

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики

«До захисту допущено»

Завідувач кафедри ПМІ

(підпис) **О. А. Дмитрієва**
(ініціали, прізвище)

« ____ » _____ 2021 р.

**Випускна кваліфікаційна робота
магістра**

на тему Алгоритмічне та програмне забезпечення методів інтелектуального
аналізу даних

Виконав: студент _____ 2 _____ курсу, групи ПІЗМ-20
(шифр групи)

спеціальності 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

(прізвище та ініціали) _____
(підпис)

Керівник _____ доц. каф. ПМІ, к.т.н., доц. Назарова І.А. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____
(підпис)

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

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

Нормоконтроль:

(підпис) І. А. Назарова

(дата)

*Засвідчую, що у цій дипломній
роботі немає запозичень з праць
інших авторів без відповідних
посилань.*

Студент _____
(підпис)

Покровськ – 2021 р.

ДВНЗ «Донецький національний технічний університет»

Факультет комп'ютерно-інформаційних технологій та автоматизації

Кафедра прикладної математики та інформатики

Освітній ступінь магістр

Спеціальність 121 Інженерія програмного забезпечення

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМІ

 /О. А. Дмитрієва/

« » 2021 року

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Лепетченку Ярославу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Алгоритмічне та програмне забезпечення методів інтелектуально-го аналізу даних

керівник роботи Назарова Ірина Акопівна, к.т.н., доцент

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

затверджені наказом від «22» вересня 2021 року № 570

2. Строк подання студентом роботи 8 грудня 2021 року

3. Вихідні дані до роботи результати переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) огляд концепції інтелектуального аналізу даних; 2) застосування мови програмування Python у задачах інтелектуального аналізу даних;

3) Salesforce та CRM-системи; 4) Heroku та хмарні платформи; 5) архітектура та розробка додатку

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Назарова Ірина Акопівна	—	—

7. Дата видачі завдання 4 жовтня 2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд концепції інтелектуального аналізу даних	04.10.21-17.10.21	
2	Застосування мови програмування Python у задачах інтелектуального аналізу даних	18.10.21-31.10.21	
3	Salesforce та CRM-системи	01.11.21-14.11.21	
4	Heroku та хмарні платформи	15.11.21-21.11.21	
5	Архітектура та розробка додатку	22.11.21-30.11.21	
6	Оформлення пояснювальної записки та опис створеної системи. Проходження нормоконтролю	01.12.21-07.12.21	
7	Перевірка пояснювальної записки антиплагіатною системою. Оформлення презентації до роботи, графічного матеріалу та рецензування роботи	08.12.21-21.12.21	
8	Захист роботи	22.12.21	

Студент

(підпис)

Лепетченко Я.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Назарова І.А.

(прізвище та ініціали)

АНОТАЦІЯ

Лепетченко Я. В. Алгоритмічне та програмне забезпечення методів інтелектуального аналізу даних / Випускна кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 121 Інженерія програмного забезпечення. – ДВНЗ ДонНТУ, Покровськ, 2021.

Об'єктом дослідження у роботі є методи та процеси інтелектуального аналізу даних при застосуванні їх до прикладних задач у CRM-системах.

Предметом дослідження є статистичні методи оцінювання багатомірних величин.

Мета роботи полягає у розробці хмарного програмного продукту для обробки багатовимірних числових величин з ціллю добуття нових раніше невідомих залежностей, тенденцій, знань.

Наукова новизна полягає у створенні безкоштовного додатку для CRM-системи Salesforce.

Практичне значення полягає у наданні можливості користувачам платформи Salesforce обробляти дані шляхом роботи з додатком для інтелектуального аналізу даних.

Ключові слова: Data Mining, Salesforce, Python, CRM, API, REST, Heroku.

ABSTRACT

Lepetchenko Y. Algorithmic and program provision for Data Mining / Graduation qualifying work for obtaining an educational degree «Master» specialty 121 Software Security Engineering. – SHEI DonNTU, Pokrovsk, 2021.

The object of research is the methods and processes of data mining in the application of data to applied problems in CRM-systems.

The subject of research is statistical methods for estimating multidimensional quantities.

The purpose of the work is to develop a cloud software product for processing multidimensional numerical quantities in order to obtain new previously unknown dependencies, trends, knowledge.

The scientific novelty is the creation of a free application for CRM-system Salesforce.

Practical importance is to enable Salesforce users to process data by working with a data mining application.

Keywords: Data Mining, Salesforce, Python, CRM, API, REST, Heroku.

ЗМІСТ

	Стор.
Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	7
Вступ.....	8
1 Огляд концепції інтелектуального аналізу даних.....	10
1.1 Data Mining – як предметна область	10
1.2 Процес інтелектуального аналізу даних	12
1.3 Типи даних, що опрацьовуються при інтелектуальному аналізі даних	13
1.4 Області застосування інтелектуального аналізу даних	14
1.5 Методи інтелектуального аналізу даних	15
1.6 Висновки за розділом	24
2 Застосування мови програмування Python у задачах інтелектуального аналізу даних	25
2.1 Про Python	25
2.2 Розширення Numpy	28
2.3 Бібліотека SciPy	28
2.4 Matplotlib	29
2.5 IPython	29
2.6 Pandas	30
2.7 Scikit-Learn	31
2.8 Висновки за розділом.....	32
3 Salesforce та CRM-системи	33
3.1 Визначення CRM-систем	33
3.2 Salesforce та переваги цієї платформи	41
3.3 Висновки за розділом.....	47
4 Heroku та хмарні платформи	48
4.1 Опис Heroku	48

4.2 Висновки за розділом	52
5 Архітектура та розробка додатку	53
5.1 Ідея додатку	53
5.2 Розміщення Python-скрипту на платформі Heroku	54
5.3 Інтеграція Heroku у Salesforce	57
5.4 Висновки за розділом.....	57
Висновки.....	59
Список використаних джерел.....	61
Додаток А Зауваження нормоконтролера.....	64
Додаток Б Програмний код.....	
Додаток В Результати роботи програми.....	
Додаток Г Роздатковий матеріал.....	

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ООП – об’єктно-орієнтоване програмування

СКБД – система керуванням базою даних

API – Applied Program Interface

BPM – Business Process Management

CRM – Customer Relationship Management

DX – Developer Experience

HTML – HyperText Markup Language

OLAP – Online Analytical Processing

OLTP – Online Transactional Protocol

REST – Representational State Transfer

RPC – Remote Procedure Call

ВСТУП

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.

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.

У цій роботі буде розроблено додаток для аналізу даних на хмарній CRM-платформі Salesforce та Data Mining, що реалізується на мові програмування Python.

Об'єктом дослідження у роботі є методи та процеси інтелектуального аналізу даних при застосуванні їх до прикладних задач у CRM-системах.

Предметом дослідження є статистичні методи оцінювання багатомірних величин.

Мета роботи полягає у розробці хмарного програмного продукту для обробки багатовимірних числових величин з ціллю здобуття нових раніше невідомих залежностей, тенденцій, знань.

Наукова новизна полягає у створенні безкоштовного додатку для CRM-системи Salesforce.

Практичне значення полягає у наданні можливості користувачам платформи Salesforce обробляти дані шляхом роботи з додатком для інтелектуального аналізу даних.

РОЗДІЛ 1

ОГЛЯД КОНЦЕПЦІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

1.1 Data Mining – як предметна область

На сьогодні, бізнес колекціонує дані з блискавичною швидкістю. Джерела цих величезних обсягів даних різні: з використання кредитних карт, транзакцій, публічних даних користувачів, даних з фінансових та банкових установ, а також з даних, що надає користувач при реєстраціях та використанні додатків або техніки.

Збереження та передача таких великих обсягів даних вимагає великого розрахункового навантаження. Для цього існують реляційні бази даних, що базуються на серверах. Протокол онлайн передачі (Online transactional protocol – OLTP) вбудовується в системи керування базами даних (СКБД) для забезпечення бізнес-процесів при роботі з великими обсягами даних.

Отже, в OLTP – системах передаються та зберігаються дані, такі як продажі, ціни, контакти тощо. Цілком логічно, що користувач захоче використати цю інформацію для розробки подальших досліджень та прийняття рішень. З цією метою використовуються системи, що базуються на протоколі онлайн аналітики (Online analytical processing – OLAP). Таким чином, при оптимізації роботи компаній використовуються не лише дані, а й автоматизовані аналітичні системи, що базуються на цих даних. OLTP та OLAP системи працюють разом [1].

У той час, як OLTP системи зберігають масивні величини даних, що поповнюються тисячами нових записів щоденно, OLAP системи отримують ці дані та враховують їх при автоматичних розрахунках нової аналітики. Як відомо, дані грають чи не найважливішу роль у рості компаній, тож дуже важливо мати науково обґрунтовані рішення, що можуть пришвидшити зростання компанії [1].

Щоб добути знання могли принести прийняття вірних рішень у бізнесі, не достатньо лише володіти ними. Світова база знань наразі надто завелика, щоб зробити певні висновки у ній без спеціальних засобів та алгоритмів. Тому Data Mining або інтелектуальний аналіз даних – саме те, що може допомогти вирішити цю проблему.

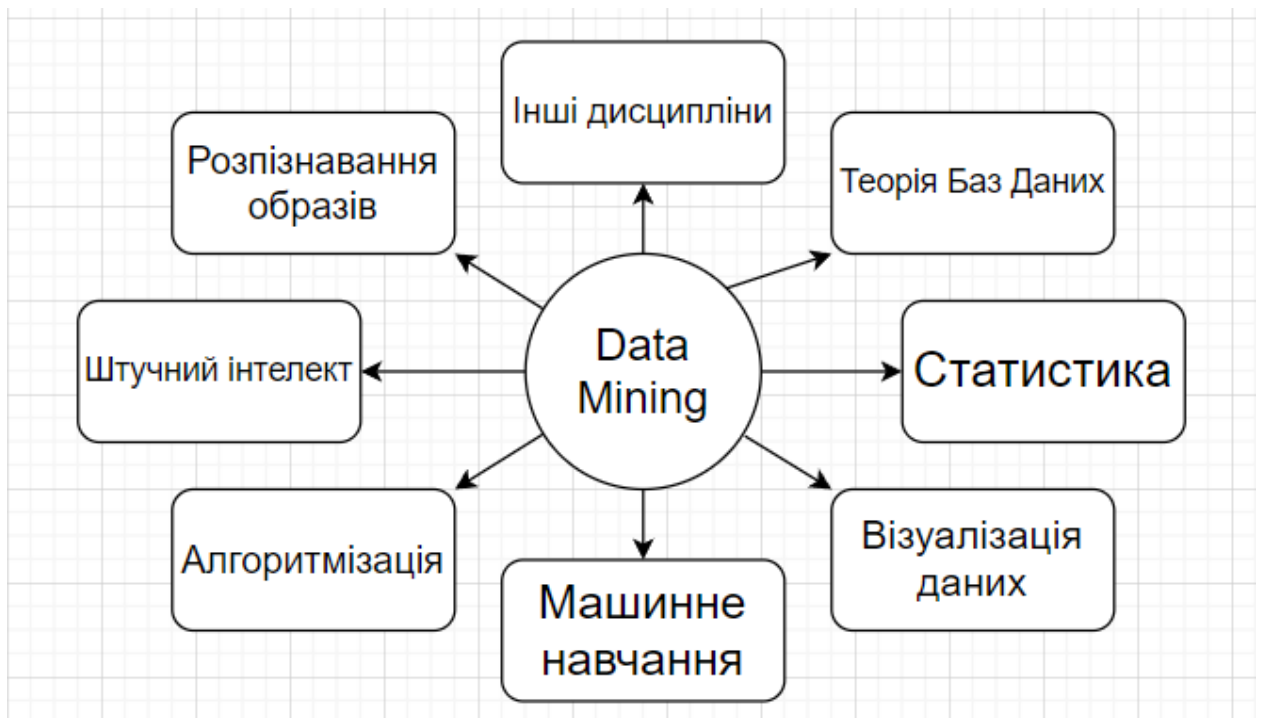


Рисунок 1.1 – Інтелектуальний аналіз даних, як мультидисциплінарна область науки

Інтелектуальний аналіз даних – процес, що дозволяє витягати інформацію з доступних наборів даних для виявлення тенденцій, шаблонів та цінної інформації. Ціль використання інтелектуального аналізу даних – зробити науково-доведені висновки з великих об’ємів даних.

