Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи обробки даних анкетування за допомогою штучного інтелекту

Виконав:

здобувач 2 року навчання,

групи СПзм-23-1

Роман ФАТІЙ
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: доц. Наталія БОЛОГОВА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет навчально-науковий центр заочної форми навчання

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

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

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Фатію Роману Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи обробки даних анкетування за допомогою штучного інтелекту

затверджена наказом по університету від " 07 " квітня 2025 р. № 53 Стз

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи 1) мова програмування Python 2) бібліотека: Anaconda.

4. Перелік питань, що потрібно опрацювати у роботі

1) аналіз проблеми та огляд існуючих рішень;

2) аналіз існуючих алгоритмів впровадження машинного навчання ;

3) аналіз вимог до створення програми;

4) розробка рішення реалізації програми;

5) програмна реалізація програми;

6) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 14 слайдів

6. Консультанти розділів роботи (заповнюється 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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	10.04.25-15.04.25	
2	Аналіз алгоритмів машинного навчання	16.04.25-20.04.25	
3	Розробка рішення реалізації програми	21.04.25-10.05.25	
4	Програмна реалізація програми	11.05.25-20.05.25	
5	Оформлення матеріалів кваліфікаційної	20.05.25-30.05.25	
	роботи		
6	Подання кваліфікаційної роботи керівникові та	30.05.25-11.06.25	
	її попередній захист		
7	Подання кваліфікаційної роботи на рецензу-	11.06.25-12.06.25	
	вання		

Дата видачі завдання " 7 " квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Наталія БОЛОГОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 49 с., 13 рис., 2 табл., 1 дод.,11 джерел.

ARTIFICIAL INTELLIGENCE, NEURAL NETWORKS, MACHINE LEARNING, PYTHON, ANACONDA.

Метою кваліфікаційної роботи є автоматизація процесу відбору анкетних даних у дистрибутиві Python із використанням алгоритмів машинного навчання.

У ході виконання кваліфікаційної роботи були виконані так завдання:

- вивчено системи штучного інтелекту;
- розглянуто програмне забезпечення для систем штучного інтелекту;
- створино та навчено класифікатор для сортування анкетних даних;
- оцінено ефективність створення проекту.

ABSTRACT

Master’s thesis: 49 pages, 13 figures, 2 tables, 1 appendices, 13 sources.

FIREWALL, GATE, INTERNET, PROTOCOL, ROUTER, SERVER, WI-FI, WIRELESS NETWORK, WLAN.

The major goal of this thesis is the qualification work is to automate the process of selecting questionnaire data in the Python distribution using machine learning algorithms.

- In order to the qualification work, the following tasks were performed:
- artificial intelligence systems were studied;
- software for artificial intelligence systems was reviewed;
- a classifier for sorting questionnaire data was created and trained;
- the effectiveness of the project creation was evaluated.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

НМ – нейронні мережі

ШІ – штучний інтелект

AI – штучний інтелект (англ., Artificial Intelligence)

ЕС – експертна система

SPSS – система статистичного аналізу (англ., Statistical Package for the Social Sciences)

ВСТУП

Тема кваліфікаційної роботи – застосування штучного інтелекту для обробки анкетних даних.

Актуальність теми обумовлена великими трудовитратами та нерелевантними результатами обробки анкетних даних.

Метою роботи є автоматизація процесу відбору анкетних даних у дистрибутиві Python із використанням алгоритмів машинного навчання.

Завдання роботи:

- вивчити системи штучного інтелекту;
- розглянути програмне забезпечення для систем штучного інтелекту;
- створити та навчити класифікатор для сортування анкетних даних;
- оцінити ефективність створення проекту.

Предмет дослідження – автоматизація процесу ранжування анкетних даних щодо релевантності.

У першому розділі розглядається обробка даних із використанням систем штучного інтелекту.

Другий розділ присвячений розробці методики використання систем штучного інтелекту при обробці анкетних даних [1].

У третьому розділі представлені системи штучного інтелекту при збиранні та обробці анкетних даних

Результати роботи: практичним результатом роботи став розроблений класифікатор, який визначає для заповненої анкети: чи буде вона врахована для аналізу ефективності навчального процесу.

1 ОБРОБКА ДАНИХ З ВИКОРИСТАННЯМ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Системи збору та обробки анкетних даних

Зі швидким розвитком штучного інтелекту (ШІ) життєво важливо розробити психометрично обґрунтовані вимірювання ставлення громадськості до цієї технології. Метою цього дослідження було вдосконалення пулу кандидатських пунктів для створення стислого, але надійного переліку для оцінки ставлення до ШІ. Загалом 96 пунктів, взяті з переформульованих шкал, пов'язаних з роботами, та нещодавно розроблених пунктів, що відображають специфічні для ШІ теми, були застосовані до вибірки з 604 дорослих із загальної популяції Сполучених Штатів (віковий діапазон: 18–89 років; 48% чоловіків). Ітеративний аналіз Раша був використаний для зменшення кількості пунктів, забезпечуючи при цьому психометричну стійкість, застосовуючи кілька критеріїв для вибору пунктів, включаючи залишки відповідності, диференціальне функціонування пунктів (DIF) та концептуальну ясність. Отримана шкала, яка отримала назву «Опитувальник ставлення до штучного інтелекту» (AI), складається з двох підшкал з 8 пунктів, що вимірюють позитивне та негативне ставлення до ШІ. Аналіз показав, що ці підшкали є окремими конструктами, а не протилежностями в одному континуумі, і вони лише слабо пов'язані з психологічним дистресом. AI надає стислий, але всебічний показник позитивного та негативного ставлення до ШІ, який можна ефективно застосовувати разом з іншими показниками. Результати підкреслюють багатогранний характер громадського сприйняття ШІ та наголошують на необхідності подальших досліджень профілів та детермінант цього ставлення. Оскільки ШІ продовжує формувати наш світ, AI пропонує цінний інструмент для розуміння та моніторингу громадських настроїв щодо цієї

трансформаційної технології.

Анкета – основний інструмент кількісних досліджень.

Збір даних здійснюється за допомогою методу анкетування, який використовується в рамках певного соціального дослідження та припускає, що цільові групи респондентів самостійно заповнять анкети та повернуть їх інтерв'юеру [2].

Мета анкетування полягає у виявленні різних фактів та тенденцій.

В результаті процесу анкетування збирають великий обсяг даних, що зумовлює необхідність використання відповідних програмних засобів обробки цих даних.

В даний час набирає популярності статистичний програмний комплекс SPSS [4].

Загальна інформація про пакет SPSS та його структуру.

Пакет SPSS (Statistical Package for the Social Sciences – статистичний пакет для соціальних наук) спочатку1 був розроблений як комп'ютерна програма для статистичної обробки даних, призначена для проведення прикладних досліджень у соціальних науках.

Маючи модульну структуру, пакет SPSS забезпечує комплексну статистичну обробку - від планування до управління даними, виконання аналізу та подання результатів. Потужні засоби аналізу та обробки даних з розвиненим графічним інтерфейсом, зручні меню та прості діалогові вікна суттєво спрощують роботу користувача [4].

