МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІМ. Ю.М. ПОТЕБНІ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на тему **Використання алгоритмів машинного навчання для побудови системи прогнозування поведінки клієнтів**

Виконав: студент 2 курсу, групи 8.1212-іпз
спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)
освітньої програми Інженерія програмного забезпечення
(код і назва освітньої програми)
В.В. Голомб
(ініціали та прізвище)
Керівник доцент, А.І. Безверхий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)
Рецензент
директор ТОВ Алтер Віжн Груп В.С. Тряпічко
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра _Кафедра електроніки, інформаційних систем та програмного забезпечення

Рівень вищої освіти _____другий (магістерський)_____

Спеціальність ___121_Інженерія програмного забезпечення_____
(код та назва)

Освітня програма ___Інженерія програмного забезпечення
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри_____Тетяна Критська

«_01_» __вересня_____2023_року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ
В.В. Голомб
(прізвище, ім'я, по батькові)

1.Тема роботи __Використання алгоритмів машинного навчання для побудови системи прогнозування поведінки клієнтів_____

керівник роботи ___доцент, _____А.І. Безверхий _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ 1577-с від 09.10.2023 р.

2. Строк подання студентом кваліфікаційної роботи_____30.11.2023_____

3. Вихідні дані магістерської роботи

- комплект нормативних документів ;
- технічне завдання до роботи.

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

- огляд та збір літератури стосовно теми кваліфікаційної роботи;
- огляд та аналіз існуючих рішень та аналогів;
- дослідження проблеми розпізнавання мов та розробка методів її вирішення;
- створення програмного продукту та його опис;
- перелік вимог для роботи програми;
- дослідження поставленої проблеми та розробка висновків та пропозицій.

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

20 слайдів презентації

6. Консультанти розділів магістерської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата Завдання прийняв

7. Дата видачі завдання 01.09.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів магістерської роботи	Примітка
1	Аналіз предметної області	02.09-10.09.23	виконано
2	Формулювання основної задачі кваліфікаційної роботи та узгодження її з науковим керівником	11.09-12.09.23	виконано
3	Аналіз існуючих методів рішення	13.09-14.09.23	виконано
4	Дослідження проблеми подібності об’єктів в багатовимірному просторі	15.09-20.09.23	виконано
5	Дослідження алгоритмів/модулів для визначення подібності об’єктів в контексті класифікації	21.09-26.09.23	виконано
6	Узгодження подальших дій з науковим керівником	27.09-28.09.23	виконано
7	Пошук, дослідження тренувального набору даних та його попередня підготовка	29.09-13.10.23	виконано
8	Розробка алгоритму класифікації за допомогою ансамблю багатошарових нейронних мереж	14.10-16.10.23	виконано
9	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	17.10-19.10.23	виконано
10	Реалізація функціоналу і користувацького інтерфейсу застосунку прогнозування поведінки клієнтів	20.10-17.11.23	виконано
11	Порівняльний аналіз результатів роботи ML-алгоритмів в системі прогнозування поведінки клієнтів	18.11-22.11.23	виконано
12	Оформлення пояснювальної записки	23.11-28.11.23	виконано

Студент _____ Голомб В.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Безверхий А.І.
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ Скрипник І.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок: 102

Рисунків: 27

Таблиць: 1

Джерел: 32

Голомб В.В. Використання алгоритмів машинного навчання для побудови системи прогнозування поведінки клієнтів : кваліфікаційна робота магістра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник А. І. Безверхий. Запоріжжя : ЗНУ, 2023. 99 с.

Мета і завдання дослідження полягають в аналізі алгоритмів класифікації структурованих даних та розробці високоефективної інформаційно-аналітичної системи для оптимізації стратегій перехресних продажів у фінансовому секторі з використанням передових методів машинного та глибокого навчання в умовах обмежено контрольованого середовища з великою кількістю нерозмічених спостережень.

В ході дослідження запропоновано метод, який використовує для PU – класифікації просту одношарову нейронну мережу. У цьому методі будується ансамбль із декількох одношарових нейронних мереж, кожну із яких тренують на різних невеликих випадкових вибірках із основного набору даних. Цей підхід дозволяє обмежити кількість невідомих об’єктів, які мережа бачить як умовно-негативний клас (0), що сприяє зменшенню ймовірності помилкового маркування невідомих (нерозмічених) об’єктів як помилково негативних (1). Фінальний прогноз класу для об'єкта формується як середнє значення прогнозів всіх мереж. Цей метод покладено в основу веб-застосунку для фінансових компаній із інтуїтивним інтерфейсом для менеджерів та агентів, який дозволяє їм ефективно вибирати клієнтів для перехресних продажів.

Ключові слова: бінарна класифікація, веб-фреймворк, відкриті дані, напівконтрольоване машинне навчання, нейронна мережа, перехресні продажі.

SUMMARY

Pages: 102

Figures: 27

Tables: 1

Sources: 32

Holomb Volodymyr. Using Machine Learning Algorithms to Build a System for Predicting Customer Behavior: Master's Thesis in Speciality 121 «Software Engineering» / Supervised by A.I. Bezverkhyi. Zaporizhzhya: ZNU, 2023. 99 pages.

The primary objective of this research is to explore various algorithms for the classification of structured data as well as develop an advanced information and analytical system for optimizing cross-selling strategies within the financial sector with the help of semi-supervised machine and deep learning techniques.

In this study, we propose an innovative method utilizing a single-layer neural network to address positive-unlabelled (PU) classification in scenarios involving a substantial number of unlabelled observations. The proposed approach involves the creation of an ensemble of multiple single-layer neural networks. Each network is individually trained on a distinct randomly selected small sample from the main dataset. This methodology effectively restricts the introduction of unlabelled samples to the network during training, treating them as pseudo-negative class (0). The primary aim is to significantly reduce the false negative rate (1). The ultimate class prediction for an object is determined by aggregating the mean predictions of all networks.

This technique has been implemented as a core component of a single-page application designed for financial companies with a user-friendly interface tailored for managers and agents, facilitating the efficient identification of leads customers for cross-selling purposes.

Keywords: binary classification, web framework, open data, semi-supervised learning, neural network, cross-selling.

ЗМІСТ

ВСТУП

Актуальність теми

Класифікація, як один із ключових аспектів аналізу даних, є необхідним інструментом у сфері маркетингу, особливо при прогнозуванні поведінки клієнтів. У зв'язку з тим, що великі обсяги даних про клієнтів неможливо аналізувати вручну, використання алгоритмів машинного навчання для класифікації стає ключовим елементом в розвитку систем прогнозування. Задача класифікації полягає в розбитті клієнтської бази на групи або класи за певними атрибутами, що дозволяє визначити спільні риси та властивості класів.

Для успішного впровадження класифікаційних алгоритмів у сфері фінансових послуг важливо враховувати особливості цього сектору. На відміну від епізодичного характеру контактів у роздрібній торгівлі, взаємодія з клієнтами фінансових компаній частіше є довгостроковою та конструктивною. Це вимагає особливого підходу до аналізу клієнтської бази та можливості прогнозування поведінки клієнтів на тривалий період. У цьому контексті, алгоритми машинного навчання, зокрема методи класифікації, стають необхідним інструментом для ефективного управління та розвитку клієнтської бази фінансових установ.

Підходи до класифікації можуть включати бінарні або багатокласові моделі в залежності від конкретних завдань. Наприклад, у фінансовому секторі може бути важливим розділення клієнтів на ті, які мають високий та низький рівень кредитоспроможності. Такі моделі дозволяють автоматично призначати клієнтам відповідні категорії, спрощуючи процес прийняття рішень та покращуючи точність прогнозування.

Важливим аспектом вдосконалення методів класифікації в фінансовому секторі є використання напівконтрольованого навчання. Цей підхід, що використовує як позначені, так і непозначені дані для навчання моделі, дозволяє покращити якість класифікації за рахунок ефективного використання обмеже-

них позначених даних та обширного обсягу непозначених даних. У фінансовому секторі, де доступ до позначених даних може бути обмеженим, цей підхід стає особливо цінним для вдосконалення класифікаційних моделей.

Перехресні продажі в сфері фінансових послуг визначаються як стратегічний метод, що передбачає пропозицію клієнтам пов'язаних чи додаткових продуктів. У контексті банківської сфери, де вже наявні довгострокові взаємини з клієнтами, перехресні продажі набувають особливої ефективності. Наприклад, якщо клієнт має іпотеку, відділ продажів може пропонувати йому особисту кредитну лінію чи інші спеціалізовані фінансові продукти.

Фактично, перехресні продажі стають ключовим методом отримання нових доходів для фінансових компаній. Однак важливо враховувати, що не всі компанії ефективно використовують цей метод, зокрема, в секторі страхування, де лише кожна п'ята компанія успішно застосовує стратегію перехресних продаж. Ця недостатня ефективність пов'язана з викликами, такими як складність визначення цільової аудиторії та продуктів для пропозиції.

На шляху вдосконалення стратегій перехресних продаж важливим кроком є використання демо-застосунків та MVP проектів. Ці інструменти дозволяють замовнику оцінити потенціал рішення, попередньо визначити його корисність та винести необхідні корективи до вимог. Використання Python-фреймворків для швидкої розробки демо-застосунків дозволяє ефективно створювати прототипи аналітичних рішень, покращуючи процес впровадження стратегій перехресних продаж.

Таким чином, актуальною є потреба для аналітичних компаній ку швидкій розробці рішень, які б надавали можливість фінансовим компаніям використовувати складні моделі машинного та глибокого навчання. Такі рішення можуть мати спрощений до кількох кроків інтерфейс взаємодії із користувачем, доступний для менеджерів з продажу та агентів, які не мають відповідного технічного і аналітичним досвіду.

Мета і завдання дослідження

Метою даного дослідження є аналіз алгоритмів класифікації структурованих даних та розробка високоефективної інформаційно-аналітичної системи для оптимізації стратегій перехресних продажів у фінансовому секторі з використанням передових методів машинного та глибокого навчання в умовах обмежено контрольованого середовища з великою кількістю нерозмічених спостережень.

Об’єкт дослідження

Об'єктом дослідження є структуровані дані з атрибутами об’єктів для визначення спільних характеристик та властивостей класифікаційних груп.

Предмет дослідження

Предметом дослідження є методи визначення ступеня схожості некласифікованих (нерозмічених) об’єктів за сукупністю атрибутів до розмічених об’єктів (наприклад, клієнтів, які вже здійснили покупку), в контексті розробки системи прогнозування поведінки клієнтів для оптимізації стратегій перехресних продажів.

Методи дослідження

Для вирішення поставленої задачі використовуються наступні методи дослідження:

1. Аналіз особливостей та існуючих рішень проблеми класифікації в умовах обмежено контрольованого навчання.
2. Аналіз передових алгоритмів машинного і глибокого навчання для класифікації структурованих даних.
3. Аналіз Python-бібліотек машинного і глибокого навчання.
4. Аналіз відкритих датасетів із фінансового сектору.

5. Експериментальне застосування статистичних і ML / DL методів для вимірювання відстаней / визначення ступеню схожості векторів атрибутів об’єктів.

6. Аналіз Python-фреймворків для побудови повнофункціональних інтерактивних веб-застосунків із використанням ML/DL моделей.

Наукова новизна одержаних результатів

Наукова новизна отриманих результатів полягає в використанні ансамблю нейронних мереж для вирішення завдання визначення подібності нерозмічених клієнтів за їх атрибутами до клієнтів, які вже здійснили покупку (розмічених клієнтів). Кожна нейронна мережа в ансамблі побудована і навчена на невеликих випадкових наборах даних. Кожен з цих наборів даних включає 90% відомих позитивних (розмічених) спостережень та випадково вибраних умовно-негативних (нерозмічених) спостережень. Такий стратегічний підхід гарантує, що кожна нейронна мережа навчається на невеликому обсягу умовно-негативних (нерозмічених) спостережень, які вважаються негативним класом, зменшуючи ймовірність помилкового прогнозування нерозмічених клієнтів як негативних.

Практичне значення одержаних результатів

Практичне значення отриманих результатів дослідження полягає у впровадженні інноваційного підходу до ефективної PU - класифікації об'єктів за допомогою ансамблю нейронних мереж. Цей підхід успішно імплементовано у вигляді інтерактивної системи прогнозування поведінки клієнтів для перехресних продажів, яка має інтуїтивно зрозумілий користувацький інтерфейс. Взаємодія із системою спрощена до кількох етапів, включаючи завантаження файлу даних клієнтів, вибір цільової аудиторії (за визначеною колонкою із цільовою міткою), оцінку прийнятності точності прогнозування та отримання переліку потенційних клієнтів для конкретного фінансового продукту чи послуги. Така система не лише надає високий рівень точності класифікації, а й

забезпечує ефективність та зручність в користуванні для працівників компаній, що впроваджують стратегії перехресних продажів.

Апробація одержаних результатів

Результати дослідження представлені на:

- III Всеукраїнської науково-практичної конференції за участю молодих науковців «Актуальні питання сталого науково-технічного та соціально-економічного розвитку регіонів України» ЗНУ [26];
- конференції «Молода наука-2023» ЗНУ [12];
- Міжнародній науково-практичній конференції «Геостратегічна трансформації та траєкторія національної безпеки в контексті відбудови і сталого розвитку України» ЗНУ [25]
- веб-порталі аналітичної компанії RBC Group у форматі SPA-застосунку для прогнозування поведінки клієнтів для перехресних продажів, доступного за адресою https://www.rbcgrp.com/en/smart-cross-sales.

Перелік використаних скорочень

API - App Programming Interface

BI - Business Intelligence

CI/CD - Continuous Integration / Continuous Deployment

CRM - Customer Relationship Management

DB / БД – DataBase / База даних

DBMS - DB Management Systems

Dev – розробницьке (середовище)

DL - Deep Learning

ERP - Enterprise Resource Planning

FSS - File Sync and Share

FTP(S) - File Transfer Protocol (Secure)

LSTM – нейронні мережі на принципах LSTM (Long Short-Term Memory)

ML / МН – машинне навчання / machine learning

ODBC - Open DB Connectivity

PU (Learning) – Positive-Unlabelled (Learning)

RDP - Remote Desktop Protocol

RnD – Research and Development

SCS – Smart Cross Selling

SFTP - Secure FTP

SMOTE - Synthetic Minority Oversampling Technique

SPA – Sinle Page Application

SSH - Secure Shell

SSL – Semi-Supervised Learning / напівконтрольоване навчання

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРОБЛЕМИ ПОДІБНОСТІ ОБ'ЄКТІВ В БАГАТОВИМІРНОМУ ПРОСТОРІ

1.1 Задача класифікації як спосіб розбиття множини об'єктів на групи за ознаками подібності

1.1.1 Опис процесу і методи класифікації

Класифікація, у широкому розумінні, представляє собою завдання групування об'єктів чи спостережень в апріорно визначені категорії, що називаються класами. Об'єкти всередині кожного класу вважаються схожими один на одного та мають приблизно ідентичні властивості та ознаки. При цьому прийняття рішення базується на аналізі значень атрибутів (ознак). Застосування класифікації розглядається в контексті маркетингу для оцінки кредитоспроможності позичальників, визначення лояльності клієнтів, розпізнавання образів, медичній діагностиці та інших областях.

Якщо відомі аналітичні характеристики об'єктів кожного класу, нове спостереження, що відноситься до певного класу, автоматично набуває властивостей цього класу. У випадку обмеження до двох класів маємо справу з бінарною класифікацією, яку можна зведення до більш складних завдань [28].

Математичний опис об'єкта, яким можна оперувати для проведення класифікації за допомогою математичних методів, найчастіше представляє собою базу даних. Кожен запис бази даних містить інформацію про певну властивість об'єкта.

Важливим етапом є розділення набору вхідних даних на навчальну та тестову вибірки Навчальна вибірка включає об'єкти, для яких відомі значення як незалежних, так і залежних змінних. Модель для визначення значення залежної змінної будується на основі навчальної вибірки, часто називаної функцією класифікації.

Для отримання точної функції класифікації важливо враховувати основні вимоги до навчальної вибірки:

- кількість об'єктів у вибірці повинна бути достатньою, щоб побудована на їх основі функція класифікації була точною.
- вибірка повинна включати об'єкти, що представляють всі можливі класи.
- для кожного класу вибірка повинна включати достатню кількість об'єктів.

Тестова множина, включаючи незалежні і залежні змінні, використовується для верифікації ефективності побудованої моделі.

Процес класифікації включає два основних етапи:

1. Побудова моделі: здійснюється на основі навчальної вибірки, і результатом є модель, представлена класифікаційними правилами, деревом рішень, математичною формулою або комп'ютерним об'єктом, таким як нейронна мережа.

2. Використання моделі: полягає у класифікації нових або невідомих значень.

Оцінка правильності моделі виконується порівнянням відомих значень з тестової множини з результатами моделі. Точність вимірюється як відсоток правильно класифікованих прикладів у тестовій множині.

Умови використання моделі: якщо точність моделі задовільна, вона може бути використана для класифікації нових прикладів, клас яких невідомий.

Основні проблеми, що виникають при розв'язанні завдань класифікації, включають низьку якість вхідних даних, присутність помилкових та пропущених значень, різні типи атрибутів (числові та категоріальні) та їх різну значимість, а також проблеми «перенавчання» та «недонавчання» [32].

Проблема «перенавчання» виникає, коли класифікаційна функція занадто точно адаптується до навчальних даних, і намагається інтерпретувати

помилки та аномалії як внутрішню структуру даних. Це може призвести до некоректної роботи моделі на нових даних.

Термін «недонавчання» використовується для позначення ситуації, коли спостерігається велика кількість помилок при перевірці класифікатора на навчальній множині, що свідчить про відсутність або неправильне виявлення закономірностей в даних.

Додатковою проблемою, що виникає при вирішенні завдань класифікації, є проблема Positive-Unlabelled Learning (PU Learning). Цей така ситуація із набором даних, в якому позначені лише екземпляри певного класу (позитивні), тоді як інші екземпляри можуть належати як цьому класу, так і «невідомому» (квазі-негативному) класу. Така невизначеність у позначенні може ускладнювати процес навчання моделі та призводити до неправильних класифікацій, особливо коли важко визначити, які саме приклади повинні вважатися «негативними».

Вирішення проблеми PU Learning вимагає розробки специфічних стратегій в навчанні моделі, щоб враховувати відсутність повноцінного набору позначених даних. Техніки, такі як вагова збалансованість класів, можуть бути використані для поліпшення точності класифікації в умовах невизначеності [22, 16].

Ще однією ключовою проблемою, з якою можна зіткнутися при вирішенні задач класифікації, є проблема дисбалансу класів. Це виникає, коли кількість прикладів одного класу значно переважає кількість прикладів іншого класу. У такому випадку модель може бути схильною приділяти більше уваги та ресурсів для класу з більшою кількістю екземплярів, що може призвести до неправильного навчання та невірної класифікації менш представленого класу [14].

Розв'язання проблеми дисбалансу класів включає в себе використання різних стратегій, таких як вагова збалансованість, введення штрафів для помилок в менш представленому класі, або використання методів синтезування даних, таких як SMOTE. Ці підходи спрямовані на забезпечення рівноваги

впливу кожного класу на процес навчання, щоб модель була адекватно навчена та ефективно класифікувала обидва класи незалежно від їхньої кількісної репрезентації в наборі даних.

В цілому, методи вирішення задач класифікації поділяються на дві групи: статистичні методи (байєсівська класифікація, логістична регресія, аналіз відстаней між векторами ознак об’єктів) і методи машинного навчання, які включають класифікацію за допомогою дерев рішень, нейронних мереж, алгоритмів покриття, методу опорних векторів та методу k-найближчих сусідів тощо.

1.1.2 Геометрична інтерпретація і оцінка якості моделі класифікатора

Задачу класифікації можна геометрично інтерпретувати, особливо при аналізі з двома незалежними змінними, що дозволяє її представити в двовимірному просторі. Кожному об'єкту ставиться у відповідність точка на площині, де символи «+» і «-» вказують на приналежність об'єкта до одного з двох класів. На площині чітко виокремлюється структура даних, де точки класу «+» зосереджені в центральній області.

Процес побудови класифікаційної функції передбачає створення поверхні, що обводить центральну область. Ця поверхня визначається як функція, що має значення «+» усередині обведеної області та «-» поза нею. Вибір форми функції залежить від застосовуваного алгоритму, як показано на рисунку 1.1.

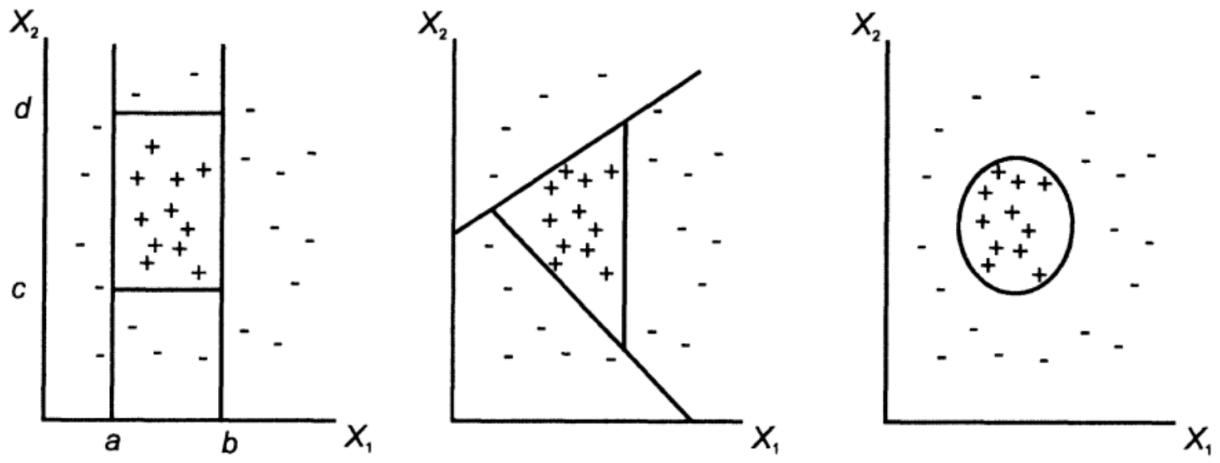

Рисунок 1.1 — Класифікація в двовимірному просторі

Оцінка точності класифікації може бути проведена через крос-перевірку (cross-validation) або тестову множину. Порівняння точності класифікації тестової множини з точністю класифікації навчальної множини дозволяє оцінити ефективність моделі. Якщо точність класифікації тестової множини аналогічна точності навчальної, це свідчить про успішність моделі.

Для візуалізації результатів перевірки часто використовується таблиця спряженості (confusion matrix). Цей процес включає застосування навчального алгоритму до тестової множини, де додається стовпець з вихідними значеннями, обчисленими за допомогою побудованої моделі. Різниці у значеннях стовпців вказують на якість класифікаційної моделі [28].

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Таблиця 1.1 — Таблиця спряженості (confusion matrix)

Нижче наведено пояснення компонентів таблиці:

TP (True Positives) - кількість вірно класифікованих позитивних прикладів.

TN (True Negatives) - кількість вірно класифікованих негативних прикладів.

FN (False Negatives) - кількість позитивних прикладів, класифікованих як негативні.

FP (False Positives) - негативні приклади, класифіковані як позитивні.

Залежно від конкретної задачі, події можуть бути визначені як позитивні чи негативні. На головній діагоналі confusion matrix вказано кількість правильно класифікованих прикладів, тоді як на побічній діагоналі – кількість неправильно класифікованих прикладів. Велика кількість неправильно класифікованих прикладів може свідчити про недоцільність моделі, і у такому випадку може бути необхідно внести зміни в параметри побудови моделі, збільшити обсяг навчальної вибірки або адаптувати набір вхідних полів. З іншого боку, якщо кількість неправильно класифікованих прикладів невелика, це може вказувати на те, що ці приклади є аномаліями. У такому випадку можливо розглянути аналіз характеристик цих прикладів і вирішити, чи слід додати новий клас для їх класифікації.

1.2 Особливості класифікації в умовах напівконтрольованого навчання

1.2.1 Загальний опис техніки напівконтрольованого навчання

Два основні методи машинного навчання — це контрольоване (з вчителем) та неконтрольоване (без вчителя) навчання. Поєднання обох цих технологій породило особливе середовище, відоме як напівконтрольоване навчання.

Напівконтрольоване навчання (SSL - semi-supervised learning або weak supervision) — це техніка машинного навчання, яка використовує невелику частину позначених даних і багато не позначених даних для навчання прогнозної моделі [21, 1].

Щоб краще зрозуміти концепцію SSL, ми повинні поглянути на неї через призму двох її основних аналогів (рис. 1.2).

