

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко
підпис

«__» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Вебзастосунок для автоматизованого машинного навчання

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408. 22120803

Здобувач

_____ Д. М. Дзундза
підпис

«__» _____ 2024 р.

Керівник канд. техн. наук, доцент

_____ Є. О. Давиденко
підпис

«__» _____ 2024 р.

Консультант канд. техн. наук, доцент

_____ А. О. Алексєєва
підпис

«__» _____ 2024 р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри ___ Є. О. Давиденко

«___» _____ 2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

Видано здобувачу групи 408 факультету комп'ютерних наук

Дзундза Дмитру Миколайовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Вебзастосунок для автоматизованого машинного навчання

Затверджена наказом по ЧНУ від «22» _____ грудня 2023 р. № 269 _____

2. Строк представлення кваліфікаційної роботи «___» _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є вебзастосунок для автоматизованого машинного навчання

4. Перелік питань, що підлягають розробці:

- дослідження предметної області та аналіз існуючих аналогів;
- формування специфікації вимог до програмного забезпечення;
- визначення архітектури для проєктування програмного забезпечення;
- моделювання та проєктування програмного забезпечення;
- розробка програмного забезпечення;
- здійснення тестування роботи програмного забезпечення;

5. Перелік графічних матеріалів

КАЛЕНДАРНИЙ ПЛАН

виконання бакалаврської кваліфікаційної роботи

Тема: «Вебзастосунок для автоматизованого машинного навчання»

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КРБ	20.02.2024	23.02.2024	Виконано
2	Огляд літератури за темою роботи	24.02.2024	26.02.2024	Виконано
3	Складання календарного плану КРБ	26.02.2024	27.02.2024	Виконано
4	Аналіз предметної області	28.02.2024	01.03.2024	Виконано
5	Розробка проектних рішень	04.03.2024	10.03.2024	Виконано
6	Моделювання та конструювання ПЗ	22.03.2024	26.03.2024	Виконано
7	Кодування ПЗ, тестування та апробація розробленого ПЗ, аналіз результатів тестування	28.03.2024	19.05.2024	Виконано
8	Розробка спеціальної частини з охорони праці	20.05.2024	22.05.2024	Виконано
9	Оформлення КРБ та презентації	23.05.2024	26.05.2024	Виконано
10	Відгук керівника КРБ	27.05.2024	31.06.2024	Виконано
11	Попередній захист	03.06.2024	05.06.2024	Виконано
12	Рецензування	06.06.2024	12.06.2024	Виконано
13	Завершення оформлення КРБ та презентації	13.06.2024	14.06.2024	Виконано
14	Захист кваліфікаційної роботи	24.06.2024	27.06.2024	Виконано

Розробив здобувач

Дзундза Д. М.

(прізвище, ім'я, по батькові)

(підпис)

«24» січня 2024 р.

Керівник роботи

канд. техн. наук, доцент Давиденко Є. О.

(посада, прізвище, ім'я, по батькові)

(підпис)

«24» січня 2024 р.

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебзастосунок для автоматизованого машинного навчання»

Здобувач 408 гр.: Дзундза Дмитро Миколайович

Керівник: зав. кафедри ІІЗ, канд. техн. наук, доцент Давиденко Євген
Олександрович.

Кваліфікаційна робота присвячена розробці програмного забезпечення для надання послуг в автоматизованому навчанні штучного інтелекту для не професіоналів.

Актуальність теми: необхідність вдосконалення та впровадження систем штучного інтелекту у сучасному світі. Зараз практично кожен стартап акцентує увагу на використанні штучного інтелекту в своїх інноваціях. Це стає ключовою точкою розвитку нових технологій, що здатні змінити парадигму нашого життя.

Об'єкт: процес автоматизованого машинного навчання на заданому набору даних.

Предмет: програмний засіб для автоматизованого машинного навчання.

Метою роботи є розробка програмного застосунку для допомоги навчання штучного інтелекту початківцям у сфері машинного навчання при отриманні набору даних.

КРБ викладена на 58 сторінки, вона містить 4 розділи, 21 ілюстрацій, 21 таблиці, 16 джерел в переліку посилань

Ключові слова: набори даних, нейронна мережа, python, штучний інтелект, база даних

ABSTRACT

Of Bachelor's thesis

"Web application for automated machine learning"

Acquirer 408 gr.: Dzundza Dmytro Mykolayovych

Head: head of the software engineering department, candidate technical of Sciences, associate professor Davydenko Eugene Oleksandrovich.

This thesis is devoted to the development of software for providing services in the automated training of artificial intelligence for non-professionals.

The relevance of the topic: the necessity to enhance and implement artificial intelligence systems in the modern world. Nowadays, virtually every startup focuses on using artificial intelligence in their innovations. This becomes a pivotal point in the development of new technologies capable of changing the paradigm of our lives.

Object: the process of automated machine learning on a given set of data.

Subject: software for automated machine learning.

The purpose of the work is to develop a software application to help artificial intelligence training for beginners in the field of machine learning when receiving a set of data.

The explanatory note of the bachelor's thesis consists of an introduction, three sections, conclusions and appendices.

KRB is laid out on 58 pages, it contains 4 sections, 21 illustrations, 21 tables, 16 sources in the list of references

Keywords: datasets, neural network, python, artificial intelligence, database.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	5
1.1 Аналіз предметної області.....	5
1.2 Аналіз готових рішень.....	7
1.3 Постановка задачі.....	10
1.4 Специфікація вимог до програмного забезпечення	11
Висновки до розділу 1.....	15
2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	16
2.1 Ескізний проєкт	16
2.2 Технічний проєкт	25
Висновки до розділу 2.....	27
3 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ТА РОЗРОБКА БАЗИ ДАНИХ.....	29
3.1 Вибір засобів розробки.....	29
3.2 Розробка фізичної моделі.....	36
3.3 Проєктування інтерфейсу	38
Висновки до розділу 3.....	42
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ	43
Висновки до розділу 4.....	49
ВИСНОВКИ.....	50
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	52

ПЕРЕЛІК СКОРОЧЕНЬ

ML – Machine learning

JWT - JSON Web Token

ПЗ – Програмне забезпечення

БД – База даних

UML - Unified Modeling Language

HTML - HyperText Markup Language

JS – Java Script

CSS - Cascading Style Sheets

НД – Набір даних

ВСТУП

На даному етапі розвитку спостерігається величезний ріст даних, що вимагає ефективної обробки. Розвиток машинного навчання спрямований на оптимізацію аналізу цих даних. Штучний інтелект стає ключовим у нашому технологічному прогресі, визначаючи нові рівні зручності та ефективності, сприяючи інноваціям, які трансформують наше сприйняття світу та покращують наше щоденне життя.

Актуальність теми визначається необхідністю вдосконалення та впровадження систем штучного інтелекту у сучасному світі. Зараз практично кожен стартап акцентує увагу на використанні штучного інтелекту в своїх інноваціях. Це стає ключовою точкою розвитку нових технологій, що здатні змінити парадигму нашого життя.

Об'єкт: процес автоматизованого машинного навчання на заданому набору даних.

Предмет роботи є програмний засіб для автоматизованого машинного навчання.

Метою роботи є розробка програмного застосунку для допомоги навчання штучного інтелекту початківцям у сфері машинного навчання при отриманні набору даних.

Завдання включають в себе розробку архітектур штучних нейронних мереж та створення інтерфейсу для ефективної взаємодії користувача з платформою.

Сфера застосування даного проекту обумовлена розширенням можливостей використання штучного інтелекту в різних галузях, зокрема в освіті, бізнесі та наукових дослідженнях.

Акцент на простоті використання робить цей проєкт важливим і актуальним для широкого кола користувачів, що допомагає зробити штучний інтелект більш доступним та інтегрованим у наше повсякденне життя.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

AutoML – інноваційна вебплатформа, яка відзначається можливістю отримання штучної нейронної мережі з найоптимальнішою архітектурою для конкретних наборів даних. Завдяки передовим алгоритмам, платформа автоматично створює архітектури штучних нейронних мереж, значно зменшуючи кількість ітерацій для визначення найкращої структури за допомогою здобутих знань з попередніх ітерацій.

Це унікальне рішення дозволяє зберігати ресурси ПК та час, спрощуючи процес отримання оптимальних результатів. Виключно клієнтський доступ до вебсервісу "AutoML" підкреслює його спрямованість на зручність та індивідуальність обслуговування.

У роботі з платформою використовуються наступні ключові поняття:

- **Нікнейм:** особисте вигадане ім'я, яке використовується користувачами для ідентифікації в онлайн-середовищах.
- **Пароль:** конфіденційна послідовність символів для підтвердження особи або її прав.
- **Авторизація:** управління рівнями та засобами доступу до захищеного ресурсу.
- **Обліковий запис або аккаунт:** інформація про користувача, його засоби та права в системі.
- **Клієнт або користувач:** особа, яка використовує обчислювальну техніку для отримання інформації або розв'язання завдань.
- **Замовлення:** доручення виготовлення, виконання або підготовки продукту.
- **Продукт:** замовлена клієнтом комп'ютерна програма чи інші результати творчої діяльності.

- **Штучний інтелект**: розділ комп'ютерної лінгвістики та інформатики, що формалізує завдання, схожі на дії людини.
- **Машинне навчання**: галузь штучного інтелекту, яка використовує статистичні прийоми для "навчання" комп'ютерів.
- **Штучні нейронні мережі або "Модель"**: обчислювальні системи, інспіровані біологічними нейронними мережами.
- **Гіперпараметр**: параметр, що керує процесом навчання, але не вивчається.
- **НД**: набір даних, на якому навчається модель.
- **Деплой**: викладення нової версії сайту на сервер.
- **Експеримент**: система досліджень зі ставленням та обробкою результатів.
- **Тюнер**: програма для підбору оптимальних параметрів моделі.
- **Токен доступу**: об'єкт для ідентифікації користувача та його привілеїв.

JWT [1] представляє собою стандарт токена доступу, побудований на основі формату JSON. Класифікація визначається як система розподілення об'єктів за певними ознаками.

При використанні вебсайту для автоматизації машинного навчання клієнт має можливість зазирнути на головну сторінку, де ознайомлюється із описом сайту та послугами. Також, клієнт може авторизуватися або зареєструвати новий обліковий запис, якщо його ще немає.

Після реєстрації клієнт отримує токен доступу, який надає йому право взаємодіяти з сайтом протягом години. По закінченню терміну придатності токена, клієнт повинен повторно авторизуватись.

Після входу на сайт, клієнт може переглядати розділ наюл. Якщо у клієнта немає НД або він не авторизований, він бачить порожню таблицю. У разі авторизації, клієнт може додавати нові НД та переглядати вже наявні.

Додавши свій набір даних, клієнт може перейти на сторінку «НД» та вибрати конкретний для подальшого перегляду. Тут клієнту виводиться

інформація про назву, розмір та завдання НД. Також, клієнту надається можливість вибору розміру моделі для подальшого навчання та видалення НД.

Під час тренування, клієнт ініціює експеримент, і тюнер починає пошук оптимальних параметрів для моделі, використовуючи певний алгоритм. Тюнер створює історію навчання та зберігає модель для подальшого використання.

Після завершення навчання, клієнт може переглянути результати та обрати оптимальну модель для завантаження та використання на сайті.

1.2 Аналіз готових рішень

На сучасний момент на ринку програмного забезпечення існує багато програмних комплексів, що забезпечують тією чи іншою мірою комплекс послуг по розробці штучного інтелекту під певну задачу. Тому перед початком розробки власного програмного забезпечення був проведений аналіз вже існуючих систем, основною метою якого було виявлення найбільш вдалих рішень з погляду користувача, а також визначення недоліків в роботі та відмінності в існуючих комплексах для їх наступного врахування під час розробки. Розглянув наступні вебсервіси:

- Google Cloud AutoML [2];
- Microsoft Automated Machine Learning [3].

Google Cloud AutoML

Cloud AutoML дозволяє розробникам з обмеженим досвідом у машинному навчанні навчати високоякісні моделі, що відповідають їх бізнес-потребам. Інтерфейс веб ресурсу «Cloud AutoML» представлено на рисунку 1.1.

