

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут
Інститут інформаційних технологій в економіці**

Кафедра математичного моделювання та статистики

Освітньо-професійна програма

Галузь знань

Спеціальність

Економічна кібернетика та Дата Сайнс

05 соціальні та поведінкові науки

051 економіка

Форма навчання: очна (денна)

**КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА
РОБОТА**

на тему **«Моделювання споживчого вибору методами машинного
навчання»**

(назва теми)

здобувача Гачківського Олександра Олексійовича

(ПІБ, підпис)

Науковий керівник: доктор економічних наук,
професор Коляда Ю.В.

(науковий ступінь, учене звання, ПІБ)

(підпис)

**Робота допущена до захисту перед екзаменаційною
комісією з атестації здобувачів вищої освіти (ЕК)**

Завідувач кафедри кандидат фізико-математичних наук,
професор Великоіваненко Г.І.

(підпис)

Київ 2023

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 ТЕОРЕТИЧНІ ЗАСАДИ МОДЕЛЮВАННЯ СПОЖИВЧОГО ВИБОРУ	6
1.1. Особливості споживчого вибору	6
1.2. Моделювання поведінки споживача	10
1.3. Використання методів машинного навчання в сфері продажів	15
РОЗДІЛ 2 МОДЕЛЮВАННЯ СПОЖИВЧОГО ВИБОРУ АЛГОРИТМОМ APRIORI В СЕРЕДОВИЩІ RSTUDIO	32
2.1. Збирання, розвідувальний аналіз та візуалізація даних по транзакціям	32
2.2. Виділення асоціативних правил алгоритмом Apriori	43
2.3. Оцінювання та підвищення ефективності виділених асоціативних правил	49
РОЗДІЛ 3 ПОРІВНЯННЯ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ АЛГОРИТМАМИ APRIORI ТА ECLAT	58
3.1. Розширені можливості пакету arules для виявлення асоціативних правил в середовищі RStudio	58
3.2. Пошук асоціативних правил алгоритмом Apriori на основі даних про клієнтів банку	61
3.3. Вивчення асоціативних правил алгоритмом Eclat на основі даних про клієнтів банку	69
ВИСНОВКИ	75
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	77

ВСТУП

Актуальність теми. Чи можна передбачити поведінку споживачів? Яким чином клієнти роблять свій вибір, і що, зрештою, відіграє вирішальну роль - низька ціна, зручність розташування чи ввічливість і привітна посмішка продавця, який спілкується з клієнтами? Як оцінити реакцію споживачів на зміни в маркетинговій стратегії або запуск нового продукту? Керівники компаній багато чого б віддали, щоб отримати відповіді на ці запитання. Ще десять-п'ятнадцять років тому така постановка питання, можливо, викликала б посмішку або неабиякий скепсис. Однак сьогодні побудова подібних моделей ринку стала реальністю завдяки новітнім технологіям моделювання.

З іншого боку, останнім часом неухильно зростає інтерес до методів "виявлення знань" у базах даних. Сучасні бази даних мають дуже великі розміри, що досягають гіга і терабайтів, і тенденцію до подальшого збільшення, що спричинило потребу в розробленні ефективних масштабованих алгоритмів, які дають змогу розв'язувати поставлені завдання за прийнятний час. Одним із популярних методів виявлення знань в даних стали алгоритми машинного навчання для знаходження різного роду закономірностей.

Машинне навчання останніми роками набуло широкого розповсюдження в багатьох галузях науки і техніки. Зростання його популярності зумовлене тим, що реалізовані ним підходи до побудови моделей ґрунтуються на автоматизованому аналізі емпіричних даних.

Універсальність алгоритмів машинного навчання зумовлена формалізацією типів математичних задач, на вирішення яких вони спрямовані. Виділяють такі завдання, як класифікація, кластеризація, регресія, афінітивний аналіз, пошук аномалій тощо. Кожен алгоритм машинного навчання пов'язаний із розв'язанням одного з цих типів завдань.

У цьому дослідженні передбачається, що існує можливість розширення прикладних можливостей алгоритмів машинного навчання за рахунок

розроблення способів перенесення алгоритмів на розв'язання інших, нетипових задач.

Аналіз останніх досліджень і публікацій. Використання машинного навчання в маркетингу було розглянуто в наукових роботах вітчизняних і зарубіжних вчених. Розробниками теорії та підходів споживчого маркетингу були К.Бове, Т.Левіт, Т.Лінч, О.Отлакан, Ю.Халліган та інші. Дослідженню теоретичних та прикладних аспектів машинного навчання присвячено наукові роботи А. Матвійчука, В. Соловійова, В.Дербенцева та інших.

Метою магістерської роботи є аналіз та моделювання споживчого вибору методами інтелектуального аналізу та алгоритмами машинного навчання.

Для досягнення поставленої мети в дослідженні вирішуються такі завдання:

- визначити сутність споживчого вибору як ефективного інструменту дослідження в маркетингу, зокрема в сфері продажів;
- зробити огляд шляхів та напрямків використання алгоритмів машинного навчання в бізнесі;
- узагальнити засоби збирання та підготовки даних для виявлення в них закономірностей;
- розглянути можливості пакету *arules* мови програмування R для застосування його при аналізі та моделюванні виявлених закономірностей;
- вивчити особливості представлення даних для вивчення асоціацій в середовищі RStudio;
- розглянути алгоритми виявлення асоціативних правил на різних прикладних задачах та провести їх порівняльний аналіз.

Об'єктом дослідження є транзакції споживчого вибору у роздрібній торгівлі та сфері банківських послуг.

Предметом дослідження є алгоритми машинного навчання вивчення закономірностей в транзакційних даних.

Методи дослідження. У ході дослідження використовувалися загальнонаукові методи аналізу та синтезу, підходи та методи дослідження споживчого вибору, висвітлені в наукових працях вітчизняних та зарубіжних вчених, рекомендації фахівців з питань розвитку та застосування алгоритмів машинного навчання, які є основними інструментами в даній роботі. Основою дослідження є праці провідних вчених, практиків з теорії й методології здійснення економіко-математичного моделювання в маркетинговій діяльності, теорії та прикладних аспектів використання методів Machine Learning та статистики.

Теоретична, методологічна та практична значимість одержаних результатів полягає в тому, що застосування базових алгоритмів асоціації розширює коло прикладних задач, які можна розв'язувати подібними алгоритмами. Практичні результати роботи можуть бути використані не лише в маркетингових дослідженнях ринків, а й для аналізу потреб клієнтів у сфері обслуговування, зокрема у сфері роздрібних банківських послуг, у крос-маркетингу, адресній розсилці клієнтам, у медицині при визначенні впливу побічних ефектів від ліків, в аналізі даних з перепису населення, у прогнозуванні збоїв телекомунікаційного обладнання тощо.

Інформаційна база дослідження. Інформаційною базою дослідження слугують відкриті дані Державної Служби Статистики України, ; наукові статті та підручники українських дослідників; зарубіжні тематичні сайти, тематикою яких є машинне навчання та інтелектуальний аналіз даних; форуми та інтернет курси стосовно особливостей програмування мовою R в системі RStudio.

Структура роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ЗАСАДИ МОДЕЛЮВАННЯ СПОЖИВЧОГО ВИБОРУ

1.1. Особливості споживчого вибору

У процесі своєї життєдіяльності людина має задовольняти свої базові потреби. Згідно піраміди Маслоу це - харчуватися, одягатися, захищатися від несприятливих погодних умов, протистояти і підтримувати свій організм у нормальному стані [3]. Потреби більшості людей практично нескінченні, однак завжди існують обмеження для їх повного задоволення. Тому виникає необхідність економічного вибору щодо розподілу наявних власних ресурсів шляхом раціонального використання різноманітних благ для задоволення потреб. Власні ресурси обмежені у своїх обсягах і можуть бути розподілені між: поточним і майбутнім споживанням; товарами довготривалого користування та одномоментного споживання; малою кількістю дорогих і великою кількістю дешевих товарів.

Особливості споживчої поведінки та необхідність її раціоналізації мають величезне практичне значення як для кожної окремої людини або груп людей у вигляді споживачів або виробників, так і для всього світового співтовариства загалом як єдиної економічної системи.

Виходячи з припущення, що споживач поводить на ринку раціонально [8], можна сказати, що він прагне отримати якісний товар за обмежених можливостей. Об'єктивного критерію для визначення найкращого набору товарів для конкретного споживача не існує, оскільки споживач здійснює свій вибір залежно від індивідуального власного сприйняття потреб. Однак слід зазначити, що поводитися раціонально на ринку зовсім не означає бути скупим і розважливим. Теорія споживчої поведінки визначає раціональним

споживачем ту особу, яка вкладає кошти найбільш прийнятним для неї способом [10]. Адже кожен споживач витрачає кошти згідно своїх бажань і при цьому прагне досягти максимального задоволення своїх потреб.

Основним обмеженням для споживача є розмір його доходу. Оскільки потреби різноманітні й безмежні, а дохід обмежений, то покупець постійно опиняється перед вибором придбання того чи іншого товару. При цьому він хоче придбати найкращий набір товарів з тих, які дозволяє йому дохід. Тому теорія споживчого вибору ґрунтується на таких постулатах [5]:

1. Різноманіття видів споживання. Кожен споживач бажає споживати безліч різноманітних індивідуальних благ.

2. Ненасиченість. Споживач прагне мати більшу кількість будь-яких товарів і послуг, він не пересичений жодним із них.

3. Транзитивність. Теорія споживчого вибору базується на сталості та відповідній узгодженості смаків споживача.

4. Субституція. Споживач згоден відмовитися від невеликої кількості блага А, якщо йому запропонують натомість більшу кількість блага-субститута.

5. Спадає гранична корисність. Гранична корисність будь-якого блага залежить від його загальної кількості, якою володіє даний споживач [7].

Здатність блага задовольняти потребу характеризується його корисністю, яка, в свою чергу, проявляється через споживання. Закон спадної граничної корисності полягає в тому, що *зі зростанням споживання якогось одного блага загальна корисність, яку отримує споживач, зростає, але зростає щоразу повільніше*. Інакше кажучи, закон спадної граничної корисності свідчить, що функція корисності є зростаючою та опуклою вгору. З цього закону випливає необхідність зниження ціни для того, щоб спонукати споживача до збільшення покупок такого продукту.

Однак цей закон не завжди діє за малих кількостей товару. Водночас із загальними принципами вибору раціонального споживача існують особливості, які визначаються його смаками та вподобаннями. Американський

економіст Х. Лейбенстайн ділить споживчий попит на дві великі групи: функціональний і не функціональний [8].

Функціональним попитом є попит на блага, який визначається винятково споживчими властивостями, що притаманні економічному благу (товар/послуга). Функціональний попит визначається при максимізації функції корисності споживача, він спадає або зростає залежно від якостей цього блага з точки зору споживача. Нефункціональний попит зумовлений чинниками, які не пов'язані з властивостями та якостями економічного блага. Тобто попит на дане благо зростає або спадає незалежно від того, якою є його якість і наскільки його споживання максимізує функцію корисності споживача. Тут "включаються" психологічні закони поведінки людей, тому поняття раціональності відступає на другий план. У нефункціональному попиті можна виокремити соціальні, спекулятивні і нераціональні чинники, що впливають на покупця [4].

Соціальний чинник зумовлений ставленням покупців до товару. Одні прагнуть витримати загальний стиль і купують товари, які купують ті, на кого вони рівняються. Інші прагнуть досягти винятковості, а не плисти за течією. Зрештою, треті досягли такого рівня життя, важливим аспектом якого стає демонстративне споживання. Тому Х. Лейбенстайн виокремлює три типові взаємні впливи [4]:

1. Ефект приєднання до більшості. Споживач, прагнучи не відставати від інших, купує те, що купують інші. Під ефектом приєднання до більшості розуміють ефект збільшення споживчого попиту, пов'язаний із тим, що споживач, дотримуючись загальноприйнятих норм, купує той самий товар, який купують інші. Саме ефектом приєднання до більшості й пояснюються різні віяння моди, коли люди купують речі, абсолютно непотрібні, але які купують усі, тому що це модно.

2. Ефект сноба. В цьому випадку у споживача яскраво виражається прагнення виділитися з натовпу. У зв'язку з цим під ефектом сноба розуміється ефект зміни попиту через те, що інші люди споживають саме цей товар. Якщо

інші збільшують споживання цього товару, то сноб його скорочує. Тому, якщо ефект сноба присутній, то крива попиту стає менш еластичною. Покупець сноб нізащо не купить те, що купують усі.

3. Ефект Веблена. Таким терміном Х. Лейбенстайн називає престижне або демонстративне споживання, яке наочно описано в [8], коли товари купують лише для того, аби справити незабутнє враження на інших. Ціна товару в цьому випадку складається з двох складових частин: її реальної вартості та престижної. Тому під ефектом Веблена розуміється ефект збільшення споживчого попиту, який спричинений тим, що товар має вищу (а не нижчу) вартість. Між ефектами Веблена та сноба є схожість. Але принципова відмінність полягає в тому, що ефект сноба залежить від розмірів споживання інших, тоді як ефект Веблена насамперед залежить від розміру ціни [2].

Крім того, у нефункціональному попиті Лейбенстайн виокремлює явища спекулятивного і нераціонального попиту. Спекулятивний попит виникає в суспільстві з високими інфляційними очікуваннями, коли небезпека підвищення цін у майбутньому стимулює додаткове споживання (купівлю) товарів у теперішньому. Нераціональний попит - це незапланований попит, що виник під впливом "одномоментного" бажання, раптової зміни настрою, забаганки чи примхи, тобто попит, який порушує передумову про раціональну поведінку споживача.

Важливо зауважити, що багато людей певною мірою схильні до спалахів нераціонального попиту і нерідко здійснюють покупки, про які зазвичай потім шкодують.

1.2. Моделювання поведінки споживача

Раніше маркетологи вивчали своїх споживачів у процесі безпосереднього контакту (торговельної діяльності), але у зв'язку зі зростанням кількості підприємств і обсягів ринку це стає доволі складним явищем, а часто й неможливим. Тому виникає питання про вивчення чинників, які впливають на зміни споживчої поведінки.

Нині споживче суспільство, зокрема України, дуже швидко розвивається, наближаючись до світових стандартів обслуговування та культури споживання. У зв'язку з тим, що основним суб'єктом ринкових відносин є споживач, який постійно формує свою поведінку, то це призводить до того, що він перебуває в перманентному стані економічного вибору. Дана тенденція пов'язана, насамперед, з тим, що відбувається зміна стилю життя українських споживачів, яка є наслідком відкритості кодонів, доступу до величезного обсягу інформації та набутого досвіду отримання різноманітних видів товарів і послуг.

Американський автор і дослідник у галузі футурологічних досліджень Джон Нейсбітт зазначає у своїх дослідженнях, що ми перебуваємо між двома видами суспільства: індустріальним та інформаційним [32]. На його думку, каталізатором процесу переходу від індустріального суспільства до інформаційного є підприємці, кількість яких, згідно зі світовими статистичними даними, з кожним роком збільшується. Проте в Україні за останні три роки склалася інша ситуація, спричинена тривалим періодом карантинних обмежень, повномасштабним вторгненням РФ, окупацією східних та південних областей, постійними військовими діями (табл. 1.1 та рис.1.1)

Таблиця 1.1. - Кількість діючих суб'єктів господарювання за видами економічної діяльності України в період 2020-2022 рр. (у % до загального показника)

Види діяльності підприємств	2020	2021	2022
Оптова та роздрібна торгівля	41,9	39,7	38,2
Інформація та телекомунікації	11,9	14,5	17,1
Професійна, наукова та технічна діяльність	7,1	7,3	7,9
Промисловість	6,4	6,2	5,9
Надання інших видів послуг	5,9	5,7	5,4
Транспорт, поштова та кур'єрська діяльність	5,0	4,8	5,1
Операції з нерухомим майном	4,9	4,8	4,6
Тимчасове розміщування й організація харчування	3,6	3,6	3,3
Сільське, лісове та рибне господарства	3,7	3,6	3,1
Будівництво	2,9	2,9	2,6
Діяльність адміністративного та допоміжного обслуговування	2,7	2,8	2,6
Охорона здоров'я та надання соціальної допомоги	1,9	1,8	1,9
Освіта	0,8	0,9	1,0
Мистецтво, спорт, розваги та відпочинок	0,8	0,8	0,8
Фінансова та страхова діяльність	0,5	0,6	0,5

Джерело: складено автором на основі [13]

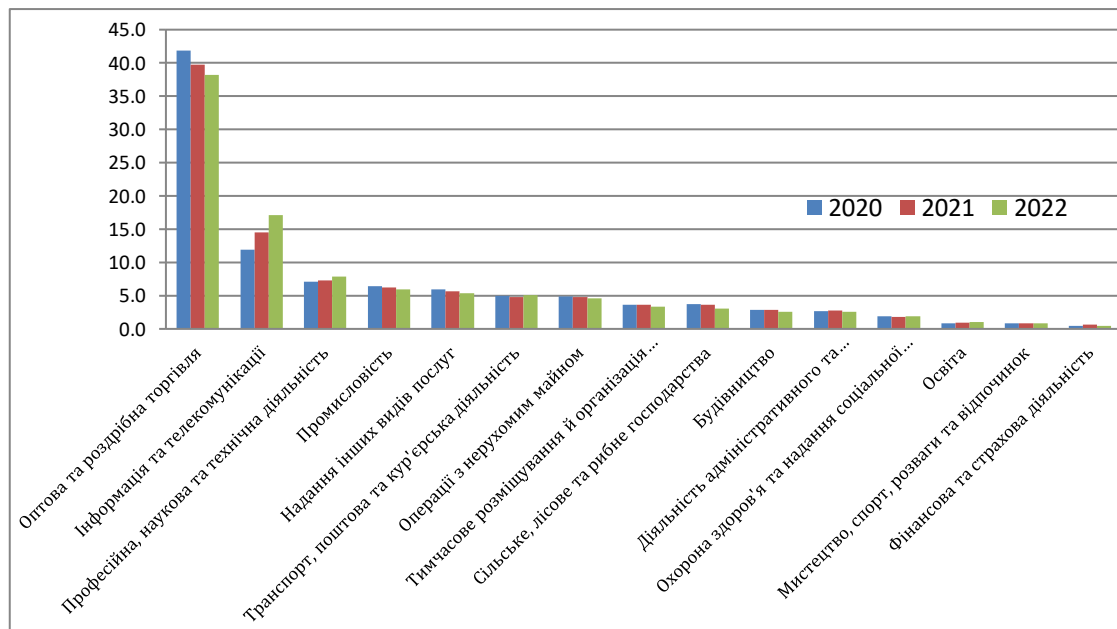


Рисунок 1.1 - Діаграма розподілу кількості діючих суб'єктів господарювання за видами економічної діяльності України за період 2020-2022рр. (у % до загального показника)

Джерело: складено автором на основі [13]

Представлені дані (табл.. 1.1 та рис.1.1) ще раз доводять той факт, що зі збільшенням підприємницької активності споживачі мають можливість отримання різноманітних видів послуг, і внаслідок цього змінюється споживча поведінка на ринку.

Американський соціолог і публіцист, творець теорії постіндустріального (інформаційного) суспільства Деніел (Деніел) Белл, який описує себе як "соціаліста в економіці, ліберала в політиці та консерватора в культурі", висунув положення про те, що постіндустріальне суспільство характеризується зниженням промислового виробництва за рахунок зростання сфери послуг та інформації [37]. Основна його ідея полягає в тому, що, незважаючи на важливість технічного прогресу, який є основним чинником, що формує попит на нові інноваційні товари, отримують свій розвиток нові стандарти споживання, орієнтовані не на "масові", а на індивідуальні потреби людей. У зв'язку з цим виробникам товарів і послуг необхідно приділяти більше уваги не тільки технологіям виробництва, а й маркетинговим стратегіям просування, тому основне завдання маркетингу полягає у визначенні тенденцій зміни поведінки споживачів у різні часові періоди.

Отже, необхідно зазначити, що основним інструментом маркетингового дослідження, а також базою для розроблення та використання моделей відгуку ринку і моделей для розроблення маркетингової політики виступає моделювання поведінки споживачів. Споживачі утворюють складну соціально-психологічну систему, яка має індивідуальні, психологічні особливості кожного конкретного індивідуума.

Незважаючи на те, що в даний час існує безліч методик щодо визначення споживчої поведінки, проте, всі вони не можуть дати однозначної відповіді щодо споживацького вибору, або пояснити чи передбачити дію споживачів. У зв'язку з цим можна говорити про використання моделей, які про описують узагальнену характеристику дій покупців. Під моделлю зазвичай розуміють спрощений логічний, графічний або математичний опис будь-якого об'єкта, що відображає істотні його властивості [38].

