МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп’ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ГРАМЯК Роман Андрійович

Інтелектуальний метод визначення конкурентного товару на основі онлайн-відгуків / Intelligent Method of Determining a Competitive Product Based on Online Reviews

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп’ютерні науки

Кваліфікаційна робота

Виконав студент групи
КНм-21
Р.А. Грамяк
_____

Науковий керівник:
к.т.н., доцент
Х.В. Ліп’яніна-Гончаренко
_____

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
Завідувач кафедри
_____ М.П. Комар

ТЕРНОПІЛЬ – 2022

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «магістр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
«____» _____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Грамяк Роман Андріійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Інтелектуальний метод визначення конкурентного товару на основі онлайн відгуків / Intelligent Method of Determining a Competitive Product Based on Online Reviews
керівник роботи к.т.н., доцент Х.В. Ліп'яніна-Гончаренко
затверджені наказом по університету від 31 грудня 2021 року № 606.
2. Строк подання студентом закінченої кваліфікаційної роботи 16 листопада 2022 року.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити
- пошук підходів для визначення конкурентного товару на основі онлайн-відгуків;
- визначення суті і особливості онлайн-відгуків;
- пошук ефективних методів і їх порівняння;
- пошук методів збору відгуків і автоматизованого визначення тексту;
- провести аналіз методів перекладу тексту та класифікації;
- проаналізувати класифікатори та порівняти їх;
- пошук методів класифікації і визначити найкращий результат;
- створити програмну реалізацію інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків.

5. Перелік графічного матеріалу у роботі
- структура інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків;
- графіки реалізації інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Н. контроль	к.т.н., доцент Комар М.П.		

7. Дата видачі завдання 11 жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів проекту	Примітка
1	Сучасний стан предметної області і постановка задачі дослідження	12.2021 р. – 03.2022 р.	
2	Інтелектуальний метод визначення конкурентного товару на основі онлайн-відгуків	03.2022 р. – 05.2022 р.	
3	Програмна реалізація інтелектуального методу визначення конкурентного товару	05.2022 р. – 11.2022 р.	
4	Повне завершення та оформлення кваліфікаційної роботи на кафедру	16.11.2022 р.	

Студент_____ Грамяк Р.А.
(підпис)

Керівник проекту_____ Ліп’яніна-Гончаренко Х.В.
(підпис)

РЕЗЮМЕ

Кваліфікаційна робота на тему «Інтелектуальний метод визначення конкурентного товару на основі онлайн-відгуків» на здобуття освітнього ступеня «Магістр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 115 сторінок і містить 7 ілюстрацій, 3 таблиці, 3 додатки та 50 використаних джерел.

Метою даної кваліфікаційної роботи є розробка інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків.

Методи досліджень: методи збору даних (онлайн-відгуки), методи автоматизованого визначення мови, методи перекладу тексту, методи класифікації тексту, інтелектуальний метод визначення конкурентного товару.

Результати дослідження: розроблено інтелектуальний метод визначення конкурентного товару на основі онлайн-відгуків, що дасть можливість інтернет-продавцям зменшити ризики при обранні товару для торгівлі.

Результати роботи будуть використовуватись в різних підприємствах та науково-дослідницьких закладах, які займаються аналізом споживачів виробленої продукції.

Ключові слова: АНАЛІЗ ДАНИХ, ПАРСИНГ, ПЕРЕКЛАД ТЕКСТУ, КЛАСИФІКАЦІЯ, МАШИННЕ НАВЧАННЯ.

ABSTRACT

Qualification work on the topic "Intelligent method of determining a competitive product based on online reviews" for the degree of "Master" in the specialty 122 "Computer Science" of the educational program "Computer Science" is written in 93 pages and contains 7 illustrations, 2 tables, 3 appendices and 50 references.

The purpose of this qualification work is to develop an intelligent method for determining a competitive product based on online reviews.

Research methods: methods of data collection (online reviews), methods of automated language detection, methods of text translation, methods of text classification, intelligent method of determining a competitive product.

Research results: an intelligent method for determining competitive products based on online reviews has been developed, which will allow online sellers to reduce risks when choosing a product for trade.

The results of the work will be used in various enterprises and research institutions that are engaged in the analysis of consumers of manufactured products.

Keywords: DATA ANALYSIS, PARSING, TEXT TRANSLATION, CLASSIFICATION, MACHINE LEARNING.

ЗМІСТ

ВСТУП

Актуальність теми. Сьогодні дедалі більше покупців ґрунтують свої рішення про покупку на онлайн-відгуках, залишених іншими покупцями. Крім сприйманої неупередженості та правдивості споживчих рецензентів, ці відгуки унікальні, оскільки вони оцінюють продукти в контексті повсякденних сценаріїв купівлі та використання. З цієї точки зору, огляди відображають тенденції та теми, актуальні для широкої аудиторії реальних і потенційних покупців продукту. Орієнтований на користувача і "живий" характер відгуків відкриває перед продавцями безліч невикористаних можливостей. Дослідники аналізують відгуки, щоб отримати уявлення про покупців і їхнє ставлення до продукту, а також для прогнозування ефективності продажів продукту. Відгуки можна аналізувати з різних точок зору. Однією з таких точок зору є поздовжнє вивчення відгуків. Типовий продукт отримує безліч відгуків з плином часу. Було показано, що ці відгуки - не просто набір думок, які приходять у випадковий час; скоріше, їхня поява слідує певним закономірностям і тенденціям.

З часової точки зору, покупців можна поділити на групи залежно від часу їхньої покупки. Минулі дослідження показали, що члени цих різних груп покупців мають певні характеристики. Наприклад, якщо ранні групи споживачів (звані новаторами і ранньою більшістю) схильні до пошуку ризику, оптимізму і достатку, то пізніші (звані пізньою більшістю і відстаючими) схильні до консерватизму, песимізму і стурбованості ціною. Подібні демографічні, психографічні та економічні характеристики в сукупності визначають те, що дослідники називають "ступенем інноваційності покупця", що є ключовим чинником, який визначає час покупки.

У зв’язку з цим можна вважати, що розробка інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків є актуальною у інтернет-торгівлі, це дасть можливість інтернет-продавцям зменшити ризики при обранні товару для торгівлі, а отже і збільшити дохід.

Метою дослідження є розробка інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків.

Для досягнення мети потрібно вирішити такі завдання:

– пошук підходів для визначення конкурентного товару на основі онлайн-відгуків;

– визначення суті і особливості онлайн-відгуків;

– пошук ефективних методів і їх порівняння;

– пошук методів збору відгуків і;

– провести опис методів визначення мови тексту та його перекладу;

– провести опис методів класифікатори та порівняти їх;

– розробити інтелектуальний метод визначення товару на основі онлайн-відгуків;

– розробити програмну реалізацію інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків.

Об’єкт дослідження є обробка природної мови у контексті методів розпізнавання тональності тексту.

Предмет дослідження є сукупність теоретико-методологічних та прикладних засад для оцінки онлайн-відгуків.

Методи дослідження. У роботі використані методи математичного програмування; системного аналізу; машинного навчання.

Наукова новизна отриманих результатів: розроблено інтелектуальний метод визначення конкурентного товару на основі онлайн-відгуків, що дасть можливість інтернет-продавцям зменшити ризики при обранні товару для торгівлі.

Апробація результатів. Основні наукові результати роботи доповідались на міжнародних наукових конференціях, серед яких: міжнародна науково-практична конференція IntelITSIS’2021 з доповіддю за темою «Intelligent method of a competitive product choosing based on the emotional feedbacks coloring» (Scopus); XIX Міжнародна науково-практична конференція молодих вчених в місті Тернопіль, 13 травня 2022 року, Україна; стаття в фаховому

журналі Вісник ХНУ на тему «Метод вибору конкурентного товару на основі емоційного забарвлення відгуків».

Практична цінність. Продавець може приймати управлінські рішення стосовно вигідного вкладання коштів в новий товар, і, тим самим зменшить ризики від неприбуткових операцій купівлі-продажу. Також зменшить затрати часу на пошук популярних та якісних товарів на основі відгуків користувачів.

Структура та обсяги роботи. Дипломна робота складається з вступу, трьох розділів, висновку, списку використаних джерел. Повний обсяг роботи становить 115 сторінок і містить 7 ілюстрацій, 3 таблиці, 3 додатки та 50 використаних джерел.

1 СУЧАСНИЙ СТАН ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Підходи до визначення конкурентного товару

Необхідність визначення меж дедалі складніших товарних ринків породила низку аналітичних методів, ґрунтованих на поведінці або судженнях покупців. Різні методи порівнюють і протиставляють залежно від того, чи відповідають вони концептуальному визначенню товарного ринку, а також від їхньої здатності давати діагностичні висновки.

Проблеми ідентифікації конкурентних товарних ринків пронизують усі рівні маркетингових рішень [11]. Такі стратегічні питання, як базове визначення бізнесу, оцінювання можливостей, що надаються прогалинами на ринку, або загроз, які створюються діями конкурентів, та основні рішення щодо розподілу ресурсів перебувають під сильним впливом широти чи вузькості конкурентної арени. Частка ринку є найважливішим тактичним інструментом для оцінки ефективності діяльності та визначення територіального розподілу бюджету на рекламу, персонал відділу продажів та інші витрати. Прискорення темпів антимонопольного переслідування є ще одним джерелом потреби в більш точному визначенні відповідних меж ринку, що дасть змогу краще зрозуміти конкурентні наслідки придбань.

Наразі їхні потреби задовольняються підходами до визначення товарних ринків, що наголошують на схожості виробничих процесів, функцій або використовуваної сировини. Ці підходи [12] рідко дають задовільну картину загроз або можливостей, з якими стикається бізнес. У відповідь на це з'явилася значна активність, спрямована на визначення товарних ринків з точки зору клієнтів.

Зрештою всі межі між продуктом і ринком є довільними. Вони існують через постійні потреби в осмисленні ринкових структур і накладання певного порядку на складне ринкове середовище. Але ця ситуація не може бути іншою.

Одна з причин полягає в широкому розмаїтті контекстів ухвалення рішень, які диктують різні визначення кордонів.

Визначення ринку і класу продукції, які підходять для ухвалення тактичних рішень, як правило, вузькі, що відображають короткострокові інтереси менеджерів з продажу і менеджерів з продукції, які розглядають ринок як "шматок попиту, що має бути заповнений ресурсами, які перебувають у моєму розпорядженні". Ці ресурси зазвичай обмежені продуктами в поточній продуктовій лінійці. Довгостроковий погляд, що відображає проблеми стратегічного планування, неминуче покаже ширший ринок продукту, щоб урахувати:

1. нині не обслуговувані, але потенційні ринки;
2. зміни в технології, цінових відносинах і постачанні, що розширюють спектр потенційних продуктів-замінників;
3. час, потрібний нинішнім і потенційним покупцям для реакції на ці зміни.

У минулому визначення ринку фокусувалися або на продукті (як у такому визначенні [13]: "...продукти можуть бути тісно пов'язані в тому сенсі, що споживачі розглядають їх як замінники", яке передбачає однорідність поведінки споживачів), або на покупцях ("...особи, які в минулому купували цей клас товарів". Ні той, ні інший підхід не дуже корисний для прояснення концепції або оцінки альтернативних підходів до визначення меж товарного ринку.

Більш продуктивний підхід можна вивести з таких передумов:

– Люди шукають вигоди, які забезпечують продукти, а не продукти як такі. Конкретні продукти або бренди являють собою доступні комбінації вигод і витрат.
– Споживачі розглядають наявні альтернативи з погляду контекстів використання, в яких вони мають досвід, або конкретних застосувань, які вони розглядають. Саме вимоги до використання диктують шукані переваги.

Виходячи з цих двох передумов, ми можемо визначити ринок продукту як сукупність продуктів, що вважаються замінниками, у тих ситуаціях використання, в яких шукають схожі моделі вигод, і клієнтів, для яких ці види використання є релевантними.

Це визначення орієнтоване на попит або клієнта, оскільки потреби та вимоги клієнтів мають першорядне значення. Альтернативою є підхід з точки зору пропозиції та визначення продукції за такими експлуатаційними критеріями, як схожість виробничих процесів, сировини, зовнішнього вигляду або функцій. Ці критерії лежать в основі системи стандартної промислової класифікації (SIC) і отримали широке визнання, тому що їх легко впровадити. Вони призводять до, здавалося б, стабільних і чітких визначень, і, що важливо, включають фактори, які значною мірою контролює фірма [14]; мається на увазі, що визначення також піддається контролю. Вони також корисні для виявлення потенційних конкурентів через схожість систем виробництва і розподілу. Критерії, орієнтовані на попит, з іншого боку, менш знайомі і, отже, видаються складнішими для реалізації (внаслідок розмаїття доступних методів і неминучих проблем емпіричного вимірювання, помилок вибірки й агрегування індивідуальних відмінностей споживачів). Ба більше, такі визначення можуть бути менш стабільними з плином часу через мінливі потреби та смаки. Зрештою, організація має ініціювати дослідницьку програму для збирання та аналізу відповідних даних і моніторингу змін, а не покладатися на урядові чи інші зовнішні джерела інформації. Наслідком цього найчастіше є рішення використовувати заходи, орієнтовані на пропозицію, незважаючи на їхню сумнівну застосовність у багатьох обставинах.

Поняття унікальної товарної категорії є надмірним спрощенням перед обличчям довільного характеру кордонів. Заміщуваність [15] - це міра ступеня. Тому краще мислити в термінах рівнів в ієрархії продуктів у рамках загального класу продуктів, що представляють усі можливі способи задоволення фундаментальної потреби або бажання споживача. Лунн проводить такі корисні відмінності між [16]:

- Абсолютно різні типи або підкласи продуктів, які існують для задоволення значно відмінних моделей потреб, що виходять за рамки фундаментальних або загальних. Наприклад, і гарячі, і холодні пластівці задовольняють одну й ту саму потребу в харчуванні на сніданок, але в іншому вони різні. У довгостроковій перспективі типи продуктів можуть поводитися як замінники.
- У межах одного загального типу доступні різні варіанти продукції, наприклад, натуральні, поживні, підсолоджені та звичайні пластівці. Існує висока ймовірність того, що між підмножинами цих варіантів відбувається деяке короткострокове заміщення (наприклад, між натуральним і поживним). Якщо заміщення занадто велике, то альтернативи всередині підмножини не заслуговують на розмежування.
- У межах одного й того самого конкретного варіанта продукту випускаються різні бренди. Хоча ці бренди можуть незначно відрізнятися за багатьма ознаками (колір, тип пакування, форма, текстура тощо), проте вони зазвичай є прямими і безпосередніми замінниками.

У такій ієрархії може бути багато або мало рівнів, залежно від широти і складності істинної потреби та різноманітності альтернатив, доступних для її задоволення. Таким чином, ця типологія є лише відправною точкою для осмислення аналітичних питань.

Визначення "продукт-ринок" [17], запропоноване вище, має на увазі субринки, що складаються з покупців зі спільним використанням або застосуванням продукту. Це сегменти відповідно до традиційного визначення груп, які мають схожу поведінку під час купівлі або використання чи реакцію на маркетингові зусилля. Корисніше розглядати їх як субринки всередині стратегічних сегментів ринку. Хоча кожен із цих субринків може слугувати фокусом для рішення про позиціонування, відмінності між ними можуть не становити істотних стратегічних бар'єрів для подолання конкурентами. Такі бар'єри можуть ґрунтуватися на таких факторах, як відмінності в географії,

кількості замовлень, вимогах до технічної допомоги та сервісної підтримки, чутливості до ціни або сприйнятої важливості якості та надійності. Перевірка стратегічної значущості полягає в тому, чи повинні сегменти, що визначаються цими або іншими характеристиками, обслуговуватися істотно різними маркетинговими комплексами. Тоді межі можуть проявлятися у вигляді розривів у структурі цін, темпах зростання, структурі частки і каналах розподілу під час переходу від одного сегмента до іншого.

1.2 Суть та особливості формування онлайн-відгуків

Розвиток інформаційно-комунікаційних технологій і, зокрема, Інтернету відкрив нові можливості для постачальників послуг і споживачів обмінюватися інформацією між собою. У цьому контексті онлайн відгуки, що являють собою особливу форму електронного сарафанного радіо [18], стали новим каналом комунікації і стають дедалі популярнішими серед громадян. Сьогодні онлайн-відгуки є одним із найвпливовіших джерел інформації для споживачів під час формування рішення про покупку і забезпечують їм великі переваги. Найголовніше, вони дають змогу географічно розкиданим споживачам обмінюватися незалежними думками про товари і послуги, допомагаючи їм приймати обґрунтовані рішення про покупку. Крім того, онлайн-відгуки також мають великий потенціал створення вартості для компаній. Будучи джерелом як джерело поліпшення продукції та послуг, вони можуть збільшити дохід і сприяти встановленню довгострокових відносин, відіграючи, таким чином, важливу роль у маркетингових зусиллях компаній. значну роль у маркетингових зусиллях компаній. Зростаюча популярність і послуги, що надаються. В останні роки зростаюча популярність і переваги онлайн-відгуків привернули велику увагу дослідників і практиків.

Попередні дослідження показують, що інформація, створена споживачами, наприклад, онлайн-огляди, є більш переконливою, ніж інформація, створена маркетологами, оскільки споживачі не мають

корисливих інтересів і тому є незалежними та більш достовірними. Більш того, попередні дослідження показують, що сприймана довіра відіграє важливу роль у процесі ухвалення рішень споживачами та зменшує. Згідно з Baek et al [19], довіра також є "найбільш важливим чинником у прийнятті електронного "слова з вуст в уста" (eWOM)". З цієї причини довіра до онлайн-відгуків також видається важливою під час ухвалення рішень про купівлю на основі таких відгуків. З огляду на важливість довіри в контексті онлайн-відгуків і пов'язаних із ними рішень про купівлю, цілком логічно, що довіра до відгуків становить великий інтерес як для споживачів, так і для маркетологів. Однак триваючі громадські дебати про фальшиві онлайн-відгуки підвищили обізнаність споживачів про цю тему, поставивши під сумнів їхню достовірність Деякі компанії, наприклад, дотримуються обманних практик і маніпулюють онлайн-відгуками про свої товари та послуги, щоб вплинути на поведінку споживачів під час купівлі.

Як наслідок, репутація і використання онлайн-відгуків і сайтів відгуків можуть опинитися під загрозою в довгостроковій перспективі, якщо побоювання і невпевненість споживачів будуть поширюватися і зміцнюватися. У зв'язку з цим Мюнцель [2] зазначає, що "зростаюча практика розміщення фальшивих відгуків в Інтернеті не тільки ставить під загрозу довіру до сайтів відгуків як важливих джерел інформації для приватних осіб, а й наражає на небезпеку цінне джерело інформації для постачальників послуг". Для того щоб протистояти цьому розвитку, ініційованому недобросовісними фірмами, надійним компаніям важливо розуміти, як споживачі сприймають та оцінюють достовірність відгуків в Інтернеті, та, зокрема, знати, які чинники визначають достовірність відгуків з точки зору споживачів. Однак, незважаючи на величезне значення достовірності відгуків - особливо стосовно рішень споживачів щодо купівлі - і зростаючу увагу дослідників і практиків до онлайн-відгуків, досі існує мало досліджень, присвячених достовірності комунікації eWOM загалом і онлайн-відгуків та її детермінантам зокрема.

Незважаючи на те, що значна кількість досліджень, присвячених онлайн-відгукам, мали вплив онлайн-відгуків на ставлення споживачів до товарів та послуг, це відбувалося в досить загальному вигляді і переважно без урахування факторів онлайн-відгуків, які впливають на наміри споживачів щодо купівлі. Між нероздрібними сайтами, що займаються збором та узагальненням відгуків споживачів (наприклад, Epinions.com, Rateitall.com, Yelp.com) та численними інтернет-магазинами і виробниками, що наслідують їхній приклад (наприклад, Amazon.com, Sears.com, Dell, Levi's), онлайн-покупці мають дедалі ширший доступ до думок і відгуків інших покупців про товари. Доступність відгуків про споживчі товари, далі іменованих відгуками, імовірно, продовжуватиме зростати, принаймні, з двох причин.

По-перше, лякаюча кількість варіантів в Інтернеті призводить до того, що споживачі цінують відгуки і як механізм вибору, і як важливе джерело інформації про характеристики продукту, які часто важко оцінити в онлайновому середовищі [20]. Дійсно, завдяки розвитку роздрібної торгівлі в Інтернеті та соціальних мереж, споживачі тепер приходять на ринок, розраховуючи отримати доступ до відгуків.

По-друге, бажання збільшити пропускну спроможність і скоротити витрати, пов'язані з обслуговуванням клієнтів і поверненням продукції, імовірно, мотивує рітейлерів полегшувати відгуки в спробах підвищити залученість клієнтів до роботи сайту та поліпшити процес ухвалення рішень споживачами. нещодавні дослідження у сфері інформаційних систем і маркетингу дають змогу зробити низку висновків щодо відгуків. З погляду передумов, дослідження визначають характеристики і мотиви тих, хто пише відгуки, включно зі стратегічними підробленими відгуками, написаними від імені організацій, і нестратегічними обманними відгуками, написаними окремими особами, які не є покупцями. Доповнюючи цю перспективу [21], ще один потік досліджень вивчає результати оглядів, виявляючи, що відгуки впливають на сукупну поведінку споживачів, що проявляється в продажах, продажах, переглядах, оцінках сайту та продукту, конкурентній розвідці та

індивідуальному споживчому виборі. Те, як споживачі вирішують, чи можуть вони покладатися на конкретний відгук, вивчено менше.

Говорячи про прикладну важливість цього питання, багато сайтів роздрібної торгівлі збирають, узагальнюють і публікують відгуки споживачів про "корисність" або "корисність" окремих відгуків [22]. Незважаючи на нещодавні дослідження, в яких вивчали вплив низки атрибутів або характеристик відгуків на сприйняття споживачами корисності, цілісна система корисності відгуків ще не сформувалася. З огляду на поширеність відгуків, а також стратегічну важливість управління цією інформацією, така система має виявитися корисною як для дослідників, так і для практиків, виявляючи фактори та умови, що підвищують або знижують корисність відгуків. Висновки про те, що рецензенти інколи надають інформацію, яка не чинить жодного або навіть шкідливого впливу на корисність, а також можливість того, що егоцентризм змушує рецензентів не усвідомлювати свою непотрібність як комунікаторів, дають підстави припустити, що вдосконалення форм рецензій саме по собі може бути недостатньо ефективним для покращення відгуків про продукти. Натомість веб-сайти, що публікують відгуки, мають змінити менталітет авторів відгуків, щоб зменшити залежність від їхніх власних знань та інформаційних уподобань. Навчання людей упередженості інтроспекції може подолати цей ефект і проблеми, які з нього випливають.

1.3 Аналіз існуючих методів для визначення конкурентного товару

Методи виявлення товарних ринків [23], орієнтовані на клієнта, можна класифікувати за тим, чи спираються вони на поведінкові або судження. Купівельна поведінка дає найкраще уявлення про те, що люди дійсно роблять або робили, але не обов'язково про те, що вони могли б зробити за обставин, що змінилися. Тому її цінність вища як керівництва для тактичного планування. Дані про судження у формі сприйняття або вподобань можуть

дати краще уявлення про майбутні моделі конкуренції та причини наявних моделей. Отже, вони можуть краще слугувати основою для стратегічного планування. Далі оцінюються сім різних аналітичних підходів [24] у рамках таких двох основних класів:

А1. У межах широкої категорії суджень споживачів про замінність буде розглянуто п'ять пов'язаних підходів, що використовують вільні асоціації, "метрику долара", пряме угруповання продуктів, аналіз продуктів за видами використання та аналіз заміщення за видами використання.

– Перехресна еластичність попиту розглядається більшістю економістів як стандарт, з яким слід порівнювати інші підходи. Незважаючи на вражаючу логіку міри перехресної еластичності, її широко критикують і рідко використовують:
– Концептуальне визначення цього заходу передбачає відсутність реакції однієї фірми на зміну ціни іншої. На практиці ця умова рідко виконується.

Зрештою, «на ринках, де зміни цін відбуваються нечасто, або всі ціни змінюються разом, або де змінюються не лише ціни, а й інші чинники, просто бракує інформації, яка міститься в даних, щоб провести достовірну статистичну оцінку еластичності».

А2. Подібність у споживчій поведінці. Цей підхід було успішно використано в дослідженні етичного фармацевтичного ринку [3]. Основне питання полягало в тому, якою мірою продукти, що складаються з різних хімічних речовин, але мають схожий терапевтичний ефект, можуть бути значущими замінниками. Ключем до відповіді на це питання стала наявність унікального набору даних про поведінку лікарів.

А3. Показники переходу на інший бренд зазвичай інтерпретуються як умовні ймовірності, тобто ймовірність купівлі бренду А з огляду на те, що востаннє було куплено бренд В. Такі показники зазвичай оцінюються на основі панельних даних, де покупки будь-якого даного респондента представлені послідовністю невизначеної довжини. Ймовірності

розраховуються на основі підрахунку частоти виникнення кожної умови в даних (наприклад, покупкам марки А передують інші марки в послідовності). Передумовою є те, що респонденти з більшою ймовірністю перемикатимуться між близькими замінниками, ніж між далекими, і що пропорції перемикання між брендами забезпечують міру ймовірності заміщення.

Як і у випадку з перехресною еластичністю, показник перемикання між брендами можна використовувати тільки після того, як буде визначено набір конкуруючих продуктів. Оскільки оцінка коефіцієнтів переходу на іншу марку ґрунтується на послідовності покупок, має існувати якась логічна основа для визначення того, які марки слід включити в таку послідовність. Подібність моделей використання, як обговорювалося вище, є однією з перспективних основ.

В.1. Аналіз послідовності ухвалення рішень використовує протоколи ухвалення споживчих рішень, які показують послідовність, у якій різні критерії використовуються для досягнення остаточного вибору. У звичайній процедурі людей просять усно розповісти, що відбувається у них у голові, коли вони ухвалюють рішення про покупку під час походу в магазин. Такий вербальний запис називається протоколом, на відміну від ретроспективного опитування випробовуваних про їхні рішення. На основі таких даних можна розробити модель того, як суб'єкт приймає рішення. Ці моделі визначають атрибути об'єктів вибору або ситуацій, які розглядаються, а також послідовність і метод комбінації цих атрибутів або підказок. Як правило, атрибути або підказки розташовуються в ієрархічній структурі, званій деревом рішень. Порядок, у якому вони розглядаються, моделюється структурою шляху дерева. Гілки ґрунтуються тільки на тому, чи задовільний рівень атрибута чи ні, або присутня певна умова ("чи не занадто висока ціна?", "чи немає в магазині моєї улюбленої марки?").

