

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

(повна назва)

Кафедра Системотехніки

(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів штучного інтелекту для оптимізації планування дня користувача

(тема)

Виконав:

студент 2 курсу, групи ІТПм-22-1

Коротич К.О.

(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформаційні

технології проектування

(повна назва освітньої програми)

Керівник проф. Колесник Л.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Гребінник І.В.

(прізвище, ініціали)

2023 р.

*Кваліфікаційна робота оформлена у відповідності до вимог діючих стандартів та методичних вказівок.*

*Матеріали кваліфікаційної роботи не містять відомостей, що заборонені для опублікування у відкритих виданнях.*

*Попередній захист проведено 13 січня 2024 року.*

*Керівник кваліфікаційної роботи*  *Л.В. Колесник*

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна

Освітня програма Інформаційні технології проектування  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Коротичу Кирилу Олеговичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів штучного інтелекту для оптимізації планування дня користувача

затверджена наказом університету від 20 листопада 2023 р. № 1373 СТ

2. Термін подання студентом роботи до екзаменаційної комісії 18 січня 2024 р.

3. Вихідні дані до роботи Функція: Дослідження та використання методів штучного інтелекту для оптимізації планування дня. Форма діалогу: мобільний застосунок. Перелік використовуваних програмних засобів: ОС Windows, інтегроване середовище розробки PyCharm та Visual Studio Code, Python, JavaScript, Typescript, Tensorflow, Pandas, NodeJS, MongoDB. Технічне забезпечення: мобільний телефон iOS чи Android, з'єднання з інтернетом, застосунок Play Market/AppStore.

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Аналіз предметної області. 4.2 Аналіз сервісів штучного інтелекту. 4.3 Порівняння алгоритмів та моделей для створення власного штучного інтелекту. 4.4 Аналіз існуючих методів планування дня. 4.5 Аналіз інтегрованого зі штучним інтелектом інтерфейсу. 4.6 Напрямок дослідження. 4.7 Постановка задачі. 4.8 Дослідження алгоритмів заснованих на дереві рішень. 4.9 Дерево рішень. 4.10 Випадковий ліс. 4.11 Градієнтне посилення. 4.12 Висновки дослідження алгоритмів заснованих на дереві рішень. 4.13 Результати експериментальних досліджень. 4.14 Опис тестового набору даних. 4.15 Математичний опис методики оцінювання результатів експерименту. 4.16 Результат досліджень. 4.17 Висновки. 4.18 Перелік джерел посилання.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) 5.1 Схема роботи алгоритму «Бінарне дерево» (1 аркуш формату А4). 5.2 Порівняння роботи алгоритму «Випадковий ліс» з «Бінарним деревом» (1 аркуш формату А4). 5.3 Завантажений набір даних для навчання моделі (1 аркуш формату А4). 5.4. Приклад тестових даних (1 аркуш формату А4). 5.5. Результат предобробки тестових даних (1 аркуш формату А4). 5.6. Архітектура мобільного додатку (1 аркуш формату А4). 5.7 Use Case діаграма процесу створення плану на день (1 аркуш формату А4). 5.8. Архітектура сервера для обробки даних від клієнта (1 аркуш

формату А4). 5.9 Діаграма варіантів, що демонструє можливості користувача (1 аркуш формату А4). 5.10 Згенерований розклад за допомогою моделі із алгоритмом «Випадковий ліс» (1 аркуш формату А4). 5.11 Згенерований розклад за допомогою моделі із алгоритмом «Бінарне дерево» (1 аркуш формату А4). 5.12 Згенерований розклад за допомогою моделі із алгоритмом «Градiєнтне посилення» (1 аркуш формату А4).

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )


| Найменування розділу | Консультант<br>(посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу |      |
|----------------------|--|---|------|
|                      |  | підпис                                      | дата |
|                      |  |   |      |

### КАЛЕНДАРНИЙ ПЛАН

| №  | Назва етапів роботи   | Терміни виконання етапів роботи | Примітка |
|----|---|---------------------------------|----------|
| 1  | Отримання завдання кваліфікаційної роботи                             | 20.11.2023                      | Виконано |
| 2  | Аналіз завдання, літератури та аналогів з теми атестаційної роботи    | 04.10.2023                      | Виконано |
| 3  | Опрацювання літератури та аналіз об'єкту дослідження                  | 15.10.2023                      | Виконано |
| 4  | Постановка та формалізація задачі                                     | 22.10.2023                      | Виконано |
| 5  | Дослідження методів та алгоритмів вирішення задачі. Розробка моделі.  | 03.11.2023                      | Виконано |
| 6  | Розробка архітектури взаємодії користувача та мобільного додатку      | 14.11.2023                      | Виконано |
| 7  | Тестування розроблення програмного забезпечення та порівняння моделей | 20.11.2023                      | Виконано |
| 8  | Оформлення пояснювальної записки                                      | 01.12.2023                      | Виконано |
| 9  | Подача кваліфікаційної роботи на допуск до захисту                    | 13.01.2024                      | Виконано |
| 10 | Підготовка доповіді до захисту кваліфікаційної роботи                 | 15.01.2024                      | Виконано |
| 11 | Подання кваліфікаційної роботи  | 18.01.2024                      | Виконано |

Дата видачі завдання 20 листопада 2023 р.

Студент Коротич К.О. 

Керівник роботи  (підпис)

проф. Колесник Л.В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 104 с., 4 табл., 12 рис., 21 джерел інформації.

ВИПАДКОВИЙ ЛІС, ДЕРЕВО РІШЕНЬ, ГРАДІЄНТНЕ ПОСИЛЕННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, ПЛАНУВАННЯ ДНЯ, JAVASCRIPT, PYTHON.

Об'єкт дослідження – штучний інтелект (ШІ) та його застосування у плануванні дня користувача. Це включає системи, які використовуються для аналізу, прогнозування та автоматизації щоденних завдань і рішень.

Предмет дослідження – методи ШІ, такі як машинне навчання, нейронні мережі та алгоритми оптимізації, що використовуються для підвищення ефективності планування дня. Вивчаються аспекти, які включають персоналізацію рекомендацій, автоматизацію рутинних завдань та адаптацію до змінних обставин.

Мета роботи – проаналізувати і оцінити, як методи ШІ можуть бути використані для покращення планування дня користувача. Основна мета полягає у визначенні найбільш ефективних стратегій ШІ для оптимізації повсякденних завдань і підвищення продуктивності.

Методи дослідження – проведення аналізу існуючих методів машинного навчання, вибір відповідного для реалізації задачі підходу, розробка моделі за алгоритмом, проведення тестування та оцінювання.

Наукова новизна результату принесе вклад у розуміння того, як адаптивні та персоналізовані методи ШІ можуть оптимізувати щоденне планування. Особливу увагу приділено алгоритмам машинного навчання заснованим на дереві рішень, які можуть класифікувати діяльність користувача за часом та створити пріоритезований список задач, що відкриває нові можливості для розробки інтелектуальних особистих помічників.

## ABSTRACT

Explanatory note to the qualification work: 104 pages, 4 tables, 12 figures, 21 sources.

RANDOM FOREST, DECISION TREE, GRADIENT BOOSTING, ARTIFICIAL INTELLIGENCE, DAY PLANNING, JAVASCRIPT, PYTHON.

The object of research is artificial intelligence (AI) and its application in planning the user's day. This includes systems used to analyze, predict and automate daily tasks and decisions.

The subject of research is AI methods such as machine learning, neural networks and optimization algorithms used to improve the efficiency of day planning. Aspects that are studied include personalization of recommendations, automation of routine tasks and adaptation to changing circumstances.

The purpose of the work is to analyze and evaluate how AI methods can be used to improve the planning of the user's day. The main goal is to identify the most effective AI strategies to optimize everyday tasks and increase productivity.

Research methods – analysis of existing machine learning methods, selection of an appropriate approach for the implementation of the task, development of a model based on an algorithm, testing and evaluation.

The scientific novelty of the result will contribute to the understanding of how adaptive and personalized AI methods can optimize daily planning. Particular attention is paid to machine learning algorithms based on decision trees, which can classify user activity by time and create a prioritized list of tasks, which opens up new opportunities for the development of intelligent personal assistants.

## ЗМІСТ

|   |    |
|---|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів .....                    | 8  |
| Вступ.....  | 9  |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....   | 10 |
| 1.1 Аналіз існуючих сервісів штучного інтелекту .....                                       | 10 |
| 1.2 Порівняння алгоритмів та моделей для створення власного штучного інтелекту.....         | 13 |
| 1.3 Аналіз існуючих методів планування дня .....  | 19 |
| 1.4 Аналіз інтегрованого зі штучним інтелектом інтерфейсу .....                             | 25 |
| 2 НАПРЯМОК ДОСЛІДЖЕННЯ.....   | 32 |
| 2.1 Постановка задачі.....  | 32 |
| 2.2 Дослідження алгоритмів заснованих на дереві рішень.....                                 | 33 |
| 2.2.1 Дерево рішень.....  | 33 |
| 2.2.2 Випадковий ліс .....  | 36 |
| 2.2.3 Градієнтне посилення .....  | 39 |
| 2.3 Висновки дослідження алгоритмів заснованих на дереві рішень.....                        | 43 |
| 3 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ.....  | 46 |
| 3.1 Опис тестового набору даних .....   | 46 |
| 3.2 Математичний опис методики оцінювання результатів експерименту ..                       | 50 |
| 3.3 Результат досліджень .....  | 52 |
| 4 РОЗРОБКА ІНТЕРФЕЙСУ МОБІЛЬНОГО ЗАСТОСУНКУ З ВИКОРИСТАННЯМ МОДЕЛІ ШТУЧНОГО ІНТЕЛЕКТУ ..... | 55 |
| 4.1 Розробка архітектури взаємодії клієнта з моделлю .....                                  | 55 |
| 4.2 Розробка серверу мобільного додатку.....  | 58 |
| 4.3 Розробка інтерфейсу мобільного додатку .....  | 60 |
| 4.4 Порівняння результатів алгоритмів класифікації .....                                    | 62 |
| ВИСНОВКИ.....   | 68 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....   | 70 |
| ДОДАТОК А .....   | 72 |
| ДОДАТОК Б .....   | 82 |
| ДОДАТОК В.....  | 93 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ

ІІІ – штучний інтелект.

CNN – Convolutional Neural Network – конволюційна нейронна мережа.

DNN – Deep Neural Network – глибока нейронна мережа.

RNN – Recurrent Neural Network – рекурентна нейронна мережа.

UML – Unified Modeling Language – уніфікована мова моделювання.

AdaBoost – Adaptive Boosting – адаптивне підсилення.

## ВСТУП

У наш час розвиток технологій з кожним роком збільшується, але особливої популярності набуває сфера розробки штучного інтелекту. З прогресом інформаційних технологій також збільшується попит на споживання більшої кількості інформації. Більшість людей стикались з проблемою відсутності якісного планування дня завдяки якому можна було б залишатись продуктивним більшу частину доби та встигати зробити й вивчити все що було заплановано. У той же час, безліч компаній шукають спосіб збільшити ефективність своїх робітників за рахунок впровадження у свою екосистему програми для покращення балансу роботи й життя.

Всі ці вимоги суспільства визначають високу актуальність проблеми оптимізації планування дня. Самостійно людині складно розробити собі розпорядок на день або навіть на тиждень. Для спрощення планування використовують різні підходи та методології, але вони показують теж невисоку ефективність. Тому, застосування штучного інтелекту для оптимізації планування дня наразі є актуальною задачею для розв'язання.

Метою даної роботи є дослідження застосування методів штучного інтелекту для оптимізації планування дня, а також для підвищення ефективності людей.

Завданнями дослідження є:

- аналіз існуючих алгоритмів, методів та підходів для розробки моделі;
- порівняльний аналіз алгоритмів заснованих на дереві рішень;
- розробка власного набору даних для навчання моделі;
- розробка моделі;
- тестування та оцінювання результатів навчання моделі.

На основі результатів дослідження відкриється можливість для реалізації моделі з використанням нейронної мережі, покращення результатів оптимізації та визначеність у необхідних параметрах набору даних.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Аналіз існуючих сервісів штучного інтелекту

У сучасному світі багато галузей зазнало швидкого розвитку, але є одна з найперспективніших та найпопулярніших на даний момент, застосування штучного інтелекту. Завдяки великій обчислювальній потужності та достатньо якісному прогнозу, машинне навчання стало тим напрямком, що зараз змінює світ та підходи до вирішення певних задач. Більшість власників компаній, зрозумівши потенціал технології, почали вкладати значні кошти та створювати бюджети на розробку штучного інтелекту, який покращить процеси роботи підприємства та збільшить їх прибуток. Даний напрямок має гарну тенденцію, що робить його привабливим для інвесторів.

Явними лідерами в розробці та популяризації штучного інтелекту на даний момент є OpenAI, Google та Midjourney [2]. Потрібно зазначити, що останній відрізняється своєю моделлю та напрямком відносно перших двох компаній. OpenAI – це компанія, що створила одну з найуспішніших моделей штучного інтелекту – ChatGPT. Ця організація також має й інші моделі, такі як DALL-E, що також є достатньо популярною, але через існуючого конкурента Midjourney, вона не так відома серед людей. ChatGPT – це штучний інтелект, заснований на моделі мовного перетворення, який здатен розуміти та генерувати текст на зразок людини. Ця технологія буквально змінила цей світ, підходи до роботи, створення відео матеріалів у соціальних мережах, а також посприяла впливу на інформаційні технології і їх розробку в тому ключі, що тепер все більше компаній переходять на використання штучного інтелекту для створення нових проектів.

Google, ще один ключовий гравець в області штучного інтелекту, заслуговує особливої уваги завдяки своїм розробкам у сфері мовних моделей. Вони створили ряд передових систем, які відрізняються інноваційними підходами до обробки та генерації природної мови. Ці системи здатні не тільки

ефективно розуміти та реагувати на людську мову, але й інтегруватися з широким спектром інших технологій та платформ, що робить їх незамінними у різних областях.

Однією з особливостей розробок Google є їхня здатність до високої інтеграції з іншими продуктами та послугами компанії, такими як пошукові системи, електронна пошта, календарі та інші офісні інструменти. Це дозволяє використовувати штучний інтелект для поліпшення користувацького досвіду, оптимізації процесів та надання більш персоналізованих рекомендацій.

Крім того, Google активно досліджує та розробляє технології у сфері навчання без нагляду та підсилення, які дозволяють мовним моделям навчатися з меншою кількістю анотованих даних. Це відкриває шлях до створення більш гнучких і масштабованих систем, здатних адаптуватися до нових завдань та обставин без необхідності великих обсягів попередньо оброблених даних.

Google також відіграє важливу роль у розробці етичних принципів для штучного інтелекту. Вони зосереджені на створенні безпечних, справедливих та відповідальних ШІ-систем, що враховують потреби та права людей, а також забезпечують прозорість та розуміння їх роботи для користувачів.

Штучний інтелект від Google має значний вплив на різні сфери, включаючи освіту, де він використовується для створення індивідуальних навчальних програм, автоматизації адміністративних завдань та підвищення якості освітнього процесу. У бізнесі, ШІ від Google допомагає компаніям аналізувати великі обсяги даних, прогнозувати тренди, оптимізувати ланцюги поставок та підвищувати загальну ефективність.

У підсумку, внесок Google у розвиток штучного інтелекту є суттєвим і впливовим. Їх інновації та розробки не тільки стимулюють прогрес у галузі ШІ, але й відкривають нові можливості для його застосування у найрізноманітніших сферах, від освіти до бізнесу, що сприяє подальшій інтеграції штучного інтелекту в повсякденне життя.

Midjourney, хоча і є менш відомим, але не менш важливим гравцем у світі штучного інтелекту, заслуговує окремої уваги за свої унікальні та інноваційні розробки. Ця компанія вирізняється своїм акцентом на глибокому навчанні та обробці візуальних даних, відкриваючи нові горизонти у створенні зображень та візуального контенту. Їхні технології, які поєднують елементи штучного інтелекту та комп'ютерного зору, здатні генерувати високоякісні, творчі візуальні образи, що відображають як конкретні запити, так і абстрактні ідеї користувачів.