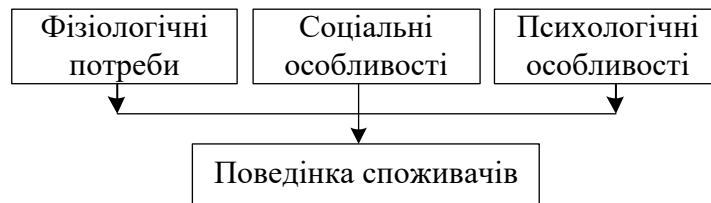


Рисунок 1.2 – Фактори моделювання поведінки споживачів

Джерело: складено автором на основі [42]

Моделювання споживчої поведінки можна уявити як поєднання соціальних, психологічних особливостей особи, а також її фізіологічних потреб (рис. 1.2). Крім цього модель споживчої поведінки можна представити як сукупність особистісних якостей споживача та його споживчого досвіду [2].

Існує кілька підходів до моделювання поведінки споживачів: психологічний, соціологічний та економічний (рис. 1.3).

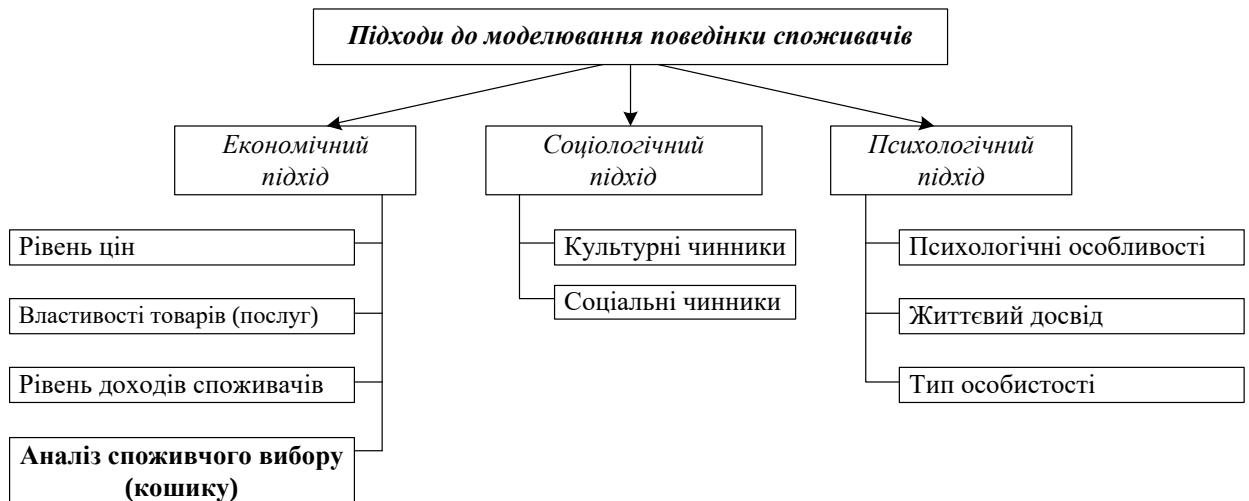


Рисунок 1.3 – Підходи до моделювання поведінки споживачів

Джерело: складено автором на основі [7]

Отже, модель споживчої поведінки – це, з одного боку, умовний образ споживача, створений для вивчення його поведінки, а з іншого - розроблення оптимальних маркетингових дій компанії з використанням таких маркетингових інструментів, як товарна пропозиція, ціноутворення, реклама, стимулювання, брендинг тощо [2].

Сутність психологічного підходу полягає в тому, що визначальну роль відіграють внутрішні якості споживача: на його купівельну поведінку впливають психологічні особливості, накопичений досвід, а також тип особистості споживача. За соціологічного підходу важливу роль відіграє

суспільне середовище, до якого належить або бажає належати споживач. У цьому разі розглядається вплив культурних і соціальних чинників на купівельну поведінку.

Економічний підхід базується на тому, що споживач у своїх діях раціональний і ґрунтується на правилі "максимізації корисності". Під час розгляду цього підходу необхідно зазначити, що основними чинниками, які впливають на поведінку споживачів, є ціни на послуги та доходи споживачів.

Серед наявних моделей споживчої поведінки загалом, зокрема моделі споживчого вибору, розглядають процес прийняття рішення покупців щодо споживчого вибору (здійснюваної покупки). Для цього загальний процес поділяють на такі етапи: осмислення/усвідомлення потреби; аналіз наявних пропозицій, здійснення покупки (рис. 1.4).



Рисунок 1.4 – Узагальнення процесу споживчого вибору

Джерело: складено автором на основі [7]

Представлені на рис.1.4 етапи прийняття купівельного рішення узагальнено відображають процес ухвалення рішення споживачів про вчинену дію. Будь-який процес розпочинається з усвідомлення споживачами своїх потреб, які зрештою або можуть бути задоволені (споживчий вибір), або відкинуті в процесі ухвалення відповідних рішень. На процес формування потреб споживачів, аналіз інформації про товари (послуги) впливають як фактори споживача (внутрішні фактори), так і фактори виробництва (зовнішні фактори), які й сприяють прийняттю відповідного рішення.

Вмотивований на здійснення покупки споживач вивчає, звертаючись до додаткових джерел інформацію, аналізує наявні пропозиції та приймає

рішення про покупку. На етапі оцінки різних альтернативних пропозицій споживач аналізує представлені йому варіанти, на підставі чого в нього формується індивідуальна думка про кожну пропозицію.

На етапі здійснення покупки споживач купує товари (послуги), що його найбільше зацікавили. Після здійснення покупки споживач або задоволений отриманою ним послугою (товаром), або ні. У результаті він, можливо, буде здійснювати повторну покупку необхідного йому товару (послуги) або ж обере нового продавця з аналогічними пропозиціями – так формуються купівельні переваги. Важливо пам'ятати, що на кожному з етапів процесу прийняття купівельного рішення необхідно проводити аналіз чинників, що впливають на покупця, аналіз його купівельного кошика, що дасть змогу професійно впливати на процес прийняття таких рішень.

1.3. Використання методів машинного навчання в сфері продажів

Моделювання споживчого кошика є однією з важливих напрямків економічного аналізу та прогнозування у галузі продажів. Цей процес дозволяє вивчати та прогнозувати зміни у виборі товарів та послуг споживачами залежно від різних факторів, таких як ціни, доходи, психологічні чинники тощо.

Технології інтелектуального аналізу даних (data mining, DM) [1, 8] орієнтовані на пошук взаємозв'язків, прихованих закономірностей у різних за своїм обсягом і структурою масивах інформації. При цьому інтелектуальний аналіз даних передбачає формування набору даних із декількох джерел та їх організацію для отримання потрібної інформації.

Data mining - це використання статистики та методів програмування для пошуку шаблонів, прихованих усередині даних, що описують і пояснюють деякі явища.

Машинне навчання (machine learning, ML) - це методи аналізу даних, які автоматизують побудову аналітичних моделей. Це галузь штучного інтелекту, заснована на ідеї, що системи можуть вчитися на даних, виявляти шаблони (патерни) та приймати рішення з мінімальним втручанням людини. Машинне навчання використовує методи інтелектуального аналізу даних і алгоритми навчання для побудови моделей на основі цих даних для прогнозування майбутніх результатів.

Штучний інтелект використовує моделі, які створені машинним навчанням, та інші способи відображення навколишнього світу для того, щоб відтворювати інтелектуальну поведінку об'єктів.

Отже, DM і ML формують екосистему прийняття обґрунтованих рішень. Ці технології доповнюють одна одну, і використовувати їх поодинокі - означає обмежувати їхній потенціал. DM і ML перетинаються з новою, але вже досить широко застосованою в різних сферах діяльності наукою про дані (Data Science).

Сучасні компанії не можуть жити в умовах браку даних. Вони повинні розвиватися і йти в ногу з розвитком технологій і майбутніми цифровими тенденціями, щоб випереджати конкурентів. Сьогодні компанії приділяють пріоритетну увагу тому, щоб бути в курсі всіх нових розробок у сфері Data Science. Необроблені дані здебільшого складаються з неструктурованих даних, які не можуть бути вставлені в стандартну електронну таблицю або базу даних. Складні алгоритми інтелектуального аналізу даних дають змогу компаніям аналізувати необроблені ("сирі") дані великих обсягів, які одержані з різних джерел, за допомогою сегментації, виявлення закономірностей і прогнозування результатів [1].

Сфера бізнесу. Багато популярних компаній використовують Data Science для оптимізації своїх процесів. Підприємства сьогодні стали все більше орієнтуватися на збирання, обробку та аналіз даних. Це означає, що підприємства всього світу використовують дані для прийняття рішень і стратегічного розвитку своєї компанії.

1. Аналітика для оптимізації ухвалення рішень. Традиційна бізнес-аналітика мала описовий та статичний характер. Але з використанням Data Science вона трансформувалася, перетворившись на більш динамічну сферу. Data Science дозволила бізнес-аналітиці охопити широкий спектр бізнес-операцій.

У зв'язку з масовим збільшенням обсягу даних компаніям потрібні фахівці з їх обробки з метою аналізу та отримання значущих висновків на їх основі. Змістовні висновки дають змогу компаніям аналізувати інформацію в великих масштабах і розробляти відповідні стратегії прийняття рішень. Прийняття рішень охоплює оцінку різноманітних чинників, що впливають на результат.

Процес прийняття рішень складається з чотирьох основних етапів:

- 1) розуміння контексту і характеру проблеми, яку потрібно вирішити;
- 2) дослідження та кількісна оцінка якості даних;
- 3) реалізація коректно обраних інструментів і алгоритмів для вирішення проблеми;
- 4) обмін досвідом і передача всередині команди знайдених інсайтів.

Отже, організаціям необхідні технології Data Science для полегшення процесу прийняття рішень та його обґрунтування.

2. Створення найкращих продуктів. Компаніям необхідно вміти залучати своїх клієнтів до продуктів. Їм необхідно розробляти продукти, що відповідають вимогам і потребам клієнтів. Процес включає в себе аналіз відгуків клієнтів. Цей аналіз проводиться з використанням передових аналітичних інструментів на основі Data Science.

Крім того, компанії враховують поточні тенденції ринку для використання розробленого продукту в широких масах споживачів. Ці тенденції ринку дають компаніям уявлення про поточну потребу у продукті. Бізнес розвивається разом з інноваціями. Зі зростанням обсягу даних компанії можуть впроваджувати не тільки нові продукти, а й різні інноваційні стратегії.

3. Ефективне управління бізнесом. Нині більшість компаній мають величезні масиви даних, які потрібно аналізувати та обробляти в подальшому. Платформи для обробки таких даних виявляють приховані закономірності, присутні всередині даних, і допомагають проводити змістовний аналіз і прогнозування подій.

Використовуючи Data Science, підприємства можуть значно ефективніше організувати своє управління. Як великі підприємства, так і невеликі стартапи можуть отримувати вигоду від застосування технологій ML і DM для подальшого зростання.

Фахівці в галузі Data Science допомагають аналізувати поточне становище бізнесу. За допомогою Data Science компанії можуть передбачати ймовірність успіху/невдачі своїх стратегій. Це допомагає в підведенні підсумків діяльності та результативності компанії. Data Science допомагає визначити ключові показники, які необхідні для діагностування ефективності бізнесу.

Базуючись на цьому, бізнес може вживати важливих заходів для кількісного оцінювання своєї діяльності та здійснення відповідних управлінських кроків. Це також може допомогти менеджерам аналізувати і виявляти потенційних кандидатів для бізнесу.

Використовуючи технології ML і DM, компанії розвивають лідерські якості серед конкурентів, відстежуючи продуктивність, рівень успіху та інші важливі показники. За допомогою аналізу робочої сили компанії можуть оцінити, що найкраще підходить для їхніх співробітників.

4. Прогнозна аналітика. Прогнозна аналітика є найважливішою частиною бізнесу. З появою передових інструментів і технологій прогнозування компанії розширили свої можливості по роботі з різними формами даних. Загалом прогнозна аналітика - це статистичний аналіз даних, який містить кілька алгоритмів машинного навчання для прогнозування майбутніх результатів з використанням історичних даних.

У бізнесі існують різні додатки прогнозової аналітики, як-от сегментація клієнтів, оцінювання ризиків, прогнозування продажів і аналіз ринку. За допомогою прогнозової аналітики компанії мають перевагу перед іншими, оскільки вони здатні передбачити майбутні події та впроваджувати відповідні заходи.

5. Оцінка бізнес-рішень.

Після прийняття управлінських рішень на основі прогнозів майбутніх подій компаніям необхідно їх проаналізувати та оцінити. Це можливо за допомогою декількох інструментів верифікації гіпотез. Після реалізації прийнятих рішень компанії повинні зрозуміти, як ці рішення впливають на їхню продуктивність і зростання. Якщо рішення призводить до якогось негативного результату, то компанії потрібно проаналізувати його й усунути проблему, яка негативно позначається на їхній роботі.

Існують різні процедури, за допомогою яких підприємства можуть оцінити свої рішення і спланувати відповідну стратегію дій. Ці рішення засновані на вимогах клієнтів, цілях компанії, а також потребах керівників проєктів. Крім того, застосовуючи нові технології ML і DM для оцінки майбутнього зростання на основі нинішнього курсу стратегії розвитку, підприємства можуть значно збільшувати прибуток.

6. Автоматизація процесів підбору персоналу. Впровадження Data Science зіграло ключову роль у забезпеченні автоматизації в різних напрямках розвитку бізнесу. Одним із таких процесів є перевірка резюме. Щодня компаніям доводиться мати справу з безліччю резюме претендентів.

Деякі великі компанії можуть навіть залучити тисячі резюме на вакансію. Щоб розібратися у всіх цих резюме і вибрати відповідного кандидата, компанії використовують технології ML і DM. Технології розпізнавання зображень здатні перетворювати візуальну інформацію з резюме в цифровий формат. Потім застосовуються різні алгоритми кластеризації або класифікації [6-8], щоб вибрати відповідного кандидата на

відповідну роботу. Крім того, компанії вивчають ринкові тенденції та аналізують потенційних претендентів на роботу. Це дає їм змогу зв'язатися з кандидатами й отримати глибоке уявлення про ринок претендентів.

У сучасному світі мільйони гігабайт інформації перебувають навколо нас, адже ми живемо в таку епоху цифрового розвитку, коли використовувати цю інформацію та обертати її на свою користь просто необхідно. Економічні, медичні, соціальні сфери, у цих і не тільки галузях використовують технологію Data mining, що відповідає на запитання "Як опрацювати цю інформацію з вигодою для себе або компанії?". На це питання дає відповідь технологія Data mining.

"Термін Data mining отримав свою назву з двох понять: пошуку цінної інформації у великих базах даних Data і видобуток гірничої руди Mining. Обидва ці процеси або вимагають величезної кількості обробки, або просіювання великої кількості матеріалу для розумного дослідження та пошуку шуканих цінностей" [1].

Суть методу Data mining полягає в тому, що за його допомогою можна відшукувати раніше невідомі знання, тобто знання мають бути новими, а не підтвердженням якихось раніше отриманих відомостей. Нетривіальних знань - таких, яких не можна просто так побачити під час побудови візуального аналізу або під час обчислення простих статистичних характеристик. Практично корисних - таких, які становлять цінність для дослідника.

Переваги Data mining:

- Мала кількість обмежень на застосування методів Data mining
- Пошук неочевидних закономірностей.
- Можливість роботи з багатовимірними і даними величезного обсягу і складної структури.
- Застосування методів Data mining - можливість отримати користь із накопиченої інформації.

Основні завдання, що розв'язують методом Data mining:

Класифікація - визначення класу об'єкта за його характеристиками.

Кластеризація - пошук незалежних груп і характеристик у всій множині даних.

Прогнозування (регресія) - подібно до класифікації, дає змогу визначити за відомими характеристиками об'єкта значення деякого його параметра, значенням параметра є множина дійсних чисел.

Асоціація - знаходження частих взаємозв'язків між об'єктами або подіями, також можуть бути використані для передбачення появи подій.

"Асоціація - це функція інтелектуального аналізу даних, яка виявляє ймовірність одночасної появи елементів у наборі. Відношення між супутними елементами виражаються у вигляді асоціативних правил" [2].

Правила асоціації часто використовують для аналізу угод купівлі-продажу. Наприклад, можна зазначити, що клієнти, які купують пластівці в продуктовому магазині, часто купують молоко одночасно. Фактично, асоціативний аналіз може виявити, що 85% касових сесій, що включають пластівці, також включають молоко" [3]. Цей додаток асоціативного моделювання називається аналіз кошика. Це цінно для прямого маркетингу, просування продажів і виявлення тенденцій у бізнесі. Аналіз кошика також можна ефективно використовувати для макета магазину, дизайну каталогу та перехресних продажів.

Алгоритми знаходження асоціативних правил. Асоціативні правила щодня зустрічаються на різних сервісах, у магазинах, під час онлайн-закупівель та інших рекомендаційних системах. У загальному вигляді ARL (правило асоціації) звучить так - "Хто купив x, також купив y" [4]. "На даний момент існує кілька алгоритмів знаходження асоціативних правил, зокрема: AIS та його продовження SETM, Apriori, ECLAT і FPgrowth" [5].

Алгоритм Apriori є базовим та найпопулярніший методом виділення асоціативних правил. Він знаходить комбінації, які часто зустрічаються в базі даних.

Кожна транзакція є бінарним вектором, де $t * [k] = 1$ якщо i_k елемент присутній в транзакції, інакше $t * [k] = 0$. Тобто, транзакція T містить деякий набір елементів X з усієї множини наборів E , якщо $X \subset T$. Тоді асоціативним правилом називається така імплікація:

$$X \rightarrow Y, \text{ де } X: E, Y \subseteq E, X \subset Y \neq \emptyset \quad (*)$$

Правило $X \Rightarrow Y$ має підтримку s (support), якщо $s\%$ транзакцій з множини D , містять:

$$X \subseteq Y, \text{supp}(X \rightarrow Y) = \text{supp}(X \cup Y). \quad (**)$$

Достовірність правила показує, яка ймовірність того, що з X випливає Y . Правило $X \Rightarrow Y$ справедливе з достовірністю (confidence) c , якщо $c\%$ транзакцій із D , що містять X , також містять:

$$Y, \text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X) \quad (***)$$

У цій роботі пропонується використовувати алгоритм Apriori, що став першим ефективним алгоритмом пошуку частих множин ознак, з урахуванням також приналежності користувача до заздалегідь визначеної групи. Алгоритм Apriori є порівневим, використовує стратегію пошуку в ширину і здійснює його знизу-вгору. В алгоритмі використовуються дві структури даних: C_i - для зберігання множини кандидатів у множині ознак довжини i та F_i - для зберігання частих множин ознак довжини i . Кожна структура має два поля - itemset, що зберігає множину ознак, і support, яке зберігає величину підтримки цієї множини ознак.

У цьому разі необхідно врахувати можливість відстеження споживчих звичок усіх покупців і використовувати зібрану інформацію, щоб запропонувати товари, які їх можуть зацікавити. Наприклад, запропонувати фільми, які споживачеві, можливо, сподобаються, хоча раніше він купував тільки книги. Аналогічно деякі сайти з продажу квитків на концерти аналізують, що користувачі відвідували раніше, і анонсують майбутні концерти, які можуть бути їм цікаві. Отже, виявлення закономірностей з використанням асоціативних правил для генерації рекомендацій користувачам

інформаційного порталу є досить актуальним завданням, що також підтверджується експериментальними дослідженнями.

Асоціативні правила мають такий вигляд:

{якщо (умова) то (результат)},

де умова - зазвичай не логічний вираз, а набір об'єктів із множини I , з яким пов'язані (асоціації) об'єкти, що включені до результату цього правила. Наприклад, асоціативне правило: "якщо (молоко, масло), то (хліб)" означає, що якщо споживач купує молоко і масло, то він купить і хліб. Основною перевагою асоціативних правил є їхнє легке сприйняття людиною і проста інтерпретація мовами програмування.

Однак, вони не всі є корисними. Виділяють три види правил:

- корисні правила - містять дійсну інформацію, яка раніше була невідома, але має логічне пояснення. Такі правила можуть бути використані для прийняття рішень, що приносять вигоду;

- тривіальні правила - містять дійсну і легко пояснювану інформацію, яка вже відома. Такі правила не можуть бути корисними, бо відображають або відомі закони в досліджуваній галузі, або результати минулої діяльності. Іноді такі правила можуть використовуватися для перевірки виконання рішень, прийнятих на підставі попереднього аналізу;

- незрозумілі правила - містять інформацію, яка не може бути пояснена.

