Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

Бакалаврська дипломна робота на тему:

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ВІДТОКУ ПОКУПЦІВ В ІНТЕРНЕТ-МАГАЗИНІ»

Виконав: студент 4 курсу, групи 2ІСТ-20б спеціальності 126 – «Інформаційні системи та технології»
_____ Наталя ІЖАКОВСЬКА
Керівник: к.т.н., доц. каф. САІТ
_____ Олексій КОЗАЧКО
«31» 05 2024 р.
Рецензент: к.т.н., доц. каф. КН
_____ Володимир ОЗЕРАНСЬКИЙ
«17» 06 2024 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
«03» 06 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____д.т.н., проф. Віталій МОКІН
"05" 03 2024 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Іжаковській Наталі Сергіївні

1. Тема роботи: «Інформаційна технологія передбачення відтоку покупців в інтернет-магазині»
керівник роботи: Козачко О. М., к.т.н., доц. каф. САІТ
затверджені наказом ВНТУ від «11» 03 2024 року №80
2. Термін подання студентом роботи 31.05
3. Вихідні дані до роботи:
Дані про відтік клієнтів з інтернет магазину.
4. Зміст текстової частини:
1) Аналіз предметної області;
2) Підготовка даних та розвідувальний аналіз;
3) Розроблення інформаційної технології для аналізу даних.
5. Перелік ілюстративного матеріалу:
1) Кореляційна матриця показників;
1) Блок-схема інформаційної технології;
2) Матриця кореляції показників
3) Матриці плутанини найкращих моделей.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11.03

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	11.03	31.03	вик
2	Підготовка даних та розвідувальний аналіз	01.04	15.04	вик
3	Проведення розвідувального аналізу	16.04	30.04	вик
4	Розроблення інформаційної технології для аналізу даних	01.05	15.05	вик
5	Оформлення матеріалів до захисту БДР	16.05	31.05	вик

Студентка _____ Наталя ІЖАКОВСЬКА

Керівник роботи _____ Олексій КОЗАЧКО

АНОТАЦІЯ

Бакалаврська дипломна робота складається з __ сторінок формату А4, на яких є __ рисунків, , список використаних джерел містить __ найменувань.

Метою роботи розроблення інфомраційної технології для підвищення точності передбачання відтоку клієнтів з інтернет магазину за рахунок використання аналізу та класифікації даних.

Виконано аналіз предметної галузі аналізу, проведено дослідження предметної області та обгрунтовано актуальність дослідження. Проведено розвідувальний аналіз, відібрано оптимальний набір ознак для дослідження. Проведено аналіз методів машинного навчання для класифікації. Розроблено інформаційну технологію та протестовано моделі класифікації, проаналізовано результат їх виконання.

Ключові слова: інформаційна технологія, аналіз даних, розвідувальний аналіз, побудова моделей, задача класифікації щодо факторів впливу.

ABSTRACT

The bachelor thesis consists of __ pages of A4 format, on which there are __ figures, the list of used sources contains __ titles.

The purpose of the work is to develop information technology to increase the accuracy of predicting the outflow of customers from the online store through the use of data analysis and classification.

The analysis of the subject area of the analysis was performed, the research of the subject area was conducted and the relevance of the research was substantiated. An exploratory analysis was carried out, the optimal set of features was selected for the study. The analysis of machine learning methods for classification was carried out. Information technology was developed and classification models were tested, the results of their implementation were analyzed.

Key words: information technology, data analysis, exploratory analysis, model building, classification problem regarding influencing factors.

ЗМІСТ

ВСТУП.....7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 8
1.1 Огляд предметної області 8
1.2 Огляд і опис методів класифікації 11
1.3 Вибір оптимальних інформаційних технологій 21
1.4 Висновки 24
2 ПІДГОТОВКА ДАНИХ ТА РОЗВІДУВАЛЬНИЙ АНАЛІЗ 25
2.1 Огляд та підготовка датасету 25
2.2 Розвідувальний аналіз 29
2.3 Висновки 43
3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ АНАЛІЗУ ДАНИХ 45
3.1 Побудова моделей класифікації 46
3.2 Розроблення алгоритму інформаційної технології 45
3.3 Висновки 56
ВИСНОВКИ 57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 58
Додаток А (обов’язковий) Технічне завдання 61
Додаток Б (обов’язковий) Протокол перевірки бакалаврської роботи на наявність текстових запозичень 63
Додаток В (довідниковий) Фрагмент лістингу програми.....64
Додаток Г (обов’язковий) Ілюстративна частина 76

ВСТУП

Актуальність теми. Використання інформаційних технологій для передбачення відтоку клієнтів з інтернет-магазину вкрай важлива в сучасному електронному бізнесі. З кожним днем обсяги даних, що збираються та обробляються магазинами, зростають. Ці дані містять важливу інформацію про клієнтські вподобання, покупкові звички, поведінку та інші параметри, які можуть бути використані для аналізу та прогнозування відтоку клієнтів.

Прогнозування відтоку клієнтів дозволяє магазинам уникнути втрати потенційного бізнесу шляхом ідентифікації клієнтів, які мають намір залишити магазин, та вжиття заходів для збереження їх. Інформаційні технології дозволяють обробляти великі обсяги даних швидко та ефективно, використовуючи алгоритми машинного навчання та аналітичні методи.

Мета і задачі дослідження. Метою дослідження є підвищення точності передбачення відтоку клієнтів з інтернет магазину за допомогою методів машинного навчання шляхом розроблення інформаційної технології. Для досягнення цієї мети передбачається виконання таких задач:

- провести аналіз проблеми відтоку клієнтів з інтернет магазину;
- здійснити аналіз інформаційних технологій;
- провести розвідувальний аналіз даних;
- розробити інформаційну технологію передбачення відтоку клієнтів.

Отримані результати та висновки дослідження можуть бути використані для покращення процесу передбачення відтоку клієнтів.

Об’єктом дослідження є процес передбачення відтоку клієнтів за допомогою методів машинного навчання.

Предметом дослідження є методи машинного навчання та інформаційна технологія, які використовуються для передбачення відтоку клієнтів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд предметної області

Інтернет-магазин – це веб-сайт або платформа, що дозволяє покупцям здійснювати покупки через інтернет. Його розвиток є частиною ширшого тренду на електронну комерцію, яка включає в себе всі форми онлайн-торгівлі. Інтернет-магазини забезпечують зручність для покупців, оскільки вони можуть робити покупки будь-якого часу і з будь-якого місця, де є доступ до інтернету. Це дозволяє їм переглядати широкий асортимент товарів, порівнювати ціни та читати відгуки інших покупців[1].

З точки зору власника бізнесу, інтернет-магазин має численні переваги. Перш за все, він дає можливість охопити глобальну аудиторію без необхідності відкривати фізичні магазини в кожному місті або країні. Це суттєво знижує витрати на оренду приміщень, заробітну плату працівникам та інші операційні витрати. Крім того, інтернет-магазин дозволяє збирати і аналізувати великі обсяги даних про покупців та їхні вподобання, що сприяє більш ефективному маркетингу і збільшенню продажів.

Технічна реалізація інтернет-магазину включає створення веб-сайту або додатку, інтеграцію з платіжними системами, забезпечення безпеки даних і захисту від кібератак. Важливим аспектом є оптимізація сайту для пошукових систем (SEO), що допомагає залучити більше відвідувачів. Крім того, велике значення має розробка інтуїтивно зрозумілого і зручного інтерфейсу, щоб покупці могли легко знаходити і замовляти потрібні товари.

Однією з головних складових успіху інтернет-магазину є логістика, яка включає організацію доставки товарів до покупців. Це може бути здійснено через власну кур'єрську службу або партнерство з логістичними компаніями. Важливим є забезпечення швидкої і надійної доставки, а також можливості відстеження замовлення в режимі реального часу.

Інтернет-магазини також стикаються з викликами, такими як високий рівень конкуренції, необхідність постійного оновлення асортименту і адаптації до змін у споживчих вподобаннях. Окрім того, важливо підтримувати високий рівень обслуговування клієнтів, вирішувати їхні проблеми і відповідати на запитання в найкоротші терміни[1].

Загалом, інтернет-магазин є важливою складовою сучасної економіки і невід'ємною частиною життя багатьох споживачів. Він відкриває нові можливості для бізнесу та спрощує процес покупки для користувачів, що робить його незамінним інструментом у сфері торгівлі.

Приклад інтернет-магазину показано на рисунку 1.1.

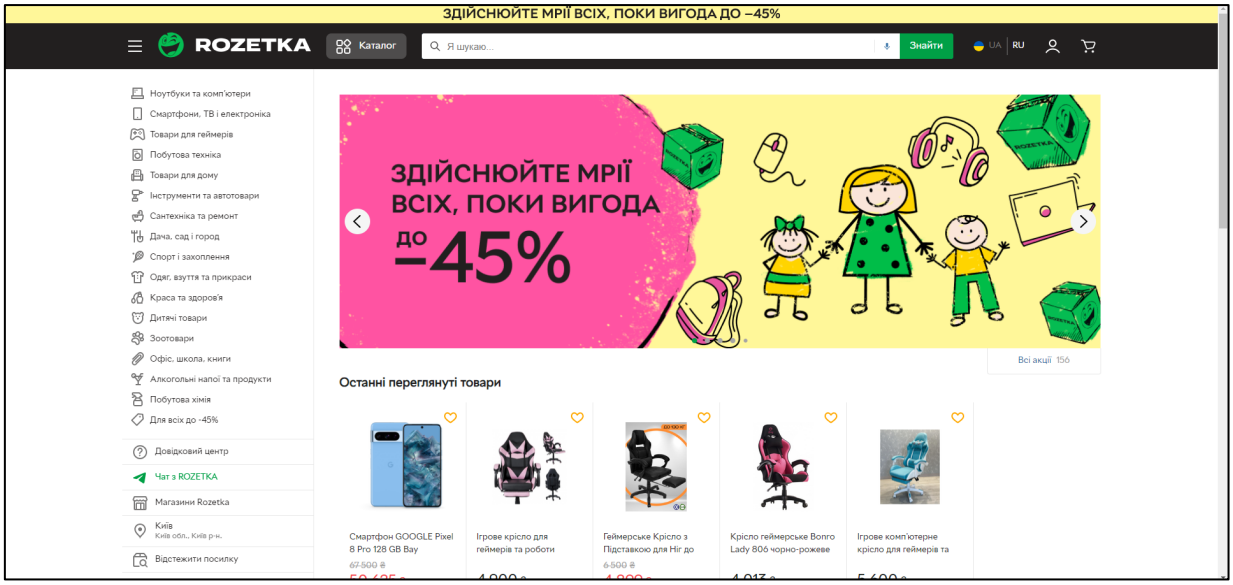


Рисунок 1.1 – Інтернет-магазин Rozetka.

Інтернет-магазини сприяють розвитку цифрової економіки, підтримуючи малий та середній бізнес, який раніше міг бути обмежений локальними ринками. Завдяки інтернет-магазинам, навіть невеликі підприємства можуть конкурувати з великими гравцями на рівних умовах, пропонуючи унікальні товари або спеціалізовані послуги. Це особливо актуально для нішевих ринків, де специфічні продукти знаходять своїх покупців по всьому світу.

Ще одним важливим аспектом є впровадження нових технологій в роботу інтернет-магазинів. Використання штучного інтелекту та машинного навчання дозволяє аналізувати поведінку користувачів, передбачати їхні потреби і пропонувати персоналізовані рекомендації. Наприклад, системи рекомендацій можуть запропонувати схожі або додаткові товари на основі попередніх покупок, що збільшує середній чек і задоволення клієнтів. Крім того, чат-боти забезпечують цілодобову підтримку, відповідаючи на поширені запитання та допомагаючи у вирішенні проблем[2].

Важливим напрямом є також використання мобільних платформ. Більшість інтернет-магазинів сьогодні має мобільні версії своїх сайтів або додатки, що дозволяють здійснювати покупки зі смартфонів і планшетів. Це відповідає тенденції зростання мобільного інтернет-трафіку і дозволяє залучити більше користувачів, які віддають перевагу мобільним пристроям.

Окремо варто згадати соціальні мережі, які стали потужним інструментом для просування інтернет-магазинів. Через платформи на зразок Facebook, Instagram або TikTok, підприємства можуть взаємодіяти зі своєю аудиторією, організовувати рекламні кампанії і залучати нових клієнтів. Соціальні мережі дозволяють не лише збільшити видимість бренду, але й створити лояльну спільноту навколо нього.

На рисунку 1.2 показано приклад рекламної кампанії інтернет-магазину з допомогою соціальних мереж.

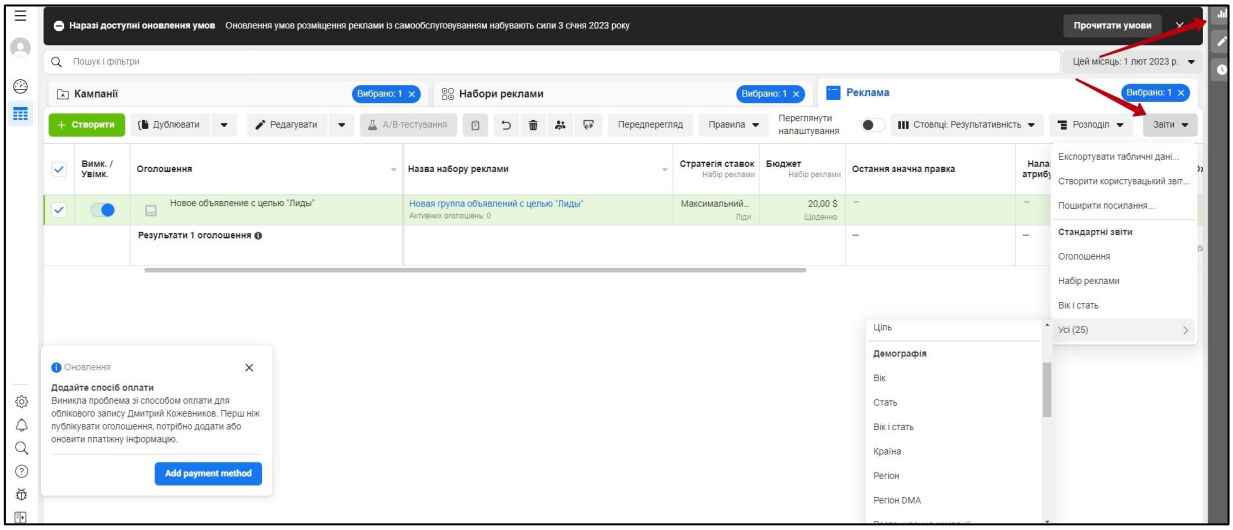


Рисунок 1.2 – Приклад налаштування рекламної кампанії для інтернет-магазину у Facebook.

Інтернет-магазини мають значний вплив на екологію. З одного боку, вони сприяють зменшенню викидів парникових газів, оскільки знижують необхідність у фізичних магазинах і відповідних перевезеннях. З іншого боку, збільшення обсягу пакування і доставки може мати негативний вплив на навколишнє середовище. Вирішенням цієї проблеми є впровадження екологічно чистих матеріалів для пакування та оптимізація логістичних процесів[3].

В цілому, інтернет-магазини продовжують розвиватися, впроваджуючи нові технології і адаптуючись до змін у поведінці споживачів. Вони стають невід'ємною частиною сучасного життя, надаючи зручність, швидкість та різноманітність вибору. У майбутньому можна очікувати ще більших змін та інновацій у цій сфері, що зробить процес покупок ще більш інтерактивним і персоналізованим.

1.2 Огляд і опис методів класифікації

Машинне навчання є галуззю штучного інтелекту, що зосереджується на розробці алгоритмів та моделей, які дозволяють комп'ютерам навчатися і

робити прогнози або приймати рішення на основі даних. Основна ідея машинного навчання полягає в тому, щоб комп'ютерна система могла автоматично вдосконалюватися, використовуючи наявні дані, без необхідності бути запрограмованою на виконання конкретних завдань[4].

Існує декілька основних типів машинного навчання, серед яких найпоширенішими є контрольоване, неконтрольоване і підкріплювальне навчання. Контрольоване навчання включає використання міткованих даних, де модель навчається на основі вхідних даних та відповідних виходів. Прикладом є задачі класифікації, коли модель навчається розпізнавати категорії, або регресії, де модель прогнозує числові значення. Неконтрольоване навчання, навпаки, працює з немітковаными даними, шукаючи приховані структури або закономірності в даних. До таких задач належать кластеризація і зниження розмірності. Підкріплювальне навчання базується на системі винагород і покарань, де агент вчиться шляхом взаємодії з навколишнім середовищем, оптимізуючи свої дії для максимізації сумарної винагороди.

Машинне навчання знайшло широке застосування у різних галузях. У медицині алгоритми машинного навчання використовуються для діагностики захворювань, прогнозування результатів лікування та персоналізованої медицини. У фінансовому секторі ці технології допомагають у виявленні шахрайства, прогнозуванні ринкових трендів і управлінні ризиками. У сфері роздрібної торгівлі та електронної комерції машинне навчання застосовується для персоналізації рекомендацій, оптимізації ціноутворення та управління запасами.

Для виконання поставленої задачі було обрано такі моделі: Random forests, K-Nearest Neighbors, Support Vector Machines, Decision Tree.

Random forests, або випадкові ліси, є одним з найпотужніших та найпоширеніших алгоритмів машинного навчання, який використовується для задач класифікації та регресії. Цей метод базується на концепції

ансамблевого навчання, де для побудови кінцевої моделі використовується множина простих моделей, зокрема дерев рішень. Основна ідея випадкових лісів полягає в тому, щоб поєднувати кілька дерев рішень, кожне з яких навчається на різних підмножинах даних і різних наборах ознак, що дозволяє зменшити ймовірність перенавчання і покращити загальну продуктивність моделі[5].

Алгоритм створення випадкових лісів починається з генерації множини підвибірок з початкового набору даних за допомогою методу бутстрепінгу, тобто випадкового вибору з поверненням. Для кожної підвибірки будується дерево рішень, причому на кожному вузлі дерева відбувається випадковий вибір підмножини ознак, які використовуються для визначення найкращого розбиття. Це забезпечує різноманітність дерев у лісі і допомагає зменшити кореляцію між ними. Після того, як усі дерева побудовані, кінцева модель об'єднує їх результати шляхом голосування або усереднення (рис. 1.3).

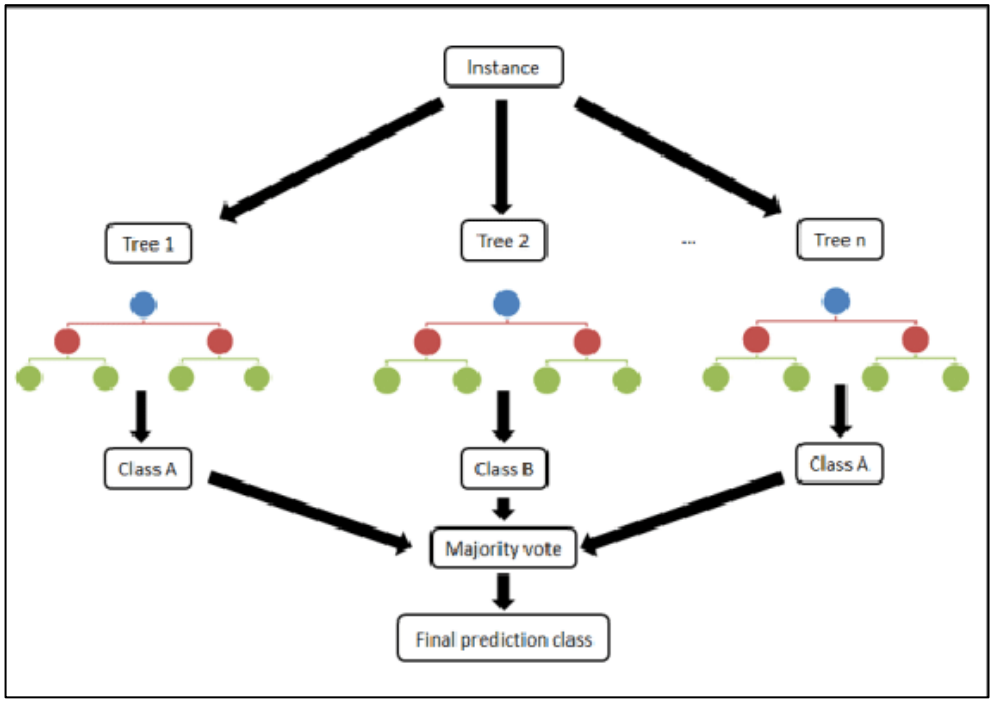


Рисунок 1.3 – Приклад роботи алгоритму Random Forest.

Однією з основних переваг випадкових лісів є їхня здатність працювати добре навіть з великим числом ознак і високою розмірністю даних. Вони стійкі до перенавчання, оскільки кожне дерево навчається на різних підмножинах даних і ознак, що робить модель більш узагальненою. Випадкові ліси також добре справляються з пропущеними даними і мають вбудовані методи для оцінки важливості ознак, що може бути корисним для відбору найбільш значущих характеристик у задачах з великою кількістю ознак[6].

Іншим важливим аспектом випадкових лісів є їхня здатність до інтерпретації. Хоча вони складаються з багатьох дерев рішень, кожне окреме дерево є простим і зрозумілим. Крім того, метод важливості ознак, що вбудований у випадкові ліси, дозволяє оцінити, які ознаки найбільше впливають на результат моделі, що може допомогти у розумінні процесу прийняття рішень.

Випадкові ліси знайшли широке застосування у різних галузях. У медицині вони використовуються для діагностики захворювань і прогнозування результатів лікування. У фінансах випадкові ліси застосовуються для виявлення шахрайства та прогнозування кредитних ризиків. У маркетингу ці алгоритми допомагають у сегментації клієнтів і прогнозуванні поведінки споживачів. У сфері екології вони використовуються для моделювання екологічних процесів та прогнозування змін у навколишньому середовищі[7].

Незважаючи на численні переваги, випадкові ліси мають і деякі обмеження. Наприклад, вони можуть бути менш ефективними для задач, де відношення між ознаками є дуже складними або нелінійними. Крім того, побудова великої кількості дерев може вимагати значних обчислювальних ресурсів, що може бути проблематичним для дуже великих наборів даних.

Алгоритм K-найближчих сусідів (K-nearest neighbors, KNN) є одним з найпростіших і найзрозуміліших методів машинного навчання, який використовується для класифікації та регресії. Основна ідея KNN полягає в тому, що об'єкт класифікується або отримує значення на основі класів або значень його найближчих сусідів у багатовимірному просторі ознак. Це означає, що для прийняття рішення про клас або значення об'єкта враховуються дані про його найближче оточення[8].

Процес роботи алгоритму KNN починається з вибору значення параметра K, яке визначає кількість сусідів, що будуть використовуватися для класифікації або регресії. Потім для кожного об'єкта, який потрібно класифікувати або оцінити, обчислюється відстань до всіх об'єктів у тренувальному наборі даних. Найчастіше використовується евклідова відстань, але можуть застосовуватися й інші метрики, такі як мангеттенська або косинусна відстань. Після обчислення відстаней обираються K найближчих сусідів, і на основі їхніх класів або значень приймається рішення.

У випадку класифікації алгоритм KNN визначає клас об'єкта шляхом голосування серед його K найближчих сусідів: об'єкт отримує клас, який має найбільшу кількість представників серед цих сусідів. У випадку регресії значення об'єкта обчислюється як середнє значення його K найближчих сусідів. Таким чином, алгоритм KNN є простим у реалізації та інтуїтивно зрозумілим.

