ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Методи та підходи прогнозування поведінки споживачів на основі аналізу даних»

на здобуття освітнього ступеня бакалавра
зі спеціальності 124 Системний аналіз
(код, найменування спеціальності)
освітньо-професійної програми Системний аналіз
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Марія САЛАЄВА*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувачка вищої освіти гр. САД-41
Марія САЛАЄВА
Ім'я, ПРІЗВИЩЕ
Керівник: PhD Михайло КУЗЬМІЧ
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма Системний аналіз

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІСТ
_____Каміла СТОРЧАК
«___» _____2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Салаєва Марія Василівна
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Методи та підходи прогнозування поведінки споживачів на основі аналізу даних

керівник кваліфікаційної роботи Михайло КУЗЬМІЧ, доцент кафедри інформаційних систем та технологій, доктор філософії, затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025 р. №56

2. Строк подання кваліфікаційної роботи «30» травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: відкритий набір даних "Customer Shopping Trends Dataset", який містить структуровану інформацію про поведінкові особливості покупців деякого онлайн-магазину; аналітичні звіти, отримані в результаті аналізу даних, що містять виявлені закономірності, патерни споживчої поведінки та результати сегментації клієнтів; візуалізації даних; практичні рекомендації.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   - Теоретичні основи прогнозування поведінки споживачів в онлайн-магазинах

- Огляд сучасних підходів і методів аналізу споживчої поведінки
- Опис і підготовка даних, характеристика датасету
- Проведення розвідувального аналізу, обробка та кодування даних
- Одновимірний, двовимірний і багатовимірний аналіз факторів впливу
- Побудова та оцінка моделей машинного навчання (SVR, KNN) для прогнозування суми покупки
- Виявлення поведінкових патернів і сегментація споживачів
- Практичні рекомендації для бізнесу
- Висновки

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	15.03.2025 – 17.03.2025	Виконано
2	Написання першого розділу	18.03.2025 – 02.04.2025	Виконано
3	Написання другого розділу	05.04.2025 – 21.04.2025	Виконано
4	Написання третього розділу	22.04.2025 – 27.04.2025	Виконано
5	Написання висновків	27.04.2025 – 28.04.2025	Виконано
6	Підготовка демонстраційних матеріалів	29.04.2025 – 10.05.2025	Виконано
7	Внесення правок	10.05.2025 – 22.05.2025	Виконано
8	Підготовка доповіді	23.05.2025 – 26.05.2025	Виконано

Здобувачка вищої освіти _____ Марія САЛАЄВА

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Михайло КУЗЬМІЧ

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра (магістра): 63 стор, 53 рис, 0 табл, 7 джерел.

Мета роботи - всебічне дослідження поведінкових характеристик споживачів на основі реальних емпіричних даних з метою виявлення ключових факторів і трендів, які впливають на прийняття рішень щодо здійснення покупок, а також розробка ефективної прогностичної моделі для практичного використання у маркетингових стратегіях.

Об'єкт дослідження - відкритий набір даних Customer Shopping Trends Dataset, що містить структуровану інформацію про поведінкові особливості покупців онлайн-магазину.

Предмет дослідження - виявлення та аналіз поведінкових закономірностей споживачів, статистично значущих взаємозв'язків між демографічними характеристиками, параметрами товарів і кінцевими діями клієнтів, а також побудова моделей прогнозування суми покупки.

Короткий зміст роботи: Основний чинник ефективності сучасного бізнесу - це здатність оперативно аналізувати великі обсяги даних про споживачів і на основі цього аналізу формувати персоналізовані маркетингові стратегії. В умовах високої конкуренції та інформаційного перенасичення традиційні підходи до маркетингу вже не забезпечують необхідного рівня залучення та утримання клієнтів. Саме тому ключовою проблемою для компаній стає впровадження інноваційних методів прогнозування поведінки споживачів, які дозволяють не лише фіксувати поточні тенденції, а й передбачати майбутні дії клієнтів.

Використання сучасних інформаційних технологій, зокрема інструментів машинного навчання та аналітики даних, відкриває нові можливості для збору, обробки та інтерпретації поведінкових патернів споживачів. Це дає змогу маркетологам, аналітикам і керівникам приймати обґрунтовані рішення щодо оптимізації рекламних кампаній, управління запасами та підвищення рівня обслуговування клієнтів. Проблема дослідження полягає у виборі та адаптації

таких аналітичних підходів, які забезпечують найбільш точне та стабільне прогнозування суми покупки на основі реальних поведінкових даних.

У дипломній роботі проведено комплексний розвідувальний, одно-, дво- та багатовимірний аналіз поведінки споживачів, що дозволило виявити ключові закономірності та сегменти з найвищим потенціалом зростання. Практична частина дослідження присвячена порівнянню ефективності моделей машинного навчання (SVR та KNN) для прогнозування суми покупки. За результатами аналізу обрано модель KNN як найбільш точну та стабільну для вирішення поставленої задачі. Отримані результати мають значну прикладну цінність для підвищення конкурентоспроможності бізнесу, формування персоналізованих пропозицій та забезпечення довгострокових взаємин із клієнтами.

КЛЮЧОВІ СЛОВА: ПРОГНОЗУВАННЯ, ПОВЕДІНКА СПОЖИВАЧІВ, МАШИННЕ НАВЧАННЯ, АНАЛІЗ ДАНИХ, KNN, SVR, МАРКЕТИНГ, СЕГМЕНТАЦІЯ, EDA, ТРЕНДИ, МАРКЕТИНГОВІ СТРАТЕГІЇ

ЗМІСТ

ВСТУП

Актуальність теми

У сучасних умовах високої конкуренції, перенасиченості ринку та зростаючих вимог споживачів, однією з ключових проблем для малого, середнього та великого бізнесу є розробка та впровадження ефективних методів прогнозування поведінки споживачів. Точне розуміння майбутніх дій інтернет-користувачів дає змогу маркетологам і рекламним агентствам оптимізувати рекламні кампанії, підвищуючи їхню цілеспрямованість, персоналізацію та результативність, що сприяє збільшенню конверсій і зацікавленості у продуктах.

Прогнозування поведінки споживачів базується на аналізі їхніх дій, уподобань, звичок та соціальних сигналів, що дозволяє формувати маркетингові стратегії, адаптовані до постійно змінних ринкових умов і очікувань клієнтів. Особливо важливо це в умовах інформаційного перенасичення та рекламної сліпоти, коли традиційні методи маркетингу втрачають ефективність.

Практична реалізація прогнозування передбачає застосування сучасних аналітичних інструментів, зокрема технологій штучного інтелекту та машинного навчання. Ці технології дозволяють обробляти великі обсяги даних, виявляти приховані закономірності та створювати точні прогностичні моделі, що допомагають бізнесу швидко реагувати на зміни, оптимізувати маркетингові заходи, управління запасами та підвищувати рівень обслуговування клієнтів.

Таким чином, дослідження методів і підходів прогнозування поведінки споживачів на основі аналізу даних є надзвичайно актуальним для підвищення конкурентоспроможності бізнесу, формування персоналізованих маркетингових стратегій і забезпечення довгострокових взаємин із клієнтами.

Мета дослідження: метою даної дипломної роботи є всебічне дослідження поведінкових характеристик споживачів на основі реальних емпіричних даних з метою виявлення ключових факторів і трендів, які впливають на прийняття рішень щодо здійснення покупок. Дослідження передбачає не лише перевірку статистичних гіпотез, а й глибоке занурення у психологічні та поведінкові аспекти споживчої

активності, що дозволяє отримати комплексне розуміння мотивацій та закономірностей покупців. У завершальній частині роботи проводиться порівняльний аналіз двох моделей машинного навчання з метою визначення найбільш ефективної, яка здатна забезпечити точне прогнозування майбутньої динаміки зростання суми покупки на основі виявлених поведінкових патернів і трендів. Такий підхід відкриває нові можливості для побудови персоналізованих маркетингових стратегій, що відповідають сучасним викликам ринку та потребам споживачів.

Завдання дослідження

У межах даного дослідження поставлено низку ключових завдань, які спрямовані на всебічне вивчення поведінки споживачів та розробку ефективних інструментів прогнозування їхніх покупкових рішень.

По-перше, необхідно здійснити одновимірний аналіз, що дозволить визначити вплив окремих змінних, таких як вік, стать, розмір одягу та категорія товару, на рівень купівельної активності. Цей етап є фундаментальним для розуміння базових закономірностей у поведінці різних сегментів споживачів.

По-друге, важливо провести двовимірний аналіз з метою виявлення взаємозв’язків між змінними, зокрема між віком та категорією товару, у контексті середнього чека. Такий підхід дозволяє глибше зрозуміти, як комбіновані фактори впливають на фінансові показники покупок, що є важливим для сегментації аудиторії та формування таргетованих пропозицій.

По-третє, реалізація багатовимірного аналізу дасть змогу побудувати комплексну модель прогнозування поведінки споживачів, яка враховуватиме взаємодію кількох змінних одночасно. Це сприятиме підвищенню точності прогнозів та ефективності маркетингових стратегій.

Четвертим завданням є ідентифікація патернів - повторюваних моделей поведінки клієнтів, які можуть бути використані для розробки таргетингових стратегій і персоналізації рекламних кампаній. Виявлення таких закономірностей допоможе краще адаптувати пропозиції до потреб конкретних груп споживачів.

П’ятим кроком стане формування практичних рекомендацій на основі отриманих результатів, що дозволить впроваджувати ефективні рішення у сфері маркетингу та підвищувати результативність рекламних заходів.

Нарешті, шостим завданням є проведення порівняльної оцінки двох моделей машинного навчання - Support Vector Regression (SVR) та K-Nearest Neighbors (KNN). Вибір найефективнішої моделі на основі показників точності прогнозування (наприклад, MAE, RMSE) стане підґрунтям для реалізації прогнозу динаміки зростання суми покупки з урахуванням виявлених поведінкових патернів.

Комплексне виконання цих завдань забезпечить глибоке розуміння споживчої поведінки та створить надійну основу для розробки інноваційних маркетингових стратегій, орієнтованих на максимальне задоволення потреб клієнтів і підвищення ефективності бізнесу.

Об’єкт і предмет дослідження

Об’єктом даного дослідження виступає відкритий набір даних Customer Shopping Trends Dataset, який містить структуровану інформацію про поведінкові особливості покупців у межах конкретного онлайн-магазину. Цей датасет включає ключові демографічні характеристики споживачів, такі як вік, стать та розмір одягу, а також параметри товарів - категорію, кількість одиниць та ціну. Завдяки цим даним можливо відтворити повний ланцюг прийняття рішення клієнтом - від першого контакту з товаром до фактичного здійснення покупки, що відкриває широкі можливості для глибокого аналізу та прогнозування споживчої поведінки.

Предметом дослідження є виявлення та аналіз поведінкових закономірностей споживачів, які проявляються у вигляді статистично значущих взаємозв’язків між демографічними характеристиками, параметрами товарів та кінцевими діями клієнтів. Дослідження охоплює вивчення впливу окремих факторів на купівельну активність, ідентифікацію повторюваних моделей поведінки (патернів), а також розробку прогностичних моделей, що дозволяють оцінити майбутню динаміку купівельної активності, зокрема прогнозувати суму покупки. Особлива увага приділяється формалізації поведінки споживача у вигляді кількісних

характеристик, які можуть бути використані для створення високоточних алгоритмів прогнозування.

Методологія дослідження

Вона ґрунтується на комплексному застосуванні кількісних та якісних методів, що забезпечують всебічний аналіз поведінки споживачів і дозволяють створити надійні прогностичні моделі.

Першим етапом є розвідувальний аналіз даних (Exploratory Data Analysis, EDA), який дозволяє виявити основні характеристики вибірки, оцінити розподіл змінних та виявити потенційні аномалії чи закономірності. Цей крок є фундаментальним для подальшої побудови аналітичних моделей.

Далі застосовується одновимірний аналіз, що дає змогу дослідити вплив окремих змінних, таких як вік, стать, розмір одягу та категорія товару, на купівельну активність. Такий аналіз дозволяє окремо оцінити внесок кожного фактора у формування поведінкових патернів.

Для глибшого розуміння взаємодії між змінними використовується двовимірний аналіз, зокрема зосереджений на вивченні зв’язку між віком та категорією товару у контексті середнього чека. Цей підхід дозволяє виявити складніші залежності, що впливають на прийняття рішень споживачами.

Багатовимірний аналіз забезпечує можливість побудови комплексної моделі поведінки споживачів, яка одночасно враховує вплив кількох змінних. Такий підхід підвищує точність прогнозів і дозволяє моделювати реальні умови ринку.

Для перевірки статистичної значущості отриманих результатів застосовуються відповідні статистичні тести, зокрема критерії Манна-Уїтні та Спірмена, що дозволяють оцінити достовірність виявлених закономірностей.

Ключовим етапом дослідження є використання методів машинного навчання для створення моделей прогнозування динаміки купівельної активності на основі поведінкових даних. Застосовано два алгоритми:

- Support Vector Regression (SVR) - метод опорно-векторної регресії, який ефективно моделює складні нелінійні залежності, використовуючи концепцію

- гіперплощин у багатовимірному просторі. Цей метод забезпечує високу точність прогнозування навіть за наявності шуму у даних.
- K-Nearest Neighbors (KNN) - алгоритм, що базується на гіпотезі про схожість об’єктів, прогнозуючи значення залежної змінної на основі найближчих за характеристиками спостережень. Цей підхід є інтуїтивним і добре адаптується до локальних варіацій у даних.

На заключному етапі здійснюється порівняльна оцінка обох моделей за критеріями точності прогнозування, такими як середня абсолютна помилка (MAE) та корінь середньоквадратичної помилки (RMSE). Обрана найефективніша модель використовується для побудови прогнозу динаміки зростання суми покупки з урахуванням виявлених поведінкових патернів.

Таким чином, поєднання класичних статистичних методів з сучасними алгоритмами машинного навчання дозволяє отримати глибоке розуміння споживчої поведінки та створити практично застосовні інструменти для підвищення ефективності маркетингових стратегій.

Інформаційна база дослідження

Вона базується на відкритому наборі даних Customer Shopping Trends Dataset, який містить детальну структуровану інформацію про поведінку споживачів у контексті онлайн-торгівлі. Цей масив даних слугує фундаментом для емпіричного аналізу та моделювання споживчої активності.

Окрім цього, у роботі використано широкий спектр наукової літератури з галузей маркетингу, аналітики, поведінкової економіки та машинного навчання. Поєднання теоретичних концепцій із сучасними практичними методами аналізу дозволяє сформувати цілісне уявлення про механізми прийняття споживацьких рішень та розробити інноваційні підходи до прогнозування їхньої поведінки.

Загалом, дана дипломна робота має на меті інтегрувати знання з аналітики та передових технологій, що створює міцну основу для формування ефективних маркетингових стратегій, здатних адекватно реагувати на виклики сучасного ринку та забезпечувати конкурентні переваги бізнесу.

1. ТЕОРЕТИЧНІ ЗАСАДИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ СПОЖИВАЧІВ

Прогнозування споживчої поведінки має глибокі історичні корені, що сягають ще до появи Інтернету та цифрових технологій. Вже багато століть тому продавці на ринках і базарах намагалися передбачити попит, орієнтуючись на власний досвід, спостереження за покупцями та сезонні коливання. Цей емпіричний підхід, що передавався з покоління в покоління, можна вважати першою формою маркетингової аналітики. Наприклад, продавці знали, що перед зимою зростає попит на теплий одяг, а напередодні свят - на подарунки та прикраси. Ведення записів про продажі і залишки товарів слугувало своєрідною базою даних, що допомагала прогнозувати попит і планувати закупівлі.

Важливим інструментом прогнозування тоді було спілкування - обмін інформацією між продавцями та покупцями, а також між самими торговцями. Це дозволяло відстежувати зміни в смаках і попиті, навіть у сусідніх регіонах, формуючи так званий «живий маркетинг». Особливо складними були 1990-ті роки, коли нестабільна економічна ситуація змушувала бізнес діяти швидко і інтуїтивно, без доступу до великих обсягів даних.

Сьогодні ситуація кардинально змінилася. Прогнозування поведінки споживачів базується на глибокому аналізі великих масивів даних із застосуванням сучасних технологій: штучного інтелекту, машинного навчання, поведінкової аналітики та інструментів типу Google Analytics. Ці інструменти дозволяють не просто спостерігати за трендами, а й виявляти приховані закономірності в поведінці користувачів, що дає змогу формувати персоналізовані маркетингові стратегії.

Поведінкові патерни та сегментація

Поведінкові патерни - це стійкі моделі дій користувачів у цифровому середовищі, які можна виявити за допомогою аналітичних інструментів. Вони відображають, наприклад, частоту відвідувань сайту, типи переглянутих товарів, тривалість перебування на сторінках, взаємодію з рекламними оголошеннями, а також дії з товарами (додавання у кошик, порівняння, купівля).

Приклад поведінкового патерну: користувач, який регулярно переглядає товари зі знижками у вечірній час, демонструє не лише інтерес до акцій, а й певний

часовий патерн активності. Це дозволяє маркетологам пропонувати йому релевантні пропозиції у відповідний час.

Поведінкова сегментація поділяє аудиторію на групи на основі реальних дій, а не лише демографічних характеристик. Серед типових сегментів можна виділити:

- Лояльні користувачі, які часто повертаються і здійснюють покупки;
- Мисливці за знижками, що реагують переважно на акції та розпродажі;
- Нові відвідувачі, які щойно почали взаємодіяти з платформою;
- «Кинули кошик», тобто користувачі, які додали товар, але не завершили покупку.

Ця сегментація дає змогу рекламодавцям точніше таргетувати повідомлення, адаптувати контент і частоту показів, а також розробляти ефективні стратегії ремаркетингу.

Приклади застосування поведінкової сегментації

- Amazon використовує алгоритми рекомендацій, які аналізують історію покупок і переглядів, пропонуючи товари, що можуть зацікавити користувача, базуючись на принципі соціального доказу.
- Coca-Cola в кампанії «Share a Coke» персоналізувала пляшки, друкуючи імена споживачів, що створило емоційний зв’язок і підвищило лояльність.

Психологічні та економічні аспекти

Розуміння психології клієнтів дозволяє створювати маркетингові кампанії, які не лише інформують, а й емоційно залучають аудиторію, підвищуючи ефективність комунікації. Економічний аналіз допомагає оптимізувати бюджет, визначати найбільш рентабельні канали та стратегії, що забезпечує максимальний ROI.

1.1 Методи аналізу даних та прогнозування

У процесі дослідження трендів і поведінкових патернів споживачів застосовуються різноманітні методи аналізу даних та прогнозування, кожен з яких виконує специфічну функцію і вносить унікальний внесок у формування цілісної картини поведінки покупців. Відсутність універсального підходу зумовлює

необхідність комбінування кількох методик, що дозволяє глибше та точніше інтерпретувати отримані дані.

Особливу увагу в роботі приділено розвідувальному аналізу даних (Exploratory Data Analysis, EDA), який не обмежується простим відображенням числових значень, а спрямований на виявлення прихованих закономірностей, структурних особливостей та мотиваційних чинників, що лежать в основі поведінкових патернів споживачів. Розглядаючи EDA як інструмент розгадування загадок, можна сказати, що дані виступають у ролі підказок, а методи аналізу - це засоби їх дешифрування, що відкривають глибше розуміння поведінкових процесів.

Застосування різноманітних методів аналізу дозволяє не лише описати поточний стан ринку та поведінку клієнтів, а й здійснити точне прогнозування майбутніх тенденцій, що є ключовим для формування ефективних маркетингових стратегій та прийняття обґрунтованих бізнес-рішень.

1.2 Одновимірний аналіз

Одновимірний аналіз є базовою і водночас надзвичайно ефективною формою розвідкового аналізу, яка зосереджується на вивченні однієї змінної. Цей підхід дозволяє дослідити розподіл, варіабельність, центральні тенденції (середнє, медіану) та наявність аномалій або викидів у даних. Наприклад, можна проаналізувати, як розподіляється вік клієнтів або як змінюється середній чек покупця. Одновимірний EDA "очищує" картину, фокусуючись на одному факторі, що допомагає виявити ключові характеристики та початкові закономірності.

1.3 Двовимірний аналіз

Двовимірний аналіз розширює межі дослідження, дозволяючи одночасно вивчати взаємозв’язок між двома змінними. Це дає змогу оцінити кореляції, тренди та потенційні взаємодії між факторами. Наприклад, аналіз поєднання віку та сезону може показати, як різні вікові групи змінюють свої купівельні звички залежно від часу року. Для візуалізації використовують діаграми розсіювання, двовимірні гістограми, а також кореляційні матриці. Двовимірний EDA допомагає розпізнати складніші закономірності, які не видно при аналізі окремих змінних.

1.4 Багатовимірний аналіз

Багатовимірний аналіз - це найбільш комплексна форма розвідкового аналізу, яка дозволяє одночасно досліджувати три і більше змінних. Він спрямований на виявлення складних взаємозв’язків, структур і факторів, що впливають на досліджувані показники. У практиці застосовують такі методи, як факторний аналіз, кластерний аналіз, багатовимірне шкалювання та інші багатовимірні статистичні підходи. Наприклад, можна вивчати одночасний вплив віку, статі, місця проживання та джерела трафіку на ймовірність здійснення покупки. Багатовимірний EDA дає змогу отримати цілісну картину даних і краще підготувати їх до подальшого моделювання чи прогнозування.

1.5 Кореляційний аналіз

Кореляційний аналіз є важливим інструментом для виявлення зв’язків між двома змінними. Особливо корисним є застосування коефіцієнта кореляції Спірмена, який ефективно працює навіть із даними, що не мають нормального розподілу або представлені у порядковій шкалі. Хоча кореляція не свідчить про причинно-наслідкові зв’язки, наявність сильного кореляційного зв’язку є вагомим сигналом для подальшого глибшого аналізу. Наприклад, виявлена позитивна кореляція між часом перебування користувача на сторінці товару та ймовірністю його придбання свідчить про важливість уваги до поведінкових патернів.

1.6 Статистичні тести

Статистичні тести виконують роль інструментів для перевірки гіпотез і підтвердження достовірності отриманих результатів. Зокрема, критерій Манна-Уїтні широко використовується для порівняння двох незалежних вибірок, наприклад, для визначення, чи відрізняється середній чек користувачів, які отримали персоналізовану знижку, від тих, хто її не отримав. Використання таких тестів дозволяє уникнути хибних висновків, базованих на випадкових збігах, і підвищує надійність аналітичних результатів.

1.7 Застосування методів у дослідженні

Початковим кроком у роботі є описова статистика та візуалізація даних, що дозволяє «побачити» структуру аудиторії, виявити піки, провали та аномалії.

Кореляційний аналіз та статистичні тести служать для перевірки гіпотез, що виникають на основі досвіду та інтуїції. Кластеризація, хоч і застосовується рідше, є потужним інструментом для виявлення однорідних груп користувачів, що є критично важливим для таргетованої реклами.

Дво- та багатовимірний аналізи дозволяють врахувати складність реальних умов і взаємодію кількох факторів, що впливають на поведінку сучасного споживача. Такий комплексний підхід є необхідним для побудови точних моделей і формування ефективних маркетингових стратегій.

1.8 Прогнозування на основі машинного навчання: вибір методів KNN та SVR

Для прогнозування майбутньої поведінки споживачів у дослідженні застосовуються два потужні алгоритми машинного навчання: K-Nearest Neighbors (KNN) та Support Vector Regression (SVR).

Метод KNN базується на принципі схожості: прогнозування здійснюється на основі найближчих за характеристиками сусідів у просторі ознак. Цей алгоритм є інтуїтивно зрозумілим, адаптивним і добре працює з локальними варіаціями в даних. Він особливо ефективний у випадках, коли поведінка споживачів має яскраво виражені кластери або сегменти.

SVR, у свою чергу, є складнішим і потужнішим методом, що дозволяє моделювати нелінійні залежності між змінними, використовуючи концепцію гіперплощин у багатовимірному просторі. Цей алгоритм забезпечує високу точність прогнозування навіть у разі наявності шуму в даних і складної структури поведінкових патернів.

