МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ І СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти	Другий (магістерський)
Спеціальність	Економіка
Освітня програма	Економічна кібернетика
Група	8.04.051.020.22.1

ДИПЛОМНА РОБОТА

на тему: «Моделювання споживацької поведінки за допомогою методів машинного навчання»

Виконала: студент Артем МИРОНЕНКО

Керівник: к.е.н., доцент Любов ЧАГОВЕЦЬ

Рецензент: к.е.н., доцент
Кафедри кібербезпеки НТУ «ХПІ» Станіслав МІЛЕВСЬКИЙ

Харків – 2023 рік

РЕФЕРАТ

Звіт про дипломну роботу: 70 сторінок, 3 розділи, 17 рисунків, 24 джерела.

Об’єктом роботи є процеси поведінки користувачів. Предметом дослідження є математичні методи та моделі оцінки споживацької поведінки.

Мета дипломної роботи – розробити комлекс моделей ідентифікації та прогнозування стану поведінки споживачів поведінки на базі методів машинного навчання, що дозволить оцінити споживацькі уподобання та підвищити якість управлінських рішень щодо маркетингових стратегій просування продукту.

Для досягення поставленої мети було розроблено та виконано такі завдання: проаналізувати особливості поведінки споживача; проаналізувати перспективи та особливості машинного навчання для моделювання поведінки споживачів; здійснити аналіз інструментальних засобів машинного навчання, зокрема, мови Python, робочого середовища Jupyter notebook, Хмарного сховища Snowflake та мови SQL до моделювання споживацької поведінки; розробити моделі прогнозування стану поведінки споживачів за індикаторами конверсії методами машинного навчання. Дипломна робота присвячена розробці моделей класифікації майбутньої поведінки споживача з використанням машинного навчання. Було детально розглянуто моделі бінарної класифікації. Було створено сім моделей бінарної класифікації за цими даними та для найкращої моделі були підібрані оптимальні параметри для збільшення точності моделі.

КЛЮЧОВІ СЛОВА: ПОВЕДІНКА СПОЖИВАЧІВ, МОДЕЛЮВАННЯ, МОДЕЛЬ, МАШИННЕ НАВЧАННЯ, БІНАРНА КЛАСИФІКАЦІЯ, ДАНІ, ГРАДІЄНТНИЙ БУСТІНГ, КЛАСИФІКАЦІЯ КОРИСТУВАЧІВ, PYTHON, JUPYTER NOTEBOOK.

ABSTRACT

Thesis report: 70 pages, 3 chapters, 17 drawings, 24 sources.

The object of the work is user behavior processes.

The subject of the research is mathematical methods and models for evaluating consumer behavior.

The aim of the thesis is to develop a complex of models for identifying and predicting the state of consumer behavior based on machine learning methods, which will allow assessing consumer preferences and improving the quality of management decisions regarding marketing strategies for product promotion.

The following tasks were developed and performed to achieve the set goal: analyze the features of consumer behavior; analyze the prospects and features of machine learning for modeling consumer behavior; analyze machine learning tools, in particular, the Python language, the Jupyter notebook working environment, the Snowflake cloud storage, and the SQL language for modeling consumer behavior; develop machine learning models for predicting the state of consumer behavior based on conversion indicators.The thesis is devoted to the development of models for classifying future consumer behavior using machine learning.

The work is based on the company's own data. The work examined binary classification models in detail. Seven binary classification models were created using this data, and for the best model, the optimal parameters were selected to increase the model's accuracy.

KEYWORDS: CONSUMER BEHAVIOR, MODELING, MACHINE LEARNING, BINARY CLASSIFICATION, DATA, GRADIENT BOOSTING, USER CLASSIFICATION, PYTHON, JUPYTER NOTEBOOK.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

МН – машинне навчання;

БД – база даних

ШІ– штучний інтелект;

SQL – структурний язик запитів;

DF – таблична структура даних в Python.

PD – бібліотека pandas

NP – бібліотека NumPy

RPA (Robotic Process Automation) – автоматизація процесів за допомогою роботизації

ВСТУП

У сучасному інформаційному суспільстві важко переоцінити роль споживача в економічних та соціальних процесах. Розуміння його поведінки є ключовим фактором для підприємств, дослідників та маркетологів у визначенні стратегій розвитку та спрямованості ринкових пропозицій. Однак, у зв'язку з розмаїттям факторів, що впливають на споживача, та швидкими змінами у вимогах ринку, стає надзвичайно важливим використання новітніх технологій для розуміння та прогнозування його поведінки.

Дипломна робота присвячена вивченню та моделюванню поведінки споживача на основі методів машинного навчання.

Об’єктом роботи є процеси поведінки користувачів.

Предметом дослідження є математичні методи та моделі оцінки споживацької поведінки.

Мета дипломної роботи – розробити комлекс моделей ідентифікації та прогнозування стану поведінки споживачів поведінки на базі методів машинного навчання, що дозволить оцінити споживацькі уподобання та підвищити якість управлінських рішень щодо маркетингових стратегій просування продукту.

Для досягення поставленої мети було розроблено та виконано такі завдання:

Проаналізувати особливості поведінки споживача

Проаналізувати перспективи та особливості машинного навчання для моделювання поведінки споживачів

Здійснити аналіз інструментальних засобів машинного навчання, зокрема, мови Python, робочого середовища Jupyter notebook, Хмарного сховища Snowflake та мови SQL до моделювання споживацької поведінки

Розробити моделі прогнозування стану поведінки споживачів за індикаторами конверсії.

Машинне навчання є потужним інструментом аналізу та прогнозування великого обсягу даних, що дозволяє автоматизувати процеси виявлення закономірностей та взаємозв'язків. Застосування цих методів у вивченні поведінки споживача відкриває нові можливості для глибокого розуміння його вподобань, потреб та реакцій на ринкові зміни. Одним із головних аспектів дослідження є визначення ключових факторів, що впливають на рішення споживача в процесі вибору товарів чи послуг. Машинне навчання дозволяє аналізувати та інтегрувати різноманітні дані, включаючи соціальні мережі, онлайн-покупки, та інші джерела, для створення комплексної картини взаємодії споживача з ринком.

Обрана тема важлива не лише для наукового співтовариства, але й для практиків у галузі маркетингу та управління бізнесом. Наші дослідження спрямовані на розробку ефективних стратегій взаємодії зі споживачем, що базуються на точних та передбачуваних моделях його поведінки. В цьому контексті, вивчення моделей машинного навчання для аналізу споживчої поведінки може внести значний внесок у розвиток сучасного маркетингового підходу та сприяти покращенню стратегій управління підприємствами.

Ця робота буде спрямована на дослідження та висвітлення можливостей застосування методів машинного навчання для моделювання поведінки споживача, а також на розробку практичних рекомендацій для бізнес-середовища.

РОЗДІЛ 1. РОЛЬ ТА МОЖЛИВОСТІ МАШИННОГО НАВЧАННЯ

1.1. Теоретичні засади моделювання поведінки користувача

У світі, де технології швидко розвиваються, а взаємодія людини з інформаційним середовищем стає все більше високоінтегрованою, розуміння та прогнозування поведінки користувачів є важливим аспектом для багатьох галузей, починаючи від маркетингу та закінчуючи розвитком продуктів і послуг. Цей розділ розглядає значення моделювання поведінки користувача в сучасному світі, висвітлює перспективи використання машинного навчання для досягнення цієї мети та розглядає його застосування в різних сферах, включаючи щоденне життя та бізнес.

Моделювання поведінки користувача є процесом створення абстрактної репрезентації реального поведінкового взаємодії із системою. Цей підхід дозволяє аналізувати та передбачати, як користувач взаємодіє з інформацією, продуктами чи сервісами. Вірно побудовані моделі дозволяють отримати глибше розуміння потреб, вподобань та мотивацій користувачів [14].

Моделювання поведінки користувача – це важливий аспект в сучасному інформаційному суспільстві. Перехід від традиційних методів до цифрових технологій вигадав нові способи вивчення та аналізу взаємодії людей з інформаційним середовищем. У цьому рефераті розглянемо важливість моделювання поведінки користувача, методи його впровадження та практичні застосування [15].

Значення моделювання поведінки користувача. Поведінка користувача є ключовим фактором для розуміння та покращення користувацького досвіду в різних сферах [12]. Моделювання цієї поведінки дозволяє прогнозувати вибір користувача, його реакції на певні події та взаємодію з різними інтерфейсами. За допомогою моделювання можна аналізувати, як користувачі сприймають нові технології та як їх використовують у реальному житті.

Методи моделювання поведінки користувача. Статистичні методи: Один з підходів - використання статистичних методів для аналізу великих обсягів даних. Методи машинного навчання, такі як класифікація та кластеризація, дозволяють виділити закономірності в поведінці користувачів на основі зібраних даних [13].

Психологічні методи: Застосування психологічних методів дозволяє враховувати особливості сприйняття інформації, мотивації та інших психологічних факторів [14]. Використання психометричних тестів та анкет може допомогти зрозуміти індивідуальні особливості користувачів.

Застосування моделей поведінки користувача. Інтернет-магазини: Моделювання поведінки користувача в інтернет-магазинах дозволяє покращити персоналізацію рекомендацій та оптимізувати процес покупок, забезпечуючи зручний та ефективний досвід користувача.

Освіта: В галузі освіти моделювання поведінки користувача може бути використане для індивідуалізації навчальних матеріалів, прогнозування успішності студентів та підтримки адаптивного навчання.

Виклики та майбутні напрямки розвитку. Використання моделей поведінки користувача також пов'язане з викликами, такими як забезпечення конфіденційності даних та етичні аспекти збору інформації. Майбутні напрямки розвитку включають вдосконалення методів штучного інтелекту, адаптацію до змінних потреб користувачів та розвиток нових методів збору даних.

Розширення методів моделювання поведінки користувача. Текстова та семантична аналітика: Використання аналізу текстів та семантичної обробки може допомогти виявити суттєві слова та поняття, які впливають на користувацьке рішення. Це особливо корисно в онлайн-середовищах, де аналіз коментарів та відгуків може призвести до кращого розуміння потреб користувачів.

Врахування контексту: Забезпечення контекстуального розуміння дій користувача є ключовим для точного моделювання. Інформація про час, місце,

пристрій та інші параметри може значно впливати на рішення користувача, тому включення цих аспектів у моделі є важливим кроком для покращення точності.

Застосування в медицині та здоров'ї: Моделі поведінки користувача можуть знайти застосування в медичних системах, де вони можуть допомагати в моніторингу здоров'я, нагадуванні про ліки та адаптації інтерфейсів для різних медичних станів користувачів.

Вивчення поведінки використовуючи різні платформи: Врахування користувацької взаємодії на різних платформах, таких як телефони, персональні компьютери або планшети дозволяє побудувати комплексні моделі, що враховують специфіку використання та відмінності в користувацькому досвіді.

Аналіз віртуальної та розширеної реальності: врахування специфіки взаємодії в середовищах віртуальної та розширеної реальності дозволяє розробляти більш ефективні інтерфейси для користувачів, а також прогнозувати їхню реакцію на віртуальні події та об'єкти.

Інтеграція з іншими системами: поєднання моделей поведінки користувача з іншими інформаційними системами (наприклад, системами управління взаємодією з клієнтами або системами аналітики) може допомагати компаніям зрозуміти потреби своїх користувачів на більш глибокому рівні та підтримувати індивідуальний підхід до кожного користувача.

Екологічна сталість: застосування моделей поведінки користувача для аналізу та вдосконалення споживання ресурсів, таких як електроенергія та вода, може сприяти створенню екологічно сталіших технологій та послуг.

Моделювання поведінки користувача - це динамічне поле досліджень, яке постійно розширюється. Розглядаючи нові напрями, такі як текстовий аналіз, врахування контексту, медичні застосування та інші, можна домогтися більш точних та комплексних моделей, що сприяють розвитку більш ефективних та інноваційних продуктів та послуг.

Моделювання поведінки користувача – це ключовий інструмент для розуміння та прогнозування взаємодії людей з інформаційним середовищем. Розглядаючи значення цього підходу, методи його реалізації та практичні застосування, можна визначити його великий вплив на різні галузі, від електронної коммерції до освіти. Однак, використання цих моделей пов'язане з етичними викликами, які вимагають уважного вирішення для забезпечення захисту приватності та справедливого використання отриманих даних [15].

1.2. Перспективи та особливості застосування методів машинного навчання

З моменту, коли машинне навчання стало невід'ємною частиною аналізу даних, його застосування в галузі моделювання поведінки користувача набуло великого інтересу. Алгоритми машинного навчання, такі як класифікація, кластеризація та регресія, дозволяють автоматизувати процеси визначення патернів у поведінці користувачів на основі великого обсягу даних.

В останні роки машинне навчання стало визначальною силою, що змінює ландшафт технологій і відкриває безліч нових можливостей в різних сферах. Застосування машинного навчання не обмежується лише інформаційними технологіями; воно проникає в медицину, науку, бізнес, виробництво та багато інших галузей. Давайте розглянемо деякі ключові перспективи застосування машинного навчання, які революціонізують наше розуміння і використання технологій.

Медицина та діагностика: Машинне навчання в сфері медицини відкриває нові можливості для точної діагностики та індивідуалізованого лікування. Навіть медичні зображення можуть бути проаналізовані алгоритмами МН, визначати патології на рентгенівських знімках та допомагати у виявленні характеристик різних захворювань.

Фінанси та прогнозування ринку: Машинне навчання стало невід'ємною частиною фінансових інструментів, які використовуються для прогнозування ринкових тенденцій, управління портфелем та виявлення фінансових аномалій. Алгоритми можуть аналізувати величезні обсяги даних та вчитися виявляти патерни, що раніше залишалися непоміченими.