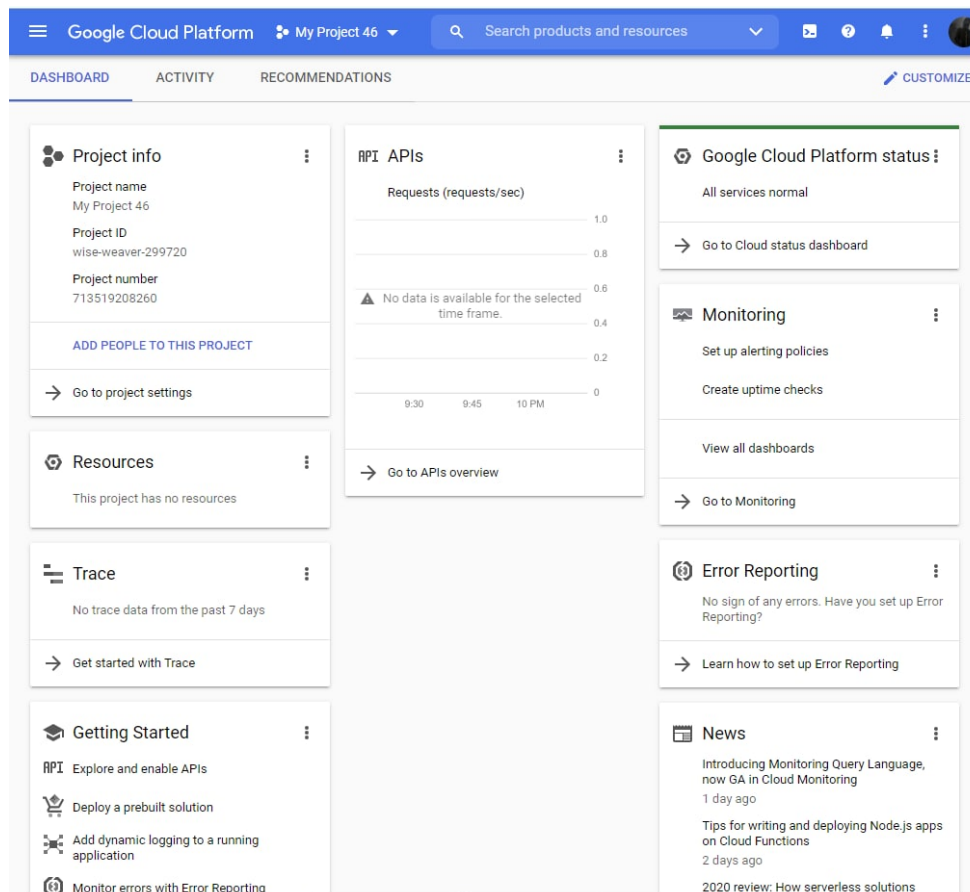


Рисунок 1.1 – Інтерфейс вебсервісу «Cloud AutoML»

В «Cloud AutoML» автоматизовані наступні функції:

- реєстрація та авторизація;
- створення нового проєкт;
- моніторинг використаних ресурсів;
- обирання розмірів штучної нейронної мережі;
- завантаження власних, або підключення зовнішніх НД;
- обирання методів підготовки даних;
- навчання штучної нейронної мережі;
- обирання кращої натренованої штучної нейронної мережі;
- автоматизований деплой готового продукту.

Переваги AutoML:

- не потрібно мати особливих навичок у створенні штучних нейронних мереж та програмуванні для створення високоякісних моделей.
- великі обчислювальні ресурси;

– існує велика кількість рішень під більшість бізнес потреб. Від регресії та класифікації, до задач, що потребують аналіз природньої мови.

Недоліки AutoML:

– неможливість завантаження моделей на власний ПК;
– високі ціни;
– потрібно мати навички з програмування та використання API для використання моделей.

Microsoft Automated Machine Learning

Automated Machine Learning надає можливість професійним та непрофесійним користувачам big data швидко створювати моделі машинного навчання. Автоматизація трудомісткі та повторюваних завдань розробки моделі за допомогою проривних досліджень та деплою моделей. Інтерфейс веб ресурсу «Cloud AutoML» представлено на рисунку 1.2.

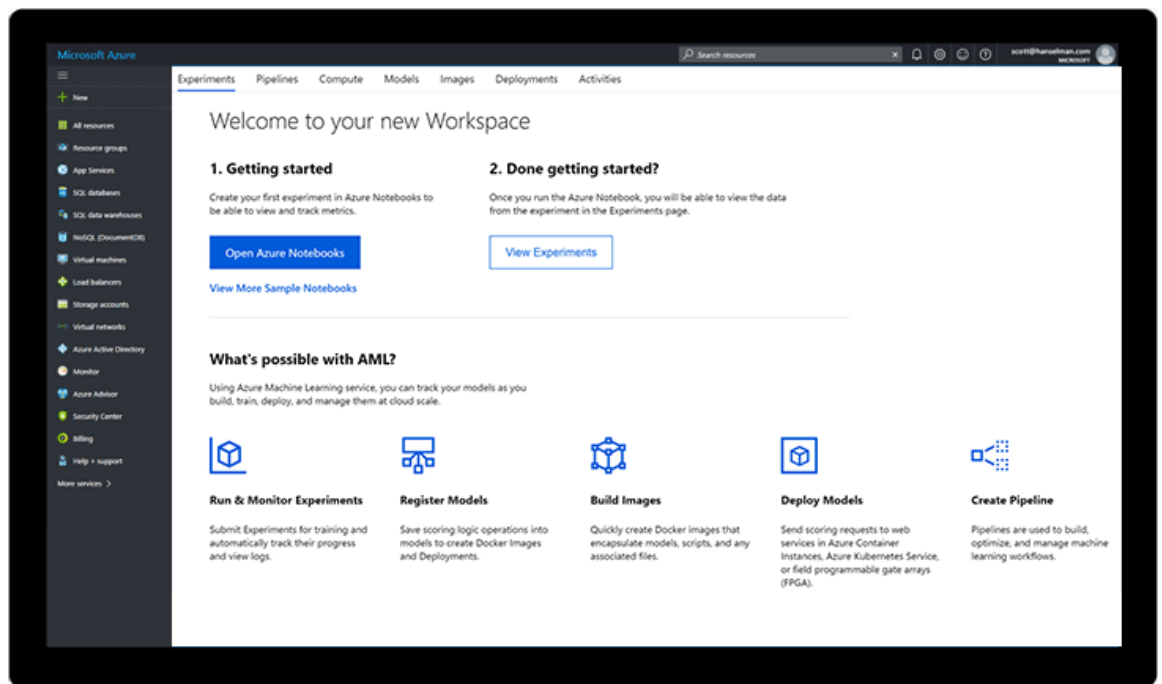


Рисунок 1.2 – Інтерфейс вебсервісу «Microsoft Automated Machine Learning»

В «Microsoft Automated Machine Learning» автоматизовані наступні функції:

- реєстрація та авторизація;
- створення нового проекту;

- моніторинг використаних ресурсів;
- завантаження власних, або підключення зовнішніх НД;
- обирання методів підготовки даних;
- навчання штучної нейронної мережі;
- обирання кращої натренованої штучної нейронної мережі;
- автоматизований деплой готового продукту.

Переваги AutoML:

- автоматичний трейн багатьох моделей та порівняння найкращих.
- великі обчислювальні ресурси;
- безкоштовний період у 750 годин.

Недоліки AutoML:

- неможливість завантаження моделей на власний ПК;
 - високі ціни незважаючи на безкоштовний період;
 - потрібно мати навички з програмування та використання API для використання моделей.
- обмеженість у виборі задач для непрофесіоналів. Присутні легкі рішення тільки для табличних структурованих даних.

1.3 Постановка задачі

Потрібно розробити вебсервіс, де користувач матиме змогу отримати найкращу модель, підігнану під його дані та вибрану задачу. Клієнтам будуть доступні наступні функції:

- реєстрація;
- логін;
- вихід;
- генерація токену;
- перевірка статусу токену;
- додавання НД;
- перегляд НД;
- видалення НД;

- вибір розміру моделей;
- трейн моделей;
- перегляд результатів;
- скачування моделі.

Вхідні дані:

- інформація для авторизації;
- токен;
- інформація про НД;
- інформація про модель;
- НД.

Вихідні дані:

- результати навчання моделі;
- результат реєстрації;
- результат входу;
- результат перевірки токену;
- моделі;
- список НД;
- опис НД;
- токен доступу;
- статус токену.

1.4 Специфікація вимог до програмного забезпечення

Призначення системи (застосунку), для якої розробляється програмне забезпечення

Призначенням ПЗ, що розробляється, є автоматизація процесу навчання штучних нейронних мереж.

Погодження, що ухвалені в програмній документації

Погоджень не ухвалено.

Межі проєкту ПЗ

Крайня дата завершення роботи над ПЗ – 01.06.2024р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Дане ПЗ не має обмежень у сферах його застосування, застосунок можна використовувати у повсякденному житті та інших сферах діяльності.

Характеристики користувачів

Основні характеристики користувачів: доступ до мережі Інтернет, наявність ПК або мобільного пристрою.

Загальна структура і склад системи

Основні частини для створення програмного забезпечення: вебзастосунок, сервер та база даних.

Загальні обмеження

Обмеження для роботи з ПЗ – наявність доступу до мережі Інтернет.

ФУНКЦІЇ ВЕБЗАСТОСУНКУ

Авторизація, реєстрація, та вихід із системи, генерація та перевірка статусу токенів забезпечують контроль доступу та безпеку користувацьких сесій. Додавання, перегляд та видалення НД дозволяють користувачам управляти даними, які вони використовують для тренування моделей. Вибір розміру моделей та їх тренування дають можливість користувачам налаштувати алгоритми машинного навчання за своїми потребами, тоді як перегляд результатів тренування та скачування моделей забезпечує можливість аналізувати ефективність та застосовувати навчені моделі у власних проєктах. Кожна дія користувача пов'язана з обробкою вхідних та вихідних даних, включаючи інформацію для авторизації, токени доступу, інформацію про НД та моделі, результати навчання, опис НД та статуси токенів, що гарантує комплексний підхід до управління даними і моделями машинного навчання в середовищі застосунку.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

В цьому застосунку основним джерелом вхідних даних є користувач. Користувачі вручну вводять інформацію, необхідну для формування та

тренування моделей машинного навчання, включаючи НД та специфікації моделей.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Вимоги до даного пункту відсутні.

Вимоги до способів організації, збереження та ведення інформації

В якості БД обрано PostgreSQL [4].

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Немає строгих вимог до апаратного забезпечення. Для роботи застосунку достатньо мати персональний комп'ютер або ноутбук, які здатні запускати сучасні веб браузері.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Архітектура складається з клієнтської частини, серверної частини та БД.

Системне програмне забезпечення

Вебзастосунок має бути розроблено з використанням python та фреймворком Flask [5] і Tensorflow [6]. В якості БД для вебзастосунку обрано PostgreSQL.

Мережне програмне забезпечення

Для створення ПЗ використано ОС Windows 11, у якості IDE обрано PyCharm [7], для перегляду вебсторінок – браузер Google Chrome.

Програмне забезпечення ведення інформаційної бази

CRUD-операції виконуються через БД PostgreSQL.

Мова і технологія розробки ПЗ

Розробка програмного забезпечення ведеться на мові Python [8] за допомогою фреймворку Flask і Tensorflow.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс користувача має бути розроблений так, щоб відповідати усім дизайнерським стандартам, забезпечуючи високу зручність та ефективність

взаємодії з системою. Це допоможе скоротити час, який користувач потребує для вивчення функціональності застосунку.

Апаратний інтерфейс

Апаратний інтерфейс включає пристрій користувача, такий як персональний комп'ютер або мобільний телефон, що використовується для доступу до вебзастосунку.

Програмний інтерфейс

Flask – це легкий фреймворк для розробки вебзастосунків, який використовує мову програмування Python. Він пропонує гнучкі інструменти для маршрутизації, роботи з базами даних та автентифікації. Flask також підтримує численні розширення, які спрощують процес розробки та управління кодом.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

ПЗ доступне для використання будь-яким користувачем, який має доступ до вебзастосунку, при наявності апаратного забезпечення і доступу до мережі Інтернет.

Переносимість

Програмне забезпечення сумісне з будь-якою операційною системою, включаючи Windows, macOS, Linux, а також мобільними платформами.

Продуктивність

Продуктивність роботи ПЗ залежить від швидкості підключення до мережі Інтернет і величини НД, обраної величини штучної нейронної мережі. Час виконання запитів може перевищувати і хвилину.

Надійність

Під час реєстрації та користування застосунком усі надані користувачем дані залишаються конфіденційними. Програмне забезпечення гарантує, що доступ до особистих даних має лише власник цих даних, для чого необхідно пройти авторизацію в системі.

Висновки до розділу 1

В першому розділі кваліфікаційної роботи бакалавра розглянуто інноваційну вебплатформу для автоматизованого машинного навчання, яка пропонує зручний та індивідуалізований підхід до отримання штучних нейронних мереж з оптимальною архітектурою для конкретних наборів даних. Платформа використовує передові алгоритми для автоматичного створення архітектур, що відзначається значним зменшенням кількості ітерацій для визначення оптимальної структури.

Описано функціонал вебсайту, що спрямований на автоматизацію машинного навчання, з підрозділами від реєстрації та авторизації користувачів до роботи з НД та тренуванням моделей. Розглянуті етапи від входу на сайт до вибору оптимальної моделі для подальшого використання.

В аналізі існуючих систем також виявлено ключові аспекти та вдалі рішення, які можна врахувати під час розробки власної платформи. Проаналізовані Google Cloud AutoML та Microsoft Automated Machine Learning як потенційні конкуренти на ринку програмного забезпечення для розвитку штучного інтелекту.

2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Ескізний проєкт

2.1.1 Вибір засобів моделювання

Розробка ПЗ виконуються з використанням об'єктно орієнтованої методології. Для проєктування комп'ютерних систем та ПЗ використовують діаграми мови UML [9], за допомогою яких можна візуалізувати, специфікувати, конструювати і документувати артефакти програмних систем.

Для моделювання було обрано програму Rational Rose [10], яка є одним з найбільш популярних інструментів для створення UML-діаграм. Rational Rose дозволяє ефективно проєктувати архітектуру програмного забезпечення, забезпечуючи інтуїтивно зрозумілий інтерфейс та потужні засоби для візуалізації і управління складними проєктами.

Rational Rose є потужним інструментом для об'єктно-орієнтованого аналізу та проєктування, який підтримує широкий спектр функціональних можливостей для розробників програмного забезпечення. Основні функціональні можливості Rational Rose включають:

- 1) підтримка UML;
- 2) зворотнє і пряме проєктування;
- 3) інтеграція з іншими інструментами;
- 4) моделювання баз даних;
- 5) генерація документації;
- 6) відстеження вимог;
- 7) підтримка різних мов програмування.