Особливість Midjourney полягає у їх здатності інтегрувати штучний інтелект у творчі процеси, що розширює границі традиційного підходу до графічного дизайну, ілюстрацій та медіа-контенту. Їхні алгоритми здатні втілювати найбільш амбітні ідеї, перетворюючи звичайні зображення на твори мистецтва, що вражають своєю унікальністю та глибиною.

Midjourney також активно досліджує можливості використання штучного інтелекту в рекламі та маркетингу. Їхні рішення дозволяють автоматизувати та персоналізувати створення рекламних матеріалів, пропонуючи компаніям інноваційні підходи до просування продуктів та послуг. Це, в свою чергу, відкриває нові перспективи для бізнесу, дозволяючи створювати більш цільові та ефективні рекламні кампанії.

У сфері освіти та навчання, Midjourney пропонує унікальні можливості для створення освітніх матеріалів. Їхні технології можуть використовуватися для генерації ілюстрацій, схем, та інших візуальних матеріалів, які покращують зрозуміння складних понять та теорій. Це значно спрощує процес навчання та робить його більш цікавим та ефективним для студентів.

Midjourney також робить внесок у розвиток етичних підходів до використання штучного інтелекту, звертаючи увагу на такі аспекти, як авторське право, конфіденційність та відповідальне використання ШІ-технологій. Це демонструє їхнє прагнення до розвитку відповідального та безпечного штучного інтелекту.

Завдяки своїм інноваціям, Midjourney значно впливає на розвиток штучного інтелекту, відкриваючи нові можливості для його застосування в творчості, бізнесі та освіті. Їхні технології сприяють не тільки технічному прогресу, але й розвитку нових форм творчого вираження та спілкування.

Таким чином, на даний момент існує багато прикладів успішного створення та реалізації штучного інтелекту. Його потенціал у плануванні та оптимізації щоденного життя людини обіцяє ще більші зміни, які можуть радикально трансформувати наші уявлення про роботу, відпочинок та взаємодію з технологіями.

## 1.2 Порівняння алгоритмів та моделей для створення власного штучного інтелекту

Маючи на увазі вже успішні реалізації моделей мовного перетворення та обробки візуальних даних, стає можливим провести глибоке порівняльне дослідження використаних алгоритмів та методів машинного навчання з метою розробки власного штучного інтелекту. На даному етапі основними критеріями вибору слід вважати наступні параметри: ефективність та швидкість навчання, мінімізація ризику недонавчання (underfitting), спрощеність розробки та зрозумілість використання даних. Кожен із цих критеріїв має вирішальне значення для кінцевого успіху дослідження та ефективності роботи розробленої моделі.

Обираючи правильні параметри для аналізу та порівняння, можна значно підвищити шанси на вибір найбільш оптимального варіанту моделі та алгоритму обробки даних. Цей підхід не тільки допоможе забезпечити високу якість кінцевого продукту, але й зробить процес розробки більш ефективним та менш часозатратним.

Серед існуючих алгоритмів та методів машинного навчання можна визначити наступні: нейронні мережі, алгоритми засновані на деревах рішень, системи на основі правил та трансформери.

Нейронні мережі представляють собою потужний клас алгоритмів машинного навчання, які імітують роботу людського мозку для обробки даних та прийняття рішень. Ці системи особливо ефективні в задачах, які вимагають високої ступені гнучкості та адаптивності, таких як розпізнавання образів, обробка природної мови, та прогнозування. Основні типи нейронних мереж включають: глибокі нейронні мережі (Deep Neural Networks, DNNs), конволюційні нейронні мережі (CNNs) та рекурентні нейронні мережі.

Глибокі нейронні мережі. Ці моделі представляють собою складні обчислювальні структури, які складаються з множини послідовно пов'язаних шарів. Кожен шар, що імітує біологічний нейрон, сприяє виявленню та абстракції високорівневих особливостей із вхідних даних, що дозволяє моделі адаптуватися до складності задачі.

Глибокі нейронні мережі знаходять застосування у різноманітних сферах, від комп'ютерного зору та обробки природної мови до складних систем автоматичного водіння та аналітики великих даних. Специфічна здатність цих мереж до ідентифікації патернів у багатовимірних даних робить їх незамінними у задачах, які вимагають глибокого розуміння контекстуальних зв'язків.

Головна перевага DNN's полягає у їх здатності моделювати складні, нелінійні відносини в даних, що дозволяє досягати високої точності в прогнозуванні та класифікації. Проте, це супроводжується підвищеними вимогами до обчислювальних ресурсів та даних для навчання, а також потенційним ризиком перенавчання при недостатньому або неякісному датасеті.

У рамках теми DNN's можуть бути застосовані для ефективного аналізу поведінкових патернів та уподобань користувачів. Водночас, потреба у великій кількості персоналізованих даних та високі обчислювальні витрати можуть ускладнити їх практичне впровадження у контексті задач оптимізації повсякденного планування.

Конволюційні нейронні мережі є особливою категорією глибоких нейронних мереж, які використовують конволюційні шари для автоматичного та ефективного виявлення важливих особливостей у вхідних даних. Вони містять фільтри, які пересуваються по вхідному простору даних (наприклад, по зображеннях), що дозволяє ефективно виявляти шаблони, такі як краї, кути та інші візуальні характеристики.

CNNs широко застосовуються в області комп'ютерного зору для задач, що включають розпізнавання образів, обробку зображень, та відеоаналітику. Їх здатність до ідентифікації візуальних патернів робить їх незамінними у таких застосуваннях, як автоматичне водіння, медична діагностика зображень та автоматизований аналіз змісту відео.

CNNs вирізняються високою точністю у виявленні візуальних особливостей та ефективністю у великих датасетах зображень. Однак, вони вимагають значної обчислювальної потужності для навчання та можуть бути складними у налаштуванні для специфічних застосувань.

Для використання даної моделі в рамках теми конволюційні нейронні мережі можуть бути використані для аналізу візуальних даних, пов'язаних з поведінковими патернами користувачів, наприклад, через аналіз зображень з їхнього повсякденного життя. Проте, у рамках дослідження не буде використано або розроблено інтеграцію з візуальними даними, тому ця модель не підходить для подальшої роботи та використання її у розробці.

Рекурентні нейронні мережі розроблені для обробки послідовностей даних, здатні зберігати інформацію про попередні входи у вигляді внутрішнього стану (пам'яті). Вони досягають цього завдяки наявності петель у своїх нейронних вузлах, що дозволяє передавати інформацію від одного кроку обробки до наступного, імітуючи тимчасову послідовність.

RNNs широко застосовуються у сферах, де важлива послідовна структура даних, як-от у обробці природної мови для машинного перекладу, генерації тексту та системах розпізнавання мови. Також вони ефективні у моделюванні

часових рядів, що корисно у фінансовому прогнозуванні, аналізі сигналів та інших застосуваннях, де дані представлені у формі послідовностей.

Основною перевагою RNN є їх здатність ефективно обробляти послідовні дані, зберігаючи контекстуальну інформацію протягом довгих послідовностей. Проте, вони стикаються з проблемами, як-от зникнення або вибух градієнтів при тренуванні на довгих послідовностях, що ускладнює навчання.

У рамках дослідження, RNNs можуть бути використані для аналізу поведінкових послідовностей та прогнозування звичок користувачів. Однак, потенційні труднощі з обробкою дуже довгих або складних послідовностей можуть вимагати використання більш розвинутих варіантів RNN, таких як LSTM або GRU, для забезпечення стабільності та ефективності в моделюванні.

Проте, важливо врахувати, що розробка та налаштування нейронних мереж вимагає значних зусиль, часу та експертизи, особливо для створення власного рішення з нуля. Необхідно забезпечити великий набір даних для навчання, а також ретельно оптимізувати параметри мережі, що може бути особливо складно без глибоких знань у галузі. Окрім того, такі моделі вимагають значних обчислювальних ресурсів, які можуть бути дорогі та часозатратними.

Враховуючи ці фактори, можна зробити висновок, що використання нейронної мережі один з найкращих варіантів для створення комплексного штучного інтелекту, що складає план на день з отриманих від користувача даних. Також, потрібно звернути увагу на те, що розробка нейронної мережі на початкових етапах буде дуже складним та не доцільним рішенням.

Окрім нейронних мереж для створення штучного інтелекту часто використовуються алгоритми засновані на деревах рішень. Такі метод мають перевагу у швидкості навчання на певних наборах даних та невеликі за часом розробки. Прикладами подібних алгоритмів є: Випадковий ліс (Random Forest), Дерево рішень (Decision Tree) та Градієнтне посилення (Gradient Boosting).

Дерево рішень: цей алгоритм працює шляхом розбиття даних на підмножини на основі певних критеріїв, створюючи деревоподібну структуру, де кожен вузол представляє рішення або розгалуження. Древа рішень прості у розумінні та інтерпретації, але мають схильність до перенавчання, особливо в разі складних даних.

Випадковий ліс (Random Forest): даний алгоритм є ансамблем дерев рішень, який працює шляхом створення багатьох дерев рішень і вибору найкращого рішення за допомогою голосування або усереднення їхніх результатів. Випадкові ліси зменшують ризик перенавчання та часто показують кращу загальну продуктивність порівняно з одиничними деревами рішень.

Градiєнтне посилення (Gradient Boosting): Boosting – це метод, що поєднує слабкі моделі (часто дерева рішень) в сильну ансамблеву модель, де кожна наступна модель виправляє помилки попередньої. AdaBoost (Adaptive Boosting) адаптується до помилок, збільшуючи вагу важких випадків, тоді як Gradient Boosting оптимізує втрати за допомогою градієнтного спуску, що робить його ефективним для широкого спектра задач.

У контексті дослідження методів штучного інтелекту для оптимізації планування дня користувача, вибір відповідного алгоритму машинного навчання відіграє ключову роль. Древа рішень та їх ансамблі, такі як випадковий ліс та методи посилення, можуть бути використані для визначення оптимальних планів дій на основі великої кількості факторів та історичних даних. Ці методи здатні аналізувати складні взаємозв'язки та забезпечувати виведення конкретних рекомендацій, що є важливим для ефективного планування.

Алгоритми на основі дерев рішень відрізняються своєю здатністю до інтуїтивно зрозумілої візуалізації та легкості інтерпретації, що робить їх ідеальними для сценаріїв, де необхідно пояснити прийняті рішення користувачам.

Застосування цих алгоритмів у сфері планування дня може включати автоматизацію розкладу, прогнозування найбільш продуктивних часових слотів для виконання завдань, аналіз звичок користувачів для рекомендації змін у розпорядку дня тощо. Важливо зазначити, що успішне впровадження цих методів вимагає ретельного збору та аналізу даних, а також постійного налаштування моделі, щоб враховувати змінні потреби та умови користувачів.

Також, серед алгоритмів та методів є ще такі як: системи на основі правил та трансформери.

Системи на основі правил – це використання примітивного способу реалізації штучного інтелекту на основі (if ... else). Такі системи підходять для використання їх у невеликих задачах та не гнучких, або не масштабуючих рішеннях, де потрібні добре визначені правила та легке їх формулювання. У випадку зі складним проектом створення щоденного плану, такий алгоритм не підходить так як неможливо створити певні правила, що будуть підходити під усі випадки, або хоча б більшість з них. Таким чином, це рішення може бути доцільним лише для демонстрації майбутньої реалізації штучного інтелекту.

Трансформери. Ці моделі, особливо в мовному перетворенні, виявилися дуже ефективними в задачах обробки природної мови. Вони можуть бути доцільними у використанні їх разом з нейронною мережею для генерації відповідей на питання користувача. Трансформери використовуються для перекладу тексту між мовами, забезпечуючи високу точність та збереження контексту, також ці моделі можуть створювати зрозумілі та зв'язні тексти, що робить їх корисними для автоматичного створення медійної інформації. Основуючись на дослідженні методів для оптимізації щоденного плану дня, можна сказати, що трансформери мають інший шлях та сферу застосування.

На основі проведеного аналізу моделей, методів та алгоритмів машинного навчання можна визначити кілька ключових висновків. Нейронні мережі демонструють значну потужність у аналізі та обробці великих обсягів даних, включаючи комплексний аналіз візуальних даних та опрацювання природної мови. Проте, ураховуючи їх високі вимоги до обчислювальних ресурсів та

професійних знань, використання нейронних мереж може бути недоцільним на початкових етапах розробки штучного інтелекту для даного дослідження.

При розгляді алгоритмів на основі дерев рішень і аналізі їх сфери використання, переваг та недоліків, було встановлено, що ці методи найбільш підходять для реалізації штучного інтелекту в контексті щоденного планування дня. Дерево рішень може ефективно аналізувати та класифікувати поведінкові патерни користувачів. Їхня здатність до швидкого аналізу та простоти інтерпретації робить їх оптимальними для розробки прототипів та на початкових етапах розробки. Цей алгоритм відповідає усім вимогам для реалізації штучного інтелекту в рамках даного дослідження, зокрема простотою у розробці, відсутністю потреби у великих обсягах даних та ефективністю навіть при обмеженій кількості даних.

Також був проведений аналіз систем на основі правил. Цей підхід виявився зручним для розв'язання простих задач і має зрозумілу структуру розробки. Такий метод може бути застосований для демонстрації базової роботи штучного інтелекту, ілюструючи його потенційну функціональність. Проте, необхідно зауважити, що цей підхід має істотні недоліки, які обмежують його використання для подальшої розробки. Системи на основі правил не відрізняються гнучкістю і є складними у масштабуванні через вимоги до чітко визначених правил, що унеможлиблює їх використання як основ для більш складних систем машинного навчання.

### 1.3 Аналіз існуючих методів планування дня

У сучасному світі багато людей прагнуть зберегти високу продуктивність протягом тривалого часу або мати добре продуманий план на тиждень. У відповідь на ці потреби, людство розробило множину практик, систем та методологій для створення ефективного розпорядку дня. Багато з цих методів мають унікальний характер, специфіку та складності в реалізації. Вибір

конкретної методології залежить від індивідуальних потреб людини, доступного часу для планування та інших факторів.

В найдавніші часи людської історії, планування дня було тісно пов'язане з природними циклами, такими як чергування дня і ночі, а також з сезонними змінами. Люди орієнтувалися на сонце для визначення оптимального часу для полювання, збору харчів та інших повсякденних діяльностей.

З появою сільського господарства та формуванням перших цивілізацій з'явилася потреба у більш складному плануванні. Це стосувалося розподілу сезонних робіт, будівництва, торгівлі, а також проведення релігійних ритуалів. У цей період були створені перші календарі, сприяючи кращій організації та плануванню цих процесів.

Розвиток наук, особливо астрономії та математики, а також поява друкарства забезпечили створення більш точних календарів та годинників, що підвищило точність планування дня, дозволяючи краще організувати як робочий час, так і відпочинок.

Промислова революція внесла суттєві зміни у планування дня. Робочий час став більш регульованим та стандартизованим, з'явилися розклади роботи на заводах, розклади поїздів та інших видів транспорту.

Розвиток технологій та наукового менеджменту призвів до подальшої систематизації планування дня. Були розроблені різноманітні методики ефективного управління часом, які знаходили застосування як у бізнесі, так і в особистому житті. Поява комп'ютерних технологій та мобільних пристроїв значно спростила та автоматизувала багато аспектів планування.

У сучасному світі планування дня перетворилося на витончену практику з використанням різноманітних цифрових інструментів, включаючи календарі, планувальники, мобільні додатки та штучний інтелект, що дозволяє не лише планувати, але й оптимізувати розклад, адаптуючи його до індивідуальних потреб і змінних умов.

У контексті активної популяризації технологій спостерігається створення різноманітних додатків, веб-сайтів та мобільних застосунків, які надають

користувачам можливість формувати плани на день або тиждень, використовуючи специфічні методології, або створювати прості списки завдань. На сьогоднішній день більшість смартфонів оснащені базовими застосунками для створення списків задач, що дозволяють планувати активності на майбутнє.

Проте, ці базові інструменти часто мають обмежений функціонал, зазвичай обмежуючись простим створенням задач без можливостей налаштування повторюваності або адаптації під індивідуальні потреби користувача. Це породило потребу у розробці спеціалізованих додатків, які пропонують більш широкий спектр функціональності, включаючи підтримку різних методологій та систем планування. Такі застосунки надають користувачам можливості для більш детального та гнучкого управління своїми завданнями та часом.