Автономні транспортні засоби: В розробці автономних автомобілів машинне навчання грає ключову роль в створенні систем, які можуть навчитися розпізнавати дорожні умови, інші транспортні засоби та адаптуватися до них. Це сприяє покращенню безпеки на дорогах та ефективності транспортних систем.

Виробництво та якість: у сфері виробництва машинне навчання використовується для оптимізації процесів виробництва, передбачення збоїв обладнання та покращення якості продукції. Інтелектуальні системи можуть виявляти аномалії в роботі обладнання та уникати непередбачених збоїв.

Навчання та освіта: В сфері освіти застосування машинного навчання означає розвиток індивідуалізованих підходів до навчання, створення програм, які можуть адаптуватися до потреб студентів, та навчальних ігор, що покращують ефективність навчання.

Захист інформації та кібербезпека: Алгоритми машинного навчання використовуються для виявлення та протидії кібератакам, а також для аналізу великих обсягів даних з метою виявлення потенційних загроз безпеці.

Сфера соціальних мереж: Машинне навчання в соціальних мережах використовується для аналізу поведінки користувачів, рекомендацій, персоналізації контенту та виявлення несанкціонованих дій.

Галузь розваг та створення контенту: У розробці відеоігор та кіноіндустрії машинне навчання використовується для створення реалістичних персонажів, оптимізації графіки та персоналізації вза ємодії з глядачем.

Загальна тенденція полягає в тому, що машинне навчання не тільки впроваджується в існуючі галузі, але й створює зовсім нові можливості.

Технологічний прогрес в цьому напрямку поки що тільки почався, і майбутні перспективи є надзвичайно захоплюючими.

Машинне навчання та Big Data. В сучасному світі, де величезні обсяги даних стають нормою, поєднання технологій машинного навчання та великих даних (Big Data) визначає новий етап в розвитку інформаційних технологій та впливає на всі сфери діяльності суспільства. Це поєднання не тільки забезпечує аналіз та інтерпретацію великих обсягів інформації, але й стає критичним фактором для досягнення стратегічного конкурентного переваги в бізнесі.

Машинне навчання (МН) — це галузь штучного інтелекту, що дозволяє комп'ютерам вчитися та приймати рішення без явно заданої програми. Ця технологія допомагає системам адаптуватися до нових вхідних даних, навчаючись самостійно і покращуючи свою продуктивність в часі. У поєднанні з великими обсягами даних, машинне навчання може виводити аналітичні здібності на новий рівень, дозволяючи виявляти складні залежності та зробити передбачення на основі цих аналізів.

З іншого боку, поняття Big Data об'єднує в собі величезні, різноманітні та швидко зростаючі обсяги даних, які важко або неможливо обробити за допомогою традиційних методів обробки даних. Big Data виникає з різних джерел, таких як сенсори в реальному часі, соціальні мережі, транзакції в мережі, інтернет речей тощо. Ці дані стають цінним ресурсом для виявлення закономірностей, трендів та можливостей для оптимізації бізнес-процесів.

Поєднання машинного навчання та Big Data відкриває безліч можливостей у різних сферах, включаючи бізнес, науку, медицину та інші. У сфері бізнесу ця симбіозна технологічна парадигма розкривається у здатності ефективно використовувати величезні обсяги даних для прийняття стратегічних рішень та оптимізації бізнес-процесів.

Машинне навчання в бізнесі дозволяє розробляти передбачувані моделі для виявлення та реагування на зміни в динаміці ринку, споживацьких уподобаннях та кон'юнктурі [15]. Аналізуючи великі обсяги даних, бізнес

може точно прогнозувати попит, оптимізувати ціноутворення та рекламні кампанії, а також виявляти нові можливості для розвитку продуктів чи послуг.

Управління ризиками в бізнесі також отримує значний приріст завдяки цій сполученій технологічній силі. Машинне навчання аналізує великі обсяги даних для виявлення та прогнозування ризиків, що допомагає компаніям вчасно реагувати на зміни у внутрішньому та зовнішньому середовищі. Окрім того, використання машинного навчання в поєднанні з Big Data розширює можливості в області клієнтського обслуговування. Аналізуючи великі обсяги даних про споживачів, компанії можуть створювати персоналізовані підходи, пропонуючи індивідуалізовані товари та послуги, що сприяє збільшенню лояльності клієнтів [15].

Наукові та технічні досягнення в області машинного навчання та Big Data нестримно розвиваються, визначаючи нові перспективи для покращення бізнес-процесів та забезпечення стійкості та конкурентоспроможності компаній в умовах постійних змін. Розвиток цих технологій є необхідним кроком для тих, хто прагне залишатися на передових позиціях у світі високих технологій. На сучасному етапі розвитку технологій, поєднання машинного навчання та Big Data стає каталізатором інновацій, прискорюючи темпи трансформації бізнес-процесів та управлінських практик. Однією з важливих сфер застосування цього симбіозу є підвищення ефективності операцій та прийняття стратегічних рішень. Машинне навчання, оперуючи великими обсягами даних, вдосконалює прогностичні моделі, надаючи бізнесу можливість швидше адаптуватися до змін у внутрішньому та зовнішньому середовищі. Системи прогнозування, побудовані на алгоритмах машинного навчання, дозволяють ефективно передбачати тенденції ринку та вчасно реагувати на них. Це стає ключовим елементом стратегічного управління, надаючи компаніям конкурентну перевагу. Великі обсяги даних, що постійно надходять в режимі реального часу, відкривають нові перспективи для сфери маркетингу. Машинне навчання допомагає аналізувати та класифікувати споживацькі поведінки, що робить можливим персоналізований підхід до

кожного клієнта. Здатність розпізнавати патерни та взаємозв'язки великих обсягів даних дає змогу оптимізувати рекламні кампанії та пристосовувати їх до унікальних потреб кожного споживача.

Динамічний розвиток інтернету речей (IoT) додав великої кількості даних, що генеруються пристроями, підключеними до мережі. Машинне навчання в цьому контексті використовується для аналізу та інтерпретації величезних потоків інформації в режимі реального часу. Застосування цих технологій у виробництві дозволяє оптимізувати процеси, підвищуючи ефективність та знижуючи витрати. Невід'ємною частиною ефективного управління великими обсягами даних є їх безпека та конфіденційність. Машинне навчання в поєднанні з Big Data розробляє та вдосконалює алгоритми кіберзахисту, що важливо в умовах зростаючого обсягу кіберзагроз.

Одним із перспективних напрямків використання машинного навчання та Big Data є розробка інтелектуальних систем управління, які здатні адаптуватися до змін у середовищі та приймати стратегічні рішення на основі аналізу великих обсягів структурованих і неструктурованих даних. Можна сказати що поєднання машинного навчання та Big Data визначає новий етап в еволюції технологій, створюючи потужний інструментарій для оптимізації бізнес-процесів та прийняття стратегічних рішень. Ця симбіозна взаємодія стає ключовим фактором конкурентоспроможності в умовах глобальних технологічних трансформацій, відкриваючи широкі можливості для розвитку та інновацій у різних секторах економіки.

Сучасний вимір інновацій та перспективи розвитку. На сучасному етапі цифрової революції, синергія машинного навчання та Big Data перетворюється в країну безмежних можливостей для бізнесу та науки. Ця динамічна взаємодія технологій визначає не тільки сучасні тренди, але і визначає стратегічний напрям розвитку компаній у майбутньому. Ще однією сферою, де спільне використання машинного навчання та Big Data виявляється особливо ефективним, є науково-дослідна діяльність. Аналіз великих обсягів даних

дозволяє вченим виявляти нові тенденції та залежності в різних галузях знань. Машинне навчання допомагає у створенні прогностичних моделей, що має значення у багатьох наукових областях, від медицини до екології.

Основною перспективою є вдосконалення алгоритмів машинного навчання для роботи з великими обсягами даних. За допомогою глибокого навчання та нейронних мереж можна досягти ще вищого рівня точності в передбаченнях та аналізі. Такі техніки дозволяють ефективніше виявляти приховані закономірності в даних та надають можливість швидко реагувати на зміни в навколишньому середовищі.

Збільшення кількості джерел даних та їх структурна різноманітність відкривають нові горизонти для застосування машинного навчання та Big Data. Зокрема, розширення використання навчання з наставником або без нього робить можливим аналіз не лише кількісних, але і якісних параметрів даних. На перехресті цих двох сфер технологій виникає новий підхід до роботи з клієнтами. Розуміння їх потреб та попередження можливих відхилень дозволяє підприємствам будувати довгострокові відносини та підвищує рівень задоволення клієнтів. У світлі швидкого розвитку машинного навчання та Big Data, необхідно активно пристосовувати бізнес-стратегії та вивчати нові можливості, які ці технології відкривають. Підприємства, які успішно інтегрують ці інновації, зможуть не лише пристосовуватися до змін, а й визначати їх, виходячи на новий рівень конкурентоспроможності.

Усе враховуючи, поєднання машинного навчання та Big Data стає тим інтелектуальним рушієм, яке визначає та перетворює сучасний бізнес-ландшафт. Розуміння величезного потенціалу цих технологій дозволяє підприємствам не лише адаптуватися до викликів, а й активно керувати їхнім майбутнім, створюючи нові горизонти можливостей та досягаючи неперевершених висот у світі бізнесу.

1.3. Використання машинного навчання в сфері бізнесу

У бізнесі машинне навчання має потенціал перетворити стратегії маркетингу, прогнозування продажів, управління клієнтськими взаємодіями та багато інших аспектів. Застосування цих технологій дозволяє бізнес-процесам стати більш ефективними та адаптивними до змін на ринку.
Завдяки аналізу цих аспектів, цей розділ розкриває значущість моделювання поведінки користувача та підкреслює потужні можливості машинного навчання для вирішення цього завдання.

В сучасному бізнес-середовищі динамічність, нестабільність ринку та швидке зростання обсягів даних вимагають нових підходів до управління та прийняття стратегічних рішень. В цьому контексті використання машинного навчання стає важливим інструментом для бізнесу, дозволяючи ефективно аналізувати та використовувати великі обсяги даних для прийняття обґрунтованих рішень. Машинне навчання в сфері бізнесу знаходить широке застосування в різних напрямках. Одним із ключових аспектів є аналіз ринкових тенденцій та прогнозування попиту на продукцію чи послуги. Алгоритми машинного навчання, використовуючи дані про попередні продажі, поведінку клієнтів та інші фактори, дозволяють бізнесу адаптуватися до змін в ринковому середовищі. Ще однією суттєвою областю використання є підвищення ефективності маркетингових стратегій. Машинне навчання допомагає персоналізувати рекламні кампанії, прогнозувати індивідуальні вподобання споживачів та оптимізувати розподіл рекламних бюджетів.

Управління ризиками є іншим важливим аспектом використання машинного навчання в бізнесі. Аналізуючи великі обсяги фінансових даних та враховуючи різноманітні чинники, системи машинного навчання допомагають у прогнозуванні ризиків та прийнятті ефективних стратегій управління ними. В сфері виробництва та логістики використання машинного

навчання призводить до оптимізації ланцюга постачання, прогнозування попиту на продукцію та автоматизації процесів управління складом.

Особливу увагу слід приділити інноваційним підходам до обслуговування клієнтів. Використання машинного навчання дозволяє розробляти персоналізовані сервіси, швидко реагувати на зміни в клієнтських потребах та підвищує рівень задоволеності споживачів. Машинне навчання стає ключовим інструментом у впровадженні концепції "бізнес 4.0", де цифрові технології та аналітика взаємодіють для створення більш гнучких та адаптивних бізнес-моделей. За допомогою машинного навчання підприємства можуть швидше реагувати на зміни у внутрішньому та зовнішньому середовищі, роблячи їх більш конкурентоспроможними.

Узагальнюючи вищесказане, використання машинного навчання в бізнесі не тільки забезпечує ефективне використання даних, але і дозволяє підприємствам побудувати стратегії на основі об'єктивних аналізів та прогнозів. Завдяки використанню RPA можливо значно вдосконалити і автоматизувати процеси, які раніше вимагали великої кількості людських ресурсів. Однак, варто визнати, що не дивлячись на усі переваги роботизації, технологія має свої обмеження у певних сценаріях. Зокрема, виявляються труднощі у випадках, коли процеси протікають за нетривіальними сценаріями, вимагають прийняття рішень або проведення аналізу.

Звісно чи обирати традиційну роботизацію бізнес-процесів чи в рази ефективнішу, дешевшу в перспективі та сучасною роботизацію бізнес-процесів за технологіями МН залежить від ваших міркувань і можливостей, аде слід зазначити, що обидві опції однозначно можуть підвищити оптимізацію бізнесу. Проте на мою думку перевага все ж таки вддається останньому варіанту. «Машинне навчання, зокрема технологія роботизації бізнес-процесів (RPA), відіграє значущу роль у трансформації бізнес-процесів». – Лілія Каневська підкреслює величезний потенціал RPA у підвищенні операційної продуктивності та зниженні операційних витрат в організаціях. Технологія виявляється особливо ефективною в оптимізації

рутинних та повторюваних щоденних процесів [26]. При поєднанні машинного навчання і роботизації та людського досвіду невпинно призводить до розвитку бізнесу таких компаній як Tesla, Google, BMW і багато інших. На рис. 1.1 зображено відносну частку великих компаній, що виділяють ШІ, як ключовий пріоритет розвитку.

