МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

Кафедра програмування
(повна назва кафедри)

МАГІСТЕРСЬКА РОБОТА

ВИЗНАЧЕННЯ РЕКОМЕНДОВАНОГО МІСЦЯ РОБОТИ НА ОСНОВІ АНАЛІЗУ НАВИКІВ КАНДИДАТА ІЗ ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

Виконала: студентка групи ПМІм-21
спеціальності
122 «Комп'ютерні науки»
(шифр і назва спеціальності)

(підпис) Клакович К.-Т. Р.
(прізвище та ініціали)

Керівник (підпис) Музичук А.О.
(прізвище та ініціали)

Консультант (підпис) Кущак П.Б.
(прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Львів – 2022

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
Кафедра програмування
Спеціальність 122 «Комп'ютерні науки»
(шифр і назва)

«ЗАТВЕРДЖУЮ»
Завідувач кафедри доцент Ярошко С.А.

" 13 " вересня 2022 року

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Клакович Ксенії-Тетяні Романівні
(прізвище, ім'я, по батькові)

1 .Тема роботи *ВИЗНАЧЕННЯ РЕКОМЕНДОВАНОГО МІСЦЯ РОБОТИ НА ОСНОВІ АНАЛІЗУ НАВИКІВ КАНДИДАТА ІЗ ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ*

керівник роботи *Музичук Анатолій Омелянович, доцент, консультант Кущак Петро Богданович, старший викладач,*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені Вченою радою факультету від " 13 " вересня 2022 року № 15

2. Строк подання студентом роботи

3. Вихідні дані до роботи *визначення рекомендованого місця роботи на основі аналізу навиків кандидата із використанням машинного навчання*

4. Зміст магістерської роботи (перелік питань, які потрібно розробити)
а) обрати тему, вивчити літературні джерела та скласти план роботи;
б) проаналізувати літературні джерела та викласти результати у першому та другому розділах роботи;
в) розробити модель рекомендацій та реалізувати програму;
г) досягти високої точності обраної моделі;
д) вдосконалити алгоритм виокремлення навиків користувачів;
е) виправити зауваження;
є) оформити роботу згідно з вимогами.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Кущак П. Б., старший викладач кафедри програмування	вересень 2022 р.	вересень 2022 р.
2	Кущак П. Б., старший викладач кафедри програмування	вересень 2022 р.	вересень 2022 р.
3	Кущак П. Б., старший викладач кафедри програмування	вересень 2022 р.	жовтень 2022 р.

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Вибір теми, вивчення літературних джерел та складання плану роботи	вересень 2022 р.	Виконано
2.	Підготовка першого та другого розділів роботи та подання їх керівнику	вересень 2022 р.	Виконано
3.	Розробка моделі та програмна реалізація	вересень 2022 р.	Виконано
4.	Покращення точності розробленої моделі	жовтень 2022 р.	Виконано
5.	Вдосконалення алгоритму виокремлення навиків	жовтень 2022 р.	Виконано
6.	Доопрацювання роботи з урахуванням зауважень керівника	жовтень 2022 р.	Виконано
7.	Оформлення магістерської роботи	листопад 2022 р.	Виконано

Студент _____ (підпис) ***Клакович К.-Т.Р.*** (прізвище та ініціали)

Керівник роботи _____ (підпис) ***Музичук А.О.*** (прізвище та ініціали)

Консультант _____ (підпис) ***Кущак П.Б.*** (прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 5
ВСТУП 6
1 ОГЛЯД ТЕХНОЛОГІЇ МАШИННОГО НАВЧАННЯ 8
1.1. Методи машинного навчання 9
1.2 Алгоритм розв’язування задачі машинного навчання 13
1.3 Вибір алгоритму навчання 14
2 ОСНОВНІ АЛГОРИТМИ НАВЧАННЯ. СИСТЕМИ РЕКОМЕНДАЦІЙ 17
2.1 Навчання з учителем 17
2.2 Огляд стратегій створення рекомендаційних систем 25
3 ДИЗАЙН ТА ПРОГРАМНА РЕАЛІЗАЦІЯ 30
3.1 Вибір моделі 30
3.2 Отримання та опрацювання даних 33
3.3 Візуалізація корпусу 35
3.4 Ініціалізація даних моделі та генерування індексу схожості 37
3.4.1 Байєсів персоналізований рейтинг 38
3.4.2 Класифікація навиків 39
4 РЕЗУЛЬТАТИ ТА ЇХ АНАЛІЗ 41
4.1 Оцінювання моделі 41
4.2 Генерування рекомендацій 43
ВИСНОВКИ 45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 47

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ОВМ	Опорно-векторна машина
API	Application Programming Interface. Прикладний програмний інтерфейс
CART	Classification and Regression Tree. Алгоритм, що вирішує задачі класифікації та регресії методом побудови дерева рішень
KNN	Алгоритм k-найближчих сусідів
NLP	Natural Language Processing. Обробка природної мови
Random Forest	Випадковий ліс
SVM	Support vector machine. Метод опорних векторів

ВСТУП

З появою електроніки розпочались перші спроби апаратного відтворення процесу мислення людини шляхом створення нейронних мереж, які здатні встановлювати складні зв’язки між елементами і приймати рішення. Зокрема, можливість навчання — одна з головних переваг алгоритмів машинного навчання над традиційними алгоритмами.

Технологія машинного навчання широко використовується провідними світовими технологічними компаніями, такими як Кортана від Microsoft, Google Assistant від Google чи Siri від Apple. Ці голосові асистенти розпізнають слова, які ми вимовляємо, за допомогою обробки природної мови, перетворюють їх у числа за допомогою машинного навчання та формують відповідний результат.

У час швидких, широко поширених і непередбачуваних соціальних та економічних змін, часто виникає необхідність працівників змінювати роботу. Така потреба виникає через закриття чи переміщення підприємств, глобальну кризу, таку як пандемія COVID-19 чи бойові дії в певному регіоні, що спричиняє масове витіснення робочої сили. Незалежно від причини, люди стикаються з проблемою зміни роботи, пошук якої є справою часто нетривіальною, адже потребує узгодження з наявними у певному регіоні вакансіями та відповідності навичок людини до вимог роботодавця.

Використання алгоритмів машинного навчання дає змогу передбачити, яке місце роботи може зацікавити користувача, надавши кандидату список посад із актуальними вакансіями, відповідно до його характеристик, що зробить процес зміни роботи легшим та ефективнішим. Типові сайти пошуку роботи надають користувачам актуальні вакансії, проте принцип роботи більшості з них обмежується тим, що кандидати вводять бажану назву посади та, можливо, кілька інших параметрів у вікно пошуку, щоб отримати відповідні вакансії. Основна проблема цього підходу полягає в тому, що кандидату не будуть представлені жодні інші пропозиції крім тих, які однозначно відповідають початковому запиту, хоч в перспективі могли б зацікавити користувача, оскільки узгоджуються з його навичкам та попереднім досвідом.

Як відомо, навички, необхідні для отримання роботи у певній сфері, часто співпадають з тими, які потрібно мати у іншій, що означає, що люди можуть здійснювати пошук роботи не безпосередньо за конкретною професією, а переліком навиків, проте така робота системи не підтримується на більшості сайтів з пошуку роботи. Цю проблему можна розв’язати використовуючи алгоритми машинного навчання: проаналізувати та встановити зв’язки між набором навиків кандидата і висунути гіпотезу щодо вакансій, які можуть зацікавити. Власне цьому дослідженню визначення рекомендованого місця роботи кандидата і присвячена магістерська робота.

Об'єктом досліджень цієї магістерської роботи є система рекомендацій у машинному навчанні з використанням алгоритму наївного Байєсового класифікатора та стратегії фільтрування вмісту. Предметом досліджень є визначення рекомендованого місця роботи на основі аналізу навиків кандидата із застосуванням машинного навчання.

Метою досліджень є підтвердження гіпотези про те, що можна з високою точністю класифікувати актуальні вакансії на основі навиків кандидата, що здійснює пошук роботи, використовуючи для цього алгоритми машинного навчання. Досягнення мети досліджень передбачає розв’язання таких завдань: теоретичний аналіз даних наукової літератури щодо алгоритмів класифікації, подання запропонованої моделі, реалізація програми із застосуванням одного з обраних методів, визначення точності отриманої моделі щодо розв’язання поставленої задачі.

1 ОГЛЯД ТЕХНОЛОГІЇ МАШИННОГО НАВЧАННЯ

Через нові обчислювальні технології машинне навчання сьогодні не схоже на машинне навчання минулого. Воно народилося завдяки розпізнаванню образів та теорії, згідно з якою комп'ютери можуть вчитися, не будучи запрограмованими на виконання конкретних завдань; дослідники, які цікавляться штучним інтелектом, мали на меті дізнатись, чи можуть комп'ютери навчатися на даних. Ітераційний аспект машинного навчання дуже важливий, оскільки тоді, коли до моделі надходять нові дані, вона може самостійно адаптуватися. Моделі навчаються на отриманих раніше результатах для отримання точних результатів. Таким чином, машинне навчання не є новою наукою, а такою, що набрала нових обертів.

Хоча багато алгоритмів машинного навчання існують вже давно, можливість автоматичного застосування складних математичних обчислень до великих даних - знову і знову, все швидше і швидше - є недавньою розробкою.

Відновлення інтересу до машинного навчання зумовлене тими ж факторами, що зробили обробку даних та байєсівський аналіз більш популярними, ніж будь-коли, а саме такі поняття, як збільшення обсягів та різновидів доступних даних, обчислювальна обробка, яка є більш дешевою та потужною, та доступне зберігання даних.

Усе це означає, що можна швидко та автоматично створювати моделі, які можуть аналізувати більші, складніші дані та забезпечувати швидкі та більш точні результати - навіть у дуже великих масштабах. А будуючи точні моделі, організація має більше шансів виявити вигідні можливості - або уникнути невідомих ризиків.

Більшість галузей, що працюють з великими обсягами даних, вміло використовують цінність технології машинного навчання. Вибираючи ці дані - часто в реальному часі - організації можуть працювати ефективніше та отримувати перевагу над конкурентами. Зокрема, технології машинного навчання часто використовуються у фінансових послугах, в галузі охорони здоров'я, транспортування чи роздрібної торгівлі.

Банки та інші підприємства у фінансовій галузі використовують технологію машинного навчання для двох ключових цілей: виявлення важливих даних у

корпусі та запобігання шахрайству. Аналіз може визначити інвестиційні можливості або допомогти інвесторам знати, коли торгувати. Видобуток даних також може ідентифікувати клієнтів з профілями високого ризику або використовувати кібернагляд, щоб визначити попереджувальні ознаки шахрайства.

