МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,
освітня програма «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

"Розробка рекомендаційної системи методами емоційного аналізу та машинного навчання"

Студента 2 курсу групи ІАВ-11

Рябова Олександра Олексійовича

Науковий керівник:

Хлевна Юлія Леонідівна

доктор технічних наук,

доц. кафедри технологій управління

Попередній захист:

_____

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри
технологій управління _____ _____ _____

(підпис) (прізвище, ініціали) (дата)

Київ – 2021

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 - Комп'ютерні науки
Освітня програма Інформаційна аналітика та впливи

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Морозов В.В.

_____
«____» _____20___року

З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент: Рябов Олександр Олексійович

Група: ІАВ-21

1. Тема кваліфікаційної роботи: «Розробка рекомендаційної системи методами емоційного аналізу та машинного навчання»
Затверджена наказом по від «___»_____20__ р. № ___.

2. Строк подання студентом готової роботи – "___"_____ 20___ р.

3. Цільова установка та вихідні дані до роботи: рекомендаційна система побудована за допомогою мови програмування Python та R на основі бази даних, отриманої від інтернет-магазину

4. Зміст роботи: проаналізовані методи підвищення функціоналу електронної комерції, концептуально продемонстровано підвищення функціоналу електронної комерції з використанням рекомендаційних систем, охарактеризовані принципи та етапи аналізу тональності тексту та застосовані задачі класифікації для якісного емоційного аналізу тексту, розроблена експертна система та надані рекомендації щодо використання отриманих систем у діючих електронних бізнесах.

5. Перелік графічного матеріалу (слайдів) 29 рисунків, 12 формул, 1 таблиця, 3 додатки та 13 слайдів презентації

6. Календарний план виконання роботи:

№ п/п	Назва частин роботи	%	Виконання роботи	
			За планом	Фактично
1.	Вибір теми дипломної роботи	3	01.10.2020	01.10.2020
2.	Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників	2	09.11.2020	09.11.2020
3.	Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи	10	08.01.2021	07.01.2021
4.	Складання розгорнутого плану кваліфікаційної роботи	5	18.01.2021	18.01.2021
5.	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.	5	19.01.2021 – 20.01.2021	20.11.2021
6.	Підготовка розділу 1	10	12.02.2021	13.02.2021
7.	Підготовка розділу 2	14	08.03.2021	08.03.2021
8.	Підготовка розділу 3	27	01.04.2021	01.04.2021
9.	Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій	15	03.05.2021	03.05.2021
10	Передача кваліфікаційної роботи науковому керівникові	2	04.05.2021	04.05.2021
11	Передача кваліфікаційної роботи рецензенту для рецензування	2	07.05.2021	07.05.2021
12	Попередній захист кваліфікаційної роботи	5	11.05.2021	11.05.2021

Дата видачі завдання «____» _____ 20__ р.

Керівник роботи: доктор технічних наук, доцент Хлевна Юлія Леонідівна
(посада, прізвище, ім’я, по батькові)

_____
(підпис)

Завдання прийняв до виконання студент групи ІАВ-21

Рябов Олександр Олексійович
(прізвище, ім’я, по батькові)

_____
(підпис)

ЗМІСТ

АНОТАЦІЯ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра технологій управління
Спеціальність 122 - Комп’ютерні науки,
освітня програма «Інформаційна аналітика та впливи»

Дипломна робота магістра Рябова Олександра Олексійовича.

Тема роботи – «Розробка рекомендаційної системи методами емоційного аналізу та машинного навчання».

Мета дипломної роботи магістра – розробка експертної системи з надання рекомендацій для підвищення функціоналу та ефективності ведення бізнесу у сфері електронної комерції.

Об'єктом дослідження є експертні системи у сфері електронної комерції.

Предметом дослідження є інформаційні інструменти та засоби управління даними і процесами у сфері електронної комерції.

Методами досліджень є математичний апарат визначення задач класифікації, дискретні методи, штучні нейронні мережі, регресійні методи, дерева рішень та інші алгоритми машинного навчання для вирішення задач класифікації, статистичні методи та інструменти для обробки та аналізу даних, інструменти візуалізації даних.

Для вирішення завдань даної роботи використано мови програмування, обробки та аналізу даних R та Python.

Практична значимість дослідження полягає в дослідженні та імплементації алгоритмів емоційного аналізу тексту, покращенні методу колаборативної фільтрації, розробці інформаційної системи для надання пропозицій клієнтам, що дозволить підвищити функціонал та ефективность ведення бізнесу у сфері електронної комерції.

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, 3 розділів, висновків. Після цього наводиться список літератури з 70 пунктів та 3 додатків. Загальний обсяг кваліфікаційної роботи становить 111 сторінок, із них 94 сторінки основного тексту, який містить 29 рисунків.

ВСТУП

Сучасні умови економічної конкуренції вимагають від підприємців створення все нових і нових технологій залучення покупців і збільшення продажів. Однією з таких технологій є рекомендаційні системи. Дані системи поширили нові способи взаємодії звичайних веб-сайтів зі своїми користувачами. На заміну надання статистичної інформації, коли потенційні покупці шукають і, ймовірно, купують товари, рекомендаційні системи збільшують ступінь інтерактивності, а також розширюють можливості, що надаються користувачеві. Рекомендаційні системи формують рекомендації незалежно для кожного конкретного користувача на основі його минулих покупок і пошуків, відгуків, а також на основі запитів інших користувачів. Також такі рекомендації збільшують середній чек клієнтів, завдяки пропозиції суміжних товарів або товарів, які певним чином пов'язані з їх минулими покупками та покупками схожих користувачів.

Метою роботи є розробка експертної системи з надання рекомендацій для підвищення функціоналу та ефективності ведення бізнесу у сфері електронної комерції.

Відповідно до поставленої мети потрібно вирішити такі задачі:

1. Визначити проблемну область дослідження.
2. Проаналізувати методи підвищення функціоналу електронної комерції.
3. Узагальнення проблематики підвищення функціоналу за допомогою експертних систем.
4. Формалізувати математичний апарат та алгоритмічні засади задач класифікації як базису емоційного аналізу текст.
5. Структурувати та формалізувати процеси емоційного аналізу тексту.
6. Проаналізувати засоби розробки та впровадження рекомендаційних систем з використанням емоційного аналізу.

7. Формалізувати базу знань для підвищення функціоналу електронної комерції.

8. Розробити модель підвищення функціоналу електронної комерції методами емоційного аналізу та машинного навчання.

9. Розробити рекомендації щодо використання отриманих систем у діючих електронних бізнесах.

Об'єктом дослідження є функціонування та ведення бізнесу у сфері електронної комерції.

Предметом дослідження є інформаційні інструменти та засоби управління даними і процесами у сфері електронної комерції.

Методами досліджень є дискретні методи, штучні нейронні мережі, регресійні методи, дерева рішень та інші алгоритми машинного навчання для вирішення задач класифікації, статистичні методи та інструменти для обробки та аналізу даних, інструменти візуалізації даних та алгоритми .

Для вирішення завдань даної роботи використано мови програмування, обробки та аналізу даних R та Python.

Практична значимість дослідження полягає в дослідженні та імплементації алгоритмів емоційного аналізу тексту, оптимізації методу колаборативної фільтрації розробці інформаційної системи для надання пропозицій клієнтам, що дозволить підвищити функціонал та ефективность ведення бізнесу у сфері електронної комерції.

Наукова новизна даної кваліфікаційної роботи полягає в оптимізації алгоритму колаборативної фільтрації та імплементації емоційного аналізу для покращення ефективності рекомендацій та розширення функціоналу та можливостей системи.

Особистий внесок здобувача. Усі наукові результати, які відображено у кваліфікаційній роботі, отримані автором самостійно. Результати співавторів сумісних публікацій до тексту кваліфікаційної роботи не включено. У надрукованих статтях, опублікованих у співавторстві, магістранту належить наступне: дослідження та імплементація алгоритмів емоційного аналізу тексту,

покращення методу колаборативної фільтрації, розробка інформаційної системи для надання пропозицій клієнтам.

Апробація результатів роботи. Автор виступав доповідачем на VI Information Technology and Interactions (Satellite): Conference Proceedings.

Публікації. Основні наукові положення, висновки і результати магістерської кваліфікаційної роботи знайшли відображення у 3 друкованих працях, з них: 1 тези доповіді на конференції, 1 монографія та 1 стаття у журналі.

Внести у список літератури:

Riabov O., Khlevna I., Recommendation system design in python by methods of emotional analysis and machine learning. Information Technology and Interactions (Satellite): Conference Proceedings, December 04, 2020, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv and [etc]; Vitaliy Snytyuk (Editor). Kyiv: Stylos, 2020.– P. 274 – 276.

Riabov O., Khlevna I., Recommendation system design in python by methods of emotional analysis and machine learning - журнал «Сучасні інформаційні технології», випуск №1 – Київ, 2021.

Структура та обсяг роботи. Кваліфікаційна робота складається зі вступу, 3 розділів, висновків. Після цього наводиться список літератури з 44 пунктів та 3 додатків. Загальний обсяг кваліфікаційної роботи становить 111 сторінок, із них 96 сторінок основного тексту, який містить 29 рисунків.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ЗАСАДИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ТА ОСОБЛИВОСТІ ВПРОВАДЖЕННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Визначення предметної області дослідження

Під електронною комерцією (e-commerce) розуміється комерційна діяльність, пов'язана з поширенням, рекламою, просуванням, а також продажем послуг і товарів через інтернет. У двох словах, це лише процес купівлі-продажу продукції електронними засобами, такими як мобільні додатки та Інтернет. Електронна комерція стосується як роздрібної торгівлі в Інтернеті, так і покупок в Інтернеті, а також електронних транзакцій. Електронна комерція за останні десятиліття надзвичайно зросла в популярності, і, певним чином, вона замінює традиційні магазини.

Хоча більшість людей розглядають електронну комерцію як бізнес для споживача (B2C), існує безліч інших видів електронної комерції . Сюди входять веб-сайти онлайн-аукціонів, Інтернет-банкінг, онлайн-квитки та бронювання, а також транзакції від бізнесу до бізнесу (B2B).

Нещодавно ріст електронної комерції поширився на продажі з використанням мобільних пристроїв що зазвичай називають "m-commerce" і є просто підмножиною електронної комерції.

На початку 2000-х багато людей скептично ставилися до передачі даних своєї картки інтернет-магазину. Тоді як транзакції в електронній комерції зараз є звичайною справою. Сертифікати SSL, шифрування та надійні зовнішні платіжні системи, такі як Paypal, Worldpay та Skrill, допомогли підвищити довіру людей до електронної комерції.

Сфера E-commerce підрозділяється на види залежно від цільової аудиторії, з якою працює компанія:

- B2B (Business-to-Business). Ніша «Бізнес для бізнесу» має на увазі комерційні відносини між юридичними особами, економічними суб'єктами ринку. Тобто компанії, виробники взаємодіють між собою - укладають угоди, партнерські контракти на поставку, продаж, покупку товарів або послуг. Для

налагодження контактів, пошуку партнерів і переговорів в B2B використовуються спеціалізовані інтернет-майданчики, інтерактивні бази даних.

- B2C (Business-to-Consumer). Сфера «Бізнес для споживача» передбачає торгівлю товарами і послугами між юридичними і фізичними особами. Це свого роду роздрібні продажі, але тільки за допомогою онлайн-майданчиків - магазини, сервіси, банки та інше. Перевага клієнтів в більшому асортименті вибору, зручність замовлення та доставки товарів додому або в офіс. Електронна комерція дозволяє підприємцю знизити витрати на утримання торгових і складських площ.

- C2C (Consumer-to-Consumer). Електронна комерція в ніші «Споживач для споживача» має на увазі здійснення угод між фізичними особами. Успіх таких інтернет-майданчиків як OLX, Авіто, Ebay та інших заснований на комерційних відносинах користувачів через електронну систему оголошень.

- C2B (Consumer-to-Business). Дещо рідше зустрічаються в електронній комерції . Мається на увазі, коли споживач продає або вносить гроші компанії. Компанії, які використовують краудсорсинг або кампанію Kickstarter для фінансування свого бізнесу, потраплять під клеймо C2B.

- M-commerce (Мобільна торгівля). Як уже було зазначено є підмножинною електронної комерції та являє мобою купівлю-продаж товарів і послуг через кишенькові пристрої (смартфони).

Нижче наведено сім причин, чому електронна комерція є таким привабливим варіантом для підприємців. До переваг еклектронної комерції відносять:

1. Глобальне охоплення - бізне може продавати будь-кому в будь-якій точці світу за допомогою свого цифрового електронного комерційного бізнесу.

2. Незалежність від часу - фізичний бізнес зазвичай має обмежений час роботи, але інтернет-магазин електронної комерції залишається «відкритим» цілодобово, без вихідних, 365 днів на рік. Це надзвичайно зручно для клієнта та відмінна можливість для бізнесу.

3. Економія коштів - підприємства електронної комерції мають значно нижчі експлуатаційні витрати порівняно з фізичними магазинами. Немає орендної плати, немає персоналу, який можна наймати і платити, і дуже мало займають постійні операційні витрати. Це робить магазини електронної комерції надзвичайно конкурентоспроможними.

4. Автоматизоване управління запасами - набагато простіше автоматизувати управління запасами за допомогою електронних онлайн-інструментів та сторонніх постачальників. Це заощаджує підприємствам електронної комерції мільярди доларів на запасах та експлуатаційних витратах.

5. Цільовий маркетинг - Інтернет-продавці можуть збирати велику кількість споживчих даних, щоб забезпечити націлювання на потрібних людей для своєї продукції. Це знижує вартість залучення клієнтів і дозволяє електронному комерційному бізнесу в Інтернеті залишатися рухливим. Підприємство може залучати, для прикладу, лише чоловіків віком від 18 до 24 років, які мешкають у міських районах. Це вузькотаргетований маркетинг, що неможливий для фізичного магазину.

6. Нішеве домінування на ринку - завдяки меншим операційним витратам, здатності орієнтуватися на ідеального клієнта , а також охопленню глобальної аудиторії, яку приносить веб-сайт електронної комерції, це забезпечує прибутковість компанії.

7. Незалежність місця розташування - власник бізнесу електронної комерції не пов’язаний ні з одним місцем під час ведення свого бізнесу.

Недоліки електронної комерції:

1. Залежність від інформаційно-комунікаційних технологій. Не у всіх регіонах є вільний доступ до Інтернету на високій швидкості, цей фактор сильно гальмує розвиток електронного бізнесу.

2. Особливості законодавства, податки. Відсутність правового регулювання онлайн-комерції часто служить перешкодою при укладанні тих чи інших угод.

3. Безпека інформації. Онлайн-торгівля і бізнес в мережі вимагає високої гарантії конфіденційності даних користувачів, покупців, учасників комерційної діяльності. Активно впроваджується сертифікація, авторизація, капча і інші варіанти боротьби з шахрайством.

4. Авторське право. Захист прав власності - це давно не нова проблема для мережі Інтернет. Піратські копії програмного забезпечення, «злиті» у вільний доступ майстер-класи, книги і інша продукція інтелектуальної праці - все це стає проблемою для електронної комерції у всьому світі

Загалом існує два типи продавців електронної комерції:

- Ті, хто продає фізичні товари. Купівля та продаж фізичних товарів за допомогою якогось електронного носія. Сюди відносять товари з будь-якої з таких ніш: мода, аксесуари, товари для дому, іграшки тощо.
- Магазини, що продають цифрові товари (які можна завантажити). Як правило, якщо потрібно отримати доступ до продукту через Інтернет-мережу або якщо потрібно його завантажити, це - "цифровий продукт".

До організаційних форм електронної комерції загалом відносять: електронні системи платежів, електронні магазини, електронні аукціони, електронні біржі, надання банківських послуг через Інтернет.

Наведемо кілька ніш успішної реалізації E-commerce:

- Онлайн-сервіси прийому комунальних платежів, штрафів, реєстрацій та інше. Вже давно пішли в минуле багатометрові черги в банк для оплати комунальних послуг. Електронні сервіси доступні з будь-якого комп'ютера або гаджета. Досить підключити інтернет-банкінг і спокійно платити з дому за квартиру, оренду, кредит та інші послуги.
- Сайти-каталоги, дошки оголошень, агрегатори послуг і товарів. Популярність сайтів торгівлі між фізичними особами, компаніями зашкалює. Можливість бронювання квитків, номерів в готелях відкриває великі перспективи для розвитку цієї ніші.
- Сайти-розповсюджувачі фільмів, книг на платній основі, онлайн-кінотеатри і інше. Інтелектуальна власність стала також і цифровим продуктом,

тому результати праці письменників, музикантів, режисерів активно продаються в онлайн-просторі.

- Онлайн-магазини, інтернет-аукціони, інтернет-банкінг, реклама, маркетинг і багато інших сфер прекрасно себе почувають у віртуальному середовищі.

Починаючи з 1998 року електронна комерція плавно вийшла на оборот 3-3,5 трильйонів доларів за 20 років діяльності. Лідери E-commerce - Китай, США. Україна займає 67 місце в рейтингу, але має хороші шанси піднятися на кілька сходинок вище.

У 2020 році близько 13,6 мільйона українців принаймні один раз відвідали сайт електронної комерції , тоді як середній споживач витратив 334 долари на онлайн-покупки.

EVO.Business прогнозує, що розмір українського ринку електронної комерції в 2021 р. становитиме 137 млрд. грн. (4,89 млрд. Дол. США), що на 28% більше порівняно з 2020 р. У той же час, враховуючи, що вакцинація в Україні тільки розпочалася і триватиме протягом усього року, можуть бути додаткові карантинні обмеження. Оскільки 67% користувачів Інтернету все ще не роблять покупок в Інтернеті, є багато можливостей для індустрії електронної комерції піднятися ще вище.

1.2 Аналіз методів підвищення функціоналу електронної комерції

Ключовою проблематикою підприємств є утримання клієнтів, тому постає необхідність у аналізі методів для досягнення мети. Кожен з цих методів перевірений часом і найвідомішими онлайн-ритейлерами.

Адвокати бренду

Кожному бренду корисно мати при собі людей, які розповідатимуть про нього в суто позитивних тонах. Люди можуть обґрунтовано не довіряти рекламі, але не довіряти своєму родичу або другу складніше. Втім, навіть відгуки на «Фламп», iRecommended або Yelp здатні дати уявлення про продукт більше, ніж

хотілося б його засновникам. Залишається тільки навчитися відрізняти фальшиві відгуки від чесних.

Інтернет-магазин нового, ношеного одягу і аксесуарів від брендів класу люкс TheRealReal стимулює своїх покупців приводити друзів і робити повторні покупки. На це їх штовхає помилка особистої послідовності, одне з безлічі когнітивних спотворень: якщо покупець озвучить своє захоплення від роботи з вашим сервісом відкрито, то висока ймовірність, що він звернеться до вас за послугами знову.

Щоденні пропозиції

Магазин Hollar кожен день пропонує користувачам купити або щось унікальне, або за дисконтною ціною. Такий спосіб утримання знаходиться в межах системи цінностей для Hollar. Він інтригує користувачів - що ж буде завтра? - і змушує їх залишатися з магазином довше. Час дії пропозиції, природно, обмежена - то одним днем, то лише кількома годинами.

Програма лояльності

Цей спосіб пройшов еволюційний шлях довжиною в десятки років і може вважатися, мабуть, найефективнішим для залучення і збереження клієнтів. Сенс програм лояльності простий - дати клієнту вигоду від регулярного звернення до сервісу.

Але вони мають кілька форм втілення і сьогодні вже не можуть агресивно волати до споживчої жадібності. Єдина вимога, яким програми лояльності повинні відповідати завжди - це простота розуміння клієнтом.

Ось які програми лояльності набули поширення в електронній комерції:

• подарунки;

• бонуси;

• дисконти.

Дисконтна карта. Тут продавець товару або послуги переходить на новий рівень відносин з клієнтом: він отримує шанс дізнатися контактні дані клієнта і розвивати відносини, інформуючи про акції та знижки і радуючи унікальними пропозиціями. Плюс до того, якщо ваш бізнес - частина партнерської мережі, дію

дисконтної карти можна поширити на всіх її учасників. Компактніше пластикової дисконтної картки - тільки її віртуальний дублер всередині мобільного додатка.

У розділі їх програма «Карти» можна оформити віртуальну дисконтну карту або активувати вже існуючу. Це працює, коли клієнт забув пластиковий аналог будинку, втратив його або не може знайти серед інших карт.

Більш ефектний хід - розбудити в користувача азарт і давати бали за різні дії. Притому не тільки за покупку, але взагалі за будь-яку дію. Додаток інтернет-магазину китайських товарів Pandao дає бали за реєстрацію, використання промокодом, підключення профайлів в соціальних мережах, запрошення друзів і відгуки. Бали служать внутрішньою валютою Pandao, яку можна витрачати на покупку речей.

Гейміфікація

Є два типи гейміфікації: розробка повноцінних ігор як розділу продукту і впровадження ігрових механік. Яскравий приклад першого - це «Медуза» з їх тестами, платформер, аркадами. Вони з великим успіхом обіграють яскраві інформаційний приводи, що створює доброзичливу атмосферу для постійних читачів і залучає нових за рахунок віральності, не кажучи вже про продовження користувальницької сесії на сайті.

Mastercard запропонував таку розвагу своїм клієнтам з Данії: після кожної оплати карткою додаток Tap, Spin & Win дає розкрутити віртуальне колесо фортуни і виграти що-небудь приємне на кшталт квитків в кіно або сертифіката на покупку. У портфоліо «Лайв Тайпінг» теж є роботи по гейміфікація бренду. Під час зимової Олімпіади в Сочі користувачі «ВКонтакте» могли пограти в гру від імені драже Тіс Тас в ролі бобслеїста, ковзанярі або сноубордиста. Гравцеві потрібно було проїхати трасу, побудовану на основі звучання якого-небудь треку з аудіозаписів, і збирати бали.

Про впровадження ігрових механік ми розповіли вище. Pandao винагороджує користувачів за кожну дію, яке вони роблять в додатку, підживлюючи залишатися в додатку як мінімум з спортивного інтересу. Інший

приклад - Swarm, сайд-додаток Foursquare, що дає користувачам бонуси за чек-іни в різних місцях, що перетворюються в знижки.

Преміум-статус

Власники такого статусу отримують привілеї і, за деякими оцінками, проводять на сервісі в два рази більше часу, ніж звичайні.

Наприклад, у клієнтів Amazon є опція оформити річну передплату на пакет послуг Amazon Prime і отримати доставку терміном на два дні і безкоштовний доступ до потокового відео і музики. З пакетом Prime Now все крутіше - доставка займає дві години, відео, музика і книжки для Kindle. За показниками на 2019 рік, 57% прибутку Amazon в Північній Америці приносять саме «преміали». Американський магазин низьких цін Thrive Market побудував на преміум-акаунтів всю бізнес-модель. Особливі користувачі отримують доступ до знижок на і без того дешеві продукти. Факт того, що вони заплатили вперед, стимулює їх замовляти більше, щоб насолоджуватися отриманої вигодою.

Останній в цьому списку, але не менш важливий для сучасної людини - китайський інтернет-магазин AliExpress. Кожен його відвідувач може вступити в спеціальний клуб, члени якого отримують бали за покупки і присвоюється статус. На кожному ступені користувачеві відкриваються нові можливості - від інформування про знижки до вирішення конфліктних ситуацій і повернення грошей в лічені дні.

Серед інших привілеїв може бути можливість повернути товар, брати участь в закритих вечірках, робити передзамовлення анонсованих товарів і отримувати знижки та подарунки до святкових дат.

Персональний досвід покупки

Опитування показують, що 82% покупців витрачали б більше, якби пропозиції будувалися з оглядкою на їх індивідуальність і цінності. За цим стоїть величезна робота зі збору даних, де належить дізнатися звички ваших покупців, на сторінках з якими товарами вони бувають, який контент їм подобається, через який пристрій вони користуються сайтом або мобільним додатком.

Обробка даних допоможе прискорити другу хвилю продаж, якщо ви розглянете загальний поведінковий патерн у кожного клієнта, який зробив покупку вперше, а саме, чому і коли люди повертаються на ваш сайт або в додаток з наміром купити.

Тут важливо не тільки знати, що робити, але і чого не робити. Наприклад, якщо ви зрозуміли, що переважно повернення відбувається через 30 днів, не заманюють клієнтів знижками в цьому відрізку часу в спробі його повернути - адже він, швидше за все, повернеться сам. А ось якщо немає, то починайте вводити в гру спеціальні пропозиції. Може допомогти прогресивне нарощування знижки: 10% на 31 день, 30% на 45 день і 50% на 60 день.

Push-повідомлення

У той час як навіть файлообмінники пропонують підписатися на повідомлення, серйозні сайти і додатки без цієї опції вже неможливо уявити. Які вигоди вони дають в сфері електронної комерції:

• Підвищують залучення на 85%.

• Повертають 65% користувачів, якщо повідомлення включені.

• Чи вважаються корисними для половини користувачів. Але що корисно для одних, марно для інших, так що сегментує користувачів і шліть їм ті push, які відповідають їх поведінки.

У push-повідомленнях можна розповідати:

• Про нові колекціх одягу, аксесуарів і косметики і т.д.

• Про розпродажі, спеціальні пропозиції та знижки.

• Про вигоди і користі, особливо якщо людина вже на сайті або в додатку, але не прагне нічого купувати і його потрібно надихнути.

• Персоналізовані пропозиції в контексті тій локації, в якій знаходиться користувач. Так робить Booking, якщо ви перебуваєте поруч з яким-небудь хостелом.

Повернення (retention)

Клієнтам, які давно у вас не були, пішли або збираються це зробити, потрібно в якийсь момент нагадати про себе: підібравши правильну інтонацію,

зателефонувати, написати лист, відправити push-повідомлення або SMS. Але попередньо непогано б з'ясувати причину.

Хороший приклад проведення ретаргетингової кампанії подає сервіс з доставки їжі Instacart. Переписка починається з пропозиції компанії завантажити додаток і спробувати безкоштовну доставку протягом п'яти днів, застосовуючи популярний прийом створення штучного дефіциту.Починаючи з другого листа, Instacart описує переваги роботи з сервісом. На шостому листі, коли ліміт в п'ять днів вичерпаний, він все переграє на очах у клієнта і продовжує термін безкоштовної доставки, додаючи ще сім днів. І, нарешті, в дев'ятому листі Instacart, крім безкоштовної доставки, обіцяє знижку в $5 на перше замовлення.

Допомога користувачам

Клієнтоорієнтованість взуттєвого рітейлера Zappos стала їх гордістю. Їх корпоративна система цінностей ставить на чільне місце сімейний дух як всередині компанії, так і у взаємодії з покупцями. Вони посилали своїм клієнтам квіти, відправляли представників в магазин конкурентів за спеціальним взуттям, коли та закінчилася на складі Zappos. А в 2011 році підкорили одного зі своїх клієнтів, чию взуття поштова служба доставила не туди: компанія дала йому VIP-статус і безкоштовні черевики. Нехай Zappos будуть для вас прикладом, адже 70% продажів генерують їхні постійні клієнти. Це дуже і дуже багато, як ви розумієте.

Рекомендаційна система

За останнє десятиліття кількість інформації зросла настільки, що для більшості людей вже не представляється можливим розділити хороші, на суб'єктивний погляд, ігри, музику, фільми і серіали і нехороші самостійно.

Цю проблему вирішують рекомендаційні системи - алгоритми, що враховують переваги та дії кожного користувача. На основі цього рекомендаційні сервіси пропонують користувачам те, що їм може сподобатися. Apple Music підказує артистів і альбоми, Amazon, Ozon і «Яндекс.Маркет» - товари, YouTube - кліпи і влогеров, Relap.io - контент, а Netflix - серіали.

Для створення рекомендаційних систем використовуються різні моделі, такі, як коллаборативна фільтрація, призначена для користувача коллаборативна фільтрація, експертна модель і гібридна модель.

Якщо брати Netflix, то з 2006 року по 2009 рік його рекомендаційна система Cinematch працювала по гібридної моделі Pragmatic Chaos, організованою групою BellKor під час конкурсу Netflix Prize. Результатом стало підвищення точності релевантних рекомендацій на 10,06%, що здається незначним тільки на перший погляд. Що рекомендаційна система дала Netflix? Глядачі стали дивитися більше. Це призвело до зниження відтоку, зростання LTV, коефіцієнту утримання сервісу і, відповідно, його доходу.