Інтелектуальний аналіз даних працює спільно з предикативним аналізом – галуззю статистичної науки, що використовує комплекси алгоритмів, що розроблені для аналізу даних та створення передбачень на їх основі. Таким чином, можна натренувати роботу певного алгоритму, що базується на такій задачі інтелектуального аналізу даних, щоб вирішувати подібні проблеми у майбутньому. На цьому базується інша галузь – машинне навчання.

Застосовуючи машинне навчання на базі інтелектуального аналізу даних, користувач повністю автоматизує роботу з даними у OLTP та OLAP системах [1].

Також можна дати визначення для інтелектуального аналізу даних, як техніки дослідження шаблонів даних, що належать до певних груп. Впливаючи з цього формулювання, інтелектуальний аналіз даних категоризує та узгоджує дані для отримання корисної інформації. У подальшому нова корисна інформація збирається на серверах баз даних або використовується у подальших алгоритмах аналізу даних та у системах прийняття рішень [2].

Data Mining може аналізувати не лише дані у контексті баз даних, а ще у текстовому форматі, у вигляді зображень або відео при їх обробці, також може аналізувати соціальні мережі та веб-ресурси.

1.2 Процес інтелектуального аналізу даних

До того, як розпочати інтелектуальний аналіз даних, треба провести підготовку даних до обробки за наступними кроками.

Крок 1. Бізнес дослідження – Перед початком треба мати повне розуміння того, які дані мають оброблятися та які корпоративні цілі будуть досягнуті у процесі; на основі бізнес досліджень будується план інтелектуального аналізу даних, що дозволяє ефективно досягти мети.

Крок 2. Перевірка якості даних – оскільки дані збираються з різних джерел, є необхідність перевіряти та порівнювати їх для забезпечення об'єктивних результатів; Перевірка якості допомагає знайти аномалії в даних, таких, як втрата інтерполяції даних.

Крок 3. Очистка даних – займає близько 90%, цей етап включає відбір, очистку, форматування та анонімізацію даних до їх аналізу; перетворення даних – складається з п'яти етапів:

- згладжування даних – процес видалення шумів з даних;
- упорядкування даних – агрегація наборів даних;

- генералізація даних – процес узагальнення даних через заміну низькорівневих даних на високорівневі концептуалізації;
- нормалізація даних – впорядкування даних у набори;
- побудова атрибутів – у відповідність кожному наборові даних ставиться певний атрибут.

Крок 4. Моделювання даних – для більшої ідентифікації шаблонів даних застосовуються математичні моделі та алгоритми[2].

1.3 Типи даних, що опрацьовуються при інтелектуальному аналізі даних

Системи керування базами даних (СКБД) розміщують у собі дані, що базуються на реляційних зв'язках. Також СКБД мають ряд програмного забезпечення.

Воно дозволяє керувати даними та забезпечувати швидкий доступ до них, а також визначати структуру баз даних, забезпечувати захист та цілісність інформації, що зберігається, керувати типами доступу до даних. Реляційні бази даних містять таблиці з певними назвами, атрибутами та можуть зберігати ряди записів великих об'ємів даних. Кожен запис має унікальний ключ. Модель реляційних зв'язків існує, щоб забезпечувати взаємозв'язки між записами серед таблиць [3].

Якщо дані, зібрані з окремих джерел за окремим уніфікованим планом, то така форма представлення даних називається складом даних. У такій формі запису дані підлягають очистці, інтеграції, оновленні та надходженні нових даних.

Транзакційні бази даних зберігають записи, що фіксують взаємодію користувача з даними. Наприклад, це замовлення білетів, покупки товарів, фіксація натискань курсора на сторінках веб-сайту тощо. Кожен такий запис має унікальний ключ.

Серед інших типів даних, що піддаються інтелектуальному аналізу даних є наступні: потоки даних, графові дані, мультимедійні дані та багато інших. Всі вони діляться за структурою та семантичними значеннями.

1.4 Области застосування інтелектуального аналізу даних

Інформація – основний фундамент для формування сучасного світу. Саме тому обробка та захист даних є пріоритетом для багатьох бізнес-компаній, структур, держав. Отже, в кожній галузі інтелектуальний аналіз даних знаходить своє застосування [4].

Медицина. Інтелектуальний аналіз даних застосовується для визначення найкращих практик, що базуються на медичних даних та аналітиці, що дозволяють фабрикам зменшити витрати та покращити медичні показники ліків.

Data Mining разом із машинним навчанням, статистикою, візуалізацією даних та іншими підходами допомагають у передбаченні одужання пацієнтів, визначенні джерел спалаху захворюваностей, що, без сумніву, покращує як роботу медицини, як галузі, так і зменшує ризики та витрати [5].

Освіта. Ціль інтелектуального аналізу даних у цій галузі – розробка технік, щоб використовувати дані з навчання для нових досліджень. Метою дослідження цих методик є вивчення того, як навчальна підтримка впливає на студентів у контексті майбутніх потреб у їхньому навчанні та прогнозуванні їх майбутніх проектів. Також ці підходи можуть допомогти у оцінюванні учнів, що дозволить викладачам зосередитись на педагогіці [5].

Аналіз покупного кошика. У основі цієї методики лежить аналіз гіпотез про те, що при покупці певних продуктів, є великий шанс того, що покупець також купить товари, не пов'язані з першим та, які не належать з першим до однієї групи. Продавці використовують цю методику для аналізу ринку покупців та дослідження їх поведінки, а також для розміщення товарів на полицях [5].

Виробництво. Часто виробництво покладається на великі набори даних та інформацію, що базується на них. Інтелектуальний аналіз даних допомагає цим компаніям у ідентифікації шаблонів у процесах, що занадто комплексні для розуміння та сприйняття людиною. Через Data Mining можуть бути виявлені шаблонні зв'язки між різними рівнями роботи виробництва. Також

методика може бути корисною для виявлення загального часу, необхідного на виробництво продукту та передбачення потреб ринку [5].

CRM-системи. Це системи, ціль яких автоматизувати бізнес процеси та покращити довіру клієнта. CRM-системи (Customer Relationship Management) базуються на даних клієнтів, тому аналіз цих даних за допомогою Data Mining рішень – вигідна стратегія [5].

Банківська та фінансова сфери. Зараз база даних та знань у банківській сфері чи не найбільша та найдіджиталізована у світі. Техніки інтелектуального аналізу даних вже довгий час використовуються для вирішення тих чи інших проблем у цій галузі. Знаходження кореляцій, тенденцій та шаблонів серед великого обсягу різнорідних даних є занадто трудомістким для того, щоб бути опрацьованими людиною [5].

Виявлення браку. У процесі промислового виробництва не рідко трапляються випадки виготовлення бракованого продукту. Інтелектуальний аналіз даних дозволяє зібрати та розділити дані на ті, що призводять до браку та ті, що допустимі при виробництві. У подальшому ці дані аналізуються та використовуються у прийнятті рішень про стан продукту, що випускається[5].

Також, інтелектуальний аналіз даних використовується у тих випадках, коли є потреба відслідкувати шаблонні прояви, аномальні випадки, закономірності їх появи, передбачення випадків їх появи.

Часто задачі кластеризації, асоціації, класифікації використовуються у тих, чи інших бізнесових або соціальних процесах та дослідженнях.

1.5 Методи інтелектуального аналізу даних

Data mining або інтелектуальний аналіз даних – це процес пошуку серед великих обсягів даних для генерації нової інформації. На перший погляд, здається, що ціль Data Mining – пошук нових даних, але це не зовсім так. З іншого боку, ця технологія використовує шаблони екстраполяції та пошук нових знань у даних, які вже було зібрано [6].

Покладаючись на техніки та технології з розділів керування базами даних, статистики та машинного навчання, спеціалісти цієї сфери присвячують свої кар'єри для кращого розуміння того, як краще розраховувати та вимальовувати висновки з менших об'ємів даних. Далі трохи детальніше про техніки, які використовуються у технології інтелектуального аналізу даних.

Таблиця 1.1 – Методи інтелектуального аналізу[7]

Назва	Опис
1	2
Статистичні методи	<p>а) дескриптивний аналіз і опис вихідних даних;</p> <p>б) аналіз зв'язків (кореляційний і регресійний аналіз, факторний аналіз, дисперсійний аналіз);</p> <p>в) багатовимірний статистичний аналіз (компонентний аналіз, дискримінантний аналіз, багатовимірний регресійний аналіз, канонічні кореляції і ін.);</p> <p>г) аналіз тимчасових рядів (динамічні моделі і прогнозування).</p>
Кібернетичні методи	<p>а) штучні нейронні мережі (розпізнавання, кластеризація, прогноз);</p> <p>б) еволюційне програмування (в т.ч. алгоритми методу групового обліку аргументів);</p> <p>в) генетичні алгоритми (оптимізація);</p> <p>г) асоціативний алгоритм;</p> <p>д) нечітка логіка;</p> <p>е) дерева рішень;</p> <p>ж) системи обробки експертних знань.</p>

Продовження таблиці 1.1

1	2
Технологічні методи	<p>а) безпосереднє використання даних, або збереження даних. Методи цієї групи: кластерний аналіз, метод найближчого сусіда;</p> <p>б) виявлення і використання формалізованих закономірностей, або дистиляція шаблонів - логічні методи, методи візуалізації, методи крос-табуляції, методи, що засновані на рівняннях.</p>

Традиційні методи аналізу даних в основному орієнтовані на перевірку наперед сформульованих гіпотез (статистичні методи) і на «грубий розвідувальний аналіз», що становить основу оперативної аналітичної обробки даних (Online Analytical Processing, OLAP), тоді як одне з основних положень Data Mining – пошук неочевидних закономірностей. Інструменти Data Mining можуть знаходити такі закономірності самостійно і також самостійно будувати гіпотези про взаємозв'язки[8]. Оскільки саме формулювання гіпотези щодо залежності є найскладнішою задачею, перевага Data Mining в порівнянні з іншими методами аналізу є очевидною.

Більшість статистичних методів для виявлення взаємозв'язків в даних використовує концепцію усереднювання по вибірці, що приводить до операцій над неіснуючими величинами, тоді як Data Mining оперує реальними значеннями.

Методи Data Mining можна поділити на дві групи:

- Supervised Learning (навчання з вчителем);
- Unsupervised Learning (навчання без вчителя).

У першому випадку завдання аналізу даних, наприклад класифікація, здійснюється в кілька етапів. Це один із способів машинного навчання, в ході якого випробувана система примусово навчається за допомогою прикладів. Між входами і еталонними виходами може існувати деяка залежність, але вона

не відома. Відома тільки кінцева сукупність прецедентів, звана навчальною вибіркою. На основі цих даних потрібно відновити залежність, тобто побудувати алгоритм, здатний для будь-якого об'єкта видати досить точну відповідь. Для вимірювання точності відповідей, так само як і в навчанні на прикладах, може вводиться функціонал якості [9].

Спочатку, за допомогою будь-якого алгоритму Data Mining будується модель аналізованих даних – класифікатор. Потім класифікатор піддається «навчанню». Іншими словами, перевіряється якість його роботи і, якщо вона незадовільна, відбувається «додаткове навчання» класифікатора. Так триває до тих пір, поки не досягнемо необхідного рівня якості або не переконаємося, що обраний алгоритм не працює коректно з даними або ж самі дані не мають структури, яку можна виявити

Відстеження шаблонів – одна з базових технік в інтелектуальному аналізі даних. Її суть полягає у розпізнаванні певних шаблонів у часто оброблюваних наборах даних. Зазвичай, метод базується на розпізнаваннях певних відхилень в даних, які стаються регулярно, але з певними інтервалами, або на зміні певної змінної з часом. Наприклад, користувач помічає, що продажі певних продуктів збільшуються перед святами, або стає видно, що тепліша погода приводить більше людей на ваш веб-сайт [9].

Класифікація – більш комплексна техніка у інтелектуальному аналізі даних, що спонукає користувача збирати різні атрибути разом у помітні категорії, які потім можна використовувати, щоб робити подальші висновки або будувати певні функції [10].