Машинне навчання - це швидкозростаюча тенденція у галузі охорони здоров'я завдяки появі носимих пристроїв та датчиків, які можуть використовувати дані для оцінки стану здоров'я пацієнта в режимі реального часу. Ця технологія також може допомогти медичним експертам проаналізувати дані, щоб виявити тенденції або червоні позначки, які можуть призвести до поліпшення діагнозів та лікування.

Також великої популярності набувають веб-сайти, що рекомендують покупцям товари на основі їх попередніх покупок, використовуючи машинне навчання для аналізу історії покупок. Крім того, роздрібні продавці покладаються на машинне навчання для збору даних, їх аналізу та використання для персоналізації досвіду покупок, реалізації маркетингової кампанії, оптимізації цін, планування постачання товарів та отримання інформації про клієнтів.

1.1. Методи машинного навчання

До методів машинного навчання відносять навчання з учителем, без нагляду, навчання підкріплення та напівавтоматичне навчання, найпоширенішими з яких є контрольоване навчання та навчання без учителя.

Рисунок 1.1 - Типи машинного навчання

У методі навчання з учителем алгоритми навчаються на прикладах з мітками, де бажаний результат відомий наперед [8]. Наприклад, дані можуть бути позначені або "F" (не істина), або "R" (істина). Алгоритм навчання отримує набір входів

разом з відповідними правильними результатами, і алгоритм навчається, порівнюючи фактичний результат з правильними результатами для пошуку помилок. Потім алгоритм відповідно модифікує модель. Завдяки таким методам, як класифікація, регресія, прогнозування та посилення градієнта, контрольоване навчання використовує шаблони, отримані внаслідок тренування моделі, для прогнозування значень міток на додаткових немаркованих даних.

Навчання з учителем зазвичай використовується в аплікаціях, де необхідно передбачити ймовірні майбутні події на основі попередніх даних. Наприклад, можна передбачити ситуації, коли операції з кредитними картками можуть бути шахрайськими або який страховий клієнт може подати претензію [9].

User ID	Gender	Age	Salary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	1
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	1
15728773	Male	27	58000	1
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	1
15727311	Female	35	65000	0
15570769	Female	26	80000	1
15606274	Female	26	52000	0
15746139	Male	20	86000	1
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1

Figure A: CLASSIFICATION

Temperature	Pressure	Relative Humidity	Wind Direction	Wind Speed
10.69261758	986.882019	54.19337313	195.7150879	3.278597116
13.59184184	987.8729248	48.0648859	189.2951202	2.909167767
17.70494885	988.1119385	39.11965597	192.9273834	2.973036289
20.95430404	987.8500366	30.66273218	202.0752869	2.965289593
22.9278274	987.2833862	26.06723423	210.6589203	2.798230886
24.04233986	986.2907104	23.46918024	221.1188507	2.627005816
24.41475295	985.2338867	22.25082295	233.7911987	2.448749781
23.93361956	984.8914795	22.35178837	244.3504333	2.454271793
22.68800023	984.8461304	23.7538641	253.0864716	2.418341875
20.56425726	984.8380737	27.07867944	264.5071106	2.318677425
17.76400389	985.4262085	33.54900114	280.7827454	2.343950987
11.25680746	988.9386597	53.74139903	68.15406036	1.650191426
14.37810685	989.6819458	40.70884681	72.62069702	1.553469896
18.45114201	990.2960205	30.85038484	71.70604706	1.005017161
22.54895853	989.9562988	22.81738811	44.66042709	0.264133632
24.23155922	988.796875	19.74790765	318.3214111	0.329656571

Figure B: REGRESSION

Рисунок 1.2 - Набір даних торгового магазину

Рисунок 1.2 А: Набір даних торгового магазину, який корисний для прогнозування того, чи купуватиме клієнт певний товар, що розглядається, на основі його статі, віку та зарплати. Тут вхідні дані - це стать, вік, зарплата. Результат передбачень: чи придбано товар, тобто 0 або 1.

Рисунок 1.2 В: Метеорологічний набір даних, який служить для прогнозування швидкості вітру на основі різних параметрів. Вхідні дані: температура, тиск, відносна вологість, напрямок вітру. Результат - швидкість вітру.

Для навчання моделі дані, як правило, ділять у співвідношенні 80:20 , тобто 80% в якості навчальних даних і решта для тестування моделі. Модель навчається лише на основі тренувальних даних, для побудови якої використовуються різні

алгоритми машинного навчання. Навчаючись, модель будує зв'язки між ознаками корпусу для подальшого використання під час тестування.

Після того, як модель завершила тренування, відбувається її тестування. На момент тестування використовуються інші 20% вхідних даних, яких модель ще ніколи не бачила. Далі модель передбачає деякі значення, які згодом порівнюються з фактичним результатом для обчислення точності моделі.

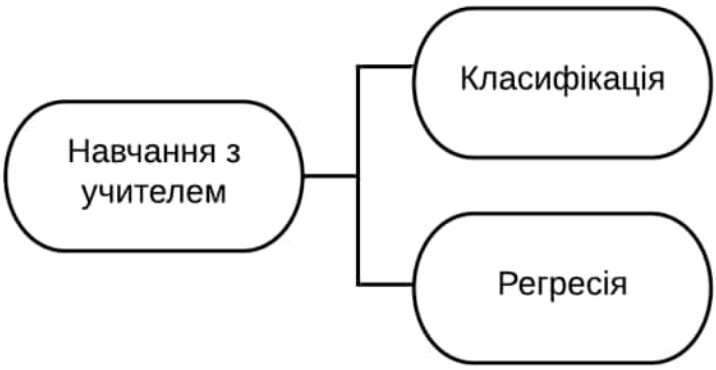

Рисунок 1.3 - Типи навчання з учителем

Розрізняють кілька типів навчання з учителем - класифікація та регресія.

Класифікація - це метод контрольованого навчання, де є наперед визначений результат з мітками(дискретні значення). Наприклад, 1 - клієнт купуватиме певний товар, або 0 - не купуватиме. Метою тут є прогнозування дискретних значень, що належать до певного класу, та оцінка моделі на основі визначення точності.

У класифікації може здійснюватись як двійкова, так і багатокласова класифікація. У двійковій класифікації модель передбачає такі значення, як 0 або 1; а у випадку класифікації з кількома класами модель передбачає більше одного класу.

Прикладом використання методу класифікації машинного навчання може слугувати те, як Gmail поділяє електронні листи на кілька груп, таких як: соціальні мережі, рекламні листи, оновлення системи, спам [9].

Ще одним методом контрольованого навчання є регресія. У цьому випадку множина можливих відповідей нескінченна (відповіді є дійсними числами або векторами). Метою регресії є передбачення значення, настільки близького до фактичного вихідного значення, наскільки може передбачити модель. Після цього

проводиться оцінка моделі шляхом обчислення значення похибки. Чим менша похибка, тим більша точність регресійної моделі.

Для навчання без учителя використовуються дані, де немає попередніх результатів. Тобто систему не відомі "правильні відповіді", а сам алгоритм повинен визначити результат. Метою є дослідження даних та знаходження певної структури всередині. Навчання без учителя добре працює з даними про транзакції. Наприклад, можна ідентифікувати сегменти клієнтів з подібними атрибутами, з якими потім можна поводитися подібним чином у маркетингових кампаніях. Також можна знайти основні атрибути, що відокремлюють сегменти клієнтів один від одного. До популярних методів відносять також самоорганізовані карти, відображення найближчих сусідів, кластеризацію методом k-середніх та декомпозицію. Ці алгоритми також використовуються для сегментації текстових даних, рекомендації елементів та виявлення неточностей.

Навчання без учителя використовується у тих самих програм, що і контрольоване навчання. Проте у випадку першого для навчання використовуються як марковані, так і немарковані дані - як правило, невелика кількість маркованих даних із великою кількістю немаркованих даних (оскільки немарковані дані вимагають менших зусиль для їх отримання). Цей тип навчання можна використовувати з такими методами, як класифікація, регресія та прогнозування.

Метод напівавтоматичного навчання лежить між навчанням з учителем та неконтрольованим навчанням. Цей вид навчання використовується у тому випадку, коли маємо справу з даними, мала частина яких є промаркованою, а решта - залишається без маркування. У цьому випадку можна використовувати некеровану техніку для прогнозування міток, а потім подавати ці мітки до контрольованих методів навчання. Цей прийом в основному застосовується у випадку наборів даних зображень, де, як правило, всі зображення не марковані.

Наступною галуззю машинного навчання є навчання з підкріпленням, яке часто використовується для робототехніки, ігор та навігації. Під час навчання з підкріпленням алгоритм виявляє шляхом спроб і помилок те, які дії приносять

найбільші винагороди. Цей тип навчання має три основні компоненти: агент (той, хто навчається або приймає рішення), середовище (те, з чим взаємодіє агент) та дії (те, що може робити агент). Мета навчання полягає в тому, щоб агент вибирав дії, які максимізують очікувану винагороду протягом заданого періоду часу. Агент досягне мети набагато швидше, дотримуючись належної політики, тому метою навчання з підкріпленням є вивчення найкращої політики.

1.2 Алгоритм розв’язування задачі машинного навчання

Для вирішення проблеми із використанням методу навчання з учителем необхідно кілька основних кроків.

По-перше, необхідно визначити тип навчальних прикладів. Для цього користувач повинен вирішити, який тип даних використовуватиметься як навчальний набір. Наприклад, у випадку аналізу рукописного тексту це може бути окремий символ, слово або цілий рядок тексту, написаного від руки.

По-друге, необхідно зібрати навчальний набір, який повинен відповідати реальному використанню функції. Таким чином, набір вхідних об'єктів збирається разом з відповідними результатами у, наприклад, людей-експертів в обраній галузі.

Далі необхідно визначити подання вхідних ознак вивченої функції. Точність вивченої функції сильно залежить від того, як представлений вхідний об'єкт. Як правило, введений об'єкт перетворюється у вектор, який містить ряд ознак, що описують заданий об'єкт. Кількість функцій не повинна бути занадто великою через так зване прокляття розмірності, але має бути достатньо інформації для точного прогнозування результату.

На наступному кроці необхідно визначити структуру вивченої функції та обчати алгоритм навчання. Наприклад, можна обрати метод опорних векторів або дерево рішень.

Далі необхідно запустити алгоритм навчання на зібраному навчальному наборі. Деякі контрольовані алгоритми навчання вимагають від користувача визначення певних параметрів управління. Ці параметри можуть бути скориговані шляхом оптимізації продуктивності підмножини навчального набору або за допомогою перехресної перевірки.

На завершення необхідно оцінити точність вивченої функції. Після налаштування параметрів та вивчення, результативність отриманої функції слід вимірювати на тестовому наборі, який є незалежним від навчального набору.