KNN має кілька важливих переваг. По-перше, це непараметричний метод, що означає, що він не робить жодних припущень щодо розподілу даних. Це робить його гнучким і здатним працювати з широким спектром задач. По-друге, алгоритм легко адаптується до нових даних, оскільки не вимагає тренування моделі: усі обчислення здійснюються під час запиту, що дозволяє додавати нові дані без необхідності повторного навчання.

Проте KNN має і свої недоліки. Один з основних викликів – це обчислювальна складність, оскільки для кожного запиту необхідно обчислити відстані до всіх об'єктів у тренувальному наборі, що може бути проблематичним для великих обсягів даних. Іншою проблемою є вибір оптимального значення K: занадто мале значення K може призвести до перенавчання, тоді як занадто велике – до недонавчання. Крім того, алгоритм KNN є чутливим до масштабування ознак, тому попередня нормалізація або стандартизація даних часто є необхідною[9].

Приклад роботи алгоритму можна побачити на рисунку 1.4.

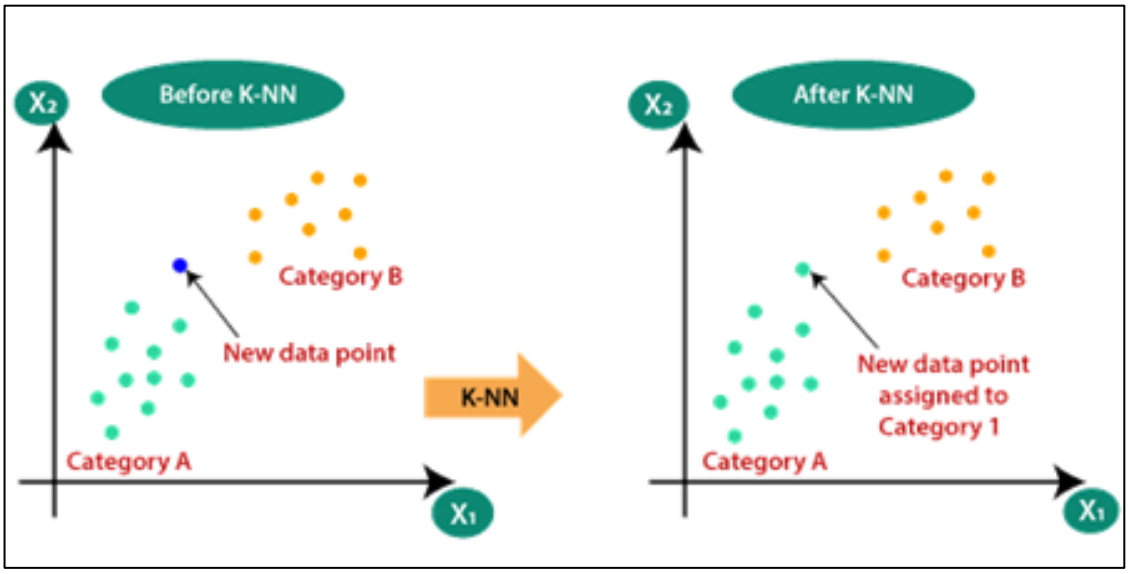


Рисунок 1.3 – Приклад роботи K-Nearest Neighbors

KNN знаходить застосування в різних галузях. У системах рекомендацій він використовується для знаходження схожих користувачів або продуктів. У медицині KNN застосовується для діагностики захворювань на основі симптомів пацієнтів, знаходячи схожі випадки в базі даних. У сфері розпізнавання образів цей алгоритм використовується для класифікації зображень на основі їх схожості з відомими зразками. У фінансовій сфері KNN може допомагати в прогнозуванні ринкових тенденцій на основі історичних даних.

Support Vector Machines (SVM), або машини опорних векторів, є потужним і широко використовуваним алгоритмом машинного навчання, який застосовується для задач класифікації та регресії. Основна ідея SVM полягає в тому, щоб знайти гіперплощину, яка найкраще розділяє дані на різні класи в багатовимірному просторі ознак. Гіперплощина, яка максимізує відстань до найближчих точок кожного класу, називається оптимальною гіперплощиною, а ці найближчі точки — опорними векторами.

Початково SVM був розроблений для лінійно роздільних задач. У таких випадках алгоритм шукає лінію (у двовимірному просторі) або гіперплощину (у багатовимірному просторі), яка чітко розділяє дані на дві категорії. Основна мета полягає в тому, щоб максимізувати "зазор" або "маржу" між двома класами, тобто відстань між гіперплощиною і найближчими точками кожного класу. Це забезпечує більшу узагальнювальну здатність моделі[10].

Однак у реальних задачах дані часто не є лінійно роздільними. Для вирішення таких проблем SVM використовує трюк, відомий як ядровий метод. Ядрові функції дозволяють перетворити вхідні дані у вищий вимір, де вони стають лінійно роздільними. Найбільш поширеними ядровими функціями є поліноміальне ядро, гауссове (радіально-базисне) ядро (RBF) та сигмоїдне ядро. Використання ядерних функцій робить SVM дуже гнучким і дозволяє моделювати складні нелінійні взаємозв'язки між ознаками.

Однією з головних переваг SVM є його здатність ефективно працювати у високовимірних просторах і навіть коли кількість ознак значно перевищує кількість зразків. Крім того, SVM є стійким до перенавчання, особливо в умовах високої розмірності, завдяки максимізації маржі між класами. Це робить його придатним для широкого спектру застосувань, таких як розпізнавання образів, біоінформатика, текстова класифікація і багато інших.

Проте SVM має й свої обмеження. Алгоритм є обчислювально інтенсивним, особливо для великих наборів даних, що може вимагати значних ресурсів для навчання моделі. Крім того, вибір правильного ядра та його параметрів є критичним для продуктивності моделі, що може бути складним і вимагати експериментів та крос-валідації. Також SVM погано справляється з великим об'ємом шуму в даних, оскільки шумові точки можуть сильно впливати на положення гіперплощини.

Важливою частиною роботи з SVM є регуляризація, яка дозволяє контролювати компроміс між максимізацією маржі і кількістю помилок на тренувальних даних. Це досягається шляхом введення параметра регуляризації, який можна налаштовувати для досягнення оптимальної продуктивності моделі на нових даних[11].

Приклад роботи алгоритму показано на рисунку 1.5.

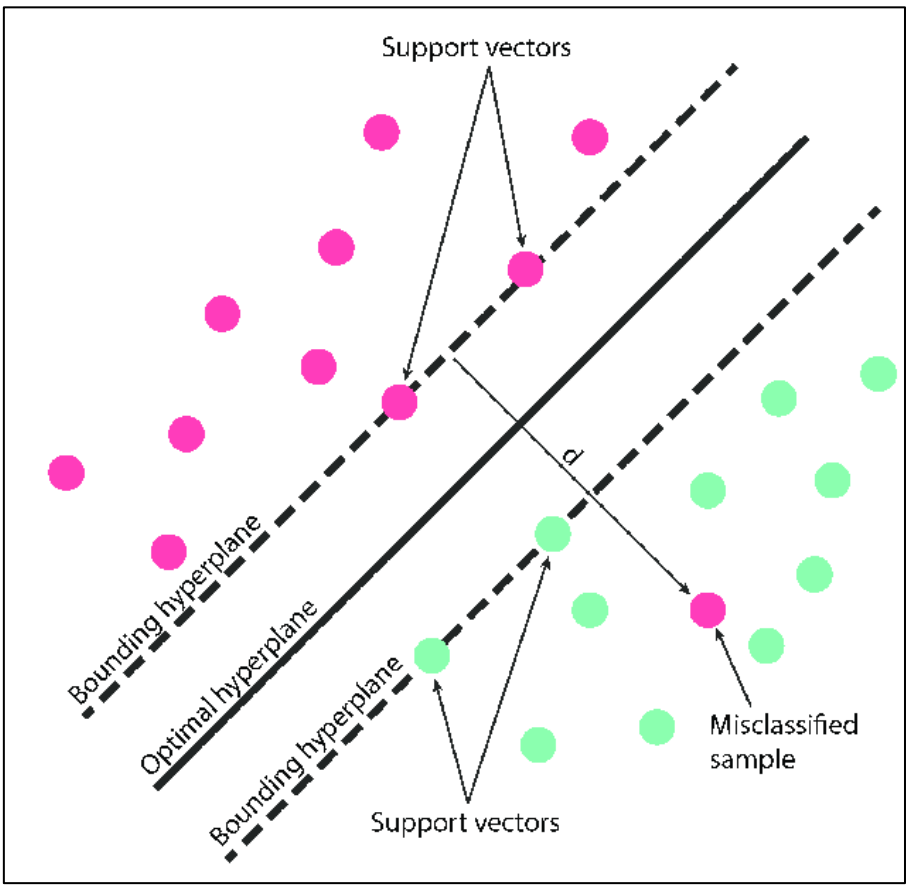


Рисунок 1.5 – Приклад роботи алгоритму Support Vector Machines.

Support Vector Machines застосовуються у багатьох галузях. У медицині SVM використовуються для діагностики захворювань на основі аналізу медичних зображень і біомаркерів. У фінансовій сфері вони допомагають у прогнозуванні ринкових тенденцій та виявленні аномалій. У сфері розпізнавання образів SVM використовуються для класифікації зображень та розпізнавання рукописного тексту. У текстовій класифікації та аналізі тональності SVM дозволяють ефективно розподіляти документи по категоріях і визначати емоційне забарвлення текстів.

Дерева рішень (Decision Trees) є одним з найпопулярніших і найзрозуміліших алгоритмів машинного навчання, які використовуються для задач класифікації та регресії. Основна ідея дерева рішень полягає в побудові моделі у вигляді дерева, де кожен внутрішній вузол відповідає за перевірку певної ознаки, кожна гілка представляє результат цієї перевірки, а кожен листовий вузол відповідає за кінцевий результат або клас. Цей підхід дозволяє створювати інтуїтивно зрозумілі та легко інтерпретовані моделі[12].

Процес побудови дерева рішень починається з вибору ознаки, яка найкраще розділяє дані на основі певного критерію, такого як інформаційний приріст (information gain) або критерій Джині (Gini impurity). Потім дані розбиваються на підмножини на основі значень обраної ознаки. Цей процес повторюється рекурсивно для кожної отриманої підмножини, доки всі дані в листових вузлах не будуть належати до одного класу або доки не буде досягнуто заданої глибини дерева.

Дерева рішень мають кілька ключових переваг. Перш за все, вони є простими для розуміння та інтерпретації. Це особливо корисно в ситуаціях, де важлива прозорість моделі, наприклад, у медицині або фінансах. Крім того, дерева рішень здатні працювати як з числовими, так і з категорійними даними, що робить їх універсальними для різних типів задач. Вони також не вимагають попередньої нормалізації або стандартизації даних.

Проте дерева рішень мають і свої недоліки. Одним з основних викликів є схильність до перенавчання, особливо коли дерево стає занадто глибоким і починає запам'ятовувати тренувальні дані замість того, щоб узагальнювати закономірності. Це може призвести до поганої продуктивності на нових даних. Для боротьби з перенавчанням часто використовуються методи обрізання (pruning), які видаляють деякі гілки або листові вузли після початкового створення дерева, щоб зменшити його складність.

Іншим важливим аспектом є вибір правильного критерію для розбиття даних. Різні критерії можуть призвести до побудови різних дерев, що може вплинути на точність і стабільність моделі. Крім того, дерева рішень можуть бути чутливими до невеликих змін у даних: незначні зміни в тренувальному наборі можуть призвести до значної зміни структури дерева[13].

Приклад роботи алгоритму показано на рисунку 1.6.

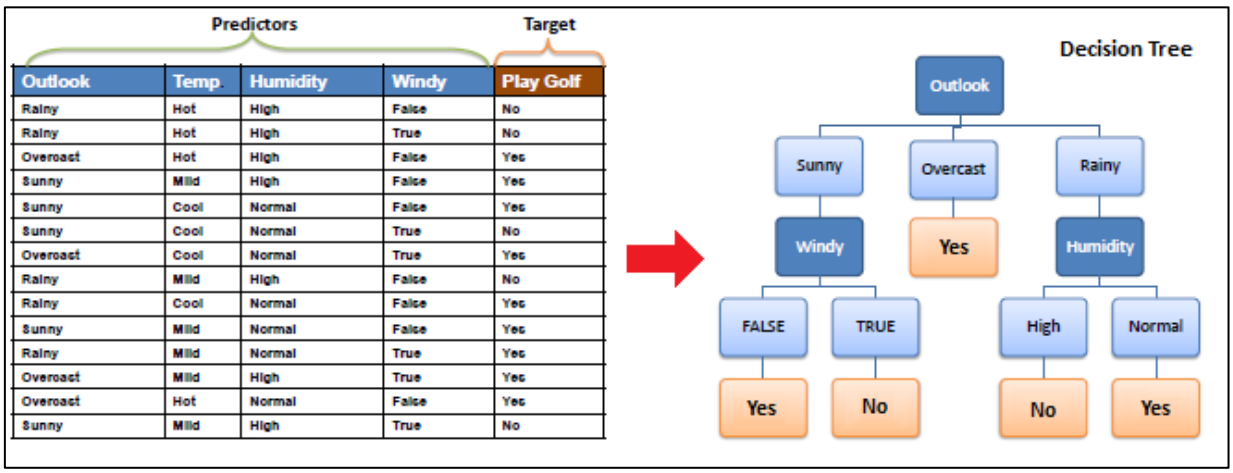


Рисунок 1.6 – Приклад роботи алгоритму Decision Tree.

Дерева рішень знаходять широке застосування у різних галузях. У маркетингу вони використовуються для сегментації клієнтів та прогнозування відтоку. У медицині дерева рішень допомагають у діагностиці захворювань та визначенні прогнозів для пацієнтів. У сфері фінансів вони застосовуються для оцінки кредитного ризику та виявлення шахрайства. У машинному навчанні дерева рішень часто використовуються

як базовий компонент у складніших ансамблевих методах, таких як випадкові ліси (Random Forests) та градієнтний бустинг (Gradient Boosting).

1.3 Вибір оптимальних інформаційних технологій

Виконання поставленого завдання було проведено у середовищі Kaggle, оскільки ця платформа має усі необхідні інструменти для роботи з великими обсягами даних та машинним навчанням.

Kaggle — це онлайн-платформа для змагань з машинного навчання та аналітики даних, яка стала важливою складовою спільноти спеціалістів з обробки даних та машинного навчання. Заснована у 2010 році, Kaggle надає користувачам можливість змагатися у вирішенні реальних задач, пропонованих різними компаніями та організаціями. Змагання на платформі дозволяють фахівцям демонструвати свої навички, співпрацювати з іншими, вчитися новим методам і вигравати призи[14].

Одна з основних переваг Kaggle — це доступ до великих наборів даних, що надаються для кожного змагання. Це дозволяє учасникам працювати з реальними даними та вирішувати практичні проблеми, такі як передбачення продажів, розпізнавання зображень, обробка природної мови та багато інших. Користувачі можуть завантажувати ці дані, аналізувати їх і створювати моделі машинного навчання, які потім оцінюються за допомогою метрик, специфічних для кожного змагання. Крім того, платформа надає інструменти для співпраці, такі як форуми та спільні ноутбуки, що дозволяє учасникам обмінюватися ідеями та кодом[15].

Kaggle також пропонує ресурси для навчання, включаючи курси та туторіали з різних аспектів науки про дані та машинного навчання. Ці матеріали охоплюють основи програмування на Python, статистику, використання бібліотек, таких як TensorFlow та PyTorch, а також більш просунуті теми, як глибоке навчання та обробка природної мови. Це робить

платформу корисною як для новачків, так і для досвідчених професіоналів, які хочуть покращити свої навички або вивчити нові техніки (рис. 1.7).

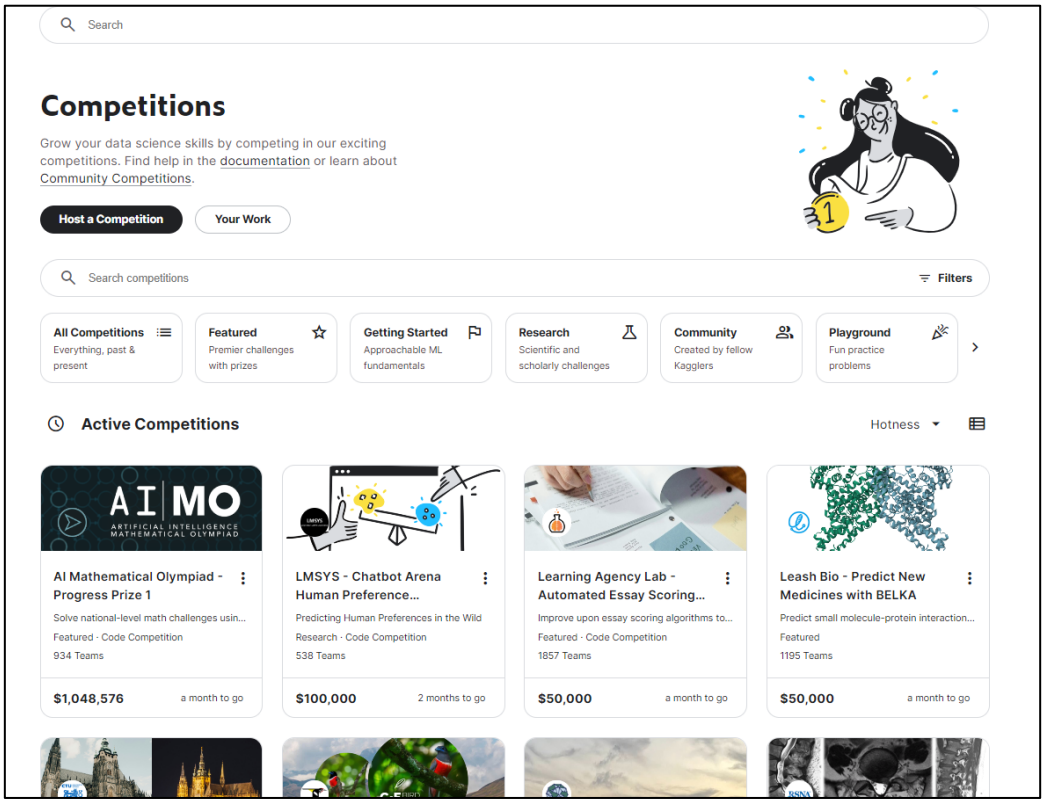


Рисунок 1.7 – Платформа Kaggle.

Для виконання поставленої задачі було обрано мову програмування Python.

Python — це високорівнева мова програмування загального призначення, яка набула великої популярності завдяки своїй простоті, читабельності та широкому спектру застосувань. Створена Гвідо ван Россумом і вперше випущена в 1991 році, Python вирізняється своєю лаконічною і зрозумілою синтаксичною структурою, що робить її особливо привабливою для новачків у програмуванні. Водночас, мова володіє достатньою потужністю та гнучкістю, щоб бути ефективним інструментом для досвідчених розробників[16].

Однією з ключових особливостей Python є його філософія, що відображається у принципах, викладених у документі "The Zen of Python". Ці

принципи наголошують на важливості зрозумілого та чистого коду, що сприяє легкому спільному використанню і підтримці програм. Простота і читабельність Python досягається завдяки відступам, які використовуються для позначення блоків коду, а не фігурних дужок або ключових слів, як у багатьох інших мовах програмування.

Python підтримує декілька парадигм програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне програмування. Це дозволяє розробникам обирати найбільш підходящий стиль кодування для вирішення конкретних задач. Мова також має динамічну типізацію і автоматичне управління пам'яттю, що спрощує процес розробки і зменшує кількість помилок[17].

Одна з головних переваг Python полягає у великій кількості бібліотек і фреймворків, які розширюють його функціональні можливості. Наприклад, бібліотека NumPy забезпечує потужні засоби для роботи з багатовимірними масивами і математичними функціями, а Pandas пропонує інструменти для аналізу і маніпуляції даними. Matplotlib і Seaborn дозволяють створювати високоякісні візуалізації даних. TensorFlow і PyTorch є провідними бібліотеками для розробки моделей машинного навчання і глибокого навчання. Django і Flask використовуються для розробки веб-додатків, а Pygame — для створення ігор[18].

Окрім вищезазначених бібліотек та фреймворків, Python має багато інших інструментів та розширень, які роблять його ще більш потужним і універсальним. Наприклад, бібліотека SciPy розширює можливості NumPy, надаючи інструменти для наукових і технічних обчислень. SymPy використовується для символічної математики, що дозволяє виконувати алгебраїчні маніпуляції і розв'язувати рівняння символічно.

Для роботи з великими даними та аналізу використовується бібліотека Dask, яка дозволяє обробляти великі масиви даних, що не вміщуються в оперативну пам'ять, за допомогою паралельних обчислень. Веб-скрапінг

можна здійснювати за допомогою бібліотек BeautifulSoup і Scrapy, які надають інструменти для збирання даних з веб-сайтів.

Python також активно використовується в автоматизації тестування програмного забезпечення. Бібліотеки, такі як unittest, pytest і Selenium, дозволяють створювати тести для перевірки функціональності програм і автоматизувати процес тестування[19].

Ще однією важливою областю застосування Python є розробка інтерфейсів користувача (GUI). Інструменти, такі як Tkinter, PyQt і Kivy, дозволяють створювати графічні інтерфейси для настільних додатків.

У галузі обробки тексту і природної мови (NLP) популярними є бібліотеки NLTK (Natural Language Toolkit) і spaCy, які надають інструменти для аналізу тексту, токенізації, частиномовного розбору та інших задач обробки природної мови.

Python також знайшов своє застосування в області Інтернету речей (IoT). Бібліотеки, такі як MicroPython і CircuitPython, адаптовані для роботи на мікроконтролерах, дозволяючи створювати додатки для IoT пристроїв.

1.4 Висновки

У першому розділі було визначено об’єкт дослідження та обґрунтовано його актуальність. Також були розглянуті та вибрані методи та моделі класифікації, проаналізовано можливі варіанти їх реалізації та обрано оптимальні інформаційні технології для досягнення поставленої мети. В результаті для виконання поставленої задачі було обрано мову програмування Python, середовище Kaggle та такі моделі машинного навчання: Support Vector Machines, Random Forest, KNN, Decision Tree.

2 ПІДГОТОВКА ДАНИХ ТА РОЗВІДУВАЛЬНИЙ АНАЛІЗ

2.1 Огляд та підготовка датасету

Перед початком аналізу даних необхідно здійснити імпорт необхідних бібліотек та виконати первинну підготовку даних для подальшого аналізу. Спочатку було здійснено імпорт усіх необхідних бібліотек для виконання поставлених завдань. Це включало бібліотеки для обробки даних, візуалізації, статистичного аналізу та машинного навчання.

Імпорт бібліотек показано на рисунку 2.1.