Аналіз протоколів проводиться на індивідуальному рівні. Перевага цього методу полягає в тому, що він дає змогу розпізнати індивідуальні відмінності в знаннях і переконаннях щодо альтернативних продуктів і

критеріїв вибору. У принципі, індивіди можуть бути згруповані в сегменти на основі схожих процедур прийняття рішень. Показники ступеня конкуренції між брендами можна отримати з протоколів різних сегментів, зазначаючи, які альтернативи взагалі розглядають і коли їх виключають із подальшого розгляду за критеріями, які використовують на кожному етапі процесу ухвалення рішення (альтернативи, виключені на пізніших етапах, мають бути конкурентоспроможнішими за ті, що були виключені раніше).

В.2. Перцепційне картування охоплює велике сімейство методів, які використовують для створення геометричного уявлення сприйняття покупцями якостей, якими володіють продукти/бренди, що входять у попередньо визначений товарний ринок. Бренди представлені у вигляді місць (точок або, можливо, регіонів) у просторі. Розміри цього простору розрізняють конкурентні альтернативи і являють собою вигоди або витрати, що сприймаються як важливі для покупки. Таким чином, будь-який продукт/бренд можна розташувати в такому просторі відповідно до набору координат, які представляють ступінь, у якій продукт, як вважають, має кожну перевагу або атрибут витрат. Відносні "відстані" між альтернативними продуктами можуть бути інтерпретовані як міри сприйманої замінності кожної альтернативи будь-якою іншою.

Існує кілька різних методів [25], які можуть бути використані для створення перцептивних конфігурацій ринків продуктів (наприклад, пряме шкалювання, факторний аналіз, множинний дискримінантний аналіз, багатовимірне шкалювання). Аналіз може ґрунтуватися на показниках сприйманої загальної схожості/відмінності, сприйманої доречності у звичайних ситуаціях використання і кореляції між рівнями атрибутів для пар продуктів. На жаль, таке розмаїття критеріїв і методів може призвести до дещо різних карт сприйняття і, можливо, до різних визначень ринку продуктів. Для порівняння альтернатив і оцінки того, які визначення продуктів є більш обґрунтованими для конкретних цілей, все ще потрібно багато емпіричних досліджень.

В.3. Аналіз заміщення технологій адаптує ідею переваги, пов'язаної з відстанню в багатоатрибутивному просторі, до проблеми прогнозування заміни одного матеріалу, процесу або продукту іншим - наприклад, алюмінію на мідь в електротехніці або полівінілу на скло в пляшках для лікерів. Кожна успішна заміна має тенденцію слідувати S-подібній або "логістичній" кривій, що являє собою повільний старт, оскільки початкові проблеми та опір змінам треба подолати, потім швидший прогрес у міру набуття визнання та поширення інформації про застосування, і, нарешті, уповільнення темпів заміни в міру досягнення насичення.

Лендз і Лаунд [4] зазначали, що простий підхід до прогнозування перебігу і швидкості процесу заміщення полягає в проектуванні функції, що має відповідну логістичну криву, з використанням історичних даних для визначення її параметрів. Цей метод припасування кривої не бере до уваги багато потенційних чинників, що впливають на процес, як-от: вік, стан і швидкість застаріння капітального устаткування, використовуваного в старій технології; цінова еластичність попиту; і "корисність у використанні" або відносна перевага в продуктивності. Останні зусилля з моделювання коефіцієнтів заміщення були зосереджені на відносній "корисності" як основі для поліпшення здатності прогнозування. Процедура оцінювання "корисності у використанні" включає: спочатку визначення відповідних атрибутів і експлуатаційних характеристик кожного з конкуруючих продуктів або технологій, потім оцінювання експертами міри, в якій кожна альтернатива має кожний атрибут, і сприйнятої важливості кожного атрибута на кожному ринку кінцевого використання. Нарешті, загальна корисність для кожного продукту в кожній ситуації використання отримується шляхом множення оцінки володіння атрибутами на оцінки важливості, підсумовування отриманих продуктів і коригування на різницю в ціні одиниці продукції. Незважаючи на те, що структуру моделі можна піддати критиці, а для визначення атрибутів, як видається, слід покладатися на вимірні фізичні властивості, цінність базового підходу не слід скидати з рахунків. У результаті виходить вельми

корисна кількісна міра корисності, яку можна використовувати для оцінки замінності конкуруючих продуктів або технологій у конкретних ситуаціях використання.

В.4. Судження покупців про замінність можуть бути отримані різними способами. Найпростіший - попросити вибірку покупців вказати ступінь замінності між можливими парами брендів за такою шкалою оцінювання, як: ні, низька, деяка або значна замінність. Крім цього звичного підходу, нещодавно було розроблено кілька методів використання суджень споживачів, які дають змогу отримати набагато глибше діагностичне уявлення про моделі конкуренції, серед яких метод вільної відповіді. Респондентам представляють різні марки і просять вільно асоціювати назви аналогічних або таких, що замінюють, марок. Отримують два види даних [26]. По-перше, це частота згадування одного бренду як замінника іншого, що може бути використано як міру подібності двох брендів для встановлення перцептивного простору. По-друге, порядок згадування брендів-замінників можна розглядати як дані рангового порядку. Ці дані являють собою сукупність суджень у різних ситуаціях і залишають респондентові право вирішувати, наскільки схожими мають бути два бренди, щоб вони стали замінниками.

Корисний варіант запитання з вільною відповіддю запитує респондентів, що б вони зробили, якби не змогли купити бажану марку. Перевага цього запитання в тому, що його можна реально адаптувати до конкретних ситуацій. Наприклад, в одному дослідженні респондентів, які п'ють скотч, запитували, що б вони зробили, якби скотч був недоступний у різних ситуаціях, наприклад, на великій коктейльній вечірці рано ввечері. Очевидно, були ситуації, коли біле вино було кращою альтернативою.

Метод доларової метрики. Респондентам спочатку пред'являються всі можливі пари марок, кожна з яких позначена їхньою звичайною ціною. У кожному випадку респондент обирає марку, яку він/вона купив би в разі вимушеного вибору. Потім їх запитують, до якої ціни має зрости марка, якій надають перевагу, щоб вони змінили свою первісну перевагу. Сила переваги

вимірюється з точки зору цього цінового приросту. Такі дані мають бути додатково "оброблені" для обчислення агрегованих показників переваг.

Ця процедура деякою мірою аналогічна лабораторному вимірюванню перехресної еластичності попиту. Набір потенційно конкуруючих брендів має бути визначено заздалегідь. Процедура досить проста в проведенні й аналізі; хоча простота може бути зведена нанівець, якщо включити міркування щодо передбачуваного використання, знайомства з брендом і сегментації ринку. Схоже, що респонденти здатні виявити свої переваги щодо різних альтернатив у ситуації вимушеного вибору. Питання про те, чи можуть вони достовірно пояснити, як вони дійшли до таких уподобань, оцінивши мінімальну зміну ціни, яка призведе до перемикання, залишається відкритим.

Основні висновки з аналізу природи кордонів і різних емпіричних методів виявлення конкурентних товарних ринків такі:

- межі рідко бувають чіткими - зрештою, всі межі довільні,
- придатність різних емпіричних методів сильно залежить від характеру ринкового середовища,
- загалом, ті емпіричні методи, які явно визнають розмаїття ситуацій використання, мають найширшу застосовність і дають максимальну проникливість. Концепція ситуації використання видається найпоширенішим спільним знаменником ринкового середовища, який може бути використаний як основа для емпіричних методів,
- більшість методів, особливо заснованих на поведінкових показниках, статичні й насилу справляються зі змінами в перевагах або додаванням і видаленням альтернатив вибору на ринку,
- незалежно від методу, найпостійнішою проблемою є відсутність надійних критеріїв для визнання меж.

1.4 Постановка задачі

На основі даної інформації потрібно визначити різні методи класифікації текстів, слабкі та сильні сторони, можливості і їхні недоліки. Потрібно усвідомити проблеми, які присутні в методах класифікації текстів, для подальшого оцінювання класифікаторів. Але, судячи з джерел літератури, класифікації тексту набувають більшого значення в аналізі тексту в інтелектуальному визначенні. Це знижує витрати в подальшій роботі. Інші проблеми: збільшення продуктивності, вибір додаткових функцій, зони документів і дисбаланс даних. Потрібно зробити висновок, що недоцільно вибирати один класифікатор для одної проблеми. Але, кількість помилок для вибору класифікатора може бути мінімізована на основі поданої інформації. Потужніші і простіші алгоритми для оптимізації параметрів і обробки даних – це області, які будуть вивчатись в майбутньому.

Метою дослідження є аналіз емоційного відгуку за коментарями, описування принципів роботи і також реалізація. Далі аналіз різних методів класифікації текстів, визначити слабкі і сильні сторони, щоби визначити різні можливості вилучання знань в області аналізу даних. Метою аналізу тональності є знаходження думок в тексті і визначення їх властивостей. Цей метод знаходить своє практичне застосування в таких областях як соціологія, політологія, маркетинг, медицина та багато інших. Ці знання вкрай важливі для фахівців за даними. Вони могли б використовувати результати цього дослідження для розробки індивідуальних моделей даних.

Завдання полягає в зосередженні на аналізі різних методів, які будуть використовуватися для навчання та тестування:

1. пошук підходів для визначення конкурентного товару на основі онлайн-відгуків;
2. визначення суті і особливості онлайн-відгуків;
3. провести аналіз методів для визначення конкурентного товару;

4. пошук методів збору відгуків і автоматизованого визначення тексту;
5. провести аналіз методів перекладу тексту та класифікації;
6. проаналізувати класифікатори та порівняти їх;
7. пошук методів класифікації і визначити найкращий результат;
8. створити програмну реалізацію інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків.

Завдання 1-3 розглянуто в параграфах 1.1, 1.2 та 1.3 відповідно, а виконання завдань 4-8 представлені у параграфах 2.1-3.4.

Висновки до розділу 1

1. Питання про те, як визначити межі між продуктом і ринком, не можна відокремити від способів використання результатів. Стратегічні або довгострокові визначення структури ринку неминуче мають більше значення, навіть якщо вони в основному отримані з суджень споживачів, а не з їхньої поведінки. Дуже вузько визначені межі видаються адекватними для короткострокових, тактичних рішень у більшості категорій товарів. Цінність правильного і стратегічно значущого визначення ринку продукції полягає в "розтягуванні" уявлень компанії досить далеко, щоб не проґавити значних загроз і можливостей, але не настільки далеко, щоб розпилювати зусилля зі збирання та аналізу інформації на "дальні постріли". Цього балансу важко досягти, враховуючи величезну кількість нинішніх і потенційних конкурентів, з якими стикається більшість компаній.

2. Метод визначення конкурентного товару на основі онлайн-відгуків збагачує розуміння онлайн-відгуків і робить висновки про те, як компанії можуть краще використовувати ці відгуки і розробляти чудові онлайн-системи підтримки прийняття рішень. Він розширює попередні роботи в цій галузі як теоретично, так і методологічно. Визнається, що можуть існувати альтернативні і не менш продуктивні методи аналізу змісту відгуків [14]. Наприклад, дослідники з досвідом у царині лінгвістики або теорії дискурсу можуть вивчити відгуки із семантичного, прагматичного, просторового і часового поглядів, щоб дізнатися більше про те, коли і як споживачі використовують різні типи мови у відгуках. Крім того, хоча споживачі загалом оцінили відгуки в вибірці як корисні, відносно непотрібні відгуки також можуть вплинути на ставлення і реакцію споживачів на продукти, канали або ритейлерів [15]. У будь-якому разі, з огляду на важливість привернення уваги споживачів у цьому дедалі більш інформаційно перевантаженому світі, дуже важливо продовжувати аналіз та оцінювання інформаційного змісту відгуків, щоб визначити наслідки для

розроблення веб-сайтів, які покращують результати для споживачів, виробників, роздрібних торговців та інших зацікавлених сторін.

3. Ці висновки призводять до ситуації, коли стан знань не встигає ні за поточною потребою в розумінні, ні за мінливими технологічними, соціальними та економічними чинниками, які постійно змінюють ринкове середовище. Щоб виправити цю ситуацію, існує явна необхідність у стратегічно орієнтованій програмі досліджень у різних ринкових ситуаціях [27]. Дослідження на кожному ринку мають характеризуватися використанням безлічі методів для пошуку підтвердження шляхом перехресної валідації та поздовжніх підходів, у яких за судженнями слідують поведінкові методи, здатні підтвердити висновки. Як вже зазначали, різні методи мають різні сильні та слабкі сторони, і необхідно більше дізнатися про чутливість результатів до недоліків кожного методу. Крім того, неминуче виникнуть точки суперечності та узгодженості в уявленнях, отриманих на основі меж, установлених різними методами. Процес розв'язання суперечностей має бути найпоказовішим як з погляду розуміння конкурентної позиції фірми, так і з погляду пропозиції альтернативних варіантів стратегії.

2 ІНТЕЛЕКТУАЛЬНИЙ МЕТОД ВИЗНАЧЕННЯ КОНКУРЕНТНОГО ТОВАРУ НА ОСНОВІ ОНЛАЙН-ВІДГУКІВ

2.1 Опис методів збору онлайн-відгуків

Дані є надзвичайно важливим фактором, коли йдеться про отримання інформації про конкретну тему, дослідження, розвідки чи навіть людей. Саме тому вони вважаються життєво важливим компонентом усіх систем, що складають наш сучасний світ. Фактично, дані пропонують широкий спектр застосування та використання в сучасну епоху. Тому незалежно від того, розглядаєте ви можливість цифрової трансформації чи ні, збір даних - це той аспект, від якого ніколи не слід відмахуватися, особливо якщо ви хочете отримувати інформацію, робити прогнози та керувати своєю діяльністю таким чином, щоб створювати значну цінність. Проте багато людей все ще тяжіють до плутанини, коли стикаються з ідеєю збору даних.

Збір даних визначається як систематичний метод отримання, спостереження, вимірювання та аналізу точної інформації для підтримки досліджень, що проводяться групами фахівців незалежно від галузі, до якої вони належать [51]. Хоча методи і цілі можуть відрізнятися в різних галузях, загальні методи збирання даних, що використовуються в процесі, по суті, однакові. Інакше кажучи, існують конкретні стандарти, яких потрібно суворо дотримуватися і виконувати, щоб забезпечити точний збір даних. Не кажучи вже про те, що якщо не надати належного значення відповідним процедурам, можуть виникнути різні проблеми, які вплинуть на проведене дослідження. Найпоширенішим ризиком є нездатність визначити відповіді та зробити правильні висновки для дослідження, а також нездатність підтвердити правильність отриманих результатів. Ці ризики також можуть призвести до сумнівного дослідження, що може сильно вплинути на довіру до вас.

Скраппінг даних - це процес збору даних з веб-сайту та збереження їх у локальному файлі на комп'ютері. Це один із найефективніших інструментів збору даних, який можна використовувати для збору інформації з Інтернету.

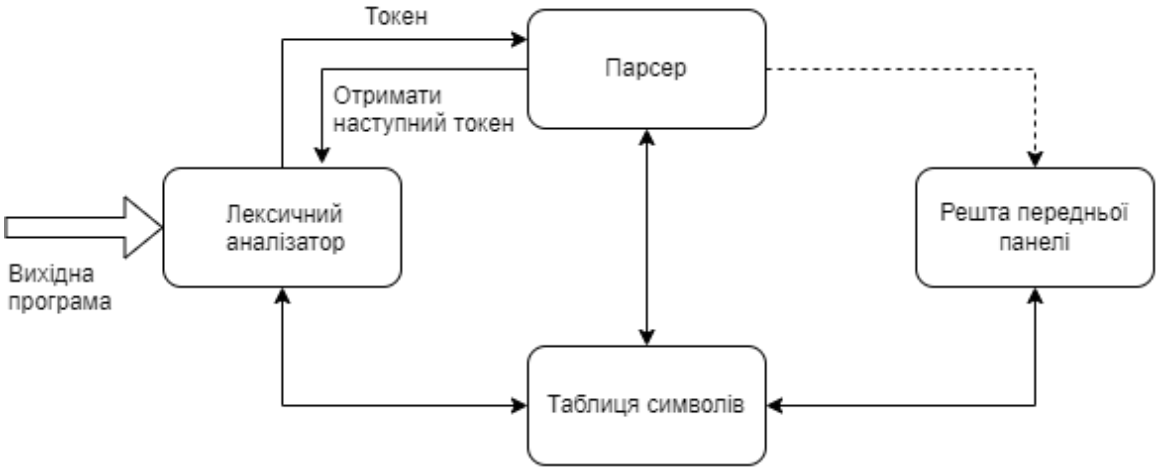


Рисунок 2.1 – процес роботи парсера

Деякі з найбільш популярних способів збору даних включають наступне:

– пошук потенційних покупців;
– проведення маркетингових досліджень;
– пошук бізнес-аналітики;
– надсилання даних про продукт.

Можна налаштувати свої критерії або параметри для вибіркового пошуку конкретних атрибутів, особливо при використанні відповідного інструменту для пошуку даних. Також легко збирати якісні та кількісні дані таким чином, щоб їх можна було легко впровадити у ваше дослідження або бізнес-процедури.

Традиційна граматична вправа синтаксичного розбору, іноді звана аналізом клаузи, передбачає розбивку тексту на складові частини мови з поясненням форми, функції та синтаксичних відносин кожної частини [52]. Це визначається значною мірою на основі вивчення відмінювань і відмінювань мови, які можуть бути доволі складними для мов з великою кількістю слів, що відмінюються. Для розбору фрази типу "людина кусає собаку" необхідно зазначити, що однина іменника "людина" є суб'єктом речення, дієслово

"кусає" - це третя особа однини теперішнього часу дієслова "кусати", а однина іменника "собака" - це об'єкт речення. Для позначення зв'язку між елементами речення іноді використовують такі прийоми, як діаграми речень.

Раніше синтаксичний розбір посідав центральне місце у викладанні граматики в усьому англомовному світі та вважався основою для використання і розуміння письмової мови. Однак нині викладання таких методів уже не актуальне.

Алгоритми розбору природної мови не можуть покладатися на те, що граматика має "приємні" властивості, як це відбувається з граматиками, розробленими вручну для мов програмування. Як згадувалося раніше, деякі граматичні формалізми дуже складно розібрати обчислювально; у загальному випадку, навіть якщо необхідна структура не є контекстно-вільною, для першого проходу використовують якесь контекстно-вільне наближення до граматики. Алгоритми, що використовують контекстно-вільні граматики, часто покладаються на той чи інший варіант алгоритму CYK, зазвичай із деякою евристикою для відсіювання малоймовірних аналізів для економії часу. Однак деякі системи обмінюють швидкість на точність, використовуючи, наприклад, версії алгоритму shift-reduce з лінійним часом. Дещо нещодавньою розробкою став повторний розбір, за якого синтаксичний аналізатор пропонує велику кількість аналізів, а складніша система обирає найкращий варіант. У додатках для розуміння природної мови семантичні синтаксичні аналізатори перетворюють текст на представлення його сенсу.

Завдання синтаксичного аналізатора полягає в тому, щоб визначити, чи можна отримати вхідні дані з початкового символу граматики, і якщо так, то яким чином. Це може бути зроблено двома способами:

Розбір зверху вниз - розбір зверху вниз можна розглядати як спробу знайти найлівіші похідні вхідного потоку шляхом пошуку дерев розбору з використанням розширення зверху вниз заданих правил формальної граматики. Токени споживаються зліва направо. Інклюзивний вибір використовується для розв'язання неоднозначності шляхом розширення всіх

альтернативних правих сторін граматичних правил. Цей підхід відомий як "первісний суп". Дуже схожий на діаграмування речень, "первісний суп" розбиває складові частини речень.

Розбір від низу до верху - синтаксичний аналізатор може почати з вхідних даних і спробувати переписати їх на початковий символ. Інтуїтивно зрозуміло, що синтаксичний аналізатор намагається знайти найосновніші елементи, потім елементи, що містять ці елементи, і так далі. Парсери LR є прикладами парсерів "знизу вгору". Інший термін, що використовується для цього типу синтаксичного аналізатора, - Shift-Reduce parsing.

LL-синтезатори та рекурсивно-висхідні синтаксичні аналізатори є прикладами низхідних синтаксичних аналізаторів, які не вміщують ліві рекурсивні похідні правила. Хоча вважалося, що прості реалізації низхідного синтаксичного аналізу не можуть врахувати пряму й непряму ліву рекурсію та можуть вимагати експоненціального часу та простору при розборі неоднозначних контекстно-вільних граматик, Фрост, Гафіз та Каллаган створили більш складні алгоритми низхідного синтаксичного аналізу, що враховують неоднозначність та ліву рекурсію за поліноміальний час та генерують поліноміального розміру представлення потенційно експоненціального числа дерев розбору. Їхній алгоритм здатний виробляти як крайні ліві, так і крайні праві похідні вхідних даних щодо заданої контекстно-вільної граматики.

Найпростіші API парсеру зчитують весь вхідний файл, виконують деякі проміжні обчислення, а потім записують весь вихідний файл. (Наприклад, багатопрохідні компілятори in-memory).

Ці прості синтаксичні аналізатори не працюватимуть, якщо не вистачає пам'яті для зберігання всього вхідного файлу або всього вихідного файлу. Вони також не працюватимуть з нескінченними потоками даних з реального світу.

Деякі альтернативні підходи API для розбору таких даних:

- Push парсери, які викликають зареєстровані обробники (callbacks), щойно парсер виявляє відповідні лексеми у вхідному потоці (наприклад, Expat);
- Парсери, що тягнуть;
- Інкрементні парсери (наприклад, інкрементні парсери діаграм), які в міру редагування тексту файлу користувачем не вимагають повного повторного розбору всього файлу;
- Активні та пасивні парсери.

2.2 Опис методів автоматизованого визначення мови тексту

Ідентифікація мови ("LI") - це завдання визначення природної мови, якою написаний документ або його частина [28]. Розпізнавання тексту певною мовою відбувається природно для людини, знайомої з цією мовою.

Дослідження в галузі LI спрямовані на імітацію цієї людської здатності розпізнавати конкретні мови. За минулі роки було розроблено низку обчислювальних підходів, які, завдяки використанню спеціально розроблених алгоритмів і структур індексування, здатні визначити мову, що використовується, без втручання людини. Можливості таких систем можна назвати надлюдськими: звичайна людина може визначити кілька мов, а підготовлений лінгвіст або перекладач може бути знайомий із багатьма десятками, але більшість із нас у якийсь момент стикалися з письмовими текстами мовами, які вони не можуть визначити. Однак дослідження в галузі LI спрямовані на розробку систем, здатних ідентифікувати будь-яку людську мову, набір яких налічується тисячами. У широкому сенсі LI може бути застосована до будь-якої модальності мови, включно з мовленням, мовою жестів і рукописним текстом, і актуальна для всіх засобів зберігання інформації, у яких задіяно мову, цифрову чи іншу. Дослідження LI дотепер традиційно фокусувалися на одномовних документах . У монолінгвальному LI завдання полягає в тому, щоб присвоїти кожному документу унікальну мовну

мітку. У деяких роботах повідомлялося про практично ідеальну точність ЛІ великих документів невеликою кількістю мов, що спонукало деяких дослідників назвати її "вирішеним завданням" [29]. Однак для досягнення такої точності необхідно зробити спрощувальні припущення, як-от згадана вище монолінгуальність кожного документа, а також припущення щодо типу та кількості даних і кількості мов, які розглядаються. Можливість точного визначення мови, якою написаний документ, є технологією, що підвищує доступність даних і має широкий спектр застосування. Наприклад, подання інформації рідною мовою користувача виявилося вирішальним фактором у залученні відвідувачів веб-сайту. Методи опрацювання тексту, розроблені в царині опрацювання природної мови та інформаційного пошуку (IR), зазвичай припускають, що мова вхідного тексту відома, а багато методів припускають, що всі документи написані однією мовою. Для застосування методів опрацювання тексту до даних реального світу використовують автоматичний ЛІ, який гарантує, що подальшому опрацюванню піддаватимуться тільки документи відповідними мовами. Під час зберігання та пошуку інформації заведено індексувати документи в багатомовній колекції за мовою, якою вони написані. LI необхідний для колекцій документів, де мови документів не відомі, наприклад, для даних, зібраних із Всесвітньої павутини. Інше застосування LI, яке з'явилося ще до появи обчислювальних методів, - це визначення мови документа для направлення його відповідному перекладачеві. Це застосування стало ще більш помітним завдяки появі методів машинного перекладу (MT) [30]: для того щоб застосувати MT для перекладу документа цільовою мовою, зазвичай необхідно визначити мову джерела документа, і саме це є завданням LI. LI також відіграє певну роль у забезпеченні підтримки документації та використання мов із низькими ресурсами. Однією з царин, де LI часто використовується в цьому відношенні, є створення лінгвістичних корпусів, де LI використовується для обробки цільових веб-пошуків з метою збору текстових ресурсів для мов із низьким рівнем ресурсів.

LI у певному сенсі є особливим випадком категоризації тексту, і попередні дослідження розглядали застосування стандартних методів категоризації тексту до LI дає визначення категоризації тексту, яке можна узагальнити як задачу зіставлення документа із заздалегідь визначеним набором класів. Це дуже широке визначення, і воно справді може бути застосоване до широкого спектра завдань, до числа яких входить і сучасна LI. Архетиповим завданням категоризації тексту є, мабуть, класифікація статей новинних стрічок відповідно до тем, які в них обговорюють, прикладом чого може слугувати набір даних Reuters-21578 [31]. Однак у LI є особливості, які роблять її відмінною від типових завдань категоризації тексту:

1. Категоризація тексту зазвичай використовує статистику частоти слів для моделювання документів, але для цілей ЛІ не існує універсального поняття слова: LI має враховувати мови, у яких пробільні символи не використовуються для позначення меж слів. Крім того, визначення відповідної стратегії токенізації слів для даного документа.

2. У завданнях категоризації тексту набір міток зазвичай застосовний тільки до певного набору. Наприклад, безглуздо запитувати, які з міток Reuters-21578 застосовні до анотації статті з біомедичного журналу. Однак у LI існує чітке поняття мови, яке не залежить від галузі: можна розпізнати, що текст написаний англійською мовою, незалежно від того, з якого він журналу - біомедичного, мікроблогу чи газетної статті.