Завдяки цьому, сучасні технології планування дня стають все більш інтегрованими та адаптивними до потреб сучасного користувача, надаючи інструменти для ефективного управління часом та завданнями у швидкому та динамічному світі.

Більшість корпорацій прагнуть інтегрувати специфічні методології чи системи у свою організаційну структуру з метою підвищення ефективності та продуктивності своїх співробітників. Такі підходи забезпечують структурованість робочого процесу та сприяють збалансуванню професійної діяльності з особистим життям.

Як приклад, можна навести компанію ЕРАМ, яка активно використовує власні ресурси для формування кодексу та етики серед своїх співробітників. У рамках своєї освітньої ініціативи, ЕРАМ регулярно організовує набір студентів до своєї корпоративної школи та дослідницької лабораторії. Студенти, які успішно проходять вступні екзамени, продовжують навчання, на етапі якого отримують набір подарунків. Серед цих подарунків є спеціально розроблений блокнот, який містить рекомендації щодо планування дня, спрямовані на підтримку здорового способу життя та балансу між роботою та особистим

часом («Work-Life Balance»). Цей аспект також акцентується під час вступних лекцій.

Такий підхід демонструє, як корпорації можуть використовувати структуровані методології планування для підвищення загальної ефективності та добробуту своїх співробітників, одночасно враховуючи їхні індивідуальні потреби та сприяючи створенню гармонійного робочого середовища.

Для глибшого розуміння потреб індивідів та забезпечення якісних результатів, було здійснено всебічний аналіз популярних методологій та систем планування. Цей аналіз охоплює вивчення різних підходів до управління часом та завданнями, враховуючи їх здатність задовольняти конкретні потреби користувачів. Метою цього аналізу є визначення ключових факторів, що впливають на ефективність різних методологій, з метою розробки більш адаптивних і ефективних систем планування.

Метод пріоритезації задач базується на систематичному розподілі завдань за ступенем їх важливості та терміновості. Він включає оцінювання кожної задачі з погляду її впливу на загальні цілі та визначення оптимального часу для її виконання. Цей підхід дозволяє раціонально розподіляти ресурси та час, сконцентрувавши увагу на найбільш значущих завданнях.

Метод пріоритезації задач широко використовується у різних сферах, але особливо популярний у сфері ІТ, де ефективно управління часом є ключовим аспектом продуктивності. В ІТ-секторі, де проекти часто характеризуються високою складністю та обмеженими термінами, цей метод дозволяє максимально ефективно розподілити робочий час та ресурси команди.

Основні переваги методу полягають у його спроможності забезпечити чіткий фокус на ключових задачах та підвищити загальну ефективність роботи. Проте, він може бути складним у впровадженні при відсутності чіткого розуміння стратегічних цілей та пріоритетів. Також може виникнути ризик знехтування менш терміновими, але довгостроково важливими завданнями.

Для оптимізації щоденного планування, метод пріоритезації задач може бути вкрай корисним. Він може допомогти в ідентифікації найбільш значущих

завдань, на які слід сконцентрувати ресурси та час. Однак, існує потенційний ризик недостатньої адаптації до змінних обставин чи нових інформаційних потоків, який може вплинути на гнучкість та адаптивність планування.

Метод «Time Blocking» полягає в сегментуванні робочого дня на визначені блоки часу, призначені для конкретних завдань або активностей. Цей підхід передбачає строге відведення певного періоду часу для кожної задачі, що дозволяє індивідуально зосередитися на кожній з них без перерв або переключення на інші справи. Він включає детальне планування та алокацію часу, що може варіюватися від години до цілого дня.

«Time Blocking» здобув широку популярність серед професіоналів у різних галузях, особливо у сфері ІТ, де потреба у фокусуванні на складних завданнях є високою. Цей метод використовується для оптимізації робочих процесів, збільшення продуктивності та кращого управління часом, особливо в умовах високої завантаженості та множинності проектів.

Основні переваги «Time Blocking» полягають у забезпеченні більшої концентрації на завданнях та ефективності роботи, а також у зниженні рівня відволікань. Проте, метод може бути менш ефективним у ситуаціях, коли потрібна висока гнучкість або зустрічаються непередбачені завдання. Також цей метод вимагає строгого дотримання запланованого графіку, що може бути викликом для деяких індивідів.

У рамках дослідження методів штучного інтелекту для оптимізації планування дня, «Time Blocking» може виявитися корисним для підвищення продуктивності та ефективності роботи. Використання цього методу може допомогти в ідентифікації та алокації часу для конкретних завдань з ШІ, сприяючи більш глибокому зосередженню на складних аспектах роботи. Однак, потреба у строгому дотриманні часових рамок може обмежувати гнучкість у випадку непередбачених змін чи термінових завдань.

На основі проведеного аналізу існуючих методів планування дня, було прийнято рішення інтегрувати системи «Time Blocking» та «Пріоритезація Задач» у єдину методологію. Ця інтеграція практик спрямована на суміщення

переваг обох методів для підвищення ефективності та якості планування. Такий підхід має на меті забезпечити більш точне та гнучке структурування розпорядку на день або навіть на тиждень.

Головні переваги такого інтегрованого підходу включають гнучкість, точність та можливість детального налаштування кожного сегменту дня. Користувачі зможуть чітко розуміти, що початок кожного сегменту повинен містити найважливіші завдання, тоді як завдання, розташовані ближче до кінця сегменту, можуть бути відкладені у разі нестачі часу або необхідності підготовки до наступного сегменту.

Водночас, інтеграція цих методів передбачає і взаємні недоліки. Користувачам може бути складніше визначити пріоритети завдань у межах певного сегменту або вирішити, в який сегмент розмістити конкретне завдання. Крім того, детальне розподілення дня на численні сегменти може призвести до створення надмірно складного та перевантаженого розкладу, де більшість завдань мають високий пріоритет.

Для вирішення цих викликів, важливо розробити механізм, що дозволяє уникнути необхідності повністю самостійного створення плану на день. В цьому контексті, штучний інтелект може стати ефективним інструментом, який допоможе користувачам оптимізувати свій розпорядок, враховуючи їхні індивідуальні потреби та пріоритети.

В контексті аналізу даної методології, можна зробити висновок про доцільність її інтеграції з технологіями штучного інтелекту. Штучний інтелект, який навчається на основі зібраних даних, може ефективно синтезувати інформацію, отриману через користувацький інтерфейс, та формувати оптимізований розпорядок дня. Такий підхід забезпечує користувачам можливість внесення власних корективів у запропонований розклад.

Ключовою метою цієї інтеграції є значне підвищення продуктивності людини через розробку найбільш ефективного варіанту розпорядку дня. Штучний інтелект в цьому контексті виступає як потужний інструмент, який адаптує методологію планування до індивідуальних потреб та умов

користувача, забезпечуючи високий рівень персоналізації та гнучкості у процесі планування.

#### 1.4 Аналіз інтегрованого зі штучним інтелектом інтерфейсу

Штучний інтелект (ШІ) представляє собою винятково потужний інструмент, чия значимість та ефективність у сучасному світі важко переоцінити. Наразі, технології ШІ демонструють найвищий рівень популярності серед сучасних технологічних розробок. Важливим аспектом ефективного використання алгоритмів машинного навчання є наявність інтуїтивно зрозумілого візуального інтерфейсу для збору та обробки даних.

Взаємодія користувача з алгоритмом ШІ значною мірою залежить від якості та функціональності інтерфейсу. З огляду на мету даного дослідження, доцільно провести детальний аналіз оптимального інтерфейсу для інтеграції штучного інтелекту. Такий підхід дозволить забезпечити максимальну ефективність взаємодії між користувачем та системою, сприяючи досягненню поставлених цілей дослідження.

У сучасному цифровому світі існує множинність способів взаємодії з програмним забезпеченням, включаючи веб-сторінки, мобільні телефони, планшети та персональні комп'ютери. Ці пристрої стали невід'ємною частиною повсякденного життя, оскільки велика кількість людей володіє принаймні одним з цих пристроїв. Через ці девайси користувачі взаємодіють з різноманітними сервісами та додатками.

Кожен тип пристрою має свою визначену цільову аудиторію і часто використовується для специфічних видів діяльності. Наприклад, персональні комп'ютери забезпечують розширені можливості для перегляду веб-сторінок та користування програмами, розробленими для відповідних операційних систем.

Важливим етапом у виборі інтерфейсу є аналіз способів використання цих пристроїв користувачами. Розуміння переваг і особливостей взаємодії з

різними типами пристроїв дозволить розробити більш ефективні та зручні для користувачів інтерфейси, які відповідають їхнім потребам та звичкам.

У сучасному світі широко поширено використання різноманітних мобільних пристроїв, особливо смартфонів, які стали невід'ємною частиною повсякденного життя. Аналіз різних джерел дозволяє оцінити масштаби їх використання у порівнянні з іншими пристроями.

Згідно з даними "The Global Media Intelligence Report 2023", в США час, проведений з мобільними пристроями, вперше перевищив час використання ПК/ноутбуків/планшетів. Середній час використання мобільних пристроїв склав 3 години 37 хвилин на добу, що на 1 хвилину більше, ніж час, проведений з ПК/ноутбуками/планшетами. Водночас, в Канаді власність смартфонів є майже універсальною, але час, проведений з мобільними медіа, залишається нижчим порівняно з іншими аналогічними країнами.

Дослідження веб-сайту Zippia показує, що 81.6% американців володіють смартфонами станом на 2023 рік. Середній американець проводить близько 5 годин 24 хвилин на день, користуючись своїм мобільним пристроєм. Це свідчить про інтенсивне використання смартфонів у повсякденному житті. Загальна кількість користувачів смартфонів по всьому світу оцінюється приблизно в 6.92 мільярди.

В Європі, станом на серпень 2023 року, розподіл частки ринку серед пристроїв показує, що мобільні пристрої лідирують з показником 51.89%. Настільні комп'ютери займають друге місце з часткою 45.64%, а планшети становлять решту відсотків. Це свідчить про зростаючу перевагу мобільних пристроїв у регіоні.

З цих даних можна зробити висновок, що у сучасному світі смартфони та інші мобільні пристрої мають вищий рівень використання порівняно з настільними комп'ютерами, планшетами та іншими пристроями. Вони впливають на повсякденне життя людей, спосіб взаємодії з цифровим світом та є ключовими інструментами у повсякденному плануванні та виконанні завдань.

На основі аналізу ринку мобільних операційних систем станом на жовтень 2023 року можна зробити висновки щодо популярності платформ Android і iOS. Згідно з даними Statcounter, Android має частку ринку 69.64%, тоді як iOS займає 29.67%. Ці дані свідчать про перевагу Android над iOS у глобальному масштабі, що може бути пов'язано з більшою доступністю та різноманітністю пристроїв на платформі Android. Однак, важливо враховувати, що розподіл ринку може варіюватися в залежності від географічного регіону та специфіки ринку.

Результати аналізу використання мобільних пристроїв свідчать про їх домінуюче положення в цифровому ландшафті. Ці дані надають цінну інформацію щодо вибору типу пристроїв, на які слід орієнтуватися при розробці мобільних додатків. Крім того, детальний аналіз ринкових часток операційних систем iOS та Android показав, що платформа від Google використовується приблизно на 40% частіше, ніж система від Apple. Ці відомості забезпечують ключовий внесок у вибір оптимального користувацького досвіду при інтеракції з додатками. Така інформація може бути важливою при інтеграції штучного інтелекту, оскільки вона впливає на ефективність та результативність такої інтеграції.

У контексті розробки інтерфейсів для взаємодії зі штучним інтелектом, що створює розпорядок на день, важливо розглянути аспекти реалізації крос-платформних мобільних застосунків у порівнянні з веб-сторінками.

Крос-платформні мобільні застосунки - це програми, які розроблені таким чином, що вони можуть працювати на різних операційних системах (наприклад, Android і iOS) з використанням єдиного базового коду. Це дозволяє розробникам створювати додатки, які сумісні з множиною пристроїв, не вдаючись до написання окремого коду для кожної платформи.

Переваги крос-платформного додатку включають зниження витрат і часу на розробку, оскільки розробникам потрібно писати менше коду. Також це забезпечує ширший охоплення ринку, оскільки додаток доступний для більшої кількості користувачів. Однак, можуть виникати компроміси щодо

продуктивності та доступу до специфічних функцій платформи, а також потенційні труднощі з уніфікацією інтерфейсу користувача на різних пристроях.

Використання крос-платформного додатку як інтерфейсу для взаємодії з штучним інтелектом у контексті оптимізації планування дня користувача має свої переваги. Воно забезпечує доступність системи на широкій гамі пристроїв, сприяючи більш широкому впровадженню та прийняттю користувачами. Однак, можуть виникати виклики з оптимізацією продуктивності та інтеграцією з функціями конкретних пристроїв, що може обмежити повний потенціал штучного інтелекту в деяких сценаріях використання.

Веб-сторінки, як крос-платформний інтерфейс, означають, що одна й та сама веб-сторінка може бути доступна та функціональна на будь-якому пристрої з веб-браузером, незалежно від його операційної системи. Це досягається за допомогою використання стандартів веб-розробки, таких як HTML, CSS та JavaScript.

Веб-сторінки як крос-платформний інтерфейс пропонують значну універсальність та доступність, дозволяючи користувачам легко доступити додаток з будь-якого пристрою. Вони не вимагають встановлення додаткового програмного забезпечення та спрощують процес оновлень. Однак, можуть мати обмеження в продуктивності та інтерактивності, особливо у випадку складних або ресурсоємних додатків.

Використання веб-сторінок як інтерфейсу для взаємодії з штучним інтелектом у контексті "Дослідження методів штучного інтелекту для оптимізації планування дня користувача" має значні переваги. Веб-інтерфейси забезпечують легкий доступ до інструментів ШІ з будь-якого пристрою, що має інтернет-з'єднання. Однак, це може обмежити використання деяких функцій штучного інтелекту, які вимагають високої обчислювальної потужності або специфічних функцій пристрою.

На основі проведеного аналізу реалізації, можливостей, переваг та недоліків веб-сторінки та крос-платформного мобільного застосунку, можна зробити висновок. Розробка веб-сторінки є більш простим варіантом з точки зору затрат часу та ресурсів, а також забезпечує доступність на різних пристроях, потенційно збільшуючи охоплення аудиторії. Серед недоліків потрібно зауважити, що користувацький досвід веб-сторінок може бути обмежений, особливо для постійного відстеження змін у виконаних завданнях. Мобільний застосунок, особливо крос-платформний є більш складним у розробці, а також має певні недоліки через потенційні конфлікти у реалізації певної функціональності на різні платформи. Перевагами такого підходу можуть бути кращий користувацький досвід, так як мобільний телефон майже завжди поруч і якщо потрібно подивитись план на день або зробити зміни, то кращим варіантом буде мобільний застосунок. Він має більш стабільну роботу, а також може працювати без підключення до інтернету.

Аналіз причин, чому компанія OpenAI спочатку запустила ChatGPT як веб-сайт, а потім у форматі мобільного застосунку, можна провести, виходячи з кількох ключових аспектів.

Запуск ChatGPT спочатку як веб-сайту дозволив OpenAI швидко та ефективно тестувати та оптимізувати свій продукт. Веб-платформи забезпечують гнучкість у внесенні змін та оновленнях, що є критично важливим для швидко еволюціонуючого ШІ-продукту.

Веб-сайти є легко доступними для широкого спектру користувачів незалежно від операційної системи або пристрою. Це дозволило OpenAI забезпечити максимально широке охоплення і залучити велику аудиторію для отримання зворотного зв'язку та використання продукту.

Розробка мобільного застосунку вимагає додаткових ресурсів та зусиль, пов'язаних із оптимізацією продуктивності, забезпеченням сумісності на різних пристроях та операційних системах. Випуск ChatGPT спочатку на веб-платформі дозволив компанії OpenAI зосередитися на розробці основного

функціоналу та його удосконаленні перед розширенням на мобільні платформи.

Ці аспекти свідчать про стратегічний підхід OpenAI до випуску та розвитку ChatGPT, де первинний акцент був зроблений на тестуванні, оптимізації та масштабуванні продукту через веб-платформу, а подальше розширення на мобільні застосунки було здійснене після забезпечення стабільності та ефективності основного продукту.

Аналіз рішення компанії Midjourney зробити реліз сервісу у Discord замість створення веб-додатку або мобільного застосунку.