Віртуальна і доповнена реальність

Різниця між віртуальною і доповненої реальністю в тому, що віртуальна повністю підміняє навколишню дійсність симуляцією (наприклад, за допомогою окулярів Oculus Rift, HoloLens або PlayStation VR), в той час як доповнена реальність привносить в цю дійсність згенеровані компоненти (покемонів в Pokemon Go або татуювання в Inkhunter).

Перший серйозний крок mainstream віртуальна реальність зробила тоді, коли в 2014 році Facebook купив компанію Oculus VR. Через два роки більше половини споживачів, які пройшли опитування компанії Greenlight Insights, погоджувалися з тим, що довіряють і хотіли б мати справу з брендами, що підтримують віртуальну реальність. І бренди йдуть їм назустріч.

У 2016 році eBay і австралійський ритейлер Myer першими в світі відкрили двері віртуального магазину. Все, що потрібно споживачеві, це телефон с будь-який з двох операційних систем, VR-окуляри і додаток eBay Virtual Reality Department Store. Купівля оформляється довгим поглядом на товар.

IKEA випустила додаток IKEA Place. З його допомогою можна розставити меблі з каталогу магазину по квартирі і оцінити, чи варто покупка того. Додаток доступний на обох мобільних платформах, але кількість підтримуваних пристроїв обмежена.

Доповнена реальність в електронній торгівлі нерідко виявляє себе за принципом телемагазину: користувачеві пропонують приміряти одяг, окуляри і прикраси прямо вдома - за допомогою телефону або планшета.У 2013 році рітейлер glasses.com, що продає оправи та лінзи, використовував AR для того, щоб підбирати для своїх покупців максимально підходящі окуляри. За допомогою однойменного додатка користувач вибирає тип очок (жіночі, чоловічі, звичайні, сонцезахисні) і створює 3D-модель своєї голови, повертаючи її вліво-вправо перед фотокамерою. Потім додаток накладає на обличчя обраний тип окуляр, і користувач може оцінити результат з різних ракурсів. Модель зберігається в акаунті користувача - тепер рекомендації сайту спираються на неї.

Контент

Кілька останніх років навколо контент-маркетингу йде справедливий ажіотаж, адже це дуже сильний інструмент, особливо якщо у вашого продукту є конкуренти.

Вам важливо виробити те, що називають "tone of voice" - прийоми спілкування, за допомогою яких ви будете розповідати аудиторії про свій товар або послугу, доносити свої цінності, ділитися новинами індустрії, і все це - на зрозумілій для аудиторії мовою і в зрозумілому вигляді . Коштів багато: хтось заводить блог, хтось веде соцмережі, хтось знімає відео і пише подкасти, а кому-то вистачає медійної реклами.

У своєму дослідженні 2000 року співробітники консалтингової фірми Bain & Company Фредерік Ф. Райхельд і Філ Шефтер відзначають, що при підвищенні кількості утриманих клієнтів на 5% прибуток може вирости від 25% до 95%. Не дивлячись на давність дослідження, на нього продовжують посилатися. Також утримання підвищує LTV користувача, відкриває нові канали для продажів і підвищує середній чек замовлення.

Таким чином імплементувавши рекомендаційні системи, бізнес отримує безліч можливостей для росту. Такі моделі і методи можуть бути застосовані в електронній комерції при створені інтернет-магазинів, інтернет-аукціонів, веб-порталів, електронних дошок оголошень та інформаційно-пошукових системах.

1.3 Узагальнення проблематики підвищення функціоналу електронної комерції за допомогою рекомендаційних систем

Рекомендаційні системи - це великий клас моделей, мета яких підвищити бізнес-показники за рахунок релевантних рекомендацій користувачеві в правильному місці, в правильний час і через правильний канал комунікації.

Щодня мільйони людей зайняті пошуком в інтернеті: хтось шукає фільми або одяг, хтось - машину або турпутівку, - і всі користувачі об'єднані однією метою: знайти те, що потрібно саме їм. Якщо в минулому столітті про появу нових товарів дізнавалися з поштових розсилок, то потім цей процес прискорився в десятки (а то і сотні) разів за рахунок появи спочатку телебачення, а потім і Інтернету.

А в останні рік-два одним з провідних трендів в удосконаленні найрізноманітніших пошукових систем стає застосування алгоритмів машинного навчання. Як доповнення до процесу самостійного пошуку (серед мільйонів найменувань всіляких товарів і послуг), на ресурсах стали з'являтися рекомендаційні системи, які пророкують що саме було б цікаво даному користувачеві. Такі рекомендаційні алгоритми в процесі своєї роботи постійно донавчаються, адаптуються і трансформуються, з часом все краще розуміючи користувача, і в результаті свого функціонування 50% і більше рекомендованих товарів або послуг в тій чи іншій мірі задовольняють пошуковим запитам користувачів.

Рекомендаційні системи - це комплекси алгоритмів, програми та сервіси, основне завдання яких передбачити, які об'єкти (товари або послуги) будуть цікаві користувачеві, маючи інформацію про його профілі або інші дані.

Існує 4 типи рекомендаційних систем:

1. Коллаборативна фільтрація (Collaborative Filtering) - рекомендації засновані на історії оцінок як самого користувача, так і інших користувачів. Цей підхід має теоретично високу точність, але при цьому має одну важливу проблему - високий поріг входу.

2. Засновані на контенті (content-based) - рекомендації засновані на даних, зібраних про кожен конкретний товар. Користувачеві рекомендуються об'єкти, схожі на ті, якими він раніше цікавився або вже купував. Схожість оцінюється виходячи зі вмісту об'єктів. Великий плюс - можливість зацікавити нового користувача пропозиціями з перших споживчих кроків. Для цього не потрібно довго збирати дані про переваги, а можна відразу включити споживача в роботу з ресурсом. Можливо рекомендувати навіть ті об'єкти, які не отримали оцінку інших користувачів. Основні недоліки - сильна залежність від предметної області, зниження точності і обмеженість корисності рекомендацій.

3. Засновані на знаннях (knowledge-based) - рекомендації засновані на знаннях про предметну область (а не про кожен товар). Такий тип рекомендацій має високу точність, пропонуючи користувачеві те, що йому потрібно. Крім цього, система вивчає і аналізує взаємозв'язки між об'єктами, враховує ряд додаткових опцій, що відносяться до індивідуальних властивостей конкретного користувача. До таких властивостей відносяться призначені для користувача побажання (наприклад, їх використовує Яндекс.Маркет) і демографічні особливості (вихідні дані, які використовують найбільші соціальні мережі, такі як Facebook, LinkedIn та інші). Основний мінус - складність розробки та збору даних.

4. Гібридні (hybrid) - рекомендації засновані на комбінуванні коллаборативного і контентного підходів, що дозволяє уникнути більшості недоліків, властивих кожній системі.

У гібридних рекомендаційних системах зустрічаються такі типи комбінування:

- реалізація окремо коллаборативного і контентного алгоритмів і об'єднання їх припущень;

- включення деяких контентних правил в коллаборативну методику;

- включення деяких коллаборативних правил в контентну методику;

- побудова загальної моделі, що включає в себе правила обох методик.

Основний недолік - складність розробки.

Завдання рекомендаційних систем прості і зрозумілі - застосовуються для пропозиції клієнтові тих продуктів або послуг, які з високим ступенем ймовірності його зацікавлять.

Рекомендаційні системи працюють на двох "рівнях":

1 рівень - це глобальні оцінки; особливості та переваги, які змінюються дуже повільно; цікаві сторінки; залежність від характерних для користувача рис, таких як стать, місце проживання і т.д .;

2 рівень - короткочасні тренди і швидкі зміни інтересу в часі.

Для складання якісних рекомендацій використовується явний або неявний збір даних:

- при явному зборі від користувача необхідно отримати заповнені анкети для виявлення переваг. Недолік методу в тому, що досить складно змусити користувача поставити оцінку.
- при неявному зборі протоколюються дії користувача: що користувач подивився, який товар доданий в кошик, що прокоментував, яку покупку зробив. Складання рейтингів відбувається автоматично. Недолік методу - невизначеність: якщо користувач подивився товар, не відомо, чи сподобався він йому чи ні; якщо користувач не купив товар - то знову ж таки невідомо, чим було зумовлене таке рішення.

Також можливий варіант комбінування двох підходів: якщо немає транзакционной історії - використовуються опитування, коли ж вона з'являється - починають враховувати і транзакції.

Сфери застосування рекомендаційних систем різноманітні: пошук фільмів, музики, наукових статей, роздрібна торгівля, соціальні мережі, електронна комерція, онлайн банкінг і т.д.

Мабуть, одним з найбільш широковідомих прикладів впровадження і використання рекомендаційних систем є компанія Netflix - постачальник відеоконтенту на умовах оренди і у вигляді потокового сервісу.

Компанія починала з того, що розсилала клієнтам по підписці VHS-касети і DVD. Користувач дивився і відправляв диски назад, отримував наступні. Для

Netflix було важливо підвищити якість рекомендацій. Чим краще Netflix рекомендує користувачам фільми, тим більше фільмів беруть в прокат. Відповідно, зростає і прибуток компанії.

У 2006 році компанія Netflix оголосила конкурс на вдосконалення своєї рекомендаційної системи під назвою Cinematch. В основу було закладено принцип коллаборативної фільтрації. Рекомендації формувалися з урахуванням як оцінок користувача, так і оцінок інших глядачів - для цього система підбирала користувачів зі схожими уподобаннями, чиї оцінки близькі до їх власних. На підставі цього глядачеві автоматично давалася рекомендація: подивитися той чи інший фільм. Власний алгоритм Netflix передбачав оцінки користувачів з якістю 0.9514 за метрикою RMSE. Завдання було поліпшити прогноз хоча б на 10% - до 0.8563. Переможцю обіцяли приз в $1 000 000.

Netflix виклав у відкритий доступ зібрані дані: близько 100 мільйонів оцінок за п'ятибальною шкалою з зазначенням ID користувачів, які поставили оцінку. Учасники змагання повинні були якомога точніше передбачати, яку оцінку поставить конкретному фільму той чи інший користувач.

Змагання тривало три роки. За перший рік якість поліпшили на 7%, далі все трохи сповільнилося. Проміжні номінації вручалися щороку до тих пір, поки дві команди з невеликою різницею в часі надіслали рішення, кожне з яких проходило поріг в 10%. Перший приз дістався компанії BellKor's Pragmatic Chaos, групі вчених з AT & T, яким вдалося домогтися поліпшення точності рекомендацій на 10,06%.

На даний момент система Netflix (BellKor), що є комбінацією 27 рекомендаційних алгоритмів, вважається найбільш технологічною в світі.

Висновки, які можна зробити. Можливість збору даних, спростила і одночасно ускладнила передбачення щодо поведінки і уподобань користувачів. Особливої уваги потребує і дотримання конфіденційності в роботі рекомендаційних алгоритмів, адже часто вони можуть спрогнозувати такі результати або виявити такі закономірності, про які користувач навіть і не підозрював, або ж не хотів, щоб про це стало відомо. Хороша рекомендаційна

система повинна справлятися не тільки з цим, але і з проявами нечесної конкуренції, вираженими в навмисному піднятті рейтингів одних товарів і заниженні у конкуруючих, наприклад, за допомогою негативних відгуків і коментарів.

Однак існує поширений стереотип, який до цих пір заважає широкому застосуванню в бізнесі рекомендаційних систем. Багатьом здається, що в реальності впровадження рекомендаційних алгоритмів - це занадто складно і вимагає глобальної перебудови всього процесу збору і обробки даних, а також зміни бізнес-процесів, логістики і так далі. Багато хто сумнівається і не можуть оцінити, який же ROI (повернення від інвестицій) в подібні трансформації. Ці сумніви абсолютно необгрунтовані, адже насправді рекомендаційні системи можуть бути корисні практично кожному бізнесу, а щоб почати рекомендувати, часто цілком достатньо тих даних, які вже збираються.

Релевантні рекомендації скорочують час, необхідний для пошуку товарів і послуг, і значно збільшують вірогідність попадання в поле зору користувача інших об'єктів, які зможуть його зацікавити. В результаті підвищується лояльність і задоволеність користувачів веб-сервісами. Як правило, користувачі також взаємодіють з більшою кількістю товарів, і це призводить до збільшення споживання і зростання прибутку. Крім того, інформаційні бюлетені, персоналізовані рекламні матеріали і push-повідомлення спонукають користувачів повертатися, збільшують частоту відвідувань постійними користувачами і зменшують відтік клієнтів.

Сьогодні кожній компанії просто необхідно налагодити процес збору даних і вміти грамотно і ефективно використовувати їх в бізнесі, тим самим оптимізуючи і покращуючи призначений для користувача контент, знижуючи витрати, збільшуючи виручку і середній чек, і підвищуючи рентабельність бізнесу в цілому.

Формалізуємо нашу задачу. Нехай у нас є безліч користувачів $U = \{u_1, u_2, \ldots, u_n\}$, безліч об'єктів $P = \{p_1, p_2, \ldots, p_m\}$ і матриця рейтингів

$R = (r_{i,j})$розміру $n \times m$, де $i \in 1 \ldots n,\ j \in 1 \ldots m.$. Можливими значеннями рейтингів можуть бути числа від 1 (зовсім не по-подобалося) до 5 (дуже сподобалося), або значеннями сподобалося / не сподобалося, виражені 0 і 1, а також інші. якщо користувач *i* ніяк не оцінив елемент *j,* то на відповідному місці $r_{i,j}$ буде стояти порожнє значення.

Описану матрицю рейтингів можна представити у вигляді таблиці, показаної на рисунку.

Таблиця 1.1 - Матриця рейтингів

№ Продукту	№ 1	№ 3	№ 3	№ 4	№ 5	№ 6
Користувач 1	2	4	?	3	4	?
Користувач 2	1	?	1	4	5	?
Користувач 3	3	4	?	1	4	?
Користувач 4	?	?	4	2	?	2
Користувач 5	?	4	5	3	?	?

Позначимо через $\hat{r}_{i,j}$ j наш прогноз щодо того, яку оцінку користувач *i* поставить продукту *j*. Наша задача найкращим чином передбачити, які оцінки $r_{i,j}$ повинні стояти на місці пропусків в матриці рейтингів, тобто розрахувати $r_{i,j}$. Потім для кожного користувача *u* на основі прогнозованих оцінок $r_{i,j}$, нам потрібно сформувати список з *N* продуктів, які найбільш точно задовольняють потреби користувача, і, які ще не були ним оцінені. Список цих *N* продуктів позначимо через *N*- мірний вектор $(p_{i_1}, p_{i_2}, \ldots, p_{i_N})$. Таким чином, математична задача звучить так:

Дано:

$U = \{u_1, u_2, \ldots, u_n\}$ - множина користувачів,

$P = \{p_1, p_2, \ldots, p_m\}$ - множина продуктів,

$R = (r_{i,j})$ - матриця рейтингів розміру $n \times m$, де на місці $r_{i,j}$ буде стояти певна кількість, якщо користувач u_i оцінив продукт p_j і пусто в іншому випадку.

N - необхідне число рекомендацій, які хочемо отримати від системи.

Потрібно знайти:

Для даного користувача u_i знайти *N*- мірний вектор $(p_{i_1}, p_{i_2}, \ldots, p_{i_N})$, де продукти $p_{i_k}, k \in N$ ще не оцінені цим користувачем, тобто в матриці рейтингів $R = (r_{i,j})$ варто порожньо на місці r_{i,i_k}, а також, щобці продукти найбільш точно задовольняли уподобанням користувача, тобто прогнозні рейтинги r_{i,i_k} були найбільшими.

Алгоритми, які це роблять, можуть бути дуже різними, і використовувати різні вхідні дані. Одні з них формують рекомендації, грунтуючись тільки на даних по відомих рейтингах. Інші використовують додаткові характеристики продуктів, на основі рейтингів визначають, які з цих характеристик найбільш точно задовільняють вподобання користувача, а потім підбирають продукти з такими характеристиками.

А тепер більш детально поговоримо про кожен з цих алгоритмів, проблеми, з якими стикається дослідник в процесі їх використання, переваги та недоліки кожного методу.

1. Системи колаборативної фільтрації

Основна ідея таких систем полягає в тому, що якщо користувачі мали однакові інтереси в минулому, то в майбутньому їх уподобання також будуть збігатися. Як приклад розглянемо книжковий інтернет-магазин. Нехай в минулому історії покупок користувачів А і Б в даному магазині сильно перетиналися. При появі користувача А на сайті ми хочемо запропонувати йому нову книгу, яку він ще не читав, а користувач Б, якраз недавно придбав книгу,

яку А ще не бачив. У такій ситуації буде розумно запропонувати користувачеві А прочитати її.

Вище описаний принцип роботи класу алгоритмів коллаборативної фільтрації, які в англомовній літературі називаються user- based, тобто засновані на статистиці про користувачів. Як видно з прикладу, ми порівнюємо схожість між користувачами, спираючись на рейтинги оцінених об'єктів. У той же час, чому б нам не використовувати ту ж статистику для порівняння продуктів між собою, а потім зіставити результати двох підходів.

Такий метод, заснований на порівняння схожості об'єктів, називається item-based. Тут основна ідея полягає в тому, що якщо користувачам, які оцінили два продукти, сподобалися обидва, то користувачам, які спробували тільки один, можна пропонувати другий, який найімовірніше, їм сподобається. Тобто, якщо в прикладі з книгарнею ми помітили, що користувачі, які купували книгу А, також купували книгу Б, то тим споживачам, які вже купили А, але ще не звернули увагу на Б розумно буде її запропонувати.

Першою дослідною роботою по рекомендаційних системаз вважається робота L. Giles "An autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications" 1998 року. Однак першою друкованою правильніше назвати роботу 1992 року D. Goldberg, D. Nichols "Using collaborative filtering to weave an information Tapestry"

У даній роботі описаний принцип роботи експериментальної поштової системи Taperstry. Розробники Tapestry першими використовували термін "коллаборативна фільтрація" як метод збору якісних даних. Дана система була розроблена в Xerox PARC як спосіб обробки великої кількості повідомлень електронної пошти та повідомлень, що відправляються в групи новин. Особливістю даної системи було те, що система збирала і аналізувала дані про реакцію людей на прочитані ними документи, в наслідок чого процес фільтрації став більш ефективним.

Одночасно з Taperstry розвивалися і інші рекомендаційні системи на основі коллаборативної фільтрації:

• У 1995-1996 роках були розроблена відразу три системи для рекомендації музики: Helpful Online Music Recommendations, Ringo, Firefly.

• Також в той час активно розвивалися системи рекомендацій най-цікавіших і популярних сторінок в інтернеті: Point's Top 5%, PHOAKS (People Helping One Another Know Stuff), Webdoggie, Alexa Internet.

Метод Item-based був винайдений і використаний Amazon.com в 1998 році. Вперше представлений публіці на науковій конференції в 2001, а його автори в 2016 отримали нагороду Test of Time.

Даний алгоритм допоміг впоратися з деякими з проблем, що мали методи, засновані на схожості користувачів:

• системи працювали погано, коли у них було багато продуктів, але порівняно небагато оцінок

• занадто багато роботи обчислити подібності між усіма парами користувачів

• профілі користувачів швидко змінювалися, і всю модель необхідно було перераховувати

Але питань, з якими доводиться мати справу розробнику в процесі створення такої системи залишається ще багато. Ось одні з них:

1) Як для даного користувача, для якого ми хочемо зробити рекомендацію, визначити користувачів, які мають схожі інтереси?

2) Як вимірювати схожість між користувачами?

3) Що робити якщо у нас є мало даних про рейтинги?

Переваги методу:

1) Є достатньо універсальним підходом, тому часто дає високі результати.

2) Для роботи даного методу не потрібна детальна інформація про продукти. У прикладі з книгарнею - автор, жанр, описання книги. Замість цього

використовується як історія оцінок самого користувача, так і інших користувачів.

Недоліки методу:

1) Як працювати з новими користувачами, для яких ще немає історії покупок (завдання холодного старту).

2) Що робити з новими об'єктами, які ще ніхто не оцінив.

3) Ресурсоємкість обчислень, яка уповільнює час роботи системи.

4) Необхідний великий обсяг даних для високої точності передбачень.

2. Системи фільтрації на основі вмісту

Цей тип систем заснований на наявності інформації про опис і профілі, що складається з набору характеристик елемента. Якщо знову розглянути приклад книгарні, то в якості характеристик можна взяти жанр, тему або автора книги. Потім для кожного користувача створюється профіль шляхом присвоєння характеристик подібних до характеристик елементів, виходячи з аналізу його поведінки в минулому, або явно запитуючи про його вподобання. Далі користувачеві рекомендуються об'єкти, схожі на ті, які цей користувач уже купував, або вказав як кращі. Схожості оцінюються за признакам вмісту об'єктів.

У прикладі з книгарнею система могла визначити, що автору подобаються детективи і новели певних авторів, і як наслідок рекомендувати книги цих жанрів або авторів.

У процесі вивчення систем фільтрації на основі вмісту також виникають цікаві питання:

1) Як система може автоматично створити профіль користувача і потім покращувати в процесі оновлення даних?

2) Як визначити який елемент відповідає перевагам користувача?

3) Як автоматично отримувати інформацію про продукт, щоб запобігти ручному заповненню?

Переваги методу:

1) Не вимагає великої групи користувачів для досягнення високої точності рекомендацій.

2) Нові елементи можна рекомендувати відразу, як тільки у них з’являються заповнені характеристики.

Недоліки методу:

1) Сильна залежність від предметної області, корисність рекомендацій обмежена.

2) Профіль користувачів і елементів повинен складатися з одинакового набору характеристик, щоб їх можна було порівнювати.

3. Системи, засновані на знаннях

Рекомендації, засновані на знаннях, використовуються зазвичай в таких областях, як електроніка, де покупці роблять покупки раз в пару років, так як в даній області ми не можемо покластися на історію покупок, яка використовується в якості вхідних даних для методів коллаборативної фільтрації і методів, заснованих на утриманні.

Розглянемо для прикладу рекомендаційну систему, яка допомагає користувача вибрати фотокамеру. Звичайний користувач купує нову камеру тільки один раз в декілька років. Таким чином, рекомендаційна система не може побудувати профіль користувача або запропонувати камери, які сподобалися іншим користувачам, так як в іншому випадку пропонуватися будуть тільки бестселери. Тому алгоритми, засновані на знаннях, зазвичай використовують додатково тількі дані, як про користувачів, так і про сам продукт, для формування списку рекомендацій. В області фотокамер така система може використовувати детальну інформацію про характеристики камер, таких як роздільна здатність, вага, ціна.

Просто представляти продукт, що задовольняє обраному користувачем набору характеристик, є недостатнім, оскільки в такому випадку кожен користувач отримує однакові рекомендації з тими, хто вибрав такий же набір характеристик. Таким чином, дані системи повинні не просто збирати

інформацію про бажані характеристики, а також формувати певний профіль користувача.

Тому важливим аспектом побудови таких систем є налаштування взаємодії між користувачем і системою. Якщо згадати приклад з книгарнею і алгоритмом коллаборативної фільтрації, то можна помітити, що користувач може взаємодіяти з програмою обмеженим числом способів. Безліч застосунків допускає тільки можливість ставити рейтинги від 1 до 5. Повертаючись до прикладу з фотокамерою, коли у нас немає інформації про історію покупок користувача, нам необхідно налаштувати діалог між користувачем і системою, в процесі якого програма задасть питання про вимоги покупця, таких як максимальна ціна, мінімальна роздільна здатність і т.д.

Такий підхід вимагає не тільки детального технічного розуміння характеристик продукту, але також будує приблизний сценарій на основі обраних характеристик. У такій ситуації обмежуючі фактори можуть бути використані для опису контексту, в якому певні характеристики є релевантними для покупця. Наприклад, камера високої роздільної здатності є кращою, якщо користувач планує друкувати фотографії великого розміру.

В цілому при розгляді систем такого виду виникає досить багато питань:

1) У яких областях може бути застосовний даний метод?

2) Як отримати профіль користувача в областях, де немає історії його покупок, і як врахувати переваги користувача?

3) Як налаштувати взаємодію з користувачами?

4) Яким чином можна персоніфікувати процес взаємодії, щоб максимізувати точність процесу збору інформції про вподобання користувачів?

Переваги методу:

1) Вимоги користувачів можуть бути визначені точніше, завдяки явній взаємодії.

2) Метод дає хороші результати в сфері, де немає достатньої інформації про історію покупок.

Недоліки методу:

1) Від користувача потрібні додаткові дії, щоб система могла зібрати дані про його вподобання.

2) Дані про вимоги користувача можуть бути неправильно інтерпретовані системою.

4. Гібридні системи

Кожен з вищеописаних методів має свої переваги і недоліки в залежності від поставленого завдання. Досить очевидним рішенням всіх цих проблем є об'єднання різних підходів для того, щоб забезпечити більшу точність рекомендацій..

Якщо, наприклад, у нас є дані про опис продуктів, профіль користувачів і історія його покупок, то ми можемо поліпшити рекомендаційну систему шляхом об'єднання методів коллаборативної фільтрації і алгоритмів фільтрації за вмістом. Таким чином, в разі появи нового користувача в системі, про який немає історії покупок, ми зможемо використовувати рекомендації на основі алгоритмів фільтрації за вмістом, а в разі великого обсягу статистичних даних будувати більш точний прогноз, використовуючи методи коллаборативної фільтрації.

Незважаючи на те, що гібридні системи допомагають боротися з недоліками описаних раніше методів, вони все ж залишають достатньо питань, на які потрібно відповісти при проектуванні такої системи:

1) Які підходи можуть бути об'єднані, і які умови повинні виконуватися, щоб це могло бути зроблено?

2) Чи повинні різні техніки використовуватися в різних ситуаціях, або результат кожної повинен братися з певною вагою?

3) Як результати різних методів повинні бути зважені, щоб на виході отримати один результат?

РОЗДІЛ 2
МАТЕМАТИЧНЕ ОБГРУНТУВАННЯ ЗАДАЧ КЛАСИФІКАЦІЇ ТА МЕТОДОЛОГІЧНІ ОСНОВИ ЕМОЦІЙНОГО АНАЛІЗУ

2.1 Математичний апарат та алгоритмічні засади задач класифікації як базису емоційного аналізу тексту

Визначення задач класифікації

Класифікація - це форма аналізу даних, яка виділяє моделі, що описують важливі класи даних. Такі моделі, які називаються класифікаторами, передбачають категоріальні (дискретні, невпорядковані) мітки класів. Наприклад, ми можемо побудувати модель класифікації, щоб класифікувати заявки на банківські позики як безпечні, так і ризиковані. Такий аналіз може допомогти нам краще зрозуміти ці дані. Багато методів класифікації було запропоновано дослідниками в галузі машинного навчання, розпізнавання образів та статистики. Більшість алгоритмів є резидентами пам'яті, як правило, припускаючи невеликий обсяг даних. Нещодавні дослідження з видобутку даних базуються на такій роботі, розробляючи масштабовані методи класифікації та прогнозування, здатні обробляти великі обсяги даних, розміщених на диску. Класифікація має численні програми, включаючи виявлення шахрайства, цільовий маркетинг, прогнозування ефективності, виробництво та медичну діагностику.

Наприклад, співробітник банківських позик потребує аналізу її даних, щоб дізнатись, які претенденти на позику є "безпечними", а які "ризикованими" для банку. Медичний дослідник хоче проаналізувати дані про рак молочної залози, щоб передбачити, яке з трьох конкретних методів лікування пацієнт повинен отримати. У кожному з цих прикладів завданням аналізу даних є класифікація, де модель або класифікатор побудовані для прогнозування міток класів (категоріальних), таких як "безпечний" або "ризикований" для даних заявки на позику; “Так” чи “ні” для маркетингових даних; або "лікування А", "лікування В" або "лікування С" для медичних даних. Ці категорії можуть бути представлені

дискретними значеннями, де впорядкування між значеннями не має значення. Наприклад, значення 1, 2 та 3 можуть бути використані для представлення методів лікування A, B та C, де серед цієї групи режимів лікування не передбачається впорядкування.