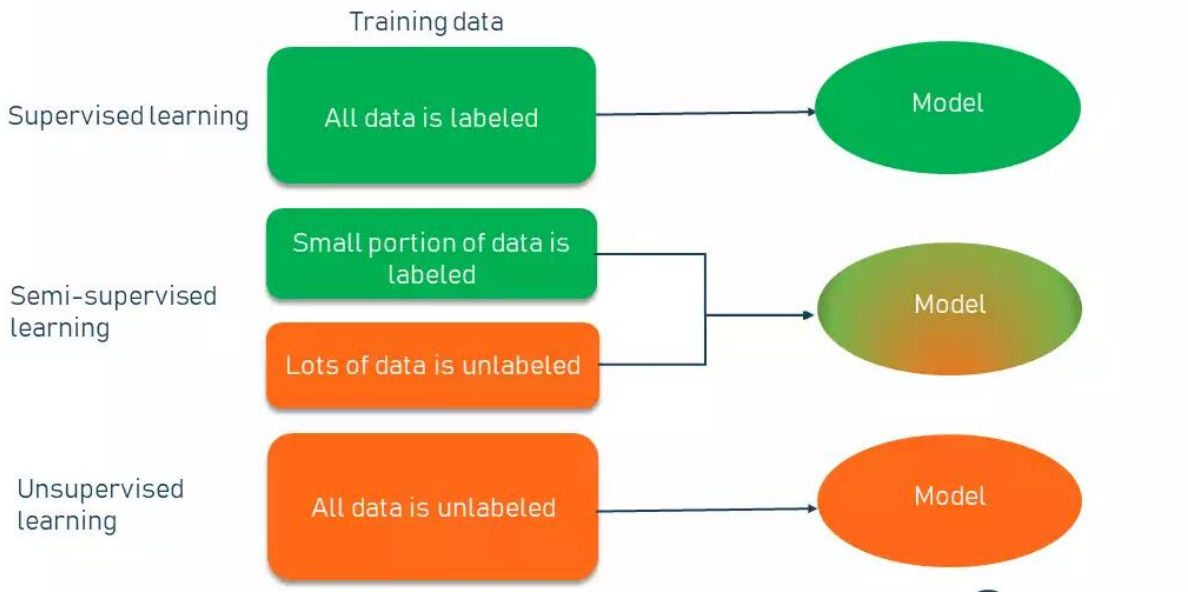

Рисунок 1.2 — Контрольоване і неконтрольоване навчання

Контрольоване навчання — це навчання моделі машинного навчання за допомогою позначеного набору даних. Органічні мітки часто доступні в даних, але процес може залучати людину-експерта, яка додає теги до необроблених даних, щоб показати моделі цільові атрибути (відповіді). Простіше кажучи, мітка — це в основному опис, який показує моделі, що вона повинна передбачити.

Контрольоване навчання має кілька обмежень. Цей процес є повільнішим (потрібно, щоб люди-експерти вручну позначали навчальні приклади один за одним) і набагато дорожчим (модель має бути навчена на великих обсягах вручну позначених даних, щоб забезпечити точні прогнози).

З іншого боку, неконтрольоване навчання — це коли модель намагається самостійно, без нагляду людини, виявити приховані шаблони, відмінності та подібності в немаркованих даних. У цьому методі точки даних групуються в кластери на основі подібності [3, 2].

Хоча навчання без вчителя є дешевшим способом виконання навчальних завдань, це не ідеальний засіб. Зазвичай сценарій має обмежену сферу застосування (в основному для цілей кластеризації) і забезпечує менш точні результати.

Напівконтрольоване навчання поєднує контрольоване навчання та методи неконтрольованого навчання для вирішення своїх ключових проблем. З його допомогою початкову модель навчається на кількох позначених зразках, а потім ітеративно застосовується до більшої кількості непозначених даних [21].

На відміну від неконтрольованого навчання, SSL працює для вирішення різноманітних проблем: від класифікації та регресії до кластеризації та асоціації.

На відміну від навчання з вчителем, метод використовує невеликі обсяги позначених даних, а також великі обсяги непозначених даних, що зменшує витрати на ручне анотування та скорочує час підготовки даних [11].

Оскільки немаркованих даних є багато та їх легко отримати, то напівконтрольоване навчання знаходить багато сфер застосування, при цьому точність результатів не надто страждає.

Розглянемо один із реальних сценаріїв, наприклад виявлення шахрайства. Скажімо, компанія з 10 мільйонами користувачів проаналізувала п’ять відсотків усіх транзакцій, щоб класифікувати їх як шахрайські чи ні, тоді як решта даних не була позначена тегами «шахрайство» та «не шахрайство». У цьому випадку напівконтрольоване навчання дозволяє задіяти всю інформацію компанії без необхідності витрачати додаткові ресурси на анотацію даних або жертвувати точністю.

Принцип напівконтрольованого навчання полягає у тому, що замість додавання тегів до всього набору даних, ви переглядаєте та позначаєте вручну лише невелику їх частину і використовуєте їх для навчання моделі, яка потім застосовується до всього обсягу непозначених даних (рис. 1.3).

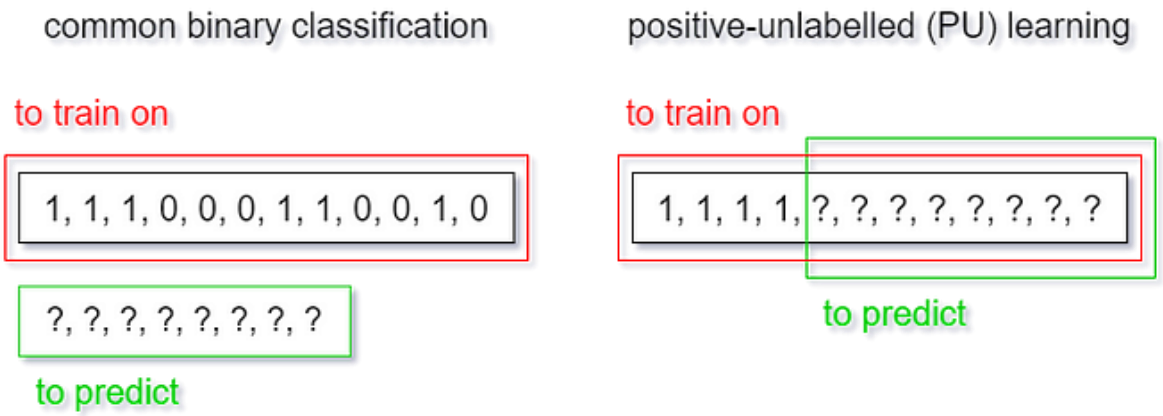

Рисунок 1.3 — Відмінність напівконтрольованого навчання

1.2.2 Самонавчання як метод напівконтрольованого навчання

Одним із найпростіших прикладів напівконтрольованого навчання є самонавчання.

Самонавчання — це процедура, за якої можна взяти будь-який контрольований метод для класифікації чи регресії та змінити його, щоб він працював у напівконтрольований спосіб, використовуючи переваги позначених і немаркованих даних. Стандартний робочий процес представлено на рисунку 1.4.

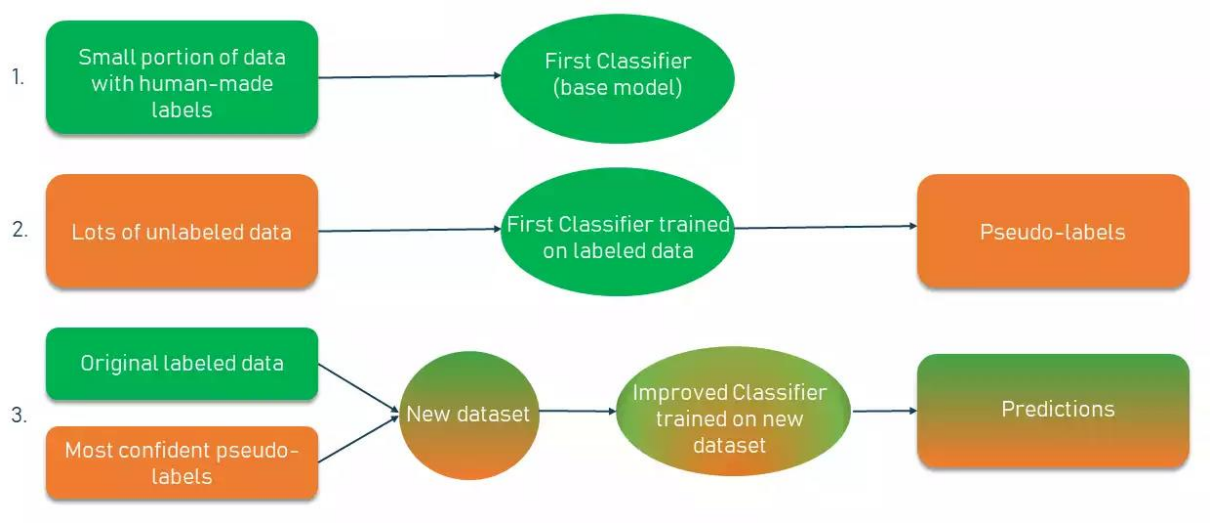

Рисунок 1.4 — Самонавчання як метод напівконтрольованого навчання

При цьому вибирається невелика кількість позначених даних, наприклад, зображень котів і собак з відповідними мітками, і використовується цей набір даних для навчання базової моделі за допомогою звичайних контрольованих методів.

Потім ви застосовуєте процес, відомий як псевдомітки, коли ви берете частково навчену модель і використовуєте її, щоб робити прогнози для решти даних, які ще не позначені. Згенеровані після цього мітки називаються псевдомітками, оскільки вони створюються на основі початково позначених даних, які мають обмеження (скажімо, може бути нерівномірне представлення класів у наборі, що призводить до упередженості — більше собак, ніж котів).

З цього моменту ви берете найбільш достовірні прогнози, зроблені за допомогою моделі (наприклад, вам потрібна впевненість понад 80 відсотків, що на певному зображенні зображено кота, а не собаку). Якщо будь-яка з псевдоміток перевищує цей рівень достовірності, ви додаєте їх до позначеного набору даних і створюєте новий об’єднаний вхід для навчання удосконаленої моделі [21].

Процес може проходити через кілька ітерацій (10 часто є стандартною кількістю), щоразу додаючи все більше псевдоміток. За умови, що дані підходять для процесу, продуктивність моделі зростатиме на кожній ітерації.

Хоча існують успішні приклади використання самонавчання, слід підкреслити, що продуктивність може сильно відрізнятися від одного набору даних до іншого. І є багато випадків, коли самопідготовка може знизити продуктивність порівняно з проходженням навчання із вчителем [7].

1.2.3 Спільне навчання як метод напівконтрольованого навчання

Ще одним методом напівконтрольованого навчання, який використовується, коли доступна лише невелика частина позначених даних є метод спільного навчання. На відміну від самонавчання, спільне навчання навчає двох окремих класифікаторів на основі двох представлень даних [21].

Представлення даних - це набори різних ознак, якими описується кожний екземпляр даних (кожне спостереження). Набір таких ознак має бути незалежним від класу. Крім того, такі набори (представлення) мають бути самодостатніми — за кожним із наборів ознак можна доволі точно передбачити позитивний / негативний клас.

В оригінальному дослідницькому документі [3] щодо спільного навчання стверджується, що цей підхід можна успішно використовувати, наприклад, для завдань класифікації веб-контенту. Опис кожної веб-сторінки можна розділити на два види: один зі словами, які зустрічаються на цій сторінці, а інший зі словами-прив’язками в посиланні, що веде на неї (рис. 1.5).

Принцип спільного навчання такий. Спочатку тренується окремий класифікатор (модель) для кожного представлення за допомогою невеликої кількості позначених даних. Потім для отримання псевдоміток додається більший пул немаркованих даних. Класифікатори спільно навчають один одного, використовуючи псевдомітки з найвищим рівнем достовірності. Якщо перший класифікатор впевнено передбачає справжню мітку для зразка даних, тоді як інший робить помилку передбачення, тоді дані з впевненими псевдомітками, призначеними першим класифікатором, оновлюють другий класифікатор і навпаки. Останнім кроком є об’єднання прогнозів із двох оновлених класифікаторів для отримання одного результату класифікації.

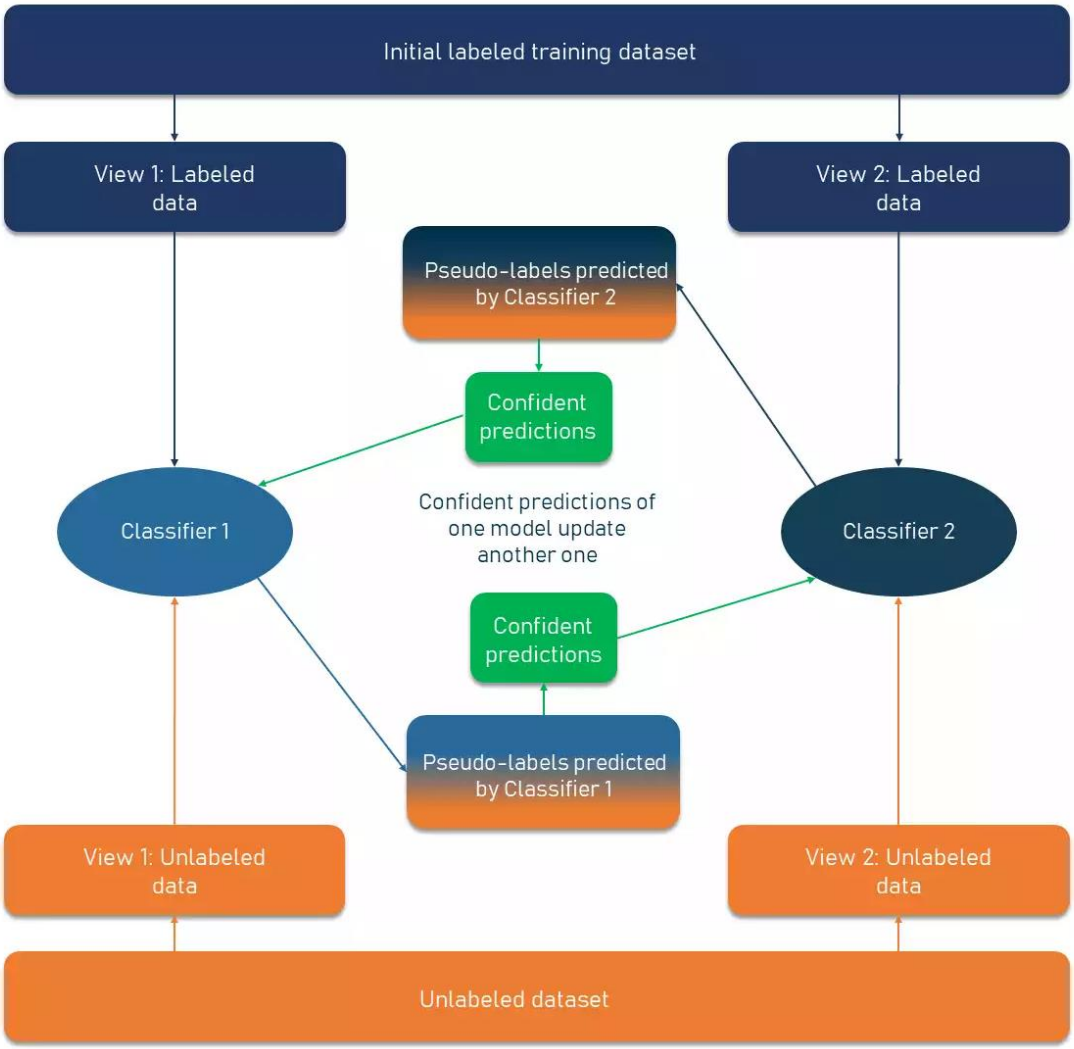

Рисунок 1.5 — Спільне навчання

Як і у випадку з самонавчанням, спільне навчання проходить через багато ітерацій, щоб побудувати додатковий навчальний набір даних з мітками з великої кількості непозначених даних [8].

1.2.4 Використання графів в контексті напівконтрольованого навчання

Ще одним популярним методом SSL є представлення позначених і непозначених даних у вигляді графів, а потім застосування алгоритму розповсюдження міток. Він поширює створені людиною анотації по всій мережі передачі даних (рис 1.6).

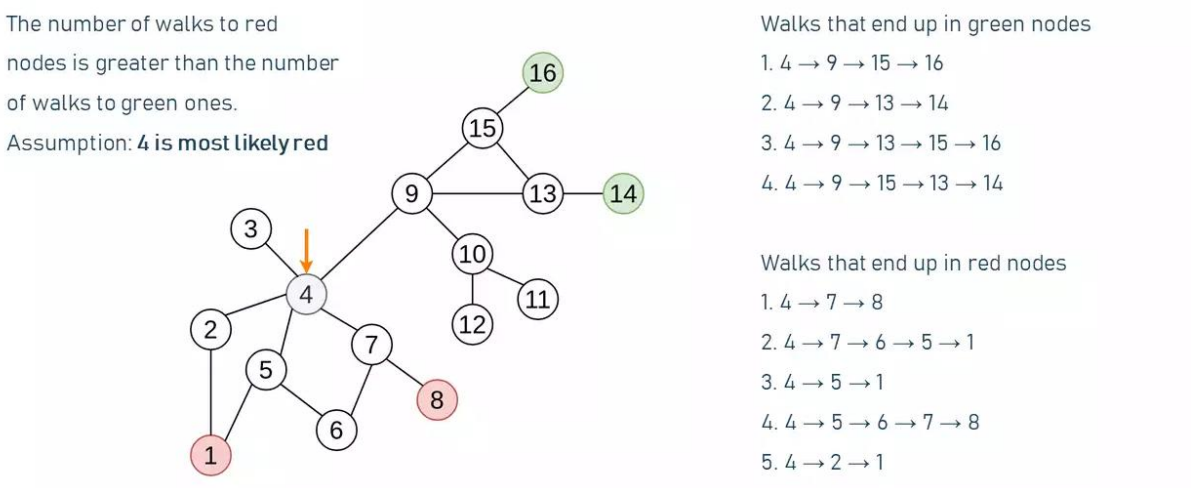

Рисунок 1.6 — Використання графів в напівконтрольованому навчанні

На графі представлено мережу точок даних, більшість із яких нерозмічені, і чотири точки позначені мітками (дві червоні 1, 8 та дві зелені точки 14, 16 для представлення різних класів). Завдання полягає в тому, щоб поширити ці кольорові мітки по всій мережі. Один із способів зробити це — вибрати, скажімо, точку 4 і підрахувати всі різні шляхи, які проходять мережею від 4 до кожного кольорового вузла. Якщо ви це зробите, то побачите, що є п’ять кроків, що ведуть до червоних точок, і лише чотири – до зелених. Звідси можна припустити, що точка 4 належить до червоної категорії. Потім ви повторите цей процес для кожної точки у графі [4, 15].

Практичне використання цього методу можна побачити в системах персоналізації та рекомендаціях [19]. За допомогою розповсюдження міток ви можете передбачити інтереси клієнтів на основі інформації про інших клієнтів. Тут ми можемо застосувати варіацію припущення безперервності — якщо

двоє людей, наприклад, пов’язані в соціальних мережах, дуже ймовірно, що вони поділятимуть однакові інтереси.

1.2.5 Огляд задач класифікації із використанням методів напівконтрольованого навчання

З огляду на те, що кількість даних постійно зростає часто просто неможливо вчасно позначити їх. Наприклад, активний користувач TikTok завантажує в середньому до 20 відео на день. І є 1 мільярд активних користувачів. У такому випадку напівконтрольоване навчання має значні переваги у широкому спектрі задач від розпізнавання зображень і мови до класифікації веб-вмісту та текстових документів [20].

Приклади вирішення задач класифікації із використанням методів напівконтрольованого навчання.

1) Розпізнавання мови

Додавання міток до аудіо – це завдання, яке потребує ресурсів і часу, тому для подолання труднощів і забезпечення кращої продуктивності можна використовувати напівконтрольоване навчання. Facebook (тепер Meta) успішно застосував напівконтрольоване навчання (а саме метод самонавчання) до своїх моделей розпізнавання мови та вдосконалив їх. Вони почали з базової моделі, яка була навчена на 100 годинах аудіо даних, анотованих людиною. Потім було додано 500 годин немаркованих аудіо даних і використано самонавчання для підвищення продуктивності моделей. Що стосується результатів, рівень помилок у словах (WER) знизився на 33,9%, що є значним покращенням [9].

2) Класифікація веб-контенту

З мільярдами веб-сайтів, які представляють різноманітний вміст, для класифікації потрібна величезна команда людських ресурсів, щоб впорядкувати інформацію на веб-сторінках шляхом додавання відповідних міток. Варіанти напівконтрольованого навчання використовуються для анотування веб-

контенту та його відповідної класифікації для покращення взаємодії з користувачем. Багато пошукових систем, у тому числі Google, застосовують SSL до свого компонента рейтингу, щоб краще розуміти людську мову та відповідність результатів пошуку кандидатів запитам. За допомогою протоколу SSL пошук Google знаходить вміст, який найбільше відповідає конкретному запиту користувача [16].

3) Класифікація текстових документів

Іншим прикладом успішного використання напівконтрольованого навчання є створення класифікатора текстових документів. Тут цей метод ефективний, оскільки анотаторам-людям дійсно важко прочитати багатослівні тексти, щоб призначити основну мітку, як-от тип або жанр.

Наприклад, класифікатор можна створити на основі нейронних мереж глибокого навчання, таких як мережі LSTM, які здатні знаходити довгострокові залежності в даних. Зазвичай навчання нейронної мережі вимагає великої кількості даних з мітками та без них. Напівконтрольоване навчання працює чудово, оскільки ви можете навчити базову модель LSTM на кількох текстових прикладах із позначеними вручну найбільш відповідними словами, а потім застосувати її до більшої кількості непозначених зразків.

Текстовий класифікатор SALnet, створений дослідниками з Університету Йонсей у Сеулі, Південна Корея, демонструє ефективність методу SSL для таких завдань, як аналіз емоційного забарвлення документа [16].

1.3 Ранні перехресні продажі як техніка збільшення прибутку компанії

1.3.1 Особливості перехресних продажів у сфері роздрібної торгівлі

Існує ряд методів для збільшення обсягу продажів, при цьому збільшення рекламного бюджету не завжди є найефективнішим способом. Продуктивнішими можуть бути прості та ефективні стратегії, такі як збільшення середнього чеку через методи допродажів і крос-продажів.

Допродажі, або «up-sell» (дослівно - «підняття суми продажу»), є маркетинговою тактикою, спрямованою на стимулювання покупця витрачати більше коштів. Зазвичай покупець спрямовується на заощадження, обираючи товар за найнижчою ціною, але вирішенням цієї проблеми може бути пропозиція альтернативного товару. Впровадження стратегій, таких як допродажі, може допомогти в переконанні клієнта вибрати товар більшої вартості, проте із значною перевагою, що може бути обумовлено його новизною, актуальністю та функціональністю [24]. Іншим підходом може бути збільшення обсягу чека за рахунок надання додаткових послуг.

Застосування цих стратегій виглядає не лише як спроба заробити на клієнтові, але й як турбота про нього. Важливо продемонструвати, наприклад, що взуття з натуральної шкіри, навіть якщо його ціна вища, ніж у штучного замінника, буде забезпечувати вищий рівень комфорту. Аналогічно, ефективний міксер чи праска із функцією автоматичного вимикання можуть визначити різницю між звичайним пристроєм і тим, який дбає про безпеку вашого дому.

На прикладі відомого інтернет-магазину «Rozetka» можна побачити, як ці стратегії застосовуються на практиці. Припустимо, відвідувач сайту розглядає пилососи. Враховуючи попередні перегляди на інших сайтах, читання відгуків та порівняльний аналіз, покупець може залишатися у сумнівах. У такому випадку пропозиція більш привабливої альтернативи може бути вирішальною (рис. 1.7).

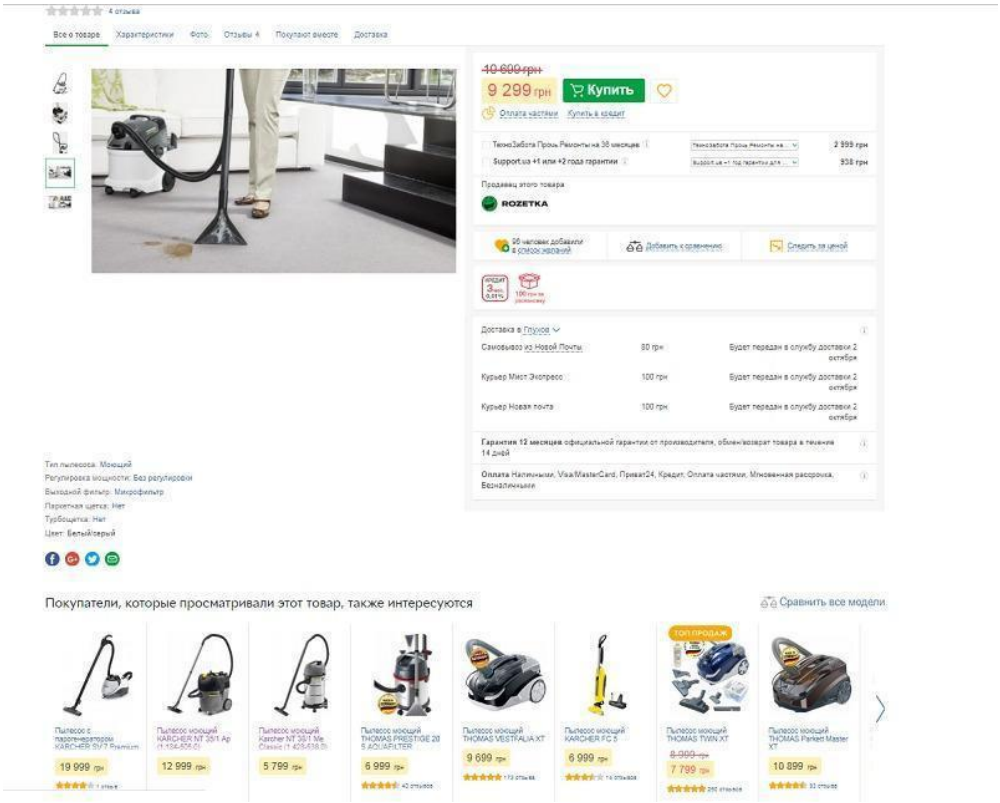

Рисунок 1.7 — Скріншот сторінки Інтернет-магазину Rozetka

У нижній частині сторінки товару представлені позиції, які зацікавили інших відвідувачів. Зазначимо, що більшість пилососів у цьому списку мають вищу вартість, ніж обрана покупцем модель. Хоча наявні й доступніші альтернативи, вирішення витратити конкретну суму знижує ймовірність вибору менш вартісної моделі.

Додатково, на картці цього товару доступна кнопка «Купують разом». Перехід за посиланням розкриває додаткові послуги обслуговування. Це створює у покупця враження, ніби він робить вибір самостійно.