Спочатку використання Discord не було запланованим. Команда Midjourney працювала віддалено та використовувала Discord для внутрішнього налагодження системи. Вони виявили, що бот Midjourney дозволяє їм ефективніше співпрацювати, надсилаючи запити до бота та одночасно бачити результати.

Під час тестування бота на Discord з групою користувачів команда Midjourney виявила, що він користується великим успіхом. Користувачі полюбили спільне відкриття та очікування на появу своїх зображень, які відображалися для інших учасників групи.

Дейвід Гольц, засновник Midjourney, не вважає Midjourney звичайним чат-ботом. Він бачить його як інструмент, який допомагає людям досягти своїх цілей (відповідно до назви Midjourney). Гольц вважає соціальний аспект критично важливою частиною досвіду Midjourney.

Ці фактори пояснюють, чому Midjourney зробив ставку на Discord як основний канал взаємодії з користувачами, замість створення традиційних веб-сайтів чи мобільних застосунків.

Отже, після проведеного аналізу інтеграції штучного інтелекту з різними платформами, визначення популярності пристрою та операційної системи, можна зробити висновок. Веб-додаток є більш вигідним варіантом розробки та початкового запуску, цей підхід може охопити більшу кількість людей та дозволяє проводити ефективну оптимізацію за рахунок великої кількості

користувачів. Мобільний застосунок можна виділити як додатковий спосіб поширення продукту, бо розробка займає багато ресурсів та зусиль, що може виявитись потенційною проблемою. Потрібно зауважити, що так як ChatGPT це штучний інтелект для розпізнавання природньої мови у вигляді тексту, а також генерації відповідним способом, то веб-додаток у випадку OpenAI був об'єктивно кращим варіантом. На основі даних статті [1] користувацький досвід з використанням тексту, особливо набиранням великої його кількості, є більш сприятливим на таких пристроях як персональні комп'ютери. В нашому випадку дослідження методів штучного інтелекту для оптимізації планування дня головним є взаємодія з отриманим результатом у вигляді списку задач, які користувач постійно буде переглядати. Тому, було вирішено розробити крос-платформний мобільний застосунок.

## 2 НАПРЯМОК ДОСЛІДЖЕННЯ

### 2.1 Постановка задачі

Метою даного дослідження є вибір та адаптація оптимального алгоритму машинного навчання для ефективного щоденного планування. Дослідження зосереджується на розробці власної методології, яка дозволить ідентифікувати найбільш підходящий метод штучного інтелекту для цієї задачі, враховуючи актуальність та перспективність вибраних підходів.

На даний момент основними викликами є вибір відповідного методу штучного інтелекту для початкового етапу реалізації, а також визначення способу його застосування в контексті планування дня.

Задачі дослідження:

- провести детальний аналіз існуючих алгоритмів машинного навчання, зокрема дерев рішень, таких як градієнтне посилення, випадковий ліс та дерево рішень. Оцінити їхню придатність для оптимізації планування дня;
- розробити інтерфейс для взаємодії користувача з системою штучного інтелекту, з врахуванням висновків, зроблених у розділі "Аналіз інтегрованого зі штучним інтелектом інтерфейсу";
- створити прототип системи штучного інтелекту, базуючись на результаті проведеного аналізу, з метою його подальшого тестування та вдосконалення.

Дослідження буде включати аналіз наукових статей та публікацій, що демонструють застосування алгоритмів машинного навчання у сферах, наближених до контексту планування дня. Також буде використано емпіричні методи для розробки та тестування прототипу.

Планується, що результати дослідження допоможуть в подальшому розробити мобільний застосунок як інтерфейс для взаємодії користувачів із розробленою системою штучного інтелекту для щоденного планування дня або тижня.

## 2.2 Дослідження алгоритмів заснованих на дереві рішень

У сфері розробки штучного інтелекту застосовується широкий спектр алгоритмів та підходів, серед яких особливе місце займають методи, базовані на алгоритмі «Дерево рішень». Згідно з поверхневим аналізом у розділі «Порівняння алгоритмів та моделей для створення власного штучного інтелекту», було встановлено, що алгоритми цього типу відповідають ключовим параметрам дослідження. Однак, для точного визначення найбільш ефективного підходу на початковому етапі розвитку ШІ необхідно провести глибоке вивчення. У фокусі аналізу знаходяться такі алгоритми, як «Дерево рішень», «Гرادієнтне посилення» та «Випадковий ліс».

### 2.2.1 Дерево рішень

До нашого часу було створено безліч різних алгоритмів, що мають одну спільну мету – оптимізувати певний процес. Розвиток методу дерева рішень почався ще задовго до винайдення комп'ютерів, але у ті часи він мав архаїчну структуру, що мало схожа на сучасний вид.

«Категорії» Аристотеля, ієрархічна структура людського мовлення та подій, ілюструють ранню форму систематичної категоризації, схожу на основи дерев рішень. Ця робота демонструє архаїчну версію алгоритмічної логіки, необхідної для еволюції дерев рішень [2].

Стаття Вільяма А. Белсона 1959 року «Зіставлення та прогнозування за принципом біологічної класифікації» часто вважається основою сучасних алгоритмів дерева рішень. У цій роботі були представлені методи класифікації та прогнозування з використанням деревоподібних структур, які є важливими для поточного використання дерев рішень у машинному навчанні [2].

Дерево рішень є одним з найпопулярніших алгоритмів сьогодення, який використовується не тільки в машинному навчанні, а й для вирішення інших задач та проблем.

Алгоритм дерево рішень використовується у різних сферах, включаючи фінанси, медицину, інженерію, дослідження споживчих ринків та інші. Його гнучкість та здатність до моделювання різноманітних сценаріїв робить його ідеальним для аналізу рішень у цих областях.

Даний алгоритм має домінуюче положення у сфері бізнес-аналітики у порівнянні з іншими підходами. Дерево рішень використовується для аналізу бізнес-процесів та для моделювання різних сценаріїв прийняття рішень. Це може включати стратегічне планування, оптимізацію ресурсів, аналіз ризиків і визначення найбільш ефективних стратегій дій. Більш того, він зайняв певне місце також в інших сферах, які було наведено вище.

Основою на меті дослідження, потрібно проаналізувати доцільність його використання для створення оптимального плану на день. Для того, щоб дізнатися, чи підходить цей метод, потрібно розглянути його структуру, алгоритм побудови та провести оцінку.

Алгоритм дерево рішень складається зі специфічної структури, яка включає вузли та гілки. Розглянемо основні елементи структури:

– кожен вузол в Дереві рішень представляє певний тест або умову, що має бути перевірена. Наприклад, вузол може містити питання «Чи велике число X?». Гілки вказують на можливі відповіді або наступні дії, залежно від результату тесту. Наприклад, якщо відповідь на питання «Чи велике число X?» є «Так», то дерево перейде до наступного вузла, якщо відповідь є «Ні», то до іншого;

– вхідні дані, що використовуються для побудови «Дерева рішень», можуть бути представлені у вигляді таблиці або матриці. Кожний рядок таблиці представляє окремий об'єкт, а кожний стовпець вказує на певну властивість цих об'єктів. Наприклад, у рамках теми дослідження, вхідні параметри можуть включати наявність занять спортом, захоплення, роботу, освіту тощо;

– критерії розгалуження визначають, які тести або умови слід використовувати для прийняття рішень при побудові «Дерева рішень».

Наприклад, критерієм може бути ентропія або чистота вузлів, які оцінюються для вибору найкращого розгалуження.

Метод машинного навчання «Дерево рішень» побудоване за алгоритмом, який включає кілька етапів. На рис. 2.1 зображено просту візуалізацію структури алгоритму.

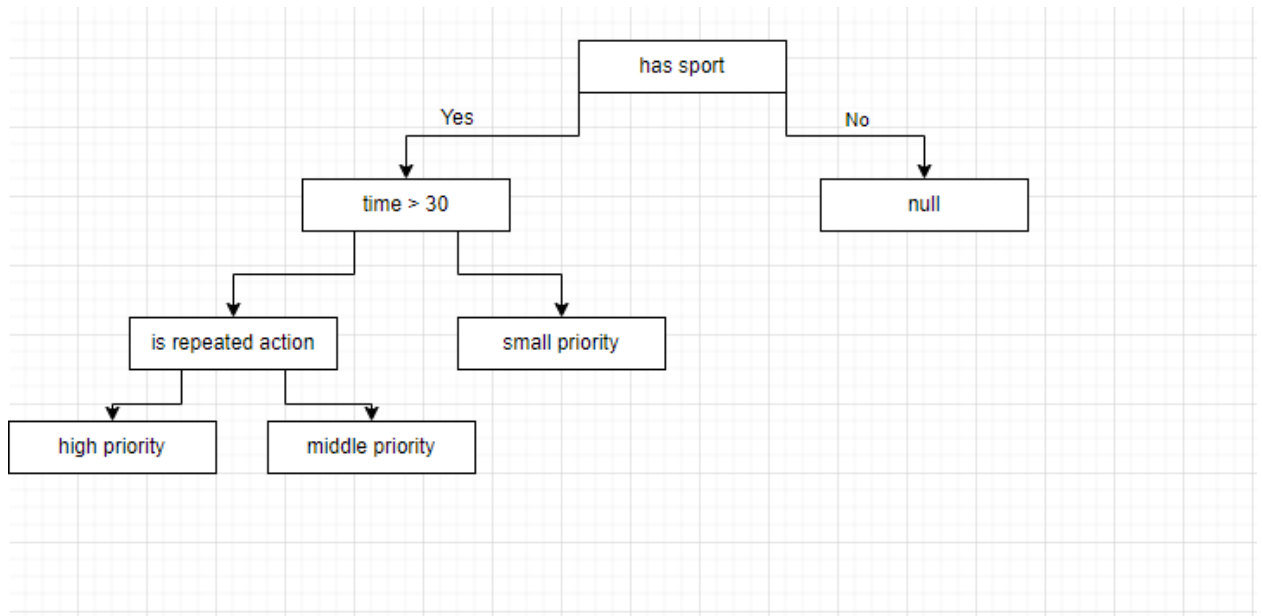


Рисунок 2.1 – Структура алгоритму «Дерева рішень». Вузли та гілки

Початковий вузол дерева, відомий як кореневий вузол, обирається на основі критеріїв розгалуження. Вибір кореневого вузла визначається таким чином, щоб максимізувати інформаційну цінність або зменшити ентропію системи.

Після вибору кореневого вузла, рекурсивний процес розгалуження починається. Кожен вузол дерева розгалужується на основі певного тесту або умови. Наприклад, вузол може розгалужуватись на дві гілки – «Так» і «Ні», в залежності від заданого критерію.

Рекурсивний процес розгалуження продовжується досягнення певної умови зупинки. Ця умова може включати досягнення певної глибини дерева, вичерпання всіх вхідних параметрів або досягнення певної кількості об'єктів у вузлі.

Дерева рішень не завжди є оптимальними і для того, щоб дізнатись, що даний алгоритм працює найкращим чином потрібно провести оцінку. Даний метод може бути оцінений за допомогою різних критеріїв. Таких як:

– критерії оцінки: одним з основних критеріїв оцінки є точність або ефективність моделі «Дерева рішень» при класифікації або прогнозуванні. Іншим критерієм є розмір дерева, де більші дерева можуть бути складнішими для інтерпретації та обчислень;

– відбір найкращого «Дерева рішень»: оцінюючи критерії, можна вибрати найкраще «Дерево рішень» з декількох побудованих варіантів. Це може включати вибір дерева з найвищою точністю або оптимальним балансом між точністю та складністю.

Виходячи з власного аналізу роботи алгоритму «Дерева рішень», можна визначити, що він добре підходить для швидкого аналізу отриманих даних. На основі цих даних метод запускає рекурсивний підхід для того, щоб дістатись певної глибини дерева або його кінця, якщо не було знайдено необхідне рішення.

### 2.2.2 Випадковий ліс

Алгоритм «Випадковий ліс» (Random Forest), який є значним внеском у сферу машинного навчання, був розроблений професором Лео Брейманом у 2001 році. Цей алгоритм є кульмінацією кількох ключових розробок у сфері ансамблевого навчання, зокрема інтеграції підходу Бреймана до вибірки пакетів із 1996 року і випадковий вибір функцій. Остання концепція була представлена доктором Тін Кам Хо в 1995 і 1998 роках, а також доктором Амітом і професором Джеманом в 1997 році. Примітно, що перший алгоритм для «Випадкових лісів» рішень, що використовує метод випадкового підпростору, був фактично створений доктором Хо в 1995 році. Досягнення Бреймана в техніці пакетування в другій половині 1990-х років сприяли відкриттю алгоритму випадкового лісу в січні 2001 року [3].

Випадковий ліс (Random Forest) є ансамблем алгоритмів "Дерева рішень" (Decision Trees). Він створюється шляхом побудови багатьох дерев рішень, кожне з яких навчається на випадковій вибірці даних з набору тренувальних даних. У процесі прогнозування випадковий ліс об'єднує результати всіх дерев, щоб отримати більш точний та стабільний прогноз, ніж це можливо з одного дерева рішень.

Алгоритм Random Forest, відомий своєю надійністю та точністю, знаходить широке застосування в різних областях завдяки своїй здатності виконувати завдання як класифікації, так і регресії. Цей алгоритм особливо цінується за його здатність генерувати різноманітні та випадкові дерева рішень, що веде до більш надійних і точних прогнозів.

Однією з ключових сфер застосування Random Forest є банківський сектор. Тут він використовується для прогнозування платоспроможності позичальників. Ця можливість має вирішальне значення для того, щоб кредитні установи могли приймати обґрунтовані рішення щодо схвалення кредитів. Крім того, Random Forest використовується для виявлення шахрайства, допомагаючи банкам у виявленні шахрайських дій [4-5].

У галузі охорони здоров'я алгоритм Random Forest відіграє значну роль у діагностиці пацієнтів. Це допомагає визначити тенденції захворювань і оцінити ризики, пов'язані з різними захворюваннями. Ця програма має життєво важливе значення для прогнозу медичної допомоги, оскільки вона дозволяє медичним працівникам приймати більш обґрунтовані рішення щодо догляду за пацієнтами та стратегій лікування [6].

Окрім цих секторів, алгоритм Random Forest має широкий спектр застосувань в інших галузях. Він використовується у фінансах для прогнозування коливань цін на акції, у маркетингу для розуміння попиту та поведінки споживачів, а також в електронній комерції для процесів прийняття рішень. Ці програми підкреслюють універсальність алгоритму Random Forest у підвищенні ефективності бізнес-процесів і наукових досліджень [7-8].

Нехай навчальна вибірка складається з  $N$  прикладів, розмірність простору ознак дорівнює  $M$ , і заданий параметр  $m$ .

$$m \approx \sqrt{M}.$$

Усі дерева комітету будуються незалежно один від одного. Процедура побудови є наступною. Згенеруємо випадкову підвбірку з повторенням розміром  $n$  з навчальної вибірки. (Таким чином, деякі приклади потраплять в неї кілька разів, а приблизно  $N/3$  прикладів не ввійдуть у неї взагалі). Далі випадково оберемо  $m$  предикторів (ознак) із  $M$ . Побудуємо дерево рішень, яке класифікує приклади даної підвбірки, причому в ході створення чергового вузла дерева будемо вибирати ознаку, на основі якої проводиться розбиття, не з усіх  $M$  ознак, а лише з  $m$  випадково вибраних. Вибір найкращого з цих  $m$  ознак може здійснюватися різними способами. В оригінальному кодї Брейман використовується критерій Джині, що застосовується також в алгоритмі побудови вирішальних дерев CART. У деяких реалізаціях алгоритму замість нього використовується критерій приросту інформації. Розділимо ознаку  $X$  на два класи  $X_i \geq S_i$  та  $X_i < S_i$ . Виміряємо гомогенність у двох нових класах за допомогою критерію Джині. Оберемо таке значення «спліт-поінту»  $S_i$  ознаки  $X$ , для якого досягнуто максимальної гомогенності класу. Дерево будується до повного вичерпання підвбірки і не піддається процедурі відсікання. Повертаємося генерації випадкової підвбірки, генеруємо нову вибірку і повторюємо вибір предиката будуючи наступне дерево.