Вибір саме цих двох методів зумовлений їхньою доповнюваністю: KNN добре справляється з локальними особливостями та інтуїтивно зрозумілий, а SVR - здатен моделювати складні глобальні залежності. Порівняльний аналіз їхньої ефективності дозволяє обрати оптимальну модель для точного прогнозування динаміки купівельної активності, що є ключовим для прийняття обґрунтованих маркетингових рішень.

1.9 Досвід з моєї компанії

На сьогоднішній день автор має можливість працювати в динамічному середовищі рекламної мережі, що спеціалізується на просуванні товарів і послуг партнерів через інтернет. Це середовище характеризується постійним впровадженням новітніх інструментів, аналітичних підходів та вдосконаленням маркетингових стратегій. Саме тут щоденно застосовуються теоретичні знання з маркетингу та прогнозування поведінки споживачів у практичній діяльності.

На початковому етапі професійної діяльності автор ознайомився з базовими інструментами, зокрема Google Analytics, і поступово усвідомив, наскільки глибоко можна аналізувати поведінку користувачів. Кожен клік, додавання товару до кошика чи перегляд певної категорії стають цінними сигналами, які підлягають детальному вивченню.

Процес аналізу поведінки користувачів починається одразу після запуску рекламної кампанії. Відстежуються ключові метрики: кількість переходів за оголошенням, дії на сайті, тривалість перебування на сторінках, переглянуті товари, додані до кошика позиції та ігноровані продукти. Ці дані є надзвичайно цінними для маркетолога, оскільки дозволяють формувати повну картину взаємодії користувача з платформою.

Застосування Google Analytics дає змогу інтерпретувати такі показники, як глибина перегляду, відсоток відмов, конверсії та середній час на сторінці. Наприклад, якщо користувачі часто додають товари до кошика, але не завершують покупку, це сигналізує про потенційні проблеми з ціною, умовами доставки або іншими факторами, що впливають на рішення про покупку. Отримані інсайти оперативно передаються партнерам для корекції стратегії.

Особливе значення має не лише збір даних, а й їх правильна інтерпретація та прийняття рішень на основі аналітичних висновків. Команда аналітиків формує детальні звіти та рекомендації щодо оптимізації асортименту, підвищення ефективності просування та точного таргетування аудиторії. Цей процес є безперервним і спрямований на постійне вдосконалення маркетингових кампаній.

Персоналізація реклами виступає ключовим фактором підвищення її ефективності. Рекламні повідомлення адаптуються під інтереси користувачів: якщо клієнт цікавився певною категорією товарів, йому демонструються релевантні пропозиції; якщо він порівнював ціни - пропонуються вигідніші варіанти. Такий підхід значно збільшує ймовірність конверсії, формує довіру до бренду та оптимізує витрати рекламного бюджету.

Результати застосування аналітичного, персоналізованого та точного підходу до маркетингу є вражаючими. Партнери компанії фіксують збільшення продажів на 30-40% після впровадження рекомендацій, що свідчить про ефективність організації процесу. Це підтверджує, що бізнес, який глибоко розуміє своїх клієнтів, завжди має конкурентну перевагу.

Сучасний маркетинг перестав бути лише креативною діяльністю - він став наукою про дані, точність і глибоке розуміння поведінки споживачів. Саме поєднання аналітики та емоційного підходу дозволяє створювати дійсно результативні стратегії просування, що відповідають викликам сучасного ринку.

2. ПРАКТИЧНА РЕАЛІЗАЦІЯ

Практичний досвід є невід'ємною складовою глибокого розуміння аналітики. Теоретичні знання набувають справжньої цінності лише в процесі роботи з реальними даними, коли можна побачити, як працюють методи на практиці та які інсайти вони дозволяють отримати. У межах цієї дипломної роботи було прийнято рішення поєднати академічні підходи з практичними навичками, що щоденно застосовуються у професійній діяльності. Такий підхід дозволяє створити аналітичний продукт, який має реальне застосування у бізнес-середовищі.

Для реалізації дослідницького проєкту використовується платформа Kaggle - це не просто онлайн-ресурс, а глобальна спільнота аналітиків, студентів і спеціалістів у галузі data science. Kaggle надає доступ до великої кількості відкритих датасетів, інтерактивних ноутбуків з кодом, обговорень і прикладів робіт, що сприяє обміну знаннями та колективному навчанню. Щодня тисячі користувачів з усього світу запускають аналітичні експерименти, діляться результатами та вдосконалюють свої навички - і автор цієї роботи є активним учасником цієї спільноти.

Особливо цінним є можливість навчатися на реальних прикладах, спостерігати за мисленням професіоналів і одночасно розробляти власні проєкти, що відповідають сучасним вимогам аналітики.

Для дипломного дослідження обрано набір даних «Customer Shopping Trends Dataset», який містить понад 3900 записів про покупки клієнтів. Цей структурований датасет включає такі ключові параметри, як вік, стать, розмір одягу, категорія товару, сума покупки, спосіб оплати, відгуки та сезонність. Наявність цих змінних забезпечує комплексний аналіз поведінки споживачів, дозволяючи виявити закономірності, тренди та фактори, що впливають на прийняття рішень щодо покупок.

2.1 Підготовка до аналізу

Імпорт бібліотек у Python є початковим і надзвичайно важливим етапом будь-якого аналітичного дослідження. Саме від правильного вибору інструментів залежить ефективність подальшої роботи з даними. Для аналізу та прогнозування

поведінки споживачів використовуються ключові бібліотеки, що забезпечують широкий спектр функціоналу - від обробки та очищення даних до їх візуалізації і побудови складних моделей.

Pandas виступає базовою бібліотекою для роботи з табличними даними, дозволяючи зручно імпортувати, переглядати, фільтрувати та агрегувати інформацію. Вона створює фундамент для подальшого аналізу, роблячи дані доступними та зрозумілими.

NumPy доповнює Pandas, забезпечуючи потужні можливості для виконання математичних операцій над великими масивами даних, формування векторів і матриць, що є необхідним для статистичних розрахунків і моделювання.

Seaborn використовується для створення інформативних та естетично привабливих візуалізацій, що допомагають виявляти закономірності, взаємозв’язки та аномалії у даних. Теплові карти, коробкові діаграми, парні графіки - це лише частина інструментарію, що сприяє глибокому розумінню структури даних.

Matplotlib є фундаментальною бібліотекою для візуалізації, що надає повну свободу у створенні різноманітних графіків - від простих гістограм до складних кругових діаграм, забезпечуючи індивідуальний підхід до подання результатів.

Завдяки поєднанню цих бібліотек дослідник здатен провести повний цикл аналітичної роботи: від очищення та підготовки даних, через одновимірний і багатовимірний аналіз, до візуалізації результатів і побудови прогностичних моделей. Такий підхід не лише підвищує якість дослідження, а й робить його практично значущим для сучасних бізнес-завдань.

Імпорт бібліотек у Python - це своєрідний початок подорожі у світ даних, що відкриває широкі можливості для дослідження, розуміння та прогнозування складних поведінкових процесів у сфері споживчої активності.

Основні бібліотеки, які були використані для дослідження:

[1]:
```python
# Додаємо бібліотеки, що будуть корисними для обробки числових та табличних даних
import numpy as np
import pandas as pd

# Підключаємо бібліотеки, що будуть необхідні для візуалізації трендів та аналізу патернів
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as mcolors
import seaborn as sns

# Для гарненьких графіків та налаштувань з кольору
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.colors as pc
```

Рис. 2.1 Імпорт бібліотек з моїми коментарями

Під час вивчення машинного навчання на четвертому курсі ми неодноразово стикалися з проблемою численних попереджень (warnings), які виводяться під час виконання коду. Хоча ці повідомлення не призводять до зупинки програми, вони суттєво відволікають увагу і ускладнюють аналіз результатів, особливо при роботі з великими обсягами даних. Для підвищення зручності роботи та концентрації уваги на ключових аспектах дослідження, застосуємо спеціальний рядок коду, який дозволяє приглушити ці попередження, забезпечуючи більш чистий і зрозумілий вивід результатів:

[1]:
```python
# Додаємо бібліотеки, що будуть корисними для обробки числових та табличних даних
import numpy as np
import pandas as pd

# Підключаємо бібліотеки, що будуть необхідні для візуалізації трендів та аналізу патернів
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as mcolors
import seaborn as sns

# Для гарненьких графіків та налаштувань з кольору
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.colors as pc

# Коли ми проходили на 4 курсі машинне навчання, виникало багато попереджень під час роботи коду.
# Тому я додаю ігнорування попередженнь (warnings), які можуть з'являтися під час виконання програми:
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

Рис. 2.2 Оновлений код з імпортом бібліотек та ігноруванням попереджень з моїми коментарями

Це команда, яка дозволяє "вимкнути" відображення попереджень. Вона дуже зручна під час роботи над дипломом або при створенні візуальних звітів, де не хочеться бачити ніякого зайвого виводу.

2.2 Завантаження даних

Після імпорту необхідних бібліотек наступним ключовим етапом є завантаження датасету для подальшого аналізу. У роботі використовується «Customer Shopping Trends Dataset» - відкритий, структурований набір даних у форматі CSV, доступний на платформі Kaggle. Цей датасет містить понад 3900 записів про покупки клієнтів і включає як кількісні, так і якісні змінні, що забезпечує широкий спектр можливостей для застосування різноманітних методів аналізу - від описової статистики до складних моделей машинного навчання.

Для імпорту даних у середовище Python застосовується бібліотека pandas, яка є стандартним інструментом для роботи з табличними даними. Використання функції read_csv() дозволяє швидко і зручно завантажити датасет у форматі DataFrame, що забезпечує подальшу ефективну обробку, очищення та аналіз інформації:

```
[2]: df = pd.read_csv('/kaggle/input/customer-shopping-trends-dataset/shopping_trends_updated.csv')
```

Рис. 2.3 Код, що завантажує датасет «Customer Shopping Trends Dataset»

2.3 Попередній огляд

Після завантаження датасету у середовище Python одним із перших кроків є перевірка коректності імпорту даних. Це необхідно для впевненості, що інформація з файлу повністю і правильно відобразилася у робочій структурі, що забезпечить точність подальшого аналізу. Для цього застосовується метод:

[3]:
```
df.sample(10)
```

[3]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
2621	2622	48	Male	Shoes	Footwear	46	Maine	M	Beige	Winter	3.0	No	Express	No	No	21	Bank Transfer	Annually
1298	1299	29	Male	Pants	Clothing	81	Indiana	S	Indigo	Summer	3.0	No	Free Shipping	Yes	Yes	14	Bank Transfer	Fortnightly
620	621	33	Male	Backpack	Accessories	89	Rhode Island	M	Black	Fall	3.7	Yes	2-Day Shipping	Yes	Yes	10	Venmo	Every 3 Months
2089	2090	37	Male	Jacket	Outerwear	57	Maryland	L	Gold	Spring	3.6	No	Free Shipping	No	No	41	Credit Card	Every 3 Months
868	869	40	Male	Dress	Clothing	83	North Carolina	M	Green	Summer	3.2	Yes	Next Day Air	Yes	Yes	2	Credit Card	Every 3 Months
175	176	53	Male	Coat	Outerwear	86	Oklahoma	L	Violet	Fall	2.8	Yes	Next Day Air	Yes	Yes	2	Bank Transfer	Annually
3796	3797	40	Female	Shorts	Clothing	81	Utah	M	Lavender	Fall	3.2	No	Free Shipping	No	No	47	Bank Transfer	Quarterly
1318	1319	52	Male	Shorts	Clothing	93	Connecticut	M	Maroon	Spring	2.7	No	Express	Yes	Yes	19	Credit Card	Bi-Weekly
869	870	38	Male	Hat	Accessories	54	New Hampshire	M	Silver	Spring	2.7	Yes	2-Day Shipping	Yes	Yes	33	PayPal	Weekly
547	548	38	Male	Jacket	Outerwear	50	Nevada	M	Blue	Winter	2.8	Yes	Free Shipping	Yes	Yes	26	Venmo	Weekly

Рис. 2.4 Результати виклику методу df.sample().

Даний метод дозволяє вивести десять випадкових рядків із таблиці, що відрізняється від традиційного використання функції df.head(), яка демонструє лише перші записи. Відображення випадкових зразків є надзвичайно корисним для первинної валідації даних, оскільки дозволяє отримати більш репрезентативне уявлення про зміст датасету. На мою думку, це один із найпростіших, але водночас ефективних способів швидко ознайомитися з «живими» даними. Якщо під час перегляду виявляються аномалії, такі як дивні символи, порожні комірки або невідповідні типи даних, це свідчить про можливі проблеми з якістю даних.

Після попереднього огляду необхідно провести детальну перевірку на наявність пропущених або нульових значень, які можуть ускладнити подальший аналіз, спричинити помилки у коді та спотворити результати дослідження. Для цього застосовуються два ключові методи:

1. Аналіз структури таблиці за допомогою df.info(). Цей метод надає повну інформацію про кожну колонку датасету, включно з назвою, кількістю непорожніх значень та типом даних (числові, об’єктні, категоріальні). Це дозволяє оперативно виявити:

- Наявність пропущених значень (nulls);
- Колонки, які потребують очищення або перетворення перед аналізом;
- Типи даних, що можуть вплинути на точність обробки (наприклад, числові дані, представлені у вигляді тексту).

2. Визначення розмірності датасету за допомогою print('Shape: ', df.shape). Цей метод виводить дві числові величини: кількість рядків (записів про покупки) та

кількість колонок (характеристик кожного запису). Для дослідника важливо розуміти співвідношення цих параметрів, оскільки воно визначає обсяг роботи та складність подальшого аналізу.

Таким чином, поєднання вибіркового перегляду даних із системним аналізом структури таблиці створює надійну основу для якісної підготовки даних і забезпечує коректність подальших аналітичних процедур.

[4]:
```
# Переглянемо всю інформацію про наш DataFrame
print(df.info())
print('Shape: ',df.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Customer ID             3900 non-null   int64  
 1   Age                     3900 non-null   int64  
 2   Gender                  3900 non-null   object 
 3   Item Purchased          3900 non-null   object 
 4   Category                3900 non-null   object 
 5   Purchase Amount (USD)   3900 non-null   int64  
 6   Location                3900 non-null   object 
 7   Size                    3900 non-null   object 
 8   Color                   3900 non-null   object 
 9   Season                  3900 non-null   object 
 10  Review Rating           3900 non-null   float64
 11  Subscription Status     3900 non-null   object 
 12  Shipping Type           3900 non-null   object 
 13  Discount Applied        3900 non-null   object 
 14  Promo Code Used         3900 non-null   object 
 15  Previous Purchases      3900 non-null   int64  
 16  Payment Method          3900 non-null   object 
 17  Frequency of Purchases  3900 non-null   object 
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
None
Shape:  (3900, 18)
```

Рис. 2.5 Результати виклику методів print(df.info()) та print('Shape: ', df.shape) з моїм коментарем

На цьому етапі було проведено ретельну перевірку даних, що підтвердила відсутність пропущених значень у всіх колонках, кожна з яких містить понад 3900 записів. Структура таблиці виявилася логічною та послідовною, а більшість змінних мають коректні типи даних (int, float, object), що суттєво спрощує подальшу обробку та аналіз інформації.

Після оцінки загальної структури даних наступним кроком стала оперативна перевірка розподілу категоріальних змінних:

```
[5]: # Переглянемо скільки було куплено речей кожного виду товарів:
     df["Item Purchased"].value_counts()
```

```
[5]: Item Purchased
     Blouse        171
     Jewelry       171
     Pants         171
     Shirt         169
     Dress         166
     Sweater       164
     Jacket        163
     Belt          161
     Sunglasses    161
     Coat          161
     Sandals       160
     Socks         159
     Skirt         158
     Shorts        157
     Scarf         157
     Hat           154
     Handbag       153
     Hoodie        151
     Shoes         150
     T-shirt       147
     Sneakers      145
     Boots         144
     Backpack      143
     Gloves        140
     Jeans         124
     Name: count, dtype: int64
```

Рис. 2.6 Код та результат коду з кількістю покупок кожного з товарів з моїм коментарем

Та кількості куплених речей кожного з розмірів:

```
[6]: df["Size"].value_counts()
```

```
[6]: Size
     M     1755
     L     1053
     S      663
     XL     429
     Name: count, dtype: int64
```

Рис. 2.7 Код та результат коду з кількістю куплених речей в кожному розмірі

В результаті проведеного аналізу розподілу категоріальних змінних було виявлено, які розміри товарів користуються найбільшим попитом, а які практично не мають покупців. Ця інформація є надзвичайно цінною при плануванні сегментації аудиторії та формуванні рекомендацій щодо оптимізації асортименту.

Після завершення повної первинної перевірки даних можна з упевненістю констатувати високу якість завантаженого датасету. Дані є повними, структурованими та не містять явних помилок чи пропусків, що створює надійну основу для подальшого глибокого розвідувального аналізу даних (Exploratory Data Analysis). Саме з цього етапу починається процес виявлення ключових

закономірностей, побудови інформативних візуалізацій та формування гіпотез, які становлять сутність аналітичної роботи.

2.4 Розвідувальний аналіз даних (РАД)

Розвідувальний аналіз даних (Exploratory Data Analysis, EDA) є фундаментальним і надзвичайно важливим етапом роботи з будь-яким набором даних. Його сутність полягає у всебічному ознайомленні з даними, вивченні їхньої структури, виявленні закономірностей та потенційних проблем, які можуть ускладнити подальший аналіз або побудову моделей.

Основні функції розвідкового аналізу даних можна окреслити таким чином:

1. Ознайомлення з даними - визначення розмірності набору (кількість рядків та стовпців), типів змінних, наявності пропущених або аномальних значень. Цей крок формує базове уявлення про структуру даних.
2. Описова статистика - розрахунок ключових статистичних показників (середнє, медіана, мінімум, максимум, розподіли), що дозволяє охарактеризувати основні властивості змінних.
3. Візуалізація - побудова графічних представлень даних (гістограми, діаграми розсіювання, коробкові діаграми), які сприяють кращому розумінню розподілів, трендів та взаємозв’язків між змінними.
4. Виявлення залежностей - застосування кореляційного аналізу та інших методів для ідентифікації взаємозв’язків між змінними та оцінки їхнього впливу одна на одну.
5. Пошук проблем - ідентифікація пропущених даних, дублікатів, аномалій або помилок, які потребують корекції або врахування у подальшій роботі.
6. Формування гіпотез - на основі отриманих результатів формулюються припущення щодо факторів, які можуть впливати на поведінку споживачів, та визначаються напрямки для подальшого дослідження.

Таким чином, розвідувальний аналіз даних закладає міцний фундамент для подальшого, більш глибокого аналізу, зокрема одновимірного, який стане наступним кроком у дослідженні.

2.5 Одновимірний аналіз

Першим кроком у проведенні одновимірного аналізу є розмежування даних за типами - числовими та категоріальними змінними, що дозволяє застосовувати до них відповідні методи обробки та аналізу.

У межах числових даних необхідно виключити стовпець «Customer ID», який є унікальним ідентифікатором кожного клієнта. Цей атрибут не містить інформації, релевантної для моделювання поведінки споживачів, і тому не використовується у подальшому аналізі.

Далі здійснюється вибірка лише тих стовпців, які містять числові ознаки. До них належать, зокрема, вік клієнта, сума покупки, рейтинг відгуку, кількість попередніх покупок та інші подібні параметри. Такий підхід дозволяє сфокусуватися на кількісних характеристиках, що є основою для подальшого статистичного аналізу та побудови моделей прогнозування:

Відокремимо числові дані:

[7]:
```
# Видаляємо непотрібний для статистики стовпець "Customer ID" із DataFrame, тоді вибираємо лише числові стовпці для наступних дій
numerical_features = df.drop('Customer ID', axis=1).select_dtypes(include=[np.number])
numerical_features.columns
```

[7]:
```
Index(['Age', 'Purchase Amount (USD)', 'Review Rating', 'Previous Purchases'], dtype='object')
```

Рис. 2.7 Код та результати коду для відокремлення числових ознак з моїм коментарем

В результаті проведених операцій було сформовано окремий набір даних, що містить виключно числові змінні, які є зручними та релевантними для подальшої роботи в контексті машинного навчання. Наступним логічним кроком є аналогічне відокремлення та обробка категоріальних ознак, що дозволить комплексно врахувати всі типи даних і забезпечити всебічний аналіз поведінкових характеристик споживачів:

Відокремимо категорії:

[8]:
```
# Вибираємо всі нечислові (категоріальні) стовпці DataFrame та виводимо:
categorical_features = df.select_dtypes(exclude=[np.number])
categorical_features.columns
```

[8]:
```
Index(['Gender', 'Item Purchased', 'Category', 'Location', 'Size', 'Color',
       'Season', 'Subscription Status', 'Shipping Type', 'Discount Applied',
       'Promo Code Used', 'Payment Method', 'Frequency of Purchases'],
      dtype='object')
```

Рис. 2.8 Код та результати коду для відокремлення категоріальних ознак з моїм коментарем

Розділивши початкову таблицю на дві окремі - з числовими та категоріальними даними - ми отримуємо можливість сфокусовано приступити до проведення описової статистики для числових змінних. Першим кроком є ознайомлення з основними характеристиками числових колонок у створеному піднаборі даних. Для цього формується короткий звіт, який містить ключові статистичні показники, що дозволяють оцінити розподіл, центральні тенденції та варіативність кожної змінної. Такий підхід забезпечує глибше розуміння структури даних і закладає основу для подальшого аналітичного дослідження:

[9]:
```
# Швиденько обчислюємо основні статистичні характеристики для числових стовпців
numerical_features.describe()
```

[9]:

	Age	Purchase Amount (USD)	Review Rating	Previous Purchases
count	3900.000000	3900.000000	3900.000000	3900.000000
mean	44.068462	59.764359	3.749949	25.351538
std	15.207589	23.685392	0.716223	14.447125
min	18.000000	20.000000	2.500000	1.000000
25%	31.000000	39.000000	3.100000	13.000000
50%	44.000000	60.000000	3.700000	25.000000
75%	57.000000	81.000000	4.400000	38.000000
max	70.000000	100.000000	5.000000	50.000000

Рис. 2.9 Код та результати коду з основними характеристиками числових ознак з моїм коментарем

Для подальшого візуального аналізу числових змінних було обрано створення коробкових діаграм (boxplots) для кожної колонки. Цей метод візуалізації, рекомендований моїм колегою-аналітиком Олексієм, є надзвичайно ефективним інструментом для швидкої і наочної оцінки розподілу даних. Коробкові діаграми дозволяють виявити наявність «стрибків» або «викидів» - аномальних значень, які можуть суттєво вплинути на результати подальшого аналізу. Завдяки такому підходу можна своєчасно ідентифікувати потенційні проблеми у даних та прийняти обґрунтовані рішення щодо їх обробки, що підвищує якість і надійність аналітичних висновків:

[10]:
```
def plot_boxplots(data, columns, rows):
# Створюємо багатопанельний графік для покращення виводу:
    fig = make_subplots(rows=rows, cols=columns, subplot_titles=[f'Діаграма розмаху {col}' for col in data.columns])

    for i, column in enumerate(data.columns):
        row = (i // columns) + 1 # Визначення номера рядка
        col = (i % columns) + 1 # Визначення номера колонки

        data_ = data[column].dropna() # Видаляємо NaN-значення, щоб уникнути помилок

        # Додаємо діаграму розмаху (коробковий графік), яка показує розподіл значень у стовпці
        fig.add_trace(go.Box(y=data_, name=f'{column} коробковий графік', marker=dict(color='orange',
        line=dict(color='black', width=1))), row=row, col=col)