Такі правила можуть бути отримані на основі аномальних значень, або суто прихованих знань. Безпосередньо такі правила не можна використовувати для ухвалення рішень, їхня нез'ясовність може призвести до непередбачуваних результатів.

Для кращого розуміння потрібен додатковий аналіз. Асоціативні правила будуються на основі часто повторюваних наборів. Так правила, побудовані на підставі набору F , є можливими комбінаціями об'єктів, які входять до нього. Отже, кількість асоціативних правил може бути дуже великою і важкосприйнятною для людини. Крім того, не всі з побудованих

правил несуть у собі корисну інформацію. Для оцінки їхньої корисності вводяться такі величини:

- *Підтримка (support)* - показує, який відсоток транзакцій підтримує це правило. Оскільки правило будується на підставі набору, то, отже, правило має підтримку, що дорівнює підтримці набору F , який складають X і Y :

$$Supp_{X \Rightarrow Y} = Supp_F = \frac{|D_{F=X \cup Y}|}{D}. \quad (1.1)$$

Зрозуміло, що правила, побудовані на підставі одного й того самого набору, мають однакову підтримку, наприклад, рівень підтримки $Supp\{\text{якщо (вода, олія), то (горіхи)}\} = 0,5$.

- *Достовірність (confidence)* - відображає ймовірність того, що з наявного набору X в транзакціях впливає наявність і набору Y . Достовірністю правила є відношення числа транзакцій, які містять X та Y , до числа транзакцій, які містять лише множину X :

$$Conf_{X \Rightarrow Y} = \frac{|D_{F=X \cup Y}|}{|D_X|} = \frac{Supp_{X \cup Y}}{Supp_X}. \quad (1.2)$$

Чим більша достовірність, тим правило краще. причому у правил, побудованих на підставі одного і того ж набору, достовірність буде різна, наприклад:

- $Conf\{\text{якщо (вода), то (горіхи)}\} = 2/3$;
- $Conf\{\text{якщо (горіхи), то (вода)}\} = 2/3$;
- $Conf\{\text{якщо (вода, олія), то (горіхи)}\} = 1$;
- $Conf\{\text{якщо (вода), то (горіхи, олія)}\} = 2/3$.

Достовірність не дає змоги визначити корисність правила. Відсоток наявності в транзакціях набору Y за умови наявності в ньому набору X менший, ніж відсоток безумовної наявності набору Y :

$$Conf_{X \Rightarrow Y} = \frac{Supp_{X \cup Y}}{Supp_X} < Supp_Y. \quad (1.3)$$

Імовірність випадково вгадати наявність у транзакції набору Y більша, ніж передбачити це за допомогою правила. Для виправлення такої ситуації вводиться міра - покращення.

- *Покращення (improvement)* - показує, чи є правило випадкового вгадування корисним. Покращення правила - це відношення числа транзакцій, котрі містять набори X і Y , до добутку кількості транзакцій, які містять лише набір X та кількості транзакцій, які містять набір Y :

$$impr_{X \Rightarrow Y} = \frac{|D_{F=X \cup Y}|}{|D_X| |D_Y|} = \frac{Supp_{X \cup Y}}{Supp_X * Supp_Y}. \quad (1.4)$$

Наприклад, $impr\{\text{якщо (вода, олія), то (горіхи)}\} = 0,5/(0,5*0,5) = 2$.

Якщо поліпшення більше одиниці, то це означає, що за допомогою правила передбачити наявність набору Y ймовірніше, ніж випадкове вгадування, якщо менше одиниці, то навпаки.

В останньому випадку можна використовувати від'ємне правило, яке передбачає відсутність певного набору:

$$\text{набір } Y: X \geq \text{не } Y$$

Алгоритм Apriori безпосередньо пов'язаний з аналізом кошика покупця. Постановка задачі полягає в тому, що є база даних одного, наприклад, магазину X , яка складається з чеків (транзакцій) клієнтів, які містять різні набори покупок. Іншими словами, кожна покупка - це транзакція, що визначає куплений набір товарів одним покупцем. Необхідно за допомогою алгоритму Apriori виявити асоціативні правила.

Для застосування алгоритму необхідно спочатку провести процедуру предобробки вхідних даних: привести всі дані до бінарного вигляду, а також необхідно змінити саму структуру представлення даних. Далі виконати ще два етапи роботи алгоритму Apriori. На першому зчитуються набори, які часто зустрічаються, при цьому необхідно заздалегідь задати потрібні значення підтримки та достовірності, а на другому - формуються з отриманих даних безпосередньо самі правила.

Крім аналізу споживчого вибору покупця цей алгоритм можна застосувати до безлічі інших сфер. Наприклад: медицина, фінанси, транспорт та різноманітні інші дослідження.

Узагальнені асоціативні правила. Під час пошуку асоціативних правил передбачалося, що всі аналізовані елементи є однорідними. Якщо повернутися до аналізу ринкового кошика, то це ті товари, які мають абсолютно однакові атрибути, за винятком назви.

Доповнимо транзакцію інформацією про те, до якої товарної групи входить товар, і побудуємо ієрархію товарів. Розглянемо приклад такого групування у вигляді ієрархічної моделі (рис. .

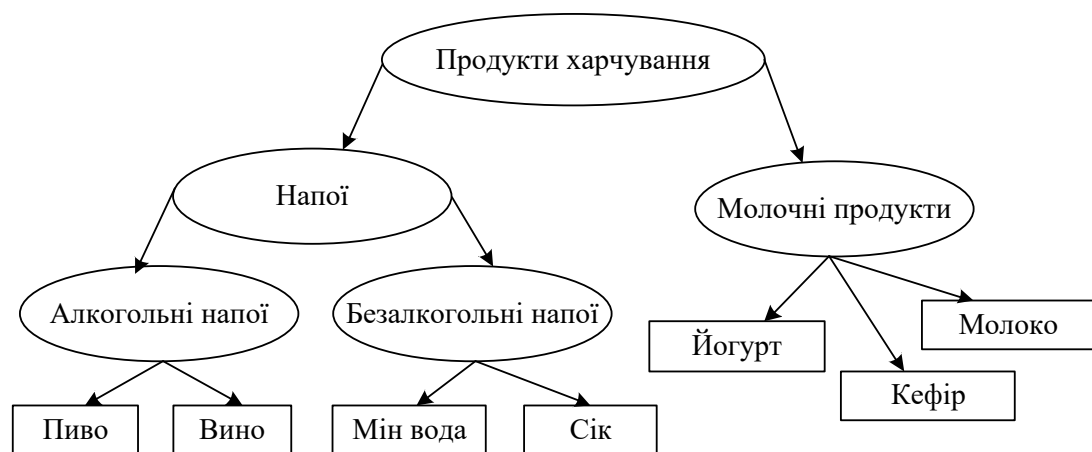


Рисунок 1.5 – Приклад ієрархічної моделі групування товарів

Джерело: складено автором на основі []

Нехай для деякої бази транзакцій D відомо, до яких груп входять елементи. Тоді можна добувати з даних правила, які пов'язують групи з групами або окремі елементи з групами. Наприклад, якщо Покупець купив товар із групи 'Безалкогольні напої', то він купить і товар із групи 'Молочні продукти' тобто 'Сік' - 'Молочні продукти'.

Такі правила називаються узагальненими асоціативними правилами, які можна відобразити такою імплікацією: $X \rightarrow Y$, де $X \subset I$, $Y \subset I$, $X \cap Y = \emptyset$

Де жоден з елементів, які входять до набору Y , не є предком жодного елемента з набору X . Підтримку і достовірність підраховують так само, як і у

разі асоціативних правил. Введення додаткової інформації про групування елементів у вигляді ієрархії може дати такі переваги:

- допомагає встановити асоціативні правила не тільки між окремими елементами, а й між різними рівнями ієрархії (групами);
- окремі елементи можуть мати недостатню підтримку, але загалом група може задовольняти поріг `minsupport`.

Для знаходження таких правил можна використовувати будь-який зі згаданих алгоритмів. Для цього кожному транзакцію потрібно доповнити всіма предками кожного елемента. Проте, безпосереднє застосування цих алгоритмів неминуче призведе до таких проблем:

- елементи на верхніх рівнях ієрархії тяжіють до значно більших значень підтримки порівняно з елементами на нижніх рівнях.
- з додаванням у транзакції груп збільшується кількість атрибутів і відповідно розмірність вхідного простору. Це ускладнює завдання, а також веде до генерації більшої кількості правил;
- поява надлишкових правил, котрі суперечать визначенню узагальненого асоціативного правила, наприклад, 'Сік' - 'Прохолодні напої'. Вочевидь, що практична цінність такого 'відкриття' нульова за 100% достовірності.

Отже, потрібні спеціальні оператори, здатні видаляти подібні надлишкові правила.

Для знаходження узагальнених асоціативних правил використовують спеціалізований алгоритм обчислення узагальнених асоціативних правил (FP-Growth, FPG), що усуває вищеописані проблеми і до того ж працює у 2-5 разів швидше, ніж стандартний Apriori. Групувати елементи можна не тільки за входженням у певну товарну групу, а й за іншими характеристиками, наприклад за ціною, брендом, рекомендаціями друзів.

В розділах 2 та 3 цієї роботи алгоритми Apriori та Eclat будуть розглянуті більш детально та з практичними прикладами.

Метод головних компонент (Principal component analysis). "Метод головних компонент - це технологія багатовимірного статистичного аналізу, що використовується для скорочення розмірності простору ознак з мінімальною втратою корисної інформації. Запропонований К. Пірсоном у 1901 р., а потім детально розроблений американським економістом і статистиком Г. Хоттелінгом" [8].

"З точки зору математики, цей метод є ортогональним лінійним перетворенням, яке дозволяє відображати дані вхідного простору в новому просторі меншої розмірності зі збереженням основних (значущих) даних" [9, с. 78-81]. При цьому побудову нової бази будують так, щоб дисперсія в новій компоненті була максимальною. Другу компоненту будують перпендикулярно до першої, щоб дисперсія даних була максимальною з решти можливих і так далі.

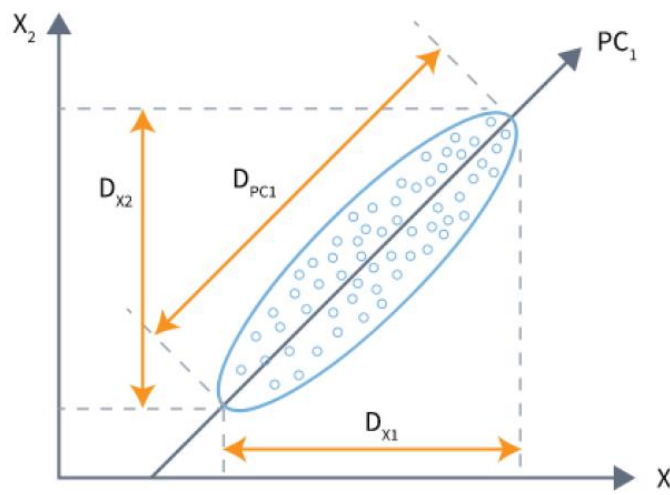


Рисунок 1.5 – Зниження розмірності вхідного 2-вимірного простору за допомогою методу головних компонент до 1-мірного

Джерело: складено автором на основі []

На рис.1.5 можна побачити, що проекція дисперсії даних на вісь першої головної компоненти більша за її проекції на вихідні осі D_{X1} і D_{X2} , але менша

за їхню суму. Іншими словами, першою головною компонентою відобразити всю дисперсію даних не вийшло. У такому разі будують другу, третю тощо головні компоненти, поки вони сумарно не відобразять більшу частину дисперсії" [8].

Отже, сенс методу головних компонент полягає у тому, що кожна головна компонента пов'язана з певною часткою загальної дисперсії вихідної бази даних. У даному разі, при знаходженні середньорічного кошика споживача дисперсія, може відображати рівень інформативності даних.

"Основною метою є відбір максимально мінливих компонентів, тому перша головна компонента має максимальну вибірккову дисперсію" [10]. Алгоритм в такий спосіб підбирає ваги, щоб розкид першої головної компоненти був максимально можливим за умови, що сума дорівнює 1. Після формування першої головної компоненти можна сформувати другу, її ваги алгоритм підбирає так, щоб вона була некорельована з першою головною компонентою, і щоб сама компонента була з максимальною дисперсією для себе. У такий самий спосіб підбирають ваги наступних головних компонент.

Перспективи аналізу асоціацій для управління бізнесом.

Описані вище показники використовують для встановлення порога значущості асоціативних правил. Якщо фактичне значення обраного для оцінки показника асоціації виявляється нижчим за встановлений аналітиком поріг, то воно не розглядається як бізнес-правило.

Тобто найскладнішим практичним завданням асоціативного аналізу є встановлення порогових значень. Основний принцип встановлення цих порогів полягає в тому, що вони мають акцентувати увагу на найчастіших наборах. Відповідно до властивості антимонотонності, набір, що складається з піднаборів, які не є частими, також не може бути частим. Ця властивість дає змогу виключити рідкісні піднабори й автоматично - всі укрупнені набори, до яких входить один або кілька рідкісних піднаборів.

Прийняття управлінських рішень на основі аналізу наборів визначатиметься багатьма факторами, здебільшого фінансовими характеристиками. Фінансові показники, зокрема показники прибутку, є основними підсумковими показниками діяльності бізнесу. Отже, асоціативний аналіз потрібно проводити не тільки щодо кількісних (натуральних) ознак (зокрема, кількості угод, як у наведеній вище ситуації), а й щодо сумових (виражених у грошовому вимірнику) показників. Так, у ситуації, що розглядається, доцільність реалізації шкарпеток може оцінюватися на основі розрахунку прибутку від продажів шкарпеток, а також прибутку від продажів товарного набору ("взуття-шкарпетки"). При цьому до списку ознак угоди додається сума угоди і підсумовується не кількість угод кожного виду, а сума за кожною угодою.

За наявності грамотно організованого управлінського обліку витрат можливий розрахунок не тільки виручки (яку легко визначити за накладними, чеками та іншими платіжними документами), а й прибутку за кожною угодою, що дасть змогу реалізувати розрахунки на практиці. Що стосується питань розміщення товарів у торговельному залі, то для їх вирішення також необхідно враховувати безліч чинників, зокрема психологічного (сприйняття товару, схильність до купівлі) і фізичного (можливість розміщення, сумісність товарів) характеру.

Це означає, що асоціативний аналіз є початковим етапом прийняття управлінських рішень про асортимент. Однак він є корисним засобом контролю результатів прийнятих управлінських рішень. В цьому разі діє порівняння значень показників асоціації до і після схвалення асортиментного рішення. Наприклад, якщо після нового викладення товарів (які, як припускають, мають бути пов'язані сильніше, ніж спостерігалось) правило посилюється і прибутки зросли, то можна розглядати ухвалене рішення як правильне.

РОЗДІЛ 2

МОДЕЛЮВАННЯ СПОЖИВЧОГО ВИБОРУ АЛГОРИТМОМ APRIORI В СЕРЕДОВИЩІ RSTUDIO

Як зазначено раніше, аналіз споживчого кошика покладено в основу рекомендаційних систем, які використовуються як в звичайних, так і в інтернет-магазинах. Виявлені асоціативні правила будуть вказувати на поєднання товарів, які часто купуються разом. Знання цих патернів дає змогу створити нові способи оптимізації товарів у мережі продуктових магазинів, рекламних акцій або розкладки товарів у магазині. Наприклад, якщо покупці часто купують на сніданок каву або апельсиновий сік разом із випічкою, тоді, можливо, вдасться збільшити щоденний дохід від реалізації, розмістивши випічку ближче до кави та соків. Однак ці методи можна застосовувати і до багатьох інших типів завдань, від рекомендацій фільмів, сайтів знайомств і до виявлення небезпечних залежностей між ліками.

2.1. Збирання, розвідувальний аналіз та візуалізація даних по транзакціям

В цьому розділі виконано аналіз споживчого кошика на основі даних про транзакції продуктового магазину на основі алгоритму Apriori, який здатен ефективно обробляти потенційно великі набори асоціативних правил.

Крок 1. Збирання даних

Практична реалізація аналізу споживчого кошика проведена на реальних даних про покупки протягом одного місяця в продуктовому магазині «Лавка». Ці дані містять 9835 транзакцій, тобто приблизно 327 транзакцій на день (в середньому це 30 транзакцій на годину протягом 12-годинного робочого дня)

– це може свідчити про те, що цей магазин не дуже великий і не маленький. Використаний датасет було створено на основі набору даних groceries з R-пакета arules [4, 5].

Будь-який типовий продуктовий магазин має широкий асортимент різноманітних товарів. Наприклад, в ньому може бути п'ять марок молока, дюжина видів прального порошку, три найменування кави тощо. Оскільки магазин «Лавка» можна вважати магазином середнього розміру, то припустимо, що в ньому не надто переймаються пошуком правил, які можна застосувати тільки до конкретної марки молока або прального засобу. Тому в датасеті були видалені назви торгових марок продуктів. Це дало змогу скоротити їх кількість до трохи більше ніж 169 видів, якими можна керувати, використовуючи широкі категорії, наприклад, куряче м'ясо, заморожені продукти, маргарин і газована вода.

Якщо потрібно визначити більш конкретні асоціативні правила - наприклад, клієнти вважають за краще купувати з арахісовим маслом виноградний або полуничний мармелад, - тоді знадобиться більш розширена база даних про транзакції. Великі роздрібні мережі, щоб знайти асоціації між конкретними брендами, кольорами або запахами товарів, використовують бази даних з десятками-сотнями мільйонів транзакцій.

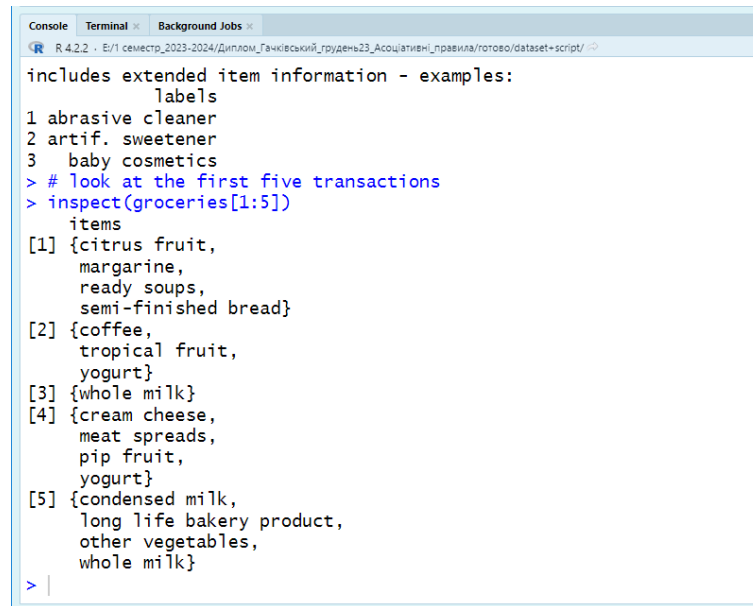
На наступному кроці потрібно дати відповіді на такі питання: чи має керівництво магазину припущення про те, які типи товарів зазвичай купують разом? чи є звичайним поєднання вина та твердого сиру? Хліба, молока та масла? чай і мед?

Крок 2. Дослідження та підготовка даних - розвідувальний аналіз

Дані про транзакції зберігаються в не зовсім звичному, форматі. В більшості прикладів аналізу, дані подаються у вигляді матриці (таблиці, дата фрейму), рядки якої відповідають записам покупки, а стовпці - ознакам. У матричному форматі у всіх прикладах має бути однаковий набір ознак. Однак дані про транзакції продуктів покупок подано у більш вільному форматі. Зазвичай, кожен рядок даних відповідає одному прикладу - в даному разі одній

транзакції. Але замість фіксованої кількості ознак кожен запис містить розділений комами список покупок і складається з довільної кількості (одного або декількох) елементів. Фактично, кожен приклад може мати свою кількість ознак.