Основні блоки SPSS:

а) редактор даних – гнучка система, зовні схожа на електронну таблицю, служить для визначення, введення, редагування та перегляду даних;

б) багатовимірні мобільні таблиці – служать для відображення результатів аналізу, дозволяючи досліджувати таблиці, переміщуючи рядки, стовпці та шари та, таким чином, виявляти важливі моменти, які можуть загубитися у стандартних звітах. Забезпечують також порівняння груп,

розщеплюючи таблиці таким чином, щоб щоразу на екран виводилася лише одна група;

в) високоякісна графіка - засіб формування повнокольорових діаграм з високою роздільною здатністю: кругових і лінійчастих діаграм, гістограм, діаграм розсіювання, об'ємних діаграм і багатьох інших [5];

доступ до баз даних - конструктор читання баз даних, що дозволяє легко, кількома натисканнями кнопки миші завантажувати дані з будь-яких джерел;

г) перетворення даних - засіб перетворення даних, що допомагає готувати дані до аналізу: можна легко виділяти підмножини в даних, об'єднувати категорії, додавати, агрегувати, зливати, розщеплювати та транспонувати файли, а також проводити інші перетворення;

ґ) довідкова система комплекс коштів підтримки початківця користувача, що включає [6]:

1) електронний підручник, що містить детальний огляд засобів та можливостей пакета;

2) контекстну довідку, допомагаючу розібратися конкретних завдання при роботі з діалоговими вікнами;

3) спливаючі визначення, що пояснюють статистичні терміни 4) мобільних таблицях;

4) репетитор по статистика, допомагаючий в пошуку необхідною процедури обробки;

5) приклади аналізу, що полегшують інтерпретувати результати в аналогічних типових задачах;

д) командна мова, що забезпечує доступ до додаткових функціональних можливостей пакета, недоступних через меню та діалогові вікна, і дозволяє зберігати та автоматизувати різні повторювані процеси та завдання. Повна документація по командній мові інтегрована до довідкової системи і доступна у вигляді окремого PDF-документа «Посібник із синтаксису», який можна викликати в меню команди «Довідка» [7].

е) передбачені різні варіанти постачання продуктів IBM SPSS Statistics:

є) IBM SPSS Statistics Standart – включає основні аналітичні можливості для вирішення широкого спектру господарських та дослідницьких завдань;

ж) IBM SPSS Statistics Professional – додатково містить засоби, пов'язані із забезпеченням якості даних та їх повноти, а також автоматизації функцій статистики та прогнозування;

з) IBM SPSS Statistics Premium - включає повний набір аналітичних методик, систему моделювання на основі структурних рівнянь (SEM), засоби детальної оцінки та перевірки вибіркових даних, процедури прямого маркетингу;

и) IBM SPSS Statistics for Education (тільки англійською мовою) – включає основні модулі, популярні у навчальному процесі (для студентів, співробітників та викладачів) [7].

1.2 Системи штучного інтелекту

Штучний інтелект це властивість автоматичних обчислювальних систем виконувати окремі функції інтелекту людини, наприклад, вибирати та приймати оптимальні та ефективні рішення на основі отриманого досвіду та раціонального аналізу зовнішніх впливів [8].

Діяльність мозку, яка спрямована на рішення завдань, що потребують розумових зусиль. Інтелект чи мислення безпосередньо пов'язані з рішенням таких завдань, як логічний аналіз, розпізнавання ситуацій, доказ теорем, планування поведінки, а також управління в умовах невизначеності. Характерними рисами інтелекту, які виявляються в процесі вирішення завдань, є наступні: здатність до узагальнення, і акумулювання досвіду (знань і навичок), навчання та адаптації до умов, що змінюються в процесі вирішення завдань. Вирішувати різноманітні завдання завдяки перерахованим вище якостям, а також бути мобільним, легко

перебудовуватися з вирішення одного завдання на інше. Отже, мозок, який наділений інтелектом, є універсальним засобом розв'язання широкого спектра завдань, зокрема неформалізованих, котрим немає заздалегідь відомих методів рішення [6].

Безумовно, можна виключити завдання, які не наважуються стандартними методами, із класу інтелектуальних. Як приклади таких завдань можуть виступати обчислювальні завдання: рішення системи лінійних алгебраїчних рівнянь, чисельне інтегрування диференціальних рівнянь та інше. Вирішенням подібних завдань виступають алгоритми, які є певною послідовністю елементарних. операцій; дана послідовність може бути легкореалізована у вигляді комп'ютерної програми. Хоча для широкого класу інтелектуальних завдань, таких, гра в шахи, як розпізнавання образів, доказ теорем і т.п., навпаки, це формальне розбиття процесу пошуку рішення на окремі елементарні кроки часто виявляється дуже скрутним, навіть якщо саме їхнє рішення є поверховим [8].

Існують дві наукові школи з різними підходами до проблеми штучного інтелекту (ШІ): обчислювальний ШІ та конвенційний ШІ. Обчислювальний ШІ під собою має на увазі ітеративну розробку та навчання. Навчання засноване на емпіричних даних та асоціюється з не- символьним ШІ та нечіткими системами. У конвенційному ШІ переважно використовуються методи машинного самонавчання, засновані на статистичному аналізі та формалізмі. Методи конвенційного ШІ реалізуються в системах та підходах наведених нижче:

- експертні системи: програми, що діють за певними правилами та опрацьовують велику кількість інформації. в результаті своєї роботи видають висновок чи рекомендацію;
- міркування за аналогією;
- байєсовські мережі довіри є імовірнісні моделі - систему з безлічі змінних та їх імовірнісних залежностей;
- поведінковий підхід: модульний метод побудови систем ШІ, система

розбивається на кілька порівняно незалежних програм поведінки, останні ініціюються залежно від змін довкілля [9].

Основні методи обчислювального ШІ:

- нечіткі системи: методики для міркування за умов невизначеності;

- нейронні мережі: симуляція нервової системи, демонструють, зокрема, високі здібності розпізнавання образів;

- еволюційні обчислення: моделі, що використовують поняття природного відбору, що забезпечує відсіювання найменш відповідних згідно із заданим критерієм рішень. у цій групі методів виділяють генетичні алгоритми та мурашиний алгоритм [10].

Експертна система (ЕС) є комп'ютерною програмою, яка здатна виступати в ролі спеціаліста-експерта у вирішенні проблемної ситуації. Ці системи почали розроблятися дослідниками ШІ у 1970-х роках, а у 1980-х набули комерційного поширення. У випадку структура експертної системи може бути виражена наступною схемою (рисунок 1.1).