    # Оновлюємо макет
    fig.update_layout(height=1000, width=1000, title_text="Діаграми розмаху", showlegend=False)
    fig.show()
```

Рис. 2.10 Функція для створення коробкових діаграм для кожної числової ознаки з моїми коментарями

Для візуалізації описаної функції у вигляді діаграм необхідно виконати відповідний код, який побудує графіки розмаху для числових ознак:

[11]:
```
# Будуємо boxplot-діаграму для числових ознак:
plot_boxplots(numerical_features, columns=2, rows=2)
```

Рис. 2.11 Вивід діаграми розмаху для числових ознак з моїм коментарем

Внаслідок застосування цього підходу отримуємо чіткі та інформативні діаграми розмаху, які наочно демонструють розподіл даних, наявність потенційних викидів і варіативність показників. Такий спосіб подання інформації є ефективним інструментом для подальшого аналізу і прийняття обґрунтованих рішень у дослідженні:

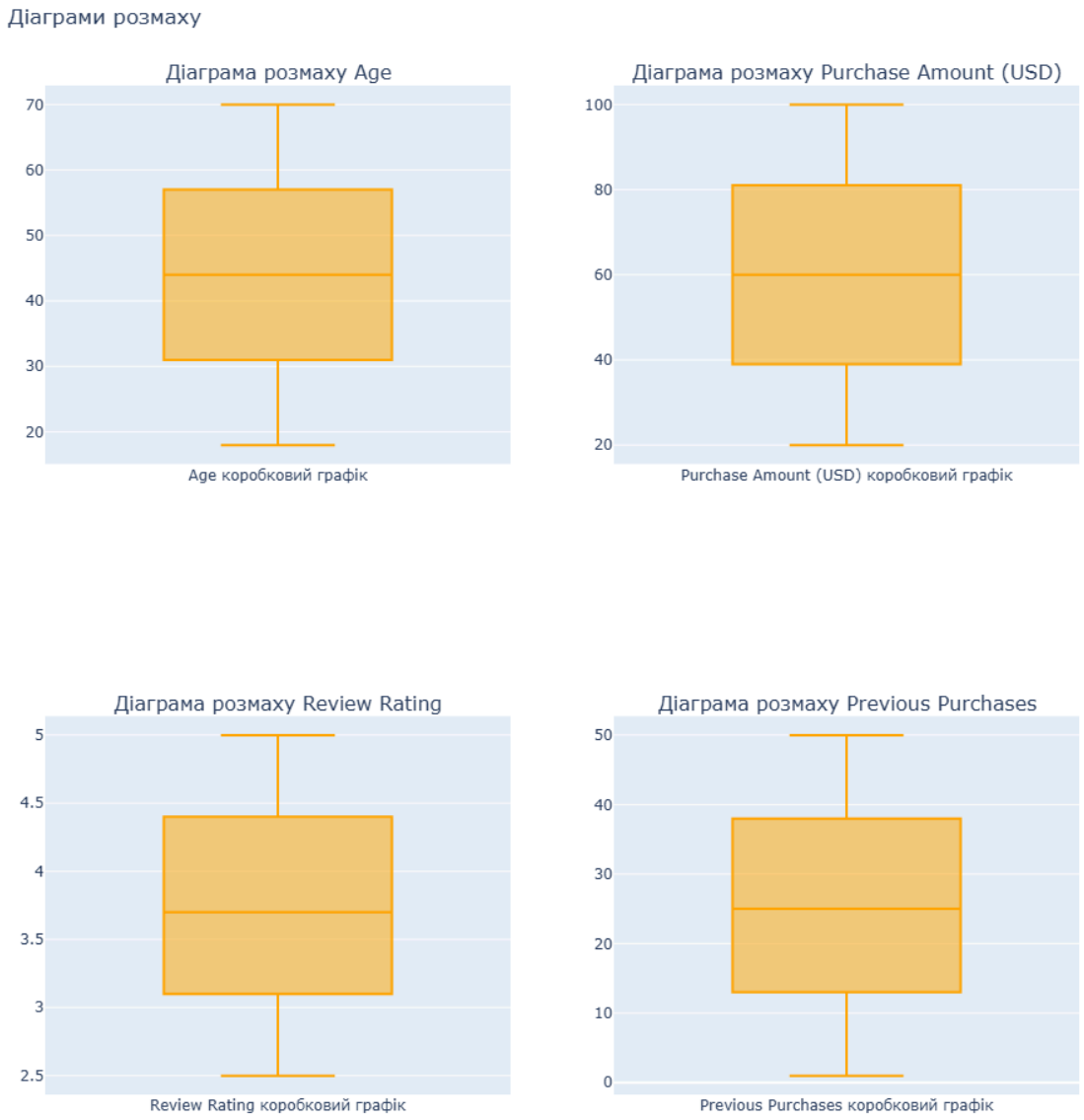


Рис. 2.12 Готові коробкові діаграми для числових ознак

Аналіз створених коробкових діаграм дозволив зробити низку важливих спостережень:

1. З першої діаграми розмаху видно, що більшість покупців належать до вікової групи приблизно від 30 до 60 років. Медіана, позначена темно-оранжевою лінією всередині «коробки», розташована близько до середини інтервалу й становить приблизно 45 років. Хоча серед клієнтів є як молодші, так і старші особи, суттєвих викидів у віковому розподілі не виявлено.

2. Друга діаграма демонструє, що основна маса покупок здійснюється у ціновому діапазоні від 40 до 80 доларів, із середнім значенням близько 60 доларів.

Існують покупки як на менші, так і на більші суми, проте більшість транзакцій зосереджена саме в цьому інтервалі.

3. Аналіз третьої діаграми, що відображає рейтинги відгуків, свідчить, що оцінки переважно коливаються в межах від 3 до 4,5, при цьому медіана знаходиться трохи вище 3,5. Це свідчить про те, що більшість клієнтів залишають позитивні відгуки, а низьких оцінок небагато.

4. Четверта діаграма розкриває, що більшість покупців мають досвід від 13 до 38 попередніх покупок, із медіаною близько 25. Хоча зустрічаються як нові клієнти з невеликою кількістю покупок, так і активні покупці з великим їх числом, вони є винятками.

Отже, на основі цих даних можна зробити попередній висновок, що основна аудиторія магазину X - це люди середнього віку, які здійснюють покупки на середні суми, залишають переважно позитивні відгуки та мають певний досвід взаємодії з магазином. Втім, не варто поспішати з остаточними висновками, адже для більш глибокого розуміння поведінки споживачів доцільно застосувати додаткові методи візуалізації, зокрема гістограми, які дозволять детальніше дослідити розподіл числових змінних:

[12]:
```python
# Будуємо гістограми для всіх стовпців з числовими значеннями і додаємо на кожен графік середнє значення та медіану
def plot_distributions(data, columns, rows):

    data = data.replace([np.inf, -np.inf], np.nan).dropna()
    # Створюємо фігуру з підсюжетами
    fig, axes = plt.subplots(rows, columns, figsize=(15, 10))
    axes = axes.flatten()

    for i, column in enumerate(data.columns):
        ax = axes[i]

        sns.histplot(data[column], kde=True, ax=ax, color='orange', bins=30, edgecolor='black', linewidth=0.5)

        # Обчислюємо середнє значення та медіану
        mean_value = data[column].mean()
        median_value = data[column].median()

        # Додаємо вертикальні лінії для середнього значення та медіани
        ax.axvline(mean_value, color='red', linestyle='--', label=f'Середнє значення: {mean_value:.2f}')
        ax.axvline(median_value, color='green', linestyle='-', label=f'Медіана: {median_value:.2f}')

        # Додаємо назву сюжету та позначок
        ax.set_title(f'Розподіл {column}', fontsize=14)
        ax.set_xlabel(column)
        ax.set_ylabel('Частота')

        # Додаємо легенду
        ax.legend()

    # Візуалізуємо
    plt.tight_layout()
    plt.show()
```

Рис. 2.12 Код для створення гістограми з моїми коментарями

```
[13]: # Будуємо гістограми для числових ознак:
plot_distributions(numerical_features, columns=2, rows=2)
```

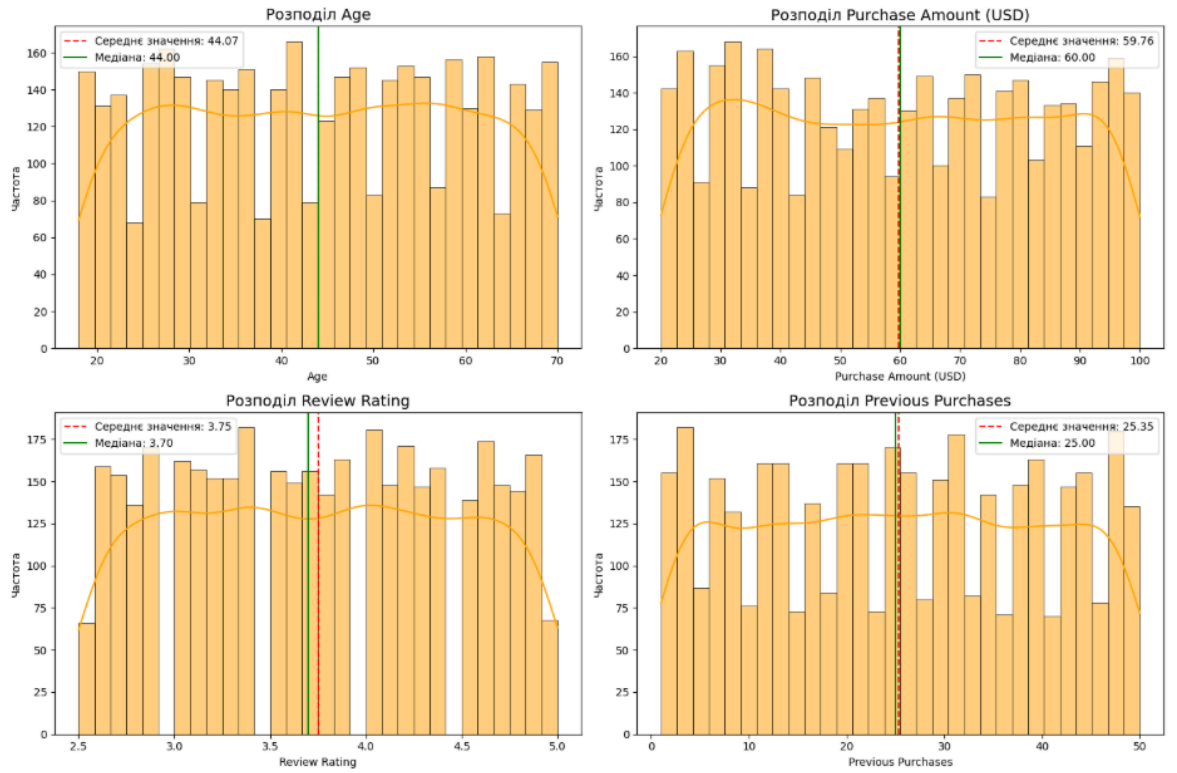


Рис. 2.13 Вигляд готових гістограм

Аналіз новостворених гістограм підтверджує раніше висловлені спостереження щодо числових ознак, отримані за допомогою коробкових діаграм. Зокрема, можна впевнено констатувати, що:

1. Усі числові змінні мають відносно центрований розподіл, при якому середнє значення та медіана розташовані близько одна до одної. Це свідчить про симетричний або майже нормальний характер розподілу даних.

2. Візуалізація не виявляє суттєвих викидів чи екстремальних відхилень, що означає, що більшість значень знаходиться в межах очікуваного діапазону без різких стрибків або аномалій.

Такий характер розподілу є сприятливим для подальшого застосування статистичних методів і моделей машинного навчання, оскільки відсутність значних викидів знижує ризик спотворення результатів та підвищує надійність аналітичних висновків.

Після успішного завершення аналізу числових ознак настав час перейти до дослідження категоріальних змінних. Аналогічно до попереднього етапу, першим

кроком є ознайомлення з новоствореним піднабором даних, що містить категоріальні колонки. Для цього формується короткий звіт, який відображає основні характеристики цих змінних - їх кількість, унікальні значення, частоти та інші ключові параметри. Такий огляд дозволяє отримати загальне уявлення про структуру категоріальних даних, виявити можливі аномалії чи нерівномірності у розподілі, а також визначити подальші напрямки їх обробки та аналізу:

Описова статистика для категоріальних ознак

[14]:
```
# Швиденько обчислюємо основні статистичні характеристики вже для категорій:
categorical_features.describe()
```

[14...

	Gender	Item Purchased	Category	Location	Size	Color	Season	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Payment Method	Frequency of Purchases
count	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900	3900
unique	2	25	4	50	4	25	4	2	6	2	2	6	7
top	Male	Blouse	Clothing	Montana	M	Olive	Spring	No	Free Shipping	No	No	PayPal	Every 3 Months
freq	2652	171	1737	96	1755	177	999	2847	675	2223	2223	677	584

Рис. 2.14 - Код та результати коду з основними характеристиками категоріальних ознак з моїм коментарем

Враховуючи результати проведеного аналізу та значну кількість категоріальних змінних текстового типу, доцільність використання коробкових діаграм для їх візуалізації є обмеженою. З огляду на це, було прийнято рішення застосувати стовпчикові діаграми, відомі також як діаграми Ганта. Цей метод є ефективним, зручним та наочним інструментом для представлення категоріальних даних, що дозволяє одночасно оцінити розподіл багатьох змінних.

Стовпчикові діаграми сприяють кращому розумінню структури даних, допомагають виявити найбільш популярні та рідкісні категорії, а також дозволяють простежити ключові тенденції у поведінці споживачів. Саме на цьому етапі аналізу відкриваються важливі інсайти, які можуть стати основою для формування рекомендацій щодо оптимізації асортименту та маркетингових стратегій магазину.

```python
# Якщо говорити коротко, то тут функція plot_barplots будує стовпчикові діаграми (bar chart) для кожного стовпця переданих даних,
# розташовуючи їх у сітці (rows × columns). Використовуємо Plotly для візуалізації та кольорову шкалу YlGn для створення красивеньких барів
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.colors as pc

def plot_barplots(data, columns, rows):
    fig = make_subplots(rows=rows, cols=columns, subplot_titles=[f'Діаграма Ганта {col}' for col in data.columns])

    colorscale = pc.sequential.YlGn

    for i, column in enumerate(data.columns):
        row = (i // columns) + 1 # Визначення номера рядка
        col = (i % columns) + 1 # Визначення номера колонки
        data_ = data[column].dropna().value_counts()

        # Створюємо список кольорів для кожної категорії
        color_list = [colorscale[j % len(colorscale)] for j in range(len(data_))]

        fig.add_trace(go.Bar(x=data_.index.astype(str), y=data_, name=f'{column} бар-чарт',
                             marker=dict(color=color_list, line=dict(color='black', width=1))),
                      row=row, col=col)

    fig.update_layout(height=1250, width=1250, title_text="Стовбчикові горизонтальні діаграми", showlegend=False)
    fig.show()
```

```python
# Будуємо горизонтальні діаграми (їх ще називають діаграмами Ганта):
plot_barplots(categorical_features,3,5)
```

Рис. 2.15 Код для створення діаграми Ганта для категоріальних ознак з моїми коментарями

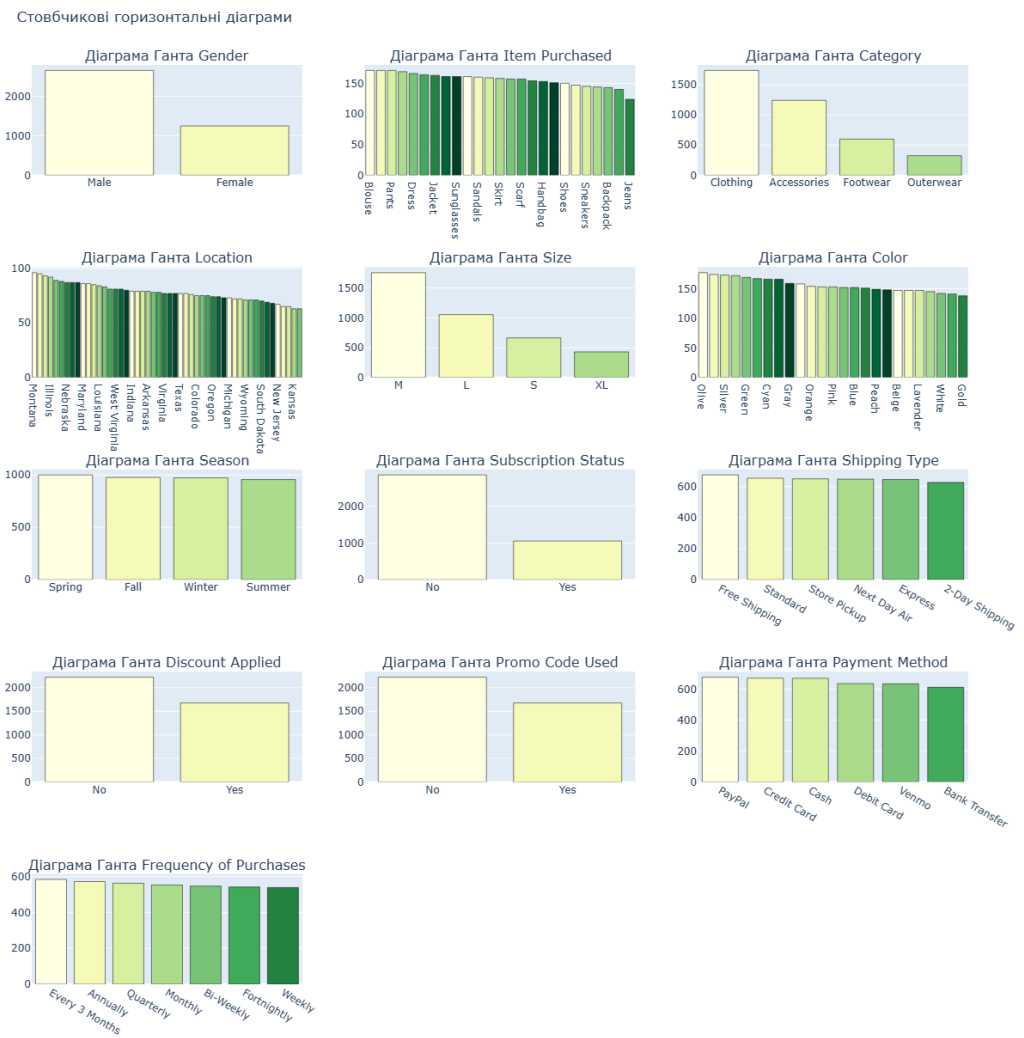


Рис. 2.16 Вигляд готових стовбчикових діаграм

Аналіз створених стовпчикових діаграм дозволяє сформувати низку важливих висновків щодо поведінки покупців та особливостей асортименту магазину:

- Перша діаграма свідчить про те, що серед покупців переважають чоловіки, що вказує на виражену перевагу певної демографічної групи.
- Друга діаграма демонструє, що найпопулярнішим товаром є блузки, тоді як джинси користуються найменшим попитом, що відображає конкретні вподобання споживачів.
- Третя діаграма підтверджує, що категорія «одяг» є домінуючою серед покупок, що визначає основний напрямок комерційної діяльності магазину.
- Четверта діаграма показує, що найбільший обсяг продажів припадає на штат Канзас, тоді як у Монтані спостерігається найнижчий рівень покупок, що має важливе значення для регіонального маркетингу та таргетингу.
- П’ята діаграма вказує на те, що середній розмір (M) є найбільш затребуваним, що дозволяє оптимізувати асортимент і зменшити витрати на менш популярні розміри.
- Шоста діаграма демонструє, що сріблястий колір є трендовим, що важливо враховувати при закупівлях і рекламних кампаніях.
- Сьома діаграма свідчить про рівномірний розподіл покупок за сезонами, що свідчить про відсутність вираженої сезонності.
- Восьма діаграма показує, що користувачі без підписки купують більше, ніж підписані, що є цікавим трендом для роботи з лояльністю.
- Дев’ята діаграма відображає майже рівномірне використання усіх доступних способів оплати, без явного лідера.
- Десята діаграма демонструє, що всі типи доставки використовуються приблизно однакову кількість разів, близько 600.
- Одинадцята діаграма вказує, що більшість транзакцій здійснюється без знижок, що може свідчити про особливості цінової політики або поведінкові звички покупців.

- Дванадцята діаграма підтверджує, що більшість покупок відбувається без використання промо-кодів.
- Тринадцята діаграма свідчить про рівномірний розподіл бажань щодо способів оплати серед клієнтів.
- Чотирнадцята діаграма демонструє рівномірний розподіл частоти покупок між категоріями, кожна з яких має близько 600 транзакцій.

Отже, на основі проведеного аналізу категоріальних ознак можна зробити висновок, що найбільший вплив на збільшення обсягів продажів мають такі фактори, як стать покупців, популярність конкретних товарів, регіональні особливості, а також переваги щодо розмірів і кольорів. Ці тенденції дають чітке розуміння пріоритетів для маркетологів і допомагають оптимізувати асортимент перед закупівлею з метою підвищення ефективності продажів.

Водночас у деяких аспектах поведінка покупців є більш рівномірною. Зокрема, це стосується способів оплати, типів доставки та сезонності, що свідчить про відсутність виражених трендів у цих сферах і вказує на їхню стабільність та менший вплив на динаміку продажів.

Наступним етапом дослідження стане більш глибокий порівняльний аналіз, зокрема двовимірний, який дозволить виявити взаємозв’язки між різними змінними та уточнити ключові чинники, що впливають на поведінку споживачів і результати діяльності магазину.

2.6 Двовимірний аналіз

Двовимірний аналіз є важливим інструментом, який широко застосовується аналітиками і лежить в основі багатьох сучасних аналітичних алгоритмів. Його суть полягає у порівнянні двох ознак з метою виявлення взаємозв’язків між ними. Такий підхід дозволяє глибше зрозуміти, як одна змінна впливає на іншу, наприклад, як стать покупця впливає на середню суму покупки або як частота покупок змінюється залежно від віку чи регіону. Завдяки цьому аналізу можна краще охарактеризувати поведінку окремих груп споживачів і точніше сформулювати рекомендації для маркетингових стратегій.

Для початку дослідження я зосереджуюсь на вивченні впливу віку на суму покупки. Цей зв’язок буде проаналізовано за допомогою діаграми розсіювання, що наочно відображає співвідношення між двома числовими змінними, а також за допомогою кореляційного аналізу Спірмена, який дозволяє виміряти силу та напрямок монотонної залежності між цими показниками.

Спершу побудуємо діаграму розсіювання, що стане наочним інструментом для візуального вивчення взаємозв’язку між віком покупця та сумою його покупки. Це дозволить виявити закономірності, тренди або аномалії, які можуть бути важливими для подальшого аналізу та прийняття рішень:

```
[17]: # Будуємо діаграму розсіювання (scatter plot) для змінних Age і Purchase Amount (USD) з лінією регресії за допомогою sns.regplot().
      # Точки (покупки) мають помаранчевий колір із прозорістю (alpha=0.5), а лінія регресії - темно-помаранчева.
      # Написання цього коду допомагає візуалізувати взаємозв'язок між віком і сумою покупки.
      sns.regplot(x='Age', y='Purchase Amount (USD)', data=df,
                  scatter_kws={'color': 'orange', 'alpha': 0.5},  # Колір точок
                  line_kws={'color': 'darkorange'})  # Колір лінії

      plt.title("Вплив віку на суму купівлі", fontsize=14)
      plt.xlabel('Age', fontsize=12)
      plt.ylabel('Purchase Amount (USD)', fontsize=12)
      plt.show()
```

Рис. 2.17 Код з коментарями для побудування діаграми розсіювання для змінних Age і Purchase Amount.

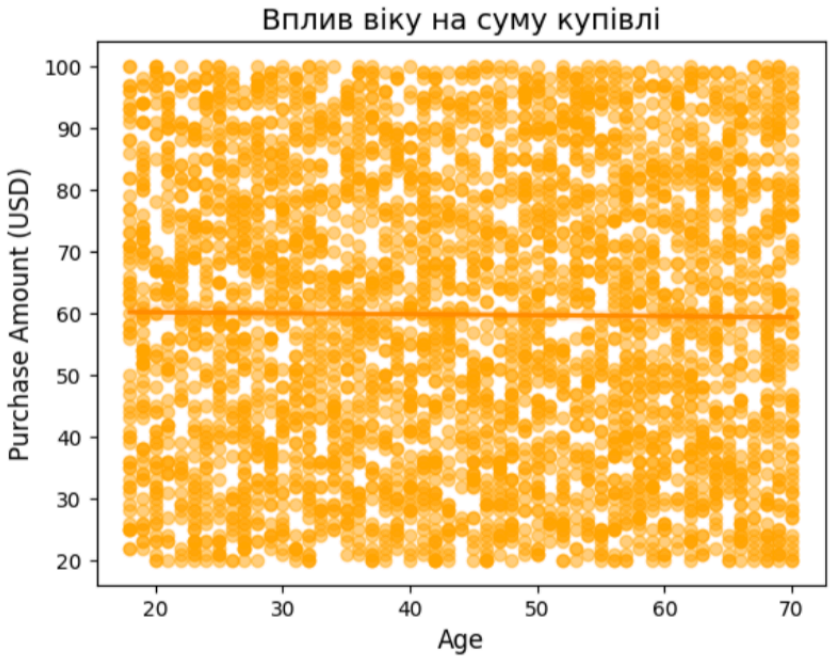


Рис. 2.18 Вигляд готової діаграми розсіювання.

Аналіз отриманої діаграми розсіювання дозволяє зробити висновок, що вік покупця не має суттєвого впливу на суму здійсненої покупки у цьому магазині. Це

свідчить про те, що рекламні кампанії або спеціальні пропозиції, спрямовані на збільшення середнього чека, не потребують орієнтації на певні вікові групи.

Для більш обґрунтованого підтвердження цього спостереження доцільно застосувати кореляційний аналіз за методом Спірмена. Цей статистичний показник дозволяє оцінити силу та напрямок монотонної залежності між двома змінними, незалежно від їх розподілу. Розрахунок кореляції Спірмена дасть змогу кількісно визначити ступінь взаємозв'язку між віком клієнтів та сумою їхніх покупок, що є важливим для прийняття обґрунтованих управлінських рішень:

```python
# Оцінюємо, чи є взаємозв'язок між віком і сумою покупки та наскільки він сильний

from scipy.stats import spearmanr

# Рахуємо кореляцію Спірмена:
spearman_corr, p_value = spearmanr(df['Age'], df['Purchase Amount (USD)'])
print(f"Кореляція Спірмена: {spearman_corr:.2f}, p-значення: {p_value:.3f}")
```

```
Кореляція Спірмена: -0.01, p-значення: 0.514
```

Рис. 2.17 Код та результати коду кореляції Спірмена з моїми коментарями.

Результати розрахунку кореляції Спірмена підтверджують попередні висновки: вік покупця не має суттєвого впливу на суму його витрат у даному магазині. Спостерігається широкий діапазон сум покупок серед усіх вікових груп без виражених тенденцій чи закономірностей. Цей факт є важливим для стратегічного планування, оскільки свідчить про відсутність доцільності орієнтувати рекламні кампанії чи спеціальні пропозиції виключно на певні вікові категорії з метою збільшення середнього чека.

Наступним кроком у дослідженні є вивчення можливих залежностей між статтю покупця та сумою його покупки. Для цього доцільно застосувати коробкові діаграми, які вже успішно використовувалися раніше, оскільки вони дозволяють наочно оцінити розподіл сум у різних групах та виявити потенційні відмінності чи аномалії. Такий підхід сприятиме глибшому розумінню поведінкових особливостей різних демографічних сегментів клієнтів:

```
[19]:
# Порівняймо розподіл витрат між чоловіками та жінками
fig = px.box(df, x='Gender', y='Purchase Amount (USD)', title='Вплив статі на суму купівлі')
fig.update_traces(marker_color='orange', line_color='orange')
fig.update_layout(width=1000)
fig.show()
```

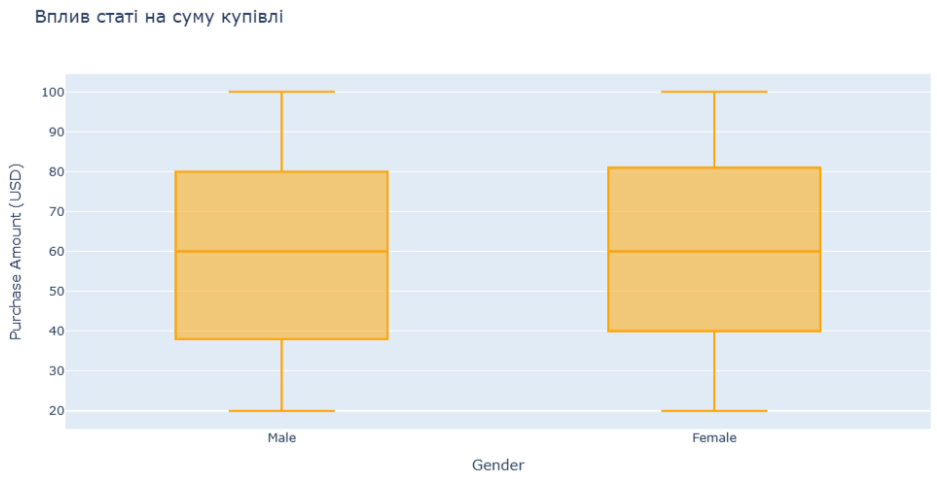


Рис. 2.19 Код та результати коду у вигляді готових коробкових графіків з моїм коментарем

Аналіз готових коробкових діаграм свідчить про відсутність суттєвих відмінностей у сумі покупок між чоловіками та жінками. Медіани обох груп розташовані на приблизно однаковому рівні, а розміри «коробок» та межі «вусів» (мінімальні та максимальні значення) також майже співпадають. Це свідчить про те, що в контексті даного магазину немає вираженого гендерного тренду: як чоловіки, так і жінки витрачають приблизно однакові суми на покупки.

Для більш обґрунтованого підтвердження цього спостереження доцільно застосувати статистичний критерій Манна-Уітні. Цей непараметричний тест дозволяє оцінити, чи існує статистично значуща різниця між двома незалежними вибірками, у нашому випадку - сумами покупок чоловіків і жінок. Використання цього методу допоможе визначити, чи можна вважати відсутність різниці випадковим збігом, чи вона є суттєвою з точки зору статистичного аналізу:

```
[20]:
from scipy.stats import mannwhitneyu
u_statistic, p_value = mannwhitneyu(df[df['Gender'] == 'Male']['Purchase Amount (USD)'],
                                    df[df['Gender'] == 'Female']['Purchase Amount (USD)'])
print(f"Критерій Манна-Уітні U: U-статистика= {u_statistic:.2f}, p-значення = {p_value:.3f}")
```

```
Критерій Манна-Уітні U: U-статистика= 1625536.50, p-значення = 0.372
```

Рис. 2.20 Код та результати коду критерія Манна-Уітні.

Результати статистичного тесту свідчать, що p-значення значно перевищує поріг 0,05, що свідчить про відсутність статистично значущої різниці між сумами покупок чоловіків і жінок. Іншими словами, немає підстав стверджувати, що одна з груп витрачає суттєво більше або менше за іншу, що повністю підтверджує попередні висновки, зроблені на основі аналізу коробкових діаграм.

Наступним об’єктом дослідження стане можливий вплив рейтингу відгуків на суму покупки. Для цього знову застосуємо діаграму розсіювання, яка дозволить візуально оцінити взаємозв’язок між цими двома змінними, а також розрахуємо кореляцію Спірмена, що дасть кількісну характеристику сили та напрямку монотонної залежності між рейтингом відгуків та сумою покупок. Такий підхід сприятиме більш глибокому розумінню факторів, що впливають на поведінку споживачів:

[21]:
```
# Оцінюємо, чи існує взаємозв'язок між рейтингами відгуків і сумою витрат покупців:
sns.regplot(x='Review Rating', y='Purchase Amount (USD)', data=df, scatter_kws={'color': 'orange', 'alpha': 0.5},  # Колір точок
            line_kws={'color': 'darkorange'})  # Колір лінії)
plt.title('Вплив рейтингів відгуків на суму купівлі')
plt.xlabel('Review Rating')
plt.ylabel('Purchase Amount (USD)')
plt.show()
```

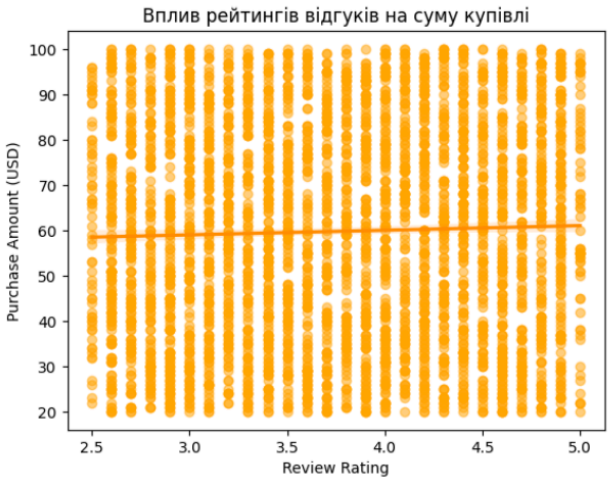


Рис. 2.21 Код та результати коду з готовою діаграмою розсіювання

[22]:
```
# Розраховуємо кореляцію Спірмена:
spearman_corr, p_value = spearmanr(df['Review Rating'], df['Purchase Amount (USD)'])
print(f"Кореляція Спірмена: {spearman_corr:.2f}, p-значення: {p_value:.3f}")
```

```
Кореляція Спірмена: 0.03, p-значення: 0.058
```

Рис. 2.22 Код та результати коду кореляції Спірмена з моїм коментарем

Аналізуючи результати, спочатку звертаю увагу на графічне представлення даних, де точки розташовані хаотично, без вираженого тренду чи нахилу. Це свідчить про відсутність очевидної залежності між оцінкою відгуку покупця та сумою його покупки: незалежно від того, чи залишена оцінка дорівнює 5 або 2,5, сума покупки може варіюватися від 20 до 100 доларів і більше.

Ці спостереження підтверджує розрахунок кореляції Спірмена, значення якої становить 0,03 - що свідчить про дуже слабкий або взагалі відсутній зв'язок між цими двома змінними. Отже, можна зробити висновок, що рейтинг відгуків та сума покупки є незалежними характеристиками у цьому магазині. Відсутність закономірностей означає, що бізнесу не варто очікувати, що більші суми покупок автоматично призводять до кращих оцінок, або навпаки - що низькі оцінки пов'язані з невеликими чеками.

Наступним питанням, яке заслуговує на увагу, є можливий вплив віку покупців на частоту їхніх покупок у магазині. Для цього необхідно розподілити клієнтів за відповідними віковими групами, створивши новий категоріальний стовпець, наприклад, «Age Group». Лише після цього можна буде побудувати інформативну стовпчикову діаграму, що дозволить наочно оцінити розподіл частоти покупок серед різних вікових категорій та виявити можливі тенденції:

[23]:
```
# Визначаємо вікові групи та мітки
bins = [18, 25, 35, 45, 55, 65, 70]
labels = ['18-24', '25-34', '35-44', '45-54', '55-64', '65-70']

# Створюємо нову колонку 'Age Group'
df['Age Group'] = pd.cut(df['Age'], bins=bins, labels=labels, include_lowest=True)
```

Рис. 2.23 Код для створення нового стовбця таблиці Age Group з віковими групами

Будуємо стовбчикову діаграму:

```
[24]: # Будуємо стовпчикову діаграму задля того, щоб переглянути як частота покупок змінюється в залежності від вікової групи
sns.countplot(x='Age Group', hue='Frequency of Purchases', data=df, palette='YlGn', edgecolor='black', linewidth=0.4)
sns.set(style='whitegrid')
plt.title('Вплив віку на частоту купівель')
plt.xticks(rotation=45, ha='right')
plt.show()
```

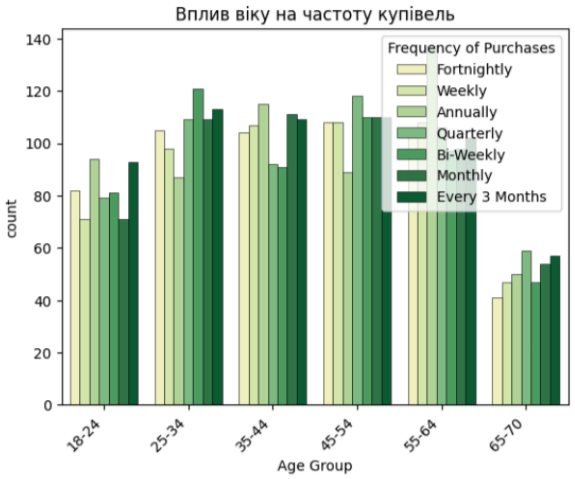


Рис. 2.24 Код та результати коду з готовою стовбчиковою діаграмою для вікових груп з моїм коментарем

Аналіз отриманої діаграми дозволяє виявити низку важливих тенденцій у поведінці покупців цього магазину:

1. Вікові групи 25-34, 35-44 та 45-54 років демонструють найвищу активність у кількості покупок за всіма розглянутими періодами (тиждень, місяць, квартал тощо). Особливо помітною є група 35-44 років, де спостерігається стабільно висока частота покупок у всіх категоріях.

2. Наймолодша та найстарша вікові категорії характеризуються найнижчими показниками купівельної активності в усіх часових проміжках.

3. Спостерігається чітка тенденція до зниження частоти покупок із зростанням віку, що стає особливо помітним після 55 років.

Отже, на основі цих спостережень можна зробити висновок, що основною цільовою аудиторією магазину є особи віком від 25 до 54 років, які проявляють найбільшу активність у здійсненні покупок незалежно від частоти. Водночас молодші та старші вікові групи є менш залученими, що вказує на необхідність розробки спеціалізованих маркетингових стратегій для підвищення їхньої зацікавленості та активності.

Для подальшого дослідження доцільно проаналізувати вплив сезонності на популярність різних категорій товарів. У цьому контексті ефективним інструментом візуалізації стане кругова діаграма, яка дозволить наочно відобразити співвідношення продажів за категоріями у різні пори року та виявити сезонні тренди:

[25]:
```python
# Фіксована палітра для сезонів
season_colors = {
    'Spring': '#66BB6A',  # світло-зелений
    'Summer': '#FFEB3B',  # теплий жовтогарячий
    'Fall':   '#FFA726',  # рудо-коричневий
    'Winter': '#64B5F6'   # блакитний
}

# Перебір по категоріях
for category in df['Category'].unique():
    category_data = df[df['Category'] == category]

    fig = px.pie(
        category_data,
        names='Season',
        title=f'Найпоширеніші сезони для категорії: {category}',
        color='Season',
        color_discrete_map=season_colors
    )

    fig.update_traces(
        textposition='inside',
        textinfo='percent+label',
        marker=dict(line=dict(color='black', width=0.2))
    )

    fig.update_layout(
        title_font_size=18,
        legend_title='Сезон',
        legend=dict(orientation="h", y=-0.1)
    )

    fig.show()
```

Рис. 2.25 Код для створення кругових діаграм

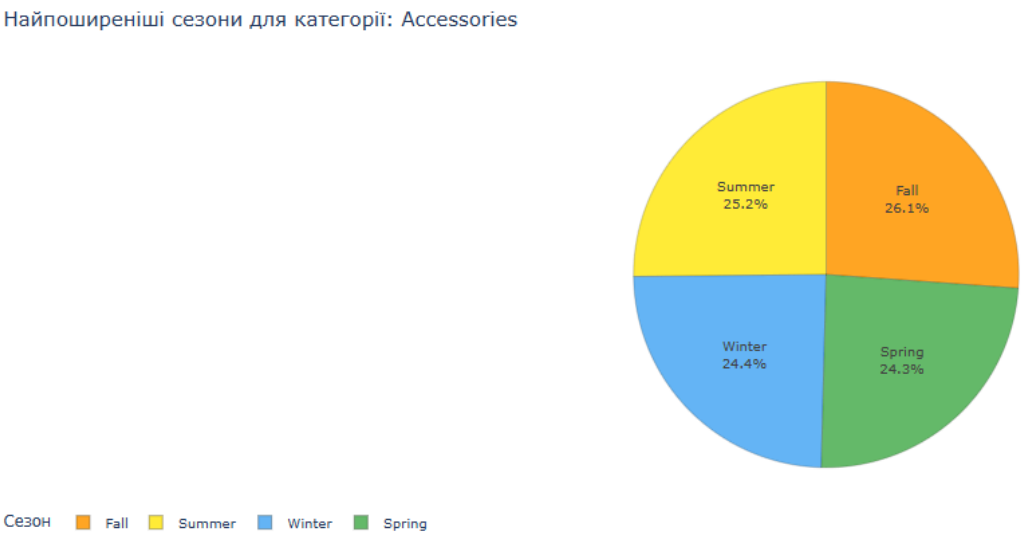


Рис. 2.26 Результати створення кругової діаграми для категорії «Accessories»

Після побудови кругової діаграми одразу можна зробити низку важливих спостережень. По-перше, для категорії аксесуарів у цьому магазині не спостерігається вираженої сезонної переваги. Хоча найбільша частка продажів припадає на осінній період, вона лише незначно - приблизно на 1% - перевищує показники інших сезонів. Це свідчить про те, що попит на аксесуари є стабільним протягом усього року.

Такий рівномірний розподіл продажів вказує на високу актуальність цієї категорії товарів незалежно від пори року. Для магазину це означає можливість планувати закупівлі та маркетингові активності без значних сезонних коливань, що сприяє більш ефективному управлінню асортиментом і ресурсами.

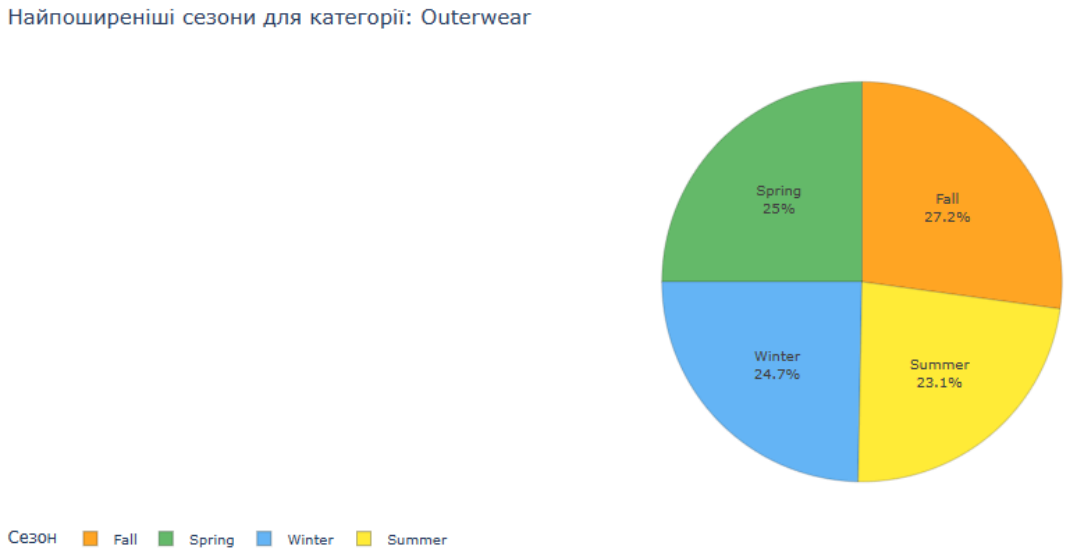


Рис. 2.27 Результати створення кругової діаграми для категорії «Outerwear»

Аналізуючи представлену діаграму, можна зробити висновок, що осінь є найпопулярнішим сезоном для придбання верхнього одягу в цьому магазині, що цілком логічно з огляду на погодні умови та сезонні потреби споживачів. Водночас весна та зима також демонструють високий рівень попиту на цю категорію товарів.

Враховуючи ці тенденції, магазину доцільно готуватися до пікових продажів верхнього одягу саме в осінній період, не ігноруючи при цьому значущість весняного та зимового сезонів. Влітку ж, навпаки, рекомендується зменшити

обсяги закупівель цієї категорії та зосередитися на розпродажах залишків, що дозволить оптимізувати товарні запаси та мінімізувати витрати.

Отже, ключова тенденція полягає в тому, що осінь є провідним сезоном для продажу верхнього одягу, проте високий попит зберігається і в весняно-зимовий період, що слід враховувати при плануванні асортименту та маркетингових активностей.

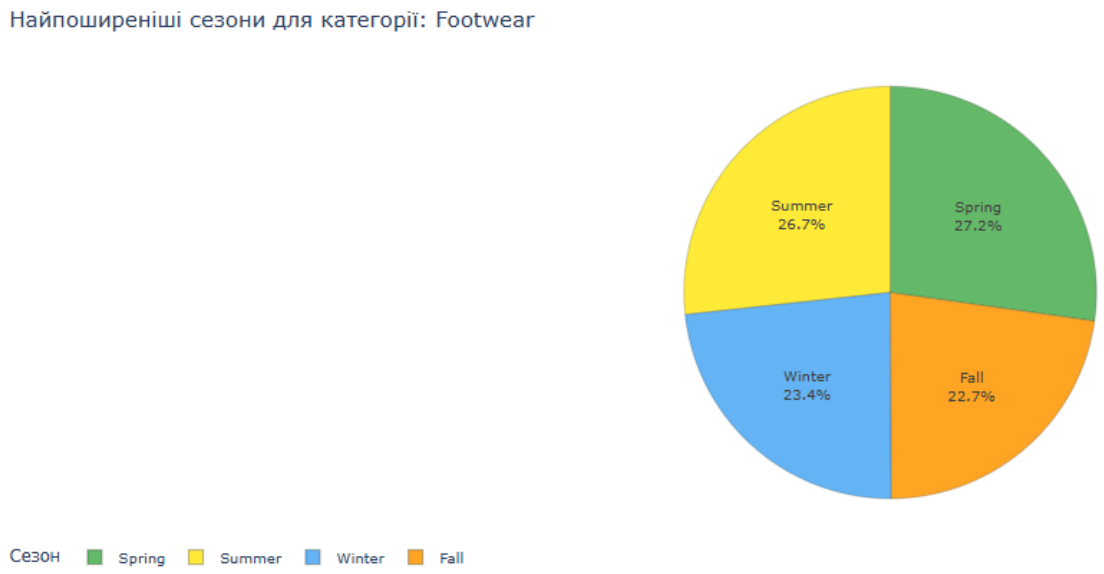


Рис. 2.28 Результати створення кругової діаграми для категорії «Footwear»

Аналізуючи цю діаграму, одразу помітно, що весна та літо є найпопулярнішими сезонами для придбання взуття в цьому магазині, тоді як зима та осінь демонструють менш активний попит. Така сезонність цілком відповідає загальним тенденціям ринку, де теплі місяці сприяють більшій активності покупців у сегменті взуття.

Це означає, що маркетологам і власникам магазину варто враховувати сезонні коливання попиту, підтримуючи різноманітний і актуальний асортимент протягом усього року. Зокрема, весняно-літній період слід розглядати як ключовий для активного просування взуття, водночас не ігноруючи потреби покупців у холодніші сезони, коли попит, хоч і знижується, але залишається стабільним.

Таким чином, збалансоване планування закупівель і маркетингових кампаній із урахуванням сезонності дозволить забезпечити ефективне управління асортиментом та максимізувати продажі упродовж року.

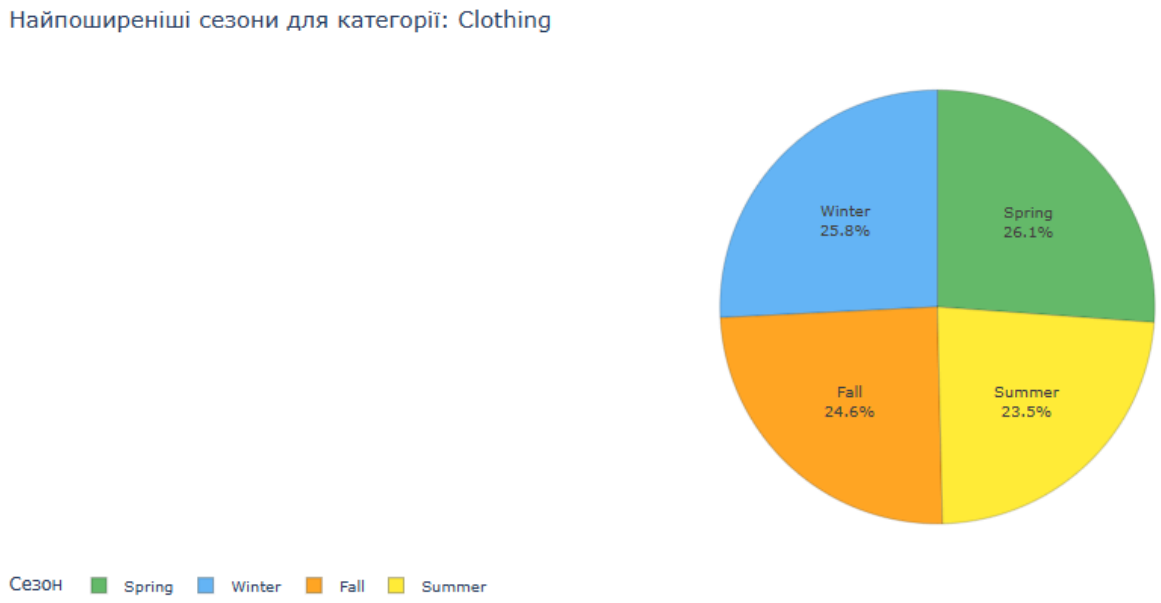


Рис. 2.29 Результати створення кругової діаграми для категорії «Clothing»

Аналізуючи цю кругову діаграму, одразу помітно, що попит на одяг у цьому магазині розподілений досить рівномірно між сезонами. Хоча весняний період демонструє дещо більшу популярність порівняно з іншими сезонами, ця перевага не є суттєвою. Найменш активним сезоном для продажу одягу виявився літній період.

Такі результати мають важливе значення для маркетологів, оскільки свідчать про необхідність формування спеціальних акцій та промоцій у менш активні сезони з метою стимулювання продажів. Врахування цих сезонних коливань дозволить збалансувати товарообіг і підвищити ефективність маркетингових заходів.

Наступним кроком стане дослідження впливу частоти застосування знижок у різних вікових групах покупців. Для цього буде використано раніше створений категоріальний стовпець «Age Group», що дозволить провести більш детальний аналіз поведінки клієнтів залежно від їх віку та схильності користуватися знижками:

```
[26]: # Будуємо графік, що покаже скільки покупок з різними знижками було зроблено в кожній віковій групі.
plt.figure(figsize=(10, 6))
sns.countplot(x='Age Group', hue='Discount Applied', data=df, palette='YlGn', edgecolor='black', linewidth=0.5)
plt.title('Використання знижок за віковою групою')
plt.xticks(rotation=45, ha='right')
plt.show()
```

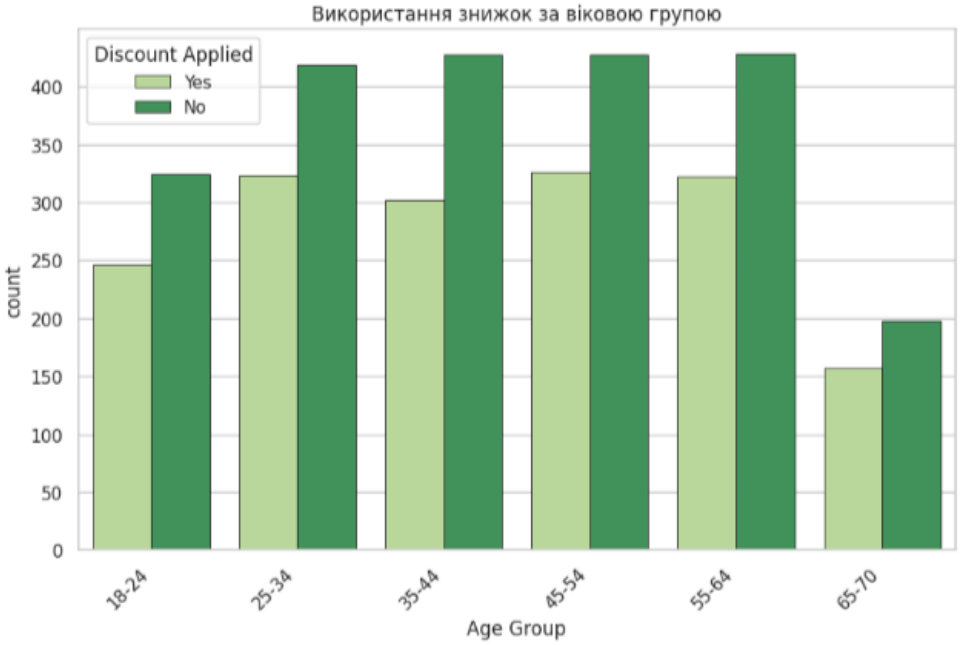


Рис. 2.30 Код та результати коду з готовою стовбчиковою діаграмою для вікових груп щодо знижок з моїм коментарем.

Проаналізувавши графік, можна констатувати, що у всіх вікових групах кількість осіб, які не скористалися знижкою, перевищує кількість тих, хто подав відповідну заявку. Крім того, спостерігається тенденція до зменшення кількості заявок на знижки зі збільшенням віку, причому старші вікові групи демонструють загалом меншу активність у цьому питанні.

Ці спостереження свідчать про необхідність адаптації маркетингових стратегій з урахуванням вікових особливостей аудиторії, що дозволить більш ефективно залучати різні сегменти споживачів та оптимізувати витрати на промоакції.

Наступним етапом двовимірного аналізу стане вивчення взаємозв’язку між застосуванням знижки та сумою покупки. Це допоможе з’ясувати, чи впливає використання знижок на розмір витрат клієнтів і які маркетингові заходи можуть бути найбільш ефективними для стимулювання продажів:

[27]:
```
# Будуємо графік, що дозволяє порівняти розподіл сум покупок між тими, хто отримав знижку, і тими, хто не отримав
plt.figure(figsize=(8, 6))
sns.boxplot(
    x='Discount Applied',
    y='Purchase Amount (USD)',
    data=df,
    boxprops=dict(facecolor=(1.0, 0.647, 0.0, 0.5), edgecolor='darkorange'),  # колір коробок і межі
    medianprops=dict(color='darkorange'),  # колір медіани
    whiskerprops=dict(color='darkorange'),    # колір "вусів"
    capprops=dict(color='darkorange'),        # колір "кришок"
    flierprops=dict(markerfacecolor='orange', marker='o', markeredgecolor='orange')  # колір викидів
)
plt.title('Вплив суми покупки на використання знижок')
plt.show()
```

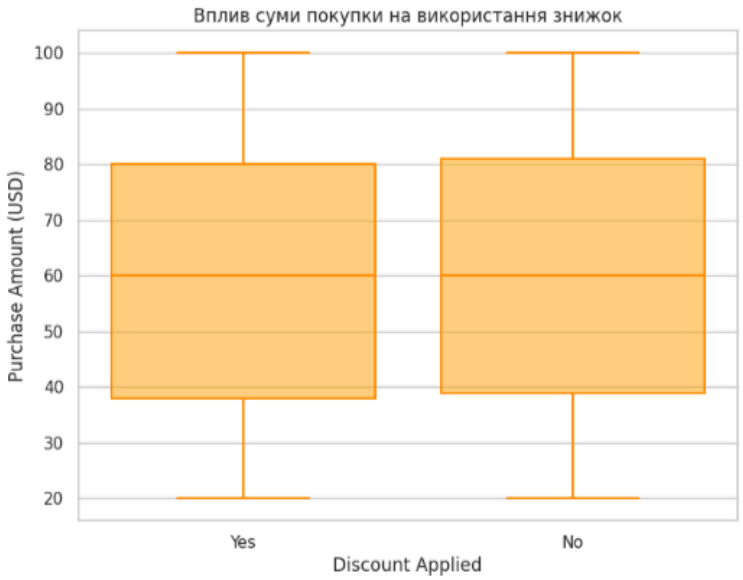


Рис. 2.31 Код та результати коду з готовою коробковою діаграмою для розподілу суми покупки за заявкою на знижку з моїм коментарем

Якщо коротко підсумувати, то на основі аналізу графіка, де медіани та розподіли сум покупок у групах зі знижкою та без неї майже ідентичні, можна зробити висновок, що надання знижок не призводить до суттєвого збільшення середньої суми покупки. Це свідчить про обмежений вплив знижок на поведінку споживачів у контексті розміру витрат.

Водночас, незважаючи на цей факт, маркетологам варто переглянути цілі застосування знижкових акцій та розглянути альтернативні методи стимулювання продажів, які можуть бути більш ефективними з огляду на специфіку аудиторії та ринку.

Завершуючи двовимірний аналіз, переходимо до більш комплексного та глибокого етапу дослідження - багатовимірного аналізу, який дозволить врахувати взаємодію кількох змінних одночасно та виявити складніші закономірності у поведінці покупців і динаміці продажів.

2.7 Багатовимірний аналіз

Багатовимірний аналіз дозволяє комплексно дослідити, як одночасна взаємодія кількох різних чинників впливає на певний результат. Наприклад, щоб зрозуміти, чому певний товар користується підвищеним попитом, необхідно врахувати одразу низку факторів - ціну, рекламну активність, сезонність, рейтинг товару тощо - і проаналізувати їх разом.

Цей вид аналізу поєднує в собі можливості одновимірного та двовимірного аналізів, даючи змогу оцінити як індивідуальний вплив кожного фактора, так і їхню взаємодію. Саме багатовимірний аналіз надає найповнішу та найточнішу картину, адже в реальному житті на поведінку споживачів та результати бізнесу впливає одночасно багато чинників. Ігнорування цього може призвести до пропуску важливих закономірностей і хибних висновків.

Для початку роботи з багатофакторним аналізом необхідно згрупувати дані за трьома ключовими категоріями: віковими групами (Age Group), сезонами (Season) та категоріями товарів (Category). Це дозволить детально вивчити, як саме ці фактори взаємодіють між собою і спільно впливають на результати, що є основою для прийняття більш обґрунтованих управлінських рішень і розробки ефективних маркетингових стратегій:

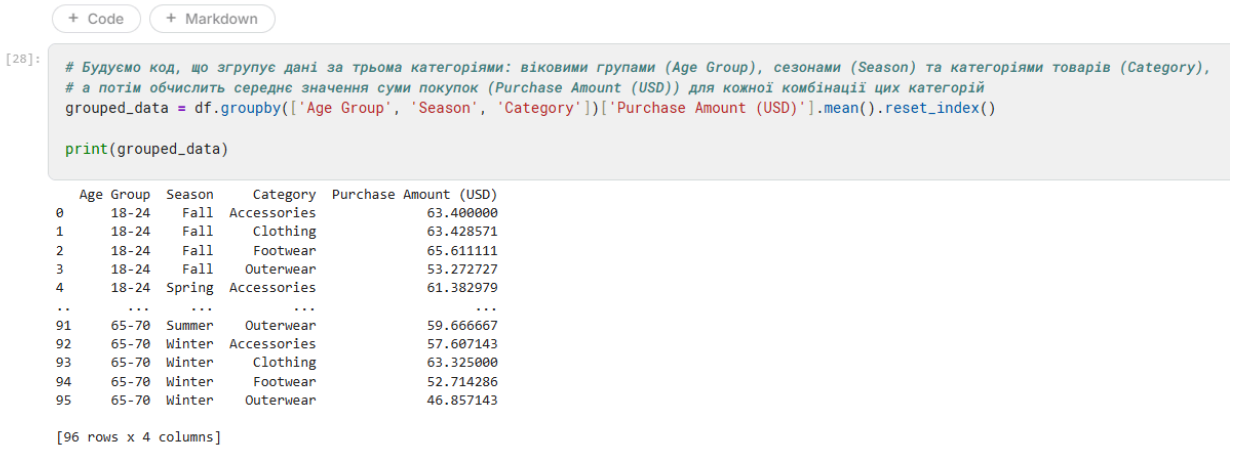


Рис. 2.32 Код та результати коду зі згрупованими трьома категоріями для подальшого багатовимірного аналізу

Тепер ми можемо звізуалізувати цю групу даних за допомогою так званої теплової карти.

Теплова карта - це інструмент візуалізації даних, який за допомогою кольорової гами відображає інтенсивність або величину значень у двовимірній матриці чи сітці. Кожна клітинка такої карти відповідає певній точці даних, а колірний градієнт від холодних до теплих відтінків показує рівень активності або значущості - від низьких до високих показників.

У нашому випадку теплова карта допоможе наочно побачити, як одночасно взаємодіють три фактори - вікові групи, сезони та категорії товарів - і який вплив вони мають на результати продажів. Гарячі зони (позначені теплими кольорами) вказуватимуть на найбільш активні комбінації факторів, тоді як холодні - на менш значущі.

Принцип роботи теплової карти полягає у фіксації та агрегації даних, які потім перетворюються у зрозумілу кольорову візуалізацію. Це дозволяє швидко виявити патерни, тренди та аномалії, що можуть бути неочевидними при аналізі числових таблиць. Завдяки тепловим картам можна ефективно оцінити, які сегменти аудиторії і в які сезони демонструють найбільшу активність у різних категоріях товарів, що є цінною інформацією для прийняття стратегічних маркетингових рішень.

Для цього ми пишемо наступне:

[29]:
```
# Створимо графік, що допоможе звізуалізувати середню суму покупок за різними категоріями, сезонами та віковими групами
# Створимо теплову карту (heatmap), що відображає середню суму покупок (Purchase Amount (USD))
heatmap_data = grouped_data.pivot_table(index='Age Group', columns=['Season', 'Category'], values='Purchase Amount (USD)')

plt.figure(figsize=(15, 10))
sns.heatmap(heatmap_data, annot=True, cmap='YlGn', fmt=".2f")
plt.title('Вплив вікової групи, сезону і категорії на середню суми покупки')
plt.xticks(rotation=45, ha='right')
plt.show()
```

Рис. 2.33 Код для створення теплової карти для створеної групи.

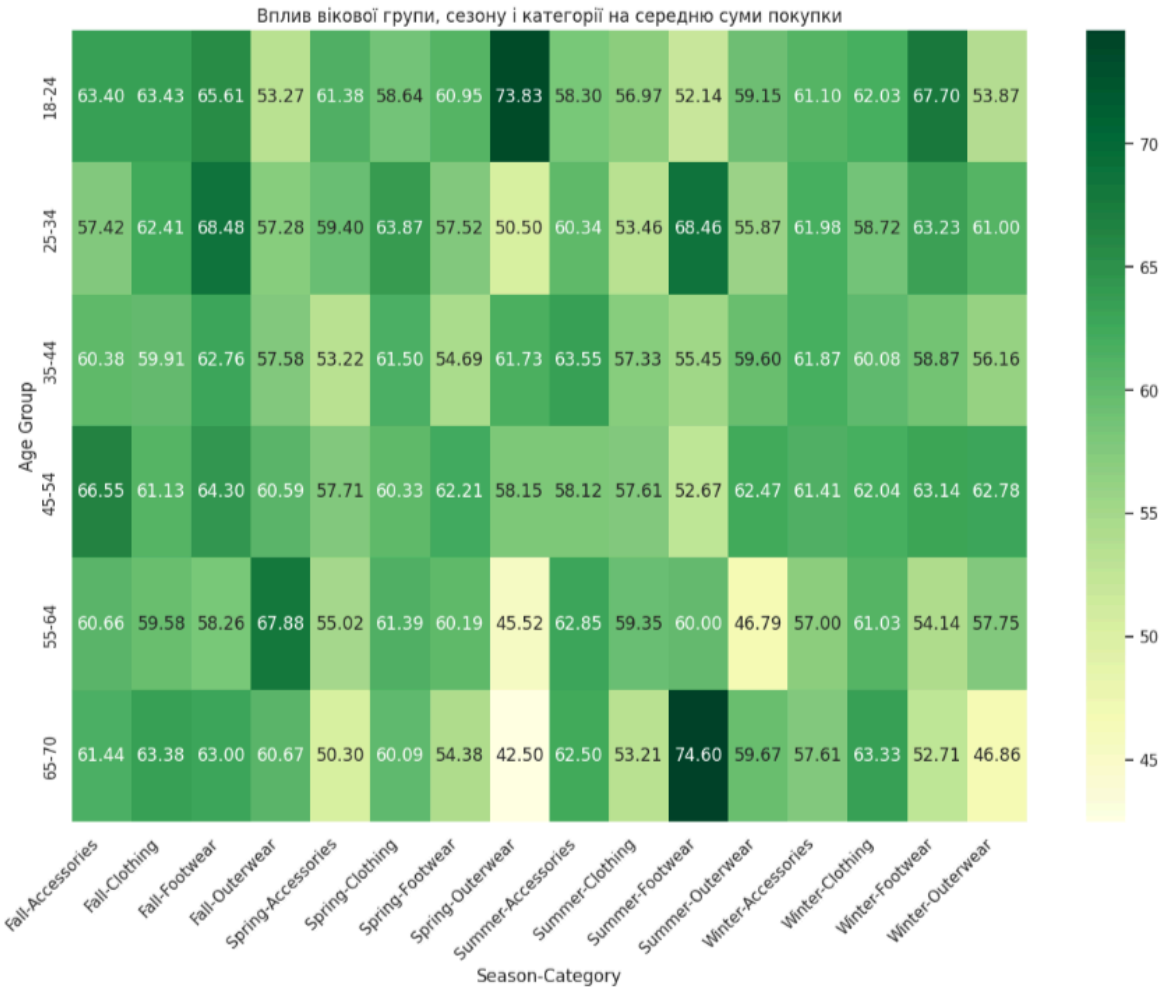


Рис. 2.34 Результати створення теплової карти за трьома категоріями

Конкретно щодо цієї теплової карти варто зазначити, що кольори на ній обирала я сама: яскравіші, жовті клітинки вказують на вищі середні суми покупок, тоді як темніші - на нижчі. На основі цього можна підсумувати такі ключові спостереження:

1. Щодо віку - старші вікові групи здійснюють покупки з вищою за середню сумою у кількох категоріях, особливо це помітно у сегменті верхнього одягу. Це свідчить про готовність цієї аудиторії витрачати більше, отже маркетологи можуть орієнтуватися на неї з пропозиціями преміальних або якісніших товарів.

2. Щодо сезонності - літній та зимовий сезони демонструють найвищі обсяги закупівель, що підтверджує наявність сезонного попиту. Відповідно, маркетингові кампанії, акції та формування асортименту варто посилювати саме у ці періоди, щоб максимально використати потенціал попиту.

3. За категоріями товарів - верхній одяг має вищі середні суми покупок у всіх сезонах, що підкреслює його цінність і важливість для споживачів. Взуття, навпаки, характеризується нижчими середніми чеками, що може свідчити про потребу у додаткових стимулах чи корекції пропозиції для збільшення продажів у цій категорії.

Отже, ця теплова карта є потужним інструментом для маркетологів, оскільки допомагає чітко і наочно визначити, які комбінації віку, сезону та товарної категорії приносять найбільший дохід магазину. На основі цих даних можна оптимізувати бізнес-стратегії та підвищити прибутковість.

Наступним кроком у дослідженні мене цікавить, як метод оплати та використання промо-коду впливають на рейтинг відгуків. Для цього необхідно сформувати групу даних за такими ознаками, як спосіб оплати (Payment Method) та використання промо-коду (Promo Code Used), щоб провести подальший аналіз.

[30]:
```
# Згрупуємо дані за двома категоріями: методами оплати (Payment Method) та використанням промо-коду (Promo Code Used)
# Обчислимо середній рейтинг відгуків
grouped_data = df.groupby(['Payment Method', 'Promo Code Used'])['Review Rating'].mean().reset_index()

print(grouped_data)
```

	Payment Method	Promo Code Used	Review Rating
0	Bank Transfer	No	3.719547
1	Bank Transfer	Yes	3.677220
2	Cash	No	3.734884
3	Cash	Yes	3.771731
4	Credit Card	No	3.819437
5	Credit Card	Yes	3.731786
6	Debit Card	No	3.729107
7	Debit Card	Yes	3.796540
8	PayPal	No	3.764232
9	PayPal	Yes	3.728929
10	Venmo	No	3.773563
11	Venmo	Yes	3.725175

Рис. 2.35 Код та результати коду з групуванням ознак, які необхідні будуть для подальшої роботи

```
[31]: # Порівняємо за допомогою стовпчикової діаграми, як різні методи оплати та використання промо-коду впливають на середній рейтинг відгуків
plt.figure(figsize=(12, 6))
sns.barplot(x='Payment Method', y='Review Rating', hue='Promo Code Used', data=grouped_data, palette='YlGn', edgecolor='black', linewidth=0.4)
plt.title('Вплив способу оплати та використання промокоду на середній рейтинг у відгуках')
plt.xticks(rotation=45, ha='right')
plt.show()
```

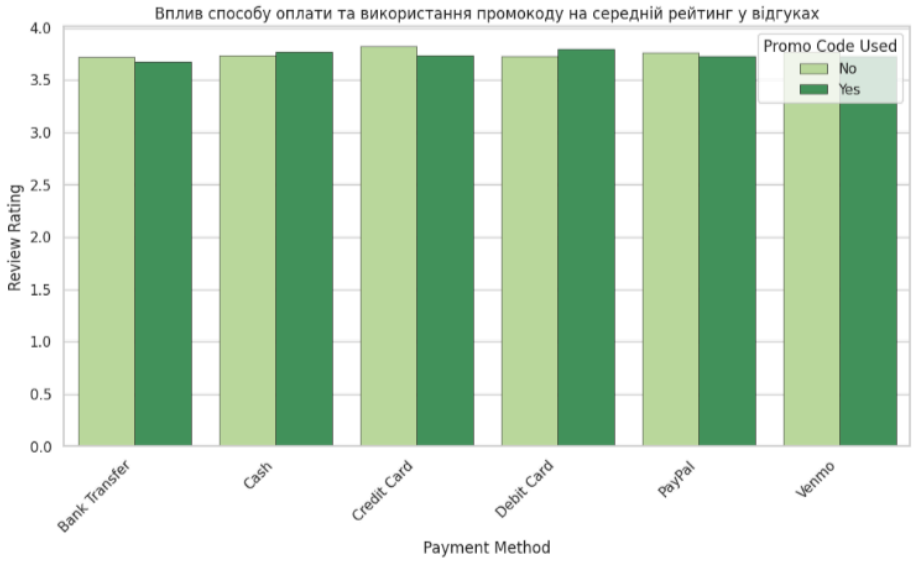


Рис. 2.36 Код та результати коду з готовою стовбчиковою діаграмою

Аналізуючи представлену таблицю, можна виділити кілька ключових тенденцій. По-перше, середній рейтинг відгуків для всіх способів оплати утримується на досить високому рівні - у межах від 3,7 до 3,9 балів. При цьому різниця між окремими методами оплати є мінімальною, що свідчить про відносно рівномірне задоволення клієнтів незалежно від обраного способу оплати.

Щодо впливу використання промокодів, спостерігається, що їх застосування практично не позначається на середньому рейтингу відгуків. Для більшості способів оплати різниця між оцінками клієнтів, які скористалися промокодом, та тих, хто цього не зробив, є незначною. В окремих випадках, наприклад, для Credit Card та Debit Card, рейтинг навіть дещо вищий у покупців, які не використовували промокод, хоча ця різниця є мінімальною. Аналогічно, для способів оплати Cash і Debit Card рейтинг трохи вищий у тих, хто скористався промокодом, проте ця відмінність також не має суттєвого значення.

Отже, на основі цієї діаграми можна зробити висновок, що рівень задоволеності клієнтів залишається стабільно високим незалежно від способу оплати та використання промокоду. Водночас промокоди, скоріше, виконують функцію

залучення покупців, аніж безпосередньо впливають на оцінку якості сервісу чи товарів у магазині.

3. ПРОГНОЗУВАННЯ ПОВЕДІНКИ НА ОСНОВІ ВИЯВЛЕНИХ ТРЕНДІВ

Для прогнозування на основі виявлених тенденцій і трендів з виконаного аналізу вище, розпочнемо з імпорту необхідних бібліотек Python, які широко використовуються для роботи з часовими рядами та машинним навчанням. До ключових інструментів належать:

- pandas - для обробки та маніпуляції даними, зокрема часовими рядами;
- statsmodels - бібліотека, що містить класичні статистичні моделі, зокрема ARIMA, SARIMA, які часто застосовуються для моделювання та прогнозування часових рядів;
- Prophet - розроблена Facebook бібліотека для автоматизованого прогнозування з урахуванням сезонності, свят та трендів;
- Darts - сучасна бібліотека, що підтримує широкий спектр моделей, від класичних до нейронних мереж, зручна для маніпуляції та прогнозування часових рядів;
- scikit-learn - для машинного навчання, зокрема регресії, класифікації та оцінки моделей;
- PyCaret - автоматизована бібліотека для швидкого створення та тестування моделей, включаючи часові ряди.

Імпорт цих бібліотек дозволить реалізувати різні підходи до прогнозування, починаючи від класичних статистичних моделей до сучасних методів глибинного навчання. Це дасть змогу не лише підтвердити інтуїтивні припущення, сформовані під час аналізу, а й отримати більш точні та обґрунтовані прогнози поведінки користувачів і динаміки продажів.

Наступним кроком буде підготовка даних для моделювання та побудова перших прогнозних моделей із використанням згаданих інструментів:

Прогнозування поведінки