Для того щоб продовжити виконання аналізу наявного датасету, завантажуюмо файл `groceries.csv` і зберігаємо його в робочому каталозі R.



```

Console Terminal Background Jobs
R 4.2.2 - E:/1 семестр_2023-2024/Диплом_Гачківський_грудень23_Асоціативні_правила/rotooo/dataset=script/
includes extended item information - examples:
  labels
1 abrasive cleaner
2 artif. sweetener
3 baby cosmetics
> # look at the first five transactions
> inspect(groceries[1:5])
items
[1] {citrus fruit,
    margarine,
    ready soups,
    semi-finished bread}
[2] {coffee,
    tropical fruit,
    yogurt}
[3] {whole milk}
[4] {cream cheese,
    meat spreads,
    pip fruit,
    yogurt}
[5] {condensed milk,
    long life bakery product,
    other vegetables,
    whole milk}
>
  
```

Рис. 2.1 – Скріншот завантажених даних файлу `groceries.csv` в робочому просторі RStudio

Джерело: розраховано автором самостійно

Перші п'ять рядків вхідного файлу `groceries.csv` наведені на рис. 2.1. Як бачимо, перша транзакція містить такі продукти як цитрусові, маргарин, готові супи, хлібні вироби; друга — кава, тропічні фрукти, йогурт; третя — незбиране молоко; четверта — вершковий сир, м'ясні паштети, сухофрукти, йогурт; п'ята — згущене молоко, хлібобулочні вироби тривалого зберігання, свіжі овочі, незбиране молоко. Відповідно маємо, що перша, четверта і п'ята транзакції включають в себе по чотири товари, друга — три, а третя — лише одне найменування.

Припустимо, що дані завантажено за допомогою стандартної функції `read.csv()`. На рис. 2.2 показано, який тоді вигляд матиме матрична форма представлення даних в R.

```

1 read.csv("groceries.csv")
2
3 citrus.fruit      semi.finished.bread      margarine      ready.soups
4 tropical fruit   yogurt
5 whole milk
6 pip fruit        yogurt      cream cheese      meat spreads
7 other vegetables whole milk      condensed milk long life bakery product
8 whole milk      butter      yogurt      rice
9 abrasive cleaner
10 rolls/buns
11 other vegetables      UHT-milk      rolls/buns      bottled beer
12 liquor (appetizer)
13 potted plants
14 whole milk      cereals
15 tropical fruit   other vegetables      white bread      bottled water
16 chocolate
17 citrus fruit     tropical fruit      whole milk      butter
18 curd             yogurt      flour      bottled water
19 dishes
20 beef
21 frankfurter      rolls/buns      soda
22 chicken          tropical fruit
23 butter           sugar      fruit/vegetable juice      newspapers
24 fruit/vegetable juice
25 packaged fruit/vegetables
26 chocolate
27 specialtv bar

```

Рисунок 2.2 – Скриншот завантаження вхідного датасету groceries.csv в матричній формі в робочому просторі RStudio

Джерело: розраховано автором самостійно

Як бачимо на рис. 2.2, R для зберігання даних створив чотири стовпці про транзакції: V1, V2, V3 і V4. Таке зберігання датасету може здатися доцільним, проте в подальшій роботі саме при виявленні асоціативних правил це може викликати проблеми в їх реалізації. R створює чотири змінні, оскільки в першому рядку було рівно чотири значення, які розділені комами між собою.

Попередній аналіз даних показав, що до складу кошика може входити більше ніж чотири продукти; за способом розділення (як на рис. 2.2) такі транзакції будуть розбиті на кілька рядків матриці. Можна було б спробувати виправити це, помістивши на початок файлу з даними транзакцію з найбільшою кількістю елементів, але тоді ігнорується ще одна, більш серйозна проблема. Структуруючи дані в такий спосіб, R створив набір ознак, які зберігають не тільки елементи транзакцій, а й їхню послідовність. Якщо уявити алгоритм навчання як спробу знайти залежності між V1, V2, V3 і V4, то незбиране молоко в ознаці V1 означатиме не те саме, що незбиране молоко в ознаці V2. Отже, потрібен набір даних, у якому транзакції були б представлені не як множина позицій, що заповнюються (або не заповнюються) конкретними елементами, а скоріше як споживчий кошик, який містить або не містить кожен конкретний товар.

Підготовка даних: створення розрідженої матриці даних про транзакції. Розв'язання цієї проблеми полягає у використанні структури

даних, яка називається розрідженою матрицею. Кожен рядок такої матриці відповідає одній транзакції. Однак у розрідженій матриці кожному товару, який може зрештою опинитися в кошику покупця, відповідає окремий стовпець (ознака). Оскільки в даних продуктового магазину присутні 169 різних товарів, розріджена матриця міститиме 169 стовпців.

Виникає питання: чому б просто не зберегти такі дані як фрейм даних? Причина полягає в тому, що під час додавання нових транзакцій і товарів звичайна структура датасету швидко стає занадто великою і перестає вміщуватися в пам'яті комп'ютера. Навіть для відносно невеликого набору даних про транзакції матриця буде містити майже 1,7 мільйона клітинок, у більшості з яких буде записані нулі (звідси і слово "розріджена" в назві матриці - у ній дуже мало ненульових значень). Оскільки немає сенсу зберігати всі ці нулі, розріджена матриця практично ніколи не зберігається в пам'яті повністю; зберігаються тільки ті комірки, в яких щось записано. Це дає змогу розміщувати структуру в пам'яті ефективніше, ніж матрицю еквівалентного розміру або фрейм даних.

Для того щоб створити розріджену матричну структуру на основі даних про транзакції, можна скористатися функціоналом пакета `arules` (назва від *association rules* – «асоціативні правила») [10].

Завантажуючи дані про транзакції, не можна, як раніше, просто використовувати функцію `read.csv()`. У пакеті `arules` є функція `read.transactions()`. Вона схожа на `read.csv()`, але її результатом є розріджена матриця, що підходить для даних про транзакції. Параметр `sep=","`, "вказує на те, що елементи вхідного файлу розділяються комою. Щоб прочитати дані з `groceries.csv` у розріджену матрицю з ім'ям `foods`, потрібно виконати такий скрипт:

```
# скрипт 2.1
```

```
> foods <- read.transactions("groceries.csv", sep = ",")
```

```
> summary(foods)
```

```

> foods <- read.transactions("groceries.csv", sep = ",")
> summary(foods)
transactions as itemMatrix in sparse format with
9835 rows (elements/itemsets/transactions) and
169 columns (items) and a density of 0.02609146

most frequent items:
  whole milk other vegetables    rolls/buns      soda      yogurt
      2513      1903      1809      1715      1372
  (other)
   34055

element (itemset/transaction) length distribution:
sizes
 1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46   29   14
 19   20   21   22   23   24   26   27   28   29   32
 14    9   11    4    6    1    1    1    1    3    1

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.000  2.000   3.000   4.409  6.000  32.000

includes extended item information - examples:
  labels
1 abrasive cleaner
2 artif. sweetener
3  baby cosmetics
>

```

Рисунок 2.3 – Лістинг результатів скрипта 2.1

Джерело: розраховано автором самостійно

Для того, щоб переглянути основну інформацію про створену матрицю продуктів, можна скористатися функцією `summary()` для об'єкта.

Перший блок описової інформації у результатах виведення цієї функції (рис. 2.3) представляє зведену статистику по розрідженій матриці, яка складається з 9835 rows, що означає кількість транзакцій (записів датасету), а 169 columns - 169 видів товарів (стовпців), які можуть бути в кошику покупця. Якщо товар був придбаний у відповідній транзакції, то комірка матриці дорівнює 1, в іншому випадку — 0. Значення щільності, що дорівнює 0.02609 (2,609 %), означає частку ненульових комірок матриці.

Оскільки в матриці всього $9835 \times 169 = 1662115$ позицій, то за 30 днів роботи магазину було придбано загалом $1\,662\,115 \times 0.02609 = 43367$ товарів (не враховуючи, що деякі товари купували декілька разів). Виконавши ділення, можна визначити, що середня транзакція містила $43367/9835=4,409$ різних товарів, тобто приблизно чотири елементи. Зрозуміло, якщо уважніше проаналізувати результат, то стане зрозуміло, що середня кількість елементів у транзакції там уже є.

У наступному блоці результатів `summary()` (рис. 2.3) перераховуються елементи (продукти), які найчастіше зустрічаються в даних про транзакції. До списку товарів, які часто купують, входять незбиране молоко (whole milk),

свіжі овочі (other vegetables), рогаики/булочки (rolls/buns), газована вода (soda), йогурт (yogurt). Наприклад, незбиране молоко зустрічається в 2513 транзакціях, тоді легко розрахувати, що цей продукт зустрічається у 25,6% транзакцій ($2513 / 9835 = 0,2555$).

Третій блок функції `summary()` (рис. 2.3) містить статистичні дані про розмір та центральні статистичні характеристики транзакцій. Тобто, 2159 транзакцій містили лише один елемент, тоді як найбільша транзакція містила 32 найменування. Перший кuartиль і середній розмір покупки становили відповідно по два та три товари. Це означає, що 25 % транзакцій містили два або один товар, а приблизно половина транзакцій - три товари та менше. Середнє значення (Mean) означає що 4,409 елементів на транзакцію відповідає значенню, яке розраховано вручну.

У пакеті `arules` також є кілька корисних функцій для дослідження даних про транзакції. Щоб переглянути вміст розрідженої матриці, можна використовувати функцію `inspect()` у поєднанні з векторними операторами R:

скрипт 2.2 – перегляд перших п'яти транзакцій

```
> inspect(foods [1:5])
```

```
> inspect(foods [1:5])
items
[1] {citrus fruit, margarine, ready soups, semi-finished bread}
[2] {coffee, tropical fruit, yogurt}
[3] {whole milk}
[4] {cream cheese, meat spreads, pip fruit, yogurt}
[5] {condensed milk, long life bakery product, other vegetables, whole milk}
>
```

Рисунок 2.4 – Лістинг результатів скрипта 2.2

Джерело: розраховано автором самостійно

Ці транзакції відповідають представленню вхідного CSV-файлу. Щоб дослідити конкретний елемент (тобто стовпчик даних), можна використати матричний запис `[row,col]`. Використання такої команди в поєднанні з функцією `itemFrequency()` дасть змогу побачити частку транзакцій, у яких міститься вказаний елемент. Наприклад, для того, щоб дізнатися таке значення для перших чотирьох елементів у даних про продукти, можна скористатися такою командою:

скрипт 2.3

```
> itemFrequency(groceries[, 1:3])
```

```
> itemFrequency(foods[, 1:4])
abrasive cleaner artif. sweetener baby cosmetics baby food
0.0035587189 0.0032536858 0.0006100661 0.0001016777
> |
```

Рисунок 2.5 – Лістинг результатів скрипта 2.3

Джерело: розраховано автором самостійно

Як видно з рис. 2.5, елементи розрідженої матриці сортуються за стовпчиками в алфавітному порядку. І в результаті маємо, що абразивні засоби для чищення (abrasivecleaner) та штучні підсилювачі солодоців (artif.sweetener) трапляються приблизно в 0,3% транзакцій кожна, а дитяча косметика (babycosmetics) — приблизно лише в 0,06 %, а дитячі продукти – в 0,01% покупках.

Крок 3. Візуалізація даних по транзакціям

Щоб візуально представити *частотні діаграми елементів* (тобто Візуалізація підтримки елементів), скористаємося функцією `itemFrequencyPlot()`. Ця функція будує стовпчасту діаграму, яка відображає частку транзакцій із зазначеними елементами. Оскільки дані про транзакції містять дуже велику кількість елементів, то доводиться обмежувати кількість виведення елементів на графік, щоб отримати не громіздку діаграму. Для того, щоб відобразити на діаграмі елементи, які присутні в мінімальній кількості транзакцій, можна застосувати параметр `support` в функції `itemFrequencyPlot()`:

скрипт 2.4

```
> itemFrequencyPlot(foods, support = 0.1)
```

На рис. 2.6 наведено гістограму, на якій показані вісім елементів, для яких значення підтримки становить щонайменше 10%.

Якщо потрібно обмежити діаграму певною кількістю продуктів, то можна використати для функції `itemFrequencyPlot()` параметр `topN`:

скрипт 2.5

```
> itemFrequencyPlot(foods, topN = 15)
```

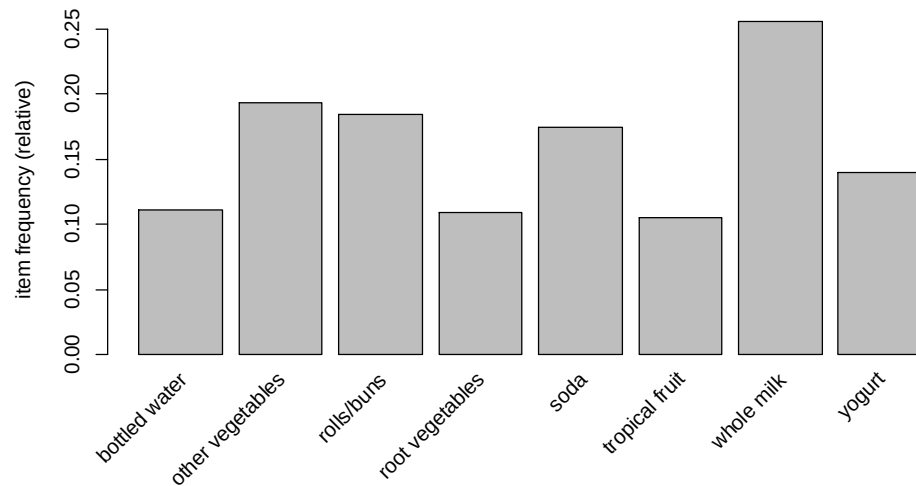


Рисунок 2.6 – Гістограма відбору продуктів (транзакцій) із значенням підтримки яких становить щонайменше 10 %

Джерело: розраховано автором самостійно

Тоді продукти, які будуть представлені на гістограмі, впорядковуються за зменшенням підтримки. Такий результат наведено на рис. 2.7 для 15 найкращих товарів із датасету про продукти.

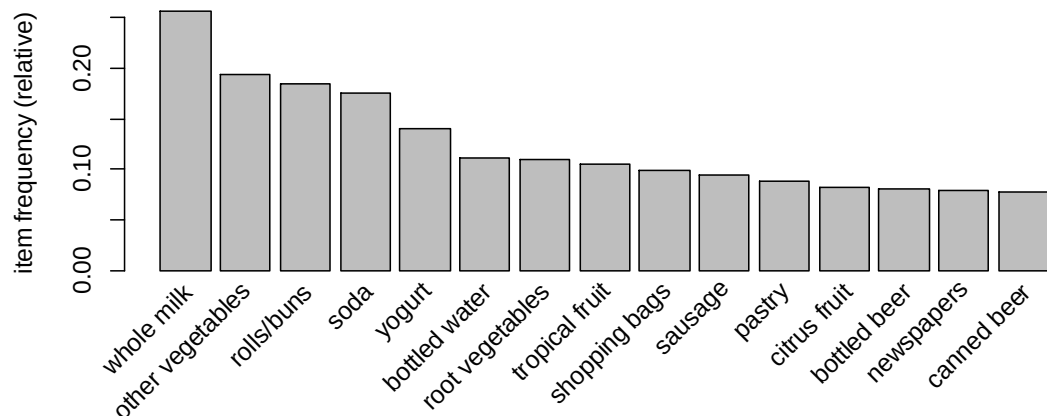


Рисунок 2.7 – Рівні підтримки для відсортованих 15 найкращих продуктів

Джерело: розраховано автором самостійно

Візуалізація даних про транзакції: *побудова розрідженої матриці*. Крім візуалізації окремих елементів, також можна розглянути розріджену матрицю загалом, використовуючи функцію `image()`. Звісно, оскільки сама матриця надто велика, краще візуалізувати лише якусь підмножину матриці. Команда для відображення розрідженої матриці для перших п'яти транзакцій має такий вигляд:

скрипт 2.6

> image(foods[1:5])

> image(foods[1:100])

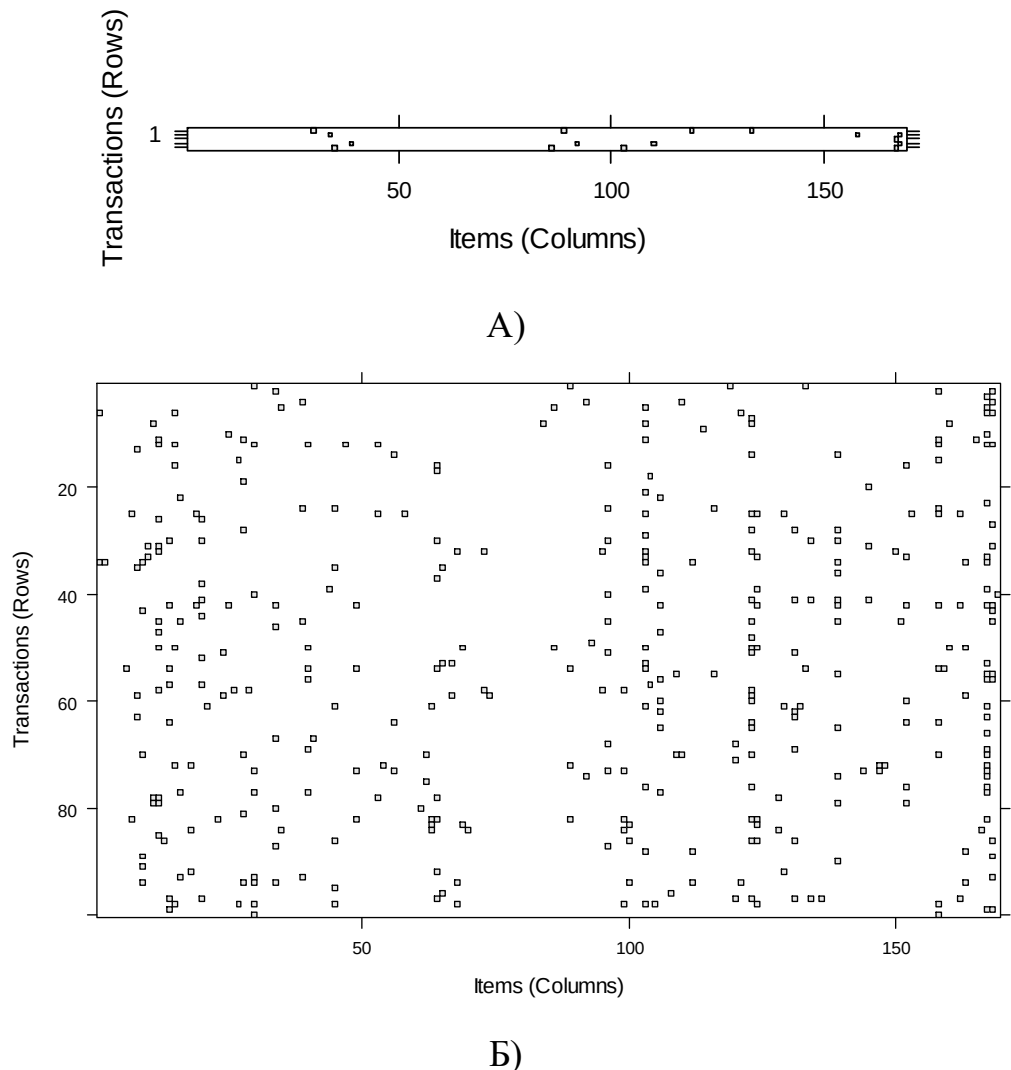


Рисунок 2.8 — Діаграма розрідженої матриці для перших п'яти (А) та ста (Б) транзакцій

Джерело: розраховано автором самостійно

Діаграма (рис.2.8) представляє собою матрицю з п'яти (або ста) рядків і 169 стовпців, що відповідають п'яти (ста) запитаним транзакціям і 169 можливим товарам. Ті комірки матриці, де у відповідній транзакції (рядку), присутній даний елемент (стовпчик), заповнюються сірим кольором. Незважаючи на те, що діаграма надто мала і її, можливо, не дуже зручно читати, можна помітити, що перша, четверта і п'ята транзакції містили по чотири елементи, оскільки в їхніх рядках заповнені чотири комірки (рис.

2.8.A). Як видно праворуч на діаграмі, рядки 3 і 5, а також рядки 2 і 4 мають спільний елемент.

Така візуалізація може бути корисним інструментом для дослідження даних про транзакції. З одного боку, вона може допомогти виявити потенційні проблеми з даними. Повністю заповнені стовпці можуть вказувати на товари, що купуються в кожній транзакції, - проблема, що може виникнути, наприклад, у разі ненавмисного включення до набору даних про транзакції імені або ідентифікаційного номера продавця.

Крім того, патерни, виявлені на діаграмі, допомагають виявляти цікаві сегменти транзакцій і елементів, особливо якщо дані відсортовані відповідними способами. Наприклад, якщо впорядкувати транзакції за датою, то патерни, представлені у вигляді чорних крапок, можуть вказати на сезонні зміни кількості або типів товарів, що купуються. Можливо, на Різдво або Новий Рік частіше купують іграшки, а ближче до Хелловіну попитом користуються цукерки. Такий тип візуалізації може бути особливо корисним, якщо відсортувати елементи ще й за категоріями. Однак у більшості ситуацій графік буде доволі випадковим, як статичні перешкоди на телевізійному екрані.

