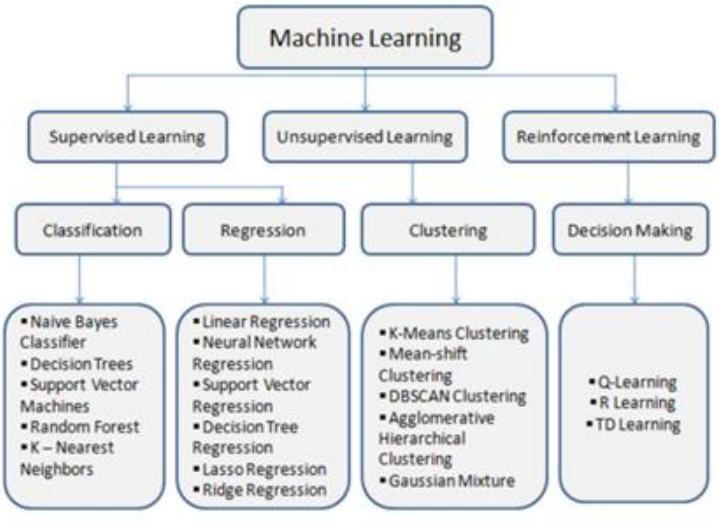

Рис. 4. Алгоритми машинного навчання

Іншим важливим аспектом машинного навчання великих даних є його здатність масштабуватись відповідно до розміру набору даних. Зі зростанням обсягу даних алгоритми машинного навчання можна навчати на більших і різноманітніших наборах даних, що забезпечує точніші прогнози і розуміння.

Отже, машинне навчання є потужним інструментом для аналізу та обробки великих даних. Використовуючи алгоритми машинного навчання, організації можуть отримувати інформацію з великих наборів даних і приймати обґрунтовані рішення на підставі закономірностей і зв'язків, виявлених у даних.

Виділяють чотири основні типи машинного навчання.

Контрольоване навчання: алгоритм навчається на позначеному наборі даних. Це означає, що вхідні дані супроводжуються відповідним виходом. Алгоритм навчається зіставляти вхідні дані з правильними результатами, використовуючи позначені приклади. Контрольоване навчання зазвичай використовують у таких програмах, як розпізнавання зображень, розпізнавання мови та обробка природної мови.

Неконтрольоване навчання: алгоритм навчається на непозначеному наборі даних. Це означає, що вхідні дані не супроводжуються жодним виходом. Завдання алгоритму полягає в пошуку закономірностей і зв'язків у даних без будь-яких попередніх знань про те, що шукати. Неконтрольоване навчання використовують переважно в таких програмах, як кластеризація, виявлення аномалій і зменшення розмірності.

Напівконтрольоване навчання: у напівконтрольованому навчанні алгоритм навчається на комбінації позначених і не позначених даних. Ідея цього підходу полягає в тому, щоб використати інформацію, наявну в немаркованих даних, для підвищення продуктивності моделі на міткових даних. Напівконтрольоване навчання використовують здебільшого в програмах, де отримання позначених даних є дорогим або потребує багато часу.

Навчання з підкріпленням: у навчанні з підкріпленням алгоритм навчається, взаємодіючи з навколишнім середовищем і отримуючи зворотний зв'язок у формі винагород або покарань. Мета алгоритму – вивчити політику, яка максимізує кумулятивну винагороду за послідовність дій. Це навчання використовують переважно в таких програмах, як ігри, робототехніка та автономне водіння.

3. SQL

SQL (Structured Query Language) – мова програмування, яка використовується для керування та обробки даних, що зберігаються у реляційних базах даних. SQL дає змогу користувачам створювати, змінювати та запитувати бази даних, а також виконувати широкий спектр завдань маніпулювання даними, таких як вставляння, оновлення та видалення даних.

SQL є декларативною мовою. Це означає, що користувачі визначають результат, якого вони хочуть досягти, а не вказують, як його досягти. Запити SQL складаються із серії команд і операторів, які використовують для взаємодії з базою даних і виконання завдань обробки даних.