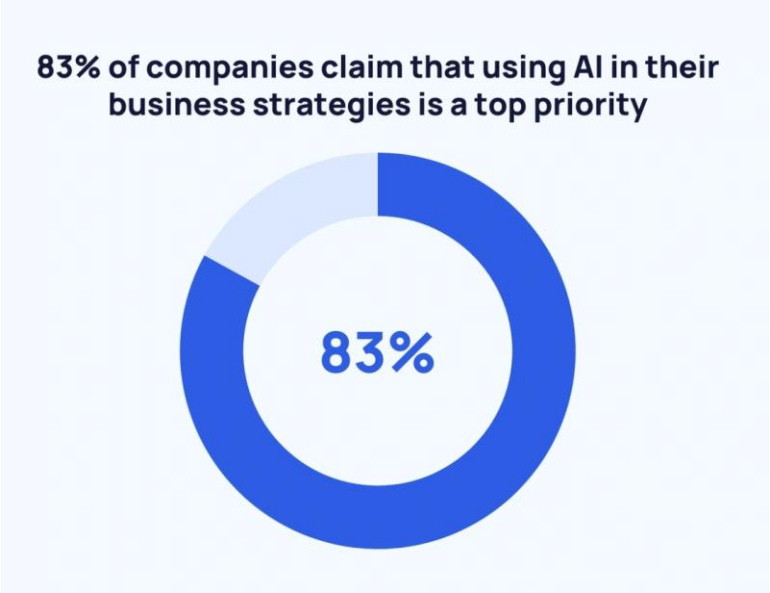

Рис 1.1. Відносна частка компаній, що виділяють ШІ, як ключовий пріоритет бізнес-стратегії

Сучасний бізнес в умовах стрімкого розвитку технологій неперервно шукає інноваційні підходи для підвищення ефективності та конкурентоспроможності. Однією з ключових технологій, яка революціонізує підхід до управління та прийняття стратегічних рішень, є машинне навчання (МН). У цій статті розглядається актуальність та перспективи використання МН в сфері бізнесу. Однією з ключових областей застосування МН в бізнесі є аналіз великих обсягів даних. Моделі МН дозволяють ефективно виявляти патерни та взаємозв'язки в даних, що надає бізнес-лідерам засоби для прийняття обгрунтованих рішень. Прогностичні алгоритми МН забезпечують можливість передбачення тенденцій ринку та оптимізації стратегій розвитку.

На рис. 1.2 зображено зростання популярності використання МН в академічному середовищі.

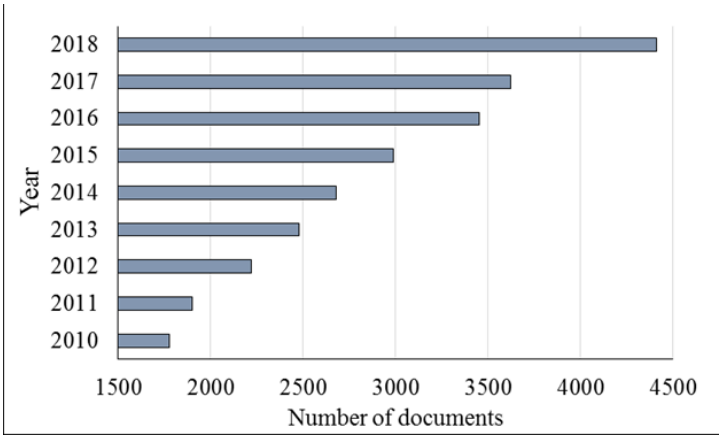

Рис. 1.2. Кількість академічних документів, де згадується МН

Використання МН у сфері маркетингу та продажів дозволяє створювати персоналізовані стратегії залучення клієнтів. Аналіз споживацьких поведінок на основі великих обсягів даних дозволяє точно визначати інтереси та потреби клієнтів, що робить рекламні кампанії більш ефективними. Використання ШІ у Email Marketing може суттєво підвищити конверсію, та дохід компаній, що показано на рис. 1.3.

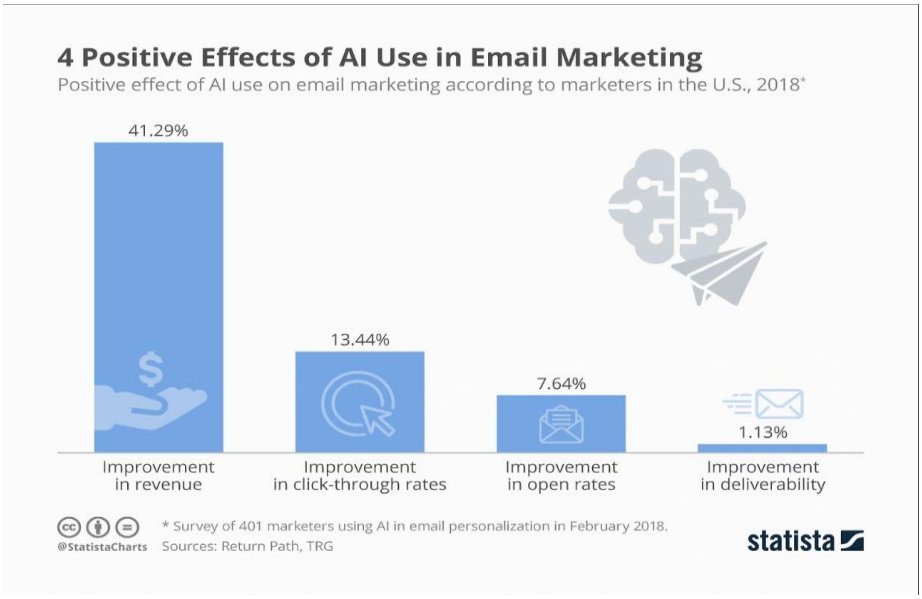

Рис. 1.3. Позитивні ефекти використання ШІ в Email Marketing

У сфері виробництва та логістики МН використовується для оптимізації ланцюга постачання, прогнозування попиту та планування виробничих процесів. Автоматизація за допомогою алгоритмів МН дозволяє підприємствам знижувати витрати та підвищувати продуктивність. МН забезпечує підприємствам засоби для управління взаємовідносинами з клієнтами. Автоматизовані системи обробки даних дозволяють розробляти індивідуальні підходи до кожного клієнта, що сприяє підвищенню рівня їхньої задоволеності.

Незважаючи на численні переваги, використання МН в бізнесі викликає певні труднощі, такі як необхідність висококваліфікованих кадрів та проблеми з конфіденційністю даних. Проте, розвиток технологій та вдосконалення алгоритмів дозволяють подолати ці виклики. Отже, використання МН в бізнесі є перспективною та необхідною стратегією для досягнення конкурентної переваги. МН надалі буде тільки зростати, що демонструє прогноз на рис. 1.4.

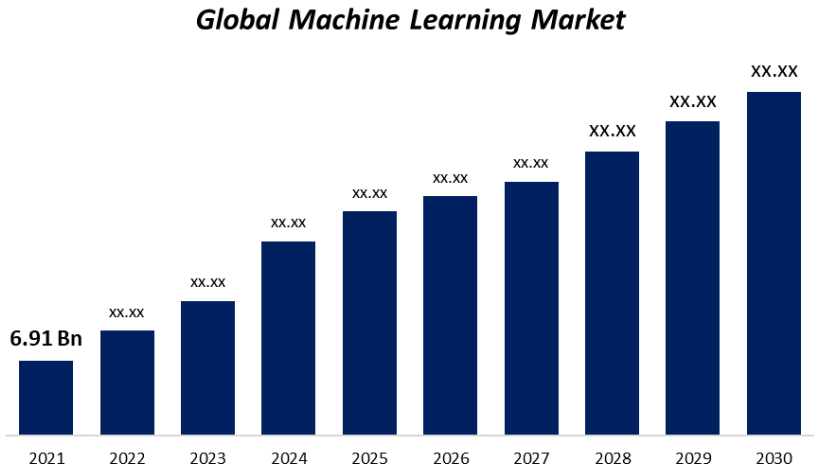

Рис. 1.4 Прогноз зростання ринку МН

Його потужність у поєднанні з великими обсягами даних створює нові можливості для ефективного управління, розвитку та інновацій в сучасному бізнес-середовищі.

РОЗДІЛ 2. ІНСТРУМЕНТИ ДОСЛІДЖЕННЯ ТА ОПИС МЕТОДІВ БІНАРНОЇ КЛАСИФІКАЦІЇ

2.1. Концептуальна схема дослідження та вибір робочих інструментів: Python, робоче середовище Jupyter notebook, Хмарне сховище Snowflake та мова SQL

Концептуальна схема дослідження передбачає в першому блоку визначення ключових аспектів поведінки споживача, які будуть моделюватися, та обґрунтування вибору конкретної моделі та показників. Другий блок передбачає використання інструментів, таких як SQL для витягування даних та Jupyter Notebook для їхньої обробки. У третьому блоку відбувається побудова моделей поведінки споживача методами машинного навчання, включаючи нормалізацію даних, формування датасетів, створення словників моделей, їх навчання, тестування та відбір оптимальних параметрів. У цьому блоку використовуються різні методи машинного навчання, такі як Логістична регресія, Метод опорних векторів, Дерева рішень, Випадковий ліс, Наївний баєсівський класифікатор, k-Найближчих сусідів та Градієнтний бустінг. Ця схема описує комплексний підхід до дослідження та моделювання поведінки споживача з використанням різноманітних методів машинного навчання та аналізу даних. на рис 2.1.

Python є важливою мовою програмування в сучасному інформаційному середовищі. Запропонований Гвідо ван Россумом у 1991 році, Python швидко здобув популярність завдяки своїй простоті, ефективності та широкому спектру застосувань. Однією з ключових особливостей Python є його читабельність коду, яка базується на філософії "читабельність має значення". Використання пробілів для форматування коду дозволяє легко розуміти структуру програм, сприяючи як швидкому написанню коду, так і його подальшій підтримці.

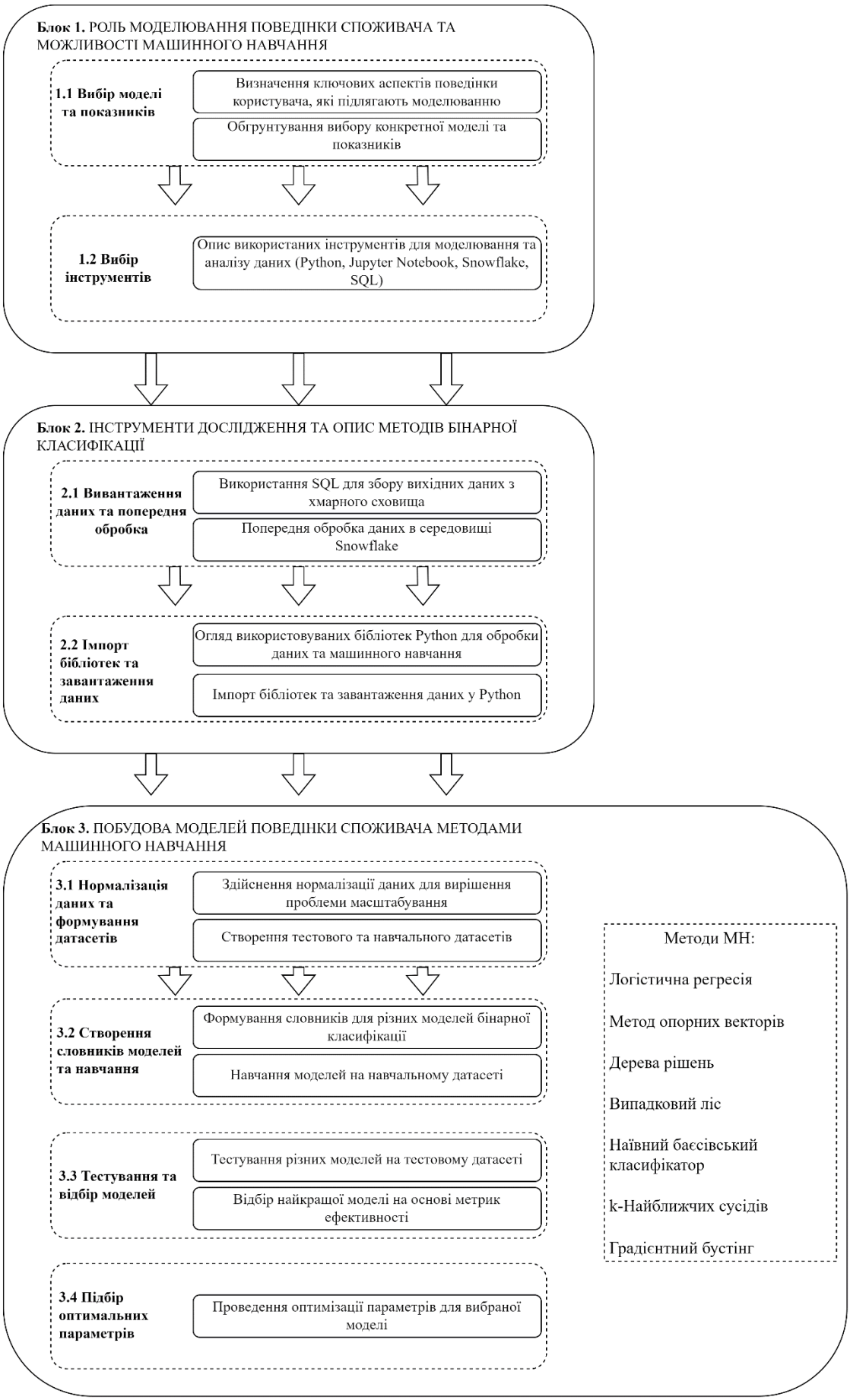

Рис 2.1 Концептуальна схема дослідження

Python визначається своєю універсальністю та багатофункціональністю. Ця мова програмування застосовується в різних сферах, включаючи веб-розробку, наукові дослідження, аналіз даних, ШІ, машинне навчання та інші галузі. Висока гнучкість Python робить його ідеальним вибором для великої кількості завдань. Розглядаючи контекст веб-розробки, за допомогою Python в мережі з'являється все більше динамічних сайтів, які існують завдяки взаємодії з базами даних. В області аналізу даних та наукових досліджень, Python є популярним інструментом завдяки своїм бібліотекам, таким як NumPy та Pandas, які дозволяють ефективно обробляти та аналізувати великі обсяги даних. Застосування Python у штучному інтелекті та машинному навчанні розширюється завдяки багатьом бібліотекам, зокрема TensorFlow та PyTorch. Це надає можливість розробки та навчання складних моделей штучного інтелекту з використанням інтуїтивного синтаксису Python.