Потрібно розуміти, що для надвеликих баз даних транзакцій така візуалізація не принесе особливої користі, тому що клітинки будуть занадто маленькими і їх неможливо буде розрізнити. Корисним може бути поєднання функцій `image()` та `sample()`, що дозволить переглядати розріджену матрицю для випадкового набору транзакцій (рис. 2.9). Команда для візуалізації діаграми випадкового набору зі 150 транзакцій має такий вигляд:

скрипт 2.7

```
> image(sample(foods, 150))
```

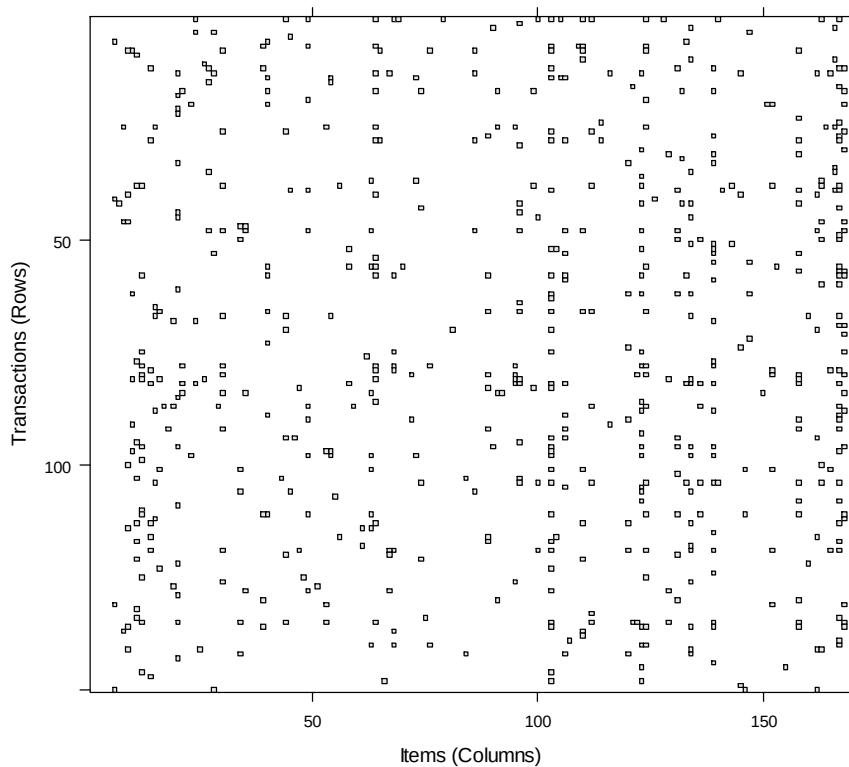


Рисунок 2.9 - Візуалізація розрідженої матриці для 150 випадкових транзакцій

Джерело: розраховано автором самостійно

У результаті такого скрипта буде побудовано матричну діаграму зі 150 рядками та 169 стовпчиками (рис. 2.9). Кілька стовпців здаються досить щільно заповненими, і це свідчить про наявність у магазині кількох дуже популярних товарів. Проте загалом розподіл точок є досить випадковим.

2.2. Виділення асоціативних правил алгоритмом Apriori

Після підготовки та аналізу даних, можна шукати залежності товарів в споживчому кошику. Для цього скористаємося функцією `apriori()`, яка реалізує алгоритм Apriori.

Крок 3. Навчання моделі алгоритмом Apriori на даних. В таблиці 2.1 наведено загальний синтаксис для створення наборів правил за допомогою функції `apriori()`.

Таблиця 2.1 Синтаксис побудови асоціативних правил за допомогою функції `apriori()`

<i>Пошук асоціативних правил</i>
<pre>myrules <- apriori(data = mydata, parameter = list(support = 0.1, confidence = 0.8, minlen = 1))</pre> data — розріджена матриця даних про транзакції (покупки); support — необхідна мінімальна підтримка правил; confidence — необхідна мінімальна довіра правила; minlen — мінімальна кількість елементів в транзакції. Функція створює набір правил, які відповідають вказаним мінімальним критеріям.
<i>Перевірка асоціативних правил</i>
<pre>inspect(myrules)</pre> myrules – множина асоціативних правил, отриманих функцією apriori() Функція виводить асоціативні правила на екран. Для вибору перегляду одного або декількох правил можна застосувати векторні оператори до об'єкта myrules.
Приклад: <pre>foods_ruls <- apriori(foods, parameter = list(support = 0.01, confidence = 0.25, minlen = 2)) inspect(foods_ruls[1:7])</pre>

Джерело: складено автором на матеріалах [4]

Функція `apriori()` виконується просто, але іноді лише методом проб та помилок можна знайти значення підтримки та довіри, які дадуть змогу отримати розумну кількість асоціативних правил. З одного боку, якщо встановити ці рівні надто високими, то можна не знайти правил або знайти такі правила, які виявляться надто загальними, щоб бути корисними. З іншого - занадто низький поріг може призвести до громіздкого набору правил. Гірше того, операція може зайняти дуже багато часу або вичерпати пам'ять компютера на етапі навчання моделі.

Якщо застосувати для даного датасету про продукти, запропоновані за замовчуванням значення `support=0.1` і `confidence=0.8`, то отримаємо набір 0 кількості правил (рис. 2.10):

```
# скрипт 2.8  
> apriori(foods)
```

```

> apriori(foods)
Apriori

Parameter specification:
  confidence minval smax arem aval originalsupport maxtime support minlen maxlen target ext
      0.8      0.1    1 none FALSE              TRUE     5   0.1     1    10 rules TRUE

Algorithmic control:
  filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE     2     TRUE

Absolute minimum support count: 983

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [8 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [0 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
set of 0 rules

> |

```

Рисунок 2.10 – Лістинг результатів скрипта 2.8

Джерело: розраховано автором самостійно

Якщо проаналізувати (0 правил), то такий результат не викликає подиву. Оскільки за замовчуванням $\text{support} = 0.1$, це означає, що для генерації правила, елемент має з'являтися принаймні в $0,1 \times 9,385 = 938,5$ транзакціях. Але в датасеті `foods` так часто зустрічаються лише 8 елементів, тому не дивно, що за таких критеріїв не знайдено відповідних правил. Зрозуміло, що потрібно розширити умови пошуку правил, за рахунок зміни значень підтримки та довіри.

Один зі способів розв'язати проблему вибору *мінімального значення підтримки* полягає в тому, щоб зрозуміти, при якій найменшій кількості транзакцій, патерн становитиме інтерес. Наприклад, можна вирішити, що якщо якийсь товар купують двічі на день (тобто приблизно 60 разів на місяць), то це може бути важливо. Отже, можна обчислити значення підтримки, яке необхідне для пошуку правил, що відповідають щонайменше такій кількості транзакцій. Оскільки 60 із 9835 транзакцій - це 0,006, то ми спочатку спробуємо встановити рівень підтримки саме таким.

Вибір *мінімального значення довіри* передбачає пошук тонкого балансу. З одного боку, якщо рівень довіри занадто низький, то алгоритм згенерує велику кількість недостовірних правил - наприклад, десятками правил, що вказують на товари, які зазвичай купують разом з акумуляторами. Як дізнатися, на що спрямувати рекламний бюджет? З іншого боку, якщо

встановити надто високий рівень довіри, то ми будемо обмежені правилами, які очевидні та неминучі, - наприклад, таким правилом: в поєднанні з акумуляторами завжди купується детектор диму. В такому разі, розміщення детекторів диму ближче до акумуляторів навряд чи принесе додатковий дохід, оскільки ці два предмети і так майже завжди купують разом. Відповідний мінімальний рівень довіри багато в чому залежить від мети дослідження. Якщо почати з консервативного значення, то його завжди можна зменшити, щоб розширити пошук, якщо досі не було знайдено нічого краще придатного.

Нехай початковий поріг довіри дорівнює 0,25. Це означає, що для того, аби правило потрапило в результати, воно повинно бути коректним принаймні у 25% випадків. В такий спосіб можна виключити найбільш ненадійні правила, одночасно залишаючи деякий простір для маневру, щоб пізніше змінити поведінку покупців за допомогою цільових рекламних акцій.

Тепер згенеруємо першу сукупність асоціативних правил. Крім мінімальних значень підтримки і довіри, для параметру `minlen` встановимо значення 2. Це дозволить виключити правила, які містять менше двох елементів. Так можна попередити створення неінформативних правил, які існують лише тому, що якийсь продукт часто купують, наприклад, `{ }=>wholemilk`. Це правило виконує вимоги мінімальної підтримки та довіри, тому що незбиране молоко купується більш ніж у 25% транзакцій, але це не дуже корисна інформація.

Повна команда для пошуку набору асоціативних правил з використанням алгоритму `Apriori` має такий вигляд:

```
# скрипт 2.9
```

```
> foods_rules <- apriori(foods, parameter =  
    list(support = 0.006, confidence = 0.25, minlen = 2))  
> foods_rules
```

Після такого скрипта, створені правила будуть збережені в об'єкті правил під назвою, наприклад, `foods_rules`. Для їх перегляду достатньо ввести його ім'я.

```

> foods_rules <- apriori(foods, parameter =
+   list(support = 0.006, confidence = 0.25, minlen = 2))
Apriori

Parameter specification:
confidence minval smax arem aval originalsupport maxtime support minlen maxlen target ext
0.25 0.1 1 none FALSE TRUE 5 0.006 2 10 rules TRUE

Algorithmic control:
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 59

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.02s].
sorting and recoding items ... [109 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 done [0.01s].
writing ... [463 rule(s)] done [0.00s].
creating S4 object ... done [0.01s].
> foods_rules
set of 463 rules
>

```

Рисунок 2.11 – Лістинг результатів скрипта 2.9

Джерело: розраховано автором самостійно

Як видно з рис. 2.11 об'єкт `foods_rules` виявив 463 асоціативних правил.

скрипт 2.10

```

> plot(foods_rules, measure = c("support", "lift"),
      shading = "confidence", jitter=0)

```

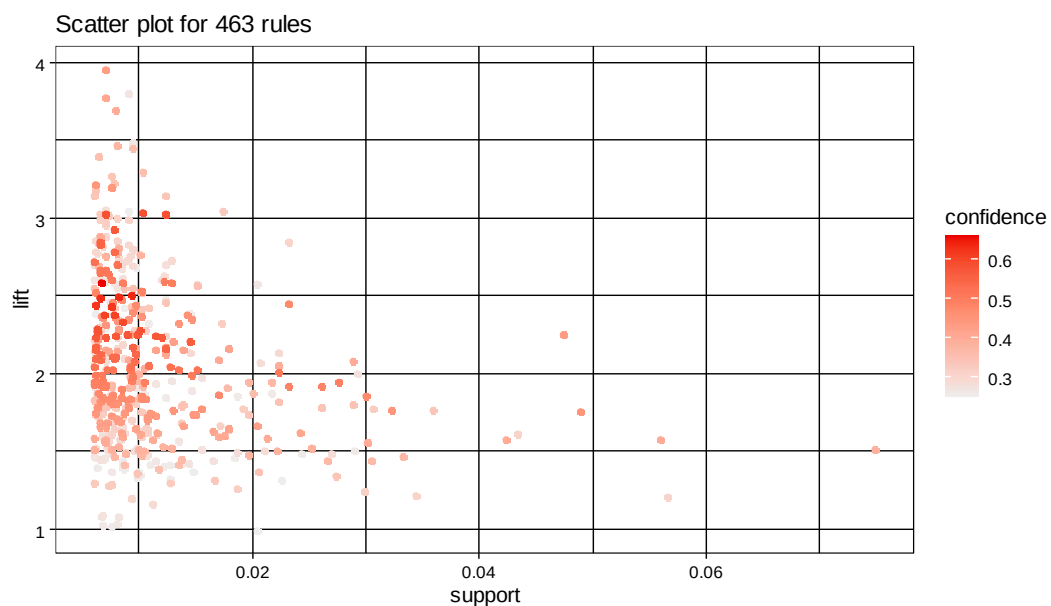


Рисунок 2.12 – Діаграма розсіювання добутих 463 асоціативних правил з рівнем підтримки 0,6%, довіри 25% та кількості елементів більше 2

Джерело: розраховано автором самостійно

На рис. 2.12 наведено діаграму розсіювання добутих 463 асоціативних правил з підйомом по осі Y. На ньому легко побачити правила з високим рівнем довіри.

скрипт 2.11

```
> plot(foods_rules, method = "two-key plot", jitter = 0)
```

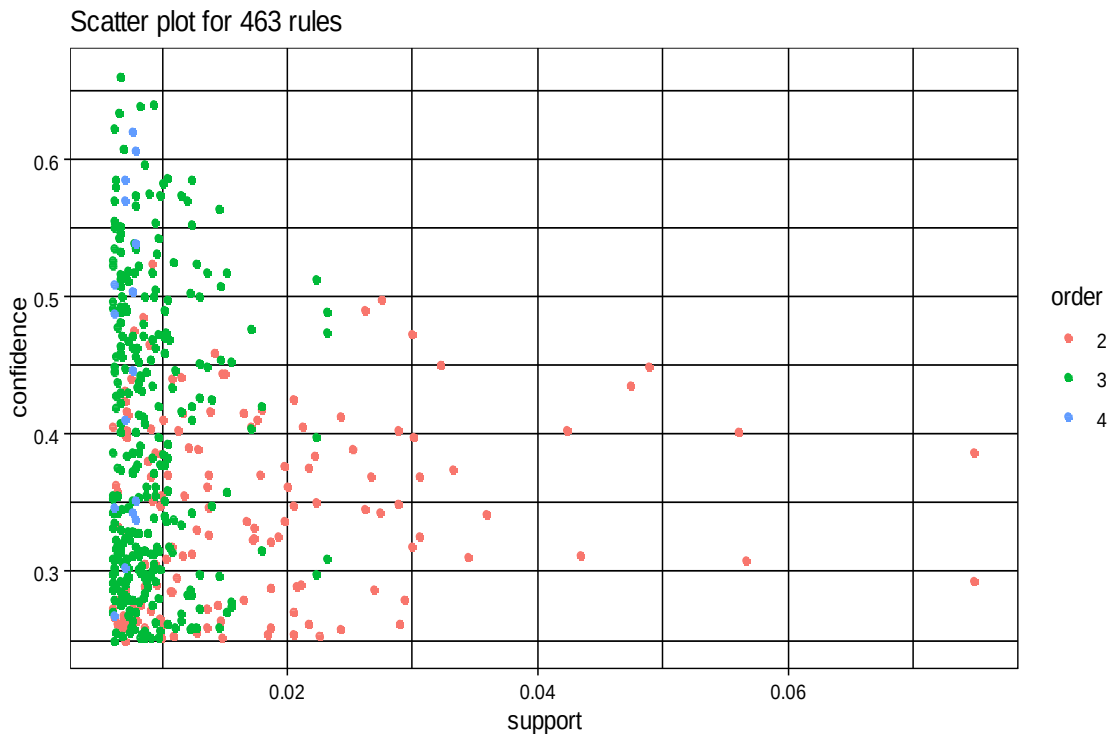


Рисунок 2.13 – Діаграма розсіювання добутих 463 асоціативних правил побудована методом «двох ключів» з рівнем підтримки 0,6%, довіри 25% та кількості елементів більше 2

Джерело: розраховано автором самостійно

На рис. 2.13 наведено діаграму розсіювання добутих 463 асоціативних правил побудована методом «двох ключів». З графіка видно, що міри порядку та підтримки мають дуже сильний обернений зв'язок, що є відомим фактом для асоціативних правил.

Для того, щоб визначити, чи усі добути правила є інформативними та корисними, потрібно провести ретельне їх дослідження. Тому переходимо до наступного етапу.

2.3. Оцінювання та підвищення ефективності виділених асоціативних правил

Крок 4. Оцінювання ефективності моделі. Щоб отримати загальні статистичні описові характеристики про добути асоціативні правила, можна скористатися функцією `summary()`. Розподіл правил за довжиною говорить про те, скільки правил складається з відповідної кількості елементів. У даному наборі 150 правил складаються з двох елементів, 297 - з трьох, а 16 - з чотирьох. Також у виведенні функції представлено зведену статистику цього розподілу:

скрипт 2.12

> `summary(foods_rules)`

```
> summary(foods_rules)
set of 463 rules

rule length distribution (lhs + rhs): sizes
  2   3   4
150 297  16

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 2.000  2.000   3.000   2.711  3.000   4.000

summary of quality measures:
      support      confidence      coverage      lift      count
Min. :0.006101 Min. :0.2500   Min. :0.009964 Min. :0.9932 Min. : 60.0
1st Qu.:0.007117 1st Qu.:0.2971   1st Qu.:0.018709 1st Qu.:1.6229 1st Qu.: 70.0
Median :0.008744 Median :0.3554   Median :0.024809 Median :1.9332 Median : 86.0
Mean   :0.011539 Mean   :0.3786   Mean :0.032608 Mean :2.0351 Mean :113.5
3rd Qu.:0.012303 3rd Qu.:0.4495   3rd Qu.:0.035892 3rd Qu.:2.3565 3rd Qu.:121.0
Max.   :0.074835 Max.   :0.6600   Max. :0.255516 Max. :3.9565 Max. :736.0

mining info:
 data ntransactions support confidence
foods      9835      0.006      0.25

                                call
apriori(data = foods, parameter = list(support = 0.006, confidence = 0.25, minlen = 2))
> |
```

Рисунок 2.14 — Лістинг результатів скрипта 2.12

Джерело: розраховано автором самостійно

Як видно з рис. 2.14, в наведеному висновку функції `summary()`, розмір правила дорівнює сумі його лівої (lhs) і правої частин (rhs). Наприклад, розмір правила {хліб} => {масло} дорівнює 2 елементам, а розмір правила {арахісове масло, мармелад} => {хліб} - 3.

Далі ми бачимо зведену статистику показників якості правила: значення підтримки (support), достовірності/довіри (confidence) і ліфта (lift). Параметри

support і confidence не повинні викликати заперечення, оскільки їх використано як критерії відбору правил. Якби значення підтримки і довіри для більшості або всіх правил виявилися дуже близькими до мінімальних порогів, це, ймовірно, викликало б занепокоєння, оскільки могло б означати, що, можливо, встановлено занадто високу планку. Однак у цьому разі це не так, оскільки багато правил мають набагато вищі значення кожного з цих показників (рис. 2.14).

Третій стовпчик - це показник, який ще не редагувано, називається ліфтом правила. Він показує, наскільки ймовірність придбання одного або кількох товарів вища, якщо відомо, що було придбано інший товар або кілька товарів. Ліфт обчислюється за такою формулою:

$$Lift_{X \Rightarrow Y} = \frac{Conf_{X \cup Y}}{Conf_Y}. \quad (2.1)$$

На відміну від довіри, коли має значення послідовність елементів, $lift(X \rightarrow Y)$ - це те саме, що $lift(Y \rightarrow X)$.

Наприклад, припустимо, що в продуктовому магазині більшість покупців купують молоко та хліб. Зрозуміло, що має бути багато транзакцій, в яких присутнє молоко або хліб. Однак якщо $lift(\text{молоко} \rightarrow \text{хліб})$ більше одиниці, то це свідчить, що ці два товари разом зустрічаються частіше, ніж окремо. *Тому велике значення ліфта є явною ознакою важливості правила і відображає справжню залежність між елементами.*

В останньому фрагменті (mining info) виведення результатів функції `summary()` (рис. 2.14) міститься інформація про аналіз даних, де показано, яким чином були обрані правила. Тут видно, що дані про продукти містили 9835 транзакцій і використовувалися для побудови правил з мінімальною підтримкою 0,006 і мінімальною довірою 0,25.

