Міністерство освіти й науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

Аналіз текстових повідомлень із використанням методів обробки природної мови та машинного навчання для виявлення симптомів депресії

Текстова частина до кваліфікаційної роботи
за спеціальністю 122 – «Комп’ютерні науки»

Керівник кваліфікаційної роботи
с.в. Борозенний С.О.

_____
(підпис)
«____» _____ 2025 р.
Виконала студентка КН-4
Дехтяренко М.С.
«____» _____ 2025 р.

Київ 2025

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на кваліфікаційну роботу

студентці 4 р.н. бакалаврської програми Комп’ютерні науки

Дехтяренко Мар’яні Сергіївні

Тема: «Аналіз текстових повідомлень із використанням методів обробки природної мови та машинного навчання для виявлення симптомів депресії»

Зміст текстової частини до кваліфікаційної роботи:

Анотація

Вступ

Розділ 1. Теоретичні засади дослідження депресії за допомогою машинного навчання та NLP

Розділ 2.Розроблення застосунку

Розділ 3.Отримані результати та їх аналіз

Висновки

Список використаних джерел

Дата видачі «____»_____ 2023 р.

Керівник _____ (підпис)

Завдання отримав _____ (підпис)

Календарний план виконання роботи:

Тема: Аналіз текстових повідомлень із використанням методів обробки природної мови та машинного навчання для виявлення симптомів депресії

№	Назва етапу кваліфікаційного проєкту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на кваліфікаційну роботу	04.09.2024	
2.	Огляд літератури за темою роботи	07.09.2024-02.10.2024	
3.	Проведення дослідження	05.10.2024-25.10.2024	
4.	Аналіз отриманих результатів	14.02.2025-20.02.2025	
5.	Розробка застосунку	22.03.2025-03.05.2025	
6.	Написання текстової частини курсової роботи	03.05.2025-15.05.2025	
7.	Захист кваліфікаційної роботи		

Студентка Дехтяренко М.С.

Керівник Борозенний С.О

“____” _____

АНОТАЦІЯ

Кваліфікаційну роботу присвячено розробці системи автоматизованого виявлення депресії за допомогою методів машинного навчання та обробки природної мови. У роботі описано процес збору та підготовки текстових даних, зокрема очищення, токенізації, лематизації та векторизації текстів різними підходами (Bag-of-Words, TF-IDF, Word2Vec, N-грами). Проведено порівняння трьох популярних алгоритмів класифікації (Naive Bayes, Support Vector Machine, Random Forest) для задачі бінарної та багатокласової класифікації психічних станів. Розроблено клієнт-серверний вебзастосунок, що дає змогу користувачам оперативно отримувати попередню оцінку ризику депресії за текстовими повідомленнями. Запропоновані методи можуть стати ефективним інструментом первинного скринінгу для підвищення доступності психологічної допомоги.

Ключові слова: машинне навчання, обробка природної мови, класифікація тексту, векторизація, депресія, психічне здоров’я, вебзастосунок.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....2
ВСТУП.....3
РОЗДІЛ 1.....7
Теоретичні засади дослідження депресії за допомогою машинного навчання та NLP.....7
1.1. Інструменти NLP і машинного навчання для автоматизованої діагностики ментального стану.....8
1.1.1 Підготовка текстових даних..... 8
1.1.2 Представлення тексту в числовому просторі.....10
1.1.3 Алгоритми машинного навчання для класифікації тексту..... 12
1.2 Висновки до розділу 1.....17
РОЗДІЛ 2.....18
Розроблення застосунку..... 18
2.1 Збір та підготовка даних.....18
2.1.1 Очищення даних.....19
2.1.2 Представлення тексту в числовому просторі.....20
2.2 Навчання та порівняння моделей.....21
2.3 Оцінка якості моделей.....25
2.4 Розроблення інтерфейсу.....27
2.5 Висновки до розділу 2.....34
РОЗДІЛ 3.....36
Отримані результати та їх аналіз.....36
3.1 Оцінка моделей машинного навчання..... 36
3.2 Аналіз матриці невідповідностей.....38
3.3 Висновки розділу 3.....42
ВИСНОВКИ..... 44
Список використаних джерел.....46

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CSV – Comma-Separated Values

GDPR – General Data Protection Regulation

ML – Machine Learning

NLP – Natural Language Processing

SVM – Support Vector Machine

TF-IDF – Term Frequency-Inverse Document Frequency

CSRF – Cross-Site Request Forgery

ВСТУП

Актуальність проблеми

У сучасному світі депресія є однією з провідних причин зниження якості життя та працездатності: поширеність депресивних розладів сягає 28 % у світі [1]. Водночас багато людей вагаються звернутися до кваліфікованих фахівців через страх стигматизації, невпевненість у власних симптомах або брак часу й ресурсів. Саме в таких випадках інструменти автоматизованого аналізу мовного контенту можуть стати першою лінією підтримки.

Зі зростанням популярності ChatGPT та аналогічних великих мовних моделей у суспільстві та науковому середовищі посилився інтерес до машинного навчання та до обробки людської мови. Це призвело до того, що автоматизований аналіз великих обсягів тексту став невід'ємною частиною людського життя. Поширення великих мовних моделей стимулювало зростання досліджень у напрямі автоматизованої оцінки психологічного стану за текстом. Завдяки цьому з'явилися численні експериментальні роботи, які демонструють можливість машинного навчання виявляти маркери депресії, тривожності та інших психічних розладів на основі лінгвістичних патернів. Такий прогрес дає змогу отримувати попередню оцінку емоційного стану користувача за кілька секунд, що може спонукати людей, які були невпевнені у своєму ментальному стані частіше звертатися по допомогу. Для фахівців подібні системи можуть виступати, як ефективний інструмент попереднього відбору пацієнтів, що дає змогу сконцентрувати ресурси на найбільш уразливих випадках та підвищити оперативність втручання.

Також пандемія COVID-19 та війна в Україні додатково погіршили моральний стан населення, що призвело до різкого зростання запитів на

психологічну підтримку. Водночас багато безплатних онлайн-тестів не проходять професійну валідацію і дають хибні результати, що лише поглиблює невпевненість користувачів або дає їм хибне уявлення про їхній ментальний стан.

У такий спосіб, застосування машинного навчання та методів NLP у галузі ментального здоров'я є надзвичайно актуальним: вони уможливлюють оперативно здійснювати первинний скринінг депресії, підвищують доступність психологічної підтримки. Це сприяє своєчасному виявленню та зверненню за допомогою й, у кінцевому підсумку, може зменшити соціальне й економічне навантаження, пов'язане з депресією.

Мета й завдання дослідження

Метою дослідження є впровадження системи автоматизованого аналізу текстового контенту для уможливлення виявлення симптомів депресії та інших психічних розладів за допомогою методів машинного навчання та NLP. Вагомою частиною дослідження є порівняння ефективності різних класифікаційних моделей, як-от: Naive Bayes, Support Vector Machine та Random Forest, для визначення найкращого алгоритму для цього завдання. У процесі дослідження зроблено акцент, не лише на порівнянні методів векторизації(Bag of words, N-grams, Word2vec, TF–IDF), а й на тому, як вибір моделі впливає на точність прогнозів і її здатність правильно виявляти симптоми депресії на основі векторизованих текстів.

Об'єктом дослідження є процес автоматизованого оброблення текстових даних для виявлення депресивних ознак.

Предметом дослідження є алгоритмічні та програмні засоби попереднього оброблення тексту, векторизації, побудови та порівняння класифікаційних моделей, а також їхнє впровадження в прикладну систему скринінгу депресії.

Для досягнення цієї мети виконано такі *завдання*:

- Проведення стандартизованого оброблення англомовних текстів, видалення пунктуації та зайвих символів, токенізацію і лематизацію – для зниження шуму та покращення якості вхідних даних.
- Реалізація чотирьох методів векторизації (Bag-of-Words, N-грами, TF-IDF, Word2Vec) та виконання порівняльного аналізу їхнього впливу на ефективність класифікаційних алгоритмів.
- Розроблення та порівняння роботи кількох класифікаторів (Naive Bayes, Support Vector Machine, Random Forest) у задачах бінарної (депресивний/нормальний) і багатокласової (стрес, депресія, біполярний розлад тощо) класифікації.
- Оцінювання продуктивності моделей за допомогою метрик accuracy, precision, recall, F1-score та матриці невідповідностей (confusion matrix) для виявлення оптимальних поєднань методів векторизації та алгоритмів машинного навчання.
- Створення клієнт-серверного вебзастосунка, який надасть користувачам можливість оперативно отримувати попередню оцінку свого ментального стану.

Методи дослідження

Методи наукового дослідження, що використано в роботі: емпіричний метод збору та обробки текстових даних із відкритих джерел; аналітичний метод порівняння підходів у векторизації та класифікації; метод системного підходу з аналізом, синтезом і декомпозицією завдання на етапи; методи машинного навчання та статистичні методи оцінки результатів (accuracy, precision, recall, F1-score, матриця невідповідностей); крос-валідація та оптимізація гіперпараметрів моделей;

Наукова новизна отриманих результатів

У процесі проведених досліджень, отримано такі наукові результати:

- Уперше реалізовано модель, що уможливлює визначення ментального стану людини за допомогою машинного навчання та NLP для українськомовних користувачів.
- Удосконалено метод, у якому вибрано найефективнішу комбінацію методів векторизації та класифікаційних алгоритмів для задачі виявлення депресивних симптомів.
- Удосконалено метод аналізування динаміки емоційного стану користувачів, і впроваджено збір часових рядів результатів скринінгу та їхню візуалізацію у вебінтерфейсі й надано змогу користувачам відстежувати зміни настрою.

Практичне значення отриманих результатів

Практичне значення одержаних результатів дослідження полягає в створенні інструменту для автоматизованого виявлення депресивних симптомів на основі текстового контенту. У розробленому вебзастосунку поєднано методи машинного навчання та обробки природної мови, й уможливлено здійснення первинного виявлення депресії. Ці результати можуть бути впроваджені в практику як інструмент для психологічної підтримки та раннього виявлення депресії, що може сприяти заохоченню більшої кількості людей своєчасно звертатися по допомогу до кваліфікованих фахівців.

РОЗДІЛ 1
Теоретичні засади дослідження депресії за допомогою машинного навчання та NLP

У сучасному суспільстві дедалі зростає увага до проблем ментального здоров’я, особливо в умовах обмеженого доступу до офлайн-послуг через географічні, фінансові або соціальні бар’єри. З огляду на це, загальнодоступні онлайн-ресурси набувають дедалі більшої популярності: все більше людей звертаються до інтернет-платформ за консультаціями з питань психотерапії, а також шукають безоплатні тести для первинної самодіагностики. Водночас бракує надійних і підтверджених інструментів, здатних своєчасно відсіяти випадки високого ризику.