Або припустимо, що менеджер з маркетингу хоче передбачити, скільки витратить той чи інший клієнт під час шопінгу. Це завдання аналізу даних є прикладом числового передбачення, де побудована модель передбачає безперервну функцію або впорядковане значення, на відміну від мітки класу. Ця модель є предиктором. Класифікація та числове передбачення - два основних типи проблем прогнозування.

Наступне питання - "Як саме працює класифікація?" Класифікація даних - це двоступеневий процес, що складається з етапу навчання (де побудована модель класифікації) та етапу класифікації (де модель використовується для прогнозування міток класів для даних даних). Процес показаний на прикладі даних заявки на позику на Рис. 2.1 (дані спрощені для ілюстративних цілей. Насправді ми можемо очікувати, що буде розглянуто набагато більше атрибутів).

На першому кроці будується класифікатор, що описує заздалегідь визначений набір класів даних або понять. Це крок навчання (або фаза навчання), де алгоритм класифікації створює класифікатор, аналізуючи або “навчаючись на” навчальному наборі, що складається з кортежів бази даних та пов’язаних з ними міток класів. Кортеж X представлений n-мірним вектором атрибутів ***X*** D .x_1, x_2, : : :, x_n/,, що відображає n вимірювань, виконаних на кортежі з n атрибутів бази даних, відповідно, A_1, A_2, : : :, A_n.[1] Кожен кортеж, X, передбачається, що належить до визначеного класу, що визначається іншим атрибутом бази даних, який називається атрибутом мітки класу. Атрибут мітки класу має дискретне значення та не упорядковується. Категоричним (або номінальним) є те, що кожне значення служить категорією або класом. Окремі кортежі, що складають навчальний набір, називаються навчальними кортежами і випадковим чином відбираються з бази даних, що аналізується. У контексті класифікації набори

даних можна називати зразками, прикладами, екземплярами, точками даних або об'єктами.

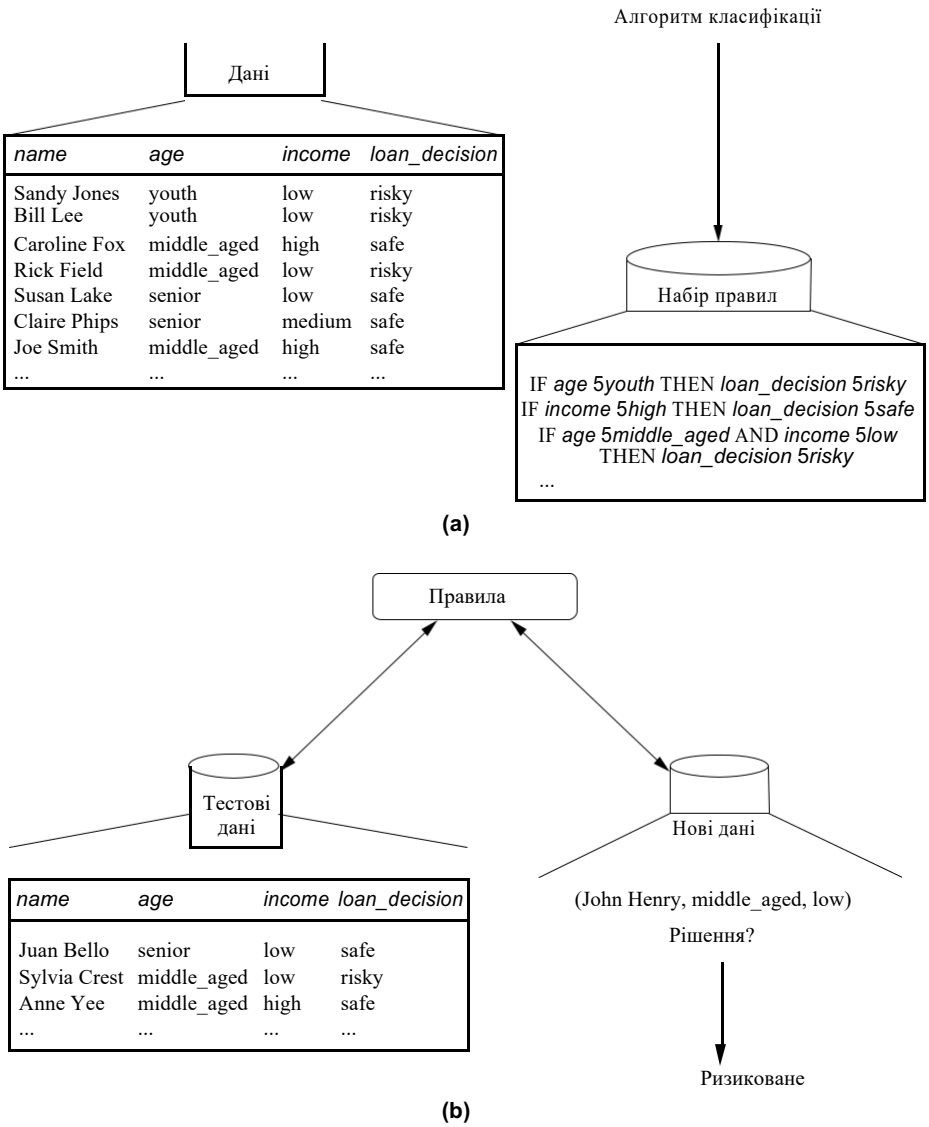

Рис. 2.1. Приклад процесу класифікації [60]

Отже, зі схеми визе можна виокремити:

(a) Навчання: дані навчання аналізуються за допомогою алгоритму класифікації. Тут атрибутом мітки класу є рішення про позику, а вивчена модель або класифікатор представлені у вигляді правил класифікації.

(b) Класифікація: Дані тестів використовуються для оцінки точності правил класифікації. Якщо точність вважається прийнятною, правила можуть застосовуватися до класифікації нових наборів даних.

Оскільки мітка класу кожного навчального кортежу надана, цей крок також відомий як контрольоване навчання (тобто навчання класифікатора є «контрольованим», оскільки повідомляється, до якого класу належить кожен кортеж). Це контрастує з неконтрольованим навчанням (або кластеризацією), коли позначка класів кожного кортежу не відома, а кількість або набір класів, які потрібно вивчити, може бути невідома заздалегідь. Наприклад, якщо у нас не було даних про рішення про позику для навчального набору, ми могли б використовувати кластеризацію, щоб спробувати визначити “групи подібних кортежів”, які можуть відповідати групам ризику в даних заявки на позику.

Цей перший крок процесу класифікації також можна розглядати як вивчення пінг-картки або функції, яка може передбачати відповідну мітку класу у даного кортежу X. У цьому поданні ми хочемо вивчити відображення або функцію, яка розділяє класи даних. Як правило, це відображення представлене у вигляді правил класифікації, дерев рішень або математичних формул. У нашому прикладі відображення представлене як правила класифікації, які визначають заявки на позики як безпечні, так і ризиковані (пункт «а» попередньої схеми). Правила можуть бути використані для класифікації майбутніх наборів даних, а також забезпечують більш глибоке розуміння вмісту даних. Вони також забезпечують представлення стиснених даних.

Наступне питання - точність класифікації. На другому етапі (пункт «б») модель використовується для класифікації. Спочатку оцінюється прогнозована точність класифікатора. Якби ми використовували навчальний набір для вимірювання точності класифікатора, ця оцінка, швидше за все, була б оптимістичною, оскільки класифікатор має тенденцію перевантажувати дані (тобто під час навчання він може включати деякі особливі аномалії навчальних даних, яких немає у загальний набір даних загальний). Тому використовується тестовий набір, який складається з тестових кортежів та пов'язаних з ними міток класів. Вони не залежать від навчальних кортежів, що означає, що їх не використовували для побудови класифікатора.

Точність класифікатора для даного тестового набору - це відсоток кортежів набору тестів, які класифікуються правильно класифікатором. Пов’язана мітка класу кожного тестового кортежу порівнюється з прогнозом класу вивченого класифікатора для цього кортежу. Якщо точність класифікатора вважається прийнятною, класифікатор може бути використаний для класифікації майбутніх наборів даних, для яких мітка класу невідома. Наприклад, правила класифікації, вивчені на малюнку 8.1 (а) з аналізу даних попередніх заявок на позику, можуть бути використовується для схвалення або відхилення нових або майбутніх заявників позики.

Дерева рішень для вирішення задач класифікації

Дерево рішень - це схожа на блок-схему деревоподібна структура, де кожен внутрішній вузол (не-листяний вузол) позначає тест на атрибут, кожна гілка являє собою результат тесту, а кожен листовий вузол (або термінальний вузол) містить мітку класу.

Наприкінці 1970-х - на початку 1980-х років Дж. Росс Квінлан, дослідник машинного навчання, розробив алгоритм дерева рішень, відомий як ID3 (Ітеративний дихотомайзер). Ця робота розширила попередню роботу над концептуальними системами навчання, описану Е. Б. Хант, Дж. Маріном та П. Т. Стоуном. Пізніше Квінлан представив C4.5 (наступник ID3), який став еталоном, з яким часто порівнюються новіші керовані алгоритми навчання. У 1984 р. Група статистиків (Л. Брейман, Дж. Фрідман, Р. Ольшен та К. Стоун) опублікувала книгу "Класифікація та дерева регресії" (CART), де описано створення бінарних дерев рішень. ID3 та CART були винайдені незалежно один від одного приблизно в той же час, проте дотримуються подібного підходу для вивчення дерев рішень на основі навчальних кортежів. Ці два наріжні алгоритми породили безліч робіт з індукції дерева рішень.

ID3, C4.5 та CART застосовують жадібний (тобто невідслідковуючий) підхід, при якому дерева рішень будуються рекурсивним способом поділу та завоювання зверху вниз. Більшість алгоритмів для індукції дерева рішень також

дотримуються підходу зверху вниз, який починається з навчального набору кортежів та пов'язаних з ними міток класів. Набір тренувань рекурсивно розподіляється на менші підмножини під час побудови дерева. Базовий алгоритм дерева рішень узагальнений на Рис. 2.2.

Найвищий вузол у дереві - це кореневий вузол. Він представляє концепцію покупки комп’ютера, тобто передбачає, чи може клієнт придбати комп’ютер. Внутрішні вузли позначаються прямокутниками, а листові - овалами. Деякі алгоритми дерева рішень створюють лише бінарні дерева (де кожен внутрішній вузол розгалужується рівно до двох інших вузлів), тоді як інші можуть створювати небінарні дерева.

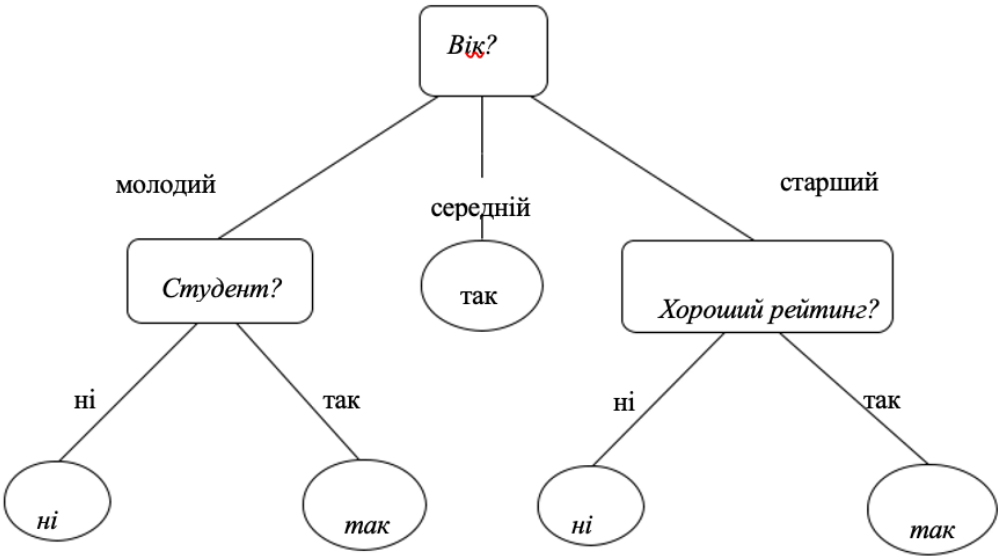

Рис 2.2. Приклад дерева рішень [60]

Враховуючи кортеж X, для якого пов'язана мітка класу невідома, значення атрибутів кортежу перевіряються на основі дерева рішень. Шлях простежується від кореня до листового вузла, який містить передбачення класу для цього кортежу. Дерева рішень можна легко перетворити на правила класифікації.

Побудова класифікаторів дерева рішень не вимагає знань доменів або встановлення параметрів, а тому підходить для дослідницького виявлення знань. Дерева рішень можуть обробляти багатовимірні дані. Їх представлення набутих знань у формі дерева інтуїтивно зрозуміле і загалом легко засвоюється

людьми. Етапи навчання та класифікації введення дерева рішень прості та швидкі. Загалом класифікатори дерев рішень мають хорошу точність. Однак успішне використання може залежати від наявних даних. Алгоритми індукції дерева рішень використовувались для класифікації у багатьох областях застосування, таких як медицина, виробництво та виробництво, фінансовий аналіз, астрономія та молекулярна біологія.

Коли побудовано дерево рішень, багато гілок відображатимуть аномалії в навчальних даних через шум або викиди. Методи обрізки дерев вирішують цю проблему даних. Такі методи зазвичай використовують статистичні заходи для видалення найменш надійних гілок. Обрізані дерева, як правило, менші та менш складні, а отже, легші для сприйняття. Зазвичай вони швидше та краще правильно класифікують незалежні дані випробувань (тобто раніше не бачених кортежів), ніж необрізані дерева.

Існує два загальноприйняті підходи до обрізки дерев: до та після побудови.

У підході до попередньої обрізки дерево «обрізають», зупиняючи свою конструкцію достроково (наприклад, вирішивши більше не розділяти підгрупу навчальних кортежів на даному вузлі). Після зупинки вузол стає листком. Листок може містити найпоширеніший клас серед підмножин кортежів або розподіл ймовірностей цих кортежів.

При побудові дерева для оцінки ефективності розбиття можна використовувати такі показники, як статистична значимість, приріст інформації, індекс Джині тощо. Якщо розділення кортежів на вузлі призведе до розщеплення, яке опускається нижче заданого порогу, тоді подальше розділення даної підмножини припиняється. Однак при виборі відповідного порогу виникають труднощі, адже високі пороги можуть призвести до спрощення дерев.

Другим і більш поширеним підходом є післязрізання, яке видаляє піддерева з “повністю вирощеного” дерева. Піддерево на даному вузлі обрізають, видаляючи його гілки та замінюючи його листом. Листок позначений найпоширенішим класом серед замінених піддерев.

Алгоритм обрізки складності витрат, що використовується в CART, є прикладом підходу післяобробки. Цей підхід розглядає складність витрат на дерево як функцію від кількості листків у дереві та рівня помилок дерева (де коефіцієнт помилок - це відсоток кортежів, неправильно класифікованих деревом), що починається з нижньої частини дерева. Для кожного внутрішнього вузла N він обчислює складність витрат піддерева на N та складність витрат піддерева на N, якщо його потрібно було обрізати (тобто замінити на листовий вузол). Порівнюються два значення. Якщо обрізання піддерева у вузлі N призведе до меншої складності витрат, тоді піддерево обрізається. В іншому випадку воно зберігається.

Набір обрізань маркованих класом кортежів використовується для оцінки складності витрат. Цей набір не залежить від навчального набору, що використовується для побудови необрізаного дерева, і від будь-якого набору випробувань, що використовується для оцінки точності. Алгоритм генерує набір поступово обрізаних дерев. Взагалі, віддають перевагу найменшому дереву рішень, яке мінімізує складність витрат.

Метод опорних векторів

Метод опорних векторів (SVM) - це алгоритм, який працює наступним чином. Він використовує нелінійне відображення для перетворення вихідних навчальних даних у вищий вимір. У цьому новому вимірі він шукає лінійний оптимальну розділяючу гіперплощину (тобто, “межу рішення”, що відокремлює кортежі одного класу від іншого). За умови відповідного нелінійного відображення до досить великого виміру, дані з двох класів завжди можна розділити гіперплощиною. SVM знаходить цю гіперплощину, використовуючи вектори підтримки (“основні” навчальні кортежі) та поля (визначені векторами підтримки).

Перший документ про опорні векторні машини був представлений у 1992 р. Володимиром Вапником та колегами Бернгардом Бозером та Ізабель Гюйон, хоча основи для SVM існують з 1960-х років (включаючи ранні роботи Вапніка

та Олексія Червоненкіса з теорії статистичного навчання). Хоча час навчання навіть найшвидших SVM може бути надзвичайно довгим, вони дуже точні завдяки своїй здатності моделювати складні нелінійні межі прийняття рішень. Вони набагато менше схильні до перенавчання, ніж інші методи. Знайдені вектори підтримки також дають компактний опис вивченої моделі. SVM можна використовувати для числового прогнозування, а також для класифікації. Вони були застосовані до ряду областей, включаючи рукописне розпізнавання цифр, розпізнавання об’єктів та ідентифікацію динаміків, а також базові тести прогнозування часових рядів.

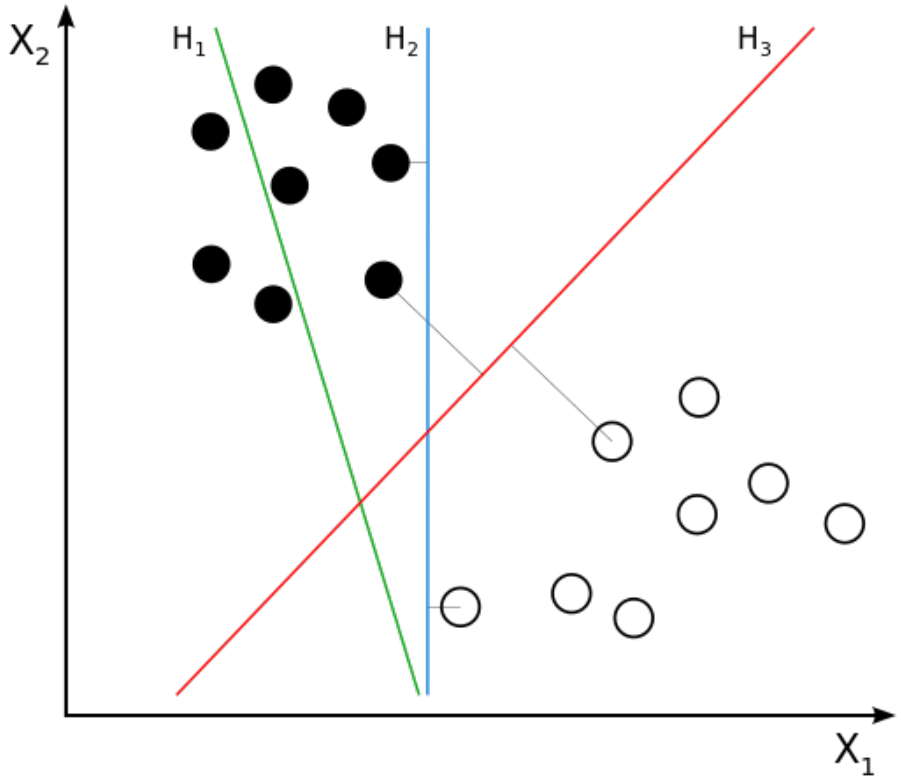

Рис. 2.3. Схематичне зображення класифікації методом опорних векторів [60]

Виходячи з Рис. 2.3, найкраще розділення відбувається гіперплощиною, яка має найбільшу дистанцію до найближчих точок тренувальних даних будь-якого з класів (так зване функціональне розділення) - як бачимо, H1 не розділяє ці класи. H2 - розділяє, але не є оптимальним. H3 розділяє їх із максимальною дистанцією, тому і є фінальним класифікатором.

Байєсові класифікатори

Баєсові класифікатори - це статистичні класифікатори. Вони можуть передбачити такі ймовірності членства в класі, як імовірність належності даного кортежу до певного класу.

Теорема Байєса названа на честь Томаса Байєса, нонконформістського англійського священнослужителя, який працював над теорією ймовірностей та прийняття рішень протягом 18 століття. Нехай X - кортеж даних. Якщо говорити по-байєсівськи, X вважається «доказом». Як зазвичай, це описується вимірами, проведеними за набором з n атрибутів. Нехай H є якоюсь гіпотезою, наприклад, що набір даних X належить до зазначеного класу C. Для задач класифікації ми хочемо визначити P.HjX /, тобто ймовірність того, що гіпотеза H виконується з урахуванням “доказів” або спостережуваного набору даних X. Іншими словами, ми шукаємо ймовірність того, що кортеж X належить до класу C, враховуючи те, що ми знаємо опис атрибута X.

Наївські байєсівські класифікатори припускають, що вплив значення атрибута на даний клас не залежить від значень інших атрибутів. Це припущення називається умовно-класовою незалежністю. Це зроблено для спрощення обчислень і, в цьому сенсі, вважається "наївним". Більше того, гіпотеза про незалежність ознак суттєво спрощує завдання, тому що оцінити n одномірних щільностей куди менш складно, ніж одну n-мірну щільність.

Класифікатор в кожному окремому випадку може бути як параметричним, так і непараметричним, це варіюється від метода, за яким відновлюються одновимірні щільності.

Серед переваг метода можна виокремити нескладну реалізацію та низькі обчислювальні витрати під час навчання та класифікації. У тих надзвичайно нечастих випадках, коли ознаки дійсно взаємо незалежні, наївний класифікатор Байєса – (майже) оптимальний.

Найчастіше його беруть до уваги для порівняння різних моделей алгоритмів або як структурну одиницю в складніших ансамблях алгоритмів.

Метод найближчих сусідів

Метод k-найближчого сусіда вперше був описаний на початку 1950-х років. Метод є трудомістким, коли йому дають великі навчальні набори, і він набув популярності лише в 1960-х роках, коли з'явилася підвищена обчислювальна потужність. З тих пір він широко використовується в області розпізнавання зразків.

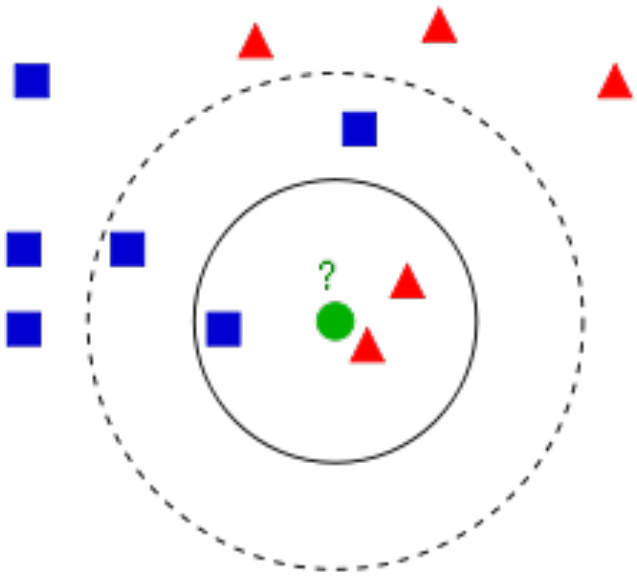

Рис. 2.4. Приклад класифікації k найближчих сусідів [60]

Класифікатори найближчих сусідів засновані на навчанні за аналогією, тобто шляхом порівняння даного тестового кортежу з навчальними кортежами, подібними до нього. Навчальні кортежі описуються n атрибутами. Кожен кортеж представляє точку в n-мірному просторі. Таким чином, всі тренувальні кортежі зберігаються у n-вимірному просторі шаблону. Коли дається невідомий кортеж, класифікатор k-найближчого сусіда здійснює пошук у просторі шаблонів для k-тренувальних кортежів, найближчих до невідомого кортежу. Ці k навчальні кортежі є k «найближчими сусідами» невідомого кортежу.

"Близькість" визначається через метрику відстані, таку як евклідова відстань. Відстань Евкліда між двома точками або кортежами, скажімо, X1 = (x11, x12, ..., x1n) та X2 = (x21, x22, ..., x2n), дорівнює

$$dist(\boldsymbol{X}_1, \boldsymbol{X}_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}. \qquad (2.1)$$

Іншими словами, для кожного числового атрибута ми беремо різницю між відповідними значеннями цього атрибута в кортежі X1 і кортежі X2, квадратуємо цю різницю і накопичуємо її. Квадратний корінь береться із загальної кількості накопичених відстаней. Як правило, ми нормалізуємо значення кожного атрибута перед використанням рівняння. Це допомагає запобігти перевищуванню атрибутів із початково великими діапазонами (наприклад, доходу) над атрибутами з початково меншими діапазонами (наприклад, двійкові атрибути). Наприклад, мінімальна нормалізація, наприклад, може бути використана для перетворення значення v числового атрибута A на v ′ в діапазоні [0, 1] шляхом обчислень

$$v' = \frac{v - min_A}{max_A - min_A}, \tag{2.2}$$

де *minA* та *maxA* - мінімальні та максимальні значення атрибута A.

Для класифікації k-найближчого сусіда невідомому кортежу присвоюється найбільш поширений клас серед його k-найближчих сусідів. Коли k = 1, невідомому кортежу присвоюється клас навчального кортежу, який є найближчим до нього в просторі шаблонів. Класифікатори найближчих сусідів також можуть бути використані для числового прогнозування, тобто для повернення реального значення прогнозу для даного невідомого кортежу. У цьому випадку класифікатор повертає середнє значення справжніх міток, пов’язаних з k-найближчими сусідами невідомої кортежі.

Попереднє обговорення передбачає, що всі атрибути, що використовуються для опису кортежів, є числовими. Для номінальних атрибутів простим методом є порівняння відповідного значення атрибута в кортежі X1 із значенням у кортежі X2. Якщо обидва ідентичні (наприклад, кортежі X1 і X2 обидва мають синій колір), тоді різниця між ними приймається рівною 0. Якщо обидва вони різні (наприклад, кортеж X1 синій, а кортеж X2 червоний), то різниця вважається рівною 1. Інші методи можуть включати більш складні схеми диференціального

оцінювання (наприклад, коли більший бал різниці призначається, скажімо, для синього та білого, ніж для синього та чорного).

Якщо ж значення заданого атрибута A відсутнє в кортежі X1 та / або кортежі X2, ми припускаємо максимально можливу різницю. Припустимо, що кожен з атрибутів був зіставлений у діапазон [0, 1]. Для номінальних атрибутів ми приймаємо значення різниці як 1, якщо одне або обидва відповідні значення A відсутні. Якщо A числовий і відсутній в обох кортежах X1 і X2, тоді різниця також приймається рівною 1. Якщо лише одне значення відсутнє, а друге (яке ми будемо називати v ′) присутнє та нормалізоване, тоді ми можемо взяти різниця повинна бути або | 1 - v ′ | або | 0 - v ′ | (тобто 1 - v ′ або v ′), залежно від того, що більше.

Наступне питання - як я можна визначити хороше значення k, кількість сусідів. Це можна визначити експериментально. Починаючи з k = 1, ми використовуємо тестовий набір для оцінки рівня помилок класифікатора. Цей процес можна повторити щоразу, збільшуючи k, щоб забезпечити ще одного сусіда. Може бути вибрано значення k, яке дає мінімальну частоту помилок. Загалом, чим більша кількість навчальних кортежів, тим більшим буде значення k (так що рішення щодо класифікації та числового прогнозування можуть базуватися на більшій частині збережених кортежів). Оскільки кількість навчальних кортежів наближається до нескінченності і k = 1, коефіцієнт помилок може бути не гіршим, ніж подвоєний коефіцієнт помилки Байєса (останній є теоретичним мінімумом). Якщо k також наближається до нескінченності, рівень помилок наближається до рівня помилки Байєса. Класифікатори найближчих сусідів використовують порівняння на основі відстані, яке суттєво присвоюється рівна вага кожного атрибута. Тому вони можуть страждати від поганої точності, якщо їм надаються шумні або невідповідні атрибути. Однак метод був модифікований для включення зважування атрибутів та обрізки шумних кортежів даних. Вибір метрики відстані може бути критичним. Також може використовуватися відстань на Манхеттені (міський квартал) (Розділ 2.4.4) або інші вимірювання відстані.