Для того, щоб побачити конкретні правила, можна скористатися функцією `inspect()`. Наприклад, щоб переглянути перші сім правил в об'єкті `groceryrules`, можна використати такий скрипт:

скрипт 2.13

> inspect(foods_rules[1:7])

```
> inspect(foods_rules[1:7])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{potted plants}	=> {whole milk}	0.006914082	0.4000000	0.01728521	1.565460	68
[2]	{pasta}	=> {whole milk}	0.006100661	0.4054054	0.01504830	1.586614	60
[3]	{herbs}	=> {root vegetables}	0.007015760	0.4312500	0.01626843	3.956477	69
[4]	{herbs}	=> {other vegetables}	0.007727504	0.4750000	0.01626843	2.454874	76
[5]	{herbs}	=> {whole milk}	0.007727504	0.4750000	0.01626843	1.858983	76
[6]	{processed cheese}	=> {whole milk}	0.007015760	0.4233129	0.01657346	1.656698	69
[7]	{semi-finished bread}	=> {whole milk}	0.007117438	0.4022989	0.01769192	1.574457	70

Рисунок 2.15 — Лістинг результатів скрипта 2.13

Джерело: розраховано автором самостійно

На рис. 2.15 наведено Лістинг результатів функції `inspect()`. Перше правило в перекладі на звичайну мову говорить про таке: «Якщо покупець купує рослини в вазонах, то він також купує незбиране молоко». Зі значень підтримки близько 0,007 і довіри 0,40 випливає, що це правило охоплює близько 0,7 % транзакцій і є вірним для 40 % покупок, до складу яких входять такі рослини. Значення ліфта говорить про те, яка ймовірність того, що покупець купить незбиране молоко, якщо він придбав квітку в вазоні, порівняно зі звичайним покупцем. Оскільки відомо, що приблизно 25,6% покупців купили незбиране молоко (support) і 40 % покупців, які купують квітку, купили незбиране молоко (confidence), то ліфт буде дорівнювати $0,40/0,256 = 1,56$ (рис. 2.15), що відповідає значенню, отриманого функцією `summary()` на рис. 2.14.

На рис. 2.15, в стовпчику `support` вказується підтримка для всього правила, а не тільки для `lhs` або `rhs`. Незважаючи на високі значення достовірності та ліфта, чи можна вважати корисним правило {горшкова рослина} → {цільне молоко}? Швидше за все, ні, оскільки не існує логічної причини, за якої хтось буде з більшою ймовірністю купувати молоко, якщо він купив рослину. Тим не менш, дані свідчать про протилежне. Як пояснити цей факт?

Загальний підхід полягає в тому, щоб розділити асоціативні правила на такі категорії:

- корисні;

- тривіальні;
- нез'ясовні.

Очевидно, що мета аналізу споживчого кошика - виявити *корисні правила*, які дають чітку та зрозумілу інформацію. Не всі правила одночасно і зрозумілі, і корисні; поєднання обох цих чинників трапляється набагато рідше, ніж кожне з них окремо.

Так звані *тривіальні правила* - це будь-які правила, які настільки очевидні, що про них не варто згадувати: вони зрозумілі, але марні. Припустимо, що ви є консультантом з маркетингу. Вам платять великі гроші за пошук нових можливостей для спільного просування товарів. Якщо ви повідомите про те, що {дитячі підгузки} → {молочна суміш}, то вас навряд чи запросять на консультацію ще раз.

Тривіальні правила можуть бути замасковані під цікаві/інформативні результати. Припустимо, що знайдено взаємозв'язок між певною маркою дитячих пластівців і деякою книгою. Такий висновок не буде інформативним, якщо головний герой твору намальований на коробці з пластівцями. Правила є нез'ясовними, якщо взаємозв'язок між елементами настільки незрозумілий, що збагнути, як використовувати цю інформацію, неможливо або майже неможливо. Також правило може бути просто випадковим патерном у даних. Наприклад, правило, яке стверджує, що {солоні огірки} → {шоколадне морозиво}, може бути пов'язане з одним єдиним клієнтом, чия вагітна дружина на той момент регулярно відчувала тягу до дивних поєднань харчових продуктів.

Найкращі (корисні) правила - це приховані скарби: невідомі досі ідеї, які стають очевидними лише після того, як їх було відкрито. За наявності достатньої кількості часу можна було б дослідити кожне правило, щоб знайти ці коштовності. Однак дослідники даних, які виконують аналіз, не завжди є найкращими суддями відносно того, чи є правило корисним, тривіальним або нез'ясовним. Тому найкращі правила можуть з'являтися завдяки співпраці з експертами в конкретній предметній галузі - у даному разі в галузі

менеджменту роздрібної торгівлі, - які здатні допомогти в інтерпретації результатів.

Далі використаємо методи сортування та експорту вивчених правил в такий спосіб, щоб найцікавіші результати опинилися першими в списку.

Крок 5. Підвищення ефективності моделі. Експерти предметної галузі дуже швидко визначають корисні правила, але було б неефективним витрачанням часу просити їх оцінити сотні або тисячі правил. Тому доцільним є відсортувати правила за різними критеріями та витягти їх із бази загальних правил, і далі передати у відділ маркетингу для ретельного вивчення. Так можна підвищити ефективність виявлення правил, зробивши результати більш корисними.

Сортування набору асоціативних правил. Залежно від мети аналізу споживчого кошика корисними можуть виявитися правила з найбільшою підтримкою або довірою, або ліфтом. При реалізації в RStudio можна використати функцію `sort()`, яка входить до складу пакета `arules`. Вона застосовується для зміни послідовності у списку правил (тобто для сортування правил), даючи змогу ставити на перше місце правила з найвищими або найнижчими якісними показниками.

Щоб змінити послідовність правил в об'єкті `groceryrules`, потрібно в функції `sort()` надати параметру `by` значення `"support"`, або `"confidence"` або `"lift"`. Поєднуючи сортування з векторними операторами, можна отримати кілька цікавих правил. Наприклад, за допомогою такої команди можна отримати сім найкращих правил з найбільшим значенням статистики `lift`:

скрипт 2.14

```
> inspect(sort(foods_rules, by = "lift")[1:7])
```

Лістинг результатів цього скрипта наведено на рисунку 2.16.

```

> inspect(sort(foods_rules, by = "lift")[1:7])
  lhs                                     rhs      support  confidence coverage  lift  count
[1] {herbs}                             => {root vegetables} 0.007015760 0.4312500 0.01626843 3.956477 69
[2] {berries}                           => {whipped/sour cream} 0.009049314 0.2721713 0.03324860 3.796886 89
[3] {other vegetables, tropical fruit, whole milk} => {root vegetables} 0.007015760 0.4107143 0.01708185 3.768074 69
[4] {beef, other vegetables}            => {root vegetables} 0.007930859 0.4020619 0.01972547 3.688692 78
[5] {other vegetables, tropical fruit}    => {pip fruit}       0.009456024 0.2634561 0.03589222 3.482649 93
[6] {beef, whole milk}                  => {root vegetables} 0.008032537 0.3779904 0.02125064 3.467851 79
[7] {other vegetables, pip fruit}        => {tropical fruit}  0.009456024 0.3618677 0.02613116 3.448613 93
>

```

Рисунок 2.16 — Лістинг результатів скрипта 2.14

Джерело: розраховано автором самостійно

Ці правила є більш інформативними та припадними, ніж ті, які нам зустрічалися раніше (наприклад, рис. 2.15).

Розглянемо детально отримані правила на рис. 2.16. Перше правило, у якого значення lift дорівнює приблизно 3,956, говорить про те, що люди, які купують зелень (herbs), майже в чотири рази частіше купують коренеплоди (root vegetables), ніж інші відвідувачі магазину. Друге правило теж є цікавим. Збиті вершки (whipped/sour cream) більш ніж утричі (3,796) частіше зустрічаються у кошику разом із ягодами, ніж в інших наборах. Можливо, це якесь поєднання для десерту?

За замовчуванням правила відсортовані в порядку убутання, тобто найбільші значення йдуть першими. Щоб змінити цю послідовність, можна додати у функцію inspect() додатковий параметр decreasing = FALSE.

Виділення підмножин асоціативних правил. Припустімо, що у відділі маркетингу врахували попереднє правило і створили нову рекламу для просування ягід (наприклад, в сезон ягід). Однак перш ніж запускати кампанію, маркетологи попросили з'ясувати, як часто ягоди купують з іншими товарами. Щоб відповісти на це запитання, потрібно знайти всі правила, які містять у собі ягоди в тій чи іншій формі.

Функція subset() дозволяє відібрати підмножини транзакцій, елементів або правил за певною умовою. Щоб використовувати цю функцію для пошуку правил, у яких зустрічаються ягоди, потрібно скористатися наведеним нижче скриптом. Тоді правила будуть збережені в новому об'єкті з ім'ям berryrules:

скрипт 2.15

```
> berryrules <- subset(foods_rules, items %in% "berries")
```

```
> berryrules
> inspect(foods_rules)
```

У результаті отримаємо набір правил, який наведено на рис.2.17

```
> berryrules <- subset(foods_rules, items %in% "berries")
> berryrules
set of 4 rules
> inspect(berryrules)
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{berries}	=> {whipped/sour cream}	0.009049314	0.2721713	0.0332486	3.796886	89
[2]	{berries}	=> {yogurt}	0.010574479	0.3180428	0.0332486	2.279848	104
[3]	{berries}	=> {other vegetables}	0.010269446	0.3088685	0.0332486	1.596280	101
[4]	{berries}	=> {whole milk}	0.011794611	0.3547401	0.0332486	1.388328	116

Рисунок 2.17 — Лістинг результатів скрипта 2.15

Джерело: розраховано автором самостійно

Із чотирьох правил, в яких є ягоди, перші два здаються досить цікавими, щоб їх можна було назвати корисними. Крім збитих вершків, ягоди також часто купують з йогуртом - поєднання, яке добре підходить для сніданку або ланчу, а також як десерт.

У функції `subset()` багато можливостей. Критерії вибору підмножини можуть визначатися за допомогою таких ключових слів і операторів.

- Ключове слово `items`, що вже зустрічалося раніше, позначає елемент, який може з'являтися в будь-якому місці правила. Щоб обмежити підмножину правилами, у яких елемент присутній тільки з лівого або тільки з правого боку, замість цього слова слід використовувати `lhs` або `rhs` відповідно.

- Оператор `%in%` означає, що в правилі має бути присутнім хоча б один з елементів, що входять до зазначеного списку. Якщо потрібно відібрати всі правила, де присутні ягоди або йогурт, то можна написати `items%in%c("berries", "yogurt")`.

- Також є додаткові оператори для часткової (`%pin%`) та повної відповідності (`%ain%`). Часткова відповідність дає змогу відібрати правила для цитрусових і тропічних фруктів, використовуючи загальний критерій пошуку `items%pin%"fruit"`. Для повної відповідності потрібна наявність усіх перерахованих у списку елементів. Наприклад, за критерієм

```
items%ain%c("berries", "yogurt")
```

будуть знайдені правила, в яких присутні і ягоди, і йогурт.

Оцінювання виділених асоціативних правил за допомогою Cross таблиці.

Функція `crossTable()` дозволяє реалізувати та представляти результати методу для перехресного представлення спільних випадків у всіх парах елементів. В результаті цієї функції отримуємо симетричну матрицю розміром $n \times n$, де n - кількість елементів у x . Тобто матриця відображає кількість збігів/незбігів між парами елементів. В загальному вигляді скрипт цієї функції має такий вигляд:

```
crossTable( x, measure = c("count", "support", "probability", "lift"),
            sort = FALSE )
```

з аргументами: x - об'єкт для крос-табулювання (тобто набір даних по транзакціям); `measure` - показник для отримання результатів, за замовчуванням кількість збігів (`count`); `sort` вказує, чи потрібно відсортувати елементи за підтримкою (може приймати значення `TRUE` або `FALSE`).

Проведемо крос-табулювання для даних про транзакції:

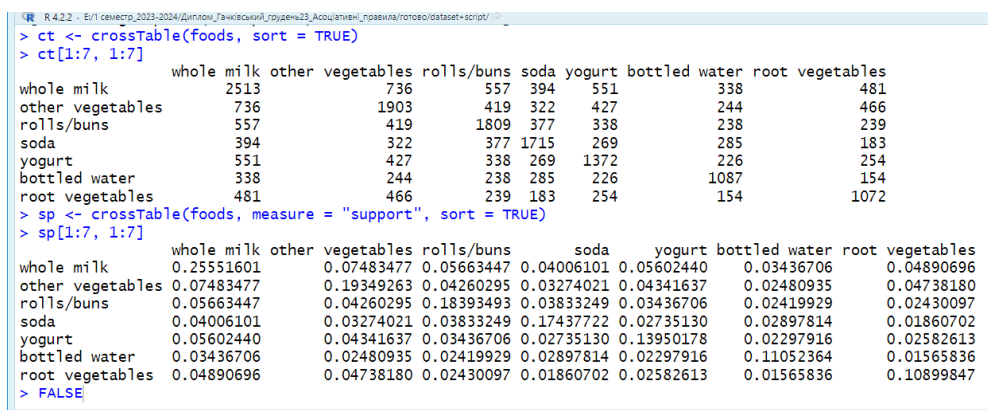
скрипт 2.16

```
ct <- crossTable(foods, sort = TRUE)
```

```
ct[1:7, 1:7]
```

```
sp <- crossTable(foods, measure = "support", sort = TRUE)
```

```
sp[1:7, 1:7]
```



```
R 4.2.2 - E:\1 семестр_2023-2024\Диплом_Гачевський_Григорій_Григорійович\dataset\script/
> ct <- crossTable(foods, sort = TRUE)
> ct[1:7, 1:7]
```

	whole milk	other vegetables	rolls/buns	soda	yogurt	bottled water	root vegetables
whole milk	2513	736	557	394	551	338	481
other vegetables	736	1903	419	322	427	244	466
rolls/buns	557	419	1809	377	338	238	239
soda	394	322	377	1715	269	285	183
yogurt	551	427	338	269	1372	226	254
bottled water	338	244	238	285	226	1087	154
root vegetables	481	466	239	183	254	154	1072

```

> sp <- crossTable(foods, measure = "support", sort = TRUE)
> sp[1:7, 1:7]
```

	whole milk	other vegetables	rolls/buns	soda	yogurt	bottled water	root vegetables
whole milk	0.25551601	0.07483477	0.05663447	0.04006101	0.05602440	0.03436706	0.04890696
other vegetables	0.07483477	0.19349263	0.04260295	0.03274021	0.04341637	0.02480935	0.04738180
rolls/buns	0.05663447	0.04260295	0.18393493	0.03833249	0.03436706	0.02419929	0.02430097
soda	0.04006101	0.03274021	0.03833249	0.17437722	0.02735130	0.02897814	0.01860702
yogurt	0.05602440	0.04341637	0.03436706	0.02735130	0.13950178	0.02297916	0.02582613
bottled water	0.03436706	0.02480935	0.02419929	0.02897814	0.02297916	0.11052364	0.01565836
root vegetables	0.04890696	0.04738180	0.02430097	0.01860702	0.02582613	0.01565836	0.10899847

```
> FALSE
```

Рисунок 2.17 — Лістинг результатів скрипта 2.16

Джерело: розраховано автором самостійно

скрипт 2.17

```
lft <- crossTable(foods, measure = "lift", sort = TRUE)
```

```
lft[1:7, 1:7]
```

```
> lft <- crossTable(foods, measure = "lift", sort = TRUE)
> lft[1:7, 1:7]
```

	whole milk	other vegetables	rolls/buns	soda	yogurt	bottled water	root vegetables
whole milk	NA	1.5136341	1.205032	0.8991124	1.571735	1.216940	1.7560310
other vegetables	1.5136341	NA	1.197047	0.9703476	1.608457	1.160101	2.2466049
rolls/buns	1.2050318	1.1970465	NA	1.1951242	1.339363	1.190373	1.2121013
soda	0.8991124	0.9703476	1.195124	NA	1.124368	1.503577	0.9789636
yogurt	1.5717351	1.6084566	1.339363	1.1243678	NA	1.490387	1.6984751
bottled water	1.2169396	1.1601012	1.190373	1.5035766	1.490387	NA	1.2997827
root vegetables	1.7560310	2.2466049	1.212101	0.9789636	1.698475	1.299783	NA

```
> |
```

Рисунок 2.18 — Лістинг результатів скрипта 2.17

Джерело: розраховано автором самостійно

Результати крос-табулювання, які наведені на рис. 2.17 та 2.18 показують, що проведення процедури побудови асоціативних правил була ефективною.

РОЗДІЛ 3

ПОРІВНЯННЯ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ АЛГОРИТМАМИ APRIORI ТА ECLAT

3.1. Розширені можливості пакету `arules` для виявлення асоціативних правил в середовищі RStudio

Як зазначалось в попередньому розділі, результатом аналізу даних про транзакції в пакеті `arules` є виявлення асоціацій. Концептуально, асоціації - це набори об'єктів, які описують зв'язок між певними елементами (наприклад, у вигляді набору елементів або правила) на основі заданих значень різних мір якості. Це можуть бути міра важливості (наприклад, рівень підтримки), або міра інтересу (наприклад, довіра, ліфт), та інші показники (наприклад, вигода, яку покриває асоціація).

Всі ці типи асоціацій мають єдину реалізацію в пакеті `arules`, який складається з таких функцій:

- `summary()` для короткого огляду всього датасету та `inspect()` для відображення окремих об'єднань,
- `length()` для визначення кількості елементів у множині,
- `items()` виведення для кожної асоціації (асоціативного правила) набору елементів, які є в транзакції (наприклад, об'єднання елементів у `lhs` та `rhs` для кожного правила),
- `sort()` - сортування набору за значеннями різних мір якості,
- `subset()` або `[]` – дозволяє виділити певну/конкретну підмножину,
- операції над множинами (`union()`, `intersect()` та `setequal()`), а також
- порівняння елементів з двох множин (`match()`).

Асоціативні правила, які на даний час реалізовано в пакетах мови R, є множинами наборів елементів (наприклад, використовуються для поширених наборів елементів їх замкненої або максимальної підмножини) та наборів правил (наприклад, правил об'єднання).

Обидва класи, набори елементів і правила, безпосередньо розширюють асоціації віртуальних класів і надають функціональність, описану вище.

У `itemsets` використовується один об'єкт `itemMatrix` в якості бінарної матриці для збереження елементів, де кожен рядок матриці є набором елементів по одній транзакції. Крім того, він може містити списки ідентифікаторів транзакцій у вигляді об'єктів `tidLists`. Важливо пам'ятати, що при представленні транзакцій `tidLists` зберігає для кожного елемента список транзакцій, але тут для кожного набору елементів зберігається список ідентифікаторів транзакцій, у яких додатково з'являється `itemset`. Подібні списки наразі повертає/створює лише функція `eclat()`.

Елементи класу `rules` складаються з двох об'єктів `itemMatrix`, що представляють відповідно ліву (`lhs`) та праву (`rhs`) частини асоціативних правил.

За потреби, параметрами асоціацій та мірами їх якості можна управляти та маніпулювати, змінюючи значення елементів, значення `lhs` і `rhs` та рівня якості. Крім того, методи перевірки валідності виділених асоціацій мають вбудовану перевірку, яка гарантує, що всі компоненти мають прийнятні розміри.

До існуючих асоціацій легко додати нові показники якості. Оскільки шаблон якості містить `data.frame`, можна додати додаткові стовпці з новими мірами якості. Ці нові міри можна використовувати для сортування або вибору асоціацій за допомогою `sort()` або `subset()`. Додавання створення нового списку асоціацій до `arules` також є простим процесом. Для цього розробнику потрібно створити додатковий шар, що розширює клас віртуальних асоціацій, і реалізувати загальний функціонал, описаний вище.

З пакету *arules* використаємо програмні реалізації *Apriori* та *Eclat* Крістіана Боргельта [9], які є у вільному доступі. Код викликається безпосередньо в R за допомогою функцій *apriori()* і *eclat()*, а об'єкти даних безпосередньо передаються з R у код C і назад без запису у зовнішні файли. Розроблені алгоритми можуть шукати набори елементів, що часто зустрічаються, а також замкнені та найбільш поширені набори елементів. Крім того, *apriori()* також може видобувати правила асоціацій.

Дані, які вводяться у функції *apriori()* та *eclat()*, мають бути транзакціями або чимось, що може бути перетворено на транзакції (наприклад, матриця або список). Параметри алгоритму представлені аргументами *parameter* і *control*, які поділяються на дві групи. Параметри видобування (*parameter*) змінюють характеристики виявлених наборів елементів або правил (наприклад, мінімальну підтримку), а параметри керування (*control*) впливають на роботу алгоритму (наприклад, вмикають або вимикають початкове сортування елементів за їх частотою). Ці аргументи мають бути визначені як елементи класів *ARparameter* та *ARcontrol* для функції *apriori()* або *ECparameter* та *ECcontrol* для функції *eclat()*, відповідно. Крім того, до цих класів можна передати дані, які можна примусово передати (наприклад, *NULL*, який надасть значення за замовчуванням, або іменований список з іменами, що дорівнюють іменам слотів, для зміни значень за замовчуванням). У цих класах кожен слот визначає окремий параметр та його значення. Параметри за замовчуванням дорівнюють значенням за замовчуванням автономних програм на мові C [10], за винятком того, що стандартне визначення підтримки правила [11] використовується для вказаної мінімально необхідної підтримки (Боргельт визначає підтримку правила як підтримку його антецедента).