На основі проведеного аналізу алгоритму «Випадковий ліс», можна зробити візуальне зображення роботи та відмінності цього методу від «Дерева рішень». На рис. 2.2 зображено роботу підходу «Випадковий ліс» у порівнянні з «Деревом рішень».

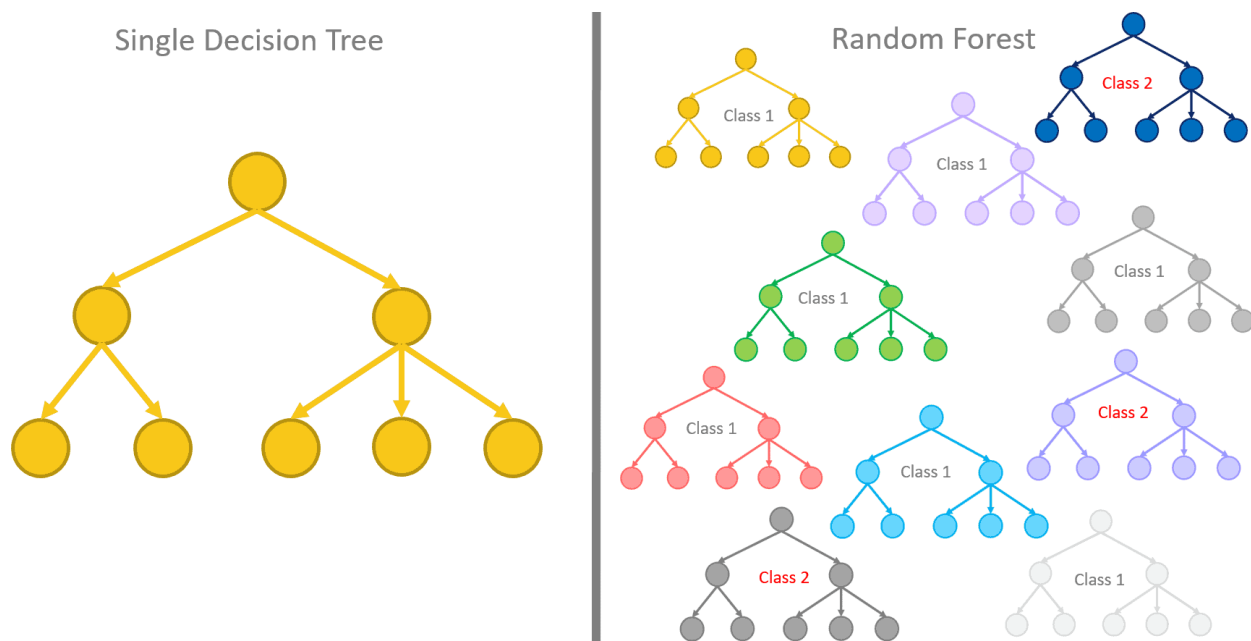


Рисунок 2.2 – Порівняння роботи «Дерева рішень» з «Випадковим лісом»

Виходячи з отриманого результату аналізу, алгоритм ансамбль «Випадковий ліс» є видозміненою версією «Дерева рішень». Даний алгоритм за рахунок збільшення варіантів дерев поки що є більш ефективним варіантом для досягнення мети дослідження. Так як, алгоритм буде використовувати більшу вибірку рішень, це може добре вплинути на оптимізацію навчання штучного інтелекту, а також покращити результат відповіді. Завдяки своїй здатності надавати точні та надійні результати його можна використати для початкового етапу розробки штучного інтелекту.

### 2.2.3 Градієнтне посилення

Розробка алгоритму посилення градієнта, основного методу в машинному навчанні, є історією ітераційних інновацій та вдосконалення. Першим прототипом був AdaBoost (Adaptive Boosting), він виявився досить успішним ансамблевим алгоритмом підвищення, представлений в 1998 році Лео Брейманом. AdaBoost функціонував шляхом зважування спостережень і послідовного навчання набору слабких учнів, як правило, тих, хто приймає

рішення, з більшою вагою, наданою зразкам, які були погано підігнані на попередніх етапах. Цей метод зрештою об'єднав усіх учнів у єдиний складний класифікатор [9].

Джером Фрідман у 1999 році значно розвинув цю концепцію, узагальнивши алгоритми посилення, що призвело до створення машин підвищення градієнта (GBM). На відміну від AdaBoost, яка мінімізувала експоненціальну функцію втрат і була вразливою до викидів, GBM дозволяла використовувати будь-яку диференційовану функцію втрат, підвищуючи стабільність перед лицем викидів. Крім того, GBM представив новий підхід прогнозування помилки, залишеної попередньою моделлю, за допомогою оптимізації функції втрат, проведеної за допомогою градієнтного спуску [9].

Однак посилення градієнта не обійшлося без проблем. Він був схильний до переобладнання, що могло погіршити продуктивність. Щоб пом'якшити це, було введено такі методи, як підсилення стохастичного градієнта, який включає підвибірку навчального набору даних для навчання окремих учнів. Цей підхід зменшив кореляцію між результатами окремих учнів, таким чином підвищивши точність. Інший метод, shrinkage, передбачав зважування прогнозів кожного дерева, щоб уповільнити процес навчання, хоча ціною збільшення часу обчислення. Обмеження на дерева, як-от обмеження глибини дерева та кількості спостережень на розбиття, також допомогли контролювати переобладнання [9].

Алгоритм Gradient Boosting, відомий своєю високою точністю та ефективністю, особливо з великими та складними наборами даних, має широкий спектр застосувань у різних галузях. Його здатність мінімізувати помилки зміщення в моделях робить його популярним вибором у рішеннях машинного навчання для бізнесу та змагань Kaggle. На відміну від AdaBoost, який дозволяє змінювати базову оцінку, Gradient Boosting має фіксовану базову оцінку у формі дерев рішень. Його основний механізм передбачає послідовне створення моделей для зменшення помилок попередніх моделей, використовуючи різні функції втрат для проблем регресії та класифікації.

У фінансовому секторі Gradient Boosting використовується для прогнозування цін на акції, оцінки кредитного ризику та виявлення шахрайських операцій. Інвестиційні фірми аналізують історичні фондові дані, щоб спрогнозувати майбутні тенденції, допомагаючи приймати обґрунтовані інвестиційні рішення. Банки та компанії, що видають кредитні картки, використовують його для оцінки кредитоспроможності клієнтів шляхом аналізу таких факторів, як дохід, зайнятість і кредитна історія, таким чином покращуючи рішення щодо кредитування та мінімізуючи втрати. Крім того, він використовується для виявлення аномальних даних транзакцій, що сигналізує про потенційне шахрайство.

Досі ми бачили, як градієнтне посилення працює в теорії. Тепер ми зануримося в математику та логіку, що стоять за цим, щоб усе було дуже зрозуміло.

Різниця між прогнозом і фактичним значенням відома як залишкова (або в даному випадку псевдозалишки), на основі якої підсилення градієнта будує послідовні дерева.

Нехай, вихідна модель  $y$  яка відповідає лише одному дереву рішень визначається як наведено у формулі (2.1)

$$y = A_1 + (B_1 * x) + e_1,$$

де  $e_1$  це залишок від цього дерева рішень.

У градієнтному посиленні ми підбираємо послідовні дерева рішень за залишком від останнього. Отже, коли до цієї моделі застосовано посилення градієнта, послідовні дерева рішень будуть математично представлені у формулі (2.2) та (2.3)

$$e_1 = A_2 + (B_2 * x) + e_2;$$

$$e_2 = A_3 + (B_3 * x) + e_3.$$

Потрібно зауважити, що тут зупиняємося на 3 деревах рішень, але в фактичній моделі посилення градієнта кількість учнів або дерев рішень набагато більше. Об'єднавши всі три рівняння, остаточна модель дерева рішень буде задана за формулою (2.4)

$$y = A_1 + A_2 + A_3 + (B_1 * x) + (B_2 * x) + (B_3 * x) + e_3.$$

Математичне розуміння побудови алгоритму дає можливість краще використати його для розробки штучного інтелекту. Враховуючи отриманні знання роботи алгоритму «Градiєнтне посилення», можна створити візуальне зображення роботи цього методу та проаналізувати його роботу. Для цього було розроблено зразок, що використовує набір даних «PIMA Indians Diabets» [10] та який містить інформацію про параметри здоров'я людини та результат 0 або 1, залежно від того, чи хворіє він на діабет.

Завдання полягає в тому, щоб класифікувати людину як діабетика, якщо отримати необхідні дані про її здоров'я. Завантажений набір даних зображений на рис. 2.3

|   | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI  | DiabetesPedigreeFunction | Age | Outcome |
|---|-------------|---------|---------------|---------------|---------|------|--------------------------|-----|---------|
| 0 | 6           | 148     | 72            | 35            | 0       | 33.6 | 0.627                    | 50  | 1       |
| 1 | 1           | 85      | 66            | 29            | 0       | 26.6 | 0.351                    | 31  | 0       |
| 2 | 8           | 183     | 64            | 0             | 0       | 23.3 | 0.672                    | 32  | 1       |
| 3 | 1           | 89      | 66            | 23            | 94      | 28.1 | 0.167                    | 21  | 0       |
| 4 | 0           | 137     | 40            | 35            | 168     | 43.1 | 2.288                    | 33  | 1       |

Рисунок 2.3 – Завантажений набір даних для навчання моделі

Після завершення попередньої обробки даних, настав час перейти до детальнішого вивчення та налаштування класифікатора з використанням методу посилення градієнта, включаючи тонке налаштування його гіперпараметрів. Наступним кроком буде застосування цього класифікатора до

навчального набору даних. По завершенню навчання моделі, можна оцінити її результативність. Одним із способів оцінки є використання матриці плутанини, яка демонструє кількість правильних та неправильних класифікацій здійснених моделлю. В даному випадку, за допомогою класифікатора Gradient Boosting Classifier було неправильно класифіковано 42 випадки у порівнянні з 112 правильними класифікаціями. Загальна точність моделі склала 73%, що є середнім показником ефективності.

Отриманий результат є лише малою частиною множини методів посилення градієнта і дає загальне розуміння принципів його роботи, вказуючи на потенціал глибшого дослідження цієї теми.

### 2.3 Висновки дослідження алгоритмів заснованих на дереві рішень

Було проведено глибокий аналіз методів машинного навчання, що засновані на алгоритмі дереві рішень. На основі отриманих знань та даних, а також досліджених статтях потрібно визначити який саме алгоритм буде використано для розробки штучного інтелекту планування дня. У таблиці 2.1 наведено основні переваги та недоліки кожного з алгоритмів.

Таблиця 2.1 – Переваги та недоліки алгоритмів заснованих на дереві рішень

| Критерій      | Дерево рішень                          | Випадковий ліс  | Градiєнтне посилення  |
|---------------|--|---|---|
| Переваги      |  |   |   |
| Інтуїтивність | Прості для розуміння та інтерпретації. | Кращі загальні результати порівняно з окремим деревом рішень. | Часто надає найкращу продуктивність на різноманітних наборах даних. |

Продовження таблиці 2.1

| Критерій                   | Дерево рішень                      | Випадковий ліс  | Гradientне посилення                            |
|----------------------------|------------------------------------|---|---|
| Швидкість навчання         | Швидке навчання.                   | Швидше, ніж gradientне посилення, але повільніше, ніж одне дерево рішень. | Повільніше через послідовне побудову моделей.   |
| Робота з великими даними   | Ефективне на невеликих даних.      | Добре масштабується на великі набори даних.                               | Добре масштабується на великі набори даних.     |
| Недоліки                   |                                    |   |   |
| Схильність до перенавчання | Висока схильність до перенавчання. | Менша схильність до перенавчання, ніж у дерева рішень.                    | Можливе перенавчання, особливо на шумних даних. |
| Інтерпретованість          | Висока.                            | Середня (складніше інтерпретувати, ніж окреме дерево).                    | Низька (складні моделі, важко інтерпретувати).  |
| Чутливість до змін у даних | Висока.                            | Нижча, ніж у одного дерева рішень.  | Нижча, ніж у одного дерева рішень.              |

У контексті оцінки алгоритмічних рішень для оптимізації планування дня штучним інтелектом, детальний аналіз переваг і обмежень різних методів машинного навчання є необхідним. Вибір оптимального алгоритму вимагає зваженого аналізу його характеристик, включаючи здатність до обробки

великого масиву даних, швидкість навчання, та стійкість до проблем перенавчання (overfitting) або недонавчання (underfitting).

Алгоритм "Випадковий ліс" (Random Forest) вирізняється серед інших підходів завдяки його унікальним властивостям. Цей алгоритм забезпечує високу точність прогнозування через використання ансамблю дерев рішень, що сприяє зниженню ризику перенавчання. Він ефективно використовує великі набори даних, забезпечуючи глибоке і точне навчання, а також здатний оперативно обробляти нові дані, що робить його ідеальним для динамічних сценаріїв, таких як планування дня. Крім того, "Випадковий ліс" легко масштабується, адаптуючись до збільшення обсягу даних без істотного зниження продуктивності.

На відміну від інших складних моделей, таких як нейронні мережі, "Випадковий ліс" пропонує порівняно просту інтерпретацію. Це означає, що результати, отримані з цього алгоритму, можуть бути більш зрозумілими для користувачів, що є критично важливим для прийняття рішень у сфері планування. Отже, "Випадковий ліс" представляє собою оптимальний баланс між складністю, точністю та інтерпретаційною здатністю, що робить його ідеальним вибором для використання в системах штучного інтелекту, орієнтованих на планування та оптимізацію розкладу дня.

## 3 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

### 3.1 Опис тестового набору даних

Дослідження в області оптимізації планування дня використовуючи методи машинного навчання представляє собою складне завдання, яке вимагає ретельного підходу до вибору та обробки даних. Першочерговим етапом є визначення відповідних параметрів, які будуть використовуватися для тестування алгоритму. На основі попереднього аналізу, визначено ключові параметри: наявність освітньої або професійної діяльності (школа, університет, курси), інтереси та хобі, регулярні фізичні заняття, чітко визначені цілі та завдання з низьким рівнем пріоритету. Важливим аспектом є врахування часового компоненту для кожного з цих параметрів, щоб забезпечити точність і реалістичність планування.

Під час пошуку відповідного набору даних на платформах, таких як Kaggle, Google, MavenAnalytics, не було знайдено достатньо відповідних даних, які б повністю задовольняли б усі визначені критерії. У зв'язку з цим було прийнято рішення про розробку спеціалізованого набору даних, що враховує усі потрібні параметри. У процесі створення цього набору даних було розроблено власний скрипт за допомогою мови програмування Javascript, що дозволило синтезувати високоякісні та релевантні дані, що відповідають усім вимогам дослідження. Використання такого підходу є важливим кроком у розробці ефективного алгоритму машинного навчання для оптимізації планування дня.

У рамках підготовки до дослідження оптимізації планування дня за допомогою методів машинного навчання, було розроблено базовий набір даних, що слугує фундаментом для подальшого аналізу та розширення. Цей первинний набір даних обмежений трьома ключовими параметрами: професійна діяльність, хобі та фізичні заняття. Важливим аспектом кожного

параметра є включення часового виміру, тобто, кількість часу, витраченого на кожну діяльність.

В рамках створення даних для аналізу ефективності алгоритмів оптимізації планування дня, було здійснено структурування записів у базі даних з урахуванням унікальності кожної активності для окремого користувача. Це означає, що кожен запис у датасеті містить унікальний набір хобі для кожного індивідуума, виключаючи повторення одного і того ж хобі.

Далі, для кожного виду діяльності (професійна робота, хобі, та фізичні заняття) було реалізовано алгоритм, який призначає випадкову кількість часу, витраченого на цю діяльність, згідно з попередньо визначеними часовими рамками. Для роботи максимальний часовий інтервал становить 8 годин 59 хвилин, для хобі – 2 години 59 хвилин, а для фізичних занять – 1 година 59 хвилин.

Крім того, в датасеті введено різноманітність у кількість хобі та спортивних занять для кожного користувача. Кожен запис може містити від 1 до 3 різних хобі та від 1 до 2 видів спортивних занять, тоді як професійна діяльність присутня у кожному записі. Така структура датасету дозволяє моделювати реалістичні сценарії повсякденного життя, забезпечуючи комплексний аналіз і розуміння того, як різні типи діяльності впливають на загальну продуктивність і часове планування.