Підсумовуючи вищесказане, Python визначається своєю простотою, читабельністю та універсальністю, що робить його важливим інструментом для розробників у різних галузях. Від веб-розробки до наукових досліджень, від штучного інтелекту до аналізу даних, Python залишається високопопулярним інструментом в сучасному програмуванні.

У сфері індустріального програмування та бізнес-розробок Python також виявляється надзвичайно корисним. Мова програмування підтримує розробку різноманітних додатків для автоматизації виробничих процесів та оптимізації управлінських завдань. Використання Python у бізнес-додатках забезпечує швидкість розробки та зручність супроводу. Однією з ключових переваг Python є його розширюваність через велику кількість бібліотек та фреймворків. Django та Flask визначають стандарти для розробки веб-додатків, пропонуючи високу продуктивність та ефективність у веб-просторі.

Python, як крос-платформена мова програмування, дозволяє розробникам створювати додатки, які працюють на різних операційних системах, зменшуючи трудомісткість розгортання та забезпечуючи високу переносимість коду. Python володіє активною та дружелюбною спільнотою

розробників, яка не лише активно розвиває мову, але й надає підтримку новачкам. Регулярні випуски нових версій мови та удосконалення покращують її функціональність та надійність.

Python – це важливий гравець у світі програмування. Від його простого синтаксису та ефективності до розширюваності через різні бібліотеки, мова виявляється необхідною для широкого спектру завдань. Невпинне розширення та вдосконалення дозволяють Python залишатися актуальним інструментом для розробників у різних галузях.

В сучасному світі, де технології стають невід'ємною частиною життя, питання оптимальної робочої середи для розробників виявляється крайне актуальним. У цьому контексті, різноманітні інструменти розробки стають ключовими аспектами продуктивної робочої діяльності. Однією з найважливіших платформ, яка привертає увагу індустрії розробки програмного забезпечення, є PyCharm. В умовах сучасної аналітики та обробки даних зростає значення використання інтелектуальних інструментів. Одним із визнаних представників цієї сфери є Jupyter, яке виявляється ключовим інструментом у світі розробки програмного забезпечення та аналітики.

У своїй основі Jupyter базується на концепції блокнотів. Це структурна компонента, що об'єднує в собі код, текст та графіку, створюючи зручне середовище для аналітиків. Його мовні можливості вражають гнучкістю, дозволяючи використовувати різні мови програмування в єдиному блокноті.

Однією з ключових особливостей Jupyter є його здатність до обробки та візуалізації даних. Інструменти для обробки даних, такі як Pandas та NumPy, забезпечують ефективну маніпуляцію та агрегацію інформації. Вбудовані засоби візуалізації, такі як Matplotlib та Seaborn, розширюють можливості аналітиків в плані представлення результатів.

Унікальність Jupyter полягає в його здатності об'єднувати код, результати та пояснення в одному блокноті, що допомагає підтримувати прозорість та взаєморозуміння учасників проекту. Це особливо важливо в

міждисциплінарних командах. Jupyter використовується не лише в аналітиці, але і в науковому процесі. Його зручний інтерфейс та можливості співпраці роблять його важливим інструментом у дослідницьких лабораторіях та наукових проектах. Розглядаючи робоче середовище Jupyter як складову інтелектуального програмування та аналітики, можна визнати його великий внесок у покращення робочих процесів та забезпечення ефективної роботи з даними. Jupyter Notebook визначається своєрідністю та гнучкістю. Цей інструмент дозволяє створювати інтерактивні блокноти, які об'єднують в собі код, виконання команд, графіку та пояснення, створюючи таким чином інтегроване середовище для аналізу даних та візуалізації результатів. Використання блокнотів дозволяє аналітикам не лише створювати ефективний код, а й зберігати коментарі та пояснення безпосередньо поруч із відповідним кодом.

Однією з найважливіших особливостей Jupyter Notebook є його мовна гнучкість. Цей інструмент підтримує універсальні мови програмування, такі як R, Julia, і звичайно ж Python, що робить його універсальним інструментом для аналізу даних на різних рівнях складності та масштабу проекту. Крім того, вбудовані інструменти для обробки даних, такі як Pandas та NumPy, сприяють ефективному аналізу та маніпуляції інформацією, тоді як графічні бібліотеки Matplotlib та Seaborn дозволяють створювати вражаючі візуалізації результатів. Особливе значення має здатність Jupyter Notebook об'єднувати код, результати та пояснення в єдиному документі. Це створює зручні умови для взаємодії між учасниками проекту та забезпечує чітку структуру та зрозумілість в робочому процесі.

У відношенні до наукового співтовариства, Jupyter Notebook є незамінним інструментом для досліджень та документації експериментів. Використання його у наукових лабораторіях дозволяє створювати документовані та відтворювані експерименти, що є важливим аспектом в наукових дослідженнях.

Враховуючи всі переваги та можливості Jupyter Notebook, можна визнати його ключовим інструментом для аналізу даних, розробки програмного забезпечення та досліджень у різних галузях науки та техніки. В даній роботі для дослідження використовувався саме Jupyter Notebook. Jupyter Notebook має свої унікальні переваги, особливо в контексті аналізу та досліджень. Jupyter Notebook має такі переваги:

інтерактивність та експерименти: Jupyter Notebook дозволяє виконувати код по частинам, дозволяючи аналітику проводити експерименти та перевіряти результати на кожному етапі. Це особливо корисно в аналізі даних та експлораторському дослідженні.

можливість використовувати блокноти для створення візуалізацій, додавання пояснень та проведення аналізу дозволяє створювати чіткі та зрозумілі звіти.

підтримка різних мов програмування: Jupyter підтримує не тільки мову програмування Python, але і інші, такі як R та Julia. Це дозволяє використовувати різні інструменти та аналітичні бібліотеки в одному документі.

легкість співпраці: Jupyter Notebook може бути легко ділений з іншими користувачами, що дозволяє створювати відкриті та взаємодійливі звіти. Інші користувачі можуть взаємодіяти з кодом, вносити зміни та додавати свої коментарі.

візуалізація результатів: Зручна інтеграція з візуалізаційними бібліотеками дозволяє легко створювати графіки та діаграми безпосередньо в блокноті, що полегшує розуміння результатів.

ефективність у навчанні та викладанні: Jupyter Notebook часто використовується в освітніх цілях для викладання програмування та аналізу даних. Він дозволяє студентам та викладачам легко ділитися матеріалами, виконувати код та переглядати результати безпосередньо в навчальних блокнотах.

Отже для аналізу даних та досліджень Jupyter Notebook може надавати великі можливості для гнучкості та зручності.

Хмарні сховища даних в сучасному інформаційному середовищі відіграють ключову роль у забезпеченні ефективного управління та аналізу великих обсягів інформації. Однією з провідних платформ у цьому напрямку є Snowflake – хмарне сховище, яке вирізняється своєю унікальною архітектурою та високою продуктивністю. Однією з ключових особливостей Snowflake є архітектура з розділенням обчислення та зберігання даних. Це дозволяє оптимізувати роботу з великими обсягами інформації та забезпечує високу швидкодію в операціях аналізу та опрацювання даних. Забезпечуючи повністю управліність витратами, Snowflake дозволяє користувачам платити лише за той об'єм ресурсів, який вони використовують. Це робить його ефективним рішенням для підприємств з різних галузей, зокрема для малих та середніх підприємств, які цінують ефективність і економію.

Однією з переваг Snowflake є його гнучкість у використанні різних джерел даних. Він інтегрується з різноманітними інструментами та джерелами даних, що дозволяє користувачам легко об'єднувати дані з різних джерел для комплексного аналізу. Це робить Snowflake відмінним вибором для компаній, які працюють з розмаїттям даних та вимагають консолідації інформації.

Безпека є ще однією ключовою аспектом, на якому акцентує Snowflake. Забезпечуючи високий рівень шифрування та контролю доступу, ця платформа враховує сучасні стандарти безпеки даних. Це особливо важливо в умовах сучасного цифрового середовища, де конфіденційність та цілісність даних є пріоритетом.

Ще однією перевагою Snowflake є його можливість масштабування. Платформа дозволяє легко збільшувати обсяги даних та ресурси в залежності від потреб користувача. Це робить його ідеальним рішенням для компаній, які планують розвиватися та збільшувати обсяги обробки даних у майбутньому.
У контексті аналізу даних Snowflake надає широкі можливості для використання різноманітних інструментів та мов програмування. Його

інтеграція з Python, R, та Java робить його відмінним вибором для аналітиків та науковців, які використовують різні інструменти для виконання своїх завдань. Окрім цього, Snowflake відзначається високою доступністю та надійністю. Його хмарна інфраструктура дозволяє забезпечити стабільність та доступність даних навіть при великих навантаженнях. Це є важливим фактором для підприємств, які вимагають неперервного доступу до критичних даних. Загалом, Snowflake – це високоефективне, гнучке та безпечне рішення для управління та аналізу даних в умовах сучасного бізнесу. Його функціональність та унікальні особливості роблять його важливим інструментом для компаній, що прагнуть до ефективного використання своїх даних в умовах динамічного інформаційного ландшафту.

Мова SQL (Structured Query Language) є однією з найважливіших мов програмування в сфері баз даних та управління інформацією. Розроблена в 1970-х роках Дональдом Чемберліном та Реймондом Бойсом в лабораторії IBM, SQL стала стандартом для взаємодії з реляційними базами даних та забезпечення доступу до даних.

Історія SQL сягає часів, коли комп'ютерні технології тільки входили в повсякденне використання. Виникнення реляційних баз даних призвело до потреби в стандартизованій мові для управління цими базами. У 1974 році Едгар Ф. Кодд запропонував модель даних, що лежить в основі реляційних баз, а саме табличну структуру, і це визначило основи для мови SQL.

SQL став прийнятним стандартом в 1986 році, коли American National Standards Institute (ANSI) вперше стандартизував мову як SQL-86. З тих пір було кілька оновлень, таких як SQL-89, SQL-92 та подальші. Ці стандарти визначають основні концепції та синтаксис мови SQL, але деякі бази даних розширили свої функції, додаючи власні розширення. Мова SQL складається з різних компонентів, які визначаються стандартами та їхніми реалізаціями. Основні команди SQL включають оператори SELECT, INSERT, UPDATE та DELETE, які забезпечують можливість вибору, вставки, оновлення та видалення даних з бази даних. Ключовим аспектом SQL є його здатність

взаємодіяти з реляційними базами даних. Оператор SELECT використовується для вибору даних з однієї або кількох таблиць. Параметри SELECT дозволяють фільтрувати та упорядковувати дані, агрегаційні функції, такі як SUM або AVG, дозволяють проводити обчислення на вибраних даних. INSERT додає нові рядки в таблицю бази даних. Оператор UPDATE змінює існуючі дані в таблиці, а DELETE видаляє рядки. Кожен з цих операторів вимагає вірного використання синтаксису та ключових слів для досягнення очікуваного результату.

Окрім базових операцій з даними, SQL також надає можливості для створення та управління структурою бази даних. CREATE використовується для створення нових таблиць, індексів та інших об'єктів. ALTER дозволяє змінювати структуру бази даних, додаючи або вилучаючи колонки, індекси чи обмеження. DROP видаляє об'єкти з бази даних. SQL також має можливості для об'єднання таблиць за допомогою операцій JOIN, які дозволяють з'єднувати дані з різних джерел. Підзапити дозволяють виконувати запити в межах іншого запиту, що розширює функціональність SQL.

Іншим важливим аспектом мови SQL є можливість управління транзакціями. COMMIT фіксує зміни, зроблені в рамках транзакції, та забезпечує їхню постійність у базі даних. ROLLBACK скасовує всі зміни, внесені під час транзакції, у випадку помилки чи іншого непередбаченого стану.

Мова SQL є важливим інструментом для розробників, адміністраторів баз даних та аналітиків. Її стандартизований синтаксис та потужні можливості забезпечують ефективну роботу з даними в реляційних базах даних. Різні системи керування БД, такі як MySQL, PostgreSQL, та Oracle, використовують SQL як основну мову для взаємодії з користувачами та програмами [18]. Мова SQL вирізняється численними перевагами, які роблять її невід'ємною частиною аналізу та досліджень в області баз даних. Ось деякі з них:

Ефективний запит даних: SQL дозволяє виконувати потужні та складні запити для отримання необхідних даних з бази. Оператор SELECT разом із

умовами, групуванням, впорядкуванням і агрегаційними функціями забезпечують гнучкість у виборі і аналізі даних.

Операції зі з'єднанням та об'єднанням: SQL надає можливість об'єднувати дані з різних таблиць за допомогою операцій JOIN, що є ключовим аспектом для аналізу зв'язаних даних. Це спрощує проведення складних аналітичних операцій.

Агрегаційні функції: Мова SQL має вбудовані функції, такі як SUM, AVG, COUNT, MAX та MIN, які дозволяють проводити агрегаційний аналіз. Це корисно для визначення статистики та отримання загального уявлення про дані.

Підзапити: SQL дозволяє використовувати підзапити, що полегшує виконання вкладених запитів. Це корисно для складних аналітичних завдань, де потрібно використовувати результати одного запиту в іншому.

Можливості фільтрації та умов: SQL надає широкий спектр можливостей для фільтрації даних за допомогою WHERE, HAVING та інших умов, що важливо для точного визначення даних для аналізу.

Масштабованість: SQL забезпечує ефективну роботу з великими обсягами даних. Індексація та оптимізація запитів дозволяють швидко виконувати операції навіть в умовах великого обсягу інформації.