3. В LI класи можуть бути певною мірою мультимодальними, тобто текст однією і тією ж мовою іноді може бути написаний у різних орфографіях і зберігатися в різних кодуваннях, але відповідати одному класу.

4. У LI мітки не перекриваються і є взаємовиключними, що означає, що текст може бути написаний тільки однією мовою. Це не виключає існування багатомовних документів, що містять текст більш ніж однією мовою, але в цьому випадку документ завжди можна однозначно розділити на одномовні сегменти. Це протилежно категоризації тексту в багатомовних документах, де,

як правило, неможливо пов'язати конкретні сегменти документа з конкретними мітками.

LI - це багата, складна і багатогранна проблема, яка привертає до себе увагу найрізноманітніших дослідницьких спільнот. Точність LI дуже важлива, оскільки часто це перший крок у довших конвеєрах оброблення тексту, тому помилки, допущені в LI, поширюватимуться і знижуватимуть продуктивність наступних етапів. У контрольованих умовах, наприклад, за умови обмеження кількості мов невеликим набором західноєвропейських мов і використання довгих, граматичних і структурованих текстів, як-от урядові документи, як навчальні дані, можна домогтися практично ідеальної точності. Це призвело до того, що багато дослідників вважають LI вирішеною проблемою, як стверджує Макнамі [32]. Однак LI стає набагато складнішим, якщо взяти до уваги особливості реальних даних, як-от дуже короткі документи (наприклад, запити пошукових машин), нелінгвістичний "шум" (наприклад, HTML-розмітка), нестандартне використання мови (наприклад, як у даних соціальних мереж) і документи змішаними мовами (наприклад, повідомлення на багатомовних веб-форумах). Сучасні підходи до LI зазвичай ґрунтуються на даних і засновані на порівнянні нових документів з моделями кожної цільової мови, отриманими з даних. Типи моделей, а також джерела навчальних даних, що використовуються в літературі, різноманітні, і до теперішнього часу в роботах не проводили їхнього систематичного порівняння й оцінювання, що робить диверсифікованим висновок про те, який метод є "найкращим" для LI. Автоматична ідентифікація мови в текстах: Наявні роботи з LI слугують ілюстрацією того, що масштаб і глибина проблеми набагато більші, ніж може здатися на перший погляд. Не будучи розв'язаною проблемою, аспекти LI роблять її архетипічним завданням навчання з тонкощами.

Більшість застосунків НЛП [33], як правило, орієнтовані на конкретну мову і тому вимагають одномовних даних. Для того щоб створити застосунок цільовою мовою, може знадобитися застосувати техніку попереднього опрацювання, яка відфільтрує текст, написаний не цільовою мовою [37]. Для

цього необхідно правильно визначити мову кожного вхідного прикладу. Нижче перераховуються інструменти, які можна використовувати як модулі Python для опрацювання, і наводиться еталон продуктивності, що оцінює швидкість і точність кожного з них.

langdetect - це переробка бібліотеки Google для визначення мови з Java на Python. Просто передати текст імпортованій функції detect, і вона виведе двобуквений код ISO 693 [34] мови, для якої модель дала найвищу довірчу оцінку. Якщо замість цього використовується функція detect_langs, вона виведе список найкращих мов, які передбачила модель, разом з їхніми ймовірностями. Творці бібліотеки рекомендують задавати в DetectorFactory seed деяке число. Це пов'язано з тим, що алгоритм langdetect є недетермінованим, а отже, якщо спробувати запустити його на занадто короткому або неоднозначному тексті, можна отримати різні результати під час кожного запуску. Встановлення приманки забезпечує [35] отримання послідовних результатів під час розробки/оцінки.

spaCy - використовуючи spaCy для своїх потреб NLP, можна додати користувацький компонент визначення мови до існуючого конвеєра spaCy, який дозволить встановити атрибут розширення під назвою .language для об'єкта Doc. Доступ до цього атрибута можна отримати через Doc._.language, який поверне передбачену мову разом із її ймовірністю.

Langid - зокрема, langid може похвалитися своєю швидкістю (докладніше про це нижче). Він працює аналогічно перерахованим вище інструментам, але його можна використовувати і як інструмент командного рядка, виконавши python langid.py.

FastText. Компанія fasttext зазначає, що її попередньо навчена модель [36] ідентифікації мови займає менше ніж 1 МБ пам'яті й водночас здатна класифікувати тисячі документів за секунду.

Було завантажено з сайту Rozetka набір даних з 10000 в середньому відгуків. Вони були анотовані на предмет приналежності до російської або

неросійської мови, а також за іншими ознаками. Нижче наведено короткий порівняльний аналіз чотирьох описаних вище інструментів.

Таблиця 2.1 – порівняльні результати методів

Методи	langdetect	fasttext	langid	spacy-langdetect
Швидкість	130 мс	8 с	1 хв 30 с	2 хв 2 с
Точність	0.860	0.827	0.845	0.845

2.3 Опис методів перекладу тексту

Переклад пройшов через різні визначення, починаючи з просто лінгвістичної діяльності в шістдесяті роки, культурної та комунікативної діяльності в сімдесяті роки і закінчуючи соціально значущою діяльністю, що залучає соціальних агентів, у новітній літературі. Переклад традиційно відомий як "заміна" текстового матеріалу однією мовою еквівалентним текстовим матеріалом іншою мовою. Хаус [5] згадує, що переклад розглядається як заміна тексту вихідною мовою семантично і прагматично еквівалентним текстом мовою перекладу. Він тісно пов'язаний із культурами вихідної та вихідної мов, вимагає високого рівня володіння мовами, усвідомлення всіх контекстуальних чинників, у межах яких виникає дискурс. Немає потреби згадувати, як він вплинув на цивілізації в різних галузях історії людства, гарний приклад - арабо-ісламська цивілізація. Завжди існували різні напрями щодо того, як слід чи не слід перекладати. В арабській традиції існувало два основні методи, які повністю розходилися між собою, а саме: буквальний і вільний метод. У західній традиції нині існують різноманітні методи:

– лінгвістичний;
– герменевтичний;

– інтерпретаційний;
– функціоналістський;
– інтервенціоністський;
– полісистемний;
– метод типологізації тексту.

Насправді, чим більше розвивалося перекладознавство і включало в себе нові методи дослідження, запозичені з інших дисциплін, тим більше змінювалися методи перекладу і текстові стратегії.

Метод перекладу охоплює різні техніки та стратегії, що застосовуються на текстовому рівні для передання тексту з вихідної мови на мову перекладу у світлі завдання на переклад і контекстуальної ситуації спілкування. Рішення про використання того чи іншого методу ухвалюють до початку перекладу, і воно залежить від кількох чинників. На думку Шаффнера [6], первинні норми, попередні норми та процедурні норми керують актом перекладу до і під час процесу. Попередні норми визначають причину вибору конкретного тексту для перекладу, первинні норми визначають, яку загальну стратегію буде використано (іностранізація або одомашнення), процедурні норми стосуються мікроприйомів, які використовуються на рівні слів і речень. Завжди існували суперечки про те, як краще перекладати: "Чи повинні перекладачі перекладати лише якомога точніше та ближче до оригіналу (SL), чи вони повинні взяти певну свободу дій щодо оригіналу та привести свої переклади у відповідність з усіма нормами та нюансами мови перекладу (TL)? Чи повинні вони зберігати колорит оригіналу за допомогою запозичень, транслітерацій, калькування та інших прийомів, чи вони повинні перетворювати засоби вираження смислу вихідного тексту на звичайні, ідіоматичні способи, характерні для цільової аудиторії, мови та тексту? Це можуть бути різні прийоми перекладу: від процедурних прийомів Віная та Дарбельнета [7] (дослівний переклад, калька, транспозиція, адаптація, еквівалентність), зрушень Кетфорда [8], прийомів іншомовного перекладу та доместикації Венуті [9]. Ньюмарк [10] також

класифікував стратегії перекладу на: перенесення, натуралізацію, культурний еквівалент, функціональний еквівалент, описовий еквівалент.

Залежно від того, як ми підходимо до перекладу і які методи використовуємо, його можна поділити на кілька категорій. Сьогодні переклад є високорозвиненою практикою, і існує безліч точок зору та класифікацій, що систематизують різні підходи до нього. Однак найпопулярнішими та найчастіше використовуваними є такі методи:

Інтерпретативний і комунікативний переклад - цей метод спрямований на розуміння та відтворення оригінального тексту, без внесення будь-яких радикальних змін; зазвичай це стосується синхронного та послідовного перекладу. При цьому зберігається мета оригіналу і досягається бажаний ефект. Функція і жанр також залишаються незмінними, стилістичні зміни неприпустимі.

Дослівний переклад - цей метод дуже схожий на перший. Він характеризується конкретним відтворенням мовних елементів вихідного тексту; дослівний переклад. Жодних стилістичних або лінгвістичних змін не повинно бути. Морфологія і синтаксис та/або сенс оригіналу мають бути суворо дотримані. Функція перекладу може бути дещо змінена, оскільки пріоритетом тут є не зміна, а відтворення мовної системи або форматування вихідного тексту.

Вільний переклад - мета вільного перекладу зберегти функцію вихідної мови, навіть якщо від цього страждає загальний зміст. Зміст має залишатися незмінним. Деякі зміни в таких категоріях, як соціальне та культурне середовище, жанр або комунікативний аспект (тональність, діалект), допустимі. Ці зміни залежать від цільової аудиторії (наприклад, якщо текст призначений для дітей), нового призначення (сценічна адаптація), зміни контексту або особистого вибору. Цей метод найпростіший у застосуванні, але він не може бути застосований до будь-якого типу тексту; перш ніж підходити до перекладу "вільно", необхідно взяти до уваги перераховані категорії, що підлягають зміні.

Філологічний переклад - застосовуючи цей метод, перекладач може додавати до перекладу примітки філологічного та історичного характеру з метою не тільки правильного розуміння специфічних термінів і слів, а й уточнення знайомих значень; у цьому разі початковий текст часто стає предметом вивчення, а переклад орієнтований на спеціалізовану аудиторію або студентів.

Крім того, ключову роль у всьому процесі відіграє так звана стратегія перекладу. Існує кілька стратегій, які являють собою окремі методології, підходи і беруть участь у загальному процесі перекладу. Стратегія перекладу залежить від перекладача та його особистого підходу до процесу. Існують також певні техніки перекладу, тобто конкретні вербальні процедури, які можна впізнати в кінцевому результаті; з їхньою допомогою досягається перехід і зачіпаються більш дрібні одиниці тексту.

Azure Cognitive Services пропонує безліч служб штучного інтелекту та когнітивних API, які допоможуть створювати інтелектуальні додатки. Однією з таких служб є Azure Translator. З його допомогою можна перекладати текст у режимі реального часу більш ніж 60 мовами, використовуючи останні інновації в галузі машинного перекладу. Він підтримує широкий спектр варіантів використання, наприклад, переклад для центрів обробки викликів, багатомовних розмовних агентів або спілкування в додатках.

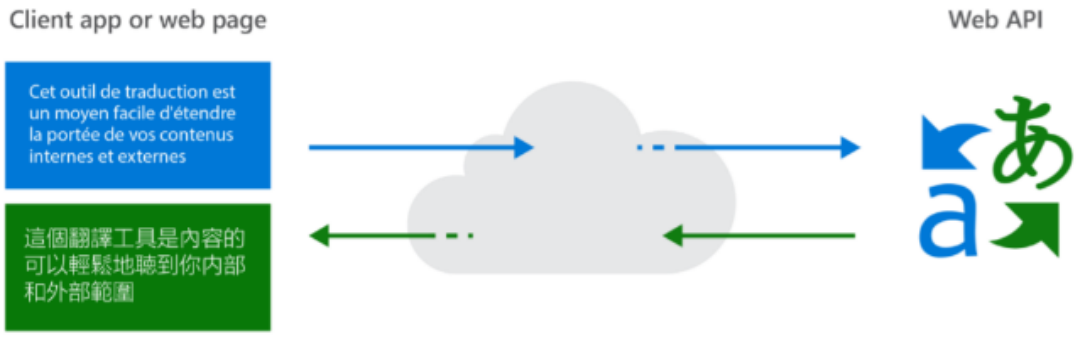


Рисунок 2.2 – принцип роботи Azure Translator

Широкі можливості забезпечення безпеки та відповідності нормативним вимогам в Azure Translate відповідають таким вимогам безпеки:

– Дані клієнта не записуються в постійне сховище. Це відповідає вимозі
– Перегляд і видалення користувацьких даних і моделей у будь-який час.

Тепер служба Azure Translator нативно виконує 2 з 5 вимог без написання коду. Отже, давайте поговоримо про деякі проблеми:

Обмеження API: служба Azure Translator Service має обмеження API в 5 000 символів на виклик. У HTML, де співвідношення тегів до тексту велике, хороше співвідношення тексту до HTML становить від 25 до 70 відсотків. Це означає, що можна легко подолати ліміт у 5 000 символів за допомогою одного виклику для перекладу заголовка HTML, якщо заголовок має досить великий зміст:

– Підтримувати структуру HTML-документа. Це означає, що необхідно:
– Перевірити загальний зміст і вирішити, що потрібно перекласти в першу чергу.
– Пропускати певні теги та вміст.
– Змінити вирівнювання LTR/RTL між мовами.

Якщо метод перекладу строго конкретний, то стратегія і використовувані прийоми індивідуальні і є предметом переваги перекладача. Вони не пов'язані одна з одною, наприклад, під час використання методу вільного перекладу ми можемо використовувати різні стратегії та техніки перекладу. Звісно, за умови, що це дає змогу зберегти кінцевий результат.

2.4 Опис методів класифікації тексту

Завдання класифікації [38] передбачає категоріальні значення для міток, хоча можна використовувати і безперервні значення як мітки. Останній варіант називається проблемою регресійного моделювання. Проблема класифікації тексту тісно пов'язана з проблемою класифікації записів із заданими ознаками ; однак у цій моделі припускають, що в документі використовують тільки інформацію про наявність або відсутність слів. Насправді частота слів також відіграє корисну роль у процесі класифікації, а

типовий розмір області текстових даних (весь обсяг лексикону) набагато більший, ніж типове завдання класифікації із заданими значеннями. Широкий огляд різноманітних методів класифікації можна знайти в [42], а огляд, специфічний для текстової області, - у. Відносну оцінку різних видів методів класифікації текстів можна знайти в [32]. Низка методів, розглянутих у цьому розділі, також була перетворена на програмне забезпечення і перебуває у відкритому доступі за допомогою різних інструментальних засобів, як-от BOW [13], Mallot [36], WEKA 1 і LingPipe 2.

У минулому було випробувано безліч методів, таких як Naive Bayes, SVM, n-gram, n-gram на основі графів [40], пророкування часткової відповідності (PPM) [41], лінійна інтерполяція з після-залежною оптимізацією ваги та мажоритарне голосування для об'єднання кількох класифікаторів [42] тощо.

Дослідники працювали над різними важливими завданнями, пов'язаними з вимірами цієї теми, включно з, окрім іншого, підтримкою мов із низькими ресурсами, як-от непалі, урду та ісландська [33], опрацюванням неструктурованих коротких текстів, які створюють користувачі, тобто мікроблогів, створенням рушія, не залежного від домену. Наявні еталонні рішення підходять до проблеми LID по-різному: LogR [11] використовує дискримінативний підхід із регуляризованою логістичною регресією, TextCat і Google CLD [13] набирають алгоритм на основі N-грам, langid.py [5] покладається на класифікатор Наїва Байєса з мультиноміальною моделлю подій. Видатні на той час результати стали де-факто стандартом LID і сьогодні [7]. Істотним компонентом їхнього методу є використання статистики рангового порядку, званої мірою відстані "поза місцем" [14]. Проблема в їхньому підході полягає в тому, що вони генерують n-грами зі слів, що вимагає токенізації. Однак багато мов, включно з японською та китайською, не мають меж між словами. З огляду на те, що російська - друга за частотою використання мова в Rozetka, потрібен досконаліший підхід для масштабування рішення на всі мови. Як розв'язання проблеми Даннінг [43]

запропонував кращий підхід із включенням n-грам рівня байтів усього рядка замість n-грам рівня символів слів.

Проблема класифікації тексту знаходить застосування в найрізноманітніших галузях текстового аналізу. Деякі приклади галузей, у яких широко використовують класифікацію текстів, наведено нижче:

Підготовка та організація новин: Більшість новинних служб сьогодні є електронними за своєю природою, в яких велика кількість новинних статей створюється організаціями щодня. У таких випадках важко організувати новинні статті вручну. Тому автоматизовані методи можуть бути дуже корисними для категоризації новин на різних веб-порталах [38]. Це застосування також називають сортуванням тексту.

Організація та пошук документів: Вищезазначене застосування загалом корисне для багатьох додатків, крім сортування та організації новин. Для організації документів у багатьох галузях можна використовувати різні методи, засновані на спостереженні. До них належать великі електронні бібліотеки документів, веб-колекції, наукова література або навіть соціальні канали. Ієрархічно організовані колекції документів можуть бути особливо корисні для перегляду та пошуку [19].

Пошук думок: Відгуки або думки клієнтів часто являють собою короткі текстові документи, які можуть бути витягнуті для визначення корисної інформації з відгуку. Подробиці про те, як класифікацію можна використати для пошуку думок, обговорюють у [29].

Класифікація електронної пошти та фільтрація спаму: Часто небажано класифікувати електронну пошту [23], щоб визначити тему листа або визначити небажану пошту [13] автоматизованим способом. Це також називається фільтрацією спаму або фільтрацією електронної пошти.

Для класифікації текстів розроблено безліч методів. Ці класи методів зазвичай існують і для інших даних, таких як кількісні або категоріальні дані. Оскільки текст може бути змодельований як кількісні дані з частотами слів-атрибутів, можна використовувати більшість методів для кількісних даних

безпосередньо в тексті. Однак текст - це особливий вид даних, у якому атрибути слів розріджені, мають високу розмірність і низьку частоту для більшості слів. Тому дуже важливо розробити методи класифікації, які ефективно враховують ці характеристики тексту. Нижче перераховано деякі ключові методи, які зазвичай використовують для класифікації текстів [41].

Дерева рішень: Дерева рішень розроблені з використанням високоієрархічного поділу простору даних з використанням різних текстових ознак. Ієрархічний поділ простору даних створюється для того, щоб створити розділи класів, які є більш перекошеними з точки зору їхнього розподілу за класами. Для даного екземпляра тексту визначається розділ, до якого він з найбільшою ймовірністю належить, і використовується для цілей класифікації [43].

Класифікатори, засновані на шаблонах (правилах) [47]: У класифікаторах, заснованих на правилах, ми визначаємо шаблони слів, які з найбільшою ймовірністю належать до різних класів. Будується набір правил, у яких ліва частина відповідає шаблону слова, а права - мітці класу. Ці правила використовуються для цілей класифікації.

SVM Classifiers [45]: SVM-класифікатори намагаються розділити простір даних за допомогою лінійних або нелінійних меж між різними класами. Ключовим моментом у таких класифікаторах є визначення оптимальних меж між різними класами та їх використання для цілей класифікації.

Нейромережеві класифікатори [46]: Нейронні мережі використовують у найрізноманітніших галузях для цілей класифікації. У контексті текстових даних основною відмінністю нейромережевих класифікаторів є адаптація цих класифікаторів з використанням характеристик слів. Ми зазначаємо, що нейромережеві класифікатори пов'язані з SVM-класифікаторами; справді, вони обидва належать до категорії дискримінативних класифікаторів, що протилежні генеративним класифікаторам [10].

Байєсівські (генеративні) класифікатори [44]: Класифікація проводиться шляхом виведення максимального апостеріорного значення, яке є максимальним P(Ci|X) з урахуванням вищенаведеного припущення, застосовуючи теорему Байєса.

$$P(\mathbf{X} \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)$$

Це припущення значно знижує обчислювальні витрати за рахунок підрахунку тільки розподілу класів. Незважаючи на те, що в більшості випадків це припущення недійсне, оскільки атрибути є залежними, на подив Наївний Байєс показав вражаючі результати. Наївний Байєс - дуже простий алгоритм для реалізації, і в більшості випадків були отримані хороші результати [44]. Він може бути легко масштабований на великі набори даних, оскільки вимагає лінійного часу, а не дорогого ітераційного наближення, як це використовується для багатьох інших типів класифікаторів. Наївний Байєс може страждати від проблеми, званої проблемою нульової ймовірності. Коли умовна ймовірність дорівнює нулю для певного атрибута, він не може дати достовірний прогноз. Це необхідно усунути явним чином за допомогою оцінювача Лапласіана.

Штучна нейронна мережа - це набір пов'язаних між собою вхідних і вихідних блоків, де кожне з'єднання має вагу, пов'язану з ним, створений психологами та нейробіологами для розроблення та тестування обчислювальних аналогів нейронів. На етапі навчання мережа навчається, регулюючи ваги таким чином, щоб мати можливість передбачати правильну мітку класу вхідних кортежів. Нині існує безліч мережевих архітектур, таких як Feed-forward, Convolutional, Recurrent тощо. Вибір відповідної архітектури залежить від сфери застосування моделі. У більшості випадків моделі з прямим передаванням дають досить точні результати, а для додатків обробки зображень краще підходять конволюційні мережі. У моделі може бути кілька прихованих шарів залежно від складності функції, яку відображатиме модель.

Наявність більшої кількості прихованих шарів дає змогу моделювати складні взаємозв'язки, як, наприклад, глибокі нейронні мережі. Однак за наявності великої кількості прихованих шарів потрібно багато часу для навчання та налаштування ваг. Іншим недоліком є погана інтерпретованість моделі порівняно з іншими моделями, такими як Дерева рішень, через невідоме символічне значення, що стоїть за вивченими вагами.

k-Nearest Neighbor - це ледачий алгоритм навчання, який зберігає всі екземпляри, що відповідають навчальним точкам даних у n-вимірному просторі. Коли надходять невідомі дискретні дані, він аналізує найближчу k кількість збережених екземплярів (найближчих сусідів) і повертає найпоширеніший клас як прогноз, а для даних із реальними значеннями повертає середнє значення k найближчих сусідів.

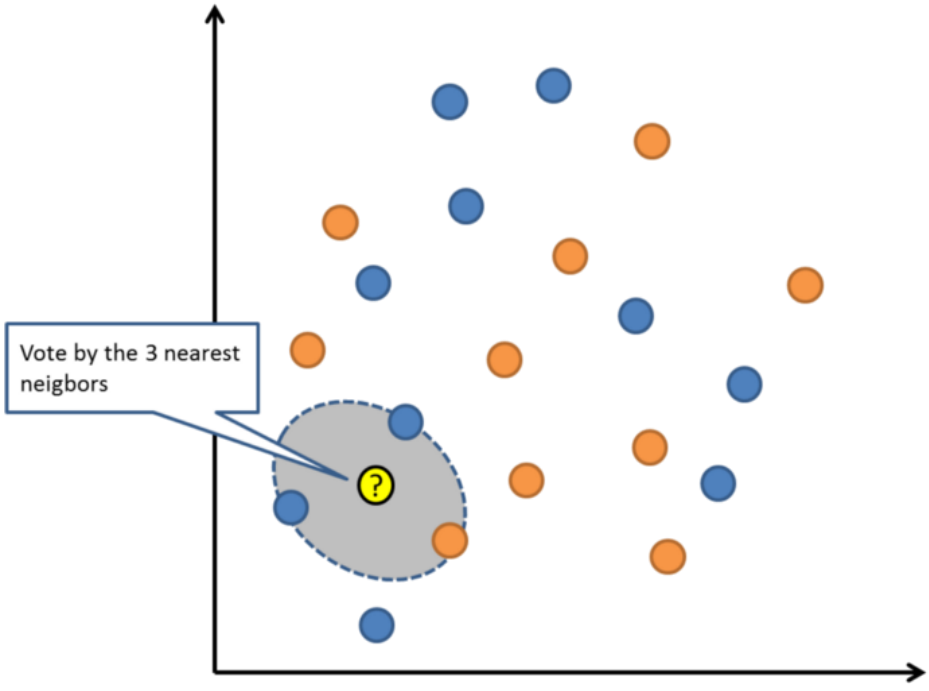


Рисунок 2.3 – принцип роботи метода класифікації тексту kNN

В алгоритмі найближчого сусіда, зваженого за відстанню, зважується внесок кожного з k сусідів відповідно до їхніх відстаней, використовуючи наступний запит, що надає більшу вагу найближчим сусідам.

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$ - Запит на обчислення відстані

Зазвичай KNN стійкий до зашумлених даних, оскільки він усереднює k найближчих сусідів.

Крос-валідація. Перепідгонка [46] - поширена проблема машинного навчання, яка може виникнути в більшості моделей. Для перевірки того, що модель не перепідганяється, можна провести k-кратну крос-валідацію. У цьому методі набір даних випадковим чином розбивається на k взаємовиключних підмножин, кожна з яких приблизно однакового розміру, і одна з них зберігається для тестування, а решта використовуються для навчання. Цей процес повторюється протягом усіх k підмножин.

ROC-крива використовується [47] для візуального порівняння моделей класифікації, яка показує компроміс між коефіцієнтом істинних позитивних результатів і коефіцієнтом помилкових позитивних результатів. Площа під ROC-кривою є мірою точності моделі. Коли модель знаходиться ближче до діагоналі, вона менш точна, а модель з ідеальною точністю матиме площу 1,0

Однак штучні нейронні мережі показали вражаючі результати в більшості реальних застосувань. Вони мають високу стійкість до зашумлених даних і здатні класифікувати ненавчені шаблони. Зазвичай штучні нейронні мережі краще працюють із входами та виходами з безперервними значеннями.

Перед виконанням будь-якого завдання класифікації одним із найфундаментальніших завдань, яке необхідно розв'язати, є представлення документа та вибір ознак. Хоча вибір ознак бажаний і в інших завданнях класифікації, він особливо важливий під час класифікації тексту через високу розмірність текстових ознак та існування нерелевантних (шумних) ознак. У загальному випадку текст може бути представлений двома різними способами. Перший - як мішок слів, у якому документ подано як набір слів разом з їхньою частотою в документі [43]. Таке подання по суті не залежить від послідовності слів у колекції. Другий метод полягає в поданні тексту безпосередньо у вигляді

рядків, у яких кожен документ є послідовністю слів. Більшість методів класифікації тексту використовують подання "мішок слів" через його простоту для цілей класифікації.