Справи стають ще простішими у випадку товарів, що відрізняються великою кількістю характеристик, наприклад, смартфонів. Для багатьох вивчити всі технічні деталі і порівняти їх важко. Тому споживачі часто керуються наявним бюджетом або обирають популярні моделі (рис. 1.8).

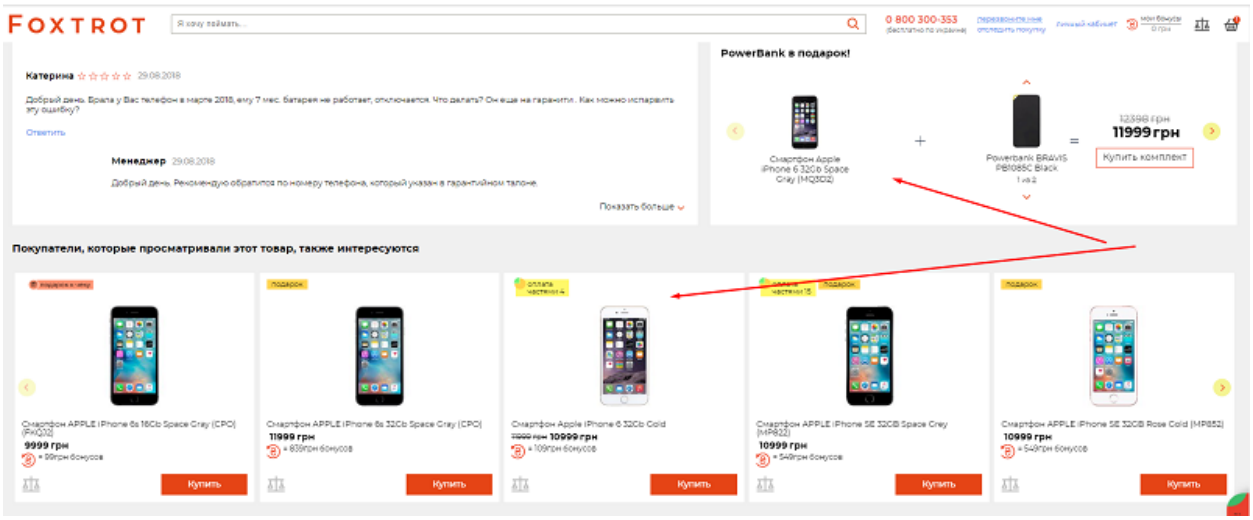

Рисунок 1.8 — Скріншот сторінки Інтернет-магазину Foxtrot

На прикладі скріншоту можна спостерігати застосування двох стратегій одночасно: допродажу і крос-продажу. Маркетингова тактика допродажів має ряд переваг, серед яких:

- легкість впровадження, яка не вимагає складних налаштувань;
- можливість підкреслити право вибору покупця без надмірного тиску;
- сприяння встановленню більш тісних відносин з відвідувачами.

Крос-продажі, як ще один метод стимулювання покупця витрачати більше, здійснюються через придбання товарів з інших категорій. Супутніми товарами можуть бути:

- канцелярські товари, пропоновані під час покупки шкільного рюкзака;
- чохли, які часто купують разом із ноутбуками, смартфонами чи планшетами;
- змінні гумки та миючі засоби для скла, які можна запропонувати під час придбання автомобільних двірників.

У випадку, якщо акційний комплект не зацікавив клієнта, пропонується альтернатива інших товарів під час етапу оформлення замовлення. Існують різні стратегії реалізації перехресних продажів, де товари можуть бути пропоновані в інших контекстах, таких як e-mail-розсилка, тематичні статті на блозі та інші.

Застосування веб-аналітики допомагає зробити метод перехресних продажів більш ефективним. Оптимальна стратегія включає створення цікавих комплектів та розширення асортименту товарів [29].

Переваги перехресних продажів включають:

- позитивне сприйняття покупцями, оскільки їм пропонуються дійсно потрібні товари;
- підвищення лояльності клієнтів, оскільки можна не лише придбати корисний товар, але й зекономити на доставці;
- збільшення інформованості відвідувачів сайту про асортимент товарів, що збільшує ймовірність додаткових покупок;
- метод не вимагає додаткових витрат.

Для ефективного функціонування перехресних продажів важливо забезпечувати наявність актуальних, корисних і нових пропозицій, уникаючи враження спроби продажу неактуальних товарів.

1.3.2 Особливості перехресних продажів у сфері фінансових послуг

На відміну від сфери роздрібної торгівлі, де контакти з покупцями мають епізодичний характер, практика здійснення перехресних продажів фінансових послуг передбачає можливість встановлення довгострокових, конструктивних взаємин з потенційними клієнтами. Фінансові компанії мають значну клієнтську базу. Для її подальшого розвитку існує два методи: можливість її кількісного розширення шляхом залучення нових клієнтів; можливість її якісного розширення шляхом надання більшої кількості послуг існуючим клієнтам («перехресний продаж»). Другий спосіб більш легкий, саме тому що перший контакт з клієнтом вже відбувся та клієнт має деякі визначені очікування щодо банку [24].

Перехресний продаж означає продаж пов’язаних або додаткових продуктів клієнту. Перехресні продажі є одним з найефективніших методів марке-

тингу. Наприклад, якщо клієнт банку має іпотеку, відділ продажів може спробувати перехресно продати цьому клієнту особисту кредитну лінію або ощадний продукт.

Перехресні продажі існуючим клієнтам є одним із основних методів отримання нових доходів для багатьох фінансових компаній. Це, мабуть, один із найпростіших способів розвитку свого бізнесу, оскільки вони вже встановили стосунки з клієнтом і знайомі з його потребами та цілями [29].

Разом з тим просте направлення клієнта до іншого відділу, який фактично продає та обробляє новий для клієнта продукт, може призвести до ситуацій, коли рекомендації надається незалежно від того, потрібні вони чи ні, оскільки співробітник відділу може не зрозуміти, коли клієнт дійсно потребує додаткової послуги або готовий її придбати.

В такому випадку перехресні продажі матимуть негативний вплив на лояльність клієнтів. Якщо зробити це неправильно, це може виглядати як настирлива тактика продажу. Це очевидно, коли продавець агресивно намагається продати пов’язаний продукт або намагається продати, не розуміючи потреби клієнта в ньому. Це не тільки впливає на продажі, але й негативно впливає на репутацію бренду [30].

Крім того, перехресні продажі неправильному типу клієнтів можуть бути контпродуктивними. Деякі клієнти мають високі вимоги до обслуговування, і чим більше продуктів вони купують, тим більше послуг вони отримують. У міру зростання потреб у їхніх послугах зростають і витрати, пов’язані з наданням цих послуг.

Так трапляється тому, що деякі клієнти час від часу повертають або обмінюють товари. При перехресних продажах непотрібних товарів (послуг) неправильно визначеному сегменту клієнтів прибуток не реалізується. Спочатку їх покупки приносять значні доходи ; однак вони часто повертають або не сплачують платежі, що обходиться компанії дорожче, ніж додатковий дохід від клієнта.

Компанії мають на 60–70% більшу ймовірність продати наявному клієнту, тоді як ймовірність продажу новому клієнту становить від 5% до 20%.

Таким чином компанії легше надавати нові послуги своїм наявним клієнтам, ніж залучати нових клієнтів. Існуючі клієнти довіряють бренду та знаходять цінність у продуктах та/або послугах. Ця довіра є рушієм успіху збільшення продажів. Наприклад, якщо клієнт довіряє бренду, він, як правило, довірятиме бренду, коли він пропонує кращий (дорожчий) варіант товару / послуги.

Методи перехресних продажів включають рекомендації, пропозиції знижок і комплектування супутніх товарів. Подібно до збільшених продажів, компанія прагне заробити більше грошей на клієнта та підвищити сприйняту цінність шляхом вирішення та задоволення потреб споживачів.

1.3.3 Практика ефективних перехресних продажів

Отже техніка перехресних продажів має свої недоліки і переваги.

Основні переваги перехресних продажів:

- може потенційно збільшити дохід за рахунок збільшення обсягів продажу, особливо менш популярних товарів;
- може підвищити лояльність до бренду, оскільки клієнти ще більше ознайомляться з асортиментом продукції однієї компанії.

Потенційні проблеми, пов’язані із неправильно реалізованими перехресними продажами:

- може призвести до збільшення витрат, пов’язаних із обслуговуванням, оскільки перехресний продаж може бути дорожчим порівняно з іншими стратегіями;
- може негативно вплинути на відносини, якщо техніка перехресних продажів виявиться настирливою;
- може призвести до негативного сприйняття громадськістю вимог або вимог поєднання кількох продуктів разом.

Є кілька стратегій, які можна застосувати, щоб зробити перехресний продаж ефективним:

- використовувати електронні кампанії, щоб періодично представляти додаткові продукти та послуги;
- почекати, поки компанія не налагодить стосунки та не досягне успіху з клієнтом;
- переконатися, що ваші продукти та послуги відповідають потребам і цілям клієнта;
- пропонувати щось, що не є контпродуктивним і не може зменшити задоволеність клієнтів.

Таким чином, під час перехресних продажів потрібно звертати увагу на найбільш лояльних клієнтів компанії, які з більшою ймовірністю зроблять покупку знову. Створюйте кампанії, орієнтовані на задоволених клієнтів, і рекламуйте їм додаткові продукти. Навчіть співробітників розпізнавати задоволених клієнтів і оцінювати їхні потреби [17].

З іншого боку, далеко не завжди клієнти обізнані про інші пропозиції (товари / послуги) компанії. Навчіть їх і допоможіть їм зрозуміти, як ці продукти можуть приносити користь. Розмовляючи з клієнтом, робіть це в привітній манері; інакше це виглядає як рекламна пропозиція [23]. Нарешті, уникайте незадоволених клієнтів, оскільки це може посилити розрив між ними та вашим брендом.

1.4 Висновки до розділу 1

Задача класифікації передбачає побудову ефективних моделей для розподілу об'єктів за заданими класами. Розділення набору даних на навчальну і тестову частини допомагає побудувати та оцінити класифікаційну модель машинного навчання.

Машинне навчання включає контрольоване і неконтрольоване навчання, а також напівконтрольоване навчання, яке ефективно поєднує обидва підходи,

зменшуючи витрати і час. Застосування класифікації розглядається в різних галузях, таких як маркетинг, медицина і розпізнавання образів.

Самонавчання використовує обмежену кількість позначених даних і багато непозначених, використовуючи контрольовані методи, псевдомітки та ітеративне оновлення моделі. Спільне навчання, як ще один метод напівконтрольованого навчання, використовує два класифікатори на основі різних представлень даних. Використання графів для представлення позначених і непозначених даних, разом з алгоритмом розповсюдження міток, є ефективним методом в умовах обмеженої доступності позначених даних. Всі ці методи можуть бути використані для вирішення завдань класифікації в різних сферах залежно від конкретного контексту та вимог задачі.

В сфері маркетингу збільшення продажів досягається за допомогою стратегій, серед яких ключовими елементами є допродажі та перехресні продажі. Допродажі спрямовані на підвищення середнього чеку, пропонуючи клієнтам альтернативні та вдосконалені товари, тим самим підвищуючи їхнє задоволення від покупки. Перехресні продажі, у свою чергу, полягають у пропозиції додаткових товарів з інших категорій, що задовольняє різні потреби клієнтів та сприяє підвищенню рівня лояльності.

У сфері фінансових послуг перехресні продажі визначають стратегію розвитку клієнтської бази, сприяючи установленню довгострокових та конструктивних відносин. Перехресні продажі в цьому контексті передбачають пропозицію клієнтам додаткових фінансових продуктів, використовуючи вже існуючі відносини для спрощення процесу продажу.

Загалом, вдалі перехресні продажі можуть призвести до збільшення доходів та задоволення клієнтів, але їх неправильна реалізація може мати негативні наслідки. Бізнесу потрібно пропонувати додаткові продукти лише тоді, коли вони дійсно відповідають потребам та очікуванням клієнтів.

РОЗДІЛ 2 ДОСЛІДЖЕННЯ АЛГОРИТМІВ/МОДУЛІВ ДЛЯ ВИЗНАЧЕННЯ ПОДІБНОСТІ ОБ'ЄКТІВ

2.1 Відкриті джерела даних і особливості їх підготовки для відтворення умов напівконтрольованого навчання

2.1.1 Доступні джерела даних для побудови аналітичних моделей

Відкриті джерела даних пропонують дослідникам безпрецедентний доступ до різноманітних і загальнодоступних наборів даних, уможливлюючи співпрацю та полегшуючи тиражування результатів досліджень. Роблячи дані відкритими, ці джерела сприяють прозорості, підвищують обґрунтованість наукових досліджень і дозволяють ідентифікувати потенційні упередження. Крім того, відкриті джерела даних заохочують інновації, сприяючи обміну знаннями, методами та техніками в спільноті інформатики. Завдяки використанню відкритих даних дослідники можуть спиратися на наявну роботу, перевіряти твердження та робити внесок у колективне розуміння різноманітних феноменів інформатики.

У мережі Інтернет розміщено велику кількість відкритих джерел даних, починаючи від загальнодоступних сховищ і державних баз даних до академічних джерел. Ці джерела надають дослідникам широкий вибір наборів даних, що охоплюють різні сфери інформатики. Доступ до таких різноманітних наборів даних дозволяє дослідникам досліджувати широкий спектр дослідницьких питань і сценаріїв у галузі. Цей різноманітний діапазон наборів даних сприяє повному розумінню різних феноменів науки про дані та дозволяє дослідникам розробляти надійні моделі та алгоритми.

Kaggle став надійною платформою для дослідників і практиків, яким потрібні надійні та високоякісні набори даних у галузі науки про дані. Kaggle забезпечує надійність і цілісність наборів даних своїх хостів за допомогою суворих заходів контролю якості та активної спільноти користувачів. Дослід-

ники можуть покладатися на систему зворотного зв’язку та перегляду, яку надає Kaggle, щоб оцінити якість наборів даних перед включенням їх у свої проекти. Це гарантує, що набори даних, отримані від Kaggle, є надійними, високоякісними та придатними для ретельного дослідження та аналізу.

Kaggle надає набори даних і стандартизовані набори даних, які служили еталоном для оцінки моделей і алгоритмів машинного навчання. Ці контрольні набори даних широко використовуються в конкурсах і дослідженнях машинного навчання, що забезпечує справедливі порівняння та надійні оцінки. Використовуючи встановлені Kaggle оціночні показники та доступ до базових міток істинності, дослідники можуть перевірити ефективність своїх моделей і алгоритмів. Порівняльні набори даних Kaggle сприяють розробці найсучасніших методів і забезпечують стандартизовану основу для оцінки ефективності різних підходів у галузі інформатики.

Kaggle пропонує ряд комерційних наборів даних, які є дуже цінними для загальних бізнес-додатків. Наприклад, одним із таких наборів даних, доступних на Kaggle, є «Перехресні продажі у страховій галузі». Цей набір даних зосереджений на перехресних продажах, які включають ідентифікацію продуктів або послуг, які задовольняють додаткові потреби, не задоволені оригінальним продуктом, яким володіє клієнт. Набір даних фіксує різні атрибути клієнта. Дослідники та компанії можуть використовувати цей набір даних для розробки моделей і алгоритмів машинного навчання для прогнозування реакції клієнтів на стратегії перехресних продажів, оптимізації маркетингових кампаній і підвищення прибутку.

Доступність наборів даних на Kaggle виявилася важливою для загальних бізнес-додатків. Наприклад, компанії електронної комерції можуть використовувати набори даних про транзакції клієнтів для розробки систем рекомендацій, які реалізують ефективні стратегії перехресних продажів (рис. 2.1). Аналізуючи поведінку клієнтів, моделі веб-перегляду та історію покупок, моделі ML можуть визначати можливості для перехресних продажів і рекомендувати відповідні продукти клієнтам під час процесу оформлення замовлення або за

допомогою персоналізованих кампаній електронною поштою. Такий підхід покращує взаємодію з клієнтами, стимулює повторні покупки та розширює обізнаність клієнтів про каталог продуктів роздрібного продавця.

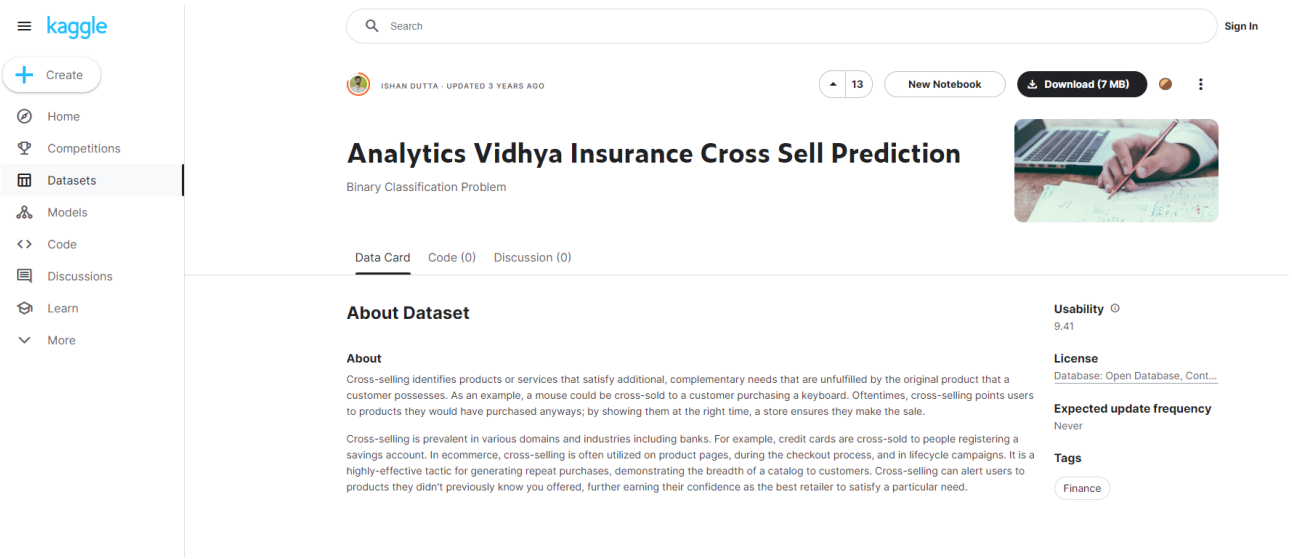

Рисунок 2.1 - Kaggle як джерело порівняльного аналізу

Так само банки можуть використовувати набори даних від Kaggle для розробки моделей, які оптимізують перехресний продаж фінансових продуктів. Аналізуючи профілі клієнтів, моделі транзакцій і фінансову поведінку, алгоритми ML можуть визначити потенційні можливості перехресних продажів і запропонувати відповідні продукти або послуги клієнтам. Цей цілеспрямований підхід підвищує задоволеність клієнтів, покращує отримання прибутку та зміцнює лояльність клієнтів.

2.1.2 Підготовка відкритих даних до машинного навчання

2.1.2.1 Адаптація набору даних до умов бізнес-задачі

Дані для подальшого аналізу взяті з відкритого джерела. Це типовий набір даних для бінарної класифікації, проте він має як раз потрібну нам бізнес-специфіку – дані для перехресних продажів (рис. 2.2).

Позитивні респонденти - це ті клієнти в нашій БД, які вже придбали товар / замовили послугу. Задача: знайти схожих на них серед всіх інших клієнтів

в нашій БД, щоб зробити першу пропозицію щодо придбання аналогічного товару. З точки зору бізнесу перевага буде надаватися методам, які забезпечують коректний пошук більшої кількості потенційних позитивних респондентів (recall), тобто навіть за рахунок зменшення точності алгоритму (за умови низької собівартості контакту - наприклад, розсилка через e-mail).

```
1  data.sample(10)
```

id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
230918	Male	49	1	28.0	1	1-2 Year	No	36963.0	26.0	46	0
110883	Female	21	1	13.0	1	< 1 Year	No	19740.0	152.0	282	0
91465	Male	68	1	28.0	0	> 2 Years	Yes	53709.0	124.0	11	1
2192	Male	40	1	39.0	0	1-2 Year	Yes	29700.0	26.0	243	0
195451	Female	21	1	29.0	1	< 1 Year	No	23651.0	152.0	228	0
25355	Female	22	1	15.0	0	< 1 Year	Yes	38089.0	152.0	112	0
264155	Male	22	1	46.0	1	< 1 Year	No	31385.0	152.0	206	0
127523	Female	35	1	28.0	0	1-2 Year	Yes	31192.0	122.0	185	0
87535	Male	42	1	8.0	0	> 2 Years	Yes	39170.0	124.0	226	0
123901	Male	33	1	30.0	1	1-2 Year	No	2630.0	29.0	31	0

Рисунок 2.2 – Робочий набір даних

Для того, щоб наш набір даних відповідав бізнес-задачі, ми його трохи модифікуємо:

- зменшимо для прискорення всіх розрахунків;
- залишимо в датасеті 15% позитивних респондентів;
- залишимо відомою нам лише вкрай незначну частку позитивних респондентів (1% від тієї кількості, що залишиться);
- при цьому всіх інших позитивних респондентів залишимо як «прихованих» для розрахунку всіх метрик по застосованим алгоритмам.

Лістинг 1 — Модифікація набору даних

```
data =
pd.read_csv('https://gist.githubusercontent.com/woldemarg/ec4df9c4319b8408f6178fc9d7c04b2d/raw/2b72490786a08ebcaf089ea2ac60f3597948a3c5/cross_train.csv')

TRG = 'Response'   # target
TST = 0.25
```

```
data.set_index('id', inplace=True)

# these features are known to be categorical from
description
data = data.astype(
    {'Region_Code': str,
     'Policy_Sales_Channel': str})
data = data.sample(frac=TST, random_state=RND).copy()

pos_idx = np.where(data[TRG])[0]
neg_idx = np.where(data[TRG] - 1)[0]

# shape data to make share of positives equal to given
POS_CNV
max_neg_len = int(len(pos_idx) * (1 - POS_CNV) / POS_CNV)

if max_neg_len > len(neg_idx):

    max_pos_len = int(len(neg_idx) * POS_CNV / (1 -
POS_CNV))

    pos_idx_cut = random.Random(RND).sample(list(pos_idx),
max_pos_len)

    data = data.iloc[list(pos_idx_cut) + list(neg_idx)]

else:

    neg_idx_cut = random.Random(RND).sample(list(neg_idx),
max_neg_len)

    data = data.iloc[list(pos_idx) + list(neg_idx_cut)]

y_orig = data[TRG].copy()

data.groupby(TRG).agg(
    count=(TRG, 'size'),
    share=(TRG, lambda x: x.shape[0] / data.shape[0]))

print(f'[INFO] {datetime.datetime.now()}: data shape:
{data.shape}')
```

```
[INFO] 2023-06-30 18:46:28.019825: data shape: (77113, 11)
```

```
data.sample(10)
data = data.drop(TRG, axis=1)
known_pos_len = math.ceil(len(np.where(y_orig)[0]) *
POS_KNW)
```

```
y_known = np.zeros(data.shape[0])
y_known[:known_pos_len] = 1
```

2.1.2.2 Дата-інжиніринг датасету

Всі алгоритми будемо застосовувати на підготовленому датасеті після простого етапу підготовки даних:

- неперервні предиктори перетворюємо на дискретні (розбиваємо по корзинах);
- таким чином BCI предиктори робимо категоріальними;
- до категоріальних предикторів застосовуємо one-hot-encoding (рис. 2.3).

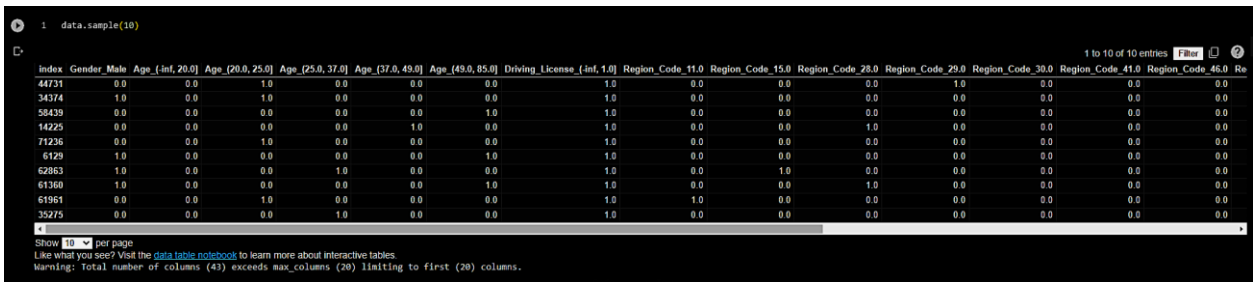

Рисунок 2.3 - Підготовка даних

Лістинг 2 — Код одного із блоків для підготовки даних

```
def bucketize_data(dat: np.array, num_bins: int) ->
pd.Series:

    # remove outliers
    flt = (dat[np.where(np.abs(scipy.stats.zscore(
           dat,
           ddof=1,
           nan_policy='omit')) <= 3)[0]])

    _, bins = pd.qcut(flt, num_bins, retbins=True,
duplicates='drop')

    return pd.cut(dat,
                  #
https://stackoverflow.com/a/47804292/6025592
                  np.concatenate(([-np.inf], bins,
[np.inf])),
```

```
                    precision=1)
num_cols = data.select_dtypes(include=np.number).columns

data[num_cols] = (data[num_cols]
                    .apply(bucketize_data, raw=True,
num_bins=NUM_BINS)
                    .astype(str))
```

Розрахунок кореляції категоріальних змінних із цільовою змінною Крамера в подальшому використаємо для розбиття простору ознак на підпростори.