У відповідь на зростаючий попит на турботу про ментальне здоров’я відзначається тенденція до широкого застосування платних і безплатних онлайн-платформ, що пропонують віддалений доступ до психотерапевтичних послуг, самодіагностики та підтримки. Проте значний відсоток користувачів вдається до безплатних онлайн-тестів, які часто не піддаються професійній валідації та можуть надавати хибно-позитивні або хибно-негативні результати, що загрожує відтермінуванням звернення до спеціалістів та поглибленням симптомів.

У цьому контексті машинне навчання постає як потужний інструмент аналізу великих обсягів текстової інформації та виявлення патернів, які свідчать про наявність депресивних симптомів. ML дає змогу створювати моделі, що навчаються на зразках текстів із підтвердженими випадками депресії, а згодом автоматично класифікувати нові тексти (повідомлення в соціальних мережах, щоденники, форуми тощо) за наявністю ризику депресивного стану. Такий підхід забезпечує швидку та

масштабовану діагностику, навіть серед тих груп населення, які не мають доступу до традиційної психотерапевтичної допомоги.

1.1. Інструменти NLP і машинного навчання для автоматизованої діагностики ментального стану

Обробка природної мови (NLP) – це галузь штучного інтелекту, що фокусується на взаємодії комп'ютерів із людською мовою. Сучасні методи NLP дають змогу автоматично аналізувати та інтерпретувати великі обсяги текстових даних, видобуваючи з них змістовну інформацію. Зокрема, NLP відіграє важливу роль у завданнях аналізу текстів для виявлення стану психічного здоров'я: вона забезпечує інструменти для автоматичного виявлення ознак депресії за текстом на основі мовних патернів і лінгвістичних характеристик.

Для того, щоб алгоритми могли ефективно аналізувати текст, його спочатку необхідно привести до стандартизованого вигляду.

1.1.1 Підготовка текстових даних

Попереднє оброблення спрямовано на приведення сирого тексту в стандартизований і очищений формат, придатний для подальшої векторизації. Тобто текст проходить декілька ключових етапів обробки, щоб залишити лише найсуттєвішу інформацію: токенізація, видалення стоп-слів, лематизація та стемінг

Токенізація – поділ суцільного тексту на окремі фрагменти – токени. Здебільшого токен відповідає окремому слову, числу або символу, іноді використовуються фрагменти слів чи цілі фрази. Токенізація перетворює рядок тексту на список одиниць, з якими зручно працювати на наступних етапах. Наприклад: фраза «I feel bad» може бути токенізована як послідовність токенів: [«I», «feel», «bad»]. Отримані токени є основою для

подальшого аналізу, оскільки саме на рівні окремих слів найчастіше будуються ознаки для моделей.

Наступним важливим етапом є видалення стоп-слів. Стоп-слова – це слова, що дуже часто зустрічаються в мові та зазвичай не несуть ключового змістового навантаження (зазвичай це службові частини мови: прийменники, сполучники, займенники, частки тощо). Приклади таких слів в англійській мові – «the», «and», «is», «in» та інші. Видалення стоп-слів зі списку токенів зменшує «шум»у даних, відкидаючи надто загальні слова, і тим самим дає змогу моделі зосередитися на більш значущих термінах тексту.

Лематизація – приведення слів до їхньої базової словникової форми. На цьому етапі алгоритм знаходить для кожного токена його нормальну форму: для дієслів – інфінітив, для іменників – називний відмінок однини тощо. Лематизація зменшує кількість унікальних слів, зводячи докупи різні словоформи одного слова. Наприклад, англійські слова «caring», «cares» і «cared» буде зведено до єдиної леми «care». Це спрощує подальше аналізування, оскільки знімає варіативність, пов’язану зі словозміною.

Стемінг – це алгоритмічний процес скорочення слова до його кореня шляхом видалення закінчень і суфіксів, що дає змогу зменшити варіативність словоформ і знизити розмір словника.

Зазвичай після перелічених кроків виконують додаткове очищення: приведення тексту до нижнього регістру, видалення розділових знаків, цифр, специфічних символів, нормалізацію абревіатур тощо – залежно від потреб конкретного завдання.

Після завершення базової попередньої обробки тексту ми отримуємо набір лексем у їхніх початкових (лематизованих) формах, готових до подальшої конвертації в числові ознаки. На наступному етапі очищені дані необхідно перевести в простір числових векторів, аби

зробити їх придатними для обчислювальних алгоритмів машинного навчання.

1.1.2 Представлення тексту в числовому просторі

Векторизація тексту полягає в представленні кожного слова чи токена у вигляді числового вектора, який відображає його зміст та статистичні властивості. Існує кілька основних підходів до цієї трансформації:

Bag-of-Words – кожен документ описується вектором довжини |V|, де V – словник усіх токенів. На позиції «i» стоїть кількість входжень і-го слова в документі. Отримана матриця розміром N×|V| (N – число документів) фіксує лише частоти слів, ігноруючи їх порядок. Так, фрази «I’m happy today» і «Today I’m happy» матимуть тотожні вектори. Попри простоту, BoW часто служить відправною точкою в задачах класифікації тексту, оскільки слова з високою частотою можуть корелювати з тематичними ознаками документа. Метод вирізняється простотою та швидкістю і слугує зручною стартовою точкою для класифікації, адже частотні слова добре корелюють із тематичними ознаками. Водночас він повністю ігнорує порядок слів і контекст, тож не здатен коректно обробляти фрази із запереченням на кшталт «not happy».

TF-IDF (Term Frequency–Inverse Document Frequency)— це вдосконалений метод BoW, який зважує слово за двома факторами: як часто воно зустрічається в одному тексті та наскільки рідко – у всій колекції, щоб виділити найбільш важливі терміни. Даний метод зважує кожне слово пропорційно його частоті в документі (TF) та обернено пропорційно частоті в усіх документах колекції (IDF). Таким чином, словам, які зустрічаються часто в аналізованому тексті, але рідко в інших текстах, буде призначена висока вага TF-IDF, тоді як загальні слова, які присутні часто, отримають низьку вагу. Такий механізм приглушує

стоп-слова та акцентує на ключовій лексиці, що робить TF-IDF популярним в інформаційному пошуку і класифікації. Водночас метод не враховує семантичну близькість синонімів, а за сильного дисбалансу класів може ігнорувати рідкісні, але критично важливі депресивні маркери.

Word2Vec — метод, який навчається на великій кількості текстів і перетворює кожне слово на «щільний вектор» (зазвичай розмірністю в сотні елементів), що відображає його значення в контексті. На відміну від BoW чи TF-IDF, які враховують лише частоти, Word2Vec розташовує семантично близькі слова поруч у векторному просторі. Завдяки цьому модель «розуміє», що слова «сумний» і «смуток» близькі за змістом, навіть якщо вони не мають спільного кореня. Така здатність враховувати контекст допомагає, точніше розпізнавати депресивну лексику, особливо коли людина використовує синоніми чи перефразування. Втім, цей підхід вимагає великого обсягу навчальних даних, а багатозначні слова можуть набувати різних сенсів (наприклад, «hurt» – і «біль», й «образити»).

N-грамні моделі — розширюють ідею методу Bag-of-Words, та беруть до уваги не тільки окремі токени, а і їхні послідовності. N-грама – це підрядок із N кількості слів. При формуванні вектору кожному документу окрім уніграм (поодиноких слів) додають біграми та/або триграми. Це дає змогу моделі «побачити» стійкі словосполучення і контекстні кліше, які втрачаються в класичній BoW: наприклад, біграма «не щасливий» вказує на негативну оцінку, тоді як окремі токени «не» й «щасливий» можуть зустрічатися в різних контекстах. Особливо корисно враховувати n-грами при розпізнаванні типових для депресивного мовлення виразів на кшталт «нічого не хочу» чи «втрачати надію». Щоб уникнути надмірної розрідженості, зазвичай обмежуються унірамами та біграмами з певною мінімальною частотою.

У результаті після стадії векторизації кожен текст представлено у вигляді числового вектора великої розмірності (сотні, тисячі й більше ознак залежно від методів).

Наступним кроком є застосування алгоритмів машинного навчання, які на цих ознаках будують модель для класифікації.

1.1.3 Алгоритми машинного навчання для класифікації тексту

Машинне навчання – це напрямок штучного інтелекту, який дає змогу комп’ютерам навчатися на основі даних і робити прогнози або приймати рішення без явного програмування кожного кроку. Замість того, щоби писати чіткі інструкції для кожної задачі, ми надаємо комп’ютеру багато прикладів, і він сам вчиться розпізнавати закономірності.

Машинне навчання поділяють на декілька видів:

Навчання з учителем (supervised learning) – це базовий підхід у текстовій класифікації, коли модель тренується на множині прикладів із відомими відповідями. Кожному тексту присвоюється мітка: наприклад, фразу «I’m depressed» позначають як 1 (депресивний текст), а «I’m very happy» – як 0 (нормальний текст). Під час навчання алгоритм аналізує ознаки кожного вхідного тексту (частоти слів, тональність, інші вектори) і коригує свої внутрішні параметри так, щоб мінімізувати помилки на цих розмічених прикладах. Після цього він здатен застосувати набуті знання до нових, раніше невідомих текстів і правильно передбачати, чи містять вони ознаки депресії.

Дослідження показують високу ефективність supervised learning у завданнях детекції депресивних повідомлень: зокрема, моделі Naive Bayes, Support Vector Machine та Random Forest демонструють точність класифікації на рівні 80–90 % при розподілі текстів на депресивні та недепресивні[4]. Перевага цього підходу полягає в тому, що модель «вчиться» саме на цільовій задачі, мінімізуючи помилки на розмічених

даних, і з достатньою кількістю якісних прикладів здатна давати стабільно високі результати.

Навчання без учителя (unsupervised learning) не спирається на готові мітки в даних, а самостійно шукає в них закономірності. У задачах аналізу депресії такий підхід часто використовують для кластеризації текстів. Наприклад, алгоритм k-means може згрупувати дописи пацієнтів у кілька кластерів за схожістю вживаної лексики: один кластер міститиме тексти з акцентом на смуток і самотність, інший – із гнівними чи агресивними висловами, третій – зі згадками втоми та апатії. Таке розподілення допомагає виявити приховані підгрупи серед людей із депресією і показує, що симптоми можуть проявлятися по-різному.