Класифікатори найближчих сусідів можуть бути надзвичайно повільними при класифікації тестових кортежів. Якщо D - це навчальна база даних | D | кортежі та k = 1, тоді для класифікації даного тестового кортежу потрібні порівняння O (|D|). Завдяки попередньому упорядкуванню та розташуванню збережених кортежів у деревах пошуку кількість порівнянь можна зменшити до O (log (|D|). Паралельна реалізація може зменшити час роботи до константи, тобто O (1), яка не залежить з |D|.

Інші методи для прискорення часу класифікації включають використання часткових розрахунків відстані та редагування збережених кортежів. У методі часткової відстані ми обчислюємо відстань на основі підмножини n атрибутів. Якщо ця відстань перевищує поріг, подальші обчислення для даного збереженого кортежу припиняються, і процес переходить до наступного збереженого кортежу. Метод редагування видаляє тренувальні кортежі, які виявляються марними. Цей метод також називають обрізанням або конденсацією, оскільки він зменшує загальну кількість збережених кортежів.

Нейронні мережі

Поле нейронних мереж спочатку досліджували психологи та нейробіологи, які прагнули розробити та перевірити обчислювальні аналоги нейронів. Грубо кажучи, нейронна мережа - це сукупність підключених вхідних / вихідних одиниць, у яких кожне з'єднання має вагу, пов'язану з ним. На етапі навчання мережа вчиться, регулюючи ваги, щоб мати можливість передбачити правильну мітку класу вхідних кортежів.

Нейронні мережі передбачають тривалий час навчання. Вони вимагають ряду параметрів, які зазвичай найкраще визначаються емпірично, таких як топологія мережі або „структура”. Нейронні мережі критикували за погану інтерпретацію. Наприклад, людям важко інтерпретувати символічне значення, що стоїть за вивченими вагами та «прихованими одиницями» в мережі. Спочатку ці особливості робили нейронні мережі менш бажаними для видобутку даних.

Однак переваги нейронних мереж включають їх високу толерантність до шуму в даних, а також їх здатність класифікувати моделі, на яких вони не навчались. Їх можна використовувати, коли ви можете мало знати про взаємозв'язок між атрибутами та класами. Вони добре підходять для безперервних входів і виходів, на відміну від більшості алгоритмів дерев рішень. Вони досягли успіху в широкому спектрі реальних даних, включаючи рукописне розпізнавання символів, патологію та лабораторну медицину, а також навчання комп’ютера для вимови англійського тексту. Алгоритми нейронної мережі є спадково паралельними; для прискорення процесу обчислень можна використовувати методи паралелізації. Крім того, нещодавно було розроблено кілька методів для вилучення правил із навчених нейронних мереж. Ці фактори сприяють корисності нейронних мереж для класифікації та чисельного прогнозування при обробці даних.

Існує багато різних видів нейронних мереж та алгоритмів нейронних мереж. Найпопулярнішим алгоритмом нейронної мережі є зворотне розповсюдження, яке здобуло репутацію у 1980-х.

Алгоритм зворотного розповсюдження виконує навчання на багатошаровій нейронній мережі зворотного зв'язку. Він ітеративно вивчає набір вагових коефіцієнтів для прогнозування мітки класу кортежів. Багатошарова нейронна мережа прямої передачі складається з вхідного рівня, одного або декількох прихованих шарів та вихідного рівня. Приклад багатошарової мережі прямої передачі наведено на Рис. 2.5.

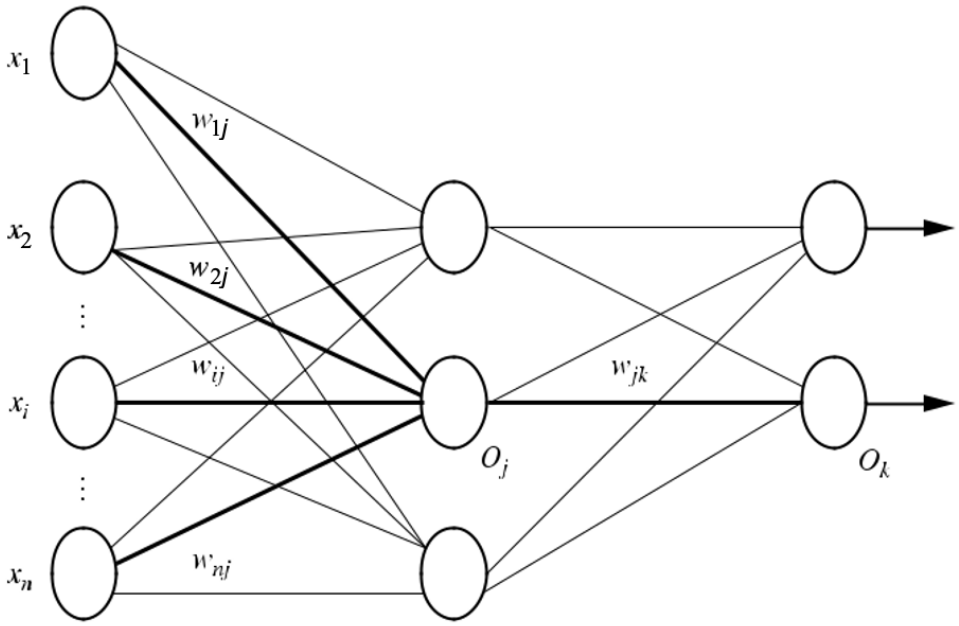

Рис. 2.5. Приклад багатошарової мережі прямої передачі (шар входу -> прихований шар -> шар виходу) [37]

Кожен шар складається з одиниць. Входи в мережу відповідають атрибутам, виміряним для кожного навчального кортежу. Входи подаються одночасно в блоки, що складають вхідний рівень. Ці входи проходять через вхідний шар, а потім зважуються і подаються одночасно на другий шар "нейроноподібних" одиниць, відомий як прихований шар. Виходи одиниць прихованого шару можуть бути введені на інший прихований шар тощо. Кількість прихованих шарів довільна, хоча на практиці зазвичай використовується лише один. Зважені результати останнього прихованого шару вводяться в одиниці, що складають вихідний рівень, який випромінює прогноз мережі для заданих кортежів.

Одиниці на вхідному рівні називаються вхідними одиницями. Одиниці в прихованих шарах і вихідному шарі іноді називають нейронами через їх символічну біологічну основу або вихідними одиницями. Багатошарова нейронна мережа, показана на Рис. 2.5, має два шари вихідних одиниць. Тому ми говоримо, що це двошарова нейронна мережа. (Вхідний рівень не враховується, оскільки він служить лише для передачі вхідних значень наступному шару.) Подібним чином, мережа, що містить два приховані шари, називається тришаровою нейронною мережею тощо. Це мережа прямого пересилання, оскільки жоден з вагових коефіцієнтів не повертається до вхідного блоку або до

вихідного блоку попереднього рівня. Він повністю пов'язаний тим, що кожен блок забезпечує введення даних для кожного блоку в наступному прямому рівні. Також він застосовує нелінійну (активаційну) функцію до зваженого вводу. Багатошарові нейромережі прямої передачі здатні моделювати передбачення класу як нелінійну комбінацію входів, тобто зі статистичної точки зору вони здійснюють нелінійну регресію. Багатошарові мережі прямого пересилання, маючи достатньо прихованих блоків та достатню кількість навчальних зразків, можуть наблизити будь-яку функцію.

Отже, підведемо підсумки: класифікація - це форма аналізу даних, яка виділяє моделі, що описують класи даних. Класифікатор або модель класифікації передбачає категоріальні мітки (класи). Числові передбачення моделюють безперервні функції. Класифікація та числове передбачення - два основних типи проблем прогнозування.

Індукція дерева рішень - це алгоритм індукції рекурсивного дерева зверху вниз, який використовує міру вибору атрибутів для вибору атрибута, що перевіряється для кожного нелистового вузла в дереві. ID3, C4.5 та CART є прикладами таких алгоритмів, що використовують різні заходи вибору атрибутів. Алгоритми обрізки дерев намагаються підвищити точність, видаляючи гілки дерев, що відображають шум у даних.

Байєсівський класифікатор базується на теоремі Байєса про ймовірність. Він передбачає умовну незалежність класу - що вплив значення атрибута на даний клас не залежить від значень інших атрибутів.

Побудова та оцінка класифікатора вимагає розподілу маркованих даних на навчальний набір та набір тестів.

Тести значимості та криві ROC є корисними інструментами для вибору моделі. Тести на значимість можуть бути використані для оцінки того, чи є різниця в точності двох класифікаторів випадковою.

2.2 Структурування та формалізація процесів емоційного аналізу тексту

Проблема виявлення думок людей, висловлених письмовою мовою, є відносно новою і дуже активною галуззю досліджень. Наявність величезного обсягу даних через всюдисущість Інтернету дозволило дослідникам у різних областях - таких як обробка природних мов, машинне навчання та видобуток даних, видобуток тексту, управління та маркетинг та навіть психологія - проводити дослідження з метою виявлення думки та настрої людей із загальнодоступних джерел даних.

Зі зростанням популярності аналітики та науки про дані, обчислювальні методи інтелекту виявляються важливими конкурентними інструментами у багатьох галузях. Наприклад, у бізнес-аналітиці дані видобуваються для моделей, які допоможуть краще зрозуміти клієнтів та покращать продажі та маркетинг. Ці методи дозволяють використовувати імовірнісні методи для пошуку закономірностей у даних.. Крім того, величезна кількість даних, що вимагає аналізу, тепер формується в письмовій формі. Наприклад, користувачі можуть залишати письмові коментарі щодо товару чи послуги на таких веб-сайтах, як Yelp та TripAdviser. Написаний текст підлягає інтерпретації, і подання даних в абсолютному синтаксисі (наприклад, в двійковій системі) складно. Однак обчислювальні методи інтелекту дозволяють обійти таку нечіткість і можуть бути найбільш підходящими методами пошуку закономірностей у таких даних.

Аналіз настрою об'єднує різні напрямки досліджень, такі як обробка природних мов, видобуток даних та видобуток тексту, і швидко стає важливим для організацій, оскільки вони прагнуть інтегрувати методи обчислювальної розвідки у свої операції, а також намагаються пролити більше світла та вдосконалити , їх продукція та послуги. При аналізі настроїв або аналізі думок (SAOM) метою є виявлення думок людей, висловлених письмовою мовою (текстом). Термін означає "те, що хтось до чогось відчуває", "особистий досвід, власні почуття", "ставлення до чогось" або "думка".

Думки є центральними для майже всієї людської діяльності та є ключовими факторами, що впливають на нашу поведінку. Наші переконання та уявлення про реальність, а також вибір, який ми робимо, значною мірою зумовлені тим, як інші бачать і оцінюють світ. З цієї причини, коли нам потрібно прийняти рішення, ми часто шукаємо думки інших. Це стосується не лише приватних осіб, але й організацій. Традиційні, закриті форми опитувальників задоволеності споживачів будуть використовуватися для визначення значущих компонентів або аспектів загальної задоволеності споживачів. Однак розробка та виконання анкет є дорогими або можуть бути недоступними. У деяких випадках державним установам навіть заборонено законом збирати анкети задоволення від споживачів. У таких випадках єдиною альтернативою може бути аналіз загальнодоступних текстових коментарів у вільній формі.

Організації все більше зосереджуються на розумінні того, як їх діяльність, що створює цінність, сприймається споживачами. Думки клієнтів визначають імідж організації та попит на їх продукцію чи послуги. Для некомерційних та урядових організацій краще обслуговування потреб споживачів допомагає отримувати політичну підтримку та підтримку платників податків, тоді як у комерційних організаціях покращення розуміння споживачів стимулює організаційну здатність отримувати доходи та конкурувати на ринку. Отже, розуміння сприйняття клієнтів є запорукою успіху організації.

Сприйняття споживачами товарів чи послуг часто визначається шляхом опитувань, фокус-груп, спостереження та інших досить трудомістких та дорогих методів. З появою Інтернету інструменти опитування стали більш доступними (і дешевшими у використанні), але отримання точних та відповідних даних з опитувань клієнтів все ще залишається проблемою. У той же час Інтернет створив цілу галузь послуг з перегляду продуктів / послуг, таких як Yelp, яка надає нові можливості та цілий океан інформації щодо думок клієнтів. Прикладами можуть бути такі сайти, як Trip Advisor для подорожей, Yelp та Urban Spoon для ресторанів, Patagonia для відкритого одягу та спорядження, Lands ’End для одягу та Epinions для оглядів товарів стали звичними явищами.

Ці веб-сайти дозволяють клієнтам як читати, так і надавати відгуки про товар або послугу. Відгуки клієнтів, які часто відкрито доступні в Інтернеті, містять велику кількість інформації, яка може використовуватися керівництвом, конкурентами, інвесторами та іншими зацікавленими сторонами для розпізнавання проблем клієнтів, що зумовлюють загальне задоволення клієнтів певною послугою чи товаром (для простоти, ми використовуємо терміни загальне задоволення та загальний рейтинг взаємозамінні). Зазвичай кількість онлайн-оглядів про об'єкт дуже великих масштабів, наприклад, тисячі тисяч людей, і кількість постійно зростає, оскільки все більше і більше людей продовжують робити свої внески в Інтернеті. Таким чином, ми зараз стикаємося з відносно новим викликом вилучення думок та настроїв споживачів із (часто) неструктурованих даних у вигляді текстових коментарів.

Ідеальний метод - це той, який дозволить нам автоматично визначати виміри, що обумовлюють задоволеність споживачів, без апріорних знань про ці розміри та їх вплив на загальну задоволеність споживачів. Переваги такої системи полягають у тому, що вона зможе виявити компоненти та їх вплив на загальне задоволення без втручання людини. Така система дозволяє аналізувати великі обсяги даних, які інакше люди не могли б проаналізувати.

Як приватні особи, так і організації можуть скористатися перевагами аналізу настроїв та аналізу думок. Коли особа хоче придбати товар або вирішить, користуватися послугою чи ні, вона / вона має доступ до великої кількості відгуків користувачів, але читання та аналіз усіх їх може бути тривалим і, можливо, розчаруванням. Крім того, коли організація або менеджер прагне викликати громадську думку про їх продукцію, продати свою продукцію, виявити нові можливості, передбачити тенденції продажів або керувати своєю репутацією, їй потрібно мати справу з переважною кількістю наявних коментарів клієнтів. За допомогою методів аналізу настроїв ми можемо автоматично аналізувати велику кількість доступних даних та витягувати думки, які можуть допомогти як клієнтам, так і організації досягти своїх цілей. Це одна з причин,

чому аналіз настроїв отримав широке поширення від інформатики до управління та соціальних наук.

Аналіз настрою може бути використаний як доповнення до інших систем, таких як системи рекомендацій, вилучення інформації та системи відповідей на запитання. Ефективність систем рекомендацій може покращитися, якщо не рекомендувати елементи, які отримують багато негативних відгуків. Системи вилучення інформації можуть отримати користь від систем видобування думок, відкидаючи певні типи інформації, що містяться в суб'єктивних реченнях. У системах відповідей на запитання різні типи питань (орієнтовані на думку та дефініційні питання) можуть набувати різних типів обробок. Це ще одне поле, яке може використати видобуток думок.

2.2.1 Постановка проблеми та огляд викликів при аналізі тексту

У цьому розділі ми визначимо проблему SAOM, зокрема проблему аналізу настроїв на рівні аспекту. Спочатку корисно визначити деякі терміни та поняття, які пов'язані з проблемою.

Співавтор - це особа чи організація, яка висловлює свої думки письмовою мовою чи текстом.

Об'єкт - це сутність, яка може бути товаром, послугою, особою, подією, організацією або темою. Це може бути пов'язано з набором компонентів та атрибутів, які в літературі з аналізу настроїв ці компоненти та атрибути називаються аспектами.

Рецензія - це текст, створений автором, який містить думки учасника щодо деяких аспектів об'єкта. Рецензію також можна назвати довіреним документом.

Загальний рейтинг: користувач повідомляє про загальне задоволення об'єктом, наприклад за шкалою від 1 до 5.

Думка: це позитивна, нейтральна чи негативна емоція, ставлення чи оцінка щодо цього аспекту учасника.

Аспект: важливий атрибутом об’єкта щодо загального задоволення споживачів, про який автор заявив у своєму огляді.

Огляд може бути створений багатьма учасниками або містити думки з багатьох джерел. Також слід зауважити, що огляд може бути прямим оглядом окремого об'єкта або порівняльним оглядом, який порівнює 2 або більше об'єктів між собою. Загалом огляд di - це сукупність речень $d_i = \{s_{i\,1}, s_{i\,2}, \ldots, s_{i\,m}\}$, які містять думки різних учасників про різні аспекти різних об’єктів. Хоча ця модель не є всебічною моделлю, яка містить всю інформацію про всі випадки, вона є достатньою для практичних застосувань. Варто згадати, що більшість літератури розглядали спрямованість думки як двійкову змінну: позитивну чи негативну. Є деякі роботи, що включають і нейтральний клас. Загалом орієнтацію на думку можна розглядати в різних масштабах, а не лише в двійковій чи трійковій формі.

Мета SAOM може бути записана як:

Дано збірник відгуків $D = \{d_1, d_2, \ldots, d_D\}$ все про об’єкт, викриті всі аспекти та відповідні настрої, виражені в цій колекції.

Цю мету можна досягти на різних рівнях. Якщо фокус зосереджений на кожному документі і виявлено орієнтацію настроїв усього документа, це називається аналізом настроїв на рівні документа. Хоча документ може передавати загальний позитивний чи негативний настрій, цілком ймовірно, що не всі речення в документі є позитивними чи негативними. Наприклад, якщо цікаві відгуки - це огляди фільмів, конкретний співавтор може дати загальну позитивну оцінку фільму, однак їм можуть не сподобатися всі аспекти цього фільму, і в своєму огляді вони згадують, які аспекти їм подобаються, а що не подобаються про той фільм. Коли SAOM зосереджується на реченнях і виявляється орієнтація настрою для речень рецензії, це називається аналізом настрою на рівні речення. Під час більш точного аналізу, можливо, буде цікаво дізнатись, які конкретні аспекти об’єкта коментує учасник, а також подобаються вони їм чи ні. Цей рівень SAOM називається аспектом SAOM.

Візуалізація та звітування про результати SAOM є важливими питаннями, які потребують особливої уваги. Існують різні способи візуалізації результатів

аналізу. Одним із шляхів може бути складання переліку виявлених аспектів та прив'язка до кожного з них відповідних позитивних та негативних відгуків. Це можна доповнити рейтингом для кожного аспекту. Таким чином, люди можуть легко дізнатися думку своїх однолітків щодо різних аспектів і конкретно вибрати, які саме відгуки вони хочуть прочитати. Цілком імовірно, що деякі учасники дають позитивні / негативні оцінки об’єкту лише тому, що вони (не) задоволені певним аспектом об’єкта. Інші користувачі можуть / не піклуватися про ці конкретні аспекти так само, як інші.

Наявність звіту про огляд, орієнтований на аспекти, дає змогу більше зосередитися на конкретних аспектах, які вас цікавлять, і шукати думки інших щодо них. Якщо ви хочете отримати уявлення про популярні або непопулярні аспекти об'єкта, створення частотного списку різних виявлених аспектів може бути корисним. Однією з переваг SAOM є те, що він надає можливість окремим особам та організаціям відстежувати думку людей з часом. Те, як речі змінюються з часом, має критичне значення для людей та організацій. Наприклад, Blackberry роками була лідером на ринку мобільних телефонів, але програла тиску на ринку через відсутність функцій у своїх мобільних телефонах. Коли важливість відсутніх аспектів телефону Blackberry, таких як відсутність здатності до текстових повідомлень, переважала важливість захищеної електронної пошти, їх частка на ринку сильно зменшилася.

Підсумуємо нижче деякі основні проблеми, пов'язані з SAOM.

Синонімія та полісемія

Користувачі в різному контексті або з різними потребами, знаннями чи мовними звичками описуватимуть одну і ту ж інформацію, використовуючи різні терміни (синонімія). Наприклад, Фурнас та інші дослідники цієї теми показували, що люди генерують одне й те саме ключове слово для опису добре відомих об’єктів лише у 20 відсотках часу. Оскільки шукачі та автори часто використовують різні слова, відповідні матеріали пропускаються. Люди також використовують одне й те саме слово для позначення різних речей (полісемія).

Такі слова, як сатурн, ягуар чи чіп, мають кілька різних значень. У різних контекстах або при використанні різними людьми один і той же термін набуває різного посилального значення. Загалом синонімія та багатозначність спричинені можливою мінливістю у вживанні слів. Проблеми синонімії та полісемії становлять складні завдання для аналізу настроїв та проблеми видобутку думок.

Для вирішення проблем синонімії та полісемії було запропоновано кілька підходів. Популярним методом є стемлінг, який можна розглядати як нормалізацію якоїсь мінливості на рівні поверхні. Метою стемінгу є зближення варіантних форм слова (до їх морфологічних коренів). Алгоритм стемінгу для англійської мови зводить слова “stemmer”, “stemming”, “stemming” та “stems” до кореневого слова “stem”. Алгоритми стемлінгу вивчаються в комп'ютерній науці з 1960-х років. Стермінг іноді може допомогти отримати інформацію, але він не стосується випадків, коли споріднені слова не пов'язані морфологічно (наприклад, лікар та лікар).

Контрольована лексика - це ще один підхід, який продемонстрував свою ефективність у вирішенні проблем, спричинених мінливістю у вживанні слів. Однак, оскільки підхід до контрольованої лексики вимагає обмеження термінів заздалегідь визначеним списком слів, це не застосовується в SAOM (неможливо обмежити учасників обмежити свій словниковий запас попередньо визначеним словом). На оглядових веб-сайтах учасники, як правило, можуть висловлювати свої думки будь-яким способом, який їм більше подобається, і неможливо застосувати певний набір слів.

Латентний семантичний аналіз (LSA) - це інший підхід до вирішення проблеми, пов’язаної із синтонімією. З появою великомасштабних колекцій текстових даних все частіше використовуються статистичні методи для виявлення взаємозв'язку між термінами та документами. LSA одночасно моделює взаємозв'язок між документами на основі складових слів та взаємозв'язок між словами на основі їх появи в документах. LSA можна розглядати як метод лінійного зменшення розмірності, заснований на

декомпозиції особливих значень матриці термінових документів. Зменшуючи розміри та використовуючи менше розмірів, ніж кількість унікальних слів, LSA викликає подібність між словами. LSA створює семантичний простір, в якому терміни та документи, які тісно пов'язані, розташовані один біля одного. Варто зазначити, що, хоча метод LSA добре розглядає проблему синонімії, він пропонує лише часткове рішення проблеми полісемії. Оскільки значення слова може бути обумовлене іншими словами в документі, LSA надає певну допомогу щодо проблеми полісемії. Однак помилка виникає через те, що кожне окреме слово має лише одну репрезентативну точку в семантичному просторі LSA не більше одного представника.

Сарказм

Ідентифікація сарказму є дуже важким завданням для людини, а ще важчим для машин. Здатність надійно ідентифікувати сарказм у тексті може покращити ефективність багатьох завдань з обробки природних мов, особливо SAOM. Сарказм - це форма вираження, де буквальне значення протилежне задуманому. «Ресторан був чудовий тим, що зробить усі майбутні страви більш смачними», - це приклад саркастичного речення, в якому, хоча технічно не існує негативного терміна в мові, він покликаний передавати негативні настрої. Цей приклад наочно демонструє деякі труднощі у роботі з саркастичними фразами. Для вирішення саркастичної ситуації потрібно добре розуміти контекст, культуру ситуації, тему, людей, а також мову, яка бере участь у саркастичній заяві. Отримати доступ до всіх цих відомостей є складним завданням саме по собі, але спроба використати їх є особливо складною для машини. Хоча явище сарказму широко вивчалось у таких галузях, як психологія, когнітивна наука та лінгвістика, зроблено дуже мало спроб його обчислювального аналізу. Відсутність набору даних із надійно позначеними екземплярами сарказму та несарказму є однією з причин того, що обчислювальний аналіз сарказму дуже молодий. Аналіз сарказму - це сфера, яка потребує більш ретельного дослідження в науковому співтоваристві SAOM.

Складені речення

Складносурядне речення має два самостійні речення або речення. Два незалежні речення можуть об'єднуватися координуючим сполучником (наприклад, "і", "або", "але" і "за") або крапкою з комою. Робота зі складнопідрядними реченнями ускладнює проблему SAOM. Наприклад, такі речення, як "Діти насолоджувались пляжем, але ми ні," або "Незважаючи на приємний досвід, я не можу підтримати багато відгуків, що це був чудовий ресторан", є складними для аналізу настроїв. Робота зі складнопідрядними реченнями все ще залишається відкритою сферою досліджень у SAOM.

Неструктуровані дані

Відгуки, написані авторами для кожного об’єкта, є вхідними даними у форматі простого тексту. Проблемою проблеми є перетворення неструктурованих вхідних даних, які доступні у формі письмових оглядів, у напівструктуровані дані. Напівструктуровані дані - це дані, які не є ані вихідними даними, ані не відповідають офіційній структурі моделей даних, пов'язаних з реляційними базами даних або іншими формами таблиць даних, але тим не менше містять теги або інші маркери для відокремлення семантичних елементів.

Ідентифікація аспекту

Збірник робіт у літературі розглядав проблему SAOM на рівні аспектів у два етапи: перший, ідентифікація аспекту та другий, ідентифікація настрою. Метою ідентифікації аспектів є виявлення конкретних аспектів об'єкта, щодо яких учасники висловлюють свою думку. У літературі є 2 категорії праць, що стосуються ідентифікації аспектів:

1. Автоматичне вилучення: Немає попередніх знань про аспекти, а аспекти автоматично витягуються з даних оглядів. Цю категорію можна розділити на підконтрольні та неконтрольовані підкатегорії.

2. (Напів)вилучення вручну: апріорі відомі або підмножина аспектів, або весь набір бажаних аспектів. У випадках, коли відома підмножина аспектів, підмножина використовується як набір насіння та певним чином розширюється.

Ху і Лю пропонують спочатку знайти найчастіші аспекти за допомогою асоціативного майнінгу, а потім, використовуючи їх, вони витягують рідкісні аспекти. Будь-яке речення, що містить один із частих аспектів, аналізується, щоб з'ясувати рідкісні аспекти.

Інші автори прагнуть представити неконтрольований алгоритм ідентифікації аспектів, який використовує кластеризацію речень із кожним кластером, що представляє аспект. Нарешті, вони запропонували застосувати емпіричну схему зважування до переліку термінів, які відсортовані відповідно до частоти їх появи. Кластеризація речень була використана для того, щоб знайти подібні речення, які, швидше за все, стосуються подібних аспектів. У роботах замість того, щоб представляти речення за допомогою загальноприйнятого методу "Мішок слів" (BOW), вони пропонують використовувати "Мішок іменників", що робить кластеризацію більш ефективною.

В той же час Блер-Гольденсон запропонувати систему узагальнення настроїв для місцевих служб. У їхній системі аспекти поділяються на два типи: динамічні аспекти (на основі рядкових частих іменників / іменних фраз) та статичні аспекти (загальні та грубозернисті), які виділяються розробкою класифікатора для кожного з використанням рукописних речень.

PLSA - це імовірнісна версія LSA, що виникла з неї. PLSA - це модель для колекції документів, в якій кожен документ моделюється як суміш тем. Лу та ін. скористалися PLSA та структурою речень для ідентифікації аспектів. Кожен документ представлений у вигляді пакета фраз, і кожна фраза є парою термінів head і modifier <hi,mi>. Кожен аспект також моделюється як розподіл за загальними термінами. У неструктурованій версії документ розглядається як

сукупність головних термінів (модифікатори ігноруються). У цій моделі журнал ймовірності всіх документів може бути записаний як:

$$logp(D|\Lambda) = \sum_{d=1}^{D} \sum_{w_h \in V_h} \{c(w_h, d) log \sum_{k=1}^{K} [\pi_{d,k} p(w_h|\Theta_k)]\} \quad , \quad (2.3)$$

де V_h - сукупність усіх головних термінів у словниковому запасі, $\pi_{d,k}$ - частка теми k у документі d, $c\ (w_h, d)$ - кількість разів, коли відбувся головний термін w_h у документі d_k , що є k-м аспектом-темою і являє собою набір усіх параметрів моделі.