Наприклад, якщо користувач оцінює данні покупної історії та фінансової основи окремого клієнта, класифікуючи їх за кредитним ризиком, як «низький», «середній» «високий», - то клієнти можуть бути віднесені до однієї з цих груп

Алгоритм C4.5 – один з найчастіше використовуваних алгоритмів у класифікації. Його було розроблено Росом Куінланом для генерації

класифікатору у формі дерева рішень з набору даних, що вже були класифіковані попередньо.

Задум алгоритму полягає у тому, що кожна точка даних має власні атрибути. Дерево рішень, створене C4.5, ставить питання про величину атрибута та на основі цього класифікує дані [10].

Асоціація – техніка, схожа з шаблонами відстеження, але частково використовує залежні змінні. Для цієї техніки використовується пошук спеціальних подій або атрибутів, які мають сильну кореляцію з іншим атрибутом або подією; наприклад, користувач може помітити, коли клієнт купує певний товар, то він купує також і другий товар. Ця техніка часто застосовується у алгоритмах для блоків онлайн-магазинів з заголовком «люди також купують» [10].

Алгоритм апріорі визначає асоціативні правила, тобто визначає та вивчає кореляції між змінними у базі даних. При роботі алгоритму визначаються цікаві шаблони та взаємні зв'язки. Хоча алгоритм і є корисним, він потребує багато пам'яті, дискового простору та займає багато часу.

Знаходження надзвичайних подій. В багатьох випадках розпізнавання шаблонів не може дати користувачеві розуміння набору даних, з яким він працює.

Іноді, при роботі з даними потрібно виявляти аномалії. Для прикладу, якщо покупці представляють певну групу, може виникнути необхідність дослідити пікові значення на функціях та що їх спричинило, це допоможе більш ретельно зрозуміти аудиторію.

Кластеризація – підхід, дуже схожий на класифікацію, але, шляхом групування шматків даних, метод виявляє їх схожість та ступінь схожості. Наприклад, користувач хоче кластеризувати аудиторію за демографічними ознаками у різні групи, базуючись на їх доході, або на тому, як часто вони відвідують магазин [11].

Мета кластеризації – пошук існуючих структур. Кластеризація є описовою процедурою, вона не робить жодних стратегічних висновків, проте дає можливість провести розвідчий аналіз та вивчити структуру даних.

Нехай X – множина об'єктів, Y – множина номерів (імен, міток) кластерів. Задана функція відстані між об'єктами є кінцева навчальна вибірка об'єктів. Потрібно розбити вибірку на непересічні підмножини, які називаються кластерами, так, щоб кожен кластер складався з об'єктів, близьких за метрикою, а об'єкти різних кластерів істотно відрізнялися. При цьому кожному об'єкту приписується номер кластера y_i [11].

Алгоритм кластеризації – це функція, яка будь-якому об'єкту ставить у відповідність номер кластера. Множина Y в деяких випадках відома заздалегідь, однак частіше ставиться завдання визначити оптимальне число кластерів, з точки зору того чи іншого критерію якості кластеризації [11].

Кластер можна охарактеризувати як групу об'єктів, що мають спільні властивості. Характеристиками кластера можна назвати дві ознаки:

- внутрішня однорідність;
- зовнішня ізолюваність.

Існує велика кількість підходів до кластеризації:

- алгоритми, засновані на поділі даних (Partitioning algorithms), в тому числі ітеративні: поділ об'єктів на k кластерів; ітеративний перерозподіл об'єктів для поліпшення кластеризації;
- ієрархічні алгоритми (Hierarchy algorithms);
- методи, засновані на концентрації об'єктів (Density-based methods);
- ґрид-методи (Grid-based methods);
- модельні методи (Model-based).

Слід зазначити, що в результаті застосування різних методів кластерного аналізу можуть бути отримані кластери різної форми. В результаті застосування різних методів кластеризації можуть бути отримані неоднакові результати, це є особливістю роботи того чи іншого алгоритму.

Однак створення подібних кластерів різними методами вказує на правильність кластеризації.

Існує ряд складнощів при проведенні кластеризації, наведемо їх нижче.

По-перше, складність вибору характеристик, на основі яких проводиться кластеризація. Необдуманий вибір призводить до неадекватного розбиття на кластери і, як наслідок, – до невірної рішення задачі.

По-друге, складність вибору методу кластеризації. Цей вибір вимагає хорошого знання методів і передумов їх використання. Щоб перевірити ефективність конкретного методу в певній предметній області, доцільно застосувати таку процедуру: розглядають кілька апріорі різних між собою груп і перемішують їх представників між собою випадковим чином. Далі проводиться кластеризація для відновлення вихідного розбиття на кластери. Частка збігів об'єктів в виявлених і вихідних групах є показником ефективності роботи методу.

Проблема вибору числа кластерів. Якщо немає ніяких відомостей щодо можливого числа кластерів, необхідно провести ряд експериментів і в результаті перебору різного числа кластерів вибрати оптимальне їх число; 4. Проблема інтерпретації результатів кластеризації. Форма кластерів в більшості випадків визначається вибором методу об'єднання. Проте слід враховувати, що конкретні методи прагнуть створювати кластери певних форм, навіть якщо в досліджуваному наборі даних кластерів насправді немає [11].

Одним з найбільш поширених алгоритмів кластеризації – алгоритм k-середніх. Суть алгоритму полягає у створенні k груп, що складаються з наборів даних, що базуються на спорідненості даних. Цей алгоритм не гарантує, що група членів буде однаковою, але чим більш схожими є члени групи, - тим більшим буде розрив з членами поза поточною групою.

Регресія – використовується здебільшого при формуванні та моделюванні, для знаходження відповідних змінних, що виправдують наявність інших змінних.

Наприклад, метод може використовуватись для підтвердження певної ціни на товар, що базується на інших факторах, таких як конкурентоспроможність, затребуваність у покупця та доступність. При більш детальному розгляді цього методу, ціль регресії – допомогти відкрити зв'язки між двома (або більше) змінними у даному наборі даних [12].

Головна особливість регресійного аналізу: за допомогою нього можна отримати конкретні відомості про те, яку форму та характер має залежність між досліджуваними змінними.

Послідовність етапів регресійного аналізу:

- формулювання задачі – на цьому етапі формуються попередні гіпотези про залежність досліджуваних явищ;
- визначення залежних та незалежних (пояснюючих) змінних;
- збір статистичних даних – дані повинні бути зібрані для кожної зі змінних, включених в модель;
- формулювання гіпотези про форму зв'язку (простий чи множинний, лінійний чи нелінійний);
- визначення функції регресії (полягає в розрахунку чисельних значень параметрів рівняння регресії);
- оцінка точності регресійного аналізу;
- інтерпретація отриманих результатів – отримані результати регресійного аналізу порівнюються з попередніми гіпотезами, оцінюється коректність та правдоподібність отриманих результатів;
- передбачення невідомих значень залежної змінної.

За допомогою регресійного аналізу можливе вирішення завдання прогнозування і класифікації. Прогнозні значення обчислюються шляхом підстановки в рівняння регресії параметрів значень пояснюючих змінних. Рішення завдання класифікації здійснюється таким чином: лінія регресії ділить всю множину об'єктів на два класи, і та частина множини, де значення функції більше нуля, належить до одного класу, а та частина, де воно менше нуля, – до іншого класу

Характер і форма залежності між змінними можуть утворювати такі різновиди регресії: позитивна лінійна регресія (виражається в рівномірному зростанні функції); позитивна рівноприскорено зростаюча регресія; позитивна рівноуповільнено зростаюча регресія; негативна лінійна регресія (виражається в рівномірному падінні функції); негативна рівноприскорено спадна регресія; негативна рівноуповільнено спадна регресія [12].

Предиктивний аналіз – одна з найбільш цінних технік у інтелектуальному аналізі даних, оскільки, використовується для проектування даних, що можуть виникнути у майбутньому. В багатьох випадках просте розпізнання та розуміння історичних тенденцій може допомогти у розумінні та прогнозуванні того, що може трапитись у майбутньому. Наприклад, можна зробити огляд кредитних історій покупців та їх минулих покупок для передбачення їх майбутніх кредитних ризиків [13].

Дерева рішень – це метод, що дозволяє передбачати приналежність спостережень або об'єктів до того чи іншого класу категоріальної залежної змінної в залежності від відповідних значень однієї або декількох предикторних змінних. Мета побудови дерев рішень полягає в передбаченні (або поясненні) значень категоріальної залежної змінної, і тому використовувані методи тісно пов'язані з більш традиційними методами дискримінантного аналізу, кластерного аналізу, непараметричної статистики [13].

Широка сфера застосування дерев рішень робить їх вельми привабливим інструментом аналізу даних, але не слід тому вважати, що його рекомендується використовувати замість традиційних методів статистики. Навпаки, якщо виконані більш строгі теоретичні припущення, що накладаються традиційними методами, і вибірковий розподіл має деякі спеціальні властивості, то більш результативним буде використання саме традиційних методів.

В найбільш простому вигляді дерево рішень – це спосіб представлення правил в ієрархічній, послідовній структурі. Основа такої структури – відповіді «Так» чи «Ні» на ряд питань.

Отже, інтелектуальний аналіз даних – найкращий набір підходів для якнайдетальнішого розгляду даних. Допоки користувач використовує вірні підходи та логіку, ставить правильні питання та вимоги до набору даних, - він може опрацювати висновки, які потенційно можуть максимально розвинути виробництво.

1.6 Висновки за розділом

Інтелектуальний аналіз даних, також відомий, як Data Mining, - це комплекс наукових методів, ціллю якого є здобуття нових раніше невідомих тенденцій, закономірностей, висновків з визначеного набору даних. Добувати знання можна кібернетичними, технологічними та статистичними методами. Окремі підходи до інтелектуального аналізу даних можуть бути застосовані у різних суспільних галузях.

РОЗДІЛ 2

ЗАСТОСУВАННЯ МОВИ ПРОГРАМУВАННЯ PYTHON У ЗАДАЧАХ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

2.1 Про Python

Мова програмування Python – одне з найуживаніших та найпопулярніших рішень для використання в області Data Science, та, зокрема, в інтелектуальному аналізі даних. Python – інтерпретативна мова програмування, що є об'єктно-орієнтованою, високорівневою та має динамічну семантику.

Цю мову програмування було розроблено Гвідо ван Россумом у 1990 році. Модель роботи зі структурами та даними високого рівня роблять цю мову програмування вигідним рішенням для швидкої побудови програм, поєднання існуючих компонентів програм та написання сценаріїв, що виконують різні завдання, базуючись на широкому пласті бібліотек та пакетів, що динамічно підключаються до середовища.

Об'єктно-орієнтована модель мови Python підтримує основні принципи ООП, такі як поліморфізм, спадкування та інкапсуляція. Це дозволяє підтримувати сценарії, написані на інших об'єктно-орієнтованих мовах програмування, та будувати додатки, що мають структурований код, який легко правити та доповнювати. При сполученні елементів програм, написаних, наприклад, на мовах Java або C++, використовуються мікросервіси, що називаються API (Applied Program Interface), таким чином, стає можливим успадкування класів та їх подальше використання для реалізації та повторного використання коду[14].

Форма підтримки та розповсюдження Python безкоштовна та базується на широкому співтоваристві у мережі Інтернет. Таким чином, до зростання та розвитку цієї мови програмування може долучитись будь хто, якщо його

нововведення пройдуть тестування, що відоме, як «Програма вдосконалення продукту» (у оригіналі PEP).

Реалізація мови Python здійснюється на ANSI, що здатен до швидкого переносу з платформи на платформу, завдяки чому він працює практично на всіх основних платформах. Тому писати програми та запускати їх можна практично на кожній відомій платформі, починаючи з надолонних комп'ютерів та закінчуючи суперкомп'ютерами[14]. Нижче наведено короткий перелік систем, які підтримують мову Python:

- UNIX-подібні системи, такі як Linux, Ubuntu;
- сучасні версії MS Windows та DOS;
- Mac OS;
- VMS, QNS;
- VxWorks та інші системи реального часу;
- суперкомп'ютери від компанії IBM;
- надолонні комп'ютери на системах PocketPc, PalmOS;
- Symbian системи;
- ігрові консолі;
- стільникові телефони, що працюють на основі операційних систем.

Окрім інтерпретатора, в складі мови Python є широка стандартна бібліотека модулів, що дозволяє взаємодію з багатьма функціями операційних систем. Також, слід зазначити, що програми на цій мові здатні компілюватись у байт-код, що працює сумісно з встановленою версією Python.