Ці можливості роблять Rational Rose потужним і зручним інструментом для моделювання та проєктування складних програмних систем, що значно підвищує ефективність і якість розробки програмного забезпечення.

2.1.2 Розробка контекстної моделі системи

Проектування програмного забезпечення починається із розробки контекстної діаграми, яка характеризує функції системи і зв'язки системи з навколишнім середовищем. Тому, було виділено основну діючу особу: користувач.

Контекстну діаграму можна переглянути на рисунку 2.1.

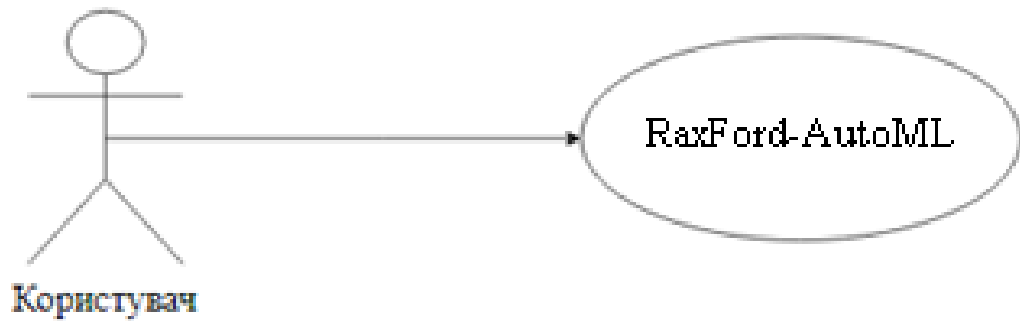


Рисунок 2.1 – Контекстна діаграма для вебсайту «AutoML»

2.1.3 Діаграма варіантів використання

Для визначення вимог, яким повинен задовольняти сайт було використано діаграму варіантів використання. Діаграма варіантів використання сайту з представлена на рисунку 2.2.

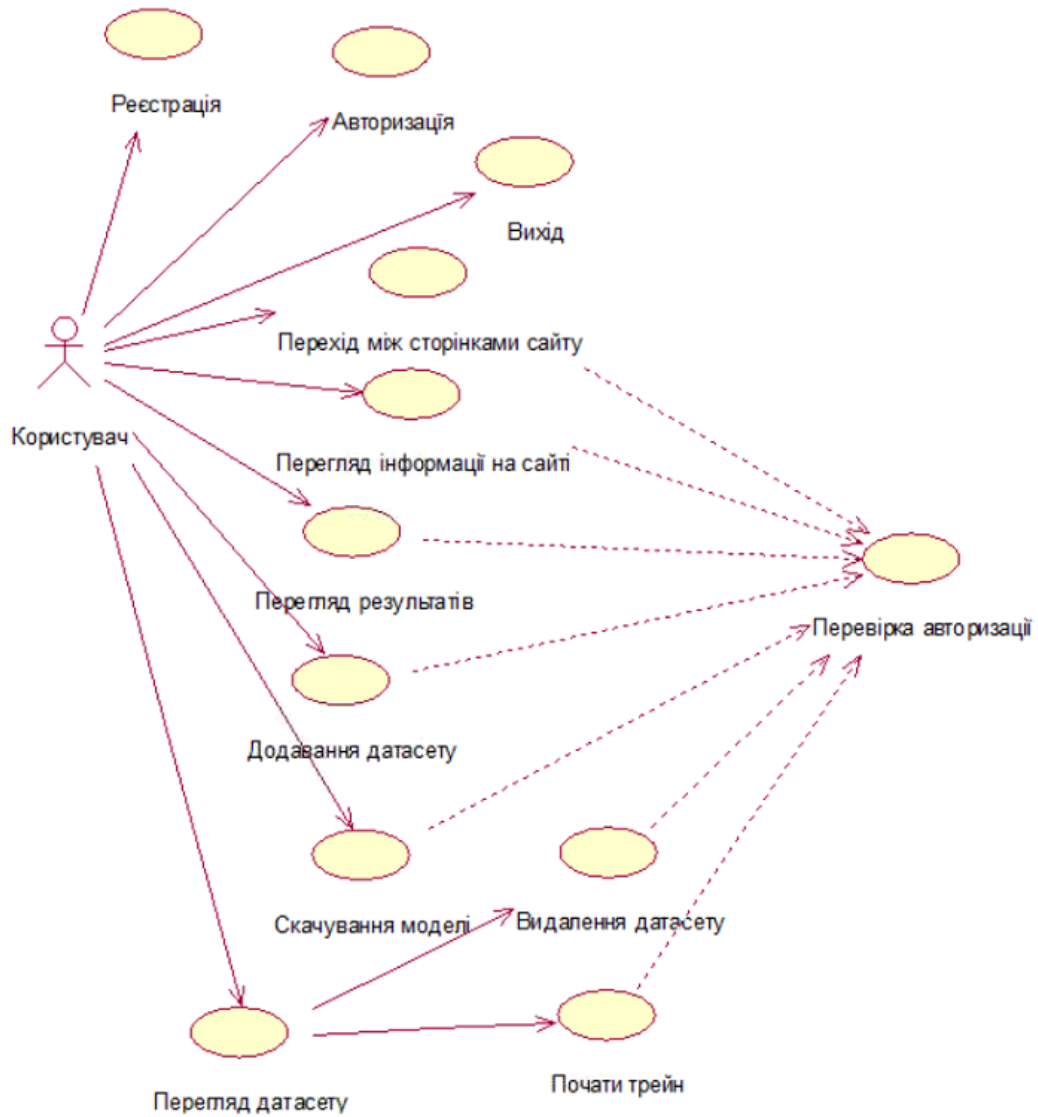


Рисунок 2.2 – Діаграма варіантів використання вебсайту «AutoML»

Специфікації варіантів використання представлені в таблицях 2.1-2.11.

Таблиця 2.1 – Специфікація прецеденту «Реєстрація»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу зареєструватись у системі.
Передумови	Не визначені.
Основний потік подій	Користувач ініціює відкриття інтерфейсу реєстрації. Система відкриває інтерфейс реєстрації. Користувач заповнює необхідні поля інтерфейсу, система перевіряє валідність введених даних і створює новий акаунт. Якщо дані невалідні – виконується альтернативний потік подій.

Продовження таблиці 2.1

1	2
Альтернативний потік подій	Система повідомляє користувачу, про невалідність введених даних. Дійовій особі пропонується можливість повторити введення або завершити варіант використання..
Пост умови	Не визначені.

Таблиця 2.2 – Специфікація прецеденту «Перевірка ідентифікації»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу пройти перевірку статусу токена доступу.
Передумови	Користувач повинен мати ідентифікатор (токен) доступу в систему.
Основний потік подій	Користувач ініціює перевірку ідентифікації. Користувач надає системі токен доступу. Система розшифровує токен та перевіряє його на термін пригодності. Система повертає статус токена. Якщо термін пригодності токена вийшов – виконується альтернативний потік подій.
Альтернативний потік подій	Невірний токен: ввели невірний токен або термін дії токена закінчився; дійовій особі пропонується можливість повторити введення або завершити варіант використання.
Пост умови	Не визначені.

Таблиця 2.3 – Специфікація прецеденту «Авторизація»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу авторизуватись у системі.
Передумови	Не визначені.
Основний потік подій	Користувач ініціює відкриття інтерфейсу авторизації. Система відкриває інтерфейс авторизації. Користувач заповнює необхідні поля інтерфейсу, система перевіряє валідність введених даних і повертає згенерований токен. Якщо дані невалідні – виконується альтернативний потік подій.
Альтернативний потік подій	Система повідомляє користувачу, про невалідність введених даних. Дійовій особі пропонується можливість повторити введення або завершити варіант використання.
Пост умови	Не визначені.

Таблиця 2.4 – Специфікація прецеденту «Вихід»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу вивести токен з використання.
Передумови	Користувач надає ідентифікатор (токен) доступу в систему.
Основний потік подій	Користувач ініціює виведення токена з експлуатації. Система перевіряє токен на валідність та запам'ятовує потрібний токен та не дозволяє у майбутньому його використовувати. Якщо токен невалідний – виконується альтернативний потік подій.
Альтернативний потік подій	Система повідомляє користувачу, про невалідність введених даних. Дійовій особі пропонується можливість повторити введення або завершити варіант використання.
Пост умови	Не визначені.

Таблиця 2.5 – Специфікація прецеденту «Перехід між сторінками сайту»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу перейти між сторінками.
Передумови	Користувач надає ідентифікатор (токен) доступу в систему.
Основний потік подій	Користувач ініціює перехід між сторінками на сайті. Система перевіряє валідність ідентифікатору та повертає спеціальну версію сторінки. Якщо ідентифікатор невалідний – виконується альтернативний потік подій
Альтернативний потік подій	Дійовій особі надається стандартна версія сторінки.
Пост умови	Не визначені.

Таблиця 2.6 – Специфікація прецеденту «Перегляд інформації»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє Користувачу переглянути інформацію на сайті.
Передумови	Користувач надає ідентифікатор (токен) доступу в систему.
Основний потік подій	Користувач ініціює перегляд інформації на сторінці. Система перевіряє валідність ідентифікатору та повертає спеціальні дані для Користувача. Якщо ідентифікатор невалідний – виконується альтернативний потік подій
Альтернативний потік подій	Користувачеві повертається стандартна інформація.
Пост умови	Не визначені.

Таблиця 2.7 – Специфікація прецеденту «Додавання НД»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє Користувачу додати НД на сервер.
Передумови	Користувач надає (токен) доступу в систему.
Основний потік подій	Користувач ініціює додавання НД система відкриває інтерфейс додавання НД. Система перевіряє валідність ідентифікатора, валідність НД та завантажує його на диск. Якщо ідентифікатор невалідний -альтернативний потік подій
Альтернативний потік подій	Дійовій особі пропонується можливість повторити введення або завершити варіант використання.
Пост умови	Не визначені.

Таблиця 2.8 – Специфікація прецеденту «Перегляд НД»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє Користувачу переглянути НД.
Передумови	Користувач надає ідентифікатор (токен) доступу в систему. Користувач обирає НД.
Основний потік подій	Користувач ініціює перегляд НД. Система перевіряє валідність ідентифікатора. Система повертає інформацію про НД. Якщо ідентифікатор невалідний – виконується альтернативний потік подій.
Альтернативний потік подій	Дійовій особі пропонується можливість ввести новий токен або завершити варіант використання.
Пост умови	Не визначені.

Таблиця 2.9 – Специфікація прецеденту «Видалення НД»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє Користувачу видалити НД з системи.
Передумови	Користувач надає ідентифікатор (токен) доступу в систему. Користувач обирає НД.
Основний потік подій	Користувач ініціює видалення НД. Система перевіряє валідність ідентифікатора. Система видаляє НД. Якщо ідентифікатор невалідний – виконується альтернативний потік подій.
Альтернативний потік подій	Дійовій особі пропонується можливість ввести новий токен або завершити варіант використання.
Пост умови	Не визначені.

Таблиця 2.10 – Специфікація прецеденту «Почати трейн»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє Користувачу почати трейн.
Передумови	Користувач надає ідентифікатор (токен) доступу в систему. Користувач обирає НД.
Основний потік подій	Користувач ініціює видалення НД. Система перевіряє валідність ідентифікатора. Система починає трейн. Якщо ідентифікатор невалідний – виконується альтернативний потік подій.
Альтернативний потік подій	Дійовій особі пропонується можливість ввести новий токен або завершити варіант використання.
Пост умови	Не визначені.

Таблиця 2.11 – Специфікація прецеденту «Перегляд результатів»

Складова	Опис
1	2
Короткий опис	Даний варіант використання дозволяє Користувачу переглянути результати навчання.
Передумови	Користувач надає ідентифікатор (токен) доступу в систему.
Основний потік подій	Користувач ініціює перегляд результатів навчання. Система перевіряє валідність токена. Система повертає сторінку з поточними результатами навчання. Якщо ідентифікатор невалідний – виконується альтернативний потік подій
Альтернативний потік подій	Користувачеві не повердаються результати.
Пост умови	Не визначені.

2.1.4 Концептуальна модель

Для опису предметної області було побудовано концептуальну модель.

Опис зв'язків між сутностями вебсайту «AutoML» показаний у таблиці 2.12.

Таблиця 2.12 – Таблиця зв'язки між сутностями

Сутності	Тип зв'язку	Зміст зв'язку
1	2	3
Користувач НД	1:М	Один користувач може загрузити багато НД
НД Експеримент	1:М	Один НД може бути в багатьох експериментах

Продовження таблиці 2.12

1	2	3
Експеримент Тюнер	1:M	Один експеримент може мати багато тюнерів
Тюнер Модель	1:M	Один тюнер може мати багато моделей
Модель Оптимізатор	1:1	Одна модель може мати один оптимізатор

Концептуальна модель вебсайту «AutoML» представлена на рисунку 2.3.