У структурованій версії (структурована PLSA), оскільки один термін-модифікатор може модифікувати різні терміни заголовка, кожен термін-модифікатор моделюється як суміш K тем. Кожен термін модифікатора w_m представляється як набір головних термінів, які він модифікує, і знову кожен аспект є розподілом за загальними термінами (k). Модифікатор можна розглядати як зразок наступної моделі:

$$p_{d(w_m)}(w_h) = \sum_{k=1}^{K} [\pi_{d(w_m),k} p(w_h)|\Theta_k)] \quad , \quad (2.4)$$

а ймовірність збору модифікаторів V_m становить:

$$logp(V_m|\Lambda) = \sum_{w_m \in V_m} \sum_{w_h \in V_h} \{c(w_h, d(w_m)) log \sum_{k=1}^{K} [\pi_{d(w_m),k} p(w_h|\Theta_k)]\} \quad (2.5)$$

Варто зазначити, що структурований PLSA оцінював параметри на основі співіснування головних термінів на рівні модифікаторів, а не на рівні документів. Оскільки огляди, як правило, є короткими і мають мало фраз, структурований PLSA, як правило, є більш інформативним.

2.2.2 Процес аналізу тексту та основні підходи до ідентифікації настроїв

Ідентифікація настрою - це процес виявлення орієнтації на думку цікавих фрагментів тексту. Орієнтація настрою може бути виражена в різних масштабах. Рейтингові шкали широко використовуються в Інтернеті, намагаючись надати вказівки на думку споживачів щодо продуктів. Багато сайтів, таких як TripAdvisor.com, Amazon.com, Epinions.com, використовують загальну 5-зіркову шкалу оцінок. Користувачі можуть проголосувати за фільми за шкалою оцінок 1–10 на IMDb та за шкалою 5 зірок на сайті rottentomatos.com.

У SAOM боротьба з негативними настроями є особливо складною. Як сказав Лєв Толстой у своїй книзі "Анна Кареніна" "Всі щасливі сім'ї нагадують одна одну, кожна нещасна сім'я нещасна по-своєму", у SAOM всі щасливі почуття майже однакові, кожне нещасне почуття однозначно виражене. У деяких випадках люди вирішують висвітлювати свою критику м’яко, щоб бути ввічливими (навіть в Інтернеті), і в більш широкому сенсі, люди, як правило, проявляють більше творчості в тому, як вони вирішують описувати те, що їм не подобається. Все це робить автоматичне вирішення проблеми складною. Автоматичні методи використовують алгоритми машинного навчання для вирішення проблеми, і ці алгоритми спрямовані на пошук закономірностей на основі даних та спробу навчання на прикладах навчання. Коли недостатньо прикладів навчання або є приклади тренувань з дуже обмеженим розмаїттям, можна уявити, наскільки важким буде вилучення шаблону.

У текстовому документі є 2 типи фрагментів: об’єктивні фрагменти - це ті фрагменти тексту, які виражають фактичні дані, та суб’єктивні фрагменти, що виражають особисті почуття. Хоча як суб'єктивні, так і об'єктивні фрагменти можуть містити думку щодо об'єкта, швидше за все суб'єктивне речення має міркування. Процес ідентифікації настрою можна записати наступними кроками:

1. Витягнути усі міркувальні фрагменти.

2. Визначити почуття кожного фрагмента.

3. Зробити висновок про загальні настрої з думки окремих фрагментів.

Щоб виконати перший крок, необхідно здійснити класифікацію суб’єктивності, тобто визначити, чи є фрагмент тексту суб’єктивним чи об’єктивним. Якщо фрагмент тексту класифікується як суб’єктивний, то дуже ймовірно, що це твердий фрагмент тексту, і тоді потрібно визначити його орієнтацію на думку.

Розумно припустити, що тверді слова та фрази є домінуючими індикаторами часу. Оскільки прикметники, прислівники, а також деякі іменники та дієслова є сильними індикаторами настрою, теги Part-Of-Speech (POS) та синтаксична структура речень також корисні для вилучення непомічених фрагментів. Позначення POS - це процес розмітки слова в тексті (корпусі) як відповідного певній граматичній категорії, такі як іменники, прикметники, прислівники, дієслова тощо. Алгоритми позначення POS базуються на визначеннях термінів, а також на контексті, тобто взаємозв'язку терміна з сусідніми та спорідненими словами та фразами.

У SAOM на рівні аспектів має бути визначена орієнтація настроїв усіх фрагментів тексту, що містить один із вилучених аспектів. Отже, якщо на першому етапі аспекти були визначені, вони можуть бути використані для вилучення сумнівних фрагментів.

Ідентифікація настрою за допомогою неконтрольованих методів

Після того, як на попередньому кроці витягнуто фрагмент думки, слід визначити його орієнтацію. Турні визначив полярність настрою слова, обчислюючи його схожість із двома насіннєвими термінами: "Відмінно" для позитивної та "Погано" для негативної полярності. Подібність між невідомим словом та кожним із початкових термінів обчислюється шляхом підрахунку кількості випадків та спільних випадків їх використання за допомогою

результатів веб-пошукової системи. Залежно від того, який із насіннєвих термінів невідомий термін більше схожий, визначається орієнтація настрою.

Ідентифікація настрою за допомогою контрольованих методів

Ідентифікацію настрою можна легко сформулювати як проблему класифікації. У літературі SAOM використовуються різні класифікаційні методи від машинного навчання для виявлення настроїв. Панг та інші дсолідники експериментували з різноманітними функціями як з наївними класифікаторами Байєса, так і з підтримкою SVM, щоб класифікувати весь вхідний документ (огляди фільмів) як позитивний, так і негативний. Найкращі результати отримали уніграми в моделі частоти на основі присутності, що проходить через SVM. Для класифікації речень сентиментів ефективний і наївний класифікатор Байєса, використовуючи максимізацію очікувань та завантаження з невеликого набору мічених даних до великого набору немічених даних. Розробка особливостей, пошук найбільш підходящого набору ознак для ідентифікації настроїв, є критичним питанням при розробці класифікаторів. У SAOM використовуються різні функції, такі як терміни та їх частота, теги POS, слова та фрази думки, синтаксичні структури та заперечення.

Інший інженерно-технічний підхід, заснований на ієрархічній системі глибокого навчання, який одночасно виявляє аспекти та їх відповідні настрої, був запропонований у роботах дослідників Лакараджу та Менінга. У їх рамках розроблені класифікатори, засновані на глибокому навчанні (рекурсивні нейронні мережі), які спільно передбачають мітки аспектів та настроїв вхідної фрази. Основна ідея полягає у вивченні векторного (або матриці, або тензору) подання слів за допомогою ієрархічного глибокого навчання, яке може пояснити мітки настрою аспекту на рівні фрази. У цій структурі кожен вузол у дереві синтаксичного аналізу даної фрази представлений вектором (або вектором та матрицею), а модель визначається за допомогою синтаксичної структури фрази, кодованої у дереві синтаксичного аналізу. На Рис. 2.6 показано приклад дерева синтаксичного аналізу, в якому кожен вузол пов'язаний d-мірним вектором.

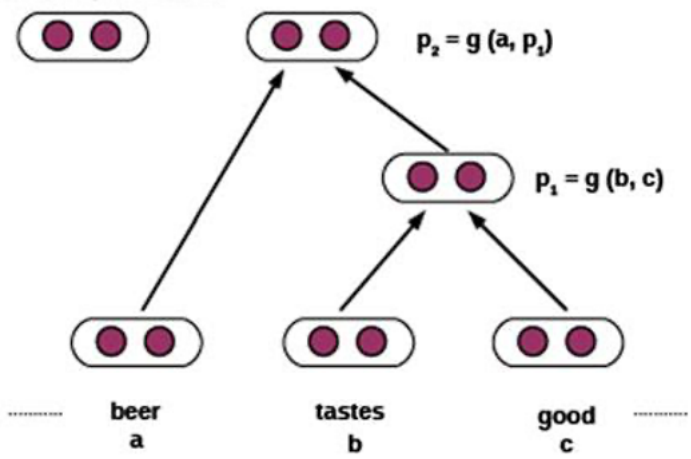

Рис. 2.6. Приклад дерева синтаксичного аналізу [37]

Векторне представлення вузлів обчислюється знизу вгору як:

$$p_1 = f\left(W\begin{bmatrix} b \\ c \end{bmatrix}\right), \qquad p_2 = f\left(W\begin{bmatrix} a \\ p_1 \end{bmatrix}\right), \tag{2.6}$$

де p_1 і p_1 - батьківські вектори, а b і c - листові вузли, а $f = t\,anh$ - стандартна нелінійна функція в елементах.

Метою є вивчення комбінованої матриці $W \in R^{d \times 2d}$, а також композиційного представлення ознак слів (пов'язаних векторів та матриць). У цій моделі мітка класу кожної даної фрази передбачається за допомогою векторного представлення її кореневого вузла дерева синтаксичного аналізу як:

$$y_i = softmax(W_s p_i^{\text{root}}), \tag{2.7}$$

де $W_s \in R^{C \times d}$ - матриця класифікації, і її потрібно оцінити.

Справжньою міткою даної фрази у навчальному наборі є $t_i \in R^{C \times 1}$, що має запис 1 на правильній мітці та 0 на інших індексах. Оскільки аспекти та настрої, призначені для спільного захоплення, мітки повинні передбачатися як пари аспектів і настроїв. Наприклад (Taste, Positive) відповідає одній мітці класу.

Якщо весь набір параметрів моделі позначено як, тоді оцінюється таким чином, що функція softmax векторного подання на кореневому рівні дерева синтаксичного аналізу $y_i \in R^{C \times 1}$ відповідає мітці класу і-го фрагмента тексту так само, як можливо. Цього можна досягти, мінімізуючи похибку поперечної ентропії між y_i і t_i. Функція помилки, яку слід мінімізувати:

$$E(\Theta) = \sum_i \sum_j t_{i,j} log y_{i,j} + \lambda \|\Theta\|^2 \tag{2.8}$$

Цю неопуклу цільову функцію в рівнянні можна звести до мінімуму за допомогою процедур оптимізації Adagrad. Оцінка включає обчислення підградієнтів E (прямий розрахунок векторів і матриць та зворотне поширення похибки softmax у кореневому вузлі).

Ідентифікація настрою за допомогою лексиконів

Багато робіт у літературі визначають орієнтацію настроїв фрагмента, що цікавить, за допомогою деяких лексиконів думок. Підходи, засновані на лексиконі, виявляють орієнтацію на почуття самовпевненої фрази інтересу, шукаючи її з існуючих лексиконів. Процес генерації лексикону, як правило, починається із насіннєвого списку слів думок, які їх орієнтації думок відомі апріорі, а потім за допомогою різних підходів перелік розширюється і до нього додається більше слів думок. Існує 2 основні стратегії розширення початкового списку слів думок: стратегії, засновані на словниках та корпусах. Методи, засновані на словниках, використовують онлайн-словники, такі як WordNet та деяку додаткову інформацію в них (наприклад, їхні глоси), щоб розширити початковий набір насіння. Ці підходи використовують переваги взаємозв'язків синонімів та антонімів між словами, а також деякі методи машинного навчання для створення кращого списку. Сформований лексикон, заснований на словниках в Інтернеті, є незалежним від контексту лексикою думок. Для того, щоб сформувати специфічний для домену лексикон, який може фіксувати слова

думки в певному домені, були запропоновані стратегії, засновані на корпусі. У цій стратегії початковий перелік початкових слів думки розширюється з використанням деяких синтаксичних правил та моделей співіснування. У цьому підході правила розроблені для сполучних термінів, таких як «І», «Або», «Але», «Ні–або», «Ні–ні», і, використовуючи їх, набір насіння розширюється. Використовуючи конкретний корпус інтересів, можна створити специфічні лексикони.

Для ідентифікації настроїв часто дотримуються підходу, заснованого на лексиконі, який передбачає виявлення найближчих прикметників до іменників у нижченаведеному реченні та пошук цих прикметників або їх синонімів у їхніх списках позитивних та негативних прикметників. Блер-Гольденсон спочатку обчислюють єдину оцінку настрою для кожного терміна в лексиконі. Ці бали обчислюються, починаючи з першого насіння з довільними балами, а потім розповсюдження цих балів за допомогою модифікованої версії стандартних алгоритмів розповсюдження міток на графіку. Недоліком цього підходу є те, що це ітераційний алгоритм, і незрозуміло, коли припиняти розповсюдження балів. Використовуючи обчислювані оцінки та враховуючи сусідів кожного речення, вони розробляють класифікатори максимальної ентропії для позитивних та негативних класів. Підходи, засновані на концепціях, використовують веб-онтології або семантичні мережі для здійснення семантичного аналізу тексту. У цьому підході документ представлений мішком понять замість мішка слів. У підході, заснованому на концепції, для визначення настроїв кожного документа використовується словник (SenticNet), який містить афективні позначення понять. Однак незрозуміло, як ідентифікувати аспекти за допомогою представлення пакета концепцій.

2.2.3 Тематичні підходи до ідентифікації настроїв

Фундаментальна характеристика вживання людських слів полягає в тому, що люди використовують найрізноманітніші слова для опису одного і того ж

предмета чи поняття. Як було згадано в пункті 2.2.1 боротьба з синонімією та полісемією природною мовою є складним питанням у багатьох завданнях з аналізу тексту. Це ілюструє необхідність імовірнісних особливостей обчислювального інтелекту для вирішення SAOM. При моделюванні тем кожна тема визначається як розподіл ймовірностей за термінами (терміни розглянутого списку словникових запасів). Одним із результатів роботи алгоритмів моделювання тем є розподіл ймовірностей на терміни для кожної теми, який визначає, які слова мають більшу ймовірність у контексті певної теми. Розумно очікувати, що в кожному контексті існують певні терміни, які мають вищу ймовірність появи, ніж інші терміни. Технічні прийоми моделювання за визначенням здатні охопити цю характеристику природної мови.

Моделі тем - це імовірнісні прийоми, засновані на ієрархічних байєсівських мережах, для виявлення основних тем, що існують у колекції неструктурованих документів. Більшість існуючих у літературі робіт вирішують проблему SAOM у 2 етапи: ідентифікація першого аспекту та ідентифікація другого настрою.

Однією з переваг методів, заснованих на моделюванні тем, є те, що вони здатні знаходити аспекти та настрої одночасно. Крім того, ці алгоритми не вимагають позначених навчальних даних, і вони знаходять теми з аналізу оригінальних текстів.

Прихований розподіл Діріхле (LDA) було запропоновано для того, щоб знайти короткі описи членів колекції. LDA забезпечує ефективну обробку великих колекцій, зберігаючи суттєві статистичні взаємозв'язки, які корисні для основних завдань, таких як класифікація, виявлення новизни, узагальнення та судження про подібність та релевантність. LDA припускає, що кожен документ сформовано із сукупності тем. Тому для кожного документа, що генерується, повинна бути відома частка кожної теми в цьому документі, а також для кожного слова документа, до якої теми це слово належить. Генеративний процес LDA можна узагальнити як:

1. Для кожного документа:

- Випадково обирається розподіл за темами.

2. Для кожного слова в документі:

- Випадково обирається тема з розподілу над темами на кроці 1.
- Випадково обирається слово із відповідного розподілу за словниковим запасом (темою).

Варто згадати, що LDA базується на припущенні, що існує прихована структура, яка створила дану колекцію документів. З огляду на збір документів (спостереження) метою є пошук тем *({β 1, β 2,..., B K})*, розподіл за темами для кожного документа (θ_d) та призначення тем $(z_{d,n})$ для кожного слова $w_{d,n}$ у кожному документі *d.{β,θ,Z}* будують приховану структуру за спостережуваними даними, і тому модель називається прихованим розподілом Діріхле.

Акуратний спосіб описати тематичні моделі - це використання графічних моделей, що забезпечують графічну мову для опису сімейств розподілу ймовірностей. Графічна модель LDA зображена на Рис. 2.7.

У кожній графічній моделі кодується ряд статистичних залежностей, які визначають цю конкретну модель. Наприклад, на Рис. 2.7, ймовірність n-го слова у документі d $(w_{d,n})$ залежить від призначення теми $z_{d,n}$ цього слова та всіх тем $\{\beta\ 1{:}\ K\}$. Завдання теми визначає, яку тему використовувати для отримання ймовірності $w_{d,n}$. Також розподіл тем слів у документі d залежить від розподілу тем $\theta\ d$ у цьому документі. Спільний розподіл прихованої та спостережуваної змінної графічної моделі на Рис. 2.7 можна записати як:

$$p(\boldsymbol{\beta}_{1:K}, \boldsymbol{\theta}_{1:D}, \mathbf{Z}_{1:D}, \mathbf{W}_{1:D}) = \prod_{k=1}^{K} p(\boldsymbol{\beta}_k) \prod_{d=1}^{D} p(\boldsymbol{\theta}_d) \prod_{n=1}^{N} p(z_{d,n}|\boldsymbol{\theta}_d)\, p(w_{d,n}|z_{d,n}, \boldsymbol{\beta}_{1:K}))$$

(2.9)

Метою навчання даної колекції документів $D = \{d1, d2, \ldots, d_D\}$ є вивчити приховану структуру *{β, θ, Z}*. У байєсівських рамках конкретною обчислювальною проблемою для використання моделі є проблема виведення, при якій обчислення апостеріорного розподілу прихованих змінних з

урахуванням спостережуваних змінних представляє інтерес. Точне обчислення апостеріорної ймовірності є нерозв'язним, і повинні бути розглянуті апроксимаційні методи.

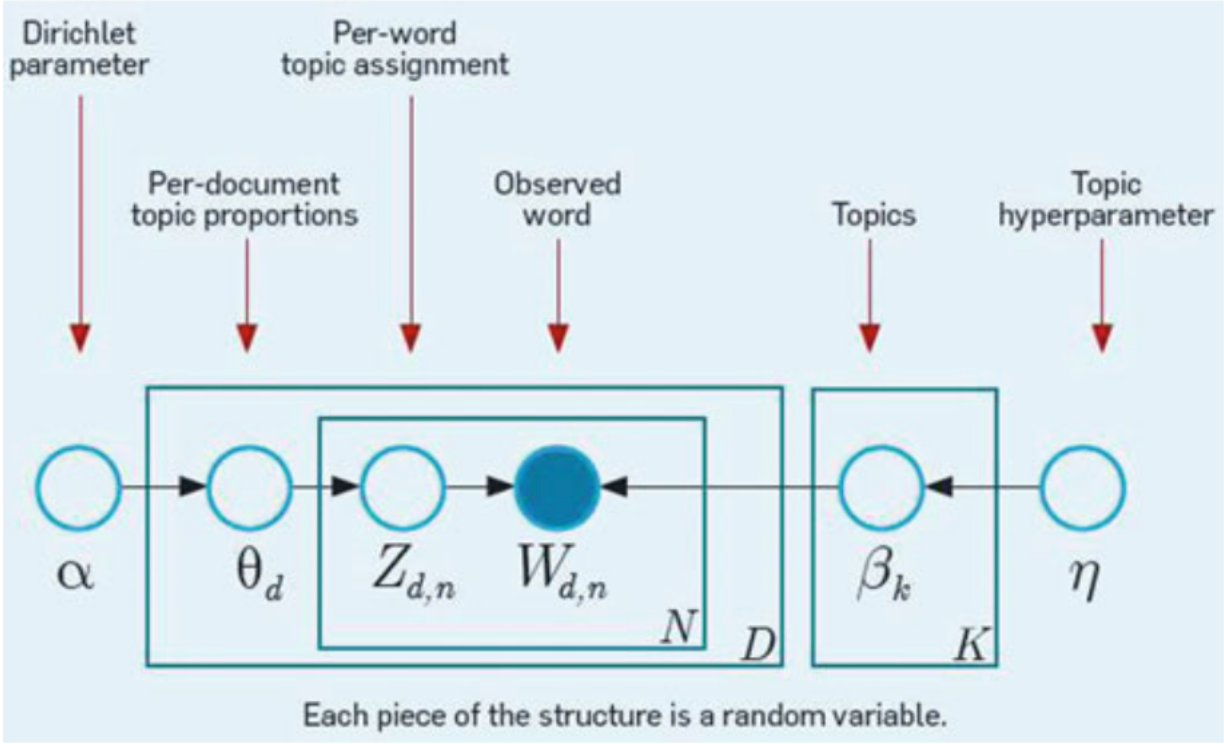

Рис. 2.7. Графічна модель прихованого розподілу Діріхле (затінений вузол є спостережуваною випадковою величиною, а незатінені - прихованими випадковими величинами) [37]

У алгоритмах моделювання тем існує 2 загальних методи наближення:

1. Алгоритми на основі вибірки: Ці алгоритми є методами Монте-Карло марковських ланцюгів (MCMC), які забезпечують принциповий спосіб наближення інтегралу (очікуваного значення). Методи Монте-Карло - це алгоритми, які спрямовані на отримання бажаного значення шляхом виконання моделювання, що включає імовірнісний вибір. Алгоритми, що базуються на вибірці, такі як вибірки Гіббса, намагаються наблизити апостеріорний розподіл, витягуючи з нього зразки без явного обчислення подальшого розподілу. У вибірці Гіббса прихована структура визначає вектор, який представляє стан системи. Основна ідея вибірки Гіббса полягає в тому, що замість вибірки

багатовимірного випадкового вектора відразу, кожен вимір відбирається з використанням зразків інших вимірів.

2. Варіаційні алгоритми: Ці алгоритми апроксимують апостеріорну ймовірність за допомогою параметризованого сімейства розподілу за прихованими змінними та за допомогою методів оптимізації знаходять набір параметрів, який робить наближений розподіл найближчим до точного апостеріорного. Ми можемо покласти варіаційні методи іншими словами: ці алгоритми базуються на нерівності Дженсена і намагаються отримати регульовану нижню межу вірогідності журналу. По суті, розглядається сімейство нижчих меж, індексоване набором варіаційних параметрів. Варіаційні параметри вибираються методом оптимізації, який намагається знайти максимально жорстку нижню межу.

Модель LDA - це найпростіша тематична модель, яка забезпечує потужний інструмент для виявлення прихованої структури у великій колекції документів. З часу впровадження LDA багато в чому розширено та модифіковано. Однією з областей, до якої можуть дуже добре відповідати тематичні моделі, є область SAOM. Зокрема, при аналізі настроїв на рівні аспектів люди говорять про різні аспекти об'єкта. Різні учасники можуть використовувати різні терміни, щоб вказати на один і той же аспект об’єкта. Наприклад, у відгуках про камеру ви можете побачити людей, які коментують якість зображення, і людей, які коментують роздільну здатність камери, і, швидше за все, вони говорять про той самий аспект. Тому, коли люди говорять про певний аспект, найімовірніше буде використаний ряд термінів. У термінології моделювання тем це означає, що кожен аспект має розподіл за термінами або повинен розглядатися як тема в нашій моделі. Крім того, коли учасники хочуть дати певний рейтинг певному аспекту в своєму огляді, конкретний діапазон слів має вищу ймовірність, а для іншого рівня оцінки інший набір слів має більшу масу ймовірності, ніж решта словникового запасу. Це означає, що кожен рейтинг (сентимент), як і кожен

аспект, слід розглядати як тему, щоб розширити алгоритми моделювання тем для SAOM.

Простого збільшення кількості тем (для врахування як аспектних тем, так і рейтингових тем) у LDA недостатньо, щоб застосувати її до SAOM. Дослідники почали модифікувати LDA багатьма способами, щоб адаптувати його для вирішення проблеми аналізу настроїв на рівні аспектів.

Одним із припущень LDA є припущення "Сумка слова" (BOW). У LDA передбачається, що порядком термінів у документі можна знехтувати, і документ може бути представлений як сукупність слів. Оскільки це припущення нереальне, дослідники намагалися усунути цей недолік. Щоб вийти за рамки BOW, була запропонована модель теми біграму. У моделі біграм-тема інтегровані підходи до моделювання документів на основі біграм та теми. У цій моделі кожна тема замість того, щоб бути єдиним розподілом за словами, має кілька розподілів за словами залежно від попереднього слова. Якщо розмір списку словникового запасу становить N, тоді кожна тема характеризується N розподілами, специфічними для цієї теми. Неважко зрозуміти, що простір параметрів у цьому підході буде швидко розширюватися. Модель колокації LDA (LDA-Col) - ще одна спроба вийти за рамки припущення BOW про LDA. У LDA-Col для кожного слова в кожному документі вводиться нова прихована змінна, яка визначає, чи слід слово брати з моделі уніграми чи з моделі біграму з урахуванням попереднього слова. Актуальна модель N-грамів (TNG) дуже схожа на LDA-Col, і єдина відмінність полягає в тому, що в TNG можна вирішити, чи формувати біграму для тих самих двох послідовних лексем слів, залежно від їх сусіднього контексту. Модель TNG автоматично визначає формувати n-грам (і надалі призначати тему) чи ні, виходячи з навколишнього контексту. Приклади тем, знайдених bt TNG, є більш зрозумілими, ніж аналоги LDA.

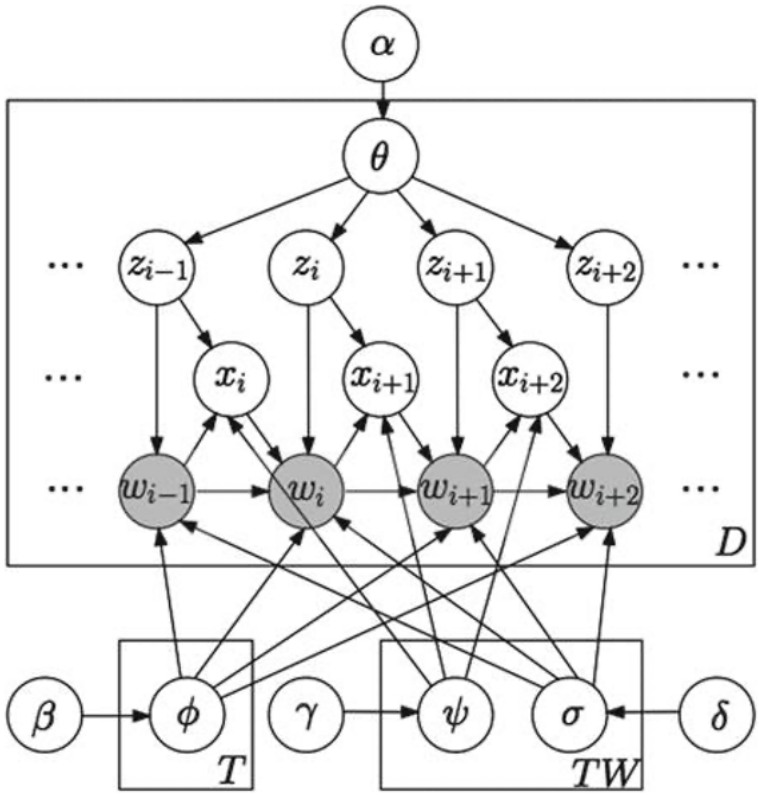

Рис. 2.8. N-грамова графічна модель. Затінений вузол є спостережуваною випадковою величиною, а незатінені - прихованими випадковими величинами [37]

У моделі теми біграму немає додаткової прихованої змінної, яка вирішує, формувати чи ні біграму, ні уніграму (всі терміни взяті з біграмних моделей), у LDA-Col двійкова випадкова змінна x_i для кожного слова w_i визначає уніграму або біграму статус, однак значення хі не залежить від попередньої теми z_{i-1}. Графічна модель LDS-Col точно така ж, як на рис. 2.8, за винятком того, що зв'язку від z_i до х і +1 немає.

У черговій спробі в напрямку послаблення BOW припущення LDA в тому, що воно ігнорує порядок слів, вводячи поняття узгодженості аспектів і сенсів, а також спробував використати існуючу синтаксичну структуру в природних реченнях. У цій моделі (CFACTS) вони розглядали кожен документ як сукупність вікон, і всі слова всередині вікна мають однаковий аспект або тему настрою (узгодженість). Тому, незважаючи на LDA, що кожне слово документа має присвоєння теми, кожне вікно має змінну присвоєння теми або настрою, і всі

слова у цьому вікні мають однакове призначення. У моделі CFACTS кожне слово може бути як аспектним словом, словом настрою, так і фоновим словом, тому вони розглянули нову приховану змінну сі для кожного слова, яка визначає, до якої категорії це слово належить. Синтаксичні залежності серед слів були відображені за допомогою c_{i-1} та сі у моделі.

