

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
(повна назва)  
Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Дослідження методів персоналізованого прогнозування  
настрою з використанням Core ML  
(тема)

Виконав:  
здобувач \_\_\_\_\_ другого \_\_\_\_\_ року навчання,  
групи \_\_\_\_\_ СШМ-24-1 \_\_\_\_\_

Дмитро ВЕЛИЧКО  
(власне ім'я, прізвище)

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

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

Освітня програма Системи штучного інтелекту  
(повна назва освітньої програми)

Керівник проф. Марина КУДРЯВЦЕВА  
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ \_\_\_\_\_ Лариса ЧАЛА \_\_\_\_\_  
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_  
Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)  
Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)  
Освітня програма \_\_\_\_\_ Системи штучного інтелекту \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Величку Дмитру Олексійовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів персоналізованого прогнозування настрою з використанням Core ML

затверджена наказом університету від 6 квітня 2026 р. № 269Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 19 травня 2026 р.

3. Вихідні дані до роботи наукові статті та публікації у сфері аналізу емоційного стану людини та машинного навчання, документація та матеріали щодо використання алгоритмів класифікації Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting, документація бібліотек scikit-learn та coremltools, матеріали щодо технології Core ML та її інтеграції у мобільні застосунки iOS.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Аналіз предметної галузі та постановка задачі

2) Теоретичні дослідження

3) Експериментальні дослідження

### КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи                         | Строк / термін виконання етапів роботи | Примітка |
|---|---------------------------------------------|----------------------------------------|----------|
| 1 | Отримання завдання на кваліфікаційну роботу | 06.04.2026                             | виконано |
| 2 | Визначення проблеми                         | 10.04.2026                             | виконано |
| 3 | Дослідження алгоритмів машинного навчання   | 15.04.2026                             | виконано |
| 4 | Проведення експериментальних досліджень     | 18.04.2026                             | виконано |
| 5 | Проектування та реалізація додатку          | 21.04.2026                             | виконано |
| 6 | Оформлення пояснювальної записки            | 05.05.2026                             | виконано |
| 7 | Захист кваліфікаційної роботи               | 19.05.2026                             | виконано |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |
|   |                                             |                                        |          |

Дата видачі завдання 6 квітня 2026 р.

Здобувач   
(підпис)

Керівник роботи \_\_\_\_\_ проф. Марина КУДРЯВЦЕВА  
(підпис) (посада, власне ім'я, прізвище)

## РЕФЕРАТ

Пояснювальна записка: 100 с., 44 рис., 1 табл., 1 дод., 21 джерело.

АНАЛІЗ ДАНИХ, КЛАСИФІКАЦІЯ, МАШИННЕ НАВЧАННЯ, МОБІЛЬНИЙ ЗАСТОСУНОК, НАСТРІЙ, ПРОГНОЗУВАННЯ, CORE ML, IOS, SCIKIT-LEARN, SWIFTUI.

Об'єктом дослідження є прогнозування емоційного стану користувача на основі поведінкових показників, отриманих із мобільного середовища.

Предметом дослідження є методи машинного навчання для класифікації та прогнозування настрою користувача на основі аналізу поведінкових даних.

Метою роботи є розроблення математичної моделі визначення інтегрального показника настрою на основі поведінкових ознак та експериментального дослідження методів машинного навчання для його класифікації у мобільному середовищі Core ML.

У роботі використано методи машинного навчання для задач класифікації, зокрема логістичну регресію, метод k-найближчих сусідів, дерево рішень, Random Forest та Gradient Boosting.

У результаті виконання роботи запропоновано математичну модель визначення інтегрального показника настрою користувача на основі поведінкових показників, розроблено методику перетворення інтегрального показника у класи емоційного стану, проведено експериментальне дослідження алгоритмів машинного навчання та визначено найбільш ефективну модель для локального прогнозування настрою.

Отримані результати можуть бути використані під час розроблення інтелектуальних мобільних застосунків для моніторингу психологічного стану користувачів та підтримки їх емоційного благополуччя.

## **ABSTRACT**

Master's thesis contains: 100 pp., 44 fig., 1 tabl., 1 ann., 21 references.

CLASSIFICATION, CORE ML, DATA ANALYSIS, IOS, MACHINE LEARNING, MOBILE APPLICATION, MOOD, PREDICTION, SCIKIT-LEARN, SWIFTUI.

The object of research is the prediction of a user's emotional state based on behavioral indicators obtained from a mobile environment.

The subject of research is machine learning methods for classification and prediction of a user's mood based on the analysis of behavioral data.

The aim of the work is to develop a mathematical model for determining an integral mood indicator based on behavioral features and to conduct an experimental study of machine learning methods for its classification in the Core ML mobile environment.

The work uses machine learning methods for classification tasks, in particular Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, and Gradient Boosting.

As a result of the work, a mathematical model for determining the integral mood indicator based on behavioral data is proposed, a method for transforming the integral indicator into emotional state classes is developed, an experimental study of machine learning algorithms is conducted, and the most effective model for on-device mood prediction is identified.

The obtained results can be used in the development of intelligent mobile applications for monitoring users' psychological state and supporting their emotional well-being.

## ЗМІСТ

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів .....                       | 9  |
| Вступ.....                                                                                     | 10 |
| 1 Аналіз предметної галузі та постановка задачі.....                                           | 12 |
| 1.1 Моніторинг емоційного стану людини.....                                                    | 12 |
| 1.1.1 Поняття настрою та його вплив на психологічне благополуччя особистості.....              | 15 |
| 1.1.2 Значення систематичного відстеження емоційного стану користувача.....                    | 16 |
| 1.1.3 Роль цифрових технологій у підтримці та аналізі ментального здоров'я .....               | 18 |
| 1.2 Методи машинного навчання для прогнозування психологічних показників .....                 | 19 |
| 1.2.1 Поведінкові та фізіологічні фактори, що впливають на формування настрою .....            | 20 |
| 1.2.2 Джерела та особливості формування наборів даних для аналізу емоційного стану.....        | 21 |
| 1.2.3 Використання ансамблевих алгоритмів у задачах класифікації психологічних показників..... | 22 |
| 1.3 Технології on-device штучного інтелекту в iOS-застосунках.....                             | 23 |
| 1.3.1 Використання Core ML для інтеграції моделей машинного навчання .....                     | 24 |
| 1.3.2 Інструменти підготовки та конвертації моделей для мобільного середовища.....             | 26 |
| 1.4 Аналіз потреб та очікувань користувачів.....                                               | 27 |
| 1.4.1 Визначення цільової аудиторії застосунків для моніторингу та аналізу настрою.....        | 28 |
| 1.4.2 Огляд існуючих рішень для моніторингу та аналізу настрою ..                              | 28 |

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| 1.4.3 Потреби користувачів у функціональності аналізу та прогнозування настрою .....  | 29 |
| 1.5 Визначення функціональних можливостей розроблюваного застосунку .....             | 30 |
| 1.5.1 Основні функції системи відстеження настрою .....                               | 30 |
| 1.5.2 Використання машинного навчання для прогнозування емоційного стану .....        | 31 |
| 1.5.3 Засоби аналітики та візуалізації змін настрою .....                             | 31 |
| 1.6 Постановка задачі .....                                                           | 32 |
| 2 Теоретичні дослідження .....                                                        | 33 |
| 2.1 Поведінкові показники у задачах прогнозування настрою .....                       | 33 |
| 2.1.1 Основні поведінкові фактори, що впливають на настрій .....                      | 34 |
| 2.1.2 Формалізація поведінкових даних для задач машинного навчання .....              | 36 |
| 2.1.3 Обмеження та проблеми використання поведінкових даних ...                       | 37 |
| 2.1.4 Математична модель визначення інтегрального показника настрою користувача ..... | 38 |
| 2.2 Алгоритми машинного навчання для задач класифікації настрою ...                   | 41 |
| 2.2.1 Логістична регресія .....                                                       | 43 |
| 2.2.2 Метод k-найближчих сусідів .....                                                | 44 |
| 2.2.3 Дерево рішень .....                                                             | 46 |
| 2.3 Ансамблеві методи машинного навчання .....                                        | 48 |
| 2.3.1 Random Forest .....                                                             | 49 |
| 2.3.2 Gradient Boosting .....                                                         | 50 |
| 2.3.3 Порівняльні характеристики ансамблевих моделей .....                            | 51 |
| 2.4 Особливості підготовки даних для моделей машинного навчання ...                   | 53 |
| 2.4.1 Проблема незбалансованості класів .....                                         | 54 |
| 2.4.2 Складність та реалістичність структури даних .....                              | 56 |
| 2.4.3 Методика порівняння моделей машинного навчання .....                            | 56 |
| 2.5 Технологія Core ML для інтеграції моделей машинного навчання ...                  | 57 |

|                                                                                   |     |
|-----------------------------------------------------------------------------------|-----|
| 2.5.1 Архітектура Core ML .....                                                   | 58  |
| 2.5.2 Конвертація моделей scikit-learn у формат Core ML .....                     | 59  |
| 2.5.3 Переваги використання on-device машинного навчання в iOS. ....              | 60  |
| 3 Експериментальні дослідження .....                                              | 62  |
| 3.1 План експериментів .....                                                      | 62  |
| 3.2 Опис використаного набору даних .....                                         | 62  |
| 3.3 Реалізація експериментального середовища .....                                | 66  |
| 3.3.1 Використані програмні бібліотеки та інструменти .....                       | 66  |
| 3.3.2 Попередня обробка та підготовка даних до навчання .....                     | 68  |
| 3.3.3 Побудова моделей машинного навчання .....                                   | 69  |
| 3.3.4 Крос-валідація моделей та оцінювання їх узагальнювальної<br>здатності ..... | 71  |
| 3.4 Результати дослідження .....                                                  | 73  |
| 3.4.1 Результати класифікації для моделі Logistic Regression .....                | 74  |
| 3.4.2 Результати класифікації для моделі K-Nearest Neighbors .....                | 77  |
| 3.4.3 Результати класифікації для моделі Decision Tree .....                      | 79  |
| 3.4.4 Результати класифікації для моделі Random Forest .....                      | 82  |
| 3.4.5 Результати класифікації для моделі Gradient Boosting .....                  | 86  |
| 3.4.6 Порівняльний аналіз результатів моделей .....                               | 90  |
| 3.5 Інтеграція Gradient Boosting у Core ML .....                                  | 92  |
| 3.5.1 Визначення оптимальної моделі прогнозування настрою .....                   | 92  |
| 3.5.2 Конвертація моделі у формат Core ML .....                                   | 93  |
| 3.5.3 Підготовка моделі до інтеграції у мобільний застосунок .....                | 94  |
| Висновки .....                                                                    | 95  |
| Перелік джерел посилання .....                                                    | 98  |
| Додаток А Відомість кваліфікаційної роботи .....                                  | 100 |

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

AUC – Area Under Curve – площа під ROC-кривою, метрика оцінювання якості класифікації;

Core ML – фреймворк машинного навчання від Apple для інтеграції моделей у iOS-застосунки;

Decision Tree – алгоритм класифікації, що використовує деревоподібну структуру рішень;

F1-score – гармонійне середнє між точністю та повнотою класифікації;

Gradient Boosting – ансамблевий метод машинного навчання, що поєднує слабкі моделі у сильну;

iOS – операційна система для мобільних пристроїв Apple;

K-Nearest Neighbors – алгоритм класифікації на основі найближчих сусідів;

Logistic Regression – статистичний метод класифікації, що використовує логістичну функцію;

Mood Prediction – задача прогнозування емоційного стану користувача;

Precision – метрика оцінювання, що визначає частку правильних позитивних передбачень;

Random Forest – ансамблевий алгоритм, що використовує множину дерев рішень;

Recall – метрика оцінювання, що визначає частку знайдених позитивних випадків;

ROC – Receiver Operating Characteristic – характеристична крива, що відображає співвідношення між чутливістю та специфічністю моделі;

SwiftUI – фреймворк для побудови інтерфейсів користувача в iOS-застосунках.

## ВСТУП

Сучасний розвиток інформаційних технологій та стрімке поширення мобільних пристроїв створюють нові можливості для використання інтелектуальних систем у сфері повсякденного життя людини. Однією з важливих тенденцій останніх років є зростання інтересу до технологій цифрового моніторингу психологічного стану та інструментів самопостереження за емоційним самопочуттям. Такі системи дозволяють користувачам аналізувати власний емоційний стан, виявляти закономірності між поведінковими факторами та настроєм, а також отримувати рекомендації щодо покращення психологічного благополуччя. У зв'язку з цим зростає актуальність створення мобільних застосунків, здатних не лише фіксувати дані про настрій користувача, але й здійснювати інтелектуальний аналіз накопиченої інформації.

Одним із перспективних напрямів розвитку таких систем є використання методів машинного навчання для прогнозування емоційного стану на основі поведінкових показників. Сучасні алгоритми машинного навчання дозволяють виявляти складні залежності між різними характеристиками діяльності людини, такими як тривалість сну, фізична активність, рівень використання цифрових пристроїв або інші аспекти повсякденного життя. Використання подібних моделей дає змогу формувати прогноз настрою користувача, що може бути корисним як для особистої аналітики, так і для досліджень у сфері психології та поведінкових наук.

Водночас важливою особливістю сучасних мобільних застосунків є необхідність забезпечення високого рівня конфіденційності персональних даних. Оскільки інформація про психологічний стан людини належить до чутливих даних, обробка таких відомостей повинна здійснюватися з дотриманням принципів приватності та безпеки. Одним із ефективних підходів до вирішення цієї проблеми є використання технологій on-device

машинного навчання, які дозволяють виконувати обчислення безпосередньо на мобільному пристрої користувача без передачі даних на віддалені сервери. У середовищі мобільної платформи iOS для реалізації подібних можливостей використовується технологія Core ML, що забезпечує інтеграцію моделей машинного навчання у мобільні застосунки та виконання прогнозування безпосередньо на пристрої користувача.

Актуальність теми кваліфікаційної роботи зумовлена необхідністю дослідження методів персоналізованого прогнозування настрою на основі поведінкових показників користувача та інтеграції таких методів у мобільні застосунки. Розробка подібних систем дозволяє не лише розширити можливості мобільних технологій у сфері аналізу психологічного стану, але й створити інструменти, що можуть сприяти підвищенню рівня саморефлексії користувачів та покращенню їхнього емоційного благополуччя.

Підставою для виконання даної роботи є необхідність дослідження сучасних підходів до прогнозування психологічних показників із використанням методів машинного навчання, а також можливостей їх інтеграції у мобільні застосунки платформи iOS. У межах дослідження розглядаються поведінкові фактори, що можуть впливати на емоційний стан користувача, аналізуються алгоритми машинного навчання для задач класифікації настрою та вивчаються технології реалізації інтелектуальних функцій у мобільних застосунках.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1 Моніторинг емоційного стану людини

Моніторинг емоційного стану людини є важливою складовою сучасних систем підтримки психічного здоров'я та персоналізованих цифрових сервісів. Розвиток мобільних технологій [11], сенсорних пристроїв та методів машинного навчання дозволяє автоматизувати процес збору та аналізу поведінкових і фізіологічних даних користувача. Типова структура таких систем (рисунок 1.1) включає джерела даних, модуль обробки сигналів, виділення ознак та модель машинного навчання, яка здійснює оцінювання або прогнозування емоційного стану. На практиці джерелами інформації можуть виступати мобільні пристрої, фітнес-браслети або інші носимі сенсори, що реєструють активність користувача, параметри сну, фізичну активність та інші показники щоденної поведінки.

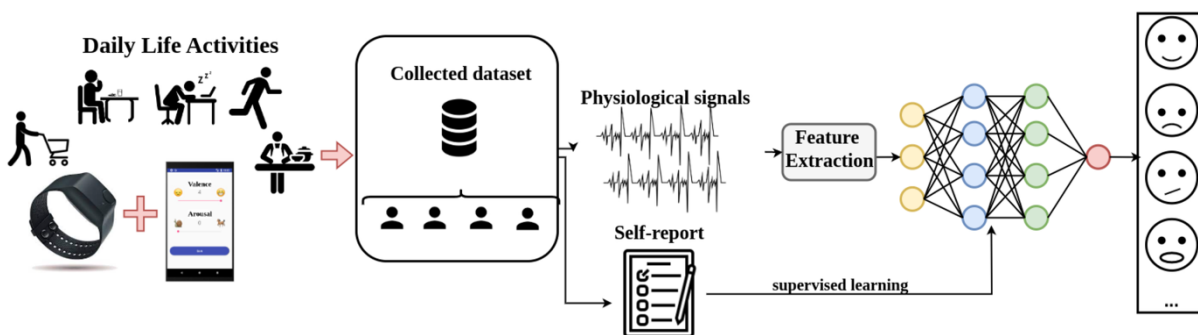


Рисунок 1.1 – Загальна схема системи моніторингу емоційного стану на основі поведінкових та фізіологічних даних

Одним із поширених підходів до формалізації емоційного стану людини є двовимірна модель емоцій, яка базується на параметрах валентності та активації (рисунок 1.2). Валентність характеризує позитивність або негативність емоційного переживання, тоді як активація

відображає рівень фізіологічного або психологічного збудження. Комбінація цих двох параметрів дозволяє описати широкий спектр емоційних станів, таких як задоволення, збудження, пригніченість або сонливість. Використання подібної моделі дозволяє представити емоційний стан у вигляді координат на площині та застосовувати математичні методи класифікації або кластеризації для подальшого аналізу.

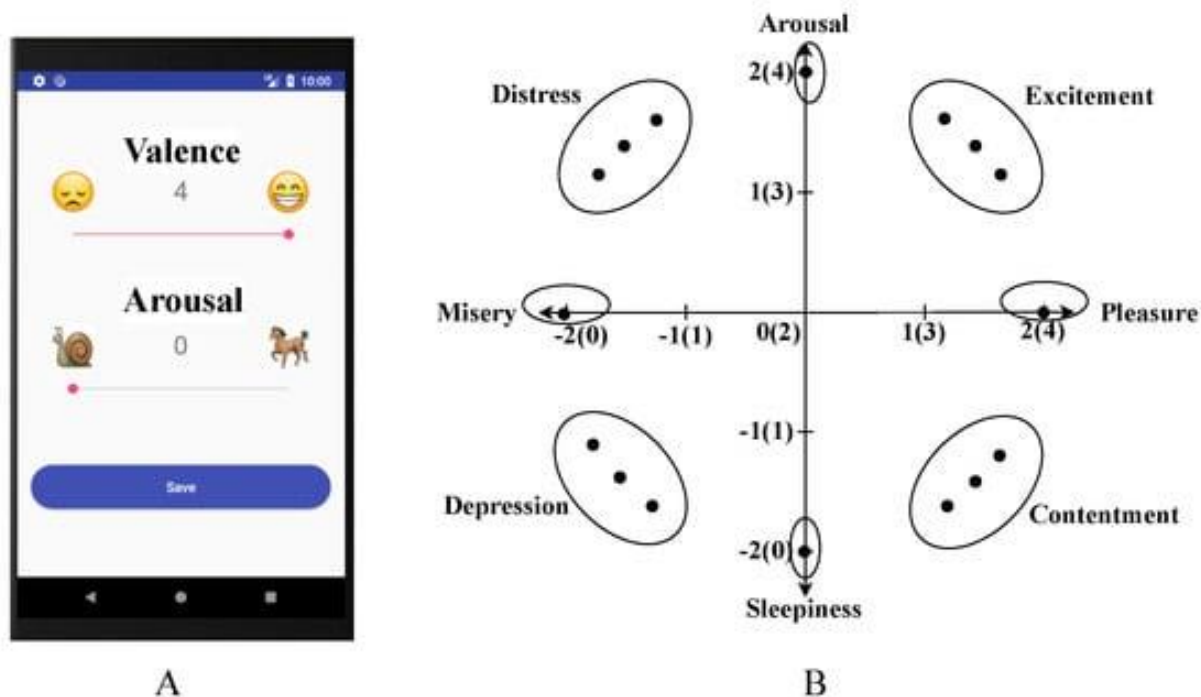


Рисунок 1.2 – Модель представлення емоційного стану у координатах валентності та активації

Сучасні підходи до моніторингу емоційного стану також активно використовують методи машинного навчання для аналізу зібраних даних. У таких системах інформація може надходити як із фізіологічних сенсорів, так і з суб'єктивних оцінок користувача, які вводяться вручну у мобільному застосунку. Отримані дані використовуються для формування навчальної вибірки, на основі якої будується модель класифікації або прогнозування емоційного стану (рисунок 1.3).

Після навчання модель здатна автоматично визначати або передбачати настрій користувача на основі поточних показників активності. Подібний підхід забезпечує можливість створення персоналізованих систем спостереження за емоційним станом та формування індивідуальних рекомендацій щодо способу життя.

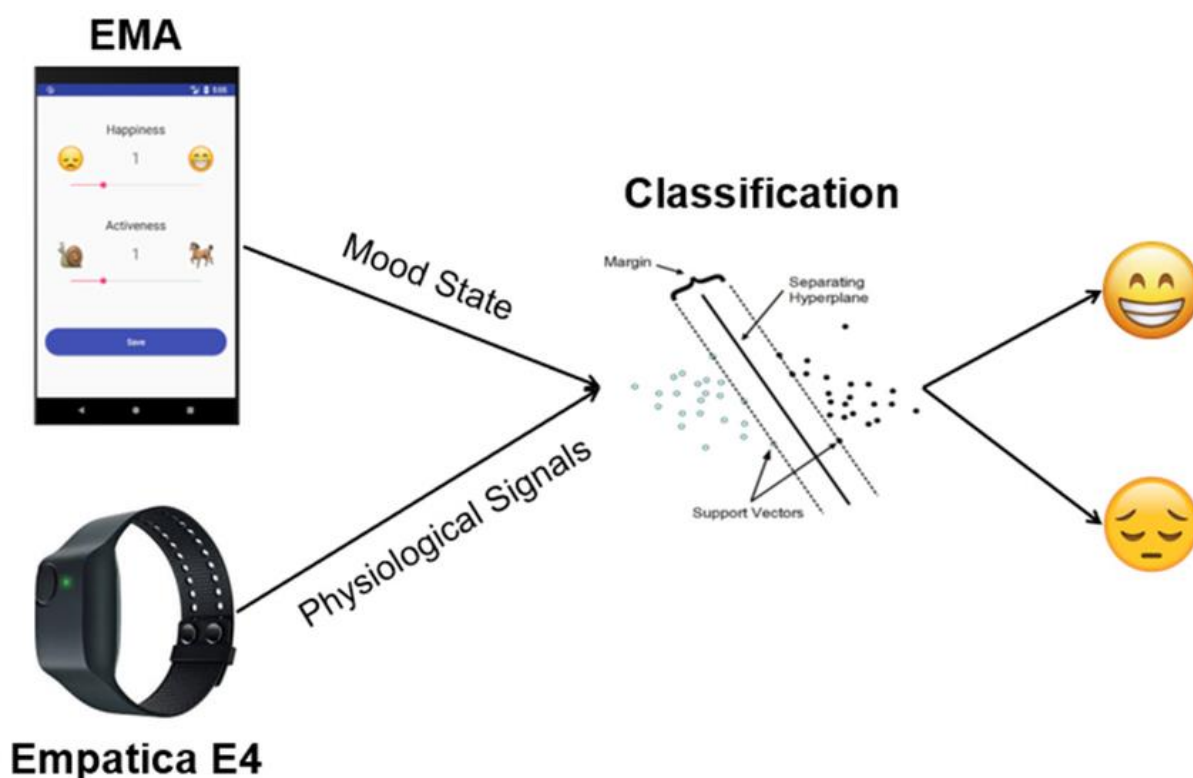


Рисунок 1.3 – Схема класифікації емоційного стану на основі фізіологічних сигналів та суб'єктивних оцінок користувача

Розробка мобільних застосунків для моніторингу настрою передбачає інтеграцію методів збору даних, алгоритмів машинного навчання та засобів візуалізації результатів. У сучасних системах значна увага приділяється конфіденційності персональних даних, тому обробка інформації часто виконується безпосередньо на мобільному пристрої користувача без використання зовнішніх серверів.

### 1.1.1 Поняття настрою та його вплив на психологічне благополуччя особистості

Настрій є складним психоемоційним станом людини, який характеризується відносно тривалим емоційним фоном і впливає на сприйняття подій, поведінку та прийняття рішень. На відміну від короткочасних емоцій, настрій формується під впливом сукупності внутрішніх і зовнішніх факторів, таких як фізичний стан організму, соціальна взаємодія, рівень активності та індивідуальні психологічні особливості. Для систематизації різноманітних емоційних проявів у психології застосовуються спеціальні моделі класифікації емоцій. Однією з найбільш відомих є модель Роберта Плутчика (рисунок 1.4), представлена у вигляді так званого «колеса емоцій».



Рисунок 1.4 – Колесо емоцій Плутчика як модель класифікації базових емоцій людини

У цій моделі емоції організовані у вигляді концентричних секторів, що відображають базові емоційні стани людини та їх різні рівні інтенсивності. Комбінації емоцій можуть формувати більш складні емоційні переживання, що дозволяє більш детально бачити емоційний стан людини.

Емоційний стан та настрої людини мають значний вплив на її психологічне благополуччя. Саме тому у сучасних дослідженнях значна увага приділяється методам спостереження, аналізу та прогнозування настрою людини, зокрема із застосуванням цифрових технологій та методів машинного навчання.

У контексті розробки інформаційних систем моніторингу емоційного стану важливим є можливість формалізації настрою у вигляді кількісних параметрів або категорій [12]. Це дозволяє використовувати математичні моделі та алгоритми машинного навчання для аналізу поведінкових даних користувача. Зокрема, мобільні застосунки можуть збирати інформацію про щоденну активність, фізіологічні показники та суб'єктивні оцінки настрою користувача, після чого на основі отриманих даних формуються моделі прогнозування емоційного стану. Такий підхід створює передумови для побудови персоналізованих систем підтримки психологічного благополуччя, здатних аналізувати індивідуальні закономірності поведінки та виявляти фактори, що впливають на зміну настрою.

#### 1.1.2 Значення систематичного відстеження емоційного стану користувача

Систематичне відстеження емоційного стану користувача є важливим інструментом аналізу психологічного благополуччя та формування усвідомленого ставлення до власного емоційного стану. Регулярна фіксація настрою дозволяє накопичувати історію емоційних змін у часі (рисунок 1.5), що створює основу для подальшого аналізу закономірностей поведінки та впливу різних факторів на психологічний стан людини.

