




Я, як студент ХНУРЕ, розумію та підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Підпис  \_\_\_\_\_

Гладкий Д. П.

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Попередній захист проведений «17» червня 2024 р.

Керівник кваліфікаційної роботи  \_\_\_\_\_ доц. Ситнікова П.Е

Харківський національний університет радіоелектроніки

Факультет Комп'ютерні науки

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва)

Тип програми освітньо-наукова  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові Гладкому Дмитру Петровичу  
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження та розробка алгоритму машинного навчання для рекомендаційної системи в сфері онлайн-бронювання»

затверджена наказом університету від 01 квітня 20 24 р. № 259Ст

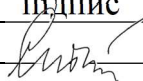
2. Термін подання студентом роботи до екзаменаційної комісії 17 червня 2024 р.

3. Вихідні дані до роботи Дослідити методи та розробити алгоритм машинного навчання рекомендаційної системи для сфери онлайн-бронювання. Дослідження включає аналіз предметної області, актуальність дослідження та існуючі методи. Розробка алгоритму включає формування концептуальної моделі, реалізацію спроектованого алгоритму та аналіз результатів роботи алгоритму; Операційна система – Windows 11, програмне забезпечення: текстовий редактор Microsoft VS Code; мова програмування Python; бібліотеки Pandas, Scikit-learn, Keras; засіб проектування UML-діаграм StarUML.

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ. 4.2 Аналіз предметної області. 4.3 Аналіз актуальності дослідження. 4.4 Дослідження видів та методів машинного навчання. 4.5 Аналіз існуючих рішень. 4.6 Проектування алгоритму машинного навчання. 4.7 Аналіз потенційних проблем. 4.8 Розробка алгоритму машинного навчання. 4.9 Аналіз результатів. 4.10 Висновки. 4.11 Перелік джерел посилання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)  
5.1 Організаційна структура видів машинного навчання. 5.2-5 Алгоритм роботи виду машинного навчання 5.6 Діаграма функціональних вимог. 5.7 Діаграма нефункціональних вимог. 5.8 Діаграма діяльності підготовки даних. 5.9 Діаграма діяльності підбору рекомендацій. 5.10-12 Алгоритм тестування моделі. 5.13 Схема роботи моделі надання рекомендацій.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Аналіз предметної області	доц. Ситнікова П.Е.		

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1	Отримання завдання кваліфікаційної роботи	01.04.2024	Виконано
2	Аналіз предметної області та постановка завдання	08.04.2024	Виконано
3	Дослідження видів та алгоритмів машинного навчання	20.04.2024	Виконано
4	Аналіз існуючих рішень	24.04.2024	Виконано
5	Визначення вимог до розроблюваного алгоритму	01.05.2024	Виконано
6	Формування структури алгоритму	13.05.2024	Виконано
7	Аналіз потенційних проблем	14.05.2024	Виконано
8	Розробка моделі надання рекомендацій	27.06.2024	Виконано
9	Аналіз результатів розробленого алгоритму	30.06.2024	Виконано
10	Оформлення пояснювальної записки	07.06.2024	Виконано
11	Подача кваліфікаційної роботи на допуск до захисту	14.06.2024	Виконано
12	Підготовка доповіді до захисту роботи	16.06.2024	Виконано
13	Подання кваліфікаційної роботи	17.06.2024	Виконано

Дата видачі завдання 01 \_\_\_\_\_ 04 \_\_\_\_\_ 20 24 р.

Студент \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_ доц. Ситнікова П.Е.

(підпис)

(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 75 арк., 17 рис., 26 джерел посилань, 2 додатки.

МАШИННЕ НАВЧАННЯ, РЕКОМЕНДАЦІЙНІ СИСТЕМИ, ОНЛАЙН-БРОНЮВАННЯ, АЛГОРИТМИ МАШИННОГО НАВЧАННЯ, ОПТИМІЗАЦІЯ РЕКОМЕНДАЦІЙ, ПЕРСОНАЛІЗАЦІЯ КОРИСТУВАЦЬКОГО ДОСВІДУ, ВИЯВЛЕННЯ ШАБЛОНІВ, ІНТЕЛЕКТУАЛЬНЕ БРОНЮВАННЯ

Об'єктом дослідження є процеси вибору та бронювання послуг онлайн, які включають взаємодію між користувачами та платформами онлайн-бронювання.

Предметом дослідження є алгоритми машинного навчання, які використовуються для підвищення ефективності рекомендаційних систем у контексті онлайн-бронювання.

Мета дослідження – розробка та валідація алгоритму машинного навчання, який здатен підвищити точність і релевантність рекомендацій в системах онлайн-бронювання, забезпечуючи більш персоналізований досвід користувача.

Результати дослідження – розроблено новий алгоритм машинного навчання для рекомендаційних систем, який покращує точність та релевантність рекомендацій, виявлено ключові фактори, які впливають на ефективність рекомендацій у контексті онлайн-бронювання, продемонстровано переваги нового алгоритму через експериментальну валідацію та порівняння з існуючими методами, розроблено рекомендації для подальшого впровадження та оптимізації алгоритму в реальних системах онлайн-бронювання.

Галузь застосування – сфера електронної комерції, зокрема в онлайн-платформах для бронювання готелів, квитків, туристичних послуг та інших подібних сервісів. Воно може бути корисним для розробників програмного забезпечення, аналітиків даних та менеджерів продуктів, що працюють над покращенням користувацького досвіду та ефективності рекомендаційних систем.

## ABSTRACT

Explanatory note to the master's qualification work: 75 pages, 17 figures, 26 references, 2 appendices.

MACHINE LEARNING, RECOMMENDER SYSTEMS, ONLINE BOOKING, MACHINE LEARNING ALGORITHMS, RECOMMENDATION OPTIMISATION, PERSONALISATION OF USER EXPERIENCE, PATTERN DETECTION, INTELLIGENT BOOKING

The object of research is the processes of choosing and booking online services, which include the interaction between users and online booking platforms.

The subject of research is machine learning algorithms used to improve the efficiency of recommender systems in the context of online booking.

The purpose of the study is to develop and validate a machine learning algorithm that can improve the accuracy and relevance of recommendations in online booking systems, providing a more personalized user experience.

Results of the study - a new machine learning algorithm for recommender systems that improves the accuracy and relevance of recommendations, identifies the key factors that affect the effectiveness of recommendations in the context of online booking, demonstrates the advantages of the new algorithm through experimental validation and comparison with existing methods, develops recommendations for further implementation and optimization of the algorithm in real online booking systems.

The application area is e-commerce, particularly in online platforms for booking hotels, tickets, travel services and other similar services. It can be useful for software developers, data analysts, and product managers working to improve the user experience and performance of recommender systems.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	8
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Актуальність предметної області .....	11
1.2 Актуальність використання машинного навчання .....	15
1.3 Постановка задачі.....	18
2 ДОСЛІДЖЕННЯ ВИДІВ ТА АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ.....	21
2.1 Види машинного навчання.....	21
2.1.1 Кероване навчання .....	21
2.1.2 Некероване навчання .....	23
2.1.3 Напівконтрольоване навчання .....	25
2.1.4 Навчання з підкріпленням .....	26
2.2 Алгоритми машинного навчання.....	28
2.2.1 Лінійна регресія.....	28
2.2.2 Логістична регресія.....	29
2.2.3 Дерево рішень.....	30
2.2.4 Випадкові ліси .....	31
2.2.5 Градієнтний бустинг .....	32
2.2.6 Кластеризація k-середніх.....	33
2.2.7 Нейронні мережі.....	34
2.3 Огляд функціонування існуючих рішень .....	35
2.3.1 Amazon Recommendation System .....	36
2.3.2 Netflix Recommendation Engine .....	37
2.3.3 Google Search Engine .....	39
2.3.4 Результати аналізу функціонування існуючих рішень .....	40
3 ПРОЕКТУВАННЯ АЛГОРИТМУ МАШИННОГО НАВЧАННЯ.....	44
3.1 Загальний огляд.....	44
3.2 Аналіз вимог .....	45

3.3	Формування структури алгоритму .....	49
3.4	Деталізація компонентів алгоритму .....	52
3.5	Планування процесів тестування та валідації .....	54
3.7	Аналіз потенційних проблем .....	57
4	РОЗРОБКА АЛГОРИТМУ МАШИННОГО НАВЧАННЯ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-БРОНЮВАННЯ .....	60
4.1	Вибір моделі та засобів .....	60
4.2	Збір та обробка даних .....	62
4.3	Розробка моделі .....	63
4.4	Навчання моделі .....	65
4.5	Валідація моделі .....	66
4.6	Аналіз результатів .....	67
4.6.1	Оцінка ефективності .....	67
4.6.2	Вплив на бізнес-процеси .....	68
4.6.3	Подальші дослідження .....	69
	ВИСНОВКИ .....	71
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....	73

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

МН – машинне навчання.

ІІІ – штучний інтелект.

KNN – метод K-найближчих сусідів.

ЗНМ - згорткові нейронні мережі.

РНМ - рекурентні нейронні мережі.

e-Commerce – електронна комерція.

ІоТ – інтернет речей.

API - application programming interface - прикладний програмний інтерфейс.

## ВСТУП

У сучасному світі, де цифровізація проникає у всі сфери життя, значення ефективних та інтуїтивно зрозумілих систем онлайн-бронювання стає надзвичайно високим. Ці системи не тільки спрощують процес вибору та бронювання послуг, але й надають можливість для персоналізації користувацького досвіду, заснованого на індивідуальних перевагах та попередніх виборах користувача. Ефективність таких систем безпосередньо залежить від якості рекомендаційних алгоритмів, які вони використовують. Враховуючи це, розвиток і вдосконалення алгоритмів машинного навчання для рекомендаційних систем стає критично важливим завданням.

Проте, незважаючи на значні досягнення в області машинного навчання та штучного інтелекту, існує низка викликів, пов'язаних з точністю, ефективністю та адаптивністю рекомендаційних систем. Ці виклики включають обробку великих обсягів даних, врахування змінних користувацьких уподобань, і вибудовування моделей, здатних адаптуватися до нових трендів без втрати релевантності рекомендацій.

Це дослідження зосереджується на розробці та аналізі нового алгоритму машинного навчання, який має на меті вирішення зазначених викликів у контексті систем онлайн-бронювання. Його задача не тільки підвищити точність рекомендацій, але й забезпечити їхню високу релевантність для користувачів, враховуючи їхні індивідуальні потреби та переваги.

Основну увагу у дослідженні приділено аналізу існуючих алгоритмів, виявленню їхніх слабких місць та розробці нового підходу, який би враховував специфіку сфери онлайн-бронювання. Виконується оцінка ефективності запропонованого алгоритму через ряд експериментів, що включають аналіз великих обсягів даних, тестування в реальних умовах, та порівняння з існуючими рішеннями.

Дослідження спрямоване на вдосконалення технологій рекомендаційних систем, з метою їх подальшого застосування в різних галузях, від електронної

комерції до індустрії розваг. Завдяки цьому, робота має значний потенціал для впливу на подальший розвиток цифрових послуг, забезпечуючи більш ефективне та користувацько-орієнтоване середовище онлайн-бронювання.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Актуальність предметної області

В умовах глобалізації, інформатизації, інтелектуалізації праці, а також загострення кризових явищ все більш очевидною стає необхідність активізації інноваційної діяльності підприємств і організацій усіх галузей економіки, в тому числі сфери послуг. Докорінні структурні зміни у світовій економічній системі пов'язані із загальним зниженням темпів розвитку індустріального сектору при одночасній диверсифікації сервісного комплексу, прискореному зростанні частки сфери послуг у національному доході країн та загальних обсягах споживання, а також глибокою трансформацією структури людських потреб, фінансової системи та зайнятості населення. Сфера послуг стає важливим сектором національного і світового господарства, впливаючи на функціонування і розвиток матеріального виробництва та забезпечуючи якість економічного зростання і підвищення конкурентоспроможності країн. В той же час сфера послуг відіграє суттєву роль у розвитку людського капіталу, створюючи можливості для повнішого задоволення і розвитку потреб людей і суспільства (сприяючи збільшенню вільного часу та підвищенню якості його використання), тому ця сфера є найважливішим елементом формування сучасної якості життя [1].

В свою чергу, сучасний розвиток інформаційних технологій авторитетно впливає на всі сфери життєдіяльності людини. Всесвітня мережа має унікальні якості передачі текстової, візуальної інформації в реальному часі, надає можливість споживачеві отримати широкий спектр різноманітних туристських послуг за короткий період часу. Наприклад, у 60-ті роки ХХ століття, коли цивільна авіація перебувала на етапі активного розвитку, технологія бронювання авіаквитків по телефону виявилася досить трудомісткою в зв'язку зі зростаючим пасажиропотоком. Це призвело до необхідності автоматизації процесу виписки авіаквитків. Перші електронні системи бронювання і резервування туристських послуг отримали назви Apollo і Sabre, їх розробили американські авіакомпанії

United і American Airlines. Подальше впровадження систем бронювання призвело до істотного скорочення часу обслуговування клієнтів і забезпечення резервування в режимі онлайн, що дозволило знизити собівартість пропонованих послуг і підвищити їх якість. Так стало можливим бронювання не тільки авіаквитків, а й інших послуг туристичної індустрії. На сьогоднішній день жодне турагентство не може працювати без використання в своїй роботі глобальної або альтернативної системи бронювання. З активною комп'ютеризацією суспільства люди все частіше бронюють свої подорожі онлайн самостійно [2].

В цьому контексті, важливу роль відіграють рекомендаційні системи. Рекомендаційна система – це система, що використовується для прогнозування індивідуальних вподобань користувачів веб-сайту на основі зібраної про них інформації для створення їм рекомендацій з метою збільшення загального інтересу до об'єктів на веб-сайті (товарів, фільмів, новин, тощо) [3].

Системи рекомендацій зайняли центральне місце в споживчому й професійному досвіді користувачів. Вони вдало інтегрувались «під капот» багатьох продуктів та зробили їх лідерами у своїх індустріях.

Одним з популярних прикладів є Spotify - один з найпопулярніших стримінгових сервісів потокового аудіо, сьогодні має понад 500 мільйонів користувачів. Понад третина з них відкриває для себе нових виконавців завдяки функції «Зроблено для вас». Такий показник — це результат поєднання масштабної бази контенту та майстерного використання систем рекомендацій.



Рисунок 1.1 – Результат підбору рекомендацій Spotify

Ще однією компанією, яку варто розглянути на цьому етапі, є Amazon. На червень 2024 року ринкова капіталізація Amazon становить \$1,911 трлн, що робить її п'ятою найдорожчою компанією у світі за цим показником, згідно з даними Companies Market Cap. Важливість аналізу Amazon полягає не лише в її масштабному впливі на світовий ринок e-Commerce, але й у значному внеску у розвиток рекомендаційних систем. Завдяки передовим алгоритмам та інноваційним підходам, які використовує Amazon, компанія суттєво підвищила ефективність персоналізованих рекомендацій, що робить її лідером у цій сфері та корисним прикладом для вивчення та впровадження подібних рішень у інших компаніях. Приклад надання персоналізованих рекомендацій від Amazon відображено на рис. 1.2.

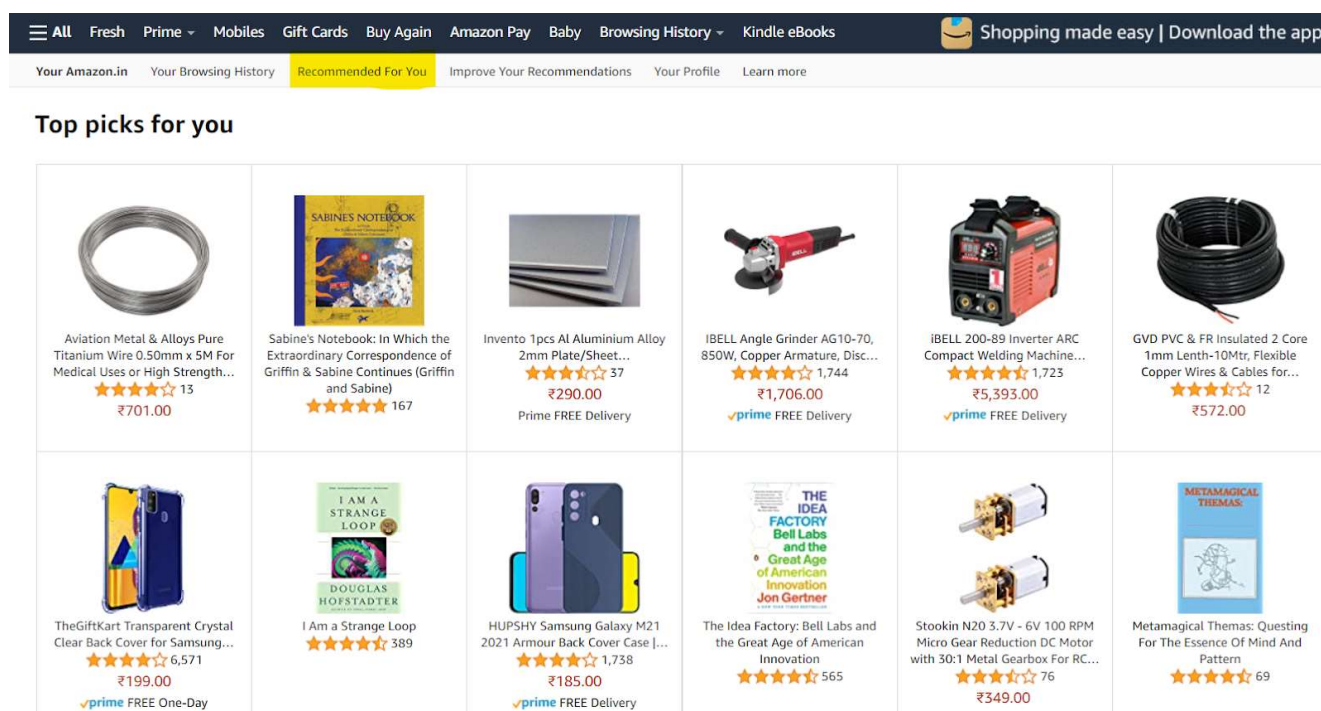


Рисунок 1.2 – Надання рекомендацій Amazon.com

У 2024 році існуючі клієнти очікують, що інтернет-магазин надасть їм персоналізований контент і урізноманітнить їхній досвід покупок.