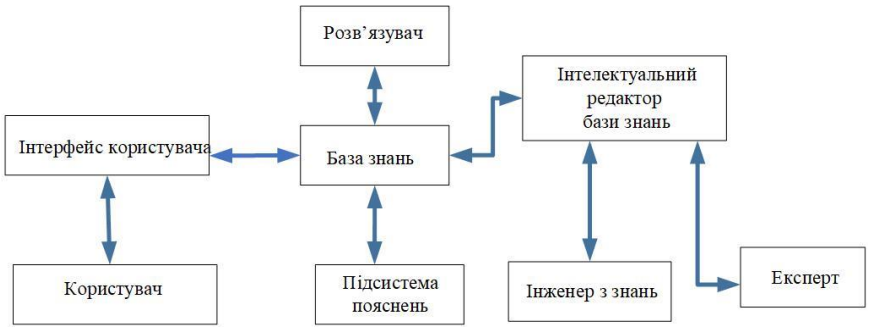

Рисунок 1.1 – Узагальнена структура експертної системи

База знань (БЗ) – головний елемент експертної системи. Вона складається з правил аналізу даних від користувача щодо конкретної проблеми. Блок логічного висновку є програмою, що моделює хід міркувань експерта на підставі знань, що містяться в БЗ.

Підсистема пояснень – це програма, за допомогою якої користувач може отримати відповіді на запитання: «Чому система прийняла те чи інше рішення?» і «Як було отримано ту чи іншу рекомендацію?». Відповідь

питанням «як» – це трасування всього процесу отримання рішення із зазначенням використаних фрагментів БЗ, тобто. всіх кроків ланцюга висновків. Відповідь питанням «чому» – це посилання висновку, яке передує отриманому рішенню, тобто. відхід один крок назад.

ЕС створюється за допомогою інженерів зі знань (аналітиків), які розробляють ядро ЕС і знаючи організацію бази знань, заповнюють її за допомогою експерта зі спеціальності. Інтелектуальний редактор БЗ – програма, що надає інженеру знань можливість створювати БЗ у діалоговому режимі.

Інтерфейс користувача є комплекс програмних засобів, що реалізують діалог користувача з ЕС як для введення інформації, так і для отримання результатів роботи ЕС [11].

Завдання, які вирішуються за допомогою експертних систем, найчастіше відносяться до однієї з наступних областей:

- інтерпретація даних – це одне із традиційних завдань для експертних систем. Під інтерпретацією розуміється визначення змісту даних, результати якого мають бути узгодженими та коректними. Приклади існуючих ЕС: SIAP (виявлення та ідентифікація різних типів океанських суден), АВТАНТЕСТ, МІКРОЛЮШЕР (визначення основних властивостей особистості за результатами психодіагностичного тестування);

- діагностика – це виявлення несправності у певній системі. Трактування несправності як відхилення від норми дозволяє з єдиних теоретичних позицій розглядати і несправність обладнання в технічних системах, захворювання живих організмів, і всілякі природні аномалії. Приклади існуючих ЕС: ANGY (діагностика та терапія звуження коронарних судин), CRIB (діагностика помилок в апаратурі та математичному забезпеченні комп'ютера);

- моніторинг – це безперервна інтерпретація даних у реальному масштабі часу та сигналізація про вихід тих чи інших параметрів за допустимі межі. Головні проблеми – «перепустити тривожну ситуацію» та

інверсне завдання «хибного» спрацьовування. Складність цих проблем полягає у розмитості симптомів тривожних ситуацій та необхідності врахування тимчасового контексту.;

- проектування полягає у підготовці специфікацій на створення «об'єктів» із наперед визначеними властивостями. p align="justify"> Під специфікацією розуміється весь набір необхідних документів - креслення, пояснювальна записка і т.д. Приклади існуючих ЕС: XCON (проектування конфігурацій ЕОМ), SYN (синтез електричних кіл);

- прогнозування – це логічний висновок можливих наслідків із заданих ситуацій. У системі прогнозування зазвичай використовується параметрична динамічна модель, в якій значення параметрів «підганяються» під задану ситуацію. Наслідки, що виводяться з цієї моделі, становлять основу для прогнозів з ймовірнісними оцінками [5]. Приклади існуючих ЕС: WILLARD (пророцтво погоди), PLANT (оцінки майбутнього врожаю), ECON (економічні прогнози);

- планування – знаходження планів дій, які стосуються об'єктів, здатних виконувати деякі функції. У таких ЕС використовуються моделі поведінки реальних об'єктів для того, щоб логічно вивести наслідки планованої діяльності. Приклади існуючих ЕС: STRIPS (планування поведінки робота), ISIS (планування промислових замовлень), MOLGFN (планування експерименту);

- навчання – процес діагностування помилки щодо будь-якої дисципліни з допомогою комп'ютера і підказують правильні рішення. Вони акумулюють знання про гіпотетичне «учні» та його характерні помилки, а потім у ході роботи здатні діагностувати слабкості у знаннях учнів і знаходити відповідні засоби їх ліквідації. Крім того, вони планують процес спілкування з учнем залежно від успіхів учня з метою передачі знань. Приклади існуючих ЕС: PROUST (навчання мови програмування Pascal) [3].

Міркування за аналогією. CBR-системи є реалізацією методології ШІ, яку застосовують при побудові комп'ютеризованих консультаційних систем,

що базуються на накопиченому досвіді. На відміну від класичних експертних систем, що діють на основі логічних правил, CBR-системи зберігають успішні рішення низки реальних проблем, такі рішення називають прикладами або прецедентами і з появою нової проблеми знаходять за певним алгоритмом (часто за допомогою машини логічного висновку, з кількісною оцінкою) найбільш підходящі (подібні) прецеденти, після чого пропонує відповідно змінену комбінацію їх рішень. Якщо нову проблему вдається вирішити у такий спосіб, це рішення заноситься у основу прецедентів підвищення ефективності роботи системи у майбутньому. Головний недолік CBR-систем у тому, що вони створюють моделей чи правил, які узагальнюють накопичений досвід [7].

Байєсівські мережі довіри. Як було сказано вище, байєсовська мережа - це імовірнісна модель, що представляє собою безліч змінних та їх імовірнісних залежностей. Наприклад, байєсовська мережа може бути використана для обчислення ймовірності того, на що хворий пацієнт за наявністю або відсутністю ряду симптомів, ґрунтуючись на даних про залежність між симптомами та хворобами. Існують ефективні методи, які використовуються для обчислень та навчання байєсівських мереж. Байєсовські мережі використовуються для моделювання в біоінформатиці (генетичні мережі, структура білків), медицині, класифікації документів, обробці зображень, обробці даних та системах прийняття рішень [7].

Нейронні мережі. Нейронна мережа (НМ) –- це розподілений паралельний процесор, що складається з елементарних одиниць обробки інформації, які накопичують експериментальні знання та надають їх для подальшої обробки [8]. Вона являє собою діючу модель нервової системи і подібна до мозку з двох точок зору:

- знання надходять у нейронну мережу з навколишнього середовища та використовуються у процесі навчання;
- для накопичення знань застосовуються зв'язки між нейронами, які називаються синаптичними вагами.

Структурна схема одиничного нейрона представлена рисунку 1.2.