SQL використовує багато організацій та окремих осіб, від малих підприємств, які керують даними клієнтів, до великих корпорацій, що аналізують мільйони транзакцій. Це стандартна мова, яку підтримує більшість систем керування реляційними базами даних, ураховуючи MySQL, Oracle, Microsoft SQL Server і PostgreSQL.

4. NOSQL

NoSQL (не-SQL) належить до типу бази даних, яка відрізняється від традиційних реляційних баз даних (RDBMS). Тоді як RDBMS покладаються на структуровану табличну модель даних із фіксованими стовпцями та рядками, бази даних NoSQL використовують різні моделі даних і структури, оптимізовані для обробки неструктурованих, напівструктурованих або змінних даних.

Бази даних NoSQL часто використовують для великих даних і вебдодатків у реальному часі, оскільки вони розроблені для обробки великих обсягів даних, до яких можна швидко отримати доступ і обробити. Вони також гнучкіші, ніж RDBMS, і можуть легко масштабуватися горизонтально, додаючи більше вузлів до розподіленої системи.

Існує кілька типів баз даних NoSQL, ураховуючи базу даних, орієнтовану на документ, базу даних типу "ключ – значення", базу даних із графіками та базу даних із сім'єю стовпців. Кожен тип має переваги і недоліки та підходить для різних випадків використання.

Мета статті

Однією з особливостей проєктів великих даних є використання спеціальних інструментів і створення сховища DWH.

DWH розшифровується як Data Warehouse, що є типом великомасштабної системи зберігання та керування даними, яку застосовують для аналітики. Сховище DWH розроблено для підтримки процесів прийняття бізнес-рішень, воно забезпечує централізоване сховище даних із багатьох джерел, які можна використовувати для звітності, аналізу та аналізу даних.

Системи зберігання DWH зазвичай використовують схему бази даних, оптимізовану для запитів великих обсягів даних, і підтримують ефективні процеси завантаження та перетворення даних. Вони часто передбачають використання процесів вилучення, перетворення, завантаження (ETL) для інтеграції даних із різних джерел і методів моделювання даних, таких як розмірне моделювання.

Зберігання DWH можна реалізувати за допомогою різноманітних технологій, ураховуючи традиційні реляційні бази даних, бази даних у стовпцях, хмарні рішення та платформи з відкритим кодом, такі як Apache Hadoop і Apache Spark. Вибір технології залежить від конкретних потреб організації та виду даних, що зберігаються.

Сховище DWH відіграє вирішальну роль у бізнес-аналітиці та аналітиці, дає змогу організаціям зберігати й аналізувати великі обсяги даних із багатьох джерел у централізованому місці.

Сховища даних (DWH) і машинне навчання можуть взаємодіяти кількома способами. Ось кілька прикладів:

Підготовка даних: DWH можна використовувати для зберігання та підготовки даних для програм машинного навчання. Дані можна очищати, трансформувати та організовувати у відповідні формати для обробки алгоритмами машинного навчання.

Розробка функцій: алгоритми машинного навчання потребують вхідних функцій, які мають відношення до проблеми, що вирішується. DWH можна використовувати для отримання та організації функцій із різних джерел даних для використання в моделях машинного навчання.

Навчання: моделі машинного навчання потребують навчальних даних, щоб вивчати закономірності та робити прогнози. DWH може забезпечити централізоване сховище навчальних даних, які можна використовувати для навчання моделей машинного навчання.

Прогнози: моделі машинного навчання можна розгортати, щоб прогнозувати нові дані. DWH застосовують для зберігання даних і надання механізму для моделі машинного навчання, щоб мати доступ до них і робити прогнози.

Цикл зворотного зв'язку: DWH можна використовувати для збирання відгуків про продуктивність моделей машинного навчання у виробництві. Цей відгук можна використовувати для вдосконалення та покращення моделей з часом.