Згідно з останніми дослідженнями з персоналізації, до 91% клієнтів інтернет-магазину стверджують, що вони, швидше за все, скористаються пропозицією бренду, яка персоналізує їхній досвід. З іншого боку, 98% власників сайтів

електронної комерції скажи це персоналізація покращує їхні стосунки з клієнтами [4]. Надаючи рекомендації, які максимізують потенційну цінність для окремих клієнтів, Amazon може підтримувати зацікавленість споживачів і пропонувати цікаві для них продукти, про які вони можуть навіть не думати.

Виходячи з наведеного аналізу раніше, основа перевага рекомендаційних систем, зокрема в сфері онлайн-бронювання, полягає саме в їхній здатності аналізувати масивний об'єм інформації та створювати персоналізований досвід. Вони враховують різноманітні аспекти — від особистих уподобань до соціальних взаємодій юзерів, а також динамічних змін у контенті.

Однак з розвитком технологій і зростанням кількості даних, з якими працюють рекомендаційні системи, виникає низка викликів:

- обробка великих обсягів даних - забезпечення швидкості та точності обробки даних у реальному часі є важливим аспектом для покращення користувацького досвіду;

- динамічність переваг користувачів - інтереси користувачів не статичні і можуть змінюватися з часом або в залежності від контексту, що вимагає від систем бути гнучкими та адаптивними;

- різноманітність контенту - рекомендаційні системи мають справу з різноманітним контентом, від готельних номерів до авіаквитків, кожен з яких має унікальні атрибути та фактори, що впливають на рекомендацію.

Розвиток алгоритмів машинного навчання відкрив нові можливості для рекомендаційних систем. Застосування методів глибокого навчання, таких як нейронні мережі, дозволило значно покращити якість рекомендацій завдяки здатності ефективно обробляти великі обсяги даних та виявляти складні шаблони в поведінці користувачів. Персоналізація стає все більш важливою в рекомендаційних системах. Використання даних про минулу поведінку користувачів, їх переваги та контекстуальну інформацію дозволяє створювати індивідуалізовані пропозиції, які значно покращують досвід користувача та збільшують вірогідність бронювання.

Новітні технологічні розвитки, такі як обчислення в хмарі, великі дані, та штучний інтелект, надають рекомендаційним системам додаткові інструменти для покращення їх ефективності. Ці технології дозволяють обробляти і аналізувати дані в масштабі та зі швидкістю, яка раніше була недосяжною, відкриваючи шлях для реалізації складних алгоритмів та моделей.

Наведений аналіз предметної області підкреслює важливість рекомендаційних систем у сфері онлайн-бронювання та виклики, з якими вони стикаються. Водночас він виявляє потенціал машинного навчання та штучного інтелекту у вирішенні цих викликів, пропонуючи напрямки для подальших досліджень і розробок в цій області.

## 1.2 Актуальність використання машинного навчання

У нинішню епоху Четвертої промислової революції (4IR або Industry 4.0) цифровий світ має безліч даних, таких як дані Інтернету речей (IoT), дані про кібербезпеку, мобільні дані, бізнес-дані, дані з соціальних мереж, дані про здоров'я тощо. Для інтелектуального аналізу цих даних і розробки відповідних інтелектуальних і автоматизованих додатків, знання штучного інтелекту (ШІ), зокрема, машинного навчання (МН), є ключовим фактором [5].

Машинне навчання – це підгалузь штучного інтелекту, що базується на створенні алгоритмів, здатних вчитися на даних і приймати рішення без заздалегідь запрограмованих інструкцій. Машинне навчання дозволяє комп'ютерам вирішувати складні задачі, аналізуючи великі масиви даних і виявляючи закономірності [6].

Ринок машинного навчання швидко зростає. З 2016 року його обсяг подолав позначку в \$1 млрд, а до 2025 року, судячи з прогнозів, він може збільшитися до \$39,98 млрд.

В кінці 2016 року MIT Technology Review і Google Cloud провели спільне дослідження на тему «Машинне навчання: новий спосіб отримати конкурентну перевагу». Було опитано 375 кваліфікованих респондентів з різних країн світу, які

працюють в дрібних і великих компаніях з різних галузей (промисловість, послуги, фінанси). В результаті дослідження з'ясувалося, що 60% компаній вже використовують машинне навчання (ML), а в третини з них ця технологія перейшла зі стадії інноваційної в стадію зрілості. Більш того, 26% компаній вже отримують за рахунок ML конкурентну перевагу. Чверть компаній інвестують в ML понад 15% від коштів, спрямованих на розвиток ІТ, і в значній мірі повертають зроблені інвестиції [7].

Що ж стосується теперішнього часу, то з 2020 року глобальний ринок рішень ML виріс щонайменше вдвічі, а поява великих мовних моделей та генеративного ШІ привернула до індустрії увагу широкого загалу. Сьогодні скористатись можливостями машинного навчання може буквально кожен користувач. Для цього достатньо увімкнути камеру на смартфоні (адже мобільна зйомка працює на алгоритмах), або відкрити в браузері безплатну версію ChatGPT [8].

Також важливо відзначити, що найважливіші тренди машинного навчання сьогодні розвиваються у світі великого бізнесу. За оцінками Statista, у 2024 році обсяг ринку рішень ML має досягти позначки в \$204,3 млрд. Найближчими роками ринок буде розвиватися із середньорічним темпом зростання в 17%; вже до 2030 року він має збільшитись більш ніж вдвічі – до \$528 млрд [8].

Які індустрії сьогодні інвестують в машинне навчання? Розподіл ринку ML по основних секторах наразі виглядає так:

- 18,8% – рішення для промисловості;
- 15,4% – інструменти для фінансового сектору;
- 12,2% – технології охорони здоров'я;
- 10,6% – рішення для транспорту;
- 10,1% – рішення для сектору безпеки;
- 9,8% – інструменти для бізнесу та юридичних послуг.

Технології ML можна застосовувати практично у будь-якій сфері, що потребує обробки даних. Розглянемо сучасні приклади використання машинного навчання в різних індустріях [8].

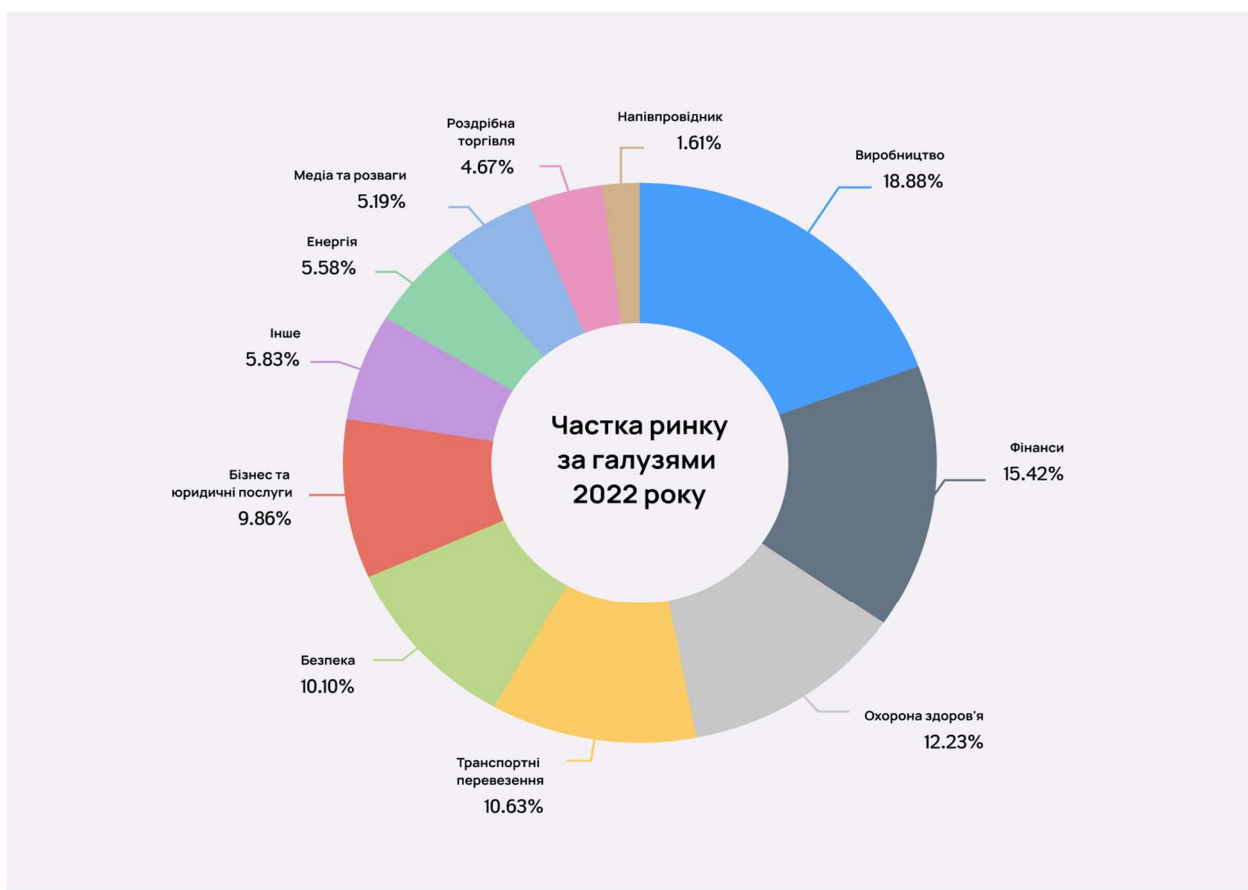


Рисунок 1.1 – Розподіл за використанням МН в різних індустріях

Машинне навчання і, зокрема, нейронні мережі доцільно використовувати для вирішення бізнес-завдань у випадках, коли:

- накопичено велику кількість різних даних, але програми для їх обробки і систематизації відсутні;
- наявні дані спотворені, не повні або не систематизовані;
- дані настільки різні, що важко виявити зв'язку і закономірності, що існують між ними.

Бізнес-завдання, які можуть вирішуватися засобами машинного навчання і нейронних мереж:

- прогнозування: попиту, обсягу продажів, наповнення складу, завантаження устаткування і інших ресурсів, подальшого розвитку підприємства;
- виявлення: тенденцій, прихованих взаємозв'язків, аномалій, повторюваних елементів;

- розпізнавання: фото-, відео-, аудіоконтенту, спроб шахрайства, брехні, внутрішніх загроз, зовнішніх атак на систему безпеки;
- автоматизація: роботи операторів в онлайн-чатах, телефонних операторів;
- класифікація: аналіз складу покупців, клієнтів, замовників і сегментація їх за різними параметрами;
- кластеризація: класифікація за параметрами, які з самого початку не були відомі;
- розробка: чат-боти.

Застосування машинного навчання в рекомендаційних системах, зокрема в сфері онлайн-бронювання, відкриває нові можливості для підвищення задоволеності клієнтів та оптимізації процесів вибору та бронювання. З використанням алгоритмів машинного навчання можливо аналізувати великі обсяги даних про поведінку користувачів, їх переваги та історію пошуку, щоб надавати персоналізовані та високо релевантні пропозиції.

Машинне навчання стає фундаментальним інструментом у сучасному цифровому світі, здатним трансформувати дані в знання та інсайти, які можуть бути використані для покращення продуктів, послуг та користувацького досвіду. Його вплив на ринок та потенціал у сфері онлайн-бронювання є значним, пропонуючи нові можливості для інновацій та конкурентоспроможності.

### 1.3 Постановка задачі

Завданням цієї роботи є дослідження та розробка алгоритму машинного навчання для формування рекомендацій в сфері онлайн-бронювання. Цей алгоритм, використовуючи детальний набір даних про поведінку користувачів та об'єкти на платформі онлайн-бронювання, має аналізувати дані та створювати рекомендації.

Аналітична частина роботи включає порівняльний аналіз різних алгоритмів машинного навчання, вивчення їх переваг та недоліків, а також складностей

імплементації у контексті онлайн-бронювання. Результати розробки та аналізу тестування мають на меті реалізувати найбільш ефективний алгоритм рекомендації.

Виходячи з наведеного раніше, для розробки алгоритму машинного навчання визначено наступні завдання:

1) аналіз існуючих рішень та методів машинного навчання, що застосовуються в рекомендаційних системах для онлайн-бронювання. Це завдання передбачає детальний огляд літератури та аналіз різноманітних підходів до побудови рекомендаційних систем, їхніх переваг та недоліків. Особлива увага приділяється аналізу ефективності існуючих моделей, методів обробки даних та алгоритмів машинного навчання;

2) визначення ключових вимог та специфікацій для розробки алгоритму машинного навчання в контексті онлайн-бронювання. Це завдання включає ідентифікацію основних викликів, з якими стикаються рекомендаційні системи у сфері онлайн-бронювання, та формулювання вимог до нового алгоритму, що повинен враховувати ці виклики для підвищення точності, швидкості та релевантності рекомендацій;

3) розробка концептуальної моделі алгоритму машинного навчання для рекомендаційної системи. Розробка теоретичної моделі, яка буде лягати в основу практичної реалізації алгоритму. Модель повинна враховувати специфіку даних в онлайн-бронюванні та бути адаптованою до різноманітних типів користувачів та їхніх вимог;

4) реалізація алгоритму машинного навчання. Практична реалізація розробленої моделі у вигляді алгоритму, включаючи програмування, налаштування параметрів та інтеграцію з існуючою інфраструктурою рекомендаційної системи. Задача також включає проведення тестування для оцінки ефективності алгоритму за допомогою визначених метрик;

5) аналіз результатів та їх оцінка. Оцінка результатів тестування алгоритму для визначення його ефективності у порівнянні з існуючими рішеннями.

Аналіз результатів дозволить визначити сильні сторони розробленого алгоритму та області, які потребують подальшого вдосконалення;

б) формулювання рекомендацій для подальшої роботи. На основі аналізу результатів формулювання рекомендацій для подальших досліджень та розвитку алгоритму машинного навчання в контексті рекомендаційних систем онлайн-бронювання.

Ці завдання дозволять не тільки розробити новий ефективний алгоритм для рекомендаційних систем, але й зробити внесок у теоретичну базу знань у сфері машинного навчання та його застосування.

## 2 ДОСЛІДЖЕННЯ ВИДІВ ТА АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ

### 2.1 Види машинного навчання

Розвиток машинного навчання (МН) є ключовим елементом у багатьох сферах сучасних технологій, від автоматичного розпізнавання зображень до розробки рекомендаційних систем. В контексті онлайн-бронювання, його ефективність безпосередньо впливає на здатність системи надавати релевантні та точні пропозиції користувачам.

Машинне навчання можна класифікувати за різними критеріями, але в основному їх можна поділити на чотири категорії: навчання під контролем, навчання без контролю, напівконтрольоване навчання та навчання з підкріпленням, як показано на рис. 2.1 [5].

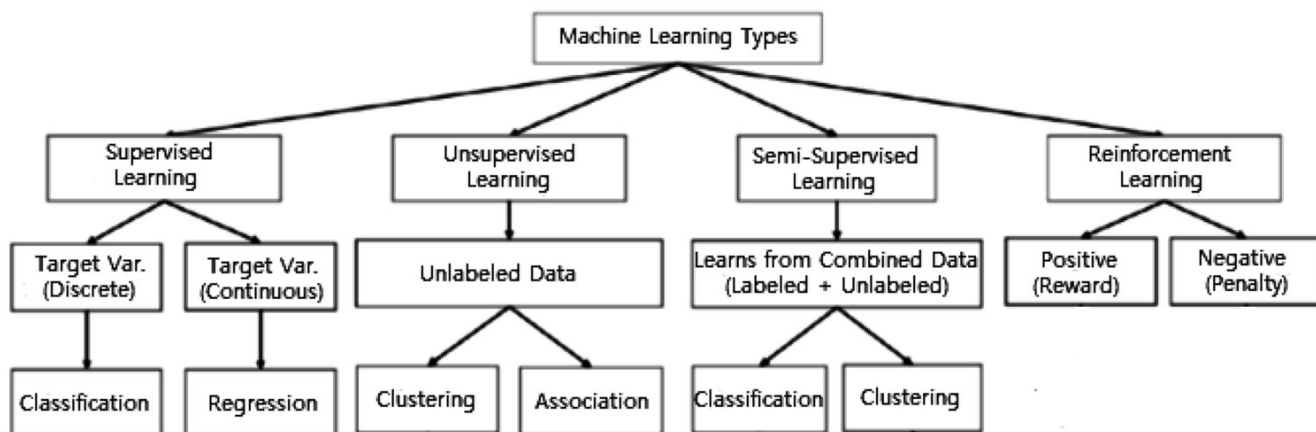


Рисунок 2.1 – Види машинного навчання

Далі опишемо кожен вид машинного навчання та їх застосовність для вирішення реальних проблем.

#### 2.1.1 Кероване навчання

Кероване навчання (навчання з учителем) – це метод машинного навчання, заснований на використанні даних, які мають присвоєні класи або ярлики (labels). У цьому процесі інженер контролює навчання, надаючи алгоритму великі обсяги розмічених даних. Шляхом аналізу цих даних модель виявляє закономірності та

структури, що дозволяє їй точно визначати класи об'єктів нових даних. Однак для досягнення високої точності потрібен значний обсяг розмічених даних, що може бути трудомістким етапом у процесі навчання [9].