Лістинг 3 — Розрахунок кореляції категоріальних змінних із цільовою змінною Крамера

```
def cramers_v(x: pd.Series, y: pd.Series) -> float:

    # https://towardsdatascience.com/the-search-for-
categorical-correlation-a1cf7f1888c9
    # https://stackoverflow.com/a/65439219/6025592

    # possible alt approach
    # https://stackoverflow.com/a/46755405/6025592

    conf_matrix = pd.crosstab(x, y).values
    chi2 = scipy.stats.chi2_contingency(conf_matrix)[0]
    n = conf_matrix.sum()
    phi2 = chi2 / n
    r, k = conf_matrix.shape
    phi2corr = max(0, phi2 - ((k-1)*(r-1))/(n-1))
    rcorr = r - ((r-1)**2)/(n-1)
    kcorr = k - ((k-1)**2)/(n-1)

    return np.sqrt(phi2corr / min((kcorr-1), (rcorr-1)))
objects = data.select_dtypes(include='object').columns

with warnings.catch_warnings():

    warnings.simplefilter(«ignore»)

    cat_corr = (data[objects]
                .apply(cramers_v, y=y_known)
                .fillna(0)
                .pipe(lambda x: x[x != 0])
                .rename('corr'))
```

```
cat_corr

Gender                          0.008376
Age                                   0.026450
Previously_Insured              0.034181
Vehicle_Age                     0.026924
Vehicle_Damage                  0.037031
Policy_Sales_Channel            0.006915
Name: corr, dtype: float64

enc = OneHotEncoder(
    drop='if_binary',
    max_categories=MAX_CTG,
    handle_unknown='infrequent_if_exist',
    sparse_output=False)

enc.fit(data[objects])

data = pd.DataFrame(
    data=enc.transform(data[objects]),
    columns=enc.get_feature_names_out())

data.sample(10)

X_pos, X_unknown, y_true = (
    data.iloc[:known_pos_len].copy(),
    data.iloc[known_pos_len:].copy(),
    y_orig.iloc[known_pos_len:].copy())

test_class_weights =
y_true.value_counts(normalize=True).to_dict()
print(
    f'[INFO] {datetime.datetime.now()}: num of known
positive responders: {X_pos.shape[0]}')

print(
    f'[INFO] {datetime.datetime.now()}: num of responders
to identify leads: {X_unknown.shape[0]}')
```

```
[INFO] 2023-06-30 18:46:33.726645: num of known positive
responders: 116
[INFO] 2023-06-30 18:46:33.731401: num of responders to
identify leads: 76997
```

2.2 Методи вимірювання відстаней між векторами ознак об'єктів в пакеті scipy

2.2.1 Огляд підходів і метрик відстаней

Оцінка відстаней і подібності між векторами характеристиками об’єктів можна назвати одним із фундаментальних завдань науки про дані. Від розпізнавання зображень до систем рекомендацій, здатність кількісно визначити схожість або відмінність між об’єктами є основою для аналізу даних і прийняття рішень.

Міри відстані служать основними інструментами для кількісного визначення подібності або відмінності між об’єктами. У цьому розділі розглядаються основні вимірювання відстані, які використовуються в науці про дані, надаючи інтуїтивно зрозумілі пояснення та практичні наслідки для кожного (рис. 2.4).

Евклідова відстань. Евклідова відстань, натхненна теоремою Піфагора, обчислює відстань по прямій лінії між двома точками в багатовимірному просторі. Обчислюючи квадратний корінь із суми квадратів різниць між відповідними елементами двох векторів, евклідова відстань є мірою загальної різниці. Ця метрика дозволяє ідентифікувати закономірності, кластери та подібності в даних, що робить її основним інструментом для виконання різноманітних завдань аналізу даних.

Манхеттенська відстань. Манхеттенська відстань, також відома як відстань міських кварталів або норма L1, походить від того, як пішоходи рухаються кварталами міста. Він вимірює відстань між двома точками шляхом підсумовування абсолютних різниць між їхніми координатами. Манхеттенська відстань особливо корисна під час роботи з об’єктами різного масштабу або розрідженими даними. Вона фіксує «поблочний» рух, необхідний для подорожі між двома точками. Ця міра відстані широко використовується в системах кластеризації, виявлення аномалій і рекомендацій.

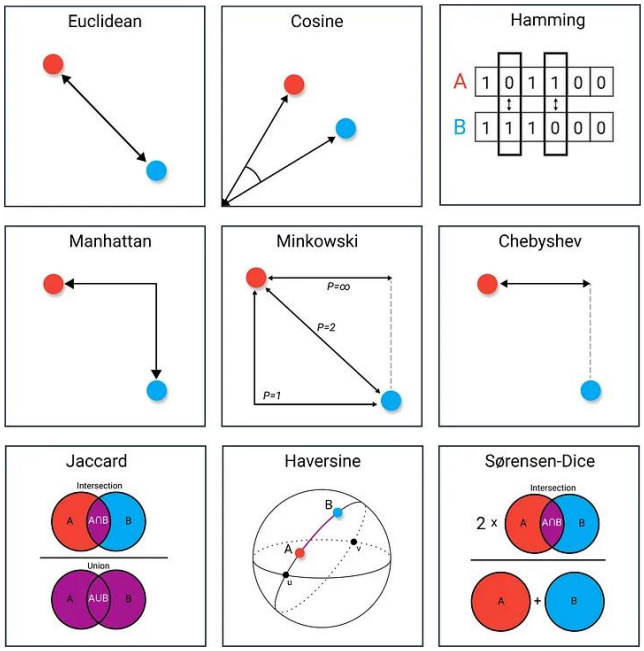

Рисунок 2.4 — Графічне інтерпретація відстаней між векторами ознак об’єктів

Косинусна подібність. Косинусна подібність вимірює косинус кута між двома векторами, забезпечуючи міру подібності незалежно від їх величин. Ця метрика оцінює орієнтацію векторів, а не їх довжину, що робить її особливо корисною в сценаріях, де величина об’єктів не має вирішального значення. Косинусна подібність широко використовується в інтелектуальному аналізі тексту, аналізі схожості документів і спільному фільтруванні, що забезпечує ефективне порівняння та класифікацію на основі семантичної подібності.

Відстань Хеммінга. Відстань Хеммінга спеціально розроблена для порівняння двійкових даних або категоріальних змінних. Він кількісно визначає кількість позицій, у яких відповідні елементи двох векторів відрізняються. Підраховуючи відмінності між двійковими або категоріальними ознаками, відстань Хеммінга допомагає виявити та виправити помилки, аналізувати послідовність ДНК і кластеризувати дані. Ця міра відстані дає цінну інформацію про відмінності та закономірності в окремих даних.

Scipy. Підмодуль spatial.distance пропонує повний набір показників відстані, включаючи евклідову відстань, манхеттенську відстань і косинусну подібність. Ці функції сприяють ефективному обчисленню відстаней між масивами або векторами, дозволяючи проводити комплексний аналіз і порівняння. Використовуючи можливості Scipy, дослідники та практики можуть легко впровадити ці вимірювання відстані та отримати цінну інформацію зі своїх даних.

Sklearn. Підмодуль sklearn.metrics.pairwise в бібліотеці Sklearn надає багатий набір метрик відстані та заходів подібності. Такі функції, як попарні евклідові відстані, косинус подібності та відстані Хеммінга, пропонують потужні інструменти для точного обчислення відстаней і подібностей. Дослідники можуть використовувати можливості Sklearn для розробки надійних моделей машинного навчання, ефективного аналізу даних і прийняття обґрунтованих рішень на основі схожості та відмінності в своїх даних.

2.2.2 Застосування методів до вирішення бізнес-задачі

Тут ми розраховуємо попарні відстань (або схожість) між векторами всіх невідомих клієнтів і векторами позитивних респондентів. Схожість невідомого клієнта на всіх відомих усереднюємо – це і буде наш прогноз для визначення клієнта як 0/1.

Лістинг 4 — Розрахунок схожості між векторами

```
# ALL FEATURES SIMILARITY

# jaccard dissimilarity
# 0 - equal, 1 - different
# https://stats.stackexchange.com/a/123611/363056
#
https://docs.scipy.org/doc/scipy/reference/generated/scipy.
spatial.distance.cdist.html

start_time = time.perf_counter()

sim_full_mtx = (
```

```
    1 -
pairwise_distances(X_unknown.astype(bool).to_numpy(),
                           X_pos.astype(bool).to_numpy(),
                           metric='jaccard',
                           n_jobs=-1))

sim_full_rate = minmax_scale(np.nansum(sim_full_mtx,
axis=1))

end = time.perf_counter()

print(
    f'[INFO] {datetime.datetime.now()}: classification
based on vector similarity is done in {end -
start_time:.1f} sec.')
```

```
[INFO] 2023-06-30 18:46:34.235823: classification based on
vector similarity is done in 0.5 sec.
```

Розвиток цього методу – визначення схожості за ознаками, які згруповані у підпростори (авторський метод).

Тут спочатку пробуємо згрупувати ознаки між собою так, щоб вони утворили деякі підпростори, в яких ми будемо розраховувати схожість (і робити субпрогнози) за першим методом. Фінальний прогноз для клієнта є середньозваженою всіх прогнозів по підпросторах. Ідея виникла з того, що експертним шляхом можна розбити всю сукупність ознак клієнта, скажімо, на демографічні і споживчі (як він купую товар, із якою частотою тощо), а тоді шукати клієнтів, які схожі і за демографією (по векторам із одним набором ознак) і за споживчими патернами (вектор із другим набором ознак). Ознаки в векторах (підпросторах) не перетинаються. Тут цей метод доведений до деякого ступеня автоматизму: як початковий критерій для розбиття ознак на групи ми використовуємо коефіцієнт Крамера, і утворюємо групи, в яких сума коефіцієнтів Крамера кожної ознаки будуть приблизно рівними.

Лістинг 5 — Розбиття ознак на групи

```
X_train_enc, X_test_enc, y_train, y_test =
train_test_split(
    data,
```

```
    y_known,
    test_size=0.2,
    random_state=RND)
def get_scores_many_spaces(
        dat: dict,
        iter_idx: int,
        X_dat_pos: pd.DataFrame,
        X_dat_neg: pd.DataFrame) -> np.array:

    factor_dists = []

    for group in dat[iter_idx][0][f'-
group_{iter_idx}'].unique():

        cols_set = (
            dat[iter_idx][0].loc[dat[iter_idx][0][f'-
group_{iter_idx}'] == group,
                                 'cols'])

        pos_clients, rst_clients = (
            X_dat_pos[cols_set],
            X_dat_neg[cols_set])

        dist_mtx = (
            1 -
pairwise_distances(rst_clients.astype(bool).to_numpy(),

pos_clients.astype(bool).to_numpy(),
                                   metric='jaccard',
                                   n_jobs=-1))

        factor_dists.append(minmax_scale(np.sum(dist_mtx,
axis=1)))

    return np.average(
        np.concatenate(
            [g.reshape(-1, 1) for g in factor_dists],
axis=1), axis=1,
        weights=res[iter_idx][1])

def split_chunks(ser: pd.Series,
                 random_state: int,
                 num: int = 2) -> pd.DataFrame:

    # https://stackoverflow.com/a/6856593/6025592

    ser = ser.sample(frac=1, random_state=random_state)  #
shuffling
```

```
    res = defaultdict(list)
    sums = {i: 0 for i in range(num)}
    c = 0
    for e in ser.index:
        for j in sums:
            if c == sums[j]:
                res[j].append(e)
                break
        sums[j] += ser[e]
        c = min(sums.values())

    # https://stackoverflow.com/a/42869605/6025592
    return pd.DataFrame(
        [el for sublist in
         (list(zip(v, [k] * len(v))) for k, v
          in res.items()) for el in sublist],
        columns=['index',
'group']).set_index('index').squeeze()
start_time = time.perf_counter()

start, stop, step = (2, len(cat_corr) // 2, 5)

iters = set(map(int, np.linspace(start, stop, step)))

res, cat_corr_df = (defaultdict(list), pd.DataFrame())

preds = pd.Series(
    X_train_enc.columns,
    index=['_'.join(c.split('_')[:-1]) for c in
X_train_enc.columns],
    name='cols')

while iters:

    print(f'[INFO] {datetime.datetime.now()}: subspace grid
is: {iters}')

    for i in iters:

        cat_corr_splitted = split_chunks(
            cat_corr,
            num=i,
            random_state=RND)

        cat_corr_df = cat_corr_df.assign(
            **dict([(f'group_{i}', cat_corr_splitted)]))

        preds_splitted = cat_corr_df[[f'group_{i}']].merge(
            preds, right_index=True, left_on='index')
```

```
        res[i].append(preds_splitted)

        tmp = preds_splitted.merge(
            cat_corr,
            left_on='index',
            right_index=True)

        raw = tmp.groupby(f'-
group_{i}')['corr'].sum().to_numpy()

        res[i].append(raw / sum(raw))

        scores = get_scores_many_spaces(
            res,
            i,
            X_train_enc.iloc[np.where(y_train)[0], :],
            X_test_enc)

        res[i].extend([auc(*np.array(
            precision_recall_curve(y_test, scores),
            dtype='object')[[1, 0]]), scores])

    res_max = max(res, key=lambda k: res.get(k)[2])

    res_keys = sorted(list(res.keys()))

    res_max_idx = res_keys.index(res_max)

    start, stop = (
        res_keys[res_max_idx - 1]
        if res_max_idx != 0 else res_keys[0],
        res_keys[res_max_idx + 1]
        if res_max_idx != len(res_keys) - 1 else res_keys[-
1])

    iters = set(map(int, np.linspace(start, stop,
step))).difference(res_keys)

print(
    f'[INFO] {datetime.datetime.now()}: choose to split
vector into {res_max} sub-spaces')
```

```
[INFO] 2023-06-30 18:46:34.290940: subspace grid is: {2, 3}
[INFO] 2023-06-30 18:46:34.500372: choose to split vector
into 2 sub-spaces
```

```
print(
```

```
    f'[INFO] {datetime.datetime.now()}: choose to split
vector into {res_max} sub-spaces')

sim_factor_rate = get_scores_many_spaces(
    res, res_max, X_pos, X_unknown)

# cm_sim_factor = confusion_matrix(y_true, sim_factor_rate
> THR)

end = time.perf_counter()

print(
    f'[INFO] {datetime.datetime.now()}: classification
based on multi-vector similarity is done in {end -
start_time:.1f} sec.')
```

```
[INFO] 2023-06-30 18:46:34.526590: choose to split vector
into 2 sub-spaces
[INFO] 2023-06-30 18:46:35.057176: classification based on
multi-vector similarity is done in 0.8 sec.
```

2.3 Сучасні алгоритми машинного навчання для бінарної класифікації в пакеті sklearn

2.3.1 Використання класифікаторів Random Forest Classifier / Gradient Boosting Classifier

Random Forest — це алгоритм навчання ансамблю, який використовує набір дерев рішень для прогнозування. Він об’єднує прогнози з кількох дерев рішень, щоб отримати остаточну класифікацію. У цьому розділі ми заглибимося в базові концепції алгоритму Random Forest, включаючи побудову дерев рішень і процес агрегування ансамблю. Крім того, ми опишемо переваги Random Forest, такі як його здатність обробляти великі набори даних, вміщувати відсутні значення та надавати рейтинги важливості функцій. Розуміння цих концепцій і переваг дозволить дослідникам і практикам ефективно використовувати потужність Random Forest для завдань бінарної класифікації.

Gradient Boosting — це ще один алгоритм навчання, який послідовно створює слабких учнів, часто дерева рішень, для створення надійної прогнозної моделі. Він ітеративно покращує продуктивність моделі, зосереджуючись

на примірниках, які раніше були неправильно класифіковані. У цьому розділі ми надаємо детальне пояснення принципів посилення градієнта, включаючи процес посилення, функції втрат і концепцію слабких учнів. Ми також підкреслюємо сильні сторони Gradient Boosting, такі як його здатність фіксувати складні зв'язки, обробляти взаємодію функцій і забезпечувати високу точність прогнозування. Розуміючи принципи та сильні сторони Gradient Boosting, дослідники та практики можуть використовувати цей алгоритм для ефективного вирішення завдань бінарної класифікації (рис. 2.5).

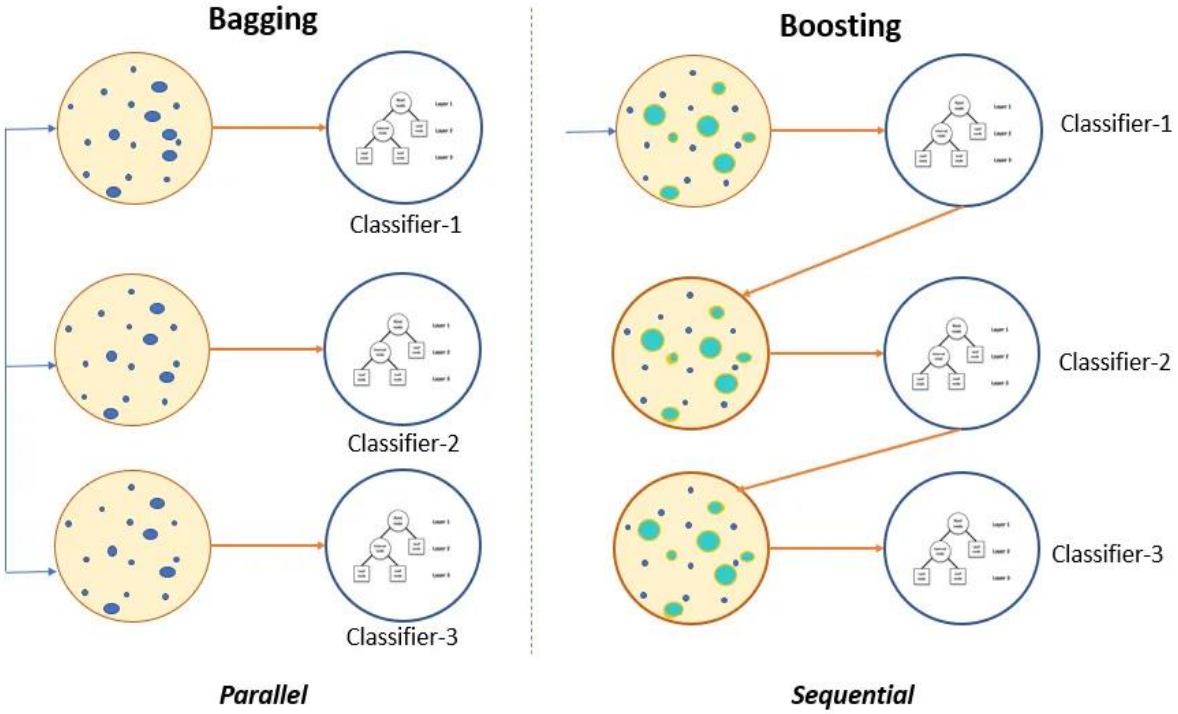

Рисунок 2.5 — Графічна інтерпретація алгоритмів

Незважаючи на те, що алгоритми Random Forest і Gradient Boosting є потужними інструментами для бінарної класифікації, вони стикаються з обмеженнями в сценаріях, де позначаються лише позитивні екземпляри, відомі як позитивні немарковані дані. У цьому розділі ми заглибимося в проблеми, пов'язані з позитивними немаркованими даними, наприклад, класовий дисба-

ланс і упереджені моделі. Ми обговорюємо наслідки цих проблем для продуктивності алгоритмів Random Forest і Gradient Boosting у таких сценаріях. Крім того, ми досліджуємо потенційні стратегії та альтернативні алгоритми, які можуть допомогти пом'якшити ці обмеження та покращити продуктивність моделей двійкової класифікації на позитивних немаркованих даних.

2.3.2 Використання Gradient Boosting для нетипової задачі класифікації

Тут спрощення підходу полягає в тому, що ми приймаємо всі невідомі таргети за негативний клас (0), робимо upsampling позитивного класу (1) і на такому датасеті намагаємося навчити наш алгоритм. При цьому, фактично, отримуємо прогнози для того ж датасету (із невідомими), який і використовували для тренування (як негативний клас). Чим більшою буде фактична частка позитивних респондентів серед невідомих, тим швидше деградуватиме такий підхід (бо ми помилково приймаємо всіх невідомих позитивних респондентів за негативних!).

Лістинг 6 — Використання Gradient Boosting

```
def resample data(*args_ordered_by_target: list,
                  random_state: int = RND) -> tuple:

    dfr dct = dict(enumerate(args_ordered_by_target))

    lst_srt = sorted(dfr_dct.items(), key=lambda x:
x[1].shape[0])

    dfr_up = resample(
        lst_srt[0][1],
        random_state=random_state,
        n_samples=lst_srt[1][1].shape[0],
        replace=True)

    dfr_resampled = np.concatenate((dfr_up, lst_srt[1][1]))

    target = np.array(
        [lst_srt[0][0]] * lst_srt[1][1].shape[0] +
        [lst_srt[1][0]] * lst_srt[1][1].shape[0])

    return (dfr resampled, target)
```

```
X_train_resampled, y_train_resampled =
resample_data(X_unknown, X_pos
print(
    f'[INFO] {datetime.datetime.now()}: upsampled data
shape: {X_train_resampled.shape}')
```

```
[INFO] 2023-06-30 18:46:35.120933: upsampled data shape:
(153994, 43)
```

```
# NAIVE GRADIENT BOOSTING

clf_rf = GradientBoostingClassifier(random_state=RND)

start = time.perf_counter()

clf_rf.fit(X_train_resampled, y_train_resampled)

clf_rf_preds =
clf_rf.predict_proba(X_unknown.to_numpy())[:, 1]

# cm_clf_rf = confusion_matrix(y_true, clf_rf_preds > THR)

end = time.perf_counter()

print(
    f'[INFO] {datetime.datetime.now()}: classification with
naive GB is done in {end - start:.1f} sec.')
```

```
[INFO] 2023-06-30 18:47:04.958254: classification with
naive GB is done in 29.8 sec.
```

2.4 Класифікація табличних даних за допомогою багатошарових нейронних мереж фреймворка TensorFlow

2.4.1 Концепція Deep Learning

Сфера глибокого навчання зробила революцію в науці про дані, дозволивши машинам автоматично аналізувати величезні обсяги даних і робити точні прогнози. Ця стаття має на меті вивчити застосування багаторівневих нейронних мереж у бінарній класифікації табличних даних, розглядаючи зростаючу потребу в ефективних методах класифікації. Використовуючи пакет TensorFlow Python, ми прагнемо пролити світло на потенціал нейронних ме-

реж у вирішенні складних завдань класифікації. Мета полягає в тому, щоб дослідити їхні можливості, порівняти їх із традиційними алгоритмами машинного навчання та продемонструвати їхні реальні застосування в реальних бізнес-завданнях.

Нейронні мережі, натхненні нейронною структурою людського мозку, є потужними обчислювальними моделями, які широко використовуються в глибокому навчанні. У цій статті ми пропонуємо вичерпний огляд нейронних мереж, заглиблюючись у їхні фундаментальні поняття, не перевантажуючи читача надмірними математичними деталями. Ми обговорюємо структуру багаторівневих нейронних мереж, наголошуючи на їх здатності автоматично вивчати ієрархічні представлення даних. Процес навчання, включаючи методи зворотного поширення та оптимізації, пояснюється коротко, щоб забезпечити міцну основу для розуміння їх функціонування.

Нейронні мережі та алгоритми машинного навчання Щоб оцінити ефективність нейронних мереж у задачах бінарної класифікації, ми проводимо порівняльний аналіз із традиційними алгоритмами машинного навчання. Вивчаючи Random Forest і Gradient Boosting, два широко використовувані алгоритми в класифікації, ми прагнемо висвітлити сильні та слабкі сторони кожного підходу. Цей аналіз дозволяє всебічно зрозуміти контексти, де нейронні мережі перевершують або доповнюють традиційні алгоритми. Ми представляємо ключові фактори, такі як вивчення складних патернів в даних, обробка великих даних і їх адаптивність до даних великої розмірності, які сприяють їх перевазі в певних сценаріях.

Далі наведено реальні приклади, які ілюструють успішне використання простих нейронних мереж для завдань бінарної класифікації в важливих бізнес-сферах. Згадуючи такі кейси, як виявлення шахрайства, оцінка кредитного ризику та прогнозування відтоку клієнтів, ми демонструємо практичну значущість нейронних мереж у вирішенні актуальних бізнес-завдань. Кожен приклад містить загальний огляд постановки проблеми, використовуваного набору

даних, методів попередньої обробки даних, вибору архітектури мережі та стратегій налаштування гіперпараметрів.

2.4.2 Використання простої нейронної мережі для нетипової задачі класифікації

Те саме, що і метод, описаний в р. 2.3.2, проте тепер використовуємо просту одношарову нейронну мережу. Кращий результат отримаємо на батчах більшого розміру.