Виявлення аномалій: DWH можна використовувати для зберігання даних, які використовуються для виявлення аномалій, таких як кібербезпека або виявлення шахрайства. Моделі машинного навчання можна навчити виявляти закономірності, які можуть вказувати на аномальну поведінку, і за необхідності сповіщати зацікавлені сторони.

Підводячи підсумок, DWH і машинне навчання можуть взаємодіяти багатьма способами, ураховуючи підготовку даних, розроблення функцій, навчання, передбачення, цикли зворотного зв'язку та виявлення аномалій. Поєднуючи сильні сторони цих технологій, організації можуть отримати цінну інформацію та приймати кращі рішення на підставі даних.

Структурна модель системи

Перед побудовою цієї системи необхідно враховувати цінність даних і важливість їх обробки в системі та подальшого аналізу, а також їх обсяги. Це неможливо, а саме немає сенсу будувати складні рішення з великими даними для невеликих проектів зі схожими даними але невеликою їх кількістю, це просто неоптимально. Ось чому система повинна мати доступ до великих обсягів різноманітних даних різного типу. Такі рішення оптимальні, для прикладу великих інтернет-магазинів або різних сервісів із великою кількістю користувачів і даних. Для вирішення цієї проблеми була створена система великих даних з обробкою великих обсягів даних, їх подальшим аналізом і візуалізацією за допомогою машинного навчання.

Комунікація з системою

Для загального розуміння системи опишемо її частини. Перша – клієнтська частина реалізована за допомогою React, back-end – за допомогою фреймворку Python Django, бази даних – нереляційні та реляційні бази даних, а саме PostgreSQL і MongoDB.

1. FRONT-END

Дисплей для клієнта розроблено у вигляді сайту з використанням таких технологій, як html, css, JavaScript, react.

HTML і CSS – це дві важливі технології веброзробки, які використовуються для створення та оформлення вебсторінок.

HTML (Hypertext Markup Language) – мова розмітки, використовувана для створення структури та вмісту вебсторінок. Забезпечує спосіб визначення елементів і вмісту вебсторінки, таких як заголовки, абзаци, зображення, посилання, форми та таблиці. Код HTML складається з тегів, які описують структуру вмісту та спосіб його відображення у веббраузері.

CSS (каскадні таблиці стилів) – це мова таблиць стилів, яка використовується для додавання стилю та візуального подання вебсторінок. Надає спосіб визначити, як мають відображатися елементи HTML, такі як шрифти, кольори, фон, рамки, інтервали та макет. Код CSS використовується для керування зовнішнім виглядом елементів HTML на вебсторінці, окремо від самого вмісту.

Програма має багато активних кнопок, користувач може вибрати, який тип даних він хоче ввести, тому програма використовує JavaScript.

JavaScript – динамічна мова програмування високого рівня, використовувана переважно для створення інтерактивних та динамічних вебсайтів. Це одна з основних технологій веброзробки, яку підтримують усі сучасні веббраузери.

Код JavaScript можна вставляти безпосередньо у вебсторінки HTML або додавати в окремі файли, які завантажує вебсторінка. Його можна використовувати для широкого кола завдань, серед яких перевірка форм, інтерактивна анімація, покращення інтерфейсу користувача та маніпулювання даними.

Основою бекенду є фреймворк Python – Django.

Django – це високорівневий вебфреймворк Python, який дає змогу розробникам швидко й ефективно створювати вебдодатки. Він дотримується архітектурного шаблону Model-View-Controller (MVC), який розділяє модель даних, інтерфейс користувача та логіку програми на окремі компоненти, що полегшує керування та підтримку великих кодових баз.