Ці фактори свідчать про те, що застосування цієї мови програмування є крос-платформним та однаково корисним при роботі з різними операційними системами. Крім того, у бібліотеках Python є специфічні механізми взаємодії при роботі з кожною з підтримуваних систем, що дозволяє розширювати можливості програм у межах специфічних систем.

Функціональні можливості Python дозволяють покривати діапазон між мовами розробки програмних систем, як Java, C#, C++ та мовами написання сценаріїв, як Perl, JavaScript, Ruby, тому Python може вважатися гібридною

мовою, що забезпечує простоту мов сценаріїв та функціонал мов, що компілюються.

Python забезпечує динамічну типізацію, що дозволяє розробнику не звертати увагу на приведення типів у програмі, уникаючи важких кодових формулювань та перевірок для забезпеченості приведення відповідності типів, а також це дозволяє динамічно обробляти цілий діапазон об'єктів.

Автоматичне керування пам'яттю дає можливість працювати, незважаючи на збір сміття у програмах та вивільненні пам'яті при роботі програми. Об'єкти у Python програмі здатні самі збільшувати або зменшувати обсяг пам'яті, що виділяється для роботи, тому розробник може не звертати увагу на низькорівневі операції з роботи з пам'яттю.

Для забезпечення багаторівневості розробки програм Python надає такі можливості, як модулі, класи та виключення. Тому, застосовуючи такі принципи розробки, як ООП, SOLID, GRASP або патерни проектування, дозволяють розробляти рішення багаторазового використання та системи, що можуть бути легко підтримувані та доповнені, а також опрацьовувати події та помилки, що виникають при роботі програми.

При програмуванні на Python розробнику доступні вбудовані типи об'єктів, що підтримують типові та прості типи даних, такі як списки, рядки, словники. Ці типи забезпечують високу гнучкість та зручність при розробці. Наприклад, вбудовані об'єкти, підтримуючи концепцію автоматичного керування пам'яттю, можуть стискатися або розширюватися по мірі необхідності, можуть комбінуватися для представлення складних структур даних, а також багато інших можливостей[15].

Для роботи та забезпечення вбудованих типів в складі Python є потужні стандартні засоби, що забезпечують операції над даними, такі як конкатенація, отримання частин колекцій, сортування тощо.

Для більш вузьких завдань застосовуються окремі бібліотеки, що підтримують функціонал, який може знадобитись при розробці, - від обробки та аналізу даних до роботи в мережі та підтримки мікро серверів.

Оскільки Python, як зазначалося вище, є відкритим програмним продуктом, розробники можуть створювати свої, попередньо скомпільовані, інструменти виконання завдань. У Мережі є реалізація підтримки COM, засоби для обробки зображень, інструменти для роботи з XML, робота з розподіленими об'єктами, механізми для доступу до баз даних тощо.

Запуск програм на мові Python відбувається інтерактивно без виконання проміжної компіляції та зв'язування. Такий принцип роботи інтерпретатора дозволяє отримувати результати роботи програми одразу ж після внесення змін та правок. Також важливим аспектом забезпечення зручності та низького порогу входу в Python є простий синтаксис та широкий набір інструментів.

2.2 Розширення Numpy

Numpy – розширення мови Python, що реалізує підтримку багатовимірних масивів та матриць даних, а також широкий спектр математичних операцій та функцій для операцій з ними. Це розширення є відкритим програмним забезпеченням та підтримується багатьма розробниками[16].

Математичні алгоритми можуть працювати повільніше, ніж у скомпільованих мовах, оскільки Python – інтерпретована мова програмування, але, за рахунок того, що розширення Numpy використовує для роботи масиви та матриці, більшість математичних операцій представлені як послідовність операцій над багатовимірними обсягами даних, що дозволяє наблизити роботу алгоритмів до швидкості, аналогічній деяким скомпільованим мовам [16].

2.3 Бібліотека SciPy

SciPy – відкрита бібліотека наукових інструментів для мови програмування Python. Містить рішення для оптимізації, інтегрування, обробки сигналів, зображень, генетичних алгоритмів, розв'язання диференціальних рівнянь та багатьох інших наукових задач, що застосовуються в інженерній розробці[17].

Часто разом з SciPy використовують бібліотеку Matplotlib для візуалізації даних. Наводяться наступні пакети:

- `scipy.integrate`: підпрограми числового інтегрування та розв’язку диференціальних рівнянь;
- `scipy.linalg`: підпрограми лінійної алгебри та розкладання матриць;
- `scipy.optimize`: алгоритми для оптимізації функцій та пошуку коренів;
- `scipy.sparse`: алгоритми роботи з розрідженими матрицями та рішеннями розріджених систем лінійних рівнянь;
- `scipy.stats`: стандартні неперервні та дискретні розподілення імовірностей (функції щільності імовірностей, формування вибірки, функції безперервного розподілення імовірності);
- `scipy.weave`: засіб для вбудови коду на C++ з метою прискорення розрахунків з масивами.

Спільно Numpy та Scipy утворюють практично повну заміну значної частини MATLAB та багаточисельних додатків до неї.

2.4 Matplotlib 25

Matplotlib – один з найпопулярніших в Python інструментів для створення графіків та інших засобів візуалізації двомірних даних. Була створена Джоном Хантером, а тепер підтримується великою групою розробників. Бібліотека чудово підходить для створення графіків, що можуть бути опубліковані.

Інтегрована з IPython, що дозволяє організовувати зручне інтерактивне оточення для візуалізації та дослідження даних. Графіки також є інтерактивними, що дозволяє керувати масштабами окремих ділянок графіків та панорамувати за допомогою панелі інструментів у вікні графіку[18].

2.5 IPython

IPython – компонент стандартного набору інструментів наукових розрахунків на Python, що забезпечує надійне високопродуктивне середовище

для інтерактивних та дослідницьких розрахунків. Це оболонка для Python з додатковими можливостями, що слугує для пришвидшення написання, тестування та відладку коду.

Особливо є корисним при роботі з даними та у комплекті з `matplotlib`. Компонент надає такі можливості:

- HTML-блокнот, схожий з програмами Mathematica для підключення IPython за допомогою веб-браузера;
- консоль з графічним інтерфейсом користувача на базі бібліотеки Qt, що включає засоби побудови графіків, багаторядковий редактор та підсвічування синтаксису;
- інфраструктуру для інтерактивних паралельних та розподілених обчислень[18].

2.6 Pandas

Pandas – вільна бібліотека Python, створена для проведення зручних маніпуляцій з даними та аналізу. Зокрема, дана бібліотека пропонує можливості роботи з числовими таблицями та часовими рядами. Назву бібліотека отримала від терміну з економетрії «panel data» – багатовимірною структурованого набору даних.

Даний пакет робить Python потужним інструментом для аналізу даних. Пакет дає можливість будувати зведені таблиці, виконувати угруповання, надає зручний доступ до табличних даних, а при наявності пакету `matplotlib` дає можливість малювати графіки на отриманих наборах даних[18].

Wes McKinney почав працювати над Pandas в 2008, коли у AQR Capital Management виникла потреба у високопродуктивному, гнучкому інструменту, для здійснення кількісного аналізу фінансових даних. Перед тим, як покинути AQR він зміг переконати керівництво дозволити йому віддати власні напрацювання у відкритий доступ. Інший співробітник AQR Chang She приєднався в 2012 як другий основний контриб'ютор бібліотеки. Приблизно в

цей час бібліотека стала популярною в Python спільноті, та збільшилась кількість контриб'юторів проекту.

На даний момент проект вважається одним з найбільш важливих і активних бібліотек аналізу даних на Python. До можливостей бібліотеки відносять:

- DataFrame – тип даних з інтегрованою індексацією, створений для маніпуляцій над даними;
- інструменти для зчитування та запису даних таких розширень, як: CSV, Excel, JSON, SAS, FWF;
- обробка відсутніх даних;
- підтримка операцій типу Group by;
- вставка та видалення строк/стовпців;
- злиття та приєднання датасетів;
- ієрархічна індексація осей для роботи з високорозмірними даними в низькорозмірних структурах даних;
- функціональність для роботи з часовими рядами.

2.7 Scikit-Learn

Scikit-Learn – це бібліотека, що надає реалізацію алгоритмів для так званого навчання з вчителем (Supervised Learning) та навчанням без вчителя (Unsupervised Learning).

Бібліотека має дистрибутиви для роботи з версіями ОС Linux, що сприяє її комерційному та науковому зростанню. Основою для scikit-learn є SciPy, що має бути встановлено до підключення scikit-learn. Основна концепція цієї бібліотеки – рівень її надійності та підтримки для ентерпрайз-рішень, тобто, велика увага приділяється якості коду, документації та оптимізації роботи бібліотеки[19].

Саме тому деякі частини scikit-learn написані за допомогою мови C, що значно пришвидшує її роботу.

Неповний перелік популярних функцій, що реалізуються на scikit-learn:

- метод k-середніх, як приклад рішення для задачі кластерного аналізу;
- проведення cross validation перевірки та оцінювання роботи незалежних даних;
- формування наборів даних з певними властивостями для дослідження поведінки моделі;
- зменшення розмірності даних для зменшення ключових атрибутів при відображенні даних;
- відбір ознак – інструмент, що дозволяє відбирати ключові ознаки для побудови моделі;
- оптимізація параметрів алгоритму;
- алгоритми навчання з учителем – дискримінантний аналіз, наївний Баєсівський класифікатор, методи опорних векторів, нейронні мережі, дерева прийняття рішень.

2.8 Висновки за розділом

Python – об’єктно орієнтована мова програмування високого рівня, що має доволі низький поріг входу та підтримується широким ком’юніті у Мережі. На даний час, ця мова програмування має доволі широкий перелік бібліотек, що постійно доповнюються та цілком покривають потреби такої наукової галузі, як інтелектуальний аналіз даних. Також, Python доволі легко інтегрується у різнорівневі рішення, оскільки є мовою програмування високого рівня.

РОЗДІЛ 3

SALESFORCE ТА CRM-СИСТЕМИ

3.1 Визначення CRM-систем

CRM (Customer Relationship Management) – це системи для керування відносинами з клієнтами, що містить методи, стратегії, інструменти, підходи та технології, що використовуються бізнесом для розвитку, утримання та залучення клієнтів. При такому підході до ведення бізнесу на перше місце для компанії ставиться клієнт та його потреби [19].

Мета впровадження CRM-стратегій – створення злагодженого, різнорівневого середовища для роботи та розвитку нових та існуючих клієнтів. Керуючи взаємовідносинами з клієнтами значить отримувати нових клієнтів, перетворювати нейтральних клієнтів на лояльних, формувати бізнес-партнерів з постійних клієнтів [19].

Основна перевага CRM-систем – те, що вони здатні позитивно впливати на рівні будь-якого організаційного робочого підрозділу – від відділу продаж та роботи з клієнтами до відділу наймів, маркетингу та бізнес розвитку. Уся інформація про клієнтів зберігається в одному місці, реєстрація проблем обслуговування, визначення можливостей, пріоритетів та інший аналіз продажів, керування маркетинговими кампаніями – усе це про можливості CRM-систем.

CRM-системи сприяють швидкому зростанню клієнтської бази. Органічний пошук, інтернет-реклама, e-mail розсилки, медіа-активності, онлайн-заходи. Система чітко визначає канал-джерело та допомагає аналізувати ефективність кожного. CRM розрахує не тільки кількість лідів по каналах, але і їх конверсію у продаж. Зібрана аналітика покаже, які канали необхідно посилити, а які відключити [19].

Комунікація відбувається за допомогою вбудованих в CRM телефонії, месенжера або e-mail. Система зберігає всю історію взаємодії та записи

розмов для їх подальшого аналізу супервайзером. CRM-система працює за принципом омніканальності – інтеграція розрізнених каналів комунікації в єдину платформу, яка забезпечує максимальну ефективність та безперервний контакт з клієнтом [19].

Розкласти робочі процеси по полицях, формалізувати їх - нетривіальне завдання, яке вирішується бізнес-аналітиками. Якщо співробітники діють згідно регламентованим процесам, зменшується кількість помилок, робота компанії прискорюється, а результати стають більш прогнозованими. Керівникам набагато легше виявляти слабкі місця в роботі і направляти зусилля команди в потрібне русло.