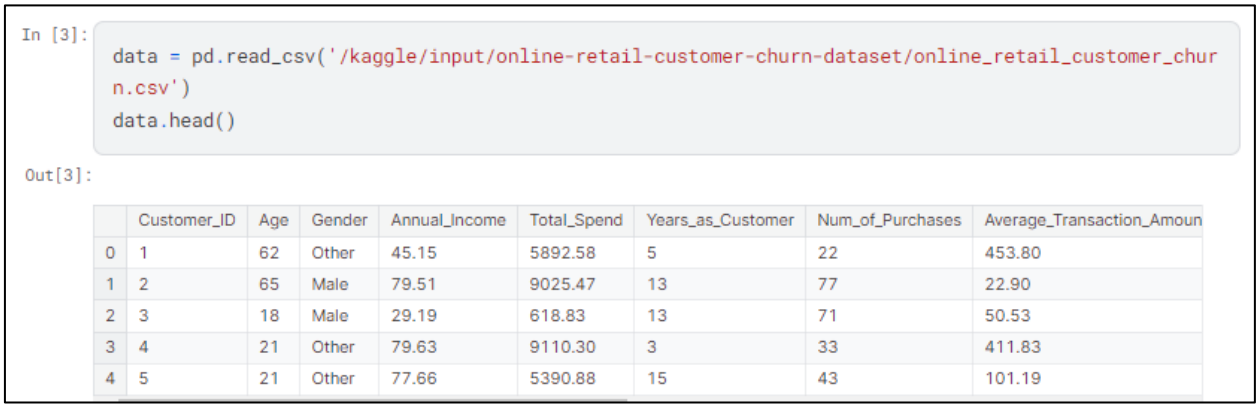
```
[ ]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')


# sklearn imports
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score, precision_score, f1_score, roc_auc_score, recall_score
```

Рисунок 2.1 – Імпорт необхідних бібліотек до проекту

Далі, після імпорту бібліотек та підготовки середовища, наступним кроком стало завантаження пакету даних та виведення певних його частин для перевірки.

Цей етап є важливим для отримання загального уявлення про набір даних і виявлення можливих проблем або особливостей, які потребують уваги під час подальшого аналізу. Результати цього етапу зберігаються для подальшого використання і можуть бути використані для додаткового аналізу та обробки.

Загальний вигляд датасету показано на рисунку 2.2.



In [3]:
```
data = pd.read_csv('/kaggle/input/online-retail-customer-churn-dataset/online_retail_customer_churn.csv')
data.head()
```

Out[3]:

	Customer_ID	Age	Gender	Annual_Income	Total_Spend	Years_as_Customer	Num_of_Purchases	Average_Transaction_Amoun
0	1	62	Other	45.15	5892.58	5	22	453.80
1	2	65	Male	79.51	9025.47	13	77	22.90
2	3	18	Male	29.19	618.83	13	71	50.53
3	4	21	Other	79.63	9110.30	3	33	411.83
4	5	21	Other	77.66	5390.88	15	43	101.19

Рисунок 2.2 – Завантаження та огляд датасету

Датасет містить такі стовпці:

– Customer_ID: унікальний ідентифікатор для кожного клієнта.

– Вік: Вік клієнта.

– Стать: стать клієнта (чоловіча, жіноча, інша).

– Annual_Income: річний дохід клієнта в тисячах доларів.

– Total_Spend: загальна сума, витрачена клієнтом за останній рік.

– Years_as_Customer: кількість років, протягом яких особа була клієнтом магазину.

– Num_of_Purchases: кількість покупок, які клієнт зробив за останній рік.

– Average_Transaction_Amount: середня сума, витрачена на транзакцію.

– Num_of_Returns: кількість товарів, які клієнт повернув за останній рік.

– Num_of_Support_Contacts: скільки разів клієнт звертався до служби підтримки за останній рік.

– Satisfaction_Score: Оцінка від 1 до 5, що вказує на задоволеність клієнта магазином.

– Last_Purchase_Days_Ago: кількість днів після останньої покупки клієнта.

– Email_Opt_In: чи погодився клієнт отримувати маркетингові електронні листи.

– Promotion_Response: відповідь клієнта на останню рекламну кампанію (відповів, проігнорував, скасував підписку).

– Target_Churn: вказує, чи відтік клієнт (істинне чи хибне).

Після більш детального розгляду датасету, було визначено що він має 1000 рядків та 14 стовпців (рис. 2.3).

```
In [11]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 15 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Customer_ID                 1000 non-null   int64
 1   Age                         1000 non-null   int64
 2   Gender                      1000 non-null   object
 3   Annual_Income               1000 non-null   float64
 4   Total_Spend                 1000 non-null   float64
 5   Years_as_Customer           1000 non-null   int64
 6   Num_of_Purchases            1000 non-null   int64
 7   Average_Transaction_Amount  1000 non-null   float64
 8   Num_of_Returns              1000 non-null   int64
 9   Num_of_Support_Contacts     1000 non-null   int64
 10  Satisfaction_Score          1000 non-null   int64
 11  Last_Purchase_Days_Ago      1000 non-null   int64
 12  Email_Opt_In                1000 non-null   bool
 13  Promotion_Response          1000 non-null   object
 14  Target_Churn                1000 non-null   bool
```

Рисунок 2.3 – Загальна інформація про датасет.

Оскільки стовпець «Customer ID» не має ніякої цінності для подальшого аналізу, його було видалено (рис. 2.4).

```
data.drop(columns='Customer_ID',axis=1, inplace=True)
data.columns
```

```
Out[12]:
Index(['Age', 'Gender', 'Annual_Income', 'Total_Spend', 'Years_as_Customer',
       'Num_of_Purchases', 'Average_Transaction_Amount', 'Num_of_Returns',
       'Num_of_Support_Contacts', 'Satisfaction_Score',
       'Last_Purchase_Days_Ago', 'Email_Opt_In', 'Promotion_Response',
       'Target_Churn'],
      dtype='object')
```

Рисунок 2.4 – Видалення стовпця.

Також, важливо розділити числові та категоріальні стовпці (рис. 2.5).

```
# split numerical and categorical columns
numerical_cols = data.select_dtypes(include=['int','float']).columns
numerical_cols
```

```
Index(['Age', 'Annual_Income', 'Total_Spend', 'Years_as_Customer',
       'Num_of_Purchases', 'Average_Transaction_Amount', 'Num_of_Returns',
       'Num_of_Support_Contacts', 'Satisfaction_Score',
       'Last_Purchase_Days_Ago'],
      dtype='object')
```

```
# categorical colums
categorical_cols = data.select_dtypes(exclude=['int','float']).columns
categorical_cols
```

```
Index(['Gender', 'Email_Opt_In', 'Promotion_Response', 'Target_Churn'], dtype='object')
```

Рисунок 2.5 – Розділення даних.

Далі було перевірено датасет на наявність пустих комірок та дублікатів значень (рис. 2.6).

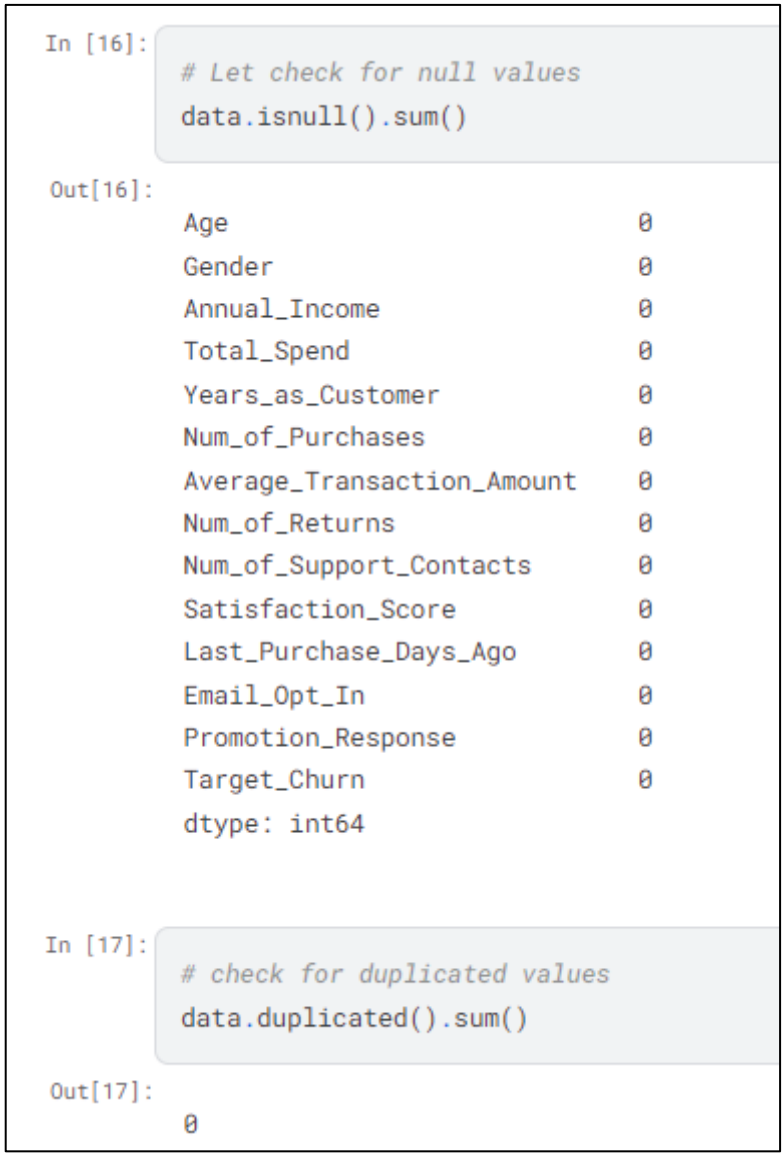


Рисунок 2.6 – Перевірка на наявність пустих комірок та дублікатів.

Таким чином, імпортувавши бібліотеки, підключивши та оглянувши датасет можна переходити до розвідувального аналізу.

2.2 Розвідувальний аналіз

Розвідувальний аналіз даних (EDA) — це етап у процесі аналізу даних, коли дослідник вивчає структуру та характеристики набору даних. Під час

EDA дослідник виявляє закономірності, визначає відмінності та формулює гіпотези, що можуть бути перевірені під час подальшого аналізу.

Основні завдання розвідувального аналізу даних включають огляд даних, перевірку типів даних, аналіз відсутніх значень, вивчення розподілу ознак, виявлення взаємозв'язків між ознаками та виявлення викидів або аномалій. Розвідувальний аналіз даних допомагає досліднику краще зрозуміти датасет і підготувати дані для подальшого використання в аналізі або моделюванні[20].

На рисунку 2.7 показано функцію яка виконує однофакторний аналіз числових стовпців.

```
def univariate_analysis_numeric(col):
    # Create subplots
    fig, ax = plt.subplots(1, 2, figsize=(10, 6))

    # Histogram and kde
    sns.histplot(x=data[col], bins=30, stat='frequency',kde=True, ax=ax[0])
    ax[0].set_title(f'Distribution of {col}')

    # Boxplot
    sns.boxplot(x=data[col], ax=ax[1])
    ax[1].set_title(f'Boxplot for {col}')

    plt.tight_layout()
    plt.show()
```

+ Code + Markdown

```
numerical_cols
```

```
Index(['Age', 'Annual_Income', 'Total_Spend', 'Years_as_Customer',
       'Num_of_Purchases', 'Average_Transaction_Amount', 'Num_of_Returns',
       'Num_of_Support_Contacts', 'Satisfaction_Score',
       'Last_Purchase_Days_Ago'],
      dtype='object')
```

+ Code + Markdown

Рисунок 2.7 – Функція однофакторного аналізу.

Однофакторний аналіз для стовпців віку показано на рисунку 2.8.

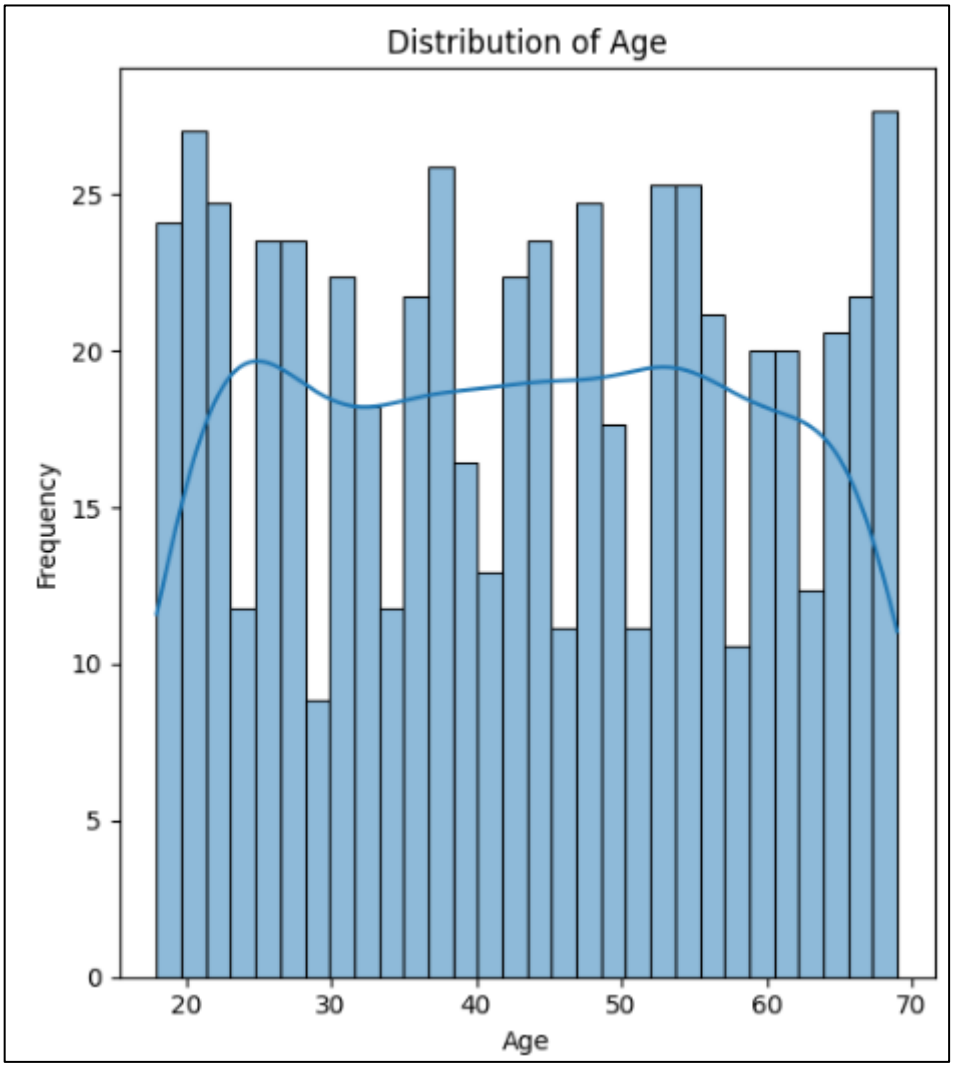


Рисунок 2.8 – Однофакторний аналіз для стовпця віку.

Графік показує розподіл віку покупців інтернет-магазину, де найбільша кількість клієнтів зосереджена у віковому діапазоні від 25 до 57 років. Це свідчить про те, що ця категорія вікова є найбільш активною серед покупців. Найменша кількість клієнтів спостерігається серед людей віком в 30 років, і серед старшого покоління, віком 65 років. Це може вказувати на неефективність маркетингових стратегій для залучення цих вікових груп.

В цілому, аналіз розподілу віку покупців може допомогти оптимізувати маркетинг та рекламні кампанії для більш ефективного залучення різних вікових категорій клієнтів.

Однофакторний аналіз для стовпця річного доходу показано на рисунку 2.9.

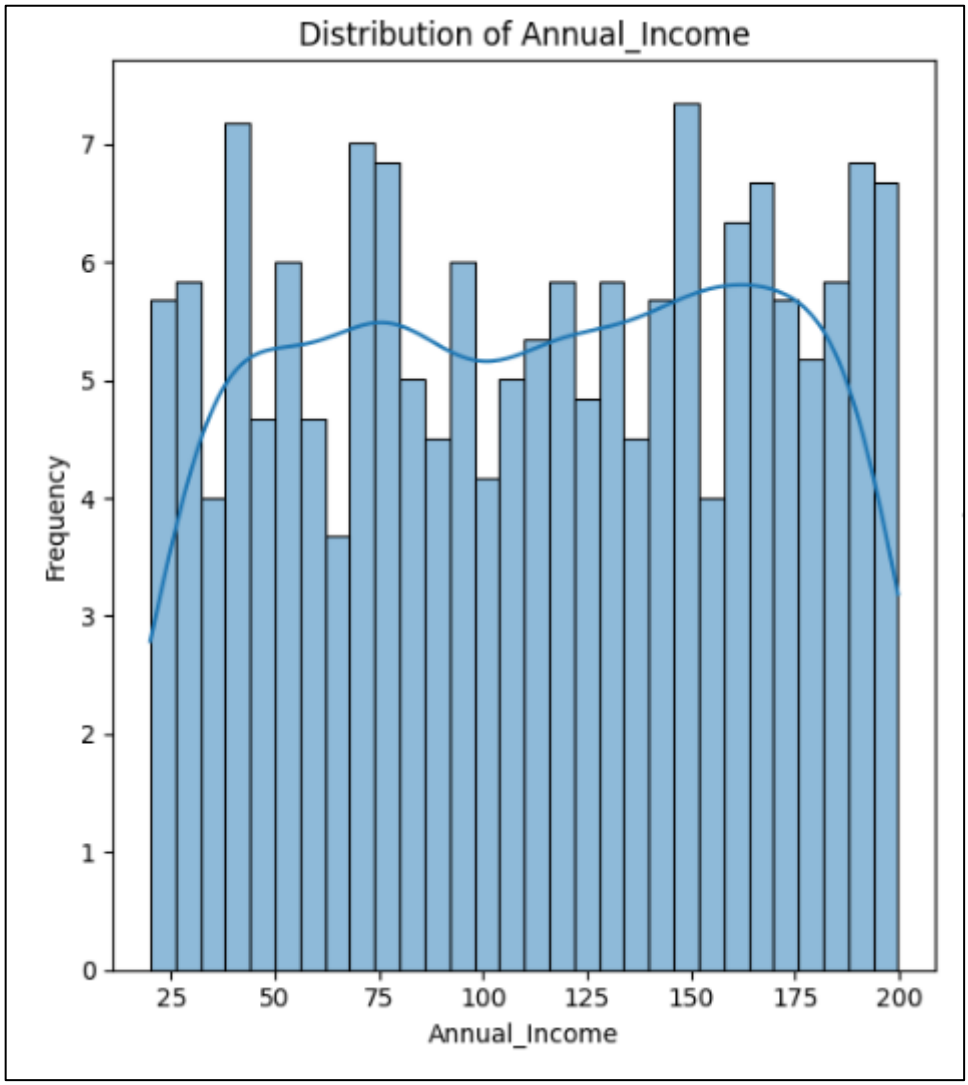


Рисунок 2.9 – Однофакторний аналіз для стовпця річного доходу.

Графік показує розподіл річного доходу покупців інтернет-магазину. За результатами аналізу видно, що більшість покупців мають річний дохід у діапазоні від 50 000 до 150 000 доларів США. Це може вказувати на те, що цільова аудиторія магазину складається з осіб середнього класу.

Найменша кількість покупців мають низький річний дохід (менше 25 000 доларів США) або високий річний дохід (більше 200 000 доларів США). Ці дані можуть бути важливими при розробці стратегій залучення та

утримання різних категорій покупців, зокрема, налаштування ціноутворення та маркетингових акцій.

Однофакторний аналіз для стовпця загальних витрат показано на рисунку 2.10.

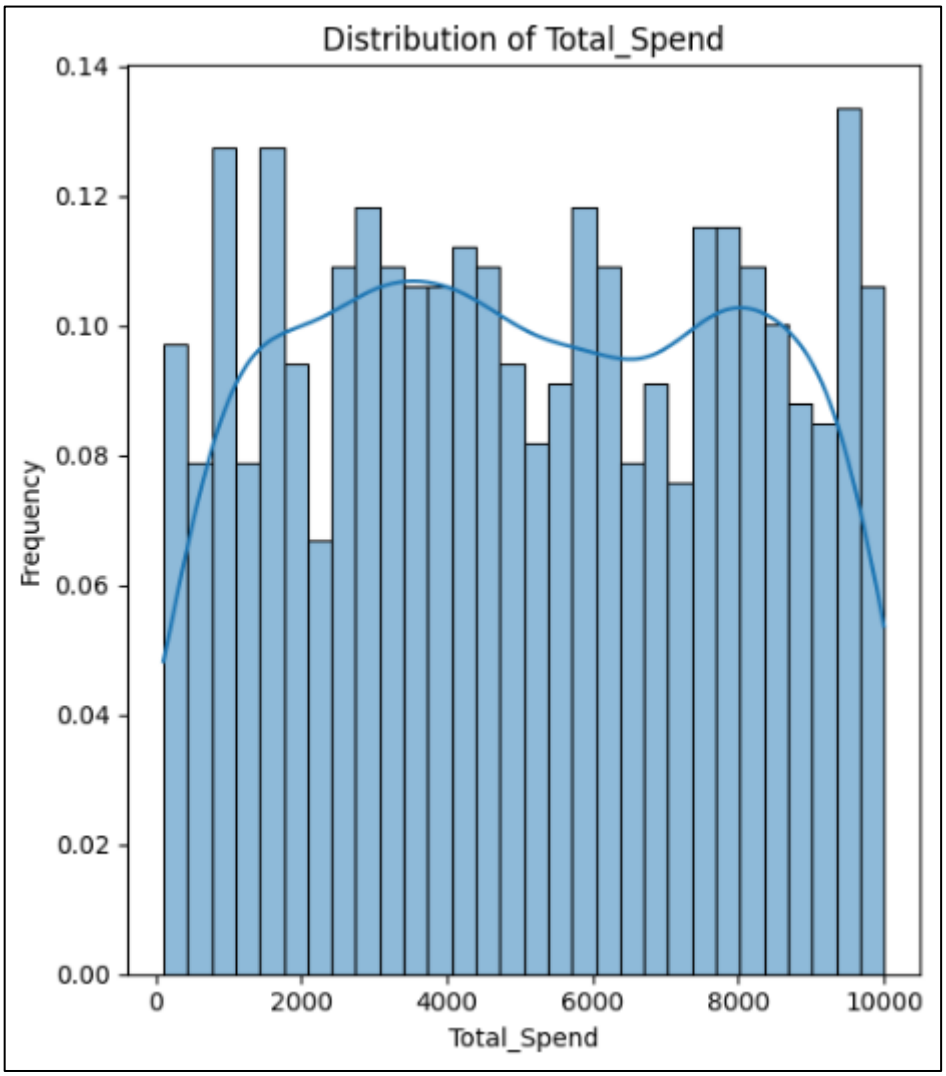


Рисунок 2.10 – Однофакторний аналіз для стовпця загальних витрат.

Графік показує розподіл загальних витрат покупців інтернет-магазину. Більшість осіб у цій групі споживачів витрачають суму від 2000 до 10 000 доларів США. Найменша кількість покупців витрачають менше 2000 доларів США або більше 10 000 доларів США, що може вказувати на те, що менша

частина споживачів має дуже обмежений або дуже великий бюджет для покупок.

Однофакторний аналіз для стовпця часу співпраці показано на рисунку 2.11.

