Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп’ютерної інженерії та управління
Кафедра Комп’ютерних інтелектуальних технологій та систем

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Виявлення депресії із постів соціальних мереж на основі машинного навчання
(тема)

Виконав:
студент II курсу, групи КІТм-22-1
Руслан АГІБАЛОВ
(власне ім'я, прізвище)

Спеціальність 123 Комп’ютерна інженерія
(код і повна назва спеціальності)

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

Освітня програма Комп'ютерні інтелектуальні технології
(повна назва освітньої програми)

Керівник проф. каф. КІТС Наталія АКСАК
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри Олег РУДЕНКО
(підпис) (власне ім'я, прізвище)

2023 р.

Харківський національний університет радіоелектроніки

Факультет	Комп'ютерної інженерії та управління
Кафедра	Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти	другий (магістерський)
Спеціальність	123 Комп'ютерна інженерія
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерні інтелектуальні технології

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____»_____202_ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Агібалову Руслану Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Виявлення депресії із постів соціальних мереж на основі машинного навчання

затверджена наказом по університету від " 03 " листопада 2023 р. № 1290Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13.01.2024

3. Вхідні дані до роботи

1) виявлення основних методів аналізу тексту
2) побудова тестової моделі нейронної мережі для виявлення наявності депресії
3) середовище моделювання – Google Collab
4) мова програмування – Python
5) нейронні мережі

4. Перелік питань, що потрібно опрацювати в роботі

1) аналіз предметної області
2) аналіз предмету дослідження
3) дослідження нейронних мереж
4) дослідження моделей нейронних мереж
5) дослідження методів обробки набору даних
6) розробка моделі виявлення ймовірності депресії
7) експериментальні дослідження
8) висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням кафедри)

6. Консультанти розділів роботи и (п.6 включається до завдання за наявністю консультантів згідно до наказу, зазначеному у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача та узгодження теми проєкту	06.11.2023	Виконано
2	Огляд стану проблеми та постановка задачі	07.11-08.11	Виконано
3	Аналіз літератури за напрямком магістерської	08.11-17.11	Виконано
	роботи		
4	Вибір технології розробки та	18.11-26.11	Виконано
	інструментальних засобів		
5	Розробка тестової нейронної мережі	29.11-13.12	Виконано
6	Експериментальні дослідження	15.12-26.12	Виконано
7	Підготовка графічних матеріалів	02.01-13.01	Виконано
8	Перевірка виконаного проекту керівником	15.01.2024	Виконано
9	Захист проекту	25.01.2024	Виконано

Дата видачі завдання 06 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____ проф. каф. КІТС Наталія Аксак
(підпис) (посада, ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 75 с., 9 рис., 1 табл., 1 дод., 20 джерел.

МАШИННЕ НАВЧАННЯ, SVM, НЕЙРОННА МЕРЕЖА, CNN, ДЕПРЕСІЯ, LSTM, СОЦІАЛЬНА МЕРЕЖА

Метою кваліфікаційної роботи є розробка системи виявлення деперсії за допомогою нейронних мереж.

У роботі пропонується система для виявлення депресивних настроїв користувачів соціальних мереж. Під час роботи було проаналізовано існуючи методи обробки текстових даних, формування критерій даних для обробки та навчання.

Об’єктом дослідження є система виявлення вірогідності наявності депресивних розладів.

Предметом дослідження є процес аналізу та підготовки набору даних для вилучення різноманітних текстових характеристик, синтактичних, лексичних і семантичних. А також знаходження ключових фраз та термінів що можуть вказувати на депресивні розлади.

ABSTRACT

Master`s thesis: 75 pages, 9 figures, 1 tables, 1 appendices, 20 sources.

MACHINE LEARNING, SVM, NEURAL NETWORK, CNN, DEPRESSION, LSTM, SOCIAL NETWORK

The major goal of the qualification work is to develop a system for detecting depression using neural networks.

The paper proposes a system for detecting depressive moods of social media users. During the work, we analyzed existing methods for processing text data, forming data criteria for processing and training.

The object of research is a system for detecting the likelihood of depressive disorders.

The subject of the study is the process of analyzing and preparing a dataset to extract various textual characteristics, syntactic, lexical, and semantic. It is also about finding key phrases and terms that may indicate depressive disorders.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп’ютерної інженерії та управління
Кафедра Комп’ютерних інтелектуальних технологій та систем

АНОТАЦІЯ

КВАЛІФІКАЦІЙНОЇ РОБОТИ

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Виявлення депресії із постів соціальних мереж на основі машинного навчання
(тема)

Виконав:
студент II курсу, групи КІТм-22-1
Руслан АГІБАЛОВ
(власне ім'я, прізвище)

Спеціальність 123 Комп’ютерна інженерія
(код і повна назва спеціальності)

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

Освітня програма Комп'ютерні інтелектуальні технології
(повна назва освітньої програми)

Керівник проф. каф. КІТС Наталія АКСАК
(посада, власне ім'я, прізвище)

2023 р.

АНОТАЦІЯ

Агібалов Р.І. Виявлення депресії із постів соціальних мереж на основі машинного навчання. – Магістерська кваліфікаційна робота.

Актуальність теми дослідження.

У сучасному світі соціальні мережі стали важливою частиною повсякденного життя, оскільки вони пропонують унікальну платформу для обміну інформацією та самовираження. Велика база даних створюється мільйонами людей по всьому світу, які в реальному часі діляться своїми думками, емоціями та враженнями. Ця база даних відображає різні аспекти людського життя. Відповідно, виявлення психічних розладів, таких як депресія, стає важливим напрямком досліджень у цьому контексті.

Депресія стає все більш поширеною проблемою громадського здоров'я. Тим не менш, діагностика може бути складною через її приховану природу та індивідуальні прояви. У таких ситуаціях машинне навчання дає чудові можливості для створення ефективних методів виявлення депресії на ранніх стадіях.

Дослідження, які спрямовані на виявлення депресивних розладів через пости в соціальних мережах, можуть допомогти діагностувати та підтримувати людей, які страждають від такого захворювання. Ця тема є актуальною через соціальні та культурні зміни, а також прагнення до прогресу в сфері охорони здоров’я, де технології можуть бути важливими для швидкого надання допомоги та підтримки.

У даній роботі розглядається потенціал машинного навчання в подоланні викликів, пов'язаних із діагностикою депресії, що робить тему надзвичайно актуальною та перспективною для подальших досліджень і розробок.

Метою даної роботи є розробка моделі виявлення депресії користувачів соціальних мереж за допомогою машинного навчання.

Об’єктом дослідження є система виявлення вірогідності наявності депресивних розладів.

Предметом дослідження є процес аналізу та підготовки набору даних для вилучення різноманітних текстових характеристик, синтактичних, лексичних і

семантичних. А також знаходження ключових фраз та термінів що можуть вказувати на депресивні розлади.

Методи дослідження. Дослідження базується на системному аналізі результатів сучасних теоретичних і прикладних розробок вітчизняних і зарубіжних вчених в галузі ІТ. Для вирішення поставлених завдань використано методи інтелектуального аналізу даних, TextMining, теорія штучних нейронних мереж.

Практична цінність отриманих результатів. Результати дослідження можуть стати основою для розроблення цифрових психологічних помічників, здатних автоматично аналізувати текстові дані та надавати рекомендації або спрямовувати користувачів до професійної допомоги в разі виявлення ознак депресії. Так само використання технологій для ширшого моніторингу психічного здоров'я в онлайн-середовищах може призвести до підвищення інформованості про цю проблему, поліпшення діагностики та своєчасного надання підтримки. Це своєю чергою може знизити суспільні втрати, пов'язані з депресивними розладами.

У першому розділі проведено аналіз предметної області та літератури щодо теми роботи. Завдяки цьому ми дізнались про основні поняття що до деперсивних розладів, як саме вони впливають на життя людини та як люди зазвичай себе поводять коли зтикаються з подібними проблемами.

Проблемою є те що зазвичай більшість людей не приділяють увагу різним симптомам виправдовуючи це для себе наприклад звичайною втомою або зіпсованим настроєм. Крім того, люди можуть соромитися відкрито обговорювати свої переживання, занижуючи їхню значущість порівняно з проблемами інших людей, які, на перший погляд, можуть видатися справді більш значущими. Саме тому поширені випадки, коли людина ділиться своїми проблемами в Інтернеті. Наприклад, ведучи свій блог у соціальних мережах або зовсім використовуючи анонімні групи, оскільки там ризик бути розкритикованим буде меншим.

Саме тому використання підходу машинного навчання по відношенню до текстових даних із соціальних мереж може стати більш ефективним вирішенням цього питання по відношенню до великих груп населення.

Машинне навчання є ефективним інструментом для автоматичного виявлення

складних залежностей і патернів у великих наборах даних. Існує багато різних методів МН для визначення депресії, які використовують текстові дані з соціальних мереж. Розглянуті статті та дослідження показують, що немає однозначно одного істинно правильного підходу. Науковці кожен окремо роблять свій внесок у розвиток цієї теми, використовуючи різні підходи, технології, початкові дані тощо.

Так само в цьому розділі розглянуто алгоритми деяких найбільш розповсюджених архітектур нейронних мереж для обробки різних типів даних. Таким чином приділено увагу згортковим нейронним мережам, рекурентним нейронним мережам і моделі LSTM. Детально описано конкретні кроки, які використовують у своїй роботі подібного роду архітектури.

Другий розділ було присвячено поясненню прийнятих проектних рішень і методів дослідження.

Було розглянуто питання того, яка мова програмування буде ефективнішою та зручнішою для розробки моделі. Для цього зроблено невеликий огляд найпоширеніших мов, які використовуються найчастіше для машинного навчання і роботи з нейронними мережами відповідно. Також у вигляді таблиці описано переваги та недоліки різних мов. Таким чином виявлено що все ж таки саме Python зможе задовольнити більшу частину потреб.

Цьому вибору сприяло кілька причин. По-перше, Python має читабельний і зрозумілий синтаксис, що робить код лаконічним і таким, що легко сприймається навіть для новачків. Це сприяє швидкій розробці та експериментам з моделями.

По-друге, багата екосистема бібліотек, як-от TensorFlow, PyTorch, і scikit-learn, робить Python ідеальним вибором для реалізації та навчання різноманітних моделей машинного навчання. Ці бібліотеки надають готові реалізації безлічі алгоритмів, спрощуючи процес розробки та оптимізації моделей.

Крім того вивчено один із найважливіших етапів у підготовці даних для подальшого навчання моделей як - попереднє опрацювання даних. Він включає в себе кілька основних моментів:

Видалення дублікатів - ідентифікація та видалення повторюваних записів у даних допомагає запобігти спотворенням і підвищує якість моделей.

Обробка пропущених значень - заповнення або видалення пропущених значень може бути необхідним для запобігання спотворенням у навчальних даних.

Нормалізація даних - приведення даних до єдиного масштабу дає змогу уникнути проблем із вагами в моделях, що базуються на відстанях, таких як метод k-найближчих сусідів.

Кодування категоріальних змінних - перетворення категоріальних змінних у числові форми забезпечує можливість включення цих змінних у моделі машинного навчання.

Витяг ознак - створення нових ознак або модифікація наявних може поліпшити продуктивність моделі, роблячи дані більш інформативними.

Обробка викидів - виявлення і корекція викидів у даних може запобігти спотворенню результатів моделі.

Поділ даних - розбиття даних на навчальний, валідаційний і тестовий набори забезпечує об'єктивну оцінку продуктивності моделі.

Масштабування ознак - приведення числових ознак до стандартних діапазонів може допомогти моделям сходитися швидше.

Перевірка на наявність викидів - аналіз даних на наявність аномалій і викидів допомагає поліпшити стійкість моделей.

У третьому розділі пропонується комплексний підхід до збору даних для подальшого використання і навчання моделі. Розглядаються різні способи отримання публікацій з популярних соціальних мереж, наприклад Twitter, Instagram тощо. Для цього використовуються API цих месенджерів для спрощеного доступу до них і подальшого аналізу контенту. Описуються різного роду обмеження, які існують для їх використання, і бібліотеки для роботи з різного роду інформацією. Наприклад, отримання частоти використання застосунку, тобто активності користувача, отримання кількості залишених коментарів і реакцій на певні публікації.

У такий спосіб визначаються вимоги для створюваного набору даних і список певних проблем, з якими можна зіткнутися в процесі формування даних для задач машинного навчання, а саме:

Класифікація текстів. Щоб досягти цього, необхідно визначити теми або

категорії текстових документів. створити набір даних, використовуючи текстові документи, які стосуються різних тем або категорій. Це можуть бути різного роду статті, блог-пости, книги або новинні статті, отримані з різних джерел.

Виявлення емоцій в тексті. Задача полягає в тому, щоб визначити емоційний тон тексту. Створений набір даних містить тексти, які є емоційно зарядженими, разом із описом емоційних характеристик, які вони містять. Це може включати відгуки, огляди та пости в соціальних мережах.

Аналіз настрою в соціальних мережах. Завдання складається в тому, щоб зрозуміти емоції користувачів соціальних мереж. У цьому місці оброблюються тексти, отримані з соціальних мереж, таких як коментарі, статуси Facebook і твіти. Після цього вони повинні дати короткий опис свого настрою (позитивний, негативний або нейтральний).

Розпізнування обьектів на зображеннях. Оскільки депресія є в основному психічним захворюванням і не може бути однозначно визначена за виглядом об’єктів на фотографіях, розпізнавання депресії на зображеннях за допомогою виявлення об’єктів є складним завданням. Тим не менш, дослідження психічного здоров’я використовують аналіз зображень і мультимедійних вмістів, щоб надати додаткову контекст і підтримати діагностику.

Четвертий розділ присвячений розробці моделі виявлення ймовірності депресії та результатам роботи. Описаний процес навчання згорткової нейронної мережі, а саме опис необхідних кроків таких як: підготовка даних, створення моделі, її навчання і так далі. Присвяено увагу факторам вибору датасету: репрезентативність датасету, його розмір та різноманіття даних, якість та достовірність та його доступність для використання згідно вимогам та ліцензійним умовам.

Описаний процес розробки та навчання мережі, опис дій основних функцій та їх лістинг. Представляються результати навчання протягом певної кількості епох та точність розробленої моделі. Крім того виявленно що протягом епох значення втрат даних зменшувались, що свідчить про прогресивне навчання моделі.

МАШИННЕ НАВЧАННЯ, SVM, НЕЙРОННА МЕРЕЖА, CNN, ДЕПРЕСІЯ, LSTM, СОЦІАЛЬНА МЕРЕЖА

Публікації здобувача за темою роботи:

1. Аксак Н.Г., Агібалов Р.І. Дослідження постів із соціальних мереж для визначення депресії на основі машинного навчання. VI International Scientific and Theoretical Conference «The current state of development of world science: characteristics and features». Lisbon, Portugal. 2023 p C. 118-122. ISBN: 979-8-88955-776-0 (series) DOI:https://doi.org/10.36074/scientia-15.12.2023

Список праць наукового керівника за темою магістерської роботи:

1. N. Axak, M. Kushnaryov, A. Tatarnykov, The Agent-Based Learning Platform. ICST-2023: *XI International Scientific and Practical Conference "Information Control Systems and Technologies",* September 21-23, 2023, Odessa, Ukraine, Vol-3513, pp. 263-275.