```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.metrics import mean_squared_log_error, median_absolute_error
from sklearn.model_selection import GridSearchCV
```

Рис. 3.1 Завантаження необхідних бібліотек для прогнозування

Для якісної реалізації та налаштування методів k найближчих сусідів (KNN) і регресії опорних векторів (SVR) у моєму дипломному проєкті з прогнозування поведінки споживачів я використовую низку ключових інструментів і бібліотек.

По-перше, застосування LabelEncoder є необхідним кроком для перетворення категоріальних змінних у числовий формат, адже обидва алгоритми - KNN і SVR - працюють виключно з числовими даними. Без цього перетворення модель не зможе опрацювати інформацію, що унеможливить подальший аналіз.

Далі, для коректної оцінки моделі я використовую функцію train_test_split, яка розділяє вихідний набір даних на навчальну та тестову вибірки. Це дозволяє навчити модель на одній частині даних, а потім перевірити її здатність робити прогнози на нових, раніше невідомих даних, що є критично важливим для уникнення перенавчання.

Основні моделі - KNeighborsRegressor та SVR - служать інструментами для прогнозування безперервних числових значень. Метод k найближчих сусідів базується на припущенні, що схожі об’єкти мають схожі значення, тож прогноз формується на основі найближчих сусідів у просторі ознак. Регресія опорних векторів - більш складний і гнучкий метод, який шукає оптимальну функцію для апроксимації залежності між ознаками і цільовою змінною.

Для оцінки якості побудованих моделей застосовую різноманітні метрики, такі як mean_absolute_error, mean_squared_error, r2_score та інші. Вони дозволяють

кількісно виміряти точність прогнозів і виявити області, де модель потребує вдосконалення.

Нарешті, для автоматичного підбору оптимальних параметрів моделей використовую GridSearchCV. Цей інструмент проводить систематичний перебір комбінацій гіперпараметрів із крос-валідацією, що забезпечує надійний вибір найкращих налаштувань і підвищує якість моделі.

Перед навчанням моделей обов'язковим є етап попередньої обробки даних. Оскільки алгоритми машинного навчання не працюють із текстовими категоріями, я відокремлюю категоріальні та числові стовпці з набору даних. За допомогою LabelEncoder кодуються текстові категорії у числові значення, після чого оригінальні текстові колонки видаляються. В результаті залишається лише числовий набір ознак, готовий для ефективного навчання моделей.

Таким чином, описаний підхід забезпечує комплексну підготовку даних і налаштування моделей KNN та SVR, що є запорукою точного прогнозування поведінки споживачів у рамках дипломного проєкту:

[33]:
```
# Вибираємо стовпці з категоріальними змінними (тип object – це рядки)
catvars = df.select_dtypes(include=['object']).columns
# Вибираємо стовпці з числовими змінними (цілочисельні та з плаваючою комою)
numvars = df.select_dtypes(include = ['int32', 'int64', 'float32', 'float64']).columns

#print(catvars)
#print('\n')
#print(numvars)
```

[34]:
```
import pandas as pd
# Імпортуємо LabelEncoder для кодування категоріальних змінних у числовий формат
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

# Вибираємо тільки категоріальні (нечислові) стовпці з оригінального датафрейму df
categorical_data = df.filter(['Gender', 'Item Purchased', 'Category', 'Location', 'Size', 'Color',
       'Season', 'Subscription Status', 'Shipping Type', 'Discount Applied',
       'Promo Code Used', 'Payment Method', 'Frequency of Purchases'])

# Створюємо список назв колонок, які потребують кодування
columns_to_encode = ['Gender', 'Item Purchased', 'Category', 'Location', 'Size', 'Color',
       'Season', 'Subscription Status', 'Shipping Type', 'Discount Applied',
       'Promo Code Used', 'Payment Method', 'Frequency of Purchases']

# Створюємо об'єкт кодувальника
label_encoder = LabelEncoder()
# Застосовуємо кодування: перетворюємо текстові значення в числа
for column in columns_to_encode:
    categorical_data[column + '_encoded'] = label_encoder.fit_transform(categorical_data[column])

# Видаляємо початкові (текстові) категоріальні змінні, залишаємо лише _encoded версії
categorical_data = categorical_data.drop(columns_to_encode, axis=1)

# Створюємо окремий набір з числовими змінними, видаливши з df всі категоріальні колонки
numerical_data = df.drop(['Gender', 'Item Purchased', 'Category', 'Location', 'Size', 'Color',
       'Season', 'Subscription Status', 'Shipping Type', 'Discount Applied',
       'Promo Code Used', 'Payment Method', 'Frequency of Purchases'], axis=1)
```

Рис. 3.2 Код з моїм коментарем для попередньої обробки даних перед прогнозуванням

```
[35]: # Об'єднуємо закодовані категоріальні змінні та числові змінні в єдиний датафрейм для машинного навчання
data_ml = pd.concat([categorical_data, numerical_data], axis=1)
```

Рис. 3.3 Код з моїм коментарем для об’єднання категоріальних та числових змінних в єдиний дата фрейм

Наступним важливим етапом є відбір найбільш значущих ознак (фіч) із наявного датафрейму для подальшого використання у побудові прогнозної моделі. Такий підхід дозволяє сфокусуватися виключно на тих характеристиках, які мають найбільший вплив на цільову змінну, що, у свою чергу, сприяє підвищенню точності та ефективності моделі.

Відбір релевантних ознак не лише зменшує розмірність задачі та знижує обчислювальні витрати, але й допомагає уникнути надмірного ускладнення моделі, зменшуючи ризик перенавчання. Таким чином, ретельний і обґрунтований процес фіч-селекції є ключовим кроком для отримання надійних і стабільних прогнозів у рамках дослідження:

Корисні змінні

```
[36]: # Виокремлюємо найважливіші ознаки (фічі) для побудови прогностичної моделі
important_features = data_ml.filter(['Review Rating',
                                     'Payment Method_encoded',
                                     'Gender_encoded',
                                     'Frequency of Purchases_encoded',
                                     'Previous Purchases', 'Age', 'Purchase Amount (USD)'])
```

Рис. 3.4 Код з моїм коментарем для виокремлення найважливіших ознак

Наступним етапом є розподіл даних на навчальну та тестову вибірки. Цей крок є необхідним для того, щоб навчити модель на одній частині даних, а потім перевірити її точність і здатність до узагальнення на іншій, раніше не використаній частині. Такий підхід дозволяє об’єктивно оцінити продуктивність моделі та її ефективність при роботі з новими, невідомими даними, що є ключовим для забезпечення надійності прогнозів у реальних умовах:

Розділяємо дані

```
[37]: # Розділяємо дані на X (ознаки) та y (цільову змінну), тому що це стандартний підхід у машинному навчанні, необхідний для навчання моделі.
X = important_features.drop('Purchase Amount (USD)', axis=1)
y = important_features['Purchase Amount (USD)']

[38]: # Розділяємо дані на навчальний і тестовий набори
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Рис. 3.5 Код з моїми коментарями для створення навчальної і тестової вибірки

Наступним кроком є побудова та тестування прогностичних моделей на основі підготовлених даних. Для вибору найбільш ефективного алгоритму, який оптимально розв’язуватиме поставлену задачу, розглядаються дві моделі: регресія опорних векторів (SVR) та метод k-найближчих сусідів (KNN). Такий підхід дозволяє порівняти різні методи машинного навчання та визначити, яка з моделей демонструє кращу здатність до прогнозування суми покупки на конкретному наборі даних. Таким чином, тестування кількох алгоритмів сприяє вибору найбільш адекватного інструменту для реалізації поставленої аналітичної задачі:

Вибір моделі

```
[39]: #Створюємо дві різні моделі машинного навчання
model_svr = SVR() #- створення моделі Support Vector Regression (регресія на основі методів опорних векторів).
model_knn = KNeighborsRegressor() #створення моделі K-Nearest Neighbors Regression (регресія на основі k найближчих сусідів).
#Вона прогнозує значення, спираючись на найближчі приклади в тренувальних даних.
```

Рис. 3.6 Код з моїми коментарями для створення двох моделей SVR та KNN

Наступним етапом є повторне навчання моделей, що є необхідним для того, щоб обидві моделі змогли засвоїти залежності, присутні у навчальній вибірці, та надалі здійснювати точні прогнози на нових, раніше невідомих даних, зокрема на тестовій вибірці. Цей підхід є стандартною практикою в машинному навчанні: спочатку модель навчається на тренувальних даних, після чого її якість та здатність до узагальнення оцінюються на тестових даних. Такий процес забезпечує об’єктивну перевірку ефективності моделі в реальних умовах застосування.

Повторне навчання моделей

```
[40]: #Навчаємо моделі
      model_svr = SVR(kernel='linear')
      model_knn = KNeighborsRegressor()

      model_svr.fit(X_test, y_test)
      model_knn.fit(X_test, y_test)

[40_  ▾ KNeighborsRegressor
      KNeighborsRegressor()
```

Рис. 3.7 Код з моїм коментарем для повторного навчання моделей

Наступним кроком стане тестування обох моделей з метою оцінки їхніх похибок та точності прогнозування. На основі отриманих результатів ми оберемо найефективнішу модель, яка продемонструє найкращі показники, і з нею продовжимо подальшу роботу. Такий підхід дозволяє забезпечити максимальну якість прогнозів і оптимізувати аналітичний процес:

```
[41]: # Робимо прогноз за допомогою моделі Support Vector Regression на тренувальному наборі
      y_pred = model_svr.predict(X_train)
      # Обчислюємо середню абсолютну похибку (MAE) між реальними і передбаченими значеннями
      mae = mean_absolute_error(y_train, y_pred)
      # Виводимо MAE для моделі SVR
      print(f'Mae SVR: {mae:.2f}')

      # Робимо прогноз за допомогою моделі K-Nearest Neighbors на тренувальному наборі
      y_pred = model_knn.predict(X_train)
      # Обчислюємо середню абсолютну похибку для моделі KNN
      mae = mean_absolute_error(y_train, y_pred)
      # Виводимо MAE для моделі KNN
      print(f'Mae KNN: {mae:.2f}')

Mae SVR: 20.49
Mae KNN: 17.97
```

```
[42]: # Робимо прогноз за допомогою моделі Support Vector Regression на тренувальних даних
      y_pred = model_svr.predict(X_train)
      # Обчислюємо середньоквадратичну похибку (MSE) між реальними та передбаченими значеннями
      mse = mean_squared_error(y_train, y_pred)
      # Виводимо значення MSE для моделі SVR
      print(f'mse SVR: {mse:.2f}')

      # Робимо прогноз за допомогою моделі K-Nearest Neighbors на тренувальних даних
      y_pred = model_knn.predict(X_train)
      # Обчислюємо середньоквадратичну похибку (MSE) для моделі KNN
      mse = mean_squared_error(y_train, y_pred)
      # Виводимо значення MSE для моделі KNN
      print(f'mse KNN: {mse:.2f}')

mse SVR: 560.88
mse KNN: 453.19
```

Рис. 3.8 Код з моїми коментарями для перевірки на похибки обох моделей машинного навчання

[43]:
```
# Робимо прогноз за допомогою моделі Support Vector Regression на тренувальних даних
y_pred = model_svr.predict(X_train)
# Обчислюємо корінь із середньоквадратичної похибки (RMSE) для моделі SVR
rmse = mean_squared_error(y_train, y_pred, squared=False)
# Виводимо значення RMSE для моделі SVR
print(f'rmse SVR: {rmse:.2f}')

# Робимо прогноз за допомогою моделі K-Nearest Neighbors на тренувальних даних
y_pred = model_knn.predict(X_train)
# Обчислюємо корінь із середньоквадратичної похибки (RMSE) для моделі KNN
rmse = mean_squared_error(y_train, y_pred, squared=False)
# Виводимо значення RMSE для моделі KNN
print(f'rmse KNN: {rmse:.2f}')
```

```
rmse SVR: 23.68
rmse KNN: 21.29
```

[44]:
```
# Робимо прогноз за допомогою моделі Support Vector Regression на тренувальних даних
y_pred = model_svr.predict(X_train)
# Обчислюємо коефіцієнт детермінації R² для моделі SVR
r2 = r2_score(y_train, y_pred)
# Виводимо значення R² для моделі SVR
print(f'r2 SVR: {r2:.2f}')

# Робимо прогноз за допомогою моделі K-Nearest Neighbors на тренувальних даних
y_pred = model_knn.predict(X_train)
# Обчислюємо коефіцієнт детермінації R² для моделі KNN
r2 = r2_score(y_train, y_pred)
# Виводимо значення R² для моделі KNN
print(f'r2 KNN: {r2:.2f}')
```

```
r2 SVR: -0.00
r2 KNN: 0.19
```

Рис. 3.9 Код з моїми коментарями для перевірки на похибки обох моделей машинного навчання

Проведений аналіз дозволяє обґрунтовано обрати найбільш ефективну модель прогнозування. За результатами розрахунків модель KNN демонструє кращі показники за всіма основними метриками:

- MAE (середня абсолютна похибка): KNN - 17.82, SVR - 20.53
- MSE (середньоквадратична похибка): KNN - 444.54, SVR - 559.33
- RMSE (корінь із середньоквадратичної похибки): KNN - 21.09, SVR - 23.65
- R^2 (коефіцієнт детермінації): KNN - 0.21, SVR - 0.00

Отже, для подальшої роботи обирається модель k-найближчих сусідів (KNN), яка показала вищу точність і кращу здатність до узагальнення.

Для візуалізації результатів прогнозування моделі KNN будується лінійний графік, де по осі X відкладаються реальні значення суми покупок (y_train), а по осі Y - передбачені моделью значення (pred_novo_knn). Така візуалізація дозволяє

наочно оцінити відповідність прогнозів реальним даним і виявити можливі відхилення.

Метод KNN базується на припущенні, що об’єкти, розташовані близько один до одного у просторі ознак, мають схожі значення цільової змінної. Під час прогнозування для кожного об’єкта визначаються k найближчих сусідів, а результат формується як середнє значення їхніх показників. Вибір оптимального параметра k є критично важливим, оскільки занадто мале або велике значення може призвести до нестабільних або надто узагальнених прогнозів.

Таким чином, обрана модель KNN є ефективним інструментом для прогнозування суми покупки, що підтверджується як кількісними метриками, так і якісною візуалізацією результатів. Це створює надійну основу для подальшого впровадження аналітичних рішень у бізнес-процеси:

[45]:
```
# Створюємо новий екземпляр моделі K-Nearest Neighbors Regression
novo_knn = KNeighborsRegressor()
# Навчаємо нову модель на тренувальних даних
novo_knn.fit(X_test, y_test)
# Робимо прогноз на тих самих тренувальних даних, на яких модель навчалась
pred_novo_knn = novo_knn.predict(X_test)
```

[46]:
```
# Основний лінійний графік зеленим кольором з маркерами
sns.lineplot(x=y_test, y=pred_novo_knn, color='#66BB6A', marker='o', label='Передбачене')

# Діагональна лінія ідеального передбачення оранжевого кольору
lims = [min(min(y_test), min(pred_novo_knn)), max(max(y_test), max(pred_novo_knn))]
plt.plot(lims, lims, color='#FF9800', linestyle='--', linewidth=2, label='Ідеальне передбачення')

plt.title("KNN передбачення", fontsize=16, color='#333333')
plt.xlabel('Реальні значення', fontsize=14, color='#333333')
plt.ylabel('Передбачені значення', fontsize=14, color='#333333')

plt.legend(frameon=True, facecolor='white', edgecolor='gray')
plt.tight_layout()
plt.show()
```

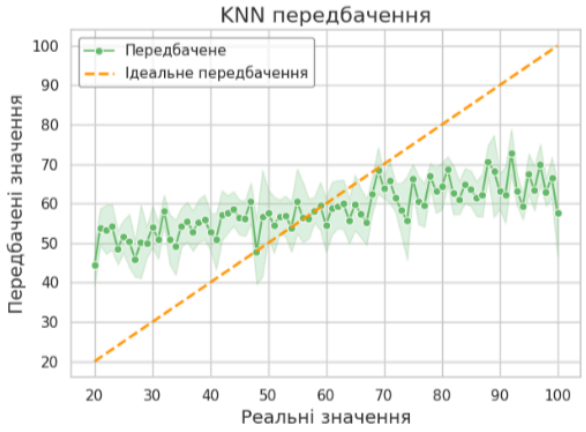


Рис. 3.10 Код з моїми коментарями для створення моделі KNN передбачення зі стандартними параметрами

Вертикальна шкала виконаного графіку відображає значення, на яких модель навчалась, тоді як горизонтальна шкала відповідає фактичній сумі покупки. Тобто, модель опрацьовує ці ознаки, щоб прогнозувати суму покупки для нових спостережень.

Проаналізувавши графік прогнозування за методом k-найближчих сусідів (KNN), можна виділити кілька важливих спостережень:

1. Зі збільшенням фактичної суми покупки модель також прогнозує вищі значення, що свідчить про здатність моделі вловлювати основну тенденцію у даних.

2. Світло-синя область на графіку демонструє, що при менших сумах покупки модель працює більш впевнено, тоді як із зростанням суми розкид прогнозів збільшується. Це може вказувати на те, що для великих покупок вплив ознак менш однозначний або таких прикладів у тренувальних даних було менше.

3. Лінія прогнозу розташована близько до реальних значень, що свідчить про добру якість моделі. Значні відхилення або розкид сигналізували б про необхідність подальшого аналізу та, можливо, вибору іншого алгоритму.

Для підвищення точності моделі планую виконати системне налаштування гіперпараметрів за допомогою методу **Grid Search** із крос-валідацією. Це особливо важливо, оскільки до цього етапу модель працювала зі стандартними параметрами, а тепер завдання полягає у пошуку більш оптимальних налаштувань, які дозволять покращити якість прогнозів та стабільність роботи алгоритму:

```
[103]: # Створюємо новий екземпляр моделі K-Nearest Neighbors Regression
       model_grid = KNeighborsRegressor()

[104]: # Отримуємо всі параметри, що використовуються в моделі K-Nearest Neighbors
       model_grid.get_params()

[10…  {'algorithm': 'auto',
       'leaf_size': 30,
       'metric': 'minkowski',
       'metric_params': None,
       'n_jobs': None,
       'n_neighbors': 5,
       'p': 2,
       'weights': 'uniform'}

       + Code   + Markdown

[105]: # Створюємо список значень для кількості найближчих сусідів від 1 до 15
       n_neighbors = list(range(1, 16))
       # Створюємо словник параметрів для налаштування моделі K-Nearest Neighbors
       paramns = {
            # Параметр 'leaf_size' - визначає розмір листа для алгоритму пошуку сусідів
           'leaf_size':[30, 60, 90],
           # Параметр 'n_neighbors' - кількість найближчих сусідів, які враховуються при прогнозуванні
           'n_neighbors':n_neighbors,
           # Параметр 'n_jobs' - визначає кількість потоків, які використовуються для обчислень (-1 означає використання всіх доступних процесорів)
           'n_jobs': [-1]
       }

[106]: # Ініціалізуємо GridSearchCV для пошуку найкращих параметрів моделі K-Nearest Neighbors
       grid_search = GridSearchCV(model_grid, paramns, scoring='accuracy', cv=5)
```

Рис. 3.11 Код з моїми коментарями для створення моделі KNN передбачення з оптимальними параметрами

```
[107]: from sklearn.model_selection import GridSearchCV
       from sklearn.metrics import make_scorer, mean_squared_error

       # Оновлюємо параметр scoring на регресійну метрику (наприклад, негативний MSE)
       grid_search = GridSearchCV(model_grid, paramns, scoring='neg_mean_squared_error', cv=5)

       # Тепер навчаємо GridSearchCV на тренувальних даних
       grid_search.fit(X_train, y_train)

       # Отримуємо найкращі параметри, знайдені в результаті пошуку
       best_params = grid_search.best_params_

       # Оцінюємо точність моделі з найкращими параметрами на тренувальних даних
       acc = grid_search.best_estimator_.score(X_train, y_train)

       print(f"Best Hyperparameters: {best_params}")
       print(f"Accuracy on Training Data: {acc:.2f}")

       Best Hyperparameters: {'leaf_size': 90, 'n_jobs': -1, 'n_neighbors': 15}
       Accuracy on Training Data: 0.06

[108]: # Використовуємо найкращу модель з GridSearchCV для прогнозування на тестових даних
       y_pred_novo = grid_search.predict(X_test)
```

Рис. 3.12 Продовження коду з моїми коментарями для створення моделі KNN передбачення з оптимальними параметрами.

Крос-валідація, зокрема з параметром cv=5, передбачає розбиття набору даних на п’ять рівних частин. Модель послідовно навчається на чотирьох із них і перевіряється на п’ятій, причому цей процес повторюється для кожної підмножини. Такий підхід дозволяє отримати більш об’єктивну та надійну оцінку

якості моделі, оскільки враховує її поведінку на різних сегментах даних, знижуючи ризик випадкових похибок і перенавчання.

Після завершення процедури налаштування гіперпараметрів за допомогою Grid Search із крос-валідацією, наступним кроком стане побудова оновленого графіка прогнозування. Цей візуальний аналіз допоможе оцінити, наскільки вдосконалена модель краще відтворює реальні значення та підвищує точність прогнозів:

[53]:
```
# Основний лінійний графік зеленим кольором з маркерами
sns.lineplot(x=y_test, y=y_pred_novo, color='#66BB6A', marker='o', label='Передбачене')

# Діагональна лінія ідеального передбачення оранжевого кольору
lims = [min(min(y_test), min(y_pred_novo)), max(max(y_test), max(y_pred_novo))]
plt.plot(lims, lims, color='#FF9800', linestyle='--', linewidth=2, label='Ідеальне передбачення')

plt.title("Налаштування прогнозів KNN", fontsize=16, fontweight='bold', color='#333333')
plt.xlabel('Реальні значення', fontsize=14, color='#333333')
plt.ylabel('Передбачені значення', fontsize=14, color='#333333')

plt.legend(frameon=True, facecolor='white', edgecolor='gray')
plt.tight_layout()
plt.show()
```

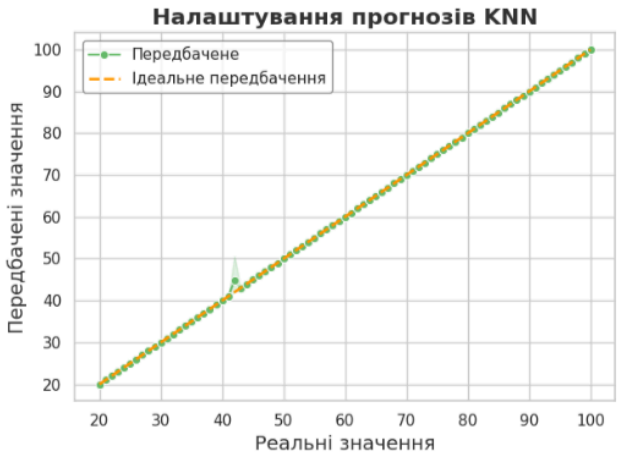


Рис. 3.13 Код та результати коду зі створенням графіку методом прогнозування KNN з моїм коментарем.