У машинному навчанні застосовується велика кількість алгоритмів для класифікації текстових даних. У цьому розділі детально розглянуто ті методи, які найчастіше використовуються в задачах автоматизованого аналізу тексту:

Наївний баєсівський класифікатор – це ймовірнісний метод, який використовує теорему Баєса для обчислення апостеріорної ймовірності належності тексту до того чи іншого класу:

$$P(C|X) = P(X|C)P(C)/P(X) \quad (1.1)$$

де C – розглянутий клас (наприклад, «депресивний» або «нейтральний»), а X=(w1, w2,…, wn) – множина ознак тексту (слова або токени). Щоб зробити розрахунки практично здійсненними в текстових задачах, вводиться припущення умовної незалежності: кожне слово вважається незалежним від інших, тобто: $P(C \mid X) = \prod_{i=1}^{n} P(wi|\, C) \quad (1.2)$

З навчальної вибірки рахуються апріорні ймовірності класів P(C)(частота документів даного класу) та умовні ймовірності появи кожного слова P(wi| C). Щоб уникнути проблем із нульовими

ймовірностями для слів, відсутніх у тренувальних даних, застосовують згладжування (найпоширеніше – Лапласове):

$$P(wi \mid C) = count(wi, C) + 1/\Sigma i\, count(wj, C) + V \quad (1.3)$$

де V – розмір словника.

Для чисельної стабільності всі множення замінюють сумою логарифмів:

$$log\ P(C \mid X) \propto log\ P(C) + \sum_{i=1}^{n} log\ P(wi \mid C) \quad (1.4)$$

Завдяки такому підходу навчання зводиться лише до підрахунку частот слів у класах і обчислення логарифмів, а класифікація – до швидкого підсумовування цих значень. Це робить модель вкрай швидкою та масштабованою навіть для корпусів із десятками тисяч документів і словником у десятки тисяч слів. Крім того, наївний Баєс ефективно працює на невеликих навчальних вибірках і з високорозмірними векторними представленнями, оскільки не потребує врахування кореляцій між ознаками. Проте його головна слабкість – ігнорування послідовностей і контексту: конструкції на кшталт «не добре» з рівними шансами трактуються як сумісне поєднання слів «не» й «добре», а не як вираз із негативною семантикою. Щоб частково подолати це, у векторизацію часто додають n-грамні ознаки. Цей алгоритм залишається одним із найпоширеніших стартових рішень у задачах текстової класифікації.

Ще одним популярним алгоритмом є Support Vector Machine(SVM) – це класичний алгоритм машинного навчання для бінарної класифікації, який шукає вектор-гіперплощину в багатовимірному просторі ознак, що найкраще розділяє дані двох класів. Під терміном «найкраще» розуміють максимізацію відстані від цієї гіперплощини до найближчих точок обох класів (опорних векторів). Завдяки такому підходу SVM зазвичай

демонструє високу здатність до узагальнення та зниженого рівня перенавчання.

У початковій лінійній формі алгоритм знаходить розв’язок оптимізаційної задачі

$$\min_{\mathbf{w},b} \ \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \quad \text{за умов} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0 \tag{1.5}$$

де w і b задають гіперплощину, C – параметр регуляризації, а – змінні відступів для некоректно класифікованих прикладів.

Головна перевага SVM полягає в застосуванні «kernel trick». За допомогою ядрових функцій (наприклад, радіальної базисної функції RBF або поліноміального ядра) можна неявно відобразити вхідні дані в простір вищої розмірності, де класи стають лінійно роздільними, без явного обчислення нових координат. Це дає змогу будувати як лінійні, так і нелінійні класифікатори з тією самою процедурою навчання.

Метод опорних векторів (SVM) вирізняється високою ефективністю на розріджених та високорозмірних даних, що особливо актуально для текстових представлень із великою кількістю термінів, при цьому він зберігає стійкість до перенавчання завдяки максимізації маржі та контролю регуляризаційним параметром C. Гнучкість вибору ядрової функції дає змогу застосовувати як лінійне ядро – яке часто дає оптимальні результати в задачах обробки тексту – так і складніші нелінійні ядра для моделювання тонших залежностей. Водночас SVM має вищі обчислювальні витрати та значне споживання пам’яті під час навчання на великих збірках даних, вимагає ретельного підбору гіперпараметрів (тип ядра, C, параметри ядра) із застосуванням крос-валідації та досвіду, а також не забезпечує наочної інтерпретації результатів, на відміну від інших моделей, де можна легко простежити ваги ознак або правила прийняття рішення.

Метод випадкових дерев (Random Forest) є потужним підходом до багатодеревної класифікації, у якому фінальне рішення формується шляхом усереднення результатів багатьох окремих дерев рішень. Кожне дерево навчається на випадковій вибірці прикладів із навчального набору і, побудувавши в кожному вузлі дерево на основі випадкового множення ознак, рекурсивно розбиває простір даних пороговими умовами. Завдяки цьому випадковості – як у виборі зразків, так і в підборі ознак для розбиття – окремі дерева виходять доволі різноманітними й менш схильними до перенавчання. Під час класифікації нового документа кожне дерево голосує за свій клас, а для остаточного прогнозу вибирається той клас, який набрав більшу кількість голосів.

Завдяки ансамблевій структурі, Random Forest зазвичай досягає вищої точності порівняно з одним деревом: поєднання багатьох слабких моделей формує стійку сукупність, у якій помилки окремих дерев компенсуються більшістю. Цей алгоритм добре масштабується на великі вибірки – навчання дерев можна робити паралельно й це не потребує детального попереднього опрацювання ознак: він успішно працює з пропущеними значеннями, необов’язковою нормалізацією і розрідженими даними, характерними для текстів. Крім того, Random Forest надає можливість оцінити важливість ознак, що допомагає виокремити слова чи групи слів із найбільшим внеском у класифікацію.

Водночас ансамбль із сотень дерев споживає більше пам’яті й часу на навчання та прогноз, ніж одиночне дерево або лінійні моделі, і його результати важче інтерпретувати: хоча можна отримати загальні оцінки важливості ознак, уявити чіткі правила ухвалення рішення складно. В умовах невеликих навчальних вибірок переваги ансамблю можуть не виявитися, а для задач із чіткими лінійними залежностями іноді знадобиться надто велика кількість дерев. Попри це, Random Forest залишається одним із найнадійніших базових алгоритмів класифікації

тексту, забезпечуючи високу точність і відмовостійкість у широкому спектрі прикладних завдань.

1.2 Висновки до розділу 1

У першому розділі розглянуто основні інструменти обробки природної мови (NLP) та методи машинного навчання, що використовуються для автоматизованої діагностики ментального стану на основі текстових даних. Розглянуто ключові етапи попередньої обробки тексту, включно з токенізацією, видаленням стоп-слів, лематизацією та стемінгом, що забезпечують стандартизацію та очищення даних для подальшого аналізу. Представлення тексту у вигляді числових векторів за допомогою методів Bag-of-Words, TF-IDF, Word2Vec та n-грамних моделей дозволяє врахувати частотні та контекстні характеристики слів, що суттєво підвищує ефективність аналізу ментального стану.

Детально розглянуто алгоритми машинного навчання, які найчастіше застосовуються для класифікації тексту у задачах виявлення депресії: наївний баєсівський класифікатор, Support Vector Machine та метод випадкових дерев (Random Forest). Кожен з них має свої переваги та обмеження, які впливають на точність, швидкодію та інтерпретованість моделей. Зокрема, навчання з учителем показує високу ефективність завдяки навчанню на розмічених даних, що дозволяє моделі адаптуватись до специфіки цільового завдання. Водночас використання різних методів векторизації та класифікації дає змогу враховувати як частотні, так і семантичні ознаки тексту, що підвищує якість діагностики ментального стану за текстовими даними.

Таким чином, поєднання методів NLP із сучасними алгоритмами машинного навчання створює потужну базу для розробки автоматизованих систем, здатних ефективно і точно аналізувати текстові висловлювання з метою виявлення ознак депресії та інших психічних розладів.

РОЗДІЛ 2

Розроблення застосунку

Цей розділ присвячено розробці застосунку для надання швидкого та об’єктивного скринінгу текстового контенту без необхідності формулювання складних запитів чи проходження численних неінформативних тестів на виявлення симптомів депресії та інших розладів.

У цьому розділі розкривається практична реалізація системи автоматизованого скринінгу депресії на основі методів машинного навчання та NLP. У застосунку використано кілька класичних алгоритмів машинного навчання (Random Forest, SVM, Naive Bayes) та різні методи векторизації тексту, що дає змогу напряму продемонструвати вплив вибору векторизації на точність результатів різних алгоритмів.

2.1 Збір та підготовка даних

Перед створенням моделі, здатної виявляти депресію за текстом, критично важливо було знайти відповідні дані на яких модель буде навчатися. Зі зрозумілих причин, як-от: приватність та просто небажання багатьох людей ділитися власними психологічними проблемами, доступні збірки даних часто мають обмеження за обсягом, недостатню різноманітність міток або охоплюють конкретну та невеличку галузь. Тому для тренування і тестування застосунку було обрано дві безплатні збірки даних «Reddit Mental Health Data» [2] та «Student-Depression-Text» [3] з платформи Kaggle. Перша збірка містить деталізацію типу стану:

- 0 = Stress
- 1 = Depression
- 2 = Bipolar disorder
- 3 = Personality disorder

- 4 = Anxiety

тоді як друга передбачає лише бінарну класифікацію «депресія / норма».

Використання обох збірок даних було зумовлене спостереженням, що моделі, натреновані виключно на «Reddit Mental Health Data», схильні видавати діагнози депресії навіть для текстів із чітко позитивним емоційним забарвленням. Додавання «Student-Depression-Text» допомогло відновити баланс між чутливістю та специфічністю алгоритму й підвищити узагальненність моделі на різнорідних прикладах.

У якості вхідних файлів використовується CSV файл із ключовими стовпцями: «text»(оригінальні повідомлення користувачів), «title» (заголовок допису) та «target» (числова мітка класу) та xlsx файл із чотирма основними стовпцями «text», «age», «gender», «age category»

Така уніфікована структура даних дає змогу не лише виконувати бінарну класифікацію «депресія / не депресія», але й – у разі позитивного результату на виявлення депресії – ідентифікувати конкретний тип психічного розладу. Такий формат стає єдиним вхідним набором для наступних етапів NLP-попередньої обробки та побудови моделей машинного навчання, забезпечуючи більш точний і детальний скринінг.