Лістинг 7 — Використання одношарової нейронної мережі

```
def train_model(x: np.array,
                y: np.array,
                epochs: int = 10
                ) -> tf.keras.Sequential:

    #
https://www.tensorflow.org/api_docs/python/tf/keras/utils/set_random_seed
    random.seed(RND)
    np.random.seed(RND)
    tf.random.set_seed(RND)

    callback = tf.keras.callbacks.EarlyStopping(
        monitor='binary_accuracy',
        min_delta=1e-2,
        patience=3,
        restore_best_weights=True)

    model = tf.keras.Sequential(
        [tf.keras.layers.Dense(
            # https://stackoverflow.com/a/14267669/6025592
            units=min(128,
2**math.ceil(math.log2(x.shape[1]))),
            activation='relu',
            input_shape=(x.shape[-1],)),
         tf.keras.layers.Dropout(0.1),
         tf.keras.layers.Dense(
            units=1,
            activation='sigmoid')])

    model.compile(
        loss=tf.keras.losses.BinaryCrossentropy(),
```

```
        metrics=tf.keras.metrics.BinaryAccuracy())

    batch_size = min(
        MAX_BATCH,
        max(
            MIN_BATCH,
            2**math.floor(math.log2(x.shape[0] /
BATCH_COEF))))

    model.fit(
        x=x,
        y=y,
        batch_size=batch_size,
        epochs=epochs,
        callbacks=[callback],
        verbose=False)

    return model

# NAIVE NEURAL NET

# upsampling performs better than official tutorials
#
https://keras.io/examples/structured_data/imbalanced_classi
fication/
#
https://www.tensorflow.org/tutorials/structured_data/imbala
nced_data

start = time.perf_counter()

clf_nn = train_model(X_train_resampled, y_train_resampled)

clf_nn_preds = clf_nn.predict(X_unknown, verbose=False)

end = time.perf_counter()

print(
    f'[INFO] {datetime.datetime.now()}: classifiaction with
naive NN is done in {end - start:.1f} sec.')
```

```
[INFO] 2023-06-30 18:47:58.316067: classifiaction with
naive NN is done in 53.3 sec.
```

2.4.3 Побудова ансамблю нейронних мереж для вирішення бізнес-задачі

При такому підході ми будуємо багато одношарових нейронних мереж, які тренуємо на різних випадкових датасетах невеликого обсягу. Кожний датасет - 90% відомих нам позитивних респондентів (розміром X) + випадково обрані невідомі клієнти (розміром X * 3). Таким чином, кожна із нейронних мереж бачить невелику кількість невідомих клієнтів як негативний клас, і таким чином ми зменшуємо ймовірність помилкового маркування невідомих як негативних. Кращий результат на батчах меншого розміру (пропорційно розмірам датасету). Фінальний прогноз 0/1 для клієнта - середня прогнозів всіх мереж.

Лістинг 8 — Використання ансамблю нейронних мереж

```
# NN ENSEMBLE

k_neg = math.ceil(X_unknown.shape[0] * 0.05)

X_neg_idx =
random.Random(RND).sample(range(X_unknown.shape[0]), k_neg)

# https://stackoverflow.com/a/64100245/6025592
chunks = np.array_split(X_neg_idx, MAX_NN)

def do_parallel(
        _fun: Callable,
        _itr: Iterable,
        concatenate_result: bool = True,
        **kwargs: dict) -> np.array:

    with parallel_backend('loky', n_jobs=-1):
        lst_processed = Parallel()(
            delayed(_fun)(el, i=i, **kwargs)
            for i, el in enumerate(_itr))

    if concatenate_result:
        return np.concatenate(
            [arr.reshape(-1, 1) for arr in lst_processed],
axis=1)

    return lst_processed

def make_model(
```

```
        idx: np.array,
        data_neg: np.array,
        data_pos: np.array,
        fun_train: Callable,
        fun_resample: Callable,
        pos_size: int,
        i: int,
        **kwargs: dict) -> tf.keras.Sequential:

    pos_rnd = data_pos.iloc[random.Random(i).sample(
        range(data_pos.shape[0]),
        k=pos_size), :]

    neg_rnd = data_neg[idx, :]

    # pass neg first
    model = fun_train(*fun_resample(neg_rnd, pos_rnd),
**kwargs)

    return (model,
model.history.history['binary_accuracy'][-1])

def get_preds(model: tf.keras.Sequential,
              dat: np.array,
              **kwargs: dict
              ) -> np.array:

    return model.predict(dat, verbose=False)
start = time.perf_counter()

with warnings.catch_warnings():

    warnings.simplefilter('ignore')

    models = do_parallel(
        make_model,
        chunks,
        data_neg=X_unknown.to_numpy(),
        data_pos=X_pos,
        fun_train=train_model,
        fun_resample=resample_data,
        pos_size=math.ceil(X_pos.shape[0] * 0.9),
        concatenate_result=False)

    preds = do_parallel(
        get_preds,
        [m[0] for m in models],
        dat=X_unknown)
```

```
accuras = [model[1] for model in models]
weights = [e / sum(accuras) for e in accuras]

preds_many_nn = np.average(preds, axis=1, weights=weights)

end = time.perf_counter()

print(
    f'[INFO] {datetime.datetime.now()}: classifiaction with
NN ensemble is done in {end - start:.1f} sec.')
```

```
[INFO] 2023-06-30 18:48:52.569679: classifiaction with NN
ensemble is done in 54.2 sec.
```

2.5 Висновки до розділу 2

Відкриті джерела даних, такі як Kaggle, дають нові можливості для наукових досліджень та бізнесу в галузі науки про дані. Kaggle надає високоякісні набори даних, які можуть служити еталоном для оцінки моделей машинного навчання. У цьому контексті, компанії можуть використовувати ці дані для розробки стратегій перехресних продажів, що призводить до підвищення прибутку та клієнтського задоволення.

Для бізнес-задачі перехресних продажів використовується набір даних, де позитивні респонденти визначаються як клієнти, які вже придбали товар чи послугу. Основна мета полягає в знаходженні схожих клієнтів для пропозицій щодо придбання аналогічного товару. Важливо надавати перевагу методам з високим recall у бізнесі, навіть за рахунок зниження точності алгоритму, особливо при низькій собівартості контакту.

Оптимізація набору даних включає зменшення обсягу, залишення 15% позитивних респондентів і використання лише 1% для навчання, із «прихованими» позитивними респондентами. Аналіз буде проведено на підготовленому датасеті з використанням перетворень, таких як перетворення неперервних предикторів на дискретні та one-hot-encoding для категоріальних предикторів. Також планується розрахунок кореляції категоріальних змінних із цільовою змінною за допомогою коефіцієнта Крамера для розбиття простору ознак на підпростори.

В роботі висвітлюються основні міри відстані, такі як Евклідова та Манхеттенська, а також методи подібності, включаючи косинус та відстань Хеммінга. Зручні інструменти для обчислення цих відстаней надають бібліотеки Scipy та Sklearn, що спрощує їх використання у роботі дослідників та практиків. Один із методів оцінки схожості, представлений у роботі, використовує попарні відстані або схожість між векторами невідомих клієнтів і векторами позитивних респондентів. Прогноз для клієнта обчислюється як середнє значення його схожості з усіма позитивними респондентами. Розвиток методу полягає в визначенні схожості за групами ознак, формуючи підпростори. Цей підхід дозволяє автоматизувати розбиття ознак на групи, використовуючи коефіцієнт Крамера для формування груп з рівними сумами для кожної ознаки в групі.

В роботі розглянуто два потужні алгоритми для бінарної класифікації, а саме Random Forest та Gradient Boosting, розкрито їхні принципи та переваги. Особлива увага приділяється їхнім можливостям обробки великих даних та вирішенню проблем, таких як класовий дисбаланс. На прикладі використання цих алгоритмів продемонстровано низьку ефективність «класичного» підходу до класифікації у сценаріях PU - навчання.

В кінці розділу запропоновано метод, який використовує просту одношарову нейронну мережу. У цьому методі будується кілька одношарових нейронних мереж, кожну тренують на різних невеликих випадкових датасетах. Цей підхід дозволяє обмежити кількість невідомих клієнтів, яких кожна мережа бачить як негативний клас, що сприяє зменшенню ймовірності помилкового маркування невідомих як негативних. Особливо ефективні результати досягаються на менших батчах, пропорційних розміру датасету. Фінальний прогноз для клієнта формується як середнє значення прогнозів всіх мереж.

РОЗДІЛ 3 РОЗРОБКА КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ КЛІЄНТІВ ДЛЯ ПЕРЕХРЕСНИХ ПРОДАЖІВ

3.1 Загальний огляд архітектури рішень з аналітики даних компанії RBC Group

Найбільш типова / загальна архітектура / сценарій рішення з аналітики даних компанії RBC Group передбачають узгоджену роботи блоків завантаження / вивантаження даних та блоків скриптів із машинного навчання під управління системи оркестрування (як правило із графічним інтерфейсом), наприклад Apache Airflow.

Типова архітектура може бути змінені та/або адаптована відповідно до IT-ландшафту замовника і специфікації проєкту, та забезпечує оптимальну сумісність з його цілями та вимогами.

Наріжним каменем будь-якого рішення із аналітики даних виступає блок скриптів із машинного навчання. Послідовність виконання та функціональне наповнення скриптів цього блоку визначаються специфікою проєкту, обсягом і ступенем підготовки вхідних даних та результатами проведеної RnD-фази проєкту. Зокрема, цей блок відповідає за очистку та обробку даних, feature engineering, створення / навчання / тестування та оптимізацію ML/DL-моделей, їх використання для отримання прогнозувань залежно від задач. Після визначення конкретної архітектури внутрішнього DS-конвеєра, його опис, як правило, включений до супровідної документації проєкту (рис. 3.1).

Розробка рішення виконується в середовищі розробки (рис. 3.2), що розташоване на сервері під управлінням операційної системи Windows. Це може бути як сервер замовника, так і сервер RBC Group. За наявності корпоративних облікових записів на платформах GitLab або GitHub, замовник ініціює процес створення та налаштування репозиторію (в іншому випадку, замовнику надається доступ до такого репозиторію).

Репозиторій призначений для зберігання та оновлення вихідного коду, пов'язаного з розробкою. Команда RBC Group вносить зміни до вихідного коду в цьому репозиторії відповідно до вимог проєкту. Після завершення певних етапів розробки, код може бути помічений відповідними тегами, які вказують на стан «релізу» або версії проєкту.

Реліз проєкту включає повний пакет скриптів для створення Docker-image. Замовник розгортає (налаштовує автоматичне розгортання) Docker-image на своєму сервері, запускаючи застосунок у контейнері. Docker-image буде включати поточний реліз Airflow-image (або подібної системи) під задану версію Python (3.9-3.11), а також визначену БД (в режимі PVC / connected locally) за необхідності. Подальше розгортання / формування Docker-image передбачає встановлення набору open-source Python-модулів для аналізу даних (із файлу requirements.txt).

За такого підходу в процесі розробки будуть задіяні окремі елементи методології CI/CD (зберігання коду в репозиторії, використання тегів для позначення релізів), що дозволятиме швидко оновлювати рішення на сервері Замовника шляхом заміни Docker-image новою версією. Розгортання застосунку, як правило, відбувається в одному контейнері, для чого замовнику достатньо використовувати менеджер Docker.

Для перевірки функціональності ізольованого контейнера створюється тестове середовища на сервері з операційною системою Linux. Рішення RBC Group, як розробника, не передбачають додаткову оркестрацію, горизонтальне масштабуванням або використанням Helm для керування цими процесами на кластері Kubernetes. Виходячи із завдань проєкту, обсягів та характеру вихідних (табличних) даних, на проєкті може передбачатися використання GPU.

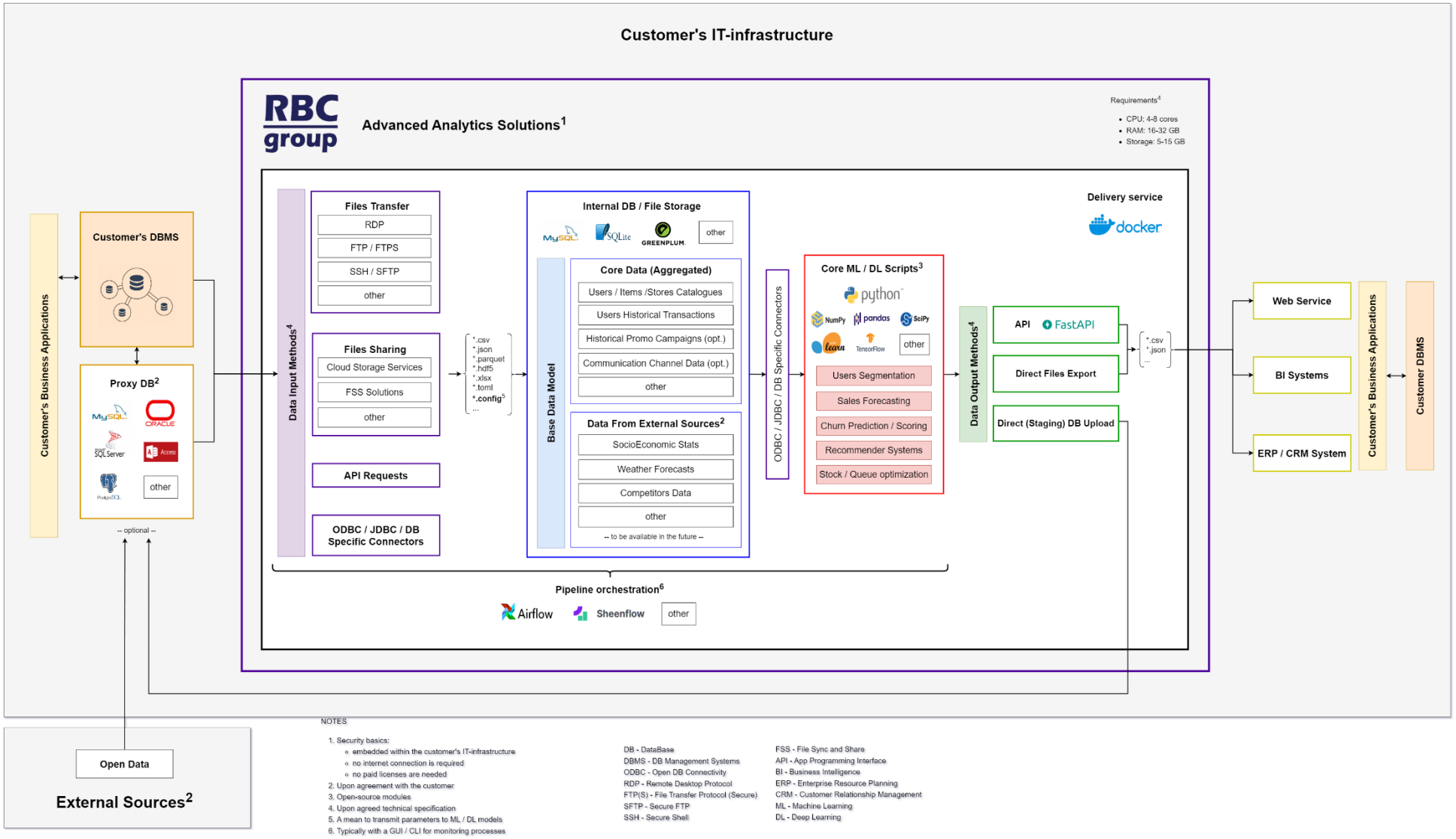

Рисунок 3.1 — Загальна архітектура програмної системи

Рішення з аналітики даних, як правило, представлять собою back-end, який забезпечуватиме наявність необхідних даних для «зовнішніх» систем замовника. Рішення RBC Group часто не мають повноцінного інтерфейсу (UI) для управління. Проте, за необхідності, замовник має можливість налаштувати визначені параметри застосунку в цілому та/або окремих його елементів (ML/DL моделей) шляхом редагування конфігураційних файлів, які будуть завантажуватися в загальному потоці даних

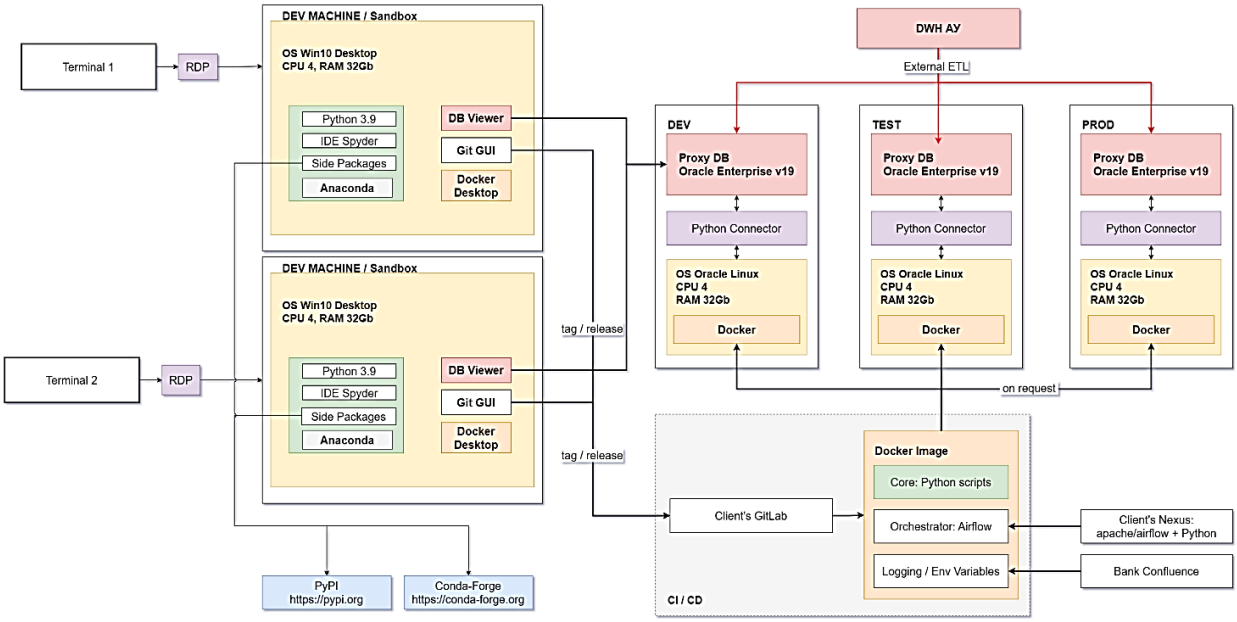

Рисунок 3.2 — Розгортання dev environment в IT-ландшафті замовника

Для оркестрування окремих процесів всередині контейнеру із застосунком використовуватися платформа Apache Airflow (або аналогічна система), яка має вбудований графічний інтерфейс (GUI) та командний рядок (CLI) для моніторингу стану виконання всіх завдань. В разі потреби та за умови узгодженням із замовником, така система може бути налаштована для відправки сервісних повідомлень через електронну пошту та/або месенджери. Замовник самостійно визначає методи авторизації та рівні доступу свого персоналу до можливостей налаштування (через редагування конфігураційних файлів) та моніторингу стану процесів у застосунку. Крім того, він узгоджує з RBC Group необхідні налаштування для сервісу відправлення повідомлень з контейнеру.

Всередині Docker-контейнера Apache Airflow (або подібна системи) створює службову внутрішню БД (як правило, MySQL), яка зберігає конфігураційні дані, журнали та метадані оркестратора. На додаток до неї, всередині контейнера може бути створена допоміжна внутрішня БД (в режимі PVC / connected locally) для зберігання проміжних результатів розрахунків та/або артефактів основних ML/DL скриптів. Ці дані можуть включати агреговані таблиці, вектори для аналізу, метрики моделей тощо.

На відміну від внутрішніх БД, в архітектурі застосунку Proxy DB розглядається як зовнішнє (по відношенню до Docker-контейнера) джерело для тимчасового зберігання вхідних даних (потенційно, акумульованих із розрізнених систем Замовника та/або зовнішніх джерел). Також, Proxy DB може служити проміжним етапом для тимчасового зберігання кінцевих результатів розрахунків, наприклад, статистики та міток клієнтів або кластерів.

Замовник самостійно визначає формат і перелік доступних джерел даних для аналізу. В залежності від задач це можуть бути дані профілів клієнтів та інформація щодо їх транзакцій, в т.ч. у вигляді узгоджених агрегатів, наприклад, середніх сум/частоти транзакцій, статистики користування продуктами замовника тощо. Відповідно до наявних джерел даних замовник визначає прийнятний і зручний для себе спосіб їх передачі - пряме вивантаження файлів, прямий доступ до Customer DBMS, акумуляція даних і прямий доступ до Proxy DB. Якщо в ході реалізації RnD-фази проєкту буде обґрунтовано доцільність використання в подальшому також і External Data Sources, то команда розробника узгоджує з замовником їх перелік і спосіб отримання (з урахуванням вимог безпеки). Дані з усіх джерел формують загальну модель даних (data model: core tables + external data sources), із якою працюють основні ML/DL скрипти всередині контейнера.

Розгортання та обслуговування Proxy DB / Customer DBMS здійснює замовник, в т.ч. визначає технологію зовнішніх БД (по відношенню до Docker-контейнера). Технологію БД всередині Docker-контейнера визначає команда

розробника. Внутрішні БД контейнера не інтегруються прямо із DBMS замовника, тому вибір технологій / способів передачі даних замовником не обмежується архітектурою застосунку.

Стандартним методом видачі результатів розрахунків є завантаження даних безпосередньо в Proxy DB / Customer DBMS. В разі потреби та за умови узгодженням із замовником, може бути розроблено розширений функціонал щодо видачі результатів розрахунків, який передбачає використання API.

3.2 Вимоги та особливості побудови застосунку для прогнозування поведінки клієнтів

3.2.1 Огляд веб-фреймворків для Python

Аналітика даних – ключовий інструмент для підвищення ефективності бізнесу, який передбачає етапи збору, аналізу та інтерпретації даних з різних джерел для виявлення прихованих закономірностей та тенденцій.

Демо-застосунки та MVP проекти стають важливим кроком у розробці аналітичних рішень, дозволяючи замовнику оцінити потенціал рішення та внести необхідні корективи до вимог. Використання фреймворків для швидкої розробки демо-застосунків та дата-вітрин дозволяє розробникам швидко та ефективно створювати прототипи аналітичних рішень.

Рішення з аналітики даних від компанії RBC Group, як правило, є back-end, який забезпечує передачу підготовлених даних для «зовнішніх» систем замовника. Однак до початку проекту часто виникає потреба продемонструвати замовнику потенціал його даних, описати, як виглядатиме рішення, обговорити концептуально процес обробки даних та бізнес-результати для замовника.

Такі реалії вимагають наявності онлайн-вітрин, які допомагають замовнику краще уявити всі можливості аналітичних продуктів. Тому у компанії виникає необхідність швидко розробляти демо-застосунки та MVP проекти, ви-

користовуючи переважно мову програмування Python, якою користуються розробники для написання аналітичних скриптів. Отже, на певному етапі стає важливим вибрати найбільш підходящі для цих цілей інструменти, зокрема фреймворки для швидкої розробки демо-застосунків та дата-вітрин.

У даному розділі коротко представимо порівняльний аналіз фреймворків для швидкої розробки демо-застосунків та дата-вітрин, оцінимо їх придатність для використання в аналітичній компанії.

Веб-розробка може бути досить складним завданням, що вимагає залучення мультидисциплінарної команди, яка має досвід у розробці інтерфейсу, бекенду та серверного програмного забезпечення. Традиційно розробники повного циклу, які володіють ноу-хау для всього процесу розробки, вдавалися до використання JavaScript, PHP або Perl для розробки веб-додатків, а Python висували на другий план як локальну мову сценаріїв. Це було пов'язано з тим, що Python за своєю суттю не призначений для роботи в Інтернеті і потребує веб-фреймворку для взаємодії з веб-серверами та браузерами. Однак з роками спільнота розробила кілька нових фреймворків, які дозволяють ефективно використовувати Python в Інтернеті. А враховуючи те, що Python робить акцент на простоті, читабельності, багатій екосистемі бібліотек та відкритому вихідному коду, він впевнено перетворився на одну з основних мов веб-скриптів, яку обирають багато розробників.

Як правило, такі веб-фреймворки бувають двох типів: повностекові та неповностекові. Вони керують усім - від комунікацій та інфраструктури до інших низькорівневих абстракцій, необхідних для веб-додатків. Нетривіальні додатки вимагають цілого ряду функцій, включаючи, але не обмежуючись ними, інтерпретацію запитів, створення відповідей, зберігання даних і рендеринг користувацьких інтерфейсів.

Для таких додатків часто використовують фреймворк з повним стеком, який забезпечує внутрішнє рішення для всіх технічних вимог. Це контрастує з неповностековими фреймворками, також відомими як мікрофреймворки, які забезпечують мінімальний рівень функціональності, зазвичай обмежуючись

маршрутизацією HTTP-запитів до відповідних контролерів, диспетчеризацією контролера і подальшим поверненням відповіді. Такі фреймворки зазвичай об'єднуються з іншими API та інструментами для створення додатків [18, 5].

Деякі з найпопулярніших прикладів кожного типу, які компанія RBC Group розглядає / використовує для розробки демо-проєктів, коротко описано далі.

3.2.1.1 Фреймворк Flask

Flask - це неповний стек або мікрофреймворк, який надає сервер додатків, не пропонуючи багато інших компонентів. Flask складається з двох основних елементів: Werkzeug, інструменту, який забезпечує підтримку HTTP-маршрутизації, та Jinja, шаблонізатора, який використовується для рендерингу базових HTML-сторінок. Крім того, Flask використовує MarkupSafe як бібліотеку обробки рядків та ItsDangerous як безпечну бібліотеку серіалізації даних для зберігання даних сеансу у вигляді файлів cookie.

Flask - це мінімалістичний фреймворк, який оснащений мінімальним набором компонентів, необхідних для рендерингу веб-додатків. Отже, розробнику надається велика автономія, а також відповідальність за створення власного додатку. Як результат, Flask найкраще підходить для статичних веб-сайтів і для досвідчених розробників, які здатні забезпечити більшу частину власної інфраструктури та рендерити власні інтерфейси.

3.2.1.2 Фреймворк Django

Django дозволяє розробникам створювати складні додатки з відносно меншими накладними витратами порівняно з Flask. Зокрема, Django дозволяє програмістам рендерити динамічний контент з покращеною масштабованістю, а також власними можливостями для взаємодії з системами баз даних за допомогою об'єктно-реляційного відображення.

Крім того, існує безліч інших модулів, включаючи, але не обмежуючись ними, пакети для електронної комерції, автентифікації та кешування, які дозволяють розробнику з легкістю надавати розширені послуги. У поєднанні з безліччю інших сторонніх пакетів, Django дозволяє розробнику зосередитися на ідеї, не турбуючись про реалізацію.