1.3 Вибір алгоритму навчання

Важливим етапом моделювання у машинному навчанні є вибір алгоритму. Як вже згадувалось, доступний широкий спектр керованих алгоритмів навчання, кожен із яких має свої переваги на недоліки. Не існує такого алгоритму навчання, який би найкраще працював з усіма навчальними проблемами [2].

Є чотири основні пункти, які слід враховувати під час використання методу навчання з учителем:

1. Компроміс між зміщенням і дисперсією

Помилка передбачення вивченого класифікатора пов’язана із сумою упередженості та дисперсією алгоритму навчання. Як правило, існує компроміс між упередженістю та дисперсією. Алгоритм навчання з низьким зміщенням повинен бути "гнучким", щоб могти добре відповідати даним. Але якщо алгоритм навчання занадто гнучкий, він буде по-різному підходити до кожного набору навчальних даних, а отже, мати велику дисперсію. Ключовим аспектом багатьох контрольованих методів навчання є те, що вони здатні регулювати цю компромісну ситуацію між зміщенням та дисперсією (або автоматично, або шляхом надання параметра зміщення/дисперсії, які регулюються користувачем).

2. Складність функції та обсяг навчальних даних

Під час використання методу навчання з учителем варто звертати увагу на кількість наявних навчальних даних щодо складності функції (класифікатора або функції регресії). Якщо функція є досить простою, то «негнучкий» алгоритм навчання з великим упередженням і малою дисперсією може засвоїти її і з невеликою кількістю даних. Але якщо функція є більш складною (наприклад, якщо вона передбачає складні взаємодії між багатьма вхідними функціями і поводиться по-різному в різних частинах вхідного простору), то функція зможе показати добрий результат лише з дуже великої кількості навчальних даних та використання "гнучкого" алгоритму навчання з низьким упередженням та великою дисперсією.

3. Розмірність вхідного простору

Також варто враховувати розмірність вхідного простору. Якщо вектори вхідних ознак мають дуже високу розмірність, то проблема навчання може бути складною, навіть якщо справжня функція залежить лише від невеликої кількості цих ознак. Це пов'язано з тим, що безліч «зайвих» вимірів можуть заплутати алгоритм навчання та спричинити велику дисперсію. На практиці, якщо інженер може вручну видалити непотрібні функції з вхідних даних, це, ймовірно, покращить точність вивченої функції. Крім того, існує безліч алгоритмів вибору ознак, які прагнуть ідентифікувати відповідні ознаки та відкинути непотрібні.

4. Шумові значення

Якщо бажані вихідні значення часто є неправильними, тоді алгоритм навчання не повинен намагатися знайти функцію, яка точно відповідає прикладам навчання. Спроба надто ретельно підігнати дані призводить до перенавчання. У такій ситуації частина цільової функції, яку неможливо змоделювати, «псує» навчальні дані - це явище називають детермінованим шумом [7]. Коли присутній будь-який тип шуму, краще вибирати алгоритм з більшим зміщенням.

На практиці існує кілька підходів для зменшення шуму у вихідних значеннях, такі як рання зупинка для запобігання перенавчанню, а також виявлення та видалення шумових даних перед навчанням.

Інші фактори, які слід враховувати при виборі та застосуванні алгоритму навчання, включають, наприклад, неоднорідність даних. Якщо вектори ознак включають ознаки багатьох різних видів (дискретні, дискретні впорядковані, відліки, безперервні значення), деякі алгоритми застосовувати простіше, ніж інші. Багато алгоритмів, включаючи машини з підтримкою векторів, лінійну регресію, логістичну регресію, нейронні мережі та методи найближчих сусідів, вимагають, щоб вхідні функції були числовими та масштабованими до подібних діапазонів. Особливо чутливі до цього методи, що використовують функцію відстані, такі як методи найближчих сусідів та машини векторних опор з ядрами Гауса. Перевагою дерев рішень є те, що вони легко обробляють різнорідні дані [11].

Якщо вхідні функції містять надлишкову інформацію (наприклад, високо корелюючі функції), деякі алгоритми навчання (наприклад, лінійна регресія, логістична регресія та методи, засновані на відстані) будуть погано працювати через числові нестабільності. Ці проблеми часто можна вирішити шляхом нав'язування певної форми регуляризації.

Якщо кожна з функцій робить незалежний внесок у результат, то алгоритми, засновані на лінійних функціях (наприклад, лінійна регресія, логістична регресія) та функціях відстаней (наприклад, методи найближчих сусідів, метод опорних векторів), як правило, працюють досить добре. Однак, якщо між об’єктами існують складні взаємодії, то алгоритми, такі як дерева рішень та нейронні мережі, працюють краще, оскільки вони спеціально розроблені для виявлення цих взаємодій. Також можуть застосовуватися лінійні методи, але інженер повинен вручну вказувати взаємодії при їх використанні.

Розглядаючи новий програмний застосунок, можна порівняти декілька алгоритмів навчання та експериментально визначити, який з них найкраще працює з розглянутою проблемою. Проте часто налаштування продуктивності алгоритму навчання може зайняти багато часу. Враховуючи фіксовані ресурси, часто краще витрачати більше часу на збір додаткових навчальних даних та більш інформативні функції, ніж витрачати додатковий час на налаштування алгоритмів навчання.

Таким чином, у цьому розділі було розглянуто основи машинного навчання, визначено її можливості. Було описано види машинного навчання, визначено кроки розв’язання задачі із використанням навчання під наглядом, а також наведено пункти, які варто враховувати при виборі алгоритму навчання.

2 ОСНОВНІ АЛГОРИТМИ НАВЧАННЯ. СИСТЕМИ РЕКОМЕНДАЦІЙ

2.1 Навчання з учителем

Одним із способів машинного навчання, в ході якого випробувана система примусово навчається за допомогою наявної множини прикладів «стимул-реакція» з метою визначення «реакції» для «стимулів», які не належать наявній множини прикладів, є навчанням з учителем або контрольоване навчання. Розглянемо принцип роботи завдань навчання з учителем.

Нехай дано N навчальних прикладів вигляду

$$\{(x_1, y_1), \ldots, (x_N, y_N)\}, \tag{2.1}$$

де x_i - вектор ознак і-го прикладу з міткою y_i.

Алгоритм навчання шукає функцію

$$\text{g: } X -> Y, \tag{2.2}$$

де X - вхідний простір, а Y - вихідний.

Функція g є елементом деякого простору можливих функцій G, що називається простором гіпотез [3]. Це зручно представляти за допомогою функції $f: X \times Y -> R$ такої, коли g визначена, як вихідне значення y, що дає найвищий результат

$$g(x) = arg\ max\ f(x, y) \tag{2.3}$$

Нехай F - це простір таких функцій. Хоч G і F можуть представляти будь-який простір функцій, багато алгоритмів навчання є ймовірнісними моделями, де g приймає форму моделі умовної ймовірності $g(x) = P(y|x)$, або f набуває форми моделі спільного розподілу $f(x, y) = P(x, y)$. Наприклад, алгоритм наївного Байєсового класифікатора та дискримінантний аналіз використовують моделі спільного розподілу, тоді як логістична регресія - модель умовної ймовірності.

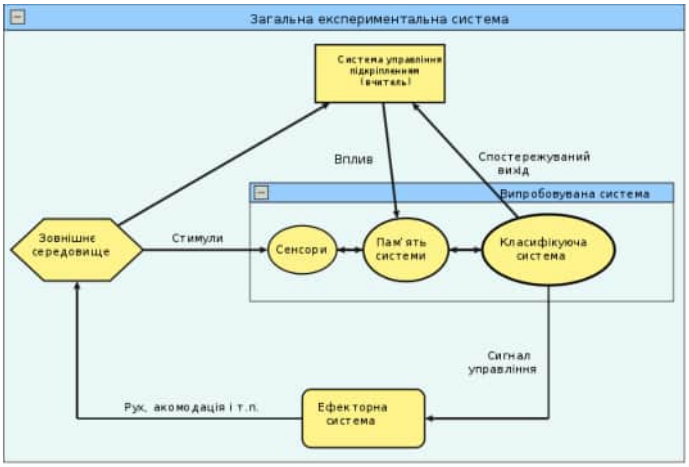

Рисунок 2.1 - Навчання з учителем

Існує 2 основних підходи до вибору функцій f або g: емпірична мінімізація та мінімізація структурних ризиків. Емпірична мінімізація ризику шукає таку функцію, що найкраще відповідає навчальним даним. З іншого боку, функція мінімізації структурних ризиків включає так звану penalty function, що контролює компроміс між зміщенням та дисперсією.

В обох випадках передбачається, що навчальний набір складається з вибірки незалежних і однаково розподілених пар (x_i, y_i). Для того, щоб виміряти, наскільки функція відповідає навчальним даним, необхідно визначити функцію втрат

$$L: Y \times Y \to R^{\geq 0} \tag{2.4}$$

Для тренувальних даних (x_i, y_i) втрата прогнозування значення $\hat{y} \in L(y_i, \widehat{y})$.

Ризик $R(g)$ функції g визначається як очікувана втрата g. Це можна оцінити з навчального набору даних як

$$R_{emp}(g) = \frac{1}{N} \sum_i L(y_i, g(x_i)) \tag{2.5}$$

Серед алгоритмів керованого навчання не існує такого, який би найкраще працював з усіма навчальними вибірками, про що твердить теорема про відсутність безкоштовних сніданків.

Деякі обчислювальні задачі розв'язуються пошуком рішень у множині допустимих розв'язків. Опис того, як швидко знаходити гарні рішення та відкидати погані, називається пошуковим алгоритмом [6]. На конкретній проблемі різні алгоритми можуть показувати різні результати, але на множині всіх задач вони розрізнювальні. Це поширюється на властивість алгоритмів платити за першість у

розв'язку одних задач значно гіршими результатами на інших задачах. З іншого боку, продуктивність пошуку зберігається.

Цікавою є теорема про відсутність безкоштовних сніданків, з якої випливає, що, теоретично, всі алгоритми показують добрі результати у процесах оптимізації майже завжди. Тобто алгоритм отримує добрі розв'язки за відносно невелике число обчислень на майже всіх цільових функціях. Причиною є те, що багато цільових функцій виявляють великий ступінь Колмогорської складності. Це призводить до надзвичайної непостійності та непередбачуваності. Всі ступені якості рівномірно розподілені між допустимими рішеннями, а добрі рішення розкидані по всій множині допустимих рішень. Пошуковому алгоритму рідко доведеться перебрати велику кількість розв'язків, щоб знайти перший дійсно хороший.

Для розв’язування задач класифікації є кілька найбільш популярних алгоритмів, які використовуються у машинному навчанні.