2. Axak N., Korablyov M., Ushakov M. Cloud Architecture for Remote Medical Monitoring // *IEEE Proceedings of the 15th International conference «Computer Sciences and Information Technologies*», (CSIT-2020). – Zbarazh-Lviv, Ukraine, September 23-26, 2020. – vol. 1, pp. 344-347.

3. Axak N., Serdiuk N., Ushakov M., Korablyov M. Development of System for Monitoring and Forecasting of Employee Health on the Enterprise. //COLINS. – 2020. – C. 979-992.

4. N. Axak, A. Tatarnykov, "The Behavior Model of the Computer User," *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*, 2022, pp. 458-461, doi: 10.1109/CSIT56902.2022.10000499.

5. Axak N., Korablyov M., Rosinskiy D. MapReduce Hadoop Models for Distributed Neural Network Processing of Big Data Using Cloud Services / *Advances in Intelligent Systems and Computing* IV. / Editors: Shakhovska, Natalya; Medykovskyy, Mykola O., Springer, 2019. – pp. 387 – 400. ISSN: 2194-5357 //doi.org/10.1007/978-3-030-33695-0

ЗМІСТ

П

омилка! Закладку не визначено.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – згорткова нейронна мережа

RNN – рекурентна нейронна мережа

SVM – метод опорних векторів (англ., Support Vector Machine).

LSTM – різновид архітектури рекурентних нейронних мереж

МН – машинне навчання

ШІ – штучний інтелект

API – інтерфейс програмування застосунків

NLP – обробка природної мови (англ., Natural Language Processing)

CSV – текстовий формат, призначений для подання табличних даних

ВСТУП

Депресія є серйозним психічним захворюванням, яке стрімко впливає на життя людей у всьому світі. На фізичний, психологічний і емоційний стан людини впливає депресія. Його симптоми включають відсутність інтересу до повсякденних дій, тривогу, розчарування, пригніченість, втрату ваги та навіть почуття ненависті до себе. Таким чином, підвищення якості життя та благополуччя людей можна досягти за допомогою досліджень, спрямованих на прогнозування та запобігання депресії.

Одним із найважливіших компонентів людського життя є здоров’я людини. На сьогоднішній день є багато приватних та державних медичних установ, включаючи спортзали та реабілітаційні зали.

Крім того, багато мобільних застосунків є популярними, оскільки вони дозволяють користувачам відстежувати різні аспекти здоров’я, такі як кількість випитої води, кількість щоденних вправ, кількість калорій, які вони споживають, і як добре вони себе почувають. Але багато людей забувають, що потрібно піклуватися про психічне та фізичне здоров’я. Отже, психологічний стан людини також впливає на якість життя людини.

Метою цієї роботи є вивчення способів використання машинного навчання для виявлення депресії у користувачів шляхом аналізу постів із соціальних мереж. Машинне навчання є потужним інструментом для автоматичного виявлення патернів і складних залежностей у великих наборах даних. Ці стратегії можуть допомогти виявити основні ознаки депресії серед користувачів соціальних мереж. Для більш ефективного вилучення інформації з великих обсягів даних у процесі дослідження необхідно використовувати сучасні методи статистичного аналізу та машинного навчання. За допомогою великої кількості даних, зібраних із різних джерел, можна провести ретельний аналіз і виявити патерни та зв’язки, які можуть допомогти в прогнозуванні депресії.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття про депресивний розлад

На сьогоднішній день депресія є глибокою проблемою для українців і всього світу, особливо звертаючи увагу на обставини в країні та по всьому світу [1]. Значною мірою це пов’язано з низьким рівнем культури турботи про власне психологічне здоров'я, особливо в країнах СНД. Депресія – є надзвичайно небезпечним психічним захворювання, яке має бурхливий вплив на життя людей у всьому світі. Таке захворювання має значний вплив на емоційне, психологічне та фізичне здоров’я людини. У числі симптомів цього захворювання є знижений інтерес до життя та відсутність бажання виконувати щоденні рутинні дії. Крім того викликає збільшене почуття тривогу, розчарування, а також може стати однією з причин втрати загальної маси тіла та зниження самооцінки.

Багатьом психіатрам важко визначити, чи є у пацієнта психічне захворювання, оскільки кожен психічний розлад унікальний. Це робить важким процес забезпечення пацієнта належним лікуванням, перш ніж це стане занадто пізно. Тим не менш, соціальні мережі зараз є частиною повсякденного життя людей, тому вони можуть надати додаткову інформацію про психічний розлад пацієнта.

Соціальні мережі стали потужною комунікаційною платформою, яка дозволяє виражати емоції та відображає душевний стан багатьох користувачів. З огляду на зростаюче занепокоєння проблемами психічного здоров'я, варто розглянути можливість використання соціальних мереж як інструменту для виявлення депресії у користувачів [2, 3].

У сучасному світі люди, як правило, використовують соціальні мережі, щоб висловити свої проблеми, що може допомогти психологам і психіатрам приймати кращі рішення, а також забезпечити раннє виявлення за допомогою даних, отриманих від суб’єктів соціальних мереж. Це підтверджено результатами дослідження, яке виявило, що студенти з депресивними симптомами користуються Інтернетом значно

частіше, ніж студенти без симптомів. Це спонукало дослідників шукати більш ефективні способи ранньої діагностики депресії. [4].

Ці проблеми дослідження щодо прогнозування та запобігання депресії можуть значно покращити якість життя людей. Тим не менш, результати цих досліджень повинні бути використані відповідно до етичних принципів і з обережністю щодо прав і безпеки досліджуваних.

Постановка проблеми. Проблеми, які необхідно вирішити, були викликані наступними факторами:

Недостатність об'єктивних методів визначення депресії в інтернет-просторі, у тому числі які враховують культурні відмінності та особливості вираження депресії в різних соціокультурних групах користувачів [5].

Відсутність систем, що дозволяють виявляти ранні ознаки депресії та надавати підтримку користувачам у соціальних мережах.

Мета дослідження – проаналізувати існуючи підходи для текстового виявлення депресії або тривожних розладів на основі активності людей в соціальних мережах та запропонувати шляхи вирішення недоліків другого аспекту шляхом поєднання методів глибокого навчання та агентно-орієнтованої технології.

Результати дослідження. Традиційно депресію виявляють за допомогою стандартизованих шкал, які вимагають суб'єктивних відповідей пацієнтів або клінічних діагнозів, поставлених лікуючими клініцистами. Ці методи мають деякі недоліки. По-перше, на реакцію людей на стандартизовані шкали, що застосовуються традиційним способом, ймовірно, впливатимуть контекст, психічний стан пацієнта на даний момент, стосунки між клініцистом і пацієнтом, поточний настрій пацієнта, а також минулий досвід пацієнта та зміщення пам’яті [6].

Традиційним методам діагностики також не вистачає часової деталізації [7]. По-друге, люди можуть не знати або соромитися своїх депресивних симптомів і навряд чи звертатимуться до професійних клініцистів, особливо на ранніх стадіях депресії. Попереднє дослідження показало, що понад 70% населення не зверталися б до професійних клініцистів, якщо б вони перебували на ранніх стадіях депресії, а це означає, що вони, ймовірно, дозволили б своїм симптомам погіршитися, перш ніж

розглянути питання про допомогу [8]. Нарешті, виявлення депресії традиційними методами, оскільки залежить від особистих інтерв'ю, є дорогим з точки зору як грошей, так і часу, і недоступне для деяких людей [9]. Тому потрібен більш економічно ефективний метод виявлення випадків депресії, застосовний до великих груп населення.

Застосування підходу машинного навчання (МН) до текстових даних із соціальних мереж може забезпечити ефективне вирішення цього питання. Соціальні медіа, такі як Twitter, Facebook, дискусійні форуми та мікроблоги, давно стали популярними платформами для вираження та запису особистостей, почуттів, настроїв, думок і поведінки людей. Різні типи соціальних медіа можуть допомогти користувачам проводити різні типи самопрезентації, наприклад на основі тексту, відео, зображень тощо. А деякі групи соціальних медіа (наприклад, блоги та сайти соціальних мереж) мають вищу ступінь саморозкриття. Отже, аналіз даних величезної кількості тексту, за допомогою якого можна шукати користувачів соціальних мереж, може мати велике значення для виявлення випадків депресії.

1.2 Огляд літератури

Машинне навчання є потужним засобом для автоматичного знаходження складних залежностей та патернів у великому обсязі даних. Існує кілька оглядів щодо підходів МН для виявлення депресії, які використовують текстові дані з соціальних мереж. Наприклад, в [10] використовують дані нейровізуалізації електроенцефалограми для прогнозування депресії. Методи МН також пропонують можливості для виявлення прихованих шаблонів у онлайн-спілкуванні та взаємодії в соціальних мережах, які можуть виявити психічні стани користувачів, такі як депресія, тривога, гнів, тощо [11].

Використовуючи методи МН можна покращити розпізнавання ключових ознак, які зазначають вірогідність наявності депресивного розладу у певного користувача соціальних мереж. Впродовж дослідження застосовано новітні засоби статистичного аналізу та машинного навчання для вилучення інформації з великих об'ємів даних.

Такий підхід до вибору даних, які було зібрано із різних джерел, надає змогу реалізувати всебічний аналіз і виявити зв'язки та патерни, що можуть бути корисними для передбачення депресії. Автори статті [12] класифікують пости на «депресивні» та «недепресивні» за допомогою машинного навчання. Для досягнення цього автори використовують різноманітні характеристики, які включають лексичні, синтаксичні та семантичні аспекти тексту постів.

Дослідження [13] має на меті вивчення методів виявлення депресії через використання машинного навчання для аналізу повідомлень користувачів у соціальних медіа. Автори проводять попереднє опрацювання даних, видаляючи стоп-слова та символи пунктуації, а також перетворюючи текст на числові ознаки за допомогою методу «мішка слів». Після цього пости класифікуються на депресивні та недепресивні за допомогою методу опорних векторів або SVM (Support Vector Machine).

У статті [14] пропонується метод, який може ефективно ідентифікувати студентів коледжу з депресією. Спочатку на теоретичному рівні аналізується можливість розпізнавання депресії на основі даних Weibo, а потім починається перевірка на практиці. Ця стаття представляє більш стабільну та точнішу модель класифікації під назвою DISVM, яка використовує стратегію інтеграції AdaBoost і глибоку нейронну мережу.

В роботі [15] було проаналізовано та оцінено 34 первинних дослідження. Twitter був найбільш дослідженою соціальною мережею для виявлення ознак депресії. Вбудовування слів було найвідомішим методом виділення мовних ознак. Метод опорних векторів (SVM) був найбільш використовуваним алгоритмом машинного навчання. Так само найпопулярнішим обчислювальним інструментом були бібліотеки Python. Нарешті, перехресна перевірка (CV) була найпоширенішим методом статистичного аналізу, який використовувався для оцінки отриманих результатів.

Загалом, головна мета цих статей полягає в тому, щоб довести, що використання машинного навчання для аналізу постів у соціальних мережах може бути ефективним інструментом для виявлення ознак депресії. Можна точно

визначити депресію, використовуючи різні елементи тексту, такі як лексика, синтаксис і емоційні характеристики, а також інформацію про користувачів.

1.3 Алгоритми нейронних мереж

CNN є однією з найпоширеніших архітектур нейронних мереж і обробляє різні види даних, такі як звук, текст і зображення. CNN може бути використана під час обробки текстових даних для виявлення локальних особливостей і зв’язків у тексті.

Застосування згорток для виявлення важливих характеристик або функцій у вхідних даних є основною метою згорткової нейронної мережі. Згортка — це процес, який використовує ядро або фільтр для обробки вхідних даних і обчислює локальну суму зважених значень. Це полегшує виявлення важливих характеристик, таких як краї, форми, текстури тощо.

Для обробки текстових даних згорткова нейронна мережа використовує наступні шари: вбудовувальний шар (embedding), згорткові шари (convolutional layers), пулінгові шари (pooling layers), повністю пов’язані шари (full-connected layers) і вихідний шар (layer).

Вбудовувальний шар (Embedding). Вбудовування слів, використовується для перетворення початкового тексту на числові вектори. За допомогою вбудовування моделі можуть розуміти семантику слів і те, як вони пов’язані з контекстом.

Згорткові шари (Convolutional Layers). Згорткові шари використовують ядра, також відомі як фільтри, для виявлення локальних характеристик у тексті. Фільтри застосовуються шляхом множення та сумування значень після того, як вони рухаються по всій вхідній послідовності. У згорткових шарах можна знайти важливі ознаки, такі як словосполучення, ключові слова або фрази, які можуть свідчити про депресію.

Пулінгові шари (Pooling Layers): Після згорткових шарів застосовуються пулінгові шари, які зменшують розмір виходу шляхом вибору найважливіших значень або інших статистичних операцій. Глобальний пулінг стискає найважливіші елементи послідовності у вектор певної довжини.

Повнозв'язані шари (Fully Connected Layers): Після згорткових та пулінгових шарів використовуються повнозв'язані шари для обробки отриманого вектору. Повнозв'язані шари виконують класифікацію або прогнозування на основі зібраних особливостей з попередніх шарів.

Вихідний шар (Output Layer): Вихідний шар використовує функцію активації, таку як сигмоїда, для вирішення задачі бінарної класифікації (наявність депресивних мислей).

Обробка послідовностей даних із довгостроковою залежністю можлива за допомогою типу рекурентних нейронних мереж (RNN), відомого як кінцеві стани довгої короткострокової пам’яті (LSTM). Його створили, щоб вирішити проблему затухання та вибуху градієнта, з якою стикаються традиційні RNN.

Модель LSTM здатна зберігати інформацію протягом тривалого періоду часу завдяки декільком блокам пам'яті, відомим як «комірки LSTM». Кожна комірка LSTM складається з вхідного вентиля, вентиля, що забуває, і вихідного вентиля.

Вхідний вентиль контролює, яку інформацію потрібно додати до стану пам'яті комірки LSTM. Вентиль, що забуває, визначає, яку інформацію слід видалити зі стану пам'яті, щоб уникнути накопичення нерелевантних даних. Вихідний вентиль вирішує, яку інформацію слід використовувати для виходу з комірки LSTM.

Здатність керувати потоком інформації за допомогою вентилів і зберігати інформацію в довгостроковій пам’яті є основною характеристикою LSTM. Це дозволяє моделі LSTM вловлювати та запам’ятовувати залежності в даних протягом тривалого проміжку часу.

Модель LSTM широко використовується в розпізнаванні мови, генерації тексту, опрацюванні природної мови, прогнозуванні часових рядів та інших областях, де опрацювання послідовностей даних із довгостроковою залежністю є важливим.

Важливо звернути увагу на те, що модель LSTM може бути розширена та змінена, щоб враховувати конкретні вимоги завдання. Наприклад, доступні варіанти LSTM з двонаправленими зв'язками, множинними шарами або комбінацією LSTM з іншими типами нейронних мереж.