Найпоширенішим методом добору ознак [47], який використовується як у контрольованих, так і в неконтрольованих додатках, є видалення стоп-слів і стеммінгу. При видаленні стоп-слів ми визначаємо загальні слова в документах, які не є специфічними або дискримінаційними для різних класів. Під час стеммінгу різні форми одного й того самого слова об'єднуються в одне слово. Наприклад, однина, множина та різні часи об'єднуються в одне слово. Зазначається, що ці методи не є специфічними для задачі класифікації, і часто використовуються в різних додатках без спостереження, таких як кластеризація та індексування. У разі завдання класифікації має сенс контролювати процес відбору ознак за допомогою міток класів. Такий процес відбору гарантує, що для процесу навчання буде відібрано ті ознаки, які мають великий перекіс у бік присутності певної мітки класу.

Використання абсолютної точності класифікації не є єдиним показником, який має відношення до алгоритмів класифікації. Наприклад, у сценаріях зі зміщеними класами, які часто виникають у контексті таких застосунків, як виявлення шахрайства та боротьба зі спамом, помилкова класифікація прикладів одного класу коштує дорожче, ніж іншого. Наприклад, хоча може бути допустимим неправильно класифікувати кілька спам-повідомлень (тим самим допускаючи їх до поштової скриньки), набагато небажаніше неправильно позначити як спам легітимне повідомлення електронної пошти. Проблеми класифікації, чутливі до витрат, природно виникають також у випадках, коли один клас трапляється рідше, ніж інший, і тому бажаніше ідентифікувати рідкісні приклади. У таких випадках бажано оптимізувати зважену за витратами точність процесу класифікації. Зауважимо, що багато загальних методів, які були розроблені для нетекстових даних [40, 42, 45], також можуть бути застосовані до текстових даних, оскільки специфічне представлення ознак не є суттєвим для того, як стандартні

алгоритми можуть бути модифіковані для чутливого до витрат випадку. Гарне розуміння класифікації з урахуванням вартості як для текстових, так і для нетекстових даних можна знайти в [40, 45, 3]. Деякі приклади того, як алгоритми класифікації можуть бути модифіковані простими способами для врахування вартісної чутливості, наведено нижче:

У дереві рішень умова поділу в даному вузлі намагається максимізувати точність його дочірніх вузлів. У випадку з чутливістю до витрат, розбиття розробляється таким чином, щоб максимізувати чутливу до витрат точність. У класифікаторах, заснованих на правилах, правила зазвичай оцінюються кількісно і впорядковуються за допомогою заходів, що відповідають їхній точності прогнозування. У випадку з правилами, чутливими до витрат, правила кількісно оцінюють і впорядковують за їхньою точністю, зваженою за витратами. У байєсівських класифікаторах апостеріорні ймовірності зважуються на вартість класу, для якого робиться прогноз. У лінійних класифікаторах оптимальна гіперплощина, що розділяє класи, визначається у вартісно-зваженому сенсі. Такі витрати, як правило, можуть бути включені в основну цільову функцію. Наприклад, найменша квадратична помилка в цільовій функції методу МНКФ може бути зважена на базові витрати різних класів. У класифікаторі k найближчих сусідів ми повідомляємо зважений за вартістю мажоритарний клас серед k найближчих сусідів тестового екземпляра.

Використання підходу, чутливого до вартості, по суті, є зміною цільової функції класифікації, яка також може бути змодельована як оптимізаційне завдання. У той час як стандартне завдання класифікації, як правило, намагається оптимізувати точність, версія з урахуванням вартості намагається оптимізувати цільову функцію, зважену за вартістю. Більш загальний підхід було запропоновано в [50], в якому було запропоновано мета-алгоритм для оптимізації специфічного показника якості, такого як точність, точність, запам'ятовування або F1-міра. Таким чином, цей підхід узагальнює цей клас методів на будь-яку довільну цільову функцію, що робить його, по суті,

об'єктно-орієнтованим методом класифікації. Узагальнений алгоритм імовірнісного розпізнавання (із заданою цільовою функцією) використовують у поєднанні з класифікатором, що цікавить, для того, щоб отримати мітки класів тестового екземпляра.

2.5 Інтелектуальний метод визначення товару на основі онлайн-відгуків

Для пришвидшення часу на вибір товару був розроблений інтелектуальний метод визначення конкурентного товару на основі онлайн-відгуків [51]. Цей метод ілюструється схемою (рис. 2.4) і є представлений такими кроками:

Крок 1. Підключаємо всі необхідні бібліотеки. В тому числі і бібліотеки для визначення тексту і його подальшого перекладу. (Блок 1).

Крок 2. Вибір сайту (Блок 2).

Крок 3. Добавляємо посилання на категорії товару. (Блок 3).

Крок 4. Виконуємо парсинг відгуків. Далі ділимо їх на id і створюємо текстовий файл. (Блок 4).

Крок 5. Створення БД усіх можливих відгуків (Блок 5).

Крок 6. Підключення відгуків з БД (Блок 5) для подальшої роботи (Блок 6).

Крок 7. Використання алгоритму розпізнавання мови тексту.

Крок 8. Очистка тексту для алгоритму, щоби легше розпізнавати настрій (Блок 7).

Крок 8. Використання векторизації (Блок 8). Для виконання процесу векторизації потрібно провести: токенізацію (Блок 8.1), ігнорування поодиноких символів (Блок 8.3), перетворення тексту в числові вектори та n-грами (Блок 8.5) після чого формуємо масив значень (Блок 8.6).

Крок 8.1 Токенізація (Блок 8.1) - розбиття вихідного тексту на невеликі фрагменти слів або речень, які називаються токенами. Якщо текст розбивається на слова, то це називається "Токенізація слів", а якщо на речення,

то це називається "Токенізація речень". Як правило, для токенізації слів використовується "пробіл", а для токенізації речень - крапки, знак оклику та символ нового рядка. При виконанні токенізації деякі символи, такі як пробіли, розділові знаки ігноруються і не будуть включені в остаточний список токенів. Оцінку якості, зроблену за тестовою вибіркою, можна застосувати для вибору найкращої моделі (Блок 8.2).

Крок 8.2 Процес ігнорування поодиноких символів, які мішають роботі програми (Блок 8.3) проводиться на основі готового набору «стоп-слів» (Блок 8.4). Стоп-слова - які відфільтровуються (тобто зупиняються) до або після обробки даних природної мови (тексту) через їхню незначущість. Не існує єдиного універсального списку стоп-слів, який використовується всіма інструментами обробки природної мови, так само як і узгоджених правил для ідентифікації стоп-слів, ба більше, не всі інструменти навіть використовують такий список.

Крок 8.3 На цьому кроці проводимо перетворення тексту в числові вектори та добавляємо n-грами (Блок 8.5). N-грами - це безперервна послідовність n елементів із заданого зразка тексту або мови. Елементами можуть бути фонеми, склади, літери, слова або базові пари залежно від програми. Зазвичай n-грами збираються з корпусу тексту або мови. Коли елементами є слова, n-грами можуть також називатися шинглами.

Крок 8.4 Створення векторизовані значення тексту для подальшого зберігання в БД (Блок 8.6).

Крок 9 Формування БД перетвореного тексту для вибору моделі класифікатора (Блок 9).

Крок 10 Введення «чистих» даних (Блок 10).

Крок 11 Далі вибір оптимального класифікатора алгоритмами мн (Блок 11).

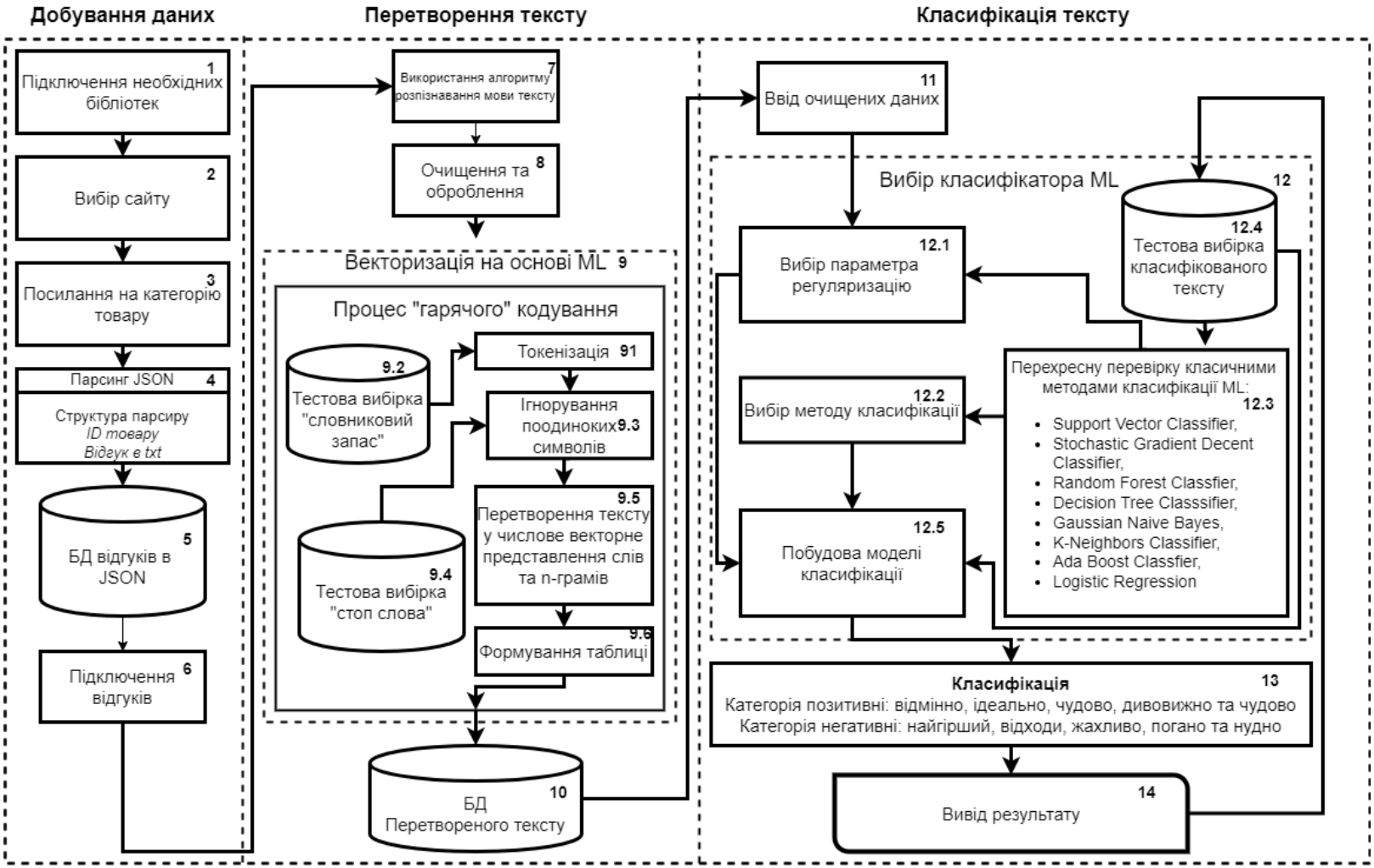


Рисунок 2.4 - Структура інтелектуального методу визначення конкурентного товару на основі онлайн-відгуків

Крок 11.1 Пошук найкращого параметру регуляризації (Блок 11.1). Потрібно використати перехресну перевірку 8 класичними методами класифікації: SVC, SGDC, RFC, DTC, GNB, KNC, ABC, LR (Блок 11.3) [82]. Для перевірки алгоритмів використано навчальну готову вибірку (Блок 11.4).

Крок 11.2 Вибираэмо найкращий метод класифікації (Блок 11.2) з найвищими параметрами регуляризації (Блок 11.1) та створюємо модель класифікації (Блок 11.5) на основі тестової вибірки (Блок 11.4).

Крок 12 Вводиться класифікація даних (Блок 12). На позитивні та негативні коментарі. А саме позитивні відгуки діляться на категорії: відмінно, ідеально, чудово, дивовижно та чудово. Негативні діляться на категорії: найгірший, відходи, жахливо, погано та нудно.

Крок 13 Виведення результату, після якого можна приймати рішення стосовно вигідного вкладання коштів в новий товар.

Далі в наступних розділах описується розробка програмної реалізації інтелектуального методу аналізу настроїв споживачів.

Висновки до розділу 2

1. Збір даних став найважливішою стратегією для багатьох професіоналів і підприємств. Хоча це може бути складним завданням для дослідників-початківців або власників бізнесу, розуміння його методів може сприяти збору даних у найточніший спосіб. Тому далі буде використовуватись активний парсер, тому що він найкраще показав результати.

2. Серед всіх методів можна виділити метод у мові програмування python. Бібліотека LangDetect має більше 100 мов і зможе легко перевірити їх наявність та виділити текст.

3. Microsoft Azure Translator найкраще показав себе і завдяки йому текст можна перекладати використовуючи останні інновації в галузі машинного перекладу.

4. Проблема класифікації є однією з найбільш фундаментальних проблем у літературі з машинного навчання та видобутку даних. У контексті текстових даних проблему можна розглядати як аналогічну проблемі класифікації дискретних наборів атрибутів, коли ігноруються частоти слів. Домени цих наборів досить великі, оскільки включають у себе весь лексикон. Тому методи аналізу тексту мають бути розроблені таким чином, щоб ефективно керувати великою кількістю елементів з різною частотою. Майже всі відомі методи класифікації, такі як дерева рішень, правила, методи Байєса, класифікатори найближчих сусідів, SVM-класифікатори та нейронні мережі, були поширені на текстові дані. Останнім часом значна увага приділяється лінійним класифікаторам, таким як нейронні мережі та SVM-класифікатори, причому останні особливо підходять до характеристик текстових даних. Останніми роками розвиток технологій Інтернету та соціальних мереж призвів до величезного інтересу до класифікації текстових документів, що містять посилання або іншу метаінформацію [44]. Недавні дослідження показали, що включення

інформації про зв'язки в процес класифікації може значно поліпшити якість базових результатів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОГО МЕТОДУ ВИЗНАЧЕННЯ КОНКУРЕНТНОГО ТОВАРУ

3.1 Парсинг онлайн-відгуків

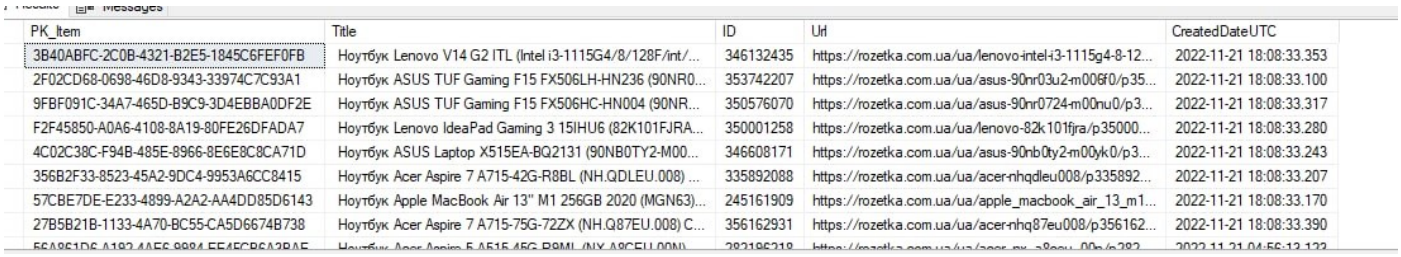
Важливість онлайн-відгуків у прийнятті споживчих рішень уже добре відома. В останні кілька років, коли електронна комерція перебуває на підйомі, її значення зростає з кожним днем. Більшість споживачів вважають за краще ознайомитися з думкою інших людей, перш ніж приступити до покупки обраного ними товару. Більше відгуків про товар, більша ймовірність того, що споживач витратить більше часу на вибір саме цього товару, а не будь-якого іншого, і, найімовірніше, в підсумку купить його. У багатьох дослідженнях уже йшлося про вплив онлайн-відгуків на рішення покупців, аналіз настрою онлайн-відгуків про товар з метою з'ясувати думку покупця про товар, рекомендаційні системи, прогнозування того, що покупець замовить у майбутньому за допомогою алгоритмів машинного навчання та глибокого навчання. Фактор, який відрізняє інші дослідження [7], відрізняється від усіх вищезазначених тим, що навіть відгуки продавців були розглянуті для подальшого аналізу. Відгуки продавців зазвичай присутні для кожного продавця на високому рівні, для всіх замовлень, які продавець виконав. За ними можна судити про роботу продавця на загальному рівні. Але коли споживач хоче вибрати конкретного продавця при замовленні певного товару, необхідно навіть уявити роботу продавця з доставки цього конкретного товару. Слід також враховувати минулий досвід доставки та відгуки споживачів про доставку цього товару. Незважаючи на те, що це більш детальна інформація, вона є важливим аспектом для продавця. Цей аспект врахування відгуків продавців, який не враховували в попередніх аналізах, допомагає нам не тільки отримати чітке уявлення про плюси та мінуси продавця, а й проаналізувати їхню роботу та прийняти рішення про вибір продавця.

Для парсингу сайту Rozetka потрібно підключити необхідні бібліотеки. GetNemItems – на основній сторінці розетка витягуються товари

```
from urllib import response
from bs4 import BeautifulSoup
import requests
import pymssql

urlToScrap = 'https://rozetka.com.ua/ua/notebooks/c80004/'
numberOfItemsToGet = 10
```

Beautiful Soup - це бібліотека на мові Python для вилучення даних з HTML та XML файлів. Вона працює з вашим улюбленим синтаксичним аналізатором, надаючи ідіоматичні способи навігації, пошуку та модифікації дерева розбору. Це зазвичай економить програмістам години або дні роботи. Вилучення інформації з Інтернету часто є єдиним способом отримати доступ до даних. Існує багато інформації, яка недоступна у зручному експорті у форматі CSV або простих у підключенні API. І самі веб-сайти часто є цінними джерелами даних - подумайте, наприклад, який аналіз ви могли б зробити, якби могли завантажити кожне повідомлення на веб-форумі. Python - це інтерпретована, об'єктно-орієнтована, високорівнева мова програмування з динамічною семантикою. Її високорівневі вбудовані структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять її дуже привабливою для швидкого розроблення застосунків, а також для використання як скриптової або клейової мови для з'єднання наявних компонентів. Простий синтаксис Python, що легко вивчається, підкреслює читабельність і, отже, знижує витрати на супровід програм. Python підтримує модулі та пакети, що сприяє модульності програми та повторному використанню коду. Інтерпретатор Python і велика стандартна бібліотека доступні у вихідному або двійковому вигляді безкоштовно для всіх основних платформ і можуть вільно поширюватися.



PK_Item	Title	ID	Url	CreatedDateUTC
3B40ABFC-2C0B-4321-B2E5-1845C6FEF0FB	Ноутбук Lenovo V14 G2 ITL (Intel i3-1115G4/8/128F/int/...	346132435	https://rozetka.com.ua/ua/lenovo-intel-i3-1115g4-8-12...	2022-11-21 18:08:33.353
2F02CD68-0698-46D8-9343-33974C7C93A1	Ноутбук ASUS TUF Gaming F15 FX506LH-HN236 (90NR0...	353742207	https://rozetka.com.ua/ua/asus-90nr03u2-m006f0/p35...	2022-11-21 18:08:33.100
9FBF091C-34A7-465D-B9C9-3D4EBBA0DF2E	Ноутбук ASUS TUF Gaming F15 FX506HC-HN004 (90NR...	350576070	https://rozetka.com.ua/ua/asus-90nr0724-m00nu0/p3...	2022-11-21 18:08:33.317
F2F45850-A0A6-4108-8A19-80FE26DFADA7	Ноутбук Lenovo IdeaPad Gaming 3 15IHU6 (82K101FJRA...	350001258	https://rozetka.com.ua/ua/lenovo-82k101fjra/p35000...	2022-11-21 18:08:33.280
4C02C38C-F94B-485E-8966-8E6E8C8CA71D	Ноутбук ASUS Laptop X515EA-BQ2131 (90NB0TY2-M00...	346608171	https://rozetka.com.ua/ua/asus-90nb0ty2-m00yk0/p3...	2022-11-21 18:08:33.243
356B2F33-8523-45A2-9DC4-9953A6CC8415	Ноутбук Acer Aspire 7 A715-42G-R8BL (NH.QDLEU.008) ...	335892088	https://rozetka.com.ua/ua/acer-nhqdleu008/p335892...	2022-11-21 18:08:33.207
57CBE7DE-E233-4899-A2A2-AA4DD85D6143	Ноутбук Apple MacBook Air 13" M1 256GB 2020 (MGN63)...	245161909	https://rozetka.com.ua/ua/apple_macbook_air_13_m1...	2022-11-21 18:08:33.170
27B5B21B-1133-4A70-BC55-CA5D6674B738	Ноутбук Acer Aspire 7 A715-75G-72ZX (NH.Q87EU.008) C...	356162931	https://rozetka.com.ua/ua/acer-nhq87eu008/p356162...	2022-11-21 18:08:33.390
66A861D6-A182-4AE6-9884-EE4FCB6A2BAE	Ноутбук Acer Aspire 5 A515-45G-R8ML (NX.A8CEU.00N)	282196218	https://rozetka.com.ua/ua/acer_nx_a8ceu_00n/p282...	2022-11-21 04:56:13.123

Рисунок 3.5 – Результати роботи парсингу витягування товарів

Часто програмісти використовують Python через підвищення продуктивності, яку він забезпечує. Оскільки відсутній етап компіляції, цикл "редагування-тестування-налагодження" неймовірно швидкий. Налагоджувати програми на Python дуже просто: помилка або погане введення ніколи не викличуть помилку сегментації. Замість цього, коли інтерпретатор виявляє помилку, він піднімає виняток. Якщо програма не перехоплює виняток, інтерпретатор друкує трасування стека. Налагоджувач на рівні вихідного тексту дає змогу переглядати локальні та глобальні змінні, оцінювати довільні вирази, встановлювати точки зупинки, проходити по коду по рядку за раз і так далі. Відладчик написаний самою мовою Python, що свідчить про інтроспективну силу Python. З іншого боку, часто найшвидшим способом налагодження програми є додавання кількох операторів друку у вихідний текст: швидкий цикл редагування-тестування-налагодження робить цей простий підхід дуже ефективним.

```
        href = row['Url'] + 'comments/'
        rozetkaHtmlPage = requests.get(href, headers=headers)
        rozetkaHtmlSoup = BeautifulSoup(rozetkaHtmlPage.content,
'html.parser')
        commentsHtmlSoup                                        =
rozetkaHtmlSoup.find_all(class_='product-comments__list-
item')
```

У даній частині коду витягується посилання на продукт з бази даних. До цього посилання додається сегмент коментарів. Після цього викачується сторінка і парситься в об’єкти пайтона, які потім використовуються для того, щоби витягнути дані з цієї сторінки.

```
    for commentHtmlSoup in commentsHtmlSoup:
        author                                          =
commentHtmlSoup.find(class_='comment__author').text
        text                                            =
commentHtmlSoup.find(class_='comment__text').text

        essentialPositive = ''
        essentialNegative = ''
        currentMark = 5
        essentialsHtmlSoup                              =
commentHtmlSoup.find_all(class_='comment__essentials-item')
        for essentialHtmlSoup in essentialsHtmlSoup:
            essentialType                               =
essentialHtmlSoup.find(class_='comment__essentials-
label').text
            if essentialType == 'Переваги:':
                essentialPositive                       =
essentialHtmlSoup.find('dd').text
            elif essentialType == 'Недоліки:':
                essentialNegative                       =
essentialHtmlSoup.find('dd').text
            else:
                continue
        starsHtmlSoup                                   =
commentHtmlSoup.find_all(class_='rating-stars__item')
        for starHtmlSoup in starsHtmlSoup:
            pathHtmlSoup = starHtmlSoup.find('path')
            color = pathHtmlSoup.attrs['fill']
            starNumber   =   pathHtmlSoup.attrs['data-index-
number']
            starNumberint = int(starNumber)
            if  color  ==  '#d2d2d2'  and  starNumberint  <
(currentMark-1):
                currentMark = starNumberint
```

Далі добуваються дані з потрібних частин сторінки і записуються в змінні. Перевіряється наявність тексту переваг та недоліків продукту і якщо вони присутні, то записуються в змінні. Те ж саме відбувається з оцінкою продукту.

```
language = detect(text)
```

Визначається мова з якою написаний текст.

```
cursor.callproc('sp_InsertComment',
(str(row['PK_Item']), text, author, language,
essentialPositive, essentialNegative, currentMark))
```

Передаються змінні, в яких записана інформацію про продукт в процедуру, яка запускається в базі даних. Вона створена для того, щоби зберегти ці дані в базі даних у таблиці “T_Comment”.

getCommentsForNewItems – створюється посилання, щоби витягувати відгуки. Після чого скачується сторінка і парситься перетворюючи хтмл в об’єкти пайтона.