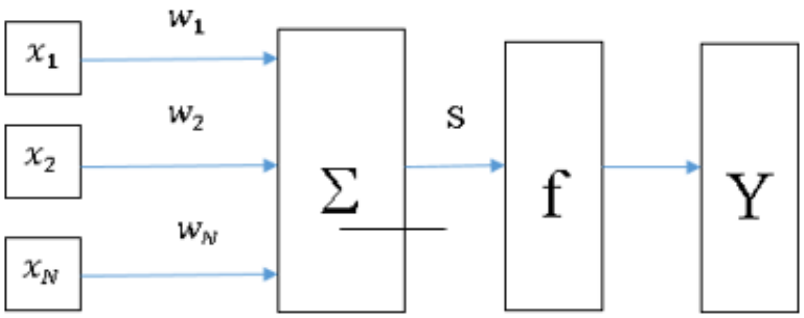

Рисунок 1.2 – Структурна схема нейрона

Сигнали x_i, які надходять на вхід нейрона, множаться на відповідні вагові коефіцієнти w_i, а потім підсумовуються. Результат підсумовування надходить на нелінійний перетворювач, що реалізує деяку нелінійну функцію. Остання називається функцією активації або функцією передавання нейрона: результат її дії надходить на вихід нейрона. Використовуються безліч способів побудови нейронних мереж окремих нейронів. Шарувата архітектура НМ є найпоширенішою, її узагальнена схема представлена рисунку 1.3. Сигнали вхідного шару надходять на входи нейронів першого шару, після проходження якого поширюються далі, доки не досягнуть виходів нейронної мережі. Для навчання нейронних мереж такого типу використовується, як правило, так званий алгоритм зворотного розповсюдження помилки, за допомогою якого розраховують зміни вагових коефіцієнтів, необхідних для узгодження вихідних значень з вибіркою зразків [9]. Використання нейронних мереж забезпечує наступні корисні властивості систем:

- нелінійність. Ця якість нейронної мережі особливо важлива у тому випадку, якщо сам фізичний механізм, який відповідає за формування вхідного сигналу, є нелінійним (наприклад, людська мова);
- адаптивність. Нейронні мережі мають здатність адаптувати свої у

нестаціонарному середовищі створюють нейронні мережі, які змінюють синаптичні ваги у реальному часі.

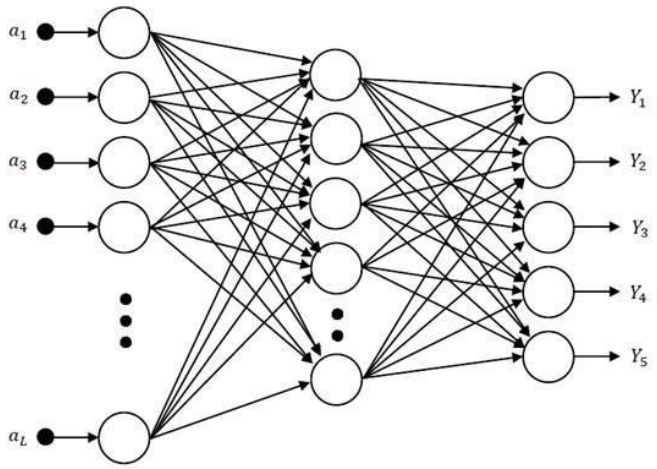

Рисунок 1.3 – Структура шаруватої нейронної мережі

Контекстна інформація Знання представляються у самій структурі нейронної мережі. Кожен нейрон мережі потенційно може бути схильний до впливу інших її нейронів.

Відмовостійкість. Мережі, які реалізовані апаратно, потенційно стійкі до відмов. Це означає, що за несприятливих умов їхня продуктивність зменшується незначно. Наприклад, якщо пошкоджено якийсь нейрон або його зв'язок, витягання запам'ятованої інформації не може. Однак, якщо брати до уваги розподілений характер зберігання інформації в нейронній мережі, то можна стверджувати, що тільки серйозні ушкодження структури нейронної мережі можуть суттєво вплинути на її працездатність. Нижче наведено деякі проблеми, які вирішуються застосуванням нейронних мереж:

Класифікація образів. Завдання полягає в тому, щоб вказати приналежності вхідного образу, який представлений набором ознак, одного або кількох попередньо визначених класів. До відомих додатків відносяться розпізнавання букв.

2 РОЗРОБКА МЕТОДИКИ ВИКОРИСТАННЯ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ ПРИ ОБРОБЦІ АНКЕТНИХ ДАНИХ

2.1 Застосування ШІ

Машинне навчання (machine learning) – це один із ключових напрямів штучного інтелекту, що базується на використанні методів математичної статистики, теорії ймовірностей, чисельних методів оптимізації, дискретного аналізу та дослідженні структурованих закономірностей у даних. На сьогоднішній день машинне навчання активно впроваджується у різноманітні галузі: біоінформатика, медицина (зокрема медична діагностика), геологія та геофізика, соціологія, економіка (включно з кредитним скорингом, прогнозуванням відтоку клієнтів, виявленням шахрайства, біржовим аналізом), техніка (розпізнавання мови), офісна автоматизація (розпізнавання тексту, класифікація документів, фільтрація спаму, обробка рукописного вводу тощо).

Сфера використання цих методів динамічно зростає. Інформатизація суспільства призводить до генерації великих обсягів даних у таких секторах як наука, промисловість, бізнес, транспорт, охорона здоров’я, внаслідок чого виникають нові виклики у сфері прогнозування, керування та прийняття рішень. Часто ці завдання вирішуються за допомогою навчання на прецедентах, тобто з використанням вже наявних прикладів. У минулому, за відсутності таких даних, ці задачі не ставилися або вирішувалися іншими, менш ефективними методами.

Системи, що базуються на машинному навчанні, на відміну від класичних алгоритмічних підходів, не потребують чітко прописаних правил – вони самостійно формують залежності на основі вхідних даних. Це забезпечує підвищену ефективність обробки текстової, графічної чи

голосової інформації. Наприклад, у сфері кадрового менеджменту (HR) такі системи можуть використовуватись для автоматизованого аналізу резюме, відповідей на типові запити, а також для створення інтелектуальних асистентів.

Одним із базових завдань машинного навчання є класифікація – це процес встановлення приналежності об’єкта до одного з наперед заданих класів. Якщо існує навчальна вибірка, тобто набір об’єктів із відомою класифікацією, то можна побудувати алгоритм (класифікатор), здатний прогнозувати належність нових об’єктів до певного класу.

У контексті даного дослідження, застосування машинного навчання полягає в автоматичному визначенні релевантності студентських анкет, надісланих у межах анкетування. Створений класифікатор, базуючись на відомих відповідях із попередньої вибірки, виконує класифікацію нових анкет – визначає, чи доцільно враховувати конкретну анкету під час оцінювання ефективності освітнього процесу.

У подальшому буде розглянуто можливість побудови такого алгоритму класифікації, який реалізує дану задачу з максимальною точністю.

Анкетування передбачає п’ять запитань, кожне з яких оцінюється респондентом за десятибальною шкалою. Збір відповідей здійснюється за допомогою платформи Google Forms, внаслідок чого була сформована об’ємна база даних.

Для опису властивостей кожної анкети було визначено такі характеристики:

- відсутність деяких відповідей;
- наявність монотонних відповідей у висхідному або низхідному порядку;
- повторювані значення відповідей;
- присутність крайніх значень (1 або 10).

Для моделювання та подальшого аналізу застосовується мова програмування високого рівня Python, яка дозволяє ефективно реалізовувати

обробку даних і має зручний синтаксис для роботи з таблицями.