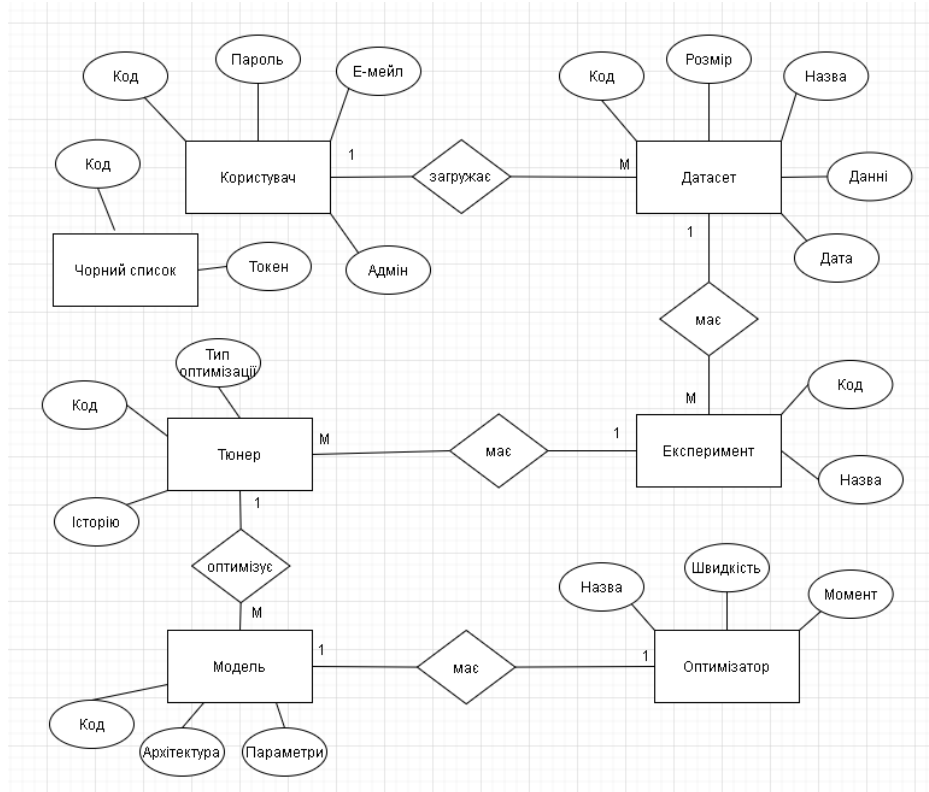


Рисунок 2.3 – Концептуальна модель вебсайту «AutoML»

2.1.5 Діаграма станів

Для відображення переходів між сторінками сайту було побудовано діаграму станів, яка представлена на рисунку 2.4. Переходи станів представлені у таблиці 2.13.

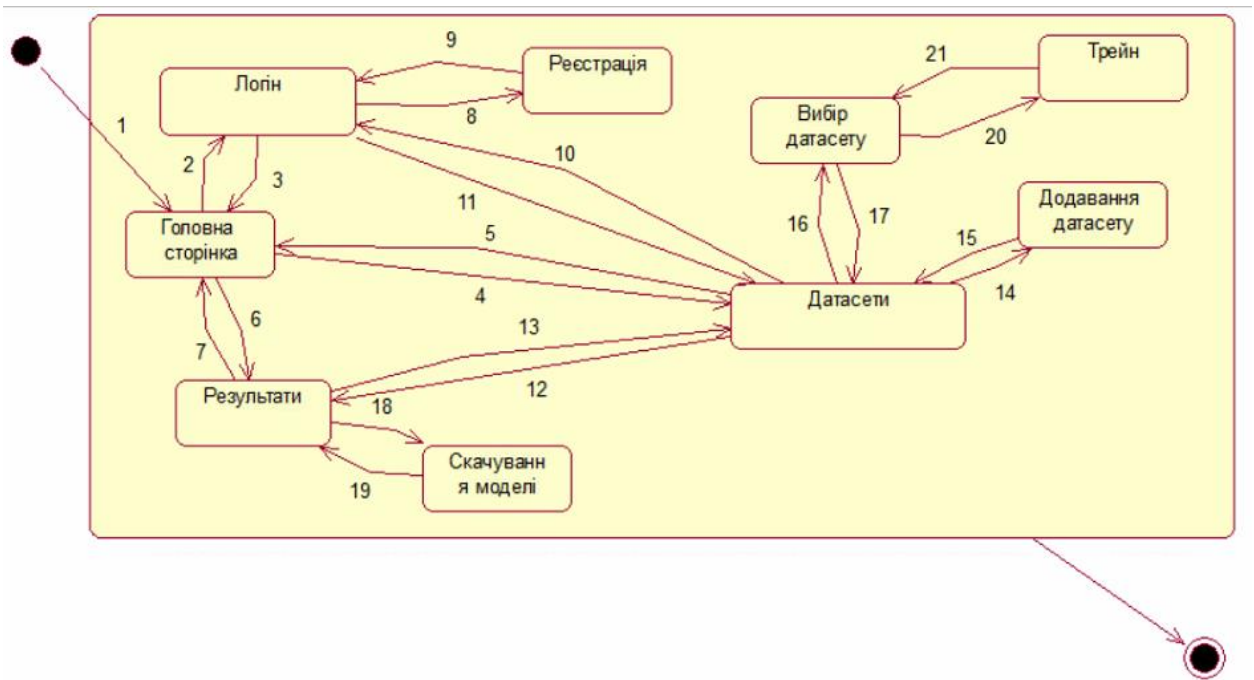


Рисунок 2.4 – Діаграма станів вебсайту «AutoML»

Таблиця 2.13 – Специфікація діаграми переходів станів

Номер	Умова/Дія
1	Ініціювання переходу до головної сторінки
3,7,5	Ініціювання переходу до головної сторінки / Повернення до головної сторінки
2,9,10	Ініціювання переходу до входу/ Перехід до сторінки логін
4,11,13,15,17	Ініціювання переходу до НД / Перехід до НД
6,12,19	Ініціювання переходу до результатів/ Перехід до результатів
8	Ініціювання переходу до реєстрації/ Перехід до реєстрації
14	Ініціювання переходу до форми додавання НД / Перехід до форми додавання НД
16,21	Ініціювання переходу до форми вибору НД / Перехід до форми вибору НД
20	Ініціювання навчання/ Перехід до навчання
18	Ініціювання скачування моделі/Скачування моделі

2.2 Технічний проєкт

2.2.1 Діаграма взаємодії

Для опису поведінки об'єктів вебсайту «AutoML»

Для приділення уваги часовим упорядкуванням подій було застосовано діаграму послідовності. Для представлення не тільки послідовності взаємодії, але і всіх структурних відносин між об'єктами, що беруть участь в цій взаємодії, було застосовано діаграму кооперації.

Діаграма послідовності для суб'єкта «Користувач» при виконанні варіанта використання «Додавання НД» представлена на рисунку 2.10.

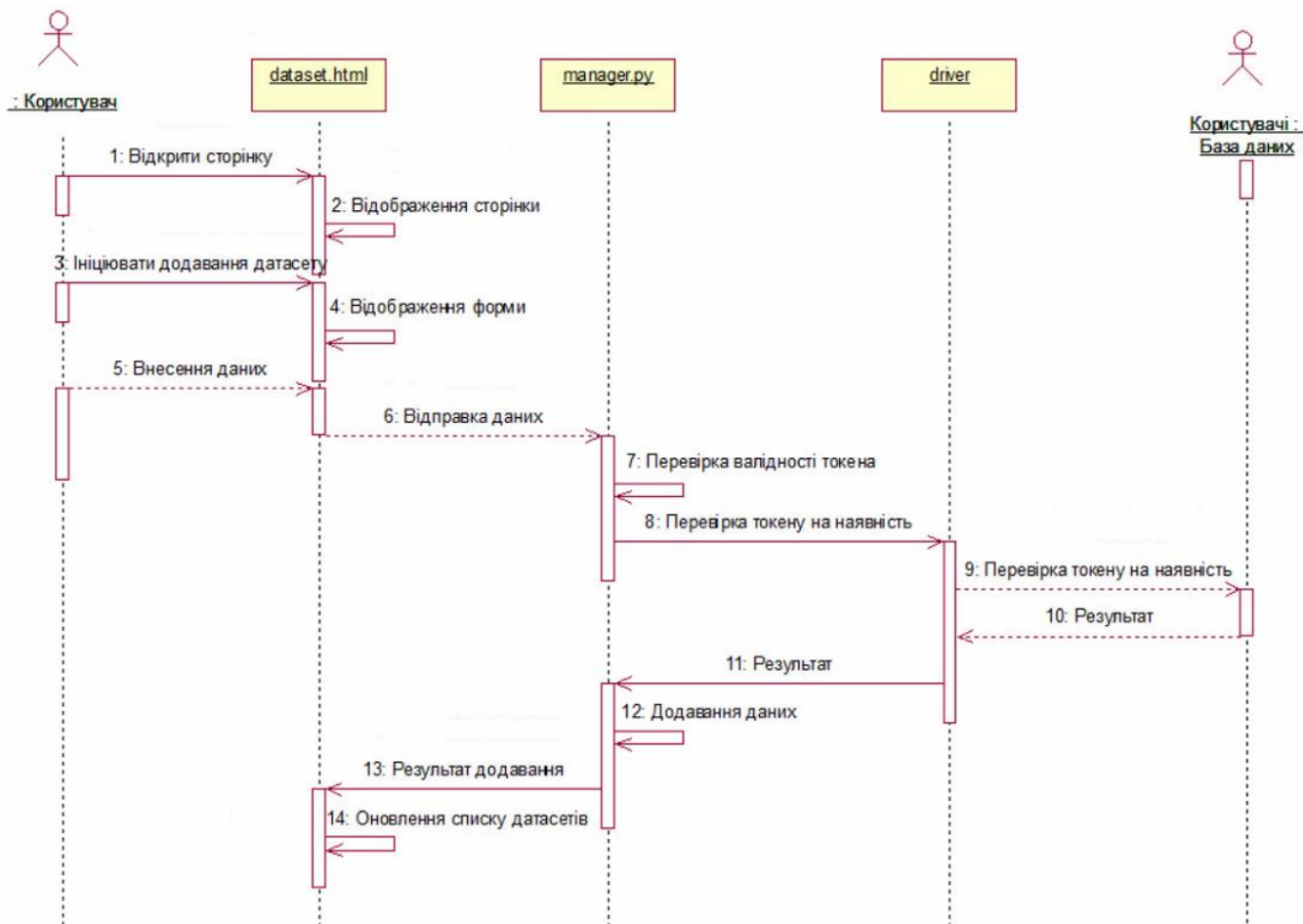


Рисунок 2.5 – Діаграма послідовності для суб'єкта «Користувач» при виконанні варіанта використання «Додавання НД»

Діаграма кооперації для суб'єкта «Користувач» при виконанні варіанта використання «Додавання НД» представлена на рисунку 2.11.



Рисунок 2.6 – Діаграма кооперації для суб'єкта «Користувач» при виконанні варіанта використання «Додавання НД»

2.2.2 Діаграма класів

Для представлення взаємозв'язків між окремими сутностями предметної області, такими як об'єкти і підсистеми, а також описує їх внутрішню структуру і типи відношень, було побудовано діаграму класів.

Діаграма класів представлена на рисунку 2.12.

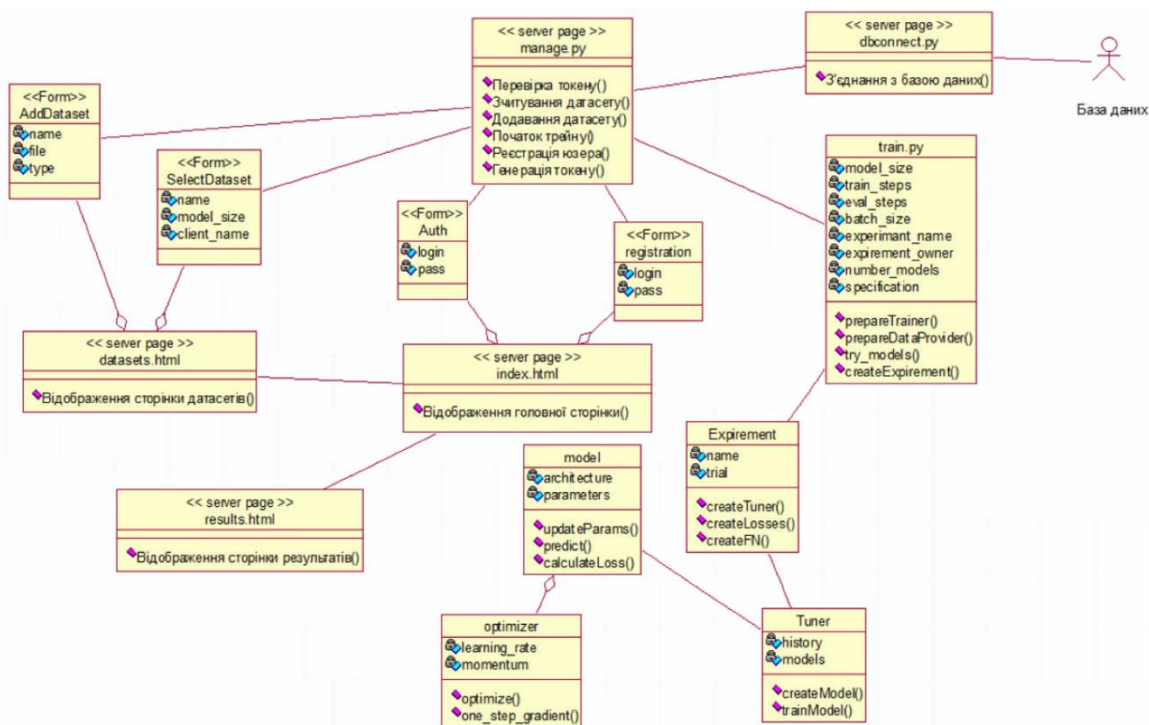


Рисунок 2.7 – Діаграма класів вебсайту «AutoML»

2.2.3 Діаграми діяльності

Побудовано діаграму діяльності для варіанта використання «Додавання НД» представлена на рисунку 2.13.