У сучасних умовах такі процеси дедалі частіше реалізуються за допомогою мобільних застосунків, які забезпечують зручні механізми введення даних та автоматичну обробку інформації.

Регулярне відстеження настрою дозволяє користувачеві краще усвідомлювати власні емоційні реакції та виявляти фактори, що впливають на їх зміну. Аналіз накопичених записів може допомогти визначити взаємозв'язок між повсякденними активностями, рівнем фізичної активності, тривалістю сну або іншими поведінковими характеристиками та загальним емоційним станом. Така інформація може використовуватися як для особистого самопостереження, так і для подальшого аналізу з використанням методів обробки даних.

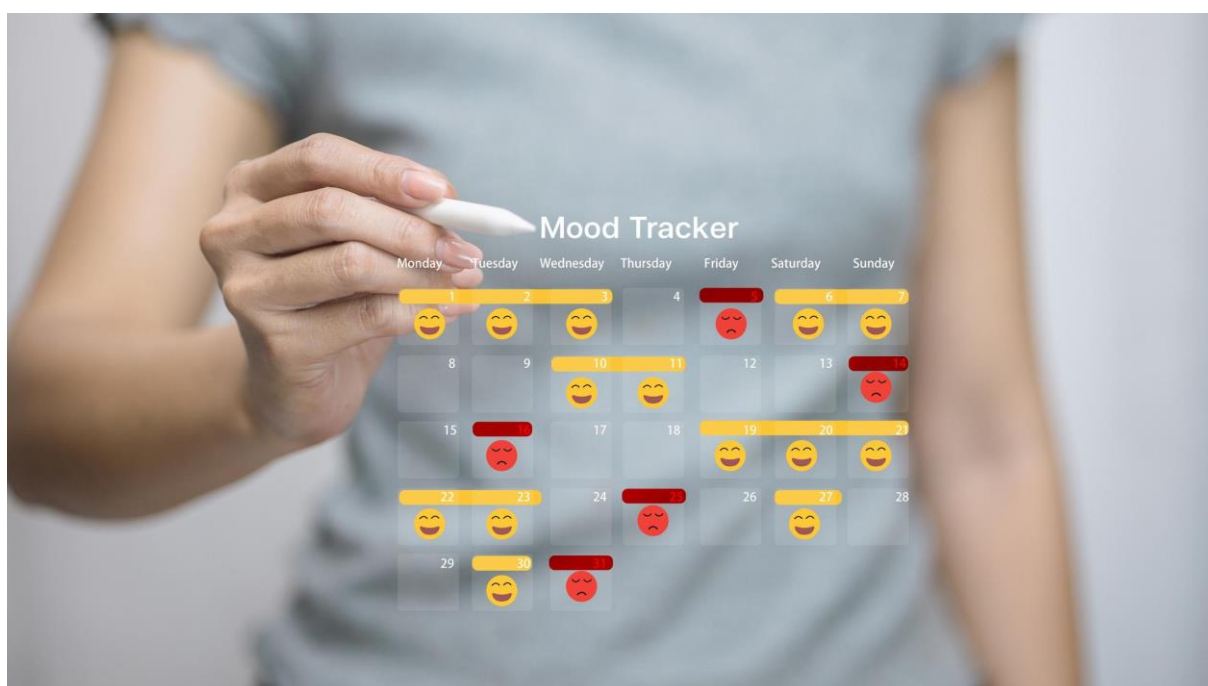


Рисунок 1.5 – Візуалізація систематичного відстеження настрою користувача у цифровому середовищі

Важливим аспектом систематичного моніторингу емоційного стану є можливість використання зібраних даних для побудови моделей машинного навчання. Накопичені історичні записи створюють навчальну вибірку, яка

дозволяє формувати моделі прогнозування настрою користувача. На основі таких моделей можна здійснювати оцінювання поточного емоційного стану або передбачати його зміну залежно від поведінкових параметрів користувача, що відкриває можливості для створення персоналізованих систем підтримки психологічного благополуччя.

### 1.1.3 Роль цифрових технологій у підтримці та аналізі ментального здоров'я

Цифрові технології відіграють важливу роль у сучасних підходах до підтримки та аналізу ментального здоров'я людини. Використання таких технологій дає можливість збирати великі обсяги персоналізованих даних, які можуть бути використані для аналізу психологічного стану людини та виявлення змін у її емоційному самопочутті.

Мобільні застосунки для підтримки ментального здоров'я забезпечують зручні механізми фіксації настрою, ведення емоційних щоденників, аналізу щоденних активностей та візуалізації отриманих результатів.

Завдяки інтеграції методів обробки даних і машинного навчання такі системи здатні виявляти закономірності між поведінковими факторами та емоційним станом користувача, а також формувати персоналізовані рекомендації щодо покращення психологічного благополуччя.

Окрім цього, сучасні цифрові рішення дозволяють автоматизувати процес аналізу ментального стану шляхом використання інтелектуальних алгоритмів, що працюють безпосередньо на мобільному пристрої. Такий підхід забезпечує підвищення рівня конфіденційності персональних даних користувача та створює умови для розробки персоналізованих систем прогнозування настрою, які можуть адаптуватися до індивідуальних особливостей поведінки та способу життя людини.

## 1.2 Методи машинного навчання для прогнозування психологічних показників

Методи машинного навчання активно застосовуються для аналізу психологічних показників та прогнозування емоційного стану людини. Використання алгоритмів обробки даних дозволяє виявляти складні закономірності між поведінковими характеристиками користувача, фізіологічними показниками та змінами емоційного стану. У сучасних дослідженнях машинне навчання застосовується для розв'язання задач класифікації, регресії та прогнозування психологічних показників. У межах такого підходу застосовуються різні групи алгоритмів (рисунок 1.6), зокрема методи навчання з учителем, методи навчання без учителя та ансамблеві моделі.

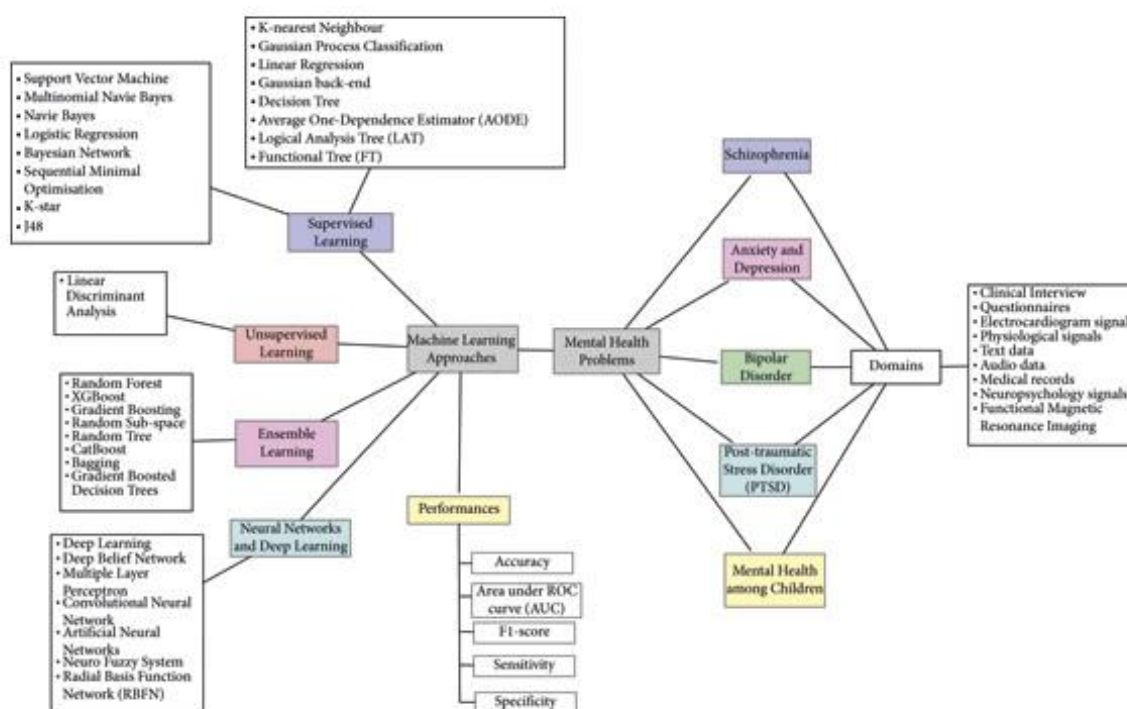


Рисунок 1.6 – Основні підходи машинного навчання для аналізу та прогнозування показників ментального здоров'я

До поширених алгоритмів належать метод опорних векторів, k-найближчих сусідів, наївний байєсівський класифікатор, дерева рішень та випадкові ліси. Крім того, для більш складних задач аналізу поведінкових даних використовуються нейронні мережі та моделі глибокого навчання, здатні обробляти великі обсяги даних та враховувати нелінійні залежності між параметрами.

Застосування машинного навчання у сфері аналізу психологічних показників передбачає використання навчальних вибірок, що формуються на основі поведінкових або фізіологічних даних користувача. Після навчання модель здатна оцінювати емоційний стан людини або прогнозувати його зміну на основі поточних показників активності. Для оцінювання ефективності таких моделей використовуються стандартні метрики якості, зокрема точність класифікації, F1-міра, чутливість, специфічність та площа під ROC-кривою.

### 1.2.1 Поведінкові та фізіологічні фактори, що впливають на формування настрою

Формування настрою людини є складним процесом, що залежить від сукупності поведінкових та фізіологічних факторів. До поведінкових факторів належать щоденні звички та активності, які характеризують спосіб життя людини, зокрема рівень фізичної активності, режим сну, тривалість використання цифрових пристроїв, соціальна взаємодія та загальна структура повсякденної діяльності. Дослідження показують, що регулярна фізична активність і достатня тривалість сну позитивно впливають на емоційний стан людини, тоді як недостатній відпочинок або надмірне використання електронних пристроїв можуть сприяти виникненню негативних емоційних переживань. Саме тому аналіз поведінкових показників є важливим елементом у системах прогнозування настрою.

Окрім поведінкових характеристик, значний вплив на формування настрою мають фізіологічні фактори, які відображають внутрішній стан організму. До таких факторів належать показники серцевого ритму, рівень фізичного навантаження, параметри сну, гормональні зміни та інші фізіологічні процеси, що можуть бути зафіксовані за допомогою носимих сенсорних пристроїв. Поєднання інформації про поведінкові та фізіологічні параметри дозволяє більш точно аналізувати емоційний стан людини та формувати моделі машинного навчання для прогнозування настрою. Використання таких даних створює основу для розробки персоналізованих систем моніторингу ментального здоров'я, здатних враховувати індивідуальні особливості способу життя користувача.

#### 1.2.2 Джерела та особливості формування наборів даних для аналізу емоційного стану

Формування наборів даних є одним із ключових етапів у задачах аналізу та прогнозування емоційного стану людини. Якість та повнота навчальної вибірки безпосередньо впливають на ефективність моделей машинного навчання, які використовуються для визначення або прогнозування настрою. У сучасних дослідженнях джерелами даних виступають різні цифрові пристрої та програмні системи, зокрема мобільні застосунки, носимі сенсорні пристрої та системи самозвіту користувача. Такі джерела дозволяють збирати інформацію про поведінкові характеристики людини, її фізичну активність, режим сну, використання цифрових пристроїв та суб'єктивну оцінку емоційного стану.

Спочатку (рисунок 1.7) відбувається збір первинних даних, що характеризують повсякденні активності користувача, такі як фізична активність, режим харчування, тривалість сну або інші поведінкові показники.

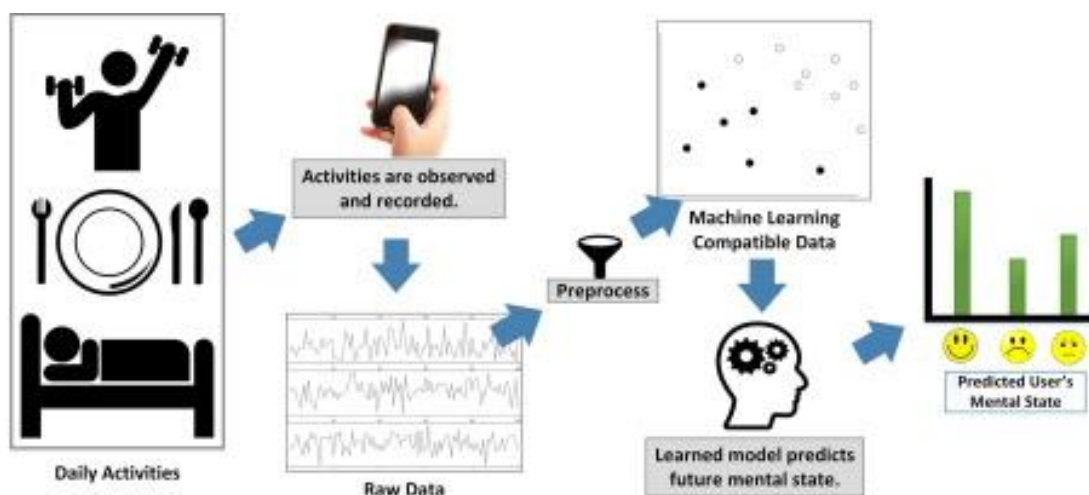


Рисунок 1.7 – Процес формування набору даних для прогнозування емоційного стану на основі поведінкових даних користувача

Отримані дані фіксуються у вигляді сирих сигналів або числових параметрів [13], після чого проходять етап попередньої обробки, який включає очищення даних, нормалізацію значень та виділення інформативних ознак. На основі підготовлених даних формується навчальна вибірка, сумісна з алгоритмами машинного навчання. Після формування та підготовки набору даних вони використовуються для навчання моделей машинного навчання, які здатні встановлювати залежності між поведінковими характеристиками та емоційним станом користувача.

### 1.2.3 Використання ансамблевих алгоритмів у задачах класифікації психологічних показників

Ансамблеві алгоритми машинного навчання є ефективним підходом до розв'язання задач класифікації, зокрема у сфері аналізу психологічних показників. Основна ідея ансамблевих методів полягає у поєднанні результатів декількох базових моделей для отримання більш точного та стабільного прогнозу. Замість використання одного класифікатора ансамблеві підходи формують сукупність моделей, кожна з яких навчається

на різних підмножинах даних або використовує різні принципи побудови моделі. Об'єднання результатів таких моделей дозволяє зменшити ризик перенавчання, підвищити узагальнювальну здатність алгоритму та покращити точність класифікації.

У задачах аналізу психологічних показників ансамблеві алгоритми дозволяють ефективно враховувати складні взаємозв'язки між поведінковими та фізіологічними характеристиками користувача. До найбільш поширених методів ансамблевого навчання належать випадковий ліс, градієнтний бустинг та методи бустингу рішень.

Застосування ансамблевих алгоритмів є особливо доцільним у задачах прогнозування настрою, де дані можуть характеризуватися високою варіативністю та наявністю шуму. Поєднання декількох моделей дозволяє підвищити надійність прогнозу та зменшити вплив окремих помилкових спостережень у навчальній вибірці.

### 1.3 Технології on-device штучного інтелекту в iOS-застосунках

Екосистема Apple включає декілька ключових технологій, що забезпечують підтримку on-device машинного навчання (рисунок 1.8). До основних інструментів належить фреймворк Core ML [1], який дозволяє інтегрувати навчальні моделі безпосередньо у застосунок та виконувати їх на пристрої користувача.

Крім того, для оптимізації обчислень використовуються апаратні можливості мобільних пристроїв, зокрема графічний процесор та нейронний процесор, що дозволяє значно підвищити продуктивність виконання моделей. Використання on-device технологій штучного інтелекту відкриває нові можливості для створення персоналізованих мобільних застосунків.

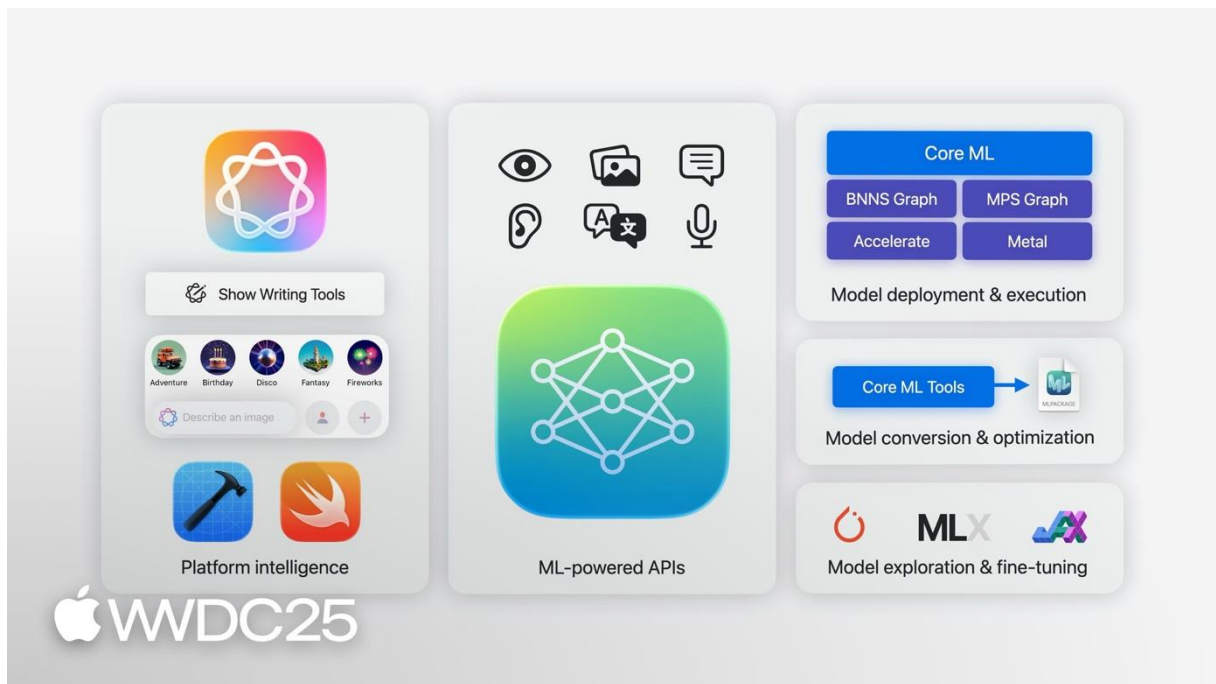


Рисунок 1.8 – Інструменти Apple для реалізації on-device моделей машинного навчання в iOS-застосунках

Завдяки локальному виконанню моделей машинного навчання застосунок може аналізувати дані користувача, формувати прогнозні оцінки та надавати індивідуальні рекомендації у реальному часі. У контексті систем моніторингу емоційного стану це дозволяє реалізувати механізми прогнозування настрою на основі поведінкових показників користувача без необхідності передачі персональних даних до хмарних сервісів, що підвищує рівень безпеки та приватності інформації.

### 1.3.1 Використання Core ML для інтеграції моделей машинного навчання

Core ML [1] є основним фреймворком платформи Apple, призначеним для інтеграції моделей машинного навчання у мобільні застосунки для операційних систем iOS, iPadOS, macOS та інших платформ екосистеми Apple. Цей фреймворк забезпечує можливість використання попередньо

навченої моделі безпосередньо у програмному застосунку та виконання прогнозування на мобільному пристрої користувача. Такий підхід дозволяє реалізувати інтелектуальні функції без необхідності використання віддалених серверів, що забезпечує підвищений рівень конфіденційності даних та зменшує затримки при обробці інформації.

Інтеграція моделей машинного навчання (рисунок 1.9) у застосунок здійснюється шляхом використання спеціального формату файлів `mlmodel`, який містить структуру навченої моделі та її параметри. Як правило, навчання моделі виконується за допомогою зовнішніх інструментів або бібліотек машинного навчання, після чого модель конвертується у формат Core ML. Після додавання такого файлу до проєкту середовище розробки Xcode автоматично генерує відповідний програмний інтерфейс, що дозволяє використовувати модель у коді застосунку для виконання прогнозування.



Рисунок 1.9 – Схема інтеграції моделі машинного навчання у застосунок за допомогою Core ML

Спочатку формується або навчається модель машинного навчання, після чого вона конвертується у формат, сумісний із Core ML. Далі модель додається до проекту застосунку та використовується для виконання прогнозування на основі даних користувача. У результаті застосунок отримує можливість виконувати інтелектуальний аналіз даних у реальному часі та формувати прогнозні оцінки безпосередньо на мобільному пристрої.

### 1.3.2 Інструменти підготовки та конвертації моделей для мобільного середовища

Підготовка моделей машинного навчання для використання у мобільному середовищі є важливим етапом створення інтелектуальних застосунків. Зазвичай навчання моделей виконується у спеціалізованих середовищах або за допомогою бібліотек машинного навчання, таких як Python-бібліотеки для аналізу даних та побудови моделей. На цьому етапі формується навчальна вибірка, проводиться попередня обробка даних, вибір алгоритму та навчання моделі. Після завершення навчання модель повинна бути підготовлена для подальшого використання у мобільному застосунку.

Для інтеграції (рисунк 1.10) моделей машинного навчання у мобільні застосунки використовується процедура конвертації у формат, сумісний із платформою виконання. У випадку розробки застосунків для операційної системи iOS моделі конвертуються у формат `mlmodel`, який підтримується фреймворком Core ML.

Конвертація може здійснюватися за допомогою спеціальних інструментів, що дозволяють трансформувати моделі, створені у різних середовищах машинного навчання, у формат, придатний для виконання на мобільному пристрої. На першому етапі відбувається введення даних користувача або отримання їх із сенсорів пристрою.

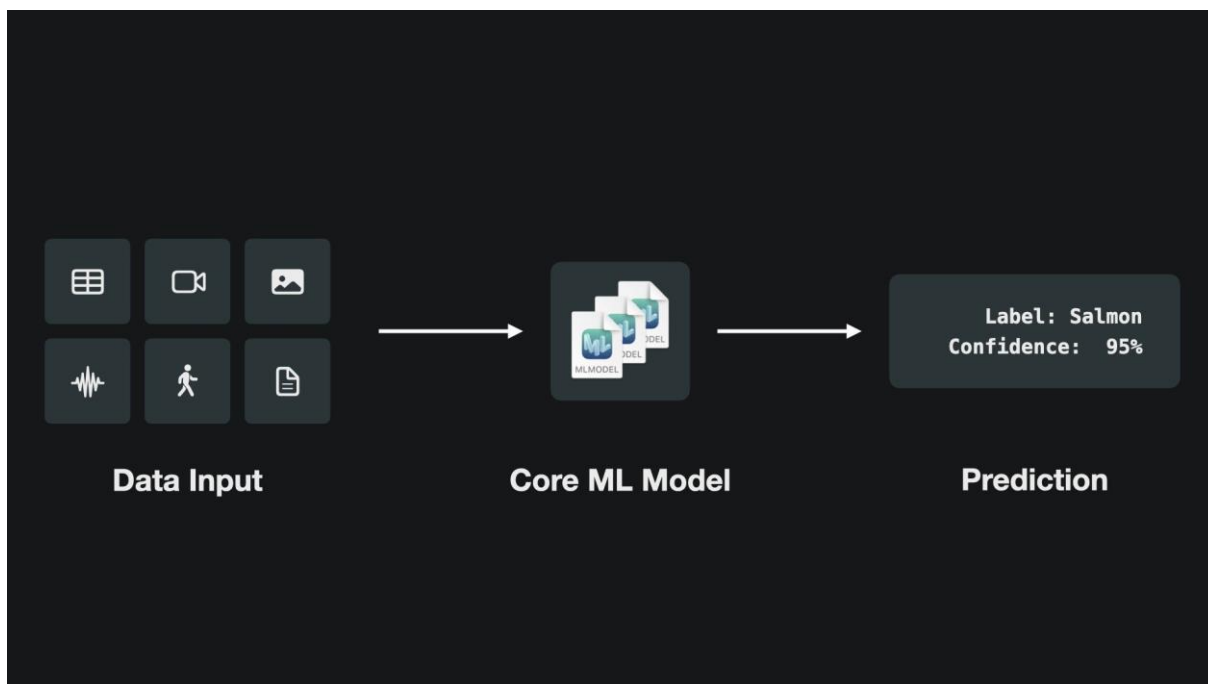


Рисунок 1.10 – Процес використання підготовленої моделі машинного навчання у мобільному застосунку

Використання інструментів підготовки та конвертації моделей дозволяє ефективно переносити результати навчання моделей машинного навчання у мобільне середовище. Це створює можливість реалізації інтелектуальних функцій безпосередньо на пристрої користувача та забезпечує ефективну роботу застосунків.

#### 1.4 Аналіз потреб та очікувань користувачів

У цьому підрозділі буде проведено аналіз потреб та очікувань користувачів щодо функціональності мобільних застосунків для моніторингу та прогнозування емоційного стану. Особлива увага приділяється визначенню ключових вимог до таких систем, зокрема зручності використання, можливості регулярного введення та аналізу даних та наочності візуалізації результатів.

#### 1.4.1 Визначення цільової аудиторії застосунків для моніторингу та аналізу настрою

Цільовою аудиторією застосунків для моніторингу та аналізу настрою є користувачі, які зацікавлені у спостереженні за власним емоційним станом та покращенні психологічного благополуччя. До цієї категорії можуть належати люди, що прагнуть краще розуміти власні емоційні реакції, аналізувати вплив повсякденних активностей на настрій та формувати більш здорові поведінкові звички. Особливо актуальними подібні системи є для студентів, працівників інтелектуальної сфери, а також людей, які ведуть активний спосіб життя та використовують цифрові технології для самоспостереження і управління особистими даними.

Крім того, потенційними користувачами таких застосунків можуть бути люди, які прагнуть відстежувати взаємозв'язок між своїм емоційним станом та щоденною активністю, зокрема режимом сну, фізичною активністю або використанням цифрових пристроїв. Мобільні застосунки дозволяють зручно фіксувати ці показники та формувати історію емоційного стану користувача.

#### 1.4.2 Огляд існуючих рішень для моніторингу та аналізу настрою

На сьогодні існує значна кількість мобільних застосунків, призначених для відстеження емоційного стану користувачів. Більшість таких систем реалізує функціональність ведення щоденника настрою, у якому користувач може фіксувати власні емоції, додавати короткі записи або відмічати настрій за допомогою шкали чи набору емоційних іконок. Також деякі застосунки пропонують інструменти візуалізації накопичених даних, що дозволяє переглядати динаміку настрою у вигляді графіків або календарних представлень.