Ці завдання можуть бути вирішені за допомогою технологій управління бізнес-процесами, які доступні в деяких потужних CRM-системах. Поєднуючи управління взаємовідносинами з клієнтами та BPM-технології, компанія досягає прискорення роботи всіх підрозділів.

Бізнес-аналітик оцінить зручний візуальний редактор, який дозволяє побудувати алгоритмічні процеси будь-якої складності. Менеджери зможуть вказати виконавців і учасників задач по процесу, зафіксувати витрачений час, а керівники отримають можливість аналізувати ефективність процесів.

Ефективне управління та облік робочого часу позитивно впливає на всі бізнес-процеси компанії. CRM-система дозволяє співробітникам організації не тільки протоколювати використання поточного робочого часу, а й планувати завантаження на майбутні періоди.

CRM програма пропонує зручний доступ до розкладу, в якому співробітник може планувати власний робочий час, відзначати результати виконання запланованих справ, переглядати розклад колег. У свою чергу, у керівництва з'являються інструменти для контролю завантаженості і ефективності роботи підлеглих.

Завдяки можливості пов'язувати всі завдання з контрагентами, контактами, угодами формується і накопичується історія роботи з кожним клієнтом. Автоматична генерація 80% завдань по бізнес-процесу дозволяє

звільнити співробітників від рутинних операцій, найбільш раціонально розподіляючи робочий час, і не забути жодної важливої справи.

Спосіб роботи з клієнтом залежить від типу продажі (коротка або довга). Якщо продаж коротка, наприклад, замовлення в інтернет-магазині, то система допоможе менеджеру швидше оформити замовлення і передати його в підрозділ доставки. Якщо мова йде про корпоративні або довгі продажі, які розділені на декілька стадій, що мають на увазі постійну взаємодію, то клієнт переходить до воронки продажів [19].

Один з найефективніших інструментів для щоденної роботи відділу продажів. Інструмент дозволяє розділити процес угоди на необхідну кількість етапів, вибудувати тактику взаємодії на кожному з них, визначати відсоток конверсії (переходу) між стадіями, а також оцінювати ефективність менеджерів. З точки зору плану продажів воронка покаже обсяг потенційних угод на фінальній стадії, а також загальні показники в поточний момент.

На всіх етапах роботи з клієнтом CRM допомагає вести документообіг, формувати рахунки на оплату за заданими шаблонами і відправляти їх клієнтам прямо з інтерфейсу системи. Система дозволяє працювати не тільки з документами для клієнтів, але і створювати/візувати внутрішні заявки, що значно прискорює і спрощує повсякденну роботу.

Неможливо підвищити рентабельність підприємства без глибокого аналізу інформації про клієнтів, їх цінності та прибутковості, виявлення «вузьких місць» в бізнес-процесах компанії, аналізу системи продажів. CRM-система дозволяє компанії отримати статистичну інформацію, провести складний аналіз даних, необхідний для прийняття стратегічно важливих бізнес-рішень.

Стандартні звіти дають можливість аналізувати і контролювати всі типові завдання бізнесу. За допомогою вбудованого генератора звітів можна створити аналітичні форми, що відповідають специфічним завданням кожного підприємства.

Планування продажів в CRM-системі організовано в різних зрізах (по регіонах, менеджерам, напрямкам і ін.). Менеджер складає план по своїм клієнтам на основі даних з урахуванням ймовірності, а керівник, проаналізувавши обсяг підтверджених платежів, може скласти для менеджера стимулюючий план.

Одна з головних функцій CRM-системи – допомагати менеджерам планувати продажі, організовувати прозоре управління угодами та оптимізувати канали продажів. Система зберігає повну історію спілкування з клієнтами, що допомагає департаменту продажів аналізувати поведінку клієнтів, формувати для них відповідні пропозиції, завойовувати лояльність.

CRM програми – це ще і організація cross-sales та up-sales. Системи дозволяють формувати матрицю крос-продажів і продуктово-сегментну матрицю, групувати клієнтів за різними параметрами і виявляти їх потенційні інтереси.

Пропонуючи інструменти прогнозування та аналізу, автоматизуючи взаємодію співробітників із клієнтами та між собою, система формує передумови для оптимізації існуючих каналів збуту та збільшення прибутку компанії.

Керівники пред'являють особливі вимоги до CRM. За допомогою готових інструментів керівники можуть контролювати якісні показники роботи менеджерів (воронку продажів), виконання планів продажів, дотримання термінів оплати і поставки. Система дозволяє оцінювати обсяг і ймовірність угод, управляти бізнес-процесами продажів, стежити за станом угоди і аналізувати дії конкурентів.

Будь-якій комерційній організації важливо надавати продукти та послуги високої якості за конкурентними цінами, постійно вдосконалювати продуктові лінійки.

CRM система дозволяє структурувати номенклатуру і управляти повним каталогом товарів і послуг компанії. Можливості CRM забезпечують ведення обліку спеціальних цін та знижок, аналіз даних і взаємозв'язків для надання

клієнтам оптимальної пропозиції пакета послуг та визначення популярних продуктів.

Крім того, функціональність систем включає інструменти для відстеження КРІ (ключових показників діяльності), аналіз яких дозволить керівництву оцінювати ефективність роботи кожного співробітника.

«Серцем» будь-якої CRM-системи є база даних як фізичних, так і юридичних осіб, які взаємодіють з вашою компанією в рамках діяльності підприємства. Це не тільки клієнти, але і філії компанії, партнери, постачальники, конкуренти. Інформація про клієнтів сама по собі цінний актив, а грамотне управління даними в системі дозволяє використовувати її в роботі з максимальною ефективністю. Клієнтська база консолідована, організація отримує повну інформацію про своїх клієнтів та їх вподобання і, ґрунтуючись на цих відомостях, будує стратегію взаємодії.

Єдина база клієнтів і повна історія взаємин з ними в сукупності з потужними аналітичними інструментами CRM дозволяє утримувати і розвивати існуючих клієнтів, виявляючи найбільш цінних, а також залучати нових клієнтів.

CRM програма скорочує час на щоденну рутину. Система передбачає всі необхідні інструменти для управління як зовнішнім, так і внутрішнім документообігом компанії. Ці інструменти надають засоби автоматичного формування документів по шаблону, підготовки друкованих форм документів, підтримки актуальної версії документів, швидкого пошуку документів в системі, створення електронного сховища документів і багато іншого.

При веденні в CRM документації можна організувати колективну роботу з документами при гнучкому розмежуванні прав доступу, електронне візування, а також облік взаємозв'язків між документами.

Якщо угода завершується продажем, то клієнт переводиться в сегмент постійних. Це означає не тільки висококласне сервісне обслуговування, а й роботу департаменту маркетингу, який буде періодично комунікувати з клієнтом, а CRM допоможе вибрати найефективніший канал і частотність

взаємодій. Результат такої стратегії – повторний продаж, якщо мова йде про товари, або продовження послуги чи сервісу.

У разі, коли клієнт бере тривалу паузу в комунікаціях або відмовляється від покупки, CRM повертає його на початкову стадію воронки для повторного взрощування. Персональні пропозиції та чіткий маркетинговий таргетинг дозволяють отримати бажаний результат. На всіх стадіях роботи система управління взаємовідносинами з клієнтами пропонує деталізовану аналітику 24/7, представлену у вигляді інформативних дашбордів. Керівники різного рівня можуть аналізувати поточний стан продажів, завантаження менеджерів, їх результативність, планувати і контролювати маркетингові витрати та багато іншого.

Низький рівень розвитку комунікацій між співробітниками та підрозділами робить роботу компанії малоефективною і призводить до збоїв основних бізнес-процесів. Як наслідок – знижується прибутковість бізнесу.

CRM система дозволяє організувати ефективну взаємодію та обмін інформацією всередині компанії, перешкоджаючи виникненню «інформаційних провалів» і втрати важливої інформації. Використання CRM-системи в компанії допоможе синхронізувати дії персоналу, контролювати виконання функціональних ролей команди в угодах, за допомогою процесів організувати автоматичний розподіл завдань між співробітниками.

Завдяки застосуванню єдиних корпоративних стандартів і кращих практик ведення бізнесу, CRM-система забезпечить швидке навчання нових співробітників.

CRM-система дозволяє оптимально організувати управління маркетингом компанії: проводити маркетингові заходи, управляти ресурсами та бюджетами на маркетинг, координувати маркетингові дії.

Співробітники відділу маркетингу отримують єдину бібліотеку маркетингових матеріалів, інструменти для сегментації клієнтів, виконання персоналізованих розсилок для цільової аудиторії. А для вимірювання

результатів кампаній і ефективності відділу маркетингу CRM-система пропонує спеціальні інструменти аналізу.

Серед базових функцій системи для автоматизації маркетингу – управління прямими маркетинговими акціями (електронна розсилка, пряма розсилка), організація досліджень, опитувань клієнтів. Автоматизація допомагає вдосконалити роботу департаменту маркетингу та підвищити конверсію лідів в покупців.

Компанії ретельно продумують спосіб розгортання CRM-системи, щоб переконатися, що вона відповідає їх бізнес-цілям. Деякі організації можуть отримати вигоду з локального програмного забезпечення, однак у хмарної CRM-системи також є свої переваги.

Роблячи вибір на користь локальної CRM, компанії необхідно заздалегідь придбати підписку на її використання. З огляду на той факт, що програмне забезпечення розташоване на серверах самої компанії, локальна CRM стане хорошим вибором для організацій з суворою політикою безпеки. При виборі локальної CRM процес розгортання потребує більше часу та ІТ-ресурсів. Завданням покупця також є забезпечення постійного обслуговування локальної системи, постачальник не несе відповідальності за бездоганну роботу серверів замовника. У той же час, постачальник надалі надає технічну підтримку, яка дозволяє вирішувати критичні проблеми [19].

З іншого боку, програмне забезпечення як послуга (SaaS) або хмарна CRM зберігає дані на серверах сертифікованого хмарного провайдера. У цьому випадку користувачі мають повний доступ до бази даних, де б вони не знаходилися.

Постачальник несе відповідальність за безперебійну роботу системи і забезпечує всі необхідні оновлення та підтримку. Важливим є те, що хмарна CRM має високу масштабованість. Це означає, що організація може легко розширювати не тільки функціональні можливості системи, але і її продуктивність по мірі зростання вимог бізнесу. Єдиною вимогою до доступності хмарної системи є надійне підключення до інтернету. Як правило,

більшість компаній вибирають хмарну версію розгортання CRM-системи, але й варіант on-site також доступний для вибору.

У сьогоденному мінливому бізнес-середовищі перевагою компанії стає об'єднання CRM і BPM технологій. BPM (Business Process Management, Управління бізнес-процесами), є підходом, який фокусується на оптимізації бізнес-операцій для підвищення організаційної ефективності та досягненню бізнес-цілей.

За своєю суттю BPM включає в себе цикл моделювання, автоматизації, виконання, контролю, вимірювання та оптимізації потоків ділової активності. Будучи ядром CRM-системи, BPM технології дозволяють організаціям швидко трансформувати процеси враховуючи зміни зовнішніх факторів. Це дає користувачам негайний доступ до всієї необхідної інформації, яка їм потрібна, що значно прискорює робочі процеси. Основною метою BPM є узгодження всіх організаційних елементів для підвищення ефективності роботи. Ось деякі переваги у злитті CRM і BPM:

- професійні інструменти для моделювання, зміни та моніторингу процесів;
- доступ до технологій для управління неструктурованими процесами;
- зрозумілі та прозорі робочі процеси з самого першого дня використання системи;
- управління фронт, мідл- і бек-офісними процесами в єдиному середовищі;
- розширена аналітика по ефективності продажів, маркетингу, сервісу, операційних процесів;
- швидкий обмін інформацією по всій компанії завдяки інтеграції в єдину ІТ-екосистему підприємства;
- масштабованість, безпеку і мобільний доступ до CRM;
- передові інструменти для кращої комунікації між відділами та споживачами.

3.2 Salesforce та переваги цієї платформи

Salesforce уже не перший рік є світовим лідером серед CRM-платформ, і, незважаючи на це, в Україні не так багато людей знайомі з цією платформою. Salesforce – це платформа, яка повністю хоститься на серверах компанії Salesforce в хмарі.