2.1.1 Очищення даних

Після завантаження файлу, дані в ньому потребують стандартизованого підходу очищення, щоб максимально зменшити шум та підготувати тексти до аналізу, тому весь текст має пройти декілька основних етапів обробки, які були детально розписані в першому розділі:

1. Нормалізація регістру і видалення пунктуації
2. Видалення зайвих елементів
3. Токенізація
4. Фільтрація стоп-слів

5. Зведення до базових форм

У результаті кожен документ перетворено в послідовність чистих, нормалізованих лексем, готових до представлення у вигляді числового вектора. Після очищення даних ми можемо чітко побачити, як багато непотрібних слів було видалено з вхідного файлу.

Приклад вхідних, ще не очищених даних:«We have to work the majority of our time in jobs we hate for people we don't care about to earn just about enough to live relatively comfortably.»

Уже очищенні дані: «work majority time job hate people dont care earn enough live relatively comfortably»

Таке очищення знизило кількість токенів із приблизно 40 до 12, що свідчить про видалення понад 65 % шумових елементів. Це дає змогу фокусуватися на лексемах, які найбільше характеризують семантичний зміст тексту.

2.1.2 Представлення тексту в числовому просторі

Після попереднього очищення та стандартизації текстів було виконано перетворення у числові вектори — ознаки, які придатні для подальшої обробки моделями машинного навчання. Оскільки алгоритми працюють виключно з числовими даними, «сирі» текстові рядки було подано у форматі, зручному для лінійних і статистичних операцій: обчислення відстаней, множення на вагові вектори, оптимізації функцій втрат тощо.

У практичній частині реалізовано кілька методів векторизації текстів для порівняння їхньої ефективності та визначення найбільш придатних для конкретних алгоритмів машинного навчання.

Перший метод — Bag-of-Words (BoW), у якому було сформовано словник усіх унікальних слів із корпусу текстів. Кожному слову присвоюється індекс, а тексти були представлені у вигляді розрідженої

матриці, де рядки відповідають текстам, а стовпці — словам. Присутність слова в тексті позначається одиницею, що дозволяє зафіксувати, які слова зустрічаються, ігноруючи їхній порядок.

Другий метод — n-grams, що розширює BoW, включає не лише окремі слова, а й двослівні послідовності (біграми). Це дає змогу врахувати стійкі словосполучення і покращити контекстуальне розуміння тексту.

Третій метод — TF-IDF, який оцінює важливість кожного слова з урахуванням його частоти у конкретному тексті та в усьому корпусі. Для цього застосовано готовий інструмент TfidfVectorizer з бібліотеки scikit-learn, що автоматично виконує трансформації та ігнорує поширені англомовні стоп-слова.

Останній метод — Word2Vec, який полягає у створенні векторних представлень для кожного слова словника. Кожному слову випадково присвоєно вектор сталої розмірності, а представлення тексту отримано шляхом усереднення векторних представлень усіх слів у ньому, що дає компактний числовий опис з урахуванням семантичної близькості слів.

Таким чином, отримано кілька типів векторних подань текстів: частотне (BoW), розширене n-grams, вагове TF-IDF та Word2Vec, які буде використано для навчання моделей машинного навчання.

2.2 Навчання та порівняння моделей

У роботі для визначення оптимальних підходів до автоматичної класифікації текстів було застосовано декілька класичних алгоритмів машинного навчання – Support Vector Machine, Naive Bayes та Random Forest – а також їхні реалізації з бібліотеки scikit-learn для порівняння ефективності та точності.

Кожен із цих методів має власні особливості:

1. Support Vector Machine

У ручній реалізації вагові коефіцієнти та зсув спочатку було встановлено рівними нулю. Навчання відбувалося методом стохастичного градієнтного спуску з додаванням L_2- регуляризації за квадратною нормою, яка обмежує збільшення значень параметрів. На початку кожного повного проходу всією вибіркою, дані перемішувалися та розбивалися на невеликі групи спостережень. Для кожної групи обчислювалася відстань кожного прикладу до граничної лінії рішення, якщо вона виявлялася меншою за певний поріг, цей приклад враховувався під час корекції ваг. Під час оновлення параметрів відбувається одночасне виконання таких процесів: зменшення значення тих коефіцієнтів, які занадто швидко зростають та внесення корекцій, орієнтованих на ті приклади, що неправильно класифіковані або розташовані близько до межі розділення.

Розмір кроку оновлення (швидкість навчання) поступово знижувалася з кожним циклом відповідно до формули $\eta = 1/(\lambda \cdot t)$, що дозволило спочатку робити великі корекції, а згодом – дедалі дрібніші, аби остаточне рішення було більш точним. Щоб зменшити випадковий вплив порядку обробки спостережень, під час навчання накопичуються проміжні значення коефіцієнтів, а після завершення всіх ітерацій остаточні параметри беруться як середнє від цих накопичених значень. Це згладжує різкі коливання та робить модель більш стійкою до нових даних.

У Scikit-learn SVM розв’язується у двоїстій формі за допомогою методу послідовної мінімальної оптимізації (SMO). Для роботи з даними, які не можна лінійно розділити у вихідному просторі, застосовується спеціальний спосіб обчислення скалярного добутку через функцію ядра, без явного збільшення розмірності – так званий метод застосування ядрової функції.

Параметр C є гіперпараметром моделі та відповідає за баланс між максимальною відстанню між класами та кількістю допущених помилок класифікації: менше значення C дає змогу більшому числу точок

потрапляти в «зону невизначеності», а більше значення C прагне мінімізувати помилки, жорстко штрафуючи неправильні класифікації, що може призвести до перенавчання. Параметр γ (для радіального базисного ядра) визначає радіус впливу кожного опорного вектора на форму кордону рішення – більші значення γ роблять кордон більш локалізованим, менші – більш гладким і глобальним.

Для задач із кількома класами використовується підхід «кожен проти всіх», коли окрему модель навчають протиставляти один клас усім іншим. Вбудовані умови зупинки (точність та максимальна кількість ітерацій) дають змогу контролювати збіжність алгоритму та час його виконання.

2. Naive Bayes

Імплементація вручну починається з підрахунку частоти появи кожного терму в документах різних класів, з урахуванням передумови Лапласового згладжування: до кожного лічильника додається сталий коефіцієнт α, щоб уникнути нульових ймовірностей. На основі цих частот обчислюються початкові ймовірності класів, як відношення кількості документів кожного класу до загальної кількості зразків у тренувальній вибірці, а також умовні ймовірності появи кожного терму в межах конкретного класу, які зберігаються в логарифмічному масштабі для уникнення роботи з надто малими числами. Під час класифікації нового документа для кожного класу обчислюється сума логарифмів початкової ймовірності цього класу та логарифмів умовних ймовірностей термінів із вхідного тексту. Клас із найбільшим значенням цієї суми обирається як передбачуваний.

Завдяки простоті обчислень і незалежному моделюванню ознак ця реалізація демонструє високу швидкість навчання та передбачення і добре справляється з розрідженими векторними поданнями тексту.

У бібліотечній реалізації класи MultinomialNB і ComplementNB написані на C і глибоко інтегровані з оптимізованими структурами даних SciPy/NumPy. Вони використовують векторизовані операції для підрахунків, підтримують різноманітні варіанти згладжування, а в ComplementNB додатково враховують баланс класів у особливо нерівномірних вибірках. Це гарантує високу швидкість роботи та стабільність у виробничому середовищі.

3. Random Forest

Ручна імплементація – модель складається з кількох окремих дерев рішень, кожне з яких навчається на випадковій вибірці з повтореннями з початкового набору даних. На кожному вузлі дерева для пошуку найкращого поділу було вибрано невелику випадкову підмножину ознак (зазвичай рівна квадратному кореню з повної кількості). Якість кожного поділу оцінено за одним із двох показників: індексом неоднорідності або мірою невизначеності розподілу (інформаційна ентропія). Щоб зменшити час обчислень у критичних ділянках – зокрема при підрахунку виграшу інформації від потенційного розщеплення – було використано бібліотеку Numba, яка переводить Python-функції в машинний код. Одночасне створення кількох дерев було організоване через joblib, що дає змогу залучити всі доступні процесорні ядра. Фінальний прогноз формується шляхом голосування або усереднення ймовірностей, що повертають окремі дерева.

Бібліотечна реалізація класу RandomForestClassifier написана на Cython і оптимізована для сучасних процесорів шляхом підтримки

одночасної обробки кількох значень одним набором команд (SIMD-інструкцій) та багатопотоковості. Модель автоматично регулює глибину дерев і мінімальну кількість зразків у листі, оцінює важливість кожної ознаки на основі середнього зменшення невизначеності при розщепленнях і ефективно управляє пам’яттю. За замовчуванням налаштовані гіперпараметри дають високу продуктивність без необхідності додаткової оптимізації.

Щоб з’ясувати, який з алгоритмів є найбільш придатним для поставленої задачі виявлення депресивних симптомів у текстах, а також який спосіб векторизації (BoW, n-грами, TF-IDF чи Word2Vec) найкраще відображає характеристику тексту, були проведені систематичні експерименти. Для кожного класифікатора здійснювалося поетапне: розбиття на тренувальну та тестову вибірки, навчання моделі на тренувальних даних із кожним із видів векторизації, оцінка її якості за метрикою точності (accuracy) і допоміжними показниками precision, recall і F1-score на тестовій вибірці.

Найбільш інформативний спосіб перетворення тексту був обраний за найвищим значенням accuracy, після чого обрана конфігурація алгоритму з відповідною векторизацією проходила фінальне порівняння з аналогічними налаштуваннями інших методів. Такий підхід дав змогу не лише виявити внутрішні переваги кожного алгоритму щодо специфіки обраного корпусу текстів, але й з’ясувати найкращий підхід із погляду балансу якості класифікації та швидкості навчання. Після завершення експериментів найкращі моделі було серіалізовано і збережено для подальшого швидкого завантаження в робочому середовищі, що дає змогу уникнути необхідності повторного тривалого навчання та гарантує відтворюваність одержаних результатів.

2.3 Оцінка якості моделей

Після побудови та навчання класифікаторів необхідно об’єктивно виміряти, наскільки точно вони розпізнають депресивні тексти. Для об’єктивного порівняння класифікаторів, у роботі використано стандартні обчислювальні процедури бібліотеки Scikit-learn. Після отримання прогнозів модель оцінюється за кількома взаємодоповнювальними показниками, кожен із яких фіксує свій аспект якості.

Насамперед обчислено accuracy – відношення кількості правильно класифікованих текстів до загальної кількості спостережень. Ця загальна метрика дає уявлення про середню точність моделі, проте її інформативність значно знижується при нерівномірному розподілі класів. У такому випадку модель може демонструвати високу точність, просто віддаючи перевагу більш представленому класу.