До цих алгоритмів належать: логістична регресія, дерево ухвалень рішень, Random forest (випадковий ліс), наївний Байєсів класифікатор і метод опорних векторів. Розглянемо детальніше кожен з них.

Алгоритм логістичної регресії використовується для двійкової класифікації точок даних. Класифікація проводиться таким чином, що результат належить до одного з двох класів (1 або 0). Наприклад, можна передбачити,чи падатиме дощ, зважаючи на погодні умови.

Дві найважливіші частини логістичної регресії - це гіпотеза та сигмоїдна крива . За допомогою гіпотези визначають ймовірність події, а дані, отримані з цієї гіпотези, поміщають у функцію log, що утворює сигмоїду. Використовуючи цю функцію, можна далі передбачити категорію класу.

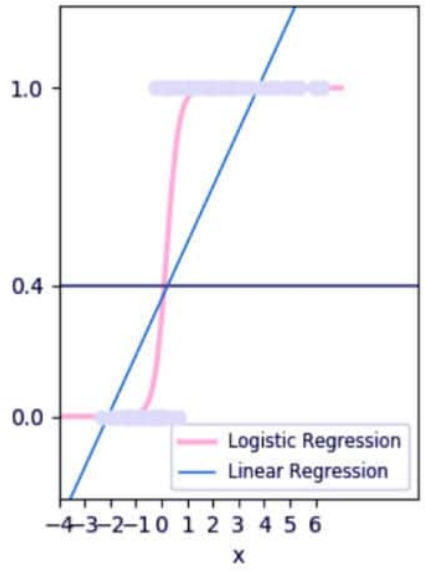

Рисунок 2.2 - Сигмоїдна функція

$$\Lambda(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}} \quad , \tag{2.5}$$

де Λ позначає логістичну функцію.

Ще одним алгоритмом розв'язування задач машинного навчання є наївний Байєсів класифікатор. Це один із потужних алгоритмів машинного навчання, який використовується для класифікації. В основі алгоритму лежить теорема Байєса, де кожна ознака є незалежною. Алгоритм використовується для різних завдань, таких як: фільтрація спаму та класифікація частин тексту.

До переваг алгоритму відносять:

- простота та швидкість передбачення класу набору даних, а також можливість виконувати багатокласові передбачення;
- коли виконується припущення про незалежність, алгоритм набагато ефективніший ніж інші алгоритми, такі як логістична регресія, і, крім того, потребує меншої кількості навчальних даних.

Недоліки алгоритму наївного Байєсового класифікатора:

- Якщо категоріальна змінна належить до категорії, яка не була проаналізована у навчальному наборі, тоді модель присвоїть їй ймовірність 0, що не дозволить їй робити будь-які подальші прогнози;

- В основі алгоритму лежить припущення про незалежність ознак. Проте у реальному житті майже немає даних, які включають абсолютно незалежні особливості.

Алгоритми дерева рішень використовуються в машинному навчанні як для прогнозування, так і для класифікації. Використовуючи дерево рішень із заданим набором вхідних даних, можна відобразити результати, які є результатом наслідків або рішень.

Розглянемо роботу алгоритму на прикладі. Припустимо, потрібно піти на ринок, щоб придбати деякі товари. Перед цим потрібно оцінити, чи справді потрібен товар. Тобто, товар буде куплено тільки якщо його запаси закінчились. Далі буде оцінено погоду надворі - дощ падає або ні. Якщо дощ не падає, то людина піде на ринок, а в іншому випадку - ні. Дану ситуацію можна представити у вигляді дерева рішень, що зображене на рис. 2.3.

Рисунок 2.3 - Дерево прийняття рішень

Це дерево рішень є результатом кількох ієрархічних кроків, які допомагають прийняти певні рішення. Для побудови цього дерева є два етапи - індукція та відсікання гілок. На практиці в результаті роботи цього алгоритму часто виходять занадто деталізовані дерева, які при їх подальшому застосуванні дають багато помилок. Це пов'язано з явищем перенавчання. Для скорочення дерев використовується відсікання гілок

Алгоритм K-найближчих сусідів(KNN) є одним з найпростіших, але в той же час і одним з найважливіших алгоритмів класифікації в машинному навчанні. KNN належать до навчання з учителем і використовується для розпізнавання, аналізу даних та у системах виявлення вторгнень. В алгоритмі для класифікації об'єктів у рамках простору властивостей використовуються відстані (зазвичай евклідові), пораховані до усіх інших об'єктів. Вибираються об'єкти, до яких відстань найменша, і вони виділяються в окремий клас.

Основним принципом методу найближчих сусідів є те, що об'єкт присвоюється тому класу, який є найбільш поширеним серед сусідів даного елемента. Сусіди беруться, виходячи з множини об'єктів, класи яких уже відомі, і, виходячи з ключового для даного методу значення k, вираховується, який клас є найчисленнішим серед них. Кожен об'єкт має кінцеву кількість атрибутів (розмірностей). Передбачається, що існує певний набір об'єктів з уже наявною класифікацією.

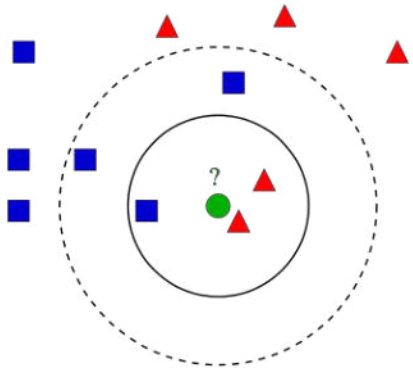

Рисунок 2.4 - Приклад класифікації методом KNN

Тестовий зразок (зелене коло) повинен бути класифікований як синій квадрат (клас 1) або як червоний трикутник (клас 2). Якщо k = 3, то класифікується як 2-й клас, тому що всередині меншого кола 2 трикутника і тільки 1 квадрат. Якщо k = 5, то він буде класифікований як такий, що належить до першого класу.

Розглянемо ще один алгоритм машинного навчання - випадковий ліс.

Нехай навчальна вибірка складається з N зразків, розмірність простору ознак дорівнює M, і заданий параметр m, як неповна кількість ознак для навчання.

Згенеруємо випадкову вибірку з повтореннями розміром N з навчальної вибірки. Таким чином, деякі зразки потраплять в неї два або більше разів, а в

середньому $(1-1/N)^N$ (при великих N приблизно N/e, де e зразків не увійдуть в неї взагалі. Ті зразки, які не були в вибірку, називаються out-of-bag (невідібрані) [2].

Побудуємо вирішальне дерево, що класифікує зразки даної підвибірки, причому в ході створення чергового вузла дерева будемо вибирати набір ознак, на основі яких проводиться розбиття (не з усіх M ознак, а лише з m випадково обраних). Вибір найкращого з цих m ознак може здійснюватися різними способами. В оригінальному варіанті використовується критерій Джині, що застосовується також в алгоритмі побудови дерев рішень Classification and Regression Tree(CART). У деяких реалізаціях алгоритму замість нього використовується критерій приросту інформації.

Дерево будується до повного вичерпання підвибірки і не піддається процедурі відсікання гілок (на відміну від дерев рішень, побудованих за таким алгоритмом, як CART).

Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету відносить об'єкт, що класифікується, до одного з класів, і перемагає клас, за який “проголосувала” найбільша кількість дерев.

Оптимальне число дерев підбирається таким чином, щоб мінімізувати помилку класифікатора на тестовій вибірці. У разі її відсутності, мінімізується оцінка помилки out-of-bag: тих зразків, які не були в навчальну подвиборку за рахунок повторень (їх є близько N/e).

Випадкові ліси, одержувані в результаті застосування технік, описаних раніше, можуть бути природним чином використані для оцінки важливості змінних в задачах регресії і класифікації. Наступний спосіб такої оцінки був описаний Брейманом.

Під час процесу побудови моделі для кожного елемента тренувального набору записується так звана out-of-bag-помилка. Потім для кожної сутності така помилка обчислюється, як середнє значення по всьому випадковому лісі.

Для того, щоб оцінити важливість j-го параметра після тренування, його значення перемішуються для всіх записів тренувального набору і out-of-bag-

помилка обчислюється заново. Важливість параметра оцінюється шляхом усереднення по всіх деревах різниці показників out-of-bag-помилок до і після перемішування значень. При цьому значення таких помилок нормалізуються на стандартне відхилення.

Параметри вибірки, які дають великі значення, вважаються більш важливими для тренувального набору. Метод має наступний потенційний недолік - для категорійних змінних з великою кількістю значень метод схильний вважати такі змінні більш важливими. Часткове перемішування значень в цьому випадку може знижувати вплив цього ефекту. З груп корелюють параметрів, важливість яких виявляється однаковою, вибираються менші за чисельністю групи.

До переваг алгоритму Random Forest належать:

1. Здатність ефективно обробляти дані з великим числом ознак і класів
2. Нечутливість до масштабування (і взагалі до будь-яких монотонним перетворенням) значень ознак.
3. Однаково добре обробляються як безперервні, так і дискретні ознаки. Існують методи побудови дерев за даними з пропущеними значеннями ознак.

З іншого боку, недоліком алгоритму є великий розмір отриманих моделей. Потрібно O(K) пам'яті для зберігання моделі, де K - число дерев.

Метод опорних векторів (SVM) - це алгоритм машинного навчання, у якому використовується подання навчальних даних у вигляді точок у просторі, розділених на категорії чітким розривом, який є якомога ширшим. Потім нові приклади відображаються в тому самому просторі та, як передбачається, належать до категорії, залежно від того, на яку сторону розриву вони потрапляють.

Параметри максимально розділової гіперплощини виводяться шляхом розв'язання задачі оптимізації. Існує кілька спеціалізованих алгоритмів для швидкого розв'язання задач КП, що виникають в опорно-векторній машині(ОВМ), вони здебільшого покладаються на евристики для розбиття задачі на менші підзадачі, з якими легше впоратися.

Іншим підходом є застосування методу внутрішньої точки, який використовує ньютоно-подібні ітерації для пошуку розв'язку умов Каруша — Куна

— Таккера прямої та двоїстої задач. Замість розв'язання послідовності розбитих задач, цей підхід безпосередньо розв'язує задачу в цілому. Для уникнення розв'язання лінійної системи з великою ядровою матрицею в ядровому методі часто використовується низькорангове наближення матриці.

Ще одним поширеним методом є алгоритм послідовної мінімальної оптимізації. У ньому площина, який розбиває задачу на 2-вимірні підзадачі, які розв'язуються аналітично, усуваючи потребу в алгоритмі числової оптимізації та в зберіганні матриці. Цей алгоритм є простим концептуально, простим у реалізації, зазвичай швидшим, і має кращі властивості масштабування для складних задач ОВМ.