Оскільки настрої та аспекти для сусідніх вікон можуть все ще залежати для конкретного оглядового документа у багатьох відношеннях, у моделі CFACTS на Рис. 2.9 була введена багаточленна змінна $\psi_{d,\,x}$

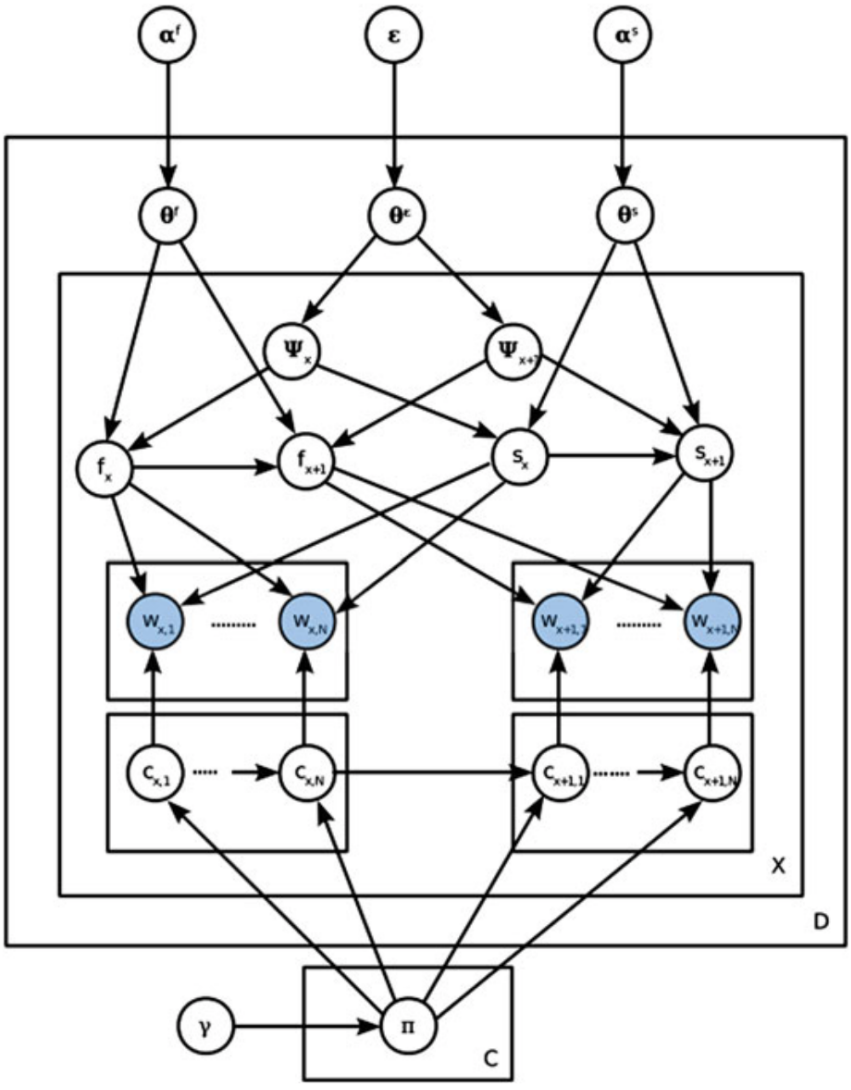

Рис. 2.9. Графічна модель когерентної грані та настроїв [37]

Затінений вузол є спостережуваною випадковою величиною, а незатінені - прихованими випадковими величинами. Ця модель вводить дві нові змінні: $\psi_{d,\,x}$,

яка визначає залежність між вікнами та c_i, яка визначає, до якої категорії належить кожне слово: аспекти, настрої або фонова категорія.

$\varphi_{d,\,x} = 0$: вказує, що як аспект, так і настрої поточного вікна збігаються з попереднім вікном.

$\varphi_{d,\,x} = 1$: вказує, що аспект поточного поточного вікна не залежить від попереднього вікна, але тема настрою поточного вікна така ж, як і попереднє.

$\varphi_{d,\,x} = 2$: вказує, що теми аспектів і настроїв поточного вікна є незалежними від попереднього вікна.

У тому ж рядку, що і ідея когерентності, модель передбачає, що всі слова в одному реченні генеруються з однієї теми та застосовує LDA до кожного речення для вилучення тем.

Підсумовуючи, обчислювальні методи відіграють центральну роль в аналізі настроїв і виявились потужними інструментами, що допомагають зрозуміти сприйняття споживачами продуктів та послуг. Незважаючи на те, що за коротку історію цієї галузі було зроблено багато досягнень, ще багато роботи ще потрібно зробити. Більша частина роботи поки що була зосереджена на розшифровці семантики письмового тексту, і на це дослідження впливають різні лінгвістичні завдання. Тим не менше, ми бачимо, що минулі дослідження змогли запропонувати методи, які розкривають настрої, думки та аспекти і які дуже добре корелюють із оцінками задоволеності споживачів. Що залишається менш зрозумілим, так це узагальнення методів за контекстами або доменами, щоб реально перевірити, наскільки ймовірнісні методи обчислювального інтелекту є узагальненими. Таким чином, можливості для подальших досліджень великі, і цей напрямок досліджень, у свою чергу, призведе до змін у розумінні організаціями своїх клієнтів і, можливо, в тому, як клієнти розуміють та оцінюють товари та послуги.

2.3 Аналіз засобів розробки та впровадження рекомендаційних систем з використанням емоційного аналізу

Для деяких завдань система повинна попередньо обробити текст, потім проіндексувати підготовлений текст для фази машинного навчання і використати класифікатор, який розрізняє методи машинного навчання. Це завдання розробляється архітектором програмного забезпечення. У його завдання також входить проектування системи, вибір програмної платформи, а також пошук заздалегідь контрольованих інструкцій з розвитку. За цю роботу відповідають бібліотеки еквівалентної мови для машинного навчання. Поточний розділ присвячений етапу розробки програмного забезпечення, аналізу декомпозиції системи та аналізу програмних платформ для аналізу текстових тонів.

Визначення тону тексту називається попередньою обробкою тексту. На першому етапі індексації текст повинен бути представлений в машинної формі (число). Існують також публічні бібліотеки, які можна використовувати для перетворення тексту в числовий вектор для цього завдання. Також використовуються різні методи взаємодії з машинами в формі належного управління бібліотекою. Після навчання класифікатор можна використовувати для визначення тональності тексту, який також пройде етапи попередньої обробки і індексації.

Огляд програмних платформ

Популярні програмні платформи включають платформу Oracle Java, платформу Microsoft .NET і безкоштовні мови програмування Python з його швидко прогресуючою екосистемою та R.

В даний час об'єктно-орієнтовані мови і системи (мова R, система Mathcad, пакет додатків MATLAB) використовуються для створення прототипів продуктів з використанням машинного навчання і обчислювальних технологій. Після створення прототипу і дослідження успішні результати зазвичай перекладаються на мови загального призначення для реалізації готового продукту. Java використовується для вирішення різних завдань і займає значну

частину ринку інформаційних технологій. У мові Java є необхідні бібліотеки для обробки природної мови [24, 25]. Однак використання двох програмних платформ для дослідження і подальшої реалізації - не оптимальне рішення. Для платформи .NET Microsoft розробила Azure Studio для машинного навчання [26], в якій в основному використовується мова програмування R. Завдяки призначеному для користувача інтерфейсу в Azure користувач може відносно швидко створити проект. За допомогою інструменту Microsoft можна вирішити основне завдання цієї роботи - створити систему аналізу тону тексту за допомогою машинного навчання [27]. Однак ця програмна платформа має суттєві обмеження для безкоштовних дослідних цілей.

Мова програмування Python і його екосистема розробляються за ліцензіями на програмне забезпечення з відкритим вихідним кодом, які не накладають жодних додаткових комерційних обмежень на використання продукту. Програмна платформа Python вибирається значним числом аналітиків даних в своїх проектах [28]. За допомогою всього лише однієї мови програмування Python можна проводити дослідження і створювати прототипи, а також готові реалізації у вигляді додатків. Дослідницька екосистема мови програмування Python зробила цю платформу дуже привабливою для різних досліджень в галузі інформатики, в тому числі для задач аналізу тону тексту. Каталог пакетів бібліотеки Python містить пакети машинного навчання і обробки природної мови (включаючи обробку української мови).

Огляд бібліотек Python для обробки природної мови

Для реалізації системи емоційного аналізу, а саме підсистеми обробки тексту, необхідно використовувати методи обробки природної мови. Основним пакетом для роботи з природним мовою в мові програмування Python є NLTK (Natural Language Toolkit) [33]. Цей пакет містить більше 50 корпусів текстових і лексичних ресурсів, а також ряд бібліотек для обробки текстів, класифікації, токенізації і маркування. Однак це рішення не підтримує українську мову в деяких випадках, наприклад при морфологічному аналізі тексту.

Для морфологічного аналізу тексту співробітником Scrapinghub був створений морфологічний аналізатор pymorphy2 [34, 35]. За допомогою цього пакета можливо:

• привести слова в звичайну форму (наприклад, слово «люди» в форму «людина»).

• виправляти помилки в словах

• відзнатися граматичну інформацію про слово (число, рід, верхній і нижній регістр, частина мови і т.д).

Описані вище бібліотеки вирішують багато завдань обробки природної мови на етапі попередньої обробки. Наступні етапи аналізу тональності - векторизация оброблюваного тексту і його класифікація.

Огляд пакетів R та порівняння з Python для машинного навчання

Для підготовки тексту до класифікації та векторизації використовуються різні алгоритми. Реалізацію алгоритму word2vec або TF-IDF можна знайти в пакеті gensim, за допомогою якого моделюються векторні простори великих текстових колекцій [36]. Цей пакет надійний і продуктивний.

Для пакетів слів або алгоритмів TF-IDF також можна використовувати універсальний пакет scikit-learn, який був розроблений для інтелектуального аналізу даних [38]. Цей інструмент містить пакети для машинного навчання та попередньої обробки даних для подальшої розробки класифікатора. Алгоритми машинного навчання, реалізовані в пакеті, включають метод опорних векторів (SVM), наївний байєсівский класифікатор, моделі для побудови нейронних мереж і багато іншого. Цей список пакетів охоплює вимоги для застосування машинного навчання в цій роботі.

R - це мова програмування для статистичної обробки даних і графіки, але в той же час це також безкоштовне програмне середовище з відкритим вихідним кодом, яке було розроблено як частина проекту GNU. R використовується всюди, де потрібна обробка даних. Це не просто статистика в строгому сенсі слова, а й «первинний» аналіз (діаграми, таблиці сполучення) і просунуті математичні моделі. Базова обчислювальна потужність R найкраще проявляє себе в

статистичному аналізі і машинному навчанні: від обчислення середніх значень до вейвлет-перетворень часових рядів. Географія використання R широко варіюється. Важко знайти американський або західноєвропейський університет, в якому ви б не працювали з R. Також багато шанованих компаній (наприклад, Boeing) встановлюють R для роботи, тому можна стверджувати, що R є глобальної мовою для статистиків.

Ще одна важлива перевага R - це наявність безлічі розширень або пакетів буквально для будь-якого випадку. При установці R на комп'ютер вже є кілька пакетів: так звані базові пакети, без яких система просто не працює (наприклад, пакет з ім'ям base або пакет grDevices, який управляє виводом діаграм). та інші рекомендовані »пакети (пакет для спеціалізованих кластерів кластерного аналізу, пакет для аналізу нелінійних nlme-моделей і ін.).

Розглянемо плюси і мінуси Python і R, щоб визначити інструмент для подальшого дослідження.

Переваги R:

• Мова була створена спеціально для аналізу даних: запис мовних конструкцій зрозуміла багатьом фахівцям в цій області.

• Багато функцій, необхідні для аналізу даних, є вбудованими мовними функціями. Для перевірки статистичних гіпотез часто потрібно всього кілька рядків коду.

• Гранично спрощена установка IDE (RStudio) і необхідних пакетів обробки даних.

• Зручне сховище пакетів і велика кількість готових тестів практично для всіх методів аналізу даних і машинного навчання.

• Ефективна робота з векторами і матрицями.

• Кілька високоякісних пакетів візуалізації даних для різних завдань (ggplot2, gattice, ggvis, googleVis, rCharts і т. Д.).

Недоліки R:

• Низька продуктивність. Однак в системі є пакети, які можуть збільшувати швидкість (pqR, Renjin, FastR, Riposte і т. Д.). При роботі з великими обсягами даних рекомендується використовувати бібліотеки data.table і dplyr.

• Специфіка в порівнянні зі стандартними мовами програмування, оскільки мова дуже спеціалізована (наприклад, індексування векторів починається з нуля замість одиниці).

• Оскільки більша частина коду на R написана людьми, незнайомими з програмуванням, деякі програми недостатньо читабельні. Крім того, не всі користувачі дотримуються рекомендацій з розробки програмного коду.

• R - відмінний інструмент для статистики та пов'язаних автономних додатків, але він спотикається в областях, де традиційно використовуються мови загального призначення.

• Можна виконувати одні і ті ж функції по-різному, але синтаксис для деяких завдань не зовсім очевидний.

• Через велику кількість бібліотек документація по деяким менш популярним бібліотекам не може вважатися повною.

Переваги Python:

• Універсальна багатоцільова мова: крім обробки даних, також можна використовувати результат обробки в веб-додатку.

Недоліки Python:

• Python - це мова динамічної типізації. Це значно прискорює розробку програми, але в той же час ускладнює пошук помилок, пов'язаних з неправильним призначенням різних даних і, отже, змінних.

На підставі проведеного аналізу виникає питання: чи можна об'єднати переваги мов в одному додатку? Наприклад, має сенс мати можливість викликати бібліотеки R з Python і запускати програми Python безпосередньо з R для статистиків, знайомих з Python. Будь-яка мова може виконувати ці операції зі сторонніми бібліотеками:

• rPython - виконати код Python за допомогою R;

• RPy2 - запускати код з R в версіях Python 2.x і 3.x.

Такі рішення дозволяють не перемикатися з однієї системи на іншу і «склеювати» програми з готових рішень всередині програми, використовуючи сучасні модулі Python і заздалегідь реалізуючи певні пакети з R.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ МЕТОДАМИ МАШИННОГО НАВЧАННЯ З ВИКОРИСТАННЯМ ЕМОЦІЙНОГО АНАЛІЗУ ТЕКСТУ

3.1 Формалізація бази знань для підвищення функціоналу електронної комерції

Для формалізації наших знань при побудові рекомендаційної системи розглянемо в завданні колаборативної фільтрації метод item-based. Вибірка являє собою (u; i; $r_{u,i}$), де u - користувач, i - фільм, $r_{u,i}$ - оцінка, яку користувач u поставив товару i. Також будемо вважати, що рейтинги нормовані на відрізку [0; 1]. Нехай є матриця користувач-ознака, складена з оцінок користувачів, тоді міру схожості товарів *i* і *j* як векторів знайдемо за допомогою коефіцієнта кореляції Пірсона за формулою:

$$¿(i,j)=\frac{\sum_u \left(r_{u,i}-\bar{r}_u\right)*\left(r_{u,j}-\bar{r}_u\right)}{\sqrt{\sum_u \left(r_{u,i}-\bar{r}_u\right)^2}*\sqrt{\sum_u \left(r_{u,j}-\bar{r}_u\right)^2}} \quad (3.1)$$

За U беремо безліч користувачів, які оцінили товари i та j, а r_u – середня оцінка, яку ставить користувач u.

Також скористаємося косинусною мірою схожості, яка розраховується за формулою:

$$¿(i,j)=\frac{\sum_u r_{u,i}*r_{u,j}}{\sqrt{\sum_u r_{u,i}}\sqrt{\sum_u r_{u,j}}} \quad (3.2)$$

Рейтинг для ще неоцінених товарів в методі item-based порахуємо за формулою:

$$\widehat{r_{u,i}} = \frac{\sum_j ¿(i,j) * r_{u,j}}{\sum_j ¿(i,j)} \tag{3.3}$$

3.2 Модель підвищення функціоналу електронної комерції методами емоційного аналізу та машинного навчання

3.2.1 Ініціалізація підвищення функціоналу електронної комерції

Наш набір даних, отриманий від інтернет-магазину одягу, містить 23 тис. відгуків з 10 характеристичними змінними, такими як ідентифікатор товару, категорія, текст, рейтинг і т. д.

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to	5	1	6	General	Tops	Blouses

Рис. 3.1. Огляд структури даних

Ми почали з методології обробки даних та обробки відгуків клієнтів. Ми розбили наш набір даних, розділивши їх за категоріями товарів, підкатегоріями і словами-триггерами, які показують настрої клієнтів, такі як любов, ненависть, фантастика чи жаль і т.д. Це дозволило нам дати набагато більш точну оцінку кожному слову в категоріях і підкатегоріях. Програмну реалізацію виокремлення ключових слів подано в Додатку А.

```
Selected Words
love     8951
great    6117
super    1726
happy     705
glad      614
dtype: int64

Class Names
Dresses     6319
Knits       4843
Blouses     3097
Sweaters    1428
Pants       1388
Name: Class Name, dtype: int64
```

Рис. 3.2. Найбільш використовувані слова в відгуках

Ми припустили, що відгуки з рейтингом 4 або вище оцінюються як позитивні (ми назвали їх True), а рейтинг 2 або нижче – як негативні (ми назвали їх False). Крім того, ми не включили в дослідження нейтральні відгуки, рівні 3.

```
df = df[df['Rating'] != 3]
df['Sentiment'] = df['Rating'] >=4
df.head()
```

	Review Text	Rating	Class Name	Age	Word Counts	Sentiment
0	Absolutely wonderful - silky and sexy and comf...	4	Intimates	33	{'absolutely': 1, 'and': 2, 'comfortable': 1, ...	True
1	Love this dress! it's sooo pretty. i happene...	5	Dresses	34	{'am': 1, 'and': 2, 'bc': 2, 'be': 1, 'below':...	True
3	I love, love, love this jumpsuit. it's fun, fl...	5	Pants	50	{'and': 1, 'but': 1, 'compliments': 1, 'every'...	True
4	This shirt is very flattering to all due to th...	5	Blouses	47	{'adjustable': 1, 'all': 1, 'and': 1, 'any': 1...	True
5	I love tracy reese dresses, but this one is no...	2	Dresses	49	{'0p': 1, 'alterations': 1, 'am': 1, 'and': 4,...	False

Рис. 3.3. Вигляд структури даних після обробки даних

Далі подивимось на розподіл кількості оцінок користувачів. Оскільки ми відфільтрували наші рейтинги, всі користувачі мають не менше 3 оцінок. Проте ми також можемо побачити, що деякі користувачі мають багато оцінок. Це цікаво, тому в наступних кроках ми можемо перевірити, чи є залежність між кількістю оцінок та їхньою якістю.

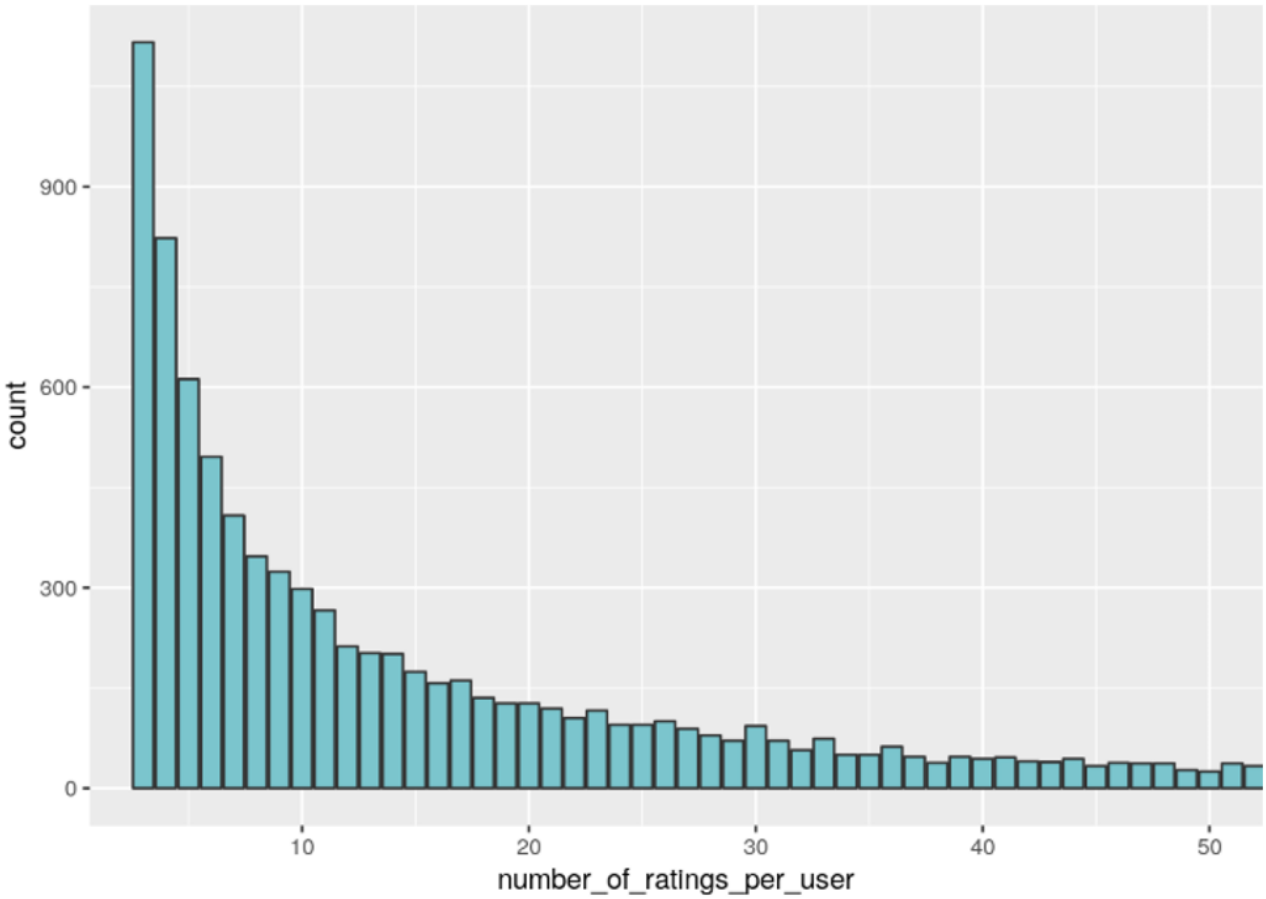

Рис. 3.4. Розподіл кількості оцінок користувачів

Люди мають різні особливості оцінювати товари. Деякі дають рейтинг 5 середній товару, а інші не оцінюють на 5 навіть ідеальні для них товари. Такі тенденції можна побачити на Рис. 3.5. Також ми бачимо велику кількість користувачів з середнім рейтингом 5, що вказує на те, що їм дуже сподобалися всі товари або що вони оцінювали тільки свої улюблені товари. Також очевидним є те, що відсутні користувачі з середнім рейтингом 1 та існує лише незначна кількість користувачів з середнім рейтингом 2 та 3. Ці спостереження важливі для нашого методу колаборативної фільтрації, оскільки вочевидь виникає необхідність подальшої нормалізації оцінок.

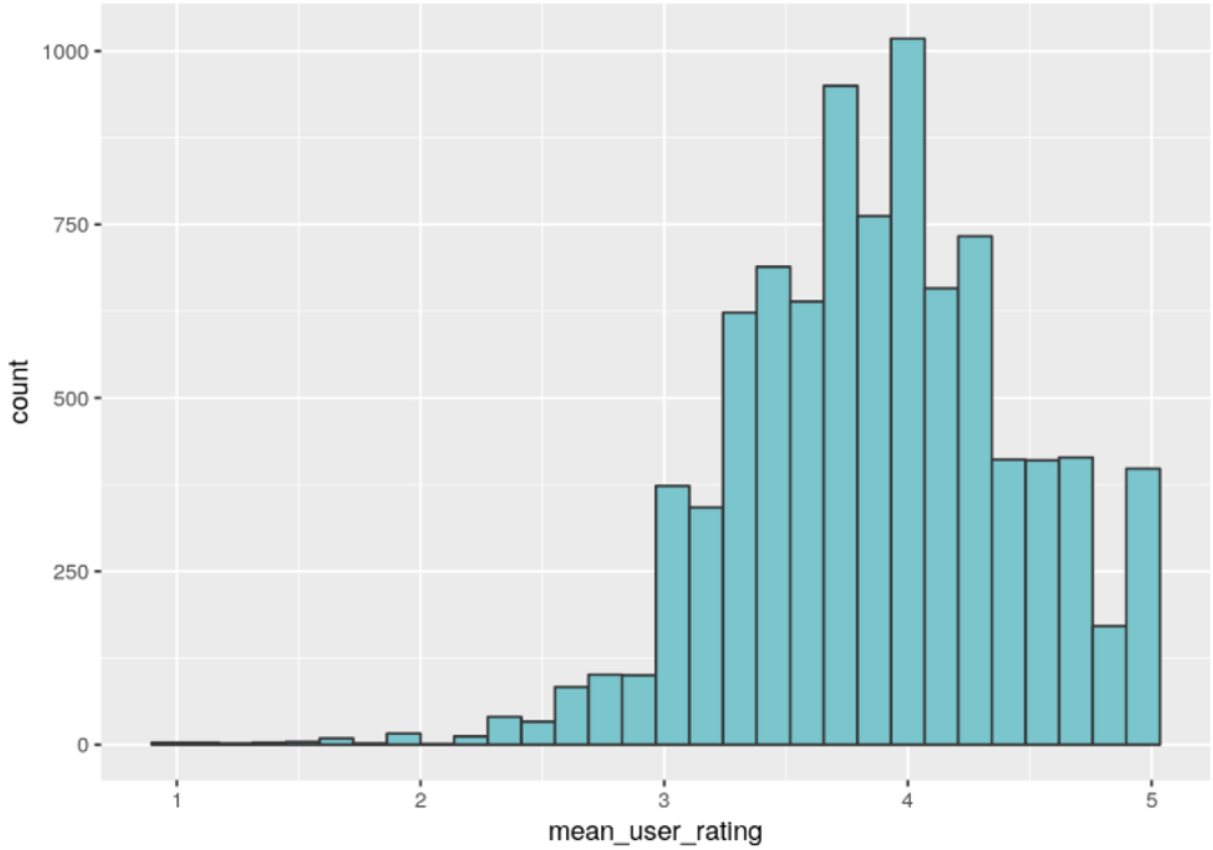

Рис. 3.5. Розподіл середніх оцінок користувачів

З Рис. 3.6. бачимо, що розподіл кількості оцінок на 1 товар є нормальним з середніми значеннями 17-23 відгуки.