Щоб оцінити, наскільки вірогідними є саме ті випадки, які модель позначила як депресивні, було використано precision – частку істинно-позитивних прогнозів серед усіх, які система класифікувала як «депресія». Високе значення «precision» означає, що небажані помилки, коли текст, який не є депресивним помилково позначається як такий, трапляються рідко. Це критично в даному застосунку, де хибні результати можуть спричинити зайве занепокоєння або непотрібні обстеження.

Дану оцінку було також доповнено метрикою recall (чутливість), яка вимірює, яку частину всіх реальних депресивних текстів модель змогла виявити. Низькі значення recall свідчать про те, що значна кількість депресивних повідомлень залишається непоміченою, що в контексті раннього виявлення симптомів неприпустимо.

Оскільки precision і recall оцінюють протилежні ризики, оптимальним рішенням було додати F1-score – що є гармонійним поєднанням цих двох показників. У роботі використано зважену версію F1, коли внесок кожного класу пропорційний його частці у вибірці; такий

підхід коректно віддзеркалює дисбаланс даних і водночас не дає змогу ігнорувати нечисленні, але критично важливі категорії.

Для детального розбору помилок будується матриця невідповідностей (confusion matrix), у якій розкладено результати на чотири базові типи: істинно-позитивні, хибнопозитивні, істинно-негативні, хибнонегативні.

У результаті такого аналізу визначено, які саме класи найчастіше плутає модель, та надано змогу спланувати подальші вдосконалення: наприклад, підсилити навчання на певних підмножинах текстів або змінити стратегії балансування вибірки.

Вибір цих метрик зумовлено специфікою задачі виявлення депресії в тексті. Так комбіноване застосування accuracy для загальної оцінки, precision для зниження числа помилкових спрацьовувань, recall для повного захоплення випадків депресії, їхнього узагальненого поєднання в F1-score та аналізу за допомогою матриці невідповідностей забезпечує всебічну й об’єктивну оцінку класифікаторів у задачі виявлення депресивних текстів.

2.4 Розроблення інтерфейсу

Для забезпечення взаємодії із системою виявлення депресії було розроблено вебінтерфейс на базі фреймворку Flask. Його вибір зумовлено браком жорстких обмежень у структурі та широкою екосистемою розширень, що дало змогу швидко реалізувати модулі аутентифікації, управління базою даних (SQLAlchemy + Flask-Migrate).

Користувачі реєструються та входять через прості форми, у яких реалізована багатоетапна валідація: поля перевіряються на порожні значення, коректність формату (email), а також захищені від CSRF-атак через вбудовані механізми Flask-WTF. Паролі зберігаються в базі у вигляді

хешів із випадковим унікальним рядком, який виключає витік чутливих даних у разі компрометації сервера.

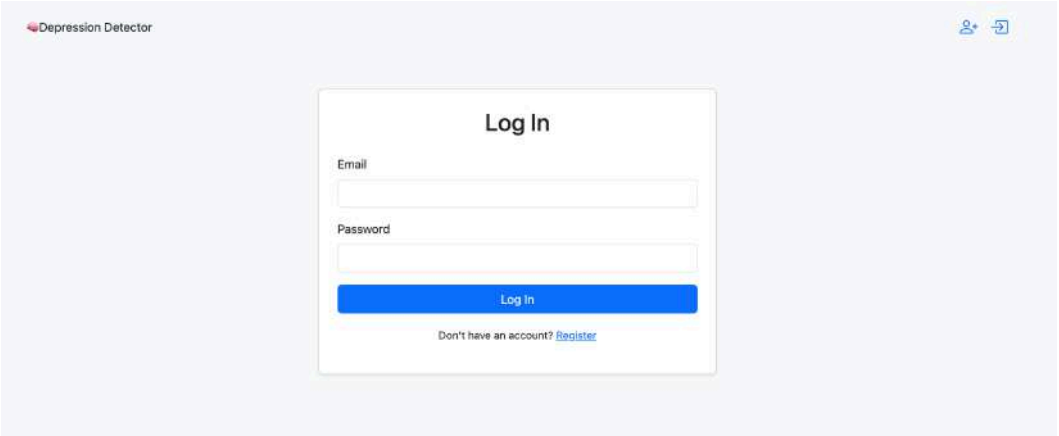

Рис 2.1. Вхід у застосунок

Після успішної аутентифікації користувач потрапляє на головну сторінку, де бачить назву сервісу, короткий опис можливостей і кнопку переходу до аналізу тексту. Звідси можна ознайомитися з інструкцією та перейти безпосередньо до завантаження повідомлень або файлів.

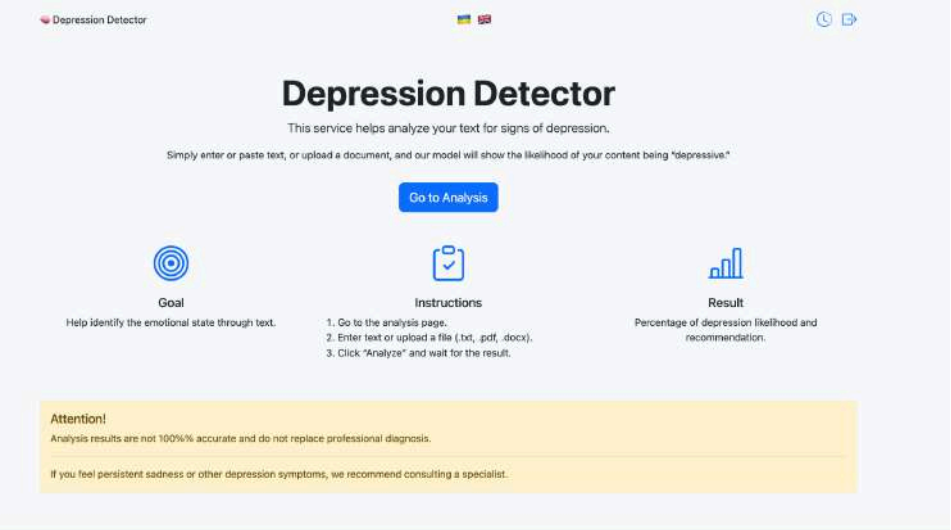

Рис 2.2. Головна сторінка

Інтерфейс підтримує як ручний ввід тексту українською чи англійською мовами, так і завантаження документів у форматах TXT, PDF та DOCX через стандартні інструменти Flask/Werkzeug. Крім того, є можливість не вводити текст вручну, а записати його голосом безпосередньо в поле вводу. Голосове введення підтримує розпізнавання мов, відповідно до обраної локалі (українська або англійська).

У зв’язку з тим, що певні групи населення, як от підлітки та жінки, статистично більш вразливі до депресії, на сторінці аналізу передбачено вказувати стать і вік користувачів. У рамках політики безпеки й відповідності GDPR усі персональні метадані (вік, стать, email) зберігаються за зашифрованими з’єднаннями й доступні лише власнику облікового запису.

Отриманий текст та супровідні метадані проходять попередню обробку (токенізація, очищення, лематизація) та векторизацію власними словниками й векторними поданнями.

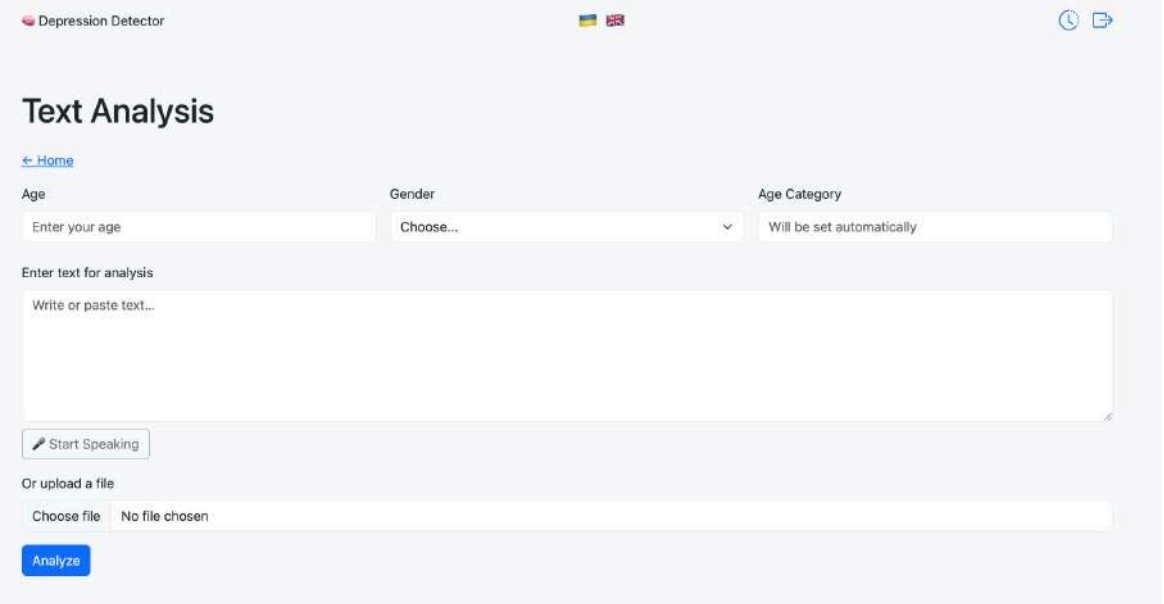

Рис 2.3. Сторінка аналізу

Модель роботи прогнозування складається з двох кроків: бінарна класифікація «депресія / не депресія» із використанням усіх ознак, зокрема трьох метаданих (вік, стать, категорія віку), які додаються до текстових ознак. Якщо результат перевищує поріг 0.5, запускається другий крок – мультикласова модель, що по тексту визначає конкретний тип розладу (стрес, депресія, біполярний чи інший стан). Обидві моделі та відповідні векторизатори завантажуються на старті через `joblib.load`, що дає змогу миттєво обробляти запити без повторного навчання.

У фоновому режимі всі ключові дії (авторизація, завантаження файлу, обробка тексту, запит до моделі) записуються в систему логування – як у файли, так і в зовнішній моніторинговий сервіс. Це дає змогу відстежити найменші збої, аналізувати продуктивність та оперативно сповіщати розробників про непередбачені помилки.

Результати аналізу відображаються у вигляді остаточного діагнозу (Depressed / Not Depressed) із відсотковою ймовірністю.