Транзакційний підхід: Мова SQL підтримує транзакції, що дозволяє виконувати безпечні та консистентні зміни в базі даних. COMMIT та ROLLBACK дозволяють керувати цим процесом.

Стандартизація: SQL є стандартизованою мовою, що робить її переносною між різними системами керування базами даних. Це дозволяє використовувати однакові або схожі запити на різних платформах.

Загалом, SQL є потужним інструментом для аналізу та досліджень в сфері баз даних. Його стандартизований синтаксис і розширені можливості роблять його невід'ємною частиною роботи з даними в бізнесі, науці та інших галузях [19].

2.2. Бібліотеки Python для обробки даних та машинного навчання

У сучасному світі обробка даних та машинне навчання стають неодмінною частиною різноманітних сфер, від науки до бізнесу. Python, завдяки своїй простоті та великому екосистему бібліотек, став однією з провідних мов програмування у цих областях. У цій статті розглянемо ключові бібліотеки Python для обробки даних та машинного навчання та їхні можливості.

Однією з найпопулярніших бібліотек для обробки даних є Pandas. Вона надає структури даних, такі як DataFrame, які спрощують читання та обробку даних. Pandas використовується для фільтрації, сортування, групування та агрегації даних.

Бібліотека Numpy дозволяє виконувати ефективні математичні операції над масивами даних і має важливе значення для числових обчислень, наприклад, лінійна алгебра та обробка сигналів.

Matplotlib та Seaborn використовуються для візуалізації даних. Matplotlib дозволяє створювати різноманітні графіки, діаграми та зображення, тоді як Seaborn надає високорівневий інтерфейс для створення привабливих статистичних графіків.

Scikit-learn - бібліотека для МН, включає широкий спектр алгоритмів класифікації, регресії, кластеризації та інших методів, а також містить інструменти для підготовки даних та оцінки моделей.Scikit-learn, як вже зазначено, є високопопулярною бібліотекою для розв'язання завдань машинного навчання. Вона включає в себе різноманітні алгоритми, такі як метод опорних векторів (SVM), дерева рішень, наївний байєсівський класифікатор і багато інших. Зручний інтерфейс Scikit-learn дозволяє легко навчати моделі, виконувати крос-валідацію та оцінювати їх ефективність.

TensorFlow та PyTorch: ці бібліотеки використовуються для роботи з нейронними мережами та глибоким навчанням. TensorFlow розроблений

Google, в той час як PyTorch став вибором для багатьох дослідників та практиків у галузі машинного навчання.

TensorFlow і PyTorch представляють собою бібліотеки для глибокого навчання і нейронних мереж. TensorFlow розроблений командою Google та широко використовується в індустрії. Він надає гнучкість та високу продуктивність, а також інструменти для розгортання моделей на різних платформах. PyTorch, в свою чергу, отримав широкий розпізнавання в наукових колах. Він відрізняється динамічним обчисленням, що полегшує експериментування з моделями.

Keras – це високорівневий інтерфейс для роботи з TensorFlow, який полегшує створення, тренування та експериментування з нейронними мережами. Keras - це високорівневий інтерфейс для TensorFlow, що дозволяє швидко створювати та навчати нейронні мережі. Він забезпечує зручний інтерфейс для визначення архітектури мережі, обчислення втрат та вибір оптимізаторів.

Бібліотеки для Візуалізації Даних: Matplotlib та Seaborn. Matplotlib та Seaborn - це потужні бібліотеки для створення різноманітних графічних візуалізацій. Matplotlib дозволяє створювати графіки на високому рівні та налаштовувати їх деталі. Seaborn надає високорівневі інтерфейси для статистичних графіків, що полегшує їхнє створення та розуміння.

Бібліотеки Python для обробки даних та машинного навчання грають ключову роль в розвитку інтелектуальних систем та аналізі великих обсягів інформації. Вони надають розробникам та дослідникам потужні інструменти для вирішення складних завдань. Вибір конкретних бібліотек залежить від конкретного завдання, але спільно вони утворюють ефективний інструментарій для роботи з даними та розвитку моделей машинного навчання.

Об'єднуючи бібліотеки для обробки даних та машинного навчання, Python стає потужним інструментом для розв'язання завдань аналізу даних. Ці бібліотеки надають розробникам і дослідникам необхідні інструменти для

створення складних моделей, ефективного аналізу даних та візуалізації результатів. Вибір конкретної бібліотеки залежить від завдань та особливостей конкретного проекту, проте використання їх в комбінації може забезпечити найкращі результати у світі обробки даних та машинного навчання.

У світі постійних технологічних інновацій, використання Python для обробки даних та машинного навчання стає необхідністю. Великий вибір бібліотек дозволяє розробникам та дослідникам створювати інтелектуальні системи, які вирішують складні завдання.

Зазначені бібліотеки - Pandas, Numpy, Matplotlib, Seaborn, Scikit-learn, TensorFlow, PyTorch і Keras - утворюють інструментальний арсенал для роботи з різноманітними завданнями. Pandas дозволяє швидко обробляти та аналізувати дані, забезпечуючи ефективні інструменти для роботи з табличними даними. Numpy дозволяє виконувати високоефективні математичні операції над масивами даних. Matplotlib та Seaborn роблять візуалізацію даних доступною та ефективною. Scikit-learn, TensorFlow, PyTorch і Keras є невід'ємною частиною арсеналу для роботи з машинним навчанням. Scikit-learn, з його різноманіттям алгоритмів, надає інструменти для класифікації, регресії та кластеризації даних. TensorFlow і PyTorch, в свою чергу, стали ключовими фреймворками для роботи з глибоким навчанням, визначаючи нові стандарти в цій області. Керування проектом та взаємодія з даними можливі завдяки бібліотекам, таким як Pandas та Numpy. Ці бібліотеки надають засоби для очищення даних, обробки пропущених значень та виконання ефективних операцій з даними. Основна перевага Python у цьому контексті полягає в його великому та активному спільноті розробників, яка невпинно розширює функціональні можливості цієї екосистеми. Завдяки цьому, мова стає справжнім лідером у сфері аналізу даних та машинного навчання.

Враховуючи швидкість розвитку технологій, важливо вдосконалювати та доповнювати наявні інструменти. Дослідження у галузі машинного

навчання продовжують визначати нові методи та моделі, що розширює можливості програмістів та аналітиків.

Усе враховуючи, Python залишається ключовим гравцем у сфері обробки даних та машинного навчання. Його простота та силу зробили популярним вибором для розв'язання різних завдань у сучасному світі обчислювальної аналітики. Застосування цих бібліотек забезпечує високий рівень ефективності та гнучкість для вирішення найскладніших викликів в області обробки даних та машинного навчання.

2.3. Основні задачі машинного навчання та моделі бінарної класифікації

Машинне навчання (МН) є ключовою галуззю штучного інтелекту, що досліджує розробку та використання алгоритмів, які навчаються на основі даних. Вона виникла від необхідності вирішення складних завдань, для яких традиційні програмні підходи виявляються неефективними. Машинне навчання знаходить широке застосування в різних галузях, від медицини та фінансів до технологій та науки про дані.

Класифікація є однією з ключових задач машинного навчання. Її мета полягає в тому, щоб призначити об'єктам одну з попередньо визначених категорій. Наприклад, це може бути використано для розпізнавання електронних листів як спам чи не спам, або для визначення, чи на зображенні зображено кішку чи собаку. Алгоритми класифікації навчаються на основі навчальних даних, де для кожного об'єкта вказана його категорія.

Регресія, інша важлива задача машинного навчання, спрямована на прогнозування числових значень на основі вхідних даних. Модель навчається побудованій функції, яка може передбачити значення цільової змінної. Це

може бути використано, наприклад, для прогнозування ціни на нерухомість або вартості акцій.

Класифікація та регресія визначають напрямок завдань, спрямованих на прогнозування та класифікацію даних. Завдяки цим задачам, ми можемо робити передбачення у різноманітних галузях, від фінансів до медицини. Використання алгоритмів класифікації дозволяє приймати важливі рішення на основі аналізу великих обсягів даних.

Кластеризація - задача групування об'єктів в класи на основі схожості між ними. Кластеризація дозволяє виявляти природні групи в даних, які можуть бути важко виявити іншими методами. Наприклад, це може бути використано для аналізу споживчого побуту та розділення споживачів на групи за їхніми покупками [15]. Кластеризація допомагає виявляти природні групи в наборах даних, що дозволяє здійснювати більш ефективний аналіз і взаємодію з даними. Знання правил надає можливість створювати моделі, які базуються на заранее визначених правилах, що сприяє прийняттю рішень у складних сценаріях.

Знання правил включає в себе вивчення правил, які можна використовувати для прийняття рішень. Моделі для цієї задачі створюються на основі великої кількості правил, які визначають залежності між різними факторами та результатом.

Зменшення розмірності - це завдання, пов'язане зі зменшенням кількості факторів, які враховуються в моделі. Це часто потрібно у випадках, коли кількість ознак у наборі даних надто велика, що може призвести до перенавчання моделі або ускладнення її роботи. Зменшення розмірності є важливою задачею у випадках, коли кількість ознак у даних є великою. Це робить моделі більш ефективними та запобігає перенавчанню. Виявлення аномалій визначає ізольовані або несподівані значення в даних, що є ключовим для вчасного реагування на потенційні проблеми.

Виявлення аномалій - це задача виявлення несподіваних або викинутих патернів в даних. Алгоритми для виявлення аномалій шукають несподівані або

викинуті значення, які можуть вказувати на потенційно проблемні об'єкти чи ситуації.

Підсилене навчання - це галузь машинного навчання, де модель навчається на основі взаємодії з навколишнім середовищем. Модель намагається максимізувати винагороду в результаті своїх дій.

Підсилене навчання, де модель вчиться взаємодіяти з навколишнім середовищем, є важливим напрямком для навчання моделей самоорганізації та самовдосконалення.

Описані задачі визначають широкий спектр досліджень у сфері машинного навчання та використовуються в різних областях, сприяючи розвитку та вдосконаленню цієї цікавої та перспективної науки. Машинне навчання стало ключовим напрямком в галузі штучного інтелекту, революціонізуючи спосіб, яким ми розуміємо та використовуємо дані. У цій статті ми розглянули різноманітні задачі машинного навчання та їхнє значення в різних сферах. У світі бізнесу машинне навчання реалізується в різних аспектах, від прийняття рішень та прогнозування попиту до виявлення аномалій та автоматизації багатьох процесів. Основні бібліотеки, такі як TensorFlow та PyTorch, надають інструменти для створення потужних моделей машинного навчання, що сприяє подальшому розвитку цієї галузі.

Загалом, машинне навчання визначає нову еру в обробці та аналізі даних. З його допомогою відкриваються нові можливості для розв'язання складних завдань, що раніше здавалися недосяжними. Продовжуючи вивчати та розвивати цю галузь, ми можемо сподіватися на ще більше інновацій та вдосконалень у майбутньому.

Машинне навчання (МН) востаннє десятиліттям визначає новий рівень розвитку інформаційних технологій, а однією з найбільш розповсюджених задач у цій галузі є бінарна класифікація. Цей напрям визначається як процес призначення кожному об'єкту одного з двох класів, тобто відповідь на питання "Так" чи "Ні". У цій статті розглянемо ключові аспекти моделей бінарної

класифікації в машинному навчанні, їхню роль у сучасному світі та перспективи подальшого розвитку цього напряму.

Теоретичні аспекти бінарної класифікації. В основі бінарної класифікації лежать математичні моделі, які вчаться на основі наборів даних. Однією з найбільш поширених моделей є логістична регресія. Вона забезпечує високу ефективність у вирішенні задач бінарної класифікації завдяки своїй простоті та здатності пристосовуватися до різноманітних умов. Математично модель логістичної регресії визначається логістичною функцією, яка відображає ймовірність належності об'єкта до одного з класів. Рішення приймається на основі порогового значення ймовірності, яке може бути адаптовано в процесі навчання моделі. Такий підхід робить логістичну регресію універсальним інструментом для багатьох задач, де необхідно розділити об'єкти на два класи.

Приклади використання бінарної класифікації: Бінарна класифікація застосовується в різних сферах, що відображає її універсальність та значущість. Однією з основних областей використання є медична діагностика. Моделі бінарної класифікації можуть допомагати у виявленні хвороб на ранніх стадіях, аналізуючи медичні зображення або результати аналізів.

У фінансовому секторі бінарна класифікація використовується для оцінки ризиків та прийняття рішень щодо інвестицій. Моделі можуть прогнозувати ймовірність дефолту або успішності інвестиційних проектів, що є критичним для прийняття фінансових рішень. В сфері електронної комерції бінарна класифікація використовується для прогнозування покупкового керівництва та визначення інтересів клієнтів. Це дозволяє персоналізувати пропозиції та покращити стратегії маркетингу.

Виклики та перспективи. Незважаючи на успіхи, моделі бінарної класифікації стикаються з деякими викликами. Один з них - це проблема невизначеності в прийнятті рішень. Моделі можуть надто впевнено робити прогнози на основі обмежених даних, що призводить до помилок.

Другий виклик пов'язаний із забрудненням даних. Наявність шуму чи викидів в навчальних даних може призвести до втрати точності класифікації. Ефективна обробка та очищення даних відіграє ключову роль у подоланні цього виклику.

Перспективи розвитку бінарної класифікації пов'язані з удосконаленням алгоритмів навчання, розширенням обсягу доступних даних та вдосконаленням взаємодії з іншими галузями машинного навчання, такими як змішане навчання. Отже, моделі бінарної класифікації в машинному навчанні є потужним інструментом для вирішення різноманітних задач в різних галузях. Їхня універсальність та ефективність роблять їх невід'ємною частиною розвитку інформаційних технологій. Незважаючи на виклики, з якими вони стикаються, постійні дослідження та вдосконалення алгоритмів дозволяють нам зробити новий крок у напрямку розуміння та використання моделей бінарної класифікації для вирішення реальних завдань.