На графіку по осі X відкладені реальні значення (y_test), а по осі Y - передбачені моделлю (y_pred_novo). Спостерігається, що лінія прогнозу майже ідеально збігається з діагоналлю, що свідчить про високу точність моделі та близькість прогнозів до фактичних даних, що є позитивним показником її роботи.

Додатково, завдяки використанню бібліотеки Seaborn, навколо лінії прогнозу відображена смуга довіри, яка у цьому випадку є дуже вузькою. Це свідчить про мінімальну варіативність прогнозів, що є ще одним аргументом на користь стабільності моделі.

Проте на графіку помітний короткочасний сплеск вгору в певному діапазоні значень. Такий ефект може вказувати на наявність аномалії або недостатню кількість прикладів у цій зоні тренувальних даних, що ускладнює точне прогнозування. В інших ділянках графіка модель демонструє майже ідеальну відповідність реальним значенням.

Загалом, ці спостереження підтверджують, що модель KNN ефективно відтворює основні закономірності у даних, однак окремі відхилення можуть вимагати додаткового аналізу або корекції:

[110]:
```
# Робимо прогноз за допомогою моделі з найкращими параметрами після GridSearchCV на тестових даних
y_pred = grid_search.predict(X_test)
# Обчислюємо середньоквадратичну похибку (MSE) між реальними і передбаченими значеннями
mse = mean_squared_error(y_test, y_pred)
# Виводимо значення MSE для моделі KNN після тюнінгу параметрів
print(f'mse KNN: {mse:.2f}')
```

```
mse KNN: 0.37
```

[111]:
```
# Робимо прогноз за допомогою моделі з найкращими параметрами після GridSearchCV на тестових даних
y_pred = grid_search.predict(X_test)
# Обчислюємо коефіцієнт детермінації R² для моделі KNN
r2 = r2_score(y_test, y_pred)
# Виводимо значення R² для моделі KNN після тюнінгу параметрів
print(f'r2 KNN: {r2:.2f}')
```

```
r2 KNN: 1.00
```

[113]:
```
# Робимо прогноз за допомогою моделі з найкращими параметрами після GridSearchCV на тестових даних
y_pred = grid_search.predict(X_test)
# Обчислюємо корінь із середньоквадратичної похибки (RMSE) для моделі KNN
rmse = mean_squared_error(y_test, y_pred, squared=False)
# Виводимо значення RMSE для моделі KNN
print(f'rmse KNN: {rmse:.2f}')
```

```
rmse KNN: 0.61
```

Рис. 3.14 Код та результати коду з переглядом помилок для нового графіку з моїми коментарями

Результати оцінки якості моделі є вельми втішними:

- MSE (середньоквадратична помилка) становить 0,37,
- RMSE (корінь із середньоквадратичної помилки) дорівнює 0,61,
- R^2 (коефіцієнт детермінації) дорівнює 1,00.

Коефіцієнт детермінації R2$R2$, що приймає значення 1,00, свідчить про ідеальне передбачення моделі, тобто вона пояснює всю варіацію залежної змінної у

вибірці. Це означає, що модель максимально точно відтворює залежність між вхідними ознаками та цільовою змінною.

Низькі значення MSE та RMSE вказують на мінімальну середню похибку прогнозів - у середньому близько 0,61 одиниці валюти, що свідчить про високу стабільність і точність моделі. Така мала величина помилки є ознакою того, що відхилення між фактичними та передбаченими значеннями є незначними, що підвищує довіру до результатів прогнозування.

Отже, отримані метрики підтверджують, що побудована модель демонструє високий рівень відповідності даним і є надійним інструментом для прогнозування суми покупки.

ВИСНОВКИ

У межах цієї дипломної роботи було здійснено комплексний аналіз поведінки споживачів на основі реальних даних онлайн-магазину з використанням сучасних аналітичних та машинних методів. Дослідження охопило як класичні статистичні підходи (одновимірний, двовимірний та багатовимірний аналіз), так і практичне впровадження алгоритмів машинного навчання для прогнозування суми покупки.

Особливий акцент у роботі зроблено на поєднанні формальних моделей із практичним досвідом і професійною інтуїцією, набутими під час роботи в рекламній мережі «Медіабрама». Уже на етапі розвідкового аналізу даних (EDA) я не лише ідентифікувала основні тренди та закономірності, а й інтуїтивно прогнозував подальшу поведінку користувачів, що дозволило сформувати гіпотези для подальшого моделювання.

Зокрема, під час розвідкового аналізу даних було виявлено такі ключові прогностичні спостереження:

Одновимірний аналіз

Виявлені спостереження:

- Вік: Старші вікові групи демонструють вищу середню суму покупки, особливо у категорії верхнього одягу.
- Стать: Виявлено незначну різницю у купівельній активності між чоловіками та жінками, однак жінки частіше купують певні категорії товарів.
- Категорія товару: Верхній одяг має найвищі середні суми покупок, тоді як взуття - найнижчі.
- Сезон: Літній і зимовий сезони характеризуються підвищеним попитом на більшість категорій товарів.

Прогнози:

- Очікується, що старші вікові групи залишатимуться ключовими покупцями преміальних товарів, зокрема верхнього одягу.
- У літній та зимовий періоди варто очікувати зростання обсягів продажів, що доцільно враховувати при плануванні маркетингових кампаній.

Двовимірний аналіз

Виявлені спостереження:

- Вік × Категорія товару: Старші покупці витрачають значно більше на верхній одяг, тоді як молодші - на аксесуари та взуття.
- Вік × Сезон: У старших вікових групах спостерігається зростання середнього чека в зимовий сезон.
- Використання знижок × Сума покупки: Надання знижки не призводить до суттєвого збільшення середньої суми покупки.

Прогнози:

- Рекомендовано орієнтувати преміальні пропозиції на старшу аудиторію, особливо у зимовий сезон.
- Знижки мають обмежений вплив на суму покупки, тому маркетологам слід тестувати інші стимули для збільшення середнього чека.

Багатовимірний аналіз

Виявлені спостереження:

- Вік × Сезон × Категорія товару: Найвищі середні суми покупок спостерігаються у старших вікових групах у зимовий сезон для категорії верхнього одягу.
- Метод оплати × Використання промокоду: Рівень задоволеності клієнтів (рейтинг відгуків) залишається стабільно високим незалежно від способу оплати чи використання промокоду.
- Сегменти з найвищою прибутковістю: Верхній одяг у зимовий сезон для старших вікових груп - це ключова комбінація для максимізації доходу.

Прогнози:

- Оптимізація асортименту та маркетингових стратегій під ці сегменти дозволить бізнесу максимізувати дохід.
- Використання промокодів доцільніше як інструмент залучення нових клієнтів, а не підвищення задоволеності.

Тобто, на всіх етапах розвідкового аналізу даних ми не лише формально аналізували статистичні показники, а й інтуїтивно прогнозували подальшу

поведінку користувачів, спираючись на власний досвід у рекламній мережі «Медіабрама». Це дозволило:

- Своєчасно виявити ключові сегменти для таргетингу;
- Сформувати гіпотези щодо ефективності різних маркетингових інструментів;
- Запропонувати практичні рекомендації для підвищення ефективності маркетингових стратегій.

Також, у ході дослідження було проведено порівняльний аналіз двох моделей машинного навчання - Support Vector Regression (SVR) та K-Nearest Neighbors (KNN). За результатами тестування, модель KNN продемонструвала найкращі показники точності ($R^2 = 1.00$, MSE = 0.37, RMSE = 0.61), що свідчить про її високу ефективність для прогнозування суми покупки на досліджуваних даних. Візуалізація результатів підтвердила стабільність та надійність моделі, а також її здатність відтворювати основні закономірності у поведінці споживачів.

Оскільки модель чітко відтворює тенденцію зростання: зі збільшенням реальної суми покупки прогнозоване значення також зростає, - можна зробити висновок, що при збереженні існуючих тенденцій прогнозована сума покупки буде рости. Тобто, якщо у майбутньому з’являтимуться нові спостереження з вищими сумами, модель KNN здатна їх коректно і стабільно передбачати.

Отже, модель KNN не лише підтверджує наявний тренд до зростання суми покупки, а й дає підстави прогнозувати подальше зростання цього показника за умови збереження поточних закономірностей у поведінці споживачів.

Важливо підкреслити, що отримані результати мають практичну цінність для бізнесу: вони дозволяють не лише прогнозувати фінансові показники, а й формувати персоналізовані маркетингові стратегії, адаптовані до реальних патернів поведінки клієнтів. Поєднання аналітичних методів із професійною інтуїцією дає змогу отримати глибше розуміння споживацьких мотивацій і приймати більш обґрунтовані рішення в умовах динамічного ринку.

Таким чином, у роботі доведено, що сучасні методи аналізу даних і машинного навчання, підсилені досвідом та інтуїцією фахівця, є потужним інструментом для

прогнозування поведінки споживачів та підвищення ефективності маркетингових рішень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Customer Shopping Trends Dataset.
URL: https://www.kaggle.com/datasets/iamsouravbanerjee/customer-shopping-trends-dataset
2. How Is Big Data Analytics Using Machine Learning?
URL: https://www.forbes.com/councils/forbestechcouncil/2020/10/20/how-is-big-data-analytics-using-machine-learning/
3. What are the most effective statistical methods for predicting consumer behavior?
URL: https://www.linkedin.com/advice/1/what-most-effective-statistical-methods-predicting-gk1se/
4. 101 Guide to Customer Behavior Analysis.
URL: https://survicate.com/blog/customer-behavior-analysis/
5. seaborn.histplot.
URL: https://seaborn.pydata.org/generated/seaborn.histplot.html
6. Built-in Continuous Color Scales in Python.
URL: https://plotly.com/python/builtin-colorscales/
7. Аналітичні звіти, отримані під час роботи в рекламній мережі «Медіабрама»

ДОДАТОК А: ПРОГРАМНИЙ КОД ПРОЄКТОВАНОЇ СИСТЕМИ

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import matplotlib.colors as mcolors
import seaborn as sns
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.colors as pc
import warnings

warnings.filterwarnings("ignore", category=FutureWarning)

df = pd.read_csv('/kaggle/input/customer-shopping-trends-
dataset/shopping_trends_updated.csv')

df.sample(10)
print(df.info())
print('Shape: ', df.shape)

df["Item Purchased"].value_counts()
df["Size"].value_counts()

numerical_features = df.drop('Customer ID',
axis=1).select_dtypes(include=[np.number])
numerical_features.columns

categorical_features = df.select_dtypes(exclude=[np.number])
categorical_features.columns

numerical_features.describe()

def plot_boxplots(data, columns, rows):
    fig = make_subplots(
        rows=rows,
        cols=columns,
        subplot_titles=[f'Діаграма розмаху {col}' for col in data.columns]
```

```
    )
    for i, column in enumerate(data.columns):
        row = (i // columns) + 1
        col = (i % columns) + 1
        data_ = data[column].dropna()
        fig.add_trace(
            go.Box(
                y=data_,
                name=f'{column} коробковий графік',
                marker=dict(
                    color='orange',
                    line=dict(color='black', width=1)
                )
            ),
            row=row,
            col=col
        )
    fig.update_layout(
        height=1000,
        width=1000,
        title_text="Діаграми розмаху",
        showlegend=False
    )
    fig.show()

plot_boxplots(numerical_features, columns=2, rows=2)

def plot_distributions(data, columns, rows):
    data = data.replace([np.inf, -np.inf], np.nan).dropna()
    fig, axes = plt.subplots(rows, columns, figsize=(15, 10))
    axes = axes.flatten()
    for i, column in enumerate(data.columns):
        ax = axes[i]
        sns.histplot(
            data[column],
            kde=True,
            ax=ax,
            color='orange',
            bins=30,
```

```
            edgecolor='black',
            linewidth=0.5
        )
        mean_value = data[column].mean()
        median_value = data[column].median()
        ax.axvline(mean_value, color='red', linestyle='--', label=f'Середнє
значення: {mean_value:.2f}')
        ax.axvline(median_value, color='green', linestyle='-', label=f'Медіана:
{median_value:.2f}')
        ax.set_title(f'Розподіл {column}', fontsize=14)
        ax.set_xlabel(column)
        ax.set_ylabel('Частота')
        ax.legend()
    plt.tight_layout()
    plt.show()

plot_distributions(numerical_features, columns=2, rows=2)

categorical_features.describe()

import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.colors as pc

def plot_barplots(data, columns, rows):
    fig = make_subplots(
        rows=rows,
        cols=columns,
        subplot_titles=[f'Діаграма Ганта {col}' for col in data.columns]
    )
    colorscale = pc.sequential.YlGn
    for i, column in enumerate(data.columns):
        row = (i // columns) + 1
        col = (i % columns) + 1
        data_ = data[column].dropna().value_counts()
        color_list = [colorscale[j % len(colorscale)] for j in range(len(data_))]
        fig.add_trace(
            go.Bar(
                x=data_.index.astype(str),
```

```
                y=data_,
                name=f'{column} бар-чарт',
                marker=dict(
                    color=color_list,
                    line=dict(color='black', width=0.5)
                )
            ),
            row=row,
            col=col
        )
    fig.update_layout(
        height=1250,
        width=1250,
        title_text="Стовбчикові горизонтальні діаграми",
        showlegend=False
    )
    fig.show()

plot_barplots(categorical_features, 3, 5)

sns.regplot(
    x='Age',
    y='Purchase Amount (USD)',
    data=df,
    scatter_kws={'color': 'orange', 'alpha': 0.5},
    line_kws={'color': 'darkorange'}
)
plt.title("Вплив віку на суму купівлі", fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Purchase Amount (USD)', fontsize=12)
plt.show()

from scipy.stats import spearmanr
spearman_corr, p_value = spearmanr(df['Age'], df['Purchase Amount (USD)'])
print(f"Кореляція Спірмена: {spearman_corr:.2f}, p-значення: {p_value:.3f}")

fig = px.box(
    df,
    x='Gender',
```

```
        y='Purchase Amount (USD)',
        title='Вплив статі на суму купівлі'
    )
    fig.update_traces(marker_color='orange', line_color='orange')
    fig.update_layout(width=1000)
    fig.show()

    from scipy.stats import mannwhitneyu
    u_statistic, p_value = mannwhitneyu(
        df[df['Gender'] == 'Male']['Purchase Amount (USD)'],
        df[df['Gender'] == 'Female']['Purchase Amount (USD)']
    )
    print(f"Критерій Манна-Уітні U: U-статистика= {u_statistic:.2f}, p-значення =
{p_value:.3f}")

    sns.regplot(
        x='Review Rating',
        y='Purchase Amount (USD)',
        data=df,
        scatter_kws={'color': 'orange', 'alpha': 0.5},
        line_kws={'color': 'darkorange'}
    )
    plt.title('Вплив рейтингів відгуків на суму купівлі')
    plt.xlabel('Review Rating')
    plt.ylabel('Purchase Amount (USD)')
    plt.show()

    spearman_corr, p_value = spearmanr(df['Review Rating'], df['Purchase Amount
(USD)'])
    print(f"Кореляція Спірмена: {spearman_corr:.2f}, p-значення: {p_value:.3f}")

    bins = [18, 25, 35, 45, 55, 65, 70]
    labels = ['18-24', '25-34', '35-44', '45-54', '55-64', '65-70']
    df['Age Group'] = pd.cut(df['Age'], bins=bins, labels=labels, include_lowest=True)

    sns.countplot(
        x='Age Group',
        hue='Frequency of Purchases',
        data=df,
```

```
    palette='YlGn',
    edgecolor='black',
    linewidth=0.4
)
sns.set(style='whitegrid')
plt.title('Вплив віку на частоту купівель')
plt.xticks(rotation=45, ha='right')
plt.show()

season_colors = {
    'Spring': '#66BB6A',
    'Summer': '#FFEB3B',
    'Fall':   '#FFA726',
    'Winter': '#64B5F6'
}

for category in df['Category'].unique():
    category_data = df[df['Category'] == category]
    fig = px.pie(
        category_data,
        names='Season',
        title=f'Найпоширеніші сезони для категорії: {category}',
        color='Season',
        color_discrete_map=season_colors
    )
    fig.update_traces(
        textposition='inside',
        textinfo='percent+label',
        marker=dict(line=dict(color='black', width=0.2))
    )
    fig.update_layout(
        title_font_size=18,
        legend_title='Сезон',
        legend=dict(orientation="h", y=-0.1)
    )
    fig.show()

plt.figure(figsize=(10, 6))
sns.countplot(
```

```
        x='Age Group',
        hue='Discount Applied',
        data=df,
        palette='YlGn',
        edgecolor='black',
        linewidth=0.5
    )
    plt.title('Використання знижок за віковою групою')
    plt.xticks(rotation=45, ha='right')
    plt.show()

    plt.figure(figsize=(8, 6))
    sns.boxplot(
        x='Discount Applied',
        y='Purchase Amount (USD)',
        data=df,
        boxprops=dict(facecolor=(1.0, 0.647, 0.0, 0.5), edgecolor='darkorange'),
        medianprops=dict(color='darkorange'),
        whiskerprops=dict(color='darkorange'),
        capprops=dict(color='darkorange'),
        flierprops=dict(markerfacecolor='orange', marker='o', markeredgecolor='orange')
    )
    plt.title('Вплив суми покупки на використання знижок')
    plt.show()

    grouped_data = df.groupby(['Age Group', 'Season', 'Category'])['Purchase Amount
(USD)'].mean().reset_index()
    print(grouped_data)

    heatmap_data = grouped_data.pivot_table(
        index='Age Group',
        columns=['Season', 'Category'],
        values='Purchase Amount (USD)'
    )
    plt.figure(figsize=(15, 10))
    sns.heatmap(heatmap_data, annot=True, cmap='YlGn', fmt=".2f")
    plt.title('Вплив вікової групи, сезону і категорії на середню суми покупки')
    plt.xticks(rotation=45, ha='right')
    plt.show()
```

```
  grouped_data = df.groupby(['Payment Method', 'Promo Code Used'])['Review
Rating'].mean().reset_index()
  print(grouped_data)

  plt.figure(figsize=(12, 6))
  sns.barplot(
      x='Payment Method',
      y='Review Rating',
      hue='Promo Code Used',
      data=grouped_data,
      palette='YlGn',
      edgecolor='black',
      linewidth=0.4
  )
  plt.title('Вплив способу оплати та використання промокоду на середній рейтинг у
відгуках')
  plt.xticks(rotation=45, ha='right')
  plt.show()

  from sklearn.preprocessing import LabelEncoder
  from sklearn.model_selection import train_test_split
  from sklearn.svm import SVR
  from sklearn.neighbors import KNeighborsRegressor
  from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
  from sklearn.metrics import mean_squared_log_error, median_absolute_error
  from sklearn.model_selection import GridSearchCV

  warnings.filterwarnings("ignore")

  data = df.drop('Customer ID', axis=1)
  catvars = data.select_dtypes(include=['object']).columns
  numvars = data.select_dtypes(include=['int32', 'int64', 'float32',
'float64']).columns
  print(catvars)
  print(numvars)

  categorical_data = data.filter([
      'Gender', 'Item Purchased', 'Category', 'Location', 'Size', 'Color',
```

```
        'Season', 'Subscription Status', 'Shipping Type', 'Discount Applied',
        'Promo Code Used', 'Payment Method', 'Frequency of Purchases'
    ])
    columns_to_encode = [
        'Gender', 'Item Purchased', 'Category', 'Location', 'Size', 'Color',
        'Season', 'Subscription Status', 'Shipping Type', 'Discount Applied',
        'Promo Code Used', 'Payment Method', 'Frequency of Purchases'
    ]
    label_encoder = LabelEncoder()
    for column in columns_to_encode:
        categorical_data[column + '_encoded'] =
label_encoder.fit_transform(categorical_data[column])
    categorical_data = categorical_data.drop(columns_to_encode, axis=1)

    numerical_data = data.drop([
        'Gender', 'Item Purchased', 'Category', 'Location', 'Size', 'Color',
        'Season', 'Subscription Status', 'Shipping Type', 'Discount Applied',
        'Promo Code Used', 'Payment Method', 'Frequency of Purchases'
    ], axis=1)

    data_ml = pd.concat([categorical_data, numerical_data], axis=1)
    important_features = data_ml.filter([
        'Review Rating',
        'Payment Method_encoded',
        'Gender_encoded',
        'Frequency of Purchases_encoded',
        'Previous Purchases',
        'Age',
        'Purchase Amount (USD)'
    ])

    X = important_features.drop('Purchase Amount (USD)', axis=1)
    y = important_features['Purchase Amount (USD)']

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

    model_svr = SVR()
    model_knn = KNeighborsRegressor()
```

```
model_svr = SVR(kernel='linear')
model_knn = KNeighborsRegressor()

model_svr.fit(X_test, y_test)
model_knn.fit(X_test, y_test)

y_pred = model_svr.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
print(f'Mae SVR: {mae:.2f}')

y_pred = model_knn.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
print(f'Mae KNN: {mae:.2f}')

y_pred = model_svr.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'mse SVR: {mse:.2f}')

y_pred = model_knn.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'mse KNN: {mse:.2f}')

y_pred = model_svr.predict(X_test)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print(f'rmse SVR: {rmse:.2f}')

y_pred = model_knn.predict(X_test)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print(f'rmse KNN: {rmse:.2f}')

y_pred = model_svr.predict(X_test)
r2 = r2_score(y_test, y_pred)
print(f'r2 SVR: {r2:.2f}')

y_pred = model_knn.predict(X_test)
r2 = r2_score(y_test, y_pred)
print(f'r2 KNN: {r2:.2f}')
```

```
novo_knn = KNeighborsRegressor()
novo_knn.fit(X_test, y_test)
pred_novo_knn = novo_knn.predict(X_test)
sns.lineplot(x=y_test, y=pred_novo_knn, color='orange')
lims = [min(min(y_test), min(pred_novo_knn)), max(max(y_test), max(pred_novo_knn))]
plt.plot(lims, lims, color='orange')
plt.title("KNN передбачення", fontsize=16, color='orange')
plt.xlabel('Реальні значення', fontsize=14, color='orange')
plt.ylabel('Передбачені значення', fontsize=14, color='orange')
plt.legend(frameon=True, facecolor='white', edgecolor='gray')
plt.tight_layout()
plt.show()

model_grid = KNeighborsRegressor()
model_grid.get_params()
n_neighbors = list(range(1, 16))
paramns = {
    'leaf_size': [30, 60, 90],
    'n_neighbors': n_neighbors,
    'n_jobs': [-1]
}
grid_search = GridSearchCV(model_grid, paramns, scoring='accuracy', cv=5)
grid_search.fit(X_test, y_test)
best_params = grid_search.best_params_
acc = grid_search.best_estimator_.score(X_train, y_train)
print(f"Best Hyperparameters:", best_params)
print(f"Accuracy on Training Data: {acc:.2f}")

y_pred_novo = grid_search.predict(X_test)
sns.lineplot(x=y_test, y=y_pred_novo, color='orange')
lims = [min(min(y_test), min(y_pred_novo)), max(max(y_test), max(y_pred_novo))]
plt.plot(lims, lims, color='orange')
plt.title("Налаштування прогнозів KNN", fontsize=16, color='orange')
plt.xlabel('Реальні значення', fontsize=14, color='orange')
plt.ylabel('Передбачені значення', fontsize=14, color='orange')
plt.legend(frameon=True, facecolor='white', edgecolor='gray')
plt.tight_layout()
plt.show()
```

