

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

“Дослідження та програмна реалізація веб-сервісу для аналізу даних із використанням методів штучного інтелекту у хмарному середовищі”

КБГЗ - 2025

Виконав здобувач вищої освіти
II курсу, групи КН-24м
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Карпіков О.С.
« ____ » _____ 2025 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
« ____ » _____ 2025 р.
Рецензент _____

АНОТАЦІЯ

Карпіков О.С. Дослідження та програмна реалізація веб-сервісу для аналізу даних із використанням методів штучного інтелекту у хмарному середовищі. 122 Комп'ютерні науки. Центральнотукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для автоматизації процесів аналізу даних та машинного навчання у хмарному середовищі.

Метою розробки є дослідження та програмна реалізація веб-сервісу для аналізу даних із використанням методів штучного інтелекту у хмарному середовищі.

Об'єктом дослідження є процес інтелектуального аналізу даних та розгортання моделей у хмарі.

Предметом дослідження є методи створення хмарних веб-сервісів для автоматизованого машинного навчання.

Методи дослідження базуються на методах хмарних технологій, методах машинного навчання, методах системного аналізу, методах об'єктно-орієнтованого програмування.

Результат роботи – програмна реалізація веб-сервісу для аналізу даних із використанням методів штучного інтелекту.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Програма може використовуватися на ПЕОМ з ОС Windows/Linux/macOS (через веб-браузер).

Програма була розроблена з використанням мов програмування Python та JavaScript у середовищі Google Cloud Platform.

Ключові слова: комп'ютерні науки, веб-сервіс, штучний інтелект, хмарні технології, машинне навчання.

ABSTRACT

Karpikov O.S. Research and software implementation of a web service for data analysis using artificial intelligence methods in a cloud environment. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for automating data analysis and machine learning processes in a cloud environment.

The purpose of the development is the research and software implementation of a web service for data analysis using artificial intelligence methods in a cloud environment.

The object of the research is the process of intelligent data analysis and model deployment in the cloud. The subject of the research is the methods of creating cloud web services for automated machine learning.

The research methods are based on the methods of cloud technologies, methods of machine learning, methods of system analysis, methods of object-oriented programming.

The result of the work is the software implementation of a web service for data analysis using artificial intelligence methods.

In the process of working on the software model, an analysis of existing hardware and software tools was performed. All components of the developed software are fully described.

A user-friendly user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows/Linux/macOS OS (via web browser).

The program was developed using Python and JavaScript programming languages within the Google Cloud Platform environment.

Keywords: computer science, web service, artificial intelligence, cloud technologies, machine learning.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень по профілю теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування	22
2.3 Розгорнута постановка завдання	26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	29
3.1 Опис функціонування системи	29
3.2 Розробка структурної схеми.....	31
3.3 Розробка функціональної схеми	33
3.4 Розробка діаграми процесів.....	35
4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ	38
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	38
4.2 Захист розробленого програмного забезпечення.....	41
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	46
6 НАУКОВА НОВИЗНА	52
7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	54
7.1 Визначення цільової аудиторії кінцевого готового продукту	54

					ВКРМ-122.25.0037.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація веб-сервісу для аналізу даних із використанням методів штучного інтелекту у хмарному середовищі</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Карпінков О.С.</i>					<i>М</i>	1	75
<i>Перев.</i>	<i>Якименко Н.М.</i>							
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							
ЦНТУ КН-24М								

7.2	Оцінка привабливості шляхом застосування методів експертних оцінок	56
7.3	Вибір методу оцінки вартості ПЗ	55
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	56
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	58
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	58
7.7	Визначення ключових факторів успіху конкретного проєкту	59
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	60
8.1.	Вступ.....	60
8.2.	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	60
8.3.	Розробка заходів з поліпшення стану охорони праці.....	63
8.4.	Техніка безпеки та протипожежна профілактика.....	65
8.5.	Розрахункова частина. Розрахунок штучного освітлення	66
9	ОСНОВНІ ВИСНОВКИ.....	70
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72