На початковому етапі виконано експорт даних з Google Forms у формат Excel. Наступним кроком стало формування навчального набору, розподіленого мною – як експертом – на дві категорії: «інформативні» (корисні) та «некорисні» (марні) анкети. До останніх віднесено анкети, у яких:

- кількість запитань менша за п’ять;
- усі відповіді однакові;
- відповіді впорядковані за зростанням або спаданням з кроком 1;
- присутні лише граничні оцінки.

У контексті машинного навчання ознаки – це формалізовані характеристики об’єкта, які дозволяють комп’ютеру навчатися на основі даних.

Зі швидким розвитком технологій штучного інтелекту, безумовно, можна очікувати зростання інтересу до вимірювання ставлення до роботи зі штучним інтелектом. Хоча вже розроблено кілька інструментів, AIAI можна описати як такий, що має чіткі переваги. Це включає той факт, що AIAI спирався на широкий спектр елементів, включаючи ті, що спеціально розроблені для контексту штучного інтелекту, а не лише на елементи, адаптовані з існуючих шкал ставлення роботів або тривожності (наприклад, NAAIS; Persson et al.,2021 рік). На відміну від деяких дуже коротких показників, таких як 4-пунктний AIAS-4 (Grassini,2023 рік) та 5-пунктовий ATAI (Sindermann et al.,2021 рік), AIAI оцінює ширший діапазон позитивного та негативного ставлення за допомогою двох підшкал по 8 пунктів кожна. Хоча AIAI структурно подібний до GAIS (Schepman & Rodway,2022 рік), він містить деякі пункти, які є неспецифічними та широкими (наприклад, «Штучний інтелект захоплює»), що пропонують обмежену інформативну цінність щодо аспектів, що сприяють позитивному чи негативному ставленню. Більше того, деякі пункти в GAIS можуть більше схилятися до вимірювання тривожності, ніж суто до аспектів ставлення (наприклад, «Я

тремчу від дискомфорту, коли думаю про майбутнє використання штучного інтелекту»), що ризикує спотворити вимірювання ставлення.реакціями (Браун та ін.,2015 рік). Розробка AIAI мінімізувала використання мови, яка стосується сильних емоційних реакцій. Таким чином, дослідники можуть досліджувати зв'язки між ставленням ШІ та психологічними змінними, такими як психологічний дистрес, без ризику хибних кореляцій, що виникають через перекриття змісту в інструментах вимірювання.

При порівнянні психометричних властивостей AIAI з іншими опублікованими шкалами ставлення на основі штучного інтелекту (ШІ) можна виявити кілька сильних сторін. Ретельна розробка AI за допомогою аналізу Раша представляє методологічну перевагу над іншими шкалами, які зазвичай спиралися виключно на підходи класичної теорії тестування. Цей сучасний психометричний підхід дозволив ретельно дослідити функціонування завдань, що є аналізом, не представленим для інших вимірювань ставлення на основі ШІ. Крім того, AI чітко продемонстрував, що позитивне та негативне ставлення являють собою різні конструкції, а не протилежні кінці континууму, що не досліджується в більшості інших вимірювань. AI також унікальний своєю підтвердженою відмінністю від тривожності, зі слабкою кореляцією зі змінними психологічного дистресу, що забезпечує докази дискримінантної валідності.

Це дослідження має кілька психометричних обмежень, які слід враховувати під час інтерпретації результатів. Хоча AI продемонстрував чудову внутрішню узгодженість (шкала Кронбахаα і ω Макдональда 0,93 для позитивної підшкали та 0,91 для негативної підшкали), часова стабільність не оцінювалася. Відсутність даних про надійність повторного тестування обмежує наше розуміння того, чи відображають бали AI стабільні риси, чи вони чутливі до ситуаційних впливів. Наш підхід до валідації в першу чергу встановив структурну валідність за допомогою аналізу Раша. Однак кілька важливих аспектів валідності залишаються нерозглянутими. Ми не досліджували зв'язки з поведінковими критеріями, пов'язаними з прийняттям

або опором ШІ (валідність критерію), порівняння з іншими встановленими показниками ставлення до ШІ не проводилися (конвергентна валідність), а зв'язки (рисунок 2.1).

```
<w:fonts>
    <w:font w:name="Arial">
        <w:panose1 w:val="020B0604020202020204" />
        <w:charset w:val="00" />
        <w:family w:val="swiss" />
        <w:pitch w:val="variable" />
        <w:sig w:usb0="20002A87" w:usb1="00000000"
            w:usb2="00000000" w:usb3="00000000"
            w:csb0="000001FF" w:csb1="00000000" />
    </w:font>
    <. . .
</w:fonts>
```

Рисунок 2.1 – Приклад шрифту

Обробка вхідних даних відбувається за певними критеріями, після чого повертається результат аналізу. Сервіс Data prediction відповідає за прогнозування, використовуючи штучні нейронні мережі для аналізу даних. Крім виведення результатів у JSON-форматі, система також генерує графіки для наочності, що реалізовано в сервісі Data visualization. Користувач може оцінити результати обробки, аналізуючи різні типи візуалізацій.

Сервіс Security-manager забезпечує управління користувачами та їхніми правами доступу до функціоналу системи. Усі дані, включаючи вхідні параметри, результати роботи та інформацію про користувачів, зберігаються в базі даних.

Як основу обрано реляційну модель бази даних, яка відзначається надійністю, структурованим зберіганням даних у вигляді таблиць та використанням SQL для запитів. Для фізичної реалізації обрано СУБД PostgreSQL через її підтримку BLOB-даних, JSON-полів, безкоштовність та можливість ефективної роботи з різними типами файлів під час аналізу даних..

Метод графового аналізу тексту передбачає представлення текстової інформації у вигляді структурованого графа, де окремі елементи (слова, фрази чи речення) виступають вузлами, а зв'язки між ними відображають семантичні та контекстуальні взаємозв'язки. Цей підхід дозволяє не лише ідентифікувати ключові концепції, але й виявляти складні відносини між ними, формуючи цілісну мережу тем.

Таке графічне представлення спрощує інтерпретацію результатів, дозволяючи аналітику оцінити як локальні залежності, так і загальну структуру тексту. Запропонований метод значно підвищує глибину та об'єктивність аналізу, пропонуючи системний інструмент для дослідження текстових даних [7].

Перший етап алгоритму TopicRank передбачає ідентифікацію основних тем документа та фраз-кандидатів, які їх репрезентують. Для цього застосовується спеціальний алгоритм, розроблений на основі методики Wan та Xiao (2008). Відповідно до цього підходу, потенційні ключові фрази визначаються шляхом виділення найдовших послідовностей іменників та прикметників у тексті.

Альтернативні методи, такі як використання синтаксично відфільтрованих n-грам, також можуть застосовуватися для виявлення статистично значимих словосполучень. Однак такі підходи мають певні недоліки.

Довші послідовності іменників забезпечують більш повний контекст порівняно з обмеженими n-грамами, які часто не охоплюють необхідний рівень деталізації.

Вони менш точно відповідають граматичній структурі тексту, що може знижувати ефективність визначення тем.