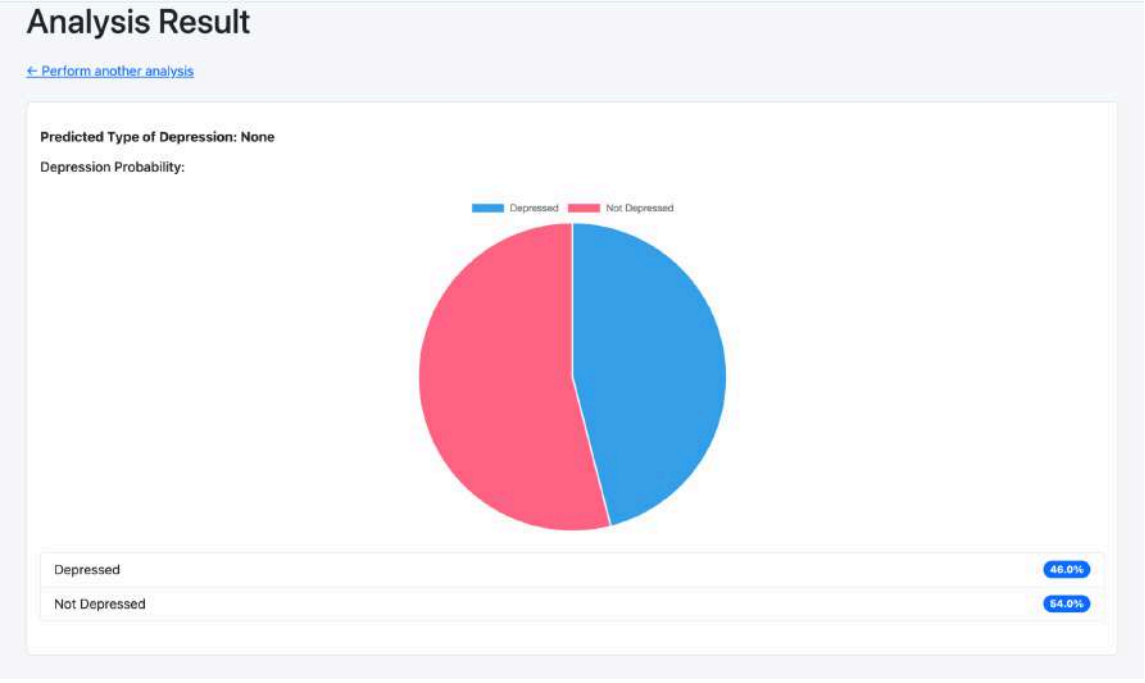

Рис 2.4. Сторінка з підсумками аналізу ментального стану користувача

У випадку, якщо користувач отримує значення depresssed понад 50 % тоді він отримає деталізований розподіл по можливих типах його розладів.

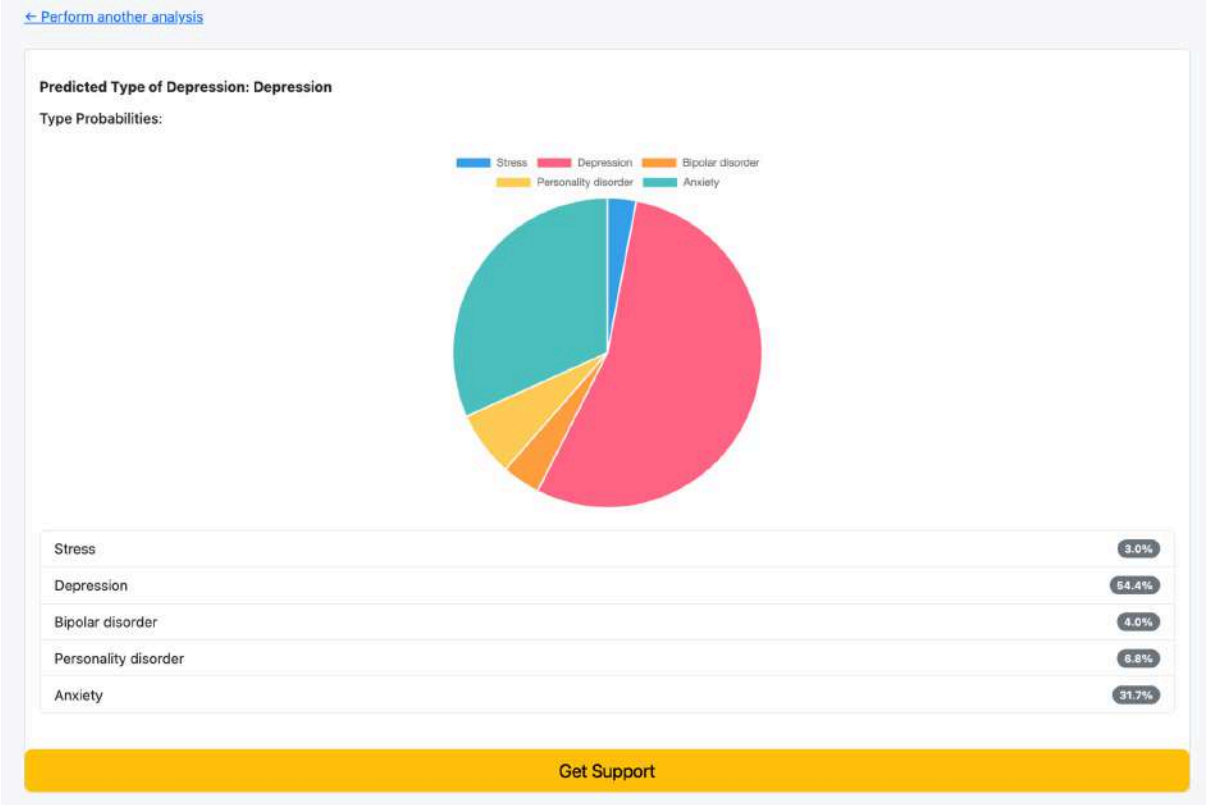

Рис 2.5. Сторінка з аналізом можливих розладів користувача

При натисканні кнопки Get Support, на основі отриманих результатів аналізу, було визначено який саме розлад користувача є найбільш вираженим. Відповідно до цього найвищого показника користувач автоматично перенаправляється на спеціальну сторінку підтримки, присвячену саме цьому розладу.

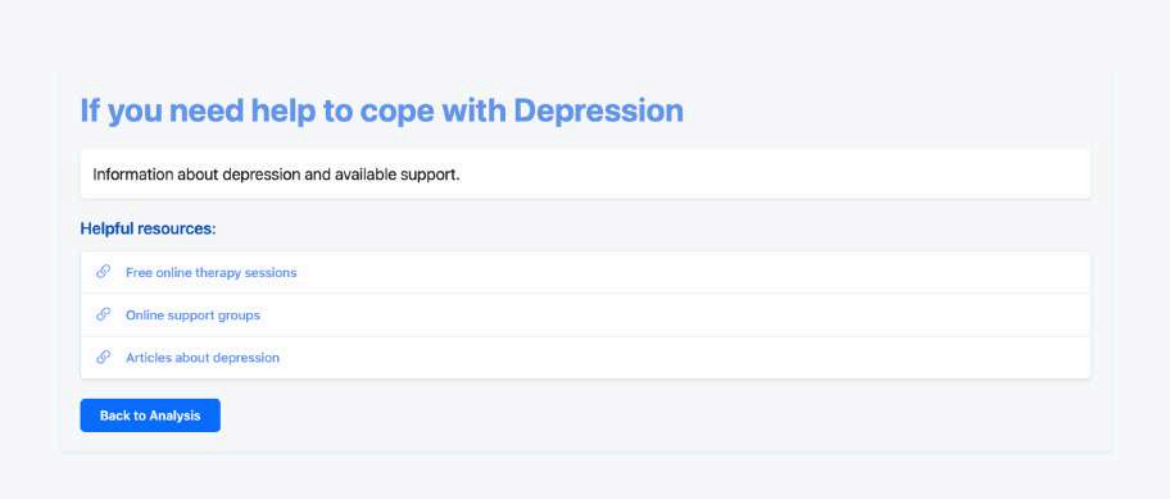

Рис 2.6. Сторінка підтримки

На сторінці підтримки користувачу надається підбірка корисних ресурсів: посилання на безплатні онлайн-терапії, групи підтримки, тематичні статті та інші джерела, які можуть допомогти впоратися з конкретним розладом. Таким чином, користувач отримує персоналізовану підтримку, яка адаптована під його актуальний емоційний чи психологічний стан.

Крім того, користувач може перейти на сторінку «Історія аналізів», де в табличному вигляді зберігаються час запиту, висновок і розподіл ймовірностей.

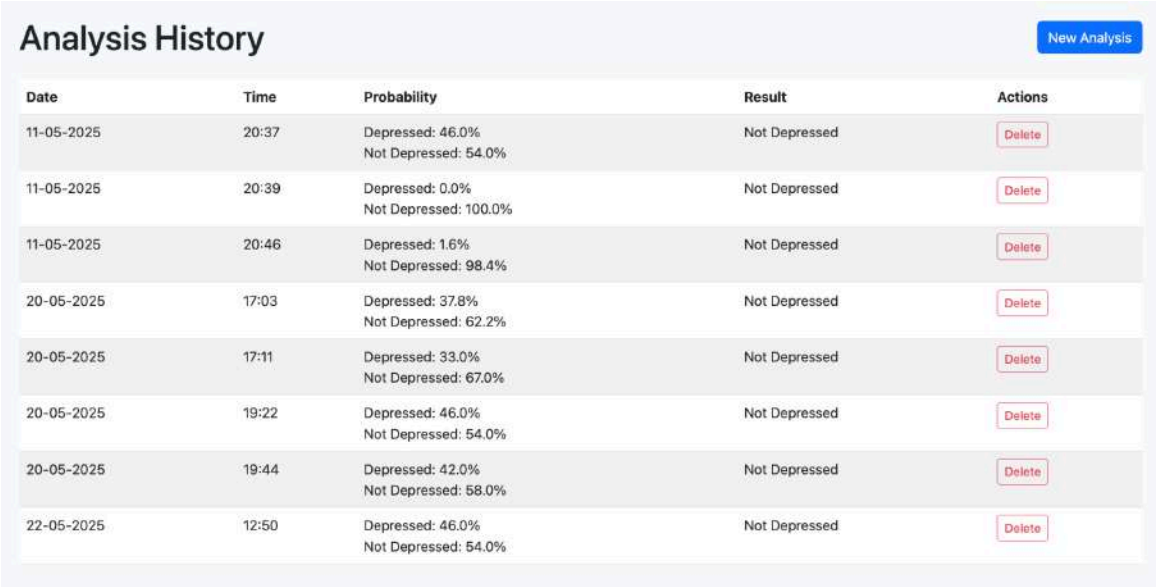

Рис 2.7. Сторінка історії аналізів користувача

Нижче таблиці формується графік, на якому відображено тенденцію ймовірності депресії на основі запитів користувача в часовому розрізі. По осі X позначено часові мітки, які відповідають моментам надходження запитів, а по осі Y – відсоткове значення ймовірності депресивного стану від 0 % до 100 %. Лінійна крива ілюструє зміну цієї ймовірності із часом, що дає змогу користувачу зручно відстежувати динаміку власного ментального стану та оцінювати коливання між депресивними та нормальними запитами.