Вище були розглянуті ключові аспекти та важливість використання моделей бінарної класифікації в машинному навчанні. На тлі швидкого розвитку цієї галузі та зростання обсягів доступних даних, бінарна класифікація стає необхідним інструментом для рішення різноманітних завдань в різних сферах. Однією з ключових переваг бінарної класифікації є її універсальність. Моделі такого типу успішно використовуються у медицині для діагностики, у фінансах для оцінки ризиків та в електронній комерції для персоналізації підходу до клієнтів. Їхній внесок у вирішення проблем та оптимізацію бізнес-процесів важливий для сучасного суспільства. Проте, незважаючи на успіхи, існують виклики, з якими стикаються моделі бінарної класифікації. Проблема невизначеності та забруднення даних вимагають уваги та пошуку ефективних стратегій для подолання. Невірне прийняття рішень може призвести до серйозних наслідків, тому вдосконалення алгоритмів та підходів до навчання моделей є актуальною задачею.

Перспективи розвитку бінарної класифікації пов'язані з пошуком нових методів врахування невизначеності, розвитком технологій збору та обробки

даних, а також пошуком оптимальних стратегій навчання для підвищення точності класифікації. Застосування новітніх методів та підходів може сприяти поліпшенню якості роботи моделей та розширенню їхнього застосування в різних галузях.

Усе враховуючи, моделі бінарної класифікації визначають новий етап розвитку машинного навчання та призначені для вирішення важливих завдань у різних галузях. Вони стають потужним інструментом для аналізу та прийняття рішень, допомагаючи вирішувати складні завдання та роблячи технології доступними та ефективними для різноманітних викликів. Розуміння їхньої ролі та постійне вдосконалення може сприяти подальшому розвитку цієї важливої галузі інформаційних технологій.

РОЗДІЛ 3. ПОБУДОВА МОДЕЛЕЙ ПОВЕДІНКИ СПОЖИВАЧА МЕТОДАМИ МАШИННОГО НАВЧАННЯ

3.1. Збір вихідних даних з хмарного сховища за допомогою SQL

Для збору вихідних даних користувачів додатку та подальшої побудови моделей машинного навчання в рамках проведення дослідження було використано SQL-запити до різних таблиць, що знаходяться на хмарному сховищі Snowflake. Для цього було створено декілька запитів, які об'єднали інформацію з десятків таблиць. Додатково, для побудови моделі було визначено незалежні фактори, які можуть впливати на залежну метрику - здійснення покупки користувачем. Серед цих факторів були такі параметри, як тип ОС (Android, iOS, MacOS, Windows), мова інтерфейсу, кількість сесій під час тестового періоду, загальна довжина сесій, кількість використань найпопулярніших функцій додатку.

Y – converted (1 чи 0) – позначає наявність подальшої конверсії у користувача;

$X1$ – кількість сесій під час тестового періоду;

$X2$ – загальна довжина сесій під час тестового періоду;

$X3$ – мова інтерфейсу

$X4$ – платформа;

$X5$ – кількість івентів типу 1;

$X6$ – кількість івентів типу 2;

$X7$ – кількість івентів типу 3;

$X8$ – кількість івентів типу 4;

$X9$ – кількість івентів типу 5;

$X10$ – час, за який був намальований перший персонаж;

$X11$ – кількість часу, що було проведено в меню покупки;

$X12$ – кількість натискань на кнопку “купити”;

X13 – кількість відкриттів меню покупки.

Оскільки дані були розподілені між різними таблицями, було необхідно написати декілька SQL запитів. У наступному розділі будуть наведені приклади таких запитів, і фінальний запит можна буде знайти в Додатку А.

На прикладі можна побачити, що запит для отримання інформації про згадані незалежні фактори для неконвертованих юзерів складається з багатьох підзапитів та має наступний вигляд (див. рис. 3.1, 3.2, 3.3).

```
WITH unconverted AS (
SELECT user_id, listagg(buys_fact.purchase_type)
FROM buys_fact
GROUP BY user_id
HAVING listagg(buys_fact.purchase_type) = 'trial'
),
trials AS (
SELECT buys_fact.user_id, min(fulldate) AS trial_date
FROM buys_fact
INNER JOIN unconverted ON unconverted.user_id = buys_fact.user_id
LEFT JOIN dates_dim ON dates_dim.id = buys_fact.created_date_id
GROUP BY buys_fact.user_id
),

session_num AS (
SELECT trials.user_id, COUNT(*) AS num_of_ses FROM trials INNER JOIN sessions_fact ON sessions_fact.user_id = trials.user_id
WHERE event_timestamp <= DATEADD(day, 3, trials.trial_date)
GROUP BY trials.user_id
),

session_duration AS (
SELECT trials.user_id, sum(duration) / 60 duration_in_minutes FROM trials INNER JOIN sessions_fact ON sessions_fact.user_id = trials.user_id
WHERE event_timestamp <= DATEADD(day, 3, trials.trial_date)
GROUP BY trials.user_id
),
```

Рис. 3.1. SQL запит для отримання даних

```
enable_system_layer AS (
SELECT trials.user_id, COUNT(*) AS num FROM events_fact
INNER JOIN trials ON trials.user_id = events_fact.user_id
WHERE event_timestamp <= DATEADD(day, 3, trials.trial_date)
AND event_type = 'enable_system_layer'
GROUP BY trials.user_id
),

search_item_selected AS (
SELECT trials.user_id, COUNT(*) AS num FROM events_fact
INNER JOIN trials ON trials.user_id = events_fact.user_id
WHERE event_timestamp <= DATEADD(day, 3, trials.trial_date)
AND event_type = 'search_item_selected'
GROUP BY trials.user_id
),

search_opened AS (
SELECT trials.user_id, COUNT(*) AS num FROM events_fact
INNER JOIN trials ON trials.user_id = events_fact.user_id
WHERE event_timestamp <= DATEADD(day, 3, trials.trial_date)
AND event_type = 'search_opened'
GROUP BY trials.user_id
),

infobox_item_selected AS (
SELECT trials.user_id, COUNT(*) AS num FROM events_fact
INNER JOIN trials ON trials.user_id = events_fact.user_id
WHERE event_timestamp <= DATEADD(day, 3, trials.trial_date)
AND event_type = 'infobox_item_selected'
```

Рис. 3.2. SQL запит для отримання даних

```
infobox_item_selected AS (
SELECT trials.user_id, COUNT(*) AS num FROM events_fact
INNER JOIN trials ON trials.user_id = events_fact.user_id
WHERE event_timestamp <= DATEADD(day, 3, trials.trial_date)
AND event_type = 'infobox_item_selected'
GROUP BY trials.user_id
),

infobox_menu_item_selected AS (
SELECT trials.user_id, COUNT(*) AS num FROM events_fact
INNER JOIN trials ON trials.user_id = events_fact.user_id
WHERE event_timestamp <= DATEADD(day, 3, trials.trial_date)
AND event_type = 'infobox_menu_item_selected'
GROUP BY trials.user_id
)

SELECT  users_dim.id,
        session_num.num_of_ses,
        session_duration.duration_in_minutes,
        CASE users_dim.locale WHEN 'en' THEN 0 WHEN 'zh-hans' THEN 1 WHEN '' THEN 2 WHEN 'fr' THEN 3 WHEN 'es' THEN 4 WHEN 'zh' THEN 5 WHEN 'de' THEN 6 END AS locale,
        CASE platform WHEN 'Android' THEN 0 WHEN 'Mac OS' THEN 1 WHEN 'Windows' THEN 2 WHEN null THEN 3 WHEN 'iOS' THEN 4 END AS platform,
        coalesce(enable_system_layer.num, 0) AS "event_1",
        coalesce(search_item_selected.num, 0) AS "event_2",
        coalesce(search_opened.num, 0) AS "event_3",
        coalesce(infobox_item_selected.num, 0) AS "event_4",
        coalesce(infobox_menu_item_selected.num, 0) AS "event_5",
        0 AS "converted"
FROM users_dim
INNER JOIN session_num ON session_num.user_id = users_dim.id
INNER JOIN session_duration ON session_duration.user_id = users_dim.id
LEFT JOIN enable_system_layer ON enable_system_layer.user_id = users_dim.id
LEFT JOIN search_item_selected ON search_item_selected.user_id = users_dim.id
LEFT JOIN search_opened ON search_opened.user_id = users_dim.id
LEFT JOIN infobox_item_selected ON infobox_item_selected.user_id = users_dim.id
LEFT JOIN infobox_menu_item_selected ON infobox_menu_item_selected.user_id = users_dim.idSELECT * FROM demo;
```

Рис. 3.3. SQL запит для отримання даних

Запит для конвертованих користувачів має незначні зміни, порівняно з цим запитом, тому його наведено окремо в додатку.

Перший CTE-запит визначає користувачів, що в історії покупок мають тільки тестовий період. Другий CTE-запит визначає дату тестового періоду, що трапився найраніше в історії покупок. Третій та четвертий CTE-запити визначають кількість сесій та їх загальну довжину у хвилинах. Серія наступних CTE-запитів визначає кількість використань основних функцій додатку під час тестового періоду, який складає три дні від дати створення запису про тестовий період.

Останній запит об’єднує дані CTE-запитів в один за допомогою унікального ідентифікатора користувача – user_id. Також цей запит надає числові значення для категоріальних даних: “en” дорівнює 0 для поля locale, це означає, що англійська мова інтерфейсу додатку в користувача буде позначатись цифрою 0. Теж саме було зроблено для даних стосовно ОС користувача:

“Android” дорівнює 0 для поля platform, що означає, що ОС Андроід у користувача буде позначатись цифрою 0. Для позначення кількості використань визначених функцій додатку було використано функцію Coalesce. Ця функція повертає перше непусте(not null) значення з наданого списку. Якщо подібне значення відсутнє – функція повертає надане значення, що у даному випадку є 0. Останнє, що робить цей запит – це вводить параметр конверсії користувача, який для цього запиту є 0, тобто користувач не зробив покупку.

3.2 Попередня обробка даних в середовищі Jupyter Notebook

Для аналізу даних на початковому етапі важливо обрати оптимальне середовище. Переваги різних опцій розглядалися у розділі 2.1. Оскільки потрібно було виконувати код поетапно та працювати з вже обробленими даними, вибір був здійснений на користь Jupyter Notebook. Це середовище дозволяє виконувати код поетапно, перевіряти проміжні результати та взаємодіяти з даними у реальному часі, а інтеграція з бібліотеками візуалізації, такими як Matplotlib, Seaborn та Plotly, дозволяє швидко створювати графіки та графічні представлення результатів досліджень.

Перший крок – імпортування бібліотек Python для роботи з даними, та машинного навчання, які будуть використовуватися надалі, та завантаження вихідних даних у змінну data, що зображено на рис. 3.4. Далі було застосовано метод `describe`, який використовується для створення статистичного опису числових даних в об'єктах Pandas DataFrame або Series. Цей метод генерує основні статистичні показники, які надають інформацію про розподіл, центр та розкид даних. Його застосування показано на рис 3.6.

```
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV

file_path = 'C:/Users/myronenko.a/JupiterWorkingDirectory/Data/diplomdata.csv'
data = pd.read_csv(file_path, sep=',', header=0)

data.head()
```

	ID	NUM_OF_SES	DURATION_IN_MINUTES	LOCALE	PLATFORM	event_1	event_2	event_3	event_4	event_5	converted
0	3503944	1	15.033333	0	0.0	2	38	42	78	14	1
1	3746483	1	3.000000	3	2.0	1	5	0	0	0	1
2	216998	3	0.083333	0	1.0	0	0	0	0	0	1
3	2701311	2	2397.383333	0	1.0	45	0	0	10	4	1
4	3883731	3	31.983333	3	1.0	29	2	2	0	0	1

Рис. 3.4. Імпорт бібліотек та даних

Спочатку переконаємося, що event стовпці не мають пустих значень (рис 3.5).

```
data['event_1'] = data['event_1'].fillna(0)
data['event_2'] = data['event_2'].fillna(0)
data['event_3'] = data['event_3'].fillna(0)
data['event_4'] = data['event_4'].fillna(0)
data['event_5'] = data['event_5'].fillna(0)
```

Рис. 3.5. Перевірка стовпчиків на пусті значення

```
In [5]: data.describe()
```

Out[5]:

	ID	NUM_OF_SES	DURATION_IN_MINUTES	LOCALE	PLATFORM	event_1	event_2	event_3	event_4	
count	9.179830e+05	917983.000000	917983.000000	917983.000000	852041.000000	917983.000000	917983.000000	917983.000000	917983.000000	9179
mean	2.694820e+06	7.849247	100.768625	0.625062	2.666441	29.776765	4.495039	3.855676	6.226624	
std	8.487772e+05	14.788351	702.219999	1.435879	1.630163	55.461588	17.136756	14.048190	21.554904	
min	1.950000e+02	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	2.003654e+06	2.000000	4.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	
50%	2.680117e+06	3.000000	17.066667	0.000000	4.000000	7.000000	0.000000	0.000000	0.000000	
75%	3.497372e+06	8.000000	68.000000	0.000000	4.000000	38.000000	1.000000	1.000000	4.000000	
max	4.108889e+06	1312.000000	446679.400000	6.000000	4.000000	2347.000000	2162.000000	984.000000	5180.000000	7:

Рис. 3.6. Опис даних за допомогою методу describe

Загальна кількість рядків 917983. Можемо побачити, що дані мають викиди по «верхній» межі розподілу для деяких стовпчиків. Наприклад, поле «DURATION_IN_MINUTES» має середнє значення 100.76, що означає, що середня загальна тривалість сесій для користувача під час тестового періоду складає трохи більше ніж півтори години. Але максимальне значення складає 446679 хвилин, що може означати, що один обліковий запис могли використовувати декілька користувачів, що не підходить для вибірки даних для навчання моделі. Приберемо викиди та подивимось на отримані дані:

```
list_to_filter = ['NUM_OF_SES', 'DURATION_IN_MINUTES', 'event_1',
'event_2', 'event_3', 'event_4', 'event_5']
filtered = data.copy()
```

```
for col in list_to_filter:
    high = data[col].quantile(0.98)
    filtered = filtered[(filtered[col] < high)]
```

```
filtered.describe()
```

	ID	NUM_OF_SES	DURATION_IN_MINUTES	LOCALE	PLATFORM	event_1	event_2	event_3	event_4	
count	8.364270e+05	836427.000000	836427.000000	836427.000000	779446.000000	836427.000000	836427.000000	836427.000000	836427.000000	8364
mean	2.687460e+06	5.752333	53.526385	0.621647	2.665847	22.774847	2.136758	1.905693	3.321727	
std	8.502297e+05	7.253346	109.309687	1.433175	1.635545	35.042618	5.998359	5.177314	7.711513	
min	1.950000e+02	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.987714e+06	1.000000	3.366667	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	
50%	2.668649e+06	3.000000	13.650000	0.000000	4.000000	6.000000	0.000000	0.000000	0.000000	
75%	3.492712e+06	7.000000	47.450000	0.000000	4.000000	33.000000	1.000000	1.000000	3.000000	
max	4.108889e+06	50.000000	841.916667	6.000000	4.000000	193.000000	50.000000	42.000000	59.000000	

Рис. 3.7. Опис даних після очищення від викидів

Спочатку, було створено змінну з інформацією про стопвці, що мають бути нормалізовані. Також створена копія вихідних даних. Далі, дані були відфільтровані для кожного з релевантних стовпців, вказаних у змінній – “list_to_filter” та надан опис оброблених даних за допомогою методу .describe().