TopicRank обрано саме через його здатність оптимально поєднувати широке охоплення контексту з точністю визначення релевантних фраз. Це забезпечує глибокий аналіз тексту, що є ключовим для якісного автоматизованого конспектування.

Графова модель алгоритму TopicRank представляє документ у вигляді повного графа, де вершини відповідають окремим темам, а ребра з ваговими коефіцієнтами відображають семантичні зв'язки між ними. Ваги ребер кількісно характеризують силу та глибину взаємозв'язків між темами, формуючи структуроване представлення текстового контенту.

Для оцінки значимості тем алгоритм застосовує метод TextRank, який реалізує ітеративний процес ранжування на основі принципу "важливість через зв'язки". Суть методу полягає в тому, що значущість теми визначається:

- вагою вхідних зв'язків;
- важливістю суміжних тем, з якими вона пов'язана.

Процес ранжування відбувається ітеративно до досягнення стабільного стану, коли оцінки важливості тем перестають суттєво змінюватись. Такий підхід забезпечує:

- об'єктивне ранжування тем за ступенем важливості;
- врахування як прямих, так і опосередкованих семантичних зв'язків;
- можливість аналізу складних текстових структур.

Результатом роботи алгоритму є ієрархічно впорядкований набір тем, що дозволяє ефективно вирішувати задачі автоматизованого аналізу та узагальнення текстових даних. Особливу цінність цей метод представляє для обробки великих масивів інформації, де ручний аналіз є трудомістким або практично неможливим.

Попередня обробка тексту для трансформерних моделей (наприклад, T5).

Попередня обробка є критичним етапом підготовки тексту для моделей Text-To-Text, таких як T5. Вона включає:

- токенізацію – розбиття тексту на токени (слова, підслова або символи) для подальшої обробки;
- очищення та нормалізацію – видалення зайвих символів, пунктуації, приведення до нижнього регістру для уніфікації даних;

- видалення стоп-слів – фільтрація неінформативних слів (наприклад, сполучників) для зменшення шуму;

- векторизацію – перетворення токенів у числові вектори (word embeddings) для подальшого аналізу моделлю.

Мета цих кроків – перетворити текст на структурований формат, придатний для обробки T5, усунувши зайві дані та підвищивши точність аналізу: кодування, декодування та пост-обробка в T;5.

Кодування. перетворює текст у векторні представлення вбудовування токенів – кожне слово отримує числовий вектор, що відображає його семантику. Позиційне кодування – додавання інформації про порядок слів (оскільки трансформери не мають вбудованого розуміння послідовності). Формування вхідної матриці – об'єднання вбудовувань і позиційних даних для подальшої обробки.

Декодування. Генерація тексту на основі закодованих даних:

- автогресія – послідовне створення токенів (кожен наступний залежить від попередніх);

- механізми уваги – визначення найважливіших частин тексту для точнішої генерації;

- критерії завершення – зупинка після спецтокена або досягнення максимальної довжини.

Пост-обробка. Покращення згенерованого тексту:

- виправлення граматичних помилок;
- видалення зайвих символів або токенів;
- оптимізація структури та стилю;
- перевірка логічної узгодженості;
- комбінація TopicRank і T5 для підсумовування.

TopicRank виділяє ключові теми та фрази, використовуючи графовий аналіз. T5 отримує ці ключові елементи та генерує абстрактивний підсумок, перефразовуючи зміст оригіналу.

Переваги:

- точність – TopicRank допомагає T5 зосередитися на найважливішому;
- стисліcть – підсумки коротші, але зберігають суть;
- автоматизація – ефективна обробка великих текстів (новини, наукові статті, документи).

Цей підхід особливо корисний у сферах, де важливі якість і швидкість обробки інформації.

3 ЗАСТОСУВАННЯ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ ПРИ ЗБОРІ ТА ОБРОБЦІ АНКЕТНИХ ДАНИХ

3.1 Машинне навчання як підрозділ штучного інтелекту

Машинне навчання (англ. machine learning) – великий підрозділ штучного інтелекту, математична дисципліна, що використовує розділи математичної статистики, чисельних методів оптимізації, теорії ймовірностей, дискретного аналізу та закономірності даних. Машинне навчання вже знайшло застосування в наступних областях: в біоінформатиці, в медицині (медична діагностика), в геології та геофізиці, в соціології, в економіці (кредитний скоринг, передбачення відтоку клієнтів, виявлення шахрайства, біржовий технічний аналіз, біржовий нагляд), в техніці (розпізнавання мови), в офісній автоматизації (розпізнавання тексту, виявлення спаму, категоризація документів, розпізнавання рукописного введення). Сфера застосування машинного навчання постійно розширюється. Повсюдна інформатизація призводить до накопичення великих обсягів даних у науці, виробництві, бізнесі, транспорті, охороні здоров'я. Виникають у своїй завдання прогнозування, управління та прийняття рішень часто зводяться навчання по прецедентам. Раніше, коли таких даних не було, ці завдання або взагалі не ставилися або вирішувалися зовсім іншими методами.

Системи з машинним навчанням та технологіями штучного інтелекту припускають, що програма проводить обробку даних не за заданими правилами, а створює ці правила самі. Такі системи допомагають краще розпізнавати тексти, зображення та голосові команди.

Тому в сфері рекрутингу вони дозволяють, наприклад, краще сканувати резюме та описи вакансій, а також створювати програми-асистенти, які

відповідатимуть на поширені запитання та надаватиме початкову інформацію. Завдання класифікації – це визначення приладдя об'єктаспостереження до певного класу. За наявності безлічі об'єктів, які вже відома приналежність до класів, можна побудувати алгоритм, який зможе визначати, якого з класів належатиме новий об'єкт. У термінах машинного навчання об'єкти з відомою приналежністю до класам називаються навчальною вибіркою, а завдання визначення приналежності нового об'єкта до класу – класифікацією.

Таким чином, застосування машинного навчання для відбору відповідних анкет студентів полягає в наступному: за раніше зібраними відповідями на питання використаний алгоритм, який визначає для анкети, що знову надійшла: чи буде вона врахована для аналізу ефективності навчального процесу. Розглянемо можливість побудови такого алгоритму [11].

3.2 Системи збору і обробки даних

Анкета є набором з 5 питань, на кожне з яких можна відповісти за шкалою від 1 до 5. Опитування студенти проходять у google forms. Зібрано велику інформаційну базу даних із відповідями питання.

Для характеристики анкети було виділено такі ознаки:

- наявність пропущених відповідей;
- наявність відповідей по порядку (низхідний/висхідний порядок);
- анкета містить однакові відповіді;
- наявність граничних відповідей в анкеті.

Для побудови моделей і аналізу даних використовується Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника та читання коду.

На першому етапі імпортуємо дані з google forms до Excel. Далі необхідно створити навчальну вибірку, яка складена мною як експертом,

тобто я розділила всі анкети на 2 групи: корисні та марні. Під марними анкетами розуміємо анкети, ознаки яких приймають такі значення:

- кількість запитань менше п'яти;
- відповіді на всі питання однакові;
- відповіді (числа) йдуть у порядку зростання/зменшення з кроком 1;
- пропуски або нерелевантні відповіді;
- в анкеті присутні максимальні та мінімальні числа (відповіді).