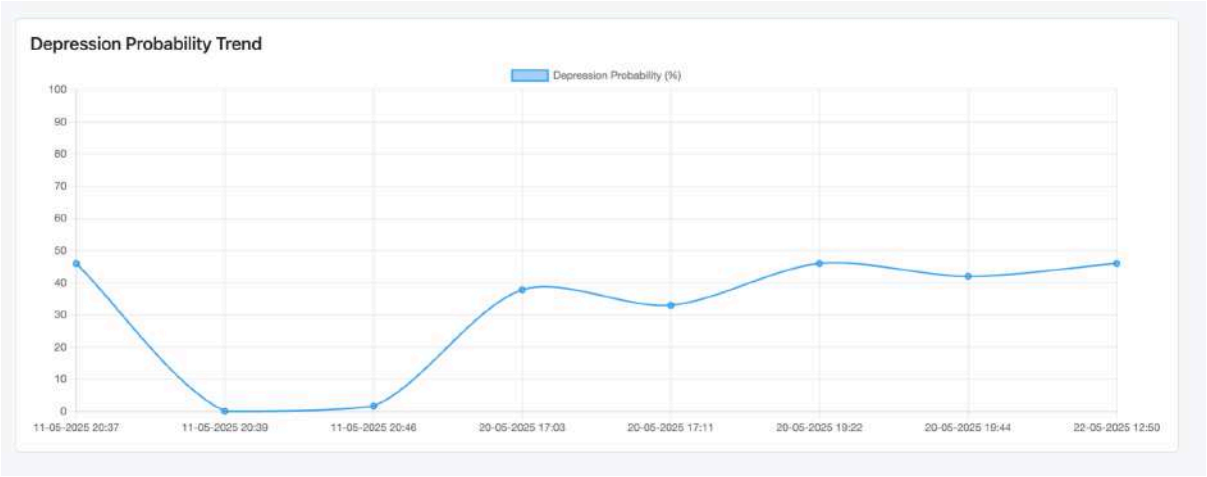

Рис 2.8. Графік для відстеження динаміки депресії за запитами користувача

2.5 Висновки до розділу 2

У другому розділі було повністю реалізовано прикладну систему скринінгу депресії – від форматування тексту зі збірок даних до створення інтерактивного вебсервісу. Поєднання двох відкритих наборів даних – Reddit Mental Health та Student-Depression-Text – дозволило збалансувати пропорцію «депресивних/нормальних» прикладів і створити вибірки як для бінарного, так і для мультикласового навчання. Стандартизований конвеєр обробки (нормалізація, токенізація, видалення стоп-слів, лематизація) прибрав понад 65 % шуму, зменшивши розмірність векторів без втрати інформативності. Для подання тексту протестовано кілька підходів: Bag-of-Words, N-грами, TF-IDF і Word2Vec. Усі конфігурації оцінювалися за accuracy, precision, recall, F1 й матрицею невідповідностей зі Scikit-learn, що дало змогу виявити сильні та слабкі сторони моделей на нерівномірних вибірках. Додавання метаданих (вік, стать, вікова категорія) дало змогу отримати більш точні результати щодо депресії.

Для швидкого передбачення найкращі моделі та векторизатори серіалізовано через joblib, що забезпечує час відповіді < 100 мс без повторного навчання. Двоступенева схема обробки даних(спершу бінарна перевірка, за потреби – мультикласова) мінімізує обчислення і зберігає

високу точність. Усі ключові дії (авторизація, завантаження файлів, виклики моделей) фіксуються і відстежуються зовнішнім моніторинговим сервісом. Персональні дані зберігаються зашифрованими й обробляються згідно з GDPR. Вебінтерфейс написаний на Flask складається зі сторінок Home, Analysis і History, Result, Login, Registration, Support підтримує ручний ввід і файли TXT/PDF/DOCX, захищений від CSRF і надає користувачеві графік динаміки ризику депресії.

Отже, розроблена система об’єднує перевірені алгоритми машинного навчання, ефективні методи векторизації та безпечний вебінтерфейс, забезпечуючи швидкий і достовірний скринінг депресивних симптомів у текстах. Впровадження персональних метаданих додатково підвищує практичну цінність автоматизованого інструменту підтримки психічного здоров’я.

РОЗДІЛ 3.
Отримані результати та їх аналіз

3.1 Оцінка моделей машинного навчання

У межах дослідження було проведено експерименти з трьома класичними моделями машинного навчання – Naive Bayes, SVM та Random Forest. Ці моделі були реалізовані двома способами: вручну, а також за допомогою бібліотеки Scikit-learn. Усі експерименти з моделями проводились на однаковій збірці даних «Reddit Mental Health Data», що містить марковані текстові повідомлення користувачів із різними психічними станами(Stress, Depression, Bipolar disorder, Personality disorder, Anxiety).

Отримані результати були ідентичними як для реалізацій написаних власноруч та для реалізацій створених через бібліотеку Scikit-learn. Це можна пояснити тим, що реалізації написані власноруч були розроблені з використанням тих самих математичних принципів, формул і алгоритмів, що лежать в основі моделей бібліотеки Scikit-learn. Наприклад, власноруч написаний клас NaiveBayesScratch використовує методи згладжування ймовірностей (Laplace smoothing), розрахунок логарифмічних ймовірностей ознак та апріорних логарифмічних ймовірностей класів, аналогічно тому, як це реалізовано в класі MultinomialNB бібліотеки Scikit-learn. Аналогічним чином, алгоритми Decision Tree та Random Forest, реалізовані вручну, також базуються на тих же математичних критеріях, як-от: критерій Джині (Gini impurity) або інформаційний приріст (Information Gain), розбиття на вузли за найкращими порогами ознак, bootstrap-вибірки та випадковий вибір підмножин ознак для розділення дерев. Реалізація з нуля SVM заснована на стохастичному градієнтному спуску, який є одним із поширених підходів, що

використовуються в бібліотечній реалізації SVM. Через це, результати моделей є передбачувано подібними до Scikit-learn.

Підтверджуються висновки викладені в статті «Benchmarking of Common Machine Learning Algorithms in Python»[5], де показано, що при відтворенні однакових гіперпараметрів результати за метриками F1 та accuracy між ручною та бібліотечними реалізаціями відрізняються не більше ніж на 0–1 %. Водночас оптимізовані бібліотеки (які використовують технології на кшталт Cython та OpenMP) забезпечують суттєво кращу швидкодію, що робить їх більш ефективними для практичного застосування.

Щодо оцінки ефективності моделей, Naive Bayes продемонстрував стабільно високі результати (F1-міра: 71 % для Bag-of-Words та 75 % для N-грамних моделей), завдяки припущенню умовної незалежності ознак, що добре відповідає розрідженим векторним представленням тексту. Використання щільних (Word2Vec) чи специфічно зважених (TF-IDF) представлень призвело до зниження точності (Word2Vec – 30 %, TF-IDF – 68 %), оскільки ці методи створюють корельовані ознаки або додають шум, порушуючи базове припущення Naive Bayes.

Модель Support Vector Machine (SVM) показала дещо гірші результати з розрідженими поданнями Bag-of-Words (точність ≈66,7 %) і N-грамами (точність ≈68,9 %), які дають змогу краще відокремлювати класи лінійною гіперплощиною. Використання TF-IDF знизило точність до ≈59,1 %, а Word2Vec – до ≈36,2 %, що пояснюється тим, що щільні семантичні вектори складніше поділяються лінійними методами за умов значного лексичного перекриття між класами.

Random Forest продемонстрував найкращі результати з використанням TF-IDF (точність = 77,85 %, precision = 79,54 %, F1 = 78,19 %), оскільки цей метод дає змогу ефективно виявляти важливі слова, що характеризують певні класи. Менш точні, але все ще високі показники

отримано при використанні BoW(точність 75,42 %) і N-грамних моделей (точність 73,07 %). Найменш ефективним для Random Forest виявився Word2Vec (точність 58,98 %), через кореляцію ознак, які ускладнюють розподіл класів методом випадкових лісів.

Таким чином, ідентичність результатів між моделями, написаними вручну, і тими, що були реалізовані за допомогою Scikit-learn, пояснюється спільною математичною основою, правильністю реалізації алгоритмів та однаковими тренувальними й тестовими даними. Отримані результати свідчать про доцільність використання розріджених векторних представлень (Bag-of-Words, N-грами та TF-IDF) для задач автоматичного визначення психічних станів у текстах, у той час, як щільні семантичні представлення (Word2Vec) потребують моделей, здатних працювати з корельованими ознаками та нелінійними межами розподілу.

3.2 Аналіз матриці невідповідностей

Для глибшого розуміння роботи моделей та оцінки їхньої здатності до правильного класифікування розглянемо також отримані матриці невідповідностей.

Confusion matrix відображають розподіл передбачень моделі по п’яти класах психічних станів(Stress, Depression, Bipolar disorder, Personality disorder, Anxiety), що дає змогу оцінити не лише загальну точність класифікації, а й характер помилок – які саме класи найчастіше плутаються між собою.

Отримані матриці для ручних реалізацій моделей також показали подібні патерни класифікації до бібліотечних аналогів. Зокрема, найбільші значення знаходяться на діагоналі матриці, що свідчить про правильне передбачення більшості прикладів. Проте, між суміжними класами

простежується певна кількість помилок класифікації, що обумовлено близькістю лексичних патернів у текстах різних психічних станів.