Цей початковий набір даних містить 10,000 записів, що забезпечує достатній обсяг даних для проведення початкового аналізу та випробувань. Розробка цього набору даних є критично важливою для першої фази дослідження, дозволяючи отримати попередні висновки щодо ефективності обраного алгоритму машинного навчання. В майбутньому планується розширення набору даних шляхом включення додаткових параметрів, що дозволить проводити більш глибокий та всебічний аналіз. Це стане основою для подальшого удосконалення алгоритму та підвищення точності прогнозування в області планування дня.

Приклад даних, отриманих з власного набору даних зображено на рис. 3.1

|    | A                | B                       | C          |
|----|------------------|-------------------------|------------|
| 1  | Category         | Name                    | Time Spent |
| 2  | hobbies          | Crocheting              | 0:41       |
| 3  | hobbies          | DJing                   | 0:32       |
| 4  | sport_activities | Squash                  | 0:53       |
| 5  | job              | Carpenter               | 4:59       |
| 6  | Category         | Name                    | Time Spent |
| 7  | hobbies          | Singing                 | 0:37       |
| 8  | sport_activities | Badminton               | 0:29       |
| 9  | job              | Nutritionist            | 6:00       |
| 10 | Category         | Name                    | Time Spent |
| 11 | hobbies          | Model building          | 1:52       |
| 12 | hobbies          | Puzzle solving          | 0:34       |
| 13 | sport_activities | Darts                   | 0:38       |
| 14 | job              | Human Resources Manager | 5:20       |
| 15 | Category         | Name                    | Time Spent |
| 16 | hobbies          | Baking                  | 1:42       |
| 17 | hobbies          | Drawing                 | 1:47       |
| 18 | sport_activities | Taekwondo               | 0:17       |
| 19 | job              | Photographer            | 6:46       |
| 20 | Category         | Name                    | Time Spent |
| 21 | hobbies          | Camping                 | 0:54       |
| 22 | hobbies          | Collecting stamps       | 0:34       |
| 23 | sport_activities | Cycling                 | 0:44       |
| 24 | job              | Data Scientist          | 5:45       |

Рисунок 3.1 – Приклад тестових даних

У контексті дослідження, що спрямоване на оптимізацію планування дня через методи машинного навчання, зокрема, за допомогою класифікаційних алгоритмів, важливо забезпечити необхідну підготовку та обробку вхідних даних. Первинна обробка даних є критичним кроком, який впливає на точність і ефективність подальшого аналізу.

Одним із ключових аспектів передобробки в даному дослідженні є конвертація часових даних, представлених у полі "Time spent", у числовий формат. Це перетворення необхідне, оскільки більшість алгоритмів машинного навчання ефективніше працюють з числовими даними. Процес конвертації включає переведення часових інтервалів, виражених у годинах та хвилинах, у однорідний числовий формат, який може бути безпосередньо використаний в алгоритмах класифікації.

Цей етап передоброби забезпечує, що вхідні дані для машинного навчання є точними, консистентними та відповідають вимогам вибраних алгоритмічних моделей. Коректна підготовка даних є фундаментальною для забезпечення високої якості прогнозів та результатів, отриманих від моделей машинного навчання, використовуваних у дослідженні оптимізації планування дня. На рис. 3.2 зображено результат передоброби тестових даних.

|    | A                | B                      | C          |
|----|------------------|------------------------|------------|
| 1  | Category         | Name                   | Time Spent |
| 2  | hobbies          | Crocheting             | 0,68       |
| 3  | hobbies          | DJing                  | 0,53       |
| 4  | sport_activities | Squash                 | 0,88       |
| 5  | job              | Carpenter              | 4,98       |
| 6  | Category         | Name                   | Time Spent |
| 7  | hobbies          | Singing                | 0,61       |
| 8  | sport_activities | Badminton              | 0,48       |
| 9  | job              | Nutritionist           | 6,00       |
| 10 | Category         | Name                   | Time Spent |
| 11 | hobbies          | Model building         | 1,86       |
| 12 | hobbies          | Puzzle solving         | 0,56       |
| 13 | sport_activities | Darts                  | 0,63       |
| 14 | job              | Human Resources Manage | 5,33       |
| 15 | Category         | Name                   | Time Spent |
| 16 | hobbies          | Baking                 | 1,7        |
| 17 | hobbies          | Drawing                | 1,78       |
| 18 | sport_activities | Taekwondo              | 0,28       |
| 19 | job              | Photographer           | 6,76       |
| 20 | Category         | Name                   | Time Spent |
| 21 | hobbies          | Camping                | 0,9        |
| 22 | hobbies          | Collecting stamps      | 0,56       |
| 23 | sport_activities | Cycling                | 0,73       |
| 24 | job              | Data Scientist         | 5,75       |

Рисунок 3.2 – Результат передоброби тестових даних

Таким чином, було виявлено відсутність відповідного до вимог набору даних на відкритих джерелах. Внаслідок цього виникла необхідність у розробці спеціалізованого інструменту – скрипта, здатного генерувати синтетичні дані, які наближені до реальних умов та вимог дослідження.

Скрипт був створений з метою генерації набору даних, який імітує реальні сценарії повсякденного життя з включенням таких аспектів, як професійна діяльність, хобі, спортивні заняття, що впливають на планування дня. Кожен аспект містить відомості про часові витрати, що є ключовими для подальшої аналізу.

Паралельно з генерацією даних, була здійснена їх передобробка з метою забезпечення оптимального формату та структури для ефективної обробки алгоритмами машинного навчання. Цей процес включав нормалізацію даних, перетворення неструктурованих або напівструктурованих даних у формат, придатний для машинного навчання, та очищення даних від можливих помилок або невідповідностей.

Такий підхід дозволив створити репрезентативний і надійний набір даних, який може бути використаний для тренування та випробування моделей машинного навчання, а також для проведення точного та об'єктивного аналізу в контексті оптимізації планування дня.

### 3.2 Математичний опис методики оцінювання результатів експерименту

В рамках дослідження з оптимізації планування дня, перший крок аналітичного процесу полягає у передобробці даних. Ця фаза критично важлива, оскільки вона дозволяє підвищити якість тестового набору даних шляхом видалення записів з шумом та нормалізації значень. Після цього, дані розділяються на навчальну та тестову вибірки зі співвідношенням 80/20, що є загальноприйнятою практикою в галузі штучного інтелекту.

Основна задача оптимізації полягає в класифікації різних типів діяльності за їх пріоритетністю і визначенні оптимального розподілу часу для них. Модель буде навчена на основі навчальної вибірки.

Для оцінки ефективності моделі будуть використані такі метрики, як F1-Score та матриця помилок. Крім того, для оцінки стійкості моделі в різних сценаріях використовуватиметься крос-валідація.

Точність (precision) та повнота (recall) є фундаментальними метриками в налаштуваннях класифікації. Вони відіграють ключову роль у визначенні ефективності моделі класифікації у контексті вибору найоптимальнішого часу для конкретної діяльності. Точність можна вирахувати за формулою (3.1)

$$tp : (tp + fp),$$

де  $tp$  – це кількість позитивних випадків;

$fp$  – це кількість негативних випадків.

Точність – це інтуїтивно зрозуміла здатність класифікатора не позначати негативний зразок як позитивний. Повноту можна визначити за формулою (3.2)

$$tp : (tp + fn),$$

де  $tp$  – це кількість позитивних випадків;

$fn$  – це кількість фактичних негативних випадків.

Повнота – це інтуїтивно зрозуміла здатність класифікатора знаходити всі позитивні зразки.

F1-Score у машинному навчанні – це метрика, яка використовується для оцінки якості моделі класифікації, особливо в ситуаціях, де розподіл класів є нерівномірним. Вона є гармонійною середньою величиною точності (precision) та повноти (recall). Вирахування F1 наведено у формулі (3.3)

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Цей показник є особливо корисним у випадках, де важливо знайти баланс між точністю та повнотою. Наприклад, у медичних діагностичних системах або в системах виявлення шахрайства, де пропуск фактичних позитивних

випадків або неправильна ідентифікація негативних випадків може мати серйозні наслідки.

Також, як ще однією метрикою оцінювання, може бути крос-валідація. Крос-валідація – це метод у машинному навчанні, який використовується для оцінки здатності моделі узагальнювати результати на новий набір даних. Цей метод особливо корисний для запобігання перенавчання, тобто ситуації, коли модель занадто добре вивчає дані, на яких вона була навчена, але погано працює на нових даних.

Крос-валідація, як правило, не має специфічної математичної формули, але її реалізацію можна описати наступними етапами:

- розділення даних: набір даних ділиться на кілька частин (або "складок"). Найчастіше використовується метод  $k$ -кратної крос-валідації, де  $k$  – це кількість складок;

- тренування та тестування: для кожної з цих складок, модель тренується на даних з усіх інших складок, а потім тестується на даній складці. Це означає, що кожна складка використовується рівно один раз як тестовий набір;

- обчислення результатів: після тренування та тестування на кожній складці, обчислюються показники продуктивності (наприклад, точність, F1-Score) для кожного тестування. Потім з цих результатів обчислюється середнє значення, яке дає загальну оцінку продуктивності моделі.

Крос-валідація дозволяє оцінити, наскільки добре модель буде працювати на незнайомих даних, і є ключовим інструментом у запобіганні перенавчання та забезпеченні достовірності моделі машинного навчання.

### 3.3 Результат досліджень

Під час дослідження, спрямованого на оцінку ефективності алгоритму "Випадковий ліс" для класифікації пріоритетності різних видів діяльності, було застосовано ряд стандартних методик вимірювання продуктивності. Ці методики включають оцінку метриками точності (Precision) та повноти

(Recall), вирахування збалансованого гармонійного середнього цих показників, відомого як F1-Score, а також проведення крос-валідації.

Для цього набір даних був розділений на навчальну та тестову вибірки. Початково модель "Випадковий ліс" була навчена на навчальній вибірці, щоб засвоїти основні закономірності та навчитися робити прогнози. Після навчання модель була протестована на тестовій вибірці, що дозволило оцінити її здатність застосовувати здобуті знання до нових, невідомих раніше даних. Цей підхід дозволяє оцінити загальну ефективність та узагальнюючу здатність моделі. На таблиці 3.1 продемонстровано результат оцінки на тестових даних навчання штучного інтелекту з використанням алгоритму «Випадковий ліс».

Таблиця 3.1 – Результат проведення оцінки тестового набору даних

| Class        | Precision | Recall | F1-Score | Support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.78      | 0.90   | 0.91     | 225     |
| 1            | 0.98      | 0.86   | 0.92     | 227     |
| 2            | 0.88      | 0.97   | 0.82     | 246     |
| accuracy     |           |        | 0.85     | 698     |
| macro avg    | 0.81      | 0.77   | 0.89     | 698     |
| weighted avg | 0.80      | 0.80   | 0.90     | 698     |

У ході оцінки ефективності моделі машинного навчання, зокрема алгоритму "Випадковий ліс", було зафіксовано, що модель демонструє майже 100% точність. Такий високий рівень точності є атиповим для більшості реальних задач і може свідчити про певні недоліки у процесі валідації або підготовці даних. Один із можливих сценаріїв – це перенавчання моделі, що часто спостерігається при використанні алгоритмів, заснованих на дереві рішень. Альтернативно, можливо, що тестовий набір даних був недостатньо складним або не повністю відображав реальні умови задачі.

Також можливо, що обмежений діапазон параметрів (лише три параметри) у синтетичному наборі даних, створеному за допомогою скрипта, спростив завдання класифікації, що призвело до надмірної точності. Це підкреслює важливість розробки репрезентативних тестових наборів даних, які можуть адекватно оцінити узагальнюючу здатність моделі та її ефективність у різноманітних умовах.

## 4 РОЗРОБКА ІНТЕРФЕЙСУ МОБІЛЬНОГО ЗАСТОСУНКУ З ВИКОРИСТАННЯМ МОДЕЛІ ШТУЧНОГО ІНТЕЛЕКТУ

### 4.1 Розробка архітектури взаємодії клієнта з моделлю

У рамках даного підрозділу «Розробка архітектури взаємодії клієнта з моделлю», визначається одна з ключових цілей нашого дослідження, яка полягає у створенні мобільного додатку. Цей додаток призначений для взаємодії користувачів з інноваційною моделлю штучного інтелекту. Проектування мобільного додатку як основного інтерфейсу для комунікації з моделлю було вибрано з метою забезпечення зручності та ефективності взаємодії.

Мобільний додаток, як платформа для взаємодії з моделлю штучного інтелекту, був обраний на основі ретельного аналізу потреб користувачів та специфіки застосування моделі. Детальний опис мотивації та обґрунтування цього вибору наведено у першому розділі дослідження. Головною метою розробки мобільного додатку було створення зручного та інтуїтивно зрозумілого інтерфейсу, який би дозволяв користувачам легко взаємодіяти з моделлю, отримувати необхідну інформацію та виконувати важливі завдання.

Основним функціоналом розробленої моделі штучного інтелекту є можливість створення, управління та оптимізації щоденного переліку завдань користувача. Мобільний додаток, у цьому контексті, надає ідеальну платформу для забезпечення постійного доступу користувачів до цієї інформації. Відзначається, що альтернативний вибір, такий як розробка веб-інтерфейсу, не був б таким ефективним, оскільки важливість мобільного доступу до інформації в сучасному світі є незаперечною. Мобільний додаток дозволяє користувачам завжди мати при собі список завдань, що сприяє підвищенню продуктивності та організованості їхнього щоденного розпорядку. Це особливо важливо у контексті стрімкого ритму сучасного життя, де ефективність управління часом та завданнями відіграє ключову роль.

У процесі розробки мобільного додатку було враховано низку технічних та дизайнерських аспектів, що забезпечують максимальну адаптивність та зручність використання. Інтерфейс додатку розроблявся з урахуванням ергономічності, інтуїтивності навігації та легкості доступу до ключових функцій. Окрім того, важливим аспектом стало забезпечення безпеки та конфіденційності даних користувачів, що є критично важливим у контексті використання штучного інтелекту та обробки персональної інформації.

Загалом, розробка мобільного додатку як інструменту для взаємодії з моделлю штучного інтелекту є значним кроком у напрямку підвищення доступності та ефективності технологій штучного інтелекту для широкої аудиторії. Це дозволяє користувачам більш активно взаємодіяти з розумними системами, отримувати персоналізовані рішення та підвищувати свою продуктивність у повсякденному житті.

Початком розробки мобільного застосунку є створення архітектури всієї послідовності взаємодії користувача. Спершу потрібно описати архітектуру. Головними сутностями взаємодії є: мобільний додаток (інтерфейс), сервер для обробки запитів користувача, база даних для збереження певної інформації та сервер де буде працювати штучний інтелект. Детальний опис архітектури наведено на рисунку 4.1.

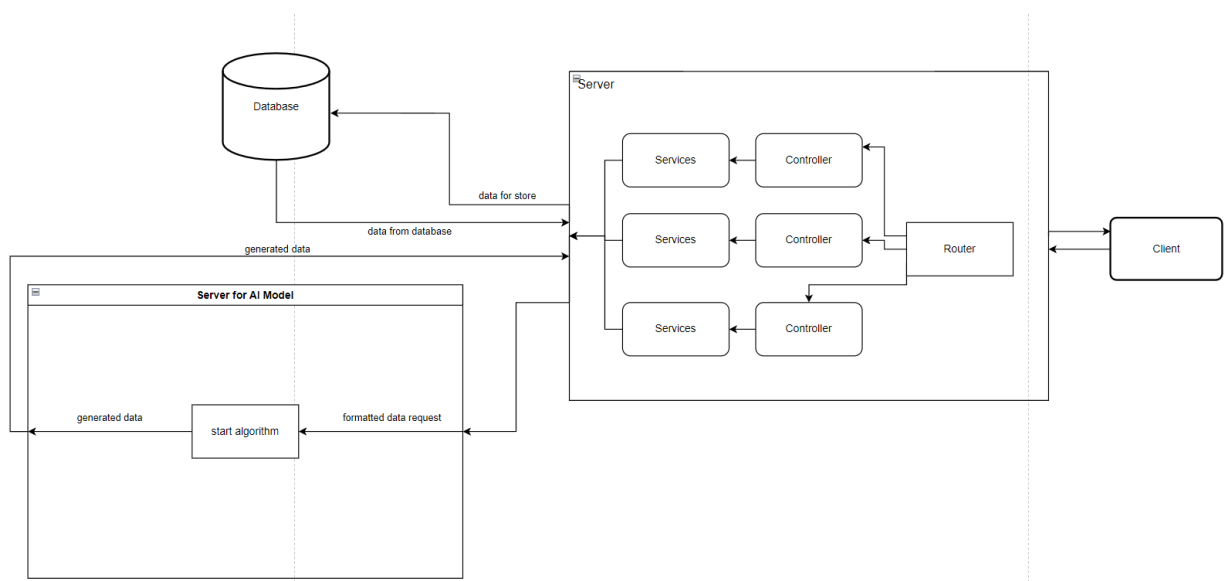


Рисунок 4.1 – Архітектура мобільного додатку

Аналіз рисунку 4.1 демонструє, як були застосовані та описані ключові сутності системи. У центрі схеми знаходиться клієнт, представлений мобільним додатком, через який користувач взаємодіє з системою. Коли користувач ініціює запит, мобільний додаток передає його на сервер. Сервер, використовуючи багаторівневу архітектуру, спершу приймає запит та спрямовує його до відповідного контролеру за допомогою маршрутизатора. Далі активується бізнес-логіка системи.