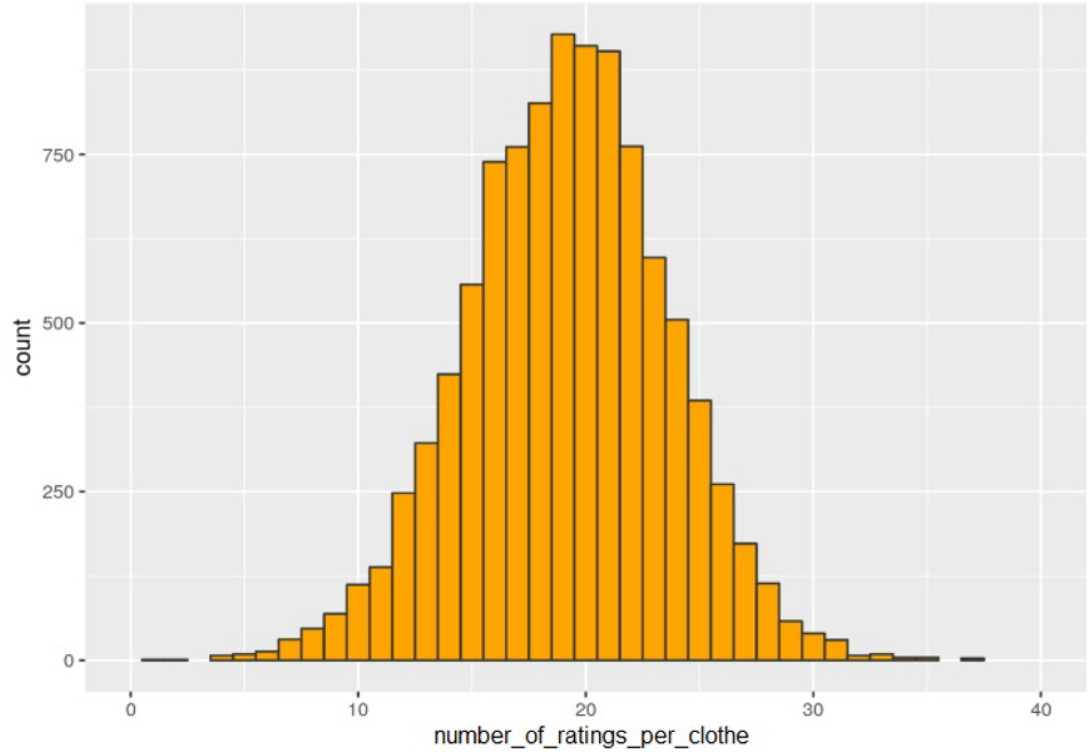

Рис. 3.6. Розподіл кількості оцінок на 1 товар

Аналогічна ситуація і з розподілом середніх оцінок товарам, де 4 – найпопулярніша оцінка товару.

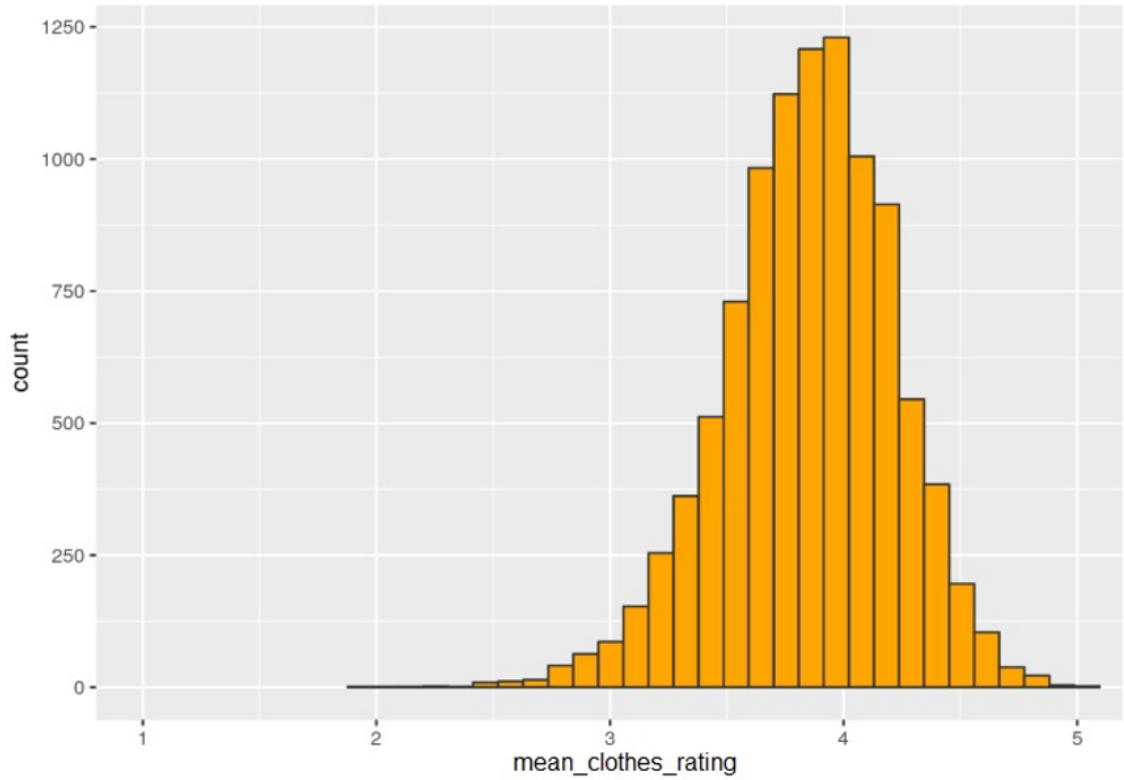

Рис. 3.7. Розподіл середньої оцінки товарів

Наступним кроком ми перевіряли гіпотезу про залежність між кількістю відгуків та їх якістю. Теоретично, могло бути, що популярність товару (з точки зору кількості отриманих рейтингів) пов'язана зі середнім рейтингом, який він отримує, і як наслідок, коли товар стає популярним, він краще оцінюється. Однак наші дані показують, що це справедливо лише в дуже малій мірі. Кореляція між цими змінними становить лише 0,045.

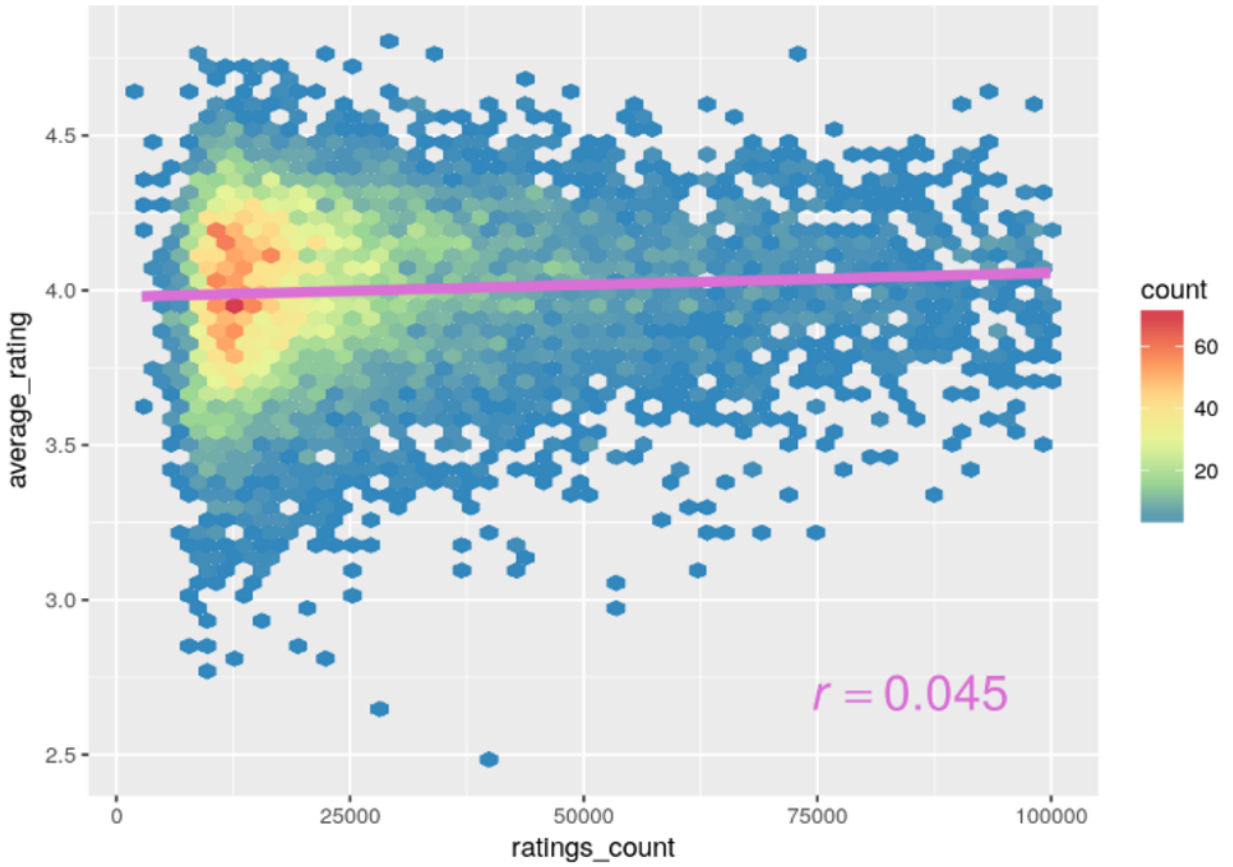

Рис. 3.8. Залежність між кількістю оцінок та її величиною

Далі було перевірено, чи отримують товари, які є частиною серії, вищий рейтинг. І насправді, чим більше обсяг серії, тим вищий середній рейтинг. Під серією мається на увазі наявність кількох кольорів (3 і більше), розмірів та інших параметрів, які можуть диференціювати товари в одному наборі.

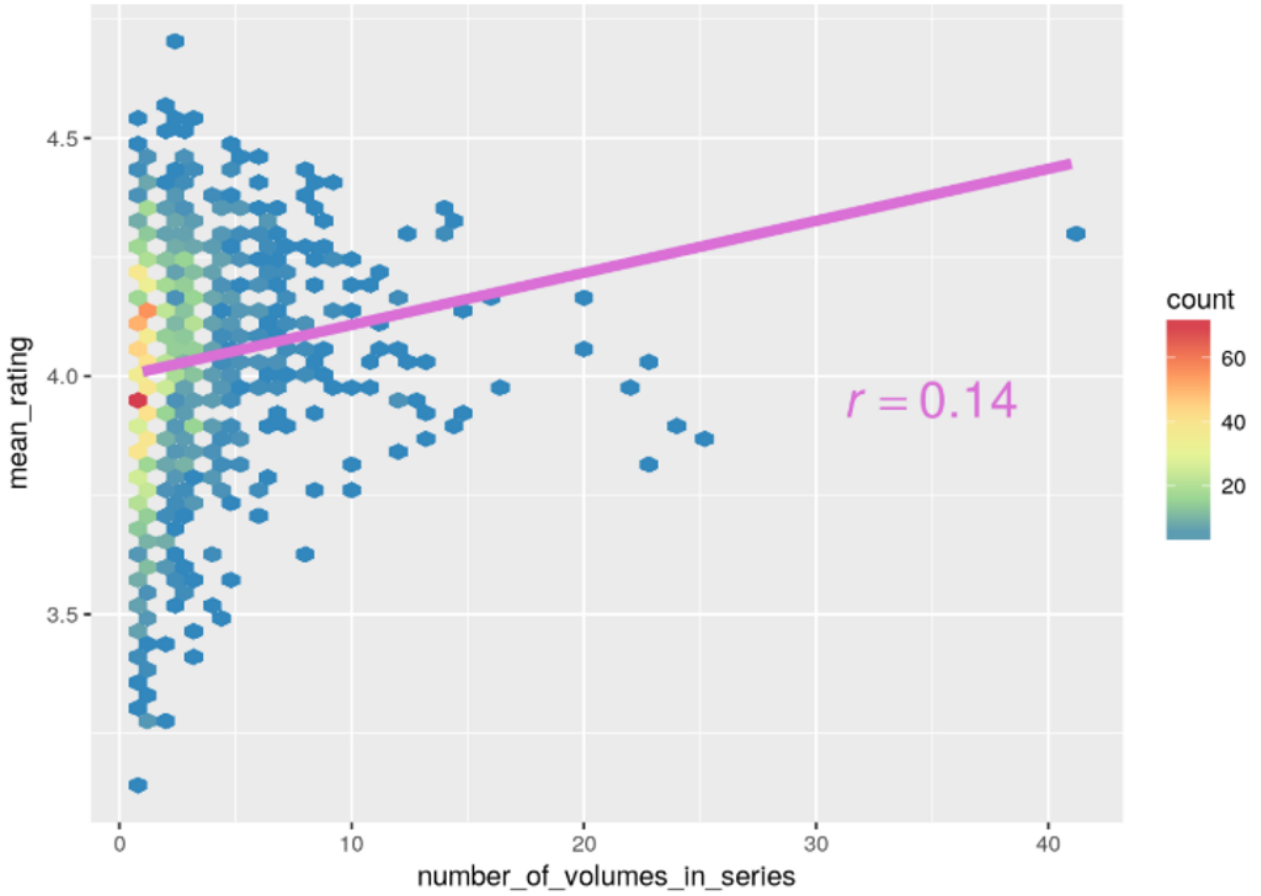

Рис. 3.9. Залежність між середньою оцінкою та кількості товарів в серії

3.2.2 Навчання моделі підвищення функціоналу електронної комерції

Наступним кроком є розподілення даних на тестову і навчальну вибірку і почали будувати нашу класифікаційну модель, використовуючи різні методи, серед яких були логістична регресія, наївний байєсівський класифікатор, метод опорних векторів і нейронна мережа. Програмну реалізацію методів класифікації подано в Додатку Б.

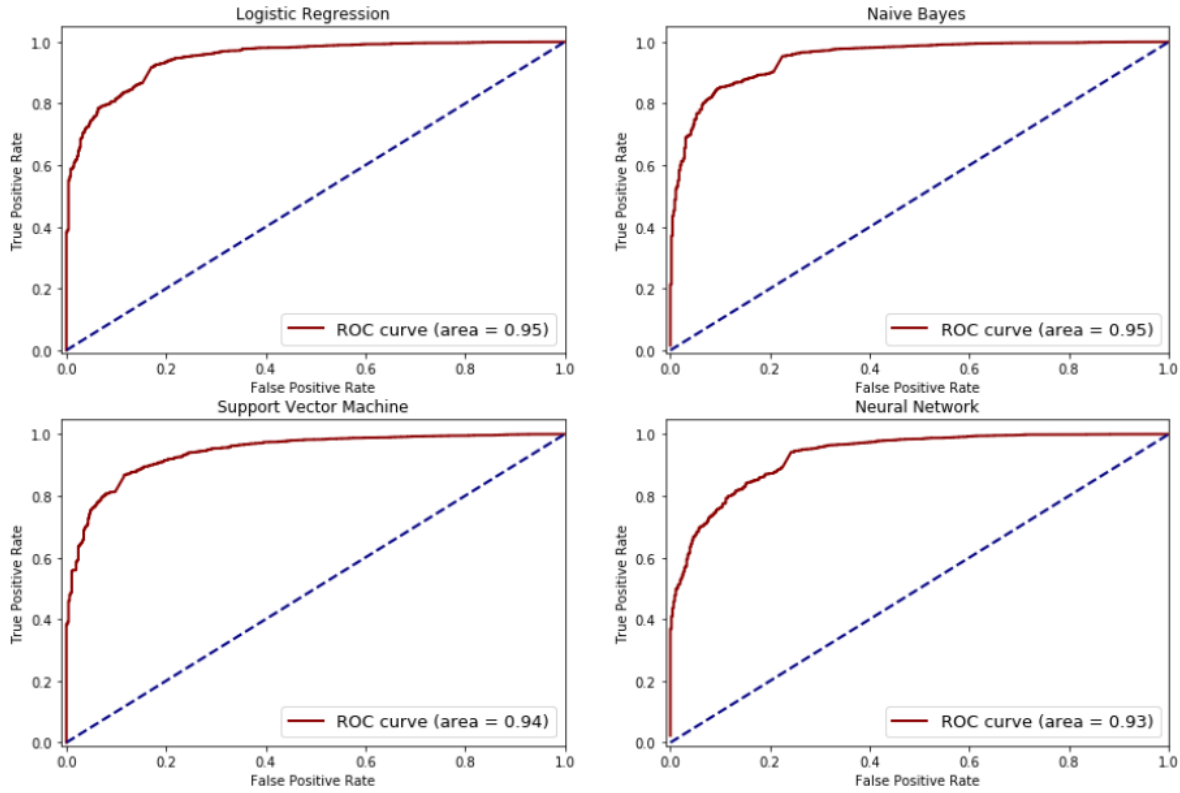

Рис. 3.10. Порівняння результати алгоритмів класифікації

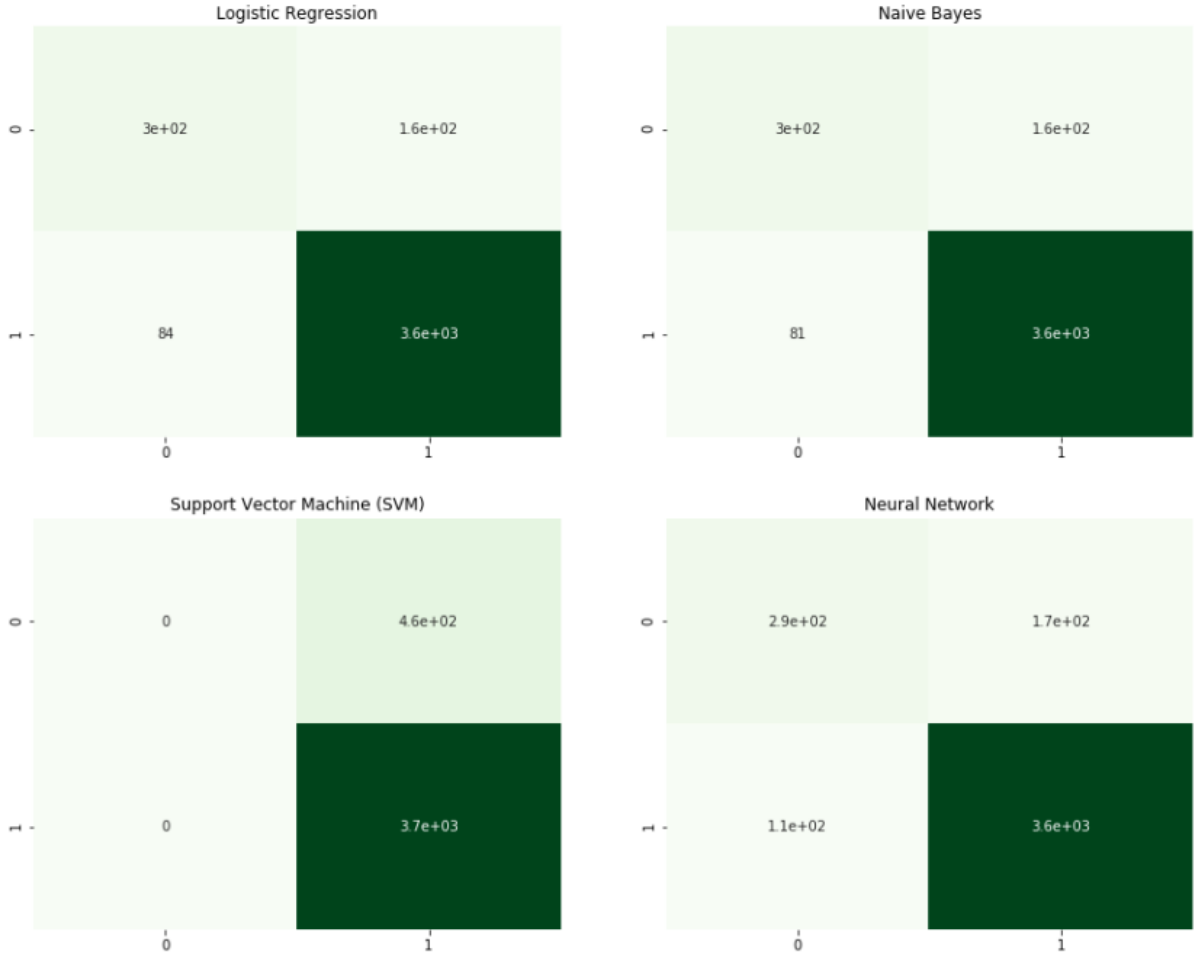

Рис. 3.11. Матриця похибок класифікаційних моделей

Ми дійшли до висновку, що наївний байєсівський класифікатор і логістична регресія дають найкращі результати (площа під ROC кривої = 0,95). Таким чином, обидва вони дуже ефективні в прогнозуванні настроїв. З іншого боку, наївний байєсівський процес займає менше часу, і коли у нас буде більший набір даних, ця різниця може збільшитися і стати важливою перевагою. Програмну реалізацію підрахунку та виведення результатів порівняння алгоритмів класифікації подано в Додатку В.

3.2.3 Проектування та тестування моделі підвищення функціоналу електронної комерції

Після вибору кращої моделі класифікації, ми використовували колаборативну фільтрацію в якості стандартного методу для рекомендацій по продукту. Ми ідентифікували інших клієнтів, схожих на поточного клієнта, з точки зору їх рейтингів тим ж товарам, використовуючи кореляцію Пірсона. Після цього ми взяли у них рейтинг товарів, які поточний покупець ще не купив. Останнім кроком системи є рекомендація користувачеві товарів з найвищим середнім рейтингом.

Зобразимо процес поетапно:

Крок 1. Пошук схожих користувачів. Для цього кроку ми вибираємо користувачів, які мають спільну оцінку тих самих товарів. Наприклад, оберемо користувача з номером 17329. Спочатку ми вибираємо користувачів, які оцінили принаймні один товар, яку також оцінив цей користувач. Загалом, 440 користувачів, які мають принаймні один спільний товар.

```
current_user <- "17329"
rated_items <- which(!is.na((as.data.frame(ratingmat[current_user, ]))))
selected_users <- names(which(apply(!is.na(ratingmat[ ,rated_items]), 1, sum)
>= 2))
head(selected_users, 40)
## [1] "35"   "153"  "158"  "202"  "343"  "368"  "958"  "1169" "1185" "1339"
```

[11] "1449" "1456" "1464" "1518" "1571" "1634" "1677" "1759" "2166" "2218"

[21] "2347" "2421" "2467" "2619" "3050" "3075" "3246" "3263" "3399" "3580"

[31] "3641" "3662" "3757" "3796" "4005" "4204" "4242" "4276" "4289" "4489"

Рис. 3.12. Програмна реалізація пошуку користувачів зі спільними товарами

Для цих користувачів ми можемо розрахувати подібність їх рейтингів з рейтингами користувача 17329. Існує ряд варіантів обчислення подібності, серед яких косинуси подібності та коефіцієнт кореляції Пірсона, який ми обрали. Тепер ми переходимо до всіх вибраних користувачів і обчислюємо подібність між їхніми рейтингами та рейтингами користувача 17329. Наведемо приклад для розрахунку кореляції Пірсона 2 користувачів (user_ids: 1339 і 21877). Ми бачимо, що подібність вища для користувача 1339, ніж для користувача 21877.

```
user1 <- data.frame(item=colnames(ratingmat),rating=ratingmat[current_user,]) %>% filter(!is.na(rating))
user2 <- data.frame(item=colnames(ratingmat),rating=ratingmat["1339",]) %>% filter(!is.na(rating))
tmp<-merge(user1, user2, by="item")
tmp
##   item rating.x rating.y
## 1 1258       4       5
## 2 1662       4       5
## 3 1757       3       3
cor(tmp$rating.x, tmp$rating.y, use="pairwise.complete.obs")
## [1] 1
user2 <- data.frame(item = colnames(ratingmat), rating = ratingmat["21877", ]) %>% filter(!is.na(rating))
tmp <- merge(user1, user2, by="item")
tmp
##   item rating.x rating.y
## 1  105       4       3
## 2 1365       3       4
## 3 1584       1       5
## 4  318       4       4
cor(tmp$rating.x, tmp$rating.y, use="pairwise.complete.obs")
## [1] -0.8660254
```

Рис. 3.13. Програмне обчислення кореляції Пірсона

Наступний виклик – нормалізація оцінок. Оскільки всі клієнти оцінюють товари по-різному – одні користувачі оцінюють більшість товарів в 5 балів, а для інших оцінка 4 є максимальною, тому перед розрахунком дані необхідно нормалізувати. Отже, передбачену оцінку потім потрібно буде перевести в вихідну шкалу зворотним перетворенням (і, якщо потрібно, округлити до найближчого цілого числа).

Ми розглядали декілька способів нормалізації:

• центруванням (mean-centering) – з оцінок користувача просто віднімаємо його середню оцінку,

•стандартизацією (z-score) – додатково до центрування ділимо оцінку її на стандартне відхилення у користувача,

• подвійною стандартизацією – перший раз нормуємо оцінки користувача, другий раз – оцінки товару.

Ми вибрали симбіоз другого і третього методу, тому що ця комбінація дає найнадійніші результати на середніх за обсягом вибірках.

```
rmat <- ratingmat[selected_users, ]
user_mean_ratings <- rowMeans(rmat,na.rm=T)
rmat <- rmat – user_mean_ratings – rmse (actual)
```

Рис. 3.14. Програмна реалізація нормалізації оцінок

Схожість між користувачами ми візуалізували за допомогою графічного пакету qpraph, де ширина ребер графа відповідає подібності між клієнтами.

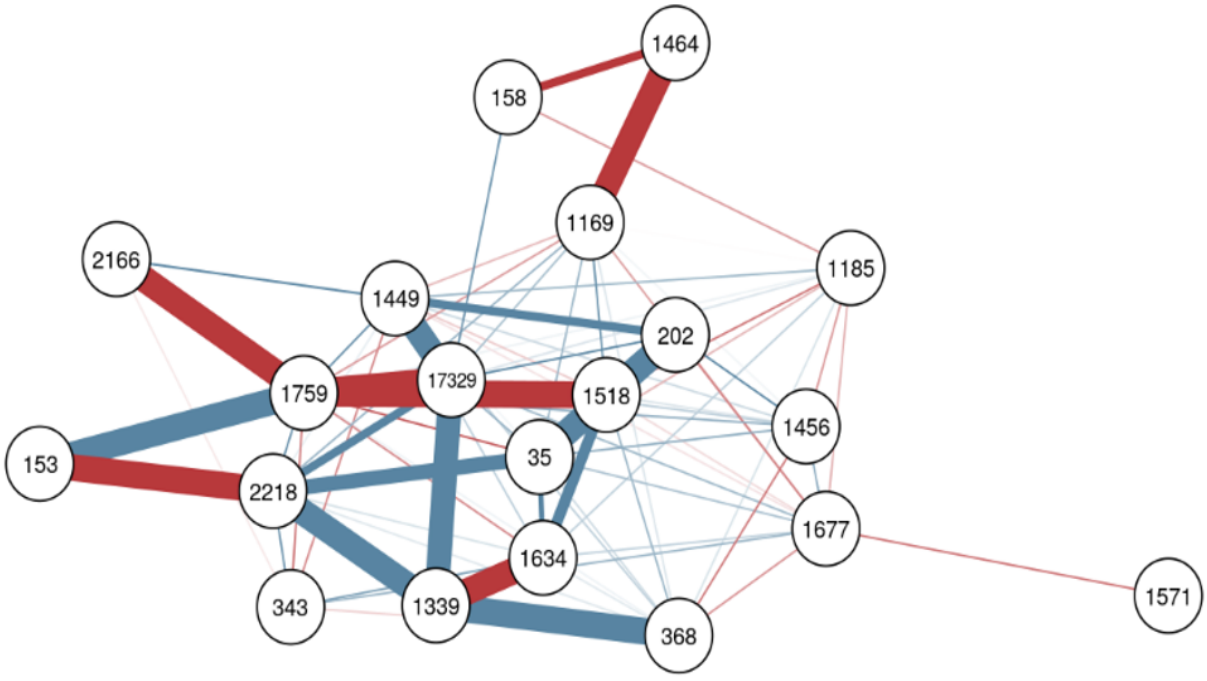

Рис. 3.15. Побудова графу для частини користувачів вибірки

Головна проблема алгоритму полягає у випадку, якщо середній рейтинг пораховано за оцінками всього декількох користувачів, така оцінка явно не буде

достовірною. Перший спосіб вирішення – показувати не середнє значення, а згладжене середнє (Damped Mean). Ідея наступна: при малій кількості оцінок відображуваний рейтинг більше тяжіє до безпечного «середнього» показнику, а як тільки набирається достатня кількість нових оцінок, «усереднюване» коригування перестає діяти. Інший підхід – розраховувати по кожному рейтингу довірчі інтервали. Математично, чим більше оцінок, тим менше варіація середнього і, отже, більше впевненість в його правильності. А в якості рейтингу можна виводити, наприклад, нижню межу інтервалу (Low CI Bound). При цьому зрозуміло, що така система буде досить консервативною, з тенденцією до заниження оцінок з нових товарів (якщо, звичайно, це не хіт) тому ми вибрали перший варіант.

Крок 2. Здійснення прогнозу для інших товарів.