Загалом, модель LSTM є популярним інструментом у багатьох застосунках

машинного навчання для опрацювання послідовних даних із довгостроковими залежностями.

1.4 Постановка проблеми дослідження

Наведені аспекти, що виходять з аналізу предметної області, обумовили проблеми, які необхідно вирішити:

- недостатність об'єктивних методів визначення депресії в інтернет-просторі, у тому числі які враховують культурні відмінності та особливості вираження депресії в різних соціокультурних групах користувачів;
- відсутність систем, що дозволяють виявляти ранні ознаки депресії та надавати підтримку користувачам у соціальних мережах.

Мета дослідження – розробити гібридну техніку аналізу виявлення депресії в соціальних мережах. Для досягнення мети дослідження необхідно вирішити наступні завдання:

- проаналізувати існуючи підходи для текстового виявлення депресії або тривожних розладів на основі активності людей в соціальних мережах;
- визначити, яка мова програмування та її технології і бібліотеки є найкращим варіантом для вирішення поставленої задачі;
- проаналізувати існуючи методи обробки даних;
- визначити вимоги для підготовки набору даних для використання моделі;
- розробити системи виявлення депресії із постів соціальних мереж.

2 ВИЗНАЧЕННЯ ПРОЕКТНИХ РІШЕНЬ ТА МЕТОДІВ ДОСЛІДЖЕНЬ

2.1 Вибір мови програмування

Вибір мови програмування для розробки системи виявлення депресії з постів у соціальних мережах залежить від багатьох факторів і вимог, а також від уподобань і потреб кінцевого продукту. Ось деякі з найпоширеніших мов програмування в цій галузі:

- Python: є однією з найпопулярніших мов що використовують для машинного та глибокого навчання. Він має велику екосистему бібліотек і фреймворків, включаючи TensorFlow, PyTorch і scikit-learn;
- R: також є популярною мовою для машинного навчання та статистичного аналізу. У ньому є багато пакетів доі бібліотек, які можна використовувати для роботи з текстовими даними та реалізації алгоритмів аналізу сентименту;
- Java: може використовуватися для розробки великих систем, які можна масштабувати. Крім того, він може бути використаний у поєднанні з бібліотеками, такими як Deeplearning4j, для створення нейронних мереж.
- JavaScript (або TypeScript): цю мову програмування можна використовувати для розробки фронтенду програми, якщо застосунок передбачає веб-інтерфейс або інтеграцію з веб-сервісами. TensorFlow.js та інші бібліотеки дозволяють виконувати інференс нейронних мереж у браузері;
- C++: використовується для створення високопродуктивних обчислень, що важливо для навчання та застосування великих моделей глибокого навчання;
- Scala: мова програмування, сумісна з Java, яка має виразний синтаксис. Цей інструмент можна використовувати для створення масштабованих систем і інтегрувати з бібліотеками машинного навчання, такими як Apache Spark MLlib.

Далі проведемо порівняльний аналіз мов програмування щодо того, наскільки вони придатні для аналізу постів із соціальних мереж у контексті роботи з машинним навчанням, (таблиця 2.1.).

Таблиця 2.1 – Порівняння мов програмування для роботи з машинним навчанням

Мова програмування	Переваги	Недоліки
Python	- Велика екосистема бібліотек для обробки текстових даних і машинного навчання; - Популярні фреймворки, як-от TensorFlow і PyTorch, забезпечують ефективні засоби для створення і навчання моделей глибокого навчання; - Простота синтаксису і швидке прототипування.	- Деякі обчислювально інтенсивні операції можуть потребувати оптимізації з використанням C/C++.
R	- Широко використовується в статистичному аналізі та обробці даних. - Багаті пакети для аналізу текстових даних і статистичних методів. - Простий синтаксис і хороша читабельність коду.	- Обмежені можливості для широкомасштабних обчислень; - Менша спільнота розробників порівняно з Python.
Java	- Платформна незалежність; - Висока продуктивність; - Велика кількість інструментів і бібліотек для машинного навчання.	- Необхідно більше коду для досягнення тих самих результатів, ніж у Python або R - Складність у використанні для початківців
JavaScript	- Використовується для розробки фронтенду і може інтегруватися з бібліотекою TensorFlow.js. - Широко поширений для веб-розробки.	- Може бути менш ефективним для обчислювально інтенсивних операцій машинного навчання.
C++	- Висока продуктивність і можливість оптимізації обчислювально інтенсивних операцій. - Використовується в бібліотеках глибокого навчання, таких як TensorFlow і PyTorch.	- Нижчий рівень абстракції, що може ускладнити розробку.
Scala	- Сумісний з Java, забезпечуючи переваги Java і більш високий рівень виразності. - Використовується в Apache Spark, що дає змогу обробляти великі обсяги даних.	- Меньш поширенний і широко використовується в галузі аналізу текстових даних, як Python або R.

Python виявляється найкращою мовою програмування для використання машинного навчання для розробки програм аналізу постів із соціальних мереж. Це пов’язано з кількома важливими факторами.

По-перше, Python має значну екосистему бібліотек, яка включає TensorFlow, PyTorch, scikit-learn і NLTK, яка надає різноманітні інструменти для обробки текстових даних, навчання та оцінювання моделей машинного навчання.

По-друге, швидке прототипування та ітеративна розробка Python ідеальні через простоту синтаксису, що особливо важливо для завдань аналізу даних. Крім того, легко освоїти мову і отримати доступ до підтримки завдяки активному співтовариству розробників і великій кількості освітніх ресурсів, спрямованих на використання Python у машинному навчанні. Таким чином, Python є кращим інструментом для створення ефективних і гнучких рішень з аналізу різних обсягів тексту з використанням методів машинного навчання.

2.2 Технології та бібліотеки

Для створення системи штучного інтелекту для аналізу постів соціальних мереж необхідно використати бібліотеки для підтримки роботи з машинним навчанням. Python надає великий інструментарій для створення систем штучного інтелекту (ШІ) і машинного навчання.

TensorFlow – є одним із найпопулярніших фреймворків глибокого навчання. Він надає гнучку платформу для створення та навчання різних моделей нейронних мереж, включно зі згортковими нейронними мережами (CNN), рекурентними нейронними мережами (RNN) і трансформерами.

Tokenizer з tensorflow.keras.preprocessing.text – ця бібліотека використовується для токенізації тексту, тобто розбиття тексту на окремі слова або токени. Tokenizer дозволяє побудувати словник слів і виконувати індексування слів у вигляді числових послідовностей для подальшої обробки моделлю;

PyTorch – ще один фреймворк глибокого навчання, який став популярним завдяки своїй простоті у використанні та динамічному обчисленню графа. Він

активно використовується дослідниками та інженерами для створення та навчання нейронних мереж.

scikit-learn – бібліотека для машинного навчання, що надає інструменти для класифікації, регресії, кластеризації, а також попереднього опрацювання даних і оцінки моделей. Добре підходить для загальних завдань машинного навчання.

pad_sequences з tensorflow.keras.preprocessing.sequence – ця бібліотека використовується для додавання заповнення (padding) до послідовностей чисел. У нашому випадку, ми використовуємо pad_sequences для додавання заповнення до текстових послідовностей слів, щоб усі послідовності мали однакову довжину. Це необхідно для використання текстових даних в нейронних мережах, які вимагають фіксованого розміру вхідних даних.

Natural Language Toolkit (NLTK) – бібліотека для обробки природної мови (NLP). Вона надає безліч інструментів для аналізу тексту, включно з токенізацією, лематизацією, синтаксичним аналізом і багато іншого.

spaCy – ще одна бібліотека для NLP, що спеціалізується на обробці текстових даних. Вона забезпечує швидке й ефективне виконання завдань, як-от витяг іменованих сутностей і аналіз залежностей.

Pandas – бібліотека для роботи з даними, що надає високопродуктивні структури даних та інструменти для маніпуляції й аналізу табличних даних. Це важливий інструмент для попередньої обробки даних.

Matplotlib і Seaborn – бібліотеки для візуалізації даних. Вони дають змогу створювати графіки та діаграми, що важливо для аналізу результатів і візуалізації інформації.

2.3 Попередня обробка даних

Процес попереднього опрацювання даних, також званий як препроцесинг даних, є однією з важливих сходинок у сфері аналізу даних. Цей етап складається з попередньої підготовки та очищення даних, так вони стають більш придатними для подальшого аналізу та моделювання.

Попереднє опрацювання даних містить у собі кілька етапів, перший з яких є завантаженням даних із різного роду джерел, зокрема файлів значень, баз даних, розділених комами, та інтерфейсів прикладного програмування. Надалі слідує дослідження даних, необхідне для складання докладного уявлення про їхній зміст і організацію.

Видалення відсутніх значень (обробка пропусків): У деяких випадках можуть бути відсутніми значення в наборі даних для певних ознак. Унаслідок чого можуть виникати проблеми під час аналізу або навчання моделей. Щоб уникнути цього можуть використовуватися методи, що дають змогу заповнити або видалити значення, які були пропущені.

- Виявлення пропущених значень. Насамперед потрібно виявити пропущені значення в наборі даних. Це можливо при використанні певних функцій, що надаються бібліотеками для аналізу даних, такими як Pandas у Python. Наприклад такі функції, як isnull() та info(), дають змогу визначити номери стовпців, у яких є пропущені значення.

- Видалення пропущених значень. Одним із найпростіших способів є видалення рядків або стовпців, які містять пропущені значення. Цей метод може призвести до втрати великої кількості даних, тому його слід застосовувати з обережністю. Наприклад, рядки та стовпці з пропущеними значеннями можна видалити за допомогою методів dropna().

- Заповнення відсутніх значень. Замість видалення рядків або стовпців відсутні значення можна заповнювати середнім, медіанним або модальним значенням, залежно від наданих даних. Цей метод добре застосовується в ситуаціях, коли видалення даних недоцільне. У Pandas для заповнення пропущених значень є метод fillna().

- Індикатори пропущених значень. Замість того щоб заповнювати пропущені дані формальними значеннями, створюються допоміжні стовпці (індикатори), які явно показують, де саме були пропущені значення. Подібний підхід зручний для збереження інформації про існування пропущених даних.

Обробка викидів: викиди є незвичайними або аномальними значеннями даних,

які можуть вплинути на якість моделі або аналізу. Можно використовувати різні методи обробки, видалити їх або змінити на більш звичайні значения. Для управління викидима доступні наступні методи:

- Виявлення викидів: Перше, що потрібно зробити, це знайти джерело даних щодо викидів. Обов'язково використовується визуалізація даних або статистичні методи.

- Візуалізація: Можна візуально визначити аномалии в даних, використовуючи діаграмми разброса (scatter plots), ящики з усами (box plots) і гистограмми.

- Очищення відходів: Видалення строк сбросів є простим методом, але не завжди рекомендованим. Тим не менш, це може призвести до втрати важливих даних. Для видалення строк з вибросами в Pandas є доступним метод drop().

- Зміна викидів: Ви можете замінити викиди значеннями, які наближаються до стандартних значень у даних. Це може бути модальне, середнє або медіанне значення. Для заміни значень у Pandas можна використовувати метод replace().

- Інтерквартильний розмах (IQR): межі викидів визначаються за допомогою інтерквартильного розмаху. Значення, які перевищують верхню та нижню межі IQR, вважаються викидами. IQR можна використовувати для видалення або заміни викидів.

Шкалювання даних: різноманітність значень різних ознак може вплинути на продуктивність алгоритмів машинного навчання. Стандартизація значень, наприклад, можна досягти шляхом скалювання ознак до однакового масштабу.

- Приведення значень ознак до того самого масштабу називається скалюванням даних. Багато алгоритмів машинного навчання залежать від того, як представлені дані; якщо ознаки мають широкий діапазон значень, вони можуть працювати погано. Є кілька способів шкалювання даних:

Шкала Мін-Макс: цей метод масштабує значення ознак таким чином, щоб вони перебували в діапазоні [0, 1]. Для кожної ознаки застосовується формула (формула 2.1). Коли розподіл ознак наближається до рівномірного, цей метод особливо корисний.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (2.1)$$

де X_{scaled} – нове масштабоване значення змінної;

X - вихідне значення змінної, яке буде масштабуватися;

X_{min} - мінімальне значення змінної набору даних;

X_{max} - максимальне значення змінної набору даних.

Стандартизація, також відома як шкала Z, використовується для масштабування значень ознак, щоб вони мали середнє значення 0 і стандартне відхилення 1. Для кожної ознаки застосовується наступна формула (формула 2.2). Для даних, які розподілені нормально або близько до нормального розподілу, стандартизація підходить.

$$z = \frac{X-\mu}{\sigma} \tag{2.2}$$

де Z – нове масштабоване значення змінної;

X – вихідне значення змінної, яке потрібно масштабувати;

μ – середнє значення (математичне очікування) змінної в наборі даних;

σ – стандартне відхилення змінної набору даних.

- Стійке масштабування: цей метод включає викиди до значення ознак. Використовуючи медіану та інтерквартильний розмах, він більш стійкий до викидів. У ситуаціях, коли дані містять викиди, потужне масштабування є корисним, для неї використовується формула (формула 2.3).

$$X_{robust=}\frac{X-median(X)}{IQR(X)} \tag{2.3}$$

де X_{robust} - нове масштабоване значення змінної;

X – початкове значення;

Median(X) – медіана масиву X;

IQR(X) – міжквартильний розмах масиву X, різниця між третім та першим квартилями.

Кодування категоріальних ознак: Оскільки багато алгоритмів машинного навчання потребують числових даних, категоріальні ознаки, такі як типи товарів або країни, необхідно перетворити в числовий формат. Це можна зробити, наприклад, за допомогою методу «однієї холодної кодування».

- Одиничне кодування (одиничне кодування): цей метод перетворює кожне значення категоріальної ознаки на нову бінарну ознаку (стовпчик), яка має значення 0 або 1. Кожен стовпчик містить одну категорію, і для кожного рядка встановлюється один у стовпчик, відповідний категорії рядка. Коли немає порядку між категоріями, одночасне кодування є особливо корисним.

- Кодування міток, також відоме як кодування міток: цей спосіб надає кожному значенню категоріальної ознаки унікальний цілочисельний код. Порядок присвоєння кодів залежить від частоти значень або алфавітного порядку. Кодування маркуванням підходить для ознак, у яких є внутрішній порядок.

- Порядкове кодування, або порядкове кодування: використовується, коли категорії мають внутрішній порядок, який потрібно зберегти. Відповідно до її порядку кожній категорії присвоюється особливе цілочисельне значення. Коли важливий порядок між категоріями, звичайний кодування особливо корисно.

Інженерія ознак: допомагає покращити продуктивність моделі, створюючи нові ознаки на основі наявних ознак. Наприклад, можна створити індикатори, об’єднати ознаки або використовувати математичні перетворення.