У процесі обробки запиту сервіси системи виконують кілька функцій. По-перше, відправляється запит до бази даних для додавання специфічної інформації про користувача. По-друге, ініціюється процес генерації плану на день для користувача. Обидва результати цих операцій обробляються у сервісах, після чого сервер надсилає до мобільного додатку згенерований план на день.

Таким чином, рисунок 4.1 ілюструє комплексну взаємодію між клієнтом та сервером, в якій відіграють важливу роль маршрутизація запитів, обробка бізнес-логіки та зворотній зв'язок з користувачем. Цей процес відображає ефективність та гнучкість багаторівневої архітектури у вирішенні комплексних завдань обробки даних та генерації корисного виводу для кінцевого користувача.

Для визначення функціональних вимог до програми, було розроблено use case діаграму, яка ілюструє взаємодію сутностей у контексті головного процесу «створення плану на день». На рисунку 4.2 зображено діаграму головного процесу.

У результаті було розроблено дві UML діаграми, які відображають структуру мобільного додатку та ключовий процес створення розкладу на день. Перша діаграма детально показує взаємодію між основними компонентами проекту, визначаючи їх структуру для подальшої розробки. Друга діаграма зосереджується на функціональних вимогах основного процесу створення денного плану, надаючи більш детальне розуміння самого процесу. Ці діаграми сприяють кращому розумінню взаємозв'язків між різними

елементами системи та ключовими процесами, що є важливим для ефективної розробки та реалізації програмного продукту.

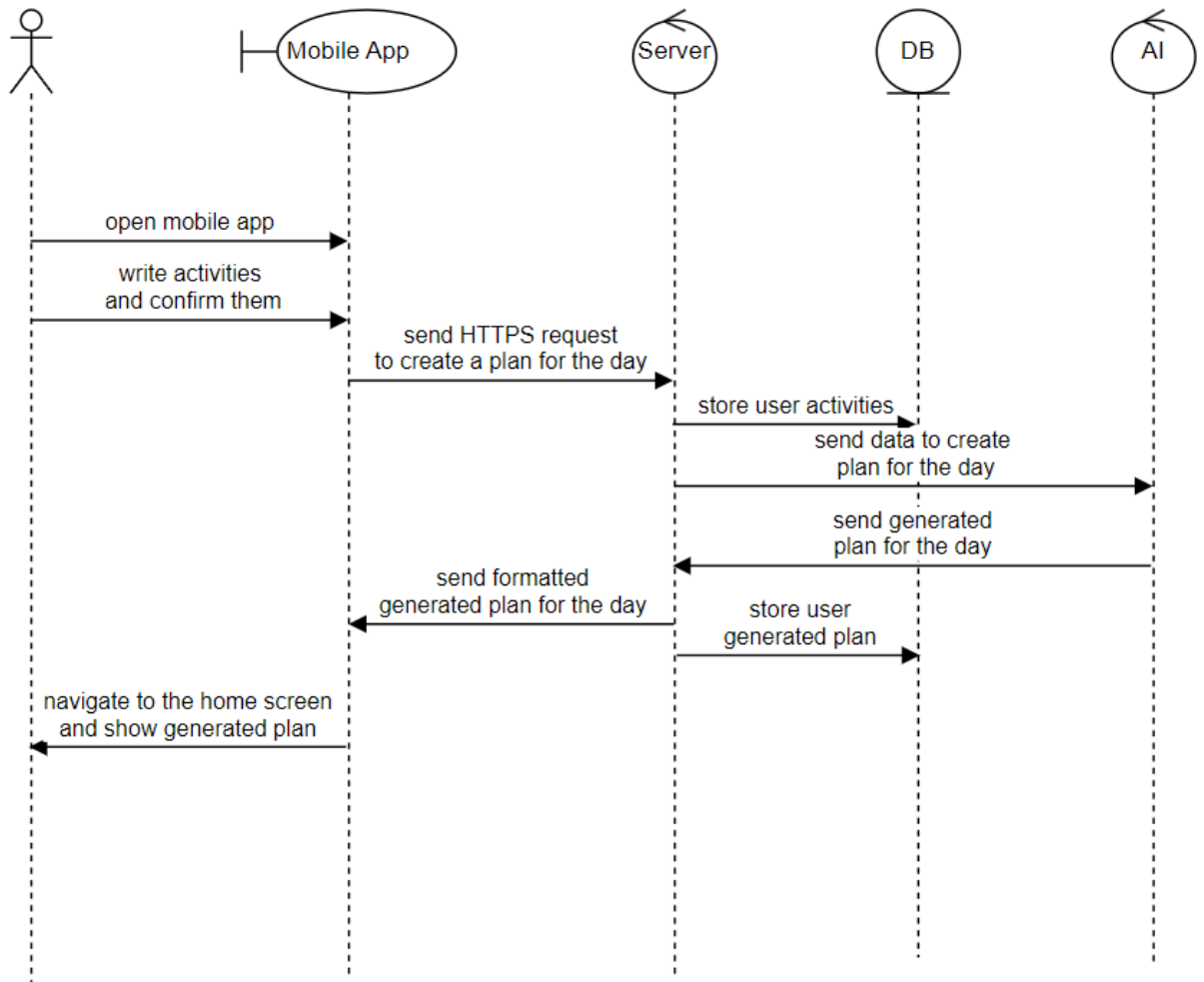


Рисунок 4.2 – Use Case діаграма процесу створення плану на день

#### 4.2 Розробка серверу мобільного додатку

Сервер – це один з ключових процесів роботи усього мобільного додатку, так як він забезпечує форматування даних отриманих від користувача, відправку їх на сервер з працюючою моделлю штучного інтелекту та форматування отриманих даних від неї для відправки їх на сторону клієнта. І для того, щоб розробити сервер згідно до вимог для здійснення усіх операцій, потрібно розібрати його архітектуру, яка наведена на рисунку 4.3.

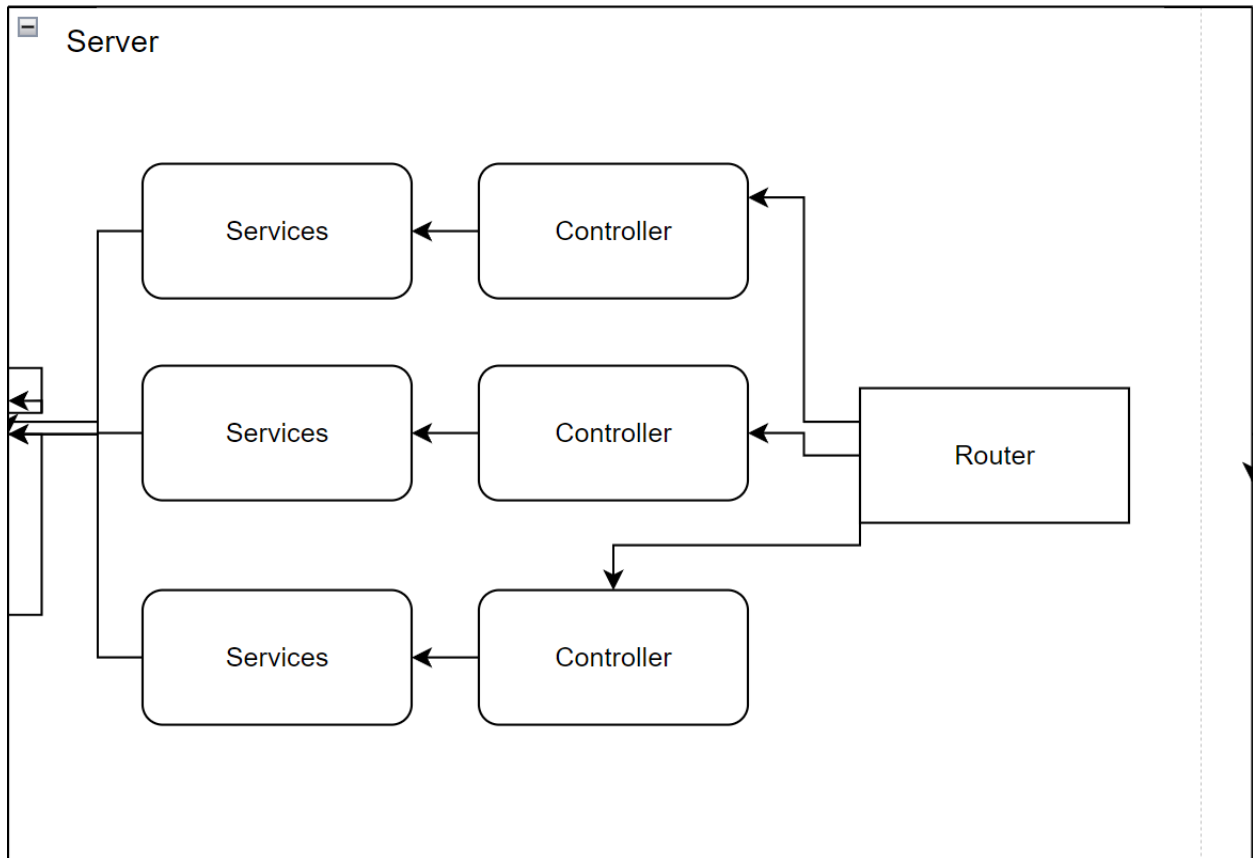


Рисунок 4.3 – Архітектура сервера для обробки даних від клієнта

На цій UML діаграмі представлена спрощена архітектура роботи серверу, спроектована для реалізації мобільного додатку, що допомагає користувачам створювати план на день. У основі діаграми лежить багаторівнева архітектура, яка є оптимальною для досягнення запланованих цілей, забезпечуючи гнучкість і масштабованість системи. Вона включає в себе різні компоненти, такі як Router (маршрутизатор), Controller (контролер) та Service (сервіс), кожен з яких відіграє свою важливу роль в архітектурі сервера.

Маршрутизатор є ключовим компонентом, який відповідає за направлення запитів до відповідних контролерів. Він також займається обробкою заголовків та даних запиту, забезпечуючи їх коректність та безпеку. Це дозволяє системі ефективно реагувати на різні типи запитів, що надходять від користувачів мобільного додатку.

Контролер має на меті форматування даних у відповідний формат для їх подальшої обробки в сервісі. Ця роль важлива, оскільки вона дозволяє

адаптувати вхідні дані до потреб бізнес-логіки, яка реалізована в сервісі. Контролери можуть також виконувати додаткові функції, такі як перевірка автентичності користувача або валідація даних.

Сервіс, у свою чергу, є центральним елементом бізнес-логіки додатку. Він відповідає за обробку даних, запити до бази даних та взаємодію з зовнішніми сервісами, такими як сервер зі штучним інтелектом. У контексті розробки мобільного додатку для створення плану на день сервіс може включати алгоритми для генерації оптимізованого розкладу, аналізу даних користувача та забезпечення персоналізованих рекомендацій.

Така структура дозволяє ефективно розділити відповідальності між компонентами системи, забезпечуючи високу якість обслуговування та легкість внесення змін та оновлень. Багаторівнева архітектура є важливою для забезпечення масштабованості та гнучкості мобільного додатку, а також для забезпечення можливості інтеграції з різними зовнішніми сервісами та технологіями.

На додаток до основних компонентів, таких як Router, Controller та Service, на діаграмі можуть бути представлені й інші важливі елементи, наприклад, модулі аутентифікації, кешування або логування. Ці допоміжні компоненти відіграють важливу роль у забезпеченні безпеки, ефективності та надійності системи.

В цілому, представлена на UML діаграмі архітектура роботи серверу є фундаментальною для розуміння загальної структури та взаємодії компонентів у мобільному додатку. Це дає можливість розробникам і архітекторам системи чітко уявляти процеси та взаємодії, що відбуваються в середині сервера, та ефективно планувати розробку та впровадження функціоналу додатку.

### 4.3 Розробка інтерфейсу мобільного додатку

Одним із ключових компонентів у будь-якій системі є інтерфейс, який забезпечує взаємодію між користувачем та моделлю. У рамках цього

дослідження було прийнято рішення створити мобільний додаток як основний інтерфейс для взаємодії користувачів із системою. Вибір мобільного додатку, зокрема кросс-платформенного, дозволяє забезпечити широкий доступ та високу зручність використання для користувачів, що є критично важливим для досягнення кінцевої мети дослідження – розробки ефективного розкладу дня за допомогою штучного інтелекту.

Мобільний додаток повинен пропонувати інтуїтивно зрозумілий та привабливий користувацький інтерфейс, який не тільки задовольняє основні потреби користувачів, але й сприяє легкій взаємодії зі складними алгоритмами штучного інтелекту. Це включає в себе такі елементи, як адаптивність до різних розмірів екранів, оптимізація для швидкої реакції на запити користувачів та забезпечення високої продуктивності додатку.

Першим кроком у процесі розробки мобільного додатку є створення UML діаграми, яка детально описує взаємодію користувача з системою. Для цього була розроблена діаграма варіантів, що представлена на рисунку 4.4.

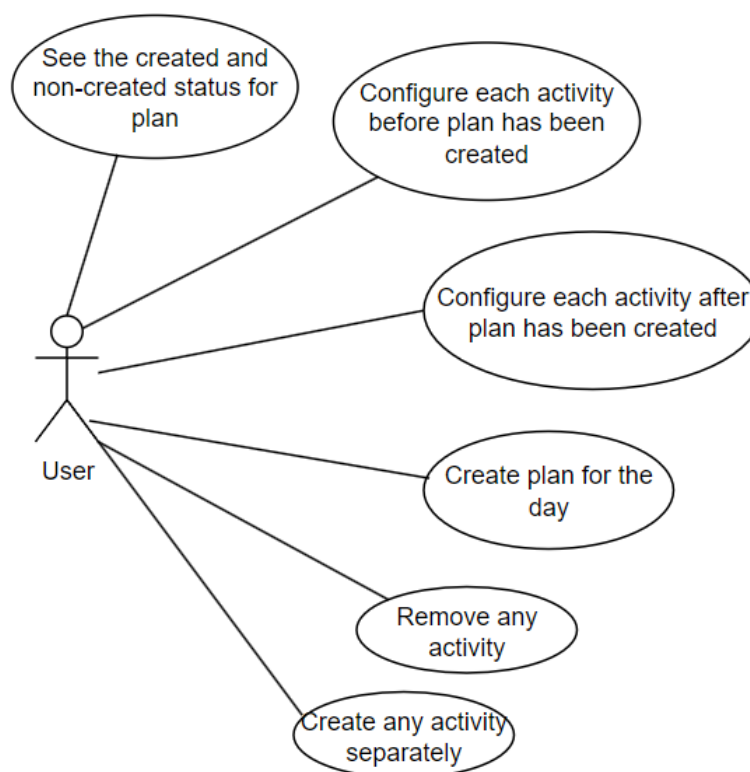


Рисунок 4.4 – Діаграма варіантів, що демонструє можливості користувача

Ця діаграма відображає які дії користувач може виконати користуючись цим додатком, підкреслюючи важливі варіанти, такі як введення даних, перегляд розкладу та взаємодія з алгоритмами штучного інтелекту. Діаграма варіантів допомагає розробникам зрозуміти, як користувачі будуть використовувати додаток, і які функції мають бути реалізовані для забезпечення ефективної роботи системи.