Окремий випадок лінійних опорно-векторних машин може розв'язуватися ефективніше алгоритмами того ж роду, що й використовуються для оптимізації їхнього близького родича, логістичної регресії; цей клас алгоритмів включає субградієнтний спуск та координатний спуск. Останній володіє деякими привабливими властивостями часу тренування. Кожна ітерація збіжності займає час, лінійний по відношенню до часу, витраченого на читання тренувальних даних, й ітерації також володіють властивістю Q-лінійної збіжності, що робить цей алгоритм надзвичайно швидким.

Звичайні ядрові ОВМ також можуть розв'язуватися ефективніше при застосуванні субградієнтного спуску, особливо якщо дозволено розпаралелювання. Цей метод машинного навчання був детальніше розглянутий у попередній роботі.

Отже, у цьому розділі було розглянуто основні алгоритми, які застосовуються при класифікації під час застосування технології машинного навчання, розглянуто принцип роботи керованих алгоритмів навчання та досліджено переваги на недоліки алгоритмів, які будуть використовуватись у практичній частині досліджень.

2.2 Огляд стратегій створення рекомендаційних систем

Багато компаній, такі як Amazon, Netflix, Linkedin та Pandora, використовують на своїх веб-сайтах системи рекомендацій, розроблені для того, щоб допомогти користувачам відкривати для себе нові та актуальні продукти,

будуючи взаємодію з користувачем, при цьому збільшуючи прибуток компанії. Система рекомендацій виявляє персоналізовані потреби та інтереси користувача шляхом аналізу поведінки користувачів і рекомендує інформацію або продукт, що може зацікавити користувачів. На відміну від пошукових систем, система рекомендацій не вимагає від користувачів точного опису своїх потреб, а моделює їх історичну поведінку, щоб проактивно надавати інформацію, яка відповідає інтересам і потребам користувачів.

Існують дві основні стратегії створення рекомендаційних систем: фільтрація вмісту і колаборативна фільтрація. Крім цього часто застосовують і гібридну рекомендаційну систему, яка є своєрідним поєднання попередніх двох.

При методі фільтрації вмісту створюються профілі користувачів і об'єктів, які досліджуються. Профілі користувачів можуть містити інформацію з особистими даними або ж відповіді на певний набір питань. Профілі об'єктів зазвичай містять інформацію з назвами жанрів, іменами акторів, виконавців, описом об’єкта тощо. Цей тип фільтрування не потребує інформації про характеристики інших користувачів.

Перевагою методу фільтрації вмісту є те, що створена за таким методом модель легко масштабується завдяки невеликій кількості даних. Крім того, оскільки, на відміну від інших моделей, цю модель не потрібно порівнювати з даними інших користувачів, вона може запропонувати більш точні результати для поточного користувача.

Недоліком рекомендаційної системи, що використовує фільтрацію вмісту є те, що модель вимагає великого обсягу знань предметної області розробників. Тому точність такої моделі значною мірою залежить від того, наскільки коректно обрані ознаки для створення векторів об’єктів.

При колаборативній фільтрації використовується інформація про поведінку користувачів у минулому — наприклад, інформація про придбання певних товарів або виставлені оцінки. В цьому разі не має значення, з якими типами об'єктів ведеться робота, але при цьому можна також брати до уваги неявні характеристики, які складно було б врахувати при створенні профілю користувача у методі

фільтрації на основі вмісту. У колаборативній фільтрації для генерування рекомендацій використовується інформація про вподобання схожих до поточного користувачів. Таким чином, обираються предмети, які придбав чи високо оцінив той користувач, який за певними характеристиками подібний до користувача, що здійснює пошук у системі.

Джерелами взаємодії користувача з елементом є явний та неявний зворотній зв'язок. Неявний зворотний зв'язок – це такий, коли оцінки користувача «подобається» та «не подобається» формуються на основі його дій, таких як відвідування сторінок, пошук та здійснення покупок. Тобто врахується активність користувача на даному веб-сайті.

Явний зворотний зв'язок – це такий, коли користувач явно вказує, що йому подобається чи не подобається, такими діями, як, наприклад, реакція на товар або його оцінка.

Основна проблема колаборативної рекомендаційної системи це «холодний старт», що позначає ситуацію, коли неможливо рекомендувати товари новим користувачам, оскільки не існує інформації про їх попередню активність. У цьому випадку новий користувач може отримувати неточні рекомендації, оскільки він ще не здійснював жодних дій чи покупок.

Також проблема холодного старту може виникнути, коли у систему додається новий товар, з яким ще не було жодної взаємодії. Якщо взаємодії недоступні, чистий алгоритм співпраці не може рекомендувати елемент. Якщо доступно лише кілька взаємодій, незважаючи на те, що алгоритм колаборативної фільтрації зможе їх рекомендувати, якість цих рекомендацій буде низькою. Тут виникає ще одна проблема, яка більше не стосується нових, а скоріше непопулярних елементів. У деяких випадках, наприклад, рекомендації фільмів, може виникнути ситуація, коли декілька елементів отримують надзвичайно велику кількість взаємодій, тоді як більшість елементів отримують лише незначну їх частину. Ця ситуація називається упередженням популярності [10].

Завдяки великій кількості доступних рекомендаційних алгоритмів, а також типу та характеристик системи були розроблені стратегії для пом'якшення

проблеми холодного запуску. Основний підхід полягає в тому, щоб покладатися на гібридні рекомендації, щоб пом’якшити недоліки однієї категорії або моделі, поєднавши її з іншою.

Одним із доступних варіантів роботи з «холодними» користувачами або предметами є попереднє отримання деяких даних. У цьому випадку користувачу пропонується надати певну інформацію про власні уподобання та інтереси, сприяючи створенню профілю користувача перш ніж система зможе самостійно надавати рекомендації.

У будь-якій системі рекомендацій важливим етапом надання рекомендацій є визначення коефіцієнта подібності між об’єктами, що розглядаються. Для визначення того, наскільки товари чи користувачі схожі між собою, використовуються метрики подібності. Хоча не існує єдиного визначення подібності, зазвичай такі показники в певному сенсі є зворотними до метрики відстані - вони приймають великі значення для подібних об’єктів і нульове або від’ємне значення для об’єктів, що відрізняються. Хоча в більш широкому сенсі функція подібності також може задовольняти метричним аксіомам.

Найчастіше використовуються такі показники подібності:

а) Косинус подібності - коефіцієнт подібності двох ненульових векторів у передгільбертовому просторі, який обчислюється як косинус кута між ними.

Однією з переваг використання метрики косинуса подібності є низька складність обчислення, особливо для розріджених векторів, у цьому випадку для обчислень достатньо брати лише координати з ненульовим значенням.

б) Евклідова відстань - це довжина вектора між двома точками, що належить одній прямій, яка проходить через ці точки.

в) Подібність Пірсона - показник кореляції (лінійної залежності) між двома змінними, який набуває значень від −1 до 1. Він широко використовується в науці для вимірювання ступеня лінійної залежності між двома змінними.

Під час вибору показника подібності варто враховувати наступні рекомендації. Подібність Пірсона варто використовувати, якщо дані залежать від упередженості користувачів або є в системі є декілька різних шкал оцінок.

Косинусну подібність потрібно використовувати, якщо дані розріджені, тобто багато оцінок є невідомими. Евклідову відстань використовують, якщо дані не розріджені, а величина значень атрибутів є значною.

3 ДИЗАЙН ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

У ході дослідження технології машинного навчання, його практичного застосування, було обрано тему “Визначення рекомендованого місця роботи на основі аналізу навиків кандидата із використанням машинного навчання”.

У попередньому розділі було розглянуто методи, які можна використовувати для реалізації системи рекомендацій для пошуку роботи. У цьому розділі розглянуто, як створити потужну модель для реалізації поставленої задачі, та покроково описано принцип її роботи.

Одним із завдань магістерської роботи є комп’ютерна реалізації програми для класифікації вхідних даних, для чого було підготовлено корпус - набір характеристик кандидатів. Корпус використовуватиметься програмою при тренуванні та тестуванні моделі, тому варто наповнити його значною кількістю даних.

Реалізація програми, що використовує машинне навчання, полягає у поетапному перетворенні вхідних даних, їх аналізу, передбаченні результату, його ілюстрації.

Розглянемо кожен етап знаходження вирішення поставленої задачі почергово.

3.1 Вибір моделі

При пошуку роботи на веб-сайті користувач прагне отримати варіанти актуальних вакансій, що підбираються не тільки на основі обраної назви професії, а й ті, що відповідають його попередньому досвіду роботи й професійним навичкам. Такі результати збільшують шанси кандидата отримати бажане місце роботи, а роботодавцям швидко відшукати людину з навиками, що відповідають поставленим вимогам. Визначення зв'язку між навичками дає змогу встановити подібність між резюме кандидата та вакансіями.

Таким чином, потрібно створити модель, яка, використовуючи дані про досвід кандидатів та професійні навички, може рекомендувати вакансії, внаслідок

аналізу та встановлення зв’язків між навичками кандидатів із вхідного набору даних.

Розроблена система видає рекомендацію щодо місця роботи людини, фільтрує актуальні вакансії зважаючи на попередній досвід працівника. У пошуковій системі здійснюється аналіз навичок людини, попереднього досвіду роботи, демографічної інформації та інших необхідних деталей. Зважаючи на отримані дані, користувачу пропонуються не тільки конкретно ті професії з ключовими словами, по яких здійснювався пошук, а й інші, які підходять даному користувачу.

Фільтрування вакансій здійснюється внаслідок прямої взаємодії з користувачем, а отримані рекомендації ґрунтуються на схожості між вакансіями, які встановила модель машинного навчання, та профілем користувача. Ці подібності залежать від уподобань користувачів, а обраний метод об’єднує всі дані для створення рейтингового списку пропозицій.

Як видно з рис. 3.1, типова система рекомендацій складається з кількох етапів.

Рисунок 3.1 - Схема роботи типової системи рекомендацій

На етапі отримання даних дані від різних користувачів збираються та зберігаються в базі даних. Ці дані містять профіль користувачів, інформацію про попередій досвід роботи та їх навички. Етап перетворення даних складається з кількох різних процесів, таких як очищення даних і формування кластерів.

Обчислювальна модель займається розрахунковою частиною роботи системи. В основному вона складається з двох частин - генерування набору результатів і фільтрування даних.

Останнім етапом роботи системи є блок рекомендацій, де користувачам надаються рекомендації залежно від відфільтрованого набору результатів з обчислювального модуля. Рекомендації щодо подібності навичок формуються на основі алгоритму наївного Байєсового класифікатора, який враховує набір навичок, надані користувачем при заповнені пошукової форми. Після цього, попередньо отримавши вектори характеристик користувачів та вакансій, система надає рекомендації топ-N елементів, що найбільше підходять.