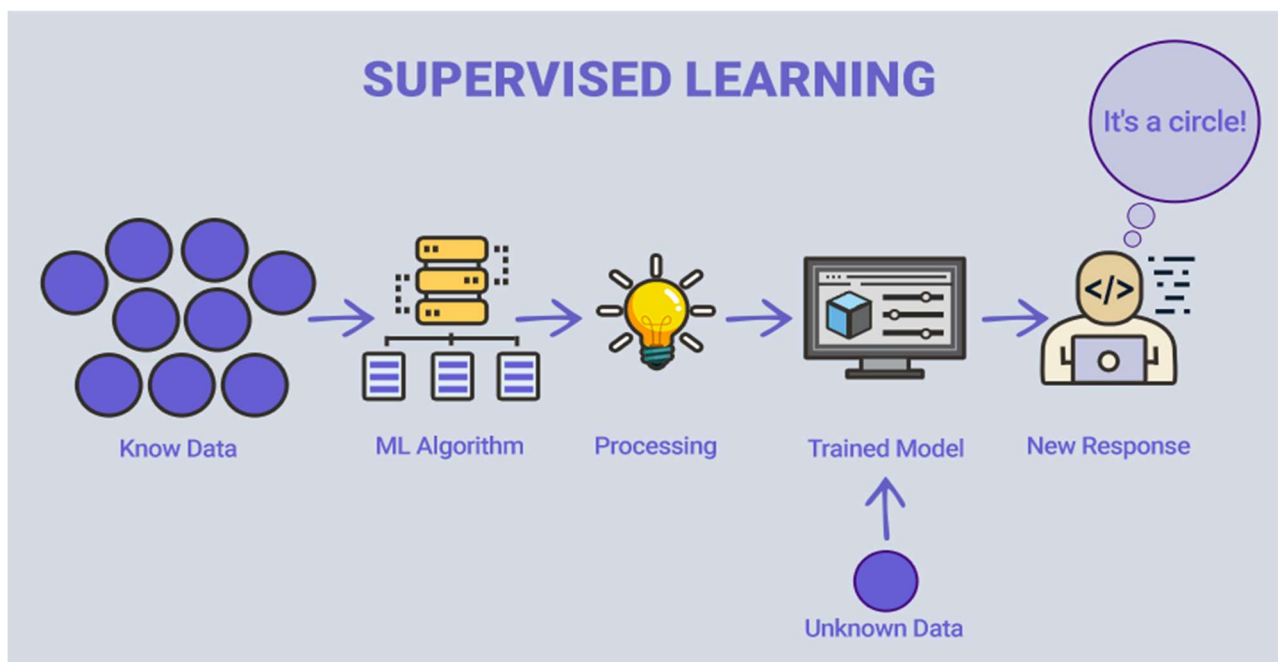


Рисунок 2.2 – Принцип роботи керованого навчання

У випадку, наведеному на рис. 2.2, модель намагається з'ясувати, чи є дані колом або іншою фігурою. Як тільки модель буде добре навчена, вона визначить, що дані це коло, і дасть бажану відповідь.

Кероване навчання може вирішувати, відповідно до розподілу вказаного на рис. 2.1, два типи завдань: класифікація та регресійний аналіз.

Класифікаційні алгоритми в машинному навчанні підходять для завдань, пов'язаних із прогнозуванням належності об'єктів до заздалегідь визначених категорій чи класів. Вони добре підходять для вирішення завдань, таких як:

- розпізнавання образів;
- фільтрування спаму;
- медична діагностика;
- фінансовий моніторинг.

Проте класифікаційні алгоритми не можуть бути оптимальними для завдань, де потрібно передбачення чисельного значення (регресія). Також вони можуть

неефективно працювати у випадках, де класи неоднорідні або перекриваються, і коли в наявних даних є багато шуму. У такому разі краще використовувати регресійні моделі чи інші типи алгоритмів, більш підходящих для конкретних умов завдання.

Регресійний аналіз у керованому навчанні – це метод, який використовується для аналізу відношення між залежною змінною (цільовою) і однією або декількома незалежними змінними (ознаками). Він дозволяє передбачати значення залежної змінної на основі значень незалежних змінних, виявляти закономірності та будувати моделі, які можуть використовуватися для прогнозування значень на основі вхідних даних.

Декілька типових завдань, для яких застосовується регресійний аналіз:

- прогнозування;
- економічний аналіз;
- маркетингові дослідження;
- медична статистика;
- наукові дослідження;
- інженерні програми.

### 2.1.2 Некероване навчання

При некерованому навчанні (навчанні без вчителя) використовуються невідомі та не помічені дані (unlabeled data), що означає, що вони не були попередньо вивчені. Без урахування наперед відомої інформації, вхідні дані передаються алгоритму машинного навчання для тренування моделі. Основна мета в машинному навчанні без вчителя полягає у виявленні структури та закономірностей у даних, без явного керівництва у вигляді цільових відповідей [9].

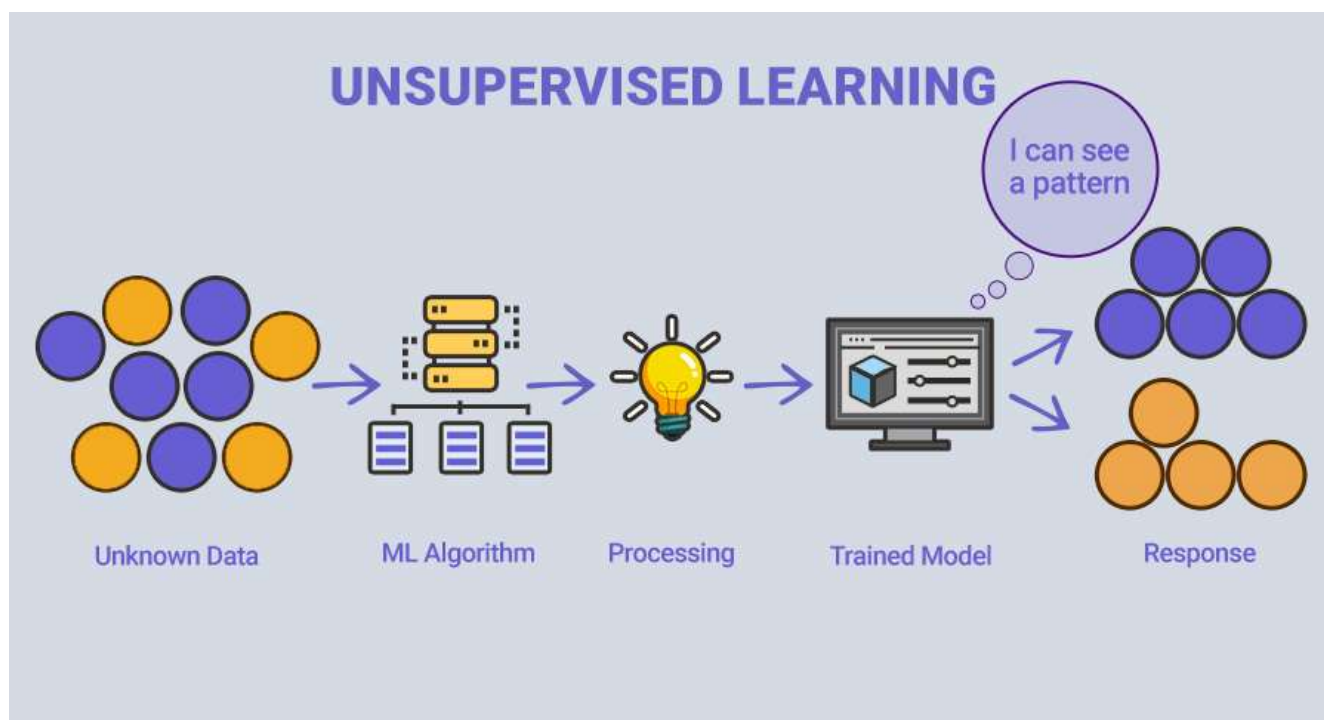


Рисунок 2.3 – Принцип роботи некерованого навчання

В основі некерованого навчання лежить робота з великими наборами даних, які можуть бути структуровані або неструктуровані. Методи машинного навчання без вчителя включають кластеризацію, зниження розмірності, навчання уявлень, виявлення викидів та асоціативні правила. Кожен з цих методів має свої особливості та застосування.

Кластеризація є однією з основних технік і використовується для групування об'єктів на основі їх подібності. Цей підхід дозволяє визначати природні групи в даних, що є корисним у багатьох галузях. Наприклад, у маркетингу кластеризація допомагає сегментувати клієнтів для створення цільових кампаній, у біології – класифікувати види на основі генетичних даних, а у сфері безпеки – виявляти аномальні поведінкові патерни, що можуть свідчити про шахрайство.

Зниження розмірності спрямоване на скорочення кількості змінних у наборі даних при збереженні його основних характеристик. Це дозволяє спростити моделі та підвищити їхню ефективність. Метод головних компонент (PCA) або t-розподілене вкладення стохастичної близькості (t-SNE) дозволяють зменшити кількість ознак і виявити ключові фактори, що впливають на дані. Це особливо

корисно для візуалізації складних даних і попереднього аналізу великих наборів інформації.

Навчання уявлень полягає в автоматичному створенні представлень даних, які містять значимі ознаки без потреби в явному програмуванні цих ознак. Нейронні мережі, такі як автокодувальники, використовуються для цього завдання, дозволяючи моделі навчатися компактних і корисних представлень вхідних даних. Це може значно полегшити подальшу обробку даних і покращити результати інших алгоритмів машинного навчання.

Виявлення викидів (аномалій) дозволяє знаходити незвичайні або аномальні зразки, які відрізняються від загальних трендів. Це надзвичайно важливо для виявлення шахрайських транзакцій у фінансових системах, моніторингу промислового обладнання для виявлення несправностей або забезпечення кібербезпеки шляхом виявлення підозрілої активності.

Асоціативні правила використовуються для виявлення зв'язків та взаємозв'язків між змінними в наборі даних. Наприклад, в аналізі купівельної поведінки цей метод може допомогти виявити, які товари часто купуються разом, що сприяє ефективному плануванню рекламних кампаній та організації товарів у магазинах.

Машинне навчання без вчителя є надзвичайно корисним для отримання попередніх інсайтів з даних та виявлення прихованих закономірностей. Воно дозволяє дослідникам та фахівцям з даних краще розуміти складні структури в наборах даних і робити обґрунтовані висновки для подальшого аналізу та прийняття рішень. Завдяки своїй здатності обробляти великі обсяги даних без потреби у попередньому маркуванні, ці методи є незамінними в багатьох галузях, від фінансів та маркетингу до біології та кібербезпеки.

### 2.1.3 Напівконтрольоване навчання

Напівконтрольоване навчання можна визначити як гібридизацію вищезгаданих методів навчання під контролем і без контролю, оскільки воно оперує як з міченими, так і з неміченими даними. Таким чином, він знаходиться

між навчанням «без нагляду» і навчанням «з наглядом». У реальному світі марковані дані можуть бути рідкісними в деяких контекстах, а немарковані дані є численними, де напівконтрольоване навчання є корисним [5]. Будова типового алгоритму з напівконтрольованим навчанням відображена на рис. 2.4.

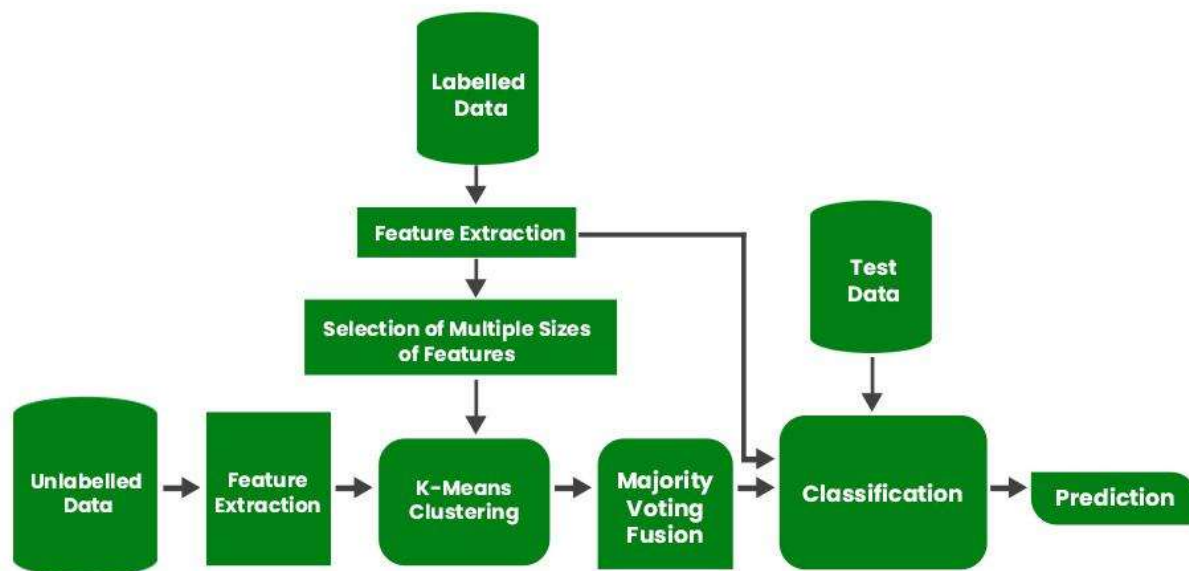


Рисунок 2.4 – Принцип роботи напівконтрольованого навчання

Кінцевою метою моделі навчання з частковим керуванням є забезпечення кращого результату прогнозування, ніж той, що отриманий з використанням лише мічених даних з моделі. Деякі прикладні області, де використовується напівкерване навчання, включають машинний переклад, виявлення шахрайства, маркування даних і класифікацію текстів [5].

#### 2.1.4 Навчання з підкріпленням

Машинне навчання з підкріпленням включає агента, середовище та дії, де агент взаємодіє з навколишнім середовищем, приймаючи рішення. Навчання полягає у виборі дій для максимізації очікуваної винагороди, особливо при дотриманні розумної стратегії. У штучному інтелекті цей процес включає отримання нагороди чи штрафу за кожну дію з метою максимізації загальної кількості балів. Це можна порівняти з новачком, який грає в гру та поступовим

поліпшенням продуктивності через аналіз взаємозв'язку дій, відображення та рахунку [9]. Наведений процес навчання відображено на рис. 2.5.

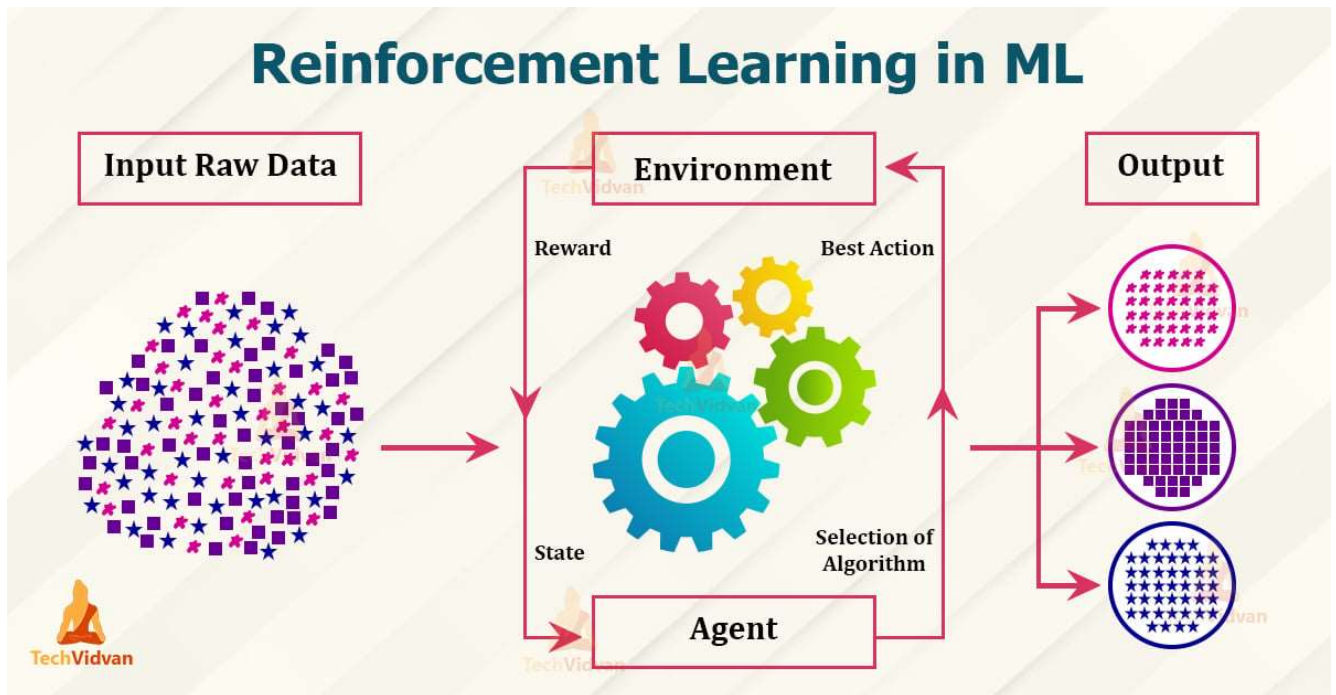


Рисунок 2.5 – Принцип роботи навчання з підкріпленням

Оскільки навчання з підкріпленням пов'язане з прийняттям рішень у непередбачуваному середовищі, що було наведено раніше, воно є основною сферою інтересу робототехніки. Для виконання простих і повторюваних завдань прийняття рішень може бути простим. Але більш складні завдання, такі як спроби імітувати людську поведінку або автоматизувати водіння, передбачають взаємодію з мінливими умовами реального світу, що характеризуються великою кількістю змінних. Дослідження показують, що глибоке навчання з підкріпленням за допомогою глибоких нейронних мереж допомагає вирішувати такі завдання, особливо щодо узагальнення і відображення високорозмірних сенсорних входів у виходи керованих систем. Також дослідження відмічають, що глибоке навчання з підкріпленням за допомогою роботів значною мірою покладається на зібрані набори даних, і тому нещодавні роботи досліджують шляхи збору реальних даних і перепрофілювання попередніх даних для вдосконалення систем навчання з підкріпленням [10].