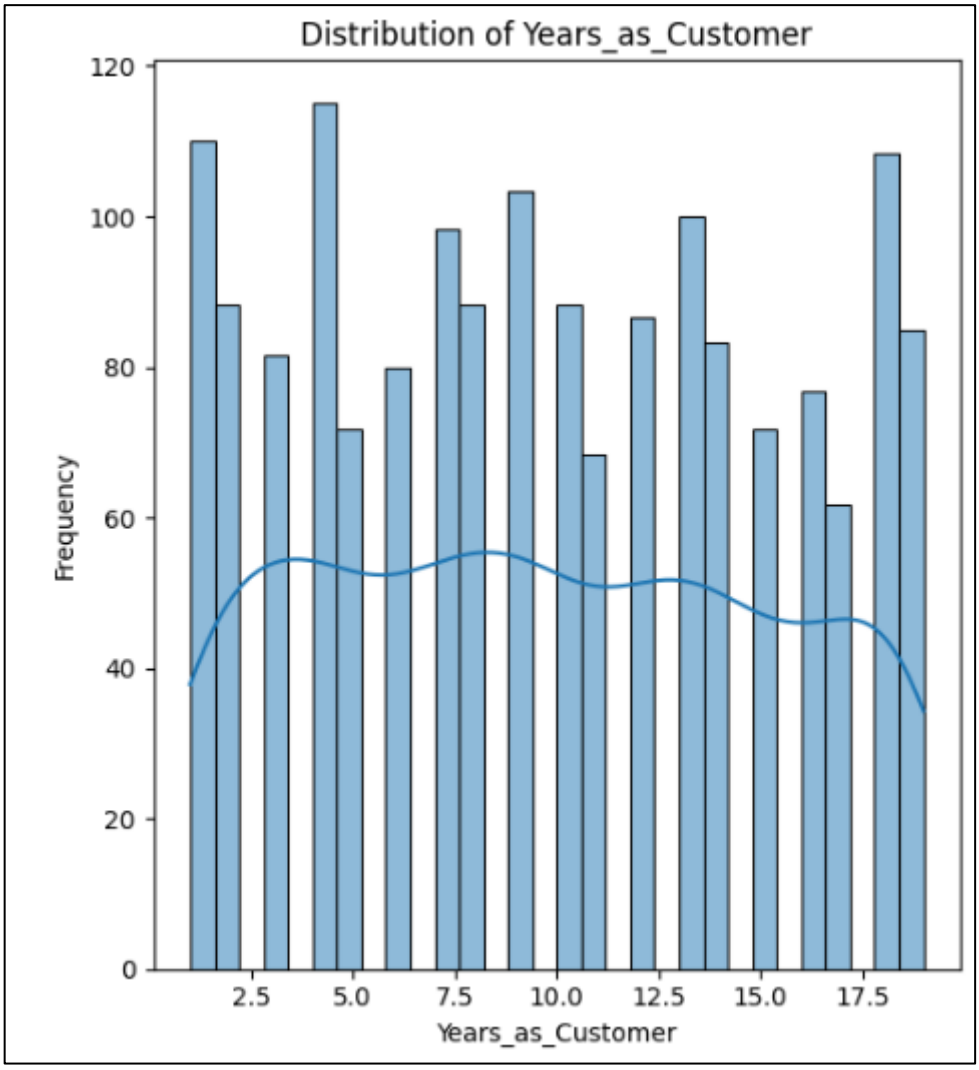


Рисунок 2.11 – Однофакторний аналіз для стовпця часу співпраці

Графік показує розподіл років співпраці покупців з компанією. Більшість споживачів у цій групі співпрацюють з компанією протягом 2,5 до 7,5 років, що може свідчити про стабільність відносин між покупцями та компанією. Цей розподіл відображає типові тенденції відтоку клієнтів і може бути корисним для розробки стратегій залучення та утримання клієнтів у майбутньому.

Однофакторний аналіз для стовпця кількості покупок показано на риснку 2.12.

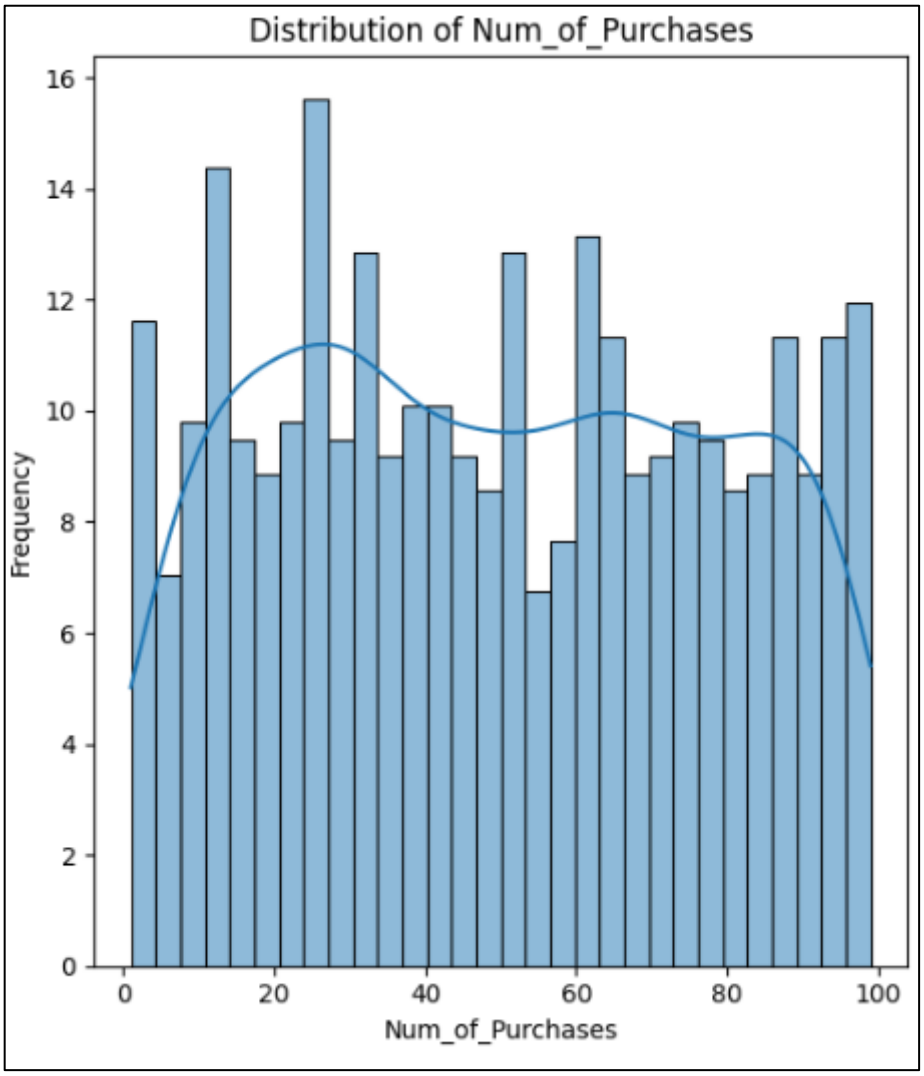


Рисунок 2.12 – Однофакторний аналіз для стовпця кількості покупок

Дані свідчать про те, що більшість покупців онлайн-магазину здійснюють лише кілька покупок протягом певного періоду. Це може свідчити про те, що магазин не надає покупцям достатньо причин для повторних покупок. Магазину слід детально проаналізувати причини, чому покупці не повертаються для здійснення повторних покупок.

Однофакторний аналіз для стовпця середньої суми транзакцій показано на рисунку 2.13.

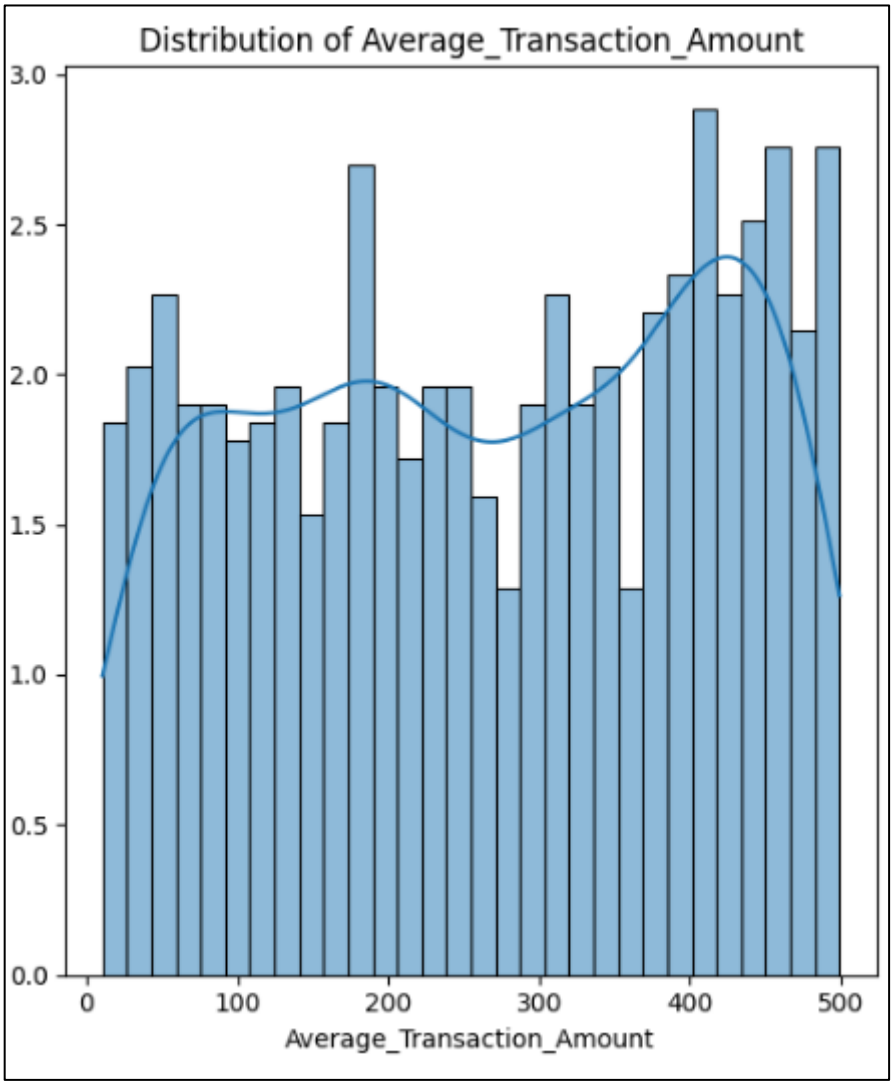


Рисунок 2.13 – Однофакторний аналіз для стовпця середньої суми транзакцій.

З графіка видно, що розподіл середньої суми транзакцій є правобічним. Це означає, що більшість транзакцій мають малу середню суму транзакції, а кількість транзакцій із великою середньою сумою транзакції значно менша.

На основі однофакторного аналізу можна зробити висновок, що середня сума транзакцій не має значного впливу на частоту транзакцій. Більшість транзакцій мають малу середню суму транзакції, незалежно від того, яка середня сума транзакції.

Однофакторний аналіз для стовпця кількості повернень показано на рисунку 2.14.

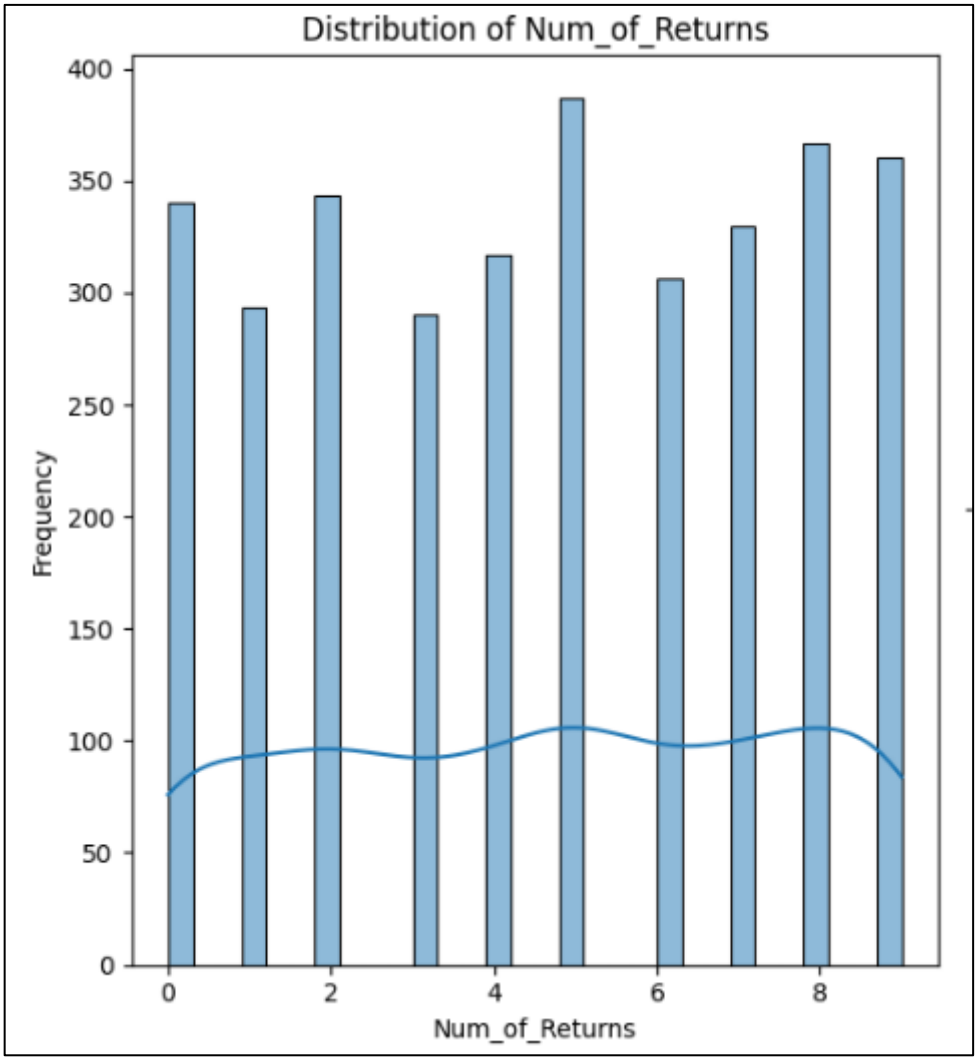


Рисунок 2.14 – Однофакторний аналіз для стовпця кількості повернень

З графіка видно, що розподіл кількості повернень є правобічним. Це означає, що більшість повернень мають малу кількість повернень, а кількість повернень з великою кількістю повернень значно менша.

Однофакторний аналіз для стовпця кількості запитів у підтримку показано на рисунку 2.15.

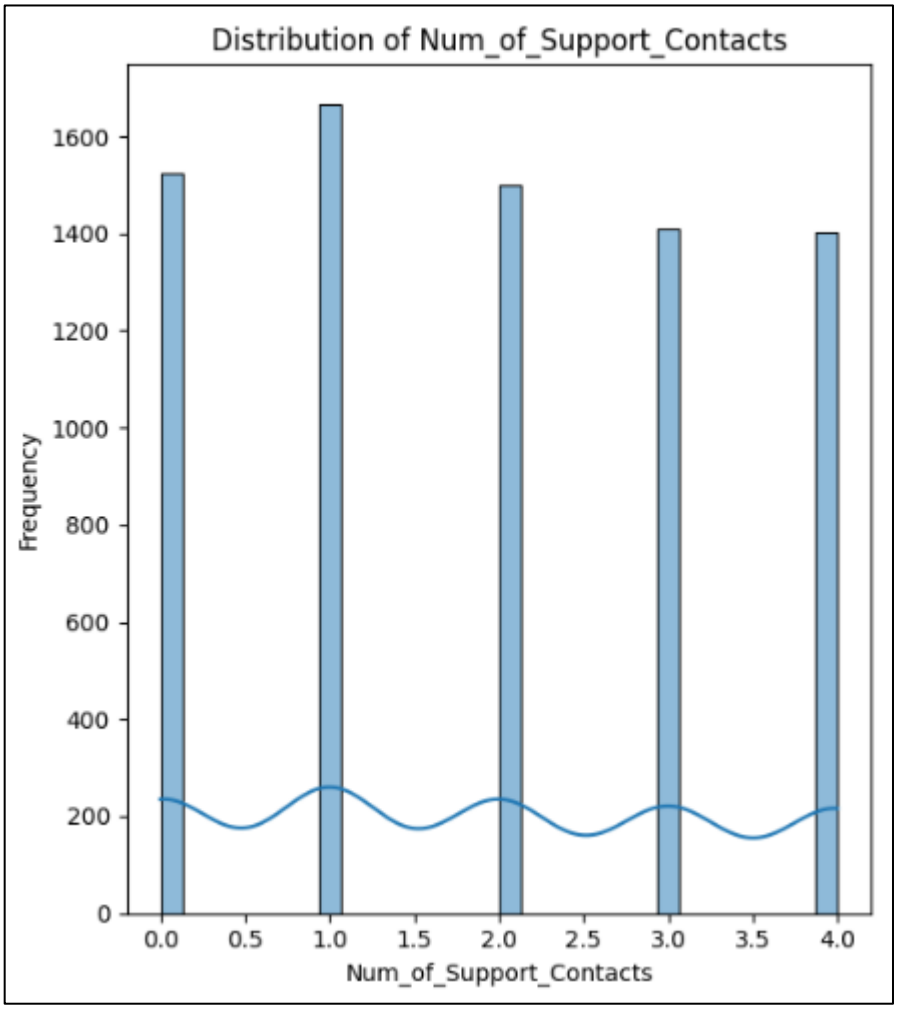


Рисунок 2.15 – Однофакторний аналіз для стовпця кількості запитів у підтримку

Аналіз даних показує, що найбільша частина клієнтів (32,4%) не мають жодних запитів до служби підтримки. Це може вказувати на те, що ці клієнти можуть бути задоволені обслуговуванням або не виявляти проблем з продуктами чи послугами магазину.

При цьому виявлено, що 27,4% клієнтів роблять лише один запит, що може вказувати на те, що вони зіткнулися з певною проблемою або мають питання, які потребують вирішення.

Додатково, 18,7% клієнтів роблять два запити, 12,5% - три запити, а 9% - чотири і більше запитів. Це може свідчити про те, що частина клієнтів може мати серйозніші проблеми або незадоволеність, яку не вдалося вирішити за один або два запити.

Враховуючи ці дані, магазин може скорегувати свою стратегію обслуговування клієнтів, зосереджуючись на покращенні якості підтримки та вирішенні проблем клієнтів з метою зниження відтоку і збереження клієнтів.

Однофакторний аналіз для стовпця оцінки задоволеності показано на рисунку 2.16.

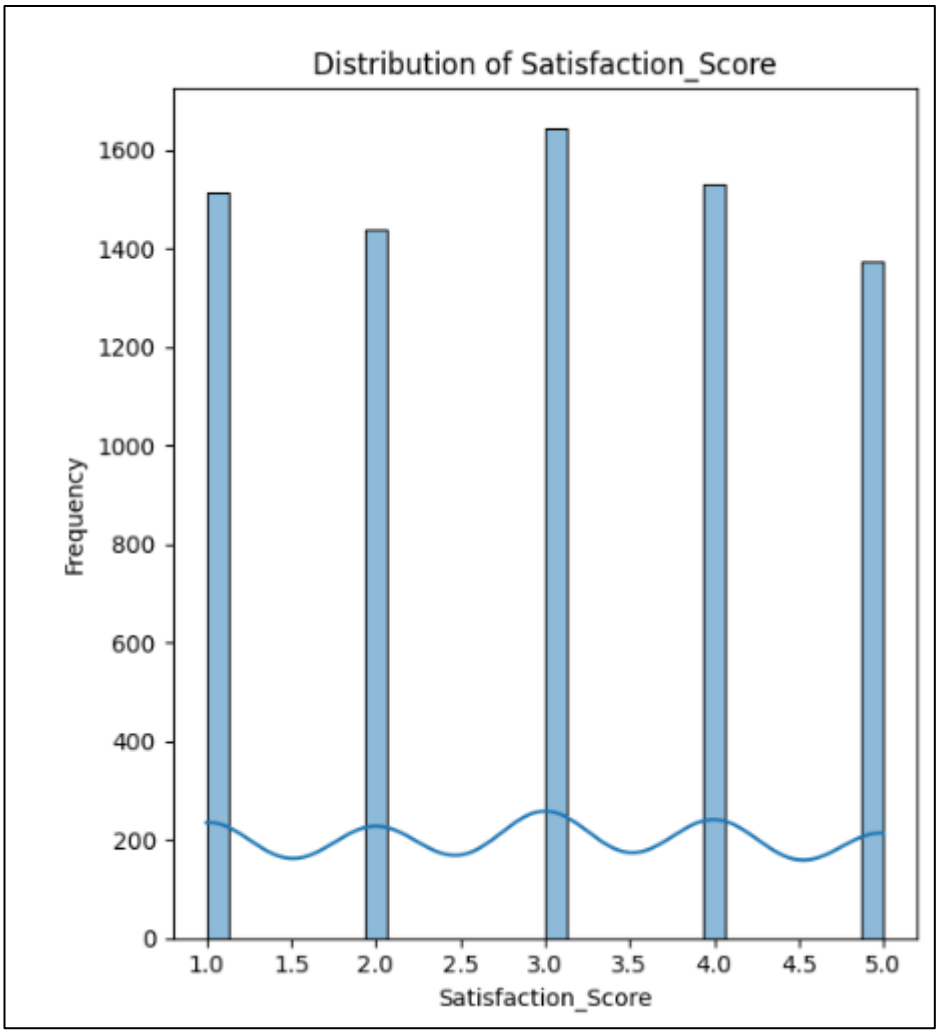


Рисунок 2.16 – Однофакторний аналіз для стовпця оцінки задоволеності

Аналіз отриманих даних показує, що більшість клієнтів онлайн-магазину (40,2%) поставили найвищу оцінку - 5, висловлюючи задоволеність своїм досвідом роботи з магазином. Додатково, 34,8% клієнтів також вважають свій досвід позитивним, оцінивши його на 4 бали.

З іншого боку, лише невелика частка клієнтів (6,1%) виразили незадоволення своїм досвідом роботи з магазином, поставивши оцінки 1 або 2. Це свідчить про те, що більшість клієнтів залишаються задоволеними обслуговуванням та якістю продуктів або послуг, що надаються магазином.

Загальний висновок полягає в тому, що більшість клієнтів цінують свій досвід роботи з магазином, що може свідчити про ефективність стратегій обслуговування та маркетингу, застосованих магазином. Однак важливо продовжувати моніторити і покращувати ці процеси, щоб забезпечити тривалу задоволеність клієнтів та зменшити кількість незадоволених клієнтів.

Однофакторний аналіз для стовпця кількості днів з останньої покупки показано на рисунку 2.17.

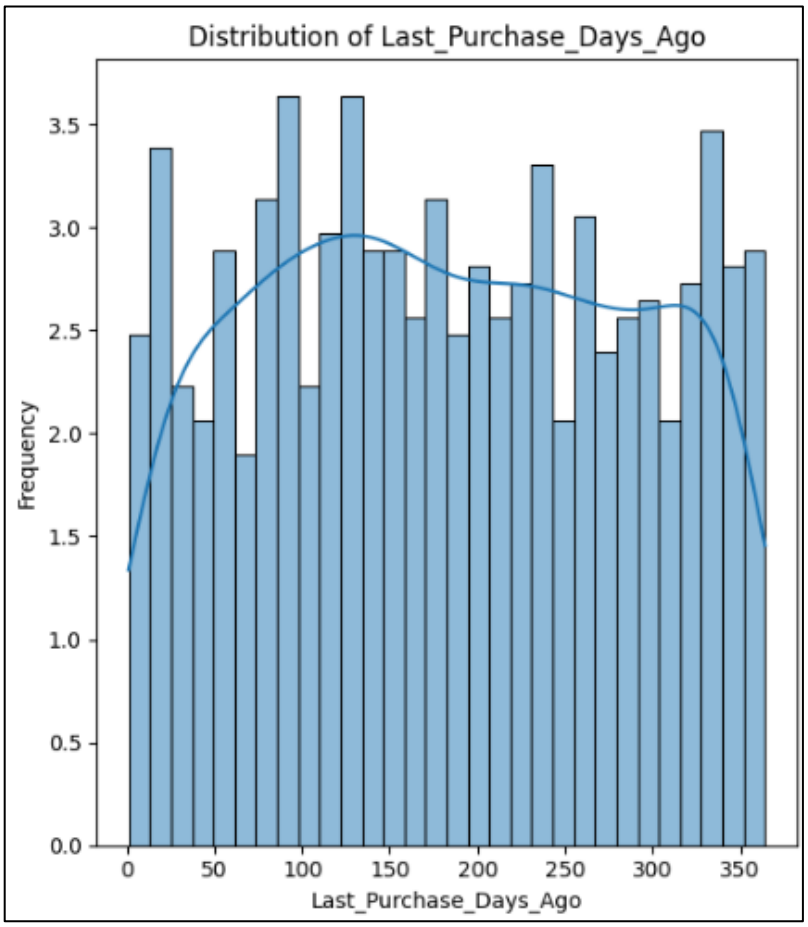


Рисунок 2.17 – Однофакторний аналіз для стовпця кількості днів з останньої покупки

За даними графіка, можна зробити висновок про активність клієнтів у здійсненні покупок протягом останніх декількох місяців.