Проте аналіз існуючих рішень показує, що більшість доступних застосунків обмежується лише функцією фіксації або візуалізації настрою. У таких системах відсутні механізми прогнозування емоційного стану користувача на основі поведінкових даних або показників щоденної активності. Таким чином, сучасні мобільні застосунки для моніторингу настрою переважно виконують роль інструментів ведення емоційного журналу, тоді як використання методів машинного навчання для автоматичного прогнозування настрою залишається недостатньо реалізованим напрямом.

#### 1.4.3 Потреби користувачів у функціональності аналізу та прогнозування настрою

Користувачі сучасних мобільних застосунків для моніторингу емоційного стану очікують не лише можливості фіксації настрою, але й отримання аналітичної інформації щодо змін емоційного стану у часі. Однією з важливих потреб є можливість автоматичного аналізу накопичених даних, що дозволяє виявляти закономірності між повсякденними активностями користувача та його настроєм. Зокрема, користувачі зацікавлені у визначенні факторів, які можуть позитивно або негативно впливати на їхній емоційний стан, таких як тривалість сну, фізична активність або рівень використання цифрових пристроїв.

Крім аналізу історичних даних, значний інтерес становить функціональність прогнозування настрою [20]. Можливість передбачення емоційного стану на основі поведінкових показників дозволяє користувачам отримувати додаткову інформацію про потенційні зміни настрою та краще планувати свою діяльність. Використання методів машинного навчання для прогнозування емоційного стану створює передумови для формування персоналізованих рекомендацій.

## 1.5 Визначення функціональних можливостей розроблюваного застосунку

У цьому підрозділі буде визначено основні функціональні можливості розроблюваного мобільного застосунку для моніторингу та прогнозування емоційного стану користувача. Зокрема, буде розглянуто перелік ключових функцій системи, пов'язаних із введенням та збереженням даних користувача, аналізом показників щоденної активності, використанням моделей машинного навчання для прогнозування настрою, а також візуалізацією результатів аналізу. На основі проведеного аналізу потреб користувачів буде сформовано структуру функціональних можливостей застосунку, що визначатиме основні компоненти та логіку його роботи.

### 1.5.1 Основні функції системи відстеження настрою

Основною функцією системи відстеження настрою є можливість регулярного введення користувачем інформації про власний емоційний стан та пов'язані з ним поведінкові показники. До таких даних можуть належати суб'єктивна оцінка настрою, тривалість сну, рівень фізичної активності, кількість кроків або час використання мобільного пристрою. Фіксація цих параметрів дозволяє формувати структуровану базу даних, яка відображає зміну емоційного стану користувача протягом певного періоду часу.

Крім введення та збереження даних, система повинна забезпечувати їх подальше опрацювання та організацію. До важливих функцій також належать можливість перегляду історії записів, редагування раніше введених даних та отримання узагальненої інформації про емоційний стан користувача. Сукупність таких функцій створює основу для формування персональної системи моніторингу настрою, що дозволяє користувачеві регулярно фіксувати та аналізувати власний емоційний стан.

### 1.5.2 Використання машинного навчання для прогнозування емоційного стану

Однією з ключових функціональних можливостей розроблюваного застосунку є використання методів машинного навчання для прогнозування емоційного стану користувача. На основі накопичених даних про поведінкові характеристики користувача формується навчальна вибірка, яка використовується для побудови моделі прогнозування настрою. Модель аналізує взаємозв'язок між такими параметрами, як фізична активність, режим сну, використання цифрових пристроїв та суб'єктивні оцінки настрою.

Після навчання модель може використовуватися для визначення ймовірного емоційного стану користувача на основі поточних показників його активності. Такий підхід дозволяє автоматизувати процес аналізу поведінкових даних та формувати прогнозні оцінки настрою. Використання машинного навчання створює можливість реалізації персоналізованих функцій у мобільному застосунку, що дозволяє більш точно враховувати індивідуальні особливості поведінки користувача.

### 1.5.3 Засоби аналітики та візуалізації змін настрою

Важливим компонентом системи моніторингу настрою є наявність інструментів аналітики та візуалізації накопичених даних. Використання графічних представлень інформації дозволяє користувачеві швидко оцінювати динаміку змін емоційного стану протягом певного періоду часу. Для цього можуть використовуватися різні типи візуалізації, зокрема лінійні графіки, календарні представлення або статистичні показники, що відображають середні значення настрою за певні часові інтервали.

Аналітичні функції застосунку дозволяють виявляти закономірності у зміні настрою та визначати фактори, що можуть впливати на емоційний стан

користувача. Наприклад, система може аналізувати взаємозв'язок між показниками фізичної активності або тривалістю сну та змінами настрою. Наявність таких інструментів аналітики сприяє підвищенню усвідомленості користувача щодо власного психологічного стану та створює умови для прийняття більш обґрунтованих рішень щодо способу життя.

## 1.6 Постановка задачі

У межах даної роботи ставиться задача дослідження методів персоналізованого прогнозування настрою користувача на основі поведінкових показників, що можуть бути отримані у мобільному середовищі. Основна увага приділяється формалізації взаємозв'язку між тривалістю сну, фізичною активністю, кількістю кроків, екранним часом та категорією настрою користувача, а також побудові математичної моделі визначення інтегрального показника настрою.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- визначити поведінкові показники, які можуть бути використані як вхідні ознаки для прогнозування настрою;
- розробити математичну модель визначення інтегрального показника настрою користувача на основі поведінкових факторів;
- сформулювати методику перетворення інтегрального показника настрою у дискретні класи емоційного стану;
- дослідити ефективність алгоритмів машинного навчання для класифікації настрою за сформованим набором ознак;
- виконати порівняльне оцінювання моделей за метриками accuracy, precision, recall, F1-score та результатами крос-валідації;
- визначити модель, що забезпечує найкращу якість прогнозування настрою користувача;
- виконати апробацію отриманої моделі шляхом її інтеграції у мобільне середовище iOS з використанням Core ML.

## 2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

### 2.1 Поведінкові показники у задачах прогнозування настрою

У задачах прогнозування настрою важливу роль відіграють поведінкові показники, які характеризують повсякденну активність користувача та можуть бути використані як вхідні параметри для моделей машинного навчання. Аналіз подібних параметрів дозволяє виявити закономірності між стилем життя людини та змінами її психологічного стану, що створює передумови для побудови систем автоматизованого прогнозування настрою. У сучасних дослідженнях подібні поведінкові індикатори часто використовуються як основні ознаки для навчання моделей класифікації або регресії, які здатні визначати ймовірний емоційний стан користувача на основі накопичених даних (рисунок 2.1).

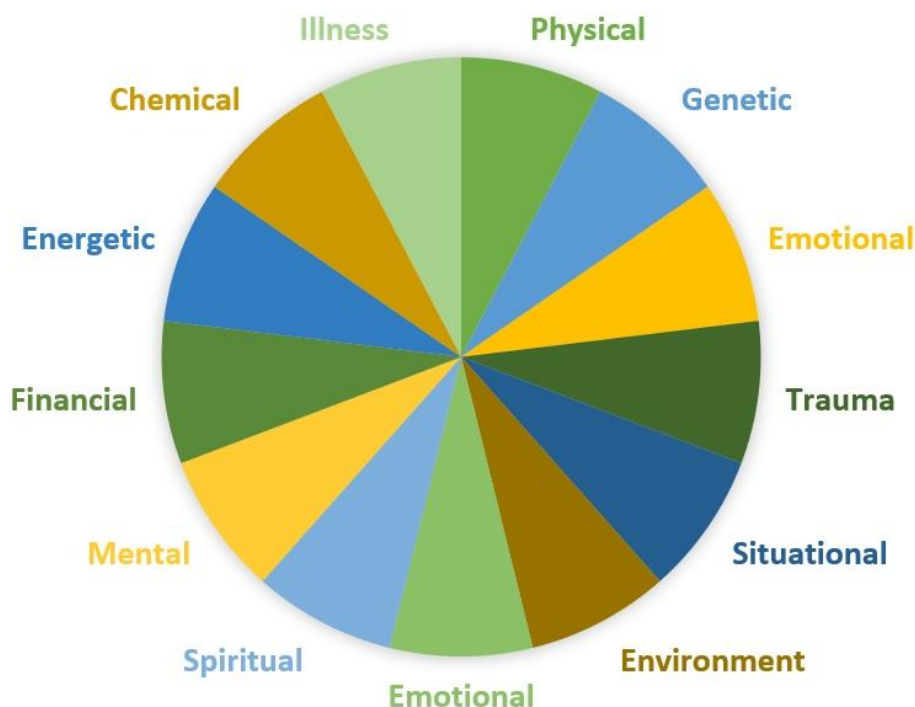


Рисунок 2.1 – Групи факторів, що впливають на емоційний стан людини

Емоційний стан людини формується під впливом великої кількості факторів, серед яких можна виділити фізичні, психологічні, соціальні та зовнішні умови середовища. Незважаючи на складність та багатофакторність цих процесів, у практичних системах прогнозування настрою зазвичай використовуються ті параметри, які можуть бути отримані автоматично за допомогою мобільних пристроїв або введені користувачем вручну. Такий підхід дозволяє значно спростити процес збору даних та забезпечити можливість побудови персоналізованих моделей прогнозування емоційного стану, що можуть працювати безпосередньо на мобільному пристрої користувача. У рамках розроблюваного застосунку для прогнозування настрою використовуються показники сну, фізичної активності та часу використання екрана, які збираються автоматично або вводяться користувачем, після чого передаються до моделі машинного навчання для формування прогнозу настрою.

### 2.1.1 Основні поведінкові фактори, що впливають на настрій

Поведінкові фактори [14] є одним із ключових джерел даних у задачах прогнозування настрою, оскільки вони безпосередньо відображають щоденну активність людини та її взаємодію з навколишнім середовищем. До таких факторів належать тривалість і якість сну, рівень фізичної активності, характер соціальної взаємодії, цифрова активність та структура щоденного розпорядку.

Кожен із цих показників може впливати на емоційний стан користувача та виступати як важлива ознака для алгоритмів машинного навчання. Використання подібних даних дозволяє будувати персоналізовані моделі прогнозування, які здатні виявляти індивідуальні закономірності між поведінкою користувача та його настроєм (рисунк 2.2).



Рисунок 2.2 – Основні поведінкові фактори, що впливають на емоційний стан користувача

Одним із найбільш значущих поведінкових факторів є сон, зокрема його тривалість та стабільність. Недостатній або нерегулярний сон може призводити до підвищеної втоми, зниження концентрації та погіршення емоційного стану. У дослідженнях у сфері цифрового здоров'я встановлено, що оптимальна тривалість сну сприяє стабілізації настрою та підвищенню психологічного благополуччя. Тому показники сну часто використовуються як одна з основних змінних у моделях прогнозування емоційного стану.

Фізична активність також є важливим індикатором емоційного благополуччя. Регулярні фізичні навантаження сприяють покращенню настрою завдяки біологічним механізмам, зокрема виділенню ендорфінів та зниженню рівня стресу. У контексті мобільних застосунків такі показники можуть визначатися автоматично за допомогою датчиків смартфона або інтеграції з фітнес-трекерами. Кількість кроків, інтенсивність активності або тривалість фізичних вправ можуть використовуватися як кількісні характеристики для подальшого аналізу та прогнозування настрою.

Окрему роль відіграє цифрова активність користувача, зокрема тривалість використання мобільних пристроїв і мобільних застосунків. Надмірний екранний час часто пов'язаний із підвищеним рівнем стресу, зниженням якості сну та погіршенням емоційного стану. З іншого боку, помірне використання цифрових технологій може мати нейтральний або навіть позитивний вплив, залежно від характеру діяльності користувача.

Не менш важливими є соціальна взаємодія та структура щоденного розпорядку. Спілкування з іншими людьми, участь у соціальних активностях та наявність стабільного розкладу дня можуть позитивно впливати на емоційний стан людини. У системах прогнозування настрою ці фактори можуть враховуватися через аналіз поведінкових патернів користувача, що дозволяє створювати більш точні та персоналізовані моделі прогнозування.

#### 2.1.2 Формалізація поведінкових даних для задач машинного навчання

У задачах прогнозування настрою поведінкові показники користувача повинні бути представлені у формалізованому вигляді, придатному для подальшої обробки алгоритмами машинного навчання. Формалізація передбачає перетворення щоденних спостережень та дій користувача у структуровані числові ознаки, які можуть використовуватися як вхідні параметри моделі. До таких ознак належать кількісні характеристики поведінки, зокрема тривалість сну у годинах, кількість зроблених кроків за добу, час використання мобільного пристрою та суб'єктивна оцінка настрою. Подібні параметри формують вектор ознак, який описує стан користувача у певний момент часу та використовується для навчання алгоритмів класифікації або регресії.

Для забезпечення коректної роботи моделей машинного навчання необхідно виконувати попередню обробку поведінкових даних. Така

обробка включає нормалізацію числових показників, усунення пропущених значень, а також уніфікацію форматів представлення інформації. У результаті формується структурований набір даних, у якому кожен запис відповідає окремому часовому інтервалу, наприклад одному дню, а кожна ознака характеризує певний аспект поведінки користувача. Подібне представлення даних дозволяє ефективно застосовувати алгоритми машинного навчання, зокрема дерева рішень, випадкові ліси або метод опорних векторів, для виявлення закономірностей між поведінковими показниками та емоційним станом.

### 2.1.3 Обмеження та проблеми використання поведінкових даних

Незважаючи на значний потенціал поведінкових даних у задачах прогнозування настрою, їх використання супроводжується рядом обмежень. Однією з основних проблем є неповнота або нерегулярність зібраної інформації. Користувач може не вводити дані щодня або пропускати окремі показники, що призводить до появи пропущених значень у наборі даних. Крім того, деякі параметри, зокрема суб'єктивна оцінка настрою, мають індивідуальний характер і можуть значно відрізнятися між різними користувачами, що ускладнює побудову універсальних моделей прогнозування.

Іншою важливою проблемою є обмеженість набору доступних поведінкових показників. Емоційний стан людини формується під впливом багатьох факторів, серед яких соціальні умови, події особистого життя, стан здоров'я та зовнішнє середовище.

Проте більшість мобільних застосунків використовує лише ті дані, які можуть бути легко отримані автоматично або введені користувачем вручну, наприклад показники сну, фізичної активності або часу використання пристрою.

#### 2.1.4 Математична модель визначення інтегрального показника настрою користувача

Одним із ключових етапів формалізації задачі прогнозування настрою є побудова математичної моделі, яка дозволяє об'єднати різноманітні поведінкові показники користувача в єдину числову оцінку.

У межах дослідження для цього використовується інтегральний показник настрою, що формується на основі сукупного впливу основних поведінкових факторів, зокрема тривалості сну, рівня фізичної активності, кількості кроків та часу використання мобільного пристрою. Такий підхід дозволяє перейти від багатовимірного опису поведінки користувача до узагальненої кількісної характеристики, яка може бути використана як основа для подальшої класифікації емоційного стану.

Основою запропонованої математичної моделі є поетапне формування інтегрального показника настрою на основі окремих поведінкових факторів користувача. На відміну від лінійного підсумовування ознак, у моделі використовується нелінійне перетворення кожного параметра, що дозволяє більш адекватно відобразити реальний характер їх впливу на емоційний стан. Для цього кожна вхідна змінна попередньо трансформується за допомогою відповідної математичної функції. Перетворення вихідних поведінкових показників здійснюється за допомогою наступних функціональних залежностей.

Для тривалості сну використовується параболічна функція, яка враховує наявність оптимального значення:

$$effect_{sleep} = -\frac{(sleep-7.5)^2}{20} + 0.5. \quad (2.1)$$

Максимальне значення досягається при  $sleep = 7.5$  годин, а відхилення в обидві сторони зменшує позитивний ефект, щоб сбалансувати фактор сну.

Для кількості кроків використовується логарифмічна залежність:

$$effect_{steps} = \frac{\log(1+steps)}{\log(2000)}. \quad (2.2)$$

Це дозволяє врахувати ефект насичення, при якому збільшення фізичної активності має зменшуваний приріст впливу.

Для тривалості фізичної активності застосовується коренева функція:

$$effect_{activity} = \frac{\sqrt{activity}}{\sqrt{120}}. \quad (2.3)$$

Вона відображає поступове зростання позитивного впливу без лінійного перевищення.

Для часу використання екрана використовується квадратична функція:

$$effect_{screen} = \left(\frac{screen}{10}\right)^2. \quad (2.4)$$

Це моделює посилення негативного впливу при збільшенні екраного часу.

Після обчислення часткових ефектів формується інтегральний показник настрою як зважена сума впливів окремих факторів:

$$score_{mood} = 0.35 \times effect_{sleep} + 0.30 \times effect_{steps} + 0.25 \times effect_{activity} + 0.30 \times effect_{screen}. \quad (2.5)$$

Вагові коефіцієнти у даній формулі відображають відносну важливість кожного з факторів у формуванні емоційного стану користувача. Позитивні коефіцієнти відповідають факторам, що покращують настрій,

тоді як від'ємний коефіцієнт для екранного часу відображає його негативний вплив. Така структура моделі дозволяє враховувати як позитивні, так і негативні аспекти поведінки користувача в єдиній системі оцінювання.

З метою підвищення реалістичності моделі до інтегрального показника додається випадкова складова, яка моделює вплив неконтрольованих факторів, що не враховані у моделі, таких як психологічний стан, соціальні події або індивідуальні особливості користувача. Це дозволяє уникнути надмірної детермінованості моделі та наблизити її до реальних умов функціонування систем прогнозування настрою.

Отримане значення інтегрального показника настрою використовується для подальшого визначення емоційного стану користувача шляхом переходу від неперервної величини до дискретних категорій.

Для цього застосовується квантильний підхід, який передбачає поділ усіх значень `mood_score` на три інтервали відповідно до їх розподілу:

- нижні 33 відсотки значень відповідають негативному емоційному стану та відносяться до класу «Sad»;
- значення у діапазоні від 33 до 66 відсотків інтерпретуються як нейтральний стан та відносяться до класу «Neutral»;
- верхні 34 відсотки значень, що перевищують 66-й перцентиль, відповідають позитивному настрою та класифікуються як «Happy».

Такий підхід дозволяє забезпечити приблизно рівномірний розподіл об'єктів між класами, що є важливим для коректного навчання моделей машинного навчання та уникнення проблеми незбалансованості даних. Таким чином, запропонована математична модель дозволяє формалізувати процес визначення настрою користувача.

## 2.2 Алгоритми машинного навчання для задач класифікації настрою

Алгоритми машинного навчання широко застосовуються для розв'язання задач класифікації емоційного стану користувачів на основі різноманітних поведінкових або сенсорних даних. У загальному випадку процес побудови моделі включає етапи попереднього аналізу даних, вибору ознак, навчання алгоритму та оцінювання його якості.

Для задач класифікації (рисунок 2.3) можуть використовуватися різні методи машинного навчання [14], зокрема логістична регресія, дерева рішень, випадковий ліс, метод опорних векторів, алгоритм k найближчих сусідів та наївний байєсівський класифікатор. Кожен із цих підходів має власні особливості, пов'язані з принципами побудови моделі, інтерпретацією результатів та вимогами до обсягу навчальних даних.

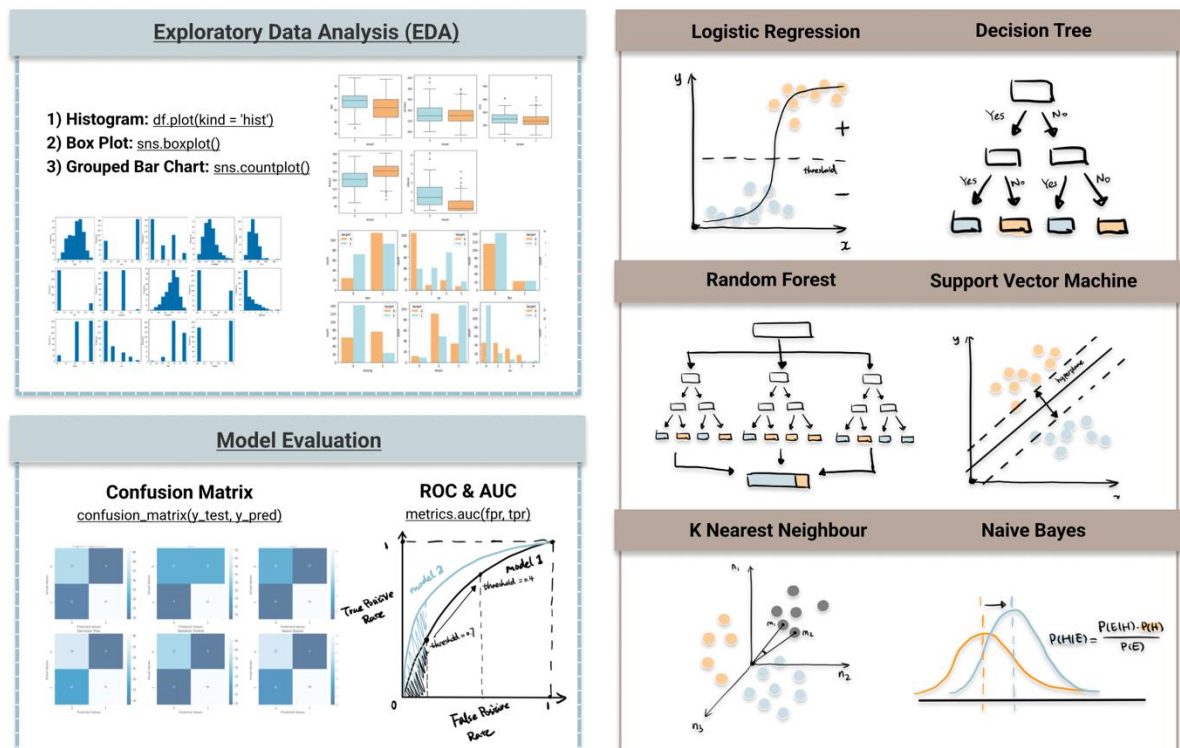


Рисунок 2.3 – Алгоритми машинного навчання та етапи аналізу даних у задачах класифікації

У задачах прогнозування настрою моделі машинного навчання використовують поведінкові показники користувача як вхідні ознаки для класифікації його емоційного стану (рисунк 2.4).

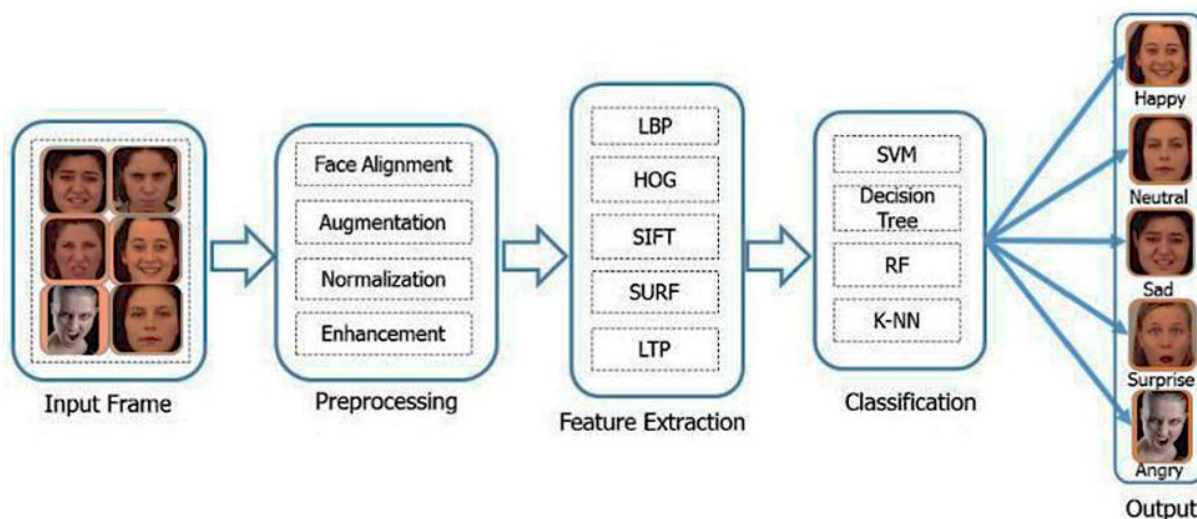


Рисунок 2.4 – Схема процесу класифікації емоційного стану з використанням алгоритмів машинного навчання

Спочатку відбувається збір та попередня обробка даних, що може включати нормалізацію значень, усунення шумів та формування вектора ознак.

Після цього здійснюється етап виділення або формування ознак, які найбільш повно характеризують поведінкові патерни користувача.

На основі отриманих ознак модель машинного навчання виконує класифікацію, визначаючи найбільш ймовірний емоційний стан серед заданих категорій, наприклад «позитивний», «нейтральний» або «негативний».

Такий підхід дозволяє автоматизувати процес аналізу поведінкових даних та формувати персоналізовані прогнози настрою користувача на основі накопиченої інформації.

### 2.2.1 Логістична регресія

Логістична регресія [14] є одним із базових алгоритмів машинного навчання, що широко застосовується для розв'язання задач бінарної та багатокласової класифікації. На відміну від лінійної регресії, яка використовується для прогнозування неперервних значень, логістична регресія дозволяє визначати ймовірність належності об'єкта до певного класу.

Як показано нижче (рисунок 2.5), у лінійній регресії результат моделі може виходити за межі інтервалу від 0 до 1, тоді як логістична регресія використовує сигмоїдну функцію, яка обмежує значення прогнозу в цьому діапазоні. Завдяки цьому модель може інтерпретувати результат як ймовірність належності до певного класу, що є особливо корисним у задачах класифікації емоційного стану користувача.