КБПЗ - 2025

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

API	– Application Programming Interface, прикладний програмний інтерфейс
AutoML	– Automated Machine Learning, автоматизоване машинне навчання
AuROC	– Area Under the Receiver Operating Characteristic curve, площа під кривою помилок
CSV	– Comma-Separated Values, текстовий формат представлення табличних даних
GCP	– Google Cloud Platform, хмарна платформа компанії Google
IDE	– Integrated Development Environment, інтегроване середовище розробки
JSON	– JavaScript Object Notation, текстовий формат обміну даними
ML	– Machine Learning, машинне навчання
MLOps	– Machine Learning Operations, методологія автоматизації життєвого циклу машинного навчання
REST	– Representational State Transfer, архітектурний стиль взаємодії компонентів
SPA	– Single Page Application, односторінковий веб-додаток
SQL	– Structured Query Language, мова структурованих запитів
UI/UX	– User Interface / User Experience, інтерфейс користувача та досвід взаємодії

ВСТУП

Актуальність теми. Сучасний світ переживає епоху інформаційного вибуху, де обсяги даних, що генеруються бізнесом та наукою, зростають експоненціально. Проте самі по собі масиви даних (Big Data) не мають цінності без ефективного аналізу для отримання практичних висновків та прогнозування тенденцій. Найпотужнішим інструментом для вирішення цієї задачі є технології штучного інтелекту (ШІ) та машинного навчання (ML).

Однак впровадження ML-рішень пов'язане з високим порогом входження, необхідністю дороговартісного обладнання (GPU) та дефіцитом вузькопрофільних фахівців. Хмарні платформи (Google Cloud, AWS, Azure) вирішують проблему ресурсів, пропонуючи AI as a Service (AIaaS), але їхні інтерфейси часто залишаються складними для кінцевого користувача. Тому розробка веб-сервісу, який виступає спрощеним "мостом" між користувачем та потужними хмарними алгоритмами, є актуальною науково-технічною задачею.

Мета і завдання дослідження. Метою роботи є дослідження методів хмарного машинного навчання та програмна реалізація веб-сервісу для автоматизованого аналізу даних з використанням платформи Google Vertex AI. Для досягнення мети вирішено такі завдання:

1. Провести аналіз існуючих хмарних платформ та методів ШІ для аналізу даних.
2. Обґрунтувати вибір технологічного стеку (Python, React, Google Cloud Platform).
3. Спроекувати архітектуру системи та розробити алгоритми взаємодії клієнтської та серверної частин.
4. Виконати програмну реалізацію веб-сервісу та інтегрувати його з сервісами Google Cloud Storage та Vertex AI.
5. Провести експериментальне дослідження роботи системи на реальному наборі даних (прогнозування відтоку клієнтів банку).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Об'єкт дослідження: процес аналізу наборів даних із застосуванням методів машинного навчання у хмарному середовищі.

Предмет дослідження: методи, моделі та програмні засоби для створення веб-сервісу інтелектуального аналізу даних.

Наукова новизна одержаних результатів полягає в тому, що:

1. **Удосконалено** метод розгортання систем машинного навчання для фінансового сектору шляхом інтеграції технології AutoML (Google Vertex AI) у легку веб-архітектуру, що, на відміну від існуючих рішень, дозволяє автоматизувати повний цикл (від завантаження даних до отримання прогнозу) без необхідності залучення інженерів MLOps.

2. **Набуло подальшого розвитку** застосування хмарних сервісів MlaaS (Machine Learning as a Service) для задач малого та середнього бізнесу, зокрема запропоновано архітектурний підхід, який мінімізує витрати на інфраструктуру завдяки використанню безсерверних обчислень та динамічного розподілу навантаження.

Практичне значення одержаних результатів полягає у створенні діючого програмного продукту, який дозволяє користувачам без глибоких знань у Data Science виконувати складні прогнозні задачі (навчання моделей, отримання прогнозів) через зручний графічний інтерфейс, використовуючи обчислювальні потужності хмари.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Створений програмний комплекс це вузькоспрямована мережева служба, чия основна функція полягає в автоматизації всього процесу проведення інтелектуального аналізу даних (Data Mining), роблячи його простішим, завдяки застосуванню технологій хмарного машинного навчання (Cloud Machine Learning).