Для *apriori()* також можна використовувати параметр *appearance*, реалізований Крістіаном Боргелтом [10]. За допомогою аргументу *appearance* функції *apriori()* можна вказати, які елементи повинні або не повинні з'являтися в наборах елементів або правилах.

Результатом роботи функцій `apriori()` та `ecslat()` є набір об'єктів класу, що розширює асоціації, який містить набори знайдених асоціацій і може бути надалі проаналізований за допомогою відповідних функцій, передбачених для цих класів.

Існує багато різних алгоритмів, які використовують матрицю інцидентності або список ID транзакцій як вхідні дані і вирішують задачі пошуку частих і складних наборів елементів. Кожен алгоритм має специфічні переваги, які можуть бути важливими для дуже великих баз даних. Такі алгоритми, наприклад, як kDCI, LCM, FP-Growth або Patricia, обговорюються в роботі Goethals and Zaki [13]. Вихідний код більшості алгоритмів доступний у відкритому доступі, і якщо потрібен спеціальний алгоритм, то його інтерфейс є простим і подібним до інтерфейсу для алгоритмів `arules`.

3.2. Пошук асоціативних правил алгоритмом `Apriori` на основі даних про клієнтів банку

Як зазначалось раніше, асоціативні правила вперше були випробувані в роздрібній торгівлі під час пошуку товарів, які купуються спільно з ними. Але на цьому їх застосування не обмежилось. Асоціативні правила широко використовують у сегментації покупців за поведінкою під час покупок, аналізі вподобань клієнтів, крос-маркетингу, адресному надсиланні, у медицині під час визначення впливу побічних ефектів від ліків, для аналізу даних з перепису населення, у прогнозуванні збоїв телекомунікаційного устаткування тощо. [2].

Ефективним є застосування асоціативних правил також у сфері роздрібних банківських послуг. Тут аналізується не ринковий кошик, а спожиті клієнтами послуги у вигляді кредитів, депозитних вкладів, дебетових карток тощо. Виявлені асоціації можна використовувати для підвищення

рентабельності програм лояльності продажу додаткових послуг клієнтам банку, а також для виділення цільового сегмента осіб у разі масової розсилки реклами.

Ще кілька років тому основною перешкодою на шляху поширення кількісних методів аналізу була відсутність як алгоритмів їх пошуку, так і відповідного програмного забезпечення. Але навіть після появи такого ефективного алгоритму, як "Apriori", минуло ще кілька років, перш ніж на ринку інформаційно-аналітичних систем почали з'являтися як комерційні, так і вільно розповсюджувані пакети програм із зрозумілим графічним інтерфейсом і багатими можливостями візуалізації отриманих результатів. Сьогодні можна говорити про іншу проблему - недостатню практичну розробленість питань використання алгоритмів виявлення асоціативних правил у банківській сфері.

Для демонстрації застосування алгоритмів асоціативних правил в банківській діяльності в даній роботі використовується база даних, що містить транзакції по клієнтам банку. Набір даних розміщено в репозиторії UCI Repository of Machine Learning Databases, створеному факультетом інформатики та комп'ютерних наук Каліфорнійського університету [16].

У початковому застосуванні даних атрибути використовувалися для прогнозування рівня доходу осіб. Для розв'язання поставленої у роботі задачі введено атрибут "дохід" з рівнями "низький" та "високий", які відповідають доходу ≤ 50 тис.\$ США та > 50 тис.\$ США відповідно. Ці дані включені до таблиць як набір даних BankCustomers UCI. Крім того, з нього вилучено неперервний атрибут fnlwgt (кінцева вага), а також education-num, оскільки він є лише числовим відображенням атрибуту education. Інші 4 неперервні атрибути перетворенні на порядкові у такий спосіб:

- вік (age): розбитий на рівні "молодь" (0-25 років), "люди середнього віку" (26-45 років), "люди старшого віку" (46-65 років) та "люди похилого віку" (66+)

- кількість робочих годин на тиждень (hours-per-week): поділено на рівні Неповний робочий день (0-25), Повний робочий день (25-40), Понаднормовий робочий день (40-60) та Занадто багато (60+)

- приріст та втрата капіталу: кожна з них поділена на такі рівні: Відсутній (0), Низький (медіанне значення банківського капіталу клієнта більше за нуль і менше за max) та Високий ($\geq \text{max}$)

Зчитані дані збережено під назвою clients:

#Скрипт 3.1

```
> data("BankCustomersUCI")
> clients <- BankCustomersUCI
> dim(clients)
> clients[1:7, ]
```

```
> populat <- AdultUCI
> dim(populat)
[1] 48842 15
> populat[1:7, ]
  age   workclass   fnlwgt   education   education-num   marital-status   occupation   relationship   race   sex
1  39   State-gov   77516   Bachelors      13   Never-married   Adm-clerical   Not-in-family   white   Male
2  50 Self-emp-not-inc 83311   Bachelors      13   Married-civ-spouse   Exec-managerial   Husband   white   Male
3  38   Private   215646   HS-grad        9   Divorced   Handlers-cleaners   Not-in-family   white   Male
4  53   Private   234721   11th          7   Married-civ-spouse   Handlers-cleaners   Husband   Black   Male
5  28   Private   338409   Bachelors      13   Married-civ-spouse   Prof-specialty     wife   Black   Female
6  37   Private   284582   Masters       14   Married-civ-spouse   Exec-managerial     wife   white   Female
7  49   Private   160187   9th           5   Married-spouse-absent   Other-service   Not-in-family   Black   Female

 capital-gain capital-loss hours-per-week native-country income
1      2174         0         40   United-States   small
2         0         0         13   United-States   small
3         0         0         40   United-States   small
4         0         0         40   United-States   small
5         0         0         40     Cuba     small
6         0         0         40   United-States   small
7         0         0         16    Jamaica     small
```

Рисунок 3.1 – Лістинг виведення результатів скриптом 3.1

Джерело: розраховано автором самостійно в RStudio

Датасет BankCustomersUCI містить набір категоріальних та кількісних атрибутів. Тому потрібно провести певну підготовку даних, перш ніж його можна буде перетворити на дані транзакцій, придатні для пошуку асоціацій. По-перше, видалено два атрибути fnlwgt і education-num. Перший атрибут - це вага, розрахована творцями набору даних на основі контрольних даних, наданих відділом народонаселення Бюро перепису населення США. Другий вилучений атрибут є лише числовим представленням атрибуту освіта, який також є частиною набору даних.

#Скрипт 3.2

```
> clients[["fnlwtgt"]] <- NULL
> clients[["education-num"]] <- NULL
```

Далі потрібно представити чотири метричні характеристики (вік, кількість годин на тиждень, приріст та втрату капіталу) з порядковими характеристиками, створивши відповідні категорії. А саме, атрибути вік та години-тиждень розділено на відповідні категорії, згідно наявної інформації про типові вікові групи та робочий час. Для двох атрибутів, пов'язаних з капіталом, створено категорію "Немає" (None) для випадків, які не мають прибутків/втрат. А далі прибутки/збитки розподілено за медіанним значенням на дві категорії "Низькі" та "Високі".

#Скрипт 3.3

```
populat[["age"]] <- ordered(cut(populat[["age"]],
                              c(15, 25, 45, 65, 100)),
                           labels = c("Young", "Middle-aged", "Senior", "Old"))
populat[["hours-per-week"]] <- ordered(cut(populat[["hours-per-week"]],
                                           c(0, 25, 40, 60, 168)),
                                       labels = c("Part-time", "Full-time", "Over-time", "Workaholic"))
populat[["capital-gain"]] <- ordered(cut(populat[["capital-gain"]],
                                         + c(-Inf, 0, median(populat[["capital-gain"]][populat[["capital-gain"]] > 0]), Inf)),
                                   labels = c("None", "Low", "High"))
populat[["capital-loss"]] <- ordered(cut(populat[["capital-loss"]],
                                         + c(-Inf, 0, median(populat[["capital-loss"]][AdultUCI[["capital-loss"]] > 0]), Inf)),
                                   labels = c("none", "low", "high"))
```

Рисунок 3.2 – Лістинг виведення результатів скриптом 3.3

Джерело: розраховано автором самостійно в RStudio

Після означених коригувань дані примусово перекодовані в бінарну матрицю частот і їх можна автоматично записати як набір даних по транзакціям.

#Скрипт 3.4

```
> popul <- as(clients, "transactions")
> popul
```

```
R 4.2.2 · E:\1 семестр_2023-2024\Диплом_Ганківський_грудень23_Асоціативні_правила/rozoce/dataset+script/
> popul
transactions in sparse format with
48842 transactions (rows) and
113 items (columns)
> |
```

Рисунок 3.3 – Лістинг виведення результатів скриптом 3.4

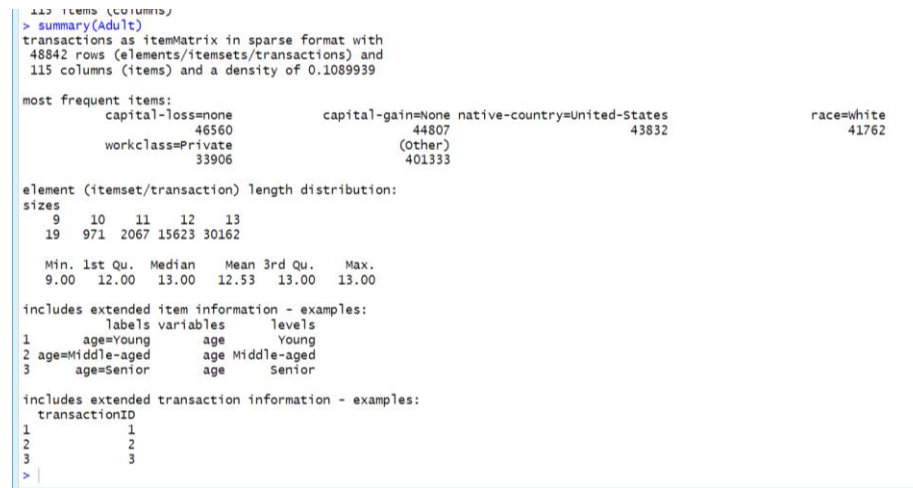
Джерело: розраховано автором самостійно в RStudio

Решта 115 категоріальних атрибутів автоматично перекодовані у 115 бінарних елементів. Під час кодування були згенеровані мітки елементів у

вигляді <ім'я змінної>=<категорія мітка>. При цьому атрибути з відсутніми значеннями елементів отримали значення нуль.

#Скрипт 3.5

```
> summary(popul)
```



```
## items (columns)
> summary(Adult)
transactions as itemMatrix in sparse format with
48842 rows (elements/itemsets/transactions) and
115 columns (items) and a density of 0.1089939

most frequent items:
      capital-loss=none      capital-gain=None native-country=United-States      race=white
      46560                44807                43832                41762
      workclass=Private      (Other)
      33906                401333

element (itemset/transaction) length distribution:
sizes
  9   10   11   12   13
19  971 2067 15623 30162

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   9.00  12.00   13.00  12.53   13.00   13.00

includes extended item information - examples:
  labels variables  levels
1 age=Young      age    Young
2 age=Middle-aged age Middle-aged
3 age=Senior     age    Senior

includes extended transaction information - examples:
transactionID
1             1
2             2
3             3
>
```

Рисунок 3.4 – Статистичні характеристики перетвореного набору транзакцій popul

Джерело: розраховано автором самостійно в RStudio

Функція `summary()` набору даних про транзакції наочно показує елементи, які найчастіше зустрічаються в наборі, розподіл транзакцій за довжиною та розширену інформацію про елементи, що показує, яка змінна та яке її значення використано для створення кожного бінарного елемента. З першого прикладу видно, що елемент з міткою `age=Middle-aged` створено за допомогою змінної `age` та рівня `middle-aged`.

Щоб побачити, які елементи є важливими в наборі даних, можна скористатися функцією `itemFrequencyPlot()`. Аби зменшити кількість елементів, потрібно побудувати графік частоти лише для елементів з підтримкою більше 10%. Для кращої читабельності підписів потрібно зменшити розмір підписів за допомогою параметра `sex.names`. Візуалізація наведена на рис. 3.5.

#Скрипт 3.6

```
> itemFrequencyPlot(popul[, itemFrequency(popul) > 0.1],
                    sex.names = 0.7)
```

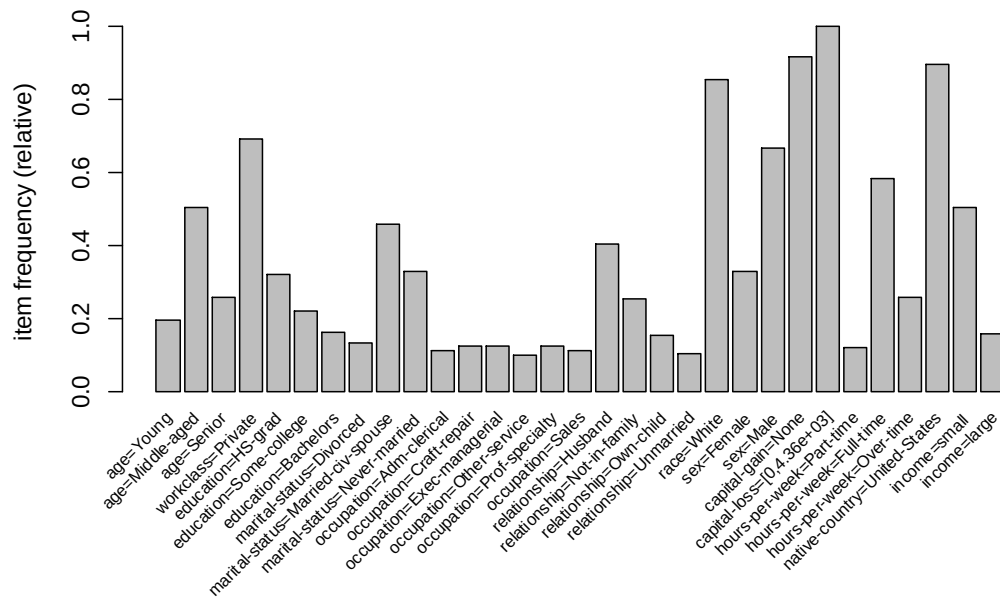


Рисунок 3.5 – Гістограма розподілу елементів у наборі даних `popul` з рівнем довіри більше 10%.

Джерело: розраховано автором самостійно в RStudio

Далі запускаємо функцію `apriori()`, щоб виявити всі асоціативні правила. За замовчуванням для алгоритму `apriori` асоціації будуть виявлятися з мінімальною підтримкою в 1% та рівнем довіри 0.6.

#Скрипт 3.7

```
> rules <- apriori(popul,
                    parameter = list(support = 0.01, confidence = 0.6))
> rules
# set of 286792 rules
```

```
> rules <- apriori(popul, parameter = list(support = 0.01, confidence = 0.6))
Apriori

Parameter specification:
 confidence minval smax arem aval originalsupport maxtime support minlen maxlen target ext
          0.6   0.1   1 none FALSE                TRUE     5   0.01     1    10 rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
  0.1 TRUE TRUE  FALSE TRUE   2    TRUE

Absolute minimum support count: 488

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[113 item(s), 48842 transaction(s)] done [0.14s].
sorting and recoding items ... [65 item(s)] done [0.03s].
creating transaction tree ... done [0.14s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [2.27s].
writing ... [286792 rule(s)] done [0.73s].
creating S4 object ... done [0.62s].
```

Рисунок 3.6 – Результати виявлення асоціативних правил в наборі `popul`

Джерело: розраховано автором самостійно в RStudio

Спочатку функція виводить використані параметри. Окрім заданих мінімального рівня підтримки та довіри, всі параметри мають значення за замовчуванням. Важливо зазначити, що параметр `maxlen`, максимальний розмір шуканих наборів частих елементів, за замовчуванням обмежено до 5. Довші асоціативні правила видобуваються лише за умови, що `maxlen` має вище значення. Після налаштування параметрів виводиться результат реалізації алгоритму на мові C з інформацією про хронометраж.

Результатом роботи алгоритму є набір з 286792 правил. Для огляду видобутих правил можна використовувати функцію `summary()`. Вона показує кількість правил, найчастіші елементи, що містяться у лівій та правій частинах та їх відповідні розподіли за довжиною, а також зведену статистику для мір якості, які повертає алгоритм видобування.

#Скрипт 3.8

> `summary(rules)`

```
> summary(rules)
set of 286792 rules

rule length distribution (lhs + rhs): sizes
  1     2     3     4     5     6     7     8     9    10
  6    420   4919 22222 53451 77178 70223 40734 14636 3003

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.000  5.000   6.000   6.325  7.000  10.000

summary of quality measures:
  support    confidence    coverage    lift    count
Min.   :0.01001  Min.   :0.6000  Min.   :0.01001  Min.   : 0.7171  Min.   : 489
1st Qu.:0.01251  1st Qu.:0.7720  1st Qu.:0.01462  1st Qu.: 1.0000  1st Qu.: 611
Median :0.01706  Median :0.9108  Median :0.02037  Median : 1.0554  Median : 833
Mean   :0.02697  Mean   :0.8682  Mean   :0.03175  Mean   : 1.3111  Mean   :1317
3rd Qu.:0.02750  3rd Qu.:0.9750  3rd Qu.:0.03276  3rd Qu.: 1.2989  3rd Qu.:1343
Max.   :1.00000  Max.   :1.0000  Max.   :1.00000  Max.   :20.6826  Max.   :48842

mining info:
 data ntransactions support confidence
popul      48842      0.01      0.6

apriori(data = popul, parameter = list(support = 0.01, confidence = 0.6))
>
```

Рисунок 3.7 – Статистичні характеристики по виявленим асоціативним правилам за транзакціями `popul`

Джерело: розраховано автором самостійно в RStudio

Як і при пошуку асоціативних правил, кількість знайдених шаблонів дуже велика. Для детального аналізу цих правил можна скористатися функцією `subset()`, щоб створити окремі підмножини правил для кожного елемента, наприклад, які в результаті мають змінну доходу в правій частині

правила. Крім того встановлюємо вимогу, що рівень ліфта повинен бути більше 1,2.

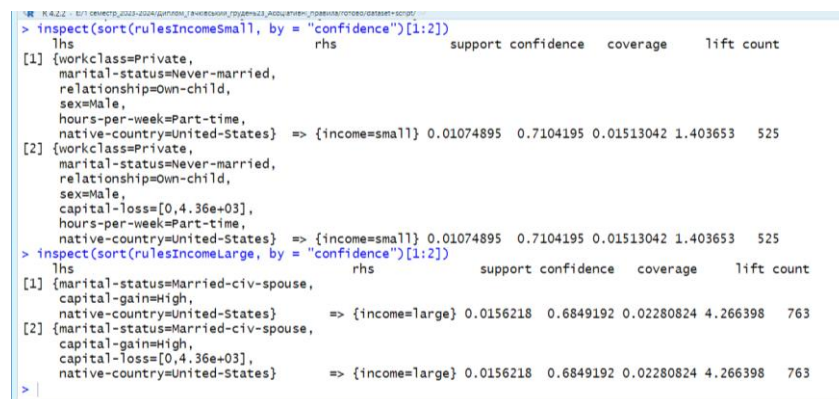
#Скрипт 3.9

```
> rulesIncomeSmall <- subset(rules,
subset = rhs %in% "income=small" & lift > 1.2)
> rulesIncomeLarge <- subset(rules,
subset = rhs %in% "income=large" & lift > 1.2)
```

Тепер у нас є набір з правилами для осіб з низьким та високим доходом окремо. Для порівняння перевіримо для обох наборів перші два правила з найвищою достовірністю (за допомогою sort ()).

#Скрипт 3.10

```
> inspect(sort(rulesIncomeSmall, by = "confidence")[1:2])
> inspect(sort(rulesIncomeLarge, by = "confidence")[1:2])
```



```
> inspect(sort(rulesIncomeSmall, by = "confidence")[1:2])
  lhs      rhs      support confidence  coverage  lift count
[1] {workclass=Private,
marital-status=Never-married,
relationship=Own-child,
sex=Male,
hours-per-week=Part-time,
native-country=United-States} => {income=small} 0.01074895 0.7104195 0.01513042 1.403653 525
[2] {workclass=Private,
marital-status=Never-married,
relationship=Own-child,
sex=Male,
capital-loss=[0,4.36e+03],
hours-per-week=Part-time,
native-country=United-States} => {income=small} 0.01074895 0.7104195 0.01513042 1.403653 525
> inspect(sort(rulesIncomeLarge, by = "confidence")[1:2])
  lhs      rhs      support confidence  coverage  lift count
[1] {marital-status=Married-civ-spouse,
capital-gain=High,
native-country=United-States} => {income=large} 0.0156218 0.6849192 0.02280824 4.266398 763
[2] {marital-status=Married-civ-spouse,
capital-gain=High,
capital-loss=[0,4.36e+03],
native-country=United-States} => {income=large} 0.0156218 0.6849192 0.02280824 4.266398 763
> |
```

Рисунок 3.8 – Лістинг виведення інформації по двом правилам в підмножинах з низьким та високим доходом

Джерело: розраховано автором самостійно в RStudio

З рис.3.8 видно, що працівники з невеликим доходом (Small) – це чоловіки (sex=Male), які працюють в приватному секторі (workclass=Private), неповний робочий день (hours-per-week=Part-time), ніколи не були одружені (marital-status=Never-married), мають одну дитину (relationship=Own-child). А люди, які мають високий дохід (Large) – це одружені, мають високий приріст капіталу та народилися в США. Цей приклад показує, що за допомогою вибору

підмножин і сортування можна проаналізувати набір видобутих асоціацій, навіть якщо він дуже великий.