3.2.1.3 Фреймворк Dash

Dash - це веб-фреймворк, розроблений компанією Plotly для візуалізації веб-додатків корпоративного рівня на Python, R і навіть Julia. Враховуючи, що Plotly переважно розробляє інструменти для аналізу та візуалізації даних, Dash частіше використовується для створення дашбордів. Тим не менш, за допомогою Dash можна створювати безліч додатків загального призначення завдяки його розширеній кастомізації.

Dash має можливість підтримувати діаграми D3.js і надає шаблони HTML і CSS за замовчуванням. Однак, для створення більш адаптованих інтерфейсів розробники повинні самі добре розбиратися у фронтенд-програмуванні. Крім того, Dash пропонує корпоративний пакет, який дозволяє досвідченим розробникам розгортати свої додатки в хмарі з можливостями виробничого рівня, такими як системи аутентифікації та бази даних.

3.2.1.4 Фреймворк Web2Py

Web2Py - це повностековий веб-фреймворк для Python, який, як і Django, використовує архітектурну парадигму контролера представлення моделі. Він дозволяє розробникам створювати примітивний, але динамічний контент з відносною легкістю і взаємодіє з системами баз даних. Новизна цього фреймворку, на відміну від інших, полягає в тому, що він постачається з власним інтегрованим веб-середовищем розробки, оснащеним системою тікетів для відстеження та управління помилками.

Однак головним недоліком є те, що Web2Py виконує об'єкти і контролери через єдине глобальне середовище, яке ініціюється при кожному HTTP-

запиті. Хоча це має свої переваги, це також несе в собі підводні камені у вигляді зниження продуктивності та проблем несумісності з певними модулями.

3.2.1.5 Фреймворк Streamlit

Можливість легко розробляти веб-додатки безпосередньо з Python зробила Streamlit цінним інструментом для створення інтерфейсів, відображення тексту, візуалізацію даних, відображення віджетів та управління веб-додатком від створення до розгортання за допомогою зручного та інтуїтивно зрозумілого інтерфейсу прикладного програмування. Streamlit використовує вбудовані ReactJS-компоненти, згруповані для створення повноцінного JavaScript-додатку, із можливістю створення кастомних і складних компонентів, які не надаються «з коробки».

3.2.2 Огляд функціоналу та UI застосунку для прогнозування поведінки клієнтів

Перехресні продажі є критично важливою стратегією для фінансових компаній, яка передбачає отримання значного економічного ефекту завдяки зниженню витрат на залучення клієнтів. Однак лише кожна п'ята компанія у страховому секторі, ефективно використовує переваги перехресних продажів. Так, понад 60% існуючих клієнтів страхових компаній наразі мають лише один страховий поліс, і лише 10% мають три поліси або більше.

Перешкоди на шляху до використання галуззю потенціалу перехресних продажів пов'язані зі складністю визначення цільової аудиторії та/або продукту для здійснення пропозиції. Це призводить до неефективного розподілу ресурсів компанії, незручностей для клієнтів, зниження лояльності та загального незадовільного клієнтського досвіду.

Для вирішення цієї проблеми нами розроблено рішення, яке розширює можливості фінансових компаній і робить тренування/використання складних ML/DL моделей доступним для менеджерів з продажу та агентів, тобто працівників без відповідного технічного/аналітичного досвіду.

Розроблена комп’ютерна система прогнозування поведінки клієнтів для перехресних продажів має інтуїтивно зрозумілий користувацький інтерфейс, спрощений до декількох кроків: завантаження файлу даних клієнтів, вибору цільової аудиторію (визначення колонки із цільовою міткою), оцінки прийнятності точності прогнозування, отримання переліку потенційних клієнтів для обраного фінансового продукту/послуги [26].

В основі рішення знаходяться два основних функціональних блоки: блок AutoML для попередня обробка даних без участі користувача і блок навчання ансамблю нейронних мереж.

Блок AutoML виконує початкову обробку даних і видалення надлишкових ознак, включаючи перетворення часових даних у числовий формат, вилучення стовпців з надмірною кількістю відсутніх значень, високою кардинальністю або низькою дисперсією, вилучення стовпців з однією домінуючою категорією, вилучення стовпців з низькою кореляцією до цільової змінної. Після цього числові ознаки бінаризуються, і таким чином всі стовпці початкового набору даних трансформуються на категоріальні ознаки.

Блок навчання нейронних мереж відповідає за ідентифікацію клієнтів, найбільш схожих на тих, які були визначені користувачем. Цей блок використовує ансамбль нейронних мереж, кожна з яких побудована і навчена на окремих випадкових наборах даних невеликого обсягу. Кожен із таких наборів даних складається з 90% відомих позитивних спостережень і випадково вибраних умовно-негативних спостережень. Така стратегічна композиція гарантує, що кожна нейронна мережа навчається на невеликому обсягу умовно-негативних спостережень, позначених як негативний клас, зменшуючи таким чином ймовірність помилкового прогнозування непозначених клієнтів як негативних.

Фактично послідовність виконання скриптів та взаємозв’язок UI із описаними вище функціональними блоками представлено на діаграмі:

Налаштування функціональних блоків зводять до мінімуму необхідність взаємодії користувача з інтерфейсом системи. Зокрема, така взаємодія описується наступної послідовністю дії і відображенням відповідних проміжних результатів в UI (рис. 3.3.):

- збір і завантаження даних: користувач готується до взаємодії із системою шляхом отримання інформації про клієнтів з CRM-системи компанії, і завантажує профілі клієнтів у форматі електронної таблиці щоразу при ініціалізації нової промо-кампанії з перехресних продажів;
- інформування про перетворення даних: блок AutoML проводить первинну оцінку якості даних, а потім оптимізує їх для навчання нейронної мережі;
- інформування про розподіл даних для тестування: на цьому етапі без участі користувача випадковим чином відбирається невелика частина даних, щоб оцінити точність алгоритму класифікації;
- інформування про метриками точності: система формує візуалізації і роз’яснює отримані показники точності прогнозування для завантаженого набору даних. Якщо результати незадовільні, користувач може додати/змінити поля у вхідному наборі даних і завантажити його повторно;
- використання калькулятора кампанії: користувач має простий інструмент для оцінки потенційної фінансові вигоди від перехресних продажів для визначених системою клієнтів, і приймає рішення про вивантаження переліку потенційних клієнтів.

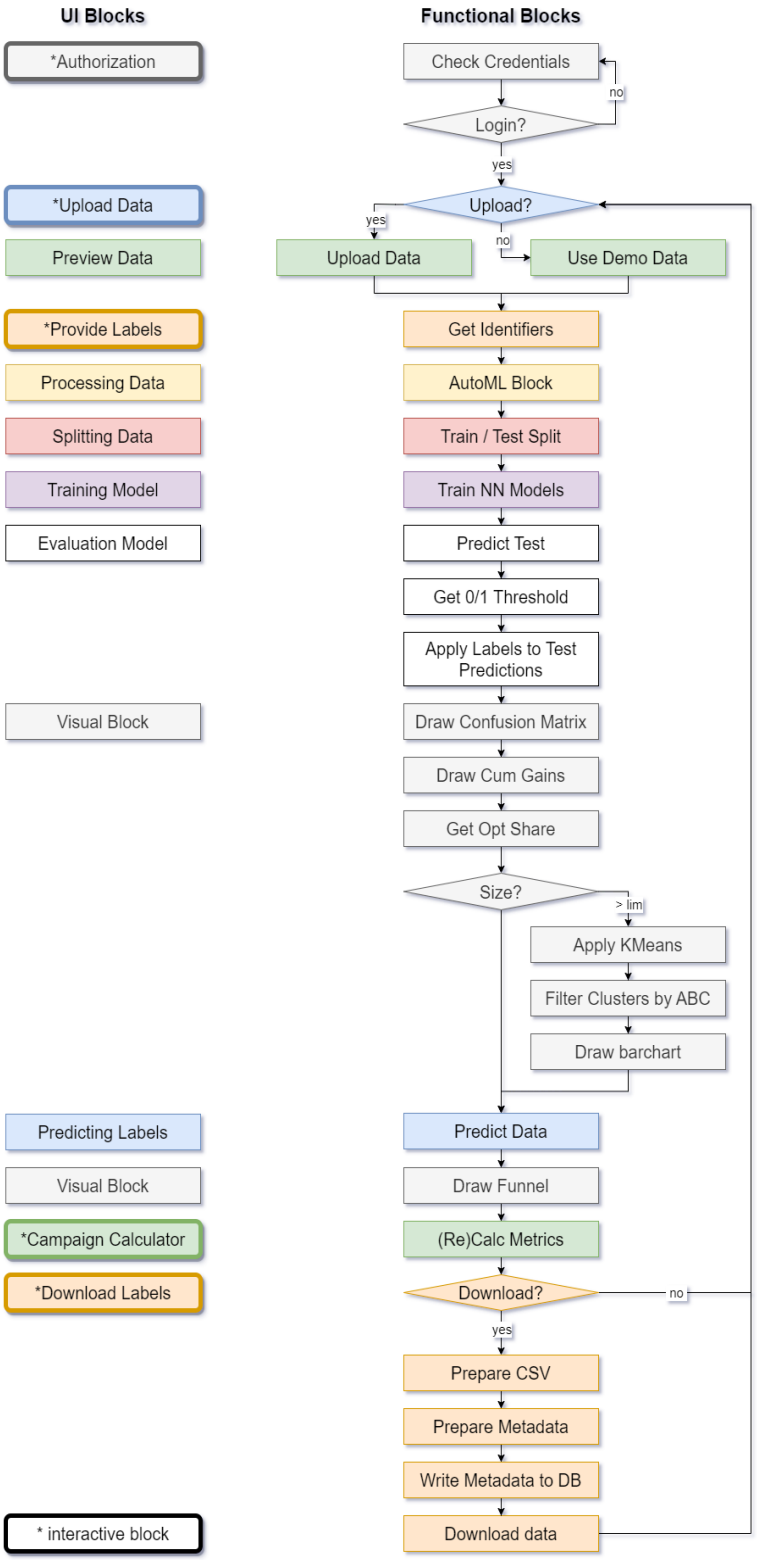

Рисунок 3.3 — Взаємозв’язок UI та функціональних блоків в застосунку

Таким чином, запропоноване рішення робить перехресні продажі простішими та ефективнішими. Воно використовує машинне / глибоке навчання

для ідентифікації клієнтів, які найбільше ймовірно зацікавляться додатковими продуктами або послугами.

З точки зору UI користувач здійснює наступні основні кроки для взаємодії із додатком для отримання результату після авторизації (рис. 3.4).

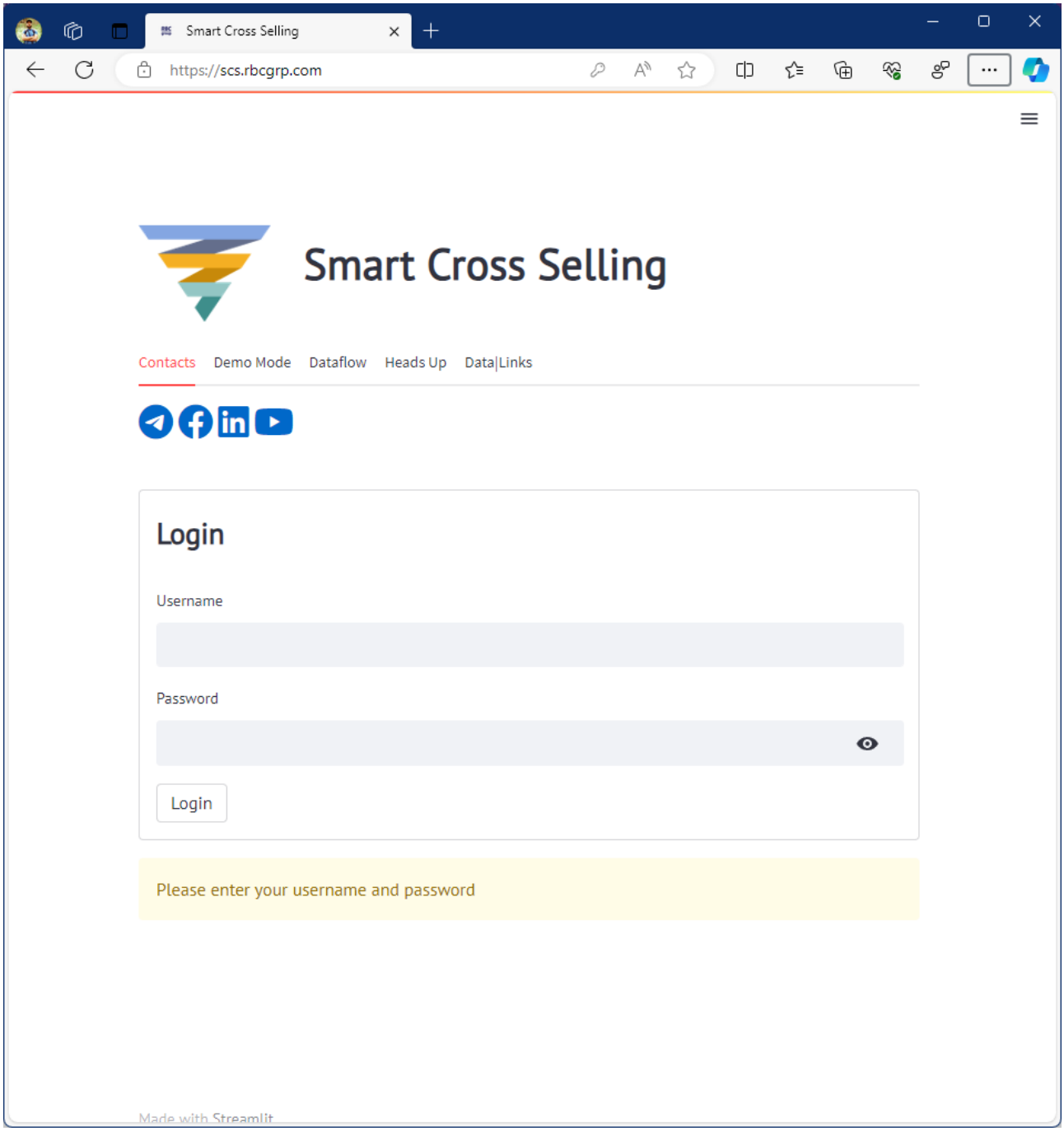

Рисунок 3.4 — Вікно ідентифікації користувача

Перший крок: завантажує інформацію із корпоративної CRM-системи або звертається до аналітичного відділу, щоб отримати оновлені профілі клієнтів у простому форматі електронної таблиці щоразу, коли він запускає нову кампанію з перехресних продажів. Таблиця може містити стільки колонок, скільки є в профілі клієнта у вашій CRM-системі. Ці колонки можуть бути категоріальними, тобто текстовими, або континуальними, тобто числовими. Допускається наявність колонок у форматі DateTime (рис. 3.5).

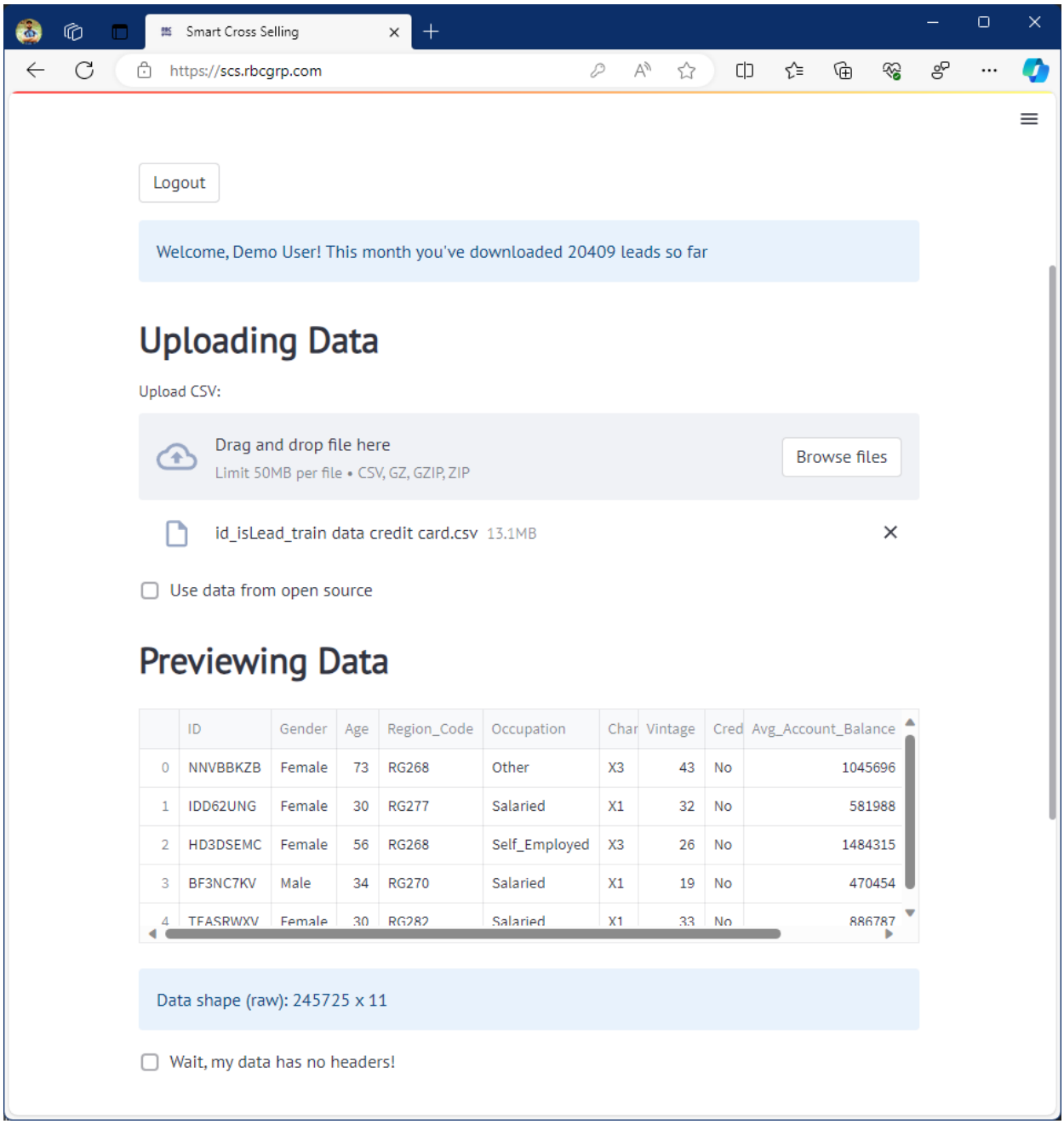

Рисунок 3.5 — Інтерфейс системи

Єдиною умовою для набору даних є наявність ідентифікатора клієнта і цільової мітки, які ви повинні вказати безпосередньо в інтерфейсі користувача (рис. 3.6). Цільова мітка повинна містити тег для клієнтів у наборі даних, які вже придбали певний продукт (в рамках поточної промо-кампанії).

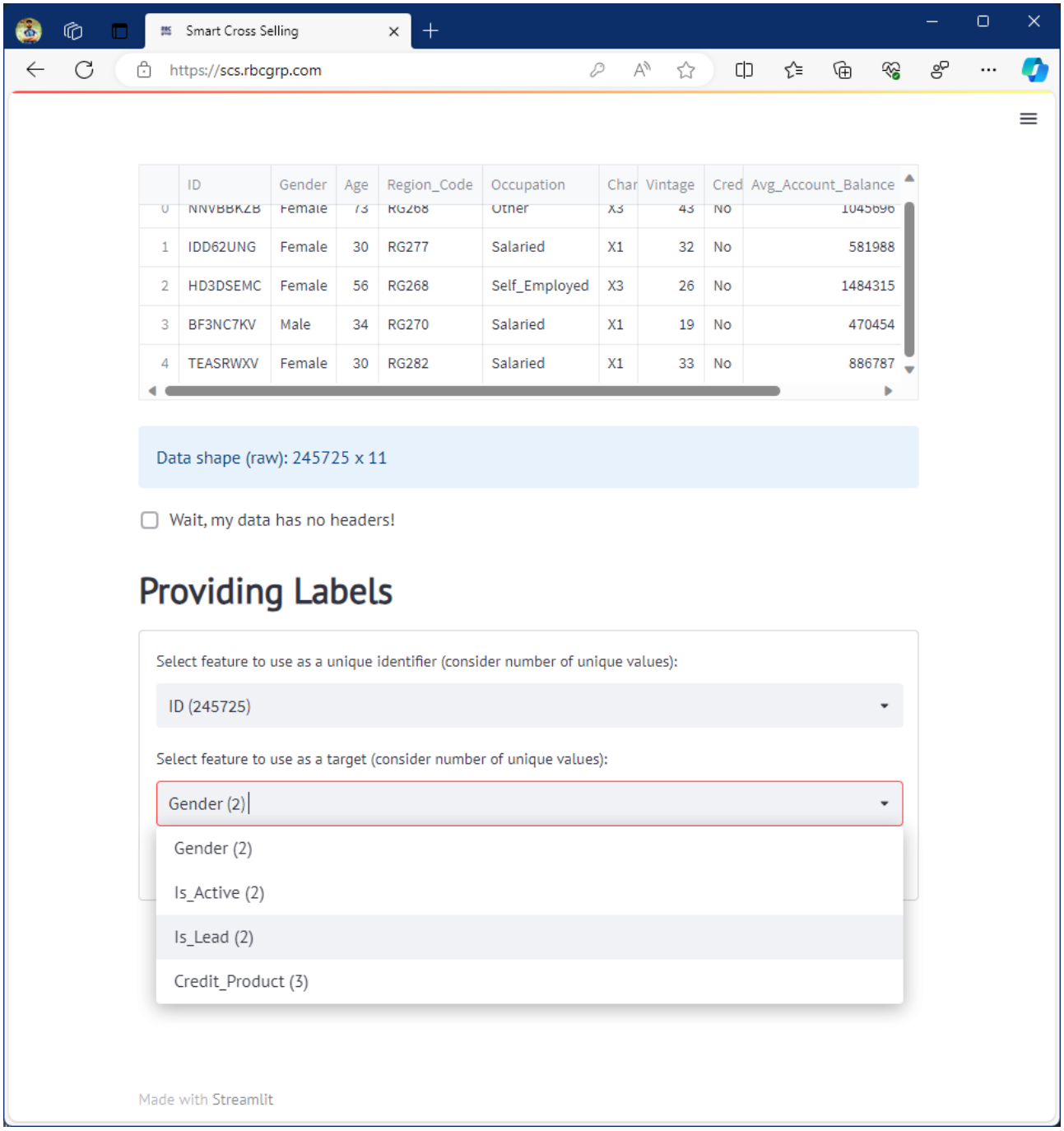

Рисунок 3.6 — Задання міток датасету

Другий крок: модуль автоматичного машинного виконує перевірку якості даних та їх подальше перетворення для навчання нейронної мережі. На цьому етапі ми можемо видалити деякі малорелевантні стовпці, але зазвичай ми модифікуємо їх і залишаємо стільки, скільки потрібно для подальшого використання.

Третій крок: ми випадковим чином виберемо невелику частину даних, щоб перевірити точність алгоритму класифікації. Використовуючи решту даних, ми спрогнозуємо схожість непозначених клієнтів із позначеними за допомогою ансамблю нейронних мереж (рис. 3.7).

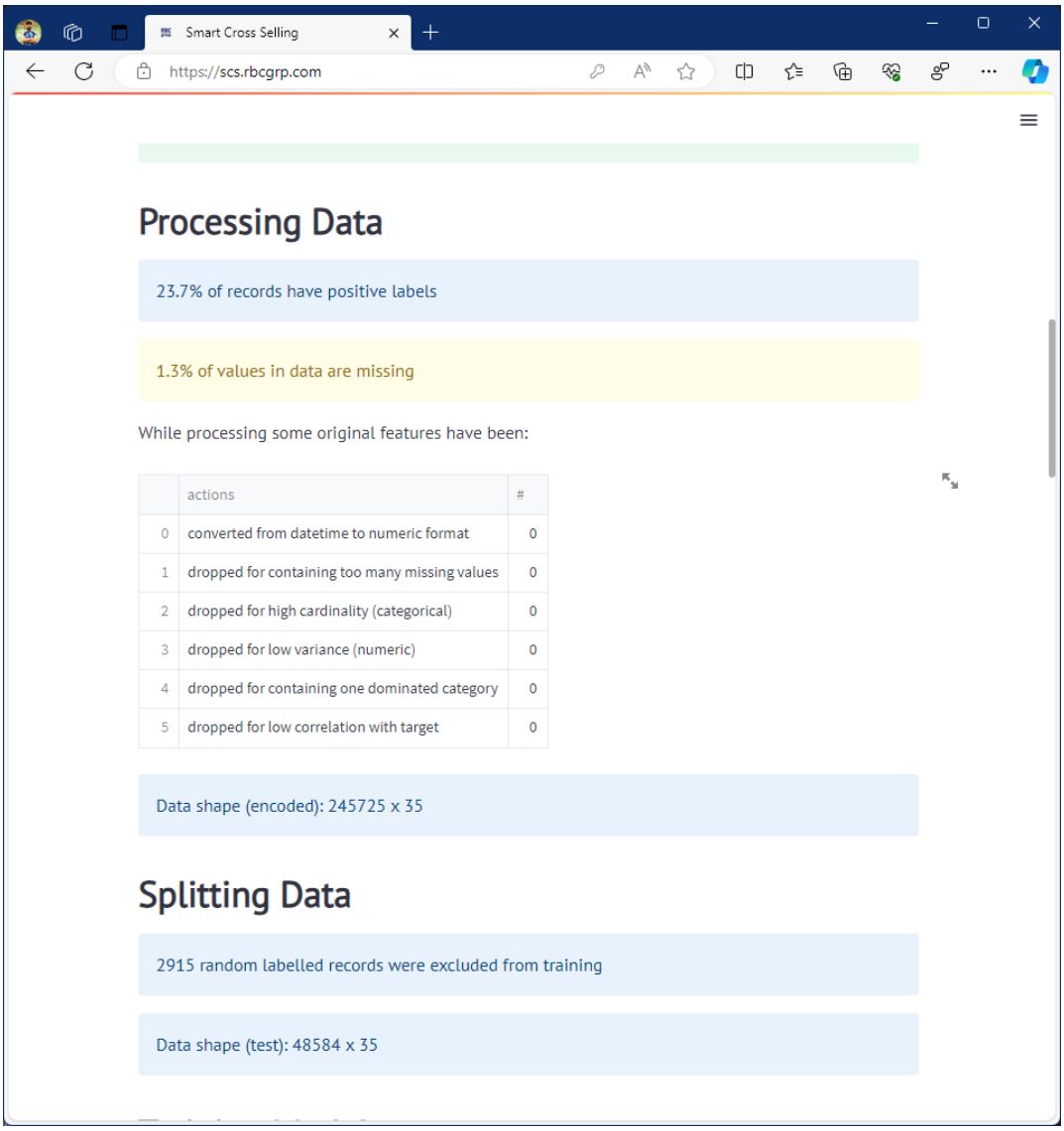

Рисунок 3.7 — Тестування системи

Четвертий крок: у додатку ми надаємо інструменти для оцінки точності прогнозів на завантаженому наборі даних. На скріншоті продемонстровано confusion matrix, яка відображає підрахунки прогнозованих і фактичних значень на тестовому наборі даних (рис. 3.8). Тут ми розраховуємо і пояснюємо показник recall: частку релевантних записів, відомих розмічених клієнтів, прихованих у тестовому наборі даних, успішно визначених нашою моделлю.