Інженерія ознак – це процес створення нових ознак з наявних даних, щоб покращити продуктивність моделей машинного навчання. Цей процес є важливим для створення точних і ефективних моделей. Ось деякі способи інженерії ознак:

- Створення нових ознак: Іноді більш інформативні ознаки можна створити шляхом комбінування або перетворення наявних ознак. Наприклад, можна створити нову ознаку для прогнозування загального доходу сім'ї, об’єднавши зарплати всіх членів сім'ї.

- Бінарні ознаки: можна бути корисним перетворити категоріальні або номінальні ознаки на бінарні індикатори. Створення бінарної ознаки «вихідний» на основі дня тижня, наприклад

- Трансформація чисельних ознак: Застосування до чисельних ознак різних математичних функцій, таких як вилучення кореня, піднесення до степеня або логарифмування, може покращити їхній розподіл і зробити їх більш придатними для використання в моделях.

- Облік часу: Наявна можливість створити нові ознаки, такі як день, тиждень, місяць або сезон, якщо ваш набір даних містить тимчасові дані.

- Групування та статистичні ознаки: можна додати додаткову інформацію в модель, використовуючи групування даних відповідно до кожної ознаки. Наприклад, стандартне відхилення, середнє або медіана для кожної групи.

- Одночасне кодування текстових даних: під час роботи з текстовими даними можна перетворити текстові дані в числовий формат для використання в моделях за допомогою методів векторизації, таких як TF-IDF (термінова частота-інверсна частота документа) або Bag of Words.

- Додаткові ознаки з зовнішніх джерел: іноді можна створити нові ознаки, що покращують модель. Наприклад, додавання географічних даних, економічних показників або погодних даних.

Зменшення розмірності даних: Якщо є багато ознак, можна використовувати методи зменшення розмірності, такі як метод головних компонентів (PCA), щоб зменшити складність даних, зберігаючи важливі характеристики.

Зменшення розмірності даних – це процес зменшення кількості ознак у наборі даних, залишивши при цьому найважливіші дані в ньому. Поліпшення продуктивності алгоритмів машинного навчання, візуалізація даних і боротьба з прокляттям розмірності – це деякі з ситуацій, у яких цей процес корисний. Ось деякі способи зменшення розмірності:

Метод головних компонентів (PCA): PCA є одним із найбільш поширених методів зменшення розмірності. Він перетворює лінійні дані в новий простір, де осі впорядковані за зменшенням дисперсії. Вихідні характеристики разом із максимальною мінливістю є основними компонентами. Бібліотека scikit-learn у Python надає реалізацію PCA:

Лістінг 2.1:

```
from sklearn.decomposition import PCA

pca = PCA(n_components=new_dimensionality)
X_pca = pca.fit_transform(X)
```

Метод нелінійного зменшення розмірності, відомий як t-дифузія стока, або t-SNE, відображає точки даних з вихідного простору в новий двовимірний або тривимірний простір таким чином, щоб близькі точки залишалися близькими, а далекі точки залишалися далекими. Приклад використання scikit-learn у Python:

Лістинг 2.2:

```
from sklearn.manifold import TSNE

tsne = TSNE(n_components=new_dimensionality)
X_tsne = tsne.fit_transform(X)
```

Лінійний дискримінантний аналіз (LDA) – це метод зменшення розмірності, який зберігає інформацію про класи, одночасно збільшуючи відстань між класами. Це особливо корисно для класифікаційних завдань. У Python, за допомогою бібліотеки scikit-learn:

Лістинг 2.3

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

lda = LinearDiscriminantAnalysis(n_components=new_dimensionality)
X_lda = lda.fit_transform(X, y)
```

Мета випадкових проекцій, також відома як метод випадкових проекцій, заснована на випадковому проектуванні даних у менший простір. Випадкові проекції відрізняються від інших методів тим, що вони не вимагають матриці коваріації або розподілу даних.

Лістинг 2.3

```
from sklearn.random_projection import SparseRandomProjection
```

```
rp = SparseRandomProjection(n_components=new_dimensionality)
X_rp = rp.fit_transform(X)
```

Зменшення розміру даних покращує продуктивність моделей і усуває надмірність інформації, особливо коли у вихідних даних багато шуму або ознаки сильно корелюють між собою. Щоб уникнути втрати важливої інформації, важливо проводити ретельний аналіз і підбирати параметри при використанні цих методів.

3 СТВОРЕННЯ НАБОРУ ДАНИХ ТА ІСНУЮЧИ API

3.1 Збір даних

Для визначення різноманітних аспектів активності користувача в соціальних мережах під час збору даних можна використовувати API соціальних мереж або інші доступні інструменти для аналізу контенту. Ці дані можуть бути використані для створення характеристик, які пізніше будуть використовуватися в моделі. Однак, збір, зберігання і використання особистої інформації користувачів повинні відповідати політиці конфіденційності і правилам соціальних мереж.

3.1.1 Кількість публікацій користувача

Для багатьох соціальних мереж є можливість отримати кількість публікацій користувача через API. Наприклад, для Facebook можна використовувати Graph API. Приклад запиту до Graph API:

```
GET /v13.0/{user-id}?fields=id,name,posts{created_time,message}.
```

Тут {user-id} - це ідентифікатор користувача.

Для Instagram використовуємо Instagram Graph API:

```
GET/v13.0/{userid}/media?fields=id,caption,media_type,thumbnail_u
rl,timestamp
```

Тут {user-id} - це ідентифікатор користувача.

Для LinkedIn використовуємо LinkedIn API:

```
GET /v2/shares?q=owners&owners=urn:li:person:{user-id}
```

Тут {user-id} - це ідентифікатор користувача.

Для Twitter можна використовувати Twitter API:

```
GET/2/tweets?tweet.fields=created_at&expansions=author_id&user.fi
elds=username
```

Вибірка може бути обмежена конкретним користувачем, заданим його ідентифікатором (author_id). Бібліотека requests необхідна для взаємодії з Twitter API для отримання інформації про користувача та його твіти. Для цього потрібно мати

ключі доступу до Twitter API, які можна отримати, створивши заявку на Twitter Developer:

Лістинг 3.1:

```
import requests
def get_tweets(api_key, api_secret_key, access_token, access_token_secret, username):
    base_url = "https://api.twitter.com"
    endpoint = f"/2/tweets"
    params = {
        "tweet.fields": "created_at",
        "expansions": "author_id",
        "user.fields": "username",
        "max_results": 10  # Кількість твітів, яку ви хочете отримати
    headers = {
        "Authorization": f"Bearer {access_token}"
    response = requests.get(
        f"{base_url}{endpoint}?{params}",
        headers=headers
    if response.status_code == 200:
        tweets_data = response.json()
        for tweet in tweets_data.get("data", []):
            created_at = tweet.get("created_at", "")
            text = tweet.get("text", "")
            print(f"{created_at}: {text}")
    else:
        print(f"Error {response.status_code}: {response.text}")
# Введіть свої ключі доступу до Twitter API та ім'я користувача
api_key = "your_api_key"
api_secret_key = "your_api_secret_key"
access_token = "your_access_token"
access_token_secret = "your_access_token_secret"
twitter_username = "desired_twitter_username"
get_tweets(api_key, api_secret_key, access_token, access_token_secret, twitter_username)
```

Використання API підлягає обмеженням, і важливо дотримуватися правил використання. Цей код наведений лише для ілюстративних цілей і не використовується для недозволених або нелегальних дій (лістинг 3.1).

3.1.2 Кількість коментарів і реакцій

Також, за допомогою API, можна отримати кількість коментарів і реакцій на кожну публікацію. Це може дати уявлення про те, наскільки активно користувач

взаємодіє з іншими.

Наприклад, для Facebook можна також використовувати Graph API. Приклад запиту до Graph API:

```
GET /v13.0/{post-id}?fields=id,comments,likes,shares
```

Тут {post-id} - це ідентифікатор публікації.

Twitter API також надає можливість отримання інформації про коментарі та реакції на твіт:

```
GET/2/tweets/{tweetid}?expansions=author_id&tweet.fields=public_metrics, created_at
```

Тут {tweet-id} - це ідентифікатор твіта.

Для Instagram використовуємо Instagram Graph API:

```
GET/v13.0/{media-id}/comments?fields=id,text,like_count,timestamp
```

Тут {media-id} - це ідентифікатор медіа-файлу.

Для LinkedIn використовуємо LinkedIn API:

```
GET /v2/shares/{share-id}/comments
```

Тут {share-id} - це ідентифікатор медіа-файлу.

Бібліотека requests використовується для отримання кількості коментарів та реакцій на публікації в Facebook. Для роботи цієї програми потрібен дійсний токен доступу до Facebook Graph API та ідентифікатор публікації.

Лістинг 3.2:

```
import requests
def get_facebook_post_activity(api_token, post_id):
    base_url = "https://graph.facebook.com/v13.0"
    endpoint = f"/{post_id}"
    params = {
        "fields":
"comments.summary(true),reactions.summary(true)",
        "access_token": api_token
    response = requests.get(
        f"{base_url}{endpoint}",
        params=params
    if response.status_code == 200:
        post_data = response.json()
        comments_count = post_data.get("comments",
{}).get("summary", {}).get("total_count", 0)
        reactions_count = post_data.get("reactions",
{}).get("summary", {}).get("total_count", 0)
        print(f"Post {post_id} has {comments_count} comments and
{reactions_count} reactions.")
    else:
```

```
        print(f"Error {response.status_code}: {response.text}")
# Введіть свій токен доступу та ідентифікатор публікації
api_token = "your_facebook_api_token"
post_id = "your_post_id"
get_facebook_post_activity(api_token, post_id)
```

Цей код демонструє, як отримати кількість коментарів та реакцій для певної публікації на Facebook (лістинг 3.2).

3.1.3 Частота входження

Отримання частоти входження (активності) користувача в соціальних мережах може бути реалізовано різними способами, залежно від можливостей, які надає кожна конкретна платформа. Це може бути виміряно через кількість входжень за день, тиждень або місяць. Таку інформацію можна отримати через API.

Facebook:

```
GET /v13.0/{user-id}/feed?fields=id,message,created_time
```

Twitter:

```
GET /2/tweets?expansions=author_id&user.fields=username
```

Instagram:

```
GET/v13.0/{userid}/media?fields=id,caption,media_type,thumbnail_u
rl,timestamp
```

LinkedIn:

```
GET /v2/activities?q=authors&authors=urn:li:person:{user-id}
```

Тут {user-id} - це ідентифікатор користувача.

Отримання активності можуть змінюватися відносно часу, і важливо дотримуватися політики конфіденційності та вимог безпеки.

Аналіз частоти входження користувача в соціальних мережах може включати в себе отримання інформації про активність та публікації користувача на платформі. Код на лістингу 3.3 використовує Twitter API для отримання твітів користувача та аналізує їхню частоту входження за допомогою бібліотеки Counter.

Лістинг 3.3:

```
import requests
from collections import Counter
```

```
    def get_tweets(api_key, api_secret_key, access_token,
access_token_secret, username, count=100):
        base_url = "https://api.twitter.com"
        endpoint = f"/2/tweets"
        params = {
            "expansions": "author_id",
            "user.fields": "username",
            "tweet.fields": "created_at",
            "max_results": count,
            "screen_name": username
        headers = {
            "Authorization": f"Bearer {access_token}"
        response = requests.get(
            f"{base_url}{endpoint}",
            params=params,
            headers=headers
        if response.status_code == 200:
            tweets_data = response.json()
            return tweets_data.get("data", [])
        else:
            print(f"Error {response.status_code}: {response.text}")
            return []
    def analyze_frequency_of_posts(tweets):
        if not tweets:
            print("No tweets to analyze.")
            return
        created_at_list = [tweet.get("created_at", "") for tweet in
tweets]
        post_frequency_counter = Counter(created_at_list)
        print("Frequency of posts:")
        for created_at, frequency in post_frequency_counter.items():
            print(f"{created_at}: {frequency} posts")
    # Введіть свої ключі доступу до Twitter API та ім'я користувача
    api_key = "your_api_key"
    api_secret_key = "your_api_secret_key"
    access_token = "your_access_token"
    access_token_secret = "your_access_token_secret"
    twitter_username = "desired_twitter_username"
    # Отримання твітів користувача
    tweets = get_tweets(api_key, api_secret_key, access_token,
access_token_secret, twitter_username)
    # Аналіз частоти входження
    analyze_frequency_of_posts(tweets)
```

3.1.4 Тип використовуваних слів

Для аналізу типів використовуваних слів в текстових данних можна використовувати методи обробки природної мови (Natural Language Processing, NLP). Такі методи дозволяють визначати частини мови, визначати емоційний тон,

знаходити ключові слова тощо.

Визначимо частоту використання певних слів або груп слів, які можуть свідчити про настрій або емоційний стан користувача. Для цього можна використовувати бібліотеки NLP, такі як NLTK чи spaCy (лістинг 3.4).

Популярна бібліотека NLTK (Natural Language Toolkit) в Python для обробки природної мови має засоби для токенізації, визначення частин мови, аналізу емоцій тощо.

Лістинг 3.4:

```
import nltk
from nltk.tokenize import word_tcokenize
from nltk.probability import FreqDist
from nltk.corpus import stopwords
nltk.download('punkt')
nltk.download('stopwords')
def analyze_word_types(text):
    words = word_tokenize(text.lower())     # токенізація та приведення до нижнього регістру
    filtered_words = [word for word in words if word.isalnum() and word not in stopwords.words('english')]   # видалення стоп-слів та неалфавітних символів
    word_types = nltk.pos_tag(filtered_words)  # визначення частин мови
    print("Word types:")
    for word, word_type in word_types:
        print(f"{word}: {word_type}")
# Приклад використання
text_to_analyze = "This is an example sentence. It demonstrates the usage of NLTK for word type analysis."
analyze_word_types(text_to_analyze)
```

Ще одна потужна бібліотека spaCy для обробки природної мови, яка має широкий функціонал приведена у лістингу 3.5.

Лістинг 3.5:

```
import spacy
def analyze_word_types_spacy(text):
    nlp = spacy.load("en_core_web_sm")
    doc = nlp(text)
    print("Word types:")
    for token in doc:
        print(f"{token.text}: {token.pos_}")
```