Більшість клієнтів онлайн-магазину (61,7%) не здійснювали покупок протягом останніх 90 днів. Це може вказувати на те, що значна частина клієнтської бази може бути неактивною або втратила інтерес до продуктів або послуг, які пропонує магазин.

Тільки 38,3% клієнтів робили покупки протягом останніх 90 днів. Це може свідчити про необхідність проведення додаткового аналізу та вжиття заходів для стимулювання активності клієнтів, таких як маркетингові акції, знижки або програми лояльності, щоб збільшити участь клієнтів у покупках та зберегти їхній інтерес до магазину.

Графіки розподілу клієнтів за статтю показано на рисунку 2.18.

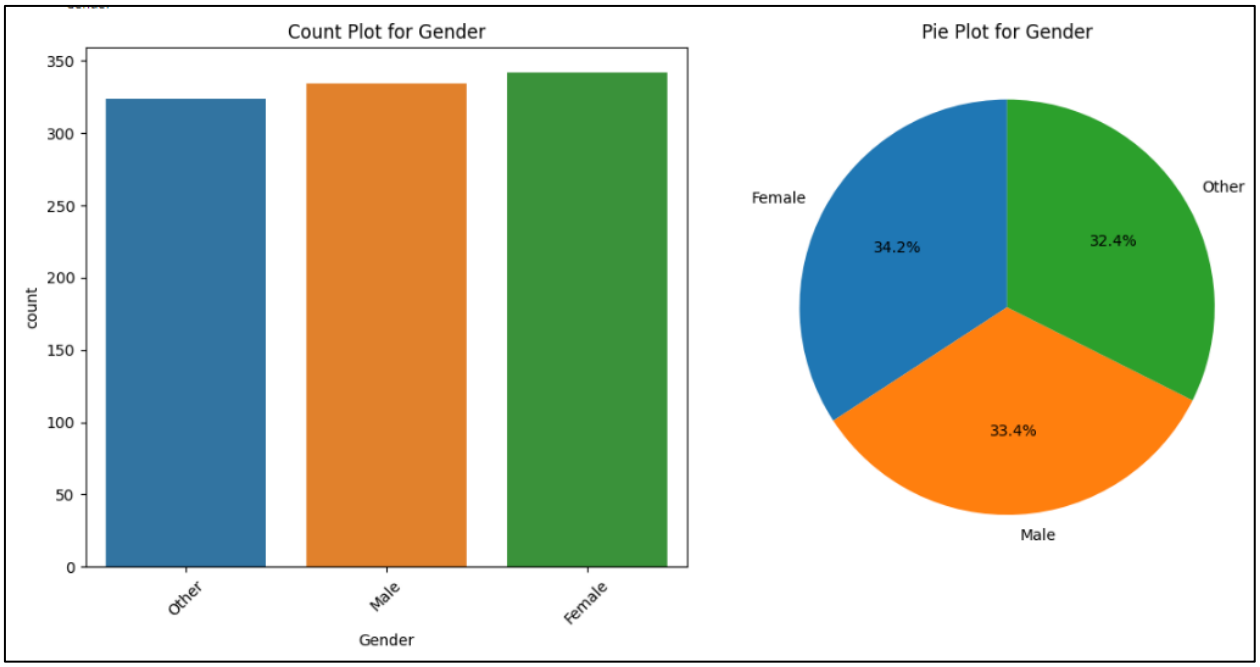


Рисунок 2.18 – Розподіл клієнтів за статтю.

Графік показує, що розподіл клієнтів за статтю є досить рівномірним. Найбільш поширеною статтю серед клієнтів є жіноча, яка становить 34,2%. На другому місці за поширеністю йде чоловіча стать, яка становить 33,4%. Найменш поширеною статтю серед клієнтів є інша, яка становить 32,4%.

Жіноча та чоловіча стать трохи переважають, але кількість клієнтів з іншою статтю також значна. Це свідчить про те, що магазин привертає увагу різних категорій клієнтів, що може бути корисним при розробці цільових маркетингових стратегій.

Графіки розподілу за тим чи вказана поштова скринька показано на рисунку 2.19.

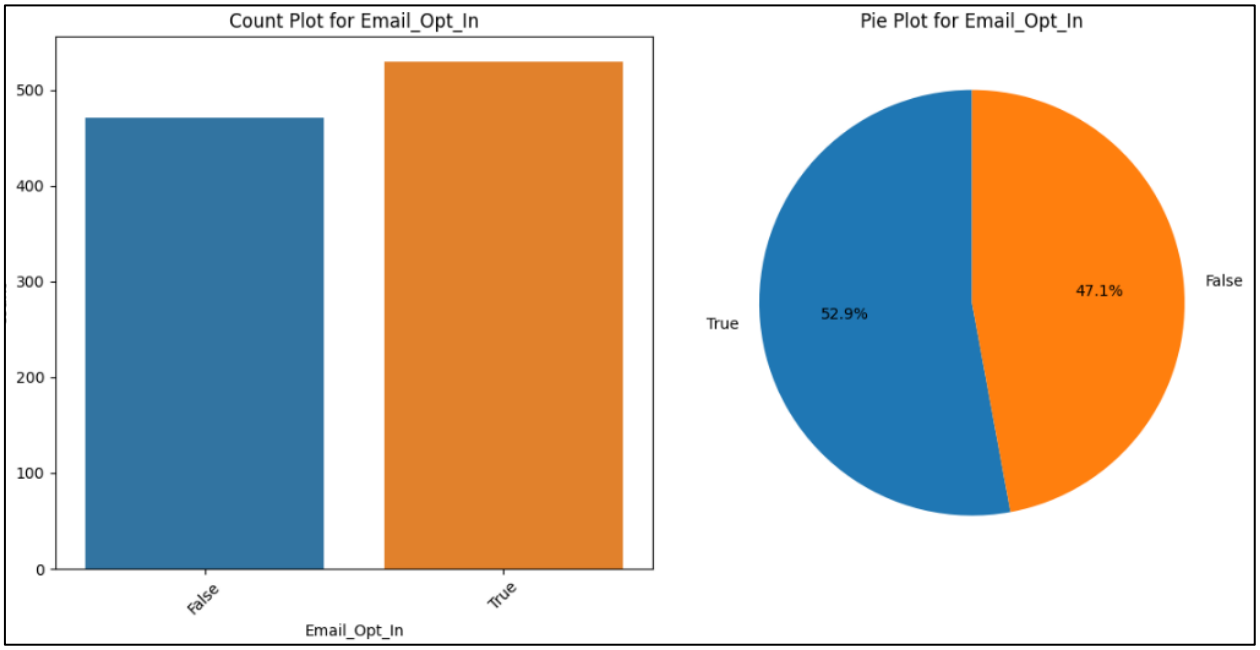


Рисунок 2.19 – Графік розподілу за тим чи вказана поштова скринька.

Ці дані вказують на те, що трохи більше половини клієнтів надали інформацію про свою поштову скриньку. Це може бути корисним для розробки маркетингових стратегій, таких як розсилки електронною поштою, оскільки магазин має доступ до контактних даних значної частини своїх клієнтів. Однак, для покращення комунікації з усіма клієнтами, варто розглянути способи заохочення інших клієнтів до надання своїх контактних даних.

Графік кореляції між стовпцями показано на рисунку 2.20.

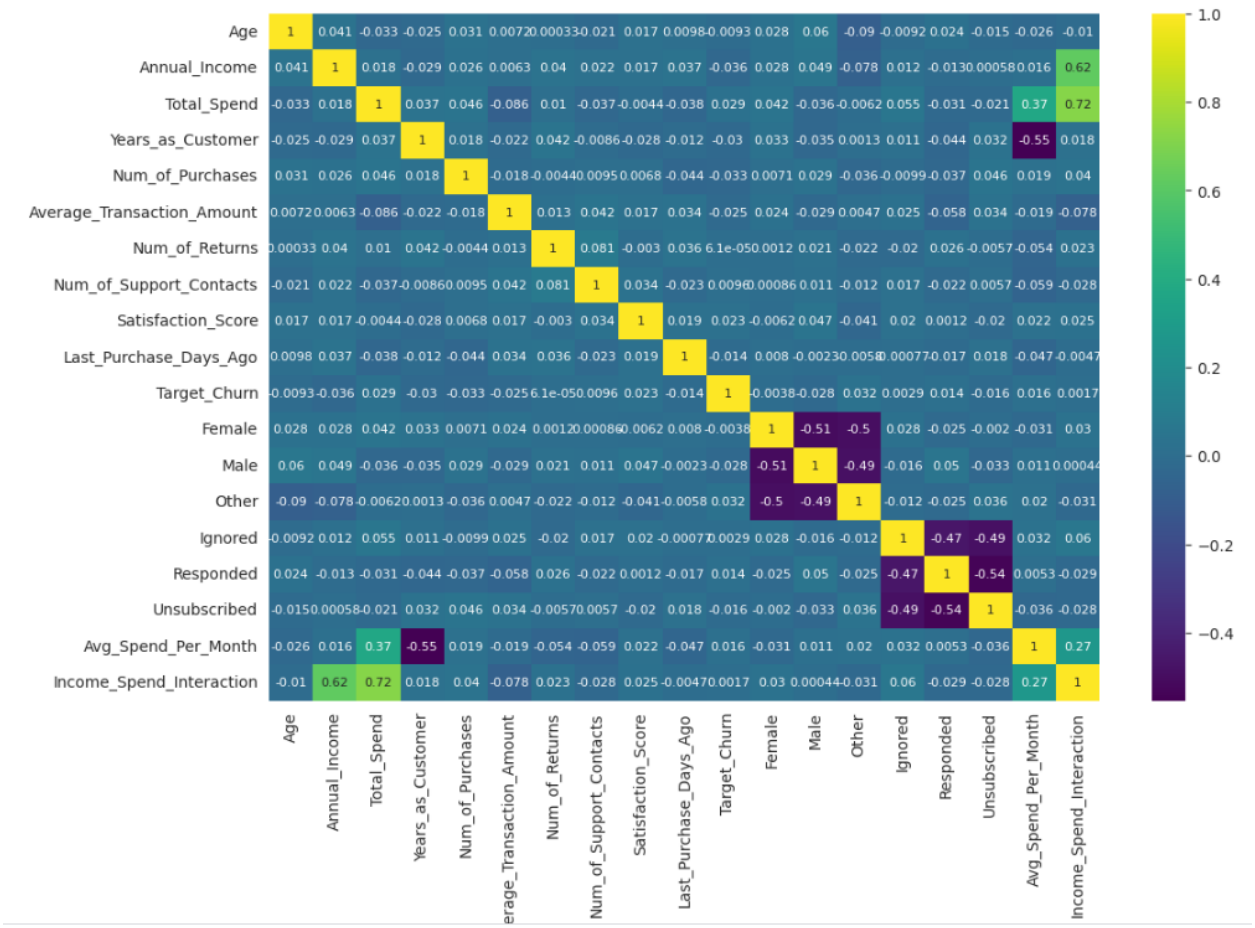


Рисунок 2.20 – Графік кореляції.

Як можна бачити на графіку кореляції показників, більшість з них мають доволі слабкий зв’язок, в той час як показники Total_Spend та Annual_Income показують гарні результати. Це означає що доволі сильний зв’язок між загальними витратами та загальним прибутком клієнта з вірогідністю його відтоку з інтернет-магазину.

2.3 Висновки

У даному розділі проведено підготовку даних та розвідувальний аналіз датасету щодо відтоку клієнтів інтернет-магазину. Цей аналіз дозволив виявити ключові групи клієнтів, які схильні до відтоку, а також визначити основні фактори, які впливають на їх рішення щодо залишення магазину. Було

визначено що показники Total_Spend та Annual_Income мають доволі сильний зв’язок з відтоком клієнтів з інтернет-магазину.

3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ АНАЛІЗУ ДАНИХ

3.1 Розроблення алгоритму інформаційної технології

У відповідності з комп'ютерним програмуванням, алгоритм описується як послідовність чітко визначених інструкцій, які спрямовані на вирішення конкретної проблеми. Ці інструкції приймають вхідні дані та забезпечують вихідний результат.

Алгоритм може бути реалізований у вигляді програмного коду або математичних формул. Він описує послідовність дій, які потрібно виконати для досягнення бажаного результату. Ефективність алгоритму часто оцінюється за допомогою його часу виконання та точності результату.

Для розробленої інформаційної технології було визначено алгоритм його роботи, що включає в себе опис послідовності операцій, які потрібно виконати системі, щоб досягти бажаного функціонального результату. Цей алгоритм може є ключовою частиною розробленої програми або системи, яка керується певними принципами та методами для досягнення певної мети.

Схему алгоритму показано на рисунку 3.1.

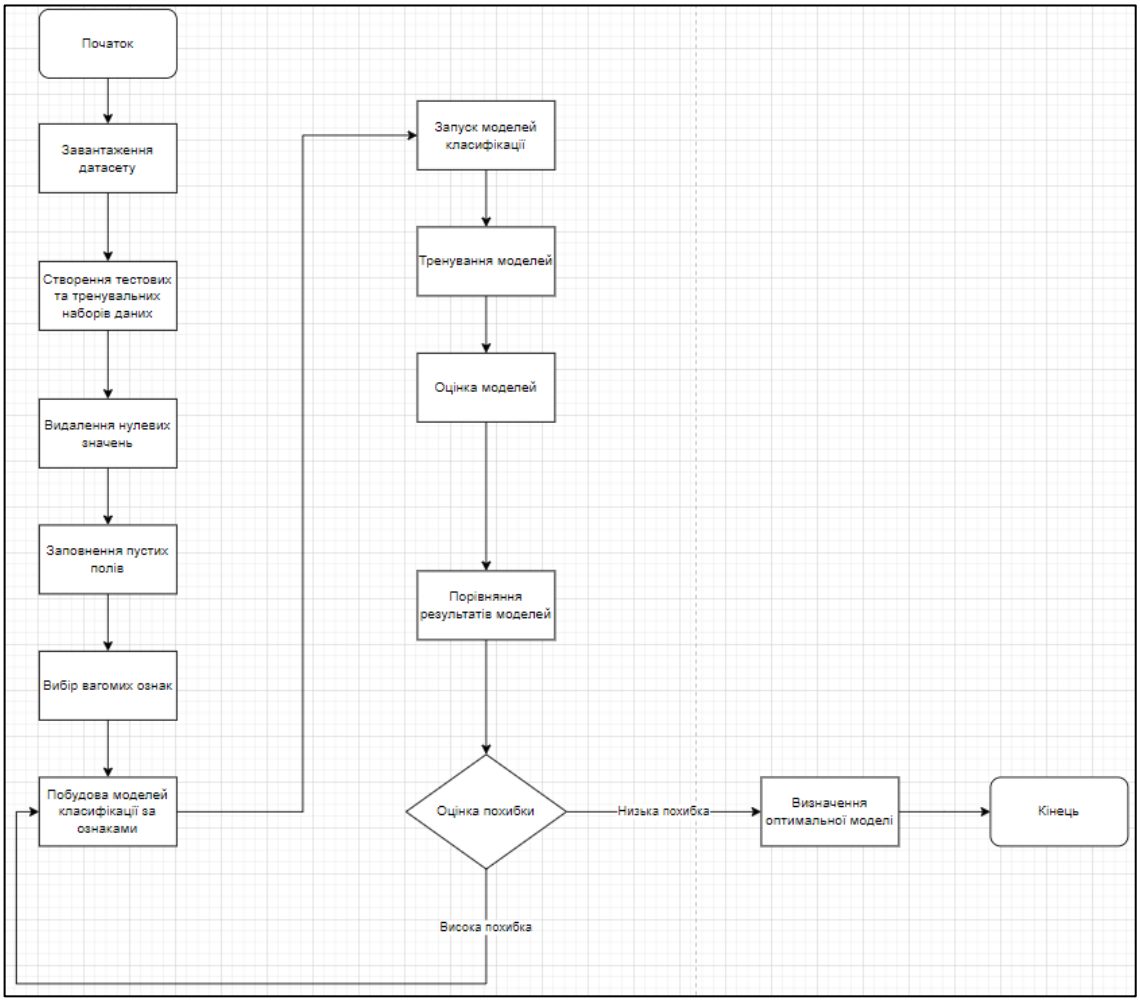


Рисунок 3.1 – Блок-схема алгоритму інформаційної технології.

Завдяки використанню блок-схеми можна ілюструвати послідовність дій, які виконує інформаційна технологія. Блок-схема надає зручний спосіб візуалізації процесу роботи програми або системи.

3.2 Побудова моделей класифікації

Першим кроком у багатьох задачах машинного навчання є розділення вихідного датасету на тренувальний та тестовий. Цей процес допомагає оцінити ефективність моделі на невидимих їй даних, що є ключовим аспектом у визначенні її загальної здатності до узагальнення.

Розділення даних показано на рисунку 3.2.

```
# train test split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.25)

print('Shape of X_train: ',X_train.shape)
print('Shape of X_test: ',X_test.shape)
print('Shape of y_train: ',y_train.shape)
print('Shape of y_test: ',y_test.shape)
```

Рисунок 3.2 – Розподіл датасету.

На рисунку 3.3 представлено вигляд тренувального датасету, який використовується для навчання моделі. Тренувальний набір даних є основною складовою частиною процесу машинного навчання, оскільки саме на цих даних модель "вчиться" виконувати своє завдання.

[43]:
```
X_train.head()
```

[43]:

	Age	Gender	Annual_Income	Total_Spend	Years_as_Customer	Num_of_Purchases	Average_Transaction_Amount	Num_of_Returns	Num_of_Support_Contacts	Satisfaction_Score	Last_Purcl
290	34	Other	66.41	5125.59	6	85	283.81	7	4	3	
122	31	Female	181.75	2127.34	10	6	181.12	5	0	1	
131	41	Female	43.82	2009.52	16	66	499.57	5	2	2	
86	29	Other	189.93	3662.12	18	21	49.15	8	1	1	
716	60	Other	114.17	7807.98	3	75	263.86	3	0	1	

Рисунок 3.3 – Тренувальний набір даних.

На рисунку 3.4 зображено код побудови моделей та налаштування гіперпараметрів для моделей класифікації. Цей етап є критично важливим у процесі машинного навчання, оскільки від правильного вибору моделі та налаштування її параметрів залежить успішність розв'язання задачі.

У відображеному коді можна побачити використання різних моделей класифікації, таких як Random Forest Classifier, Support Vector Machine (SVM), K-Nearest Neighbors (KNN) та Decision Tree.

```
# Define classifiers with hyperparameters
classifiers = {
    "Decision_Tree": DecisionTreeClassifier(criterion='entropy', max_depth=10, min_samples_split=
2),
    "Random_Forest": RandomForestClassifier(n_estimators=200, criterion='entropy', max_depth=15, m
in_samples_split=2),
    "SVM": SVC(kernel='rbf', C=1.0, gamma='scale', class_weight=None, degree=3),
    "KNN": KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30),
}

n_rows = 2
n_cols = 3

fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 10))
axes = axes.flatten()

for i, (name, clf) in enumerate(classifiers.items()):
    clf.fit(X_train_scaled, y_train)
    y_pred = clf.predict(X_test_scaled)
    accuracy = accuracy_score(y_test, y_pred)
    print(f"{name} Accuracy: {accuracy}")
    cm = confusion_matrix(y_test, y_pred)

    sns.heatmap(cm, annot=True, fmt='d', cmap='coolwarm', linecolor="orange", cbar=False, ax=axes
[i])
```

Рисунок 3.4 – Набір гіперпараметрів для моделей класифікації.

На рисунку 3.5 представлено результат виконання моделей класифікації з визначенням їхньої точності.

```
plt.tight_layout()
plt.show()

Decision_Tree Accuracy: 0.585
Random_Forest Accuracy: 0.52
SVM Accuracy: 0.495
KNN Accuracy: 0.525
```

Рисунок 3.5 – Результат виконання моделей.

У наведених результатів можна побачити значення точності для кожної з використаних моделей:

- Decision Tree Accuracy: 0.585

- Random Forest Accuracy: 0.52
- SVM Accuracy: 0.495
- KNN Accuracy: 0.525

Аналізуючи ці значення, можна зробити деякі висновки щодо ефективності моделей в даному контексті. Наприклад, модель дерева рішень показала найвищу точність серед усіх використаних моделей, досягнувши значення 0.585. Це може свідчити про те, що в даній задачі модель дерева рішень краще впоралася з класифікацією, ніж інші алгоритми.

Оскільки такий результат є незадовільним, було прийнято рішення провести додаткову роботу з датасетом. Першим кроком, було прийнято рішення використати алгоритм One-hot encoding для категоріальних змінних.

Категоріальна змінна "Gender" (стать) містить три можливі значення: Female (жінка), Male (чоловік) та Other (інше). Однак, для подальшого аналізу у машинному навчанні, де модель вимагає числові дані, категоріальні значення потрібно перетворити у числовий формат.

One-hot encoding перетворює кожне категоріальне значення у вектор з нулями та одним одиницею, де кожна позиція вектора відповідає одному можливому значенню категорії. У цьому випадку, використовуються три нові ознаки: "Female", "Male" та "Other", які приймають значення 1 або 0 залежно від того, яке значення було початково. Так само було зроблено для змінної Promotion_Response (рис. 3.6).

```
]:
# One-hot encoding для 'Gender'
gender_encoder = OneHotEncoder()
gender_encoded = gender_encoder.fit_transform(data['Gender'].values.reshape(-1, 1)).toarray()
gender_encoded_data = pd.DataFrame(gender_encoded, columns=['Female', 'Male', 'Other'])
data = pd.concat([data, gender_encoded_data], axis=1)

# One-hot encoding для 'Promotion_Response'
promo_response_encoder = OneHotEncoder()
promo_response_encoded = promo_response_encoder.fit_transform(data['Promotion_Response'].values.reshape(-1, 1)).toarray()
promo_response_encoded_data = pd.DataFrame(promo_response_encoded, columns=['Ignored', 'Responded', 'Unsubscribed'])
data = pd.concat([data, promo_response_encoded_data], axis=1)

# Label Encoding для 'Target_Churn'
label_encoder = LabelEncoder()
data['Target_Churn'] = label_encoder.fit_transform(data['Target_Churn'])
```

Рисунок 3.6 – Використання алгоритму One-hot encoding

На рисунку 3.7 показано процес створення нових ознак на основі вже існуючих у наборі даних. Цей етап відіграє важливу роль у підвищенні ефективності моделі шляхом введення нової інформації чи покращення існуючих ознак.

```
# Створення нових ознак
data['Avg_Spend_Per_Month'] = data['Total_Spend'] / (data['Years_as_Customer'] * 12)
data['Income_Spend_Interaction'] = data['Annual_Income'] * data['Total_Spend']

# Розділення даних на ознаки та цільову змінну
X = data.drop('Target_Churn', axis=1)
y = data['Target_Churn']
```

Рисунок 3.7 – Створення нових ознак на основі існуючих.