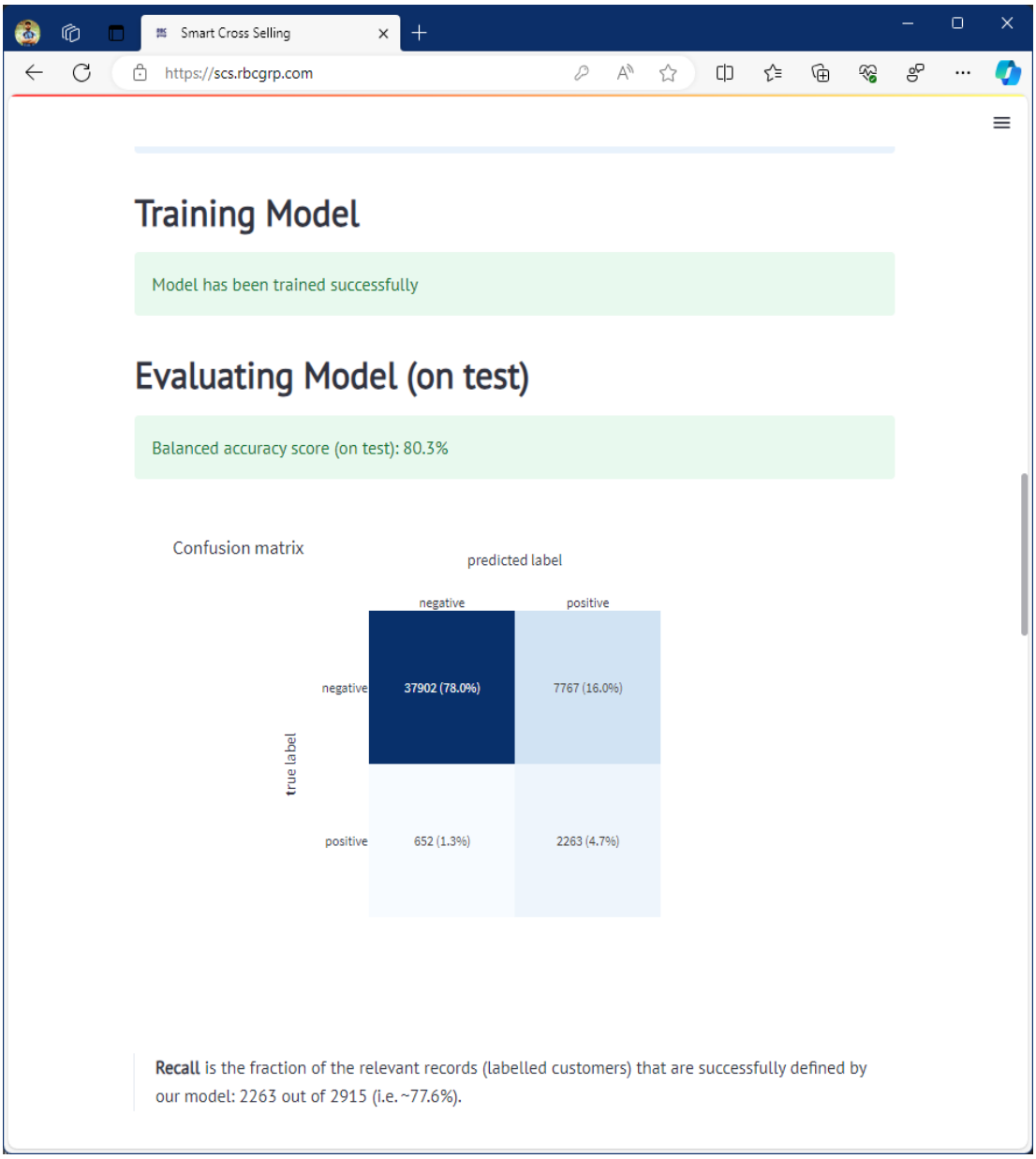

Рисунок 3.8 - Підрахунки прогнозованих і фактичних значень на тестовому наборі даних

На наступному графіку ми показуємо, що коли вибрані лише 14% немаркованих записів з вашого набору даних з найвищими показниками схожості згідно з нашою моделлю, ця вибірка містить в собі майже 68% всіх потенційних клієнтів / споживачів даного продукту в промо-кампанії перехресних продажів (рис. 3.9).

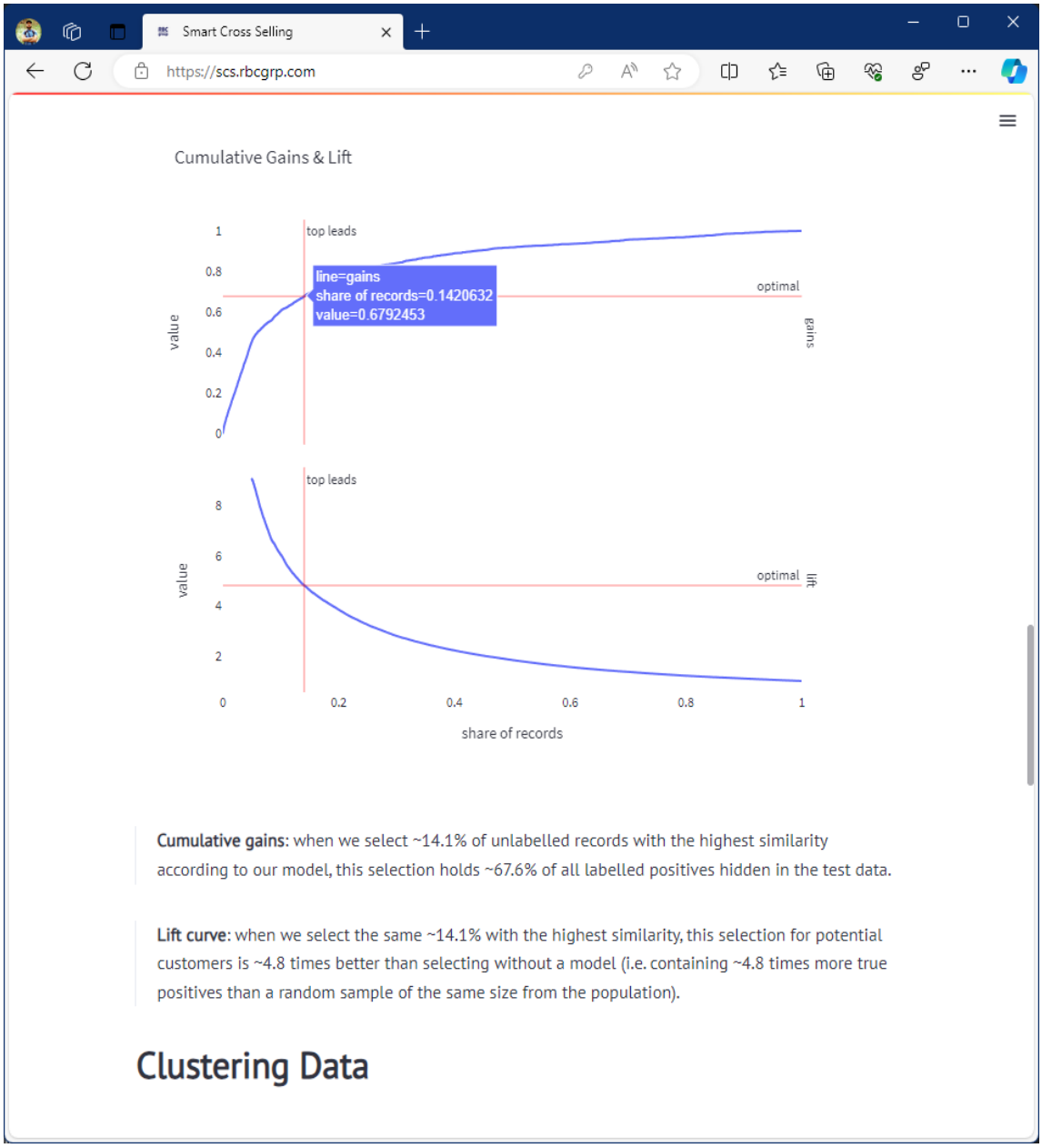

Рисунок 3.9 — Охоплення потенційних клієнтів немаркованим даними

На цьому етапі клієнт можете додати більше функцій до початкового набору даних і перезавантажити його, якщо результати ,будуть здаватися йому неприйнятними.

Останній крок: оцінка фінансової вигоди від використання нашого рішення. Тепер ви можете оцінити фінансові вигоди від використання нашого рішення. На прикладі цього демонстраційного набору даних ми рекомендуємо вам розпочати кампанію з вибірки у 20 тисяч потенційних клієнтів (із загальної клієнтської бази у майже 190 тисяч), яка, як очікується, міститиме понад 6 тисяч позитивних відповідей. (рис. 3,10)

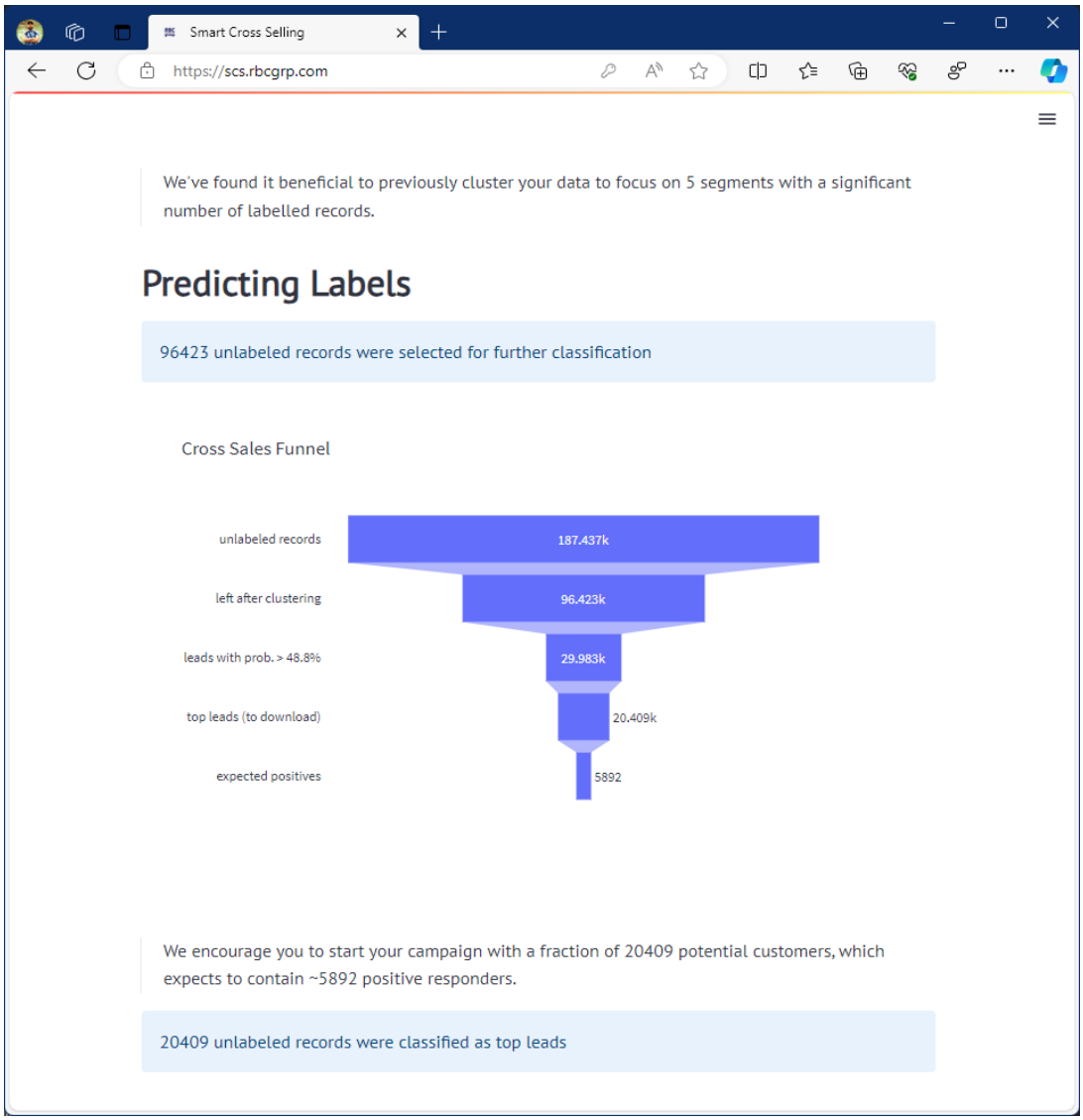

Рисунок 3.10— Оцінка фінансової вигоди рішення

Якщо все відповідає очікуванням, можна завантажити список лідів з оцінками схожості для використання у вашій маркетинговій кампанії (рис. 3.11).

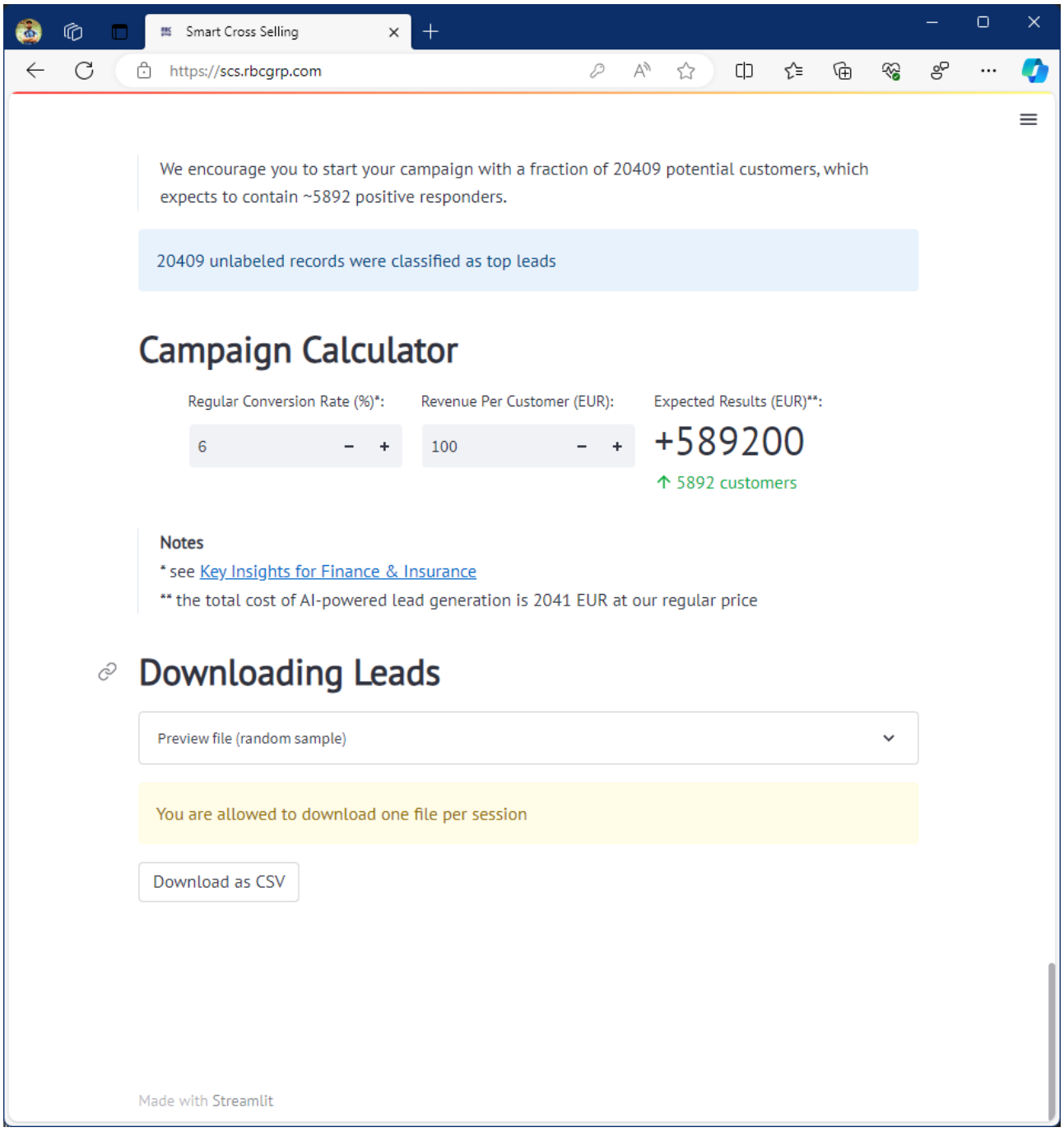

Рисунок 3.11 — Оцінка маркетингової компанії

3.3 Висновки до розділу 3

Центральною складовою аналітичних рішень компанії RBC Group є блок скриптів машинного навчання, який відповідає за обробку даних, створення ознак, та роботу з моделями машинного та глибокого навчання. Розробка відбувається на сервері під управлінням операційної системи Windows, з репозиторієм коду, який може бути розміщений на платформах GitLab або GitHub. Реліз проєкту включає Docker-image, що полегшує розгортання на сервері замовника. Для координації процесів використовується Apache Airflow. Методологія CI/CD використовується для швидкого впровадження оновлень, зберігаючи код у репозиторії та використовуючи теги для позначення релізів. Тестування функціональності проводиться в ізольованому контейнері на сервері з операційною системою Linux.

Рішення RBC Group, як правило, не потребують оркестрації чи горизонтального масштабування, оптимально використовуючи Apache Airflow для координації процесів всередині Docker-контейнера. Внутрішні бази даних використовуються для зберігання конфігураційних даних та проміжних результатів розрахунків. Замовник самостійно визначає формат та джерела даних для аналізу, а Proxy DB використовується для тимчасового зберігання вхідних даних та результатів розрахунків. Результати можуть бути вивантажені в основні бази даних або передані через API за узгодженням із замовником.

Рішення RBC Group, фактично, є back-end, який передає підготовлені дані замовникові. Використання фреймворків Python для розробки демо-проєктів, таких як Flask, Django, Dash, Web2Py та Streamlit, дозволяє швидко створювати прототипи аналітичних рішень. Онлайн-вітрини визначаються як необхідний елемент для кращого уявлення можливостей аналітичних продуктів. З цією метою, компанія використовує Python для аналітичних скриптів та розробки демо-проєктів.

Перехресні продажі в сучасних фінансових компаніях визнаються ключовою стратегією для ефективного зниження витрат на привертання клієнтів.

У сфері страхування тільки 20% компаній використовують цю стратегію ефективно, а більшість клієнтів обмежується одним страховим полісом.

Для подолання цих викликів, розроблено рішення, що спрямоване на забезпечення доступу менеджерів та агентів без технічного досвіду до складних моделей машинного та глибокого навчання. Комп'ютерна система прогнозування поведінки клієнтів для перехресних продажів має інтуїтивно зрозумілий інтерфейс, що включає лише декілька кроків: завантаження даних, вибір цільової аудиторії та оцінка точності прогнозування.

Основні функціональні блоки системи - блок AutoML для попередньої обробки даних та блок навчання ансамблю нейронних мереж - спрямовані на автоматизацію процесів відбору та тренування моделей. Блок AutoML відповідає за оптимізацію та трансформацію даних для подальшого використання нейронною мережею.

Взаємодія користувача з додатком передбачає кілька ключових кроків. Спочатку дані завантажуються з корпоративної CRM або аналітичного відділу у форматі електронної таблиці, що дозволяє включати різноманітні дані, включаючи текстові, числові та часові характеристики. Потім відбувається автоматична обробка даних та їх перетворення для навчання нейронної мережі.

Додаток надає інструменти для оцінки точності прогнозів, використовуючи confusion matrix та показник recall. Користувач може додавати нові функції до початкового набору даних та оцінювати їх вплив на результати. Останнім етапом є оцінка фінансової вигоди від використання системи, де рекомендації та результати демонстраційного набору даних допомагають приймати обґрунтовані рішення.

РОЗДІЛ 4 ДОСЛІДЖЕННЯ РЕЗУЛЬТАТІВ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ КЛІЄНТІВ ДЛЯ ПЕРЕХРЕСНИХ ПРОДАЖІВ

4.1 Метрики бінарної класифікації в пакетах scikit-learn і sci-kit-plot

Цей розділ присвячений опису особливостей метрик для бінарної класифікації, доступних в пакетів Python scikit-learn і sci-kit-plot. Ці пакети широко відомі своєю надійною функціональністю, простотою використання та розширеною підтримкою класифікаційного аналізу. Використовуючи ці пакети, дослідники та практики отримують доступ до набору інструментів оцінювання, що дозволяє їм точно оцінювати ефективність класифікатора та приймати обґрунтовані рішення.

Дисбаланс класів є поширеною проблемою в контексті бінарного прогнозного моделювання. Це відбувається, коли розподіл між двома класами сильно асиметричний. У цьому розділі ми спробуємо зробити більший акцент на виборі відповідного методу оцінки ефективності кінцевої моделі.

Неспроможність показника «точності» як метрики для незбалансованих даних добре відома. Розглянемо випадок набору даних зі співвідношенням 1 позитивний результат на 100 негативних. У цьому випадку модель, яка передбачає, що всі випадки будуть негативними, дає точність 99%. Проте ця модель є фіктивним класифікатором, який завжди прогнозує клас більшості.

Крива ROC і площа під кривою ROC (AUROC) стали найпоширенішою метрикою для оцінки моделі класифікації на незбалансованих даних з двох причин. По-перше, вони не чутливі до розподілу класів. AUROC дорівнює ймовірності того, що випадково вибраний позитивний випадок буде оцінений вище, ніж випадково вибраний негативний випадок. Метрику не цікавить, скільки позитивних і негативних випадків є в наборі даних. По-друге, AUROC не залежить від порогу. Нам не потрібно вирішувати, яким має бути поріг, що розділяє позитивні та негативні класи, щоб обчислити метрику. Це означає, що

коли негативний клас є більш поширеним, але кількість істинно негативних прогнозів є низькою, ROC може дати надто оптимістичну оцінку ефективності моделі.

Наприклад, розглянемо випадок набору даних, який містить 10 позитивних і 100 000 негативних результатів. У нас є 2 моделі:

Модель А: прогнозує 900 позитивних результатів, з яких 9 є справді позитивними

Модель В: прогнозує 90 позитивних результатів, з яких 9 є справжніми позитивними.

Очевидно, що модель В має кращі показники. Хоча обидві моделі передбачають однакову кількість правильних результатів, модель В видає менше хибних результатів. Іншими словами, модель В є більш «точною».

Однак розглянемо ROC-аналіз двох моделей, який вимірює частоту істинних спрацьовувань (TPR) у порівнянні з частотою хибних спрацювань (FPR):

Модель А: TPR = 9/10 = 0,9 і FPR = (900-9)/100 000 = 0,00891

Модель В: TPR = 9/10 = 0,9 і FPR = (90-9)/100 000 = 0,00081

Як і очікувалося, TPR є абсолютно однаковим в обох моделях. З іншого боку, оскільки кількість негативних результатів значно переважає кількість позитивних, різниця у FPR між обома моделями (0,00891-0,00081 = 0,0081) втрачається в тому сенсі, що її можна округлити майже до 0. Іншими словами, значна зміна кількості хибних спрацювань призвела до незначної зміни FPR, і, таким чином, ROC не в змозі відобразити кращу ефективність моделі.

На противагу цьому, крива Precision-Recall (PR) спеціально розроблена для виявлення рідкісних подій і є метрикою, яку слід використовувати, коли позитивний клас представляє більший інтерес, ніж негативний. На криву PR не впливає дисбаланс даних [6]. Повернімося до наведеного вище прикладу:

Модель А: згадування = TPR = 0,9 і точність = 9/900 = 0,01

Модель В: згадування = TPR = 0,9 і точність = 9/90 = 0,1

Для ефективної оцінки бінарних класифікаторів у роботі надано перевагу кривих precision-recall над кривими ROC, особливо в сценаріях із дуже незбалансованими наборами даних. Криві precision-recall забезпечують більш точне представлення ефективності прогнозування міноритарного класу та дозволяють приймати обґрунтовані рішення.