Кількість рядків скоротилась до 836427, тобто ми втратили трохи меньше 10 відсотків даних при очищенні. Оброблені дані вже не містять викидів, та можуть бути використані для нормалізації для подальшого моделювання. Нормалізація зображена на рис. 3.8.

```
min_max_scaler = preprocessing.MinMaxScaler()
normalize_list = ['NUM_OF_SES', 'DURATION_IN_MINUTES', 'event_1',
'event_2', 'event_3', 'event_4', 'event_5']
```

```
for col in normalize_list:
    filtered[col] = min_max_scaler.fit_transform(np.array(filtered[col]).reshape(-
1,1))
```

```
filtered.head()
```

	ID	NUM_OF_SES	DURATION_IN_MINUTES	LOCALE	PLATFORM	event_1	event_2	event_3	event_4	event_5	converted
1	3746483	0.000000	0.003563	3	2.0	0.005181	0.10	0.000000	0.000000	0.000000	1
2	216998	0.040816	0.000099	0	1.0	0.000000	0.00	0.000000	0.000000	0.000000	1
4	3883731	0.040816	0.037989	3	1.0	0.150259	0.04	0.047619	0.000000	0.000000	1
5	2282093	0.040816	0.011007	0	0.0	0.191710	0.00	0.000000	0.118644	0.473684	1
6	4028671	0.714286	0.084133	0	4.0	0.196891	0.06	0.023810	0.000000	0.000000	1

Рис. 3.8. Нормалізація даних для використання у моделі

Даний код використовує метод масштабування Min-Max для нормалізації числових змінних у наборі даних. Спочатку створюється екземпляр класу `MinMaxScaler` з бібліотеки `preprocessing`, який буде використовуватися для виконання цього виду масштабування. Потім визначається список змінних, які потрібно нормалізувати.

Використовується цикл для ітерації по кожній змінній у вказаному списку. Для кожної змінної виконується масштабування Min-Max, використовуючи метод `fit_transform` екземпляра `MinMaxScaler`. Змінні перетворюються, щоб мати форму вектора (одновимірного масиву) перед застосуванням масштабування. Змінні у наборі даних оновлюються змасштабованими значеннями. Нарешті, виводяться перші рядки оновленого набору даних, використовуючи метод `head()`.

Отже, цей код реалізує масштабування Min-Max для вказаних числових змінних з метою приведення їхніх значень до діапазону від 0 до 1, що може покращити стабільність та ефективність деяких моделей машинного навчання.

3.3. Побудова моделей прогнозування стану поведінки споживачів за індикаторами конверсії

У попередніх розділах дипломної роботи ми зосередилися на етапах екстракції та підготовки даних, необхідних для подальшого використання в різноманітних моделях класифікації. Зараз ми переходимо до етапу самої класифікації, в якому використовуються отримані раніше дані для прогнозування категорій чи класів об'єктів дослідження. Класифікація є ключовим етапом аналізу даних, який спрямований на визначення певних закономірностей та взаємозв'язків між об'єктами дослідження. Наш підхід до класифікації ґрунтується на ретельному відборі та підготовці ознак, що раніше були витягнуті з вихідних даних.

У даному розділі ми детально розглянемо процес класифікації та використані моделі для досягнення поставлених цілей. Відзначимо основні кроки, включаючи вибір класифікаторів, побудову моделей та їхню налаштування. Також розглянемо важливі метрики ефективності класифікації, які дозволять об'єктивно оцінити результати нашої роботи. Наступний рис. 3.9 демонструє ряд дій у контексті бінарної класифікації.

```
y = filtered['converted'].fillna(0)
X = filtered.iloc[:,1:9].fillna(0)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
models = {}
```

```
models['Logistic Regression'] = LogisticRegression(max_iter=4000)
models['Support Vector Machines'] = LinearSVC(max_iter=4000)
models['Decision Trees'] = DecisionTreeClassifier()
models['Random Forest'] = RandomForestClassifier()
models['Naive Bayes'] = GaussianNB()
models['K-Nearest Neighbor'] = KNeighborsClassifier()
models['Gradient Boosting'] = GradientBoostingClassifier()
```

```
true_positive, false_positive, true_negative, false_negative, accuracy = {}, {}, {}, {}, {}
```

Рис 3.9 Створення тренувальних та тестових сетів, підготовка словника моделей

Спочатку заповнено пропущені значення в стовпці 'converted' нулями та з нього створено новий вектор y. Далі, створено матрицю ознак X з обраних стовпців (з другого по дев'ятий) і заповнено пропущені значення нулями. Потім, використовуючи функцію train_test_split, дані розділяються на навчальний і тестовий набори, при цьому розмір тестового набору складає 20% від загальної кількості даних, що є розповсюдженою практикою. Далі, створюється словник models, де ключі – це назви різних моделей класифікації, а значення - екземпляри цих моделей, такі як логістична регресія, метод опорних векторів, дерева рішень, випадковий ліс, наївний байєсівський класифікатор, метод k-найближчих сусідів та градієнтний бустінг.

Завершується код створення порожніх словників для розрахунку метрик ефективності моделей, таких як true positive, false positive, true negative, false negative та accuracy. Навчання моделей продемонстровано на рис. 3.10:

```
for key in models.keys():
    models[key].fit(X_train, y_train)
    predictions = models[key].predict(X_test)
    true_negative[key], false_positive[key], false_negative[key], true_positive[key] = confusion_matrix(y_test, predictions).rave
    accuracy[key] = (true_positive[key]+true_negative[key]) / (true_positive[key]+false_positive[key]+true_negative[key]+false_ne
```

Рис. 3.10. Навчання моделей

Цей код взаємодіє з раніше ініціалізованими моделями, які розміщені у словнику `models`. Для кожної моделі проводяться такі етапи. Використовуючи навчальний набір (X_train і y_train), модель навчається за допомогою методу `fit`. Після цього, застосовуючи навчену модель, виконуються прогнози на тестовому наборі (X_test) за допомогою методу `predict`. З використанням функції `confusion_matrix` з бібліотеки scikit-learn обчислюються елементи матриці плутанини, такі як true negative, false positive, false negative, true positive. Отримані значення параметрів заносяться у відповідні словники для кожної моделі. Також для кожної моделі обчислюється точність, яка

визначається як відношення суми правильно класифікованих екземплярів (true positive та true negative) до загальної кількості екземплярів у тестовому наборі.

Таким чином, цей код дозволяє ітеруватися через кожну модель, проводити їх навчання на навчальному наборі, робити прогнози на тестовому наборі, обчислювати матрицю плутанини та ряд метрик ефективності для кожної моделі, включаючи точність. Результати класифікації представлено на рис. 3.11:

```
df_model = pd.DataFrame(index=models.keys(), columns=['True Positive', 'False Positive', 'True Negative', 'False Negative', 'Accu
df_model['True Positive'] = true_positive.values()
df_model['False Positive'] = false_positive.values()
df_model['True Negative'] = true_negative.values()
df_model['False Negative'] = false_negative.values()
df_model['Accuracy'] = accuracy.values()
```

```
df_model
```

	True Positive	False Positive	True Negative	False Negative	Accuracy
Logistic Regression	104847	75399	91645	62680	0.587295
Support Vector Machines	105427	76066	90978	62100	0.587035
Decision Trees	100513	70689	96355	67014	0.588419
Random Forest	105807	63922	103122	61720	0.624468
Naive Bayes	119267	83208	83836	48260	0.607055
K-Nearest Neighbor	103527	64873	102171	64000	0.614811
Gradient Boosting	114293	61214	105830	53234	0.657926

Рис. 3.11. Результати навчених моделей у класифікації

Цей код створює новий об'єкт DataFrame у бібліотеці pandas, призначений для структурування та аналізу даних у табличній формі. Кожен рядок коду виконує наступне:

Створення DataFrame: df_model = pd.DataFrame(index=models.keys(), columns=['True Positive', 'False Positive', 'True Negative', 'False Negative', 'Accuracy']). Тут створюється DataFrame з індексами, які взяті з ключів словника models, та стовпцями, що представляють різні метрики ефективності моделей.

Заповнення DataFrame зі значеннями метрик: df_model['True Positive'] = true_positive.values(), df_model['False Positive'] = false_positive.values(), df_model['True Negative'] = true_negative.values(), df_model['False Negative'] =

false_negative.values(), df_model['Accuracy'] = accuracy.values(). Тут кожен стовпець 'True Positive', 'False Positive', 'True Negative', 'False Negative' і 'Accuracy' заповнюється відповідними значеннями з відповідних словників.

Виведення DataFrame: df_model. Цей рядок повертає або виводить створений DataFrame, де рядки відповідають різним моделям, а стовпці містять значення метрик ефективності для кожної моделі. Також було проведено додаткове навчання моделі GradientBoosting з попередньо підібраними параметрами (рис. 3.12).

```
mod = GradientBoostingClassifier(n_estimators=200)
mod.fit(X_train, y_train)
pr1 = mod.predict(X_test)
cm = confusion_matrix(y_test, pr1)
```

```
TN, FP, FN, TP = confusion_matrix(y_test, pr1).ravel()
accuracy1 = (TP+TN) /(TP+FP+TN+FN)
print('True Positive(TP) = ', TP)
print('False Positive(FP) = ', FP)
print('True Negative(TN) = ', TN)
print('False Negative(FN) = ', FN)
print('Accuracy of the binary classification = {:0.3f}'.format(accuracy1))
```

```
True Positive(TP) =  56649
False Positive(FP) =  29925
True Negative(TN) =  53831
False Negative(FN) =  26881
Accuracy of the binary classification = 0.660
```

Рис. 3.12. Навчання моделі GradientBoosting та її результат

Було використано модель `GradientBoostingClassifier` з фіксованими параметрами (`n_estimators=200`) для бінарної класифікації на навчальному наборі `X_train` та `y_train`. Після навчання модель використовується для прогнозу на тестовому наборі `X_test`, і результати порівнюються з реальними мітками `y_test`.

Далі в коді використовується матриця плутанини (`confusion_matrix`) для обчислення різних метрик ефективності бінарної класифікації. Точність бінарної класифікації (accuracy обчислюється) за допомогою формули:

$$\frac{TP + TN}{TP + FP + TN + FN}$$

де TP – кількість екземплярів, які було правильно класифіковано;

FP – кількість екземплярів, які належать негативному класу, але були неправильно класифіковані як позитивні.

TN – кількість екземплярів, які належать негативному класу і були правильно класифіковані як негативні.

FN – кількість екземплярів, які належать позитивному класу, але були неправильно класифіковані як негативні.

В кінці коду виводяться значення TP, FP, TN, FN та точність (`accuracy1`). Точність моделі зросла порівняно з попереднім результат на 0.3 відсотки. Здобуті результати свідчать про ефективність використання даної моделі у передбаченні того, скільки користувачів здійснять покупку на основі їхнього взаємодії з додатком. Це важливо з точки зору підвищення рівня персоналізації та оптимізації маркетингових стратегій.

Використання моделі GradientBoosting дозволяє здійснювати прогнози, спрямовані на підвищення конверсії та оптимізацію рекламних кампаній. За допомогою цієї моделі можна ідентифікувати та залучати цільову аудиторію, що збільшує ймовірність успішних транзакцій. Такий підхід сприяє підвищенню ефективності бізнесу та покращенню стратегій взаємодії з користувачами у сфері електронної комерції.

ВИСНОВОК

Дипломна робота спрямована на вивчення та застосування методів машинного навчання в різних сферах людського життя, бізнесу та науки. Здійснено аналіз методів классифікації, кластеризації та регресії, а також проведено етапи витягування та обробки даних з використанням хмарного сховища Snowflake та мови SQL, а також мови програмування Python.

Як було визначено, метою дипломної роботи стала розробка комлексу моделей ідентифікації та прогнозування стану поведінки споживачів поведінки на базі методів машинного навчання, що дозволить оцінити споживацькі уподобання та підвищити якість управлінських рішень щодо маркетингових стратегій просування продукту.