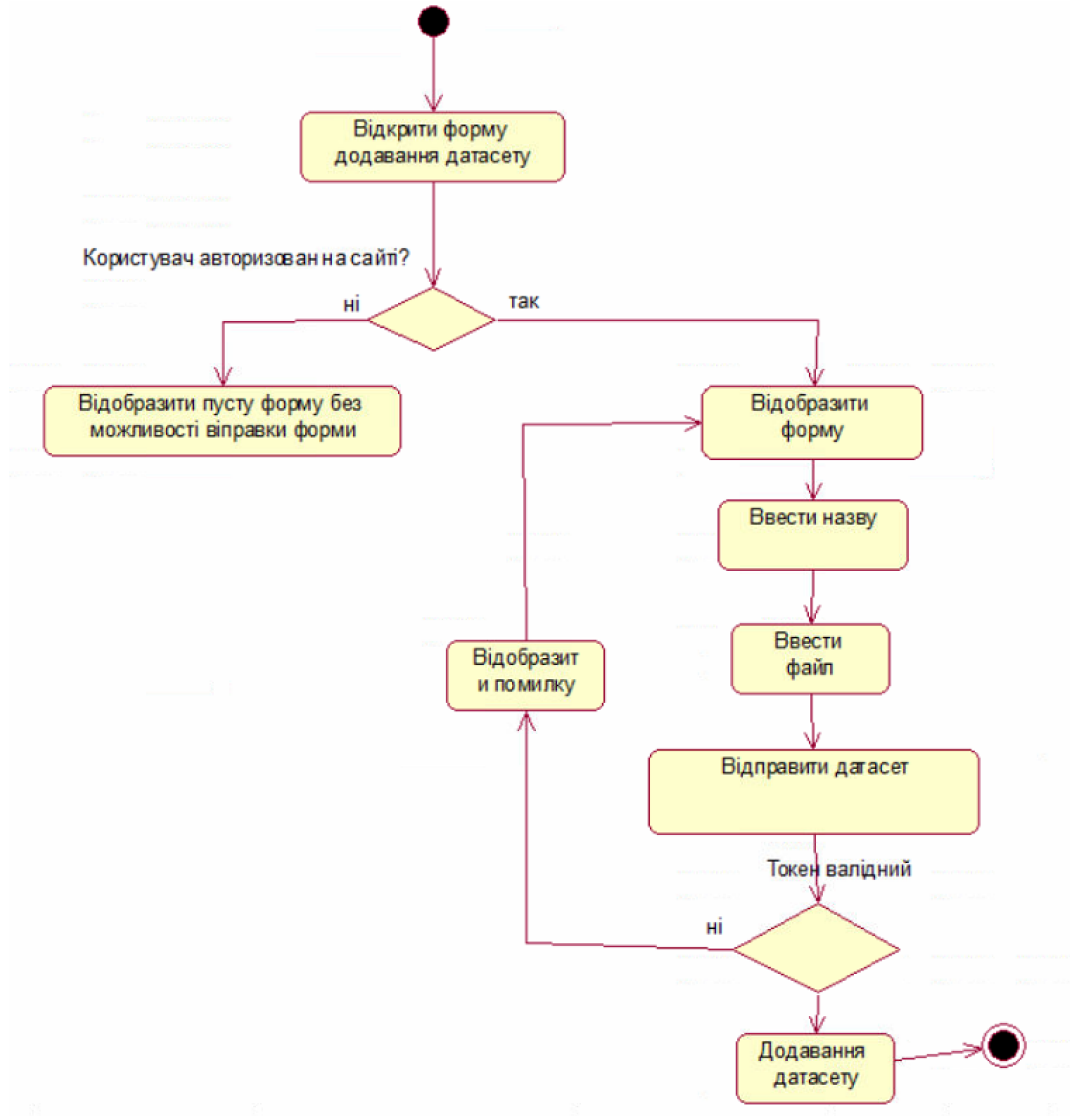


Рисунок 2.8 –Діаграма діяльності для варіанта використання «Додавання НД»

Висновки до розділу 2

У другому розділі приведено проєкт програмного забезпечення, який складається з:

- діаграми варіантів використання;
- станів;

- послідовності;
- кооперації;
- класів;
- діяльності;
- розгортання.

Також в проєкті програмного забезпечення наведено проєктування інтерфейсу та концептуальну модель.

3 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ТА РОЗРОБКА БАЗИ ДАНИХ

3.1 Вибір засобів розробки

Для створення сучасного вебсайту було обрано декілька ключових технологій, які забезпечують необхідну функціональність, продуктивність та надійність. Однією з основних мов програмування стала Python, яка славиться своєю простотою, читабельністю та багатofункціональністю. Крім того, для управління даними було обрано систему керування базами даних PostgreSQL, яка забезпечує надійне та ефективне зберігання і обробку даних.

Python — це інтерпретована мова програмування, що дозволяє швидко розробляти програмні рішення завдяки своїй зрозумілій та читабельній синтаксису. Її часто використовують для розробки вебзастосунків, обробки даних, автоматизації завдань та навіть для наукових досліджень. Однією з головних переваг Python є велика кількість доступних бібліотек і фреймворків, які значно спрощують процес розробки та розширюють можливості мови.

Для управління базами даних обрано PostgreSQL, яка є однією з найпопулярніших об'єктно-реляційних систем керування базами даних. PostgreSQL відома своєю надійністю, масштабованістю та підтримкою стандартів SQL. Вона забезпечує ефективне зберігання великих обсягів даних та високу продуктивність при виконанні складних запитів.

Завдяки поєднанню Python та PostgreSQL розробники можуть створювати високоякісні вебзастосунки, які відповідають сучасним вимогам до продуктивності та надійності. Використання цих технологій дозволяє ефективно обробляти великі обсяги даних, забезпечувати динамічну взаємодію з користувачами та підтримувати високу швидкість роботи додатка.

Цей вибір технологій дозволяє реалізувати всі необхідні функції для сучасного вебсайту, включаючи обробку користувацьких запитів, зберігання та управління даними, динамічну генерацію вебсторінок та інтеграцію з іншими сервісами. Таким чином, Python та PostgreSQL стають основою для створення

надійного та масштабованого вебзастосунку, що відповідає сучасним стандартам якості та безпеки.

Python є однією з найпопулярніших мов програмування в сучасному світі завдяки своїй універсальності та простоті використання. Вона була розроблена Гвідо ван Россумом у 1990 році з метою створення мови, яка буде проста у вивченні та використанні, але при цьому потужна та гнучка. Python є інтерпретованою мовою програмування, що означає, що код виконується безпосередньо без необхідності попередньої компіляції, що значно спрощує процес розробки та тестування.

Однією з головних особливостей Python є її об'єктно-орієнтований підхід, що дозволяє розробникам створювати модульні та повторно використовувані компоненти коду. Крім того, Python підтримує строго динамічну типізацію, що означає, що типи змінних визначаються під час виконання програми, а не на етапі компіляції. Це надає додаткову гнучкість при написанні коду.

Python широко використовується для веброзробки завдяки своїм потужним фреймворкам, таким як Django та Flask. Ці фреймворки надають розробникам інструменти для швидкого створення вебдодатків з підтримкою обробки даних, генерації динамічних сторінок, роботи з куками та іншими важливими функціями. Flask, наприклад, є мікрофреймворком, який дозволяє розробникам створювати вебдодатки з мінімальними налаштуваннями, зосереджуючись на основному функціоналі додатка.

Окрім веброзробки, Python активно використовується в інших галузях, таких як наука про дані, машинне навчання, автоматизація завдань та створення скриптів. Завдяки таким бібліотекам, як NumPy, pandas, scikit-learn та TensorFlow, Python став основним інструментом для аналітиків даних та дослідників. Ці бібліотеки забезпечують широкий спектр інструментів для обробки та аналізу даних, створення моделей машинного навчання та їхньої інтеграції у виробничі системи.

Python є безкоштовною мовою програмування з відкритим вихідним кодом. Це означає, що будь-хто може завантажити, використовувати та модифікувати Python без будь-яких обмежень. Офіційний вебсайт Python-проєкту надає останні версії мови та велику кількість ресурсів для вивчення та підтримки спільноти розробників.

Flask є мікрофреймворком для вебдодатків, який був створений для забезпечення простоти і гнучкості при розробці вебдодатків. Він побудований на базі двох потужних інструментів: Werkzeug та Jinja2. Werkzeug є інструментарієм для роботи з WSGI (Web Server Gateway Interface), який забезпечує взаємодію між вебсервером і вебдодатком. Jinja2, у свою чергу, є рушієм шаблонів, який дозволяє легко генерувати HTML-сторінки на основі даних.

Flask надає розробникам можливість швидко створювати вебдодатки з мінімальною кількістю налаштувань та обмежень. Завдяки своїй модульній архітектурі, Flask дозволяє додавати необхідні функції шляхом підключення різних розширень, що робить його дуже гнучким та адаптивним до потреб конкретного проєкту. Це особливо корисно для невеликих і середніх проєктів, де важливо швидко реалізувати основний функціонал без необхідності занурюватися в складні налаштування.

Однією з ключових особливостей Flask є його вбудований відлагоджувач, який дозволяє розробникам ефективно виявляти і виправляти помилки в коді під час розробки. Крім того, Flask підтримує юніт-тести, що дозволяє автоматизувати процес тестування коду і забезпечувати високу якість програмного забезпечення. Управління запитами RESTful також є однією з важливих функцій Flask, що робить його ідеальним вибором для створення API і вебсервісів.

Flask широко використовується в індустрії завдяки своїй простоті та потужності. Він ідеально підходить для створення прототипів, MVP (Minimum Viable Product) та невеликих вебдодатків, де важливо швидко реалізувати ідею і перевірити її на практиці. Flask також добре підходить для розробки

мікросервісів, де кожен сервіс може бути невеликим та незалежним додатком, який легко інтегрується з іншими сервісами.

Серед інших переваг Flask можна виділити його хорошу документацію і активну спільноту. Документація Flask детально описує всі основні аспекти фреймворку, що значно спрощує процес вивчення та використання. Активна спільнота розробників постійно підтримує та розширює Flask, додаючи нові функції та розширення, що робить його ще більш потужним і зручним у використанні.

Таким чином, Flask є потужним інструментом для розробки вебдодатків на Python, який поєднує в собі простоту, гнучкість і ефективність. Він дозволяє швидко реалізувати необхідний функціонал, забезпечуючи високу продуктивність і якість програмного забезпечення.

TensorFlow є однією з найбільш популярних платформ для машинного навчання з відкритим вихідним кодом, розробленою компанією Google. Вона надає широкий спектр інструментів та бібліотек, які дозволяють розробникам і дослідникам створювати, тренувати та розгортати моделі машинного навчання на різних платформах. Завдяки своїй гнучкості та масштабованості, TensorFlow використовується для вирішення широкого спектру завдань, починаючи від простих моделей регресії до складних нейронних мереж.

Однією з головних переваг TensorFlow є його комплексна екосистема, яка включає в себе різноманітні інструменти для розробки моделей машинного навчання, включаючи бібліотеки для обробки даних, інструменти для візуалізації та відладки моделей, а також підтримку різних платформ для розгортання моделей у виробничих середовищах. TensorFlow підтримує як центральні процесори (CPU), так і графічні процесори (GPU), що дозволяє значно прискорити процес тренування моделей.

TensorFlow має модульну архітектуру, яка дозволяє легко налаштовувати та розширювати платформу відповідно до потреб конкретного проєкту. Основний модуль TensorFlow надає базові функції для створення та тренування моделей, тоді як додаткові модулі, такі як TensorFlow Extended

(TFX) та TensorFlow Lite, забезпечують додаткові можливості для розгортання моделей у виробничих середовищах та на мобільних пристроях відповідно.

TensorFlow широко використовується в різних галузях, включаючи розпізнавання образів, обробку природної мови, рекомендаційні системи та багато інших. Наприклад, у галузі розпізнавання образів TensorFlow використовується для створення моделей, які можуть автоматично розпізнавати та класифікувати зображення. У галузі обробки природної мови TensorFlow використовується для створення моделей, які можуть автоматично перекладати текст, розпізнавати мовлення та аналізувати настрої.

TensorFlow також має активну спільноту розробників, які постійно працюють над вдосконаленням платформи та створенням нових інструментів та бібліотек. Це робить TensorFlow потужним та гнучким інструментом для машинного навчання, який може бути адаптований до потреб будь-якого проєкту. Крім того, Google надає широкий спектр навчальних матеріалів та курсів для вивчення TensorFlow, що значно спрощує процес вивчення та освоєння платформи.

Таким чином, TensorFlow є комплексною платформою для машинного навчання, яка надає всі необхідні інструменти та бібліотеки для створення, тренування та розгортання моделей. Завдяки своїй гнучкості, масштабованості та активній спільноті, TensorFlow є ідеальним вибором для розробників та дослідників, які хочуть використовувати машинне навчання у своїх проєктах.

PostgreSQL є однією з найпопулярніших систем керування базами даних з відкритим вихідним кодом, яка забезпечує високу продуктивність, надійність та масштабованість. Вона була розроблена як об'єктно-реляційна система керування базами даних (СКБД), що поєднує в собі переваги реляційних баз даних та можливості об'єктно-орієнтованого програмування. PostgreSQL підтримує широкий спектр функцій, включаючи транзакції, перевірку цілісності даних, контроль доступу та багато інших.