З огляду на шалений приплив даних, що продукуються нинішніми компаніями, постає наріжна потреба у засобах, здатних трансформувати неперероблені відомості (Raw Data) у значущі узагальнення для прийняття рішень (Insights). Класичні підходи до аналізу нерідко нездатні вихоплювати заплутані кореляції у велетенських нагромадженнях даних, а створення повноцінного штату спеціалістів з Data Science часто є економічно невиправданим для чималої кількості малого та середнього бізнесу через високу ціну апаратури та брак фахівців потрібного рівня.

Цей комплекс розроблено з метою подолання згаданої колізії, пропонуючи єдину точку входу до потужних розробок у царині штучного інтелекту (ШІ), які функціонують на хмарному сервісі Google Cloud Platform (GCP). Він виконує роль проміжного програмного забезпечення, що знімає з плечей фінального споживача клопоти з конфігурацією серверного забезпечення, кодуванням програм для тренування нейронних мереж та загальним адмініструванням процесів впровадження розроблених моделей.

Функціональне призначення системи полягає у вирішенні наступних завдань

1. Процедура завантажування та верифікації даних (Data Ingestion) здійснюється автоматично. Апаратне рішення гарантує стійкий спосіб трансферу

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

інформації від користувача до захищеного репозиторію у Google Cloud Storage. У ході цього відбувається розпізнавання коректності файлів (стандарт CSV), перевірка наявності усіх належних полів та приведення зведеної інформації до вигляду, який може бути оброблений моделями штучного інтелекту.

2. Надання можливості використання інструментів автоматичного машинного навчання (AutoML). Основна функція полягає у взаємодії із сервісом Vertex AI, що дає змогу оператору, навіть не володіючи розширеними пізнаннями у ділянці математичної статистики, розпочати пошук найбільш вдалої конфігурації моделі. Програма незалежно запускає процедури навчання, перевірки та оцінки моделей на віддалених обчислювальних ресурсах.

3. Керування повним життєвим циклом розробок машинного навчання (MLOps). Ця послуга сфокусована на відстеженні стану тренування у режимі реального часу, збереженні кодів ідентифікації навчених зразків та адмініструванні їхнього введення в експлуатацію (Deployment) як доступних точок програмного доступу (Endpoints) для генерації прогностичних результатів.

4. Забезпечення засобів для аналізу бізнесу та передбачення майбутнього. Суть функціонування системи полягає у представленні користувачеві легкодоступних показників ефективності моделі (наприклад, відсоток правильності, зведена таблиця помилок) та інтуїтивно зрозумілого інтерфейсу для здійснення прогнозів на свіжих інформаційних масивах. Це дає змогу застосовувати розробку задля вирішення конкретних комерційних потреб, як-от визначення рівня ризику, передбачення відмови клієнтів (Churn Prediction) чи категоризація сутностей, без потреби у залученні стороннього софту.

Отже, створений веб-ресурс має на меті зробити сучасні методики штучного інтелекту більш загальнодоступними, переміщуючи комплексні обчислювальні операції із локальних пристроїв користувача до гнучкого хмарного простору, гарантуючи водночас оперативне опрацювання даних та надійність отриманих висновків.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Сфера використання створеного комплексу охоплює чимало секторів економіки, наукової діяльності та управлінських процесів, зокрема там, де потрібна робота з впорядкованими наборами даних для розкриття неявних зв'язків та передбачення подій у перспективі. Через застосування хмарних технологій та загального призначення алгоритмів AutoML, цей веб-сервіс має здатність бути налаштованим для розв'язання практичних завдань у численних доменах без потреби втручання у вихідний програмний код.

Ключові напрямки використання цієї розробки включають:

1. Фінансова сфера та банківська справа (ФінТех). Це ключовий пріоритет у застосуванні, що було доведено під час дослідження системи на експериментальному рівні, використовуючи приклад прогнозування відходу клієнтів (Churn Prediction). У цьому напрямку веб-сервіс може бути задіяний для:

- Кредитного скорингу: автоматичне визначення спроможності позичальника брати позику, базуючись на його фінансовому минулому, віці, доходах та інших даних, що містяться у вхідних файлах формату CSV.
- Управління небезпеками: передбачення шансів на невиконання боргу або ідентифікація підозрілих фінансових операцій (детект шахрайства) через аналіз відхилень у даних.
- Групування клієнтів: розподіл існуючих клієнтів на певні категорії для створення індивідуалізованих пропозицій, що сприяє зростанню прихильності клієнтів та зниженню їхнього відтоку.