3.3. Вивлення асоціативних правил алгоритмом Eclat на основі даних про клієнтів банку

Алгоритм Eclat (Equivalence Class Transformation) є ефективним методом для видобутку асоціативних правил з набору даних. Алгоритм Eclat працює у горизонтальному форматі даних, тобто кожен запис представляється як транзакція, а не у вертикальному форматі, як в алгоритмі Apriori.

Основні кроки алгоритму Eclat такі:

Крок 1. Трансформація даних про транзакції у вертикальному форматі. Для кожного товару створюється список транзакцій, в яких цей товар міститься. Це і називається вертикальним представленням даних.

Крок 2. Пошук товарів, які часто зустрічаються в транзакціях. Алгоритм шукає часті набори товарів за допомогою методу рекурсивного пошуку. Починаючи з одного товару, алгоритм рекурсивно розглядає всі інші товари, які можна додати до поточного набору. Під час такого пошуку підраховується підтримка для кожного набору товарів.

Крок 3. Виявлення асоціативних правил. На основі частих наборів товарів, знайдених на попередньому кроці, алгоритм генерує асоціативні правила. Для кожного частого набору алгоритм розглядає всі можливі комбінації піднаборів і формує правила.

Крок 4. Рекурсивний Пошук:

Алгоритм використовує рекурсивний підхід для пошуку частих наборів та асоціативних правил, використовуючи еквівалентні класи.

Алгоритм Eclat має кілька переваг:

- Він ефективно працює з великими обсягами даних.
- Використовує горизонтальну структуру даних, що полегшує обробку.

Однак, алгоритм Eclat може бути менш ефективним порівняно з алгоритмом Apriori, коли набір даних має велику кількість унікальних товарів або коли існують набори даних зі значущими кількостями нульових значень.

Також у цьому прикладі покажемо, як можна додати нову міру повної достовірності (all-confidence) [15]. Повна достовірність (або повна довіра) для множини елементів X визначається за формулою

$$all - Conf_{X \Rightarrow Y} = \frac{Supp_X}{Supp_{I \cup X} Supp_I} \quad (3.1)$$

Ця величина має властивість $conf(I \rightarrow XI) \geq all - confidence(X)$ для всіх $I \subset X$. Це означає, що всі можливі правила, згенеровані в наборі елементів X , повинні мати принаймні достовірну ймовірність, яка дорівнює довірчій ймовірності набору елементів. В роботі [15] показано, що підтримка в знаменнику рівняння (3.1) повинна впливати з одного елемента і може бути спрощена до $\max_{i \in X} supp(\{i\})$. Щоб отримати набір елементів для обчислення повної достовірності, виявлено асоціативні правила вже в згаданому раніше наборі транзакцій `popul` за допомогою алгоритму Eclat.

#Скрипт 3.11

```
> ECrules <- eclat(popul, parameter = list(support = 0.05),
               control = list(verbose = FALSE))
> ECrules
```

```
## set of 8829 itemsets
```

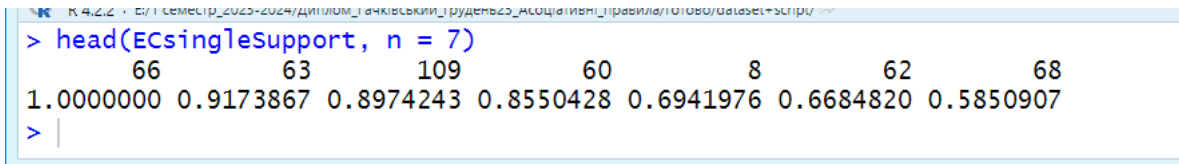
В результаті, алгоритм Eclat виявив набір з 8829 правил. Для порівняння, за Apriori було виявлено 286792 правил, але аналіз показав, що частина цих правил є неінформативними (скрипт 3.7).

Для визначення показника повної достовірності потрібно знайти всі добути одиничні елементи та їх відповідні значення підтримки. Для цього

створено іменований вектор, де іменами є номери стовпців елементів, а значення - їхня підтримка.

#Скрипт 3.12

```
> ECsingleItems <- ECrules[size(items(ECrules)) == 1]
> ECsingleSupport <- quality(ECsingleItems)$support
> names(ECsingleSupport) <- unlist(LIST(items(ECsingleItems),
                                     decode = FALSE))
> head(ECsingleSupport, n = 7)
```



```
> head(ECsingleSupport, n = 7)
      66      63     109      60      8      62      68
1.0000000 0.9173867 0.8974243 0.8550428 0.6941976 0.6684820 0.5850907
> |
```

Рисунок 3.9 – Лістинг виведення інформації по семи правилам в підмножинах з найбільшим рівнем підтримки

Джерело: розраховано автором самостійно в RStudio

Далі обчислено повну довірчу ймовірність, згідно формули (3.1) для всіх наборів елементів. Одиначний елемент підтримки, який потрібен для позначення, визначається з іменованого вектора ECsingleSupport (скрипт 3.12), і отримане при цьому значення додається до фрейму даних про якість набору.

#Скрипт 3.13

```
> ECitemsetList <- LIST(items(ECrules), decode = FALSE)
> ECallConfidence <- quality(ECrules)$support/sapply(ECitemsetList,
                                                       function(x) max(ECsingleSupport[as.character(x)]))
> ECquality <- cbind(quality(ECrules), ECallConfidence)
> summary(ECrules)
> summary(ECquality)
```

Тепер новий показник «повна довіра» є частиною даного набору елементів.

```

> summary(ECrules)
set of 8829 itemsets

most frequent items:
capital-loss=[0,4.36e+03]      capital-gain=None native-country=United-States
                        4415                        4136                        4128
                        race=white      workclass=Private      (other)
                        3926                        3274                        22946

element (itemset/transaction) length distribution:sizes
 1  2  3  4  5  6  7  8  9 10
36 304 1089 2130 2483 1782 781 197 26 1

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.00   4.00   5.00   4.85   6.00   10.00

summary of quality measures:
support      count
Min. :0.05002  Min. : 2443
1st Qu.:0.06032 1st Qu.: 2946
Median :0.07569 Median : 3697
Mean :0.10180  Mean : 4972
3rd Qu.:0.11398 3rd Qu.: 5567
Max. :1.00000  Max. :48842

includes transaction ID lists: FALSE

mining info:
data ntransactions support
popul 48842 0.05

eclat(data = popul, parameter = list(support = 0.05), control = list(verbose = FALSE))
> summary(ECquality)
support      count      EcallConfidence
Min. :0.05002  Min. : 2443  Min. :0.05002
1st Qu.:0.06032 1st Qu.: 2946  1st Qu.:0.06449
Median :0.07569 Median : 3697  Median :0.08257
Mean :0.10180  Mean : 4972  Mean :0.11436
3rd Qu.:0.11398 3rd Qu.: 5567  3rd Qu.:0.12545
Max. :1.00000  Max. :48842  Max. :1.00000

```

Рисунок 3.10 – Лістинг виведення результатів скриптом 3.13

Джерело: розраховано автором самостійно в RStudio

Цей показник (повна довіра, all-Confidence) можна використовувати для маніпулювання множинами транзакцій. Наприклад, при перегляді наборів елементів, які пов'язані з освітою (education), їх можна відсортувати за повною довірою (тобто спершу відфільтровано набори елементів довжиною 1, оскільки вони за визначенням мають довірчу ймовірність 1).

#Скрипт 3.14

```

> ECrules_Education <- subset(ECrules, subset = items %in% "education")
inspect(sort(ECrules_Education[size(ECrules_Education) > 1],
by = "allConfidence")[1:3])

```

```

  items                                support allConfidence
1 {education=HS-grad,
  hours-per-week=Full-time} 0.2090209      0.3572453
2 {education=HS-grad,
  income=small}              0.1807051      0.3570388
3 {workclass=Private,
  education=HS-grad}         0.2391794      0.3445408

```

Рисунок 3.11 – Лістинг виведення результатів скриптом 3.14

Джерело: розраховано автором самостійно в RStudio

Як видно з рис. 3.11, отримані набори елементів показують, наприклад, що елемент "випускник середньої школи (але без вищої освіти) сильно асоціюється з атрибутами: робота на повну ставку, невеликий дохід та робота в приватному секторі. Повна довіра вже реалізована в функції `arules` параметром `allConfidence()`.

На завершальному кроці проведено порівняння алгоритмів побудови асоціативних правил `Apriori` і `Eclat` з метою обрання оптимального методу.

Спочатку, при реалізації кожного алгоритму, добирався параметр підтримки. Результат цих дій можна побачити в табл. 3.1, що свідчить про значну перевагу алгоритму `Eclat`. Це, в свою чергу, можна обґрунтувати роботою з, так званими, TID-множинами.

Таблиця 3.1. Порівняння алгоритмів `Apriori` та `Eclat` за рівнем підтримки для даних `BankCustomers`

Рівень підтримки	Eclat	Apriori
0,1	0,00 сек	0,00 сек
0,01	0,00 сек	0,01 сек
0,001	3,17 сек	5,01 сек
0,0001	5,25 сек	181,15 сек
0,00001	7,39 сек	427,37 сек

Джерело: розраховано та складено автором самостійно

На наступному кроці, параметр підтримки фіксовано, змінюється кількість даних на вхід. Результат відображено в табл. 3.2.

Таблиця 3.2. Порівняння алгоритмів `Apriori` та `Eclat` за різними обсягами даних

Обсяг вибірки	Eclat	Apriori
500	0,00 сек	0,00 сек
1 000	0,00 сек	0,01 сек
2 000	0,01 сек	0,22 сек
3 000	0,08 сек	0,92 сек

4 000	1,83 сек	102,98 сек
-------	----------	------------

Джерело: розраховано та складено автором самостійно

Як можна помітити, алгоритм Apriori практично не здатний працювати на великому обсязі даних. Отже, враховуючи, що завдання аналізу даних актуальні лише при опрацювання великої кількості вхідних значень, можна зробити висновок, що перевагу варто віддати алгоритму Eclat.

ВИСНОВКИ

Дослідження споживчої поведінки (вибору), виявлення її особливостей та можливостей застосування алгоритмів машинного навчання для його вивчення - непросте завдання, під час вирішення якого зроблено такі висновки:

- обираючи блага для споживання, покупець орієнтується на свої вподобання. Поведінка споживача є раціональною, зокрема, він переслідує певні цілі та керується особистим інтересом. Споживач прагне максимізувати сукупну корисність, іншими словами, прагне обрати такий набір благ, який принесе йому найбільшу загальну корисність. На вибір споживача і його суб'єктивні оцінки корисності благ, які він купує, впливає закон спадної граничної корисності. При виборі благ можливості споживача обмежені їх цінами і його доходом, таке обмеження називається бюджетним обмеженням. Разом із загальними принципами вибору раціонального споживача існують і особливі, які визначаються впливом смаків і побажань.

- Нині розробляють галузеві та спеціалізовані за іншими ознаками інструменти аналізу. Загалом же набір аналітичних інструментів дослідження бізнес-показників доволі великий і став можливим після активного застосування до них алгоритмів машинного навчання.

При виконанні даної роботи отримано такі результати:

- досліджено можливості застосування алгоритмів асоціативних правил як до вивчення закономірностей споживчого кошику покупців продуктового магазину, так і для аналізу клієнтів банку з метою вивчення їх особистих даних.

- За допомогою пакету *arules* виявлено закономірності в даних з метою побудови асоціативних правил на даних різного походження. В ході виконання роботи виявлено, що основними перевагами пакету *arules* є його

ефективна реалізація з використанням розріджених матриць; він має простий та інтуїтивно зрозумілий інтерфейс для маніпуляції та аналізу даних транзакцій, наборів елементів та правил з можливістю вибору та сортування підмножин. Важливою особливістю пакету *arules* також є його гнучкість у додаванні нових мір якості та додаткових описів елементів і транзакцій, які можна використовувати для відбору транзакцій та аналізу отриманих асоціацій; розширення структури даних, що дозволяє легко впроваджувати нові типи асоціацій та підключення нових алгоритмів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hahsler M., Hornik K., Reutterer T. Implications of Probabilistic Data Modeling for Mining Association Rules, 2005.
2. Gaul W., Vichi M., Weihs C. From Data and Information Analysis to Knowledge Engineering. *Studies in Classification, Data Analysis and Knowledge Organization*, 2006. P. 598-605
3. Hahsler M., Gruen B., Hornik K. arules — A Computational Environment for Mining Association Rules and Frequent Item Sets. *Journal of Statistical Software*, 2005. Vol. 14.]
4. <https://cran.r-project.org/web/packages/arules/arules.pdf>
5. Hahsler, M., Grun, B., & Hornik, K. Arules - A Computational Environment for Mining Association Rules and Frequent Item Sets. *Journal of Statistical Software*. 2005, №14(15), 1–25. <https://doi.org/10.18637/jss.v014.i15>
6. Antonie L., Li J., Zaiane O. Negative Association Rules. In: Aggarwal C., Han J. (eds) Frequent Pattern Mining, *Springer International Publishing*, 2014. pp. 135-145. https://doi.org/10.1007/9783319-078212_6
7. Ситник В. Ф., Краснюк М. Т. Інтелектуальний аналіз даних(дейтамайнінг). Київ : КНЕУ, 2007.
8. Leibenstein H. Bandwagon, Snob, and Veblen Effects in the Theory of Consumers' Demand. *The Quarterly Journal of Economics*, Vol. 64, No. 2 (May, 1950), pp. 183-207. URL: <http://www.jstor.org/stable/1882692>
9. Borgelt C., Kruse R. Induction of Association Rules: Apriori Implementation. *Proc. 14th Conf. on Computational Statistics (COMPSTAT)*. Berlin, Germany 2002. https://link.springer.com/chapter/10.1007/978-3-642-57489-4_59
10. Borgelt C. Efficient Implementations of Apriori and Eclat. *Workshop of Frequent Item Set Mining Implementations*. 2003. https://borgelt.net/papers/fimi_03.pdf

11. Srikant, R. and Agrawal, R. Mining sequential patterns. San Jose [etc.]: IBM Thomas J. Watson Research Division. 1996.
12. <http://epub.wu-wien.ac.at/> from January 2003 to August 2005.
13. Goethals B., Zaki M.J. Chapter Frequent Set Mining. *Data Mining and Knowledge Discovery Handbook* 2003.
https://www.researchgate.net/publication/225880982_Frequent_Set_Mining
14. <http://epub.wu-wien.ac.at/>
15. Omiecinski ER. "Alternative Interest Measures for Mining Associations in Databases." *IEEE Transactions on Knowledge and Data Engineering*. 2003, №15(1), 57–69.
16. A. Asuncion & D. J. Newman (2007): UCI Repository of Machine Learning Databases. Irvine, CA: University of California, Department of Information and Computer Science.
17. Kohavi, R. (1996). Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.
18. Kovalenko I., Davydenko Ye., Shved A. Development of the procedure for integrated application of scenario prediction methods. *Eastern-European Journal of Enterprise Technologies*. 2019, vol. 2, iss. 4 (98). pp. 31-38. doi: 10.15587/1729-4061.2019.163871
19. Antipova K. O., Kovalenko I., Davydenko Ye., Shved A. Modelling of group expert judgments under conditions of complex uncertainty. *East European Scientific Journal*. 2019, vol 5 (45), pp. 4-10.
20. Молдавська А.В. Метод формування багаторівневих послідовних патернів. *Проблеми програмування*. 2016, № 2-3. С. 158-163.
<http://dspace.nbuv.gov.ua/handle/123456789/126401>
21. Іванченко Г. Ф. Системи штучного інтелекту : навч. посіб. — К. : КНЕУ, 2011. — 382 с.
22. Коваленко, І. І., Давиденко, Є. О., & Швед, А. В. МЕТОДИКА ПОШУКУ АСОЦІАТИВНИХ ПРАВИЛ. *Вісник Черкаського державного*

технологічного університету. 2019, №3, с. 50–55.
<https://doi.org/10.24025/2306-4412.3.2019.176909>

23. Koh Y. S., Rountree N. Rare association rule mining and knowledge discovery: technologies for infrequent and critical event detection. *New York: Information Science Reference*, 2009. doi: 10.4018/978-1-60566-754-6.

24. Інформаційні системи в економіці : монографія / под ред. Устенко С. В. — К. : КНЕУ, 2011. — 424 с.

25. *Encyclopedia of artificial intelligence* / Eds.: J. R. Dopico, J. D. de la Calle, A. P. Sierra. *New York : Information Science Reference*, 2009. Vol. 1–3. 1677 p.

26. Fournier-Viger, P et. al. (2014) SPMF: a Java open-source pattern mining library. *J. Mach. Learn. Res.* 2014, №1. pp 3389-3393.

27. Chen, Z., Roussopoulos, M., Liang, Z., Zhang, Y., Chen, Z. and Delis, A. Malware characteristics and threats on the internet ecosystem. *Journal of Systems and Software*. 2012, №85(7), pp.1650-1672.

28. Sami, A., Yadegari, B., Peiravian, N., Hashemi, S., Hamze, A. Malware detection based on mining API calls. *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10*, pp.1020-1025.

29. Черняк О. І., Ставицький А. В., Чорноус Г. О. Системи обробки економічної інформації: Підручник. Київ: Знання, 2006.

30. Ishida Y. Immunity-based systems: a design perspective. Berlin, Springer, 2004, №177 p. DOI: 10.1007/978-3-662-07863-1.

31.. Segel L.A., Cohen I. R. Design principles for immune system and other distributed autonomous systems. *New York, Oxford University Press*, 2001, 428 p.

32. Flower D., Timmis J. In silico immunology. *New York, Springer*, 2007, 451 p. DOI: 10.1007/978-0-387-39241-7.

33. Chen W., Li T., Qin J. [et al.] A new cluster based real negative selection algorithm, *Information and automation*, 2011, Vol. 86, pp. 125–131. DOI: 10.1007/978-3-642-19853-3_18.

34. Gonzalez F., Dasgupta D., Gomez J. The effect of binary matching rules in negative selection, *Genetic and Evolutionary Computation: conference GECCO-2003, Chicago, 9–11 July 2003: proceedings*. Berlin-Heidelberg, Springer-Verlag, 2003, pp. 195–206. DOI: 10.1007/3-540-45105-6_25.
35. Ong A. An adaptive anomaly detection system using data mining and artificial immune system. London, King's College London, 2007, 348 p.
36. Subbotin S. A., Olejnik An. A., Gofman E. A., Zajcev S. A., Olejnik Al. A.; pod obshh. red. S. A. Subbotina. Intellektual'nye informacionnye tehnologii proektirovaniya avtomatizirovannyh sistem diagnostirovaniya i raspoznavaniya obrazov: monografija. Har'kov, Kompanija SMIT, 2012, 318 p.
37. Clarke B. Fokoue E., Zhang H. H. Principles and theory for data mining and machine learning. New York, Springer. 2009, №781 p. DOI: 10.1007/978-0-387-98135-2.
38. Bishop C. M. Pattern recognition and machine learning. New York, Springer. 2006, 738 p. DOI: 10.1108/03684920710743466.
39. Zhao Y. Post-mining of association rules: techniques for effective knowledge extraction / Y. Zhao, C. Zhang, L. Cao. New York: Information Science Reference. 2009. 372 p.
40. Encyclopedia of machine learning / Eds. C. Sammut, G.I. Webb. New York: Springer. 2011. 1031 p.
41. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посібник / С. О. Субботін. Запоріжжя: ЗНТУ, 2008. 341с.
42. Khan M.S. Weighted Association Rule Mining from Binary and Fuzzy Data / M.S. Khan, M. Mueyba, F. Coenen // Lecture Notes in Computer Science. – 2008. – Vol. 5077. – P. 200 – 212.