Однією з головних переваг PostgreSQL є її модель "клієнт-сервер", яка забезпечує розподілену архітектуру для обробки запитів до бази даних. У цій

моделі клієнтська програма відправляє запити до сервера бази даних, який обробляє ці запити та повертає результати клієнту. Це дозволяє забезпечити високу продуктивність та ефективність роботи з базою даних, особливо при обробці великих обсягів даних.

PostgreSQL використовує мову структурованих запитів SQL (Structured Query Language) для взаємодії з базою даних. SQL є стандартною мовою для роботи з реляційними базами даних, яка дозволяє виконувати різноманітні операції, такі як створення, оновлення та видалення даних, а також складні запити для аналізу та обробки даних. PostgreSQL підтримує розширення SQL, що дозволяє додавати нові функції та можливості до мови запитів.

Однією з ключових особливостей PostgreSQL є її підтримка транзакцій, яка забезпечує цілісність та консистентність даних у базі. Транзакції дозволяють групувати декілька операцій в одну логічну одиницю, яка виконується повністю або не виконується взагалі. Це забезпечує захист даних від некоректних змін та помилок під час виконання операцій.

PostgreSQL також підтримує механізми реплікації та відновлення даних, що забезпечує високу надійність та доступність бази даних. Реплікація дозволяє створювати копії бази даних на різних серверах, що забезпечує захист від втрати даних та підвищує продуктивність системи. Механізми відновлення даних дозволяють відновити базу даних у випадку збоїв або пошкоджень, що забезпечує безперервну роботу системи.

Однією з важливих особливостей PostgreSQL є її підтримка об'єктно-орієнтованих можливостей, що дозволяє розробникам створювати складні типи даних та структури. Це дозволяє значно розширити можливості бази даних та забезпечити більш ефективну обробку складних даних. PostgreSQL також підтримує розширення, які дозволяють додавати нові функції та можливості до бази даних, що робить її дуже гнучкою та адаптивною до потреб конкретного проєкту.

Таким чином, PostgreSQL є потужною та надійною системою керування базами даних, яка забезпечує високу продуктивність, надійність та

масштабованість. Вона підтримує широкий спектр функцій та можливостей, що робить її ідеальним вибором для зберігання та обробки великих обсягів даних у сучасних вебдодатках.

HTML (HyperText Markup Language) є стандартною мовою розмітки для створення вебсторінок. Вона використовується для структурування та форматування вмісту вебсторінок, включаючи текст, зображення, відео, посилання та інші елементи. HTML дозволяє розробникам створювати структуровані документи, які можуть бути відображені в веббраузерах та інших клієнтських додатках.

Однією з ключових особливостей HTML є його простота та зручність у використанні. HTML використовує теги для визначення різних елементів документа, таких як заголовки, абзаци, списки, посилання та інші. Кожен тег оточується кутовими дужками та визначає початок та кінець елемента. Це дозволяє легко читати та розуміти структуру документа, що робить HTML доступним для широкого кола користувачів, включаючи розробників, дизайнерів та контент-менеджерів.

HTML підтримує гіпертекстові посилання, які дозволяють зв'язувати різні документи та ресурси між собою. Це є основою веб, оскільки дозволяє користувачам легко переходити між різними сторінками та ресурсами. Гіпертекстові посилання визначаються за допомогою тегу `<a>`, який включає атрибут `href` для вказання URL-адреси ресурсу.

HTML також підтримує мультимедійні елементи, такі як зображення, аудіо та відео. Це дозволяє розробникам включати різноманітний вміст у свої вебсторінки, що робить їх більш інтерактивними та привабливими для користувачів. Зображення додаються за допомогою тегу ``, а аудіо та відео — за допомогою тегів `<audio>` та `<video>` відповідно.

Python надає потужний інструментарій для розробки серверної частини додатка, забезпечуючи простоту та ефективність коду. PostgreSQL забезпечує надійне та ефективне управління даними, підтримуючи складні запити та

високу продуктивність. Flask дозволяє швидко та легко створювати вебдодатки з мінімальними налаштуваннями, забезпечуючи гнучкість та масштабованість.

TensorFlow відкриває можливості для використання машинного навчання у вебдодатках, дозволяючи створювати потужні моделі для аналізу даних та прийняття рішень. HTML забезпечує структуру та форматування вебсторінок, роблячи їх доступними та зручними для користувачів. CSS додає стилі та форматування, покращуючи вигляд вебдодатка, а JavaScript забезпечує динамічну взаємодію та інтерактивність.

Використання цих технологій дозволяє створити вебзастосунок, який буде відповідати високим вимогам до продуктивності, надійності та безпеки. Крім того, вони забезпечують гнучкість та масштабованість, що дозволяє легко адаптувати застосунок до змінних потреб бізнесу та користувачів.

3.2 Розробка фізичної моделі

Для відображення всіх артефактів бази даних, відношень між таблицями, визначень обмежень, та для досягнення мети продуктивності, побудовано фізичну модель БД, яка представлена на рисунку 3.1.

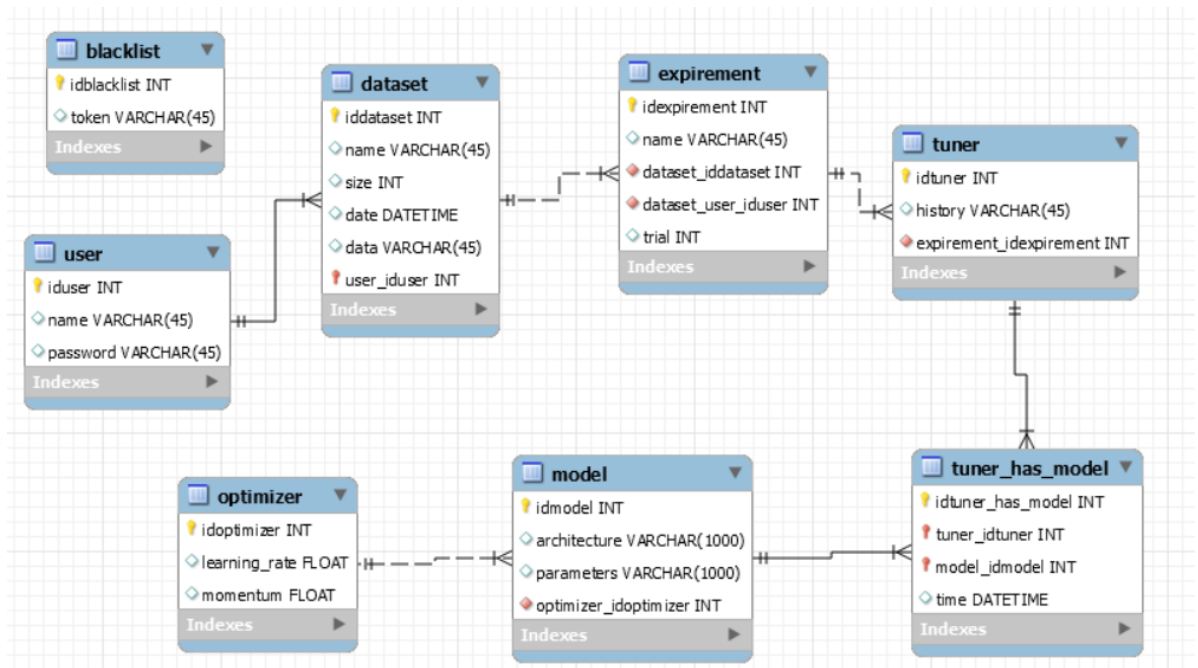


Рисунок 3.1 – Фізична модель БД застосунку

Назви, типи та обмеження полів представлені у таблицях 3.1 – 3.8.

Таблиця 3.1 – Структура таблиці User

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження
iduser	PK	INTEGER	NOT NULL AUTO_INCREMENT
name		VARCHAR(45)	NULL
password		VARCHAR(45)	NULL

Таблиця 3.2 – Структура таблиці Dataset

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження
Iddataset	PK	INTEGER	NOT NULL AUTO_INCREMENT
Name		VARCHAR(45)	NULL
Size		INTEGER	NULL
Date		DATETIME()	NULL
Data		VARCHAR(45)	NULL
user_iduser	FK	INTEGER	NOT_NULL

Таблиця 3.3 – Структура таблиці Expiration

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження
Idexpiration	PK	INTEGER	NOT NULL AUTO_INCREMENT
Name		VARCHAR(45)	NOT_NULL
Trial		INTEGER	NULL
dataset_id_dataset	FK	INTEGER	NOT_NULL
dataset_user_iduser	FK	INTEGER	NOT_NULL

Таблиця 3.4 – Структура таблиці Tuner

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження
Idtuner	PK	INTEGER	NOT NULL AUTO_INCREMENT
History		VARCHAR(45)	NULL
expiration_idexpiration		INTEGER	NOT_NULL

Таблиця 3.5 – Структура таблиці Tuner_has_model

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження
idtuner_has_model	PK	INTEGER	NOT NULL AUTO_INCREMENT
tuner_idtuner	FK	INTEGER	NOT_NULL
model_idmodel	FK	INTEGER	NOT_NULL
Time		DATETIME	NULL

Таблиця 3.6 – Структура таблиці Model

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження
idmodel	PK	INTEGER	NOT NULL AUTO_INCREMENT
architecture		VARCHAR(1000)	NULL
parameters		VARCHAR(1000)	NULL
optimizer_idoptimizer	FK	INTEGER	NOT_NULL

Таблиця 3.7 – Структура таблиці Optimizer

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження
Idoptimizer	PK	INTEGER	NOT NULL AUTO_INCREMENT
learning_rate		FLOAT	NULL
Momentum		FLOAT	NULL

Таблиця 3.8 – Структура таблиці Blacklist

Ідентифікатор поля	Ознака ключа	Тип даних	Обмеження
idblacklist	PK	INTEGER	NOT NULL AUTO_INCREMENT
token		VARCHAR(45)	NOT_NULL

3.3 Проектування інтерфейсу

Web-інтерфейс - це сукупність засобів, за допомогою яких користувач взаємодіє з Web-сайтом або будь-яким іншим додатком через браузер.

Макет сторінок сайту представлені на рисунках 2.5.

Користувач починає своє знайомство з графічним інтерфейсом з головної сторінки сайту. Головна сторінка дозволяє клієнту ознайомитись з продуктом сайту. Шаблон головної сторінки представлений на рисунку 2.5.

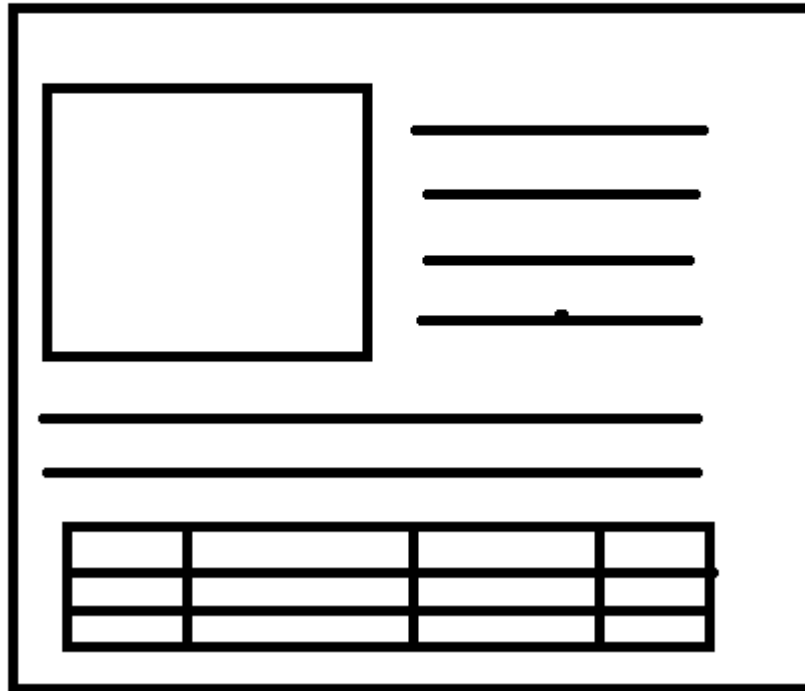


Рисунок 3.2 – Прототип головної сторінки

При натисканні «НД», користувач переходить на відповідну сторінку. Сторінка «НД» дозволяє переглянути, додати або обрати НД, що в нього є. Шаблон сторінки «НД» представлений на рисунку 2.6.

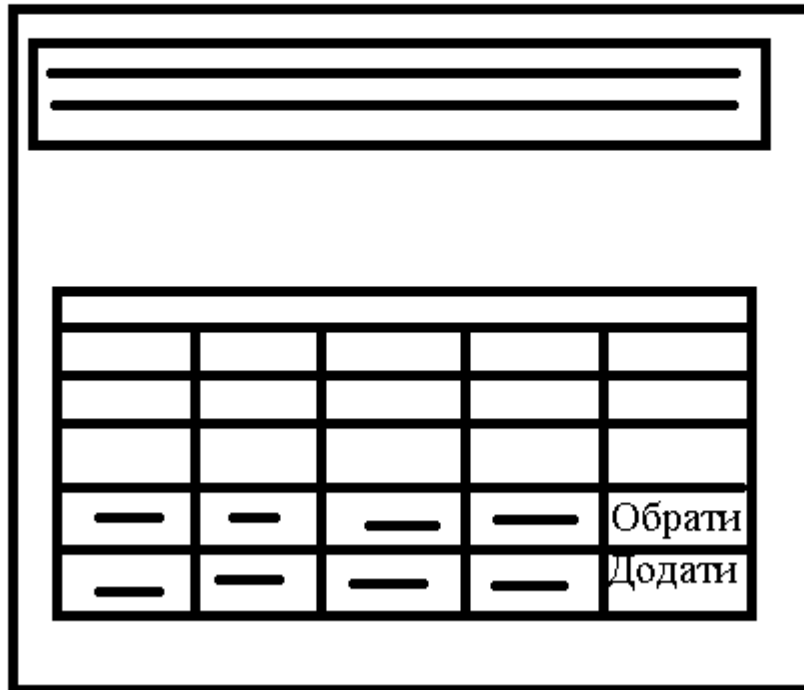
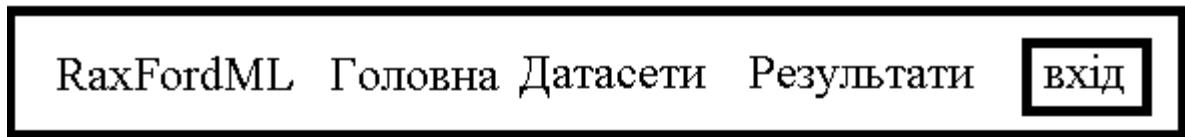


Рисунок 3.3 – Прототип сторінки «НД»

При натисканні кнопки «Додати», користувачу виводиться форма додавання НД. На формі користувач вводить назву НД, тип задачі та файл csv. Шаблон форми «Додавання НД» представлений на рисунку 2.7.