Після аналізу теоретичної літератури я вирішила реалізувати систему рекомендацій використовуючи стратегію фільтрування вмісту, оскільки обраний набір даних не містить інформації про попередню активність та уподобання користувачів.

Розглянемо структуру схеми запропонованої моделі, що зображена на рис. 3.2. Дані, отримані з веб-сайту з розміщеними актуальними вакансій, містять інформацію щодо користувачів. На основі зібраних даних необхідні навички співставляються з кожним користувачем, а кластер навичок оновлюється.

Рисунок 3.2 - Структурна схема запропонованої моделі

Індекс подібності формується на основі Евклідової відстані між наборами навичок. Згодом подібні навички ранжуються за допомогою наївного Байєсового класифікатора. Внаслідок цих дій на виході отримуємо результат.

Перевагами запропонованої моделі є те, що її робота грунтується на статистиці та частоті появи ознак. Крім того для отримання бажаного результату необхідний невеликий набір даних.

3.2 Отримання та опрацювання даних

В дослідженні було використано набір даних, взятий з офіційного веб-сайту державного центру зайнятості у Львівській області, як зазначено в [1]. Набір даних складається з 10000 рядків та містить інформацію щодо резюме, навичок людей та їх попереднього досвіду роботи. Крім того у магістерській роботі використовувався набір даних щодо актуальних вакансій станом на вересень 2022 року, що складається з 3000 доступних пропозицій працевлаштування.

Вхідними даними є резюме кандидата, що містить інформацію про попередній досвід роботи та перелік професійних навичок, а також набір вакансій, які доступні на веб-сайті. Вихідними даними є перелік рекомендованих місць роботи з обраними актуальними вакансіями.

Набір даних є досить збалансованим, хоч і містить пропущену інформацію, наприклад, щодо очікуваної зарплати працівників чи навиків, деяка частина яких дублюється і ускладнює процес опрацювання.

Корпус програми зберігається в окремому файлі з розширенням .csv. На цьому етапі програмної реалізації використано мову програмування Python та бібліотеку pandas, за допомогою якої відбувається порядкове зчитування даних.

	DateCreate	Position	Salary	Skills
0	2022-05-19T16:24:00	продавець непродовольчих товарів	7000.0	Знання асортименту, якісних характеристик това...
1	2022-05-19T16:24:00	продавець продовольчих товарів	7000.0	Знання асортименту, якісних характеристик това...
2	2022-05-19T16:24:00	касир (на підприємстві, в установі, організації)	7000.0	Знання асортименту, якісних характеристик това...
3	2022-05-19T16:19:00	головний бухгалтер	10000.0	Уміння забезпечувати контроль і відображення н...
4	2022-05-19T16:19:00	продавець непродовольчих товарів	10000.0	Уміння забезпечувати контроль і відображення н...

Рисунок 3.3 - Таблиця вхідних даних

Після формування вхідного набору даних, необхідно сформувати так званий профіль користувача та профіль посади. Профіль користувача описує його і містить перетворену інформації про, зокрема, набір його професійних навиків. Оскільки інформація про навички представлена у текстовому вигляді, то потребує

перетворення у цифровий, який, разом з усіма характеристиками користувача, утворює вектор користувача. Далі побудовані вектори будуть співставлятись між собою для визначення подібності.

Для побудови вектора користувача були обрані такі його характеристики, як Position та Skills. Далі я використала метод перетворення ознак обробки природної мови(NLP) spaCy, який є потужним інструментом надання точного синтаксичного аналізу. Крім того бібліотека spaCy містить вбудовані функції для визначення подібності між векторами користувачів на основі порівняння слів у них.

Аналогічно для побудови профіля посади вибирають ознаки з набору даних, які дають змогу найкраще описати вакансії. Після вибору ознак, дані обробляються для формування вектора посади.

На етапі попередньої обробки корпусу даних потрібно підготувати обрані дані для наступних етапів.

В цьому дослідженні було використано кілька ефективних технік для попередньої обробки вхідних даних:

- Перетворення опису навичок у текст нижнього регістру.
- Видалення спеціальних символів, таких як #, /, & *, $, а також цифр.
- Сегментація тексту навиків у набір слів.
- Видалення стоп-слів.
- Представлення слів у векторах із заздалегідь підготовленими наборами для вбудовування слів.

Техніки обробки вхідних даних можуть застосовуватись по декілька разів у різному порядку. Результатом цього етапу є очищений набір навичок для кожного користувача. Для створення загального набору навичок з наданих вхідних даних були застосовані різні алгоритми та методики, набір навиків розбито в окремі незалежні одиниці, зручні для подальшого використання.

Отримавши очищені набори даних щодо навичок користувачів та характеристик вакансій, відбувається їх перетворення у вектори ознак, процес

якого був описаний раніше. В результаті отримуємо дані з векторами користувачів та описів роботи з вакансій.

3.3 Візуалізація корпусу

Для ознайомлення з набором даних було здійснено їх аналіз із використанням візуалізації даних. Спершу було розглянуто дані кандидатів на позиції, чия назва містить частину «програм».

	DateCreate	Position	Salary	Skills
269	2022-05-19T10:08:00	Фахівець з розробки та тестування програмного ...	15000.0	Уміння розробляти документи правового характер...
346	2022-05-18T16:38:00	інженер-програміст	10000.0	Уміння розробляти та впроваджувати технологічн...
603	2022-05-17T16:33:00	оператор верстатів з програмним керуванням	10000.0	Забезпечення процесу оброблення деталей на ве...
978	2022-05-16T15:13:00	керівник проектів та програм у сфері матеріаль...	7000.0	Знання законодавчих і нормативних правових акт...
1392	2022-05-13T12:33:00	оператор верстатів з програмним керуванням	8500.0	Знання видів та призначення санітарно-технічни...
1546	2022-05-13T10:02:00	оператор верстатів з програмним керуванням	8500.0	Своєчасне і якісне обслуговування клієнтів філ...
1548	2022-05-13T10:02:00	інженер-програміст	8500.0	Своєчасне і якісне обслуговування клієнтів філ...
1985	2022-05-11T15:39:00	інженер з програмного забезпечення комп'ютерів	15000.0	Уміння виконувати операції з базами даних на к...
2756	2022-05-09T16:23:00	інженер-програміст	20000.0	Знання процесу розробки алгоритмів та програм ...
2760	2022-05-09T16:23:00	програміст системний	20000.0	Знання процесу розробки алгоритмів та програм ...
3100	2022-05-06T16:04:00	інженер-програміст	0.0	Умію виконувати операції з приймання, визначен...
3101	2022-05-06T16:04:00	технік-програміст	0.0	Умію виконувати операції з приймання, визначен...

Рисунок 3.4 - Таблиця відфільтрованих даних

У програмній реалізації для цього використовуються бібліотеки *matplotlib* та *seaborn,* що є більш високорівневою API на базі першої, з методами *countplot, despine, barplot, distplot.*

Для початку було створено графік, на якому показано розподіл співвідношення заробітної плати до кількості даних у резюме.

Як видно на рис. 3.5, більшість кандидатів претендують на зарплату в проміжку між 5000 та 8000 грн на місяць.

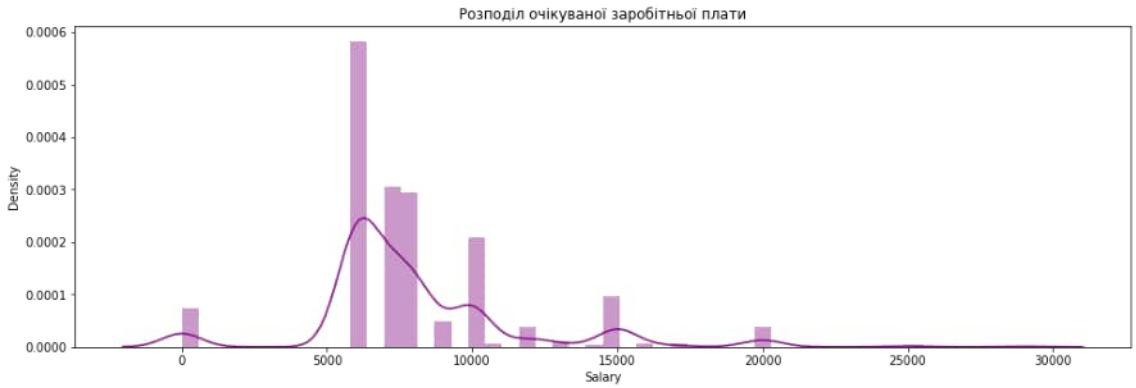

Рисунок 3.5 - Розподіл заробітної плати у наборі даних

Далі я представила візуалізацію професій за допомогою бібліотеки WordCloud, що відображає частоту та важливість кожного слова.

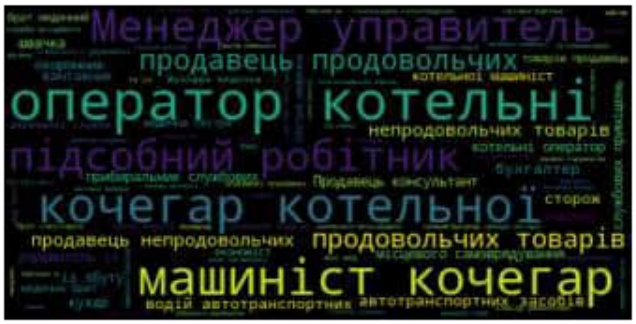

Рисунок 3.6 - Візуалізація професій із використанням WordCloud

Також було здійснено візуалізацію співставлення дати завантаження резюме кандидата з рівнем заробітної плати. На рис. 3.7. можна побачити, що більшість заявом на працевлаштування було подано у будні дні. Заявки, залишені у вихідні дні, характеризуються рівнем бажаної зарплати вищим середнього.