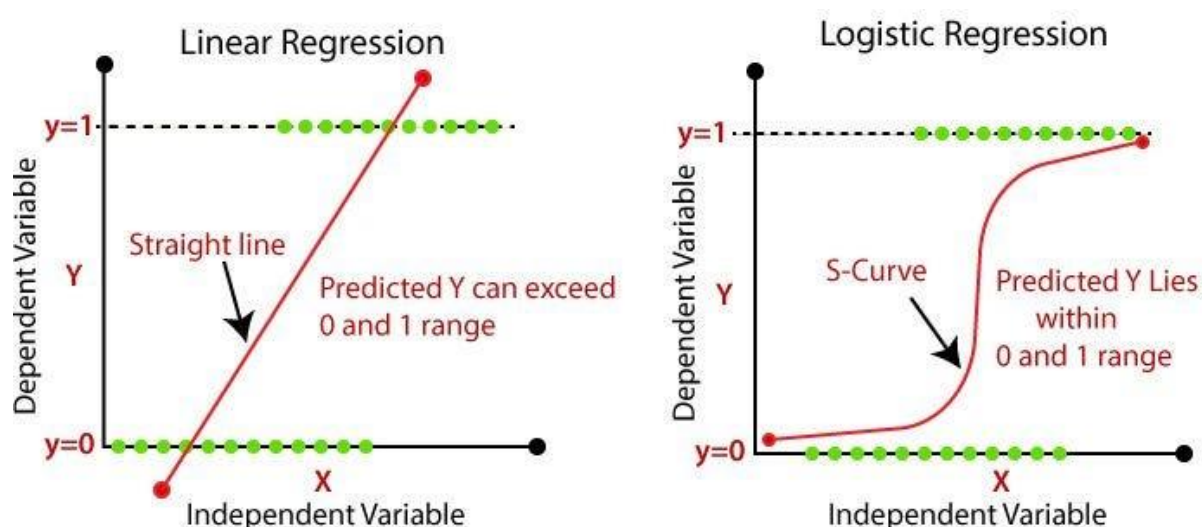


Рисунок 2.5 – Порівняння лінійної та логістичної регресії у задачах прогнозування

У задачах прогнозування настрою логістична регресія може використовуватися для визначення ймовірності того, що емоційний стан користувача належить до певної категорії. Модель отримує на вході набір

даних, після чого обчислює відповідну ймовірність для кожного класу. Кінцеве рішення про належність до певного класу приймається на основі встановленої межі класифікації, що розділяє простір ознак на відповідні області (рисунок 2.6).

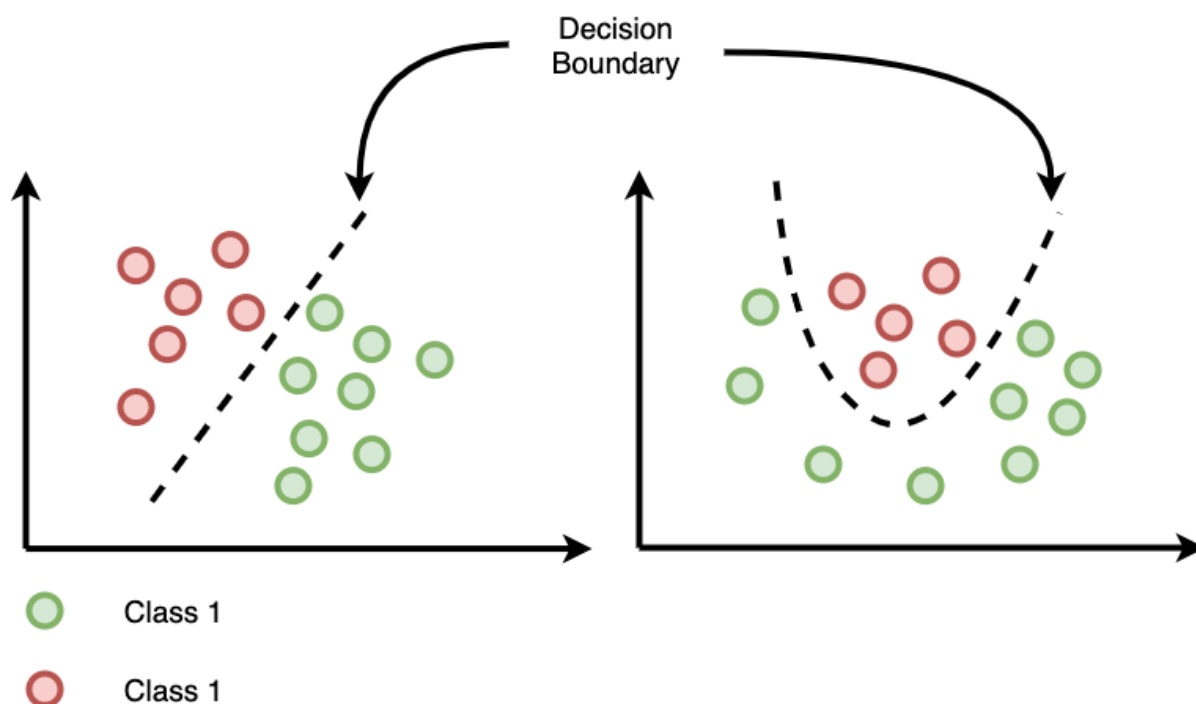


Рисунок 2.6 – Приклад формування межі прийняття рішення у задачі класифікації

Такий підхід дозволяє формувати інтерпретовані моделі прогнозування, які можуть бути використані для аналізу взаємозв'язку між поведінковими характеристиками користувача та його емоційним станом.

### 2.2.2 Метод k-найближчих сусідів

Метод k-найближчих сусідів [14] є одним із простих та інтуїтивно зрозумілих алгоритмів машинного навчання, який використовується для задач класифікації та регресії. Основна ідея цього підходу полягає у визначенні належності нового об'єкта до певного класу на основі схожості

з уже відомими прикладами навчальної вибірки. Алгоритм не виконує попереднього навчання у класичному розумінні, а використовує безпосередньо збережені дані для прийняття рішення. Для нового об'єкта обчислюється відстань до всіх інших об'єктів у просторі ознак, після чого визначається  $k$  найближчих сусідів. Клас нового об'єкта встановлюється на основі більшості серед знайдених сусідів, що дозволяє враховувати локальну структуру даних (рисунок 2.7).

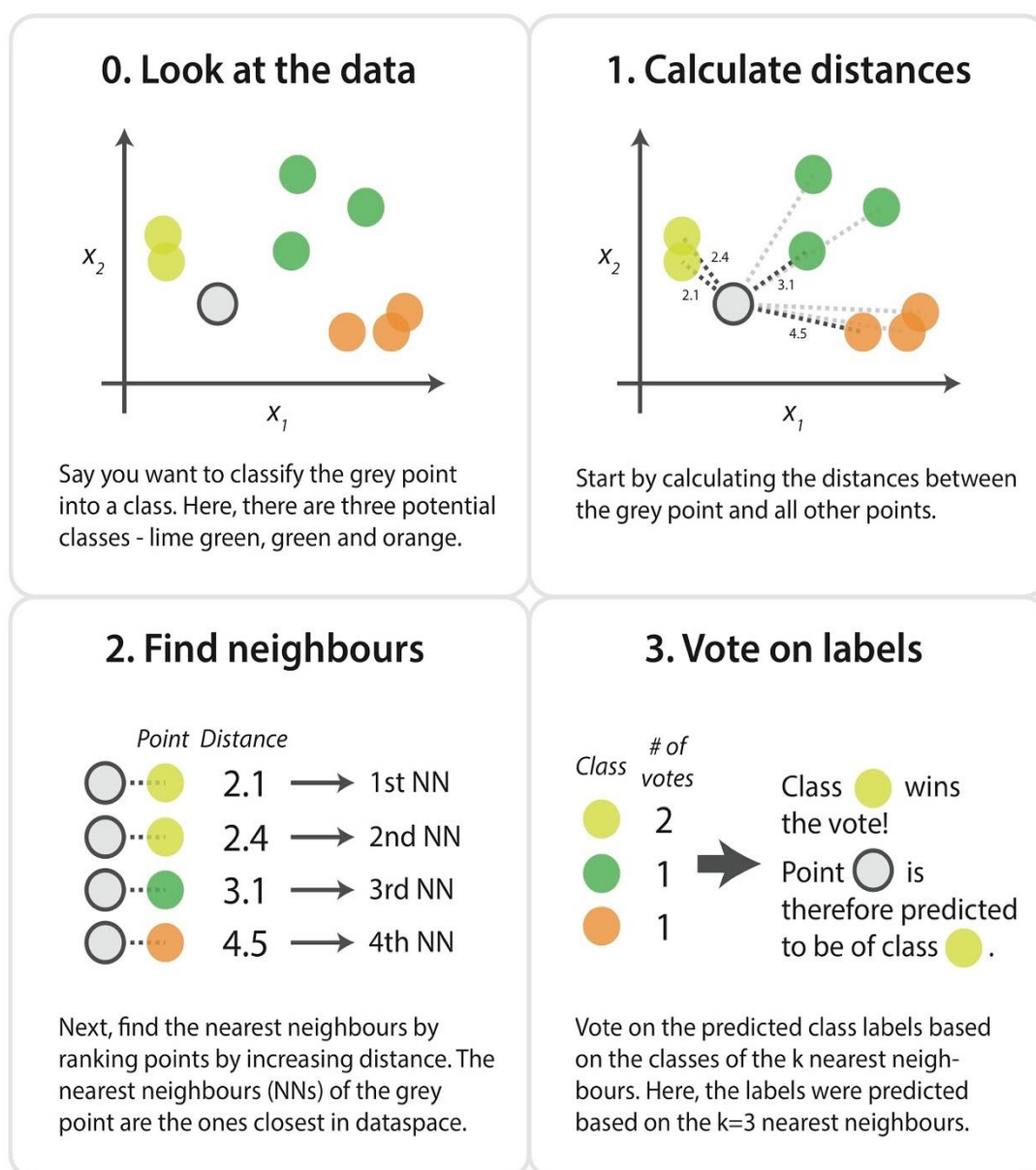


Рисунок 2.7 – Принцип роботи алгоритму  $k$ -найближчих сусідів у задачах класифікації

У задачах прогнозування настрою метод  $k$ -найближчих сусідів може використовуватися для визначення емоційного стану користувача на основі подібності його поведінкових показників до раніше зафіксованих даних. Наприклад, якщо новий запис містить певну комбінацію значень тривалості сну, кількості кроків та часу використання екрана, алгоритм знаходить у навчальній вибірці найбільш схожі записи. Далі виконується голосування між відповідними класами настрою, і новому запису присвоюється той клас, який зустрічається найчастіше серед найближчих сусідів.

### 2.2.3 Дерево рішень

Дерево рішень (рисунок 2.8) є одним із найбільш інтерпретованих алгоритмів машинного навчання [14], що застосовується для задач класифікації та регресії.

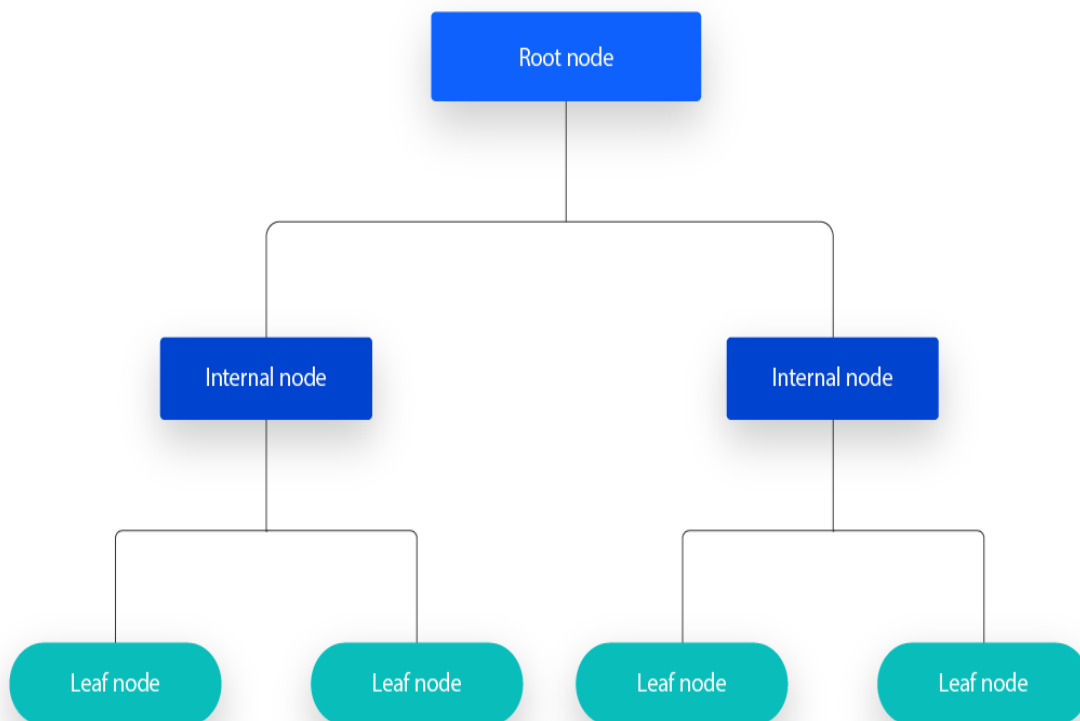


Рисунок 2.8 – Загальна структура моделі дерева рішень

Модель будується у вигляді ієрархічної структури вузлів, де кожен внутрішній вузол відповідає певній перевірці ознаки, гілки відповідають можливим результатам цієї перевірки, а листові вузли визначають кінцевий клас або прогнозоване значення. У процесі навчання алгоритм послідовно розбиває набір даних на підмножини, обираючи ті ознаки, які найбільш ефективно розділяють дані на класи. Для цього використовуються спеціальні критерії, зокрема інформаційний приріст або індекс Джині.

У задачах прогнозування настрою дерево рішень може використовуватися для визначення емоційного стану користувача на основі його поведінкових показників. Модель послідовно аналізує значення вхідних ознак, таких як тривалість сну, рівень фізичної активності або час використання мобільного пристрою, та переходить по відповідних гілках дерева до отримання кінцевого результату (рисунок 2.9).

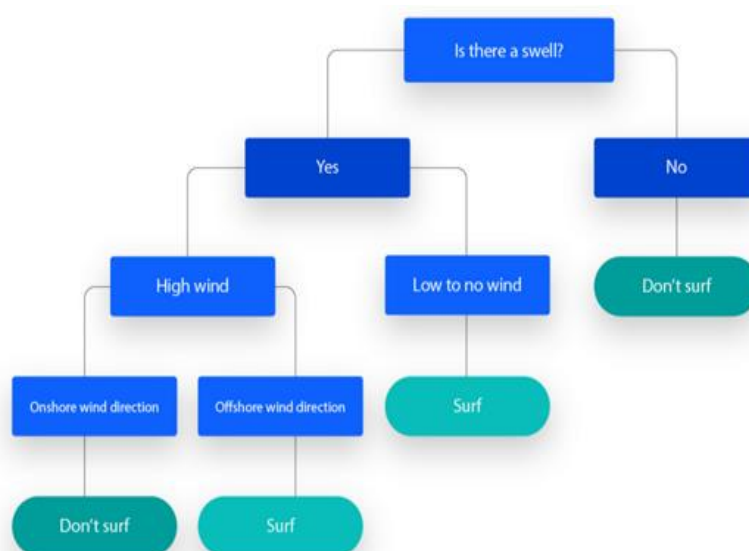


Рисунок 2.9 – Приклад процесу прийняття рішення у моделі дерева рішень

Подібний підхід дозволяє формувати зрозумілі правила класифікації, що можуть бути інтерпретовані людиною, а також забезпечує можливість аналізу впливу окремих поведінкових факторів на результат прогнозування.

### 2.3 Ансамблеві методи машинного навчання

Ансамблеві методи машинного навчання [14] представляють собою підхід, у якому для формування прогнозу використовується не одна модель, а сукупність декількох базових моделей. Основна ідея такого підходу полягає у поєднанні результатів кількох алгоритмів з метою підвищення точності та стабільності прогнозування.

Як показано нижче (рисунок 2.10), навчальний набір даних може розподілятися на декілька підвибірок, на основі яких навчаються окремі базові моделі. Кожна з цих моделей формує власний прогноз, після чого результати об'єднуються за допомогою певного механізму агрегації, наприклад голосування або усереднення. Такий підхід дозволяє зменшити вплив помилок окремих моделей та підвищити загальну якість класифікації.

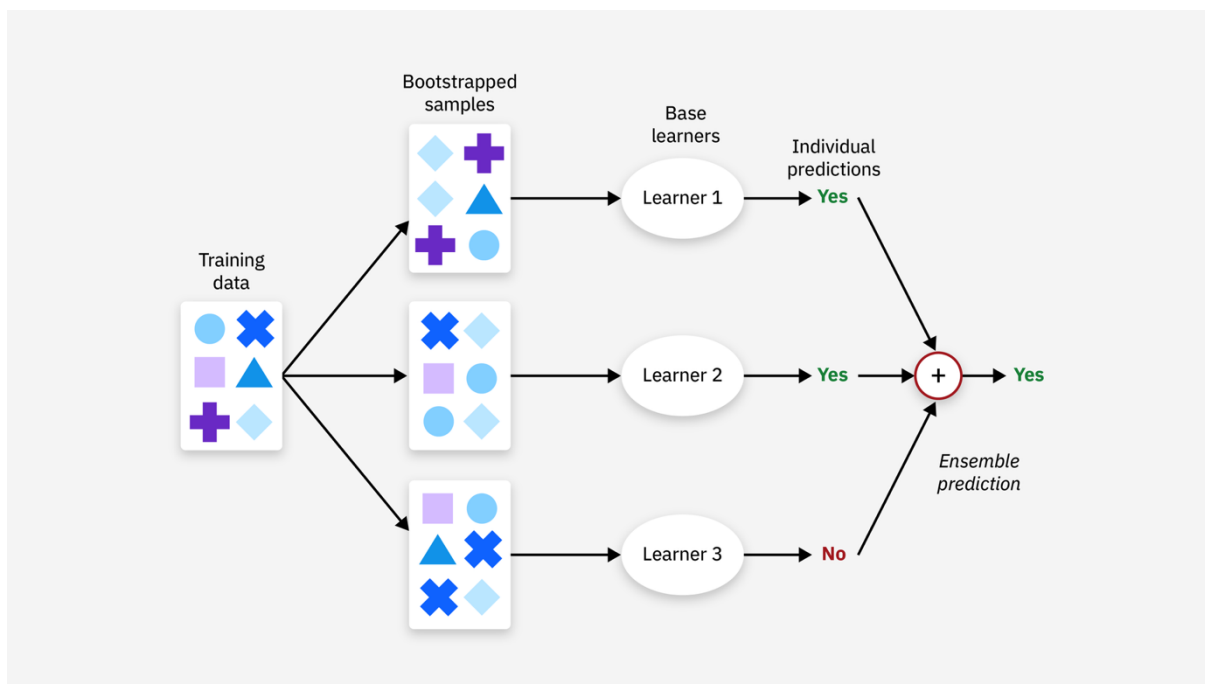


Рисунок 2.10 – Принцип роботи ансамблевих методів машинного навчання

У задачах прогнозування настрою ансамблеві методи можуть використовуватися для більш ефективного аналізу поведінкових даних

користувача. Оскільки емоційний стан людини залежить від багатьох факторів, використання декількох моделей дозволяє краще враховувати різні закономірності у даних. Одним із найпоширеніших ансамблевих алгоритмів є випадковий ліс, який поєднує множину дерев рішень та формує підсумковий прогноз на основі їх колективного голосування. Завдяки цьому зменшується ризик перенавчання моделі та підвищується її здатність до узагальнення на нових даних.

### 2.3.1 Random Forest

Алгоритм Random Forest (рисунок 2.11) є одним із найпоширеніших ансамблевих методів машинного навчання, який використовується для задач класифікації та регресії.

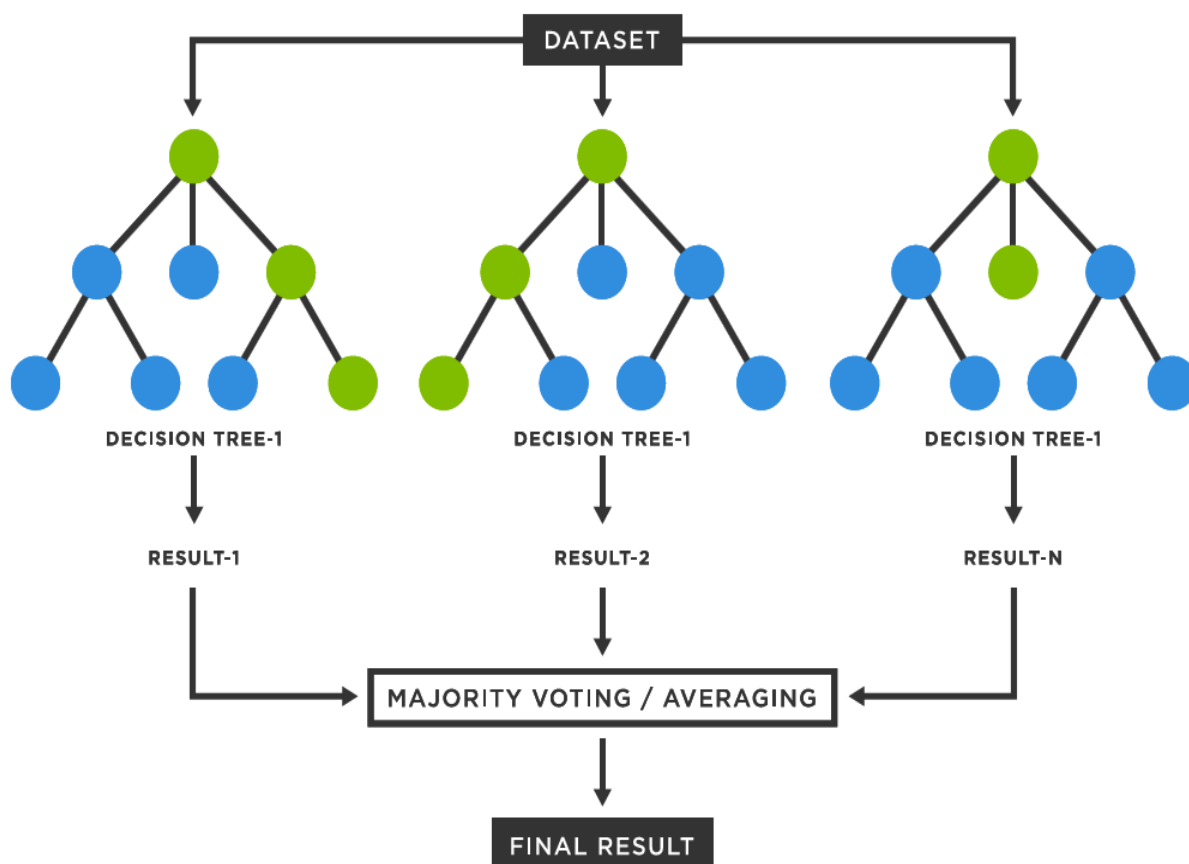


Рисунок 2.11 – Принцип роботи алгоритму Random Forest

Його основна ідея полягає у побудові великої кількості дерев рішень, кожне з яких навчається на випадковій підвибірці початкового набору даних. Кожне дерево формує власний результат класифікації, після чого всі отримані результати об'єднуються за допомогою механізму голосування або усереднення. Остаточне рішення визначається на основі більшості голосів окремих моделей, що дозволяє підвищити стабільність та точність прогнозування.

У задачах прогнозування настрою алгоритм Random Forest може ефективно використовуватися для аналізу поведінкових даних користувача. Модель отримує на вході набір ознак, таких як тривалість сну, рівень фізичної активності та час використання мобільного пристрою, після чого кожне дерево рішень формує власну оцінку емоційного стану. Завдяки використанню великої кількості дерев модель здатна виявляти складні нелінійні залежності між поведінковими показниками та настроєм користувача.

### 2.3.2 Gradient Boosting

Gradient Boosting (рисунок 2.12) є одним із ефективних ансамблевих методів машинного навчання, який базується на послідовному навчанні декількох моделей з метою поступового покращення точності прогнозування [15]. Основна ідея цього підходу полягає у тому, що кожна наступна модель навчається з урахуванням помилок, допущених попередніми моделями. Спочатку будується базова модель, яка формує початковий прогноз.

Далі аналізуються помилки цієї моделі, після чого наступна модель навчається таким чином, щоб зменшити ці помилки. Процес повторюється декілька разів, у результаті чого формується ансамбль моделей, які разом забезпечують більш точний результат.

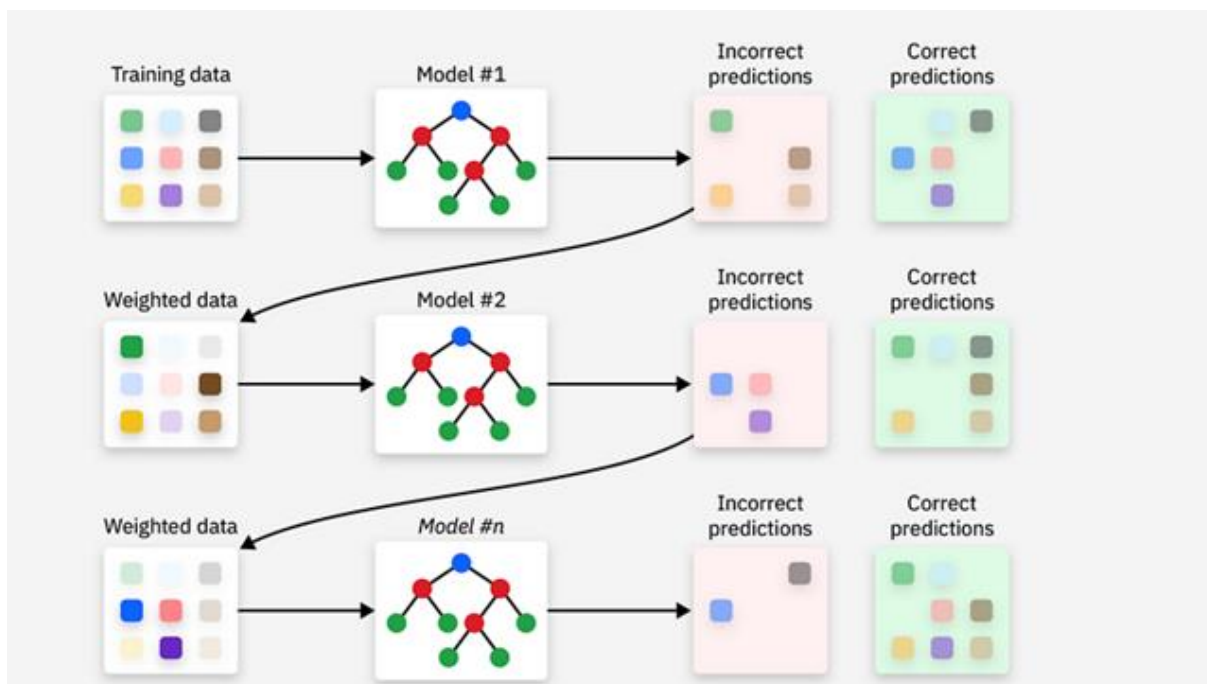


Рисунок 2.12 – Принцип роботи алгоритму Gradient Boosting

У задачах прогнозування настрою алгоритм Gradient Boosting може використовуватися для аналізу складних взаємозв'язків між поведінковими показниками користувача та його емоційним станом. Завдяки послідовному коригуванню помилок попередніх моделей алгоритм здатний ефективно враховувати навіть слабкі закономірності у даних.