На рисунку 3.8 показано перевірку дисбалансу класів у датасеті. Дисбаланс класів - це ситуація, коли кількість прикладів одного класу набагато перевищує кількість прикладів іншого класу в наборі даних. Це може виникнути, наприклад, у задачах, де один клас є значно більш поширеним або менш представленим у вихідних даних.

Перевірка дисбалансу класів є важливим кроком у процесі підготовки даних, оскільки нерівномірний розподіл класів може вплинути на навчання моделі та якість її прогнозів. Наприклад, модель, навчена на даних з великим дисбалансом класів, може бути схильна до перекосу у вподобання більш представленого класу, і, таким чином, може показувати погані результати для менш представленого класу.

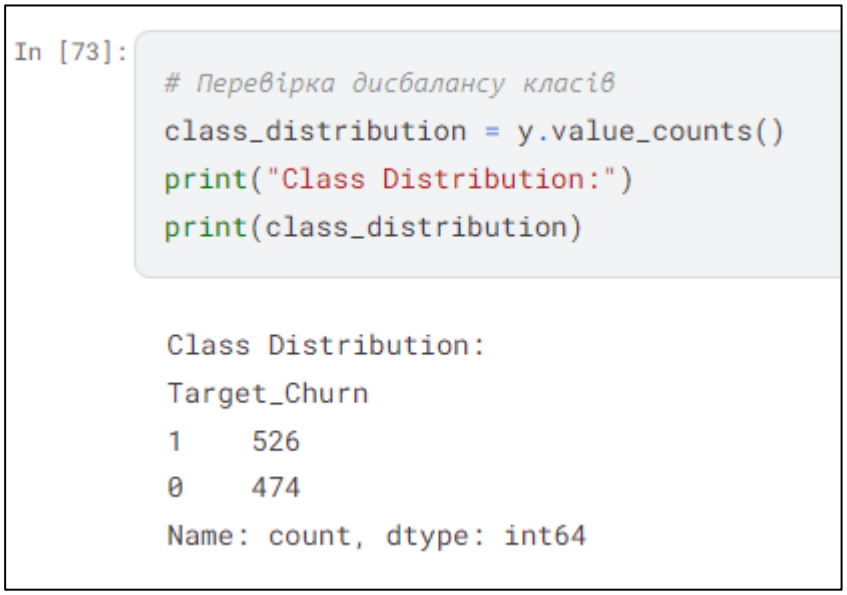


Рисунок 3.8 – Перевірка балансування класів.

Звертаючись до зображення 3.9, можна помітити, що на ньому відображено ефективність застосування алгоритму SMOTE (Synthetic Minority Over-sampling Technique) для вирішення дисбалансу класів у наборі даних.

In [74]:

```
from imblearn.over_sampling import SMOTE

# Ініціалізація SMOTE
smote = SMOTE(random_state=8)
#smote = SMOTE(random_state=35)
#smote = SMOTE(random_state=75)

# Застосування SMOTE до даних
X_resampled, y_resampled = smote.fit_resample(X, y)

# Перевірка розподілу класів після застосування SMOTE
class_distribution_resampled = y_resampled.value_counts()
print("Class Distribution after SMOTE:")
print(class_distribution_resampled)
```

```
Class Distribution after SMOTE:
Target_Churn
1    526
0    526
Name: count, dtype: int64
```

Рисунок 3.9 – Використання алгоритму SMOTE

Додавши аналіз результатів методу f_classif, можемо побачити, що після визначення найважливіших ознак для класифікації, модель стала ще ефективнішою у відрізненні одного класу від іншого (рис 3.10).

Метод f_classif використовується для визначення статистично значущих ознак, що найбільше впливають на змінну відгуку в класифікаційних задачах. Результати цього аналізу можуть надати важливу інформацію для вибору найбільш релевантних ознак та покращення якості моделі.

```python
# Визначення найважливіших ознак за допомогою методу f_classif
k_best = SelectKBest(score_func=f_classif, k=10)  # Вибрати 10 найкращих ознак
X_selected = k_best.fit_transform(X_resampled, y_resampled)

# Відбір найважливіших ознак
selected_features = k_best.get_support(indices=True)
selected_feature_names = [X.columns[i] for i in selected_features]
print("Selected features:", selected_feature_names)

# Розділення даних на тренувальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X_selected, y_resampled, test_size=0.2, random_state=42)
```

Рисунок 3.10 – Відбір ознак.

Після визначення найважливіших ознак та їх впливу на класифікацію, наступним кроком в оптимізації моделі було підібрання гіперпараметрів. Гіперпараметри моделі визначаються перед навчанням і включають параметри, такі як швидкість навчання, кількість дерев у випадковому лісі або глибина дерева в дереві рішень. Підібір правильних значень гіперпараметрів може суттєво покращити результат класифікації моделі та запобігти перенавчанню (рис. 3.11).

```python
# Створення моделей класифікації
classifiers = {
    "Decision_Tree": DecisionTreeClassifier(random_state=42,
                        max_depth=5,
                        min_samples_split=5,
                        min_samples_leaf=2,
                        max_features='sqrt'),
    "Random_Forest": RandomForestClassifier(random_state=42, n_estimators=300, max_depth=2,
                    min_samples_split=2, min_samples_leaf=2),

    "SVM": SVC(random_state=42,
               kernel='rbf',
               C=0.1,
               gamma='scale',
               class_weight='balanced'),

    "KNN": KNeighborsClassifier(n_neighbors=15),
}
```

Рисунок 3.11 – Параметри моделей класифікації.

На рисунку 3.12 можна побачити, як точність та надійність класифікації зросли після використання всіх описаних алгоритмів. Це може бути важливим досягненням в процесі розробки моделі, оскільки високі показники точності та надійності є ключовими для ефективного використання моделі у реальних умовах.

	Classifier	Train_Accuracy	Test_Accuracy
0	Decision_Tree	0.704073	0.646327
1	Random_Forest	0.758769	0.707938
2	SVM	0.635107	0.641588
3	KNN	0.733799	0.651066

Рисунок 3.12 – Результати класифікації даних.

Представлено матриці плутанини для кожної з моделей на рисунку 3.13. Ці матриці є важливим інструментом для оцінки результатів класифікації та визначення ефективності моделі в розпізнаванні класів.

Матриця плутанини відображає кількість правильно та неправильно класифікованих об'єктів для кожного класу. Вона дозволяє зрозуміти, наскільки часто модель робить помилки та які саме класи найчастіше плутається.

Аналіз матриці плутанини допомагає виявити сильні та слабкі сторони моделі. Наприклад, якщо модель часто плутає один клас з іншим, це може вказувати на потребу у покращенні якості даних або на потребу у додатковому навчанні моделі на таких прикладах.

На рисунку 3.13 можна бачити, як кожна з моделей справляється з класифікацією кожного з класів. Це дозволяє зробити висновки про загальну ефективність моделі та зробити відповідні висновки щодо подальших кроків для вдосконалення класифікаційної системи.

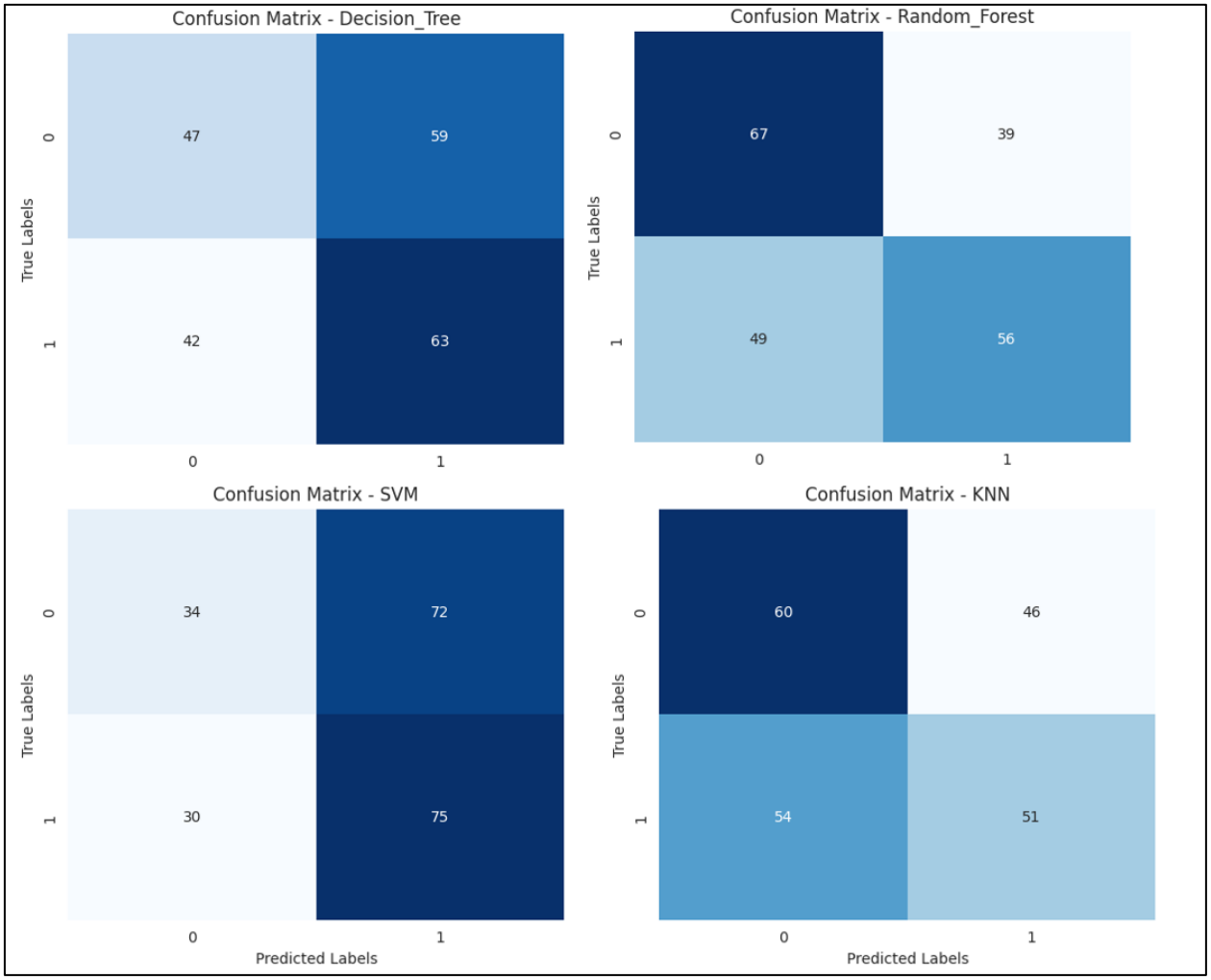


Рисунок 3.13 – Матриці плутанини.

Модель Random Forest показала найвищий результат - 0.707 на тестових даних, що свідчить про її високу ефективність в класифікації. Цей результат відображає успішне використання ансамблю дерев рішень для розв'язання задачі класифікації та підтверджує його переваги у порівнянні з іншими моделями.

На рисунку 3.14 показано діаграму важливості ознак для моделі навчання що показала найкращий результат.

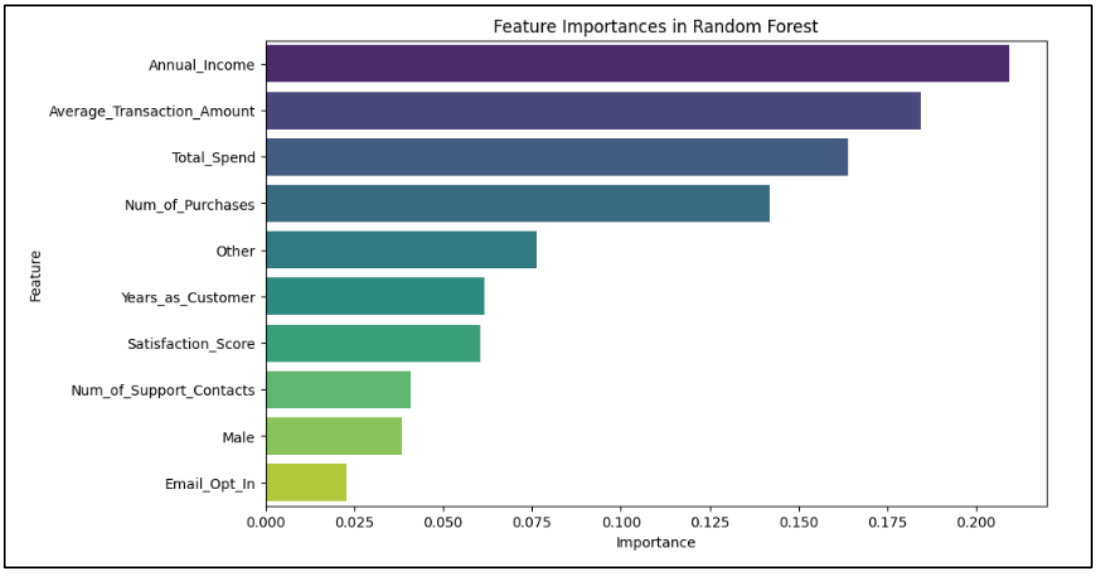


Рисунок 3.14 – Діаграма важливості ознак для моделі Random Forest.

Показана діаграма важливості ознак дозволяє побачити що найбільш вагомими в роботі моделі були такі показники: Annual_Income, Average_Stransaction_Amount та Total_Spend.

3.3 Висновки

У цьому розділі було проведено побудову та оптимізацію моделей з використанням різних алгоритмів обробки даних, таких як SMOTE, а також визначено найкращі ознаки для підвищення точності класифікації. Остаточна модель Random Forest показала точність на рівні 0,707, що свідчить про високий рівень результату та готовність використовувати побудовані моделі для подальших досліджень, зокрема у сфері аналізу відтоку клієнтів з інтернет-магазину.

ВИСНОВКИ

Під час виконання бакалаврської роботи було проведено глибоке дослідження теми, що стосується аналізу відтоку клієнтів інтернет-магазинів. Ця проблема має велике значення в сучасному інформаційному середовищі, де конкуренція на ринку дуже висока, а збереження клієнтів стає ключовим фактором успіху для будь-якого бізнесу.

У першому розділі було встановлено об’єкт дослідження та обґрунтовано актуальність проблеми відтоку клієнтів. Важливим кроком було вибір оптимальних інформаційних технологій для вирішення поставленої задачі, таких як Python та платформа Kaggle. Також було обрано набір моделей машинного навчання, включаючи Support Vector Machines, Random Forest, KNN та Decision Tree, які відповідали потребам дослідження.

Другий розділ підкреслює значення попередньої підготовки дослідницького середовища та аналізу даних. Шляхом ретельного вивчення даних, були виявлені ключові фактори, що впливають на відтік клієнтів, та визначені групи клієнтів, які мають найбільшу схильність до відтоку.

У третьому розділі було побудовано та оптимізовано моделі класифікації, які дозволили досягти рівня точності 0.707 для моделі Random Forest. Це підтверджує ефективність досліджень і надає основу для подальшого наукового розвитку в галузі аналізу відтоку клієнтів інтернет-магазинів.

У цілому, робота успішно вирішила поставлену задачу та створила основу для подальшого розвитку у галузі аналізу відтоку клієнтів інтернет-магазинів. Результати досліджень відкривають нові можливості для розуміння та прогнозування поведінки клієнтів, що допоможе бізнесу ефективніше управляти відносинами з клієнтами та збільшити їхню лояльність.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дубовик Т. Концептуальна модель довіри споживачів до інтернет-магазинів. Вісник Київського національного університету імені Тараса Шевченка. Економіка. 2014. Вип. 7 (160). С. 33–37.

2. Цуненко С. М. Особливості залучення клієнтів в Інтернет-магазин : thesis. 2015. URL: http://essuir.sumdu.edu.ua/handle/123456789/43802 (дата звернення: 18.03.2024).

3. Ліман В., Шевчук О., Коляденко С. Інтернет-магазин як етап розвитку продаж закладу традиційної форми торгівлі. Наука і техніка сьогодні. 2023. № 1(15). URL: https://doi.org/10.52058/2786-6025-2023-1(15)-62-71 (дата звернення: 19.03.2024).

4. Шаркаді М. М., Роботишин М. В., Маляр М. М. Моделі і методи машинного навчання для завдань передбачення. Науковий вісник Ужгородського університету. Серія: Математика і інформатика. № 1(36). С. 112–122. 2020.

5. Роботишин М. В., Маляр М. М. Моделі і методи машинного навчання для завдань передбачення. Науковий вісник Ужгородського університету. Серія: Математика і інформатика. № 1(36). С. 112–122. 2020.

6. Ayyadevara V. K. Random Forest. *Pro Machine Learning Algorithms*. Berkeley, CA, 2018. P. 105–116. URL: https://doi.org/10.1007/978-1-4842-3564-5_5 (date of access: 25.03.2024).

7. Keith M. Random Forest. Machine Learning with Regression in Python. Berkeley, CA, 2020. URL: https://doi.org/10.1007/978-1-4842-6583-3_5 (date of access: 03.04.2024).

8. Kramer O. K-Nearest Neighbors. Dimensionality Reduction with Unsupervised Nearest Neighbors. Berlin, Heidelberg, 2013. P. 13–23. URL: https://doi.org/10.1007/978-3-642-38652-7_2 (date of access: 11.04.2024).

9. K-Nearest Neighbors / C. El Morr et al. *International Series in Operations Research & Management Science*. Cham, 2022. P. 301–318. URL: https://doi.org/10.1007/978-3-031-16990-8_10 (date of access: 12.04.2024).

10. Computational Aspects. Support Vector Machines. New York, NY. P. 408–451. URL: https://doi.org/10.1007/978-0-387-77242-4_11 (date of access: 18.06.2024).

11. Kernels and Support Vector Machines. Support Vector Machines. 2012. P. 109–154. URL: https://doi.org/10.1201/b14297-10 (date of access: 15.04.2024).

12. Li H. Decision Tree. *Machine Learning Methods*. Singapore, 2023. P. 77–102. URL: https://doi.org/10.1007/978-981-99-3917-6_5 (date of access: 16.04.2024).

13. Decision Tree / Z. Wang et al. *Numerical Machine Learning*. 2023. P. 97–115. URL: https://doi.org/10.2174/9789815136982123010006 (date of access: 20.04.2024).

14. Bojer C. S., Meldgaard J. P. Kaggle forecasting competitions: An overlooked learning opportunity. International Journal of Forecasting. 2020. URL: https://doi.org/10.1016/j.ijforecast.2020.07.007 (date of access: 21.04.2024).

15. Al-Taie M. Z., Salim N., Obasa A. I. Successful Data Science Projects: Lessons Learned from Kaggle Competition. *Kurdistan Journal of Applied Research*. 2017. Vol. 2, no. 3. P. 40–49. URL: https://doi.org/10.24017/science.2017.3.18 (date of access: 23.04.2024).

16. Code J. Python Machine Learning: The Absolute Beginner's Guide for Understand Neural Network, Artificial Intelligent, Deep Learning and Mastering the Fundamentals of Ml with Python. Independently Published, 2020.

17. Eknath, Prof. Upasani Dhananjay, 1st, Shete, Prof. Virendra Virbhadra, 1st. Machine Learning with Python: Machine Learning with Python. INSC International Publisher (IIP), 2021.

18. Manipulating Tabular Data Using Pandas. Python® Machine Learning. Indianapolis, Indiana, 2019. P. 39–65. URL: https://doi.org/10.1002/9781119557500.ch3 (date of access: 24.04.2024).

19. Python. Machine Learning. 2011. P. 381–398. URL: https://doi.org/10.1201/9781420067194-20 (date of access: 26.04.2024).

20. Мокін О.Б., Мокін В.Б., і Мокін Б.І., «Алгоритм методу ідентифікації моделі авторегресії – ковзного середнього, який узагальнює методику Юла–Уокера, та його програмна python-реалізація», Вісник ВПІ, вип. 4, с. 41–55, Верес. 2022.

Додаток А

(обов’язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
“____”_____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську дипломну роботу

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ВІДТОКУ ПОКУПЦІВ В ІНТЕРНЕТ-МАГАЗИНІ

08-34.БДР.003.00.000 ТЗ

Керівник: к.т.н., доц.
_____ Олексій КОЗАЧКО
«__» _____ 2024 р.
Розробила студентка гр. 2ІСТ-20б
_____ Наталя ІЖАКОВСЬКА
«__» _____ 2024 р.

Вінниця 2024

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____ 2024р., та індивідуальне завдання на БДР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2024р.

2. Джерела розробки:

1) Дубовик Т. Концептуальна модель довіри споживачів до інтернет-магазинів. Вісник Київського національного університету імені Тараса Шевченка. Економіка. 2014. Вип. 7 (160). С. 33–37.

1) Шаркаді М. М., Роботишин М. В., Маляр М. М. Моделі і методи машинного навчання для завдань передбачення. Науковий вісник Ужгородського університету. Серія: Математика і інформатика. № 1(36). С. 112–122. 2020.

3. Мета і призначення роботи.

Метою дослідження є підвищення точності передбачення відтоку клієнтів з інтернет магазину за допомогою методів машинного навчання шляхом розроблення інформаційної технології.

4. Вихідні дані для проведення робіт:

Датасет Kaggle «Online Retail Customer Churn Dataset» з даними для передбачення відтоку клієнтів з інтернет-магазину.

5. Методи дослідження:
   - Підготовка даних;
   - Розвідувальний аналіз;
   - Моделі машинного навчання для передбачення даних;

6. Етапи роботи і терміни їх виконання:

a) Аналіз предметної області _____ – _____
b) Підготовка даних та розвідувальний аналіз _____ – _____
c) Розроблення інформаційної технології для аналізу даних _____ – _____
d) Оформлення матеріалів до захисту БДР _____ – _____

7. Очікувані результати та порядок реалізації

Розроблення інформаційної технології передбачення відтоку клієнтів з інтернет-магазину.

8. Вимоги до розробленої документації

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)».