Results | Messages

Text	Author	Language	CreatedDateUTC	Essentials_Positive_Text	Essentials_Negative_Text
Ноутбук 10/10 Нет никаких браков и тд. Некоторые п...	Олександра Будовська 29 жовтня 2022	ru	2022-11-21 06:47:24.907	Хорошо работает, выглядит эстетично, подсветка раб...	Нет
Ноутбук пришел вовремя, упаковка целая. При устан...	Александр Тих вчора	ru	2022-11-21 18:09:17.127	Монитор тонкий, ноутбук компактный	Крышка монитора тонкая, не
Отличный, быстрый ноутбук, стоит своих денег	Леся Шикітка 18 вересня 2022	ru	2022-11-21 18:09:19.067	Всё отлично	Нет
Приобрела macbook air с м1 в целях мобильности, ко...	Даяна К 12 жовтня 2022	ru	2022-11-21 18:09:24.453	Лёгкий, красивый, мощный	Конкретно в данной вариаци
Шикарный аппарат. Батарейка слабовата,но то мелоч...	Павел Незнаемый 02 жовтня 2022	ru	2022-11-21 18:09:22.357	Тянет все что нужно, при играх на ультрах сильно гре...	Тускловатый экран
Кидалово с бонусами при покупки А так ноутбук норм	Игорь Кручинин 21 жовтня 2022	ru	2022-11-21 18:09:24.133	Легкий	Нет
Супер все	Анастасія Дикун 02 листопада 2022	ru	2022-11-21 18:09:25.930	NULL	NULL
В целом ноутбук очень хорош для учебы и работы, но ...	Игорь Тимошицкий 18 вересня 2022	ru	2022-11-21 06:47:25.107	Для работы и для серфинга отличный ноут. Возможн...	Проблема с установкой драй
Довольне неплохой нетбук. Покупал жене для офисн...	Віталій Зайцев 05 листопада 2022	ru	2022-11-21 18:09:14.023	Хорошая картинка, мощный проц(оперы хватает с го...	Чистый китаец, из безысход
Наклейки те что слева от тачпада приклеены криво ...	Ілля 19 жовтня 2022	ru	2022-11-21 18:09:14.257	NULL	NULL
Покупал сыну для онлайн учобы в 5 клас 25 августа. ...	Игорь Мирошниченко 05 листопада 2022	ru	2022-11-21 18:09:21.890	Тонкий, экран, подсветка клавиатуры	Не обнаружил
Данный девайс у меня уже по сути два дня. Установи...	Евгений Сурков 18 листопада 2022	ru	2022-11-21 18:09:15.830	Мощный, 16 гб озу	Нет.
Сдал сразу же как принес домой и включил. Неравно...	Сергей 01 вересня 2022	ru	2022-11-21 06:47:25.310	NULL	NULL

Рисунок 3.6 - Результати роботи парсингу витягування відгуків

3.2 Визначення мови тексту та його перекладу

Автоматичне визначення мови - це перший крок до вирішення цілої низки завдань, як-от визначення вихідної мови для машинного перекладу, поліпшення релевантності пошуку шляхом персоналізації результатів пошуку відповідно до мови запиту [3], надання єдиного пошукового рядка для багатомовного словника [4], маркування потоку даних із Rozetka відповідною мовою тощо. Хоча класифікація мов, що належать до розрізнених груп, не є складним завданням, однозначна класифікація мов, що походять з одного джерела, і діалектів, як і раніше, є значною проблемою в галузі обробки природної мови. Звичайні класифікатори, засновані тільки на частоті слів, не

в змозі зробити правильний прогноз для таких схожих мов, і використання сучасних інструментів машинного навчання для виявлення структури мови стало необхідним для підвищення продуктивності класифікатора.

Для визначення мови тексту використовується бібліотека LangDetect.

```
      if      str(row['Essentials_Positive_Text'])      !=      ""      and
row['Essentials_Positive_Text'] is not None:
          translateBody_Essential_Positive = [{
              'text': row['Essentials_Positive_Text']
          }]
          request          =          requests.post(translateConstructed_url,
params=translateParams,                                headers=translateHeaders,
json=translateBody_Essential_Positive)
          response = request.json()
          if request.ok:
              translatedEssentialPositive                                    =
response[0]['translations'][0]['text']
          else:
              translatedEssentialPositive = row['Essentials_Positive_Text']
      else:
          if row['Essentials_Positive_Text'] is not None:
              translatedEssentialPositive = row['Essentials_Positive_Text']
          else:
              translatedEssentialPositive = ''
```

Перевіряється чи текст переваг продукту присутній в базі даних. Якщо ж він присутній, то цей текст записується у тіло запиту. Після цього запит з відповідним тілом та головою, у якій прописані ключі для доступу до нейронної мережі в Azure, відсилається в api сервісу Azure. Потім перевіряється успішність виклику, якщо виклик успішний – записуємо результат у відповідну змінну. Якщо ж ні, то лишаємо змінну пустою та сповіщаємо про помилку.

```
      if      str(row['Essentials_Negative_Text'])      !=      ""      and
row['Essentials_Negative_Text'] is not None:
          translateBody_Essential_Negative = [{
              'text': row['Essentials_Negative_Text']
          }]
          request          =          requests.post(translateConstructed_url,
params=translateParams,                                headers=translateHeaders,
json=translateBody_Essential_Negative)
          response = request.json()
          if request.ok:
              translatedEssentialNegative                                    =
response[0]['translations'][0]['text']
          else:
              translatedEssentialNegative = row['Essentials_Negative_Text']
      else:
          if row['Essentials_Negative_Text'] is not None:
              translatedEssentialNegative = row['Essentials_Negative_Text']
          else:
              translatedEssentialNegative = ''
```

У даному блоці коду виконується все те саме, що і у попередньому, тільки для блоку тексту з недоліками.

```
      cursor.callproc('sp_InsertTranslatedComment',  (str(row['PK_Comment']),
str(row['FK_Item']),       translatedText,       row['Author'],       'uk',
translatedEssentialPositive, translatedEssentialNegative, row['Mark']))
```

Зберігаємо перекладений текст разом із інформацією про продукт у базу даних в табличку “Comment_Translated” за допомогою виклику процедури.

Використовуємо api від Microsoft Azure Translator для перекладання відгуків, які були витягнуті раніше. Azure Translator – це миттєвий або пакетний переклад тексту на більш ніж 100 мов з використанням останніх інновацій у сфері машинного перекладу. Підтримка широкого спектра сценаріїв використання, як-от переклад для центрів оброблення викликів, багатомовні розмовні агенти або спілкування в застосунках.

```
from urllib import response
from bs4 import BeautifulSoup
from langdetect import detect, DetectorFactory
import requests
import pymssql
import uuid, json

maxNumberOfCommentsToTranslate = 10


translateKey = "API KEY"
translateEndpoint                                          =
"https://api.cognitive.microsofttranslator.com"
translateLocation = "eastus"
translatePath = '/translate'
translateConstructed_url    =    translateEndpoint    +
translatePath

translateHeaders = {
    'Ocp-Apim-Subscription-Key': translateKey,
    # location required if you're using a multi-service
or regional (not global) resource.
    'Ocp-Apim-Subscription-Region': translateLocation,
    'Content-type': 'application/json',
    'X-ClientTraceId': str(uuid.uuid4())
}
```

Підключаються потрібні бібліотеки для роботи з веб-запитами, парсингом та визначення мови. Також ініціалізуються глобальні змінні для відправки запитів в api Azure. А саме: ключі доступу, регіон сервера, сервіс, який викликається, глобальні адреса.

Results | Messages

Text	Author	Language	CreatedDateUTC	Essentials_Positive_Text	Essentials_Negative_Text
Ноутбук 10/10 Нет никаких браков и тд. Некоторые п...	Олександра Будовська 29 жовтня 2022	ru	2022-11-21 06:47:24.907	Хорошо работает, выглядит эстетично, подсветка раб...	Нет
Ноутбук пришел вовремя, упаковка целая. При устан...	Александр Тих вчора	ru	2022-11-21 18:09:17.127	Монитор тонкий, ноутбук компактный	Крышка монитора тонкая, не
Отличный, быстрый ноутбук, стоит своих денег	Леся Шикітка 18 вересня 2022	ru	2022-11-21 18:09:19.067	Всё отлично	Нет
Приобрела macbook air с м1 в целях мобильности, ко...	Даяна К 12 жовтня 2022	ru	2022-11-21 18:09:24.453	Лёгкий, красивый, мощный	Конкретно в данной вариации
Шикарный аппарат. Батарейка слабовата,но то мелоч...	Павел Незнаемый 02 жовтня 2022	ru	2022-11-21 18:09:22.357	Тянет все что нужно, при играх на ультрах сильно гре...	Тускловатый экран
Кидалово с бонусами при покупки А так ноутбук норм	Игорь Кручинин 21 жовтня 2022	ru	2022-11-21 18:09:24.133	Легкий	Нет
Супер все	Анастасія Дикун 02 листопада 2022	ru	2022-11-21 18:09:25.930	NULL	NULL
В целом ноутбук очень хорош для учебы и работы, но ...	Игорь Тимошицкий 18 вересня 2022	ru	2022-11-21 06:47:25.107	Для работы и для серфинга отличный ноут. Возможн...	Проблема с установкой драй
Довольне неплохой нетбук. Покупал жене для офисн...	Віталій Зайцев 05 листопада 2022	ru	2022-11-21 18:09:14.023	Хорошая картинка, мощный проц(оперы хватает с го...	Чистый китаец, из безысходи
Наклейки те что слева от тачпада приклеены криво ...	Ілля 19 жовтня 2022	ru	2022-11-21 18:09:14.257	NULL	NULL
Покупал сыну для онлайн учобы в 5 клас 25 августа. ...	Игорь Мирошниченко 05 листопада 2022	ru	2022-11-21 18:09:21.890	Тонкий, экран, подсветка клавиатуры	Не обнаружил
Данный девайс у меня уже по сути два дня. Установи...	Евгений Сурков 18 листопада 2022	ru	2022-11-21 18:09:15.830	Мощный, 16 гб озу	Нет.
Сдал сразу же как принес домой и включил. Неравно...	Сергей 01 вересня 2022	ru	2022-11-21 06:47:25.310	NULL	NULL

Text	Author	Language	CreatedDateUTC	Essentials_Positive_Text	Essentials_Negative_Text
Ноутбук 10/10 Немає шлюбів і так далі. Деякі писа...	Олександра Будовська 29 жовтня 2022	uk	2022-11-21 07:30:38.607	Добре працює, естетично виглядає, працює підсвіч...	Ні
Ноутбук прибув вчасно, упаковка ціла. При установ...	Александр Тих вчора	uk	2022-11-21 18:09:44.677	Монітор тонкий, ноутбук компактний	Кришка монітора тонка, на ssd нем
Відмінний, швидкий ноутбук, вартий своїх грошей	Леся Шикітка 18 вересня 2022	uk	2022-11-21 18:09:45.823	Все чудово	Ні
Придбали macbook air з м1 з метою мобільності, як...	Даяна К 12 жовтня 2022	uk	2022-11-21 18:09:47.067	Легкий, красивий, потужний	Конкретно в цій варіації - всього 25
Шикарний пристрій. Батарея слабка, але потім дріб...	Павел Незнаемый 02 жовтня 2022	uk	2022-11-21 18:09:48.320	Тягне все необхідне, при грі на ультрас сильно розі...	Тьмяний екран
Кідалово з бонусами при покупці І так норми ноутбу...	Игорь Кручинин 21 жовтня 2022	uk	2022-11-21 18:09:48.980	Світлий	Ні
Супер все	Анастасія Дикун 02 листопада 2022	uk	2022-11-21 18:09:49.270	NULL	NULL
В цілому ноутбук дуже хороший для навчання і робо...	Игорь Тимошицкий 18 вересня 2022	uk	2022-11-21 07:30:39.443	Для роботи і для серфінгу відмінний ноутбук. Можл...	Виникла проблема з установкою др
Досить непоганий нетбук. Я купив дружину для офі...	Віталій Зайцев 05 листопада 2022	uk	2022-11-21 18:09:50.620	Хороша картинка, потужний відсоток(опера досить ...	Чисто китайці, з відчаю довелося к
Наклейки, розташовані ліворуч від тачпада, криво п...	Ілля 19 жовтня 2022	uk	2022-11-21 18:09:51.043	NULL	NULL
Я купила сина на онлайн-учоб в 5 класі 25 серпня. ...	Игорь Мирошниченко 05 листопада 2...	uk	2022-11-21 18:09:52.043	Тонка, екранна, клавіатура з підсвічуванням	Не знайдено
У мене цей пристрій вже в основному два дні. Вста...	Евгений Сурков 18 листопада 2022	uk	2022-11-21 18:09:53.153	Потужний, 16 ГБ оперативної пам'яті	Ні.
Я повернув його, як тільки приніс додому, і включив...	Сергей 01 вересня 2022	uk	2022-11-21 07:30:39.903	NULL	NULL

Рисунок 3.7 – Результати визначення та перекладу мови до і після

3.3 Реалізація програмного забезпечення методу

На першому етапі потрібно спочатку імпортувати наступні бібліотеки, sklearn.feature_extraction.text:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
import numpy as np
import pandas as pd
import os
import re
```

TF-IDF - це числова статистика, що покликана відобразити, наскільки важливим є слово для документа в колекції або корпусі.[1] Її часто використовують як ваговий коефіцієнт під час пошуку інформації в інформаційному пошуку, текстовому аналізі та моделюванні користувачів. Значення tf-idf збільшується пропорційно кількості разів, коли слово трапляється в документі, і компенсується кількістю документів у корпусі, що містять це слово, що допомагає скоригувати той факт, що деякі слова взагалі трапляються частіше. tf-idf - одна з найпопулярніших схем зважування термінів на сьогоднішній день. Дослідження, проведене 2015 року, показало, що 83% текстових рекомендаційних систем в електронних бібліотеках використовують tf-idf.

Далі завантажуємо файли в Jupyter у форматі txt з розширення tf-8. Це проєкт, метою якого є розробка програмного забезпечення з відкритим вихідним кодом, відкритих стандартів і послуг для інтерактивних обчислень на різних мовах програмування. Документ Jupyter Notebook являє собою JSON-файл, що відповідає версійній схемі, який зазвичай закінчується розширенням ".ipynb". Основними частинами Jupyter Notebook є: Метадані, формат блокнота і список комірок. Метадані - це словник даних, що містить визначення для налаштування та відображення блокнота. Формат блокнота - це номер версії програми. Список комірок - це різні типи комірок для

Markdown (відображення), коду (для виконання) і виведення комірок типу коду[21].

Хоча ".ipynb" і JSON є найпоширенішими і використовуваними за замовчуванням форматами, можна відмовитися від деяких функцій (таких як зберігання зображень і метаданих) і зберігати блокнот як документи у форматі markdown, використовуючи розширення, як-от JupyText.[22] Jupytext часто використовують у поєднанні з контролем версій, щоб спростити диференціювання та об'єднання блокнотів.:

```
reviews_train = []
for line in open('./train.txt', 'r', encoding = "utf-8"):
reviews_train.append(line.strip())
reviews_test = []
for line in open('./test.txt', 'r', encoding = "utf-8"):
reviews_test.append(line.strip())
```

Перевіряємо 5-ий рядок для того, щоби зрозуміти, що всі файли були успішно завантажені та конвертовані:

```
reviews_train[5]
'12254999 :Еще хороша новина, прийшло повітрям ПО :1.1.3 Build
20180124 rel.52299 (EU) [версия v.1.0] додаткова функція режим моста
(WDS)'
```

Замінюємо символи, тому що алгоритму важко розпізнавати необроблений текст. Потрібно зробити очистку, щоби процес показав правильні результати:

```
import re
REPLACE_NO_SPACE =
re.compile("(\.)|(\;)|(\:)|(\!)|(\')|(\?)|(\,)|(\")|(\()|(\))|(\[)|(\])|(\
d+)")
REPLACE_WITH_SPACE = re.compile("(<br\s*/><br\s*/>)|(\-)|(\/)")
```

```
NO_SPACE = ""
SPACE = " "
def preprocess_reviews(reviews):
reviews = [REPLACE_NO_SPACE.sub(NO_SPACE, line.lower()) for line in
reviews]
reviews = [REPLACE_WITH_SPACE.sub(SPACE, line) for line in reviews]
return reviews
reviews_train_clean = preprocess_reviews(reviews_train)
reviews_test_clean = preprocess_reviews(reviews_test)
```

Перевіряємо новий очищений варіант того ж самого рядка:

```
reviews_train_clean[5]
' ще хороша новина прийшла по повітрю по build rel eu версія v
додаткова функція режим моста wds'
```

Забираємо стоп-слова. Стоп-слова - це слова в будь-якій мові, які не додають багато сенсу в речення. Їх можна сміливо ігнорувати без шкоди для сенсу речення. Для деяких пошукових систем це найпоширеніші короткі функціональні слова, такі як the, is, at, which і on. У цьому разі стоп-слова можуть спричинити проблеми під час пошуку фраз, що включають їх, особливо в таких назвах, як "The Who" або "Take That". Якщо перед нами стоїть завдання класифікації тексту або аналізу настроїв, то ми маємо видалити стоп-слова, оскільки вони не дають жодної інформації нашій моделі, тобто виключають непотрібні слова з нашого корпусу, але якщо перед нами стоїть завдання перекладу мови, то стоп-слова є корисними, оскільки їх доводиться перекладати разом з іншими словами.

Не існує жорсткого і швидкого правила, коли слід видаляти стоп-слова. Їх треба видаляти, якщо завдання - класифікація мов, фільтрація спаму, генерація підписів, генерація автотегів, аналіз настрою або щось пов'язане з класифікацією тексту:

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
russian_stop_words = stopwords.words('russian')
def remove_stop_words(corpus):

    removed_stop_words = []
    for review in corpus:
        removed_stop_words.append(
            ' '.join([word for word in review.split()
                      if word not in russian_stop_words])
        )
    return removed_stop_words
    no_stop_words = remove_stop_words(reviews_train_clean)
```

Перевіряємо наявність видалення стоп-слів:

```
    no_stop_words[5]
'хороша новина прийшла повітрю build rel eu версія v додаткова функція режим
мосту wds'
```

Добавляємо векторизацію. Токенізація має важливий вплив на інші етапи роботи. Токенізатор розбиває неструктуровані дані та текст природною мовою на фрагменти інформації, які можна розглядати як дискретні елементи. Зустрічі токенів у документі можуть бути використані безпосередньо як вектор, що представляє цей документ.

Це відразу ж перетворює неструктурований рядок (текстовий документ) на числову структуру даних, придатну для машинного навчання. Вони також можуть бути використані безпосередньо комп'ютером для запуску корисних дій і відповідей. Або вони можуть бути використані в конвеєрі машинного навчання як характеристики, що спричиняють складніші рішення або поведінку:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(binary=True)
cv.fit(reviews_train_clean)
X = cv.transform(reviews_train_clean)
X_test = cv.transform(reviews_test_clean)
```

Добавляємо метод нормалізації «лематизація». Під час стеммінгу частину слова просто відсікають у хвостовій частині, щоб отримати основу слова. Існують різні алгоритми, що дають змогу визначити, скільки символів потрібно відрізати, але алгоритми не знають значення слова в мові, до якої воно належить. У лематизації, з іншого боку, алгоритми володіють цими знаннями. Фактично, можна навіть сказати, що ці алгоритми звертаються до словника, щоб зрозуміти значення слова, перш ніж скоротити його до кореневого слова, або леми.

Так, алгоритм лематизації знатиме, що слово better походить від слова good, і, отже, лемма - good. Але алгоритм стеммінгу не зможе зробити те саме. Може відбутися надлишковий або недостатній стеммінг, і слово better може бути скорочено до bet або bett, або просто збережено як better. Але при стемінгу немає можливості скоротити його до кореневого слова good. У цьому, по суті, і полягає різниця між стеммінгом і лематизацією.:

```
def get_lemmatized_text(corpus):
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
return [' '.join([lemmatizer.lemmatize(word) for word in 
review.split()]) for review in corpus]
lemmatized_reviews = get_lemmatized_text(reviews_train_clean)
```

Побудова моделі класифікації методом SVM. Support Vector Machine або SVM - один із найпопулярніших алгоритмів контрольованого навчання, який використовується як для вирішення завдань класифікації, так і регресії. Однак здебільшого його використовують для розв'язання задач класифікації в машинному навчанні. Мета алгоритму SVM - створити найкращу лінію або межу прийняття рішення, яка може розділити n-вимірний простір на класи, щоб у майбутньому ми могли легко віднести нову точку даних до потрібної категорії. Ця найкраща межа прийняття рішення називається гіперплощиною. SVM вибирає крайні точки/вектори, які допомагають у створенні гіперплощини. Ці крайні випадки називаються опорними векторами, і тому алгоритм називається Support Vector Machine:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
target = [1 if i < 481 else 0 for i in range(962)]
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)
X_train, X_val, y_train, y_val = train_test_split(
X, target, train_size = 0.75
)
for c in [0.01, 0.05, 0.25, 0.5, 1]:
svm = LinearSVC(C=c)
svm.fit(X_train, y_train)
print ("Accuracy for C=%s: %s"
      % (c, accuracy_score(y_val, svm.predict(X_val))))
final_svm_ngram = LinearSVC(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
% accuracy_score(target, final_svm_ngram.predict(X_test)))
```

Логістична регресія - це процес моделювання ймовірності дискретного результату з урахуванням вхідної змінної. Найчастіше логістична регресія моделює бінарний результат; те, що може набувати двох значень, наприклад, істина/брехня, так/ні і так далі. Мультиноміальна логістична регресія може моделювати сценарії, в яких існує більше двох можливих дискретних результатів. Логістична регресія є корисним методом аналізу для розв'язання завдань класифікації, коли ви намагаєтеся визначити, чи підходить новий зразок до будь-якої категорії. Оскільки деякі аспекти кібербезпеки являють собою проблеми класифікації, наприклад, виявлення атак, логістична регресія є корисним аналітичним методом. Цей тип статистичної моделі (також відомий як логіт-модель) часто використовується для класифікації та прогностичної аналітики. Логістична регресія оцінює ймовірність настання події, наприклад, проголосував чи не проголосував виборець, на основі заданого набору незалежних змінних. Оскільки результат є ймовірністю,

залежна змінна обмежена між 0 і 1. У логістичній регресії до шансів застосовується логіт-перетворення, тобто ймовірність успіху ділиться на ймовірність невдачі:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
target = [1 if i < 481 else 0 for i in range(962)]
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)
X_train, X_val, y_train, y_val = train_test_split(
X, target, train_size = 0.75
)
for c in [0.01, 0.05, 0.25, 0.5, 1]:
lr = LogisticRegression(C=c)
lr.fit(X_train, y_train)
print ("Accuracy for C=%s: %s"
      % (c, accuracy_score(y_val, lr.predict(X_val))))
final_ngram = LogisticRegression(C=0.5)
final_ngram.fit(X, target)
print ("Final Accuracy: %s"
% accuracy_score(target, final_ngram.predict(X_test)))
```

Побудова моделі класифікації методом Random forest classifier. Random forests - це алгоритм контрольованого навчання. Він може використовуватися як для класифікації, так і для регресії. Це найбільш гнучкий і простий у використанні алгоритм. Ліс складається з дерев. Вважається, що чим більше дерев, тим надійніший ліс. Random forests створює дерева рішень на випадково обраних зразках даних, отримує передбачення від кожного дерева і вибирає найкраще рішення шляхом голосування. Він також забезпечує досить хороший показник важливості ознаки. Випадкові ліси мають безліч застосувань, таких як рекомендаційні системи, класифікація зображень і вибір ознак. Його можна використовувати для класифікації лояльних претендентів кредитів, виявлення шахрайської діяльності та прогнозування захворювань.

Він лежить в основі алгоритму Борута, який вибирає важливі ознаки в наборі даних. Random forests також пропонує хороший показник відбору ознак. Scikit-learn надає додаткову змінну з моделлю, яка показує відносну важливість або внесок кожної ознаки в передбачення. Він автоматично обчислює бал релевантності кожної ознаки на етапі навчання. Потім він зменшує значущість так, щоб сума всіх оцінок дорівнювала 1. Ця оцінка допоможе вибрати найважливіші ознаки і відкинути найменш важливі для побудови моделі.:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
target = [1 if i < 481 else 0 for i in range(962)]
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)
X_train, X_val, y_train, y_val = train_test_split(
X, target, train_size = 0.75
)
for c in [0.01, 0.05, 0.25, 0.5, 1]:
RF = RandomForestClassifier(C=c)
RF.fit(X_train, y_train)
print ("Accuracy for C=%s: %s"
      % (c, accuracy_score(y_val, RF.predict(X_val))))
final_svm_ngram = RandomForestClassifier(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
% accuracy_score(target, final_RF_ngram.predict(X_test)))
```

Побудова моделі класифікації методом дерева рішень у машинному навчанні. Методологія дерева рішень - це широко використовуваний метод інтелектуального аналізу даних для створення систем класифікації на основі декількох коваріатів або для розробки алгоритмів прогнозування для цільової змінної. Цей метод класифікує сукупність на сегменти, схожі на гілки, які будують перевернуте дерево з кореневим вузлом, внутрішніми вузлами та

вузлами листя. Алгоритм є непараметричним і може ефективно працювати з великими, складними наборами даних без нав'язування складної параметричної структури. Коли розмір вибірки досить великий, дані дослідження можна розділити на навчальні та перевірочні набори даних. Використання навчального набору даних для побудови моделі дерева рішень і валідаційного набору даних для ухвалення рішення про відповідний розмір дерева, необхідного для досягнення оптимальної кінцевої моделі:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.datasets import load_iris
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
target = [1 if i < 481 else 0 for i in range(962)]
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)
X_train, X_val, y_train, y_val = train_test_split(
X, target, train_size = 0.75
)
for c in [0.01, 0.05, 0.25, 0.5, 1]:
DT = DecisionTreeClassifier(C=c)
DT.fit(X_train, y_train)
print ("Accuracy for C=%s: %s"
      % (c, accuracy_score(y_val, DT.predict(X_val))))
final_svm_ngram = DecisionTreeClassifier(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
% accuracy_score(target, final_DT_ngram.predict(X_test)))
```

Побудова моделі класифікації методом стохастичного градієнта. Стохастичний градієнтний спуск (SGD) - Стохастичний градієнтний спуск (часто скорочено SGD) - це ітераційний метод оптимізації об'єктивної функції з відповідними властивостями гладкості (наприклад, диференційованої або субдиференційованої). Його можна розглядати як стохастичну апроксимацію оптимізації градієнтного спуску, оскільки він замінює фактичний градієнт

(розрахований за всім набором даних) його оцінкою (розрахованою за випадково обраною підмножиною даних). Особливо в задачах оптимізації високої розмірності це знижує дуже високе обчислювальне навантаження, забезпечуючи швидші ітерації в обмін на нижчу швидкість збіжності:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import SGDClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
target = [1 if i < 481 else 0 for i in range(962)]
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)
X_train, X_val, y_train, y_val = train_test_split(
X, target, train_size = 0.75
)
for d in [0.01, 0.05, 0.25, 0.5, 1]:
SGD = SGDClassifier(D=d)
SGD.fit(X_train, y_train)
print ("Accuracy for C=%s: %s"
      % (c, accuracy_score(y_val, SGD.predict(X_val))))
final_ngram = SGDClassifier(C=0.5)
final_ngram.fit(X, target)
print ("Final Accuracy: %s"
      % accuracy_score(target, final_ngram.predict(X_test)))
```

Побудова моделі класифікації методом наївного баєсівського класифікатора. Байєсівські методи останніми роками стають дедалі популярнішими під час аналізу геостатистичних даних. Байєсівська парадигма забезпечує послідовний підхід до визначення складних ієрархічних моделей для складних даних, а останні обчислювальні досягнення зробили підгонку моделей у таких ситуаціях здійсненною. Недоліком використання байєсівських методів є те, що всі спільні розподіли процесів і параметрів мають бути задані, часто через набір умовних розподілів. Припущення про розподіл не є необхідними в класичній геостатистиці, яка вимагає тільки