Модель отримує на вході набір ознак, таких як тривалість сну, кількість кроків або час використання екрана, після чого формує прогноз емоційного стану на основі комбінованого результату кількох послідовно побудованих моделей. Подібний підхід дозволяє досягати високої точності класифікації, що є важливим для систем персоналізованого прогнозування настрою.

### 2.3.3 Порівняльні характеристики ансамблевих моделей

Ансамблеві методи машинного навчання широко застосовуються у задачах класифікації завдяки здатності поєднувати результати декількох

базових моделей та підвищувати загальну точність прогнозування. До найбільш поширених ансамблевих алгоритмів належать Random Forest та Gradient Boosting, які мають спільну ідею використання множини моделей, але відрізняються принципами побудови ансамблю. Random Forest базується на паралельному навчанні великої кількості дерев рішень на випадкових підвбірках даних, після чого результати окремих моделей об'єднуються за допомогою механізму голосування. Такий підхід дозволяє зменшити варіативність моделі та підвищити її стійкість до шумів у даних.

Алгоритм Gradient Boosting використовує інший підхід до формування ансамблю моделей. На відміну від Random Forest, де всі дерева навчаються незалежно, у Gradient Boosting моделі створюються послідовно, і кожна нова модель спрямована на виправлення помилок попередніх. Завдяки цьому алгоритм здатний поступово покращувати якість прогнозування та виявляти складні залежності у даних. Проте така стратегія навчання потребує більшого обчислювального ресурсу та може бути більш чутливою до параметрів налаштування моделі.

У задачах прогнозування настрою вибір ансамблевого алгоритму залежить від характеристик доступних даних та вимог до продуктивності системи. Random Forest зазвичай демонструє високу стабільність, простоту налаштування та стійкість до перенавчання, що робить його придатним для систем, які працюють із невеликими або середніми наборами поведінкових даних.

У свою чергу, Gradient Boosting може забезпечувати більш високу точність класифікації у випадках, коли дані мають складну структуру або містять нелінійні залежності між ознаками. Тому у практичних системах прогнозування настрою доцільно проводити експериментальне порівняння різних ансамблевих методів з метою визначення найбільш ефективного алгоритму для конкретного набору поведінкових показників.

## 2.4 Особливості підготовки даних для моделей машинного навчання

Підготовка даних [16] є одним із ключових етапів побудови моделей машинного навчання, оскільки якість вхідних даних безпосередньо впливає на точність і надійність отриманих прогнозів. У задачах прогнозування настрою дані формуються на основі щоденних поведінкових показників користувача, таких як тривалість сну, рівень фізичної активності, час використання мобільного пристрою та суб'єктивна оцінка емоційного стану. Оскільки такі дані можуть збиратися як автоматично з датчиків мобільного пристрою, так і вводитися користувачем вручну, вони часто мають різний формат, масштаб значень та рівень точності. Тому перед використанням у моделях машинного навчання необхідно виконати процедури попередньої обробки та структуризації інформації.

Однією з важливих задач підготовки даних є усунення пропущених або некоректних значень. У реальних умовах користувачі можуть нерегулярно вводити інформацію про свій настрій або окремі поведінкові показники, що призводить до появи неповних записів у наборі даних. Для вирішення цієї проблеми можуть використовуватися різні методи обробки пропущених значень, зокрема їх видалення, заміна середніми значеннями або використання спеціальних алгоритмів відновлення даних. Такі підходи дозволяють зберегти цілісність набору даних та забезпечити коректне функціонування алгоритмів машинного навчання.

Ще одним важливим етапом є нормалізація та масштабування ознак. Оскільки різні поведінкові показники можуть мати різні одиниці вимірювання, наприклад кількість кроків або тривалість сну у годинах, їх значення можуть суттєво відрізнятися за масштабом. Нормалізація дозволяє привести всі ознаки до єдиного діапазону значень, що сприяє більш стабільній роботі алгоритмів машинного навчання та покращує швидкість навчання моделей.

Крім того, важливу роль відіграє формування набору ознак, які найбільш повно відображають поведінкові патерни користувача. На цьому етапі визначається, які саме параметри будуть використовуватися як вхідні змінні моделі. У контексті систем прогнозування настрою такими ознаками можуть бути середня тривалість сну за певний період, кількість фізичної активності протягом дня або середній час використання мобільного пристрою. Правильний вибір ознак дозволяє підвищити точність моделі та забезпечити більш ефективне виявлення закономірностей між поведінковими показниками та емоційним станом користувача.

#### 2.4.1 Проблема незбалансованості класів

Однією з поширених проблем у задачах машинного навчання є незбалансованість класів у навчальному наборі даних. Така ситуація виникає тоді, коли кількість прикладів одного класу перевищує кількість прикладів іншого (рисунок 2.13). У результаті алгоритм може надавати перевагу більш представленому класу та формувати зміщені прогнози.

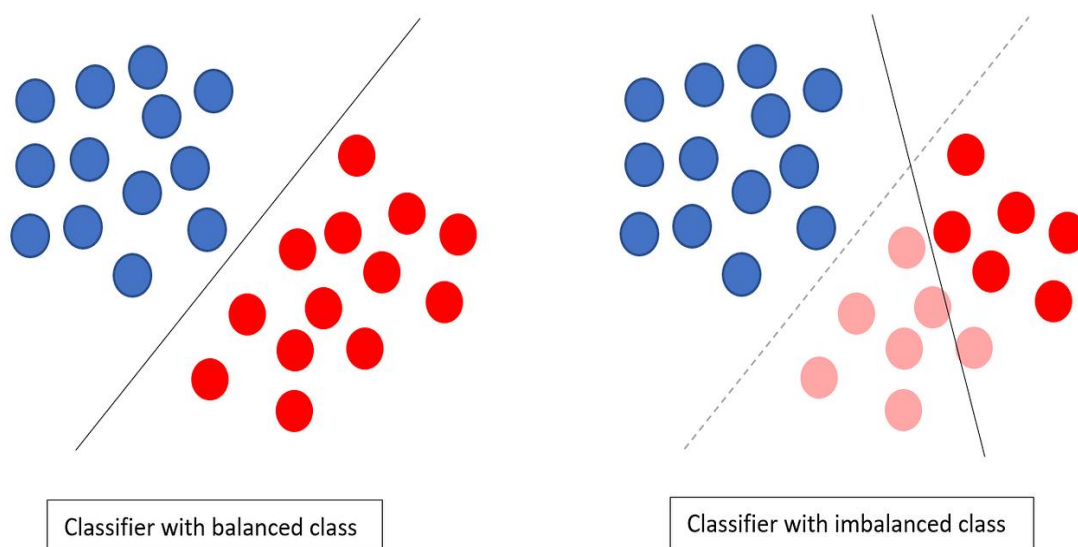


Рисунок 2.13 – Вплив незбалансованості класів на роботу класифікатора

У задачах прогнозування настрою подібна проблема може виникати, якщо певні емоційні стани фіксуються значно частіше за інші. Наприклад, користувач може частіше вводити записи про нейтральний або позитивний настрій, тоді як негативні стани можуть бути представлені меншою кількістю прикладів. Для вирішення цієї проблеми застосовуються спеціальні методи балансування даних (рисунок 2.14). До таких підходів належать збільшення кількості прикладів менш представленого класу тобто *over-sampling*, зменшення кількості прикладів домінуючого класу тобто *under-sampling*, або створення нових синтетичних прикладів на основі наявних даних.

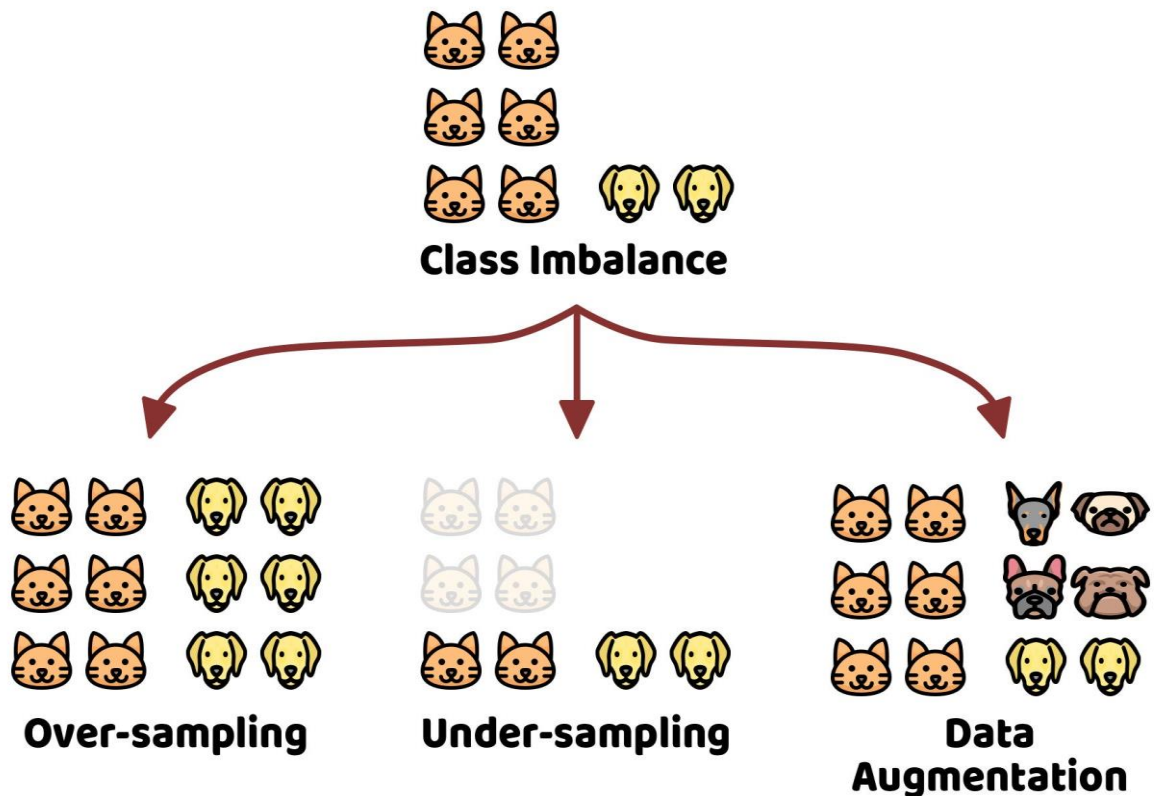


Рисунок 2.14 – Методи подолання проблеми незбалансованості класів

Використання подібних методів дозволяє покращити здатність моделі коректно розпізнавати всі класи та підвищити загальну якість класифікації у задачах прогнозування емоційного стану.

#### 2.4.2 Складність та реалістичність структури даних

У задачах прогнозування настрою структура даних характеризується значною складністю, оскільки емоційний стан людини формується під впливом великої кількості взаємопов'язаних факторів. Поведінкові показники, які використовуються як вхідні ознаки моделей машинного навчання, можуть мати нелінійні залежності, різні масштаби вимірювання та часову динаміку. Наприклад, однакова кількість фізичної активності може по-різному впливати на емоційний стан залежно від тривалості сну або рівня стресу користувача. У результаті модель повинна враховувати складні взаємозв'язки між різними ознаками для формування точного прогнозу.

Крім того, реальні поведінкові дані часто містять шум, випадкові коливання та індивідуальні особливості користувачів. Значення показників можуть змінюватися залежно від зовнішніх обставин, таких як робоче навантаження, соціальні події або зміни у повсякденному розкладі. Це ускладнює процес навчання моделей, оскільки алгоритм повинен відокремлювати закономірні залежності від випадкових змін у даних. Тому під час розробки систем прогнозування настрою важливо враховувати реалістичну структуру даних та використовувати алгоритми машинного навчання, здатні працювати з нелінійними та неоднорідними наборами даних.

#### 2.4.3 Методика порівняння моделей машинного навчання

Для визначення найбільш ефективного алгоритму у задачах прогнозування настрою необхідно виконувати порівняльний аналіз декількох моделей машинного навчання. Такий аналіз дозволяє оцінити, наскільки добре кожен алгоритм здатний виявляти закономірності у поведінкових даних та формувати коректні прогнози. У процесі

дослідження можуть розглядатися різні алгоритми класифікації, зокрема логістична регресія, метод k-найближчих сусідів, дерева рішень, Random Forest та Gradient Boosting. Кожна з цих моделей має власні переваги та обмеження, пов'язані зі складністю алгоритму, вимогами до обсягу даних та інтерпретованістю результатів.

Порівняння моделей зазвичай здійснюється на основі спеціальних метрик оцінювання якості класифікації. До найбільш поширених показників належать точність класифікації, повнота, точність передбачення та F-міра. Для підвищення достовірності результатів дослідження дані поділяються на навчальну та тестову вибірки, що дозволяє оцінити здатність моделі до узагальнення на нових даних. У результаті проведеного аналізу визначається алгоритм, який забезпечує найкраще співвідношення між точністю прогнозування, стабільністю роботи та обчислювальною ефективністю, що є важливим для подальшої інтеграції моделі у мобільний застосунок прогнозування настрою.

## 2.5 Технологія Core ML для інтеграції моделей машинного навчання

Core ML є спеціалізованим фреймворком компанії Apple, призначеним для інтеграції моделей машинного навчання безпосередньо у мобільні застосунки. Основною особливістю цієї технології є можливість виконання обчислень безпосередньо на пристрої користувача без використання зовнішніх серверів або хмарних сервісів. Такий підхід забезпечує високу швидкість обробки даних, зменшує затримки під час виконання прогнозів та підвищує рівень конфіденційності персональної інформації. Завдяки підтримці оптимізованого формату моделей та інтеграції з Swift, це дозволяє ефективно поєднувати методи машинного навчання з сучасними мобільними інтерфейсами та забезпечувати роботу інтелектуальних функцій безпосередньо на мобільному пристрої.

### 2.5.1 Архітектура Core ML

Архітектура Core ML побудована за багаторівневим принципом, що забезпечує ефективну інтеграцію моделей машинного навчання у мобільні застосунки.

Як показано нижче (рисунок 2.15), верхній рівень архітектури представлений прикладними застосунками, які використовують можливості машинного навчання для виконання інтелектуальних функцій. На цьому рівні розташовані також спеціалізовані фреймворки, зокрема Vision та SpeechKit, які дозволяють обробляти зображення, відео та мовні дані. Центральним компонентом архітектури є Core ML, що виступає основним середовищем виконання моделей машинного навчання та забезпечує взаємодію між прикладним програмним забезпеченням і обчислювальними ресурсами пристрою.

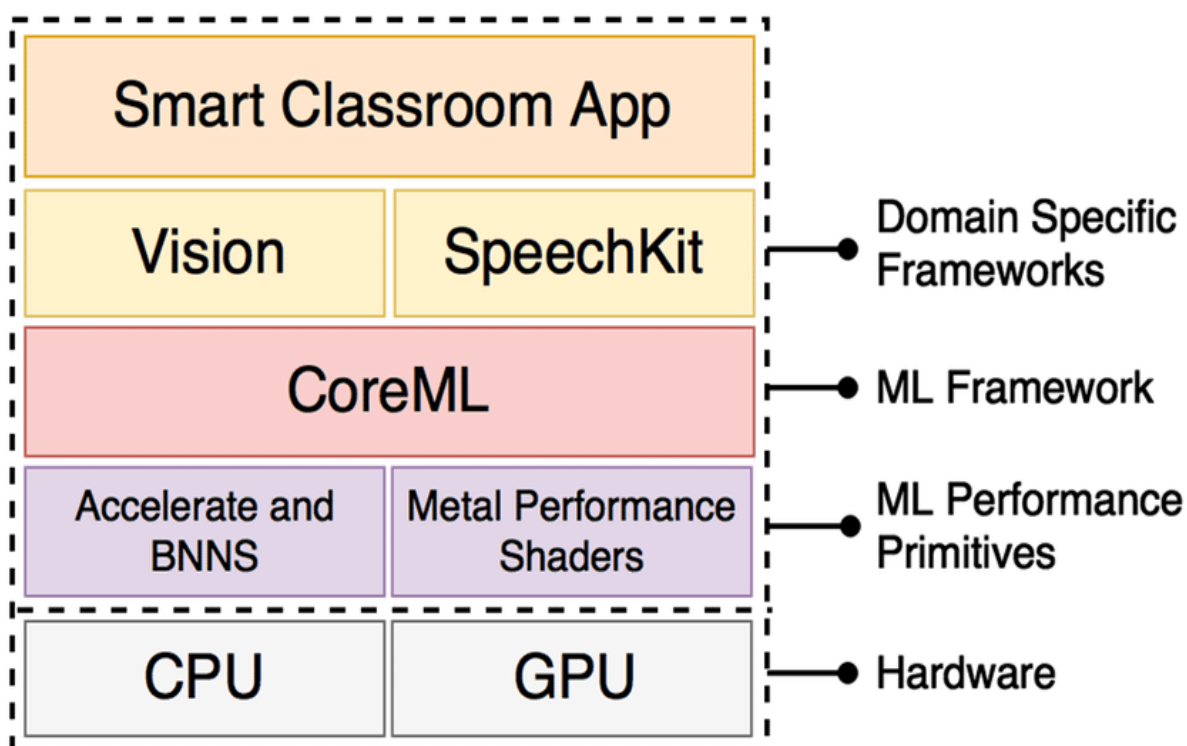


Рисунок 2.15 – Архітектура фреймворку Core ML та взаємодія його компонентів

Нижні рівні архітектури відповідають за оптимізацію обчислень та використання апаратних можливостей пристрою. Для прискорення виконання моделей використовуються спеціальні бібліотеки та обчислювальні механізми, зокрема Accelerate, Basic Neural Network Subroutines та Metal Performance Shaders. Ці компоненти забезпечують ефективну обробку математичних операцій, які виконуються під час роботи моделей машинного навчання. На найнижчому рівні архітектури знаходяться апаратні ресурси пристрою, зокрема центральний процесор та графічний процесор, що безпосередньо виконують обчислення. Така багаторівнева структура дозволяє Core ML ефективно використовувати апаратні можливості мобільних пристроїв та забезпечувати швидке виконання моделей машинного навчання у мобільних застосунках.

### 2.5.2 Конвертація моделей scikit-learn у формат Core ML

Однією з важливих можливостей технології Core ML є підтримка інтеграції моделей машинного навчання, які були попередньо навчені у зовнішніх середовищах розробки. У практичних системах моделі часто створюються за допомогою мов програмування Python та бібліотек машинного навчання, зокрема scikit-learn. Для цього використовується спеціальний інструментарій Core ML Tools, який дозволяє виконати конвертацію моделей з різних фреймворків машинного навчання у формат MLModel, що використовується у середовищі розробки Xcode.

Процес конвертації передбачає декілька етапів (рисунок 2.16). Спочатку модель навчається у середовищі Python з використанням бібліотек машинного навчання, після чого вона експортується за допомогою інструментів Core ML Tools. На цьому етапі виконується перетворення структури моделі, її параметрів та вагових коефіцієнтів у внутрішній формат Core ML.

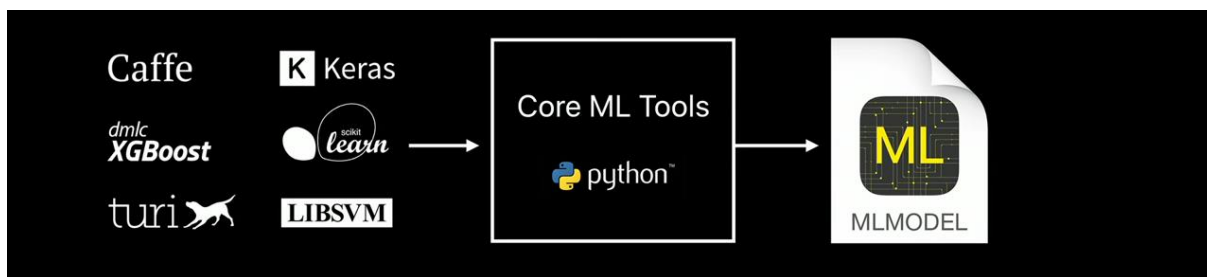


Рисунок 2.16 – Процес конвертації моделей машинного навчання у формат Core ML

У результаті формується файл з розширенням `mlmodel`, який може бути безпосередньо доданий до мобільного застосунку. Після імпорту моделі у проєкт Xcode середовище автоматично генерує програмний інтерфейс для взаємодії з моделлю, що дозволяє використовувати її для прогнозування на основі вхідних даних користувача безпосередньо у мобільному застосунку.

### 2.5.3 Переваги використання on-device машинного навчання в iOS

Використання on-device машинного навчання є важливим підходом у сучасних мобільних застосунках, оскільки дозволяє (рисунок 2.17) виконувати обробку даних та формування прогнозів безпосередньо на пристрої користувача.

Однією з ключових переваг такого підходу є забезпечення високого рівня конфіденційності, оскільки персональні дані користувача не передаються на зовнішні сервери. Усі обчислення виконуються локально, що зменшує ризики витоку інформації та підвищує рівень захисту особистих даних. Це є особливо важливим для систем, які працюють із персональними поведінковими показниками користувача, зокрема даними про фізичну активність, сон або емоційний стан.



Рисунок 2.17 – Основні переваги використання on-device машинного навчання у середовищі iOS

Ще однією важливою перевагою є висока швидкість виконання прогнозів та можливість роботи системи у режимі реального часу. Оскільки обчислення виконуються безпосередньо на мобільному пристрої, відсутня необхідність у передачі даних через мережу та очікуванні відповіді від віддаленого сервера. Це дозволяє зменшити затримки та забезпечити миттєве отримання результатів. Крім того, технологія on-device машинного навчання забезпечує глибоку інтеграцію з екосистемою Apple, що дозволяє ефективно використовувати апаратні ресурси пристрою та оптимізовані фреймворки операційної системи. Завдяки цьому мобільні застосунки можуть реалізовувати інтелектуальні функції аналізу даних і прогнозування, зберігаючи при цьому високу продуктивність та енергоефективність.

### 3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

#### 3.1 План експериментів

Експериментальне дослідження спрямоване на оцінювання ефективності різних алгоритмів машинного навчання для задачі прогнозування емоційного стану користувача на основі поведінкових показників. Основною метою експерименту є визначення моделі, яка демонструє найкращу узагальнювальну здатність під час класифікації настрою за набором кількісних ознак, що характеризують повсякденну активність користувача. Отримана модель у подальшому використовується для інтеграції у мобільний застосунок iOS з використанням технології Core ML, що дозволяє виконувати прогнозування безпосередньо на пристрої користувача.

У рамках дослідження розглядаються кілька алгоритмів машинного навчання, що підтримуються бібліотекою `scikit-learn` [7] та можуть бути конвертовані у формат Core ML [3]. До переліку досліджуваних моделей належать Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest та Gradient Boosting [14]. Використання декількох моделей дозволяє провести порівняльний аналіз їх ефективності та визначити алгоритм, який найкраще підходить для задачі прогнозування настрою з урахуванням обмежень мобільних платформ.

#### 3.2 Опис використаного набору даних

Для проведення експериментального дослідження було сформовано набір даних, що моделює поведінкові характеристики користувача мобільного застосунку. Використання синтетичного датасету дозволяє відтворити реалістичну структуру даних без використання персональної інформації, що є важливим з точки зору конфіденційності. У рамках

експерименту кожен запис набору даних представляє один умовний день активності користувача та містить набір числових показників, які можуть впливати на його емоційний стан.

Генерація набору даних реалізована програмним способом з використанням бібліотеки NumPy. Відповідний програмний код наведено у лістингу 3.1. У процесі генерації для кожного запису випадковим чином формуються значення поведінкових параметрів, що описують спосіб життя користувача. До таких параметрів належать тривалість сну, кількість пройдених кроків, час використання екрана мобільного пристрою та тривалість фізичної активності. Для формування реалістичного розподілу значень використовується нормальний розподіл із подальшим обмеженням значень у допустимих межах.

Лістинг 3.1 – Програмна реалізація генерації синтетичного набору даних поведінкових показників

```
def generate_dataset(samples=5000):
    np.random.seed(RANDOM_STATE)
    data = []

    for i in range(samples):
        sleep = np.random.normal(7.5, 1.2)
        sleep = np.clip(sleep, 4, 10)

        steps = np.random.normal(8000, 2500)
        steps = np.clip(steps, 1000, 20000)

        screen = np.random.normal(3.5, 1.5)
        screen = np.clip(screen, 1, 10)

        activity = np.random.normal(45, 20)
        activity = np.clip(activity, 0, 120)

        sleep_effect = -((sleep - 7.5) ** 2) / 20 + 0.5
```

### Продовження лістингу 3.1

```

steps_effect = np.log1p(steps) / np.log(20000)
activity_effect = np.sqrt(activity) / np.sqrt(120)
screen_effect = (screen / 10) ** 2

mood_score = (
    0.35 * sleep_effect +
    0.30 * steps_effect +
    0.25 * activity_effect -
    0.30 * screen_effect
)

mood_score += np.random.normal(0, 0.02)

data.append({
    "sleep_hours": sleep,
    "steps": steps,
    "screen_time": screen,
    "activity_minutes": activity,
    "mood_score": mood_score
})

df = pd.DataFrame(data)
return df

```

Окрім формування значень поведінкових показників, у процесі генерації датасету також обчислюється інтегральний показник настрою користувача – `mood_score`. Цей показник формується на основі зваженої комбінації впливу кожного з факторів. Зокрема, позитивний вплив на емоційний стан мають достатня тривалість сну, фізична активність та кількість пройдених кроків, тоді як надмірний час використання екрана може мати негативний вплив. Такий підхід дозволяє сформувати узагальнену числову оцінку настрою, яка враховує взаємодію кількох поведінкових факторів.