У машинному навчанні ознаки описують об'єкт у доступній та зрозумілій для комп'ютера формі.

Наступним етапом імпортуємо бібліотеки pandas для зручної роботи з даними та numpy для роботи з числами (рисунок 3.1).

```
# Імпорт бібліотек для обробки даних
import pandas as pd  # бібліотека для зчитування та аналізу табличних даних
import numpy as np   # бібліотека для роботи з числовими масивами

# Вивід підтвердження імпорту
print("Бібліотеки pandas та numpy успішно імпортовано.")
```

Рисунок 3.1 – Імпорт бібліотек

Далі зчитуємо корисні (рисунок 3.2) та марні анкети (рисунок 3.3) за допомогою бібліотеки pandas.

```
import pandas as pd

# Зчитування корисних анкет з файлу
korisni_ankety = pd.read_csv('korisni_ankety.csv')
print("Корисні анкети:")
print(korisni_ankety.head())

# Зчитування марних анкет з файлу
marni_ankety = pd.read_csv('marni_ankety.csv')
print("\nМарні анкети:")
print(marni_ankety.head())
```

Рисунок 3.2 –Зчитування корисних моделей

Анкета набуває значення 1, якщо вона задовольняє одну з чотирьох ознак марності (рисунок 3.3)

```python
import pandas as pd

# Зчитування корисних анкет з файлу
korisni_ankety = pd.read_csv('korisni_ankety.csv')
print("Корисні анкети:")
print(korisni_ankety.head())

# Зчитування марних анкет з файлу
marni_ankety = pd.read_csv('marni_ankety.csv')
print("\nМарні анкети:")
print(marni_ankety.head())
```

Рисунок 3.3 – Завдання ознак марності

Приклад реалізації логістичної регресії в Python для задачі класифікації анкет на корисні та марні за допомогою бібліотеки scikit-learn (рисунок 3.4).

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

# X - ознаки (features), y - мітки (labels)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Створення моделі логістичної регресії
model = LogisticRegression()

# Навчання моделі
model.fit(X_train, y_train)

# Прогнозування результатів на тестовій вибірці
y_pred = model.predict(X_test)

# Обчислення точності класифікації
accuracy = accuracy_score(y_test, y_pred)
print("Точність логістичної регресії:", accuracy)
```

Рисунок 3.4 – Реалізація логістичної регресії

Приклад реалізації алгоритму вирішального дерева для класифікації анкет у Python із використанням бібліотеки scikit-lear (рисунок 3.5)

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

# Розділення на тренувальні та тестові вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Ініціалізація моделі вирішального дерева
tree_model = DecisionTreeClassifier(random_state=42)

# Навчання моделі
tree_model.fit(X_train, y_train)

# Передбачення результатів
y_pred = tree_model.predict(X_test)

# Оцінка точності класифікації
accuracy = accuracy_score(y_test, y_pred)
print("Точність вирішального дерева:", accuracy)
```

Рисунок 3.5 – Реалізація алгоритму вирішального дерева

Приклад реалізації алгоритму «Випадковий ліс» (Random Forest) у Python (рисунок 3.6).

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

# Розділення даних на тренувальні та тестові вибірки
X
```

Рисунок 3.6 – Реалізація алгоритму «Випадковий ліс

Для обробки анкетних даних із застосуванням штучного інтелекту (машинного навчання) вибір методу залежить від особливостей самих даних і цілей аналізу. Ось ключові моменти, які допоможуть зрозуміти, що краще врахувати перед вибором методу (таблиця 3.1):

- розмір та структура даних: скільки анкет, які типи питань (числові, категоріальні, текстові);
- якість даних: чи є пропуски, шум, неоднорідність;
- мета аналізу: класифікація (наприклад, виявлення категорії

респондента), прогнозування, сегментація;

- інтерпретованість: чи важливо зрозуміти, чому модель приймає такі рішення.

Таблиця 3.1 – Аналіз вибору методу

Ситуація	Рекомендований метод	Чому саме він?
Невеликі або середні набори з чіткими категоріями відповіді	Логістична регресія	Проста, добре інтерпретується, дає ймовірності, легко впроваджується
Потрібна зрозуміла логіка рішень, наприклад, ієрархічна класифікація	Вирішальні дерева	Інтуїтивно зрозумілі, можна легко візуалізувати, пояснюють процес прийняття рішення
Великі обсяги даних, складні взаємозв’язки між ознаками	Випадковий ліс (ансамбль дерев)	Висока точність, стійкість до шуму, добре працює з різнорідними даними
Дуже великі обсяги та швидкість навчання	Стохастичний градієнтний спуск (SGD)	Ефективний для великих даних, гнучкий, використовується для різних моделей
Малий обсяг, простота, відсутність навчання	Метод K найближчих сусідів (KNN)	Легкий у реалізації, не вимагає навчання, але не підходить для великих даних
Потрібно виявити приховані структури або сегменти	Кластеризація (наприклад, K-means)	Добре для пошуку груп однорідних респондентів без заданих міток

ХНУРЕ – один із найбільших технічних університетів України, який щорічно навчає понад 8 000 студентів та проводить численні дослідження й опитування. Для ефективної обробки великої кількості різнорідних анкетних даних (від оцінок до текстових відгуків) пропонуються такі рішення (таблиця 3.2):

- великий обсяг анкетних даних – тисячі і навіть десятки тисяч записів;
- дані можуть бути різного типу – числові, категоріальні, можливо текстові відповіді;
- потрібно не лише класифікувати, а й отримувати інсайти для прийняття рішень;
- важлива автоматизація, масштабованість та надійність.

Таблиця 3.2 – Вибір методу машинного навчання

Метод	Чому підходить для ХНУРЕ
Випадковий ліс	Потужний і стабільний, працює з великими даними. Стійкий до шуму
Логістична регресія	Простий, швидкий. Добре для базових класифікацій і пояснень.
Градієнтний бустинг (LightGBM, XGBoost)	Підходить для великого обсягу. Але складніше у налаштуванні і важче інтерпретувати.
Стохастичний градієнтний спуск	Швидкий для дуже великих даних.
Метод KNN	Не рекомендується через обсяг даних та складність масштабування. Можна використовувати тільки для невеликих підмножин або прототипів.

Для ХНУРЕ із великим обсягом анкетних даних найкраще підходять ансамблеві методи машинного навчання (випадковий ліс, бустинг), оскільки вони поєднують точність, стійкість до шуму та працюють із різними типами даних.

Приклад коду на Python із використанням Random Forest (Випадковий ліс) для обробки анкетних даних. Цей приклад демонструє основні етапи: завантаження даних, попередню обробку, навчання моделі (рисунок 3.7).

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, accuracy_score

# Припустимо, у вас є CSV-файл з анкетними даними
# df = pd.read_csv('anketa_data.csv')

# Приклад створення тестового датасету (замініть на свій файл)
data = {
    'age': [20, 21, 22, 23, 24, 25],
    'gender': ['M', 'F', 'F', 'M', 'M', 'F'],
    'education': ['Bachelor', 'Master', 'Bachelor', 'PhD', 'Bachelor', 'Master'],
    'response_score': [7, 8, 5, 6, 9, 4],
    'target': [0, 1, 0, 1, 1, 0]  # Клас для класифікації (наприклад, задоволеність)
}

df = pd.DataFrame(data)

# Обробка категоріальних ознак (Label Encoding)
for col in ['gender', 'education']:
```

Рисунок 3.7 – Приклад коду на Python із використанням Random Forest

Код завантажує або створює датафрейм з анкетними даними, кодує категоріальні ознаки (стать, освіта) в числовий формат, розбиває дані на навчальну і тестову вибірки, навчає модель Random Forest для класифікації, оцінює якість класифікації на тестових даних.