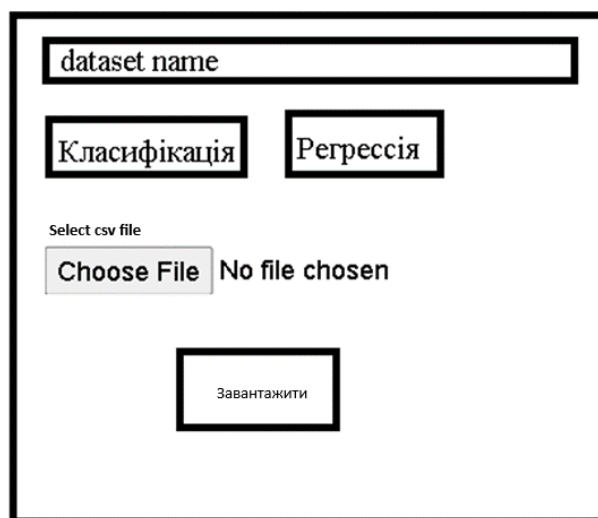


Рисунок 3.4 – Прототип форми «Додавання НД»

При натисканні «Обрати», користувачу виводиться форма з описом НД. На формі користувач має опис НД та обирає розмір тренування моделей та може розпочати тренування моделі. Або видалити НД. Шаблон форми «Вибраний НД» представлений на рисунку 2.8.

Класифікація		Регресія			

shape:
[]

Тренувати Видалити

Рисунок 3.5 – Прототип форми «Обраний НД»

При натисканні на кнопку «Результати», користувачеві виводиться відповідна сторінка. На цій сторінці користувачеві виводяться результати його тренів. Шаблон сторінки «Результати» представлений на рисунку 2.9.

RaxFordML Головна Датасети Результати

				Завантажити

Рисунок 3.6 – Прототип сторінки «Результати»

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи представлено засоби розробки вебзастосунку для автоматизованого машинного навчання, а саме:

- мова програмування Python;
- фреймворк Flask;
- платформа TensorFlow;
- база даних PostgreSQL;
- середовище розробки PyCharm.

Описано структуру бази створеної бази даних, представлена фізична модель.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

Щоб почати роботу сайту, потрібно запустити браузер та перейти на сайт.

Після запуску сайту, користувач потрапляє на головну сторінку. Після цього у вікні браузера з'явиться головна сторінка сайту вебсайту «RaxFord-AutoML». Головна сторінка включає короткий опис та відео для правильного ознайомлення з аналізом своїх даних та підготовкою даних для навчання.

Головна сторінка сайту представлена на рисунку 4.1.

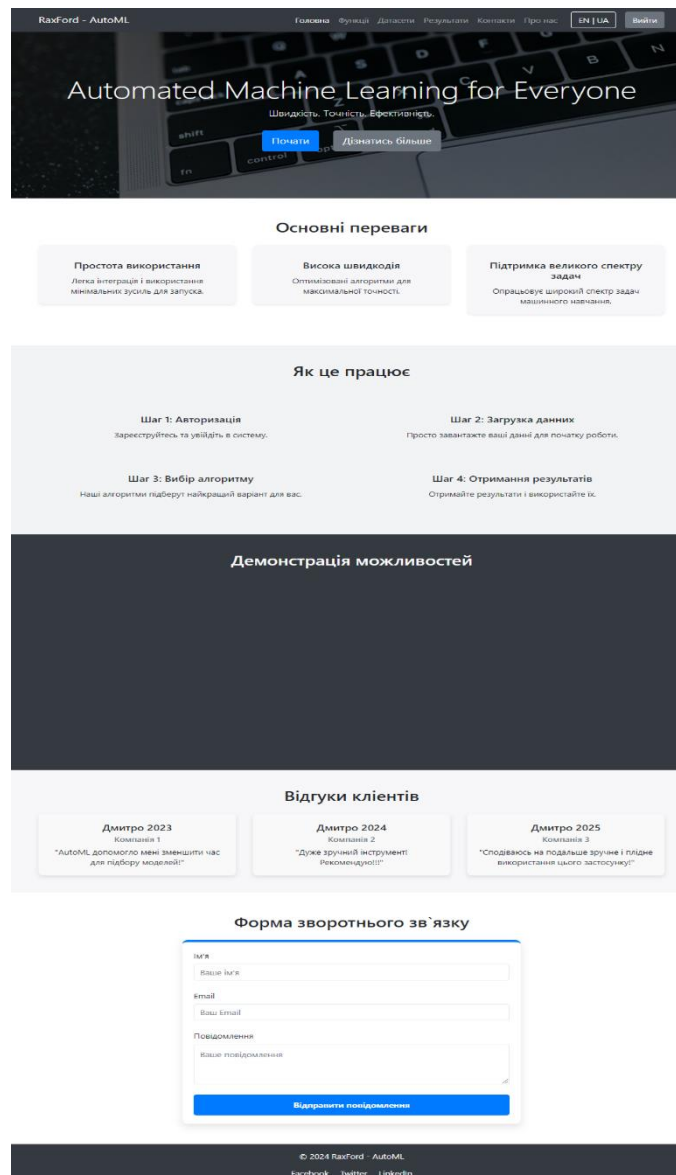


Рисунок 4.1 – Головна сторінка вебсайту «RaxFord-AutoML»

Для того щоб користуватись сайтом, потрібно зареєструватись. Натиснувши на кнопку «Вхід», з'явиться форма для входу. Ви можете ввести свої данні, якщо вже зареєстровані, або зареєструватись, натиснувши кнопку «Реєстрація». Форма реєстрації представлена на рисунку 4.2.

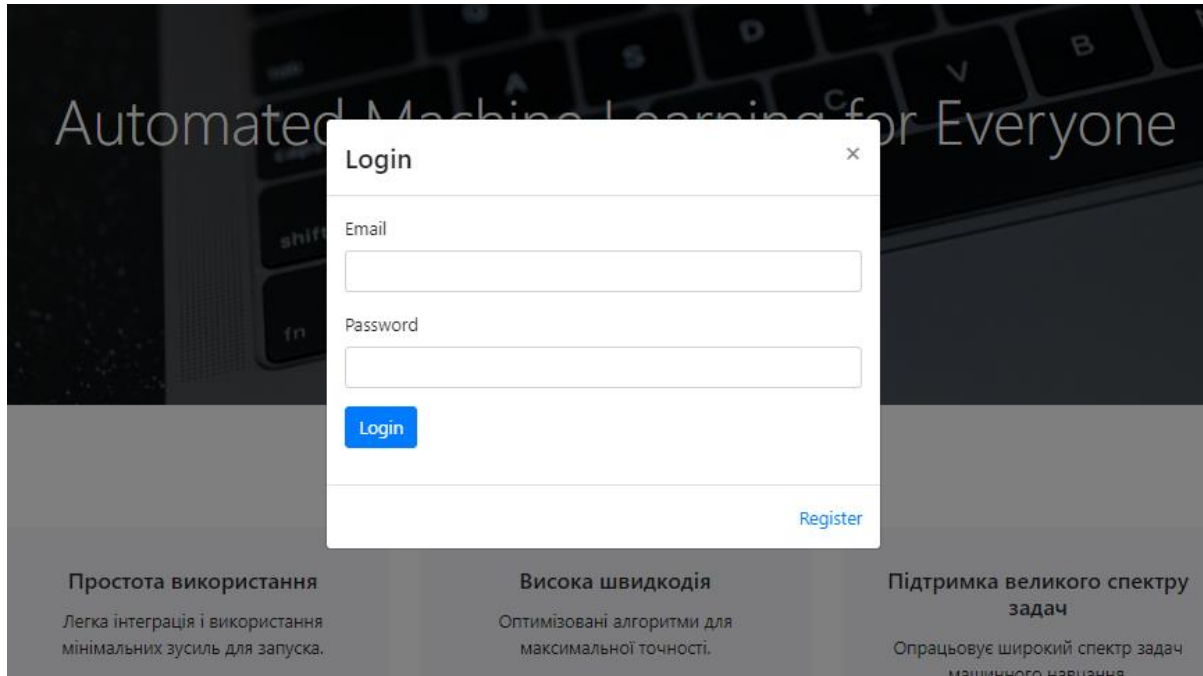


Рисунок 4.2 – Форма реєстрації

Для переходу на сторінку «НД», потрібно натиснути на кнопку «НД» у верхній панелі. Після натискання кнопки «НД», ви переходите на сторінку «НД». На цій сторінці у таблиці показуються ваші НД. Якщо ви не додавали НД, у вас буде пуста таблиця з НД.

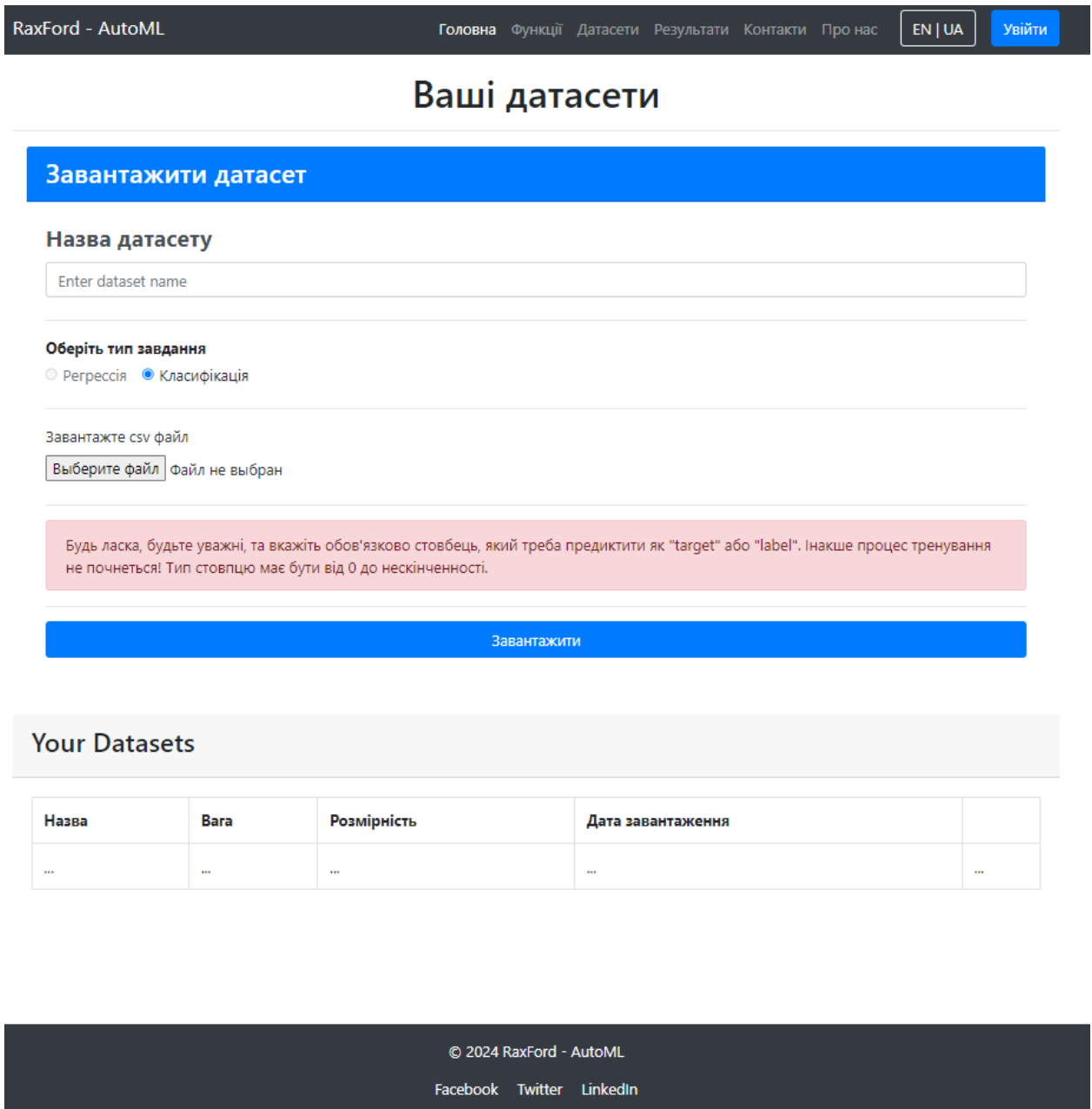


Рисунок 4.3 – Сторінка НД без наявних НД

Щоб додати НД, потрібно натиснути на кнопку «Додати». Після натискання кнопки «Додати», з'являється форма «Додавання НД». На цій формі потрібно ввести ім'я НД, який ви хочете завантажити, обрати файл з розширенням «csv» та обрати тип задачі, під який орієнтований цей НД. Після натискання кнопки «Завантажити», НД додається на вашу сторінку «НД».