Оскільки задача дослідження полягає у класифікації настрою, наступним кроком є перетворення безперервного показника `mood_score` у дискретні класи емоційного стану. Для цього використовується квантильний поділ значень інтегрального показника на три категорії. Програмна реалізація цього етапу наведена у лістингу 3.2. Нижній квантиль відповідає негативному настрою, середній діапазон – нейтральному емоційному стану, а верхній квантиль – позитивному настрою користувача.

Лістинг 3.2 – Формування цільової змінної настрою на основі квантильного поділу показника `mood_score`

```
q1 = df["mood_score"].quantile(0.33)
q2 = df["mood_score"].quantile(0.66)

df["mood"] = np.where(
    df["mood_score"] > q2,
    "Happy",
    np.where(df["mood_score"] > q1, "Neutral", "Sad")
)

df = df.drop(columns=["mood_score"])
return df
```

У результаті виконання описаної процедури формується фінальний набір даних, що містить чотири числові ознаки – `sleep_hours`, `steps`, `screen_time` та `activity_minutes` – а також цільову змінну `mood`, яка визначає категорію емоційного стану користувача.

Така структура датасету є придатною для подальшого використання у задачах машинного навчання та дозволяє застосовувати алгоритми класифікації для прогнозування настрою на основі поведінкових характеристик користувача.

### 3.3 Реалізація експериментального середовища

Експериментальне середовище дослідження було реалізовано у вигляді програмного сценарію на мові програмування Python. Такий підхід дозволяє забезпечити гнучкість під час проведення експериментів, швидке тестування різних алгоритмів машинного навчання та відтворюваність результатів. Усі етапи дослідження – від генерації набору даних до оцінювання ефективності моделей – реалізовані у вигляді єдиного програмного pipeline, що виконується послідовно в межах одного експериментального сценарію.

#### 3.3.1 Використані програмні бібліотеки та інструменти

Для реалізації експериментального дослідження було використано мову програмування Python та набір бібліотек для аналізу даних і машинного навчання. Основними інструментами роботи з даними виступають бібліотеки pandas та NumPy.

Побудова та навчання моделей машинного навчання здійснюється за допомогою бібліотеки scikit-learn. У межах експерименту використовуються алгоритми Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest та Gradient Boosting. Також бібліотека надає інструменти для поділу вибірки на тренувальну і тестову частини, нормалізації ознак за допомогою StandardScaler.

Для візуалізації результатів експерименту застосовуються бібліотеки matplotlib та seaborn, які дозволяють будувати графіки, матриці помилок та порівняльні діаграми моделей. Додатково використовується бібліотека coremltools, що забезпечує конвертацію навченої моделі у формат Core ML для подальшого використання у мобільному застосунку. Імпорт програмних бібліотек та налаштування експерименту наведено у лістингу 3.3.

### Лістинг 3.3 – Імпорт програмних бібліотек та налаштування базових параметрів експерименту

```
import pandas as pd
import numpy as np
import coremltools as ct
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.ensemble import GradientBoostingClassifier,
RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve
from sklearn.metrics import auc
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import label_binarize
RANDOM_STATE = 42
```

Отримана модель може бути використана безпосередньо у мобільному застосунку iOS. Використання наведеного набору бібліотек дозволяє реалізувати повний цикл експериментального дослідження – від генерації набору даних і навчання моделей машинного навчання до оцінювання їх ефективності та підготовки моделі до інтеграції у мобільний застосунок.

### 3.3.2 Попередня обробка та підготовка даних до навчання

Після формування набору даних виконується підготовка даних до використання у моделях машинного навчання. На цьому етапі з датасету виділяються ознаки, що використовуються для навчання моделей, а також цільова змінна класифікації. У дослідженні як ознаки використовуються поведінкові показники користувача: тривалість сну, кількість кроків, час використання екрана та тривалість фізичної активності. Цільовою змінною виступає категорія настрою користувача, що була сформована на попередньому етапі.

Виділення ознак та цільової змінної для подальшого навчання моделей реалізовано програмним способом, як показано у лістингу 3.4. Такий підхід дозволяє сформувати матрицю ознак  $X$  та вектор цільових значень  $y$ , які використовуються алгоритмами класифікації.

Лістинг 3.4 – Виділення ознак і цільової змінної для навчання моделей класифікації

```
x = df[
    [
        "sleep_hours",
        "steps",
        "screen_time",
        "activity_minutes"
    ]
]
y = df["mood"]
```

Наступним кроком є поділ набору даних на тренувальну та тестову вибірки. Тренувальна вибірка використовується для навчання моделей машинного навчання, тоді як тестова вибірка призначена для оцінювання їх здатності прогнозувати результати на нових даних.

Програмна реалізація поділу даних на тренувальну та тестову вибірки наведена у лістингу 3.5. Додатково на цьому етапі створюється об'єкт `StandardScaler`, який у подальшому використовується для нормалізації значень ознак перед навчанням моделей.

Лістинг 3.5 – Поділ набору даних на тренувальну і тестову вибірки зі стратифікацією класів

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=RANDOM_STATE,
    stratify=y
)
scaler = StandardScaler()
```

### 3.3.3 Побудова моделей машинного навчання

Після підготовки даних наступним етапом експериментального дослідження є побудова моделей машинного навчання. У межах роботи розглядається кілька алгоритмів класифікації, що підтримуються бібліотекою `scikit-learn` та можуть бути конвертовані у формат Core ML. Використання декількох моделей дозволяє провести порівняльний аналіз їх ефективності та визначити алгоритм, який демонструє найкращі результати прогнозування настрою користувача.

Для проведення дослідження було сформовано набір моделей машинного навчання, до якого входять `Logistic Regression`, `K-Nearest Neighbors`, `Decision Tree`, `Random Forest` та `Gradient Boosting`. Ініціалізація цих моделей із відповідними параметрами наведена у лістингу 3.6.

Лістинг 3.6 – Ініціалізація набору моделей машинного навчання для порівняльного дослідження

```
models = {  
    "LogisticRegression": LogisticRegression(  
        max_iter=2000,  
        class_weight="balanced",  
        random_state=RANDOM_STATE  
    ),  
    "KNN": KNeighborsClassifier(  
        n_neighbors=9,  
        weights="distance"  
    ),  
    "RandomForest": RandomForestClassifier(  
        n_estimators=1100,  
        max_depth=30,  
        min_samples_leaf=4,  
        min_samples_split=6,  
        max_features="sqrt",  
        bootstrap=True,  
        class_weight="balanced",  
        n_jobs=-1,  
        random_state=RANDOM_STATE  
    ),  
    "DecisionTree": DecisionTreeClassifier(  
        max_depth=10,  
        min_samples_split=4,  
        random_state=RANDOM_STATE  
    ),  
    "GradientBoosting": GradientBoostingClassifier(  
        n_estimators=300,  
        learning_rate=0.05,  
        max_depth=4,  
        subsample=0.8,  
        random_state=RANDOM_STATE ) }
```

Для навчання моделей використовується механізм конвеєрів обробки даних, який об'єднує нормалізацію ознак та навчання моделі в одну послідовність операцій. Це дозволяє автоматизувати процес підготовки даних та забезпечити коректне застосування нормалізації як під час навчання, так і під час прогнозування. Побудова конвеєра обробки даних та запуск процесу навчання моделей наведені у лістингу 3.7.

Лістинг 3.7 – Побудова конвеєра обробки даних і навчання моделі

```
for name, model in models.items():
    pipeline = Pipeline([
        ("scaler", scaler),
        ("model", model)
    ])
    pipeline.fit(X_train, y_train)
    pred = pipeline.predict(X_test)
```

### 3.3.4 Крос-валідація моделей та оцінювання їх узагальнювальної здатності

Після навчання моделей виконується оцінювання їх ефективності на тестовій вибірці. Для цього використовуються стандартні метрики класифікації, які дозволяють визначити якість прогнозування моделей з різних точок зору. Зокрема, обчислюються показники *accuracy*, *precision*, *recall* та *F1-score*.

Окрім оцінювання на тестовій вибірці, у дослідженні використовується метод крос-валідації. Цей підхід дозволяє більш об'єктивно оцінити здатність моделі узагальнювати знання на нових даних шляхом багаторазового навчання та тестування на різних підмножинах датасету. Реалізація обчислення метрик класифікації та виконання крос-валідації наведена у лістингу 3.8.

### Лістинг 3.8 – Обчислення метрик якості та виконання крос-валідації моделі

```
accuracy = accuracy_score(y_test, pred)
precision = precision_score(
    y_test,
    pred,
    average="weighted"
)
recall = recall_score(
    y_test,
    pred,
    average="weighted"
)
f1 = f1_score(
    y_test,
    pred,
    average="weighted"
)
cv_scores = cross_val_score(
    pipeline,
    X,
    y,
    cv=5,
    scoring="accuracy"
)
```

Отримані значення метрик використовуються для формування підсумкової структури результатів експерименту. Для кожної моделі зберігаються показники точності, повноти, F1-міри та середнє значення точності крос-валідації. Додатково здійснюється визначення моделі, яка демонструє найкращу якість класифікації за показником accuracy. Програмна реалізація збереження результатів оцінювання моделей наведена у лістингу 3.9.

Лістинг 3.9 – Збереження результатів оцінювання моделей у підсумкову структуру даних

```
results.append({
    "model": name,
    "accuracy": accuracy,
    "precision": precision,
    "recall": recall,
    "f1": f1,
    "cv_accuracy": cv_scores.mean(),
    "pipeline": pipeline
})

if accuracy > best_score:
    best_score = accuracy
    best_model = pipeline
    best_name = name
```

### 3.4 Результати дослідження

У цьому підрозділі розглядаються результати експериментального дослідження ефективності різних алгоритмів машинного навчання для задачі прогнозування настрою користувача на основі поведінкових показників.

Окрім числових показників ефективності, у підрозділі також аналізуються результати крос-валідації, що дозволяє оцінити стабільність роботи моделей на різних підмножинах даних. Додатково виконуються візуалізація результатів класифікації та порівняльний аналіз досліджуваних алгоритмів.

На основі отриманих результатів визначається модель машинного навчання, яка демонструє найкращі показники прогнозування та може бути використана для подальшої інтеграції у мобільний застосунок.

### 3.4.1 Результати класифікації для моделі Logistic Regression

Модель Logistic Regression є однією з базових моделей машинного навчання, що широко використовується для задач класифікації. У межах проведеного експерименту дана модель була навчена на сформованому наборі поведінкових показників користувача. Основні результати класифікації (рисунок 3.1), де представлено значення основних метрик оцінювання якості моделі. Отримана точність класифікації становить 0.672, а значення метрик precision, recall та F1-score знаходяться на близькому рівні, що свідчить про відносно збалансовану роботу моделі для різних класів настрою.

```

=====
Model: LogisticRegression
Accuracy: 0.672
Precision: 0.6705
Recall: 0.672
F1-score: 0.6712
Cross-val accuracy: 0.6681999999999999

```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Happy        | 0.73      | 0.74   | 0.74     | 1360    |
| Neutral      | 0.54      | 0.53   | 0.54     | 1320    |
| Sad          | 0.73      | 0.74   | 0.74     | 1320    |
| accuracy     |           |        | 0.67     | 4000    |
| macro avg    | 0.67      | 0.67   | 0.67     | 4000    |
| weighted avg | 0.67      | 0.67   | 0.67     | 4000    |

Рисунок 3.1 – Результати класифікації Logistic Regression

Більш детальний аналіз результатів класифікації можна виконати за допомогою матриці помилок (рисунок 3.2). З аналізу матриці видно, що модель достатньо добре розпізнає класи Happy та Sad, для яких кількість правильно класифікованих прикладів є найбільшою. Водночас для класу

Neutral спостерігається більша кількість помилок класифікації, що пов'язано з тим, що цей клас знаходиться між двома іншими класами за значенням інтегрального показника настрою, тому його складніше відокремити.

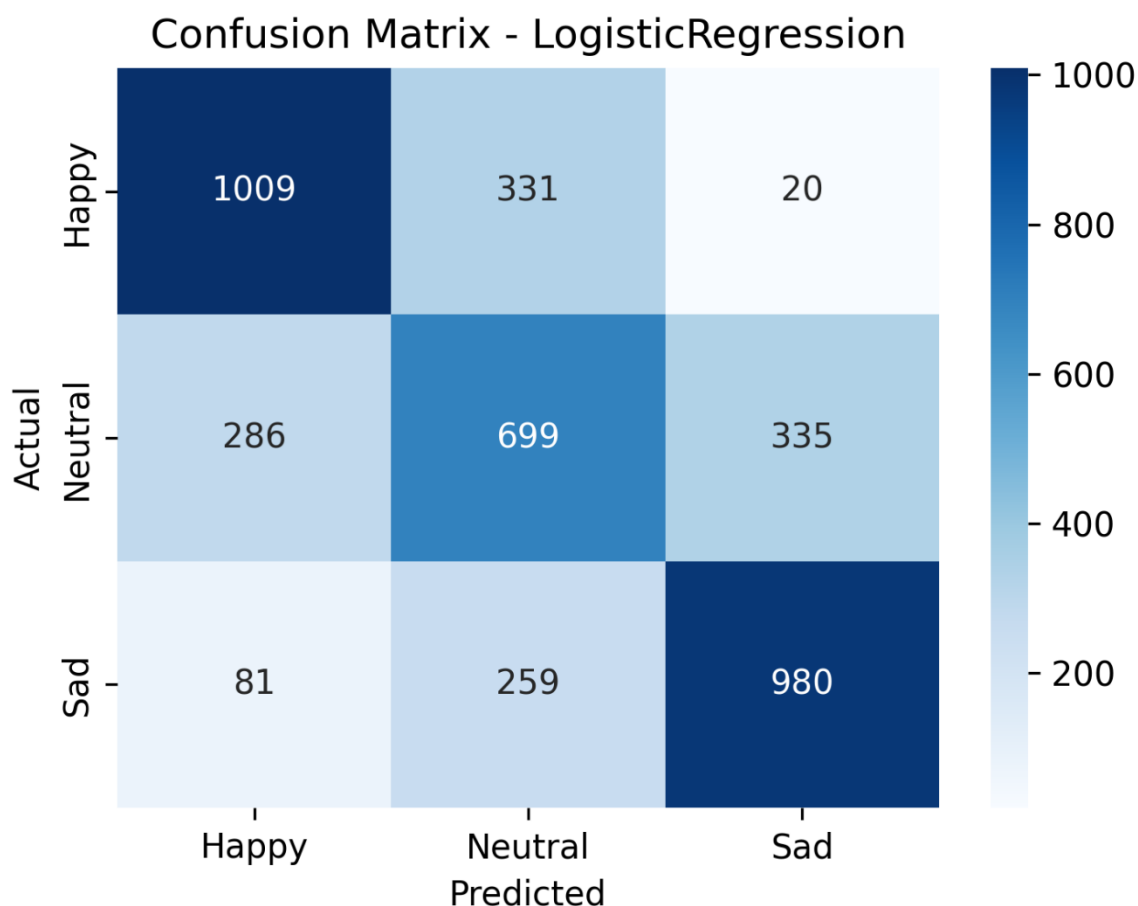


Рисунок 3.2 – Матриця помилок класифікації для Logistic Regression

Додаткову оцінку ефективності моделі можна отримати за допомогою аналізу ROC-кривих (рисунок 3.3). Як видно з графіка, для класів Happy та Sad значення площі під кривою (AUC) становлять відповідно 0.897 та 0.885, що свідчить про хорошу здатність моделі розрізняти ці класи. Для класу Neutral значення AUC є нижчим і становить 0.740, що також підтверджує складність класифікації нейтрального емоційного стану.

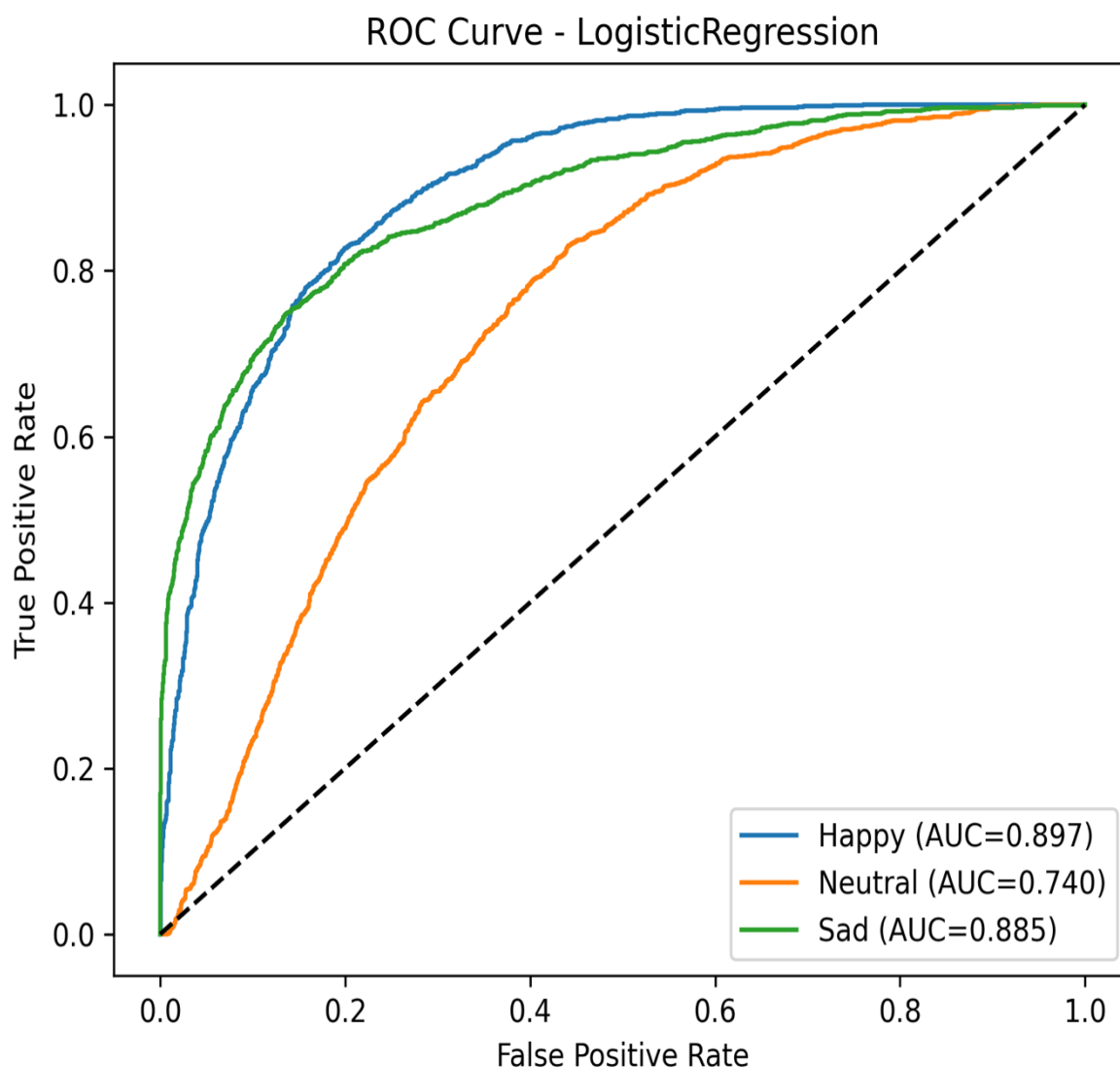


Рисунок 3.3 – ROC-криві для багатокласової класифікації настрою моделлю Logistic Regression

Загалом результати експерименту показують, що модель Logistic Regression демонструє середній рівень точності класифікації настрою користувача. Незважаючи на відносну простоту алгоритму, модель здатна ефективно розпізнавати крайні емоційні стани, проте має певні труднощі з коректною класифікацією нейтрального класу.

Отримані результати використовуються для подальшого порівняння з іншими моделями машинного навчання, що досліджуються у межах даної роботи.

### 3.4.2 Результати класифікації для моделі K-Nearest Neighbors

Модель K-Nearest Neighbors (рисунок 3.4) була використана для класифікації настрою користувача на основі поведінкових характеристик. Згідно з отриманими результатами точність класифікації становить 0.7762, що є вищим показником порівняно з попередньо розглянутою моделлю Logistic Regression. Значення метрик precision, recall та F1-score також знаходяться на рівні близько 0.78, що свідчить про більш ефективну роботу моделі під час класифікації різних емоційних станів.

```

=====
Model: KNN
Accuracy: 0.7762
Precision: 0.7778
Recall: 0.7762
F1-score: 0.7769
Cross-val accuracy: 0.7781500000000001

```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Happy        | 0.81      | 0.82   | 0.82     | 1360    |
| Neutral      | 0.66      | 0.67   | 0.67     | 1320    |
| Sad          | 0.86      | 0.83   | 0.85     | 1320    |
| accuracy     |           |        | 0.78     | 4000    |
| macro avg    | 0.78      | 0.78   | 0.78     | 4000    |
| weighted avg | 0.78      | 0.78   | 0.78     | 4000    |

Рисунок 3.4 – Результати класифікації K-Nearest Neighbors

З аналізу матриці видно (рисунок 3.5), що модель досить добре розпізнає класи Happy та Sad, де кількість правильно класифікованих прикладів є найбільшою. Водночас деякі помилки спостерігаються під час класифікації нейтрального настрою, що проявляється у плутанні класу Neutral із класами Happy та Sad.

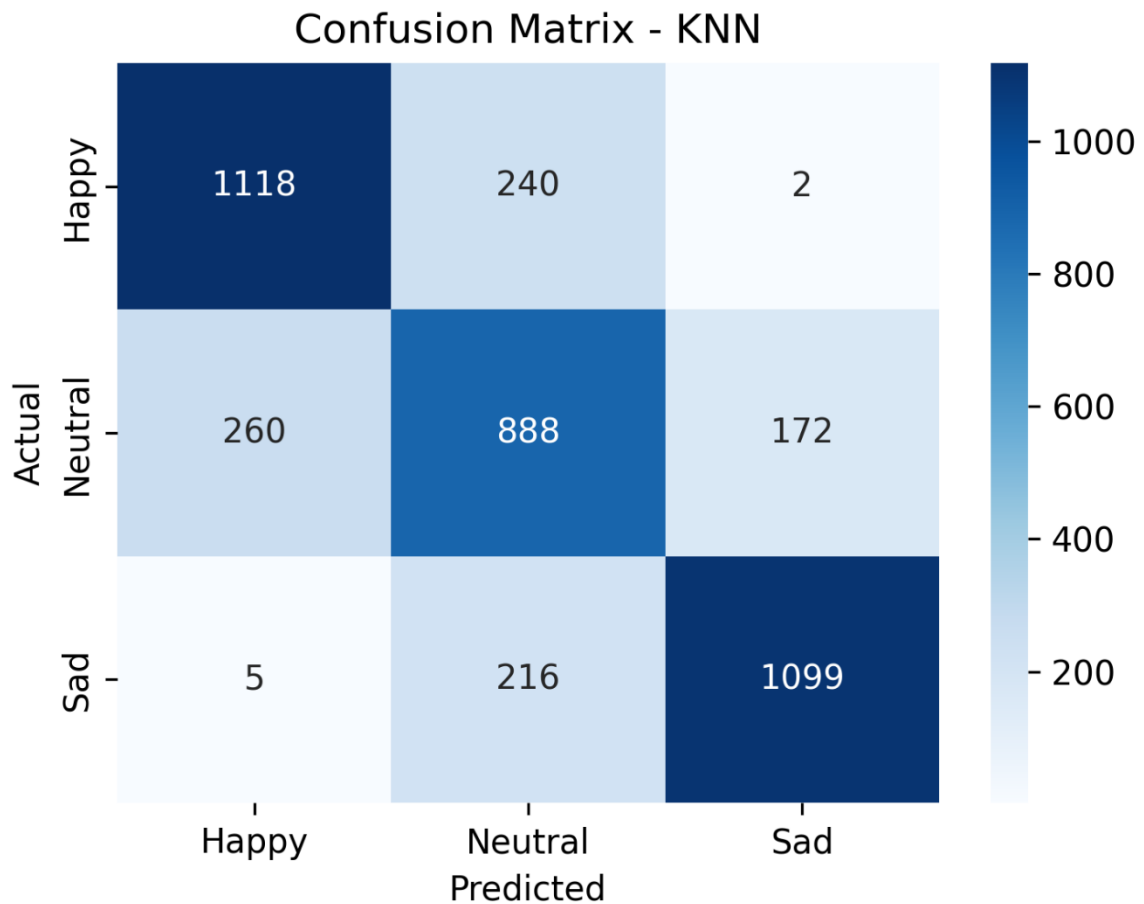


Рисунок 3.5 – Матриця помилок класифікації для K-Nearest Neighbors

Додаткову оцінку ефективності моделі можна отримати за допомогою аналізу ROC-кривих (рисунок 3.6).

Для класу Happy значення площі під ROC-кривою становить 0.942, для класу Neutral – 0.843, а для класу Sad – 0.957. Отримані значення свідчать про високу здатність моделі розрізняти відповідні класи, особливо для крайніх емоційних станів.

Загалом результати експерименту показують, що модель K-Nearest Neighbors демонструє вищу якість класифікації порівняно з базовою моделлю Logistic Regression. Підвищення точності класифікації свідчить про ефективність використання методу найближчих сусідів для задачі прогнозування настрою на основі поведінкових характеристик користувача.

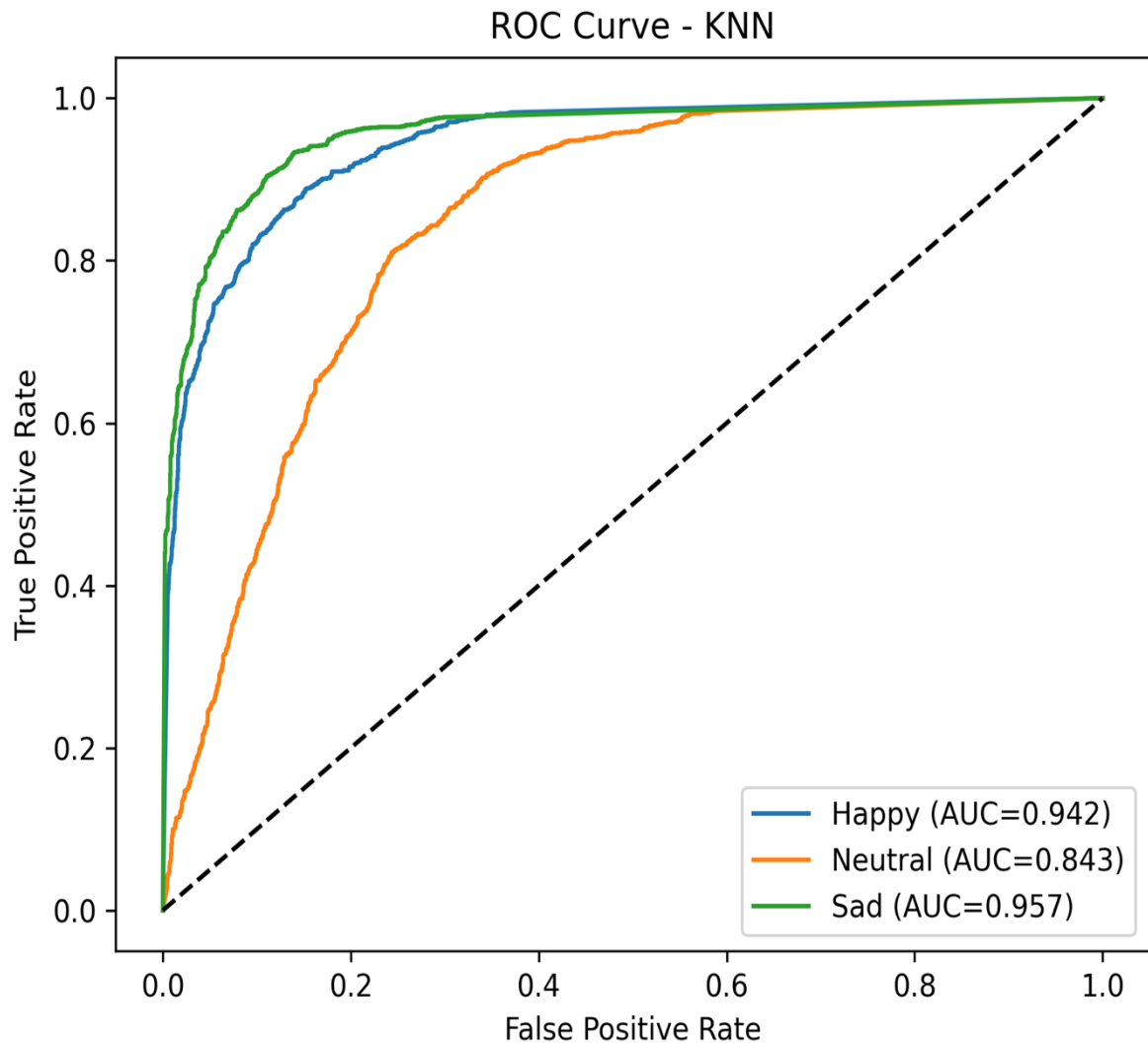


Рисунок 3.6 – ROC-криві для багатокласової класифікації настрою моделлю K-Nearest Neighbors

### 3.4.3 Результати класифікації для моделі Decision Tree

Модель Decision Tree (рисунок 3.7) була використана для класифікації настрою користувача на основі поведінкових характеристик. Згідно з отриманими результатами точність класифікації становить 0.7538. Значення метрик precision, recall та F1-score знаходяться приблизно на рівні 0.75–0.76, що свідчить про достатньо стабільну якість класифікації для різних класів емоційного стану.