Наприклад, для моделі SVM, навченої на Bag-of-Words, матриця, реалізована вручну, містить великі значення на діагоналі (більше ніж 150–180+ випадків для кожного класу), водночас помилки розкидані по суміжних класах у межах кількох десятків прикладів.

```
[[183   5   5  35   8]
 [ 12 159   8  53   9]
 [ 12  19 155  41  10]
 [ 13  22   2 188  15]
 [ 27  18   5  32 156]]
```

Рис 3.1 Матриця невідповідності для SVM(ручна реалізація)+ Bag-of-Words

Аналогічна модель, реалізована за допомогою Scikit-learn, демонструє схожу структуру матриці, хоча й із незначними відмінностями в кількості помилкових класифікацій. Ці відмінності можуть бути викликані особливостями реалізації, такими як, генерація випадкових чисел, та оптимізації бібліотеки.

```
[[206  10   0   9  11]
[ 39 141  10  33  18]
[ 36  22 148  19  12]
[ 45  27   5 147  16]
[ 39  11   1  25 162]]
```

Рис 3.2 Матриця невідповідності для SVM+ Bag-of-Words

Використання різних методів векторизації тексту суттєво впливає на розподіл передбачень. Зокрема, щільні векторні представлення, як-от: Word2Vec, значно ускладнюють модель, що проявляється в матрицях плутанини великою кількістю помилкових класифікацій і розмитістю між класами. У таких випадках матриця демонструє зниження значень на діагоналі та збільшення неправильних передбачень між усіма класами, що знижує загальну якість класифікації.

```
[[140  65   4  23   4]
[ 34 188   9   5   5]
[ 52 124  30  27   4]
[ 38 147   6  40   9]
[ 54 127  10  13  34]]
```

Рис 3.2 Матриця невідповідності для SVM+ word2vec

Для моделей на основі Naive Bayes та Random Forest, навчальних на розріджених поданнях (BoW, N-grams, TF-IDF), матриці неточності мають чітко виражену діагональ, що свідчить про високу точність та ефективність класифікації. Однак і тут спостерігається характерна плутанина між деякими класами, яка зумовлена схожістю лексичних ознак у текстах, що відображає природну неоднозначність у визначенні меж психічних станів за текстовими даними.

```
[[223   0   0  12   1]
 [135  48   0  57   1]
 [160   2  48  26   1]
 [109   2   0 129   0]
 [175   3   0  28  32]]
```

Рис 3.3 Матриця невідповідності для Naive Bayes+ *TF–IDF*

Загалом, порівняння матриць невідповідностей для реалізацій із нуля та бібліотечних моделей підтверджує їхню функціональну еквівалентність у задачі багатокласової класифікації психічних станів. Невеликі відмінності в деталях помилок можна пояснити різницею в оптимізаціях, обробці крайових випадків та випадковості, притаманній деяким алгоритмам.

Даний аналіз демонструє, що розріджені векторні подання тексту є оптимальними для точного розмежування класів, тоді як щільні семантичні вектори вимагають додаткових підходів або адаптованих моделей для підвищення розпізнавальної здатності.

3.3 Висновки розділу 3

У цьому розділі було проведено комплексну оцінку ефективності трьох класичних моделей машинного навчання – Naive Bayes, SVM та Random Forest – у задачі автоматичного виявлення депресивних текстів на основі збірки даних «Reddit Mental Health Data».

Реалізації моделей, написані власноруч без використання допоміжних бібліотек, продемонстрували, що вибір способу векторизації тексту суттєво впливає на якість класифікації. Зокрема, Naive Bayes показав найкращі результати з розрідженими поданнями (Bag-of-Words, N-грамові моделі), що відповідає базовим припущенням моделі про умовну незалежність ознак. SVM найбільш ефективний із N-грамовою векторизацією, яка розширює контекстуальне представлення, а Random Forest вирізнявся високою точністю і стабільністю при використанні TF-IDF, що враховує значущість слів з урахуванням їх частоти в тексті.

Використання щільних векторних подань Word2Vec виявилося менш ефективним для Naive Bayes і SVM, хоча Random Forest показав результати понад 50 %, але все ж нижчі за TF-IDF. Це пояснюється особливостями семантичного представлення слів, що не завжди сумісні з припущеннями класичних моделей.

Порівняння алгоритмів написаних вручну та бібліотечних реалізацій підтвердило висновки статті «Benchmarking of Common Machine Learning Algorithms in Python», згідно з якими при відтворенні однакових гіперпараметрів результати за метриками F1 та accuracy відрізняються не більше ніж на 0–1 %. Водночас оптимізовані бібліотеки, що використовують технології на кшталт Cython та OpenMP, забезпечують суттєво кращу швидкодію, що робить їх більш ефективними для практичного застосування.

Використання бібліотек, як-от: Scikit-learn, значно зменшує час розробки та навчання моделей, а також спрощує процес написання. Це дає змогу фокусуватися на аналітичній частині дослідження і оптимізації параметрів, не витрачаючи ресурси на реалізацію базових алгоритмів. Натомість реалізації написані власноруч дають глибше розуміння алгоритмів і надають можливість модифікувати код відповідно до специфічних вимог проблеми, що є незамінним у дослідницьких і експериментальних задачах.

Аналіз матриць невідповідностей показав, що бібліотечні реалізації демонструють більш чітке розмежування класів із меншою кількістю помилкових класифікацій, особливо в близьких за ознаками категоріях. Реалізації написані без використання додаткових бібліотек мають трохи більше хибних спрацьовувань, що пояснюється відсутністю додаткових оптимізацій і внутрішніх механізмів контролю якості, які є в бібліотечних моделях. Водночас помилки між схожими психічними станами менш критичні, ніж радикальні неправильні класифікації, що важливо враховувати при подальшому вдосконаленні систем.

Отже, отримані результати підтвердили доцільність застосування оптимізованих бібліотечних моделей у поєднанні з відповідними методами векторизації для створення надійних і точних систем автоматичного виявлення депресії в текстах. Водночас моделі написані власноруч залишаються корисним інструментом для глибшого розуміння процесів класифікації та адаптації алгоритмів під конкретні дослідницькі завдання.

ВИСНОВКИ

У ході виконання дослідження створено й апробовано комплексну систему автоматизованого виявлення симптомів депресії в текстах, що ґрунтується на методах обробки природної мови та машинного навчання і тим самим підтверджує як актуальність, так і практичну цінність обраної теми.

Завдання кваліфікаційної роботи виконано, а саме: побудовано стандартизований конвеєр попередньої обробки текстів, що включає нормалізацію регістру, вилучення пунктуації та стоп-слів, токенізацію, лематизацію і стемінг. Це дозволило зменшити кількість шуму в даних більш ніж на 65 % і залишити лише релевантні лексеми.

Реалізовано й порівняно чотири методи векторизації текстів (Bag-of-Words, n-грами, TF-IDF і Word2Vec) у поєднанні з трьома класичними алгоритмами класифікації (Naive Bayes, Support Vector Machine та Random Forest). Для кожної конфігурації проведено повноцінний експеримент із крос-валідацією і розрахунком метрик точності, чутливості, специфічності та F1-показника. Детальний аналіз матриць невідповідностей та метрик precision/recall дозволив визначити сильні та слабкі сторони кожного підходу, а також встановити, як вибір моделі та методу векторизації впливає на точність виявлення симптомів депресії.

Мету роботи досягнуто, оскільки було створено онлайн-інструмент для швидкого й точного первинного скринінгу депресії. Надано змогу користувачам швидко та ефективно оцінити свій психічний стан і в режимі реального часу відстежувати динаміку власних показників. Для цього було створено захищений веб-сервіс, який підтримує формати TXT, PDF, DOCX, а також голосовий ввід тексту і повністю відповідає вимогам GDPR.

У процесі здійсненої роботи вперше створено застосунок для визначення ментального стану людини за допомогою машинного навчання та NLP, який підтримує обробку текстів українською та англійською мовами. Було також реалізовано алгоритм, який автоматично вибирає найефективнішу комбінацію методів векторизації та класифікаційних алгоритмів для задачі виявлення депресивних симптомів. Удосконалено механізм аналізу динаміки емоційного стану користувачів, інтегрувавши збір часових рядів результатів скринінгу та візуалізацію їх у вебінтерфейсі, що дає змогу користувачам відстежувати зміни настрою.

Результати дослідження мають практичне значення, оскільки створений сервіс можна запускати як безплатний анонімний тестувальник для широкої аудиторії. У майбутньому його можна інтегрувати в клінічні системи для автоматичного виявлення пацієнтів із підвищеним ризиком, використовувати для довготривалого моніторингу ефективності психотерапії, а також можна додати розширення, щоби підключати його до соціальних і HR-платформ як модуль емоційної аналітики текстів. Таким чином, дослідження одночасно поглиблює теоретичні підходи до психологічної діагностики, демонструючи ефективність оптимізованих класичних ML-методів, і пропонує готовий інструмент, що робить психологічну підтримку доступнішою в реальному житті.

Список використаних джерел

1. World Health Organization (Mental disorders) URL:https://www.who.int/news-room/fact-sheets/detail/mental-disorders#:~:text=A%20mental%20disorder%20is%20characterized,in%20important%20areas%20of%20functioning.
2. Reddit Mental Health Data URL: https://www.kaggle.com/datasets/neelghoshal/reddit-mental-health-data
3. Student-Depression-Text URL: https://www.kaggle.com/datasets/nidhiy07/student-depression-text
4. Machine Learning Algorithms for Depression Detection and Their Comparison. URL: https://arxiv.org/pdf/2301.03222
5. Python Machine Learning by Sebastian Raschka and Vahid Mirjalili URL:http://radio.eng.niigata-u.ac.jp/wp/wp-content/uploads/2020/06/python-machine-learning-2nd.pdf
6. Pattern Recognition and Machine Learning by Christopher M. Bishop URL:https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf
7. Naive Bayes and Text Classification – Introduction and Theory by Sebastian Raschka URL: https://sebastianraschka.com/Articles/2014_naive_bayes_1.html
8. Rohan Saha, Influence of various text embeddings on clustering performance in NLP, Machine Learning (cs.LG), 2023
9. Xin Rong, word2vec Parameter Learning Explained, Computation and Language (cs.CL), 2014. URL: https://arxiv.org/abs/1411.2738
10. Bayesian Naive Bayes classifiers to text classification URL: https://journals.sagepub.com/doi/full/10.1177/0165551516677946
11. Neural Networks from Scratch in Python by Harrison Kinsley & Daniel Kukiela

12. Hands-on Machine Learning with Sckit-Learn, Keras and TensorFlow by Aurelien Geron
13. Handbook of Natural Language Processing by Nitin Indurkhya and Fred J. Damerau
14. Speech and Language Processing" by Daniel Jurafsky and James H. Martin. URL: https://web.stanford.edu/~jurafsky/slp3/
15. MASON-NLP at eRisk 2023: Deep Learning-Based Detection of Depression Symptoms from Social Media Texts. URL: https://www.researchgate.net/publication/374870279_MASON-NLP_at_eRisk_2023_Deep_Learning-Based_Detection_of_Depression_Symptoms_from_Social_Media_Texts
16. Depression Prediction Model based on NLP. URL: https://www.researchgate.net/publication/386533281_Depression_Prediction_Model_based_on_NLP