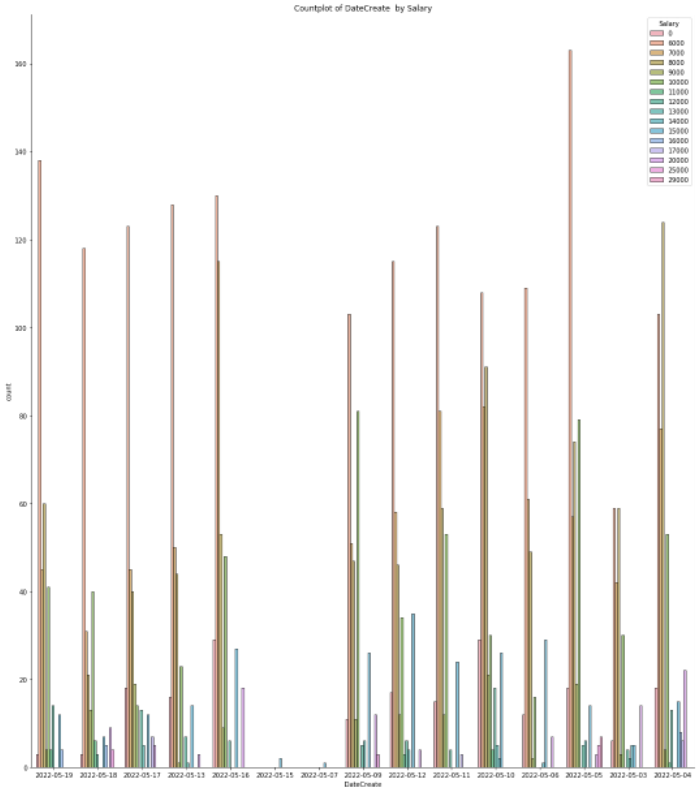

Рисунок 3.7 - Співвідношення заробітної плати до дати завантаження резюме

Аналізуючи набір вхідних даних було побудовано графік відношення 15-ти найпопулярніших професій до їх кількості у корпусі. Бачимо, що понад 200 одиниць даних описують посаду оператора котельні. Популярними також є такі професії, як кухар, вантажник, швачка та бухгалтер.

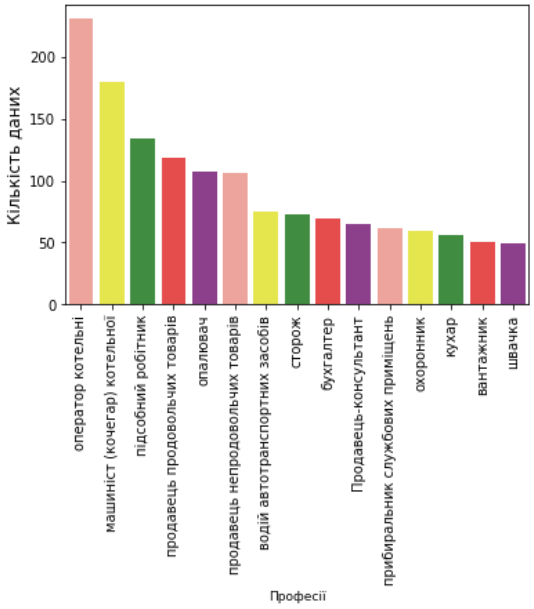

Рисунок 3.8 - Співвідношення між 15 найпопулярнішими професіями у корпусі

Тепер, коли ми проаналізували вхідні дані, варто визначити, які з них можуть найбільше вплинути на професії, які підходять конкретному користувачу.

3.4 Ініціалізація даних моделі та генерування індексу схожості

Ініціалізація даних – це етап попередньої обробки отриманих даних, що складається з двох кроків.

а) Ініціалізація системи, включаючи різні системні змінні. У моєму випадку це включає налаштування системного середовища та пов'язаних з ним функцій, а також завантаження попередньо опрацьованого корпусу даних.

б) На цьому етапі відбувається формування кластерів на основі частоти появи навичок у наборі даних. Алгоритм, який використовується для створення кластера, створює nC_2 комбінації всіх навичок і оновлює глобальний словник навичок, який використовується іншими моделями для генерування кінцевого набору результатів.

На етапі генерування індексу подібності здійснюється обчислення вагових коефіцієнтів подібності між двома наборами даних навичок для визначення взаємозв'язку між ними. Для цього використовується Евклідова відстань та коефіцієнт кореляції Пірсона. Розглянемо кожен з них:

Евклідова відстань отримує два навики – skill1 та skill2, і повертає оцінку подібності у кілька етапів:

а) Знаходження множини перетину для обох навичок.

б) Нормалізація входження навичок, що належать до набору перетину, з коефіцієнтом нормалізації 3.

в) Обчислення за формулою 3.1

$$\sqrt{\Sigma_{i=1}^{n}(x_i - y_i)^2}, \tag{3.1}$$

де $x_i \in X$ і $y_i \in Y$, де X і Y позначають набір навичок, що перетинаються для двох заданих навичок.

Коефіцієнт кореляції Пірсона також отримує два навики – skill1 та skill2, і повертає оцінку подібності у кілька етапів:

а) Знаходження множини перетину для обох навичок.

б) Нормалізація входження навичок, що належать до набору перетину, з коефіцієнтом нормалізації 3.

в) Для двох нормалізованих наборів оцінок X та Y, коефіцієнт Пірсона визначається як

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{n}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{n})((\sum Y^2 - \frac{(\sum Y)^2}{n})}}, \tag{3.2}$$

де n – кількість пар оцінок, $\sum XY$– сума добутку оцінок пар, $\sum X$ – сума оцінок множини X, $\sum Y$– сума оцінок множини Y, $\sum X^2$ – сума квадратів оцінок із множини X, $\sum Y^2$ - сума квадратів оцінок із множини Y.

На етапі фільтрування даних відбувається вилучення певних навичок із набору рекомендацій на основі попередньо визначених обмежень. Ця операція виконується для того, щоб рекомендації не містили шумових даних. У поточній моделі навички з низькою частотою фільтруються та додаються до загального набору, щоб вони не були разом із популярними навичками в кінцевому наборі результатів.

3.4.1 Байєсів персоналізований рейтинг

Для виконання завдання створення рекомендацій було обрано алгоритм наївного Байєсового класифікатора, оскільки він дозволяє легше зрозуміти та обґрунтувати прогнози моделі [9]. Крім того, під час попередніх досліджень було

з’ясовано, що результати, отримані за допомогою цього алгоритму, є конкурентоспроможними щодо інших методів.

В основі процесу побудови Байєсового рейтингу є концепція наївного Байєсового класифікатора для ранжирування кінцевого набору рекомендацій для користувача. Для цього обчислюється умовна ймовірність того, що дві навички зустрічаються разом, щоб визначити їх рейтинг відносно певної навички. Такий підхід гарантує те, що навички, які згадують у посадах, що рекомендуються, залишаються збалансованими щодо загального набору навичок, отриманих із даних користувача. Наступне рівняння використовується для знаходження байєсової оцінки між двома навичками [4]:

$$\text{Score} = -log_2\big(P(j)\big) * P(j|i), \tag{3.3}$$

де $P(j|i) = \frac{Freq\ (ji)}{Freq(i)}$ і $P(j) = \frac{Number\ of\ people\ possessing\ skill\ j}{Totol\ count\ of\ users}$,

$P(j|i)$ – ймовірність того, що навичка j буде в списку користувача, враховуючи, що вже є навичка i,

$Freq(ji)$ – кількість пар j – i в кластері навичок i,

$Freq(i)$ – кількість користувачів, які володіють навичками i.

Оскільки $P(j|i)$ не дорівнюватиме $P(i|j)$ [4], ця модель використовує асиметричну функцію подібності. Одне з обмежень використання такої функції полягає в тому, що кожен елемент *i*, як правило, матиме високі умовні ймовірності у поєднанні з елементами, які часто рекомендуються. В основі цього рішення лежить інверсне масштабування документів, що виконується в системах пошуку інформації.

3.4.2 Класифікація навиків

На етапі прогнозування результатів використовується згаданий раніше алгоритми наївного Байєсового класифікатора.

Для цього у програмі використовується модуль *sklearn.naive_bayes*, який містить клас *GaussianNB* для тренування моделі обраним методом, а також *numpy.linalg* для обчислення Евклідової відстані між навичками користувачів.

Наївний Байєс є одним з ефективних методів, який забезпечує кращий рівень точності та хороші результати після класифікації з використанням статистичного підходу та імовірнісних методів [6].

За допомогою наївного Байєсового класифікатора здійснюється класифікація навиків, згідно з їх частотою у профілі кандидатів.

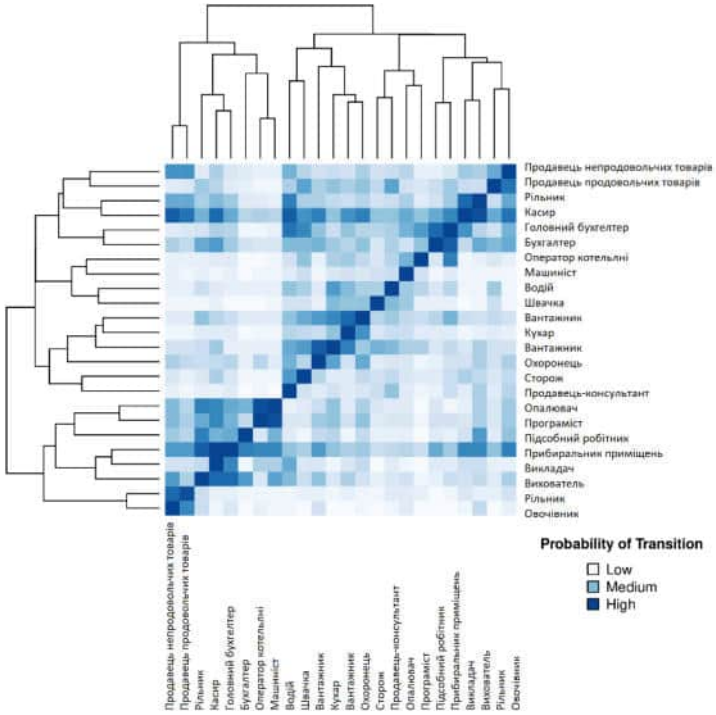

Рисунок 3.9 - Візуалізація карти переходів між професіями

В ході реалізації програми було створено візуалізацію карти переходів, де можна спостерігати 20 професій та їх попарні ймовірності переходу. У цій візуалізації відбуваються переходи від стовпців до рядків, і темно-синій зображує високу ймовірність переходу, а білий — низьку. Хоча перехід на одну і ту саму професію дає найбільші ймовірності (темно-сині діагональні квадрати), зрозуміло, що переходи є асиметричними. Діаграма показує, як схожі професії об’єднуються разом, де існує чітка різниця між послугами та професіями ручної праці.

4 РЕЗУЛЬТАТИ ТА ЇХ АНАЛІЗ

На етапі аналізу отриманих результатів дослідження необхідно оцінити розроблену систему рекомендацій.

4.1 Оцінювання моделі

Для отримання оптимальних результатів необхідно вибрати такий аналіз помилок, результати якого стабілізуються після певного коефіцієнта нормалізації.

Таким чином з рис. 4.1 видно, що відхилення частоти помилок є мінімальним із використанням алгоритму Евклідової відстані.

Рисунок 4.1 - Евклідовий аналіз помилок, аналіз помилок Пірсона

Тут вісь X позначає коефіцієнт нормалізації, а вісь Y - середню квадратичну частоту помилок, яка розраховується із використанням методу *mean_squared_error* із модуля *sklearn.metrics*.