```

=====
Model: DecisionTree
Accuracy: 0.7538
Precision: 0.762
Recall: 0.7538
F1-score: 0.7567
Cross-val accuracy: 0.7551499999999999

```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Happy        | 0.80      | 0.79   | 0.79     | 1360    |
| Neutral      | 0.62      | 0.68   | 0.65     | 1320    |
| Sad          | 0.87      | 0.79   | 0.83     | 1320    |
| accuracy     |           |        | 0.75     | 4000    |
| macro avg    | 0.76      | 0.75   | 0.76     | 4000    |
| weighted avg | 0.76      | 0.75   | 0.76     | 4000    |

Рисунок 3.7 – Результати класифікації моделі Decision Tree

Детальніше результати класифікації представлені у вигляді матриці помилок (рисунок 3.8). Матриця помилок дозволяє проаналізувати, як саме модель розподіляє передбачення між різними класами та у яких випадках виникають помилки класифікації. По діагоналі матриці розташовані значення правильних передбачень для кожного класу, тоді як інші елементи відображають випадки помилкової класифікації.

Аналіз матриці показує, що модель досить добре класифікує класи Happy та Sad, для яких кількість правильно визначених прикладів є найбільшою. Це свідчить про те, що поведінкові характеристики, притаманні позитивному та негативному емоційним станам, мають більш виражені відмінності у наборі ознак, що дозволяє алгоритму дерева рішень ефективно їх розділяти. Водночас для класу Neutral спостерігається більша кількість помилок класифікації, що пояснюється тим, що нейтральний емоційний стан має характеристики, близькі до двох інших класів.

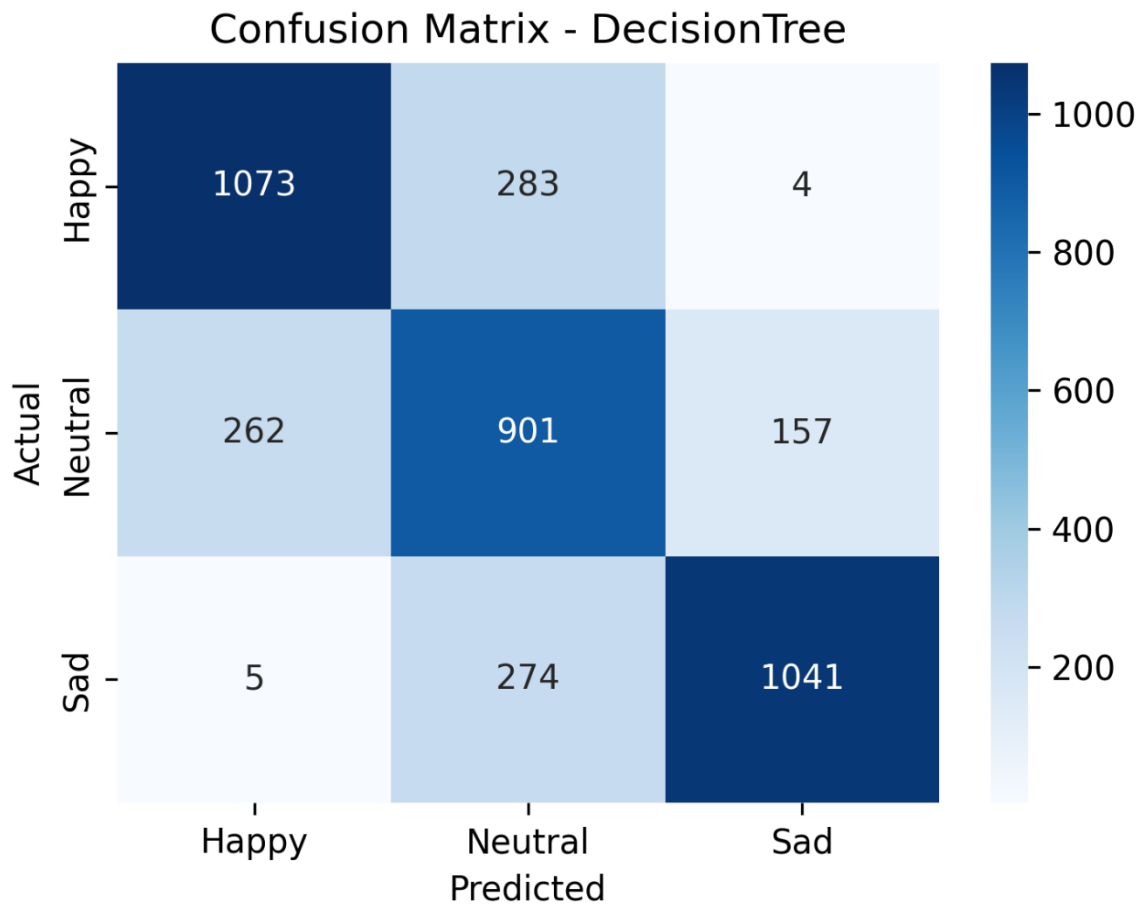


Рисунок 3.8 – Матриця помилок класифікації для моделі Decision Tree

Додаткову оцінку ефективності моделі можна виконати за допомогою аналізу ROC-кривих (рисунок 3.9). ROC-крива відображає залежність між часткою правильно визначених позитивних прикладів та часткою хибнопозитивних передбачень для різних значень порогу класифікації.

Для класу Happy значення площі під ROC-кривою становить 0.924, для класу Neutral – 0.798, а для класу Sad – 0.929. Значення площі під кривою є важливим показником якості моделі, оскільки характеризує здатність алгоритму правильно розрізняти приклади класів. Чим ближче значення AUC до одиниці, тим кращою є якість класифікації. У даному випадку для класів Happy та Sad отримані значення перевищують 0.9, що свідчить про високу ефективність моделі під час розпізнавання цих емоційних станів.

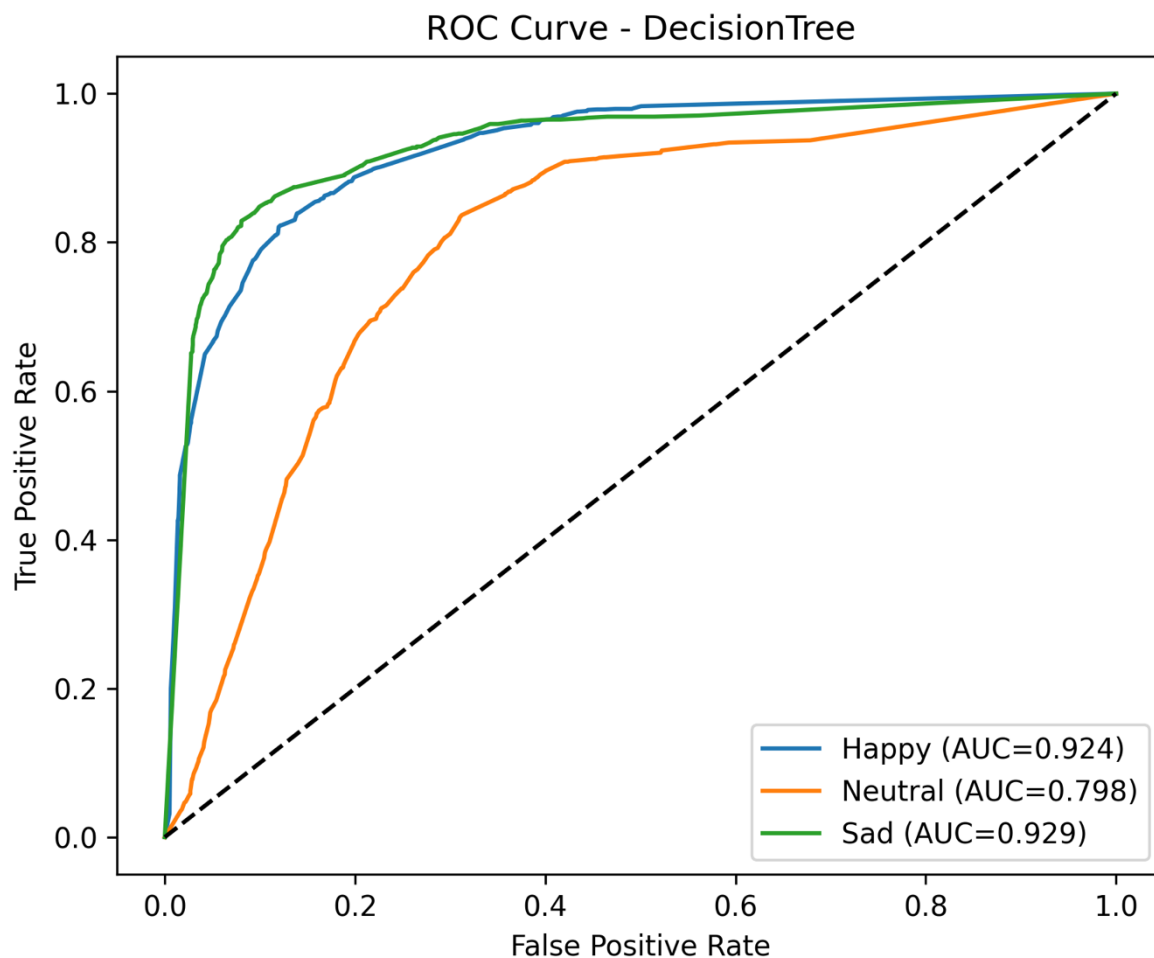


Рисунок 3.9 – ROC-криві для багатокласової класифікації настрою моделлю Decision Tree

Загалом результати експерименту показують, що модель Decision Tree демонструє стабільну якість класифікації та забезпечує достатньо високі показники точності. Проте отримані результати дещо поступаються моделі K-Nearest Neighbors, що свідчить про доцільність подальшого порівняльного аналізу з іншими алгоритмами машинного навчання.

#### 3.4.4 Результати класифікації для моделі Random Forest

Модель Random Forest є ансамблевим методом машинного навчання, що базується на використанні великої кількості дерев рішень, результати

яких агрегуються для отримання фінального передбачення. Основні результати роботи моделі наведені нижче (рисунок 3.10). Отримана точність класифікації становить 0.7897, що є одним із найвищих показників серед досліджених моделей. Значення метрик precision, recall та F1-score також знаходяться приблизно на рівні 0.79, що свідчить про стабільну та збалансовану роботу моделі під час класифікації різних класів настрою.

```

=====
Model: RandomForest
Accuracy: 0.7897
Precision: 0.7913
Recall: 0.7897
F1-score: 0.7904
Cross-val accuracy: 0.79145

```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Happy        | 0.84      | 0.81   | 0.83     | 1360    |
| Neutral      | 0.68      | 0.70   | 0.69     | 1320    |
| Sad          | 0.86      | 0.86   | 0.86     | 1320    |
| accuracy     |           |        | 0.79     | 4000    |
| macro avg    | 0.79      | 0.79   | 0.79     | 4000    |
| weighted avg | 0.79      | 0.79   | 0.79     | 4000    |

Рисунок 3.10 – Результати класифікації моделі Random Forest

Матриця помилок (рисунок 3.11) дозволяє оцінити розподіл правильних та помилкових передбачень для кожного класу емоційного стану. З аналізу матриці видно, що модель досить ефективно класифікує класи Happy та Sad, для яких кількість правильно передбачених прикладів є найбільшою. Для класу Neutral також спостерігається достатньо висока кількість правильних передбачень, проте частина прикладів цього класу класифікується як Happy або Sad.

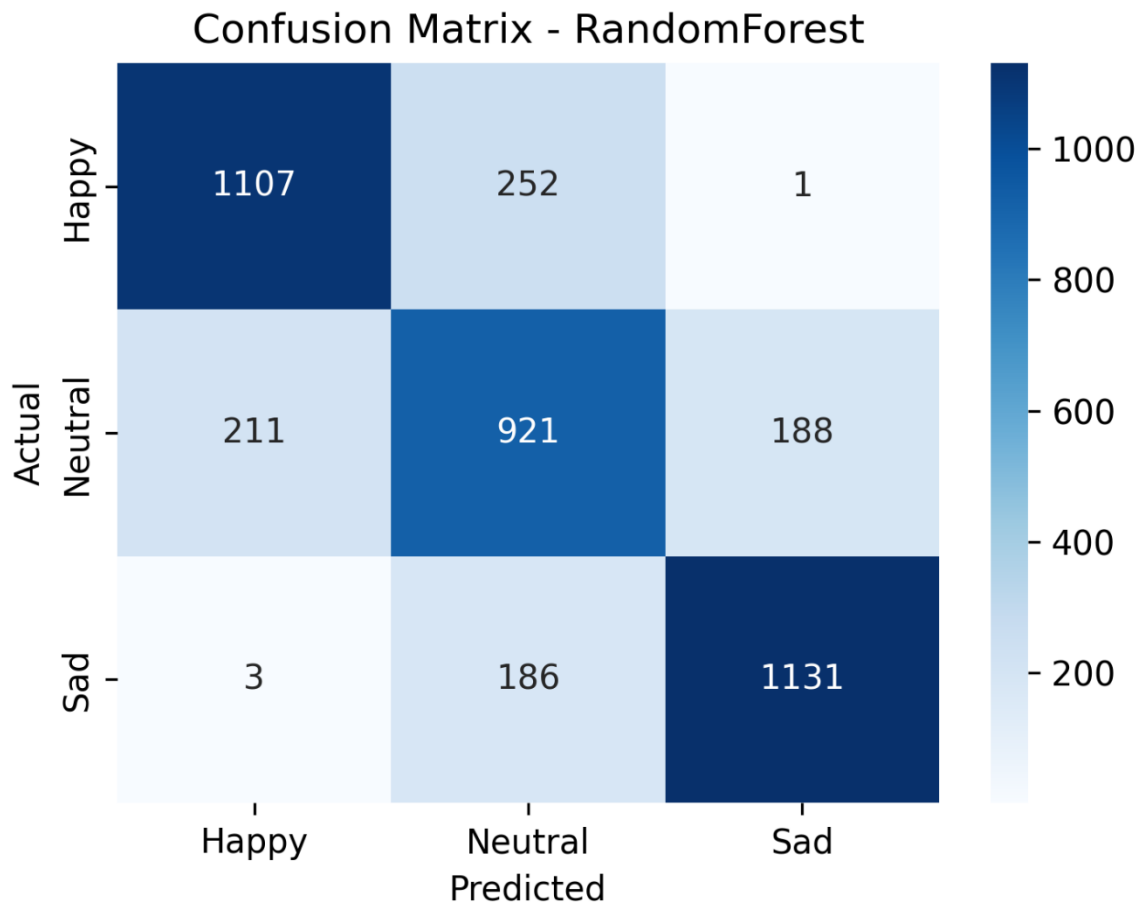


Рисунок 3.11 – Матриця помилок класифікації для моделі Random Forest

Додаткову оцінку ефективності моделі можна отримати за допомогою аналізу ROC-кривих, що наведені нижче (рисунок 3.12). Для класу Happy значення площі під ROC-кривою становить 0.954, для класу Neutral – 0.862, а для класу Sad – 0.967.

Такі значення AUC свідчать про високу здатність моделі Random Forest розрізняти різні класи настрою. Найвищі значення показника спостерігаються для класів Happy та Sad, що вказує на високу точність класифікації крайніх емоційних станів. Водночас для класу Neutral значення AUC є дещо нижчим, що узгоджується з результатами аналізу матриці помилок.

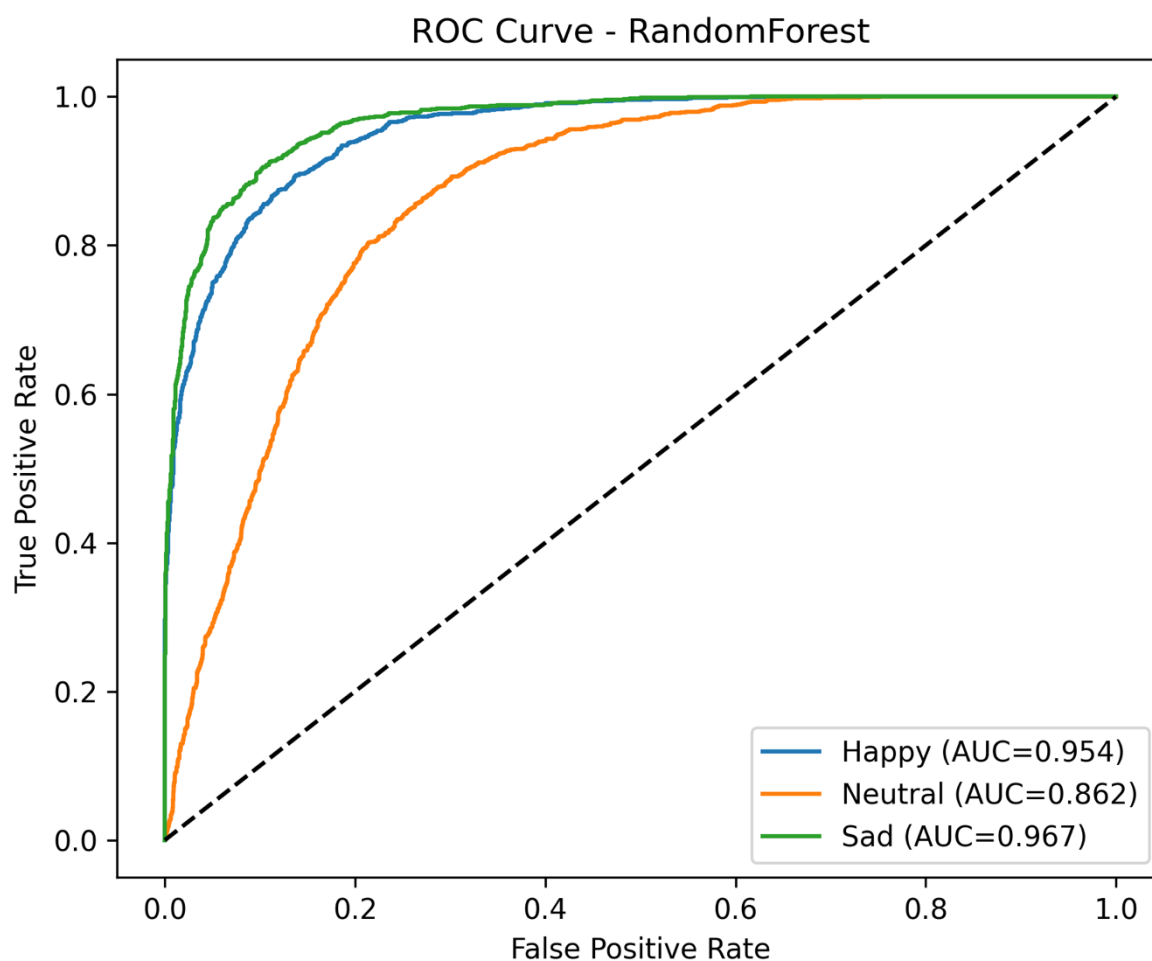


Рисунок 3.12 – ROC-криві для багатокласової класифікації настрою моделлю Random Forest

Загалом результати експериментального дослідження показують, що модель Random Forest демонструє високу якість класифікації настрою користувача та перевищує за точністю більшість інших розглянутих алгоритмів.

Використання ансамблевого підходу дозволяє зменшити вплив помилок окремих дерев рішень та підвищити стабільність прогнозування. Отримані результати свідчать про доцільність використання даної моделі для задачі прогнозування настрою на основі поведінкових показників користувача.

### 3.4.5 Результати класифікації для моделі Gradient Boosting

Модель Gradient Boosting (рисунок 3.13) є одним із сучасних ансамблевих методів машинного навчання, який базується на послідовному навчанні набору слабких моделей, зазвичай дерев рішень. Згідно з отриманими результатами точність класифікації становить 0.7928, що є найвищим показником серед розглянутих моделей. Значення метрик precision, recall та F1-score також знаходяться приблизно на рівні 0.79–0.80, що свідчить про стабільну та збалансовану роботу моделі під час класифікації різних емоційних станів користувача.