Форма «Додавання НД» представлена на рисунку 4.4.

Сторінка «НД» з кількома доданими НД представлена на рисунку 4.5.

Завантажити датасет

Назва датасету

Оберіть тип завдання

Регресія Класифікація

Завантажте csv файл

Файл не выбран

Будь ласка, будьте уважні, та вкажіть обов'язково стовбець, який треба предиктити як "target" або "label". Інакше процес тренування не почнеться! Тип стовпцю має бути від 0 до нескінченності.

Рисунок 4.4 – Форма «Додавання НД»

RaxFord - AutoML Головна [Функції](#) [Датасети](#) [Результати](#) [Контакти](#) [Про нас](#) [EN | UA](#) [Вийти](#)

Ваші датасети

Завантажити датасет

Назва датасету

Оберіть тип завдання

Регресія Класифікація

Завантажте csv файл

Файл не выбран

Будь ласка, будьте уважні, та вкажіть обов'язково стовбець, який треба предиктити як "target" або "label". Інакше процес тренування не почнеться! Тип стовпцю має бути від 0 до нескінченності.

Your Datasets

Назва	Вага	Розмірність	Дата завантаження	
iris4	868КБ	(502,4)	26.01.2021	<input type="button" value="Обрати"/>
iris3	868КБ	(502,4)	26.01.2021	<input type="button" value="Обрати"/>
iris2	868КБ	(502,4)	26.01.2021	<input type="button" value="Обрати"/>
iris	868КБ	(502,4)	26.01.2021	<input type="button" value="Обрати"/>
...

Рисунок 4.5 – Сторінка «НД» з доданими НД

Щоб обрати та переглянути НД, що ви додали, натисніть кнопку «Обрати» напроти потрібного НД. Після натискання кнопки «Обрати» напроти потрібного НД, відобразиться форма «Обраний НД». На цій формі відображена коротка інформація з НД. Ви можете видалити цей НД, натиснувши кнопку «Видалити». Або обрати потрібний розмір моделі та почати трейн, натиснувши кнопку «Почати трейн».

Форма «Обраний НД» представлена на рисунку 4.6.

Dataset Form [Close]

iris3

Выберите тип задачи

Регрессия Классификация

	Unnamed: 0	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	label
0	0	1	5.1	3.5	1.4	0.2	0
1	1	2	4.9	3.0	1.4	0.2	0
2	2	3	4.7	3.2	1.3	0.2	0
3	3	4	4.6	3.1	1.5	0.2	0
4	4	5	5.0	3.6	1.4	0.2	0
5	5	6	5.4	3.9	1.7	0.4	0
6	6	7	4.6	3.4	1.4	0.3	0
7	7	8	5.0	3.4	1.5	0.2	0
8	8	9	4.4	2.9	1.4	0.2	0
9	9	10	4.9	3.1	1.5	0.1	0
10	10	11	5.4	3.7	1.5	0.2	0
11	11	12	4.8	3.4	1.6	0.2	0
12	12	13	4.8	3.0	1.4	0.1	0
13	13	14	4.3	3.0	1.1	0.1	0
14	14	15	5.8	4.0	1.2	0.2	0
15	15	16	5.7	4.4	1.5	0.4	0
16	16	17	5.4	3.9	1.3	0.4	0
17	17	18	5.1	3.5	1.4	0.3	0
18	18	19	5.7	3.8	1.7	0.3	0
19	19	20	5.1	3.8	1.5	0.3	0
20	20	21	5.4	3.4	1.7	0.2	0
21	21	22	5.1	3.7	1.5	0.4	0
22	22	23	4.6	3.6	1.0	0.2	0
23	23	24	5.1	3.3	1.7	0.5	0
24	24	25	4.8	3.4	1.9	0.2	0

Shape:
(150, 7)

Оберіть максимальний розмір моделі

Маленька [Dropdown Arrow]

Тренувати **Видалити**

[Close]

Рисунок 4.6 – Форма «Обраний НД»

Щоб побачити результати навчання, потрібно натиснути «Результати» на верхній панелі сайту. Після натискання кнопки «Результати», відображається сторінка «Результати». На цій сторінці представлені результати ваших навчань по певним метрикам. Переглянувши ці метрики, ви можете обрати найкращу модель та завантажити її собі на комп'ютер, натиснувши на кнопку «Завантажити».

Сторінка «Результати» представлена на рисунку 4.7.

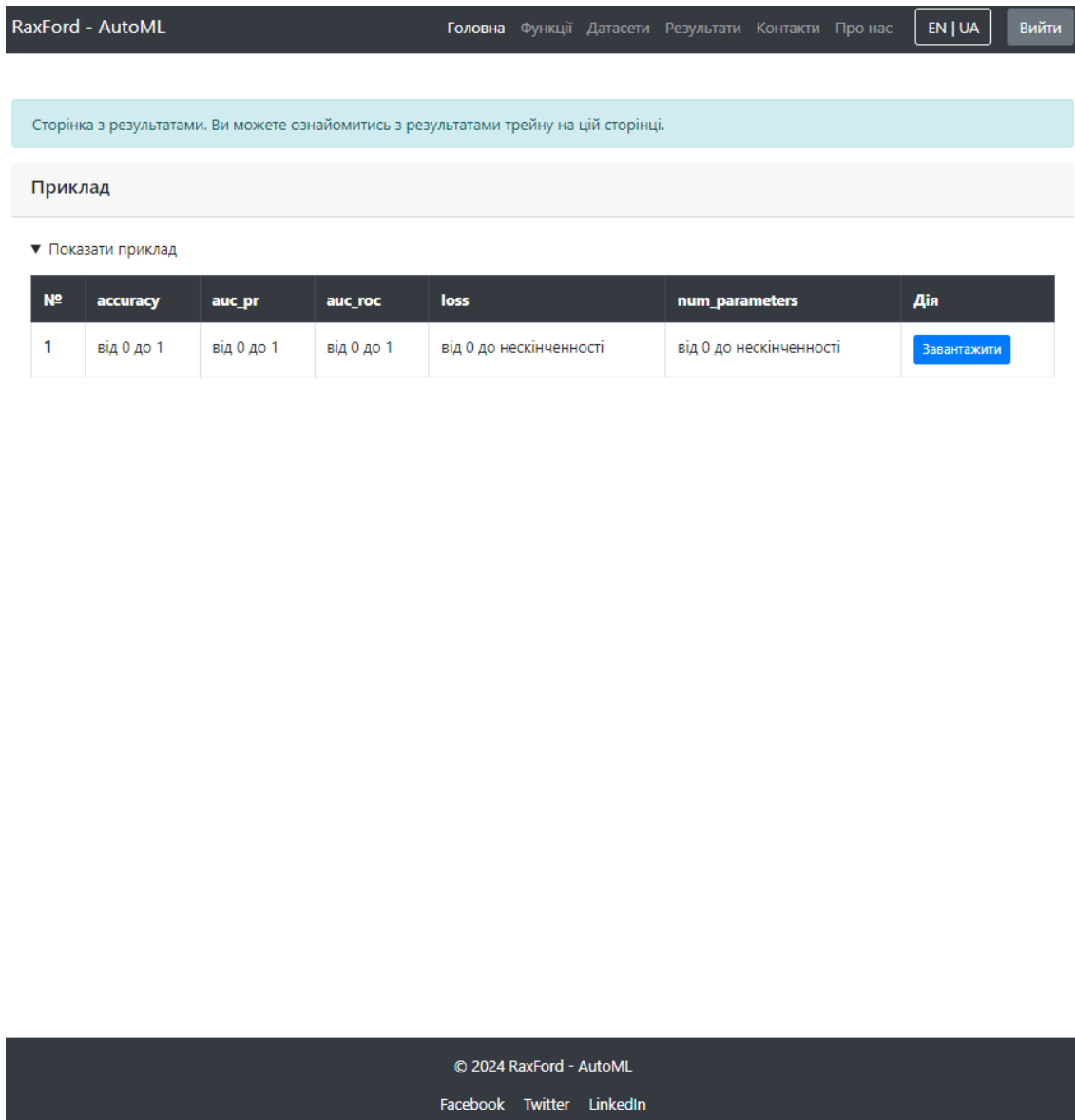


Рисунок 4.7 – Сторінка «Результати»

Створений вебзастосунок охоплює всі встановлені задачі.

Висновки до розділу 4

У четвертому розділі представлено сторінки розробленого вебзастосунку:

- реєстрація;
- логін;
- генерація токену;
- перевірка статусу токену;
- додавання токену до чорного списку;
- додавання НД;
- перегляд НД;
- видалення НД;
- початок навчання;
- навчання;
- перегляд результатів;
- виведення токену з експлуатації;
- скачування моделі.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розроблено 4 розділи.

В першому розділі кваліфікаційної роботи бакалавра розглянуто інноваційну вебплатформу для автоматизованого машинного навчання, яка пропонує зручний та індивідуалізований підхід до отримання штучних нейронних мереж з оптимальною архітектурою для конкретних наборів даних. Платформа використовує передові алгоритми для автоматичного створення архітектур нейронних мереж, що відзначається значним зменшенням кількості ітерацій для визначення оптимальної структури.

Описано функціонал вебсайту, що спрямований на автоматизацію машинного навчання, з підрозділами від реєстрації та авторизації користувачів до роботи з НД та тренуванням моделей. Розглянуті етапи від входу на сайт до вибору оптимальної моделі для подальшого використання.

В аналізі існуючих систем також виявлено ключові аспекти та вдалі рішення, які можна врахувати під час розробки власної платформи. Проаналізовані Google Cloud AutoML та Microsoft Automated Machine Learning як потенційні конкуренти на ринку програмного забезпечення для розвитку штучного інтелекту.

У другому розділі приведено проєкт програмного забезпечення, який складається з:

- діаграми варіантів використання;
- станів;
- послідовності;
- кооперації;
- класів;
- діяльності;
- розгортання.

Також в проєкті програмного забезпечення наведено проєктування інтерфейсу та концептуальну модель.

У третьому розділі кваліфікаційної роботи представлено засоби розробки вебзастосунку для автоматизованого машинного навчання, а саме:

- мова програмування Python;
- фреймворк Flask;
- платформа TensorFlow;
- база даних PostgreSQL;
- середовище розробки PyCharm.

Описано структуру бази створеної бази даних, представлена фізична модель.

У четвертому розділі представлено сторінки розробленого вебзастосунку:

- реєстрація;
- логін;
- генерація токену;
- перевірка статусу токену;
- додавання токену до чорного списку;
- додавання НД;
- перегляд НД;
- видалення НД;
- початок навчання;
- навчання;
- перегляд результатів;
- виведення токену з експлуатації;
- скачування моделі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Introduction to JSON Web Tokens: вебсайт URL: <https://jwt.io/introduction> (дата звернення: 04.04.2024).
2. Google Cloud AutoML: вебсайт URL: <https://cloud.google.com/automl> (дата звернення: 04.04.2024).
3. Microsoft Automated machine learning: вебсайт URL: <https://azure.microsoft.com/en-us/products/machine-learning/automatedml/> (дата звернення: 04.04.2024).
4. PostgreSQL documentation: вебсайт URL: www.postgresql.org/docs/ (дата звернення: 05.04.2024).
5. Flask User's Guide: вебсайт URL: flask.palletsprojects.com/en/3.0.x/ (дата звернення: 05.04.2024).
6. Get started with TensorFlow: вебсайт URL <https://www.tensorflow.org> (дата звернення: 05.04.2024).
7. PyCharm: The Python IDE for data science and web development: вебсайт URL <https://www.jetbrains.com/pycharm> (Last accessed: 05.04.2024)
8. The Python Programming Language : підручник. Ramalho L. Fluent python. "O'Reilly Media, Inc.", 2022.
9. Uml specification: вебсайт URL <https://www.omg.org/spec/UML/> (дата звернення: 05.04.2024).
10. Rational rose: вебсайт URL <https://www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-ifix001> (дата звернення: 23.05.2024).
11. The Python Programming Language: book. Raschka S., Patterson J., Nolet C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information. 2020. Т. 11. №. 4. Р. 193.
12. Tensorflow: підручник. Weber M. et al. Deeplab2: A tensorflow library for deep labeling. arXiv preprint arXiv:2106.09748. 2021.

13. Tensorflow: підручник. Joseph F. J. J., Nonsiri S., Monsakul A. Keras and TensorFlow: A hands-on experience. Advanced deep learning for engineers and scientists: A practical approach. 2021. P. 85-111.
14. Tensorflow: підручник. Janardhanan P. S. Project repositories for machine learning with TensorFlow. Procedia Computer Science. 2020. Т. 171. P. 188-196.
15. Tensorflow: paper. Demosthenous G., Vassiliades V. Continual learning on the edge with tensorflow lite. arXiv preprint arXiv:2105.01946. 2021.
16. Tensorflow: підручник. Dhanday R., Pearson J., Willis C. J. Implementation of the RX algorithm in TensorFlow for high-performance computing. Image and Signal Processing for Remote Sensing XXIX. SPIE, 2023. Т. 12733. P. 290-294.