Щоб отримати рекомендації для нашого користувача, ми візьмемо найбільш схожих на нього користувачів, отриманих з кроку 1 і середні рейтинги товарів, які користувач 17329 ще придбав. Щоб зробити ці середні показники більш надійними, за умови наявності великої бази даних можна включати лише елементи, які оцінено кількома іншими схожими користувачами.

```
similar_users <- names(res[1:4])
similar_users_ratings <- data.frame(item = rep(colnames(rmat), length(similar_users)), rating = c(t(as.data.frame(rmat[similar_users,])))) %>% filter(!is.na(rating))
current_user_ratings <- data.frame(item = colnames(rmat), rating = rmat[current_user,]) %>% filter(!is.na(rating))
predictions <- similar_users_ratings %>%
  filter(!(item %in% current_user_ratings$item)) %>%
  group_by(item) %>% summarize(mean_rating = mean(rating))
predictions %>%
  datatable(class = "nowrap hover row-border", options = list(dom = 't',scrollX = TRUE, autoWidth = TRUE))
```

	item	mean_rating
1	1005	1.52884615384615
2	1009	-0.471153846153846
3	1031	1.75438596491228
4	1041	-0.245614035087719
5	1042	0.141509433962264
6	1045	-0.105882352941177
7	1062	-0.105882352941177
8	1068	0.528846153846154
9	1072	-1.10588235294118
10	1073	0.141509433962264

Рис. 3.16. Програмна реалізація товару для рекомендації

Крок 3. Надання найкращих рекомендацій користувачу.

Враховуючи результати з Рис. 3.16, ми відсортували прогнози щодо їхнього середнього рейтингу і рекомендували користувачу товари з найвищим нормалізованим рейтингом. У нашому випадку це були товари з номером 1031, 2004, 3934, 5524, 7239.

```
predictions %>%
 arrange(-mean_rating) %>%
 top_n(5, wt = mean_rating) %>%
 mutate(item_id = as.numeric(as.character(item))) %>%
 left_join(select(items, authors, title, item_id), by = "item_id") %>%
 select(-item) %>%
 datatable(class = "nowrap hover row-border", options = list(dom = 't',scrollX = TRUE, autoWidth = TRUE))
```

Рис. 3.17. Програмна реалізація надання 5 кращих рекомендацій користувачу.

Крок 4. Оцінка якості рекомендацій.

Після здійснення пронозу та рекомендацій важливо перевірити їхню якість, тому ми оцінюємо точність моделі шляхом крос-валідації і взявши RMSE в якості ключового показника. Крім того, ми варіювали кількість клієнтів, на основі оцінок яких модель робить прогноз.

```
scheme <- evaluationScheme(real_ratings[1:500,], method = "cross-validation", k = 10, given = -1, goodRating = 5)
algorithms <- list("random" = list(name = "RANDOM", param = NULL),
                   "UBCF_05" = list(name = "UBCF", param = list(nn = 5)),
                   "UBCF_10" = list(name = "UBCF", param = list(nn = 10)),
                   "UBCF_30" = list(name = "UBCF", param = list(nn = 30)),
                   "UBCF_50" = list(name = "UBCF", param = list(nn = 50))
                   )
results <- evaluate(scheme, algorithms, type = "ratings")
```

Рис. 3.18. Програмна реалізація перевірки якості моделі

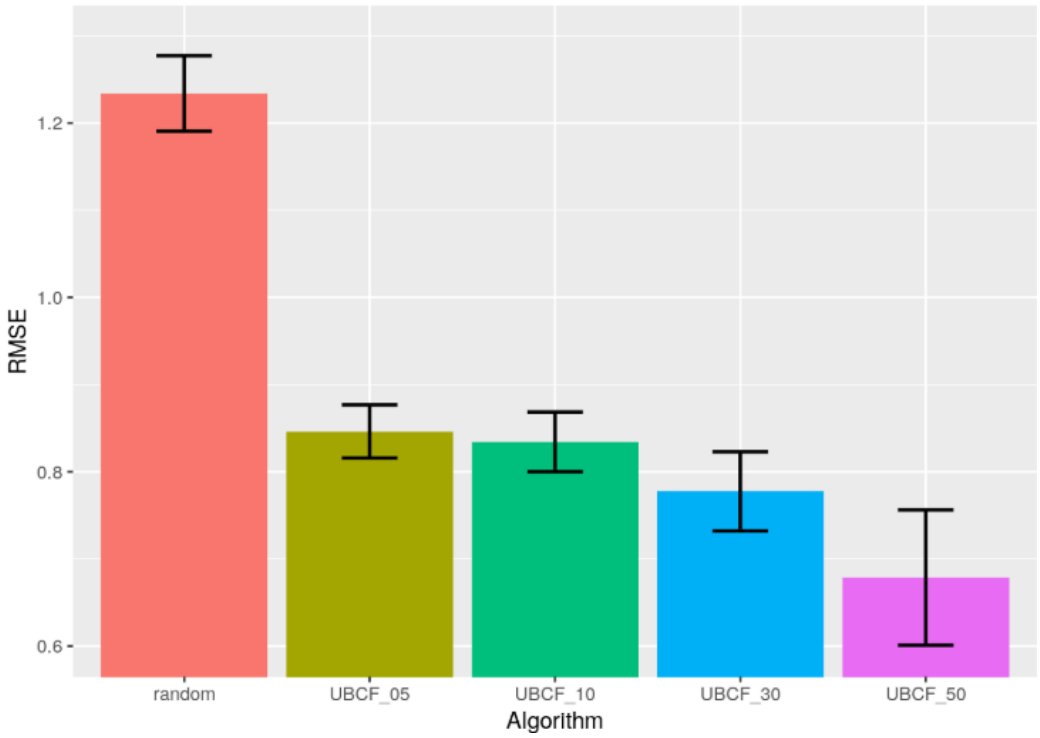

Рис. 3.19. Похибки RMSE рекомендацій залежно від кількісті вершин графу

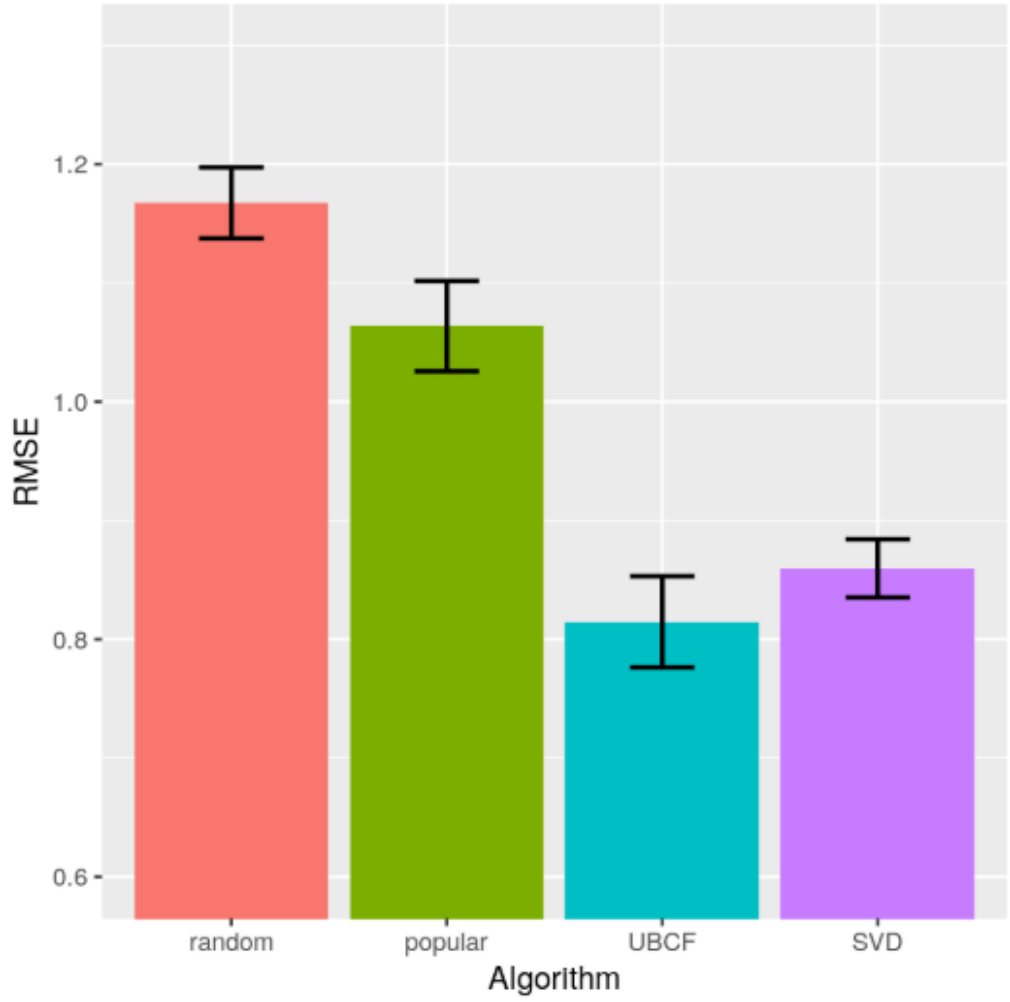

Рис. 3.20. Порівняння точності рекомендацій між різними алгоритмами

По-перше, ми бачимо, що алгоритм працює набагато краще, ніж випадкові пропозиції. По-друге, ми прийшли до висновку, що RMSE зменшується зі збільшенням числа найближчих сусідів, тому ми рекомендуємо будувати прогноз на основі не менше 30 клієнтів зі спільними товарами.

Незважаючи на високу точність, у алгоритму є недолік: його складне застосування через квадратичну складність. Дійсно, як будь-який метод найближчого сусіда, він вимагає розрахунку всіх попарних відстаней між користувачами (а користувачів можуть бути мільйони). Дану проблему ми вирішили шляхом технічного коригування алгоритму:

• оновлювати відстані не після кожної покупки, а пакетами (наприклад, раз в день),

• не перераховувати матрицю відстаней повністю, а оновлювати її інкрементально.

У фінальному вигляді ми отримуємо нашу рекомендаційну систему у вигляді Python-коду, який імплементується в back-end код сайту магазину і готовий до використання.

ВИСНОВКИ

У даній кваліфікаційній роботі розглядаються та аналізуються принципи емоційного аналізу тексту і системи рекомендацій, висвітлюються проблеми і завдання, з якими стикаються їх розробники. Оглянуто програмні засоби, що використовуються для інтелектуального аналізу даних, економіко-математичного моделювання і машинного навчання. Для аналізу емоційного забарвлення відповідей та відгуків і створення системи рекомендацій були побудовані класифікаційні моделі з використанням методів наївного байєсівського класифікатора, нейронних мереж, логістичної регресії і методу опорних векторів. Ці моделі точно визначають оцінку відповіді, а індивідуалізований метод колаборативної фільтрації з використанням нормалізації і попередньої корекції оцінок дозволяє надавати ефективні рекомендації з невеликими помилками. Крім того, було запропоновано вирішення проблеми квадратичної складності стандартного алгоритму колаборативної фільтрації. Результатом моделювання є автоматизована рекомендаційна система з модулем емоційного аналізу тексту. Якість побудованих моделей було проаналізовано з використанням кривих ROC і RMSE.

Для розробки моделі була використана база даних, отримана з інтернет-магазину. Оскільки всі моделі навчаються і тестуються на реальних даних, існує безліч способів застосування цього методу на практиці, і пропонується, щоб результати дослідження використовувалися розробниками програмного забезпечення в якості методологічного матеріалу при проектуванні і розробці систем рекомендацій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Николаев И. С. Прикладная и компьютерная лингвистика / И. С. Николаев, О. В. Митренина, под ред. Т. М. Ландо – 2-е изд. – М. : Ленанд, 2016. – 316 с.

2. Bing Liu Handbook Of Natural Language Processing – Second Edition – Chapman & Hall/CRC, 2010. – 702 с.

3. Автоматическая обработка текстов на естественном языке и анализ данных / Е. И. Большакова, К. В. Воронцов, Н. Э. Ефремова и др. – Изд-во НИУ ВШЭ Москва, 2017. – 269 с.

4. Natural Language Processing // Wikipedia – the free encyclopedia – https://en.wikipedia.org/wiki/Natural-language_processing

5. Bing Liu Sentiment analysis and opinion mining. (Synthesis Lectures on Human Language Technologies) – Morgan & Claypool Publishers – 2012. – 184 с.

6. Emma Haddi The Role of Text Pre-processing in Sentiment Analysis / Emma Haddi, Xiaohui Liu, Yong Shi // Procedia Computer Science. 2013, № 17.

C. 26-32.

7. Bo Pang Opinion Mining and Sentiment Analysis / Bo Pang, Lillian Lee // Foundations and Trends® in Information Retrieval: Vol. 2: No. 1–2, 2008 – С. 1–135.

8. Surbhi Garg Study of Sentiment Classification Techniques / Surbhi Garg, Murthal Neetu // International Journal of Innovations & Advancement in Computer Science Volume 7, 2018. – С. 241–247.

9. LINIS CROWD // Общедоступный тональный словарь и краудсорсинговая платформа для его создания – http://linis-crowd.org/

10. Rice, D. R. Corpus-based dictionaries for sentiment analysis of specialized vocabularies / Rice, D. R., Zorn, C. // Proceedings of NDATAD, 2013.

– С. 98–115.

11. Taras Zagibalov Comparable english-russian book review corpora for sentiment analysis / Taras Zagibalov, Katerina Belyatskaya, John Carroll // In Computational Approaches to Subjectivity and Sentiment Analysis, 2010. – С. 67– 72.

12. Meng X. Lost in translations? building sentiment lexicons using context based machine translation / Meng X., Wei F. [и др.] // COLING, 2012. – С. 829–838.

13. Анна Пазельская Метод определения эмоций в текстах на русском языке / Анна Пазельская, Алексей Соловьев // The international conference on computational linguistics and intellectual technologies “Dialogue 2011” : конференция. – Москва, 2011. – С. 510–522.

14. Анализ тональности текста // Википедия – свободная энциклопедия. https://ru.wikipedia.org/wiki/Анализ_тональности_текста/

15. Bhumika Gupta Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python / Bhumika Gupta, Monika Negi [и др.] // International Journal of Computer Applications, Volume 165 – No.9, 2017. – С. 29–34.

16. Клековкина М. В. Метод автоматической классификации текстов по тональности, основанный на словаре эмоциональной лексики (рус.) / М. В. Клековкина, Е.В. Котельников // RCDL-2012, Переславль-Залесский, Россия : конференция. – 2012.

17. Muqtar Unnisa, Opinion mining on twitter data using unsupervised learning technique / Muqtar Unnisa, Ayesha Ameen, Syed Raziuddin // International Journal of Computer Applications 148(12), 2016. – С. 12–19.

18. He Yulan, Self-training from labeled features for sentiment analysis / He Yulan, Zhou Deyu // Information Processing & Management, Volume 47, 2011. – С. 606–616.

19. Ortigosa-Hernández Jonathan, Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers / Ortigosa-Hernández Jonathan, Rodríguez Juan Diego [и др.] // Neurocomputing, 2012. – С. 98–115.

20. Support vector machine // Wikipedia, the free encyclopedia. – https://en.wikipedia.org/wiki/Support_vector_machine

21. Котельников Е. В., Автоматический анализ тональности текстов на основе методов машинного обучения / Котельников Е. В., Клековкина М.В. // Компьютерная лингвистика и интеллектуальные технологии. По материалам ежегодной Международной конференции «Диалог» (2012), том 2, 2012, С. 27-36.

22. Осокин В. В., Анализ тональности русскоязычного текста / Осокин В. В., Шегай М. В. // Интеллектуальные системы. Теория и приложения, Том 18, Вып. №3, 2014. – С. 163-174.

23. Sanjiv D., Yahoo! for Amazon: Extracting market sentiment from stock message boards / Sanjiv D., Chen M. // Proceedings of the Asia Pacific finance association annual conference (APFA), 2001.

24. Stanford CoreNLP – Natural Language software // CoreNLP. – https://stanfordnlp.github.io/CoreNLP/

25. About Apache OpenNLP // OpenNLP – https://opennlp.apache.org/

26. Студия машинного обучения Azure // Microsoft Azure. – https://azure.microsoft.com/ru-ru/services/machine-learning-studio/

27. Create text analytics models in Azure Machine Learning Studio // Microsoft Azure. – https://docs.microsoft.com/en-us/azure/machine-learning/studio/text-analytics-module-tutorial/

28. R, Python Duel As Top Analytics, Data Science software // KDnuggets 2016 Software Poll Results. – https://www.kdnuggets.com/2016/06/r-python-top-analytics-data-mining-data-science-software.html

29. PEP 20 – The Zen of Python // PEP Index. – https://www.python.org/dev/peps/pep-0020/

30. About Jupyter // Jupyter Project. – https://jupyter.org/

31. Why Django? // Django. – https://www.djangoproject.com/start/overview/

32. Тутубалина Е. В., Тестирование методов анализа тональности текста, основанных на словарях / Тутубалина Е. В., Иванов В. В. [и др.] // Russian Digital Libraries Journal. Volume. 18. No 3-4, 2015. – С. 138–162.

33. NLTK 3.3 Documentation // Natural Language Toolkit. – http://www.nltk.org/

34. Документация pymorphy2 // Морфологический анализатор pymorphy2. – https://pymorphy2.readthedocs.io/en/latest/;

35. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts, 2015. – С. 320–332.

36. Документация gensim // gensim. – https://radimrehurek.com/gensim/index.html

37. Radim Rehurek. Scalability of semantic analysis in Natural Language Processing, Ph.D. Thesis, Masaryk University, 2011.

38. Machine Learning in Python // scikit-learn. – http://scikit-learn.org/stable/index.html

39. Naive Bayes // Scikit learn. – http://scikit-learn.org/stable/modules/naive_bayes.html

40. SVM Classification // Sciket learn. – http://scikit-learn.org/stable/modules/svm.html#svm-classification

41. Perceptron // Scikit learn. – http://scikit-learn.org/stable/modules/linear_model.html#perceptron

42. Черняк О.І. Теорія ймовірностей та математична статистика. Збірник задач: [навч. посібник] / О.І. Черняк, О.М. Обушна, А.В. Ставицький.- 2-ге вид. – К.: Знання, КОО, 2002. — 199 с.

43. Ставицький А.В. Навчально-методичний комплекс з курсів «Прогнозування» та «Фінансове прогнозування». – К.: РВВ ІМФ, 2006. – 107 с.

44. Чорноус Г.О. Моніторинг соціально-економічних систем на основі інтелектуального аналізу даних / Г.О.Чорноус // Моделі управління в ринковій економіці. Збірник наукових праць. – Донецьк, 2012. – №15. – С. 319–335.

45. The Elements of Statistical Learning: Data Mining, Inference, and Prediction by T. Hastie, R. Tibshirani, J. Friedman - Springer 2009, 764 p.

46. Introduction To Machine Learning by Nils J Nilsson – 1997, 209 p.

47. Inductive Logic Programming: Theory and Methods by Stephen Muggleton, Luc de Raedt - ScienceDirect 1994, 51 p.

48. Information Theory, Inference, and Learning Algorithms by David J. C. MacKay - Cambridge University Press 2003, 640 p.

49. Gaussian Processes for Machine Learning by Carl E. Rasmussen, Christopher K. I. Williams - The MIT Press 2005, 266 p.

50. Задача класифікації [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Задача_класифікації. – (дата звернення 05.02.2019). – Назва з екрана.

51. Логістична регресія [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/ Логістична_регресія – (дата звернення 05.02.2019). – Назва з екрана.

52. Метод опорних векторів [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Метод_опорних_векторів. – (дата звернення 05.02.2019). – Назва з екрана.

53. Дерево ухвалення рішень [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Дерево_ухвалення_рішень. – (дата звернення 05.02.2019). – Назва з екрана.

54. Платформа онлайн-курсів Coursera. Режим доступу: https://www.coursera.org/

55. Лекции по логическим алгоритмам классификации [Електронний ресурс] / Воронцов К. В. – 2010. – С. 17-21. – Режим доступу: http://www.machinelearning.ru/wiki/images/3/3e/Voron-ML-Logic.pdf – (дата звернення 17.09.2018). – Назва з екрана.

56. Метод k-найближчих сусідів [Електронний ресурс] – 2018. – Режим доступу: uk.wikipedia.org/wiki/Метод_k-найближчих_сусідів. – (дата звернення 05.02.2019). – Назва з екрана.

57. Practical Data Analysis, 2nd Edition by Hector Cuesta, Sampath Kumar - Packt Publishing 2016, 338 p.

58. PythonDataAnalysisCookbookbyIvanIdris – PacktPublishing2016,462p.

59. Mastering Python Data Analysis by Magnus Vilhelm Persson, Luiz Felipe Martins - Packt Publishing 2016, 284 p.

60. Learning Predictive Analytics with Python by Ashish Kumar - Packt Publishing 2016, 354 p.

61. A Course in Machine Learning by Hal Daumé III - ciml.info 2012, 189 p.

62. Machine Learning, Neural and Statistical Classification by D. Michie, D. J. Spiegelhalter - Ellis Horwood 1994, 298 p.

63. Мета-методологія управління проектами на моделях взаємодії зацікавлених сторін [Текст] : автореф. дис. ... д-ра техн. наук : 05.13.22 / Хлевна Юлія Леонідівна ; Київ. нац. ун-т ім. Тараса Шевченка. - Київ, 2019. - 39 с.

64. Muqtar Unnisa, Opinion mining on twitter data using unsupervised learning technique / Muqtar Unnisa, Ayesha Ameen, Syed Raziuddin // International Journal of Computer Applications 148(12), 2016. – С. 12–19.

65. He Yulan, Self-training from labeled features for sentiment analysis / He Yulan, Zhou Deyu // Information Processing & Management, Volume 47, 2011. – С. 606–616.

66. Ortigosa-Hernández Jonathan, Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers / Ortigosa-Hernández Jonathan, Rodríguez Juan Diego [и др.] // Neurocomputing, 2012. – С. 98–115.

67. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts, 2015. – С. 320–332.

68. Radim Rehurek. Scalability of semantic analysis in Natural Language Processing, Ph.D. Thesis, Masaryk University, 2011.

69. Meng X. Lost in translations? building sentiment lexicons using context based machine translation / Meng X., Wei F. [и др.] // COLING, 2012. – С. 829–838.

70. Анна Пазельская Метод определения эмоций в текстах на русском языке / Анна Пазельская, Алексей Соловьев // The international conference on computational linguistics and intellectual technologies “Dialogue 2011” : конференция. – Москва, 2011. – С. 510–522.

ДОДАТКИ

Додаток А

Програмна реалізація виокремлення ключових слів

```
selectedwords = ['awesome','great','fantastic','extraordinary','amazing','super',
                 'magnificent','stunning','impressive','wonderful','breathtaking',
                 'love','content','pleased','happy','glad','satisfied','lucky',
                 'shocking','cheerful','wow','sad','unhappy','horrible','regret',
                 'bad','terrible','annoyed','disappointed','upset','awful','hate']

def selectedcount(dic,word):
    if word in dic:
        return dic[word]
    else:
        return 0

dfwc = df.copy()
for word in selectedwords:
    dfwc[word] = dfwc['Word Counts'].apply(selectedcount,args=(word,))

word_sum = dfwc[selectedwords].sum()
print('Selected Words')
print(word_sum.sort_values(ascending=False).iloc[:5])

print('\nClass Names')
print(df['Class Name'].fillna("Empty").value_counts().iloc[:5])

fig, ax = plt.subplots(1,2,figsize=(20,10))
wc0 = WordCloud(background_color='white',
                      width=450,
                      height=400 ).generate_from_frequencies(word_sum)

cn = df['Class Name'].fillna(" ").value_counts()
wc1 = WordCloud(background_color='white',
                      width=450,
                      height=400
                     ).generate_from_frequencies(cn)

ax[0].imshow(wc0)
ax[0].set_title('Selected Words\n',size=25)
ax[0].axis('off')

ax[1].imshow(wc1)
ax[1].set_title('Class Names\n',size=25)
ax[1].axis('off')

rt = df['Review Text']
plt.subplots(figsize=(18,6))
wordcloud = WordCloud(background_color='white',
                      width=900,
                      height=300
                     ).generate(" ".join(rt))
plt.imshow(wordcloud)
plt.title('All Words in the Reviews\n',size=25)
plt.axis('off')
plt.show()
```

Додаток Б

Програмна реалізація методів класифікації

Логістична регресія:

```
train_data,test_data = train_test_split(df,train_size=0.8,random_state=0)
X_train = vectorizer.fit_transform(train_data['Review Text'])
y_train = train_data['Sentiment']
X_test = vectorizer.transform(test_data['Review Text'])
y_test = test_data['Sentiment']
start=dt.datetime.now()
lr = LogisticRegression()
lr.fit(X_train,y_train)
print('Elapsed time: ',str(dt.datetime.now()-start))
```

Наївний байєсівський класифікатор:

```
start=dt.datetime.now()
nb = MultinomialNB()
nb.fit(X_train,y_train)
print('Elapsed time: ',str(dt.datetime.now()-start))
```

Метод опорних векторів:

```
start=dt.datetime.now()
svm = SVC()
svm.fit(X_train,y_train)
print('Elapsed time: ',str(dt.datetime.now()-start))
```

Нейронна мережа:

```
start=dt.datetime.now()
nn = MLPClassifier()
nn.fit(X_train,y_train)
print('Elapsed time: ',str(dt.datetime.now()-start))
```

Додаток В

Програмна реалізація підрахунку та виведення результатів порівняння алгоритмів класифікації

```
pred_lr = lr.predict_proba(X_test)[:,1]
fpr_lr,tpr_lr,_ = roc_curve(y_test,pred_lr)
roc_auc_lr = auc(fpr_lr,tpr_lr)

pred_nb = nb.predict_proba(X_test)[:,1]
fpr_nb,tpr_nb,_ = roc_curve(y_test.values,pred_nb)
roc_auc_nb = auc(fpr_nb,tpr_nb)

pred_svm = svm.decision_function(X_test)
fpr_svm,tpr_svm,_ = roc_curve(y_test.values,pred_svm)
roc_auc_svm = auc(fpr_svm,tpr_svm)

pred_nn = nn.predict_proba(X_test)[:,1]
fpr_nn,tpr_nn,_ = roc_curve(y_test.values,pred_nn)
roc_auc_nn = auc(fpr_nn,tpr_nn)

f, axes = plt.subplots(2, 2,figsize=(15,10))
axes[0,0].plot(fpr_lr, tpr_lr, color='darkred', lw=2, label='ROC curve (area = {:0.
2f})'.format(roc_auc_lr))
axes[0,0].plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
axes[0,0].set(xlim=[-0.01, 1.0], ylim=[-0.01, 1.05])
axes[0,0].set(xlabel ='False Positive Rate', ylabel = 'True Positive Rate', title =
'Logistic Regression')
axes[0,0].legend(loc='lower right', fontsize=13)

axes[0,1].plot(fpr_nb, tpr_nb, color='darkred', lw=2, label='ROC curve (area = {:0.
2f})'.format(roc_auc_nb))
axes[0,1].plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
axes[0,1].set(xlim=[-0.01, 1.0], ylim=[-0.01, 1.05])
axes[0,1].set(xlabel ='False Positive Rate', ylabel = 'True Positive Rate', title =
'Naive Bayes')
axes[0,1].legend(loc='lower right', fontsize=13)

axes[1,0].plot(fpr_svm, tpr_svm, color='darkred', lw=2, label='ROC curve (area = {:
0.2f})'.format(roc_auc_svm))
axes[1,0].plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
axes[1,0].set(xlim=[-0.01, 1.0], ylim=[-0.01, 1.05])
axes[1,0].set(xlabel ='False Positive Rate', ylabel = 'True Positive Rate', title =
'Support Vector Machine')
axes[1,0].legend(loc='lower right', fontsize=13)

axes[1,1].plot(fpr_nn, tpr_nn, color='darkred', lw=2, label='ROC curve (area = {:0.
2f})'.format(roc_auc_nn))
axes[1,1].plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
axes[1,1].set(xlim=[-0.01, 1.0], ylim=[-0.01, 1.05])
axes[1,1].set(xlabel ='False Positive Rate', ylabel = 'True Positive Rate', title =
'Neural Network')
axes[1,1].legend(loc='lower right', fontsize=13);
```