Важливо відмітити, що нещодавні дослідження показують, що використання методів та інструментів обробки природної мови - наприклад, великих мовних моделей (ВММ) - може покращити узагальнення в системах навчання з підкріпленням через текстове представлення реального середовища. Багато досліджень показують, як інтерактивні текстові середовища забезпечують економічно ефективні альтернативи тривимірним середовищам при навчанні агентів, що навчаються, послідовним завданням прийняття рішень. Глибоке навчання з підкріпленням також лежить в основі текстового прийняття рішень в чат-ботах. Насправді, навчання з підкріпленням перевершує інші методи покращення реакції чат-ботів на діалог [10].

## 2.2 Алгоритми машинного навчання

Алгоритми машинного навчання (МН) становлять основу аналітичних систем, що здатні вивчати закономірності та робити прогнози на основі даних. Ці алгоритми зазвичай класифікуються залежно від типу вихідних даних і завдання, яке необхідно вирішити. Серед найпопулярніших виокремлюють:

- лінійна регресія;
- логістична регресія;
- дерева рішень;
- випадкові ліси;
- градієнтний бустинг;
- кластеризація k-середніх;
- нейронні мережі, зокрема глибоке навчання.

Кожен з цих алгоритмів має свої унікальні переваги та обмеження. Вибір конкретної моделі залежить від специфіки задачі, доступності та обсягу даних, а також вимог до точності та швидкості обробки.

### 2.2.1 Лінійна регресія

Лінійна регресія - це метод керованого машинного навчання, який використовується для прогнозування та передбачення значень, що знаходяться в межах безперервного діапазону, наприклад, кількості продажів або цін на житло. Цей метод запозичений зі статистики і зазвичай використовується для встановлення зв'язку між вхідною змінною ( $X$ ) і вихідною змінною ( $Y$ ) [11].

Модель представляє залежність як лінійну функцію:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon$$

де  $\beta_0$  – це перетин з оссю  $Y$  (константа);

$\beta_1 + \dots + \beta_n$  – коефіцієнти при незалежних змінних, що вказують на міру зміни  $Y$  при зміні відповідного  $X$  на одиницю;

$\epsilon$  – випадкова помилка.

Простіше кажучи, лінійна регресія бере набір точок даних з відомими вхідними та вихідними значеннями і знаходить лінію, яка найкраще відповідає цим точкам. Ця лінія, відома як «лінія регресії», слугує прогнозною моделлю. Використовуючи цю лінію, ми можемо оцінити або передбачити вихідне значення ( $Y$ ) для заданого вхідного значення ( $X$ ).

Лінійна регресія в першу чергу використовується для прогнозного моделювання, а не для категоризації. Вона корисна, коли ми хочемо зрозуміти, як зміни вхідної змінної впливають на вихідну змінну. Аналізуючи нахил і перехват лінії регресії, ми можемо отримати уявлення про взаємозв'язок між змінними і робити прогнози на основі цього розуміння [11]. Наприклад, вона використовується для прогнозування числових значень, що може бути корисно для оцінки ймовірності бронювання на основі певних характеристик користувача або продукту. Також вона широко застосовується у фінансах, економіці, соціальних науках для прогнозування тенденцій та виявлення взаємозв'язків між змінними.

### 2.2.2 Логістична регресія

Логістична регресія — це статистична модель, яка використовується для оцінки ймовірності належності до певної категорії (зазвичай бінарної: 0 або 1). Вона виражає логарифм відношення шансів лінійної комбінації вхідних змінних:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

де  $p$  – ймовірність належності до категорії 1.

Логістична регресія прогнозує ймовірність того, що вхідні дані можуть бути віднесені до одного первинного класу. Однак на практиці її зазвичай використовують для групування виходів у дві категорії: первинний клас і непервинний клас. Для цього логістична регресія створює поріг або межу для бінарної класифікації. Наприклад, будь-яке значення результату від 0 до 0,49 можна віднести до однієї групи, тоді як значення від 0,50 до 1,00 можна віднести до іншої групи [11].

Отже, логістична регресія зазвичай використовується для бінарної категоризації, а не для прогнозного моделювання. Вона дозволяє віднести вхідні дані до одного з двох класів на основі оцінки ймовірності та визначеного порогу. Наприклад, у рекомендаційних системах вона може допомогти визначити, чи буде певний продукт цікавим конкретному користувачеві.

Виходячи з всього вище описаного, це робить логістичну регресію потужним інструментом для таких завдань, як розпізнавання зображень, виявлення спаму в електронній пошті або медична діагностика, де нам потрібно розділити дані на окремі класи.

### 2.2.3 Дерево рішень

Дерево рішень — це алгоритм машинного навчання, який використовується для вирішення задач класифікації та регресії. Він представляє собою деревоподібну структуру, де кожен внутрішній вузол відповідає «питанню» на основі одного з атрибутів, кожна гілка до наступного вузла відповідає відповіді на це питання, а

кожен листовий вузол (кінцевий вузол дерева) представляє прогнозовану мітку або значення [12].

Основний принцип роботи дерева рішень полягає у використанні набору правил рішень, що базуються на атрибутах вхідних даних, для розділення набору даних на класи або для прогнозування значення. Процес побудови дерева рішень починається з визначення кращого атрибута, який використовується в якості кореня дерева, і розбиття набору даних на піднабори. Цей процес повторюється рекурсивно для кожного отриманого піднабору відповідно до кожного атрибута, поки не будуть виконані умови зупинки, такі як досягнення максимальної глибини дерева або коли кількість елементів у піднаборі не перевищує певного порогу.

Алгоритми дерев рішень популярні в машинному навчанні, оскільки вони можуть легко і просто обробляти складні набори даних. Структура алгоритму спрощує розуміння та інтерпретацію процесу прийняття рішень. Ставлячи послідовність запитань і слідуючи за відповідними гілками, дерева рішень дозволяють нам класифікувати або прогнозувати результати на основі характеристик даних.

Важливо відмітити, що хоча дерева рішень корисні завдяки їхній простоті та інтуїтивно зрозумілій візуалізації, але вони також мають обмеження, такі як схильність до перенавчання та певні обмеження в точності порівняно з іншими більш складними моделями машинного навчання.

Дерева рішень знаходять своє застосування в багатьох областях, включаючи кредитний скоринг, медичну діагностику, машинне зору, фільтрацію спаму та багато інших завдань, де потрібні рішення на основі аналізу набору характеристик об'єктів.

#### 2.2.4 Випадкові ліси

Алгоритм «Випадкові ліси» (Random Forests) представляє собою ансамблевий алгоритм машинного навчання, який об'єднує прогнози з декількох дерев рішень для вироблення кінцевого рішення. Основна ідея полягає у побудові багатьох дерев рішень на підставі випадково вибраних підмножин навчального

набору даних та використанні голосування (для задач класифікації) або усереднення (для задач регресії) для визначення кінцевого прогнозу [13].

В свою чергу, описаний алгоритм будується шляхом декількох кроків:

1) бутстрепінг: Випадкові ліси використовують метод бутстрепінг (bootstrap sampling), що передбачає створення випадкових підмножин навчального набору даних з повтореннями. Для кожного дерева створюється своя випадкова підмножина даних;

2) розділення за допомогою випадкових ознак: При побудові кожного вузла в дереві розглядається лише обмежена кількість випадково вибраних ознак. Це забезпечує різноманітність у деревах, знижує кореляцію між ними та допомагає зменшити перенавчання;

3) голосування або усереднення для визначення прогнозу: Кінцевий прогноз моделі випадкових лісів визначається шляхом голосування (більшість голосів для класифікації) або усереднення (середнє значення для регресії) прогнозів усіх дерев в лісі.

Цей алгоритм вирішує загальну проблему, яка називається «перенавчання», що може статися з окремими деревами рішень. Перенавчання відбувається, коли дерево рішень стає занадто тісно пов'язаним зі своїми навчальними даними, що робить його менш точним при отриманні нових даних.

Випадкові ліси застосовуються в широкому спектрі задач від класифікації та регресії до виявлення аномалій і обробки зображень. Вони особливо ефективні в задачах, де набір даних містить велику кількість ознак, частина з яких може мати обмежений вплив на цільову змінну. Їхня здатність до обробки даних без попереднього видалення корельованих ознак і роботи з неповними даними робить їх цінним інструментом в арсеналі фахівців з машинного навчання.

### 2.2.5 Градієнтний бустинг

Градієнтний бустинг — це потужна техніка машинного навчання для задач класифікації, регресії та інших прогнозних задач, що базується на ідеї поєднання прогнозів з множини простіших моделей для створення більш точної та надійної

композитної моделі. Вона відноситься до класу алгоритмів ансамблевого навчання, де послідовні моделі (зазвичай дерева рішень) будуються так, що кожна наступна модель коригує помилки попередньої [13].

Основна ідея градієнтного бустингу полягає у послідовному додаванні до ансамблю нових моделей, які виправляють помилки, зроблені попередніми моделями, за рахунок мінімізації втрат, що вимірюються градієнтом функції втрат. Кожна нова модель в ансамблі фокусується на найбільш складних випадках, які були неправильно класифіковані або недостатньо добре апроксимовані попередніми моделями.

Градієнтний бустінг ефективний у роботі зі складними проблемами та великими наборами даних. Він може вловити складні закономірності та залежності, які можуть бути пропущені однією моделлю. Тому він демонструє високу ефективність у широкому спектрі задач, від конкурсів з аналізу даних до реальних застосувань у промисловості, фінансах, медицині та багатьох інших областях. Він особливо цінний завдяки своїй здатності до обробки великих обсягів даних та високій точності прогнозування.

### 2.2.6 Кластеризація k-середніх

K-середні - це некерований алгоритм, який зазвичай використовується для кластеризації та розпізнавання образів. Він спрямований на групування точок даних на основі їхньої близькості одна до одної. Подібно до методу K-найближчих сусідів (KNN), кластеризація K-середніх використовує концепцію близькості для виявлення шаблонів у даних [11].

Для обробки навчальних даних алгоритм K-середніх в інтелектуальному аналізі даних починає з першої групи випадково вибраних центроїдів, які використовуються як початкові точки для кожного кластера, а потім виконує ітеративні (повторювані) обчислення для оптимізації позицій центроїдів [14].

Він припиняє створення та оптимізацію кластерів, коли у будь-якого з них:

- центроїди стабілізувалися - їхні значення не змінюються, тобто кластеризація пройшла успішно;

- досягнуто заданої кількості ітерацій.

Алгоритми кластеризації особливо корисні для великих наборів даних і можуть дати уявлення про внутрішню структуру даних, групуючи схожі точки разом. Але, в свою чергу, потрібно пам'ятати, що вони можуть давати помилки при обробці викидів.

Описаний алгоритм застосовується в різних сферах, таких як сегментація клієнтів, стиснення зображень і виявлення аномалій.

### 2.2.7 Нейронні мережі

Нейронні мережі — це системи, натхненні будовою мозку, які використовують шари взаємопов'язаних «нейронів» для виконання складних обчислень. Кожна нейронна мережа складається з шарів вузлів, або штучних нейронів - вхідного шару, одного або декількох прихованих шарів і вихідного шару. Кожен вузол з'єднується з іншими і має власну вагу та поріг спрацьовування. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі [15].

Нейронні мережі покладаються на навчальні дані, щоб навчатися і покращувати свою точність з часом. Після того, як вони точно налаштовані на точність, вони стають потужними інструментами в комп'ютерних науках і штучному інтелекті, дозволяючи класифікувати і кластеризувати дані з високою швидкістю. Завдання з розпізнавання мовлення або розпізнавання зображень можуть займати хвилини, а не години, якщо порівнювати з ручною ідентифікацією, яку проводять люди-експерти. Одним з найвідоміших прикладів нейронної мережі є пошуковий алгоритм Google [16].

Описані мережі іноді називають штучними нейронними мережами (ШНМ) або імітаційними нейронними мережами (ІМНМ). Вони є підмножиною машинного навчання і лежать в основі моделей глибокого навчання.

Нейронні мережі можна класифікувати на різні типи, які використовуються для різних цілей. Нижче наведено найпоширеніші типи нейронних мереж, які застосовуються у найпоширеніших випадках їх використання:

- згорткові нейронні мережі (ЗНМ) схожі на мережі прямого поширення, але їх зазвичай використовують для розпізнавання зображень, розпізнавання образів та/або комп'ютерного зору. Ці мережі використовують принципи лінійної алгебри, зокрема множення матриць, для виявлення шаблонів на зображенні [16];
- рекурентні нейронні мережі (РНМ) ідентифікуються за петлями зворотного зв'язку. Ці алгоритми навчання в першу чергу використовуються при використанні даних часових рядів для прогнозування майбутніх результатів, таких як прогнози фондового ринку або прогнозування продажів [16].

Нейронні мережі знайшли застосування в широкому спектрі доменів, включаючи, але не обмежуючись ними, комп'ютерний зір, обробку природної мови, прогнозування часових рядів, автономне водіння та багато інших. Особливо значні успіхи були досягнуті в розробці глибоких нейронних мереж (Deep Learning), які використовують багат шарові нейронні мережі для аналізу великих обсягів даних, знаходження складних закономірностей та виконання завдань, таких як розпізнавання зображень, обробка природної мови та автономне керування.

### 2.3 Огляд функціонування існуючих рішень

В контексті дослідження видів та алгоритмів машинного навчання, важливим аспектом є глибоке розуміння існуючих рішень, їхніх функціональних можливостей, переваг та обмежень в заданій предметній області. Огляд існуючих рішень дозволяє ідентифікувати поточні тренди, найбільш вдалих представників ринку та основні напрямки подальших досліджень.

Основними аналогами серед сучасних рішень в даному дослідженні є:

- Amazon Recommendation System;
- Netflix Recommendation Engine;
- Google Search Engine.

Далі розглянемо їх будову, принцип функціонування та надання рекомендацій, а також яким чином вони використовують дані як про послуги так і про клієнтів.

### 2.3.1 Amazon Recommendation System

Рекомендаційна система Amazon вирізняється своєю складністю та ефективністю в електронній комерції завдяки використанню різноманітних алгоритмів та методів аналізу даних для надання персоналізованих рекомендацій. Основу її роботи складають техніки колаборативної та контентної фільтрації, які доповнюються гібридними підходами для підвищення точності рекомендацій.

Колаборативна фільтрація застосовується для аналізу матриці користувач-товар, що відображає взаємодії між користувачами та продуктами, зокрема покупки та рейтинги. Для визначення схожості між користувачами використовуються методи косинусної схожості та коефіцієнт кореляції Пірсона. Крім цього, колаборативна фільтрація товар-товар дозволяє рекомендувати продукти на основі їх схожості, наприклад, пропонуючи чохол для подорожей користувачу, який купив VR-гарнітуру, враховуючи досвід інших користувачів.

Контентна фільтрація, у свою чергу, аналізує характеристики товарів, такі як описи та категорії, щоб рекомендувати схожі продукти. Таким чином, якщо користувач переглядає ігровий ноутбук, система запропонує інші ноутбуки з подібними характеристиками. Для досягнення максимальної точності рекомендацій Amazon використовує гібридний підхід, який поєднує колаборативну та контентну фільтрацію. Використання технік факторизації матриць дозволяє зменшити розмірність матриць користувач-товар і виявити латентні фактори, що впливають на вподобання користувачів.

Для покращення рекомендаційної системи застосовуються передові методи, такі як бандитські алгоритми, що динамічно налаштовують рекомендації на основі взаємодій користувачів з продуктами в реальному часі, та причинно-наслідковий аналіз, який ідентифікує фактори, що спричиняють взаємодії користувачів з певними продуктами.

Рекомендаційні механізми інтегровані у всі платформи Amazon, забезпечуючи персоналізовані пропозиції на головній сторінці, сторінках продуктів, у email та мобільних повідомленнях, а також у голосових та фізичних магазинах, таких як Amazon Alexa та Amazon Go. Персоналізація рекомендацій базується на великій кількості даних, включаючи явні відгуки (історія покупок, рейтинги, відгуки), неявні відгуки (поведінка під час перегляду, пошукові запити, час, проведений на сторінках продуктів) та контекстуальні дані (інформація про місцезнаходження, час доби, тип пристрою).

Amazon приділяє значну увагу приватності користувачів, анонімізуючи дані та використовуючи агреговані набори даних. Однак, існують занепокоєння щодо створення інформаційних бульбашок та можливих упереджених рекомендацій, що підкреслює важливість етичного підходу в роботі рекомендаційних систем.

Виходячи з аналізу системи надання рекомендацій Amazon, то до переваг цієї системи можна віднести:

- висока персоналізація рекомендацій;
- здатність прогнозувати інтереси користувача на основі широкого спектру даних;
- покращення продажів через ефективне цільове маркетингове спрямування.

В свою чергу, до недоліків відносяться:

- потенційне порушення конфіденційності через глибокий аналіз поведінкових даних.
- ризик «фільтраційного бульбашки», коли користувачі бачать обмежений вибір, базований лише на їхніх попередніх інтересах.