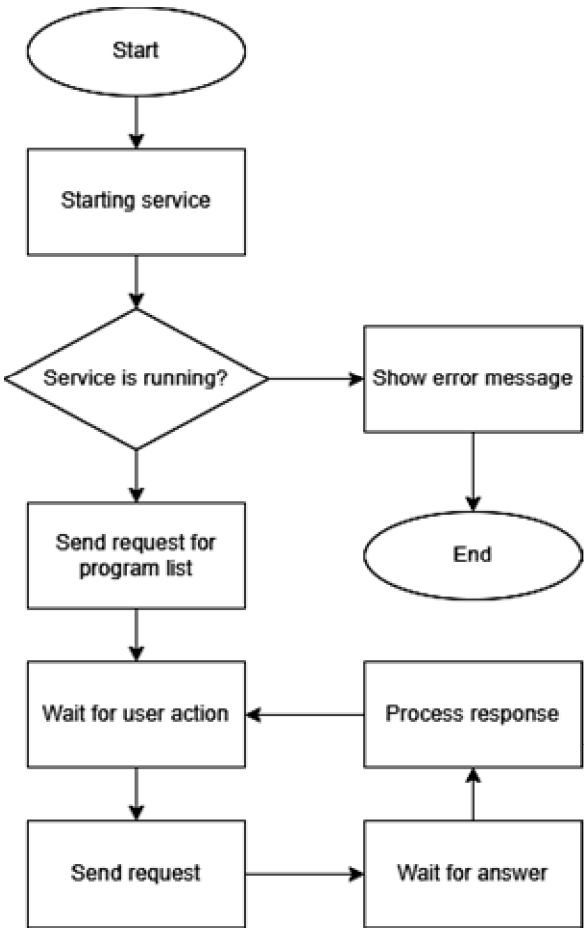

Рис. 5. Алгоритм роботи UI

2. BACKEND

Django надає широкий спектр вбудованих функцій та інструментів, ураховуючи потужну ORM (Object-Relational Mapping), яка дає змогу розробникам взаємодіяти з базами даних за допомогою коду Python, систему шаблонів для відтворення динамічних вебсторінок, вбудовану автентифікацію та авторизацію користувачів і автоматичні інтерфейси адміністратора для керування даними програм.

Django також легко настроюється та розширюється, що дає змогу розробникам забезпечувати додаткову функціональність за допомогою пакетів і бібліотек сторонніх розробників. Він часто використовується в поєднанні з іншими інструментами та фреймворками, такими як інтерфейсні бібліотеки JavaScript, як-от React або Angular, і може бути розгорнутий на різних платформах, зокрема хмарних хостингових служб, таких як Heroku та AWS.

Django має велике та активне співтовариство розробників, його документація та ресурси підтримки обширні. Це популярний вибір для створення широкого спектрв вебдодатків, зокрема систем керування вмістом, платформ електронної комерції, соціальних мереж тощо.

3. Бази даних

Вся база даних з її логікою та всіма підключеннями реалізована за допомогою postgresql і підключена до серверної частини.

PostgreSQL – це система керування реляційними базами даних (RDBMS) із відкритим кодом, відома надійністю, гнучкістю та потужними функціями. Її часто називають скорочено “Postgres”.

PostgreSQL розроблено для зберігання та керування великими обсягами структурованих даних із підтримкою розширених типів даних, таких як масиви, JSON і просторові дані. Він також надає такі функції, як транзакції, обмеження цілісності даних і повнотекстовий пошук.

Однією з ключових переваг PostgreSQL є його розширюваність. Він надає систему розширення, яка дає змогу розробникам додавати спеціальні функції до бази даних, такі як додаткові типи даних, методи індексування та процедурні мови. Це робить його популярним вибором для широкого діапазону додатків, від простих вебсайтів до складних систем корпоративного рівня.

PostgreSQL також притаманна висока масштабованість із підтримкою багатоверсійного керування паралелізмом (MVCC) і розділення таблиць. Його можна розгорнути на різноманітних платформах, зокрема локальних серверах та хмарних службах хостингу, таких як Amazon Web Services (AWS) і Microsoft Azure.