вказівки середніх, варіацій і коваріацій. Незважаючи на необхідність більш обмежувальних припущень, потужною перевагою байєсівського підходу є те, що він дає змогу глибше зрозуміти природу процесу Y:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.datasets import load_iris
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
target = [1 if i < 481 else 0 for i in range(962)]
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)
X_train, X_val, y_train, y_val = train_test_split(
X, target, train_size = 0.75
)
for c in [0.01, 0.05, 0.25, 0.5, 1]:
DT = DecisionTreeClassifier(C=c)
DT.fit(X_train, y_train)
print ("Accuracy for C=%s: %s"
      % (c, accuracy_score(y_val, DT.predict(X_val))))
final_svm_ngram = DecisionTreeClassifier(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
% accuracy_score(target, final_DT_ngram.predict(X_test)))
```

Побудова моделі класифікації k-найближчих сусідів. Алгоритм k-найближчих сусідів (k-NN) — це непараметричний контрольований метод навчання, вперше розроблений Евелін Фікс і Джозефом Ходжесом у 1951 році[1], а пізніше розширений Томасом Ковером.[2] Він використовується для класифікації та регресії. В обох випадках вхідні дані складаються з k найближчих навчальних прикладів у наборі даних. Результат залежить від того, для класифікації чи регресії використовується k-NN. У класифікації k-NN результатом є членство в класі. Об’єкт класифікується більшістю голосів його сусідів, причому об’єкт призначається до класу, який є найпоширенішим серед k його найближчих сусідів (k — додатне ціле число, зазвичай невелике).

Якщо k = 1, то об’єкт просто призначається до класу цього єдиного найближчого сусіда. k-NN — це тип класифікації, де функція апроксимується лише локально, а всі обчислення відкладено до оцінки функції. Оскільки цей алгоритм покладається на відстань для класифікації, якщо об’єкти представляють різні фізичні одиниці або мають дуже різні масштаби, то нормалізація навчальних даних може значно підвищити їх точність:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
target = [1 if i < 481 else 0 for i in range(962)]
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))

ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)
X_train, X_val, y_train, y_val = train_test_split(
      X, target, train_size = 0.75
)
for c in [0.01, 0.05, 0.25, 0.5, 1]:
      KN = KNeighborsClassifier(C=c)
      KN.fit(X_train, y_train)
      print ("Accuracy for C=%s: %s"
            % (c, accuracy_score(y_val, KN.predict(X_val))))
final_svm_ngram = KNeighborsClassifier(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
      % accuracy_score(target, final_KN_ngram.predict(X_test)))
```

Побудова моделі класифікації методом AdaBoostClassifier. AdaBoost, скорочення від Adaptive Boosting, є метаалгоритм статистичної класифікації. Його можна використовувати у поєднанні з багатьма іншими типами алгоритмів навчання підвищення продуктивності. Вихідні дані інших алгоритмів навчання («слабкі учні») об'єднуються у виважену суму, що становить остаточний результат посиленого класифікатора. Зазвичай, AdaBoost представлений для бінарної класифікації, хоча його можна

узагальнити на кілька класів або обмежених інтервалів на реальній лінії. AdaBoost є адаптивним у тому сенсі, що наступні слабкі учні налаштовуються на користь тих екземплярів, які були неправильно класифіковані попередніми класифікаторами. У деяких завданнях він може бути менш схильний до проблеми перенавчання, ніж інші алгоритми навчання. Окремі учні можуть бути слабкими, але якщо продуктивність кожного з них трохи краща за випадкове вгадування, можна довести, що остаточна модель сходиться до сильного учня.:

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
target = [1 if i < 481 else 0 for i in range(962)]
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)
X_train, X_val, y_train, y_val = train_test_split(
      X, target, train_size = 0.75
)
for d in [0.01, 0.05, 0.25, 0.5, 1]:
      Ada = AdaBoostClassifier(D=d)
      Ada.fit(X_train, y_train)
      print ("Accuracy for C=%s: %s"
            % (c, accuracy_score(y_val, Ada.predict(X_val))))
final_ngram = AdaBoostClassifier(C=0.5)
final_ngram.fit(X, target)
print ("Final Accuracy: %s"
      % accuracy_score(target, final_ngram.predict(X_test)))
```

3.4 Інтерпретація отриманих результатів

Споживчі настрої відіграють важливу роль у розробленні політики та прогнозуванні споживання у зв'язку з багатством інформації, що міститься в такій мірі [51]. На жаль, така інформація не має чіткого визначення в

літературі. Найважливішими факторами, які впливають на настрої споживачів, є їхні власні уявлення та очікування щодо економічних умов, ефективність державної політики та новинні повідомлення.

Представлення тексту є найбільш важливим питанням. У більшості літератури дається статистичне синтаксичне рішення для подання тексту. Однак модель подання залежить від необхідної інформації. Концептуальна база або семантичне представлення відгуків потребує додаткових досліджень. Найкращу класифікацію здійснюватимуть з урахуванням семантики, можливості подання документів на основі семантики. З додаванням онтології та семантики для представлення відгуків, точність і процес класифікації буде покращено. Тому визначення ознак, що відображають семантичний зміст, є однією з важливих областей дослідження. Загальна проблема множинного навчання в присутності шуму є надзвичайно складною проблемою, яка тільки зараз формулюється і, ймовірно, потребуватиме додаткової роботи для успішного розроблення стратегій пошуку природи множини, що лежить в основі. Для автоматичної класифікації документів [52] було запропоновано кілька алгоритмів або комбінацій алгоритмів у вигляді гібридних підходів. Серед цих алгоритмів SVM, NB, kNN та їхні гібридні системи з комбінацією різних інших алгоритмів і технік добору ознак показано як найбільш придатні в наявній літературі. Однак метод добре працює у фільтрації спаму, вимагає невеликої кількості навчальних даних для оцінки параметрів, необхідних для класифікації. Наївний Байєс добре працює на числових і текстових даних, простий у реалізації порівняно з іншими алгоритмами, однак припущення про умовну незалежність порушується в реальних даних і працює дуже погано, коли ознаки дуже корельовані й не враховує частоти зустрічних слів (Табл.3.1).

SVM-класифікатор був визнаний одним із найефективніших методів класифікації текстів у порівнянні алгоритмів машинного навчання під спостереженням [34]. SVM краще вловлює невід'ємні характеристики даних і використовує принцип мінімізації структурного ризику (SRM), що мінімізує

верхню межу помилки узагальнення (краще, ніж принцип мінімізації емпіричного ризику), також здатність до навчання може бути незалежною від розмірності простору ознак та глобальних мінімумів проти локальних мінімумів, проте SVM зазнає деяких труднощів у налаштуванні параметрів і виборі ядра. Якщо для k-NN використовують відповідну попередню обробку, то цей алгоритм продовжує досягати дуже хороших результатів і добре масштабується зі зростанням кількості документів, чого не можна сказати про SVM [45]. Що стосується наївного Байєса [44], то він також показав хороші результати за відповідного попереднього опрацювання. Алгоритм k-NN показав хороші результати, оскільки враховуються більш локальні характеристики документів, однак час класифікації великий і важко знайти оптимальне значення k.

Таблиця 3.1. Результати перехресної оцінки

№	Метод	Оцінка прогнозу
1	SVM	0.88
2	STD	0.84
3	RFC	0.83
4	DTC	0.77
5	GNB	0.87
6	KNC	0.86
7	ADC	0.86
8	LR	0.83

Під час упорядкування (табл. 3.2) за індексами n-грам було виділено 10 позитивних і негативних слів. Відповідно, для товару з id - 122360970 було виділено такі позитивні слова з найбільшим індексом n-грам: "топ" (2,57) і "хороший" (1,82). Також було виявлено негативні слова з найменшим n-грамним індексом - "найгірший" (-0,77), "жах" (-0,76). Із числових характеристик n-грам видно, що позитивних слів у відгуках більше, ніж негативних, відповідно, можна дійти висновку, що товар з id - 122360970, є якісним і його варто обрати для перепродажу.

Таблиця 3.2 - Упорядковані відгуки по товару з id – 122360970

ID	poswords	pos_importance	negwords	neg_importance
122360970	найкращий	22.57	найстрашніший	-0.77
122360970	добре	1.82	жах	-0.76
122360970	супер	0.85	Не рекомендую	-0.76
122360970	прохолодний	0.71	погано	-0.69
122360970	найкращий	0.71	дурниця	-0.62
122360970	супер-пупер	0.71	розбитий	-0.62
122360970	прохолодний	0.58	погано	-0.58
122360970	найкращий	0.56	перестав працювати	-0.56
122360970	відмінний	0.55	негативний	-0.56
122360970	надійний	0.53	розчарований	-0.54

Таким чином, розроблений метод відрізняється від аналогів тим, що він дасть змогу розібрати відповідні дані з цільового сайту, а класифікація перевіряється вісьмома класичними методами класифікації, а саме Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression. Це дає можливість ухвалювати управлінські рішення щодо вигідного інвестування в нові товари. Для підвищення продуктивності і точності процесу класифікації документів потрібні додаткові роботи. Необхідні нові методи і рішення для отримання корисних знань зі зростаючого обсягу відгуків. Нижче перераховані деякі можливості класифікації неструктурованих даних і виявлення знань:

1. Поліпшити і дослідити методи вибору ознак для поліпшення процесу класифікації.
2. Скоротити час навчання і тестування класифікатора і поліпшити точність класифікації, точність і повторний виклик.
3. Використання семантики та онтології для класифікації відгуків та інформаційного пошуку.
4. Потоковий текст вимагає нових прийомів і методів для управління інформацією.

5. Автоматична класифікація та аналіз настроїв, думок і вилучення з них знань. Аналіз настроїв і думок є новою активною областю текстового аналізу.
6. Класифікація і кластеризація напівструктурованих відгуків має деякі проблеми і нові можливості.
7. Необхідна реалізація процедури класифікації тексту на основі думок для відновлення думок зі слів, використовуваних у певному контексті.
8. Інформаційне вилучення корисних знань з електронних документів і веб-сторінок, таких як продукти і результати пошуку, для отримання смислових повних патернів.

Ідентифікація або зіставлення семантично схожих даних з Інтернету (які містять величезну кількість даних і кожен веб-сайт представляє подібну інформацію по-різному) є важливою проблемою, що має безліч практичних застосувань. Тому веб-інформація, інтеграція та зіставлення схем потребують глибшого вивчення.

Висновки до розділу 3

1. Було створено веб-інтерфейс програми парсингу відгуків. В подальшому пункті буде використання detect(text) і Microsoft Azure для визначення та перекладу тексту.
2. Представлена схема глибокої нейромережевої ідентифікації мов, яка досягає майже ідеальної точності при класифікації несхожих мов і близько 90% точності на дуже схожих мовах. І розширення корпусу цих мов за рахунок зовнішніх джерел не дуже допомогло, головним чином тому, що в розширену частину корпусу не потрапили n-грами слів, які є унікальними для певних мов. На даний момент подальше вдосконалення може бути досягнуто лише шляхом розробки функцій, заснованих на правилах, шляхом спілкування з експертами з мови або носіями мови.

3. Огляд процесів машинного навчання та методів представлення документів. Представлено аналіз методів відбору ознак і алгоритмів класифікації. Для автоматичної класифікації документів було запропоновано кілька алгоритмів або комбінацій алгоритмів у вигляді гібридних підходів. Серед цих алгоритмів найбільш придатними в існуючій літературі є класифікатори SVM, NB і kNN.

4. Розроблено інтелектуальний метод визначення конкурентного товару на основі онлайн-відгуків, завдяки якому продавець може приймати управлінські рішення стосовно вигідного вкладання коштів в новий товар, і, тим самим, зменшить ризики від неприбуткових операцій купівлі-продажу.

ВИСНОВКИ

У методі розглядаються дві основні ідеї - фільтрація відгуків відповідно до їхньої передбачуваної корисності та вилучення з них бізнес-аналітики. Фільтрація здійснюється у два етапи. Спочатку вибираються не переглянуті відгуки на основі коефіцієнта корисності, отриманого з голосів покупців. Потім відгуки обробляються для отримання сумарного балу, заснованого на читабельності, орфографічних помилках, рейтингу відгуку, суб'єктивності, можливо, і розкритті інформації про автора відгуку. Відгуки з найвищим балом фільтруються. Такий підхід гарантує, що обрані відгуки мають корисний зміст. Другий модуль займається пошуком думок у відгуках. Новизна полягає в тому, що замість визначення загального настрою кожного відгуку ми знаходимо оцінку настрою, пов'язану з кожною характеристикою продукту. Метод працює для будь-якого продукту, динамічно створюючи список характеристик на основі заданих вхідних даних. Система є комплексною, оскільки вона охоплює отримання відгуків покупців, їхній аналіз на кількох рівнях і подальше вилучення бізнес-інформації. Особливі випадки, як-от неявне заперечення, також було опрацьовано.

Майбутня робота - слова, що підпадають під категорію сенсу слова, не включаються до підрахунку орфографічних помилок. Написання слова правильне, але слово використовується поза контекстом. Цей випадок може бути розглянутий. Якщо в огляді присутні настрої, пов'язані з іншим продуктом, вони також відображаються на характеристиках обговорюваного продукту. Ця проблема може бути вирішена.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lunn Tony (1972), “Segmenting and Constructing Markets,” in Consumer Market Research Handbook, Worcester R. M., ed. Maidenhead, Berkshire: McGraw-Hill.
2. Plotkina, D. and A. Munzel, “Delight the Experts, but never Dissatisfy Your Customers! A Multi-Category Study on the Effects of online review source on Intention to Buy a New Product,” Journal of Retailing and Consumer Services, Vol. 29, No. Supplement C: 1–11, 2016.
3. Cocks Douglas L., and Virts John R. (1975), “Market Definition and Concentration in the Ethical Pharmaceutical Industry,” Internal publication of Eli Lilly and Co., Indianapolis.
4. Lenz Ralph C. Jr., and Lanford H. W. (1972), “The Substitution Phenomena,” Business Horizons, 15(February), 63–68.
5. Arthur S. House and Edward P. Neuburg. Toward Automatic Identification of the Lan-guage of an Utterance. I. Preliminary Methodological Considerations. The Journal of theAcoustical Society of America, 62(3):708–713, 1977
6. Schaffner, 2004, C. Schaffner. Metaphor and translation: some implication of a cognitive approach. Journal of Pragmatics, 36 (2004), pp. 1253-1269
7. Vinay, Jean-Paul, and Jean Darbelnet. 1958/1972. Stylistique comparée du français et del’anglais: méthode de traduction Nouvelle édition revue et corrigée, Paris: Didier.
8. John C. Catford. A Linguistic Theory of Translation: An Essay on Applied Linguistics. Oxford University Press, London (1965)
9. Venuti, 1998b. Lawrence Venuti. The Scandals of Translation: Towards on Ethics of Difference. Routledge, London and New York (1998)
10. Newmark, 1988. P. Newmark. A Textbook of Translation Prentice Hall, New York/London/Toronto/Sydney (1988)
11. Belk Russell (1975), “Situational Variables and Consumer Behavior,” Journal of Consumer Research, 2(December), 157–164.

12. Bettman James R. (1971), “The Structure of Consumer Choice Processes,” Journal of Marketing Research, 8(November), 465–471.
13. Bourgeois Jacques D., Haines George H., and Sommers Montrose S. (1979), “Defining an Industry,” paper presented to the TIMS/ORSA Special Interest Conference on Market Measurement and Analysis, Stanford, CA, March 26.
14. Butler Ben Jr., and Butler David H. (1970 and 1971), “Hendrodynamics: Fundamental Laws of Consumer Dynamics,” Hendry Corp., Croton-on-Hudson, NY, Chapter 1 (1970) and Chapter 2 (1971).
15. Cocks Douglas L., and Virts John R. (1975), “Market Definition and Concentration in the Ethical Pharmaceutical Industry,” Internal publication of Eli Lilly and Co., Indianapolis.
16. Day George S. (1977), “Diagnosing The Product Portfolio,” Journal of Marketing, 41(April), 29–38.
17. Day George S., Deutscher Terry, and Ryans Adrian (1976), “Data Quality, Level of Aggregation and Nonmetric Multidimensional Scaling Solutions,” Journal of Marketing Research, 13(February), 92–97.
18. Armstrong, J. S. and T. S. Overton, “Estimating Nonresponse Bias in Mail Surveys,” Journal of Marketing Research, Vol. 14, No. 3: 396–402, 1977.
19. Awad, N. F. and A. Ragowsky, “Establishing Trust in Electronic Commerce Through Online Word of Mouth: An Examination Across Genders,” Journal of Management Information Systems, Vol. 24, No. 4: 101–121, 2008.
20. Baek, H., J. Ahn and Y. Choi, “Helpfulness of Online Consumer Reviews: Readers' Objectives and Review Cues,” International Journal of Electronic Commerce, Vol. 17, No. 2: 99–126, 2012.
21. Baek, H., S. Lee, S. Oh and J. Ahn, “Normative Social Influence and Online Review Helpfulness: Polynomial Modeling and Response Surface Analysis,” Journal of Electronic Commerce Research, Vol. 16, No. 4: 290–306, 2015.

22. Doh, S.-J. and J.-S. Hwang, “How Consumers Evaluate eWOM (Electronic Word-of-Mouth) Messages,” Cyberpsychology & Behavior, Vol. 12, No. 2: 193–197, 2009.
23. Fan, Y.-W. and Y.-F. Miao, “Effect of Electronic Word-of-Mouth on Consumer Purchase Intention: The Perspective of Gender Differences,” International Journal of Electronic Business Management, Vol. 10, No. 3: 175–181, 2012.
24. Green Paul E. (1975), “Marketing Applications of MDS: Assessment and Outlook,” Journal of Marketing, 39(January), 24–31.
25. Huber Joel, and James Bill (1977), “The Monetary Worth of Physical Attributes: A Dollarmetric Approach,” in Moving A Head with Attitude Research, Wind Yoram, and Greenberg Marshall, eds., Chicago: American Marketing Association.
26. Kalwani Manohar U., and Morrison Donald G. (1977), “A Parsimonious Description of the Hendry System,” Management Science, 23(January), 476–477.
27. Lenz Ralph C. Jr., and Lanford H. W. (1972), “The Substitution Phenomena,” Business Horizons, 15(February), 63–68.
28. Kheireddine Abainia, Siham Ouamour, and Halim Sayoud. Robust Language Identificationof Noisy Texts - Proposal of Hybrid Approaches. In 25th International Workshop onDatabase and Expert Systems Applications (DEXA), pages 228–232, Munich, Germany, 2014.
29. Steven Abney and Steven Bird. The Human Language Project: Building a Universal Corpusof the World’s Languages. In Proceedings of Human Language Technologies: The 11thAnnual Conference of the North American Chapter of the Association for ComputationalLinguistics (NAACL HLT 2010), pages 88–97, Los Angeles, USA, 2010.
30. Judit ´Acs, L´aszl´o Grad-Gyenge, Thiago Bruno, and Rodriguez de Rezende Oliveira. ATwo-level Classifier for Discriminating similar Languages. In Proceedings of the JointWorkshop on Language

Technology for Closely Related Languages, Varieties and Dialects(LT4VarDial), pages 73–77, Hissar, Bulgaria, 2015

31. Wafia Adouane and Simon Dobnik. Identification of Languages in Algerian Arabic Mul-tilingual Documents. In Proceedings of The Third Arabic Natural Language ProcessingWorkshop (WANLP 2017), pages 1–8, Valencia, Spain, 2017.
32. Bashir Ahmed, Sung-Hyuk Cha, and Charles Tappert. Language Identification from TextUsing N-gram Based Cumulative Frequency Addition. In Proceedings of Student/FacultyResearch Day, pages 12.1–12.8, CSIS, Pace University, New York, USA, 2004.
33. Malepati Bala Siva Sai Akhil and J. Abhishek. Language Identification, Transliteration andResolving Common Words Ambiguity in a Pair of Languages: Shared Task on Translit-erated Search. In Working Notes of Shared Task on Transliterated Search at Forum forInformation Retrieval Evaluation (FIRE'14), Bangalore, India, 2014.
34. Nicholas Akosu and Ali Selamat. A Dynamic Model Selection Algorithm for LanguageIdentification of Under-resourced Languages. International Journal of Digital ContentTechnology and its Applications (JDCTA), 8, 2014.
35. Beatrice Alex. An Unsupervised System for Identifying English Inclusions in German Text.In Proceedings of the Student Research Workshop, ACL-05, pages 133–138, Ann Arbor,Michigan, USA, 2005.
36. Timothy Baldwin and Marco Lui. Language Identification: The Long and the Short of theMatter. In Proceedings of Human Language Technologies: The 11th Annual Conference ofthe North American Chapter of the Association for Computational Linguistics (NAACLHLT 2010), pages 229–237, Los Angeles, USA, 2010b.
37. Dario Benedetto, Emanuele Caglioti, and Vittorio Loreto. Language Trees and Zipping.Physical Review Letters, 88(4), 2002.

38. R. Bekkerman, r. El-yaniv, y. Winter, n. Tishby. On feature distributional clustering for text categorization. Acm sigir conference, 2001.
39. P. Bennett, s. Dumais, e. Horvitz. Probabilistic combination oftext classifiers using reliability indicators: models and results.acm sigir conference, 2002.
40. P. Bennett, n. Nguyen. Refined experts: improving classification inlarge taxonomies. Acm sigir conference, 2009.
41. S. Chakrabarti, s. Roy, m. Soundalgekar. Fast and accurate text classification via multiple linear discriminant projections, vldbjournal, 12(2), pp. 172–185, 2003.
42. A. Dayanik, d. Lewis, d. Madigan, v. Menkov, a. Genkin. Constructing informative prior distributions from domain knowledge intext classification. Acm sigir conference, 2006.
43. Dunning J. The Theory of International Production / J. Dunning // International Trade Journal. – 1988. – № 3;
44. A. Y. Ng, M. I. Jordan. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. NIPS. pp. 841-848, 2001.
45. H. Raghavan, j. Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. Acmsigir conference, 2007.
46. J. Rocchio. Relevance feedback information retrieval. The smart retrieval system- experiments in automatic document processing,g. Salton, ed. Prentice hall, englewood cliffs, nj, pp 313–323,1971.
47. M. Ruiz, P. Srinivasan. Hierarchical neural networks for text categorization. ACM SIGIR Conference, 1999.
48. M. Sahami, S. Dumais, D. Heckerman, E. Horvitz. A Bayesian approach to filtering junk e-mail. AAAI workshop on learning for text categorization. Tech. Rep. Ws-98-05, AAAI press. Http://robotics.stanford.edu/users/sahami/papers.html

49. H. Schutze, D. Hull, J. Pedersen. A comparison of classifiers and document representations for the routing problem. ACM SIGIR Conference, 1995.
50. N. Slonim, N. Friedman, N. Tishby. Unsupervised document classification using sequential information maximization. ACM SIGIR Conference, 2002.
51. Грамяк Р. А., Лендюк Т. Т., Комар М.П., Ліп’яніна-Гончаренко Х. В. МЕТОД ВИБОРУ КОНКУРЕНТНОГО ТОВАРУ НА ОСНОВІ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ ВІДГУКІВ. Вісник Хмельницького національного університету. 2021. № 6. С. 86–88.
52. Intelligent Method of a Competitive Product Choosing based on the Emotional Feedbacks Coloring / R. A. Gramyak et al. : 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security. 2021. P. 3–8.

ДОДАТОК А

getNewItems

```
from urllib import response
from bs4 import BeautifulSoup
import requests
import pymssql

urlToScrap = 'https://rozetka.com.ua/ua/notebooks/c80004/'
numberOfItemsToGet = 10

sqlServerName = '42yk.l.time4vps.cloud:1433'
sqlServerUser = 'sa'
sqlServerPass = 'g5qZzZ7GQ64z'
sqlServerDbName = 'RozetkaDB'

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/81.0.4044.138 Safari/537.36'}

rozetkaHtmlPage = requests.get(urlToScrap, headers=headers)
rozetkaHtmlSoup = BeautifulSoup(rozetkaHtmlPage.content, 'html.parser')

itemsHtmlSoup = rozetkaHtmlSoup.find_all(class_='goods-tile')

conn = pymssql.connect(sqlServerName, sqlServerUser, sqlServerPass, sqlServerDbName)
cursor = conn.cursor(as_dict=True)
itemNumber = 0

for itemHtmlSoup in itemsHtmlSoup:
    itemNumber += 1
    if itemNumber > numberOfItemsToGet:
        break

    id = itemHtmlSoup.find(class_='g-id').text
    href = itemHtmlSoup.find(class_='goods-tile__picture').attrs['href']
    title = itemHtmlSoup.find(class_='goods-tile__title').text


    print('executing item with ID: ' + id + ', href: ' + href)
    new_value = cursor.callproc('InsertItem', (title, id, href))
    print('SP executed.')
    for row in cursor:
        print("ID=" + row['ID'] + ", URL=" + row['Url'])

print('Applying changes in DB...')
conn.commit()
print('Changes in DB Applied.')
print('Closing DB connection...')
cursor.close()
conn.close()
print('DB connection is closed.')
print('Bye-bye')
```

getCommentsForNewItems

```
from urllib import response
from bs4 import BeautifulSoup
from langdetect import detect, DetectorFactory
import requests
import pymssql

maxNumberOfItemsToScrap = 10
maxNumberOfCommentsToGetForEveryItem = 10
skipFirstNumberOfComments = 0

sqlServerName = '42yk.l.time4vps.cloud:1433'
sqlServerUser = 'sa'
sqlServerPass = 'g5qZzZ7GQ64z'
sqlServerDbName = 'RozetkaDB'

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/81.0.4044.138 Safari/537.36'}

print('Opening Connection to DB...')
conn = pymssql.connect(sqlServerName, sqlServerUser, sqlServerPass, sqlServerDbName)
cursor = conn.cursor(as_dict=True)
print('Connection to DB Successfully established.')