### 2.3.2 Netflix Recommendation Engine

Рекомендаційна система Netflix є важливим компонентом, що забезпечує персоналізований підхід до кожного користувача, дозволяючи рекомендувати фільми та серіали на основі їхніх індивідуальних вподобань. Її робота базується на

поєднанні колаборативної та контентної фільтрації, що забезпечує високу точність та ефективність рекомендацій.

Колаборативна фільтрація Netflix використовує матрицю користувач-елемент для визначення зв'язків між користувачами та продуктами на основі їхніх взаємодій, таких як перегляди та оцінки. Алгоритми найближчих сусідів знаходять користувачів зі схожими вподобаннями, дозволяючи пропонувати їм відповідні фільми. Це дає змогу формувати рекомендації, враховуючи вподобання користувачів, які мають схожі смаки.

Одночасно, контентна фільтрація аналізує характеристики контенту, такі як жанр, актори та рік випуску, створюючи рекомендації на основі подібних характеристик. Наприклад, якщо користувач часто дивиться фільми певного жанру, система рекомендує інші фільми цього жанру. Для досягнення максимальної точності рекомендацій Netflix використовує гібридні моделі, які поєднують обидва підходи. Застосування факторизації матриць дозволяє зменшити розмірність даних та виявити латентні фактори, що впливають на вподобання користувачів, забезпечуючи більш точні та релевантні рекомендації.

Одним із ключових елементів покращення рекомендаційної системи є інтеграція глибинного навчання. Хоча початкові результати були незначними, додавання великої кількості різномірних даних суттєво покращило ефективність моделей. Вирівнювання офлайн та онлайн метрик залишається важливим викликом, який впливає на точність рекомендацій.

Інфраструктура Netflix підтримує роботу рекомендаційної системи за допомогою розподілених обчислювальних систем, що дозволяють ефективно обробляти великі обсяги даних. Відмовостійкість досягається завдяки реплікації даних на кількох серверах, а кешування забезпечує швидкий доступ до часто запитуваних даних, що зменшує затримки і підвищує ефективність системи.

Персоналізація рекомендацій Netflix базується на збиранні великої кількості даних про користувачів, включаючи історію переглядів, рейтинги, пошукові запити та контекстуальні дані, такі як час доби та тип пристрою. Це дозволяє створювати максимально точні та релевантні рекомендації. Водночас, компанія приділяє

значну увагу приватності користувачів, анонімізуючи дані та використовуючи агреговані набори даних для аналізу, що допомагає мінімізувати ризики витоку персональної інформації. Постійне вдосконалення алгоритмів та врахування зворотного зв'язку від користувачів сприяють підвищенню точності та ефективності рекомендаційної системи Netflix, забезпечуючи користувачам високий рівень задоволеності від використання платформи.

Виходячи з проведеного аналізу, рекомендаційна система Netflix має такі переваги:

- високий рівень персоналізації;
- здатність адаптуватися до змін у перевагах користувачів;
- підвищення задоволеності користувачів і залучення.

Також, як визначено раніше, ця система рекомендацій має і недоліки, а саме:

- велика залежність від точності введених даних користувачами;
- обмеження в рекомендаціях нового контенту, який ще не має великої історії переглядів.

### 2.3.3 Google Search Engine

Пошукова система Google є складною технологією, яка використовує різноманітні методи для організації, ранжування та подання інформації користувачам. Основні етапи її роботи включають організацію інформації, ранжування результатів, суворе тестування та виявлення спаму, що забезпечує користувачам найрелевантніші та найкорисніші результати для їхніх пошукових запитів.

Організація інформації здійснюється за допомогою спеціальних програм, відомих як веб-сканери або боти. Ці боти автоматично відвідують веб-сторінки, збирають інформацію та зберігають її у великій базі даних, відомій як індекс. Кожна сторінка аналізується на предмет ключових сигналів, таких як ключові слова, структура сторінки, заголовки та інші метадані. Це дозволяє створити детальну ієрархію інформації, яка згодом використовується для швидкого та ефективного пошуку.

Коли користувач вводить пошуковий запит, алгоритми Google аналізують індексовані сторінки та визначають їхню релевантність до запиту. Для цього використовуються сотні факторів, включаючи відповідність запиту (ключові слова), якість та авторитетність джерела (рейтинг сторінки, кількість та якість зворотних посилань), а також користувацький досвід (швидкість завантаження сторінки, зручність навігації). Цей процес ранжування результатів дозволяє відфільтрувати найбільш релевантні та якісні сторінки, забезпечуючи користувачам найкращий досвід пошуку.

Google постійно вдосконалює свої алгоритми через масштабне тестування. Використовуються як автоматизовані тести, так і оцінки якості пошуку, які виконуються людьми-оцінювачами. Ці оцінки допомагають визначити, наскільки результати відповідають очікуванням користувачів, що дозволяє компанії постійно адаптувати та покращувати свої алгоритми.

Одним з важливих аспектів роботи пошукової системи є виявлення спаму та маніпулятивних технік, які намагаються штучно підвищити рейтинг сторінок. Для цього Google використовує спеціальні алгоритми та ручні перевірки, що забезпечують якість та достовірність результатів пошуку. Ці заходи допомагають підтримувати високу якість інформації, що подається користувачам, і мінімізувати вплив недобросовісних практик.

Виходячи з всього вищесказаного, то до переваг Google Search Engine відноситься:

- висока релевантність і точність пошукових результатів;
- здатність швидко обробляти та аналізувати величезні обсяги інформації.

В свою чергу, до недоліків можна віднести:

- ризик створення «фільтраційної бульбашки», обмежуючи доступ до інформації;
- питання приватності, пов'язані з аналізом пошукової активності.

### 2.3.4 Результати аналізу функціонування існуючих рішень

Аналіз існуючих рекомендаційних та пошукових систем, таких як Amazon, Netflix та Google, дозволяє виділити ключові аспекти, що сприяють їхній ефективності та точності. Кожна з цих систем використовує складні комбінації алгоритмів та технологій для забезпечення персоналізованих та релевантних результатів, що відповідають потребам користувачів.

Рекомендаційна система Amazon поєднує колаборативну та контентну фільтрацію. Колаборативна фільтрація використовує матрицю користувач-товар для аналізу взаємодій, тоді як контентна фільтрація аналізує характеристики продуктів. Гібридний підхід, що поєднує обидві техніки, забезпечує високу точність рекомендацій.

Netflix також використовує комбінацію колаборативної та контентної фільтрації. Алгоритми найближчих сусідів допомагають знаходити користувачів зі схожими вподобаннями, а контентна фільтрація аналізує характеристики контенту. Інтеграція глибинного навчання та факторизація матриць сприяють покращенню рекомендацій.

Google забезпечує високу якість пошуку через організацію інформації за допомогою веб-сканерів, ранжування результатів на основі численних факторів та постійне вдосконалення алгоритмів через масштабне тестування. Спеціальні алгоритми та ручні перевірки допомагають боротися зі спамом.

Проведений аналіз існуючих рішень у сфері рекомендаційних та пошукових систем показує, що успіх цих технологій базується на інтеграції різних методів та алгоритмів, які взаємодіють між собою для досягнення високої точності та релевантності результатів. Рекомендаційні системи Amazon і Netflix, а також пошукова система Google використовують складні комбінації колаборативної та контентної фільтрації, глибинного навчання, факторизації матриць та інших передових технологій для забезпечення персоналізованих і точних результатів.

Колаборативна фільтрація, яка використовується в обох рекомендаційних системах, заснована на аналізі взаємодій між користувачами та продуктами або контентом. Вона дозволяє виявляти схожі вподобання серед користувачів, що дає змогу формувати рекомендації на основі їхніх спільних інтересів. Це є надзвичайно

ефективним методом, оскільки враховує реальні дії користувачів, такі як перегляди, покупки та рейтинги.

Контентна фільтрація доповнює колаборативну, аналізуючи характеристики продуктів або контенту, такі як жанр, актори чи технічні параметри. Це дозволяє створювати рекомендації на основі схожих характеристик, що забезпечує різноманітність та гнучкість у наданні рекомендацій. Гібридні моделі, які поєднують ці два підходи, використовують факторизацію матриць для зменшення розмірності даних та виявлення латентних факторів, що впливають на вподобання користувачів. Це дозволяє забезпечити ще більшу точність та релевантність рекомендацій.

Інтеграція глибинного навчання в рекомендаційні системи, зокрема в Netflix, значно покращила якість рекомендацій після додавання великої кількості різноманітних даних. Глибине навчання дозволяє обробляти великі обсяги даних та виявляти складні патерни, які можуть бути неочевидними при використанні традиційних методів. Це дозволяє створювати більш точні та персоналізовані рекомендації, що відповідають індивідуальним вподобанням кожного користувача.

Пошукова система Google використовує веб-сканери для збору та організації інформації, зберігаючи її в індексі, який дозволяє швидко знаходити релевантні сторінки при введенні пошукового запиту. Процес ранжування результатів базується на численних факторах, таких як відповідність запиту, якість та авторитетність джерела, а також користувацький досвід. Це забезпечує високу точність та релевантність результатів, що відповідають запитам користувачів.

Постійне вдосконалення алгоритмів через масштабне тестування, яке включає автоматизовані тести та оцінки якості пошуку, виконувані людьми-оцінювачами, дозволяє Google підтримувати високий рівень якості результатів пошуку. Важливою частиною роботи пошукової системи є виявлення спаму та маніпулятивних технік, що забезпечується спеціальними алгоритмами та ручними перевітками. Це дозволяє мінімізувати вплив недобросовісних практик та підтримувати якість і достовірність результатів пошуку.

Отже, успіх сучасних рекомендаційних та пошукових систем базується на комплексному підході до обробки та аналізу даних, що включає використання колаборативної та контентної фільтрації, глибинного навчання, факторизації матриць та інших передових технологій. Постійне вдосконалення алгоритмів та технологій, а також увага до приватності користувачів та боротьба з маніпулятивними техніками є ключовими факторами, що забезпечують високу ефективність та точність цих систем. Це дозволяє компаніям, таким як Amazon, Netflix та Google, надавати своїм користувачам персоналізовані та релевантні результати, що відповідають їхнім індивідуальним потребам та вподобанням.

## 3 ПРОЕКТУВАННЯ АЛГОРИТМУ МАШИННОГО НАВЧАННЯ

### 3.1 Загальний огляд

Відповідно до проведеного аналізу предметної області, сучасний розвиток сфери онлайн-бронювання вимагає від рекомендаційних систем високої точності, адаптивності до змінних уподобань користувачів та здатності ефективно обробляти величезні обсяги даних. У цьому контексті алгоритми машинного навчання, зокрема на основі нейронних мереж, виступають як оптимальне рішення, здатне задовольнити зазначені вимоги. Метою проектування алгоритму є розробка моделі, яка забезпечує високу точність рекомендацій при одночасному скороченні часу їх генерації.

У відповідності до проведеного дослідження та беручи до уваги тренд на застосування машинного навчання, зокрема нейронних мереж глибокого навчання у системах рекомендацій, а також з урахуванням того факту, що у комерційних проектах найкраще себе показують моделі, що беруть за основу нейромережі, визначено взяти за основу нейронні мережі для створення моделі.

Проектування ефективного алгоритму машинного навчання передбачає визначення його архітектури, включно з кількістю шарів, типами нейронів, методами активації та алгоритмами оптимізації. Основні компоненти нейронної мережі включають:

- 1) вхідний шар, який приймає сирий датасет;
- 2) приховані шари, де відбувається основна обробка даних через взаємодію нейронів;
- 3) вихідний шар, який надає результат у форматі, зручному для інтерпретації (наприклад, оцінки ймовірності належності до класу).

Також треба відзначити важливість та актуальність застосування нейромереж у рекомендаційних системах через їхні переваги, що включає:

- масштабованість - здатність ефективно обробляти великі обсяги даних;
- гнучкість - можливість адаптації до різноманітних типів даних і задач;

- висока точність - завдяки глибокому аналізу даних і виявленню неочевидних взаємозв'язків;
- адаптивність - можливість обробляти дані різних типів і підлаштовуватися під оновлення даних.

Розробка та впровадження нейронних мереж несуть певні виклики, зокрема, потребу в значних обчислювальних ресурсах, необхідності збалансування між точністю та швидкістю реакції системи, а також забезпечення прозорості та інтерпретованості моделі для користувачів.

Проектування алгоритму машинного навчання на основі нейронних мереж для рекомендаційної системи вимагає глибокого розуміння як теоретичних основ машинного навчання, так і практичних аспектів їх застосування. Подолання існуючих викликів та ефективного використання потенціалу нейронних мереж може значно покращити якість рекомендацій, забезпечити користувачам більш персоналізований досвід та зробити процес онлайн-бронювання більш ефективним та задовільним [17].

Виходячи з наведеного огляду, метою цього розділу є проведення дослідження і створення моделі для подальшого застосування у алгоритмі машинного навчання. Де, в першу чергу, потрібно звернути увагу на вимоги, що ставляться перед такого роду алгоритмами відповідно до предметної області комерційних систем рекомендацій.

### 3.2 Аналіз вимог

Аналіз вимог є критично важливим кроком у процесі проектування алгоритму машинного навчання, особливо коли мова йде про розробку рекомендаційних систем у сфері онлайн-бронювання. Вимоги поділяються на функціональні (рис. 3.1) та нефункціональні (рис. 3.2), кожна з яких впливає на вибір моделі, її структуру, навчання, тестування, та оцінку ефективності. Відповідно до SWEBOOK визначаються, що до вимог залучаються як розробник так і стекхолдер, а саме користувач.

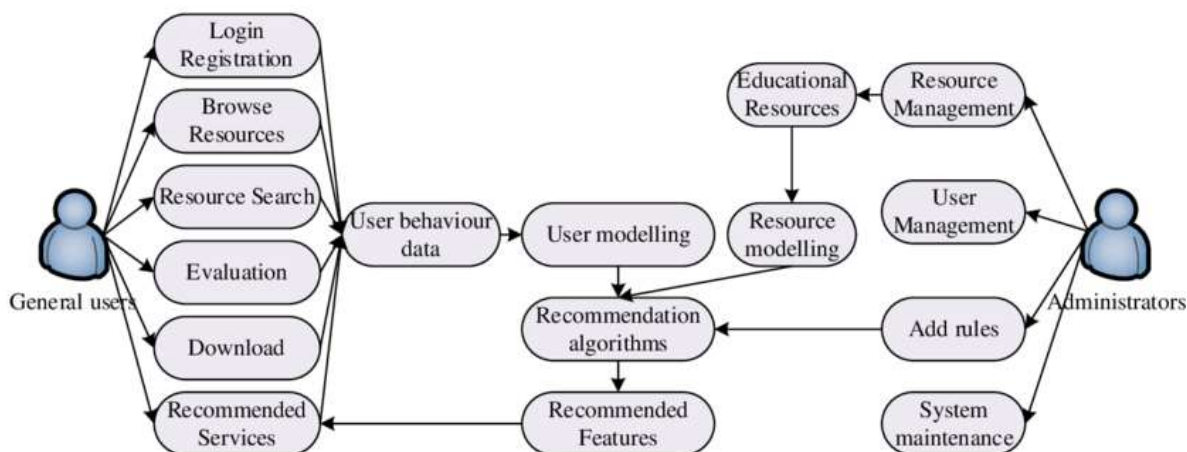


Рисунок 3.1 - Діаграма функціональних вимог для систем рекомендацій

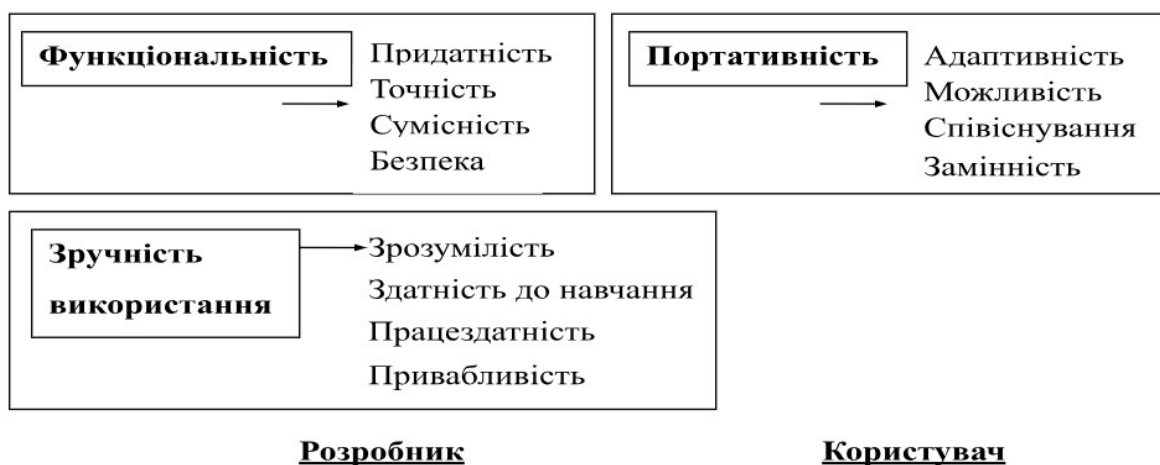


Рисунок 3.2 - Нефункціональні вимоги за SWEBOK

Функціональні вимоги націлені на визначення процесів, що будуть виконувати задачі пов'язані з заданою метою системи та визначають конкретні задачі, які алгоритм повинен виконувати. Тому оскільки ця робота має на меті створення алгоритму для системи рекомендацій, тому був створений наступний список вимог, що включає:

- точність рекомендацій: алгоритм повинен забезпечувати високу точність рекомендацій, враховуючи історію переглядів, покупок, переваги користувачів, та інші релевантні дані;
- швидкість відгуку: система має оперативно генерувати рекомендації, забезпечуючи користувачам негайний доступ до персоналізованих пропозицій;

- адаптивність до змін у поведінці користувачів: алгоритм має ефективно адаптуватися до змін у вподобаннях та інтересах користувачів;
- масштабованість: спроможність алгоритму обробляти зростаючі обсяги даних без значної втрати продуктивності.