У PostgreSQL велике та активне співтовариство розробників, його документація та ресурси підтримки великі. Його широко використовують у таких галузях, як фінанси, охорона здоров'я та телекомунікації, а також урядові організації та некомерційні групи.

4. Візуалізація

Вся візуалізація побудована за допомогою Power BI.

Power BI – це служба бізнес-аналітики від Microsoft, яка дозволяє користувачам аналізувати та візуалізувати дані з різних джерел. Надає набір інструментів для створення інтерактивних звітів, інформаційних панелей і візуалізацій даних, а також дає користувачам змогу ділитися своєю роботою та співпрацювати з іншими.

Power BI можна використовувати для підключення до широкого діапазону джерел даних, ураховуючи електронні таблиці Excel, бази даних SQL Server і хмарні служби, такі як Azure і Salesforce. Він також підтримує різноманітні інструменти моделювання та перетворення даних, що дає користувачам змогу формувати та очищати свої дані перед їх аналізом.

Power BI містить конструктор звітів із функцією перетягування, який дозволяє користувачам створювати різноманітні та інтерактивні звіти з широким набором візуалізацій, зокрема діаграми, карти, таблиці тощо. Він також містить конструктор інформаційних панелей, забезпечуючи користувачам можливість створювати власні інформаційні панелі, до яких можна отримати доступ із веббраузера чи мобільного пристрою, та ділитися ними.

Power BI тісно інтегрується з іншими продуктами Microsoft, такими як Excel, SharePoint і Teams, завдяки чомуяє користувачам легко співпрацювати та ділитися своєю роботою з іншими. Він також надає розширені функції для керування даними та безпеки, зокрема рольовий контроль доступу та політики захисту даних.

Power BI доступний і як хмарна служба, і як настільна програма для Windows. Хмарна служба, яка називається Power BI Service, надає додаткові функції та можливості для спільного використання та співпраці, а настільна програма – Power BI Desktop надає потужніший набір інструментів для створення та розроблення звітів і візуалізацій.

Алгоритм роботи системи

Спочатку клієнт взаємодіє із клієнтською частиною за допомогою інтерфейсу та завантажує файли або заповнює дані вручну, ця інформація передається на бекенд сервера за допомогою протоколу HTTP, де вона обробляється, а потім надсилається запит до бази даних, де дані вже записані, оновлені чи видалені.

Для взаємодії клієнт-сервер використовують сокети, які допомагають забезпечити певний захист пакетів і зберегти їх цілісність.

Інтерфейсна частина має облікові записи адміністратора та користувача із різним функціоналом і, відповідно, доступом до них.

Серверна частина фіксує запити на отримання та публікацію, обробляє їх і додає логіку, а потім надсилає запити до бази даних або назад до клієнта після їх обробки.

Бази даних містять усю логіку обробки вхідних даних, на них навчається програма, яка потім вносить зміни в саму систему.

Рис. 6. Проста схема проєкту

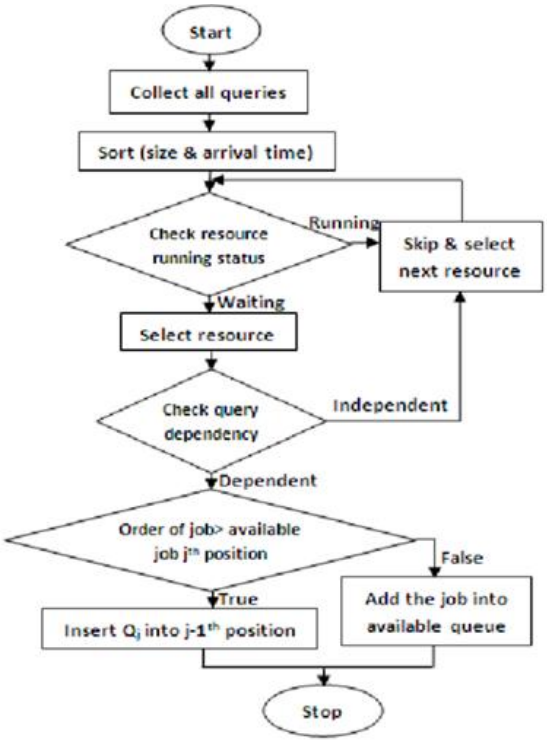

Рис. 7. Алгоритм бази даних

Оцінка поліпшення

Машинне навчання можна використовувати кількома способами, щоб удосконалити розроблення баз даних і керування ними. Ось кілька прикладів:

Якість і цілісність даних. Алгоритми машинного навчання можна використовувати для автоматичного виявлення та виправлення помилок у даних, гарантуючи, що база даних залишається точною та узгодженою. Наприклад, машинне навчання можна використовувати для виявлення дублікатів або відсутніх даних, а потім автоматично очищати або приписувати дані.

Оптимізація запитів: машинне навчання можна використовувати для оптимізації запитів до бази даних, виконуючи аналіз попередніх запитів і часу їх виконання. Вивчаючи ці дані, алгоритм може передбачити час виконання нових запитів і запропонувати зміни в плані запитів, які можуть підвищити продуктивність.

Виявлення аномалій. Алгоритми машинного навчання можна використовувати для виявлення аномалій у даних, що зберігаються у базі даних. Наприклад, алгоритм здатний виявляти незвичні моделі активності в транзакційній базі даних, що може вказувати на шахрайство.

Механізми рекомендацій. Алгоритми машинного навчання можна використовувати для розроблення механізмів рекомендацій, які пропонують користувачам релевантні елементи на основі їх

минулої поведінки. Наприклад, вебсайт електронної комерції може використовувати машинне навчання, щоб рекомендувати продукти користувачам на підставі їхніх попередніх покупок.

Обробка природної мови: алгоритми машинного навчання можна використовувати для розроблення програм обробки природної мови (NLP), які можуть аналізувати та розуміти людську мову. Наприклад, програму NLP можна використовувати для автоматичного вилучення інформації з неструктурованих текстових даних, таких як відгуки клієнтів або публікації в соціальних мережах.

Програма використовує навчання з підкріпленням, а саме навчання прийняття рішень і алгоритм q-навчання, що пришвидшує аналізування даних, тому що ми не використовуємо ресурс розробників та їх час на аналіз, система сама агрегує дані для подальших статистик. Час, витрачений на створення основних агрегацій для магазину продажів по регіонах, становить 20 хв. Зазначена система дає ті самі результати за 17,5 хв автоматично, оскільки система сама аналізує дані. Це пришвидшує пошук, витягування та операції з даними на 12,5 %. Наприклад, видобування усіх клієнтів просто з бази з покупками в Черкаській області займає 8 с, а із використанням системи час виконання цього запиту зменшується до 7 с, завдяки аналізу даних в цій базі та навчання на її даних.

Висновки

У статті наведено результат створення програми зі збереженням DWH за допомогою машинного навчання з двома репозиторіями. Ця система забезпечує аналіз великих обсягів даних і має як клієнтську, так і серверну частину, а також додатково візуалізує ці дані. Ця система є унікальним рішенням, оскільки використовує машинне навчання під час розроблення, що допомагає знайти найоптимальніші рішення для обробки та додавання нової логіки в проєкт. Описано всі аспекти програми від сторони клієнта до сторони сервера, зокрема як вони працюють і які алгоритми використовують. Розроблено сервер, який приймає запити від клієнта, а також архітектуру бази даних.

Програма допомагає вирішити проблеми наявних систем, зосереджуючись на поліпшенні обробки даних і підвищенні їх точності, що забезпечує більшу ефективність і спрощення подальшої логіки.