Вибірка складається зі 100 респондентів. З них 30 осіб оцінили якість освіти у 3 бали, 40 – у 4 бали, 25 – у 5 балів. У 3 анкетах відповідь на це питання була відсутня (missing values). Розрахувати середнє значення оцінки якості освіти на основі валідних даних, а також провести імпутацію пропущених значень середнім і повторно обчислити середнє значення (рисунок 3.8).

```python
import numpy as np
import pandas as pd

# Вхідні дані
# Кількість студентів за оцінками
data = {
    'score': [3, 4, None, 5],   # None - пропуски
    'count': [30, 40, 3, 25]
}

# Створюємо датафрейм
df = pd.DataFrame(data)

# Загальна кількість анкет
total_surveys = df['count'].sum()

# Кількість валідних анкет (без пропусків)
valid_surveys = df[df['score'].notnull()]['count'].sum()

# Обчислюємо середній бал за валідними анкетами
# Помножуємо кожен бал на кількість студентів, сумуємо і ділимо на кількість валідних анкет
weighted_sum = (df[df['score'].notnull()]['score'] * df[df['score'].notnull()]['count']).sum()
average_score = weighted_sum / valid_surveys

print(f"Загальна кількість анкет: {total_surveys}")
print(f"Кількість валідних анкет (без пропусків): {valid_surveys}")
print(f"Середній бал за якість освіти (за валідними анкетами): {average_score:.2f}")
```

Рисунок 3.8 – Приклад коду розрахунку

Вивід:

- загальна кількість анкет: 98;
- кількість валідних анкет (без пропусків): 95;
- середній бал за якість освіти (за валідними анкетами): 3.95.

Пояснення:

- загальна кількість анкет: 30 + 40 + 3 + 25 = 98 (бо 3 пропуски теж рахуємо як анкети, але вони не мають балів);
- валідні анкети: усі, крім пропусків, 30 + 40 + 25 = 95;
- середній бал: зважене середнє за 95 валідними анкетами.

Код для розрахунку (з обробкою пропусків та імпутацією) (рисунок 3.9).

```
import pandas as pd

# Вхідні дані: оцінка -> кількість студентів
data = {
    'score': [3, 4, None, 5],   # None — пропуск
    'count': [30, 40, 3, 25]
}

df = pd.DataFrame(data)

# Кількість валідних спостережень
valid_df =
```

Рисунок 3.9 – Код для розрахунку (з обробкою пропусків та імпутацією)

Запропоновані алгоритми машинного навчання вирішують описану вище проблему: після впровадження класифікатора трудовитрати знизяться, оскільки зменшиться кількість анкет, які необхідно обробляти, а отримані результати матимуть значущість.

ВИСНОВКИ

Сучасні організації стикаються з необхідністю систематичної обробки великих обсягів даних, для чого застосовують інформаційні технології, що автоматизують рутинні операції. Це дозволяє суттєво зменшити кількість помилок, скоротити витрати часу та підвищити ефективність роботи. Крім того, автоматизовано процес відбору анкетних даних у Python з використанням алгоритмів машинного навчання, що дозволило ефективно класифікувати відповіді на «корисні» та «некорисні».

Для автоматизації рекомендовано У межах дослідження було проаналізовано п’ять алгоритмів машинного навчання: логістичну регресію, вирішальне дерево, випадковий ліс, стохастичний градієнтний спуск та метод K найближчих сусідів. Порівнявши їх точність, здатність до узагальнення та стійкість до «шумових» даних, найбільш ефективним для класифікації анкет виявився алгоритм «випадковий ліс».

Це зумовлено його здатністю:

- працювати з різноманітними типами ознак;
- мінімізувати переобучення завдяки ансамблю дерев;
- надавати високу точність класифікації навіть у випадках із нерелевантними або неповними анкетами.

Таким чином, випадковий ліс рекомендовано як основний алгоритм для автоматизованого відбору анкет у системах освітнього аналізу.Проєкт є масштабованим, оскільки база відповідей щороку оновлюється, а класифікатори можна донавчати. У майбутньому можлива розробка web-сервісу для інтеграції з системою збору та обробки даних через JSON. Підсумковий висновок свідчить про успішну реалізацію проєкту, а навчений алгоритм вже готується до використання в роботі відділу з документообігу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Фатій Р. Ю. Методи обробки даних сза допомогою штучного інтелекту / Р. Ю. Фатій // Радіоелектроніка та молодь у XXI столітті : матеріали 29-го Міжнар. молодіж. форуму, 16–19 квітня 2025 р. – Харків : ХНУРЕ, 2025. – Т. 5. – С. 16–18.

2. Коломієць А., Кушнір О. Використання штучного інтелекту в освітній та науковій діяльності: Можливості та виклики. Modern Information Technologies and Innovation Methodologies of Education in Professional Training Methodology Theory Experience Problems. 2023. № 70. С. 45–57. DOI: 10.31652/2412-1142-2023-70-45-57.

3. Осадчий В. Сучасні тенденції цифровізації управлінських процесів у вищій освіті: аналітика даних, хмарні технології, штучний інтелект. Educological discourse. 2024. № 1 (44). С. 8–27. DOI: https://doi.org/10.28925/2312-5829.2024.11

4. Missing value imputation on missing completely at random data using multilayer perceptrons / E.-L. Silva-Ramírez [et al.] // Neural Networks. - 2011. - Vol. 24, iss. 1. - P. 121-129.

5. Single imputation with multilayer perceptron and multiple imputation combining multilayer perceptron and k-nearest neighbours for monotone patterns / E.-L. Silva-Ramírez [et al.] / / Applied Soft Computing. - 2015. - № 29. - P. 128-132.

6. Yoon S., Lee S. Training algoritm with incomplete data for feed-forward neural networks // Neural Processing Letters. - 1999. - № 10 (3). - P. 171-179.

7. Ерік Маттес. Пришвидшений курс Python / Маттес Ерік. - Видавництво Старого Лева, 2021. - 600 с.

8. . Aurelien Geron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 3rd Edition / Geron Aurelien. - Publisher O'Reilly, 2022. - 850 p.

9. Sebastian Raschka. Python Machine Learning, 1st Edition / Raschka Sebastian. - Publisher Packt Publishing, 2015. - 454 p.

10. Best Python libraries for Machine Learning [Електронний ресурс]. Режим доступу: https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/

11. Open Source Machine Learning Libraries For Python Developers [Електронний ресурс]. Режим доступу: https://www.c-sharpcorner.com/blogs/open-source-machine-learning-libraries-forpython-developers