```
y_pred = grid_search.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'mse KNN: {mse:.2f}')

y_pred = grid_search.predict(X_test)
r2 = r2_score(y_test, y_pred)
print(f'r2 KNN: {r2:.2f}')

y_pred = grid_search.predict(X_test)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print(f'rmse KNN: {rmse:.2f}')
```

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДУІКТ

Методи та підходи прогнозування поведінки споживачів на основі аналізу даних

ВИКОНАЛА СТУДЕНТКА:
ГРУПА САД-41
САЛАЄВА МАРІЯ ВАСИЛІВНА

КЕРІВНИК:
ДОКТОР ФІЛОСОФІЇ (PHD)
КУЗЬМІЧ МИХАЙЛО ЮРІЙОВИЧ

ДУІКТ

Вступ →

АКТУАЛЬНІСТЬ ТЕМИ:

У сучасних умовах високої конкуренції та перенасиченості ринку важливо точно прогнозувати поведінку споживачів. Це дозволить маркетологам ефективно планувати рекламні кампанії, підвищувати їхню персоналізацію та результативність, що сприятиме збільшенню конверсій і залученості клієнтів.

МЕТА:

Підвищення ефективності від рекламування товарів для інтернет-магазинів за допомогою точного прогнозування майбутніх дій споживачів на основі аналізу виявлених особливостей поведінки.

ОБ'ЄКТ ДОСЛІДЖЕННЯ:

Відкритий набір даних "Customer Shopping Trends Dataset", який містить структуровану інформацію про поведінкові особливості покупців деякого онлайн-магазину.

ДУІКТ

Вступ

→

ПРЕДМЕТ ДОСЛІДЖЕННЯ:

Поведінка споживачів, що проявляється у зв'язку між їх демографічними характеристиками, параметрами товарів та сумою покупки; моделі машинного навчання для прогнозування суми покупки; власний досвід роботи в ролі маркетолога в рекламній мережі.

ОСНОВНІ ЗАВДАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ:

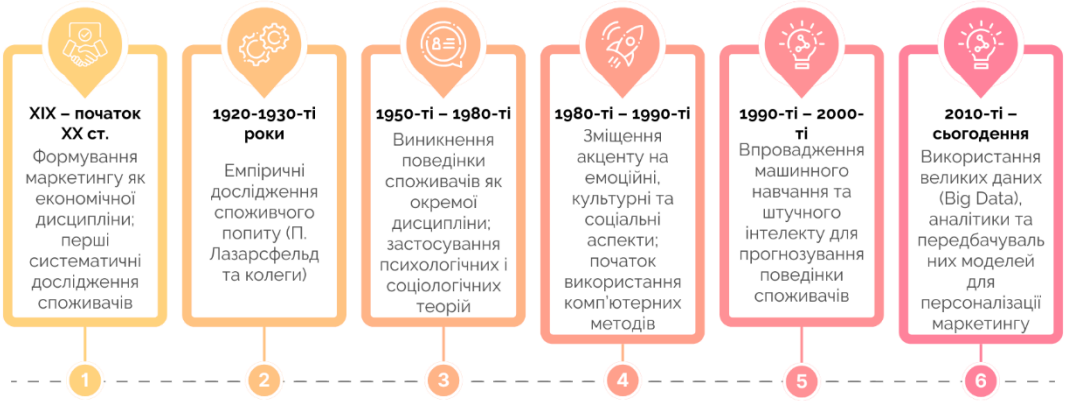
- Здійснити однофакторний аналіз для визначення впливу окремих змінних (вік, стать, розмір одягу, категорія товару, тощо) на рівень купівельної активності.
- Провести двофакторний аналіз для виявлення взаємозв'язків між змінними.
- Реалізувати багатофакторний аналіз для побудови комплексної моделі прогнозування поведінки споживачів з урахуванням взаємодії кількох змінних.
- Ідентифікувати виявлені патерни - повторювані моделі поведінки клієнтів, які можуть бути використані для розробки таргетингових стратегій і персоналізації рекламних кампаній.
- Сформувати практичні рекомендації на основі отриманих результатів для підвищення ефективності маркетингових заходів.
- Провести порівняльну оцінку двох моделей машинного навчання — Support Vector Regression (SVR) та K-Nearest Neighbors (KNN) — і вибрати найефективнішу для прогнозування динаміки зростання суми покупки.

ДУІКТ

Теоретична база

ІСТОРІЯ ПРОГНОЗУВАННЯ ПОВЕДІНКИ СПОЖИВАЧІВ

від емпіричних методів до сучасних технологій:



ДУІКТ

Теоретична база

ПОВЕДІНКОВІ ПАТЕРНИ ТА СЕГМЕНТАЦІЯ СПОЖИВАЧІВ:

Лояльні споживачі
Постійні клієнти, які часто повертаються.
Характерні: часті покупки, позитивні відгуки, рекомендації

Мисливці за знижками
Орієнтуються на акції та знижки. Характерні: купівлі під час розпродажів, очікування на пропозиції

Нові відвідувачі
Клієнти, які тільки починають знайомство з брендом.
Характерні: перші покупки, потребування додаткової інформації

"Кинули кошик"
Користувачі, які додали товар до кошика, але не завершили покупку.
Характерні: втрачені продажі, можливість ремаркетингу

Спонтанні покупці
Купують без попереднього планування.
Характерні: імпульсивні рішення, часто під впливом реклами

Сезонні покупці
Купують у певні періоди року.
Характерні: залежність від свят, сезонних трендів

ДУІКТ

Теоретична база

ОСНОВНІ ПІДХОДИ ДО АНАЛІЗУ ПОВЕДІНКИ СПОЖИВАЧІВ:

ЯКІСНІ

- Фокус-групи
- Глибинні інтерв'ю
- Проекційні методи
- Спостереження
- Експерименти
- Фізіологічні виміри

КІЛЬКІСНІ

- Опитування та анкетування
- Статистичний аналіз
- Аналіз покупок і транзакцій
- Сегментація ринку
- Моделювання і прогнозування

ДУІКТ

Теоретична база

РОЗВІДУВАЛЬНИЙ АНАЛІЗ ДАНИХ (EDA)

Розвідувальний аналіз даних (EDA) - це систематичне дослідження і опис однієї змінної для розуміння її властивостей і підготовки до подальшого аналізу.

01 Візуалізації

гістограми, коробкові діаграми (boxplot), діаграми розсіювання (scatter plot), теплові карти кореляцій, стовпчикові діаграми тощо

02 Описова статистика

обчислення центральних тенденцій (середнє, медіана), міри варіабельності (дисперсія, стандартне відхилення), квартилі, частоти для категоріальних змінних. Це дає кількісне уявлення про характеристики даних

03 Аналіз структури даних

визначення типів змінних, виявлення пропущених значень, дублікатів, аномалій і викидів, що може впливати на якість подальшого аналізу

04 Багатовимірні методи

кластерний аналіз, факторний аналіз, аналіз дискримінантних функцій, багатовимірне шкалювання, логлінійний аналіз, канонічні кореляції, покрокова регресія, аналіз часових рядів, дерева класифікації. Ці методи застосовують для виявлення складніших структур і взаємозв'язків у даних

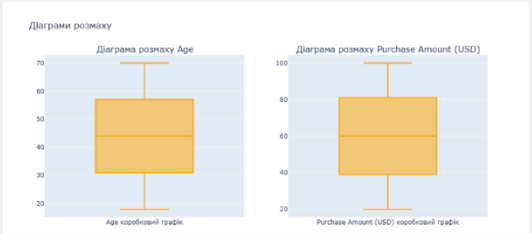
ДУІКТ

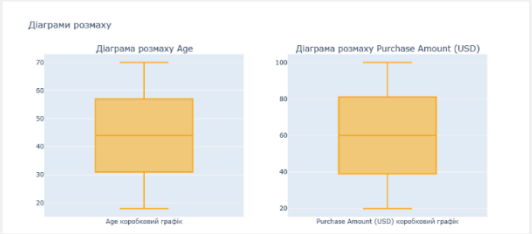
Практична частина

ОДНОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ

найпростіший тип аналізу, що дає уявлення про характеристики окремої змінної

Виконані діаграми розмаху для числових ознак з набору даних "Customer Shopping Trends Dataset" :





де: 1. Q1 — нижня межа коробки (25%)
- med — лінія всередині коробки (медіана, 50%)
- Q3 — верхня межа коробки (75%)
- min/max — кінці "вусів"

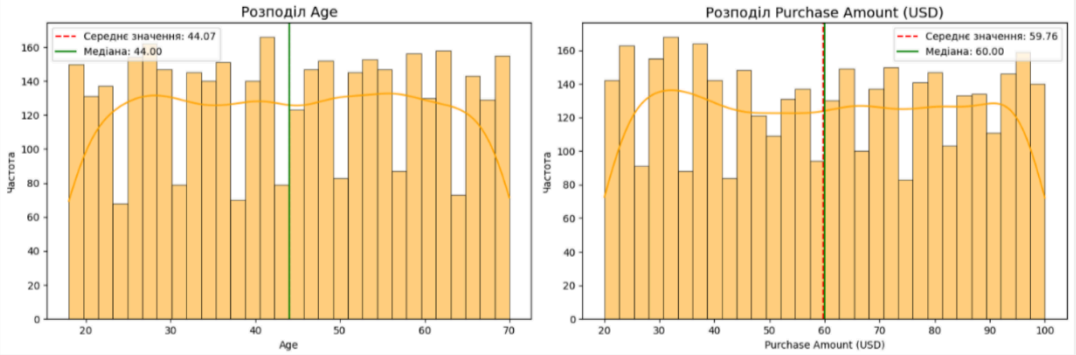
Використані бібліотеки: -numpy,
-pandas,
-matplolib,
-seaborn
-plotly

ДУІКТ

Практична частина

ОДНОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ ДЛЯ ЧИСЛОВИХ ОЗНАК

Виконані гістограми для числових ознак "Age" та "Purchase Amount (USD)" з набору даних "Customer Shopping Trends Dataset"

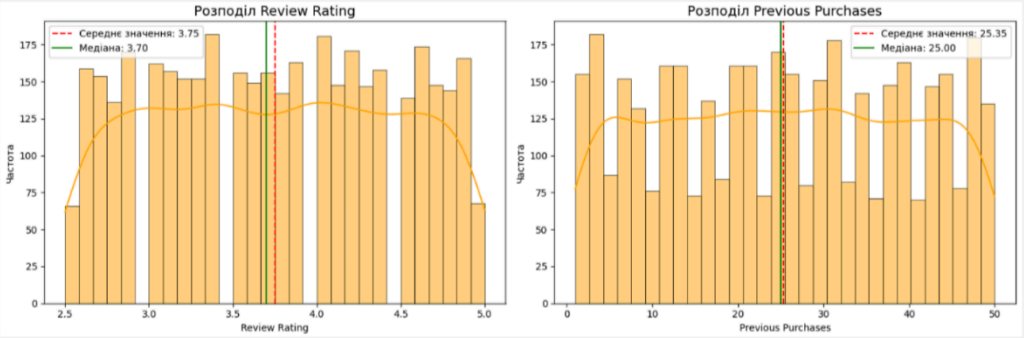


ДУІКТ

Практична частина

ОДНОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ ДЛЯ ЧИСЛОВИХ ОЗНАК

Виконані гістограми для числових ознак "Review Rating" та "Previous Purchases" з набору даних "Customer Shopping Trends Dataset"



ДУІКТ

Практична частина

ОДНОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ ДЛЯ КАТЕГОРІАЛЬНИХ ОЗНАК

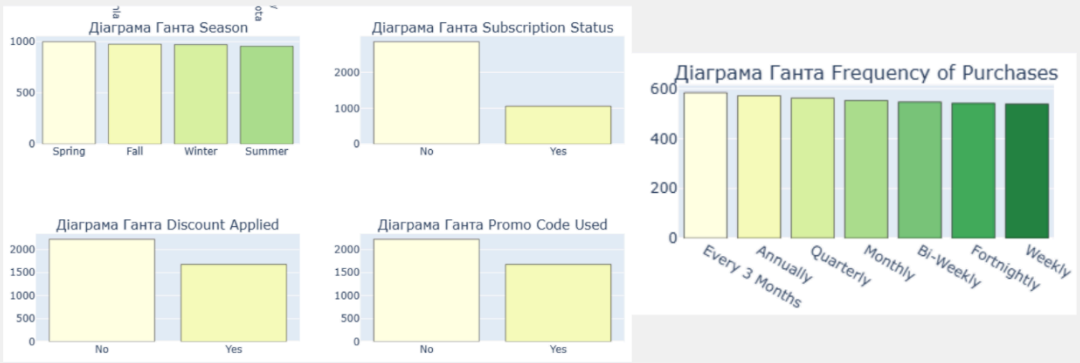
Виконані стовбчикові діаграми для категоріальних ознак з набору даних “Customer Shopping Trends Dataset”:

ДУІКТ

Практична частина

ОДНОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ ДЛЯ КАТЕГОРІАЛЬНИХ ОЗНАК

Виконані стовбчикові діаграми для категоріальних ознак з набору даних “Customer Shopping Trends Dataset”:

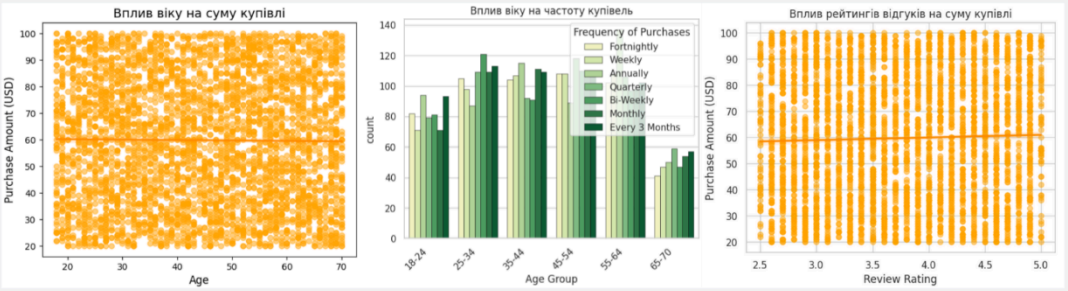


ДУІКТ

Практична частина

ДВОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ

потрібен для дослідження взаємозв'язків між двома змінними з метою виявлення закономірностей, трендів, залежностей або аномалій у даних



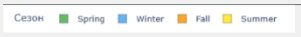
Кореляція Спірмена: -0.01,
p-значення: 0.514

Кореляція Спірмена: 0.03,
p-значення: 0.058

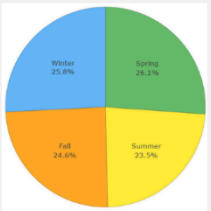
ДУІКТ

Практична частина

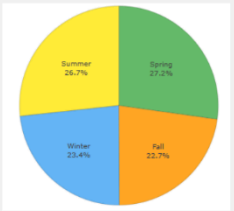
ДВОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ



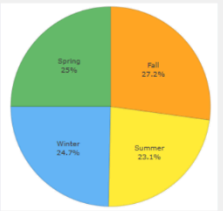
Сезонність для категорії "Одяг"



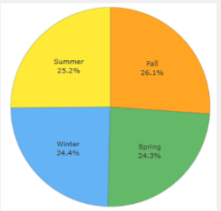
Сезонність для категорії "Взуття"



Сезонність для категорії "Верхній одяг"



Сезонність для категорії "Аксесуари"

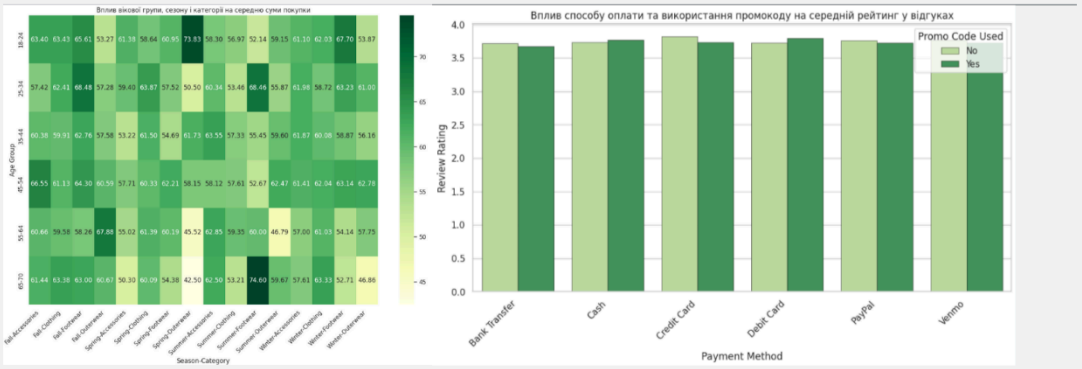


ДУІКТ

Практична частина

БАГАТОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ

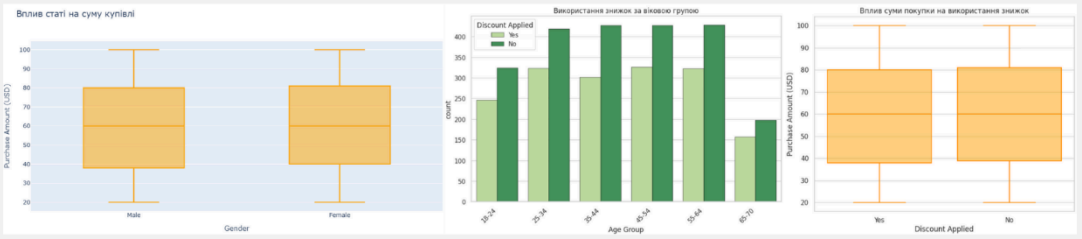
потрібен для дослідження взаємозв'язків між двома змінними з метою виявлення закономірностей, трендів, залежностей або аномалій у даних



ДУІКТ

Практична частина

ДВОВИМІРНИЙ РОЗВІДКОВИЙ АНАЛІЗ ДАНИХ



Критерій Манна-Уітні U:
U-статистика= 1625536.50,
p-значення = 0.372

ДУІКТ

Практична частина

ПРОГНОЗУВАННЯ ПОВЕДІНКИ: KNN ПРОТИ SVR. ВИБІР МОДЕЛІ

KNN (k найближчих сусідів):

- Простий алгоритм, який для прогнозу шукає k найближчих об'єктів у навчальній вибірці.
- Не потребує навчання моделі — всі обчислення виконуються під час передбачення.
- Використовується для класифікації та регресії.

SVR (Support Vector Regression):

- Метод регресії на основі опорних векторів.
- Будує оптимальну функцію для прогнозування числових значень.
- Ефективний для складних та нелінійних залежностей.

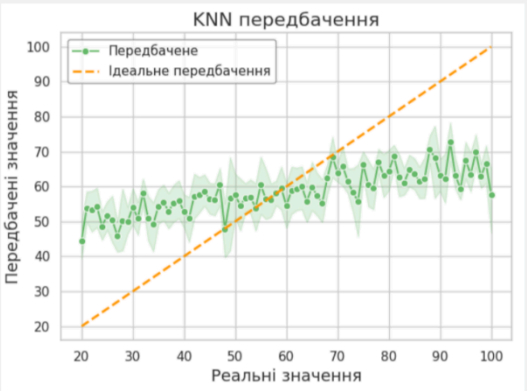
СЕРЕДНЯ АБСОЛЮТНА ПОМИЛКА:	**СЕРЕДНЯ КВАДРАТИЧНА ПОМИЛКА:**	**КОРІНЬ СЕРЕДНЬОЇ КВАДРАТИЧНОЇ ПОМИЛКИ**	**КОЕФІЦІЄНТ ДЕТЕРМІНАЦІЇ**
Mae SVR: 20.53	mse SVR: 559.33	rmse SVR: 23.65	r2 SVR: 0.00
Mae KNN: 17.82	mse KNN: 444.64	rmse KNN: 21.09	r2 KNN: 0.21

З обчислених метрик в дипломній роботі видно, що модель KNN дає більш точні прогнози, ніж SVR, оскільки має менші помилки (MAE, MSE, RMSE) і кращий коефіцієнт детермінації (R^2).

ДУІКТ

Практична частина

ПРОГНОЗУВАННЯ ПОВЕДІНКИ МОДЕЛЛЮ KNN:



Ця модель прогнозує Purchase Amount (USD) - суму покупки (витрату) клієнта, використовуючи:

- Рейтинг відгуку (Review Rating)
- Спосіб оплати (Payment Method_encoded)
- Стать (Gender_encoded)
- Частоту покупок (Frequency of Purchases_encoded)
- Кількість попередніх покупок (Previous Purchases)
- Вік (Age)

На скріншоті зображено графік, де по осі X — реальні значення Purchase Amount (USD), а по осі Y — передбачені цією моделлю значення.

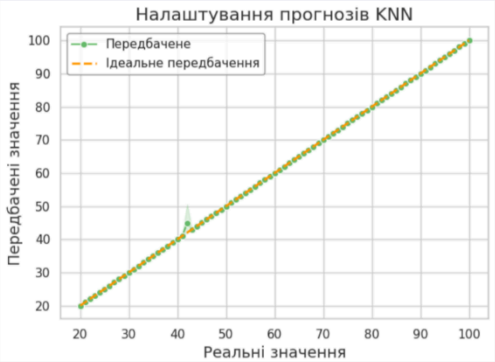
Сформована модель з прогнозуванням **НЕ Є ІДЕАЛЬНОЮ**:

- Вона не дає точних прогнозів для всіх значень.
- Може бути покращена шляхом оптимізації параметрів, вибору інших ознак

ДУІКТ

Практична частина

ПОШУК ТА РЕАЛІЗАЦІЯ НАЙКРАЩИХ ГІПЕРПАРАМЕТРІВ ДЛЯ СТВОРЕННЯ ІДЕАЛЬНОЇ МОДЕЛІ



Результати натренованої моделі:

- mse KNN: 0.37
- r2 KNN: 1.00
- rmse KNN: 0.61

Згідно з цим графіком, модель KNN працює дуже добре, адже:

- Передбачені значення (зелена лінія з точками) майже повністю співпадають з ідеальними передбаченнями (помаранчева пунктирна лінія).
- Усі точки лежать дуже близько до ідеальної діагоналі, тобто модель майже точно відтворює реальні значення.
- Видимих великих відхилень або систематичних помилок немає.

ДУІКТ

Висновки

Проведено комплексний аналіз сучасних методів прогнозування поведінки споживачів.
У ході виконання роботи було:

- Досліджено особливості застосування машинного навчання для аналізу покупок, виявлення ключових факторів впливу та формування персоналізованих маркетингових стратегій
- Розглянуто та порівняно основні підходи до аналізу даних
- Виконано одновимірний, двовимірний та багатовимірний аналіз для виявлення впливу демографічних та товарних характеристик на купівельну активність.
- Протестовано дві моделі машинного навчання: KNN і SVR і обрано найкращу за результатами тестів і проведено порівняльну оцінку моделей за метриками точності (MAE, RMSE, R^2) для вибору найефективнішої для прогнозування суми покупки.
- Встановлено, що модель KNN забезпечує кращу точність прогнозування
- Сформовано практичні рекомендації для покращення якості прогнозування

Результат можна покращити шляхом розширення набору даних, використання методів аугментації, оптимізації гіперпараметрів та впровадження додаткових ознак для підвищення точності моделі.

Створена система прогнозування може бути використана для бізнесу.

Розроблені підходи та моделі здатні підвищити ефективність маркетингових кампаній, покращити персоналізацію пропозицій та сприяти зростанню конверсій у сфері онлайн-торгівлі.

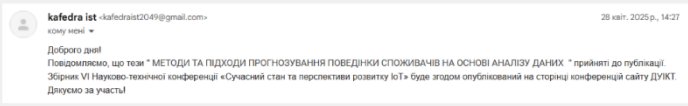
ДУІКТ

Висновки

УЧАСТЬ В НАУКОВО-ТЕХНІЧНІЙ КОНФЕРЕНЦІЇ

Результати дослідження, що вкладені у кваліфікаційній роботі, були представлені на VI Науково-технічній конференції «Сучасний стан та перспективи розвитку IoT» (15 квітня 2025 року) у вигляді тези “МЕТОДИ ТА ПІДХОДИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ СПОЖИВАЧІВ НА ОСНОВІ АНАЛІЗУ ДАНИХ”. Згодом її буде опубліковано на сторінці конференцій сайту ДУІКТ.

Це підтверджує не лише академічну цінність отриманих результатів, але й їхню практичну актуальність в контексті розвитку IoT та покращені результативності рекламних кампаній для інтернет-магазинів.



kafedra ist <kafedraist2049@gmail.com> 28 квіт. 2025 р., 14:27
кому мені

Доброго дня!
Повідомляємо, що тези " МЕТОДИ ТА ПІДХОДИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ СПОЖИВАЧІВ НА ОСНОВІ АНАЛІЗУ ДАНИХ " прийняті до публікації.
Збірник VI Науково-технічної конференції «Сучасний стан та перспективи розвитку IoT» буде згодом опублікований на сторінці конференцій сайту ДУІКТ.
Дякуємо за участь!

ДУІКТ

Дякую за увагу!