9. Порядок приймання роботи

Публічний захист «___» _____ 2024 р.
Початок розробки «___» _____ 2024 р.
Граничні терміни виконання БДР «___» _____ 2024 р.
Розробила студентка групи 2ІСТ-20б _____ Наталя ІЖАКОВСЬКА

Додаток Б

(обов'язковий)

Протокол перевірки бакалаврської дипломної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія передбачення відтоку покупців в інтернет-магазині»
Тип роботи: бакалаврська дипломна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 94,39 % Схожість 5,61 %

Аналіз звіту подібності (відмітити потрібне)

• Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
○ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
○ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ (підпис) Сергій ЖУКОВ

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Наталя ІЖАКОВСЬКА

Керівник роботи _____ (підпис) Олексій КОЗАЧКО

Додаток В

(довідниковий)

Фрагмент лістингу програми

```
# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:01.734429Z","iopu
b.execute_input":"2024-06-17T21:49:01.734874Z","iopub.status.idle":"2024-06-17T21
:49:01.746638Z","shell.execute_reply.started":"2024-06-17T21:49:01.734842Z","shel
l.execute_reply":"2024-06-17T21:49:01.745232Z"}}
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/dock
er-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list a
ll files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets
preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved ou
tside of the current session

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:01.750569Z","iopu
b.execute_input":"2024-06-17T21:49:01.751459Z","iopub.status.idle":"2024-06-17T21
:49:01.762424Z","shell.execute_reply.started":"2024-06-17T21:49:01.751388Z","shel
l.execute_reply":"2024-06-17T21:49:01.760423Z"}}
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')


# sklearn imports
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
```

```
from sklearn.metrics import accuracy_score, precision_score, f1_score, roc_auc_sc
ore, recall_score

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:01.764337Z","iopu
b.execute_input":"2024-06-17T21:49:01.765059Z","iopub.status.idle":"2024-06-17T21
:49:01.809987Z","shell.execute_reply.started":"2024-06-17T21:49:01.764998Z","shel
l.execute_reply":"2024-06-17T21:49:01.808478Z"}}
data = pd.read_csv('/kaggle/input/online-retail-customer-churn-dataset/online_ret
ail_customer_churn.csv')
data.head()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:01.811909Z","iopu
b.execute_input":"2024-06-17T21:49:01.812413Z","iopub.status.idle":"2024-06-17T21
:49:01.821043Z","shell.execute_reply.started":"2024-06-17T21:49:01.812369Z","shel
l.execute_reply":"2024-06-17T21:49:01.819554Z"}}
data.shape

# %% [markdown]
# ## 1. Summary about dataset
# - We have total 1000 customer records and total 15 features or columns in datas
et.
#

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:01.826813Z","iopu
b.execute_input":"2024-06-17T21:49:01.827295Z","iopub.status.idle":"2024-06-17T21
:49:01.840195Z","shell.execute_reply.started":"2024-06-17T21:49:01.827260Z","shel
l.execute_reply":"2024-06-17T21:49:01.838561Z"}}
# checks column names
data.columns

# %% [markdown]
# ## 2. Columns Description:-
# 1. `Customer_ID`:- This column contains unique identifier for each customer.
# 2. `Age`:- This column contains age of the customer.
# 3. `Gender`:- This column contains gender of the customer.  (Male, Female, othe
r)
# 4. `Annual_Income`:- This column contains total income of the customer in a yea
r.
# 5. `Total_Spend`:- This column contains total amount spent by customer on a pro
duct or service.
# 6. `Years_as_Customer`:- Time duration of customer has been associate with prod
uct or services
# 7. `Num_of_Purchases`:- This column contains total number of products or servic
es purchased by the customer.
# 8. `Average_Transaction_Amount`:- The average amount spent by a customer on eac
h transaction.
# 9. `Num_of_Returns`:- This column contains the number of times customer returne
d the product or services.
# 10. `Num_of_Support_Contacts`:- This column contains records of how many times
customer contacted to support services.
# 11. `Satisfaction_Score`:- A numerical representation of the customer's satisfa
ction level. (ratings by customers)
# 12. `Last_Purchase_Days_Ago`:- number of days before customer's last purchase.
# 13. `Email_Opt_In`:- An indicator (True or False) representing whether the cust
omer has opted in to receive promotional emails.
# 14. `Promotion_Response`:- An indicator representing whether the customer has r
esponded to mails. (Responded, Ignore, Unsubscribed etc.)
```

```
# 15. `Target_Churn`:- The target variable indicating whether the customer has ch
urned (1) or not (0), where churn refers to customers who have stopped using the
products or services.

# %% [markdown]
# ## 3. Mannual analysis
#

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:01.841972Z","iopu
b.execute_input":"2024-06-17T21:49:01.842379Z","iopub.status.idle":"2024-06-17T21
:49:02.249742Z","shell.execute_reply.started":"2024-06-17T21:49:01.842336Z","shel
l.execute_reply":"2024-06-17T21:49:02.248426Z"}}
# store our data as in excel file and look it manually in google sheets.
data.to_excel('data.xlsx')

# %% [markdown]
# ## 4. Programmatic analysis

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.251375Z","iopu
b.execute_input":"2024-06-17T21:49:02.251901Z","iopub.status.idle":"2024-06-17T21
:49:02.274148Z","shell.execute_reply.started":"2024-06-17T21:49:02.251859Z","shel
l.execute_reply":"2024-06-17T21:49:02.271653Z"}}
# first 5 records in data
data.head()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.276839Z","iopu
b.execute_input":"2024-06-17T21:49:02.277404Z","iopub.status.idle":"2024-06-17T21
:49:02.305513Z","shell.execute_reply.started":"2024-06-17T21:49:02.277360Z","shel
l.execute_reply":"2024-06-17T21:49:02.304144Z"}}
# random 5 records in data
data.sample(5)

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.307078Z","iopu
b.execute_input":"2024-06-17T21:49:02.307733Z","iopub.status.idle":"2024-06-17T21
:49:02.321307Z","shell.execute_reply.started":"2024-06-17T21:49:02.307647Z","shel
l.execute_reply":"2024-06-17T21:49:02.319834Z"}}
# shape of data
data.shape

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.323274Z","iopu
b.execute_input":"2024-06-17T21:49:02.323634Z","iopub.status.idle":"2024-06-17T21
:49:02.352745Z","shell.execute_reply.started":"2024-06-17T21:49:02.323606Z","shel
l.execute_reply":"2024-06-17T21:49:02.350643Z"}}
# last 5 rows in data
data.tail()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.354228Z","iopu
b.execute_input":"2024-06-17T21:49:02.354686Z","iopub.status.idle":"2024-06-17T21
:49:02.375833Z","shell.execute_reply.started":"2024-06-17T21:49:02.354632Z","shel
l.execute_reply":"2024-06-17T21:49:02.372765Z"}}
data.info()

# %% [markdown]
# #### Dataset summary
# 1. There are 1000 records and 15 columns in dataset.
#   - 2 columns have boolen data
#   - 3 columns have float type data.
#   - 8 columns have integer type data.
```

```
#   - 2 columns have string type data.

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.378294Z","iopu
b.execute_input":"2024-06-17T21:49:02.378989Z","iopub.status.idle":"2024-06-17T21
:49:02.392787Z","shell.execute_reply.started":"2024-06-17T21:49:02.378935Z","shel
l.execute_reply":"2024-06-17T21:49:02.391448Z"}}
# we don't need 'Customer_id' column, so we drop it for further analysis
data.drop(columns='Customer_ID',axis=1, inplace=True)
data.columns

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.394623Z","iopu
b.execute_input":"2024-06-17T21:49:02.395127Z","iopub.status.idle":"2024-06-17T21
:49:02.409772Z","shell.execute_reply.started":"2024-06-17T21:49:02.395093Z","shel
l.execute_reply":"2024-06-17T21:49:02.407760Z"}}
# split numerical and categorical columns
numerical_cols = data.select_dtypes(include=['int','float']).columns
numerical_cols

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.416092Z","iopu
b.execute_input":"2024-06-17T21:49:02.416703Z","iopub.status.idle":"2024-06-17T21
:49:02.427719Z","shell.execute_reply.started":"2024-06-17T21:49:02.416633Z","shel
l.execute_reply":"2024-06-17T21:49:02.425920Z"}}
# categorical colums
categorical_cols = data.select_dtypes(exclude=['int','float']).columns
categorical_cols

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.429412Z","iopu
b.execute_input":"2024-06-17T21:49:02.430271Z","iopub.status.idle":"2024-06-17T21
:49:02.449602Z","shell.execute_reply.started":"2024-06-17T21:49:02.430190Z","shel
l.execute_reply":"2024-06-17T21:49:02.448057Z"}}
# value counts for each categorical column
for i in categorical_cols:
    print(f"**** '{i}' unique values ****")
    print(data[i].value_counts())
    print()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.450930Z","iopu
b.execute_input":"2024-06-17T21:49:02.451686Z","iopub.status.idle":"2024-06-17T21
:49:02.472498Z","shell.execute_reply.started":"2024-06-17T21:49:02.451622Z","shel
l.execute_reply":"2024-06-17T21:49:02.471184Z"}}
# Let check for null values
data.isnull().sum()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.474795Z","iopu
b.execute_input":"2024-06-17T21:49:02.475328Z","iopub.status.idle":"2024-06-17T21
:49:02.489695Z","shell.execute_reply.started":"2024-06-17T21:49:02.475293Z","shel
l.execute_reply":"2024-06-17T21:49:02.488501Z"}}
# check for duplicated values
data.duplicated().sum()

# %% [markdown]
# ### Conclusion based above analysis
# - There is no any null value in data.
# - There are no duplicated records in data.

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.491167Z","iopu
b.execute_input":"2024-06-17T21:49:02.491533Z","iopub.status.idle":"2024-06-17T21
```

```
:49:02.536549Z","shell.execute_reply.started":"2024-06-17T21:49:02.491503Z","shel
l.execute_reply":"2024-06-17T21:49:02.534693Z"}}
# mathematical summary  for numerical columns
data.describe()


# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.538147Z","iopu
b.execute_input":"2024-06-17T21:49:02.538623Z","iopub.status.idle":"2024-06-17T21
:49:02.561496Z","shell.execute_reply.started":"2024-06-17T21:49:02.538582Z","shel
l.execute_reply":"2024-06-17T21:49:02.560110Z"}}
# for categorical columns
data.describe(include=['object', 'bool'])

# %% [markdown]
# ## 3. Exploratory Data Analysis
# ### 3.1. UNivariate Analysis
#

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.563060Z","iopu
b.execute_input":"2024-06-17T21:49:02.563418Z","iopub.status.idle":"2024-06-17T21
:49:02.578160Z","shell.execute_reply.started":"2024-06-17T21:49:02.563388Z","shel
l.execute_reply":"2024-06-17T21:49:02.576685Z"}}
numerical_cols

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.580072Z","iopu
b.execute_input":"2024-06-17T21:49:02.580595Z","iopub.status.idle":"2024-06-17T21
:49:02.601565Z","shell.execute_reply.started":"2024-06-17T21:49:02.580553Z","shel
l.execute_reply":"2024-06-17T21:49:02.600313Z"}}
# check for skweness
skewness_result = {}
for col in numerical_cols:
    skew_value = data[col].skew()
    if skew_value > 0.5 :
        skewness_type = 'Right Skewed'

    elif skew_value < (-0.5):
        skewness_type = 'Left Skewed'
    elif skew_value==0 :
        skewness_type='Normal'
    else:
        skewness_type='Approximately Normal'

    skewness_result[col] = [skew_value, skewness_type]

skewness_df = pd.DataFrame(skewness_result, index=['Skew_Value','Skewness_Type'])
skewness_df.T

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.603077Z","iopu
b.execute_input":"2024-06-17T21:49:02.603725Z","iopub.status.idle":"2024-06-17T21
:49:02.613309Z","shell.execute_reply.started":"2024-06-17T21:49:02.603688Z","shel
l.execute_reply":"2024-06-17T21:49:02.612006Z"}}
# create a fuction , which perform univariate analysis on numerical columns
def univariate_analysis_numeric(col):
    # Create subplots
    fig, ax = plt.subplots(1, 2, figsize=(10, 6))

    # Histogram and kde
    sns.histplot(x=data[col], bins=30, stat='frequency',kde=True, ax=ax[0])
```

```
    ax[0].set_title(f'Distribution of {col}')

    # Boxplot
    sns.boxplot(x=data[col], ax=ax[1])
    ax[1].set_title(f'Boxplot for {col}')

    plt.tight_layout()
    plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.615389Z","iopu
b.execute_input":"2024-06-17T21:49:02.615764Z","iopub.status.idle":"2024-06-17T21
:49:02.629952Z","shell.execute_reply.started":"2024-06-17T21:49:02.615735Z","shel
l.execute_reply":"2024-06-17T21:49:02.628514Z"}}
numerical_cols

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:02.631621Z","iopu
b.execute_input":"2024-06-17T21:49:02.632116Z","iopub.status.idle":"2024-06-17T21
:49:03.508317Z","shell.execute_reply.started":"2024-06-17T21:49:02.632073Z","shel
l.execute_reply":"2024-06-17T21:49:03.506917Z"}}
# Univariate analysis for Age columns
univariate_analysis_numeric('Age')

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:03.509794Z","iopu
b.execute_input":"2024-06-17T21:49:03.510148Z","iopub.status.idle":"2024-06-17T21
:49:12.150014Z","shell.execute_reply.started":"2024-06-17T21:49:03.510111Z","shel
l.execute_reply":"2024-06-17T21:49:12.148711Z"}}
# univatriate analysis for all numerical columns
for i in numerical_cols:
    print(f'== {i} ==')
    univariate_analysis_numeric(i)
    print('=='*40)

# %% [markdown]
# #### Conclusion based on univariate analysis
# - Almost all numerical columns data is symmetric means normal distributed.
# - There is no any outlier in numerical columns.

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:12.151526Z","iopu
b.execute_input":"2024-06-17T21:49:12.151985Z","iopub.status.idle":"2024-06-17T21
:49:12.160130Z","shell.execute_reply.started":"2024-06-17T21:49:12.151945Z","shel
l.execute_reply":"2024-06-17T21:49:12.158764Z"}}
# univariate analysis for categorical columns
def univariate_analysis_categorical(col):
    # Create subplots
    fig, ax = plt.subplots(1, 2, figsize=(12, 6))

    # Count plot
    sns.countplot(x=data[col], ax=ax[0])
    ax[0].set_title(f'Count Plot for {col}')
    ax[0].tick_params(axis='x', rotation=45)

    # Pie plot
    value_counts = data[col].value_counts()
    ax[1].pie(value_counts, labels=value_counts.index, autopct='%1.1f%%', startan
gle=90)
    ax[1].set_title(f'Pie Plot for {col}')

    plt.tight_layout()
```

```
    plt.show()


# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:12.161534Z","iopu
b.execute_input":"2024-06-17T21:49:12.161902Z","iopub.status.idle":"2024-06-17T21
:49:12.176543Z","shell.execute_reply.started":"2024-06-17T21:49:12.161872Z","shel
l.execute_reply":"2024-06-17T21:49:12.175129Z"}}
categorical_cols


# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:12.178136Z","iopu
b.execute_input":"2024-06-17T21:49:12.178715Z","iopub.status.idle":"2024-06-17T21
:49:12.580409Z","shell.execute_reply.started":"2024-06-17T21:49:12.178675Z","shel
l.execute_reply":"2024-06-17T21:49:12.579230Z"}}
univariate_analysis_categorical('Gender')

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:12.581966Z","iopu
b.execute_input":"2024-06-17T21:49:12.582394Z","iopub.status.idle":"2024-06-17T21
:49:14.393575Z","shell.execute_reply.started":"2024-06-17T21:49:12.582332Z","shel
l.execute_reply":"2024-06-17T21:49:14.392119Z"}}
for i in categorical_cols:
    print(f'****- {i} -****')
    univariate_analysis_categorical(i)
    print('==='*40)

# %% [markdown]
# ## 3.2. Bivariate analysis ( each column with target column)
# #### Numerical - categorical bivariate analysis

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:14.395300Z","iopu
b.execute_input":"2024-06-17T21:49:14.395809Z","iopub.status.idle":"2024-06-17T21
:49:14.406132Z","shell.execute_reply.started":"2024-06-17T21:49:14.395768Z","shel
l.execute_reply":"2024-06-17T21:49:14.404754Z"}}
import seaborn as sns
import matplotlib.pyplot as plt

def bivariate_analysis_num_cat(data, i):
    fig, ax = plt.subplots(1, 2, figsize=(10, 6))

    # Set plot style and colors
    sns.set_style("whitegrid")
    colors = ['blue', 'red', 'green']

    # Bar plot
    sns.barplot(data=data, x='Target_Churn', y=i, ax=ax[0], palette=colors)
    ax[0].set_title(f'Average {i} by Target Churn')
    ax[0].set_facecolor('white')

    # Box plot
    sns.boxplot(data=data, x='Target_Churn', y=i, ax=ax[1], palette=colors)
    ax[1].set_title(f'{i} distribution by Target Churn')
    ax[1].set_facecolor('white')

    # Set the facecolor of the figure
    fig.patch.set_facecolor('white')

    plt.tight_layout()
    plt.show()
```

```
# Example usage:
# bivariate_analysis_num_cat(data, 'your_numeric_column')


# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:14.407849Z","iopu
b.execute_input":"2024-06-17T21:49:14.408289Z","iopub.status.idle":"2024-06-17T21
:49:20.299752Z","shell.execute_reply.started":"2024-06-17T21:49:14.408248Z","shel
l.execute_reply":"2024-06-17T21:49:20.298622Z"}}
for i in numerical_cols:
    print(f'****- {i} -****')
    bivariate_analysis_num_cat(data, i)
    print('===='*40)

# %% [markdown]
# #### 3.3 multivariate analysis

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:20.301789Z","iopu
b.execute_input":"2024-06-17T21:49:20.302194Z","iopub.status.idle":"2024-06-17T21
:49:21.047290Z","shell.execute_reply.started":"2024-06-17T21:49:20.302162Z","shel
l.execute_reply":"2024-06-17T21:49:21.045875Z"}}
plt.figure(figsize=(10,6))
sns.heatmap(data[numerical_cols].corr(), annot=True, cmap='viridis', cbar=True)
plt.show()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.048856Z","iopu
b.execute_input":"2024-06-17T21:49:21.049289Z","iopub.status.idle":"2024-06-17T21
:49:21.071615Z","shell.execute_reply.started":"2024-06-17T21:49:21.049247Z","shel
l.execute_reply":"2024-06-17T21:49:21.070349Z"}}
data.head()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.072946Z","iopu
b.execute_input":"2024-06-17T21:49:21.073333Z","iopub.status.idle":"2024-06-17T21
:49:21.093608Z","shell.execute_reply.started":"2024-06-17T21:49:21.073301Z","shel
l.execute_reply":"2024-06-17T21:49:21.092443Z"}}
# store data in csv file for further framework work
data.to_csv('Cleaned_customer_churn_data.csv')

# %% [markdown]
# # Split data as dependent and independt data

# %% [markdown]
# ### Encoding Categorical Columns

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.095023Z","iopu
b.execute_input":"2024-06-17T21:49:21.095516Z","iopub.status.idle":"2024-06-17T21
:49:21.110683Z","shell.execute_reply.started":"2024-06-17T21:49:21.095486Z","shel
l.execute_reply":"2024-06-17T21:49:21.109500Z"}}
# Convert 'Gender' to numerical representation
df_encoded = pd.get_dummies(data, columns=['Gender'], drop_first=True)

# Convert 'Email_Opt_In' to numerical representation
df_encoded['Email_Opt_In'] = df_encoded['Email_Opt_In'].astype(int)

# Creating new feature as spend Income ratio
df_encoded['Spend_Income_Ratio'] = df_encoded['Total_Spend'] / df_encoded['Annual
_Income']
```

```
# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.112066Z","iopu
b.execute_input":"2024-06-17T21:49:21.112423Z","iopub.status.idle":"2024-06-17T21
:49:21.144320Z","shell.execute_reply.started":"2024-06-17T21:49:21.112393Z","shel
l.execute_reply":"2024-06-17T21:49:21.143199Z"}}
# Convert 'Promotion_Response' to numerical representation
df_encoded = pd.get_dummies(df_encoded, columns=['Promotion_Response'], drop_firs
t=True)
df_encoded.head()

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.145384Z","iopu
b.execute_input":"2024-06-17T21:49:21.145717Z","iopub.status.idle":"2024-06-17T21
:49:21.162227Z","shell.execute_reply.started":"2024-06-17T21:49:21.145682Z","shel
l.execute_reply":"2024-06-17T21:49:21.160800Z"}}
# One-hot encoding для 'Gender'
gender_encoder = OneHotEncoder()
gender_encoded = gender_encoder.fit_transform(data['Gender'].values.reshape(-1, 1
)).toarray()
gender_encoded_data = pd.DataFrame(gender_encoded, columns=['Female', 'Male', 'Ot
her'])
data = pd.concat([data, gender_encoded_data], axis=1)

# One-hot encoding для 'Promotion_Response'
promo_response_encoder = OneHotEncoder()
promo_response_encoded = promo_response_encoder.fit_transform(data['Promotion_Res
ponse'].values.reshape(-1, 1)).toarray()
promo_response_encoded_data = pd.DataFrame(promo_response_encoded, columns=['Igno
red', 'Responded', 'Unsubscribed'])
data = pd.concat([data, promo_response_encoded_data], axis=1)

# Label Encoding для 'Target_Churn'
label_encoder = LabelEncoder()
data['Target_Churn'] = label_encoder.fit_transform(data['Target_Churn'])


# %% [markdown]
# ## Machine Learning Models

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.164010Z","iopu
b.execute_input":"2024-06-17T21:49:21.164400Z","iopub.status.idle":"2024-06-17T21
:49:21.178258Z","shell.execute_reply.started":"2024-06-17T21:49:21.164370Z","shel
l.execute_reply":"2024-06-17T21:49:21.177080Z"}}
# Видалення оригінальних категоріальних змінних
data = data.drop(['Gender', 'Promotion_Response'], axis=1)

# Створення нових ознак
data['Avg_Spend_Per_Month'] = data['Total_Spend'] / (data['Years_as_Customer'] *
12)
data['Income_Spend_Interaction'] = data['Annual_Income'] * data['Total_Spend']

# Розділення даних на ознаки та цільову змінну
X = data.drop('Target_Churn', axis=1)
y = data['Target_Churn']


# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.179833Z","iopu
b.execute_input":"2024-06-17T21:49:21.180316Z","iopub.status.idle":"2024-06-17T21
:49:21.193309Z","shell.execute_reply.started":"2024-06-17T21:49:21.180277Z","shel
l.execute_reply":"2024-06-17T21:49:21.192182Z"}}
```

```
# Перевірка дисбалансу класів
class_distribution = y.value_counts()
print("Class Distribution:")
print(class_distribution)


# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.194514Z","iopu
b.execute_input":"2024-06-17T21:49:21.194876Z","iopub.status.idle":"2024-06-17T21
:49:21.223536Z","shell.execute_reply.started":"2024-06-17T21:49:21.194846Z","shel
l.execute_reply":"2024-06-17T21:49:21.222081Z"}}
from imblearn.over_sampling import SMOTE

# Ініціалізація SMOTE
smote = SMOTE(random_state=8)
#smote = SMOTE(random_state=35)
#smote = SMOTE(random_state=75)

# Застосування SMOTE до даних
X_resampled, y_resampled = smote.fit_resample(X, y)

# Перевірка розподілу класів після застосування SMOTE
class_distribution_resampled = y_resampled.value_counts()
print("Class Distribution after SMOTE:")
print(class_distribution_resampled)

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.229716Z","iopu
b.execute_input":"2024-06-17T21:49:21.230083Z","iopub.status.idle":"2024-06-17T21
:49:21.235829Z","shell.execute_reply.started":"2024-06-17T21:49:21.230055Z","shel
l.execute_reply":"2024-06-17T21:49:21.234608Z"}}
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.model_selection import train_test_split


# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:49:21.237247Z","iopu
b.execute_input":"2024-06-17T21:49:21.237605Z","iopub.status.idle":"2024-06-17T21
:49:21.262652Z","shell.execute_reply.started":"2024-06-17T21:49:21.237575Z","shel
l.execute_reply":"2024-06-17T21:49:21.261270Z"}}
# Визначення найважливіших ознак за допомогою методу f_classif
k_best = SelectKBest(score_func=f_classif, k=10)  # Вибрати 10 найкращих ознак
X_selected = k_best.fit_transform(X_resampled, y_resampled)

# Відбір найважливіших ознак
selected_features = k_best.get_support(indices=True)
selected_feature_names = [X.columns[i] for i in selected_features]
print("Selected features:", selected_feature_names)

# Розділення даних на тренувальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X_selected, y_resampled, test
_size=0.2, random_state=42)

# %% [raw]
#

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:51:48.199566Z","iopu
b.execute_input":"2024-06-17T21:51:48.199984Z","iopub.status.idle":"2024-06-17T21
:51:51.126027Z","shell.execute_reply.started":"2024-06-17T21:51:48.199952Z","shel
l.execute_reply":"2024-06-17T21:51:51.124904Z"}}
from sklearn.ensemble import GradientBoostingClassifier, AdaBoostClassifier
```

```
# Створення моделей класифікації
classifiers = {
    "Decision_Tree": DecisionTreeClassifier(random_state=42,
                         max_depth=5,
                         min_samples_split=5,
                         min_samples_leaf=2,
                         max_features='sqrt'),
    "Random_Forest": RandomForestClassifier(random_state=42, n_estimators=300, ma
x_depth=2,
                     min_samples_split=2, min_samples_leaf=2),

    "SVM": SVC(random_state=42,
                kernel='rbf',
                C=0.1,
                gamma='scale',
                class_weight='balanced'),

    "KNN": KNeighborsClassifier(n_neighbors=15),
}
# Візуалізація результатів
n_rows = 2
n_cols = 3
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 10))
axes = axes.flatten()

# Define an empty dictionary to store results
train_results = {'Classifier': [], 'Train_Accuracy': [], 'Test_Accuracy': []}

for i, (name, clf) in enumerate(classifiers.items()):
    clf.fit(X_train, y_train)
    y_train_pred = clf.predict(X_train)
    train_accuracy = accuracy_score(y_train, y_train_pred)

    y_test_pred = clf.predict(X_test)
    test_accuracy = accuracy_score(y_test, y_test_pred)
    cm = confusion_matrix(y_test, y_test_pred)

    sns.heatmap(cm, annot=True, fmt='d', cmap='coolwarm', linecolor="orange", cba
r=False, ax=axes[i])
    axes[i].set_title(f'Confusion Matrix - {name}')
    axes[i].set_xlabel('Predicted Labels')
    axes[i].set_ylabel('True Labels')

    # Add results to the dictionary
    train_results['Classifier'].append(name)
    train_results['Train_Accuracy'].append(train_accuracy)
    train_results['Test_Accuracy'].append(test_accuracy)

plt.tight_layout()
plt.show()

# Create a DataFrame from the results dictionary
results_data = pd.DataFrame(train_results)

print(results_data)
```

```
# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:51:51.273546Z","iopu
b.execute_input":"2024-06-17T21:51:51.273970Z","iopub.status.idle":"2024-06-17T21
:51:51.282019Z","shell.execute_reply.started":"2024-06-17T21:51:51.273938Z","shel
l.execute_reply":"2024-06-17T21:51:51.280473Z"}}
selected_features

# %% [code] {"execution":{"iopub.status.busy":"2024-06-17T21:51:51.694175Z","iopu
b.execute_input":"2024-06-17T21:51:51.694592Z","iopub.status.idle":"2024-06-17T21
:51:52.149184Z","shell.execute_reply.started":"2024-06-17T21:51:51.694560Z","shel
l.execute_reply":"2024-06-17T21:51:52.147877Z"}}
# Побудова діаграми важливості ознак для моделі Random Forest
random_forest_clf = classifiers["Random_Forest"]
feature_importances = random_forest_clf.feature_importances_
features = selected_feature_names

# Створення DataFrame з важливостями ознак
importance_df = pd.DataFrame({'Feature': features, 'Importance': feature_importan
ces})
importance_df = importance_df.sort_values(by='Importance', ascending=False)

# Візуалізація важливості ознак
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=importance_df, palette='viridis')
plt.title('Feature Importances in Random Forest')
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.show()
```

Додаток Г

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕДБАЧЕННЯ ВІДТОКУ ПОКУПЦІВ В ІНТЕРНЕТ-МАГАЗИНІ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2024 р.