Програма підвищила швидкість аналізу даних на 12,5 %, зменшила час агрегації цих даних, а також автоматизувала деякі функції візуалізації цих даних для звітності. У результаті більше не потрібно робити це вручну або писати код для кожної нової порції даних і економити час команд

З погляду ефективності платформа успішно мінімізувала ручне втручання для аналізу даних на рахунок простих або вже прописаних агрегацій на 100 %, оскільки вже не вимагає втручання людини, а лише запуску або встановлення автоматичних запусків агрегацій. Надалі система автоматично візуалізує потрібні статистики чи дані, наприклад продажі у регіонах чи топ-покупців, а також зменшує кількість помилок у кінцевому результаті, оскільки не потребує втручання людини. Система забезпечує до 12,5 % пришвидшення аналізу даних та створення агрегацій. Програма досягла автоматизації та пришвидшила розроблення логіки для аналізу великих обсягів даних.

Для зручності використання платформа має інтуїтивно зрозумілий інтерфейс.

З погляду безпеки та конфіденційності платформа надає пріоритет безпеці та конфіденційності своїх користувачів, впроваджуючи суворі заходи безпеки та дотримуючись найкращих галузевих практик. Передбачено такі функції, як безпечні канали завантаження, високий рівень перевірки цілісності утиліт за допомогою цифрових підписів і повне шифрування усіх конфіденційних даних користувача.

Завдяки програмі користувач отримає якісніше проаналізовані дані та швидшу їх візуалізацію.

Список літератури

[1] Katharina Morik and Peter Marwedel (2023). Machine Learning under Resource Constraints – Fundamentals [Online]. Vol. 1. Available: https://www.degruyter.com/document/doi/10.1515/9783110785944/html

[2] Katharina Morik and Wolfgang Rhode (2023). Machine Learning under Resource Constraints – Discovery in Physics [Online]. Vol. 2. Available: https://www.degruyter.com/document/doi/10.1515/9783110785968/

[3] Katharina Morik, Jörg Rahnenführer and Christian Wietfeld (2023). Machine Learning under Resource Constraints – Applications [Online]. Vol. 3. Avaliable: https://www.degruyter.com/document/doi/10.1515/9783110785982/

[4] Ralph Kimball (2008, January 10). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling [Online]. Available: https://dl.acm.org/doi/10.5555/2543973

[5] Andrii Kirk (2016). Data Visualisation: A Handbook for Data Driven Design [Online]. Available: https://dl.acm.org/doi/book/10.5555/3002857

[6] Lowrence A. Rowe and Michael Stonebraker (2018). The implementation of PostgreSQL [Online]. Available: https://dl.acm.org/doi/10.1145/3226595.3226639

[7] T. Bray, Ed. (2017, December). The JavaScript Object Notation (JSON) Data Interchange Format. [Online]. Available: https://doi.org/10.17487/RFC8259

SPECIALIZED SOFTWARE PLATFORM FOR ANALYSIS OF INFORMATION IN DATA STORES

O. Kharchenko, Y. Klushyn

Lviv Polytechnic National University,
Computer Engineering Department

This article presents the design, development, and evaluation of a specialized program for analyzing, developing aggregations of this data, and visualizing large volumes of data. The main goal of this program is to simplify data processing, speed up their analysis, and make it easier to write code for problems with large amounts of data. To achieve this goal, machine learning is used, as well as two repositories.

The program includes a convenient and easy-to-understand interface, servers that process various types of requests from users and transfer them to the database, and the database itself with two repositories.

The research methodology used in this study involves a thorough analysis of existing programs and methods for solving problems with large volumes of data. This analysis informed the design of the core features of the program, which were then subjected to extensive testing and evaluation. A user study was conducted to evaluate the effectiveness of programs with machine learning in comparison to programs that work without it, and a comparison of the speed of implementations of program development and data processing was conducted.

The results of the study show that this approach has accelerated program development, accelerated data processing, and made it more qualitative and accurate. The study concludes that the platform has significant potential to improve the performance of large businesses and that with the growth of multiple times of data and technology, without using this, the development of programs with similar logic will be completely ineffective.

Key words: DWH storage; django; react; machine learning.