Таким шляхом першим кроком для нас є формулювання системних вимог вищого порядку до ПЗ. Високорівневі системні вимоги, в першу чергу, показують нам яку модель якості хоче бачити користувач. Класичним інструментом, що дозволяє сформулювати таку модель якості є нотація SWEBOOK. Вона містить опис шістьох факторів якості. В літературі ця модель якості також отримала назву нефункціональні вимоги [18]. Нефункціональні вимоги описують якість та обмеження системи, включаючи:

- продуктивність: алгоритм повинен демонструвати високу продуктивність, навіть при обробці великих обсягів даних;
- масштабованість: система має бути спроектована таким чином, щоб легко масштабуватися у відповідь на збільшення кількості користувачів та обсягів даних;
- безпека: забезпечення захисту даних користувачів від несанкціонованого доступу та використання;
- інтеграція: здатність алгоритму інтегруватися з існуючими системами та технологічними стеками;
- інтерпретованість: моделі машинного навчання повинні бути інтерпретовані, тобто користувачі або розробники мають можливість розуміти, як були згенеровані рекомендації.

Детальний аналіз функціональних та нефункціональних вимог є фундаментом для розробки ефективного алгоритму машинного навчання для рекомендаційних систем. Врахування цих вимог на ранньому етапі проектування дозволяє створити міцну основу для розробки моделі, яка не тільки задовольнятиме потреби користувачів але й буде відповідати технічним та бізнес-вимогам.

Тому можна представити конкретизацію створених функціональних вимог:

- 1) точність рекомендацій:

- ціль - досягнення мінімум 90% точності у визначенні користувацьких переваг засобами класифікації;
- дані для навчання - використання даних про попередні пошукові запити, перегляди сторінок, історію покупок та відгуків користувачів;
- методи оцінювання - використання матриці плутанини та ROC-кривої для оцінки точності моделі;
- 2) швидкість відгуку:
  - ціль - генерація рекомендацій не повинна перевищувати 200 мс;
  - оптимізація - впровадження методів зниження розмірності даних та використання ефективних алгоритмів кешування;
- 3) адаптивність:
  - ціль - модель повинна адаптуватися до змін у поведінці користувачів з часом, з використанням інкрементного навчання;
  - дані - автоматичне включення нових даних про користувацьку взаємодію в процес навчання моделі;
- 4) масштабованість:
  - ціль - система має підтримувати збільшення кількості користувачів та обсягів даних у 10 разів без значної втрати продуктивності;
  - реалізація - використання розподілених обчислень та ефективних структур даних.

В свою чергу, конкретизація нефункціональних вимог виглядає так:

- 1) продуктивність:
  - оцінка - вимірювання часу відповіді системи під час пікового навантаження;
  - оптимізація - паралелізація обчислень і використання швидкодіючих баз даних;
- 2) безпека:
  - ціль - забезпечення захисту даних користувачів від несанкціонованого доступу;

- заходи - впровадження шифрування даних, аутентифікації та контролю доступу;
- 3) інтеграція:
  - ціль - забезпечення сумісності з існуючими веб-платформами та базами даних;
  - методи - розробка API для легкої інтеграції з іншими системами;
- 4) інтерпретованість:
  - ціль - забезпечення можливості пояснення рекомендацій, що генеруються системою;
  - реалізація - використання моделей з можливістю візуалізації вагових коефіцієнтів та впливу вхідних даних на результат.

Відповідно до цього детальний аналіз вимог є фундаментальним кроком у проектуванні алгоритму машинного навчання, що забезпечує не тільки високу ефективність рекомендаційної системи, але й відповідність її роботи очікуванням користувачів та бізнес-цілям. Врахування цих вимог на етапі проектування дозволяє створити адаптивну, масштабовану та безпечну систему, спроможну забезпечити високу якість сервісу та користувацького досвіду.

### 3.3 Формування структури алгоритму

Відповідно до проведеного аналізу вимог було обрано використання нейромережі як базис для алгоритму машинного навчання. Нейромережі мають високу здатність до навчання та адаптації, що дозволяє їм ефективно обробляти великі обсяги даних та виявляти складні залежності між ними. Це робить нейромережі ідеальним вибором для задач рекомендаційних систем, які потребують точних прогнозів на основі поведінкових даних користувачів.

Разом із використанням нейромережі залишається необхідність у ретельній обробці даних користувачів. Дані повинні бути очищені, нормалізовані та представлені у вигляді, придатному для навчання нейромережі. Відповідно до потреб комерційних систем рекомендацій, важливо мати результати та висновки,

що базуються на реальному досвіді. Для цього необхідно враховувати аналогію з колаборативною фільтрацією, яка показала свою ефективність у рекомендаційних системах [19].

Колаборативна фільтрація передбачає використання даних про взаємодію користувачів з товарами чи послугами для створення рекомендацій. Одним із ключових аспектів цього методу є класифікація клієнтів та послуг, що дозволяє краще розуміти їхні уподобання та поведінку. Таким чином, для побудови ефективної системи рекомендацій було вирішено застосовувати метод К-найближчих сусідів (K-Nearest Neighbors, KNN) для створення класифікацій покупців і окремо класифікацій послуг [20].

Метод KNN працює на основі пошуку найближчих сусідів для кожного об'єкта у багатовимірному просторі. Це дозволяє виявити клієнтів та послуги, які мають найбільшу схожість між собою за певними критеріями. Для кожного нового клієнта або послуги можна знайти їх найближчих сусідів серед існуючих даних і таким чином визначити класифікацію. Потім ці дані додатково аналізуються для визначення пріоритету між класами клієнтів по відношенню до класів послуг. Це здійснюється через аналіз вибору клієнтами певних послуг, що дозволяє відобразити відповідність між попитом і пропозицією [18].

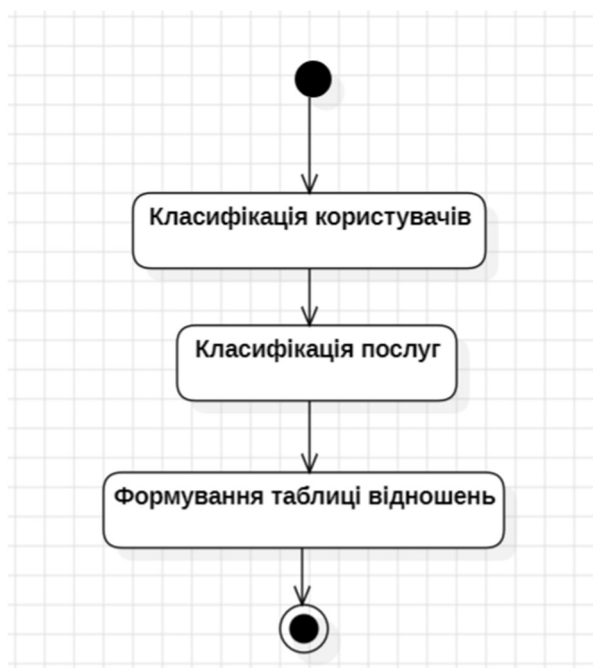


Рисунок 3.3 – Початкова класифікація користувачів та послуг

Розрахунок пріоритетів, представлений у вигляді ранжування класів товарів для класів покупців, забезпечує можливість оцінки відношення якість/ціна для окремого користувача. Це особливо важливо для комерційних систем, де цінність пропонованих рекомендацій визначається їхньою релевантністю та корисністю для клієнтів. Потім ці результати спрощуються та нормалізуються, представлені у таблицях даних кластерів покупців, послуг та пріоритетів. Ці таблиці забезпечують готові дані для роботи з моделлю нейромережі [21].

Дані про існуючих клієнтів та послуги передаються в нейромережу для визначення зв'язку між ними та моделювання поведінки клієнтів. Нейромережа, навчаючись на цих даних, може ефективно прогнозувати взаємодії клієнтів з новими послугами чи товарами. Це забезпечує можливість створення списків рекомендованих товарів чи послуг для клієнтів. На основі дій клієнтів (наприклад, покупка, ігнорування або відвідування), які можуть бути відстежені засобами веб-аналітики, модель нейромережі може зберігати актуальні зв'язки між клієнтами та товарами або перевизначати категорії клієнтів для надання нових рекомендацій.



Рисунок 3.4 – Принцип надання рекомендацій нейронною мережею

Таким чином, система рекомендацій стає динамічною та здатною адаптуватися до змін у поведінці клієнтів. З часом модель стає автономною від початкових таблиць даних і може накопичувати інформацію про клієнтів і товари самостійно. Це дозволяє підвищувати рівень оновлення класифікацій за допомогою методу KNN та ранжування таблиць відносин між класами клієнтів і послуг, що підвищує ефективність моделі і значущість показника конверсії [22].

У підсумку, обраний підхід, що базується на поєднанні нейромережі та методу KNN, забезпечує високу точність і адаптивність рекомендаційної системи. Це дозволяє створювати персоналізовані рекомендації, які максимально відповідають уподобанням та потребам клієнтів, що є ключовим фактором для підвищення їхньої задоволеності та збільшення конверсії у комерційних системах.

### 3.4 Деталізація компонентів алгоритму

На основі визначеної структури алгоритму, спрямованого на аналіз даних для рекомендаційної системи в сфері онлайн-бронювання, перейдемо до детального опису ключових компонентів. Кожен компонент відіграє важливу роль у загальній роботі системи, забезпечуючи ефективне перетворення вхідних даних у релевантні рекомендації.

В першу чергу, варто описати модуль передобробки даних. Цей модуль необхідний для приведення початкових даних до визначеного формату, так як від цього залежить правильність роботи зазначеного алгоритму класифікації а також точність його функціонування. Також важливо відмітити, що реальні дані рідко бувають вже у готовому для використання вигляді, тому без цього модуля функціонування заданого алгоритму не може бути гарантоване.

Модуль передобробки даних цього алгоритму включає:

- очищення даних;
- нормалізацію;
- лематизацію тексту;
- векторизацію тексту.

У наведеному модулі передоброби, очищення даних передбачає видалення пропущених або нерелевантних елементів, що можуть негативно вплинути на якість навчання моделі. Нормалізація, в свою чергу, здійснюється для приведення даних до єдиного масштабу, включаючи перетворення тексту в нижній регістр і видалення стоп-слів. Лематизація ж тексту зводить слова до їх базової форми, що допомагає зменшити кількість унікальних слів і покращує узгодженість даних. Векторизація тексту перетворює текстові дані у числовий формат за допомогою таких методів, як TfidfVectorizer або Word2Vec, що дозволяє подавати їх на вхід моделі машинного навчання. Використання NLP-бібліотек, таких як NLTK або SpaCy, дозволяє ефективно виконувати ці завдання.

Наступним модулем наведеного алгоритму є модуль аналізу даних. Цей модуль відповідає на початкову класифікацію користувачів та послуг. Від цього модуля залежить як в подальшому будуть визначатися рекомендації для користувача. Цей модуль включає:

- використання K-найближчих сусідів (KNN) для класифікації користувачів і послуг;
- побудову таблиці відношень користувач-послуга з ранжуванням.

Метод KNN використовується для групування схожих користувачів та послуг. Це дозволяє створити класи користувачів і класи послуг, які потім використовуються для побудови таблиці відношень між ними. Таблиця відношень користувач-послуга містить ранжування, яке показує, який клас послуги найбільше підходить для кожного класу користувача. Цей підхід допомагає зрозуміти взаємодію між різними категоріями користувачів і послуг, забезпечуючи основу для подальшої генерації рекомендацій.

Модуль генерації рекомендацій є основним модулем алгоритму. Він відповідає за наданням користувачам релевантних рекомендацій, а також забезпечує оновлення, за умови, коли підібрані рекомендації виявилися хибними.

Модуль генерації рекомендацій включає:

- визначення класів користувачів;
- визначення класів послуг;

- надання рекомендацій на основі таблиці відношень.

Цей модуль базується на раніше побудованій таблиці відношень користувач-послуга. Ранжування в таблиці визначає пріоритети для кожного класу користувачів щодо відповідних класів послуг. На основі цієї таблиці модуль генерації рекомендацій визначає, які послуги найбільш підходять конкретному користувачу, забезпечуючи персоналізовані рекомендації. Використання нейронних мереж дозволяє враховувати складні взаємозв'язки між користувачами та їхніми вподобаннями, що підвищує релевантність рекомендацій.

Останнім модулем проєктованого алгоритму є модуль валідації та тестування. Цей модуль не менш важливий ніж інші, так як він відповідає за перевірку правильності роботи визначеного алгоритму та забезпечує перевірку релевантності наданих рекомендацій.

Модуль валідації та тестування включає:

- Крос-валідацію;
- А/В тестування;

Крос-валідація використовується для оцінки загальної продуктивності моделі, забезпечуючи надійні результати через багаторазове розбиття даних на тренувальні та тестові набори. А/В тестування дозволяє порівнювати різні підходи і налаштування моделі для визначення найефективніших рішень. Важливо використовувати метрики, такі як точність (precision), відгук (recall) та F1-міра, для кількісної оцінки результатів моделі. Це допомагає визначити, наскільки добре модель справляється зі своїми задачами і де можна провести оптимізацію.

### 3.5 Планування процесів тестування та валідації

Тестування та валідація є невід'ємними складовими процесу розробки алгоритму машинного навчання, дозволяючи оцінити його ефективність, надійність та загальну придатність до використання в рекомендаційній системі. Цей процес включає застосування ряду стратегій та методів для всебічного аналізу моделі (рис. 3.5). Процеси тестування та валідації відіграють вирішальну роль у

розробці надійних та ефективних алгоритмів машинного навчання для рекомендаційних систем. Вони дозволяють не тільки оцінити загальну продуктивність моделі, але й ідентифікувати потенційні напрямки для її удосконалення, забезпечуючи високу точність та релевантність рекомендацій.

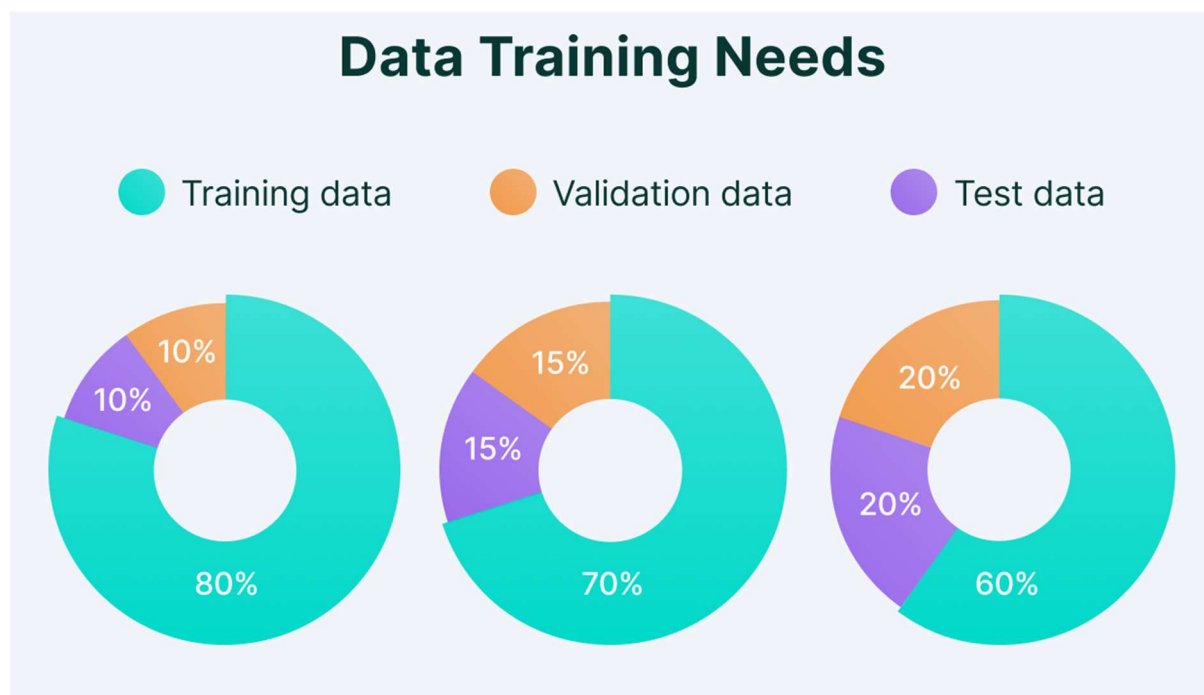


Рисунок 3.5 - Застосування тестування для валідації нейромерж

Для даного алгоритму передбачається така стратегія тестування, що складається з:

- Розділення даних (Splitting);
- Крос-валідація.

Розділення даних (рис. 3.6) виконується з метою є забезпечення об'єктивної оцінки моделі шляхом використання незалежних наборів даних для навчання, валідації та тестування. Цей процес допомагає уникнути перенавчання моделі та забезпечує більш точне вимірювання її продуктивності. Реалізація розділення даних передбачає розподіл загального датасету на три частини: навчальний, валідаційний та тестовий набори. Зазвичай, навчальний набір складає 70-80% від загального обсягу даних і використовується для навчання моделі, тобто для налаштування її параметрів. Валідаційний набір, що становить 10-15% від загального датасету, використовується для проміжної оцінки моделі під час

навчання і допомагає налаштувати гіперпараметри моделі, щоб уникнути перенавчання. Нарешті, тестовий набір, який також складає 10-15% від загального обсягу даних, використовується для остаточної оцінки моделі після завершення навчання. Це дозволяє об'єктивно оцінити здатність моделі до узагальнення на нових, невідомих даних, що є критичним для забезпечення її ефективної роботи в реальних умовах.