Вінниця 2024

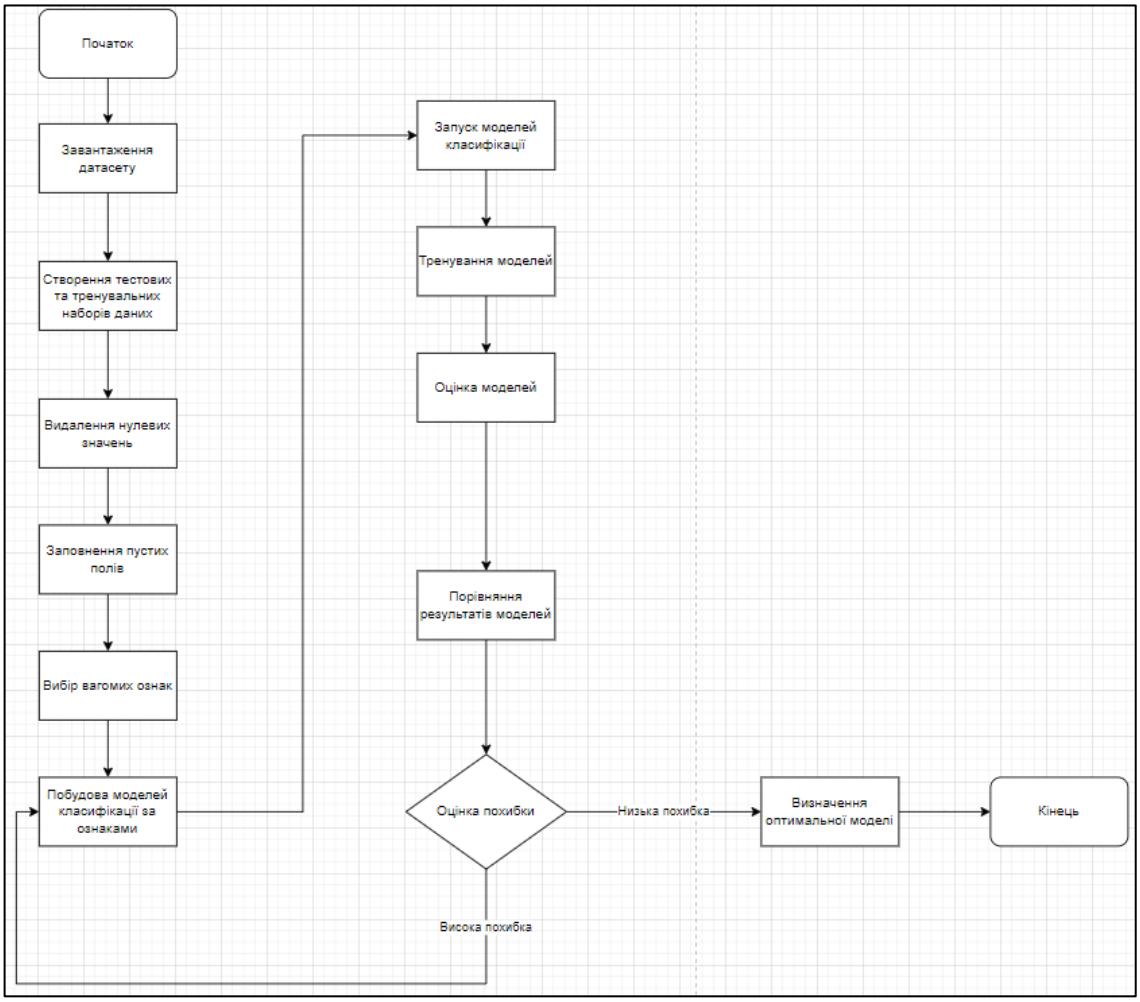


Рисунок Г.1 – Блок-схема алгоритму інформаційної технології

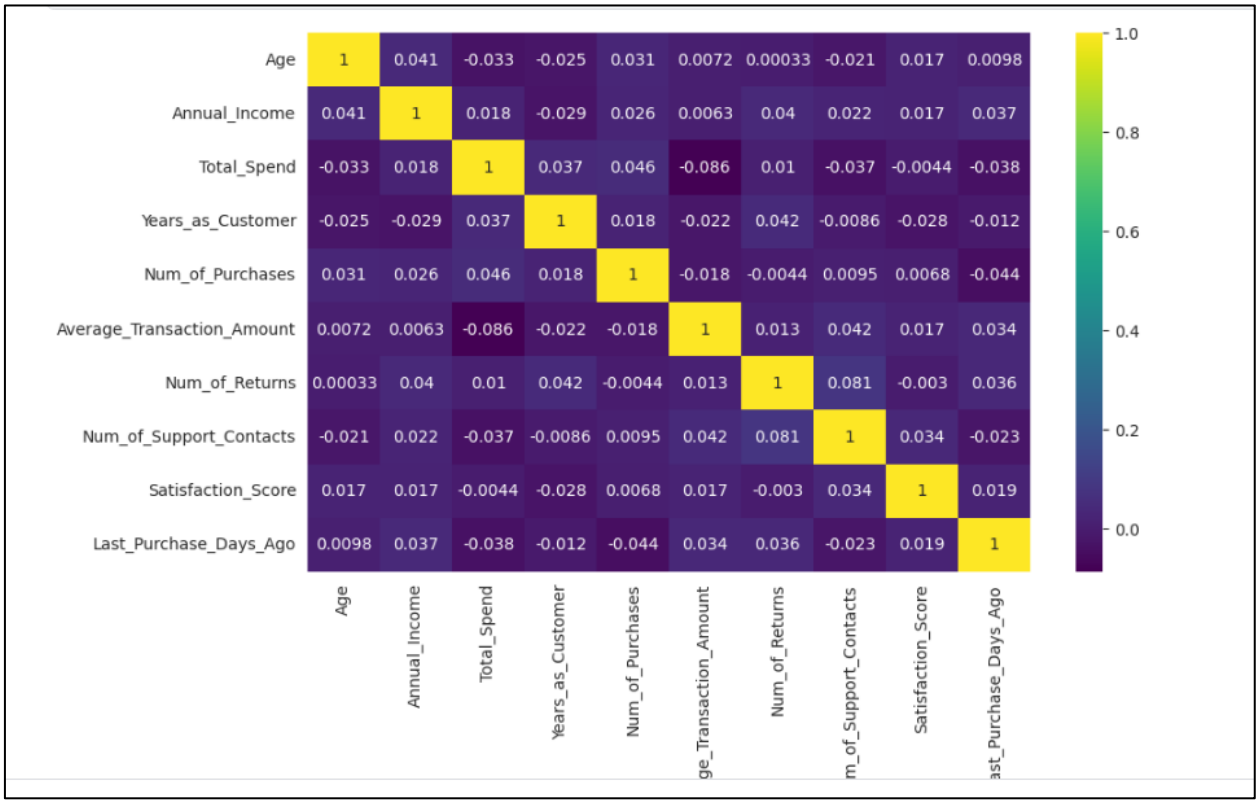


Рисунок Г.2 – Кореляційна матриця показників.

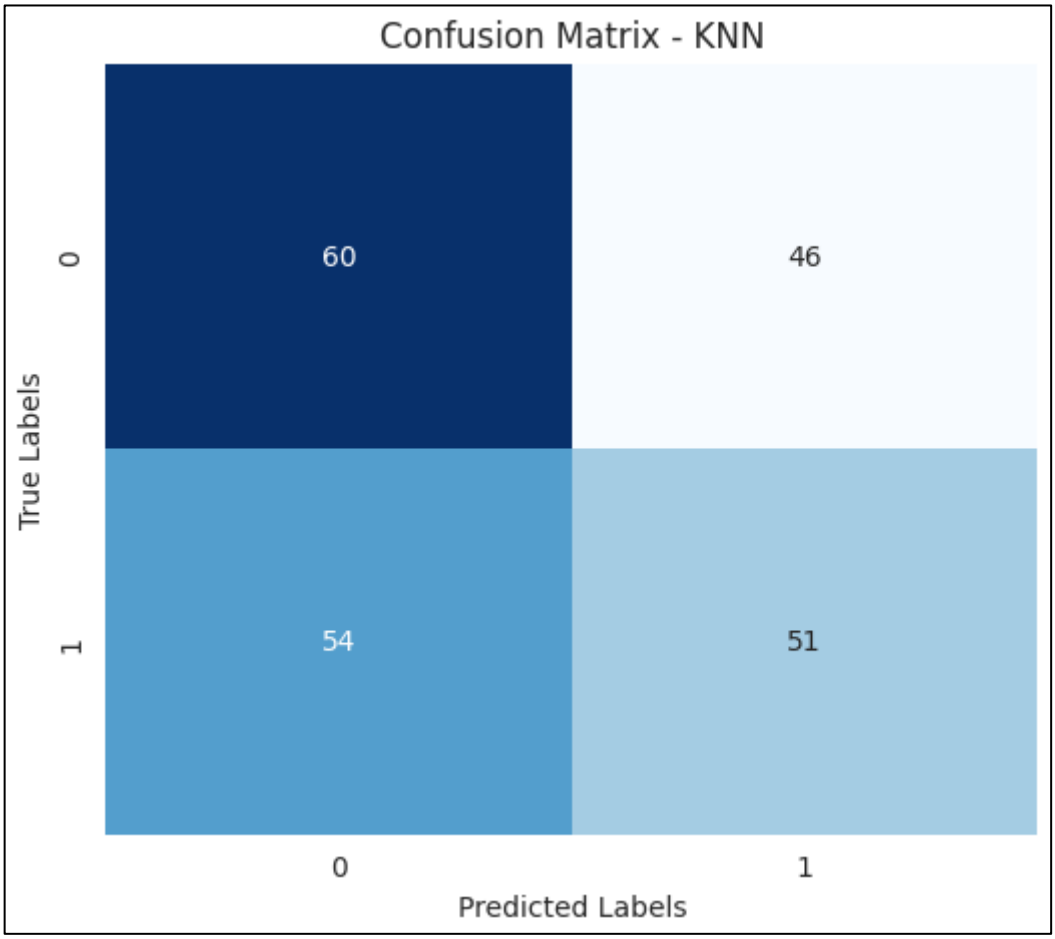


Рисунок Г.3 – Матриця плутанини моделі «KNeighbors».

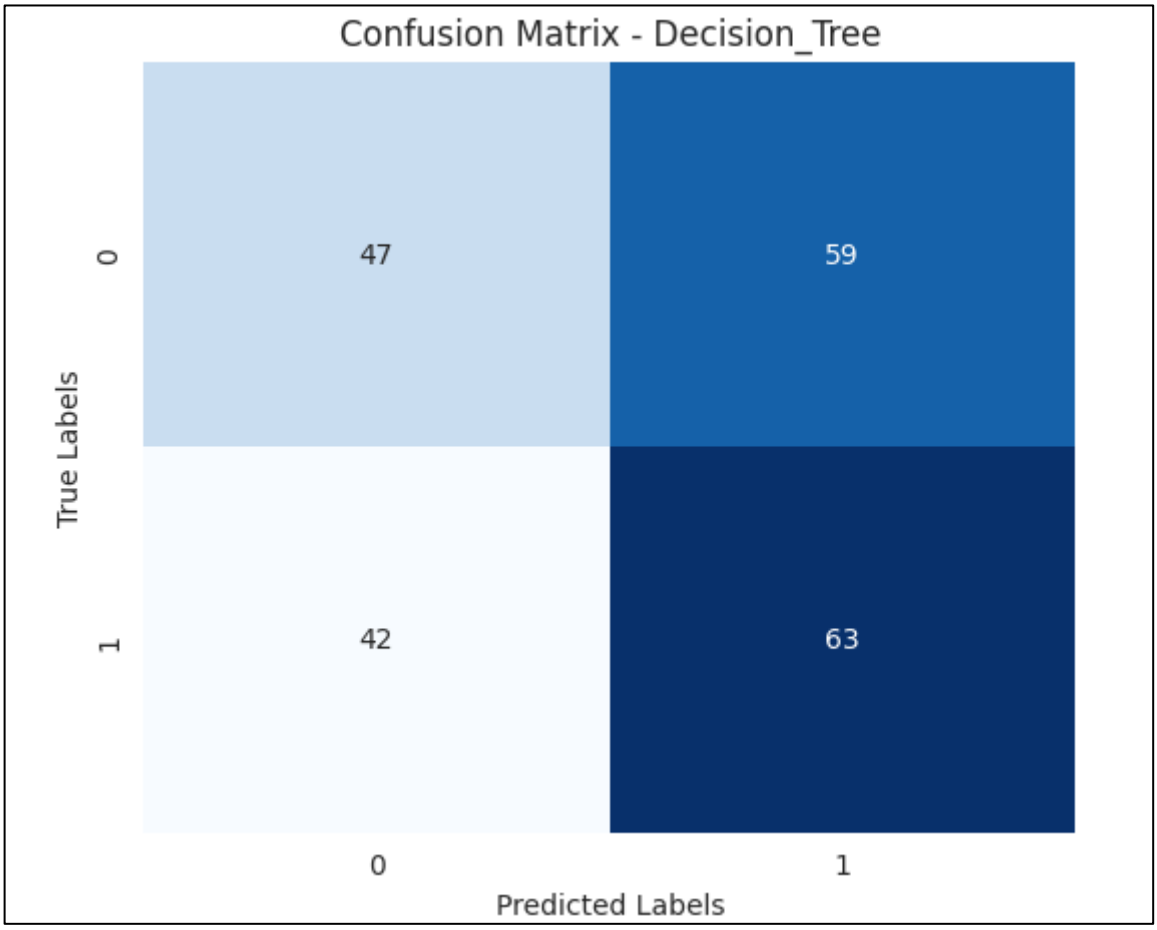


Рисунок Г.4 – Матриця плутанини моделі «Decision Tree».

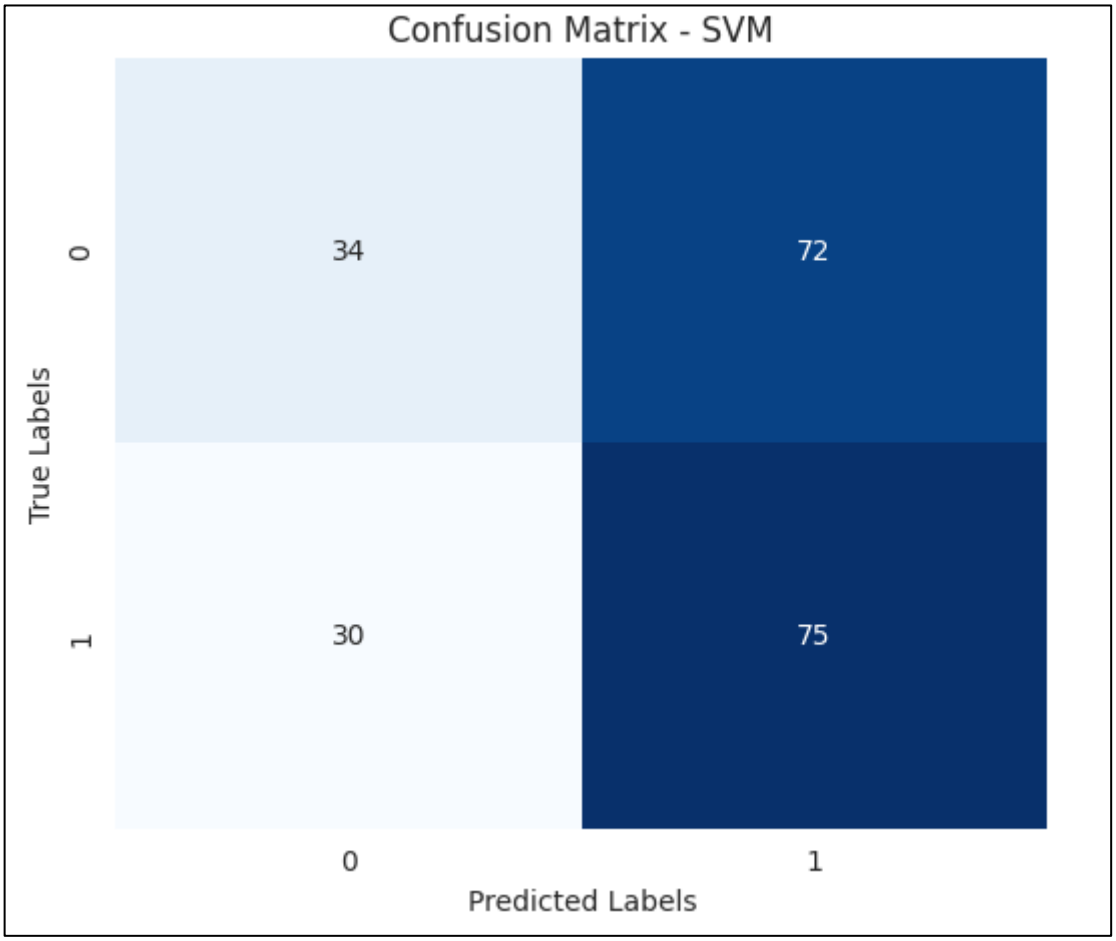


Рисунок Г.5 – Матриця плутанини моделі «SVM».

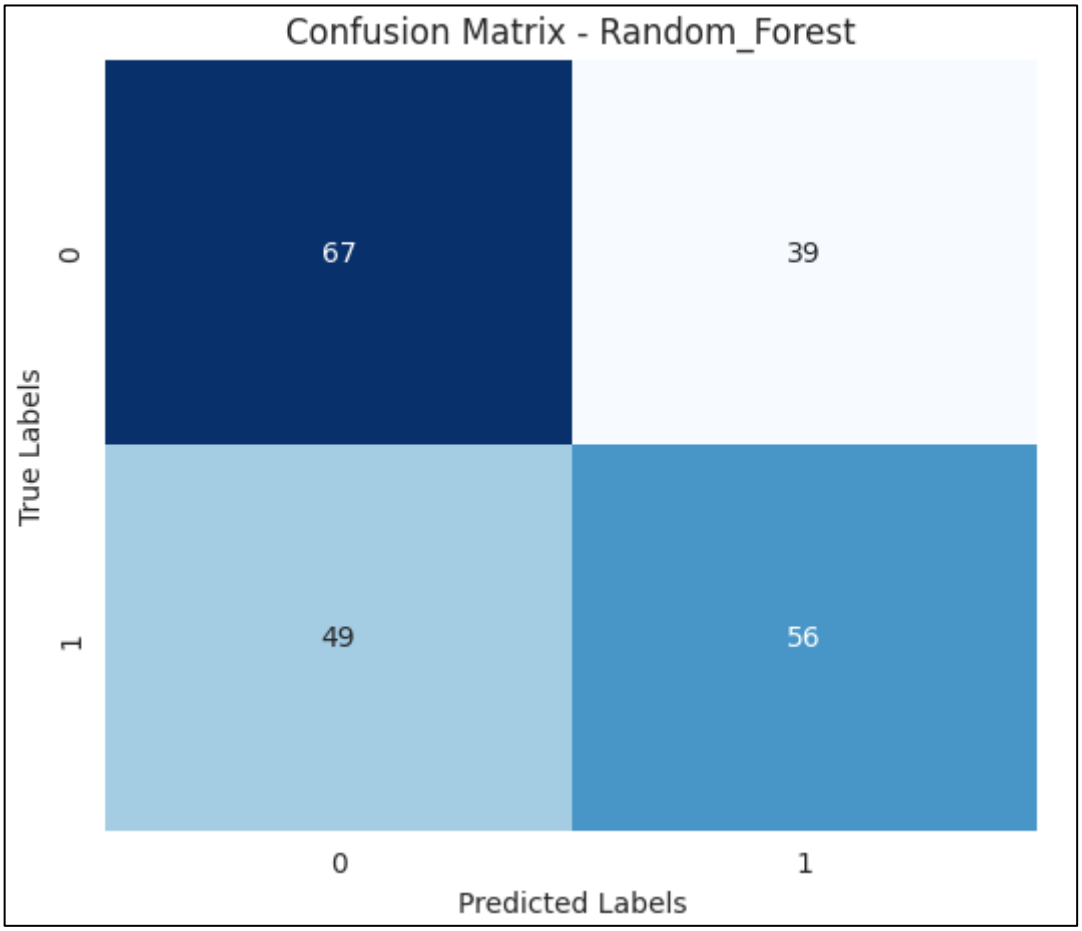


Рисунок Г.6 – Матриця плутанини моделі «Random Forest».