У порівнянні з коефіцієнтом Пірсона, в якому відхилення продовжує зростати зі збільшенням коефіцієнта нормалізації, алгоритм Евклідової відстані, навпаки, дає кращі результати для нашого випадку. З наведених рисунків видно, що графік стабілізується після коефіцієнта нормалізації 3, отже, це оптимальне значення, яке слід обрати для отримання узгоджених результатів.

Байєсові вагові коефіцієнти, як показано на наведених вище рисунках, визначають зважене співвідношення на основі їхніх частот у наборах навичок із даною навичкою. Ці вагові показники використовуються для ранжирування цих навичок у зваженому порядку зменшення.

	Skill1	Skill2	Weightage
0	робота з документами	редагування текстів	53.55
1	робота з документами	юриспуденція	42.89
2	робота з документами	маркетинг	33.17
3	робота з документами	викладання	21.69
4	робота з документами	надання першої медичної допомоги	6.45

Рисунок 4.2 - Таблиця ваг для навику «робота з документами»

Як показано на рис. 4.2, коефіцієнт відношення, з яким вхідний набір даних поділяється на набори навчальних і тестових даних, визначається коефіцієнтом поділу. Набір рекомендацій вважається успішним, якщо в тестовому випадку присутній будь-який із навиків, рекомендованих нашою системою.

Навичка	Division Ratio = 0.2		Division Ratio = 0.25	
	Pass	Fail	Pass	Fail
Робота з документами	0.91	0.09	0.9	0.1
Продаж товарів	0.9	0.1	0.93	0.07
Програмування	0.98	0.02	0.97	0.03
Викладання	0.94	0.06	0.94	0.06
Створення маркетингових стратегій	0.87	0.13	0.82	0.18

Рисунок 4.3 - Точність моделі для набору навиків

Для заданого коефіцієнта поділу тестовий набір отримується шляхом вилучення відповідного коефіцієнта загальних записів із заданого вхідного набору, а решта записів вважається навчальним набором. У нашій моделі тестування набір поділок отримано для двох коефіцієнтів поділу, а саме 0,2 та 0,25. Це було зроблено для того, щоб отримати результати для набору даних різних розмірів, поділених на дані для навчання та тестування моделі. Результати тренування наведені на рис. 4.3.

З аналізу всього вхідного набору ми робимо висновок, що для коефіцієнта розподілу 0,2 середня точність системи для всіх позицій становить 91,33%, а для коефіцієнта поділу 0,25 середня точність системи становить 92,74%.

Зважаючи на наведені вище результати, можна зробити висновок, що розроблена система може успішно рекомендувати вакансії на основі поточного набору навичок користувача, поєднуючи його з аналогічними навичками в глобальному наборі даних, який ми отримали.

4.2 Генерування рекомендацій

На етапі генерування рекомендацій відбувається визначення та отримання рекомендованих вакансій на основі встановлених зв'язків між профілем користувача та вакансіями.

Принцип генерування рекомендацій полягає у виборі користувача, для якого потрібно видати результати та завантаження його опрацьованих даних. Після цього профіль користувача співставляється з усіма профілями характеристик посад почергово, використовуючи техніки, які були розглянуті раніше. Зважаючи на коефіцієнт подібності між даними, формується список 10 вакансій, які найбільше підходять користувачу.

Результат роботи системи зображений на рис. 4.4.

```
resumeIndex = random.randint(0, len(resumes))
selectedUserResume = resumes.iloc[resumeIndex]
selectedUserResume
```

```
DateCreate                              2022-05-11T11:56:00
Position                  оператор комп'ютерного набору
Salary                                              6500
Skills       Знаю законодавчі і нормативні правові акти, ме...
Name: 2128, dtype: object
```

```
recommendJobs(resumeIndex)
```

	RegDate	Position	PositionGroup	Salary	Skills
249	2022-05-16T14:21:49	інженер з комп'ютерних систем	розробники обчислювальних систем	11132.0	Забезпечення працездатності комп'ютерної техні...
352	2022-05-10T17:09:02	адміністратор системи	розробники обчислювальних систем	6500.0	ВАКАНСІЯ АКТУАЛЬНА. Робота на посаді системног...
353	2022-05-10T16:01:52	інженер з комп'ютерних систем	розробники обчислювальних систем	7000.0	Здійснювати проектування, адміністрування та с...
795	2022-04-21T12:58:09	інженер з програмного забезпечення комп'ютерів	розробники обчислювальних систем	14440.0	встановлення та забезпечення програмного забез...
796	2022-04-21T12:54:55	інженер з комп'ютерних систем	розробники обчислювальних систем	13244.0	супровід системного та офісного програмного за...
797	2022-04-21T12:52:26	інженер з комп'ютерних систем	розробники обчислювальних систем	13244.0	супровід системного та офісного програмного за...
1417	2022-03-25T16:21:25	слюсар-електрик з ремонту електроустаткування	електромеханіки та електромонтажники	15000.0	Обов'язки:\r\n- Здійснювати комп'ютерну діагно...
1906	2022-01-21T16:01:00	спеціаліст-бухгалтер	Професіонали державної служби та місцевого сам...	6700.0	Бухгалтер із знанням роботи на комп'ютері, бух...
2293	2021-11-30T10:38:26	адміністратор бази даних	розробники обчислювальних систем	8000.0	головний спеціаліст (з інформаційних технології...

Рисунок 4.4 - Отримані рекомендації

Розроблена система була протестована для генерування результатів для різних користувачів і показала високу точність підбору даних. У більшості випадків вона дорівнювала 79%, при цьому значення F-міри було на рівні 82%, що є хорошим показником.

ВИСНОВКИ

У процесі виконання поставлених на початку магістерської роботи завдань було досліджено та опрацьовано основні алгоритми машинного навчання, які виконують задачі класифікації, та обрано метод наївного Байєсового класифікатора для її розв'язування.

При розв'язуванні поставленої задачі було використано знання основ лінійної алгебри, статистики, вміння працювати з корпусом даних і аналізувати його ознаки, а також практичні навички роботи з бібліотеками мови програмування Python. Під час досліджень було розглянуто основні алгоритми, які застосовуються при класифікації, розглянуто принцип роботи керованих алгоритмів навчання та досліджено переваги на недоліки окремих алгоритмів.

У роботі запропоновано модель, яка, використовуючи дані про досвід кандидатів та професійні навички, може рекомендувати вакансії, внаслідок аналізу та встановлення зв'язків між навичками кандидатів із вхідного набору даних.

Під час комп'ютерної реалізації задачі класифікації було з'ясовано, що відхилення частоти помилок є мінімальним із використанням алгоритму Евклідової відстані у порівнянні з коефіцієнтом кореляції Пірсона; у всіх випадках при прогнозуванні результату була досягнута середня точність системи 92,74%, що є бажаним результатом.

Крім того, було досягнуто мету, визначену на початку, і підтверджено, що що можна з високою точністю класифікувати актуальні вакансії на основі навиків кандидата, що здійснює пошук роботи, використовуючи для цього алгоритми машинного навчання.

У процесі програмної реалізації було проаналізовано набір вхідних даних і досягнуто певних висновків щодо співвідношення даних корпусу. Таким чином, вдалось встановити відповідності між набором професійних навиків та професіями, що їм відповідають, а також визначено, що на точний кінцевий результат найбільше впливають характеристики з простим та лаконічним описом.

Зважаючи на отримані результати, можна стверджувати, що розроблена система може успішно рекомендувати вакансії на основі поточного набору навичок

користувача, поєднуючи його з аналогічними навичками у створеному глобальному наборі даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Клакович К.-Т.Р., Визначення рекомендованого місця роботи на основі аналізу навиків кандидата із використанням машинного навчання // Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 71): Тези доп. Міжнародн. наук. конференц. (18-19 жовтня 2022 р.). – 2022. [Електронний ресурс]. - Режим доступу: http://www.konferenciaonline.org.ua/ua/article/id-677/
2. Курс “Машинне навчання”. [Електронний ресурс]. - Режим доступу: https://courses.prometheus.org.ua/courses/IRF/ML101/2016_T3/course/
3. Avinash Navlani. Support Vector Machines with Scikit-learn. [Electronic resource]. - 2019. – Available from: https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python
4. Carl Dawson. SVM Parameter Tuning. [Electronic resource]. - 2019. – Available from: https://towardsdatascience.com/a-guide-to-svm-parameter-tuning-8bfe6b8a452c
5. Jalaj Thanaki. Python Natural Language Processing: Advanced machine learning and deep learning techniques for natural language processing. - Mumbai, 2017. - 456 c.
6. Jeevankrishna. Job Recommendation System Using Machine Learning And Natural Language Processing. [Electronic resource]. - 2020. – Available from: https://esource.dbs.ie/bitstream/handle/10788/4254/msc_jeevankrishna_2020.pdf?sequence=1&isAllowed=y
7. Julie Yin. Understanding the data splitting functions in scikit-learn. [Electronic resource]. - 2018. – Available from: https://medium.com/@julie.yin/understanding-the-data-splitting-functions-in-scikit-learn-9ae4046fbd26
8. Machine Learning Classification – 8 Algorithms for Data Science Aspirants [Electronic resource]. – Available from: https://data-flair.training/blogs/machine-learning-classification-algorithms/

9. Nikolas Dawson, Mary-Anne Williams, Marian-Andrei Rizoiu. Skill-driven recommendations for job transition pathways. [Electronic resource]. - 2014. – Available from: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0254722
10. Pranav Dar. Popular Machine Learning Applications and Use Cases in our Daily Life. [Electronic resource]. - 2019. – Available from: https://www.analyticsvidhya.com/blog/2019/07/ultimate-list-popular-machine-learning-use-cases/
11. Savita Choudhary. Collaborative job prediction based on Naïve Bayes Classifier using python platform. [Electronic resource]. - 2022. – Available from: https://www.researchgate.net/publication/311611325_Collaborative_job_prediction_based_on_Naive_Bayes_Classifier_using_python_platform/
12. Supervised learning. [Electronic resource]. – Available from: https://en.wikipedia.org/wiki/Supervised_learning
13. Taweh Beysolow II. Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing. - San Francisco, California, USA, 2018. - 150 c.
14. Ventsislav Yordanov. Introduction to Natural Language Processing for Text. [Electronic resource]. - 2018. – Available from: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63
15. Yuyang Ye1, Hengshu Zhu. Identifying High Potential Talent: A Neural Network based Dynamic Social Profiling Approach. [Electronic resource]. - 2019. – Available from: http://staff.ustc.edu.cn/~tongxu/Papers/Yuyang_ICDM19.pdf