Для досягення поставленої мети було успішно виконано низку завдань: проаналізовано особливості поведінки споживача; проаналізовано перспективи та особливості машинного навчання для моделювання поведінки споживачів; здійснено аналіз інструментальних засобів машинного навчання, зокрема, мови Python, робочого середовища Jupyter notebook, хмарного сховища Snowflake та мови SQL до моделювання споживацької поведінки; розроблено моделі прогнозування стану поведінки споживачів за індикаторами конверсії методами машинного навчання.

В рамках роботи були побудовані сім різних моделей машинного навчання, їх порівняння та відбір оптимальної. Отримана краща модель, зокрема модель GradientBoosting, була застосована для створення класифікатора, призначеного визначати ймовірність того, чи здійснить користувач покупку у подальшому.

Застосування такого класифікатора може виявитися вкрай корисним у сучасному бізнес-середовищі. Бінарний класифікатор дозволяє ідентифікувати користувачів, які мають високу ймовірність здійснення покупки. Це сприяє зосередженню маркетингових та рекламних зусиль на

цільових аудиторіях, що може призвести до збільшення конверсії та ефективності рекламних кампаній.

Створена модель допомагає персоналізувати взаємодію з користувачами, надаючи індивідуалізовані пропозиції та послуги, що відповідають їхнім потребам та інтересам. Ідентифікація користувачів, які ймовірно здійснять покупку, дозволяє оптимізувати витрати на рекламу та маркетинг, спрямовуючи їх на ті групи, які мають найбільший потенціал. Також, застосування моделі класифікатора сприяє покращенню користувацького досвіду через надання персоналізованих рекомендацій та послуг, що відповідають індивідуальним потребам користувачів.

Отримана модель може бути використана для прогнозування та стратегічного планування розвитку бізнесу, дозволяючи компаніям ефективно реагувати на зміни в ринкових умовах та підлаштовувати свої стратегії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Machine learning [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Machine_learning.

2. Python about [Електронний ресурс]. – Режим доступу: https://python.org.

3. Jupyter project about / [Електронний ресурс]. – Режим доступу: https://jupyter.org/.

4. What is SQL? [Електронний ресурс]. – Режим доступу: https://searchdatamanagement.techtarget.com/definition/SQL.

5. BigQuery [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/BigQuery.

6. NumPy [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/NumPy.

7. What is Numpy? [Електронний ресурс]. – Режим доступу: https://numpy.org/devdocs/user/whatisnumpy.html.

8. pandas - Python Data Analysis Library [Електронний ресурс]. – Режим доступу: https://pandas.pydata.org.

9. Pandas [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Pandas

10. Pandas – Package overview [Електронний ресурс]. – Режим доступу:https://pandas.pydata.org/pandasdocs/stable/getting_started/overview.html.

11. Гуляк Р. Е. Конспект лекций в схемах и таблицах по дисциплине «Поведение потребителя» Харьк. нац. акад. гор. хоз-ва. – Харків : ХНАГХ, 2013. – 119 с.

12. Прокопенко О.В. Поведінка споживачів: Навчальний посібник. / О.В. Прокопенко, М.Ю. Троян – Київ : Центр учбової літератури, 2008. 176 с.

13. Гут І.О. Сучасні методи дослідження поведінки споживачів // Академічний огляд. 2001. № 1. С. 32 – 37.

14. Иванова Р. Х. Поведение потребителей : учеб. пособие / Р. Х. Иванова. – Харьков : ИД «ИНЖЭК». – 2003. – 120 с.

15. Шафалюк О. К. Поведінка споживачів: курс лекцій / О. К. Шафалюк. – Київ : КНЕУ, 2003. – 68 с

16. Зарицька О.Л. Бази даних та інформаційні системи: Методичний посібник. – Житомир: Вид-во ЖДУ ім. І. Франка, 2009. – 132 с.

17. Коломейчук В. В. Розробка та дослідження бази даних для систем обробки статистичної інформації. Математичні машини и системи, 1(4), (2009). С. 89 – 95.

18. Cloud Database [Електронний ресурс] // SearchCloudApplications – Режим доступу: https://searchcloudapplications.techtarget.com/definition/cloud-database.

19. Snowflake key concepts. [Електронний ресурс] – Режим доступу: https://docs.snowflake.com/en/user-guide/intro-key-concepts.

20. Бібліотеки Python, необхідні для машинного навчання [Електронний ресурс] – Режим доступу: https://techrocks.ru/2018/10/05/python-libraries-for-machine-learning.

21. Сучасні бібліотеки машинного навчання [Електронний ресурс] – Режим доступу: https://travelscode.com/suchasni-biblioteki-mashinnogo-navchannya.

22. Робота з даними по-новому: Pandas замість SQL [Електронний ресурс] – Режим доступу: https://tproger.ru/translations/rewrite-sql-queries-in-pandas.

23. Что такое Jupyter-ноутбук и как его использовать [Електронний ресурс] – Режим доступу: https://highload.today/jupyter-notebook.

24. Як впливає машинне навчання на бізнес-процеси [Електронний ресурс] – Режим доступу: https://biz.nv.ua/ukr/experts/jak-vplivaje-mashinne-navchannja-na-biznes-protsesi--2458155.html.

ДОДАТКИ

Додаток А

Запит до хмарного сховища Snowflake

```
with unconverted as (
select user_id, listagg(buys_fact.purchase_type)
from buys_fact
group by user_id
having listagg(buys_fact.purchase_type) ilike '%subscription%'
),
trials as (
select buys_fact.user_id, min(fulldate) as trial_date
from buys_fact
inner join unconverted on unconverted.user_id = buys_fact.user_id
left join dates_dim on dates_dim.id = buys_fact.created_date_id
where buys_fact.purchase_type = 'trial'
group by buys_fact.user_id
),
session_num as (
select trials.user_id, count(*) as num_of_ses from trials inner join sessions_fact on sessions_fact.user_id = trials.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
group by trials.user_id
),

session_duration as (
select trials.user_id, sum(duration) / 60 duration_in_minutes from trials inner join sessions_fact on sessions_fact.user_id = trials.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
group by trials.user_id
```

```
),

enable_system_layer as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'enable_system_layer'
group by trials.user_id
),

search_item_selected as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'search_item_selected'
group by trials.user_id
),

search_opened as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'search_opened'
group by trials.user_id
),

infobox_item_selected as (
select trials.user_id, count(*) as num from events_fact
```

```
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'infobox_item_selected'
group by trials.user_id
),

infobox_menu_item_selected as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'infobox_menu_item_selected'
group by trials.user_id
)

select  users_dim.id,
        session_num.num_of_ses,
        session_duration.duration_in_minutes,
        case users_dim.locale when 'en' then 0 when 'zh-hans' then 1 when '' then 2 when 'fr' then 3 when 'es' then 4 when 'zh' then 5 when 'de' then 6 end as locale,
        case platform when 'Android' then 0 when 'Mac OS' then 1 when 'Windows' then 2 when null then 3 when 'iOS' then 4 end as platform,
        coalesce(enable_system_layer.num, 0) as "event_1",
        coalesce(search_item_selected.num, 0) as "event_2",
        coalesce(search_opened.num, 0) as "event_3",
        coalesce(infobox_item_selected.num, 0) as "event_4",
        coalesce(infobox_menu_item_selected.num, 0) as "event_5",
        1 as "converted"
from users_dim
inner join session_num on session_num.user_id = users_dim.id
```

```
inner join session_duration on session_duration.user_id = users_dim.id
left join enable_system_layer on enable_system_layer.user_id = users_dim.id
left join search_item_selected on search_item_selected.user_id = users_dim.id
left join search_opened on search_opened.user_id = users_dim.id
left join infobox_item_selected on infobox_item_selected.user_id = users_dim.id
left join infobox_menu_item_selected on infobox_menu_item_selected.user_id = users_dim.id
union
with unconverted as (
select user_id, listagg(buys_fact.purchase_type)
from buys_fact
group by user_id
having listagg(buys_fact.purchase_type) = 'trial'
),
trials as (
select buys_fact.user_id, min(fulldate) as trial_date
from buys_fact
inner join unconverted on unconverted.user_id = buys_fact.user_id
left join dates_dim on dates_dim.id = buys_fact.created_date_id
group by buys_fact.user_id
),
session_num as (
select trials.user_id, count(*) as num_of_ses from trials inner join sessions_fact on sessions_fact.user_id = trials.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
group by trials.user_id
),
session_duration as (
```

```
select trials.user_id, sum(duration) / 60 duration_in_minutes from trials inner join sessions_fact on sessions_fact.user_id = trials.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
group by trials.user_id
),
enable_system_layer as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'enable_system_layer'
group by trials.user_id
),
search_item_selected as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'search_item_selected'
group by trials.user_id
),
search_opened as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'search_opened'
group by trials.user_id
),
infobox_item_selected as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
```

```
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'infobox_item_selected'
group by trials.user_id
),
infobox_menu_item_selected as (
select trials.user_id, count(*) as num from events_fact
inner join trials on trials.user_id = events_fact.user_id
where event_timestamp <= DATEADD(day, 3, trials.trial_date)
and event_type = 'infobox_menu_item_selected'
group by trials.user_id
)
select  users_dim.id,
        session_num.num_of_ses,
        session_duration.duration_in_minutes,
        case users_dim.locale when 'en' then 0 when 'zh-hans' then 1 when '' then 2 when 'fr' then 3 when 'es' then 4 when 'zh' then 5 when 'de' then 6 end as locale,
        case platform when 'Android' then 0 when 'Mac OS' then 1 when 'Windows' then 2 when null then 3 when 'iOS' then 4 end as platform,
        coalesce(enable_system_layer.num, 0) as "event_1",
        coalesce(search_item_selected.num, 0) as "event_2",
        coalesce(search_opened.num, 0) as "event_3",
        coalesce(infobox_item_selected.num, 0) as "event_4",
        coalesce(infobox_menu_item_selected.num, 0) as "event_5",
        0 as "converted"
from users_dim
inner join session_num on session_num.user_id = users_dim.id
inner join session_duration on session_duration.user_id = users_dim.id
left join enable_system_layer on enable_system_layer.user_id = users_dim.id
left join search_item_selected on search_item_selected.user_id = users_dim.id
```

```
left join search_opened on search_opened.user_id = users_dim.id

left join infobox_item_selected on infobox_item_selected.user_id = users_dim.id

left join infobox_menu_item_selected on infobox_menu_item_selected.user_id = users_dim.id
```

Додаток Б.
Повна версія Python коду

```
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import LinearSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV

file_path = 'C:/Users/myronenko.a/JupiterWorkingDirectory/Data/diplomdata.csv'
data = pd.read_csv(file_path, sep=',', header=0)

data.head()
data['event_1'] = data['event_1'].fillna(0)
data['event_2'] = data['event_2'].fillna(0)
data['event_3'] = data['event_3'].fillna(0)
data['event_4'] = data['event_4'].fillna(0)
data['event_5'] = data['event_5'].fillna(0)
data.describe()
list_to_filter = ['NUM_OF_SES', 'DURATION_IN_MINUTES', 'event_1',
'event_2', 'event_3', 'event_4', 'event_5']
filtered = data.copy()
for col in list_to_filter:
```

```
    high = data[col].quantile(0.98)
    filtered = filtered[(filtered[col] < high)]
filtered.describe()
min_max_scaler = preprocessing.MinMaxScaler()
normalize_list = ['NUM_OF_SES', 'DURATION_IN_MINUTES', 'event_1',
'event_2', 'event_3', 'event_4', 'event_5']
for col in normalize_list:
     filtered[col] = min_max_scaler.fit_transform(np.array(filtered[col]).reshape(-
1,1))
filtered.head()
y = filtered['converted'].fillna(0)
X = filtered.iloc[:,1:9].fillna(0)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
models = {}
models['Logistic Regression'] = LogisticRegression(max_iter=4000)
models['Support Vector Machines'] = LinearSVC(max_iter=4000)
models['Decision Trees'] = DecisionTreeClassifier()
models['Random Forest'] = RandomForestClassifier()
models['Naive Bayes'] = GaussianNB()
models['K-Nearest Neighbor'] = KNeighborsClassifier()
models['Gradient Boosting'] = GradientBoostingClassifier()
true_positive, false_positive, true_negative, false_negative, accuracy = {}, {}, {},
{}, {}
for key in models.keys():
    models[key].fit(X_train, y_train)
    predictions = models[key].predict(X_test)
    true_negative[key], false_positive[key], false_negative[key], true_positive[key]
= confusion_matrix(y_test, predictions).ravel()
    accuracy[key] = (true_positive[key]+true_negative[key]) /
(true_positive[key]+false_positive[key]+true_negative[key]+false_negative[key])
```

```
df_model = pd.DataFrame(index=models.keys(), columns=['True Positive', 'False Positive', 'True Negative', 'False Negative', 'Accuracy'])
df_model['True Positive'] = true_positive.values()
df_model['False Positive'] = false_positive.values()
df_model['True Negative'] = true_negative.values()
df_model['False Negative'] = false_negative.values()
df_model['Accuracy'] = accuracy.values()
df_model
mod = GradientBoostingClassifier(n_estimators=200)
mod.fit(X_train, y_train)
pr1 = mod.predict(X_test)
cm = confusion_matrix(y_test, pr1)
TN, FP, FN, TP = confusion_matrix(y_test, pr1).ravel()
accuracy_1 = (TP+TN) /(TP+FP+TN+FN)
print('True Positive(TP) = ', TP)
print('False Positive(FP) = ', FP)
print('True Negative(TN) = ', TN)
print('False Negative(FN) = ', FN)
print('Accuracy of the binary classification = {:0.3f}'.format(accuracy_1))
```