Компанія Salesforce була заснована в 1999 році колишнім виконавчим директором Oracle Марком Беніоффом. Головна ідея створення – це побудова доступного програмного забезпечення і впровадження його повністю онлайн в якості сервісу. Salesforce дозволяє створювати та розгортати індивідуальні рішення, автоматизувати бізнес-процеси, інтегруватися із зовнішніми додатками [20].

Більшість світових ентєрпрайз-компаній є клієнтами компанії Salesforce і використовують цю платформу як рішення для своїх бізнес-потреб. Серед них: Adidas, AWS, Canon, Philips, Toyota, American Express, Western Union, Cisco, KLM і багато інших. Salesforce, як компанія, є партнером різноманітних представників ІТ-світу і, відповідно, вони пропонують готові інтеграційні рішення, удосконалені сервіси для задоволення вимог кінцевих клієнтів і ще багато різних можливостей. Серед них – Apple, Microsoft, Google, Amazon.

Важливо зазначити, що Salesforce розроблена так, щоб мати досить низький поріг входу. Платформа не має вимог до обладнання та для роботи не потрібно встановлювати жодних програм. Розміщення в хмарі надає доступ до інформації в режимі реального часу, тобто нею можна користуватися будь де і будь коли.

Часто, характеризуючи Salesforce, зазначають про архітектуру платформи. Multitenant Architecture – платформа для багатьох користувачів. Усі користувачі Salesforce мають спільну інфраструктуру та екземпляр програмного забезпечення.

Це дає можливість автоматичного та одночасного оновлення для всіх користувачів на платформі. Таким чином, отримуються найсвіжіші та найновіші функції з автоматичним оновленням тричі на рік.

Figure 1: Magic Quadrant for the CRM Customer Engagement Center



Source: Gartner (June 2021)

Рисунок 3.1 – Частота використання популярних CRM-систем станом на 2021 рік за версією видання Gartner

Кожен користувач має змогу створити тестову безкоштовну організацію для навчання, практикування або тестування. Це фактично повнофункціональне середовище для розробки, яке можна також використовувати для створення власних продуктів – пакетів, які можуть випускатися як додатки в AppExchange.

AppExchange у Salesforce дозволяє розробляти та продавати власні продукти або отримувати доступ до тисяч корисних, захищених та перевірених продуктів чи інтеграцій, створених іншими користувачами. Наприклад, apex-lang – це відкрита бібліотека допоміжних класів, написаних на Apex, метою якої є усунення недоліків у основних класах платформи.

Пакет Rollup Helper дозволяє агрегувати дані за певним набором критеріїв у Salesforce без використання Apex коду.

Salesforce дає можливість для будь-якої інтеграції. Крім того, пропонується багато вбудованих інтеграцій – наприклад, з Heroku, Outlook, Gmail.

Для вхідних викликів використовується apex REST веб-сервіс. Для вихідних викликів використовується Apex callout – HTTP request з Apex коду.

Платформа легка у використанні та налаштуванні. Інтуїтивна і зрозуміла панель навігації. Всі вкладки розділені залежно від категорії. Для доступу до налаштувань чи кастомних розробок використовується Setup меню. Меню розділене на різні секції в залежності від мети використання [20]. Наприклад, налаштування об'єктів і полів відбувається через Object Manager.

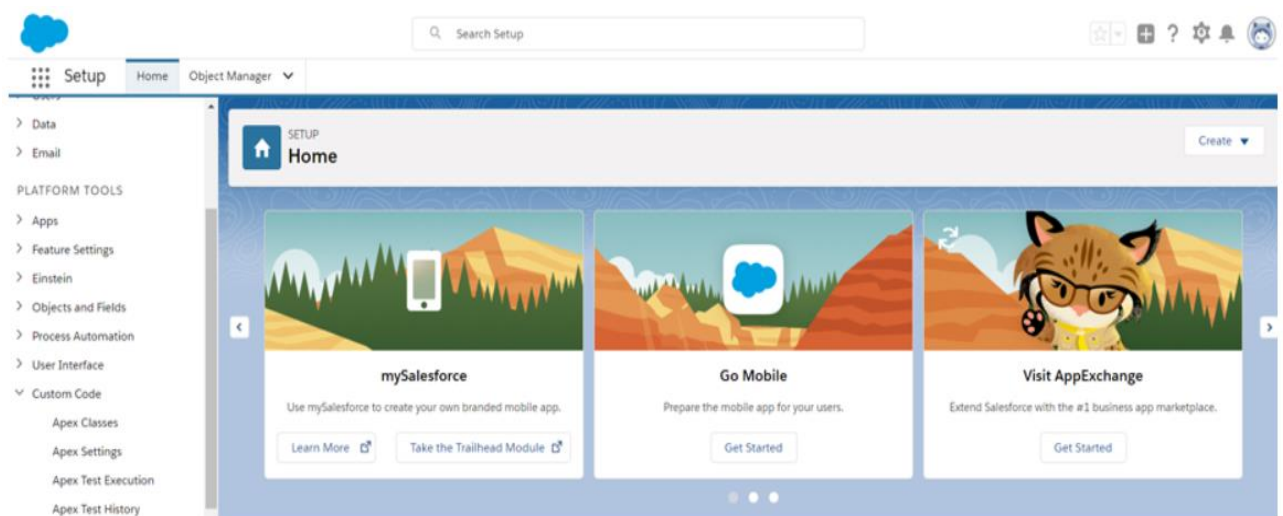


Рисунок 3.2 – Робоче середовище Salesforce

Однією з найбільших переваг Salesforce є те, що велика кількість функціоналу, яку вимагають для більшості продуктів, уже імплементована і протестована.

Наприклад, вбудований функціонал Email-to-Case дозволяє автоматично створювати Case записи в системі з електронної пошти.

Або Web-to-Lead, який створює HTML форму, яку можна вставити на свій веб-сайт і створювати записи лідів безпосередньо з сайту (Lead – потенційний клієнт в розумінні SF) [21].

Значну частину функціоналу можна просто конфігурувати без єдиної лінійки коду за короткий період часу. А це піднімає Salesforce на рівень вище порівняно з традиційною розробкою з нуля. Розробка з нуля несе набагато більші ризики для клієнта і займає набагато більше часу.

Salesforce – це платформа, яка постійно змінюється і розвивається, даючи вам нові можливості для вивчення і вдосконалення. Компанія займає активну позицію стосовно стартапів. У 2010 році була придбана одна з перших хмарних платформ Heroku, яка дозволяє розгортати, запускати та управляти додатками, написаними на різних відкритих мовах і фреймворках. Протягом останніх років Salesforce активно інтегрує елементи штучного інтелекту (AI) в свою платформу.

Технологія використовує всі дані в межах системи: дані клієнта, інформацію з листувань, електронної пошти, календаря та електронної комерції, потоки даних із соціальних мереж, такі як твіти та фотографії, для створення моделей машинного навчання.

Один з прикладів AI в Salesforce – функціонал Einstein Bots for Service Cloud, який використовує машинне навчання для кращого та швидшого обслуговування клієнтів. Це дозволяє передавати запити клієнтів ботам, які застосовують поєднання машинного навчання та обробки історії клієнтів для прийняття рішення.

Salesforce Trailhead – це онлайн-платформа для вивчення Salesforce, яка дає можливість дотримуватися вже готових способів навчання або обрати свій шлях – і все у своєму власному темпі [22].

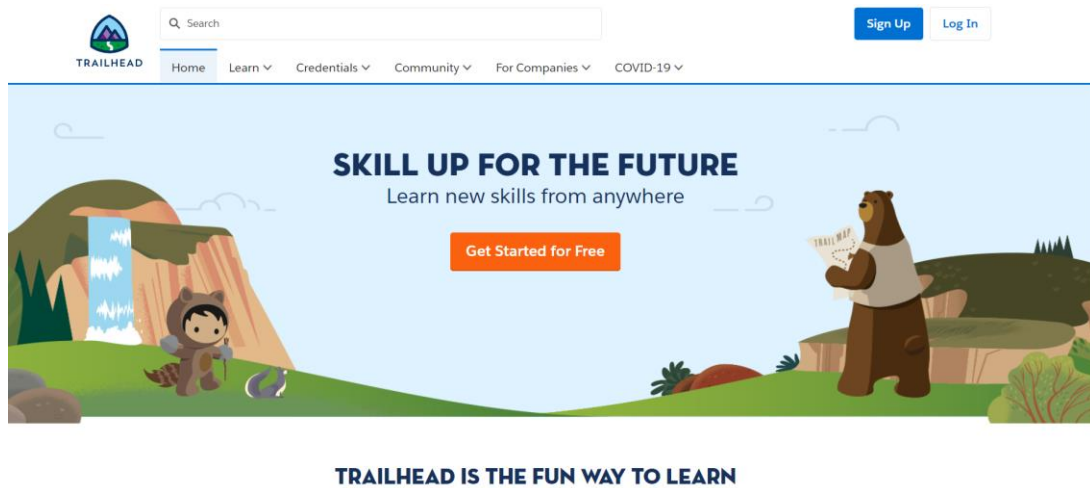


Рисунок 3.3 – Стартова сторінка Salesforce Trailhead

Якщо говорити про кастому розробку, то Salesforce має достатньо потужні інструменти для цього. Серверною мовою є Apex. Це Java-подібна мова, яка підтримує більшість можливостей, щоб писати код в стилі ООП.

Так само, як інші ООП мови програмування, Apex підтримує класи, інтерфейси, колекції, змінні та константи, умовні оператори (if-then-else), for і while цикли. Але, крім цього, apex розробляється, зберігається, компілюється і виконується в клауд, дає можливість здійснювати прямі виклики бази даних (SOQL і SOSL query).

Apex може виконуватися тригерами до або після певних DML операцій. Крім того, Apex є case-insensitive мовою.

Зазвичай Apex використовується для виконання складних бізнес-процесів, створення веб-сервісів чи email-сервісів. Apex можна викликати за допомогою тригерів – до або після змін у базі даних, є можливість запускати класи Apex в певний час.

Якщо говорити про UI – тут є два варіанти. Перший і старіший – Visualforce pages – підтримує MVC (Model-View-Controller) паттерн. В основному Visualforce сторінки розробляються у так званому Classic Mode – це інтерфейс, з якого Salesforce стартував в далекому 1999. Сторінки Visualforce складаються з 2 елементів: Visualforce Markup та Visualforce Controller (останній є не обов'язковим).

Visualforce Markup може містити стандартні теги (бібліотека Visualforce має великий набір власних тегів), а також підтримує будь-яку стандартну веб-технологію (CSS, JavaScript, HTML, Flash і ін.). Visualforce Controller може бути стандартним або кастомним – це фактично apex class з набором змінних та функцій.

У 2015 році Salesforce запустив нову платформу, яка називається Lightning, і переніс весь функціонал на неї. Це зовсім інший підхід в архітектурі, так званий Event Driven Architecture.

Для розробки власних UI компонентів використовується Lightning Component framework. Там теж є набір вбудованих бібліотек/тегів для написання UI, які пропонує Salesforce, а також підтримка стандартних веб-технологій.

Для обробки івентів всередині компонентів використовується Client-Side контроллер, написаний на JavaScript.

Для роботи з базою даних, написання бізнес-логіки використовується Server-Side контроллер – apex клас, методи якого викликаються з Client-Side контроллера [23].

Salesforce має багато переваг, але разом з тим можна виділити низку недоліків:

- Salesforce має досить багато лімітів, до прикладу, за одну транзакцію можна зробити не більше 150 DML операцій, максимальний heap size – 6 MB для синхронних і 12 MB – для асинхронних процесів;
- поганий механізм для дебагу коду, для цього потрібно включити `System.Debug('message')` в свій код. Salesforce видає лог-файл, де можна побачити свої дебаг логи;
- статичні змінні живуть тільки в межах операції. Apex-змінна живе від початку до кінця реквесту, коли реквест закінчується – змінна обнуляється;
- деплоймент досить часто вимагає дій вручну, конфігураційну роботу, яку не можна автоматизувати, потрібно робити вручну після деплойменту.