print('Getting list of items for which we need get comments')
cursor.callproc('sp_GetItemsToScrap')
c1_list = cursor.fetchall()
itemNumber = 0
for row in c1_list:
    print('Getting new Comments for item ' + row['Title'] + '...')
    if skipFirstNumberOfComments > 0:
        skipFirstNumberOfComments -= 1
        continue

    itemNumber += 1
    if itemNumber > maxNumberOfItemsToScrap:
        break

    href = row['Url']
    href += 'comments/'
    rozetkaHtmlPage = requests.get(href, headers=headers)
    rozetkaHtmlSoup = BeautifulSoup(rozetkaHtmlPage.content, 'html.parser')
    commentsHtmlSoup = rozetkaHtmlSoup.find_all(class_='product-comments__list-item')
    commentNumber = 0
    for commentHtmlSoup in commentsHtmlSoup:
        commentNumber += 1
        if commentNumber > maxNumberOfCommentsToGetForEveryItem:
            break

        author = commentHtmlSoup.find(class_='comment__author').text
        text = commentHtmlSoup.find(class_='comment__text').text

        # hotfix :)
        if(len(text) > 500):
            text = text[0 : 500]
```

```
        essentialPositive = ''
        essentialNegative = ''
        currentMark = 5
        essentialsHtmlSoup = commentHtmlSoup.find_all(class_='comment__essentials-item')
        for essentialHtmlSoup in essentialsHtmlSoup:
            essentialType = essentialHtmlSoup.find(class_='comment__essentials-label').text
            if essentialType == 'Переваги:':
                essentialPositive = essentialHtmlSoup.find('dd').text
            elif essentialType == 'Недоліки:':
                essentialNegative = essentialHtmlSoup.find('dd').text
            else:
                continue
        starsHtmlSoup = commentHtmlSoup.find_all(class_='rating-stars__item')
        for starHtmlSoup in starsHtmlSoup:
            pathHtmlSoup = starHtmlSoup.find('path')
            color = pathHtmlSoup.attrs['fill']
            starNumber = pathHtmlSoup.attrs['data-index-number']
            starNumberint = int(starNumber)
            if color == '#d2d2d2' and starNumberint < (currentMark-1):
                currentMark = starNumberint

        language = detect(text)

        print('Comment Successfully scrapped.')
        print('inserting comment to DB...')
        print('----------------------------------------------------------------------------------------')
        print('fkItem: ' + str(row['PK_Item']))
        print('text: ' + text)
        print('author: ' + author)
        print('language: ' + language)
        print('essentialPositive: ' + essentialPositive)
        print('essentialNegative: ' + essentialNegative)
        print('currentMark: ' + str(currentMark))
        print('----------------------------------------------------------------------------------------')
        cursor.callproc('sp_InsertComment', (str(row['PK_Item']), text, author, language, essentialPositive,
essentialNegative, currentMark))
        print('Comment Inserted to DB.')
print('Applying changes in DB...')
conn.commit()
print('Changes in DB Applied.')
print('Closing DB connection...')
cursor.close()
conn.close()
print('DB connection is closed.')
print('Bye-bye')
```

translateNewComments

```
from urllib import response
from bs4 import BeautifulSoup
from langdetect import detect, DetectorFactory
import requests
import pymssql
import uuid, json

maxNumberOfCommentsToTranslate = 10


translateKey = "907fe8defbb0443db4d6d557c1990314"
translateEndpoint = "https://api.cognitive.microsofttranslator.com"
translateLocation = "eastus"
translatePath = '/translate'
translateConstructed_url = translateEndpoint + translatePath

translateHeaders = {
    'Ocp-Apim-Subscription-Key': translateKey,
    # location required if you're using a multi-service or regional (not global) resource.
    'Ocp-Apim-Subscription-Region': translateLocation,
    'Content-type': 'application/json',
    'X-ClientTraceId': str(uuid.uuid4())
}

sqlServerName = '42yk.l.time4vps.cloud:1433'
sqlServerUser = 'sa'
sqlServerPass = 'g5qZzZ7GQ64z'
sqlServerDbName = 'RozetkaDB'

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/81.0.4044.138 Safari/537.36'}

print('Opening Connection to DB...')
conn = pymssql.connect(sqlServerName, sqlServerUser, sqlServerPass, sqlServerDbName)
cursor = conn.cursor(as_dict=True)
print('Connection to DB Successfully established.')

print('Getting list of Comments to translate...')
cursor.callproc('sp_GetCommentsToTranslate')
c1_list = cursor.fetchall()
print('List of Comment fetched.')
commentNumber = 0
for row in c1_list:
    commentNumber += 1
    if commentNumber > maxNumberOfCommentsToTranslate:
        break

    print('Translating comment with ID: ' + str(row['PK_Comment']) + ' ...')
    translateParams = {
        'api-version': '3.0',
        'from': row['Language'],
        'to': ['uk']
    }

    if str(row['Text']) != "" and row['Text'] is not None:
```

```
        print('Translating main text of comment...')
        translateBody = [{
            'text': row['Text']
        }]

        request = requests.post(translateConstructed_url, params=translateParams, headers=translateHeaders,
json=translateBody)
        response = request.json()

        if request.ok:
            translatedText = response[0]['translations'][0]['text']
            print('Main Text of comment translated.')
        else:
            translatedText = row['Text']
            print('Something went wrong while translating main text of comment.')
            print('----------------------------------------------------------------------------------------------------
-------------')
            print(response)
            print(request.content)
            print('----------------------------------------------------------------------------------------------------
-------------')
    else:
        if row['Text'] is not None:
            translatedText = row['Text']
        else:
            translatedText = ''

    if str(row['Essentials_Positive_Text']) != "" and row['Essentials_Positive_Text'] is not None:
        print('Translating positive essentials of comment...')
        translateBody_Essential_Positive = [{
            'text': row['Essentials_Positive_Text']
        }]

        request = requests.post(translateConstructed_url, params=translateParams, headers=translateHeaders,
json=translateBody_Essential_Positive)
        response = request.json()

        if request.ok:
            translatedEssentialPositive = response[0]['translations'][0]['text']
            print('positive essentials of comment translated.')
        else:
            translatedEssentialPositive = row['Essentials_Positive_Text']
            print('Something went wrong while translating positive essentials of comment.')
            print('----------------------------------------------------------------------------------------------------
-------------')
            print(response)
            print(request.content)
            print('----------------------------------------------------------------------------------------------------
-------------')
    else:
        if row['Essentials_Positive_Text'] is not None:
            translatedEssentialPositive = row['Essentials_Positive_Text']
        else:
            translatedEssentialPositive = ''

    if str(row['Essentials_Negative_Text']) != "" and row['Essentials_Negative_Text'] is not None:
```

```
        print('Translating negative essentials of comment...')
        translateBody_Essential_Negative = [{
            'text': row['Essentials_Negative_Text']
        }]

        request = requests.post(translateConstructed_url, params=translateParams, headers=translateHeaders,
json=translateBody_Essential_Negative)
        response = request.json()

        if request.ok:
            translatedEssentialNegative = response[0]['translations'][0]['text']
            print('negative essentials of comment translated.')
        else:
            translatedEssentialNegative = row['Essentials_Negative_Text']
            print('Something went wrong while translating negative essentials of comment.')
            print('----------------------------------------------------------------------------------------------------
--------------')
            print(response)
            print(request.content)
            print('----------------------------------------------------------------------------------------------------
--------------')
    else:
        if row['Essentials_Negative_Text'] is not None:
            translatedEssentialNegative = row['Essentials_Negative_Text']
        else:
            translatedEssentialNegative = ''

    print('Comment Successfully translated.')
    print('inserting translated comment to DB...')
    print('--------------------------------------------------------------------------------')
    print('FK_Comment_Original: ' + str(row['PK_Comment']))
    print('fkItem: ' + str(row['FK_Item']))
    print('text: ' + translatedText)
    print('author: ' + row['Author'])
    print('language: ' + 'uk')
    print('essentialPositive: ' + translatedEssentialPositive)
    print('essentialNegative: ' + translatedEssentialNegative)
    print('currentMark: ' + str(row['Mark']))
    print('--------------------------------------------------------------------------------')
    cursor.callproc('sp_InsertTranslatedComment',       (str(row['PK_Comment']),       str(row['FK_Item']),
translatedText, row['Author'], 'uk', translatedEssentialPositive, translatedEssentialNegative, row['Mark']))
    print('Comment Inserted to DB.')

print('Applying changes in DB...')
conn.commit()
print('Changes in DB Applied.')
print('Closing DB connection...')
cursor.close()
conn.close()
print('DB connection is closed.')
print('Bye-bye')
```

КОД PYTHON НОРМАЛІЗАЦІЇ ТЕКСТУ

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
import numpy as np
import pandas as pd
import os
import re


reviews_train = []
for line in open('./train.txt', 'r', encoding = "utf-8"):

    reviews_train.append(line.strip())

reviews_test = []
for line in open('./test.txt', 'r', encoding = "utf-8"):

    reviews_test.append(line.strip())


reviews_train[5]
'12254999 :Еще хорошая новость ,пришло по воздуху ПО :1.1.3 Build 20180124 rel.52299 (EU) [версия v.1.0] дополнительная функция    режим моста (WDS)'


import re

REPLACE_NO_SPACE = re.compile("(\.)|(\;)|(\:)|(\!)|(\')|(\?)|(\,)|(\")|(\()|(\))|(\[)|(\])|(\d+)")
REPLACE_WITH_SPACE = re.compile("(<br\s*/><br\s*/>)|(\-)|(\/)")
NO_SPACE = ""
SPACE = " "

def preprocess_reviews(reviews):

    reviews = [REPLACE_NO_SPACE.sub(NO_SPACE, line.lower()) for line in reviews]
    reviews = [REPLACE_WITH_SPACE.sub(SPACE, line) for line in reviews]

    return reviews

reviews_train_clean = preprocess_reviews(reviews_train)
reviews_test_clean = preprocess_reviews(reviews_test)


reviews_train_clean[5]
' еще хорошая новость пришло по воздуху по  build  rel eu версия v дополнительная функция    режим моста wds'


import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

russian_stop_words = stopwords.words('russian')
def remove_stop_words(corpus):
    removed_stop_words = []
```

```
    for review in corpus:
        removed_stop_words.append(
            ' '.join([word for word in review.split()
                      if word not in russian_stop_words])
        )
    return removed_stop_words

no_stop_words = remove_stop_words(reviews_train_clean)


no_stop_words[5]
'хорошая новость пришло воздуху build rel eu версия v дополнительная функция режим моста wds'


from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(binary=True)
cv.fit(reviews_train_clean)
X = cv.transform(reviews_train_clean)
X_test = cv.transform(reviews_test_clean)


def get_lemmatized_text(corpus):
    from nltk.stem import WordNetLemmatizer
    lemmatizer = WordNetLemmatizer()
    return [' '.join([lemmatizer.lemmatize(word) for word in review.split()]) for review in corpus]

lemmatized_reviews = get_lemmatized_text(reviews_train_clean)


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

target = [1 if i < 481 else 0 for i in range(962)]

ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)

X_train, X_val, y_train, y_val = train_test_split(
    X, target, train_size = 0.75
)

for c in [0.01, 0.05, 0.25, 0.5, 1]:

    svm = LinearSVC(C=c)
    svm.fit(X_train, y_train)
    print ("Accuracy for C=%s: %s"
           % (c, accuracy_score(y_val, svm.predict(X_val))))

final_svm_ngram = LinearSVC(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
```

```
      % accuracy_score(target, final_svm_ngram.predict(X_test)))


from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(binary=True)
cv.fit(reviews_train_clean)
X = cv.transform(reviews_train_clean)
X_test = cv.transform(reviews_test_clean)


feature_to_coef = {
    word: coef for word, coef in zip(
     ngram_vectorizer.get_feature_names(), final_model.coef_[0])
}

print('Positive Words')
for best_positive in sorted(
    feature_to_coef.items(),
    key=lambda x: x[1],
    reverse=True)[:10]:
    print(best_positive)

print('Negative Words')
for best_negative in sorted(
    feature_to_coef.items(),
    key=lambda x: x[1])[:10]:
    print(best_negative)


Positive Words
('роутер', 1.5698892132702176)
('супер', 0.8507945457301497)
('хороший', 0.8158958230405385)
('сигнал', 0.7149497955775814)
('скорость', 0.7140080167119551)
('игра', 0.7103184403087008)
('телефон', 0.5831151559369437)
('работает', 0.5639534804326038)
('будет', 0.5513231171161025)
('приложение', 0.5283152738574807)
Negative Words
('не', -1.453385061312366)
('не рекомендую', -0.9575272497306453)
('через', -0.9353636709023622)
('пылесос', -0.6886346598457043)
('месяц', -0.620311899186346)
('перестала', -0.6202484093768675)
('сломался', -0.581722554736022)
('день', -0.5614443842133172)
('года', -0.55771892712474)
('кнопка', -0.5354937201421938)


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

target = [1 if i < 481 else 0 for i in range(962)]

ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)

X_train, X_val, y_train, y_val = train_test_split(
    X, target, train_size = 0.75
)

for c in [0.01, 0.05, 0.25, 0.5, 1]:

    lr = LogisticRegression(C=c)
    lr.fit(X_train, y_train)
    print ("Accuracy for C=%s: %s"
           % (c, accuracy_score(y_val, lr.predict(X_val))))

final_model = LogisticRegression(C=0.5)
final_model.fit(X, target)
print ("Final Accuracy: %s"
       % accuracy_score(target, final_model.predict(X_test)))


from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(binary=True)
cv.fit(reviews_train_clean)
X = cv.transform(reviews_train_clean)
X_test = cv.transform(reviews_test_clean)
feature_to_coef = {
    word: coef for word, coef in zip(
        cv.get_feature_names(), final_model.coef_[0]
    )
}
for best_positive in sorted(
    feature_to_coef.items(),
    key=lambda x: x[1],
    reverse=True)[:5]:
    print (best_positive)

for best_negative in sorted(
    feature_to_coef.items(),
    key=lambda x: x[1])[:5]:
    print (best_negative)


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

target = [1 if i < 481 else 0 for i in range(962)]
```

```
ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)

X_train, X_val, y_train, y_val = train_test_split(
   X, target, train_size = 0.75
)

for d in [0.01, 0.05, 0.25, 0.5, 1]:

   Ada = AdaBoostClassifier(D=d)
   Ada.fit(X_train, y_train)
   print ("Accuracy for C=%s: %s"
         % (c, accuracy_score(y_val, Ada.predict(X_val))))

final_ngram = AdaBoostClassifier(C=0.5)
final_ngram.fit(X, target)
print ("Final Accuracy: %s"
      % accuracy_score(target, final_ngram.predict(X_test)))


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import SGDClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

target = [1 if i < 481 else 0 for i in range(962)]

ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)

X_train, X_val, y_train, y_val = train_test_split(
   X, target, train_size = 0.75
)

for d in [0.01, 0.05, 0.25, 0.5, 1]:

   SGD = SGDClassifier(D=d)
   SGD.fit(X_train, y_train)
   print ("Accuracy for C=%s: %s"
         % (c, accuracy_score(y_val, SGD.predict(X_val))))

final_ngram = SGDClassifier(C=0.5)
final_ngram.fit(X, target)
print ("Final Accuracy: %s"
      % accuracy_score(target, final_ngram.predict(X_test)))


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.datasets import make_classification
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

target = [1 if i < 481 else 0 for i in range(962)]

ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)

X_train, X_val, y_train, y_val = train_test_split(
   X, target, train_size = 0.75
)

for c in [0.01, 0.05, 0.25, 0.5, 1]:

   RF = RandomForestClassifier(C=c)
   RF.fit(X_train, y_train)
   print ("Accuracy for C=%s: %s"
        % (c, accuracy_score(y_val, RF.predict(X_val))))

final_svm_ngram = RandomForestClassifier(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
      % accuracy_score(target, final_RF_ngram.predict(X_test)))


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.datasets import load_iris
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

target = [1 if i < 481 else 0 for i in range(962)]

ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)

X_train, X_val, y_train, y_val = train_test_split(
   X, target, train_size = 0.75
)

for c in [0.01, 0.05, 0.25, 0.5, 1]:

   DT = DecisionTreeClassifier(C=c)
   DT.fit(X_train, y_train)
   print ("Accuracy for C=%s: %s"
        % (c, accuracy_score(y_val, DT.predict(X_val))))

final_svm_ngram = DecisionTreeClassifier(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
```

```
      % accuracy_score(target, final_DT_ngram.predict(X_test)))


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

target = [1 if i < 481 else 0 for i in range(962)]

ngram_vectorizer = CountVectorizer(binary=True, ngram_range=(1, 2))
ngram_vectorizer.fit(reviews_train_clean)
X = ngram_vectorizer.transform(reviews_train_clean)
X_test = ngram_vectorizer.transform(reviews_test_clean)

X_train, X_val, y_train, y_val = train_test_split(
   X, target, train_size = 0.75
)

for c in [0.01, 0.05, 0.25, 0.5, 1]:

   KN = KNeighborsClassifier(C=c)
   KN.fit(X_train, y_train)
   print ("Accuracy for C=%s: %s"
         % (c, accuracy_score(y_val, KN.predict(X_val))))

final_svm_ngram = KNeighborsClassifier(C=0.05)
final_svm_ngram.fit(X, target)
print ("Final Accuracy: %s"
      % accuracy_score(target, final_KN_ngram.predict(X_test)))
```

ДОДАТОК Б

Intelligent Method of a Competitive Product Choosing based on the Emotional Feedbacks Coloring

Roman Gramyak[a], Hrystyna Lipyanina-Goncharenko[a], Anatoliy Sachenko[a,b], Taras Lendyuk[a] and Diana Zahorodnia[a]

[a] *West Ukrainian National University, Lvivska str., 11, Ternopil, 46000, Ukraine*
[b] *Kazimierz Pulaski University of Technology and Humanities in Radom, Department of Informatics, Jacek Malczewski str., 29, Radom, 26 600, Poland*

Abstract

Finding the best products for sale is one of the most important steps in the process of a profitable company creating. That is why the choice of goods for the online store must be made carefully, taking into account both the opportunities and analysis of prospects in the niche, and a number of other important parameters. One of the methods of competitive product choosing can be products analysis in marketplaces based on the emotional feedbacks coloring. Research on product feedbacks is an extremely popular topic, as confirmed by research analysis. Feedbacks can be constantly re-read, but when there are many products in one segment, because there are more and more manufacturers, it is time consuming. Therefore, the development of technology that can automate this process is necessary for the sales business. Paper develops the intelligent method of a competitive product choosing based on the emotional feedbacks coloring, which is divided into three blocks: parser of feedbacks, emotional coloring determination and feedbacks classification. The data will help retailers manage their websites wisely and help customers make purchasing decisions. The implementation of the method was carried out on the data of the Ukrainian site Rozetka, where 4477 feedbacks were used. The classification was tested by eight classical machine-based classification methods, namely Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression.

Keywords

Product, feedback, parser, text emotional coloring, classification, machine learning.

1. Introduction

Other people's opinions have always been an important piece of information for most of us in the decision-making process. The interest shown by users in online feedback and comments, as well as the potential impact of these comments on issues in discourse and decision-making, make us pay attention to this aspect of online activity. The purpose of tone analysis is to find ideas in the text and determine their properties. This method finds its practical application in such fields as sociology, political science, marketing, medicine and many others.

The main approaches that can be used to analyze emotional mood are machine learning and a vocabulary-based approach. Currently, the analysis of attitudes is an interesting topic and direction of development, as it has many practical applications. Companies use it to automatically analyze survey feedbacks, product feedbacks, and social media comments to gain valuable information about their brands, products, and services. However, today seller can use the analysis of emotional mood and to

IntelITSIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine
EMAIL: fear3171@gmail.com (R. Gramyak); xrustya.com@gmail.com (H. Lipyanina-Goncharenko); as@wunu.edu.ua (A. Sachenko); tl@wunu.edu.ua (T. Lendyuk); dza@wunu.edu.ua (D. Zahorodnia)
ORCID: 0000-0001-8698-0377 (R. Gramyak); 0000-0002-2441-6292 (H. Lipyanina-Goncharenko); 0000-0002-0907-3682 (A. Sachenko); 0000-0001-9484-8333 (T. Lendyuk); 0000-0002-9764-3672 (D. Zahorodnia)

select the potentially most profitable product that is in good demand in the market, it will allow online sellers to reduce the risks of choosing a product for trade, and thus increase revenue.

In this regard, it can be considered that the development of a method of selection by machine learning algorithms of the intelligent method of a competitive product choosing based on the emotional feedbacks coloring is one of the most promising areas in online commerce.

This paper is structured as follows. A Section 2 discusses the analysis of related work, and the Section 3 presents the proposed intelligent method of a competitive product choosing based on the emotional feedbacks coloring. The Section 4 presents a case study and discussion, and the Section 5 is summarizing the obtained results.

2. Related works

Reviewing customer feedbacks in online stores is attracting more and more attention from practitioners and scientists. Many studies have analyzed the impact of online feedbacks on sales of goods and services: Amazon products [1], technical characteristics of products from social networks [8], Flipkart, Snapdeal and Amazon India brands [34] and Amazon, Sony and Pocketbook on Facebook, reviews of restaurants from Yelp.com [49], reviews of books (book.dangdang.com). Also, there are works that analyze research on predicting the usefulness of the review [40], identifying data sources, ML methods [23].

Important after data retrieval is the ranking of these data, [18] presents the structure of merging information to rank product feedbacks. Also, ranking approaches are proposed: method on different aspects of alternative products [11]; model [19] of regression prediction based on their quality; hierarchical approach [20], word level and review level; classifier [15] Ensemble, which uses linguistic features to praise or classify complaints; method [21], which returns the ranking lists of useful feedbacks according to their usefulness in relation to the product, a model of multiple linear regression using the method of elastic network regularization.

After structured feedback receiving, seller need to conduct a detailed analysis. In [3] there is developed the system for predicting the usefulness of the feedback for multilingual online feedbacks, which displays the best results in terms of forecasting the usefulness and classification. A [13] model for detecting product defects based on feedback on social networks is proposed. In [14] the method of analysis of competitive advantages of the product is offered, which provides an important basis for quality management and development of marketing strategy by UGC mining. An empirical study of the choice of functions for predicting the usefulness of online product feedbacks was conducted in [22]. The relationship between moods and usefulness for feedback was studied [26], a method was developed for full use of mood features in assessing usefulness for feedback and it was investigated whether the type of product affects the assessment of usefulness for feedback. Based on the bias of negativity and the theory of summation, [33] proposed a theoretical model that explains the usefulness of feedbacks on the Internet based on the specific characteristics of these feedbacks (i.e., duration, evaluation, arguments frame). [43] developed a system that takes all feedbacks containing Hindi, as well as texts in English, and determines the mood expressed in this feedback for each attribute of the product, as well as the final feedback of the product. Several utility predictions models have been studied [48] using multidimensional adaptive regression, classification and regression trees, random forest approaches, neural networks, and deep neural networks using two real Amazon product feedback datasets.

Determining the emotional state is the next step in analyzing user feedback. To analyze the emotional mood of feedbacks, seller can use several approaches, namely: models [30, 32] of multiple regression; ensemble method [9] with several revisions; clustering of text [6] focused on the business sector; model [4] consumer decision-making in terms of risk; method [16] to dynamically assign a coefficient of influence to each basic student in the ensemble; SentiCon by LSA consolidation [25]; model [29] prediction based on a combination of neural network (CNN) and TransE; method [38] of classification of action verbs in the feedback text; integrated model [35] of decision support for finding products on the Internet; approach [44] is based on language modeling (LM); model [7] of product recommendations based on product rating, filtering and hybrid classification of decision trees; method [10] of hybrid classification of moods on the basis of attributes for the purpose of receiving orientation on moods, approach [12] on the basis of fuzzy numbers of an interval of type 2; model [31] predicting the usefulness of feedback is presented using a truncated decision tree C4.5.

The most accurate in determining the emotional mood of the feedback are models [5, 36, 37] based on machine learning, deep learning [17, 39] and convolutional neural network [2]. An uncontrolled thematic model of emotion extraction has been proposed [24], which adds appropriate relationships between aspect words and thought words to express comments as a bag of aspect-thought pairs. A model is presented [27] for determining the usefulness of comments using text functions removed from feedbacks, based on different types of functions, including emotional, linguistic and textual features, values of valence, excitation and dominance, feedback duration and polarity of comments. An in-depth model has been proposed [28] for using the features of feedbacks based on content, semantics, attitudes, and metadata to predict the usefulness of the feedback. [41] presents a general structure that uses natural language processing (NLP) methods, including mood analysis, text data analysis, and clustering methods, to obtain new estimates based on consumer sentiment for different product characteristics. It is proposed [42] to use the plot of the film Wikipedia to identify genre factions using methods of text extraction, based on the model of word reasoning, where (frequency) of occurrence of each word is used as a feature to prepare a classifier. [45] various model architectures have been proposed that can be used to predict the genre and rating of a film in the different languages available in our abstract-based data set. Proposed [46] approach to the classification of moods and tweet language, the architecture includes a convolutional neural network (ConvNet) with two different outputs, each of which is designed to minimize the error of classification or assignment of distribution, or language identification. [47] The classification of the film genre is proposed only on the basis of poster images, on the basis of an in-depth neural network.

As it confirmed by the analysis above a research on product feedbacks is an extremely popular topic, the data help retailers manage their websites wisely and help them make purchasing decisions. Analysis of the emotional state of the feedbacks is an important point in solving the problem of determining the most popular product, which allows the seller to choose the most profitable product for sale.

In this regard, a goal of this paper is to develop the intelligent method of a competitive product choosing based on the emotional feedbacks coloring.

3. Proposed method

To indicate further the novelty of the proposed method authors conducted a detailed analysis of analogues where data is extracted, the emotional mood of the feedbacks is determined and classified (Table 1).