4.2 Порівняльний аналіз результатів роботи ML-алгоритмів в системі прогнозування поведінки клієнтів для перехресних продажів

Для порівняння і представлення результатів роботи алгоритмів використано стандартні методи/метрики для бінарної класифікації:

- Accuracy
- Precision
- Recall
- Confusion matrix (visual)
- Cumulative gain curve (visual)
- Precision-Recall curve (visual)

Лістинг 10 — Побудова confusion matrix

```
labels_raw = np.concatenate(
    [x.reshape(-1, 1) for x in [
        y_true.to_numpy(),
        sim_full_rate,
        sim_factor_rate,
        preds_many_nn,
        clf_rf_preds,
        clf_nn_preds.ravel()]
     ], axis=1)

labels = np.apply_along_axis(lambda x: x > THR, 0,
labels_raw)

mtx = pd.DataFrame(
    np.concatenate(
        [confusion_matrix(
            labels[:, 0],
            labels[:, i]) for i in range(1, 6)]),
```

```
    index=['true_neg', 'true_pos'] * 5,
    columns=['pred_neg', 'pred_pos'])

metrics = defaultdict(list)

for i in range(1, 6):
    for fun in (balanced_accuracy_score, recall_score,
precision_score):
        metrics[i - 1].append(fun(labels[:, 0], labels[:,
i]))

titles = ['all features similarity',
          f'{res_max}-subspace similarity',
          'ensemble NN classifier',
          'naive GB classifier',
          'naive NN classifier']

ODD = ':\naccuracy {:.1%}\nrecall {:.1%}\nprecision {:.1%}'

cm_titles = [('').join([title, ODD]).format(*metrics[i])
             for i, title in enumerate(titles)]

# https://stackoverflow.com/a/3900001/6025592
mtx['method'] = sum([[title] * 2 for title in cm_titles],
[])
def draw_heatmap(**kwargs):
    dfr = kwargs.pop('data').copy()
    dfr.drop(['method'], axis=1, inplace=True)

    dfr_cnts = [str(v) for v in dfr.values.flatten()]
    dfr_perc = [f'{v:.1%}' for v in dfr.values.flatten() /
                np.sum(dfr.values)]
    dfr_labs = [f'{v1}\n{v2}' for v1, v2 in zip(dfr_cnts,
dfr_perc)]
    dfr_labs = np.asarray(dfr_labs).reshape(-1, 2)

    sns.heatmap(dfr / dfr.sum().sum(),
                annot=dfr_labs,
                annot_kws={'fontsize': 14},
                fmt='',
                vmin=0,
                vmax=1,
                cbar=False,
                cmap='Blues',
                square=True,
                linewidth=0.1,
                **kwargs)
fg = sns.FacetGrid(mtx, col='method',
                   col_wrap=3,
```

```
                    height=4.5)

fg.map_dataframe(draw_heatmap)

fg.set_titles(col_template='{col_name}',  size=15)

fg.fig.tight_layout()
```

Порівняння роботи алгоритмів продемонстровано на рисунках 4.1 і 4.2.

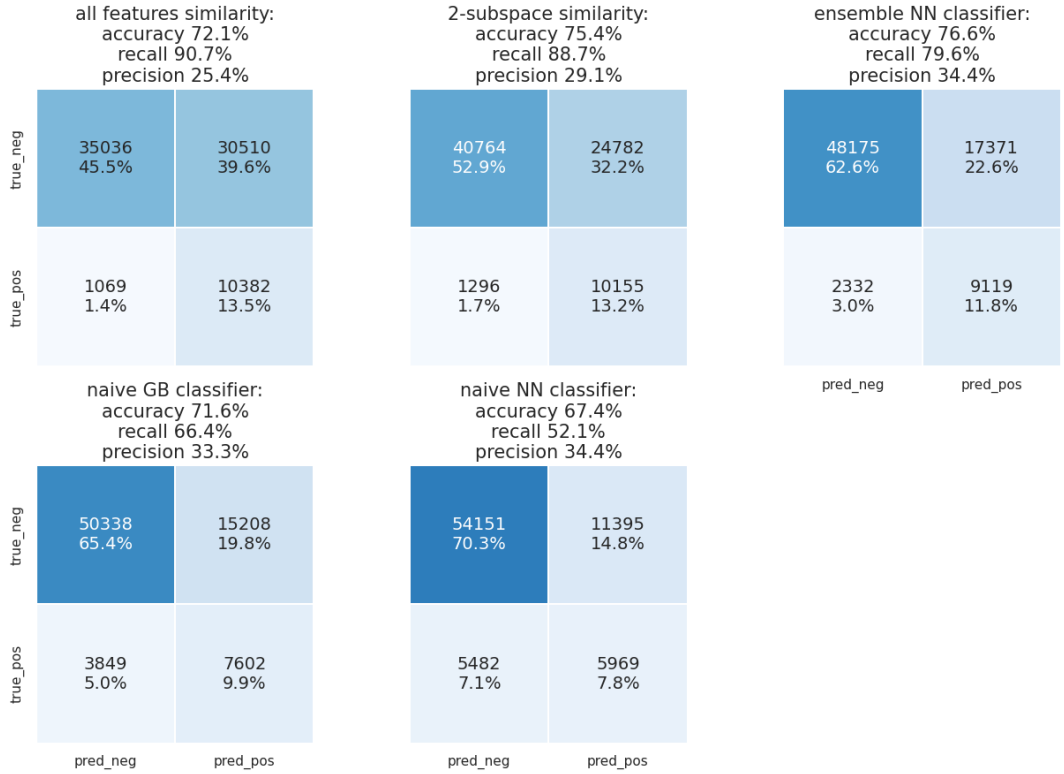

Рисунок 4.1 — Confusion matrices

Лістинг 11 — Побудова кумулятивної кривої посилення та Precision - Recall кривої

```
fig, ax = plt.subplots(ncols=2,
                       figsize=(16, 8),
                      #  sharex=True,
                      #  sharey=True
                       )

# https://github.com/reiinakano/scikit-plot/blob/2dd3e6a76df77edcbd724c4db25575f70abb57cb/scikitplot/helpers.py#L157
for i in range(1, 6):
    ax.flat[0].plot(
        *cumulative_gain_curve(labels_raw[:, 0], labels_raw[:, i], 1),
        label=titles[i-1])

# https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/
for i in range(1, 6):
    ax.flat[1].plot(
        *precision_recall_curve(labels_raw[:, 0], labels_raw[:, i])[:-1][::-1],
        label=titles[i-1])

ax.flat[0].plot((0, test_class_weights[1], 1),
                (0, 1, 1),
                color='black',
                linewidth=2,
                label='ideal classifier')

ax.flat[1].axhline(
    y=test_class_weights[1],
    linestyle='--',
    color='black',
    linewidth=1,
    label='no-skill classifier')

ax.flat[1].text(
    x=0.7,
    y=test_class_weights[1] * 0.9,
    s='no-skill classifier',
    verticalalignment='top')

ax.flat[0].set(
    xlabel='fraction of positively predicted examples',
    ylabel='fraction of examples correctly classified',
    title='Cummulative Gains')
```

```
ax.flat[1].set(
    xlabel='recall',
    ylabel='precision',
    title='Precision-Recall Curves')

ax.flat[0].legend()

plt.show()
```

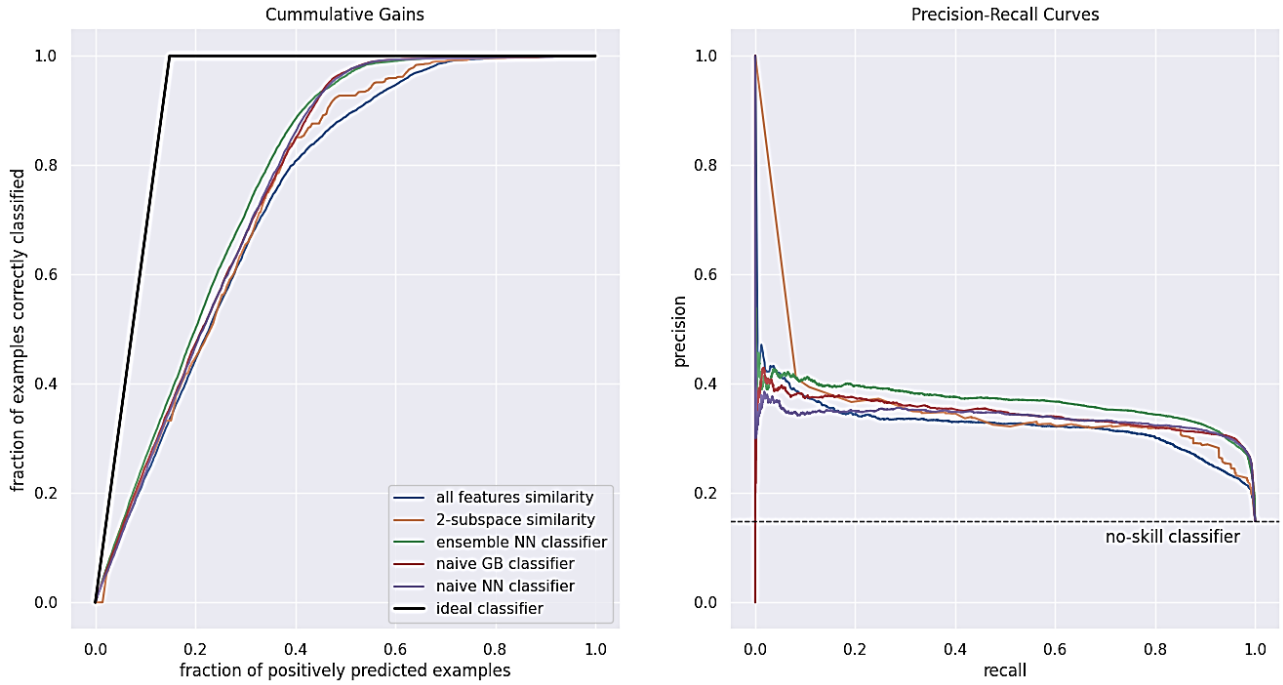

Рисунок 4.2 — Cumulative gain curve та Precision-Recall curve

Лістинг 12 — Перетин прогнозів різних алгоритмів

```
sets = [set(np.where(c > THR)[0]) for c in labels_raw.T]

plt.figure(figsize=(15, 8))

# https://github.com/gecko984/supervenn
# https://habr.com/ru/company/yandex/blog/501924/

supervenn(
    sets,
    ['y true'] + titles,
    rotate_col_annotations=True,
    col_annotations_area_height=2,
    widths_minmax_ratio=0.05,
```

```
    sets_ordering='minimize gaps',
    min_width_for_annotation=250,
    side_plots=True)
```

Перетин прогнозів різних алгоритмів представлено на рисунку 4.3.

Рисунок 4.3 — Перетин прогнозів різних алгоритмів

В цілому запропоновані автором підходи до вирішення поставленої бізнес-задачі (перехресні продажі) дають порівняно кращі результати класифікації з порівняно незначним збільшенням часу обчислень (за умови паралельного обчислення).

Шляхи для подальшого удосконалення роботи ML-алгоритмів в системі прогнозування поведінки клієнтів:

- переглянуто підхід до формування вибірки негативного класу (не брати за основу розмір позитивного класу, а брати - 5-10% датасету із невідомими спостереженнями;
- додано механізм розрахунку середньозваженого прогнозу (ваги - training accuracy моделей в ансамблі);

- подумати над тим, щоб навчати більше моделей, але для отримання прогнозів використовувати випадковим чином близько 50-75% датасету, і відповідно дробити датасет на потоки. Це має значно скоротити час для отримання результатів;
- технічні удосконалення - робити upsampling за допомогою готових (і більш інтелектуальних) методів SMOTE з пакету imblearn.

4.3 Висновки до розділу 4

У цьому розділі розглянуто ключові особливості метрик для бінарної класифікації, доступних в популярних пакетах Python, таких як scikit-learn і sci-kit-plot. Ці пакети відомі своєю надійною функціональністю та простотою використання, надаючи дослідникам та практикам інструменти для точної оцінки ефективності класифікаторів.

Особлива увага приділена проблемі дисбалансу класів у бінарних моделях, де розподіл між позитивним та негативним класами асиметричний. Представлено вибір методів оцінювання ефективності моделі в умовах незбалансованих даних.

Зосереджено увагу на використанні кривих ROC та площі під кривою ROC (AUROC) як популярної метрики для моделей на незбалансованих даних. Однак висвітлено обмеження цих метрик в умовах великого дисбалансу та великої кількості негативних випадків.

В кінці розділу підкреслено переваги кривих Precision-Recall (PR) для роботи з рідкісними подіями та даними з великим дисбалансом. Розглянуті приклади моделей та порівняння, які вказують на важливість використання PR-метрик у певних сценаріях бінарної класифікації.

Для аналіз результатів роботи ML-алгоритмів в системі прогнозування поведінки клієнтів для перехресних продажів використано метрики accuracy, precision, recall, а також візуалізовані confusion matrix, Cumulative Gain Curve та Precision-Recall Curve.

Результати свідчать про те, що запропоновані підходи на основі ансамблю нейронних мереж призвели до покращення результатів класифікації в контексті PU - навчання із незначним збільшенням часу розрахунків, що може бути подолано шляхом використання паралельних обчислень.

Для подальшого вдосконалення ML-алгоритмів для прогнозування поведінки клієнтів для перехресних продажів, рекомендується переглянути підходу до формування вибірки негативного класу та розглянути можливість навчання більшої кількості моделей. Також пропонується використовувати інтелектуальні методи балансування даних, зокрема методи SMOTE для узгодження кількості спостережень в класах. Ці підходи відкривають перспективи для поліпшення ефективності та оптимізації часу обчислень у системі прогнозування поведінки клієнтів для перехресних продажів.

ВИСНОВКИ

1. Метою дослідження є аналіз алгоритмів класифікації структурованих даних та розробка інформаційно-аналітичної системи для оптимізації стратегій перехресних продажів у фінансовому секторі.

2. В рамках дослідження розглянуто задачу класифікації як ефективного розподілу об'єктів за заданими класами. Зазначено важливість розділення набору даних на навчальну і тестову вибірки для побудови та оцінки ML / DL Досліджено контрольоване, неконтрольоване та напівконтрольоване машинне навчання, визначено роль класифікації у різних галузях, таких як маркетинг, медицина та розпізнавання образів. Розглянуті методи самонавчання та спільного навчання, які використовують обмежену кількість позначених даних і багато непозначених.

3. У галузі маркетингу розглянуто стратегії допродажів та перехресних продажів для підвищення прибутків бізнесу та ступеня задоволеності клієнтів. Перехресні продажі визначено як ключову стратегію розвитку клієнтської бази для фінансових компаній.

4. Детально розглянуто використання Kaggle як джерела відкритих даних для наукових досліджень та бізнес-задач в галузі науки про дані. Підкреслено, що Kaggle надає високоякісні дані для оцінки моделей машинного навчання. Для набору відкритих даних, завантаженого із Kaggle, проведено попередню обробку атрибутів та цільової змінної з метою його адаптації для умов бізнес-задачі та подальшого тестування методів визначення схожості об'єктів за сукупністю атрибутів.

5. Досліджено статистичні методи визначення схожості об'єктів і алгоритми Random Forest та Gradient Boosting для бінарної класифікації. Продемонстровано складності та недоліки застосування класичних алгоритмів ML / DL у вирішенні проблем класового дисбалансу в контексті PU-навчання.

6. Як альтернативу запропоновано метод, що використовує просту одношарову нейронну мережу та ансамбль із декількох одношарових нейронних

мереж. Цей підхід дозволяє обмежити кількість невідомих об'єктів, які мережа бачить як умовно-негативний клас (0), сприяючи зменшенню ймовірності помилкового маркування невідомих об'єктів як помилково негативних (1). Результати цього методу особливо ефективні при роботі з меншими батчами, пропорційними розміру тренувальної вибірки. Фінальний прогноз для об'єкта формується як середнє значення прогнозів всіх мереж.

7. Досліджено підхід компанії RBC Group до розробки аналітичних рішень з акцентом на використанні скриптів машинного навчання для обробки даних. Оглянуто Python-фреймворки для створення онлайн-вітрин та демо-застосунків, які дозволяють швидко розробляти прототипи аналітичних рішень. З використанням одного із таких фреймворків розроблено систему, яка автоматизує відбір та тренування моделей глибокого навчання для фінансових компаній. Система надає інтуїтивний інтерфейс для менеджерів та агентів, що дозволяє їм ефективно вибирати клієнтів для перехресних продажів.

8. Для дослідження ключових метрик запропонованого підходу до PU-бінарної класифікації використовуються пакети Python, такі як scikit-learn і scikit-plot. Звертається увага на проблему дисбалансу класів та вибір оптимальних метрик для оцінки якості класифікаторів. Зроблено висновок, що в сценаріях із рідкісними подіями важливо надавати перевагу кривим Precision-Recall при порівнянні ефективності моделей, незважаючи на популярність кривих ROC та площі під ними.

9. Сформульовано рекомендації для подальшого покращення ефективності системи прогнозування поведінки клієнтів, зокрема щодо формування вибірки негативного класу та створення умов для навчання більшої кількості моделей в ансамблі. Також рекомендується використовувати інтелектуальні методи балансування даних, такі як SMOTE.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alon A. Semi-Supervised Classification of Unlabeled Data (PU Learning)», 2022. URL: https://towardsdatascience.com/semi-supervised-classification-of-unlabeled-data-pu-learning-81f96e96f7cb (дата звернення: 23.06.2023)

2. Bekker J., Davis J. Learning from Positive and Unlabeled Data: a Survey, 2018. URL: https://doi.org/10.48550/arXiv.1811.04820 (дата звернення: 23.06.2023)

3. Blum A., Mitchell T. Combining Labeled and Unlabeled Data with Co-Training, 2010. URL: https://doi.org/10.1145/279943.279962 (дата звернення: 30.11.2022)

4. Brownlee J. Semi-Supervised Learning With Label Propagation, 2020. URL: https://machinelearningmastery.com/semi-supervised-learning-with-label-propagation (дата звернення: 23.06.2023)

5. Dabbas E. Interactive Dashboards and Data Apps with Plotly and Dash. Birmingham, UK : Packt, 2021. 408 pages

6. Davis J., Goadrich M. The Relationship Between Precision-Recall and ROC Curves, 2006. URL: https://doi.org/10.1145/1143844.1143874 (дата звернення: 09.10.2023)

7. Dobilas S. Self-Training Classifier: How to Make Any Algorithm Behave Like a Semi-Supervised One, 2021. URL: https://towardsdatascience.com/self-training-classifier-how-to-make-any-algorithm-behave-like-a-semi-supervised-one-2958e7b54ab7 (дата звернення: 23.06.2023)

8. Dorigatti S., Emilio H. Robust and Efficient Imbalanced Positive-Unlabeled Learning with Self-Supervision, 2022. URL: https://doi.org/10.48550/arXiv.2209.02459 (дата звернення: 23.06.2023)

9. Giannis K., Subhabrata M., Guoqing Z, Ahmed A. Self-Training with Weak Supervision, 2021. URL: https://aclanthology.org/2021.naacl-main.66.pdf (дата звернення 30.11.2022)

10. Hayes A., Howard E. What Is a Cross-Sell?, 2019. URL: https://www.investopedia.com/terms/c/cross-sell.asp#:~:text=Cross%2Dselling%20is%20the%20practice,to%20their%20existing%20client%20base (дата звернення: 30.11.2022)

11. Holomb V. An imPULSE to Action: A Practical Solution for Positive-Unlabeled Classification, 2023. URL: https://towardsdatascience.com/an-impulse-to-action-a-practical-solution-for-positive-unlabelled-classification-cd5895128e45 (дата звернення: 23.06.2023)

12. Holomb V. Enhancing Positive-Unlabeled Learning with custom semisupervised techniques. Молода наука-2023: збірник наукових праць студентів, аспірантів, докторантів і молодих вчених. Запоріжжя : ЗНУ, 2023. С. 82–83.

13. Holomb V. Practical Approach to Evaluating Positive-Unlabeled (PU) Classifiers in Business Analytics, 2023. URL: https://towardsdatascience.com/a-practical-approach-to-evaluating-positive-unlabeled-pu-classifiers-in-real-world-business-66e074bb192f (дата звернення: 23.06.2023)

14. Jain D., Shantanu S. Recovering True Classifier Performance in Positive-Unlabeled Learning, 2017. URL: https://doi.org/10.48550/arXiv.1702.00518 (дата звернення: 23.06.2023)

15. JointEntropy: Awesome ML Positive Unlabeled Learning, 2022. URL: https://github.com/JointEntropy/awesome-ml-pu-learning (дата звернення: 23.06.2023)

16. Ju-Hyoung L., Sang-Ki K., Yo-Sub H. SALNet: Semi-Supervised Few-Shot Text Classification with Attention-based Lexicon Construction, 2015. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17558/17365 (дата звернення: 30.11.2022)

17. Key Insights for Finance & Insurance, 2021. URL: https://unbounce.com/conversion-benchmark-report (дата звернення: 09.10.2023)

18. Khorasani M., Abdou M., Fernández J. Web Application Development with Streamlit: Develop and Deploy Secure and Scalable Web Applications to the Cloud Using a Pure Python Framework. Berkeley, CA : Apress, 2022. 352 pages

19. Kiyomaru H. A collection of notebooks with algorithms introduced in «Learning from Positive and Unlabeled Data: a Survey», 2020. URL: https://github.com/hkiyomaru/pu-learning (дата звернення: 23.06.2023)

20. Saunders J., Freitas A. Evaluating the Predictive Performance of Positive-Unlabelled Classifiers: a Brief Critical Review and Practical Recommendations for Improvement, 2022. URL: https://doi.org/10.48550/arXiv.2206.02423 (дата звернення: 23.06.2023)

21. Semi-Supervised Learning, Explained with Examples, 2020. URL: https://www.altexsoft.com/blog/semi-supervised-learning/ (дата звернення: 30.11.2022)

22. Xiaojin Z., Zoubin G. Learning from Labeled and Unlabeled Data with Label Propagation, 2018. URL: https://pages.cs.wisc.edu/~jerryzhu/pub/CMU-CALD-02-107.pdf (дата звернення: 30.11.2022)

23. Zhang Q., Lu J. Artificial intelligence in recommender systems, 2021. URL: https://doi.org/10.1007/s40747-020-00212-w (дата: 09.10.2023)

24. Головкіна Н. Персоналізація маркетингових комунікацій: новий стратегічний простір. Маркетинг в Україні. Київ, 2008. №2. С. 27-33

25. Голомб В. , Безверхий А. Покращення машинного навчання напів-контрольованими методами з використанням даних без позитивних міток. Ге-остратегічні трансформації та траєкторія національної безпеки в контексті від-будови і сталого розвитку України: матеріали міжнародної науково-практич-ної конференції (м. Запоріжжя, 25–26 травня 2023 р.). Запоріжжя : ЗНУ, 2023. С. 448-450.

26. Голомб В., Безверхий А. Комп'ютерна система раннього відбору клієнтів для перехресних продажів. Актуальні питання сталого науково-технічного та соціально-економічного розвитку регіонів України: матеріали III Всеукраїнської науково-практичної конференції за участю молодих науковців. Запоріжжя : ЗНУ, 2023. С. 152-154.

27. Животова А. Крос-маркетинг або ефективний інструмент співпраці. Розвиток наукової думки постіндустріального суспільства: сучасний дискурс: матеріали IV Міжнародної наукової конференції (м. Вінниця, 1 липня, 2022 р. / Міжнародний центр наукових досліджень). Вінниця : Європейська наукова платформа, 2022. С. 64-66

28. Інформаційні системи та технології в управлінні. Методичні вказівки, теоретичні відомості і завдання до лабораторних робіт для студентів та магістрів денної форми навчання спеціальності 7.803060101 Менеджмент організацій і адміністрування. Частина 3. Класифікація в бізнес-аналітиці. / Укл.: Біла Н.І. Запоріжжя : ЗНТУ, 2014. 50 с.

29. Кругова Т. Шляхи оптимізації здійснення перехресних продажів працівниками банку, 2019. URL: https://bit.ly/3RehvO6 (дата звернення: 30.11.2023)

30. Масик І., Дудинець Л Використання ІТ-інструментів у продажі банківських продуктів і послуг. Фінансово-кредитна система України в умовах інтеграційних і глобалізаційних процесів: матеріали Всеукраїнської науково-практичної конференції студентів та аспірантів (м. Черкаси, 11 квітня 2019 р. / ЧННІ ДВНЗ «Університет банківської справи»). Черкаси, 2019. С. 386-389

31. Найда Р., Олексин І. Крос-маркетингові технології в організації бізнес процесів роздрібної торгівлі. Проблеми та перспективи розвитку бізнесу в Україні: матеріали Міжнародної наук.-практ. конф. молодих вчених і студентів (м. Львів, 19 лютого 2021 р.). Львів : Львівський торговельно-економічний університет, 2021. С. 213 - 217

32. Тітова А., Іванов Д. Розробка моделі аналізу складних даних на основі класифікації machine learning. Вісник НТУ «ХПІ». Серія: Інформатика та моделювання. Харків : НТУ «ХПІ», 2018. № 42 (1318). С. 171 – 178.

Декларація
академічної доброчесності
здобувача вищої освіти ЗНУ

Я, Голомб Володимир Васильович, студент 2 курсу, форми здобуття освіти денної, Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ, спеціальність121 Інженерія програмного забезпечення, адреса електронної пошти se22m-03@stu.zsea.edu.ua,

підтверджую, що написана мною кваліфікаційна робота на тему «Використання алгоритмів машинного навчання для побудови системи прогнозування поведінки клієнтів» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлений/ознайомлена;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;
- згоден/згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою Інтернет-системи, а також на архівування роботи в базі даних цієї системи.

Дата 30.11.2023 Підпис_____ В.В. Голомб
(студент)

Дата 30.11.2023 Підпис_____ А.І. Безверхий
(науковий керівник)