3.3 Висновки за розділом

CRM-системи отримали стрімкий розвиток за останні декілька років у середовищі обробки бізнес процесів. Такі системи орієнтовані на обробку даних клієнтів та в цілому на полегшення роботи з клієнтами. Зазвичай, CRM-системи дають можливість візуалізувати та опрацьовувати інформацію клієнтів, а також організовувати комфортну та злагоджену роботу окремих підрозділів підприємства.

Salesforce.com – компанія, що підтримує власну однойменну CRM-платформу, що є лідером у своїй галузі. Ця платформа є основою для широкого загалу користувачів та компаній і повсякчас спонукає їх до співробітництва, оскільки має доволі низький поріг входу та широку базу для самостійного навчання. Salesforce має власну мережу розповсюдження додатків, що зветься AppExchange, тому кожен бажаючий користувач має можливість поділитись своєю розробкою для загалу.

РОЗДІЛ 4

HEROKU ТА ХМАРНІ ПЛАТФОРМИ

4.1 Опис 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 була створена Джеймсом Лінденбаумом, Адамом Уїгінсом та Оріон Генрі у 2007 році. У 2010 році компанія Salesforce.com придбала Heroku, зробивши її дочірньою компанією. У 2011 році було додано функції підтримки мови Ruby, Clojure, Node.js. у тому ж році було представлено функцію «Heroku for Facebook» - колаборація двох компаній для забезпечення Facebook необхідними мікросервісами. Також Heroku підтримує такі 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 зветься «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 проєкті, чи то звужуючи функціонал, чи то розширюючи.

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.

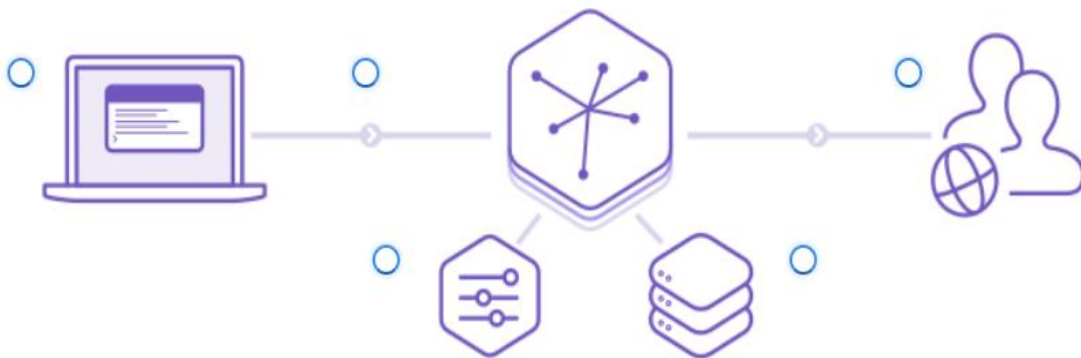


Рисунок 4.1 – Схема роботи сервісу Heroku

Процес роботи сервісу:

- користувач викладає код на платформу, за допомогою команд, що підтримуються сервісом;
- додаток запускається та працює на «дупо»-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[24].

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»[24].

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.



Рисунок 4.2 – Робота з OpEx, що демонструє завантаженість dyno-процесів та роботу запитів за хвилину

4.2 Висновки за розділом

Heroku – хостинг платформа, що є частиною компанії Salesforce.com з 2010 року. 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.

РОЗДІЛ 5

АРХІТЕКТУРА ТА РОЗРОБКА ДОДАТКУ

5.1 Ідея додатку

Програма 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. Код сценарію буде наведено у додатку А.

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.



Рисунок 5.1 – Схема додатку

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

5.2 Розміщення Python-скрипту на платформі Heroku

Перш за все, відбувається встановлення мовного середовища Heroku для поточної операційної системи. У даному випадку, це Windows x64. На відміну від UNIX-подібних систем, де установка відбувається через командну строку, для Windows встановлення відбувається через файл установки.

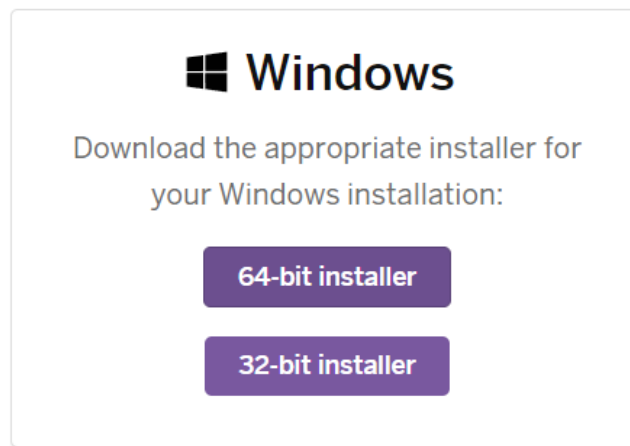


Рисунок 5.2 – Вікно завантаження Heroku CLI на офіційному сайті для Windows

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
```

Рисунок 5.3 – 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
```

Рисунок 5.4 – 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
```

Рисунок 5.5 – 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
```

Рисунок 5.6 – 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.

5.3 Інтеграція Heroku у Salesforce

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 PostgreSQL, 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 буде наведена у додатку Б.

5.4 Висновки за розділом

Реалізована архітектура системи здатна працювати незалежно від апаратної структури користувача, оскільки є хмарною.

Реалізація CRM-додатку 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 системи.

ВИСНОВКИ

У процесі дослідницької роботи було розроблено хмарний додаток, що обслуговує CRM-систему через методи Data Mining з використанням Python на віддаленій хостинг-платформі.

Переваги такого підходу до розробки додатку полягає у тому, що розроблена система не залежить від апаратних можливостей користувача та розміщена повністю на хмарних сервісах, що потребує лише доступу до Мережі.

Компоненти, що працюють у системі, мають доволі чітку інтегровану документацію, тому здатні поєднуватись за допомогою API мікросервісів, що забезпечує злагоджену та швидкісну роботу системи.

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.

CRM-система Salesforce є гарним середовищем не лише, як CRM, але і як середовище для реалізації розробника завдяки системі тренування та підвищення кваліфікації Trailhead та системі розповсюдження онлайн додатків AppExchange.

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 Salesforce.

Завдяки розробленій системі користувачі системи Salesforce можуть робити пошук раніше невідомих даних, зв'язків, закономірностей для більш детальної обробки інформації про клієнтів. Це матиме позитивний вплив на роботу як великих компаній, так і малих, покращуючи показники взаємодії

роботи з клієнтами та взаємодіями між робочими відділами та між робочими процесами.

Реалізована архітектура системи здатна працювати незалежно від апаратної структури користувача, оскільки є хмарною.

Реалізація CRM-додатку у взаємодії з Data Mining задачами здатна суттєво позитивно вплинути на роботу компанії з даними з клієнтської бази.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтелектуальний аналіз даних [Електронний ресурс]. – Режим доступу : <http://mirznanii.com/a/308854/ntelektualniy-analz-danikh>
2. Інтелектуальний аналіз даних: практикум / Фісун М. Т., Кравець І. О., Казмірчук П. П., Ніколенко С. Г.: «Новий Світ - 2000», 2020. – 162с.
3. Інтелектуальний аналіз даних [Електронний ресурс]. – Режим доступу : <https://univerfiles.com/1168907/Інтелектуальний-аналіз-даних/>
4. Марченко О.О., Россада Т.В. Актуальні проблеми Data Mining: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. — Київ. — 2017. — 150 с.
5. UpGrad Most Common Examples of Data Mining [Електронний ресурс] – Режим доступу : <https://www.upgrad.com/blog/most-common-examples-of-data-mining/>
6. Ершов, К. С. Анализ и классификация алгоритмов кластеризации. // Новые информационные технологии в автоматизированных системах. — 2016. — №19. — С. 274-279
7. UpGrad Data Mining Techniques: Types of Data, Methods, Applications [Електронний ресурс] – Режим доступу : <https://www.upgrad.com/blog/data-mining-techniques/>
8. UpGrad Classification in Data Mining Explained: Types, Classifiers & Applications 2021 [Електронний ресурс] – Режим доступу : <https://www.upgrad.com/blog/classification-in-data-mining/>
9. UpGrad Top 10 Most Common Data Mining Algorithms You Should Know [Електронний ресурс] – Режим доступу : https://www.upgrad.com/blog/common-data-mining-algorithms/#1_C45_Algorithm
10. 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/](#)

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

12. UpGrad Regression in Data Mining: Different Types of Regression Techniques [2021] [Електронний ресурс] – Режим доступу : <https://www.upgrad.com/blog/regression-in-data-mining/>

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

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

15. Python org About [Електронний ресурс] – Режим доступу : <https://www.python.org/about/>

16. Python org NumPy [Електронний ресурс] – Режим доступу : <https://wiki.python.org/moin/NumPy>

17. Python org Numeric and Scientific [Електронний ресурс] – Режим доступу : <https://wiki.python.org/moin/NumericAndScientific>

18. Уэс Маккинли Python и анализ данных/ Пер. с англ. Слинкин А. А. - М.: ДМК Пресс, 2015. - 482 с.: ил.

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

20. 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/>

21. 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

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

23. Salesforce Forum [Электронный ресурс] – Режим доступа :
<https://salesforce.in.ua/>
24. Heroku [Электронный ресурс] – Режим доступа :
<https://www.heroku.com/>

Додаток А

Зауваження нормоконтролера

Таблиця А.1 – Зауваження нормоконтролера

[illegible]

Додаток Б
Програмний код

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()
```

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()
```

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()
```

Сценарій для узгодження типізації даних з Salesforce на Python:

```
from simple_salesforce import Salesforce
sf = Salesforce(
    username='myemail@example.com',
    password='password',
    security_token='token')
"SELECT Owner.Name, store_id__c, account_number__c,
username__c, password__c, program_status__c, FROM Account
WHERE program_status__c IN ('Live','Test')"
sf_data = sf.query_all("SELECT Owner.Name, store_id__c,
account_number__c, username__c, password__c,
program_status__c, FROM Account WHERE program_status__c IN
('Live','Test')")
sf_df =
pd.DataFrame(sf_data['records']).drop(columns='attributes')
```

Додаток В

Результати роботи програми

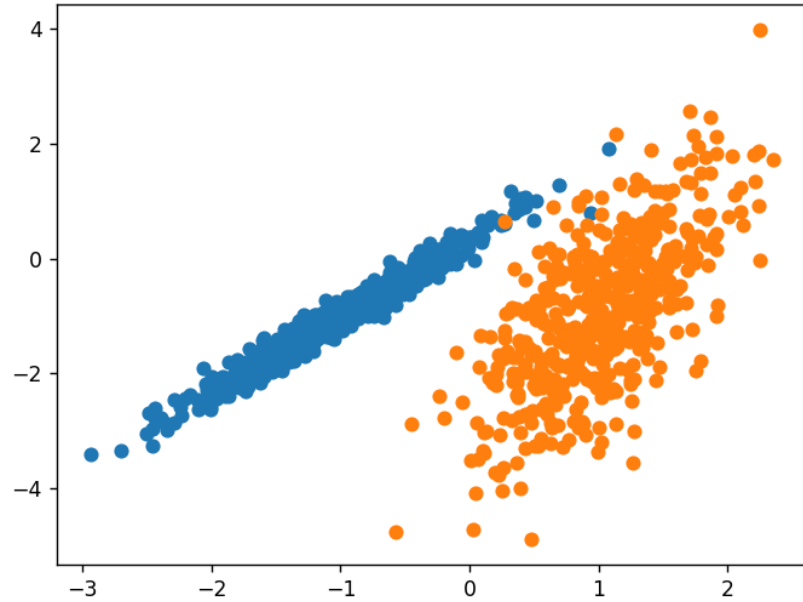


Рисунок 1 - Алгоритм кластеризації Бірча

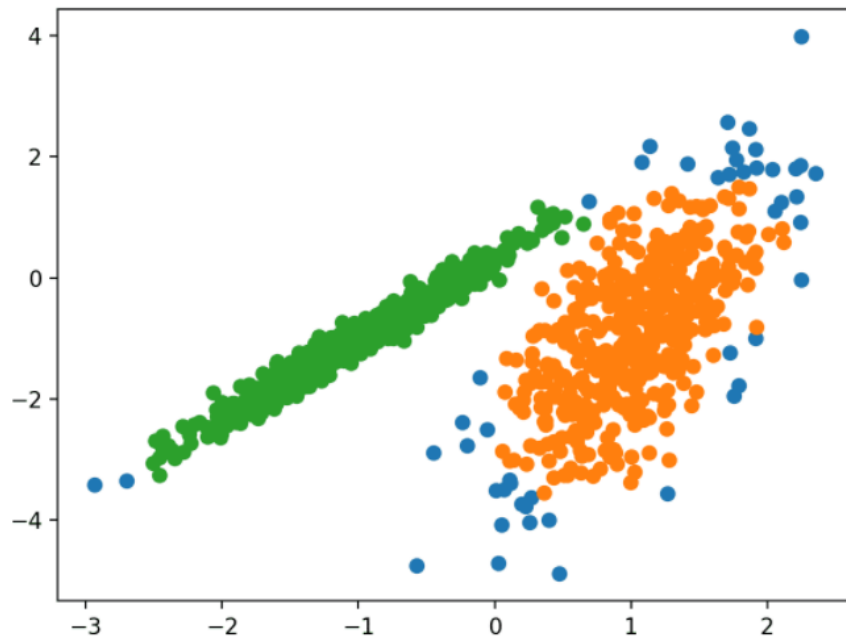


Рисунок 2 – Алгоритм DBscan

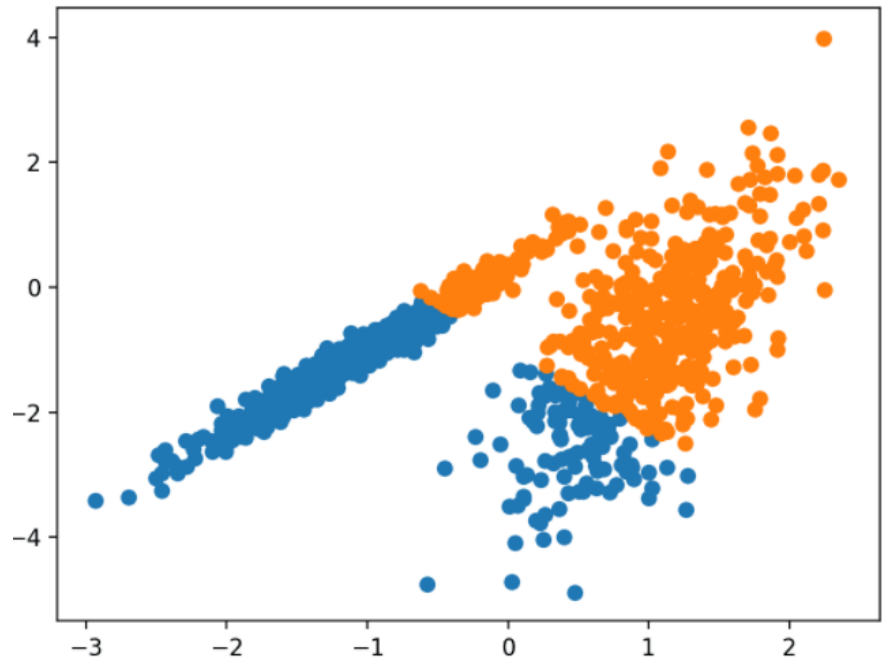










Рисунок 3 – Алгоритм k-means



All

Search Opportunities and more...



Data Mining App

Accounts

Contacts

Products








Opportunities

Opportunities

Recently Viewed

14 items • Updated 7 minutes ago

Search this list...



	Opportunity Name	Account Name	Account Site	Stage	Close Date	Opportunity O...
1	<input type="checkbox"/> Developers Course			Prospecting	06/11/2021	YLepe
2	<input type="checkbox"/> Some Courses Opp			Prospecting	06/10/2021	YLepe
3	<input type="checkbox"/> Express Logistics SLA	Express Logistics and Transport		Perception Analysis	09/03/2021	YLepe
4	<input type="checkbox"/> Edge SLA	Edge Communications		Closed Won	28/02/2021	YLepe
5	<input type="checkbox"/> GenePoint SLA	GenePoint		Closed Won	10/06/2021	YLepe
6	<input type="checkbox"/> GenePoint Lab Generators	GenePoint		Id. Decision Makers	07/06/2021	YLepe
7	<input type="checkbox"/> Express Logistics Portable Truck Generators	Express Logistics and Transport		Value Proposition	08/03/2021	YLepe
8	<input type="checkbox"/> GenePoint Standby Generator	GenePoint		Closed Won	17/04/2021	YLepe
9	<input type="checkbox"/> Express Logistics Standby Generator	Express Logistics and Transport		Closed Won	10/03/2021	YLepe
10	<input type="checkbox"/> Edge Emergency Generator	Edge Communications		Id. Decision Makers	19/06/2021	YLepe
11	<input type="checkbox"/> Burlington Textiles Weaving Plant Generator	Burlington Textiles Corp of America		Closed Won	19/04/2021	YLepe
12	<input type="checkbox"/> Edge Installation	Edge Communications		Closed Won	04/04/2021	YLepe
13	<input type="checkbox"/> Edge Emergency Generator	Edge Communications		Closed Won	13/06/2021	YLepe
14	<input type="checkbox"/> Dickenson Mobile Generators	Dickenson plc		Qualification	26/04/2021	YLepe

Рисунок 4 – Представлення таблиці даних для обробки на Salesforce


Opportunity Owner	Amount
 Yaroslav Lepetchenko	€80,000.00
Private	Expected Revenue
<input type="checkbox"/>	€40,000.00
Opportunity Name	Close Date
Express Logistics Portable Truck Generators	08/03/2021
Account Name	Next Step
Express Logistics and Transport	
Type	Stage
Existing Customer - Upgrade	Value Proposition
Lead Source	Probability (%)
External Referral	50%
Participant Quantity	Primary Campaign Source
Order Number	Main Competitor(s)
	Honda
Current Generator(s)	Delivery/Installation Status
Fujitsu	Yet to begin
Tracking Number	

Рисунок 5 – Запис у представленні даних Salesforce

Додаток Г
Роздатковий матеріал

ТЕМА МАГІСТЕРСЬКОЇ РОБОТИ:
«АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ
ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ
ДАНИХ»

Виконав:
ст. гр. ПЗМ-20
Лепетченко Я. В.

Керівник:
к.т.н., доц. каф.
Назарова І. А.

МЕТА: Дослідити методи
інтелектуального аналізу даних у
різних галузях життєдіяльності

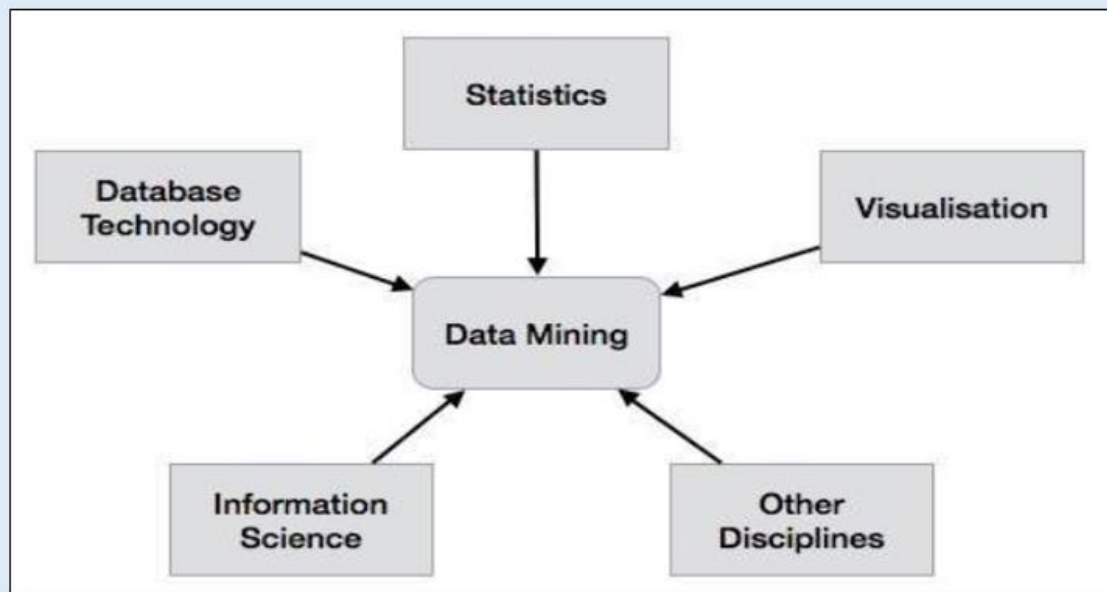
Предметом дослідження є
машинне навчання та Data mining
та їх застосування на практиці.

Все частіше для збору та аналізу великих обсягів даних використовують інтелектуальний аналіз даних, що також відомий, як Data mining (розкопка/добування даних).

Data mining – процес виявлення в сирих, раніше невідомих, нетривіальних, практично корисних і доступних для інтерпретації знань, необхідних для прийняття рішень у різних сферах людської діяльності.



Data Mining – мультидисциплінарна область





CRM-платформа Salesforce – хмарна SaaS-платформа, що використовується для автоматизації бізнес-процесів, більш зручної взаємодії компанії та клієнтської бази, а також взаємодії між підрозділами компанії.

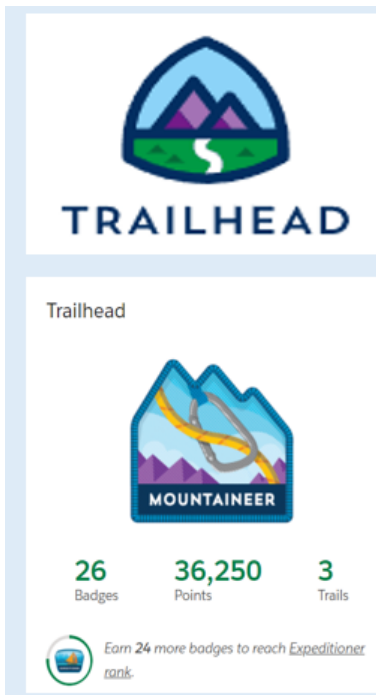
Також, Salesforce має доволі зручні можливості для інтегрування нових програмних модулів, що здатні виконувати специфічні задачі, не передбачені платформою.



Платформа має широке ком'юніті та можливості розширюватися за його рахунок.

AppExchange – маркетплейс на платформі Salesforce, де користувачі платформи можуть ділитися власними розробками.

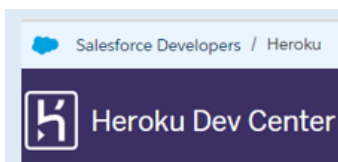




Також, спеціально для Salesforce створено середовище для підготовки та сертифікації користувачів. Ця навчальна методика називається Trailhead та має інтерактивні засоби підготовки розробників.

За проходження кожного блоку підготовлених завдань користувач отримує бали та «значки», що відображають кількість та якість пройдених завдань.

Навчання відбувається на певних кейсах застосування платформи, що описані в коротких та інформативних статтях, після яких користувач проходить квіз-тест або виконує прикладне завдання.



Компанія Salesforce.com також володіє хостинг-платформою Heroku. Heroku – хмарна PaaS платформа, що здатна розміщувати бази даних (PostgreSQL, MongoDB тощо), а також підтримує середовище таких мов, як Java, Ruby, Go, JavaScript, Python та інші.

Ця платформа дозволяє досить просто взаємодіяти різним програмним послугам, що можуть бути просто інтегровані за допомогою Heroku. У тому числі можна реалізувати інтеграцію програмного коду з платформою Salesforce.



Мова програмування Python – одна з найкращих для реалізації рішень Data Mining. Це можливе завдяки пакетам та бібліотекам, що підтримуються у актуальному стані завдяки підтримці з боку ком'юніті.

Бібліотеки NumPy, SciPy, Matplotlib, Ipython та інші мають достатньо можливостей для математичних, статистичних операцій та візуалізації даних.

Мовне середовище Python здатне реалізовувати різні архітектурні рішення завдяки взаємодії з Інтернет-протоколами, а також інтеграції з хостинг-платформами, такими, як Heroku.

Використовуючи всі вищеприведені аспекти науки та технологій, відкривається можливість створення потенційно-корисного додатку для користувачів платформи Salesforce для інтелектуального аналізу даних.

Це рішення здатне вирішити питання та задачі, що виникають при роботі з даними клієнтів та продажів, а також покращити роботу компаній.