Розробка мобільного додатку як інтерфейсу взаємодії передбачає також врахування ергономіки, доступності та універсальності дизайну, щоб додаток був зручним та доступним для широкого спектру користувачів, включаючи людей з особливими потребами. Важливо також забезпечити безпеку даних користувачів та надійну роботу додатку в різних мережевих умовах.

В цілому, мобільний додаток як інтерфейс має стати ефективним засобом для реалізації основної мети дослідження – створення адаптивного розкладу дня з використанням штучного інтелекту. Це передбачає розробку додатку, орієнтованого на користувача, з високим рівнем функціональності, зручності та інноваційних технологічних рішень.

#### 4.4 Порівняння результатів алгоритмів класифікації

Після успішної розробки мобільного додатку та його серверної частини, наступним етапом є порівняльний аналіз результатів роботи моделей у реальних умовах використання звичайними користувачами. Використання мобільного додатку як звичайним користувачем дозволяє нам провести всебічне тестування функціональності додатку, а також переконатися у досягненні визначених цілей проекту. Важливо зазначити, що процес порівняння базуватиметься на визначених критеріях, які включають швидкість генерації списку завдань, якість згенерованого контенту та відсутність галюцинацій у виведених даних.

Детальний опис критеріїв оцінювання такий:

– швидкість – цей показник аналізується на серверній стороні. Процес відліку часу розпочинається з моменту відправки запиту на сервер штучного інтелекту та закінчується з отриманням відповіді;

– якість – хоча це може бути суб'єктивним критерієм, у даному дослідженні ми намагаємося визначити його якомога об'єктивніше. Якість згенерованого списку завдань буде оцінюватися за десятибальною шкалою, де 0 означає повну невідповідність згенерованого списку до запланованого користувачем, а 10 вказує на високий рівень відповідності, у тому числі з точним розподілом завдань на частини дня та їх пріоритети;

– відсутність галюцинацій – цей критерій також оцінюється за десятибальною шкалою, де 0 відображає ситуацію, коли список завдань містить в основному або повністю завдання, які користувач не вводив, та 10 позначає повну відсутність таких неіснуючих або невідповідних активностей у згенерованому списку.

Окрім цих основних критеріїв, важливо взяти до уваги додаткові аспекти, такі як інтуїтивність інтерфейсу мобільного додатку, його стабільність під час роботи та зручність взаємодії з сервером. Ці фактори також мають велике значення для загального враження користувача від використання додатку.

Для об'єктивного оцінювання швидкості роботи додатку та якості генерації списку завдань буде використано автоматизоване тестування, що дозволить отримати точні метрики. Це допоможе визначити ефективність алгоритмів, використаних у серверній частині, та їх здатність швидко обробляти великі обсяги даних.

В контексті оцінювання якості згенерованих списків завдань, буде проведено анкетування серед користувачів, яке дозволить зібрати суб'єктивні оцінки відповідності згенерованого контенту їхнім реальним потребам та очікуванням. Це допоможе зрозуміти, наскільки добре модель штучного інтелекту враховує індивідуальні особливості та переваги користувачів.

Нарешті, оцінка відсутності галюцинацій в згенерованих списках завдань є критично важливою для забезпечення точності та надійності роботи системи.

Цей показник допоможе виявити можливі помилки або неточності в алгоритмах генерації та забезпечити їх своєчасне виправлення.

Таким чином, комплексний підхід до оцінювання роботи мобільного додатку та серверної частини дозволить не тільки підтвердити досягнення поставлених цілей, але й виявити потенційні можливості для подальшого вдосконалення системи.

Для проведення порівняльного аналізу різних алгоритмів, слід наголосити на тому, що усі експерименти виконувались за допомогою однакового набору інструкцій (завдань на день, визначених користувачем), при цьому усі моделі були навчені на ідентичному датасеті.

Першим алгоритмом, що розглядається в рамках цього дослідження, є "Випадковий ліс". Результати роботи цього алгоритму, включаючи вхідні та вихідні дані, а також час обробки, представлені на рисунку 4.5.

```

RANDOM FOREST TREE: {
  userInput: 'make easy exercises, breakfast, work from 9 am to 16 pm, prepare for exams, make some dinner, call my mom [morning: urgent]'
}
track time: 0.068ms
USER INPUT: make easy exercises, breakfast, work from 9 am to 16 pm, prepare for exams, make some dinner, call my mom [morning: urgent]
GENERATED SCHEDULE: 1. Morning (6:00 - 7:00) = Make easy exercises = Medium
2. Morning (7:00 - 8:00) = Breakfast = Low
3. Morning (8:00 - 9:00) = Call my mom = Urgent
4. Morning to Afternoon (9:00 - 16:00) = Work = High
5. Afternoon (16:00 - 18:00) = Prepare for exams = High
6. Evening (18:00 - 19:00) = Make some dinner = Medium
track time: 5.806s
track time: 5.807s

```

Рисунок 4.5 – Згенерований розклад за допомогою моделі із алгоритмом «Випадковий ліс»

На підставі аналізу виконання алгоритму "Випадковий ліс", було встановлено наступні показники ефективності: час виконання складає 5.807 секунд, якість виведених даних оцінюється в 9 з 10 балів, а відсутність галюцинацій - 10 з 10 балів. За цими результатами можна зробити проміжний висновок про високу ефективність моделі, навченої за допомогою алгоритму "Випадковий ліс", у відповідності до визначених критеріїв оцінювання.

Другим алгоритмом, який підлягає аналізу, є модель, навчена з використанням алгоритму "Бінарне дерево". Результати роботи цієї моделі представлені на рисунку 4.6.

```

BINARY TREE: {
  userInput: 'make easy exercises, breakfast, work from 9 am to 16 pm, prepare for exams, make some dinner, call my mom [morning: urgent]'
}
track time: 0.005ms
USER INPUT: make easy exercises, breakfast, work from 9 am to 16 pm, prepare for exams, make some dinner, call my mom [morning: urgent]
GENERATED SCHEDULE: 1. Evening = Make easy exercises = Low
2. Evening = Breakfast = Medium
3. Morning = Work from 9 am to 16 pm = High
4. Morning = Prepare for exams = Urgent
5. Afternoon = Make some dinner = Low
6. Evening = Call my mom = Urgent
track time: 3.978s
track time: 3.978s

```

Рисунок 4.6 - Згенерований розклад за допомогою моделі із алгоритмом «Бінарне дерево»

Вивчення результатів, отриманих від моделі, навченої за допомогою алгоритму "Бінарне дерево", дозволило визначити наступні показники ефективності: час виконання - 3.978 секунд, якість - 5 з 10 балів, відсутність галюцинацій - 10 з 10 балів. Аналіз рисунку 4.6 вказує на те, що модель демонструє неадекватну пріорітезацію завдань, оскільки розподіл таких завдань, як підготовка до екзамену та робота в ранковий період, є субоптимальним. Крім того, виявлено, що алгоритм має певні обмеження у врахуванні заданих параметрів часу та пріоритету. Наприклад, завдання "call my mom", яке було введено як термінове завдання на ранок, було оброблено алгоритмом лише з врахуванням пріоритету, без урахування часових параметрів. Це вказує на певні недоліки в алгоритмі "Бінарне дерево" щодо точності обробки завдань з комплексними вимогами.

Останнім алгоритмом буде розглянуто «Гرادієнтне посилення», його результати зображені на рисунку 4.7.

```

GRADIENT BOOSTING TREE: {
  userInput: 'make easy exercises, breakfast, work from 9 am to 16 pm, prepare for exams, make some dinner, call my mom [morning: urgent]'
}
track time: 0.067ms
USER INPUT: make easy exercises, breakfast, work from 9 am to 16 pm, prepare for exams, make some dinner, call my mom [morning: urgent]
GENERATED SCHEDULE: 1. Morning (6:00 - 7:00) = Wake up and make easy exercises = High
2. Morning (7:00 - 8:00) = Prepare and have breakfast = Medium
3. Morning (8:00 - 9:00) = Call my mom = Urgent
4. Morning to Afternoon (9:00 - 16:00) = Work = High
5. Afternoon (16:00 - 17:00) = Break and refresh = Low
6. Afternoon (17:00 - 19:00) = Prepare for exams = High
7. Evening (19:00 - 20:00) = Make some dinner = Medium
8. Evening (20:00 - 21:00) = Relax and unwind = Low
9. Evening (21:00 - 22:00) = Review for exams = Medium
10. Evening (22:00 - 23:00) = Prepare for bed = Low
track time: 12.252s
track time: 12.253s

```

Рисунок 4.7 - Згенерований розклад за допомогою моделі із алгоритмом «Градiєнтне посилення»

На основі отриманого згенерованого списку задач, можна визначити наступні значення показників: швидкість – 12.253 секунд, якість – 7/10, відсутність галюцинацій – 5/10. Модель з алгоритмом «Гرادієнтне посилення» показала найвіддаленіші від інших результати. Швидкість генерації цієї моделі була значно вищою за інші, з цього випливає твердження описане у другому розділі, що цей алгоритм є занадто ресурсозатратним. Якість була оцінена у 7 балів, так як в цілому модель повернула потрібні задачі з суб'єктивно правильною пріорітезацією, але через те, що додала від себе задачі, які є логічними у даному списку, але які користувач не додавав самостійно, тому було вирішено залишити оцінку 7. Відсутність галюцинацій: це єдина модель з усіх розглянутих що згенерувала список задач з «галюцинаціями», тобто задачами яких користувач не вводив.

Як висновки цього порівняння, приведемо таблицю з усіма результатами.

Таблиця 4.1 – Таблиця порівняння результатів моделей

|                      | Швидкість в секундах | Якість | Відсутність галюцинацій |
|----------------------|----------------------|--------|-------------------------|
| Випадковий ліс       | 5.807                | 9/10   | 10/10                   |
| Бінарне дерево       | 3.978                | 5/10   | 10/10                   |
| Градiєнтне посилення | 12.253               | 7/10   | 5/10                    |

На основі аналізу даних, представлених у таблиці 4.1, можна зробити обґрунтований висновок про те, що застосування алгоритму "Випадковий ліс" є оптимальним рішенням для вирішення завдань класифікації в контексті даного дослідження. Ефективність "Випадкового лісу" зумовлена його ансамблевою структурою, яка базується на використанні численних дерев рішень. Така структура сприяє зниженню ризиків перенавчання (overfitting) та недонавчання (underfitting), забезпечуючи більш стабільні та надійні результати.

Ключовим аспектом, що сприяє ефективності алгоритму "Випадковий ліс" в контексті класифікації задач, є його здатність до генерації множини бінарних дерев рішень, кожне з яких вносить власний вклад у загальний прогноз. Ця множинність дерев сприяє поліпшенню загальної точності класифікації, оскільки кожне дерево розглядає різні аспекти вхідних даних, а фінальне рішення формується на основі агрегації результатів кожного з них.

Таким чином, алгоритм "Випадковий ліс" виявляється ефективним у задачах класифікації завдань, забезпечуючи високу точність та стабільність роботи моделі, що підтверджується результатами, представленими у таблиці 4.1.

## ВИСНОВКИ

Метою кваліфікаційної роботи було проведення аналізу предметної області, опис напрямку дослідження, постановка задачі та результати експериментального дослідження.

Перед початком роботи над реалізацією штучного інтелекту було проведено аналіз предметної області у декілька етапів. Першим був пошук та розбір існуючих прикладів реалізації штучного інтелекту та сервісів для взаємодії з ними. Другий етап – аналіз існуючих алгоритмів та методів машинного навчання. У рамках третього було проведено розбір існуючих методологій та систем планування дня. Кінцевим етапом було проведення аналізу існуючих інтерфейсів для взаємодії користувача зі штучним інтелектом. На кожній стадії було сформовано висновки на основі яких проводилось подальше дослідження.

У другому розділі було виконано постановку задачі для дослідження та розглянуто алгоритми засновані на дереві рішень. Сформульована постановка задачі надала можливість зробити дослідження більш змістовним та з'ясувати його мету. Спочатку було проаналізовано усі недоліки, переваги та сфери використання базового алгоритму «Дерево рішень». Потім, було проведено глибоке вивчення алгоритму «Випадковий ліс», який і було обрано для подальшої реалізації мети дослідження. Останнім алгоритмом було проаналізовано «Гradientне посилення», який виявився не доцільним для використання у рамках дослідження.

У третьому розділі було згенеровано набір тестових даних та описано його. Потім, було визначено метрики оцінки досліджуваного алгоритму «Випадковий ліс». Кінцевий етап – це проведення навчання на наборі даних та отримання результатів оцінки.

У четвертому розділі було описано розробку архітектури мобільного додатку. Детальний опис дозволив більш чітко зрозуміти вимоги до системи та повноцінну взаємодію її компонентів від початку до кінця. Також, було

описано розробку архітектури серверу та його внутрішніх компонентів. Більш того, був проведений аналіз варіантів взаємодії користувача з системою на основі якого було розроблено дві UML діаграми, діаграма використання та діаграма варіантів.

Останнім етапом дослідження стало порівняння різних моделей, що були навчені з використанням різних алгоритмів, але на одному наборі даних. Аналізуючи отримані показники було зроблено висновок, що результат теоретичного дослідження відповідає практичному для вирішення даної задачі.

Подальшим дослідженням може бути удосконалення набору даних, збільшити його комплексність та покращити алгоритм машинного навчання додавши евристичний алгоритм для знаходження оптимальних рішень.

Було підготовлено доповідь на всеукраїнській конференції «Комп'ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві», яка стосується теми моєї кваліфікаційної роботи [21].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. 80% of the world's population use mobile devices. URL: <https://www.edume.com/blog/mobile-learning-statistics> (дата звернення: 25.09.2023).
2. Decision Tree History. URL: <https://holypython.com/dt/decision-tree-history/> (дата звернення: 28.09.2023).
3. Historical Developments of Random Forest. URL: <https://drjariel.medium.com/historical-developments-of-random-forest-41492deb6737> (дата звернення: 30.09.2023).
4. Random Forest Algorithm. URL: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm> (дата звернення: 01.10.2023).
5. Mbaabu O. Introduction to Random Forest in Machine Learning. URL: <https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/> (дата звернення: 01.10.2023).
6. Random Forest Algorithm. URL: <https://www.javatpoint.com/machine-learning-random-forest-algorithm> (дата звернення: 01.10.2023).
7. Logunova I. Random Forest Classifier: Basic Principles and Applications. URL: <https://serokell.io/blog/random-forest-classification> (дата звернення: 02.10.2023).
8. Random Forest. URL: <https://www.wallstreetmojo.com/random-forest/> (дата звернення: 02.10.2023).
9. Story of Gradient Boosting: How It Evolved Over Years. URL: <https://analyticsindiamag.com/story-of-gradient-boosting-how-it-evolved-over-years/> (дата звернення: 03.10.2023).
10. Pima Indians Diabetes Database. URL: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database> (дата звернення: 05.10.2023).

11. Russell, Stuart and Norvig, Peter. "Artificial Intelligence: A Modern Approach." Pearson, 2016. 1152 с.
12. Bishop, Christopher M. "Pattern Recognition and Machine Learning." Springer, 2006. 738 с.
13. Lafore, Robert. "Data Structures and Algorithms in Java." Sams Publishing, 2002. 800 с.
14. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome. "The Elements of Statistical Learning." Springer, 2009. 763 с.
15. Kleppmann, Martin. "Designing Data-Intensive Applications." O'Reilly Media, 2017. 616 с.
16. Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. "Deep Learning." MIT Press, 2016. 800 с.
17. Sipser, Michael. "Introduction to the Theory of Computation." Cengage Learning, 2012. 504 с.
18. Murphy, Kevin P. "Machine Learning: A Probabilistic Perspective." MIT Press, 2012. 1104 с.
19. Sedgewick, Robert and Wayne, Kevin. "Algorithms." Addison-Wesley, 2011. 992 с.
20. Saternos, Casimir. "Client-Server Web Apps with JavaScript and Java." O'Reilly Media, 2014. 260 с.
21. Коротич К. О., Колесник Л.В. Використання методів штучного інтелекту для оптимізації щоденного планування // «Комп'ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві», 22 листопада 2023. Харків, Україна. 2023. С. 184-188.