### Train, Test and Validation Split



Рисунок 3.6 - Типова схема розділеної валідації

Крос-валідація (рис. 3.7) спрямована на зниження впливу випадковості у розподілі даних на оцінку моделі та забезпечення більш стабільної оцінки її продуктивності. Цей метод допомагає уникнути ситуацій, коли модель демонструє хороші результати лише через сприятливий розподіл даних, а не через свою реальну здатність до узагальнення. Основним методом крос-валідації є  $k$ -кратна крос-валідація. У цьому методі набір даних розділяється на  $k$  однакових частин. Модель тестується  $k$  разів на різних комбінаціях цих частин, кожного разу використовуючи одну частину для тестування, а інші  $k-1$  частини для навчання. Таким чином, кожна частина даних використовується як для навчання, так і для тестування, що забезпечує більш надійну та стабільну оцінку продуктивності моделі. Це дозволяє отримати середнє значення результатів усіх  $k$  випробувань, що знижує вплив випадкових факторів і надає більш точне уявлення про ефективність моделі.

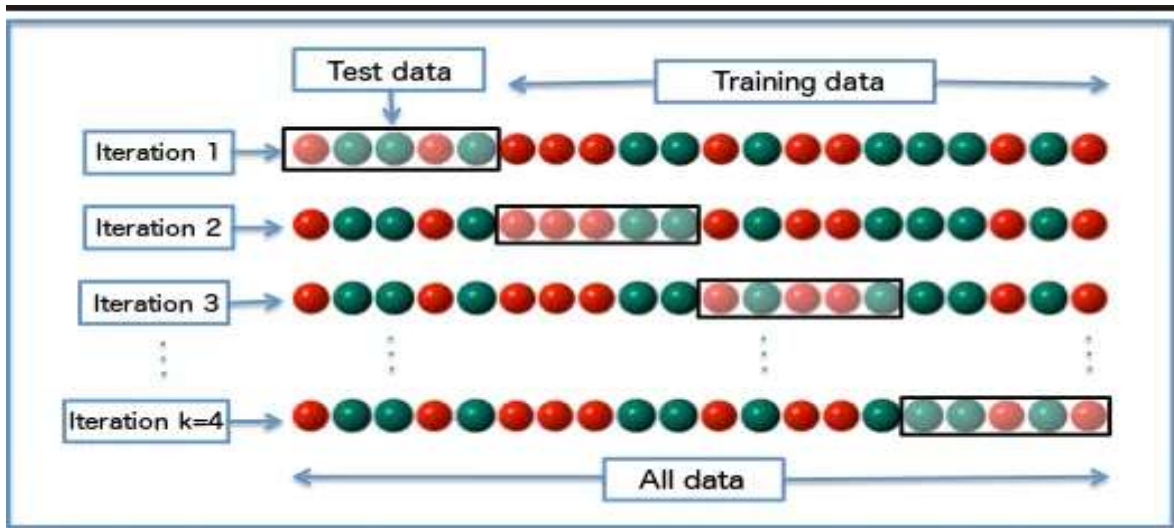


Рисунок 3.7 - Типова схема крос-валідації

Методи валідації базуються на наступних метриках як точність (Accuracy), F1-міра, Precision і Recall для оцінки ефективності класифікації моделі. А також аналіз помилково позитивних та помилково негативних результатів для уточнення чутливості моделі. Метою валідації є ідентифікація типових помилок моделі для подальшого їх усунення та оптимізації.

Виходячи з вище сказаного, важливо провести детальний аналіз випадків, де модель демонструє низьку продуктивність, для виявлення можливих причин (наприклад, недостатнє представлення даних, перенавчання).

Регулярний моніторинг показників ефективності моделі в реальному часі для виявлення зниження продуктивності або потреби в додаткового налаштуванні. Впровадження механізмів для автоматичного оновлення моделі на основі нових даних або змін у поведінці користувачів.

### 3.7 Аналіз потенційних проблем

Розробка алгоритму машинного навчання для рекомендаційних систем у сфері онлайн-бронювання стикається з рядом потенційних проблем, які можуть вплинути на ефективність, точність та надійність моделі. Розуміння та аналіз цих проблем є ключовим для розробки стратегій їх вирішення або мінімізації їхнього впливу. Важливо відмітити, що такі проблеми часто характерні для методів в

цілому, тому важливо виділити саме ті з якими може зустріти під час роботи з нейромережею:

1) Перенавчання (Overfitting):

Проблема: Модель занадто добре адаптується до навчального датасету, втрачаючи здатність узагальнювати на нових даних.

Рішення: Використання методів регуляризації (наприклад, L1, L2, dropout), крос-валідації та ранньої зупинки для запобігання перенавчанню.

2) Недонавчання (Underfitting):

Проблема: Модель недостатньо добре навчена для виявлення закономірностей у датасеті, що призводить до низької продуктивності.

Рішення: Збільшення складності моделі, оптимізація гіперпараметрів, збагачення набору даних або використання більш потужних алгоритмів машинного навчання.

3) Зміщеність даних (Data Bias):

Проблема: Наявність упереджень у навчальному датасеті може призвести до зміщених або несправедливих рекомендацій.

Рішення: Ретельний аналіз та очищення даних, використання технік збалансування датасетів, а також розробка алгоритмів для ідентифікації та корекції зміщеності.

4) Шум у даних (Noise in Data):

Проблема: Наявність невідповідностей та помилок у даних може знижувати точність моделі.

Рішення: Використання технік передобробки даних для виявлення та усунення шуму, таких як фільтрація, нормалізація та очищення даних.

5) Масштабування даних (Scalability Issues):

Проблема: Зі зростанням обсягів даних, модель може стикатися з проблемами продуктивності та масштабованості.

Рішення: Використання ефективних алгоритмів та технологій розподілених обчислень, оптимізація архітектури моделі для забезпечення високої продуктивності при масштабуванні.

б) Проблеми інтерпретації:

Проблема: Складність пояснення рішень, прийнятих моделлю, особливо в глибоких нейронних мережах.

Рішення: Використання методів інтерпретації машинного навчання, розробка більш прозорих моделей або інтеграція пояснювальних модулів.

Виявлення та аналіз потенційних проблем, з якими може зіткнутися алгоритм машинного навчання, є критичним етапом в процесі розробки рекомендаційної системи. Вирішення цих проблем передбачає комплексний підхід, який включає як технічні заходи, так і методологічні корективи, забезпечуючи створення ефективної, надійної та справедливої моделі.

## 4 РОЗРОБКА АЛГОРИТМУ МАШИННОГО НАВЧАННЯ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ОНЛАЙН-БРОНЮВАННЯ

### 4.1 Вибір моделі та засобів

Для задачі рекомендаційної системи в сфері онлайн-бронювання, відповідно до проведеного проектування алгоритму, обрано методи машинного навчання з учителем. Цей вибір зумовлений тим, що дані, які ми маємо, чітко структуровані та містять явні зразки взаємодії користувачів із системою, а саме оцінки, перегляди, що дозволяє застосувати навчання на основі попередньо відомих результатів. Ціллю є навчити модель передбачати користувацькі вподобання на основі вхідних даних, що сприятиме формуванню точних рекомендацій.

Алгоритм будується на основі компоненту класифікації вхідних даних через метод KNN та компоненту надання рекомендацій з використанням нейронної мережі. Тому для цієї задачі вирішено застосовувати мову програмування Python та бібліотеки Pandas, scikit-learn а також Keras.

По-перше, розглянемо причини вибору мови програмування Python. Python - інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією. Для аналізу даних, інтерактивних обчислень та візуалізації даних Python неминуче порівнюють з іншими мовами програмування з відкритим вихідним кодом та комерційними інструментами, такими як R, MATLAB, SAS, Stata та іншими, що широко використовуються. В останні роки вдосконалені бібліотеки з відкритим вихідним кодом (такі як pandas та scikit-learn) зробили Python популярним вибором для задач аналізу даних. У поєднанні з загальною силою Python для розробки програмного забезпечення загального призначення, вона є чудовим варіантом як основна мова для створення додатків для роботи з даними [23].

Pandas - це швидкий, потужний, гнучкий і простий у використанні інструмент аналізу та маніпулювання даними з відкритим вихідним кодом, створений на основі мови програмування Python.

Pandas надає високорівневі структури даних і функції, призначені для того, щоб зробити роботу зі структурованими або табличними даними інтуїтивно зрозумілою і гнучкою. З моменту своєї появи у 2010 році вона допомогла зробити Python потужним і продуктивним середовищем для аналізу даних. Основними об'єктами, які будуть використовуватися, є DataFrame, таблична структура даних, орієнтована на стовпці, з мітками для рядків і стовпців, і Series, одновимірний об'єкт масиву з мітками [23].

Також в розробці моделі важливе місце займає бібліотека scikit-learn. Scikit-learn - модуль Python для машинного навчання, побудований на основі SciPy і розповсюджується за ліцензією 3-Clause BSD. З моменту заснування проекту в 2007 році scikit-learn став провідним універсальним інструментарієм машинного навчання для програмістів на Python [23]. Він включає в себе підмодулі для таких моделей як:

- Класифікація: SVM, найближчих сусідів, випадкового лісу, логістичної регресії тощо.
- Регресія: Лассо, гребенева регресія тощо.
- Кластеризація: k-середні, спектральна кластеризація тощо.
- Зменшення розмірності: PCA, виділення ознак, матрична факторизація тощо.
- Вибір моделі: Пошук по сітці, перехресна перевірка, метрики
- Попередня обробка: Виділення ознак, нормалізація

Остання бібліотека, що застосовується в розробці моделі це Keras. Keras — відкрита нейромережна бібліотека, написана мовою Python. Вона здатна працювати поверх TensorFlow, Microsoft Cognitive Toolkit, R, Theano та PlaidML. Він надає доступний, високопродуктивний інтерфейс для вирішення завдань машинного навчання (ML), з акцентом на сучасне глибоке навчання. Keras охоплює кожен крок робочого процесу машинного навчання, від обробки даних до налаштування гіперпараметрів і розгортання. Він був розроблений з акцентом на швидке експериментування [24].

З Keras ви маєте повний доступ до масштабованості та крос-платформних можливостей TensorFlow. Ви можете запускати Keras на TPU Pod або великих кластерах графічних процесорів, а також експортувати моделі Keras для запуску в браузері або на мобільних пристроях. Ви також можете обслуговувати моделі Keras через веб-аплікатор.

## 4.2 Збір та обробка даних

Для підвищення ефективності рекомендаційної системи онлайн-бронювання основним завданням є збір та аналіз великої кількості даних про користувачів та їхні взаємодії з різними послугами. Джерела даних включають:

- Журнали веб-серверів: відомості про всі запити та відповіді сервера, що дозволяє аналізувати дії користувача на платформі.
- Бази даних користувачів: інформація про реєстрацію, відгуки та оцінки користувачів.
- Історія бронювань: деталізовані дані про попередні бронювання, включно з типами обраних послуг, локацією, вартістю і т. д.

Для обробки ж даних застосовуються наступні етапи:

- 1) Очищення даних: Видалення помилкових, неповних або несуттєвих записів, які можуть спотворити результати аналізу.
- 2) Нормалізація даних: Приведення всіх даних до єдиного масштабу, що забезпечує порівняння та уніфікацію вхідних параметрів для моделі.
- 3) Кодування категоріальних змінних: Перетворення категоріальних даних (наприклад, тип послуги) в числовий формат за допомогою технік One-Hot Encoding або Label Encoding, щоб можливо було включити ці дані в модель машинного навчання.

Крім того, для забезпечення конфіденційності та безпеки користувачів виконується анонімізація персональних даних перед їх обробкою. Використання таких підходів дозволяє підвищити точність та персоналізацію рекомендацій, що значно поліпшує користувацький досвід і задоволеність сервісом.

Для експериментального дослідження функціональності розроблюваного алгоритму було обрано набори даних з репозиторію Kaggle. Kaggle — платформа для змагань з аналітики та передбачувального моделювання, в рамках якого статистики та добувачі даних конкурують у створенні найкращих моделі для прогнозування та опису даних, запропонованих компаніями або користувачами. Так як це платформа, що створена для змагань з аналітики даних, то вона містить об'ємну базу різних датасетів, в тому числі в сфері онлайн-бронювання.

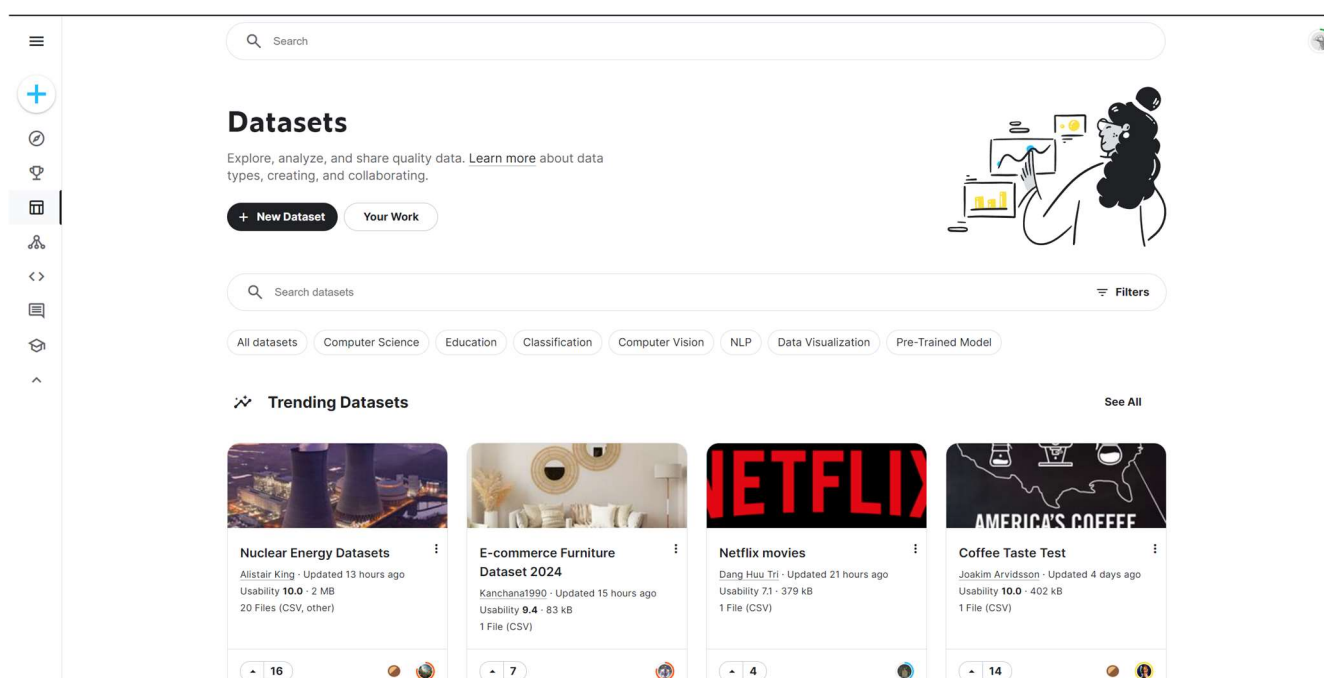


Рисунок 4.1 – Платформа для аналізу даних Kaggle

За основу, для проведення навчання, обрано датасети про клієнтів авіакомпанії та їх перельоти, а також про перельоти, резервування готелів. Ці дані були адаптовані, щоб імітувати дані рекомендаційної системи для онлайн-бронювання.

### 4.3 Розробка моделі

Для реалізації рекомендаційної системи в сфері онлайн-бронювання вирішено використати нейронну мережу, оскільки вона забезпечує гнучкість у

моделюванні складних неявних взаємозв'язків між характеристиками користувачів та предметами [25].

Для надання рекомендацій використовується нейронна мережа відповідно до спроектованого алгоритму. Топологія цієї нейромережі складається з:

- Вхідний шар - розмір входу відповідає кількості особливостей (наприклад, 30 різноманітних характеристик користувача та продукту).
- Приховані шари - дві приховані шари з 64 та 32 нейронами відповідно.
- Функція активації - ReLU для прихованих шарів, що забезпечує нелінійність і допомагає уникнути проблеми зниклих градієнтів.
- Вихідний шар - один нейрон із сигмоїдальною активаційною функцією, щоб вивести ймовірність вибору користувачем даної послуги.

Всі вхідні характеристики нормалізовано для поліпшення швидкості навчання та стабільності конвергенції. Категоріальні змінні перетворено за допомогою one-hot encoding.