```
    # Приклад використання
    text_to_analyze_spacy = "This is an example sentence. It
demonstrates the usage of spaCy for word type analysis."
    analyze_word_types_spacy(text_to_analyze_spacy)
```

Тут використовуються вбудовані алгоритми для визначення частин мови слова в тексті. Зазвичай такі аналізатори слів пов'язані з лінгвістичними моделями та можуть бути використані для різних мов. Однак, такі аналізатори не завжди можуть бути точними, особливо в разі складних текстів або текстів інших мов.

3.1.5 Створення набору даних для навчання

Для створення набору даних для навчання спочатку треба визначити конкретну задачу машинного навчання або обробки природної мови. При цьому, набір даних повинен відображати характеристики і особливості проблеми, які мають бути вирішені або вивчені. Дані потрібно розмістити, позначаючи, чи має користувач ознаки депресії, далі збалансувати набір даних, щоб уникнути перекосу в сторону одного класу.

Проблеми можуть бути поділені таким чином:

а) Класифікація текстів;

б) Виявлення емоцій в тексті;

в) Аналіз настрою в соціальних мережах;

г) Розпізнавання об'єктів на зображеннях.

1. Класифікація текстів. Для цього треба визначити теми чи категорії текстових документів. Створити набір даних із текстових документів різних тем або категорій. Це можуть бути новинні статті, блог-пости, книги або короткі тексти з різних джерел.

Для прикладу класифікації текстів новинних статей, можливо використати набір даних, де кожна новина вже привласнена до конкретної категорії. Наприклад, корпус новин Reuters, який включає набір новин та відповідних категорій. Набір даних Reuters доступний у бібліотеці nltk. Цей набір даних містить новинні статті, розподілені за різними категоріями. Він доступний для використання у великій

кількості досліджень у галузі обробки природної мови та машинного навчання. Набір даних Reuters є частиною NLTK (Natural Language Toolkit), яку ми використовуємо для тестування з текстовими даними. Кожна новина у наборі даних віднесена до однієї або декількох категорій. Категорії включають, наприклад, "earn" (заробіток), "acq" (придбання), "crude" (нафта), "corn" (кукурудза), "trade" (торгівля) та інші. Набір даних включає тисячі новинних текстів, що дозволяє використовувати його для навчання та тестування моделей машинного навчання, текстові дані у наборі даних представлені англійською мовою.

Лістинг 3.6:

```
import nltk
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
# Завантаження даних Reuters та вибір підмножини категорій
nltk.download('reuters')
from nltk.corpus import reuters
categories = ['earn', 'acq', 'crude', 'corn', 'trade']  # Вибір кількох категорій для прикладу
docs = [(category, reuters.raw(fileid)) for category in categories for fileid in reuters.fileids(category)]
# Розділення на тренувальний та тестовий набори
train_docs, test_docs = train_test_split(docs, test_size=0.2, random_state=42)
# Підготовка даних для моделі
train_texts, train_labels = zip(*train_docs)
test_texts, test_labels = zip(*test_docs)
# Векторизація текстів за допомогою TF-IDF
tfidf_vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')
X_train = tfidf_vectorizer.fit_transform(train_texts)
X_test = tfidf_vectorizer.transform(test_texts)
# Тренування класифікатора Naive Bayes
classifier = MultinomialNB()
classifier.fit(X_train, train_labels)
# Прогнозування на тестовому наборі
predictions = classifier.predict(X_test)
# Оцінка результатів
accuracy = accuracy_score(test_labels, predictions)
conf_matrix = confusion_matrix(test_labels, predictions)
class_report = classification_report(test_labels, predictions)
print(f"Accuracy: {accuracy}")
print(f"Confusion Matrix:\n{conf_matrix}")
```

```
print(f"Classification Report:\n{class_report}")
```

У цьому прикладі використовується класифікатор на основі моделі Мультиноміального наївного Баєса, який тренується на корпусі новин Reuters(лістинг 3.6).

Набір даних Reuters складається переважно з новинних статей, які зазвичай створюються у формалізованому стилі і орієнтовані на передачу об'єктивної інформації. Такі тексти не завжди мають виражені емоції, як це може бути у випадку особистих блогів, відгуків або соціальних мереж.

Однак можна використати методи обробки природної мови (Natural Language Processing, NLP) для виявлення емоційного тону в тексті. Зазвичай, для цього використовують аналіз емоційної окраски слів, визначення настрою та інші методи NLP.

2. Виявлення емоцій в тексті. Задача полягає у визначенні емоційного тону тексту (позитивний, негативний, нейтральний). Створений набір даних містить тексти, які є емоційно зарядженими з визначенням їхніх емоційних характеристик. Це може бути відгуки, соціальні медіа-пости, огляди тощо. Тут можна використовувати:

- словники, де слова мають призначені емоційні оцінки, і обчислення загального емоційного тону на основі цих оцінок;

- моделі машинного навчання для класифікації текстів за емоційним тоном. Такі моделі можуть бути навчені на відомих корпусах, де кожен текст має емоційний мітки;

- визначення, чи текст виражає позитивний, негативний чи нейтральний настрій.

Для прикладу використання словникового підходу слід використовувати словники, де кожне слово асоціюється з певною емоцією, і розраховувати емоційний тон тексту на основі суми значень слів (лістинг 3.7).

Лістинг 3.7:

```
def analyze_emotion(words):
    # Припустимий словник емоцій
    emotion_dict = {"happy": 1, "sad": -1, "excited": 1, "angry":
```

```
-1, "neutral": 0}
        # Розрахунок емоційного тону на основі словників
        total_emotion = sum(emotion_dict.get(word.lower(), 0) for word
in words)
        # Визначення емоційного тону
        if total_emotion > 0:
            return "Positive"
        elif total_emotion < 0:
            return "Negative"
        else:
            return "Neutral"
    # Приклад використання
    sample_words = ["happy", "excited", "sad", "neutral"]
    emotion_result = analyze_emotion(sample_words)
    print(f"Emotion: {emotion_result}")
```

У цьому прикладі слова асоціюються з емоційними значеннями, і їхні суми використовуються для визначення емоційного тону.

3. Аналіз настрою в соціальних мережах. Задача полягає у визначенні настрою користувачів у соціальних мережах. Тут оброблюються тексти з соціальних мереж, такі як твіти, статуси Facebook чи коментарі. Після чого треба зробити анотацію їхіх настроїв (позитивний, негативний, нейтральний).

Для аналізу настрою тексту будемо використовувати бібліотеку TextBlob для обробки природної мови (Natural Language Processing, NLP) в Python. TextBlob є простий інструмент для роботи з текстом та включає в себе аналіз емоцій та настрою. На лістингу 3.8 наведений код для аналізу настрою тексту.

Лістинг 3.8:

```
from textblob import TextBlob
def analyze_sentiment(text):
    analysis = TextBlob(text)
    # Визначення настрою
    if analysis.sentiment.polarity > 0:
        return "Positive"
    elif analysis.sentiment.polarity < 0:
        return "Negative"
    else:
        return "Neutral"
# Приклад використання
sample_text = "I love this product! It's amazing."
sentiment_result = analyze_sentiment(sample_text)
print(f"Sentiment: {sentiment_result}")
```

У цьому коді TextBlob використовується для визначення настрою тексту. Метод sentiment.polarity повертає числове значення від -1 до 1, де -1 вказує на негативний настрій, 1 - на позитивний, а 0 - на нейтральний.

4. Розпізнавання об'єктів на зображеннях. Розпізнавання депресії на зображеннях за допомогою виявлення об'єктів є складним завданням, оскільки депресія переважно є станом психічного здоров'я і не може бути однозначно визначена за виглядом об'єктів на фотографіях. Однак, деякі дослідження в області психічного здоров'я використовують аналіз зображень та мультимедійних вмістів для додаткового контексту та підтримки діагностики.

Розпізнавання депресії на зображеннях вимагає врахування приватності та етичних аспектів. Збір та аналіз зображень для цієї мети повинні дотримуватися високих стандартів конфіденційності та захисту особистої інформації.

В лістингу 3.9 наведено приклад, який використовує бібліотеку OpenCV для аналізу зображення та розпізнавання обличчя. Важливо зауважити, що цей приклад не призначений для діагностики депресії, але може бути використаний для розпізнавання основних елементів на зображенні, таких як обличчя.

Лістинг 3.9:

```
import cv2
def detect_faces(image_path):
    # Використання класифікатора обличчя Haarcascades
    face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
    # Зчитання зображення
    image = cv2.imread(image_path)
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # Знаходження облич на зображенні
    faces = face_cascade.detectMultiScale(gray_image, scaleFactor=1.3, minNeighbors=5)
    # Позначення облич на зображенні
    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 2)
    # Виведення зображення з обличчями
    cv2.imshow('Detected Faces', image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
# Виклик функції для зображення (змініть шлях до свого зображення)
image_path = 'path/to/your/image.jpg'
detect_faces(image_path)
```

Цей код використовує класифікатор обличчя Haarcascades для знаходження облич на зображенні та позначення їх прямокутниками.

3.2 Обробка текстової інформації

По результатах етапу збору, підготовки та моніторингу даних можна використовувати дерево рішень для ідентифікації ключових слів, фраз та патернів у текстах, які можуть вказувати на можливу депресію. Таке дерево рішень може бути побудоване на основі наступних кроків (рисунок 3.1).

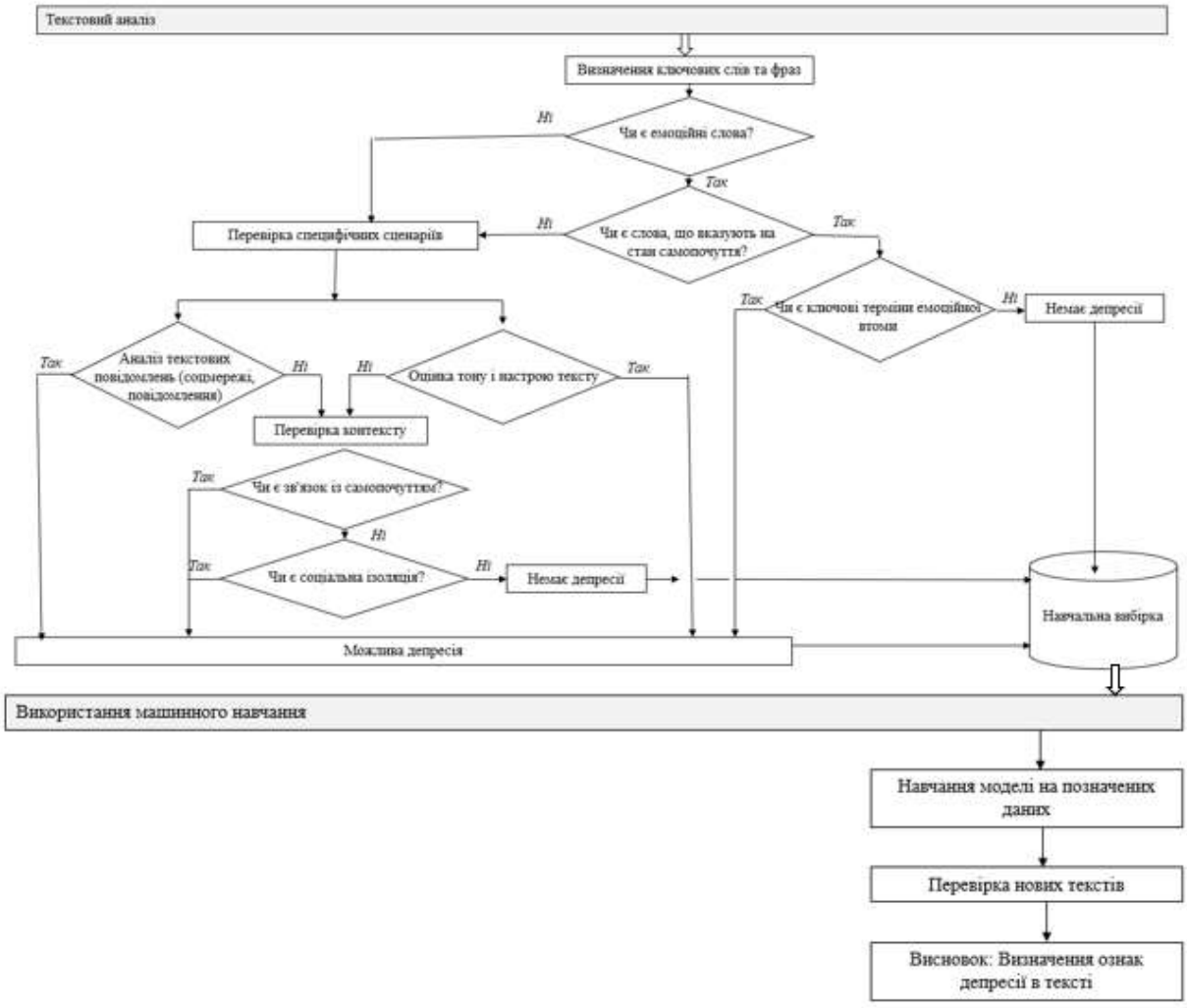


Рисунок 3.1 – Дерево рішень для ідентифікації ключових слів

Крок 1: Визначення ключових слів та фраз. На цьому етапі визначаються:

- слова, які пов'язані із емоційним станом, наприклад, тягар, безпорадність, печаль, втрата інтересу тощо;

- фрази, що вказують на стан самопочуття, наприклад, "не варто", "немає сенсу", "все погано", "ніколи не виходить", тощо;

- ключові терміни, що характеризують емоційну втомленість, наприклад, "виснажений", "вигорів", "втомився", тощо.

Крок 2: Перевірка специфічних сценаріїв, завдяки чому відбувається:

- аналіз текстових повідомлень: пошук інформації у повідомленнях, таких як соціальні мережі, де люди можуть висловлювати свої емоції;

- оцінка тону і настрою тексту, для чого використовуєть аналізатори тексту для визначення настрою тексту (позитивний, негативний, нейтральний).

Крок 3: Перевірка контексту, де визначається:

- зв'язок із самопочуттям, тобто перевіряється, чи є в тексті згадки про низьке самопочуття, невпевненість у собі, відсутність задоволення від життя;

- соціальна ізоляція, тобто перевіряється, чи вказують текстові дані на соціальну ізоляцію та віддаленість від інших.

Крок 4: Використання машинного навчання:

- навчання моделі на позначених даних, тобто тут використовуються позначені дані, які класифіковані як "з депресією" чи "без депресії", для тренування моделі машинного навчання;

- перевірка нових текстів за допомогою моделі, тобто тут використовуються модель для класифікації нових текстів та визначення ймовірності депресії.

В роботі пропонується система, яка може відправляти повідомлення або рекомендації, якщо виявлено ознаки депресії (рисунок 3.2.).

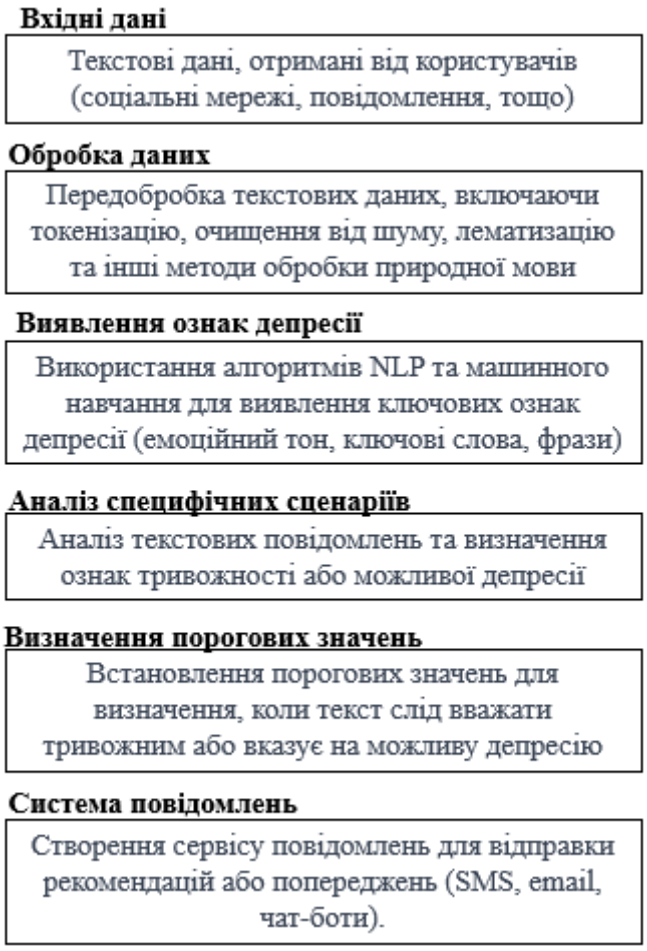


Рисунок 3.2 – Архітектура системи попередження та виявлення депресії

4 РОЗРОБКА ТА ДОСЛІДЖЕННЯ ЕКСПЕРЕМЕНТАЛЬНОЇ МОДЕЛІ

4.1 Постановка задачі

Пропонується створити комплексний підхід до аналізу постів у соціальних мережах для виявлення депресії шляхом поєднання дерев рішень, які можуть надати зрозумілі правила, що вказують на ознаки депресії, методу опорних векторів (SVM), який може розрізняти між двома класами (наприклад, "з депресією" та "без депресії") та інтелектуальних агентів (ІА), які можуть здійснювати моніторинг поведінки користувачів [16]. Для розробки запропонованої гібридної структури розглядаються наступні кроки [17. 18, 19, 20].

Збір, підготовка та моніторинг даних. На цьому етапі дерево рішень використовується для визначення ключових факторів, які можуть вказувати на можливість наявності депресії у користувачів з урахуванням текстових ознак, сентиментів, активності та інших параметрів. SVM використовується для класифікації текстових даних та для визначення, які з них можуть бути індикаторами депресії.

Інтелектуальні агенти можуть взаємодіяти із соціальними мережами, здійснювати моніторинг поведінки користувачів та збирати дані для подальшого аналізу. Вони можуть виявляти нові повідомлення та взаємодії користувачів, що можуть бути важливими для виявлення ознак депресії. На рисунку 4.1 показаний робочий процес запропонованої гібридної техніки.

Виявлення ознак депресії. Дерево рішень застосовується для ідентифікації ключових слів, фраз та патернів у текстах, що можуть вказувати на можливу депресію. SVM застосовується для аналізу сентименту та визначення тону текстів, спрямованого на визначення негативних емоцій.

Інтелектуальні агенти можуть допомагати у розширенні ознак, які використовуються для класифікації. Вони можуть виявляти контекстуальні ознаки, що не завжди очевидні в тексті повідомлення, такі як частота інтеракцій, часові

параметри, геолокація та інші.

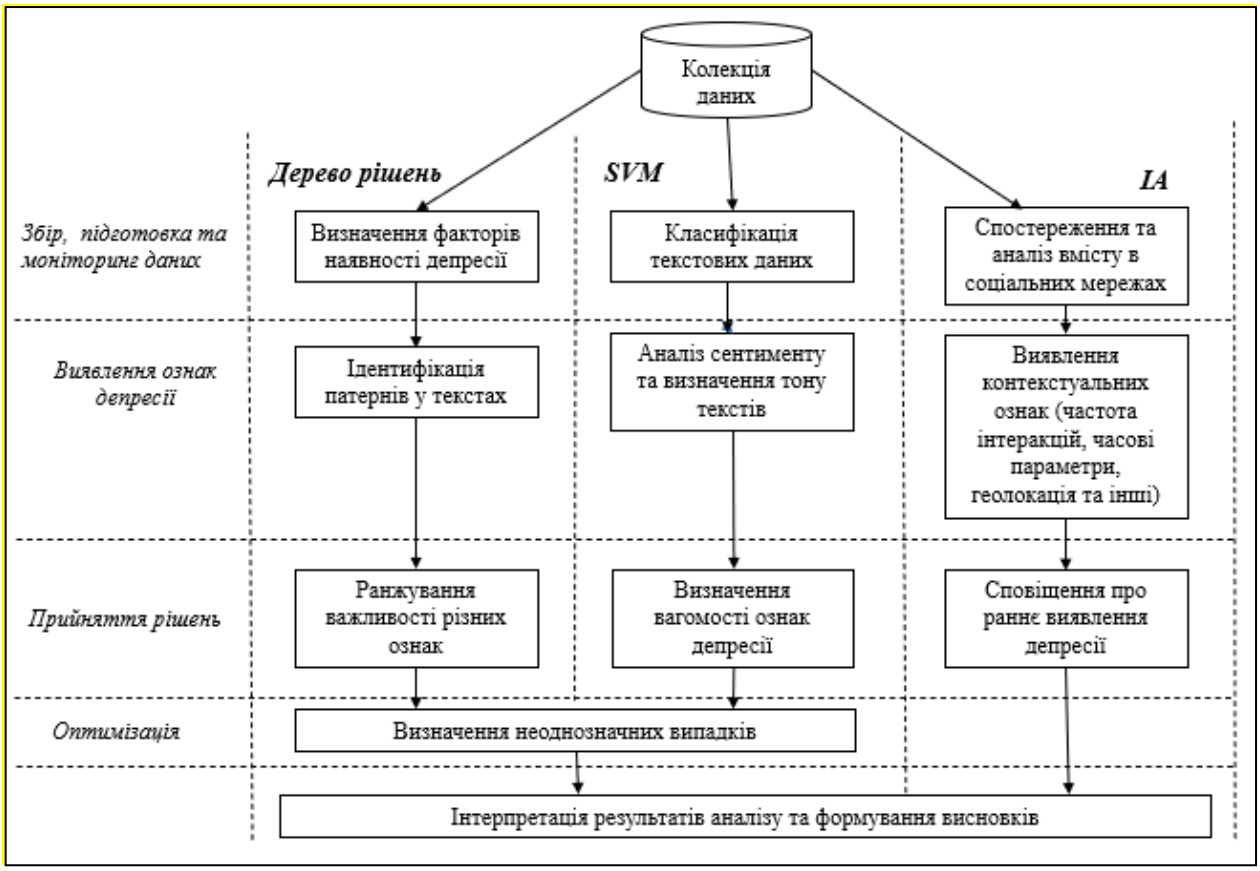


Рисунок 4.1 – Гібридна техніка аналізу виявлення депресії в соціальних мережах

Прийняття рішень. Для ранжування важливості різних ознак та визначення, які з них слід враховувати в першу чергу при виявленні депресії, використовується дерево рішень. Для створення моделей, які враховують взаємодію різних ознак та їх вагомість при визначенні класу, використовується SVM. Інтелектуальні агенти можуть бути включені в систему раннього виявлення депресії, виявляючи зміни в психічному стані користувачів та надаючи попередження або рекомендації для втручання.

Оптимізація. Дерева рішень та SVM мають свої параметри, які можна оптимізувати для покращення результатів. Комбінування обох методів може включати в себе оптимізацію цих параметрів з метою визначення неоднозначних випадків, підвищення точності та стійкості моделі.

Інтерпретація результатів аналізу постів у соціальних мережах для виявлення депресії та формування висновків є критичним етапом, який дозволяє зрозуміти, які ознаки або патерни вказують на можливість депресії у користувачів. Інтерпретація включає як кількісний, так і якісний аналіз. Вона є ключовою для того, щоб забезпечити висновки, які є релевантними та корисними для подальших дій або втручань у психічне здоров'я користувачів

Такий підхід дозволяє комбінувати сильні сторони кожного методу для отримання комплексної системи, що може ефективно виявляти депресію у текстових даних соціальних мереж, покращити загальну точність та робустність моделі. Загальний внесок інтелектуальних агентів полягає в їхній здатності пристосовуватися до змінних умов та допомагати в автоматизації інтелектуальних завдань, що може полегшити процес виявлення депресії на основі аналізу постів у соціальних мережах.

Висновки. Аналіз постів із соціальних мереж з використанням машинного навчання може допомогти:

- виявити ознаки депресії у великої кількості людей;
- виявити ранні ознаки депресії, навіть до того, як вони можуть бути діагностовано традиційними методами;
- отримати більше інформації про конкретні контексти, соціальну взаємодію та фактори, що впливають на психічне здоров'я та емоційний стан кожної конкретної особи;
- визначати осіб, яким може допомогти раннє втручання або підтримка;
- зрозуміти психічне здоров'я в цьому онлайн-середовища, яке стає все більш важливим.

4.2 Процес навчання згорткової нейронної мережі

Процес навчання згорткової нейронної мережі (CNN) включає наступні кроки:

Підготовка даних: дані, що складаються з коментарів у соціальних мережах про депресію, підготовлюються для використання в моделі. Текст перетворюється на послідовності числових векторів за допомогою вбудовування слів та інших методів

попередньої обробки.

Створення моделі: визначається архітектура CNN, включаючи згорткові шари, пулінгові шари та повнозв'язані шари. Встановлюються гіперпараметри, такі як розмір фільтрів, кількість фільтрів, розмір пулінгу та кількість нейронів в повнозв'язаному шарі.

Компіляція моделі: визначаються функція втрат (loss function), оптимізатор та метрики, які використовуються під час навчання. У випадку бінарної класифікації, часто використовується бінарна кросс-ентропія (binary cross-entropy) як функція втрати.

Навчання моделі: модель навчається на підготовлених тренувальних даних за допомогою функції fit(). Під час навчання модель змінює свої ваги та зсуви, щоб оптимізувати функцію втрат та покращити точність класифікації. Зазвичай, дані розділяють на тренувальний та перевірочний набори, щоб оцінити продуктивність моделі на навидених даних.

Оцінка та налаштування моделі: після навчання модель оцінюється на тестовому наборі даних, які раніше не бачила. Виконується аналіз результатів, такий як точність, втрати та інші метрики, для оцінки ефективності моделі. Можливо, потрібно налаштувати гіперпараметри або внести зміни в архітектуру моделі для поліпшення її продуктивності.

Це загальний опис процесу навчання згорткової нейронної мережі для класифікації коментарів про депресію. Згорткові нейронні мережі демонструють хороші результати в обробці текстових даних, виявленні зв'язків та класифікації, що робить їх ефективними інструментами для аналізу соціальних мереж і виявлення депресії. На рисунку 4.2 представлено структуру згорткової нейронної мережі.

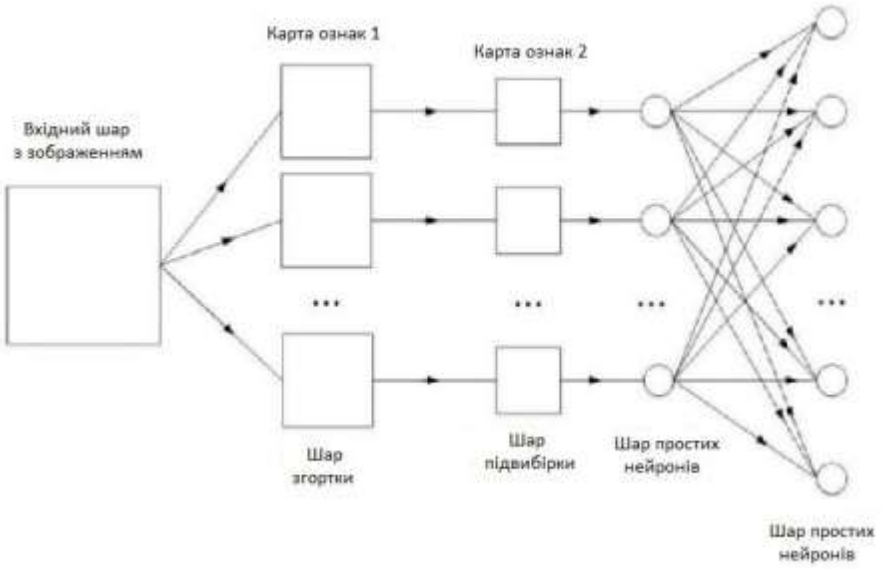


Рисунок 4.2 – Структура згорткової нейронної мережі

4.3 Датасети

Набори даних, відомі як датасети, використовуються для аналізу, вивчення або навчання моделей машинного навчання. Датасети можуть зберігати інформацію в різних форматах, таких як текстові документи, звукові файли, таблиці та зображення.

За допомогою веб-скрапінгу, відкритих баз даних, власних досліджень та інших методів можна отримати датасети. Вони можуть бути структурованими або неструктурованими.

Оптимальний датасет має бути досить різноманітним, щоб моделі машинного навчання могли узагальнювати дані, а також бути чистими та вільними від помилок. Крім того, важливо, щоб датасет відповідав задачі, яку необхідно вирішити.

Датасети є основою навчання моделей машинного навчання, тому вибір і підготовка датасету є важливими кроками в процесі розробки моделі.

Для класифікації коментарів про депресію важливо мати відповідний датасет, який містить дані, що відображають різні аспекти та варіації депресивних мислей. При виборі датасету можуть враховуватись такі фактори:

- репрезентативність: датасет повинен бути репрезентативним для цільової групи, тобто містити коментарі, які відображають депресивні мислі та досвід людей, що стикаються з депресією;

- розмір та різноманітність: бажано мати датасет достатнього розміру та з достатньою різноманітністю коментарів, що дозволить побудувати модель з високою точністю та універсальністю;

- якість та достовірність: датасет повинен бути якісним та достовірним, щоб гарантувати точність результатів моделі. Важливо перевірити джерело датасету та переконатися, що він був створений або анотований професіоналами або експертами у сфері депресії;

- доступність та ліцензування: важливо переконатися, що датасет доступний для використання згідно з вимогами та ліцензійними умовами, що задовольняють зазначеним потребам;

- обробка та очищення: датасет може потребувати попередньої обробки та очищення для видалення непотрібної інформації, виправлення помилок та підготовки даних для моделювання.

Враховуючі усі ці фактори було обрано датасет з сайту Kaggle. На рисунку 4.3 представлена сторінка з датасетом.

Рисунок 4.3 – Датасет на сайті Kaggle

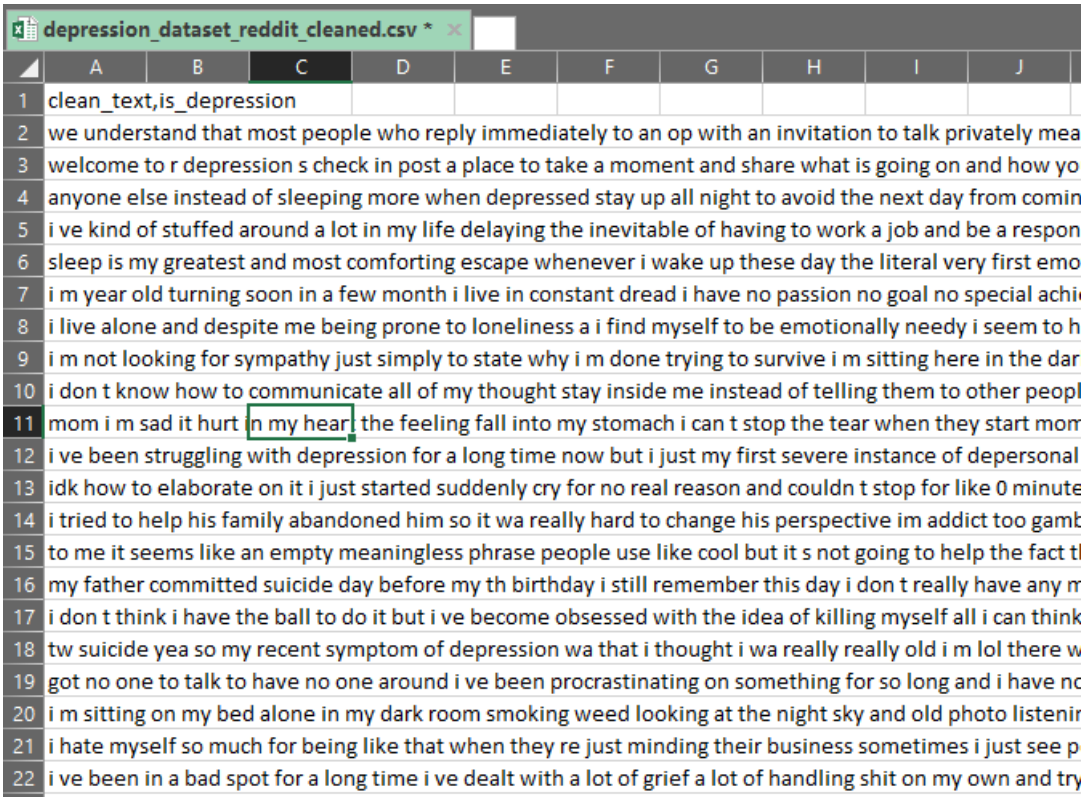
У вказаному датасеті містяться 7732 записи з двома полями: "clean_text" (очищений текст коментаря) і "is_depression" (вказує, чи містить коментар депресію).

Наприклад, ось декілька прикладів коментарів з датасету:

Коментар з депресією: "i m starting to lose hope i feel like i m on auto pilot i m not living i m existing" (1)

Коментар без депресії: "a bad nite for the favorite team astros and spartan lose the nite out with t w wa good" (0)

Кожен коментар відображається у полі "clean_text", а значення поля "is_depression" вказує, чи присутня депресія у коментарі. Значення "1" вказує на наявність депресії, а значення "0" означає відсутність депресії. На рисунку 4.4 представлено вигляд файлу датасету.


depression_dataset_reddit_cleaned.csv *

	A	B	C	D	E	F	G	H	I	J
1	clean_text,is_depression									
2	we understand that most people who reply immediately to an op with an invitation to talk privately mea									
3	welcome to r depression s check in post a place to take a moment and share what is going on and how yo									
4	anyone else instead of sleeping more when depressed stay up all night to avoid the next day from comin									
5	i ve kind of stuffed around a lot in my life delaying the inevitable of having to work a job and be a respon									
6	sleep is my greatest and most comforting escape whenever i wake up these day the literal very first emo									
7	i m year old turning soon in a few month i live in constant dread i have no passion no goal no special achi									
8	i live alone and despite me being prone to loneliness a i find myself to be emotionally needy i seem to h									
9	i m not looking for sympathy just simply to state why i m done trying to survive i m sitting here in the dar									
10	i don t know how to communicate all of my thought stay inside me instead of telling them to other peopl									
11	mom i m sad it hurt in my hear the feeling fall into my stomach i can t stop the tear when they start mor									
12	i ve been struggling with depression for a long time now but i just my first severe instance of depersonal									
13	idk how to elaborate on it i just started suddenly cry for no real reason and couldn t stop for like 0 minute									
14	i tried to help his family abandoned him so it wa really hard to change his perspective im addict too gamb									
15	to me it seems like an empty meaningless phrase people use like cool but it s not going to help the fact tl									
16	my father committed suicide day before my th birthday i still remember this day i don t really have any m									
17	i don t think i have the ball to do it but i ve become obsessed with the idea of killing myself all i can think									
18	tw suicide yea so my recent symptom of depression wa that i thought i wa really really old i m lol there w									
19	got no one to talk to have no one around i ve been procrastinating on something for so long and i have nc									
20	i m sitting on my bed alone in my dark room smoking weed looking at the night sky and old photo listenir									
21	i hate myself so much for being like that when they re just minding their business sometimes i just see p									
22	i ve been in a bad spot for a long time i ve dealt with a lot of grief a lot of handling shit on my own and try									

Рисунок 4.4 – Вигляд файлу датасету

4.4 Алгоритм навчання та збереження нейронної мережі

В програмі використовуються навчена та збережена нейронна мережа. Навчання та збереження мережі виконується за таким алгоритмом:

- завантаження датасету: зчитати датасет з CSV-файлу;

- підготовка даних: розділити дані на текстові коментарі та відповідні мітки, які вказують, чи є коментар пов'язаним з депресією;
- токенізація: перетворити текстові коментарі на послідовності чисел, використовуючи токенізатор. Кожне слово в коментарі буде представлене числовим індексом;
- заповнення даних: додати заповнення (padding) до послідовностей чисел, щоб всі коментарі мали однакову довжину;
- створення моделі: створити модель згорткової нейронної мережі з ембедінг-шаром, згортковими шарами, шарами пулінгу та повнозв'язаними шарами;
- компіляція моделі: визначити функцію втрати, оптимізатор та метрики для моделі;
- навчання моделі: навчити модель на тренувальних даних протягом певної кількості епох;
- оцінка моделі: оцінити точність моделі на тестових даних;
- збереження моделі: зберегти навчену модель для подальшого використання.

Алгоритм включає завантаження даних, підготовку, побудову та навчання моделі згорткової нейронної мережі, оцінку та збереження моделі. Використовуються різні техніки, такі як токенізація, заповнення даних та згорткові шари, щоб обробити текстові дані та виконати класифікацію коментарів на наявність депресії.

Для правильного визначення депресивних коментарів точність моделі має бути приблизно більше 90%.

4.5 Розробка та навчання нейронної мережі

Тестовий застосунок буде розроблено у середовищі розробки Google Collab. Google Colab – це безкоштовна хмарна платформа, розроблена компанією Google, яка надає можливість виконувати та редагувати код на мові програмування Python у веб-браузері без необхідності встановлення додаткового програмного забезпечення на локальний комп'ютер. Він базується на середовищі Jupyter Notebook і має ряд переваг для розробки і виконання проектів з машинного навчання та аналізу даних. Основні

особливості Google Colab включають:

Безкоштовне використання та доступ до обчислювальних ресурсів: Google Colab надає обчислювальні ресурси, такі як центральний процесор (CPU) та графічний процесор (GPU), безкоштовно. Можна використовувати потужні обчислювальні ресурси Google для швидкого навчання моделей машинного навчання та обробки великих обсягів даних;

Зручність спільної роботи та обміну даними: Можна легко ділитися своїми проектами Colab з колегами або співробітниками, дозволяючи їм переглядати, редагувати або навіть виконувати ваш код. Крім того, можна імпортувати та експортувати дані з різних джерел, таких як Google Drive або GitHub, для зручного обміну даними та кодом;

Підтримка Jupyter Notebook: Google Colab заснований на середовищі Jupyter Notebook, що дозволяє створювати інтерактивні та документовані ноутбуки. Можна додавати текстові описи, вставляти код, візуалізації та графіки, а також виконувати код почастинно, роблячи розробку більш зрозумілою та легко зрозумілою;

Велика кількість підтримуваних бібліотек: Google Colab підтримує багато популярних бібліотек Python, включаючи TensorFlow, Keras, NumPy, Pandas, Matplotlib та багато інших. Можна легко імпортувати ці бібліотеки та використовувати їх для аналізу даних, побудови моделей машинного навчання та виконання різних обчислювальних завдань;

Гнучкість середовища: Google Colab надає можливість встановлювати додаткові пакети та бібліотеки Python за необхідності. Можна виконувати команди оболонки, встановлювати залежності та налаштовувати середовище згідно зі своїми потребами.

Google Colab є потужним інструментом для розробки та виконання проектів з машинного навчання, забезпечуючи зручне середовище для експериментів, навчання моделей та обробки даних без необхідності налаштування локального середовища розробки. Його безкоштовна природа та потужність обчислювальних ресурсів Google роблять його популярним вибором для багатьох дослідників та розробників у галузі машинного навчання.

Перш за все завантажимо заздалегідь навчену модель та токенізатор. Код представлений у лістингу 4.1 виконує наступні дії:

Завантаження заздалегідь навченої моделі, зчитує модель з файлу depression_model.h5, яка була попередньо збережена після навчання.

Завантаження токенізатора – створює об'єкт токенізатора з Tokenizer та задає його параметри, такі як num_words, який вказує максимальну кількість слів, що будуть використовуватися у токенізаторі.

У цьому випадку num_words=10000 означає, що будуть використовуватися лише 10 000 найпоширеніших слів. Крім того, oov_token='<OOV>' встановлює значення, яке буде використовуватися для невідомих слів, тобто слів, які не зустрічалися в тренувальних даних.

Цей код дозволяє завантажити навчену модель та токенізатор для подальшого використання без необхідності повторного навчання або створення нового токенізатора.

Лістинг 4.1 – Код завантаження моделі та створення токенізатора:

```
model = tf.keras.models.load_model('depression_model.h5')

tokenizer = Tokenizer(num_words=10000, oov_token='<OOV>')
tokenizer.fit_on_texts([''])
```

Для предсказування депресії у коментарі напиши функцію, яка буде мати ім’я – detect_depression.

Код представлений у лістингу 4.2 представляє функцію detect_depression, яка приймає фразу в якості вхідного параметра і визначає, чи вказує ця фраза на наявність депресії.

Опис дій функції:

- sequence = tokenizer.texts_to_sequences([sentence]): Фраза перетворюється на послідовність чисел, використовуючи токенізатор;

- sequence_padded = pad_sequences(sequence, maxlen=100, padding='post', truncating='post'): Послідовність чисел доповнюється (заповнюється нулями або обрізається) до максимальної довжини 100;

- prediction = model.predict(sequence_padded)[0][0]: Модель передбачає ймовірність депресії для вхідної послідовності чисел.

Залежно від значення prediction, виводиться відповідне повідомлення: "Введена фраза указує на депресію" або "Введена фраза не указує на депресію".

Лістинг 4.2 – Код функції «detect_depression»:

```
# Функція визначення наявності депресії у фразі
def detect_depression(sentence):

    # Перетворення фрази на послідовність чисел
    sequence = tokenizer.texts_to_sequences([sentence])
    # Додавання заповнення до максимальної довжини
    sequence_padded = pad_sequences(sequence, maxlen=100,
padding='post', truncating='post')
    # Передбачення ймовірності депресії
    prediction = model.predict(sequence_padded)[0][0]
    # Висновок результату
    if prediction >= 0.5:
        print('Введена фраза вказує на депресію.')
    else:
        print('Введена фраза не вказує на депресію.')
```

В останньому блоці коду введення фрази користувачем та визначення депресії. В цьому фрагменті коду виконується безкінечний цикл, де користувачу пропонується ввести фразу. Код очікує введення фрази від користувача і зберігає його у змінну user_input.

Після введення фрази, код перевіряє, чи користувач ввів "вихід" для виходу з програми. Якщо користувач ввів "вихід", цикл припиняється і програма завершує свою роботу. Якщо користувач ввів будь-яку іншу фразу, викликається функція detect_depression(user_input).

Очікування введення фрази від користувача та визначення депресії здійснюються у циклі, що дозволяє користувачеві аналізувати різні фрази та отримувати відповіді щодо наявності депресії у введеному тексті. Код представлений у лістингу 4.3.

Лістинг 4.3 – Код прогнозування наявності депресії у користувача:

```
# Введення фрази користувачем та визначення депресії
while True:
    user_input = input('Введіть фразу (або введіть "вихід" для
```

```
виходу): ')
          if user_input.lower() == 'вихід':
              break
          detect_depression(user_input)
```

Навчання моделі починається с завантаження тестових даних, а саме датасету. Код представлений у лістингу 4.4 відповідає за завантаження датасету з файлу 'depression_dataset_reddit_cleaned.csv'. Він використовує бібліотеку Pandas, щоб прочитати CSV-файл та зберегти дані в об'єкт даних під назвою 'data'.

Після завантаження датасету, змінні 'sentences' і 'labels' містять текстові коментарі та відповідні мітки відповідно. 'sentences' містить значення стовпця 'clean_text', який містить текст коментарів, а 'labels' містить значення стовпця 'is_depression', що вказує, чи містить коментар депресію.

Лістинг 4.4 – Код завантаження датасету:

```
data = pd.read_csv('depression_dataset_reddit_cleaned.csv')
sentences = data['clean_text'].values
labels = data['is_depression'].values
```

Далі необхідно розділити дані на навчальну та тестові виборки.

У лістингу 4.5 представлено код для розділення даних. Змінні "train_sentences" і "train_labels" будуть містити відібрані коментарі та відповідні мітки для тренувальної вибірки.

Змінні "test_sentences" і "test_labels" будуть містити відібрані коментарі та відповідні мітки для тестової вибірки.

Функція train_test_split розподіляє дані у випадковому порядку, використовуючи вказаний random_state (для відтворюваності результатів), та встановлює розмір тестової вибірки в 30% від загальної кількості даних.

Таким чином, дані розділяються на тренувальну та тестову вибірки зі збереженням відповідних коментарів та міток для подальшого використання в моделі.

Лістинг 4.5 – Код для розділення:

```
    train_sentences, test_sentences, train_labels, test_labels =
train_test_split(sentences, labels, test_size=0.3, random_state=42)
```

Далі необхідно створити три важливі речі для роботи з текстом, а саме: створити токенізатор, перетворити коментарі у послідовності чисел, заповнити до максимальної довжини.

У лістингу 4.6 представлено код для виконання описаних дій. Цей код виконує підготовчі кроки для обробки текстових даних перед подачею їх на вхід моделі згорткової нейронної мережі.

Створення токенізатора: створюється об'єкт токенізатора з максимальною кількістю слів (num_words) рівною 10000. Також встановлюється спеціальний маркер для невідомих слів (oov_token), який використовується, коли зустрічається слово, яке не має індексу в словнику.

Перетворення коментарів у послідовності чисел: використовуючи токенізатор, текстові речення з навчального та тестового наборів даних перетворюються на послідовності числових індексів. Кожному слову присвоюється числовий індекс згідно зі словником токенізатора.

Додавання заповнення до максимальної довжини: послідовності чисел заповнюються нулями або обрізаються до максимальної довжини (max_length), щоб усі послідовності мали однакову довжину. Заповнення здійснюється після тексту (padding='post') та обрізання здійснюється з кінця послідовності (truncating='post').

Цей код допомагає підготувати текстові дані для подальшого використання у моделі згорткової нейронної мережі шляхом перетворення тексту на послідовності чисел та забезпечення їх однакової довжини шляхом додавання заповнення.

Лістинг 4.6 – Код для виконання описаних дій:

```
# Створення токенізатора
tokenizer = Tokenizer(num_words=10000, oov_token='<OOV>')
tokenizer.fit_on_texts(train_sentences)
# Перетворення пропозицій у послідовності чисел
train_sequences = tokenizer.texts_to_sequences(train_sentences)
test_sequences = tokenizer.texts_to_sequences(test_sentences)
# Додавання заповнення до максимальної довжини
max_length = 100
train_padded = pad_sequences(train_sequences, maxlen=max_length, padding='post', truncating='post')
```

```
    test_padded = pad_sequences(test_sequences, maxlen=max_length,
padding='post', truncating='post')
```

Наступним кроком буде створення структури згорткової нейронної мережі. Опис структури моделі:

- вбудовуваний шар (embedding): вхідний шар, який перетворює числові індекси слів в вектори фіксованої довжини;

- використовується словнику розміру 10000, кожне слово має векторний представник розміру 16. вхідна послідовність має довжину, вказану параметром input_length;

- згортковий шар (conv1d): використовує 128 фільтрів ширини 5 для виявлення локальних особливостей у векторах. активаційна функція 'relu' застосовується для активації виявлених ознак;

- пулінговий шар (globalmaxpooling1d): обирає найбільш значущі значення з усіх отриманих ознак,стискаючи їх у вектор фіксованої довжини. застосовується для виділення найбільш важливих ознак;

- повнозв'язний шар (dense): має 64 нейрони з активаційною функцією 'relu'.виконує обробку виділених ознак для подальшої класифікації;

- вихідний шар (dense): має 1 нейрон з активаційною функцією 'sigmoid'. Використовується для вирішення задачі бінарної класифікації, де вихідне значення вказує на ймовірність наявності депресії (1) або її відсутності (0). У лістингу 4.7 представлено код структури моделі нейронної мережі.

Лістинг 4.7 – Код структури моделі нейронної мережі:

```
    # Створення моделі згорткової нейронної мережі
    model = tf.keras.models.Sequential([
        tf.keras.layers.Embedding(input_dim=10000,      output_dim=16,
input_length=max_length),
        tf.keras.layers.Conv1D(128, 5, activation='relu'),
        tf.keras.layers.GlobalMaxPooling1D(),
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ])
```

Наступний крок простий, але дуже важливий. Виконується навчання моделі,

визначення точності та збереження моделі для наступного кроку. У лістингу 4.8 представлено код тренування та збереження моделі. Компіляція моделі: цей код виконує компіляцію моделі перед початком навчання. Він встановлює функцію втрати (binary_crossentropy), оптимізатор (adam) і метрики (accuracy), які будуть використовуватися під час навчання моделі.

Навчання моделі: цей код виконує навчання моделі на тренувальних даних протягом 50 епох. Він використовує train_padded (перетворені та заповнені тренувальні дані) та train_labels (мітки для тренувальних даних) як вхідні дані. Крім того, вказані test_padded та test_labels використовуються для перевірки моделі під час навчання.

Оцінка точності моделі: цей код оцінює точність моделі на тестових даних, використовуючи test_padded (перетворені та заповнені тестові дані) та test_labels (мітки для тестових даних). Результати оцінки зберігаються в змінну accuracy, яка використовується для виведення точності моделі у відсотках.

Збереження моделі: цей код зберігає навчену модель у файл "depression_model.h5". Файл містить всю архітектуру моделі, ваги та інші параметри, необхідні для повторного використання моделі пізніше. Після збереження моделі виводиться повідомлення про успішне збереження.

Лістинг 4.8 – Код тренування та збереження моделі:

```
    # Компіляція моделі
    model.compile(loss='binary_crossentropy',        optimizer='adam',
metrics=['accuracy'])
    # Навчання моделі
    model.fit(train_padded,          train_labels,          epochs=50,
validation_data=(test_padded, test_labels))
    # Оцінка точності моделі
    _, accuracy = model.evaluate(test_padded, test_labels)
    print('Accuracy: %.2f' % (accuracy * 100))
    model.save('depression_model.h5')
    print('Модель збережена у файл "depression_model.h5"')
```

На рисунку 4.5 представлено результати навчання моделі.

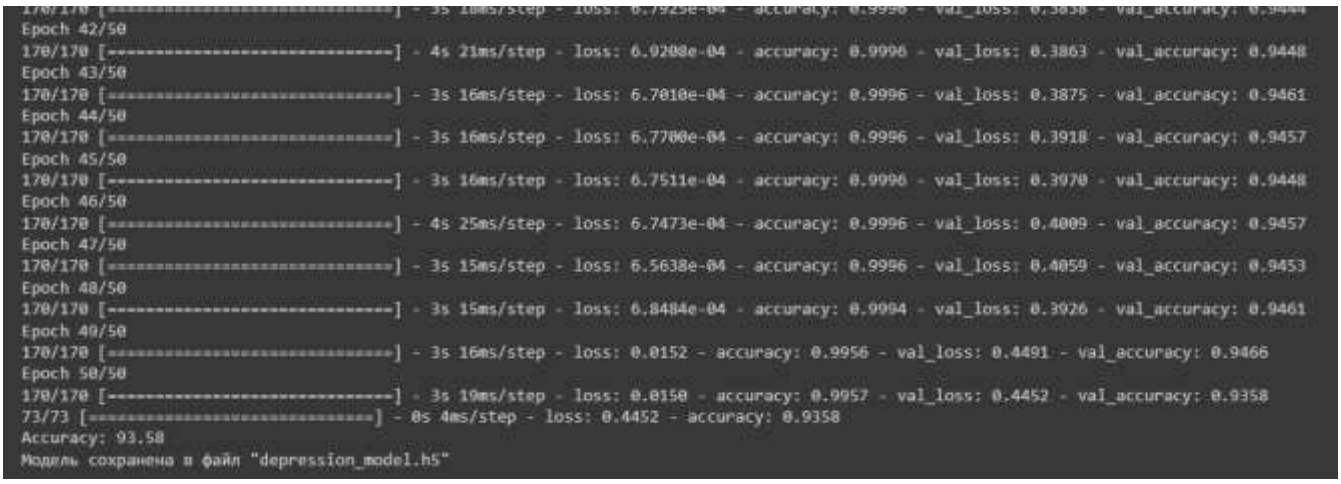


Рисунок 4.5 – Результати навчання моделі

Результати навчання моделі показують зміну значень функції втрати (loss) та точності (accuracy) протягом 50 епох. Після останньої епохи, модель була оцінена на тестових даних, де вона показала точність 93.58%. Також була збережена модель у файл "depression_model.h5".

Отже, модель досягла високої точності на тестових даних, що свідчить про її ефективність у класифікації коментарів з депресією. Значення функції втрати на тренувальних та валідаційних даних зменшувались протягом епох, що свідчить про прогресивне навчання моделі.

Збереження моделі дозволяє використовувати її у майбутньому без необхідності повторного навчання. Отримані результати свідчать про успішність моделі у вирішенні завдання класифікації депресивних коментарів на основі наданого датасету.

В результаті запуску та введення фрази («I want to live in a world filled with happiness and spending time with loved ones» - «Я хочу жити у світі, наповненому щастям і проводити час з коханими»), яка не вказує на депресію отримуємо правильну відповідь нейронної мережі.

Результат виконання представлено на рисунку 4.6.

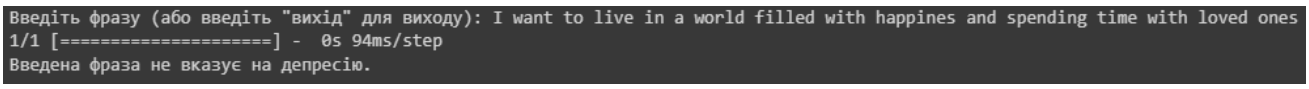


Рисунок 4.6 – Коректний результат нейронної мережі

На рисунку 4.7 представлено введення фрази «I want to die» на що маємо результат про депресію.

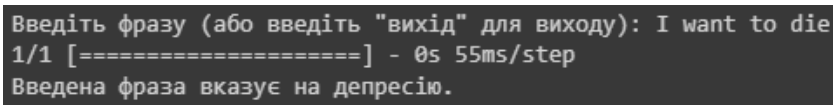


Рисунок 4.7 – Випадок коли фраза вказує на депресію

Отже програмний застосунок вірно працює та виконує покладені на нього вимоги.

ВИСНОВКИ

Діагностика депресії на основі даних із соціальних мереж широко досліджується в усьому світі за допомогою різноманітних мережевих сайтів, наборів даних, методів виділення лінгвістичних ознак, алгоритмів машинного навчання, обчислювальних інструментів і методів статистичного аналізу. Результати, отримані в більшості досліджень, свідчать про те, що використання нових цифрових інструментів, пов’язаних із психічним здоров’ям, є стимулом для продовження досліджень у цій сфері.

У роботі було проаналізовано існуючи статті та дослідження в цій області. В них розробники використовували різні соціальні мережі для формуванн даних та методи та моделі навчання такі як метод опорних векторів (SVM), DISVM яка використовує нейронну мережу для вилучення ознак і класифікації даних і так далі.

Визначено за допомогою порівняльного аналізу що саме мова програмування Python є більш придатною до роботи з машинним навчанням. Це обумовлено великою кількостю бібліотек та інструментів для обробки текстових даних у порівнянні з іншими мовами розробки.

Також в роботі проаналізовано існуючи методи попередньої обробки даних що дають змогу надати уяви про те які методи більш придатні для вирішення теми роботи. Так само визначенні вимоги що до набору даних що дає змогу побудувати дерево рішень для ідентифікації ключових слів та фраз текстах.

Результати показують, що модель, розроблена для визначення ознак депресії, є дуже точною. Це показує, що машинне навчання може виявити депресію на основі аналізу текстових даних із соціальних мереж. Такий підхід може значно вплинути на раннє виявлення та попередження депресії серед користувачів соціальних мереж.

В цілому, дані дослідження підтверджують можливість використання машинного навчання для визначення депресії за допомогою аналізу постів соціальних мереж. Автоматизовані методи виявлення депресії можуть покращити діагностику, попередження та підтримку осіб, які стикаються з депресією

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аксак Н.Г., Агібалов Р.І. Дослідження постів із соціальних мереж для визначення депресії на основі машинного навчання. VI International Scientific and Theoretical Conference «The current state of development of world science: characteristics and features». Lisbon, Portugal. 2023 p C. 118-122. ISBN: 979-8-88955-776-0 (series) DOI:https://doi.org/10.36074/scientia-15.12.2023

2. Giuntini, F. T., Cazzolato, M. T., dos Reis, M. D. J. D., Campbell, A. T., Traina, A. J., & Ueyama, J. (2020). A review on recognizing depression in social networks: challenges and opportunities. *Journal of Ambient Intelligence and Humanized Computing*, *11*, 4713-4729.

3. Guntuku, S. C., Yaden, D. B., Kern, M. L., Ungar, L. H., & Eichstaedt, J. C. (2017). Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, *18*, 43-49.

4. William, D., & Suhartono, D. (2021). Text-based depression detection on social media posts: A systematic literature review. *Procedia Computer Science*, *179*, 582-589.

5. Aguilera, J., Farías, D. I. H., Ortega-Mendoza, R. M., & Montes-y-Gómez, M. (2021). Depression and anorexia detection in social media as a one-class classification problem. *Applied Intelligence*, *51*, 6088-6103.

6. Liu, D., Feng, X. L., Ahmed, F., Shahid, M., & Guo, J. (2022). Detecting and measuring depression on social media using a machine learning approach: systematic review. *JMIR Mental Health*, *9*(3), e27244.

7. Kumar M, Dredze M, Coppersmith G, De Choudhury M. Detecting Changes in Suicide Content Manifested in Social Media Following Celebrity Suicides. *HT ACM Conf Hypertext Soc Media* 2015 Sep; 2015:85-94

8. Shen G, Jia J, Nie L, Feng F, Zhang C, Hu T, et al. Depression detection via harvesting social media: A multimodal dictionary learning solution. 2017 Presented at: *the Twenty-Sixth International Joint Conference on Artificial Intelligence* (IJCAI-17); 19-25 August 2017; Melbourne, Australia.

9. Sadeque F, Xu D, Bethard S. UArizona at the CLEF eRisk 2017 Pilot Task: Linear and Recurrent Models for Early Depression Detection. *CEUR Workshop Proc* 2017 Sep; 1866:1-11

10. Mumtaz W, Malik AS, Yasin MAM, Xia L. Review on EEG and ERP predictive biomarkers for major depressive disorder. *Biomedical Signal Processing and Control* 2015 Sep; 22:85-98.

11. Islam MR, Kabir MA, Ahmed A, Kamal ARM, Wang H, Ulhaq A. Depression detection from social network data using machine learning techniques. *Health Inf Sci Syst* 2018 Dec;6(1):8.

12. Nafiz Al Asad. et al. Depression detection by analyzing social media posts of user // *IEEE International Conference on Signal Processing, Information, Communication & Systems* (SPICSCON). – 2019.

13. Abhilash Biradar & S. G. Detecting Depression in Social Media Posts Using Machine Learning. – 2019 – Т. 13. – №. 6. – С. 716-725.

14. Ding, Y., Chen, X., Fu, Q., & Zhong, S. A depression recognition method for college students using deep integrated support vector algorithm. – 2020. – *IEEE access*, *8*, 75616-75629.

15. Salas-Zárate, R., Alor-Hernández, G., Salas-Zárate, M. D. P., Paredes-Valverde, M. A., Bustos-López, M., & Sánchez-Cervantes, J. L. (2022, February). Detecting depression signs on social media: a systematic literature review. In *Healthcare* (Vol. 10, No. 2, p. 291). MDPI.

16. N. Axak, M. Kushnaryov, A. Tatarnykov, The Agent-Based Learning Platform. ICST-2023: *XI International Scientific and Practical Conference "Information Control Systems and Technologies"*, September 21-23, 2023, Odessa, Ukraine, Vol-3513, pp. 263-275.

17. Axak N., Korablyov M., Ushakov M. Cloud Architecture for Remote Medical Monitoring // IEEE Proceedings of the 15th International conference «Computer Sciences and Information Technologies», (CSIT-2020). – Zbarazh-Lviv, Ukraine, September 23-26, 2020. – vol. 1, pp. 344-347.

18. Axak N., Serdiuk N., Ushakov M., Korablyov M. Development of System for

Monitoring and Forecasting of Employee Health on the Enterprise. //COLINS. – 2020. – С. 979-992.

19. N. Axak, A. Tatarnykov, "The Behavior Model of the Computer User," *2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT)*, 2022, pp. 458-461, doi: 10.1109/CSIT56902.2022.10000499.

20. Axak N., Korablyov M., Rosinskiy D. MapReduce Hadoop Models for Distributed Neural Network Processing of Big Data Using Cloud Services / *Advances in Intelligent Systems and Computing* IV. / Editors: Shakhovska, Natalya; Medykovskyy, Mykola O., Springer, 2019. – pp. 387 – 400. ISSN: 2194-5357 //doi.org/10.1007/978-3-030-33695-0