Table 1
Analysis of research analogues

Source	Formulation of the problem	Data mining	Emotional coloring	Classification		Data
				Method	Accuracy	
6	A model of product recommendations based on mood assessments; real-time product data is proposed	DOM Parser	Normalized Product Feedback Score	Random Tree Hoeffding Tree Adaboost +RT	0.9691 0.978 0.967	www.amazon.com
11	A new ranking method is offered through online feedbacks based on various aspects of alternative products, which combines both objective and subjective values.	-	LDA	PageRank	-	-
14	A method of analysis of competitive advantages	Parser	UGC	NTUSD Hownet	0.7425 0.6425	Social networks

Source	Formulation of the problem	Data mining	Emotional coloring	Classification		Data
				Method	Accuracy	
	of the product is proposed, which provides an important basis for quality management and development of marketing strategy through the extraction of UGC.			Trained domain-specific sentiment lexicon with H=9	0.8625	
18	The approach of networks to the detection of social networks based on the method of fuzzy clustering.	Parser	LSGDM	SVM	0.63	www.twitter.com
				SVM-DS	0.97	
21	A method with an extended list of functions for presenting the overview and a model of multiple linear regression using the method of elastic network regularization is proposed.	DOM	LS2F	Decision tree with LS2F	0.77	www.amazon.com
26	The data set used methods to obtain information and assess moods. Gradient amplification and F-estimation methods were used to classify the data set with these characteristics.	-	Python library	Random forest	0.5742	www.amazon.com
			SentiWordNet	Gradient boosting	0.5721	
33	ANOVA assays are used to identify the levels of each of these characteristics that maximize the tangible usefulness of online consumer feedbacks, and an artificial neural network approach is used to predict the usefulness of this feedback based on its characteristics.	-	ANOVA	ANN	0.84	-
37	A new approach called value-based neighbor selection (VNS) is	-	-	Classification threshold	0.706	JD.com

Source	Formulation of the problem	Data mining	Emotional coloring	Classification		Data
				Method	Accuracy	
	proposed to address the above constraints.					
39	The OCC model and the CNN-based generalization method for Chinese microblogging systems are proposed.	-	OCC	CNN	0.84	Chinese social networks
41	A general structure that uses natural language processing techniques, including mood analysis, text data analysis, and clustering techniques, to obtain new estimates based on consumer sentiment for different product characteristics.	-	-NLP	RSS	-	www.amazon.com
48	Several utility prediction models are built using multidimensional adaptive regression, classification and regression tree, random forest approaches, neural network, and deep neural network using two real-world Amazon product feedback datasets.	-	-	MAR	0.90	www.amazon.com
				CART	0.97	
				RandF	0.82	
				Neural Net	0.75	
				Deep NN	0.60	

To reduce the time spent on choosing a popular new product, the authors have developed the intelligent method of a competitive product choosing based on the emotional feedbacks coloring. The proposed method is illustrated schematically (Fig.1) and is represented by the following steps.

Step 1. Connect the necessary libraries. They provide general mathematical and numerical operations in the form of pre-compiled, fast functions. They are combined into high-level packages (Block 1).

Step 2. Select the desired site for further work (Block 2).

Step 3. Enter the link to the product categories that user needs (Block 3).

Step 4. Parsing of user feedbacks to goods is performed. Divide them by product id and add to the text file. A separate feedback for each line (Block 4).

Step 5. Creating a database of feedbacks collected together (Block 5).

Step 6. Connection of feedbacks from a DB (Block 5) for the further work (Block 6).

Step 7. The raw text is quite messy for these feedbacks, so seller need to clean the text to make it easier for the algorithm to recognize the mood (Block 7).

Step 8. Vectorization is performed (Block 8). Vectorization is a type of program parallelization in which single-threaded applications that perform one operation at a time are modified to perform several operations of the same type at the same time. To perform the vectorization process seller need to: tokenization (Block 8.1), ignoring single characters (Block 8.3), converting text into numerical vectors and n-grams (Block 8.5) and then form an array of values (Block 8.6).

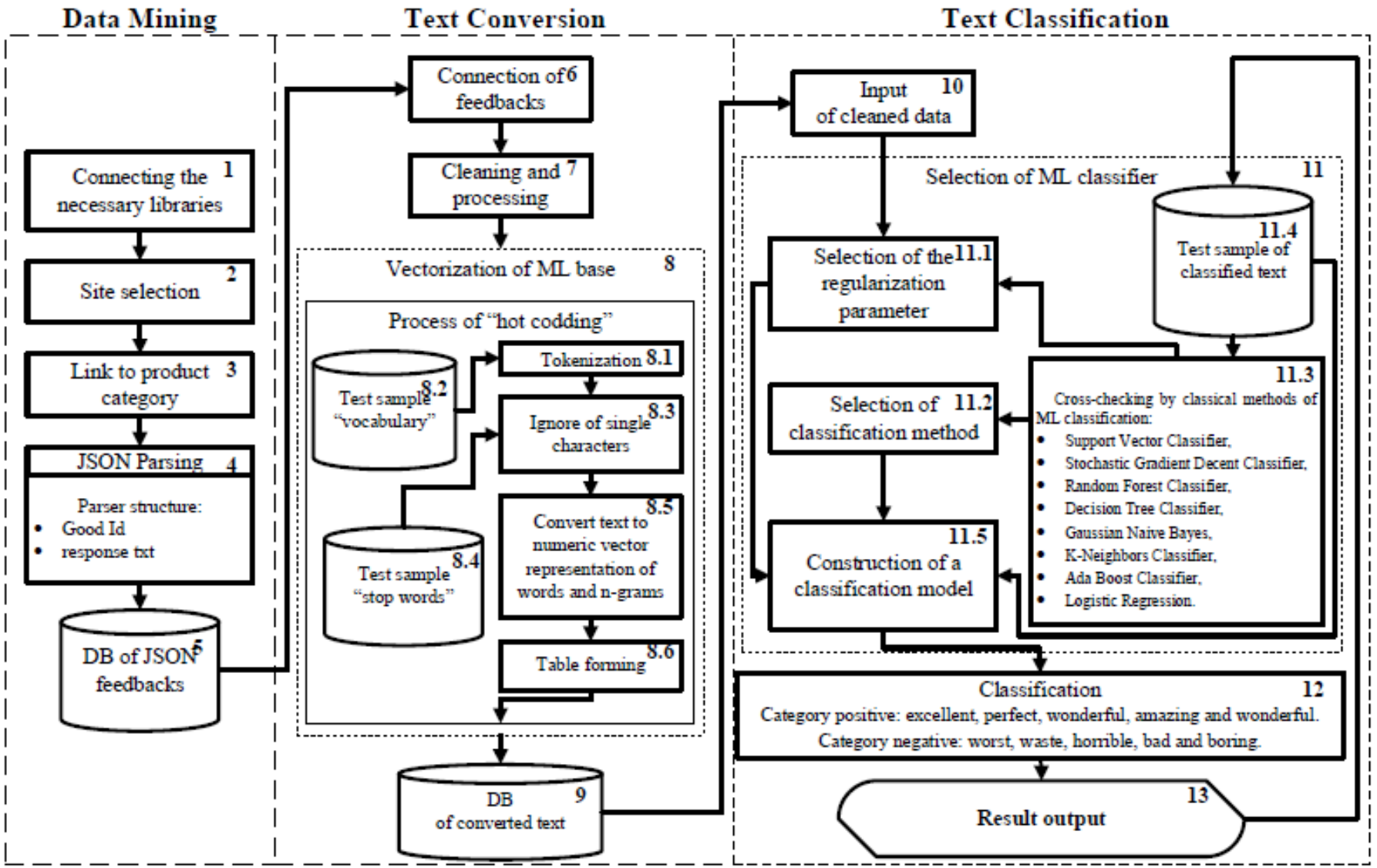


Figure 1: Structure of the intelligent method of a competitive product choosing based on the emotional feedbacks coloring

Step 8.1. Tokenization (Block 8.1) is required to replace a confidential data item with a non-confidential equivalent called a token that has no independent meaning / significance for external or internal use. Sample, which evaluates the quality of the constructed model. The quality assessment made from the test sample can be used to select the best model (Block 8.2).

Step 8.2. The process of ignoring single characters that interfere with the program (Block 8.3) is carried out on the basis of a ready set of "stop words" (Block 8.4). Stop words are very common words such as "if", "but", "we", "he", "she" and "they". Seller can usually delete these words without changing the text semantics (but not always), improving the model performance.

Step 8.3. In this step, the text is converted into numerical vectors and add n-grams (Block 8.5). N-grams are simply a sequence of n elements (sounds, syllables, words or symbols) that go in a row in a text. If seller divide the text into several small fragments represented by N-grams, they are easy to compare with each other and thus obtain a degree of similarity of the analyzed documents. N-grams are often used successfully to categorize text and language. In addition, they can be used to create functions that allow seller to gain knowledge from text data.

Step 8.4. Forming an array of vectorized text values for further storage in the database (Block 8.6).

Step 9. Forming a database of the converted text to select the classifier model (Block 9).

Step 10. Enter the cleared data (Block 10).

Step 11. The next step is to select the most optimal classifier by machine learning algorithms (Block 11).

Step 11.1. Search for the best regularization parameter (Block 11.1). The regularization parameter reduces retraining, which reduces the variance of the estimated regression parameters. Seller need to use it. Cross-checking by eight classical classification methods: Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression (Block 11.3) [50, 52-54]. A ready-made training sample (Block 11.4) was used to check the algorithms.

Step 11.2. Based on the obtained results, the best classification method is chosen (Block 11.2) of the classification with the highest regularization parameters (Block 11.1) and form a classification model (Block 11.5) based on the test sample (Block 11.4).

Step 12. Next is the data classification (Block 12) for positive and negative comments. Namely, positive feedbacks are divided into categories: excellent, perfect, wonderful, amazing and wonderful. Negatives are divided into categories: worst, waste, horrible, bad and boring.

Step 13. The last step is to derive the result on the basis of which seller can decide on a profitable investment in a new product.

The results of experimental studies confirmed the correctness of the developed method, which will be described in more detail in the camera-ready and the presentation.

4. Experimental results and Discussion

Python was chosen to conduct an intelligent method of choosing a competitive product based on the feedback emotional coloring. The following libraries were used: pandas, numpy, train_test_split, SVC, GridSearchCV, SGDClassifier, RandomForestClassifier, DecisionTreeClassifier, GaussianNB KNeighborsClassifier, AdaBoostClassifier, LogisticRegression.

4477 feedbacks from the Rozetka site were used as input [51]. Feedbacks are collected by parsing and formed into a JSON file, which presents the product id and feedback on it.

Next, the data was cleaned for the presence of characters that do not affect the content of the text: ".", ",", "-", "!", "", "?", ":", "", "()", "[]". The next step is removing the "stop words", which are in the Russian language, because most feedbacks are in Russian, further research will consider the possibility of feedback distributing by language, for better results.

After vectorization and lemmatization, the classification of the most optimal classifiers of machine learning algorithms was performed. Therefore, results will be cross-checked (Table 2) and evaluated based on 8 different methods: Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression.

Table 2
The results of cross-evaluation

#	Method	Prediction assessment
1	SupportVectorClassifier	0.73
2	StochasticGradientDecentC	0.74
3	RandomForestClassifier	0.78
4	DecisionTreeClassifier	0.67
5	GaussianNB	0.77
6	KNeighborsClassifier	0.76
7	AdaBoostClassifier	0.76
8	LogisticRegression	0.73

Table 2 shows that all methods showed poor results, due to the fact that the feedbacks include both Ukrainian and Russian languages. However, the best is RandomForestClassifier, with a forecast score of 0.78, which is a valid score for evaluation. Probably, the model accuracy can be increased by increasing the dataset size, as the four thousand dataset is quite modest. Also, it would be possible to reduce the problem to a binary classification of feedbacks to positive and negative, which would also increase accuracy.

Next, the feedbacks will be arranged into positive and negative based on the classification method RandomForestClassifier (Fig. 4). Arranging is based on the inverted n-gram index and relative to the selected product with id – 122360970 (one of the most popular headphones).

Table 3
Arranged feedbacks by product with id – 122360970

ID	poswords	pos_importance	negwords	neg_importance
122360970	top	22.57	the worst	-0.77
122360970	good	1.82	horror	-0.76
122360970	super	0.85	I do not recommend	-0.76
122360970	cool	0.71	badly	-0.69
122360970	best	0.71	nonsense	-0.62
122360970	super-duper	0.71	broke	-0.62
122360970	cool	0.58	bad	-0.58
122360970	best	0.56	stopped working	-0.56
122360970	excellent	0.55	negative	-0.56
122360970	reliable	0.53	disappointed	-0.54

When arranging (see table 3), 10 positive and negative words were separated relative to the n-gram indices. Accordingly, for the product with id – 122360970, the following positive words with the largest n-gram index were identified: "top" (2.57) and "good" (1.82). Also, negative words with the lowest n-gram index – "worst" (-0.77), "horror" (-0.76). From the numerical characteristics of n-grams, show that the positive words in the feedbacks more than negative, respectively, we can conclude that the product with id – 122360970, is of good quality and worth choosing for resale.

Therefore, the developed method differs from analogues (see table 1) in that it will allow to parse the relevant data from the target site, and the classification is tested by eight classical classification methods, namely Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression. That gives the chance to make administrative decisions concerning profitable investment in the new goods.

5. Conclusions

The intelligent method of a competitive product choosing based on the emotional feedbacks coloring, based on which the seller can make management decisions regarding profitable investments in a new product, and thus reduce the risks of non-profit sales. Also, the proposed method reduces the time spent searching for popular and quality products based on user feedback.

The method was implemented on feedbacks (4477 feedbacks) from the Rozetka website. Feedback is collected by parsing and formed into a JSON file and cleaned of unnecessary characters and "stop words". After vectorization, lemmatization and classification of the most optimal classifiers of machine learning algorithms: Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression. All methods showed poor result, due to the fact that the feedbacks include both Ukrainian and Russian feedbacks. However, the best is RandomForestClassifier, with a forecast score of 0.78. Next, the feedbacks were sorted into positive and negative based on the RandomForestClassifier classification. The words with the largest n-gram index are highlighted: positive ("top" (2.57) and "good" (1.82)) and negative ("worst" (-0.77), "horror" (-0.76)). The positive words in the feedbacks are more than the negative ones, based on the n-gram indices, respectively, it can be concluded that they are of good quality and worth choosing for resale.

Areas of further research include an in-depth study of the effectiveness of the developed method in expanding the range of goods and their geography, as well as a significant increase in the number of user feedbacks and the ability to distribute feedback by languages. In addition, it is worth trying to reduce the problem to a binary classification of feedbacks to positive and negative, which would also increase accuracy. Moreover, authors are going to explore ontology models [55] and Deep Learning [56] in domains above.

ДОДАТОК В

Technical sciences *ISSN 2307-5732*

DOI 10.31891/2307-5732-2021-303-6-86-88
УДК 004.85

ЛІП'ЯНІНА-ГОНЧАРЕНКО Х. В.
Західноукраїнський національний університет
ORCID ID: 0000-0002-2441-6292
e-mail: xrustya.com@gmail.com
КОМАР М. П.
Західноукраїнський національний університет
ORCID ID: 0000-0001-6541-0359
e-mail: mko@wunu.edu.ua
ЛЕНДЮК Т. В.
Західноукраїнський національний університет
ORCID ID: 0000-0001-9484-8333
e-mail: tl@wunu.edu.ua
ГРАМЯК Р. М.
Західноукраїнський національний університет
ORCID ID: 0000-0001-8698-0377
e-mail: fear3171@gmail.com

МЕТОД ВИБОРУ КОНКУРЕНТНОГО ТОВАРУ НА ОСНОВІ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ ВІДГУКІВ

У статті розроблено інтелектуальний метод вибору конкурентного товару на основі емоційного забарвлення відгуків, що ділиться на три блоки: парсер відгуків, визначення емоційного забарвлення та класифікація відгуків.

Ключові слова: товар; відгуки; парсер; емоційне забарвлення тексту; класифікація; машинне навчання.

KHRYSTYNA LIPIANINA-HONCHARENKO, MYROSLAV KOMAR,
TARAS LENDYUK, ROMAN GRAMYAK
Western Ukrainian National University

METHOD OF CHOOSING A COMPETITIVE PRODUCT BASED ON THE EMOTIONAL COLOR OF THE CALLS

Other people's opinions have always been an important piece of information for most of us in the decision-making process. The interest shown by users to online reviews and comments, as well as the potential impact of these comments on issues in discourse and decision-making, make them pay attention to this aspect of online activity. Finding the best products for sale is one of the most important stages in the process of creating a profitable company. That is why the choice of goods for an online store should be carried out deliberately, taking into account both the capabilities and analysis of prospects in the niche, as well as a number of other important parameters. One of the methods of choosing a competitive product may be the analysis of goods in marketplaces based on the emotional color of the calls. Product feedback research is an extremely popular topic, which is confirmed by the analysis of studies. Calls can be constantly reread, but when there are many goods in one segment, because there are more manufacturers, it is laborious. Therefore, the development of technology that will be able to automate this process is necessary for business sales. The article developed an intelligent method of choosing a competitive product based on the emotional color of the calls, which is divided into three blocks: a feedback parser, the definition of emotional coloring and the classification of calls. The findings will help retailers manage their websites wisely and help customers make product purchase decisions. In the next scientific researches, the implementation of the method will be carried out on the data of the Ukrainian site Rozetka. The classification of the most classical methods of classification based on machine learning will be carried out, namely Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression.

Keywords: product; reviews; parser; emotional coloring of the text; classification; machine learning.

Постановка проблеми

Думки інших людей завжди були важливою частиною інформації для більшості із нас в процесі прийняття рішень. Інтерес, який демонструють користувачі до онлайн відгуків та коментарів, а також потенціальний вплив цих коментарів на питання у дискурсі і прийняття рішень змушують звернути увагу на цей аспект. У зв'язку з цим можна вважати, що розробка методу вибору конкурентного товару з врахуванням емоційного настрою є одним із найбільш перспективних напрямків у інтернет-торгівлі.

Аналіз останніх джерел

Найбільш точними у визначені емоційного настрою відгуку є моделі [1–3] засновані на машинному навчанні, глибокому навчанні [4, 5] та загортковій нейронній мережі [6, 7]. Представлено [8] модель для визначення корисності коментарів за допомогою текстових функцій, вилучених з оглядів, на основі різних типів функцій, включаючи емоційні, мовні та текстові особливості, значення валентності, збудження та домінування, тривалість огляду та полярність коментарів. Представлено [9] загальну структуру, яка використовує методи обробки природних мов (NLP), включаючи аналіз настроїв, аналіз текстових даних та методи кластеризації, щоб отримати нові оцінки на основі споживчих настроїв для різних характеристик товару. Проаналізовано [10] емоційний зміст великої кількості онлайн-оглядів продуктів із використанням методів обробки натуральної мови (NLP). Запропоновано [11] експериментальне дослідження SentiME, підхід до вилучення полярності настрою повідомлення, на корпусі на основі огляду Amazon.

У зв'язку із цим, метою статті є розробка методу вибору конкурентного товару на основі емоційного забарвлення відгуків.

На відміну від аналогів [8–11] розроблений метод вибору конкурентного товару на основі емоційного забарвлення відгуків дозволить проводити парсер відповідних даних з цільового сайту. Також класифікація буде протестована вісьмома класичними методами класифікації, а саме Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression. На основі отриманого методу продавець може приймати управлінські рішення стосовно вигідного вкладання коштів в новий товар.

Виклад основного матеріалу

Для зменшення часових витрат на вибір популярного нового товару авторами розроблено метод вибору конкурентного товару на основі емоційного забарвлення відгуків. Запропонований метод представлений наступними кроками:

Крок 1. Підключаємо необхідні бібліотеки. Вони надають загальні математичні і числові операції у вигляді пре-скомпільованих, швидких функцій. Вони об'єднуються в високорівневі пакети.

Крок 2. Вибір потрібного сайту для подальшої роботи.

Крок 3. Введення посилання на категорії товару, які нам необхідні.

Крок 4. Виконується парсинг відгуків користувачів на товари. Ділимо їх по id товару і добавляємо в текстовий файл. На кожен рядок окремий відгук.

Крок 5. Створення БД зібраних разом відгуків.

Крок 6. Підключення відгуків з БД для подальшої роботи.

Крок 7. Необроблений текст досить безладний для цих оглядів, тому потрібно зробити очистку тексту, щоб алгоритму було легше розпізнавати настрій.

Крок 8. Проводиться векторизація. Векторизація – вид розпаралелювання програми, при якому однопоточні додатки, які виконують одну операцію в кожен момент часу, модифікуються для виконання декількох однотипних операцій одночасно. Для виконання процесу векторизації потрібно провести токенізацію, ігнорування поодиноких символів, перетворення тексту в числові вектори та n-грами після чого формуємо масив значень.

Крок 8.1. Токенізація потрібна для заміни конфіденційного елемента даних на неконфіденційний еквівалент, званий токеном, який не має самостійного сенсу/значення для зовнішнього або внутрішнього використання.

Крок 8.2. Процес ігнорування поодиноких символів, які мішають роботі програми проводиться на основі готового набору «стоп-слів». Стоп-слова – це дуже поширені слова, такі як «якщо», «але», «ми», «він», «вона» і «вони».

Крок 8.3. На цьому кроці проводимо перетворення тексту в числові вектори та добавляємо n-грами. N-грами – це просто послідовність з n елементів (звуків, складів, слів або символів), що йдуть в якомусь тексті поспіль.

Крок 8.4. Формування масиву векторизованих значень тексту для подальшого зберігання в БД.

Крок 9. Формування БД перетвореного тексту для вибору моделі класифікатора.

Крок 10. Введення очищених даних.

Крок 11. Наступним кроком є вибір найоптимальнішого класифікатора алгоритмами машинного навчання. Для цього потрібно використати, перехресну перевірку вісьмома класичними методами класифікації: Support Vector Classifier, Stochastic Gradient Decent Classifier, Random Forest Classifier, Decision Tree Classifier, Gaussian Naive Bayes, K-Neighbors Classifier, Ada Boost Classifier, Logistic Regression [12].

Крок 12. Далі проводиться класифікація даних. На позитивні та негативні коментарі. А саме позитивні відгуки діляться на категорії: відмінно, ідеально, чудово, дивовижно та чудово. Негативні діляться на категорії: найгірший, відходи, жахливо, погано та нудно.

Крок 13. Останнім кроком є виведення отриманого результату на основі, якого можна приймати рішення стосовно вигідного вкладання коштів в новий товар.

Висновки

Розроблено метод вибору конкурентного товару на основі емоційного забарвлення відгуків, на основі якого продавець може приймати управлінські рішення стосовно вигідного вкладання коштів в новий товар, і, тим самим, зменшить ризики від неприбуткових операцій купівлі-продажу. Також запропонований метод дозволяє зменшити затрати часу на пошук популярних та якісних товарів на основі відгуків користувачів.

Література

1. Singh, J. P., Irani, S., Rana, N. P., Dwivedi, Y. K., Saumya, S., & Roy, P. K. (2017). Predicting the "helpfulness" of online consumer reviews. Journal of Business Research, 70, 346-355.

2. Samal, B., Behera, A. K., & Panda, M. (2017). Performance analysis of supervised machine learning techniques for sentiment analysis. 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS). DOI: 10.1109/ssps.2017.8071579

3. Sun, Xinyu; Han, Maoxin; Feng, Juan (2019). Helpfulness of online reviews: Examining review informativeness and classification thresholds by search products and experience products. Decision Support Systems, 113099. DOI: 10.1016/j.dss.2019.113099

4. Khedkar S., Shinde S. (2020) Deep Learning-Based Approach to Classify Praises or Complaints from Customer Reviews. In: Bhalla S., Kwan P., Bedekar M., Phalnikar R., Sirsikar S. (eds) Proceeding of International Conference on Computational Science and Applications. Algorithms for Intelligent Systems. Springer, Singapore, 391-402. https://doi.org/10.1007/978-981-15-0790-8_38

5. Wu, P., Li, X., Shen, S., & He, D. (2020). Social media opinion summarization using emotion cognition and convolutional neural networks. International Journal of Information Management, 51, 101978

6. Smetanin, S., & Komarov, M. (2019). Sentiment Analysis of Product Reviews in Russian using Convolutional Neural Networks. 2019 IEEE 21st Conference on Business Informatics (CBI), 01, 482-486.

7. Saumya, S., Singh, J.P. & Dwivedi, Y.K. (2020) Predicting the helpfulness score of online reviews using convolutional neural network. Soft Comput 24, 10989–11005. https://doi.org/10.1007/s00500-019-03851-5

8. F. Fouladfar, M. N. Dehkordi and M. E. Basiri, (2020). Predicting the Helpfulness Score of Product Reviews Using an Evidential Score Fusion Method, in IEEE Access, vol. 8, pp. 82662-82687. DOI: 10.1109/ACCESS.2020.2988872.

9. Kauffmann, Peral, Gil, Ferrández, Sellers, Mora, (2019). Managing Marketing Decision-Making with Sentiment Analysis: An Evaluation of the Main Product Features Using Text Data Mining. Sustainability, 11(15), 4235. DOI: 10.3390/su11154235

10. Ullah, R., Amblee, N., Kim, W., & Lee, H. (2016). From valence to emotions: Exploring the distribution of emotions in online product reviews. Decision Support Systems, 81, 41-53.

11. Sygkounas E., Rizzo G., Troncy R. (2016) Sentiment Polarity Detection from Amazon Reviews: An Experimental Study. In: Sack H., Dietze S., Tordai A., Lange C. (eds) Semantic Web Challenges. SemWebEval 2016. Communications in Computer and Information Science, vol 641. Springer, Cham. 108-120. https://doi.org/10.1007/978-3-319-46565-4_8

12. Lipyanina H., Maksymovych V., Sachenko A., Lendyuk T., Fomenko A., Kit I. (2020) Assessing the Investment Risk of Virtual IT Company Based on Machine Learning. In: Babichev S., Peleshko D., Vynokurova O. (eds) Data Stream Mining & Processing. DSMP 2020. Communications in Computer and Information Science, vol 1158, 167-187 Springer, Cham. https://doi.org/10.1007/978-3-030-61656-4_11

13. Ullah, R., Amblee, N., Kim, W., & Lee, H. (2016). From valence to emotions: Exploring the distribution of emotions in online product reviews. Decision Support Systems, 81, 41-53.

14. Sygkounas E., Rizzo G., Troncy R. (2016) Sentiment Polarity Detection from Amazon Reviews: An Experimental Study. In: Sack H., Dietze S., Tordai A., Lange C. (eds) Semantic Web Challenges. SemWebEval 2016. Communications in Computer and Information Science, vol 641. Springer, Cham. 108-120. https://doi.org/10.1007/978-3-319-46565-4_8

15. Lipyanina H., Maksymovych V., Sachenko A., Lendyuk T., Fomenko A., Kit I. (2020) Assessing the Investment Risk of Virtual IT Company Based on Machine Learning. In: Babichev S., Peleshko D., Vynokurova O. (eds) Data Stream Mining & Processing. DSMP 2020. Communications in Computer and Information Science, vol 1158, 167-187 Springer, Cham. https://doi.org/10.1007/978-3-030-61656-4_11

Рецензія/Peer review : 23.12.2021 Надрукована/Printed :30.12.2021