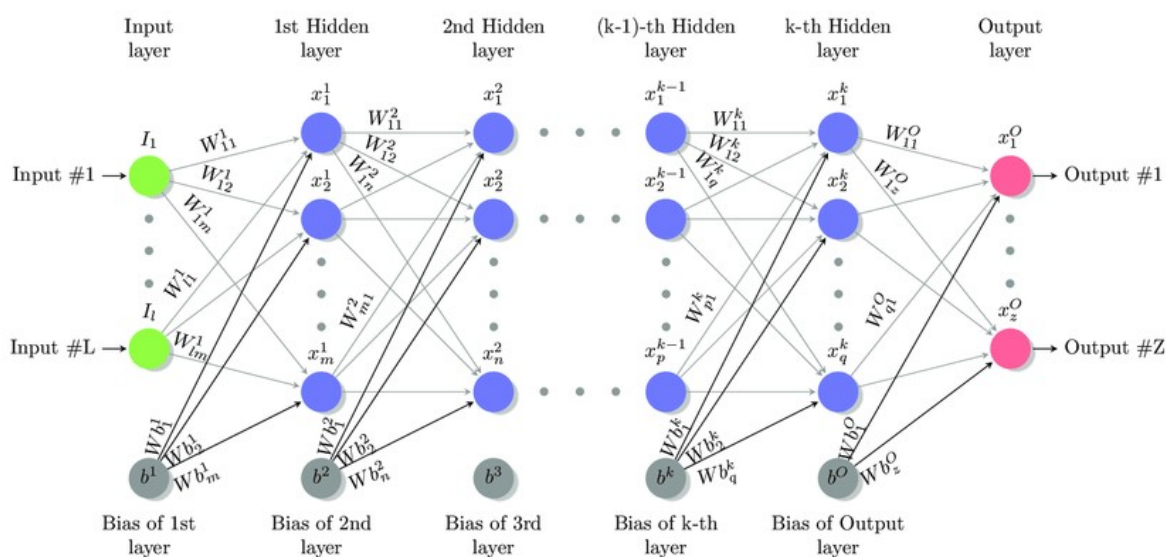


Рисунок 4.2 - Топологія нейромережі

Для навчання розробленої нейромережі визначені такі налаштування:

- Втрати: Використання бінарної крос-ентропії як функції втрат для оцінки різниці між передбаченнями та реальними мітками.

- Оптимізатор: Adam зі швидкістю навчання 0.001 для ефективної оптимізації.

- Кількість епох: 100 з ранньою зупинкою, якщо помилка на валідаційному наборі не зменшується протягом 10 послідовних епох.

Застосування такого підходу дозволяє детально оцінити потенціал моделі у вирішенні реальних завдань рекомендаційної системи.

#### 4.4 Навчання моделі

Процес навчання моделі глибокого навчання для рекомендаційної системи в сфері онлайн-бронювання включає декілька ключових кроків, що забезпечують ефективно та точно моделювання. Навчання моделі проводиться на основі заздалегідь підготовленого і розділеного набору даних, який включає тренувальні, валідаційні та тестові датасети.

При підготовці до навчання, дані для експерименту розділяються на тренувальний (70% даних), валідаційний (15% даних) та тестовий (15% даних) набори. Таке розділення дозволяє не тільки ефективно навчати модель, але й оцінювати її загальну здатність узагальнювати на нових даних, що не були використані під час навчання.

Перед навчанням модель має бути правильно ініціалізована. Використовуються випадкові значення вагових коефіцієнтів нейронів, що базуються на стандартному або зміщеному нормальному розподілі, що дозволяє забезпечити адекватний старт навчання.

Переходимо до навчання моделі. Навчання відбувається таким чином:

- 1) Дані подаються на вхід моделі, де вони проходять через шари нейронів, в кожному з яких відбувається зважування входів, додавання зміщення та застосування функції активації.

- 2) Після того, як вхідні дані проходять через модель, обчислюється відхилення отриманих результатів від фактичних міток даних за допомогою функції втрат, наприклад, крос-ентропії.

3) Використовуючи алгоритм зворотного поширення помилки, обчислюється градієнт функції втрат по кожному з параметрів моделі (вагам і зміщенням) для оновлення цих параметрів у напрямку зменшення помилки.

Для налаштування адаптації швидкості навчання використовується оптимізатор Adam, який адаптує швидкість навчання для кожного параметра окремо на основі оцінок перших і других моментів градієнтів.

Розроблена модель навчається протягом декількох епох (наприклад, 50), де одна епоха включає один прохід через весь тренувальний набір даних. Дані подаються в модель пакетами (наприклад, по 32 приклади), що дозволяє ефективно використовувати обчислювальні ресурси.

Після кожної епохи проводиться оцінка моделі на валідаційному наборі. Це дозволяє моніторити перенавчання та регулювати гіперпараметри, такі як швидкість навчання або число епох, в залежності від динаміки зміни точності та функції втрат на валідаційних даних.

Цей покроковий опис процесу навчання відображає комплексний підхід до розробки та тренування нейронної мережі, який спрямований на створення високоефективної рекомендаційної системи для онлайн-бронювання.

#### 4.5 Валідація моделі

##### Методики тестування

Для забезпечення надійності та ефективності рекомендаційної системи важливо провести ґрунтовне тестування алгоритму. Використовуємо два основні методи:

- Перехресна перевірка (Cross-validation);
- A/B тестування.

Перехресна перевірка - поділ тренувального набору даних на  $k$  частин (фолдів). В кожній ітерації один фолд використовується як тестовий набір, а решта - як тренувальні дані. Цей процес повторюється  $k$  разів, з кожним фолдом

використанам як тестовий один раз. Такий підхід дозволяє оцінити модель на різних вибірках і зменшити вплив випадковості в розподілі даних.

A/B тестування - впровадження алгоритму в реальній системі з частиною користувачів, які отримують рекомендації за допомогою нової моделі (група А), інша частина - зі старою версією (група В). Порівняння результатів обох груп дає змогу оцінити ефективність нової моделі у реальних умовах.

Для оцінки ефективності моделі застосовуються наступні метрики:

- Точність (Accuracy) - відсоток випадків, коли модель правильно передбачила рекомендацію.
- Відгук (Recall) - відношення правильно ідентифікованих позитивних результатів до загальної кількості реальних позитивних результатів.
- F1-міра - гармонійне середнє між точністю і відгуком, що дозволяє оцінити баланс між ними.
- ROC-AUC - площа під кривою характеристик роботи приймача (ROC curve), що відображає залежність між чутливістю та специфічністю на всіх можливих порогах класифікації.

Використання таких методів тестування допомагає гарантувати, що модель буде ефективною та надійною у реальних умовах експлуатації.

## 4.6 Аналіз результатів

### 4.6.1 Оцінка ефективності

Після завершення процесу навчання та тестування моделі, результати оцінювалися за допомогою визначених метрик, щоб зрозуміти, наскільки ефективно модель може рекомендувати продукти користувачам. Результати були представлені як середні значення виходячи з декількох ітерацій перехресної перевірки. Ось основні показники, які були виміряні:

Точність (Accuracy): 92%

Відгук (Recall): 82%

F1-міра: 83%

ROC-AUC: 0.9

Інтерпретація

Точність 92% вказує на те, що модель добре впоралася з задачею класифікації, правильно ідентифікуючи більшість рекомендацій. Відгук 82% свідчить про те, що модель змогла ідентифікувати більшу частину релевантних взаємодій. F1-міра, яка є балансом між точністю та відгуком, також показала хороший результат, що робить модель збалансованою в контексті її здатності до класифікації.

Аналіз причин успіху або невдачі моделі

Основною причиною успіху моделі можна вважати ефективний вибір архітектури нейронної мережі та вдале використання технік оптимізації та регуляризації. Однак, слід відзначити декілька потенційних напрямків для покращення:

Збільшення датасету: Модель може показати кращі результати, якщо буде навчена на більшій та більш різноманітній кількості даних.

Гіперпараметри: Додаткове налаштування гіперпараметрів може допомогти покращити точність і відгук.

Уникнення перенавчання: Подальше використання методів регуляризації та розширення валідаційних методик може допомогти зменшити перенавчання моделі.

Це дозволяє зрозуміти її сильні та слабкі сторони у контексті задачі рекомендації.

#### 4.6.2 Вплив на бізнес-процеси

Розроблений алгоритм машинного навчання для рекомендаційної системи в сфері онлайн-бронювання має значний потенціал для покращення бізнес-процесів, особливо в контексті збільшення задоволеності клієнтів та оптимізації продажів. Алгоритм забезпечує більш персоналізовані рекомендації на основі поведінкових даних та переваг користувачів, що сприяє зростанню лояльності клієнтів та

підвищує ймовірність бронювання, оскільки клієнти бачать пропозиції, які відповідають їхнім бажанням та потребам. Крім того, алгоритм допомагає підприємствам краще розуміти попит на певні послуги, дозволяючи оптимізувати доступність та розподіл ресурсів.

Для успішного впровадження алгоритму в реальні бізнес-сценарії слід інтегрувати його з існуючими системами управління відносинами з клієнтами (CRM) та системами управління контентом (CMS), щоб забезпечити плавне введення в експлуатацію та мінімізувати потребу в значних модифікаціях систем.

Також рекомендується спочатку запуснути пілотний проект для оцінки ефективності алгоритму на обмеженій кількості користувачів або продуктів, перш ніж здійснювати повномасштабне впровадження. Після впровадження необхідно встановити процедури для регулярного моніторингу продуктивності алгоритму та вживати заходів для оптимізації на основі отриманих даних, що може включати тонке налаштування параметрів алгоритму, оновлення навчальних даних або реагування на зміну ринкових умов. Такі заходи допоможуть забезпечити ефективне впровадження алгоритму, сприяючи досягненню бізнес-цілей та покращенню процесів онлайн-бронювання.

#### 4.6.3 Подальші дослідження

Хоча розроблена модель показала високі результати в тестуванні, існує кілька напрямків для її подальшого вдосконалення. Збільшення обсягу і різноманітності датасету може покращити здатність моделі до узагальнення і зменшити перенавчання, включаючи збір даних з різних географічних регіонів та культур для адаптації рекомендацій до глобальних користувачів. Експериментування з архітектурою мережі, зокрема використання конволюційних нейронних мереж (CNN) для обробки візуального контенту або рекурентних нейронних мереж (RNN) для аналізу послідовностей даних, може покращити розуміння контексту користувача. Застосування технік зниження розмірності, таких як PCA (головні компоненти) або t-SNE, для візуалізації та аналізу високовимірних даних допоможе ідентифікувати ключові фактори, що впливають

на рекомендації, і знайти нові напрямки для оптимізації моделі. Розширене тестування та валідація, включаючи стрес-тестування моделі за допомогою синтетично згенерованих аномалій, може виявити потенційні слабкі місця і сприяти розробці механізмів їх усунення.

Щодо подальших напрямків досліджень, можна виділити інтеграцію рекомендаційної системи з іншими бізнес-системами, такими як ERP або CRM, що дозволить точніше прогнозувати попит і краще управляти ресурсами. Дослідження методів машинного навчання без учителя, таких як кластеризація або генеративні змагальні мережі (GAN), для генерації нових користувацьких профілів або розширення існуючих датасетів, також є перспективним напрямком. Використання технологій штучного інтелекту для аналізу емоційних реакцій користувачів на рекомендації може допомогти зрозуміти, як емоційне сприйняття впливає на їхні рішення і як це можна використати для покращення точності рекомендацій. Ці напрямки дозволять не тільки покращити існуючу систему, але й розширити можливості її застосування, забезпечивши більшу адаптивність і точність у відповідності до змінних умов ринку та поведінки споживачів.

## ВИСНОВКИ

У процесі роботи проведено глибоке дослідження алгоритму машинного навчання для рекомендаційної системи в сфері онлайн-бронювання, з особливим акцентом на аналізі текстових даних. Ця робота охоплює критично важливі аспекти, починаючи з огляду існуючих рішень, аналізу вимог, визначення архітектури алгоритму, деталізації його компонентів, стратегій навчання та оптимізації, планування процесів тестування та валідації, а завершується аналізом потенційних проблем.

Основною метою досліджуваного алгоритму було створення високоефективної системи, спроможної аналізувати великі обсяги текстових даних - відгуки користувачів, описи об'єктів бронювання, цінові параметри - для генерації точних і персоналізованих рекомендацій. Робота акцентувала увагу на важливості використання нейронних мереж, зокрема рекурентних нейронних мереж і механізмів уваги, які демонструють виняткову здатність до обробки і аналізу послідовностей тексту.

Під час розробки було визначено ключові вимоги до алгоритму, включаючи необхідність забезпечення високої точності рекомендацій, швидкості відгуку системи, адаптивності до змінних уподобань користувачів, а також масштабованості системи. Врахування цих вимог дозволило забезпечити комплексний підхід до проектування алгоритму та його компонентів, а також ефективно спланувати процеси навчання, тестування та валідації.

Аналіз потенційних проблем, таких як перенавчання, недонавчання, зміщеність даних, шум у даних, масштабування даних та проблеми інтерпретації, підкреслив важливість ретельного планування та використання передових методів машинного навчання для подолання зазначених викликів. Вирішення цих проблем вимагало застосування комплексного підходу, що включає як технічні рішення, так і методологічні корективи, з метою створення більш ефективної, надійної та справедливої моделі.

У підсумку, ця робота не лише демонструє потенціал машинного навчання у розробці рекомендаційних систем для онлайн-бронювання, але й підкреслює значення всебічного аналізу та планування на всіх етапах розробки алгоритму. Результати дослідження вказують на широкі перспективи застосування глибокого навчання та нейронних мереж у цій галузі, відкриваючи нові можливості для покращення якості та точності рекомендаційних систем.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Швець Л., Редько Н., Дудка Т. СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ СФЕРИ ПОСЛУГ. Економіка та суспільство. 2021. № 34. URL: <https://doi.org/10.32782/2524-0072/2021-34-57> (дата звернення: 10.04.2024).
2. Чуєва І., Жестков С., Сидорук А. СУЧАСНІ ТЕНДЕНЦІЇ РОЗВИТКУ ОНЛАЙН БРОНЮВАННЯ ТУРИСТИЧНИХ ПОСЛУГ В УКРАЇНІ. Економіка та суспільство. 2021. № 27. URL: <https://doi.org/10.32782/2524-0072/2021-27-11> (дата звернення: 16.04.2024).
3. Meleshko Y. ПРОБЛЕМИ СУЧАСНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ТА МЕТОДИ ЇХ РІШЕННЯ. Системи управління, навігації та зв'язку. Збірник наукових праць. 2018. Т. 4, № 50. С. 120–124. URL: <https://doi.org/10.26906/sunz.2018.4.120> (дата звернення: 01.05.2024).
4. Amazon Product Recommendation System: How Does Amazon's Algorithm Work? - Stratoflow. Stratoflow. URL: <https://stratoflow.com/amazon-recommendation-system/> (дата звернення: 01.05.2024).
5. Sarker I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science. 2021. Vol. 2, no. 3. URL: <https://doi.org/10.1007/s42979-021-00592-x> (дата звернення: 16.04.2024).
6. Машинне навчання: всебічний погляд на можливості і проблеми штучного інтелекту - BIZMAG. BIZMAG. URL: <https://bizmag.com.ua/mashynne-navchannya/> (дата звернення: 05.05.2024).
7. Machine Learning, ML. IT-Enterprise – your one-stop platform for digital transformation | www.it.ua. URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning> (дата звернення: 15.05.2024).
8. Тренди машинного навчання в індустріях та технологіях | Wezom. IT-компанія повного циклу розробки програмних продуктів WEZOM - Київ, Україна. URL: <https://wezom.com.ua/ua/blog/maybutnje-mashinnogo-navchannya-industriyi-ta-trendi> (дата звернення: 20.05.2024).

9. Липа М. Матеріали з дослідження, розробки та навчання ШІ. QuData.com. URL: <https://qudata.com/uk/blog/what-is-machine-learning/> (дата звернення: 10.05.2024).
10. What is reinforcement learning? | IBM. IBM - United States. URL: <https://www.ibm.com/topics/reinforcement-learning> (date of access: 14.05.2024).
11. 10 Machine Learning Algorithms to Know in 2024. Coursera. URL: <https://www.coursera.org/articles/machine-learning-algorithms> (дата звернення: 25.05.2024).
12. Breiman L. Machine Learning. 2001. Vol. 45, no. 1. P. 5–32. URL: <https://doi.org/10.1023/a:1010933404324> (дата звернення: 01.06.2024).
13. Bengio Y., Courville A., Goodfellow I. Deep Learning. MIT Press, 2016. 800 с.
14. Understanding K-means Clustering in Machine Learning. medium.com. URL: <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-бабе67336aa1> (дата звернення: 01.06.2024).
15. LeCun Y., Bengio Y., Hinton G. Deep learning. Nature. 2015. Vol. 521, no. 7553. P. 436–444. URL: <https://doi.org/10.1038/nature14539> (дата звернення: 05.06.2024).
16. What is a Neural Network? | IBM. IBM - United States. URL: <https://www.ibm.com/topics/neural-networks> (дата звернення: 12.05.2024).
17. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2016. 758 p.
18. Hastie T., Tibshirani R., Friedman J. H. The Elements of Statistical Learning. Springer, 2003. 549 p.
19. П.Е. Ситнікова, М.О. Гребенюк, Рекомендаційна система на основі компактної гібридної моделі користувача // Автоматизовані системи управління і прилади автоматики, 2023, Vol. 179, pp. 32-42. doi: 10.20837/0135-1710.2023.179.032.

20. K-Nearest Neighbor (KNN) Algorithm - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/k-nearest-neighbours/> (дата звернення: 17.05.2024).
21. Subramanian V. Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch. Packt Publishing, 2018. 262 с.
22. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 2017. Vol. 60, no. 6. P. 84–90. URL: <https://doi.org/10.1145/3065386> (дата звернення: 03.06.2024).
23. McKinney W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter. O'Reilly Media, 2022. 550 с.
24. Keras: The high-level API for TensorFlow | TensorFlow Core. TensorFlow. URL: <https://www.tensorflow.org/guide/keras> (дата звернення: 10.06.2024).
25. Moroney L. AI and Machine Learning for Coders: A Programmer's Guide to Artificial Intelligence. O'Reilly Media, Incorporated, 2020. 300 p.
26. Гладкий Д. П. ДОСЛІДЖЕННЯ ТА РОЗРОБКА АЛГОРИТМУ МАШИННОГО НАВЧАННЯ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ В СФЕРІ ОНЛАЙН-БРОНЮВАННЯ. РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ У ХХІ СТОЛІТТІ: МАТЕРІАЛИ ХХVІІІ МІЖНАР. МОЛОДІЖ. ФОРУМУ, м. Харків, 16–18 квіт. 2024 р. Харків, 2024. С. 693–694.