```

=====
Model: GradientBoosting
Accuracy: 0.7928
Precision: 0.7952
Recall: 0.7928
F1-score: 0.7938
Cross-val accuracy: 0.7970999999999999

```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| Happy        | 0.84      | 0.81   | 0.83     | 1360    |
| Neutral      | 0.68      | 0.71   | 0.69     | 1320    |
| Sad          | 0.87      | 0.86   | 0.86     | 1320    |
| accuracy     |           |        | 0.79     | 4000    |
| macro avg    | 0.79      | 0.79   | 0.79     | 4000    |
| weighted avg | 0.80      | 0.79   | 0.79     | 4000    |

Рисунок 3.13 – Результати класифікації моделі Gradient Boosting

Більш детальний аналіз результатів класифікації представлений у вигляді матриці помилок, що наведена нижче (рисунок 3.14). Матриця помилок дозволяє оцінити кількість правильних та помилкових передбачень для кожного класу настрою. З аналізу матриці видно, що модель досить ефективно класифікує класи Happy та Sad, де кількість правильних

передбачень є найбільшою. Для класу Neutral також спостерігається значна кількість правильно визначених прикладів, однак частина з них класифікується як Happy або Sad. Це пояснюється тим, що поведінкові характеристики нейтрального стану можуть частково перетинатися з характеристиками інших емоційних станів, що ускладнює процес їх чіткого розмежування.

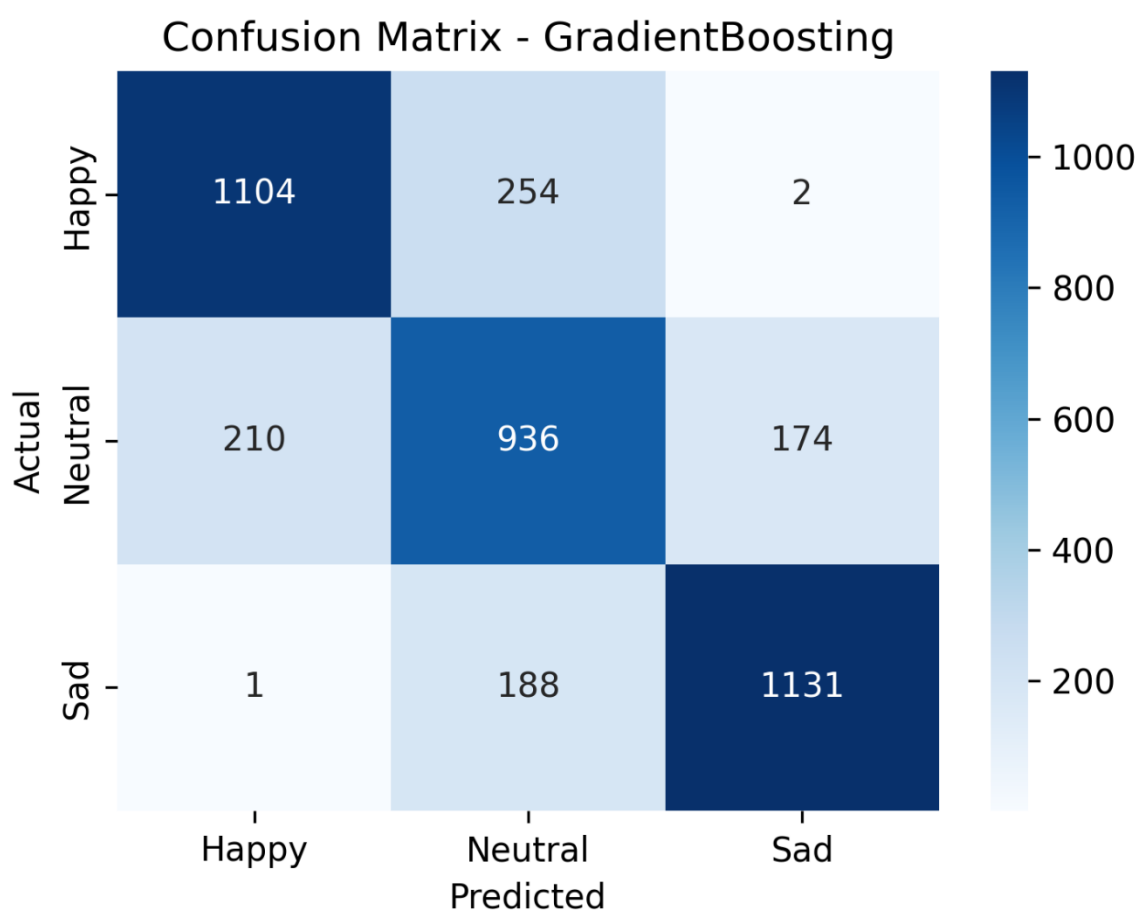


Рисунок 3.14 – Матриця помилок класифікації для моделі Gradient Boosting

Додаткову оцінку ефективності моделі можна виконати за допомогою аналізу ROC-кривих (рисунок 3.15).

Для класу Happy значення площі під ROC-кривою становить 0.956, для класу Neutral – 0.869, а для класу Sad – 0.969. Отримані значення свідчать про високу здатність моделі Gradient Boosting розрізняти різні

класи настрою користувача. Найкращі результати спостерігаються для класів Happy та Sad, тоді як для класу Neutral значення AUC є дещо нижчим, що також підтверджує складність класифікації нейтрального емоційного стану.

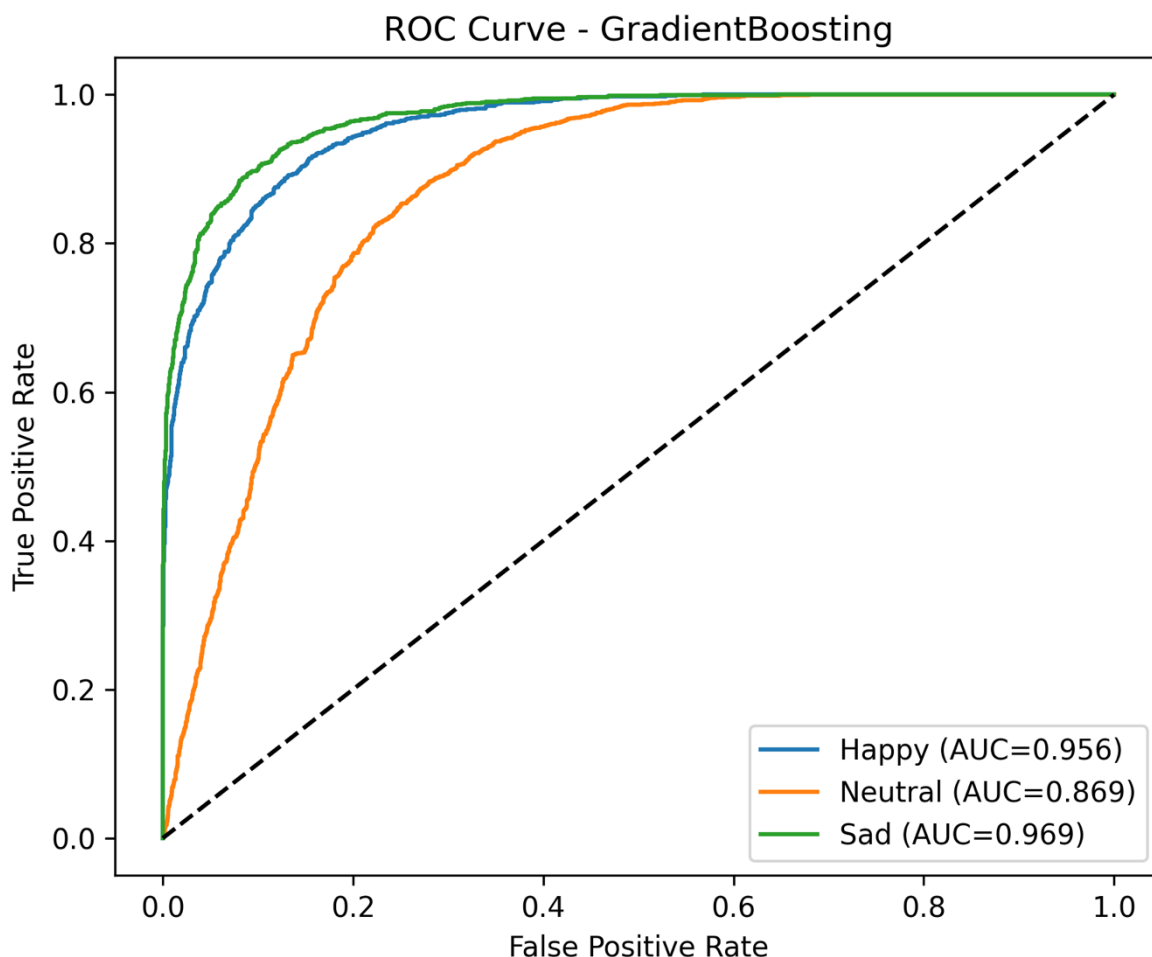


Рисунок 3.15 – ROC-криві для багатокласової класифікації настрою моделлю Gradient Boosting

Аналіз важливості ознак дозволяє визначити, які поведінкові параметри мають найбільший вплив на результат прогнозування настрою користувача. З аналізу графіка (рисунок 3.16) видно, що найбільший вплив на прогноз настрою має показник `activity_minutes`, значення важливості якого становить 0.356. Це свідчить про те, що рівень фізичної активності

користувача є одним із ключових факторів, що впливають на його емоційний стан. Наступними за значущістю є показники `screen_time` та `sleep_hours`, важливість яких становить відповідно 0.306 та 0.283. Дані результати підтверджують припущення про те, що тривалість використання екрана та якість сну мають суттєвий вплив на настрій користувача.

Найменший внесок у процес класифікації має ознака `steps`, значення важливості якої становить 0.056. Це може пояснюватися тим, що кількість кроків частково корелює з показником фізичної активності, тому її вплив на результати прогнозування є менш вираженим.

Загалом отримані результати свідчать про те, що поведінкові показники користувача мають різний ступінь впливу на прогнозування настрою, а найбільш значущими факторами є фізична активність, використання екрана та тривалість сну.

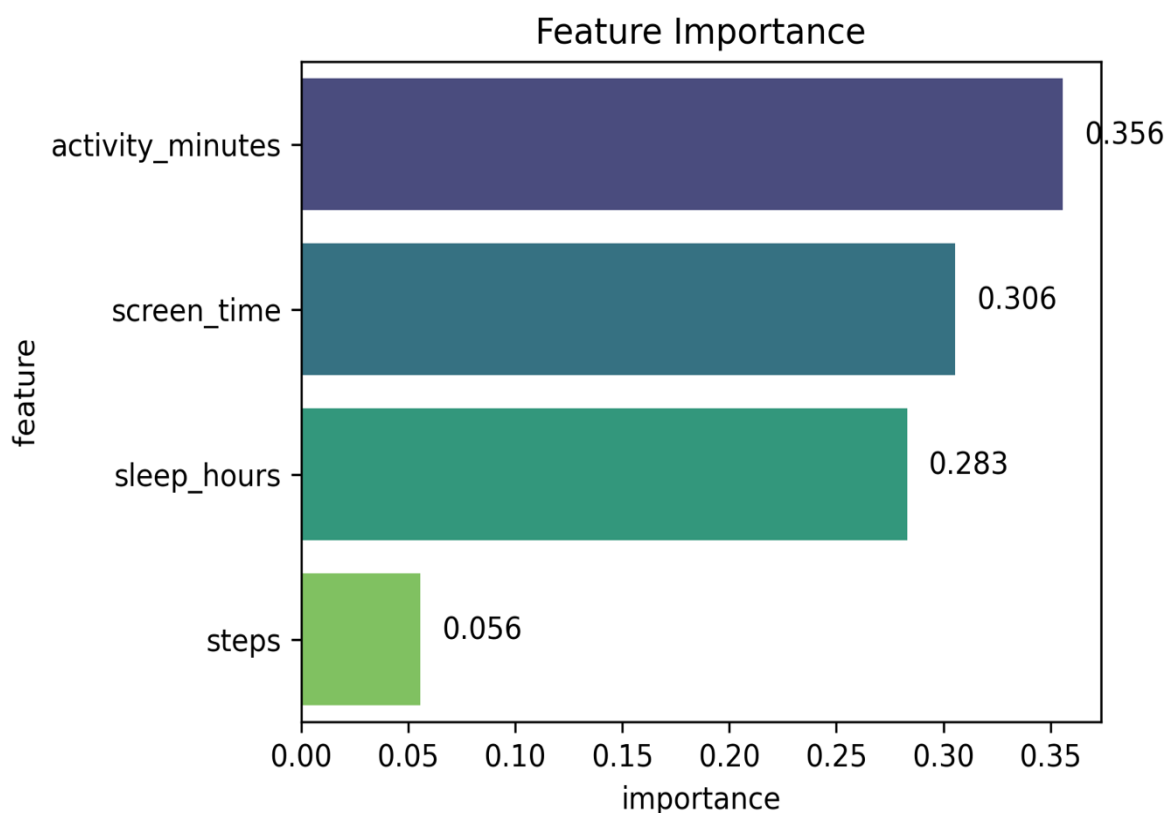


Рисунок 3.16 – Важливість ознак у моделі Gradient Boosting

### 3.4.6 Порівняльний аналіз результатів моделей

У цьому підрозділі виконується порівняльний аналіз результатів класифікації, отриманих для різних моделей машинного навчання. Для оцінювання ефективності алгоритмів використовуються метрики accuracy, precision, recall, F1-score, а також середнє значення точності, отримане за результатами крос-валідації. Узагальнені результати роботи моделей наведені у таблиці 3.1.

Таблиця 3.1 – Порівняння точності моделей машинного навчання

| Модель                 | Точність | Precision | Recall | F1-score | Cross-val<br>точність |
|------------------------|----------|-----------|--------|----------|-----------------------|
| Logistic<br>Regression | 0.672    | 0.67      | 0.67   | 0.67     | 0.668                 |
| KNN                    | 0.77625  | 0.77      | 0.77   | 0.77     | 0.778                 |
| RandomForest           | 0.78975  | 0.79      | 0.78   | 0.79     | 0.791                 |
| DecisionTree           | 0.75375  | 0.76      | 0.75   | 0.75     | 0.755                 |
| GradientBoosting       | 0.79275  | 0.79      | 0.79   | 0.79     | 0.797                 |

З аналізу таблиці 3.1 видно, що всі моделі демонструють достатньо близькі значення основних метрик класифікації, однак їх ефективність відрізняється. Найнижчі результати показала модель Logistic Regression, точність якої становить 0.672. Це пояснюється тим, що дана модель є лінійною та має обмежені можливості для опису складних нелінійних залежностей між поведінковими характеристиками користувача та його емоційним станом.

Модель K-Nearest Neighbors показала значно кращі результати, досягнувши точності 0.776. Даний алгоритм використовує інформацію про найближчі приклади у просторі ознак, що дозволяє ефективніше враховувати локальні закономірності у даних. Модель Decision Tree

продемонструвала точність 0.754, що дещо поступається результатам KNN, однак забезпечує зрозумілу структуру прийняття рішень.

Ансамблеві методи Random Forest та Gradient Boosting показали найкращі результати серед усіх досліджених алгоритмів. Точність моделі Random Forest становить 0.789, тоді як модель Gradient Boosting демонструє найвищий показник точності – 0.793. Це пояснюється тим, що ансамблеві методи поєднують результати кількох базових моделей, що дозволяє зменшити вплив помилок окремих алгоритмів та підвищити загальну якість прогнозування.

Також нижче можна побачити (рисунок 3.17) графічне порівняння точності моделей. З аналізу діаграми видно, що модель Gradient Boosting демонструє найкращий результат, тоді як Logistic Regression має найнижчу точність.

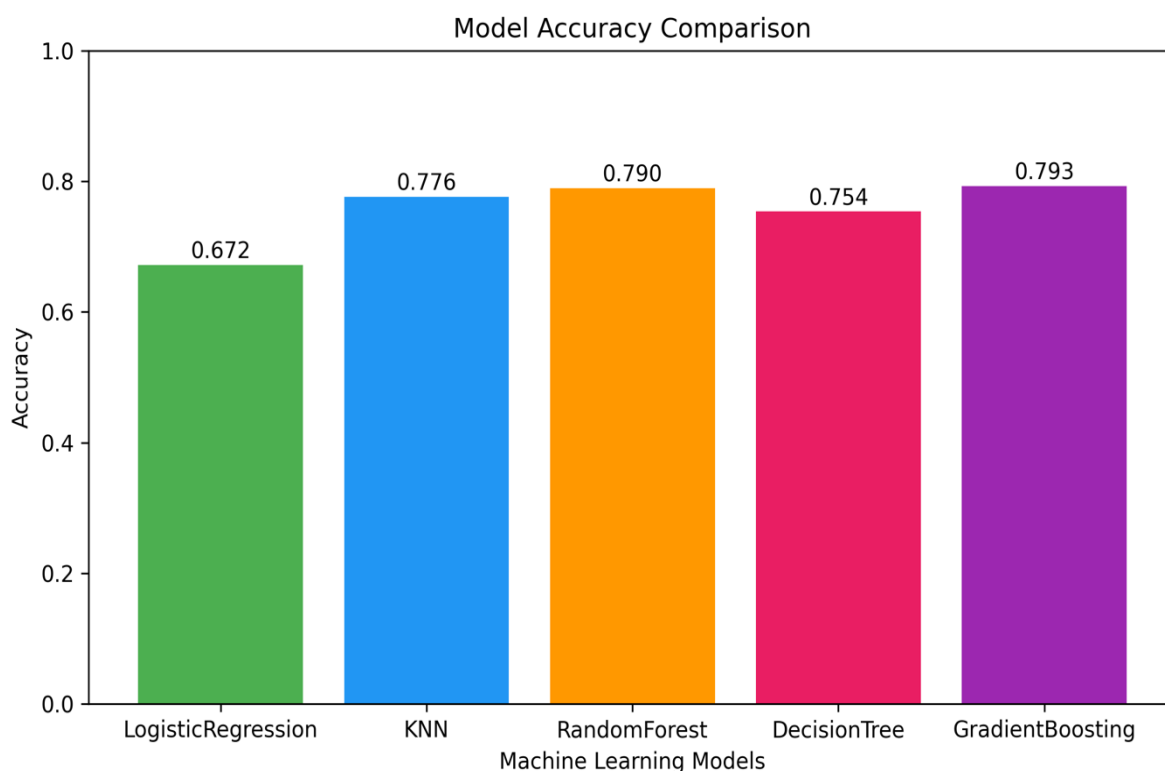


Рисунок 3.17 – Порівняльні результати оцінювання моделей машинного навчання

Отримані результати підтверджують доцільність використання більш складних ансамблевих алгоритмів для задачі прогнозування настрою користувача на основі поведінкових характеристик. На основі проведеного аналізу модель Gradient Boosting може бути обрана як найбільш ефективна для подальшої інтеграції у мобільний застосунок.

### 3.5 Інтеграція Gradient Boosting у Core ML

Після проведення експериментального дослідження та порівняльного аналізу моделей машинного навчання наступним етапом є інтеграція найбільш ефективної моделі у мобільний застосунок, тобто Gradient Boosting. Для цього використовується технологія Core ML, яка дозволяє запускати моделі машинного навчання безпосередньо на пристроях iOS. Такий підхід забезпечує швидке виконання прогнозування, зменшує залежність від серверної інфраструктури та підвищує рівень конфіденційності даних користувача.

Процес інтеграції моделі включає кілька етапів: визначення оптимальної моделі за результатами експерименту, конвертацію навченої моделі у формат Core ML та підготовку отриманої моделі до використання у мобільному застосунку. На кожному з цих етапів виконуються відповідні програмні операції, що забезпечують коректну роботу моделі у середовищі iOS.

#### 3.5.1 Визначення оптимальної моделі прогнозування настрою

На основі проведеного експериментального дослідження було визначено модель машинного навчання, що демонструє найкращі результати класифікації. Для цього використовується показник точності класифікації ассурасу, який характеризує частку правильно класифікованих прикладів у тестовій вибірці. Визначення найкращої моделі здійснюється

шляхом пошуку запису з максимальним значенням показника точності у підсумковій таблиці результатів. Програмна реалізація визначення оптимальної моделі наведена у лістингу 3.10.

Лістинг 3.10 – Визначення найкращої моделі за показником точності класифікації

```
best_row = results_df.loc[
    results_df["accuracy"].idxmax()
]
best_model_name = best_row["model"]
print("Best model:", best_model_name)
```

У результаті виконання цієї операції визначається модель машинного навчання, яка демонструє найкращі показники ефективності серед усіх досліджених алгоритмів. Саме ця модель використовується для подальшої інтеграції у мобільний застосунок.

### 3.5.2 Конвертація моделі у формат Core ML

Після визначення оптимальної моделі необхідно виконати її конвертацію у формат Core ML, який підтримується мобільною платформою iOS. Для цього використовується бібліотека `coremltools`, що забезпечує можливість перетворення моделей, створених у середовищі Python, у формат `.mlmodel`.

Під час конвертації необхідно вказати перелік вхідних ознак, які використовуються моделлю для виконання прогнозування. У даному дослідженні такими ознаками є поведінкові характеристики користувача, зокрема тривалість сну, кількість кроків, час використання екрана та тривалість фізичної активності. Програмна реалізація процесу конвертації моделі наведена у лістингу 3.11.

### Лістинг 3.11 – Конвертація навченої моделі машинного навчання у формат Core ML

```
input_features = [  
    "sleep_hours",  
    "steps",  
    "screen_time",  
    "activity_minutes"  
]  
  
coreml_model = ct.converters.sklearn.convert(  
    best_model,  
    input_features=input_features,  
    output_feature_names="mood"  
)  
  
coreml_model.save("MoodPredictor.mlmodel")
```

У результаті виконання даного коду формується файл моделі з розширенням `.mlmodel`, який може бути безпосередньо використаний у мобільному застосунку.

#### 3.5.3 Підготовка моделі до інтеграції у мобільний застосунок

Після конвертації моделі у формат Core ML необхідно підготувати її до використання у мобільному застосунку. Отриманий файл моделі додається до проєкту в середовищі розробки Xcode, де він автоматично перетворюється у відповідний програмний інтерфейс для мови Swift. Це дозволяє розробнику використовувати модель як звичайний програмний компонент.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було вирішено актуальну науково-прикладну задачу, що полягає у дослідженні методів персоналізованого прогнозування настрою користувача на основі поведінкових показників із подальшою інтеграцією отриманих моделей у мобільне середовище із використанням технології Core ML. Поставлена мета роботи досягнута у повному обсязі, що підтверджується реалізацією всіх визначених етапів дослідження, починаючи від аналізу предметної галузі та завершуючи експериментальною перевіркою ефективності розроблених підходів і їх практичною реалізацією у вигляді мобільного застосунку.

У ході виконання роботи було проведено комплексний аналіз сучасних підходів до моніторингу емоційного стану людини та визначено ключові поведінкові фактори, що впливають на формування настрою. Зокрема, встановлено, що такі показники, як тривалість сну, рівень фізичної активності, кількість кроків та час використання мобільного пристрою, можуть виступати інформативними ознаками для побудови моделей прогнозування емоційного стану. Це дозволило сформулювати основу для подальшої формалізації задачі та побудови математичної моделі визначення настрою користувача.

Одним із основних наукових результатів роботи є розробка математичної моделі визначення інтегрального показника настрою, яка враховує нелінійний вплив поведінкових факторів на емоційний стан користувача. Запропонована модель базується на використанні різних функціональних залежностей для кожного з параметрів та дозволяє отримати узагальнену числову оцінку настрою. Додатково було розроблено методику перетворення інтегрального показника у дискретні класи емоційного стану на основі квантильного підходу, що забезпечує збалансованість класів у навчальному наборі даних. Такий підхід усуває

проблему перекосу класів та підвищує якість навчання моделей машинного навчання.

У процесі дослідження було реалізовано повний цикл машинного навчання, що включає підготовку даних, нормалізацію ознак, поділ на тренувальну та тестову вибірки, побудову моделей та їх оцінювання. Для розв'язання задачі класифікації настрою було використано декілька алгоритмів машинного навчання, зокрема Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest та Gradient Boosting. Проведений порівняльний аналіз дозволив оцінити ефективність кожного з підходів за такими метриками, як точність класифікації, повнота, точність передбачення та F1-міра.

За результатами експериментального дослідження встановлено, що ансамблеві методи машинного навчання, зокрема Gradient Boosting та Random Forest, демонструють найвищі показники якості класифікації. Це пояснюється їх здатністю враховувати складні нелінійні залежності між поведінковими показниками користувача та його емоційним станом. Застосування крос-валідації підтвердило стабільність отриманих результатів та здатність обраної моделі до узагальнення на нових даних.

Практичним результатом роботи є розробка програмного прототипу мобільного застосунку для моніторингу та прогнозування настрою користувача, який реалізує концепцію локального виконання моделей машинного навчання. Особливістю реалізованого підходу є використання технології Core ML, що дозволяє виконувати прогнозування безпосередньо на мобільному пристрої без передачі даних на зовнішні сервери. Це забезпечує високий рівень конфіденційності персональної інформації та відповідає сучасним вимогам до безпеки даних. Крім того, застосунок реалізує функціональність збору поведінкових даних, прогнозування настрою, відображення статистики та візуалізації змін емоційного стану користувача.

Аналіз досягнутих кількісних показників показав, що використання ансамблевих моделей дозволяє досягти високої точності класифікації настрою, що підтверджує доцільність їх застосування у подібних задачах. Якісні результати дослідження полягають у розробці узагальненої математичної моделі, яка враховує поведінкові фактори, а також у створенні інтегрованої системи прогнозування настрою, здатної працювати у реальному мобільному середовищі. Отримані результати свідчать про ефективність запропонованого підходу та його практичну придатність для створення інтелектуальних систем підтримки психологічного благополуччя.

Окремим напрямом розвитку є персоналізація моделей машинного навчання шляхом їх адаптації до індивідуальних особливостей користувача. Це може включати використання методів онлайн-навчання або донавчання моделей безпосередньо на пристрої користувача. Такий підхід дозволить підвищити точність прогнозування та забезпечити більш індивідуалізований аналіз емоційного стану.

Крім того, перспективним є вдосконалення методів інтерпретації результатів роботи моделей машинного навчання, що дозволить підвищити прозорість системи та зробити її більш зрозумілою для користувача. Це особливо важливо у контексті застосунків для моніторингу психологічного стану, де користувачі очікують не лише прогнозу, але й пояснення причин зміни настрою.

Таким чином, у роботі було досягнуто поставленої мети, отримано науково обґрунтовані результати та створено практичну реалізацію системи прогнозування настрою користувача. Запропонований підхід може бути використаний як основа для подальших досліджень у сфері інтелектуальних мобільних систем, цифрового здоров'я та персоналізованих сервісів аналізу поведінкових даних.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Core ML. Apple Developer Documentation. URL: <https://developer.apple.com/documentation/coreml> (дата звернення: 01.02.2026).
2. Create ML. Apple Developer Documentation. URL: <https://developer.apple.com/documentation/createml> (дата звернення: 03.02.2026).
3. Machine Learning with Core ML. Apple Developer. URL: <https://developer.apple.com/machine-learning/core-ml/> (дата звернення: 05.02.2026).
4. SwiftUI Documentation. Apple Developer. URL: <https://developer.apple.com/documentation/swiftui> (дата звернення: 07.02.2026).
5. Swift Charts Framework Documentation. Apple Developer. URL: <https://developer.apple.com/documentation/charts> (дата звернення: 09.02.2026).
6. Core ML Execution Provider. ONNX Runtime Documentation. URL: <https://onnxruntime.ai/docs/execution-providers/CoreML-ExecutionProvider.html> (дата звернення: 11.02.2026).
7. Scikit-learn: Machine Learning in Python. URL: <https://scikit-learn.org/stable/> (дата звернення: 13.02.2026).
8. Core ML Tools Documentation. URL: <https://coremltools.readme.io> (дата звернення: 15.02.2026).
9. TensorFlow Lite. Machine Learning for Mobile and Embedded Devices. URL: <https://www.tensorflow.org/lite> (дата звернення: 17.02.2026).
10. HealthKit Framework. Apple Developer Documentation. URL: <https://developer.apple.com/documentation/healthkit> (дата звернення: 20.02.2026).

11. Pedregosa F., Varoquaux G., Gramfort A. та ін. Scikit-learn: Machine Learning in Python // *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830.
12. Breiman L. Random Forests // *Machine Learning*. 2001. Vol. 45, № 1. P. 5–32.
13. Cortes C., Vapnik V. Support-Vector Networks // *Machine Learning*. 1995. Vol. 20, № 3. P. 273–297.
14. Bishop C. *Pattern Recognition and Machine Learning*. New York : Springer, 2006. 738 p.
15. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. Cambridge : MIT Press, 2016. 775 p.
16. Géron A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Sebastopol : O'Reilly Media, 2019. 856 p.
17. Chollet F. *Deep Learning with Python*. New York: Manning Publications, 2018. 384 p.
18. Zhang Z., Wang X., Li X. Mood Prediction Using Machine Learning Techniques // *IEEE Access*. 2020. Vol. 8. P. 102247–102256.
19. Vijjala S., Majumder B., Gupta A., Surana H. *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*. Sebastopol : O'Reilly Media, 2020. 325 p.
20. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. 4th ed. Hoboken : Pearson, 2021. 1136 p.
21. Bodyanskiy Y., Zolotukhin O., Yerokhin A., Kudryavtseva M., Yerokhin M. *Fast stacking neuro-neo-fuzzy system for inverse modeling in online mode*. International Journal of Computing. 2025. Vol. 24, No. 4. P. 661–667.