2. Роздрібний сектор та цифрова торгівля (E-commerce). Для торговельних підприємств ця система слугуватиме засобом для передбачення потреби та вдосконалення реалізації продукції. Сфери використання охоплюють:

- Передбачення обсягів збуту: дослідження минулих транзакцій з метою визначення очікуваного попиту на певні групи товарів, що дає змогу раціоналізувати рівень складських запасів.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- Оцінка дієвості рекламних заходів: зважування впливу різноманітних чинників (дисконтні програми, сезонні коливання, промоактивності) на купівельну поведінку клієнтів.

- Системи персоналізованих пропозицій: застосування накопичених відомостей для прогнозування, який саме продукт, імовірно, приверне увагу конкретного споживача.

3. Охорона здоров'я та сфера медицини. У цьому напрямі веб-сервіс спроможний працювати як інструментарій для асистування у постановці клінічних рішень (CDSS – система підтримки клінічних рішень). Сюди входить:

- Попередній медичний огляд: первинна обробка лабораторних даних та історії хвороби пацієнтів з метою ідентифікації груп осіб із підвищеним ризиком набуття певних недуг (наприклад, цукрового діабету чи кардіоваскулярних проблем) на основі зібраної медичної бази знань.

- Дослідження результативності терапії: встановлення взаємозв'язків між обраними методами лікування та досягнутими ефектами одужання у масивах деперсоналізованих медичних відомостей.

4. Керування Людськими Ресурсами (HR-Аналітика). Ця платформа надає змогу відділам кадрів великих підприємств впроваджувати принципи керованості на основі даних задля роботи з персоналом:

- Передбачення відтоку персоналу: завчасне ідентифікування працівників, у яких висока імовірність розірвання трудових відносин (схоже на проблему відтоку клієнтів у банківській сфері), що дає змогу керівництву застосувати превентивні кроки.

- Визначення потенційної результативності претендентів: аналіз резюме та результатів вхідних випробувань з метою прогнозування майбутніх успіхів кандидата на конкретній посаді.

Цільова група споживачів. Якщо розглядати з позиції кінцевих користувачів, сфера використання системи сфокусована на трьох типах фахівців:

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

1. Бізнес-аналітики: для стрімкої валідації припущень та здобуття прихованих знань без потреби залучення програмістів.

2. Фахівці з аналізу даних (Data Scientists): з метою формування початкових моделей (Baseline models) та швидкого створення робочих прототипів перед розробкою більш складних індивідуальних архітектур.

3. Керівники малих та середніх підприємств: для забезпечення доступу до потужних аналітичних інструментів, які раніше були прерогативою лише великих корпорацій, завдяки моделі оплати відповідно до фактичного використання (Pay-as-you-go) через хмарні платформи.

Отже, цей розроблений веб-сервіс постає як універсальний засіб, придатний для інтеграції у бізнес-потоки будь-якої установи, яка акумулює формалізовані дані і прагне оптимізувати процес прийняття рішень.

КБПЗ_2025

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень з профілю теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Google Cloud AI Platform (Vertex AI)

Google Cloud AI Platform, котра нещодавно була об'єднана під назвою Vertex AI, є однією з найповніших та найпотужніших послуг машинного навчання як сервіс (MLaaS). Вона пропонує уніфікований простір та набір утиліт для опрацювання всього циклу ML-проектів, що цілком підходить для потреб моєї розробки.

Цей сервіс дає змогу не перейматися складнощами керування інфраструктурою, зосереджуючись безпосередньо на розробці моделей та аналітиці даних.

Основні складові, важливі для мого проекту:

1. Vertex AI Workbench

Це контрольоване робоче місце, що базується на Jupyter Notebooks. Воно дає змогу спеціалістам з даних (Data Scientists) оперативно отримувати доступ до хмарних потужностей (включно з GPU/TPU), виконувати дослідження даних (EDA), перевіряти гіпотези та тренувати первинні моделі. Для моєї роботи це ідеальний початковий майданчик.

2. Vertex AI Training (а також AutoML)

Сервіс пропонує два головні підходи до тренування моделей:

Custom Training: Дозволяє розробникам створювати власний код для навчання (з використанням таких фреймворків, як TensorFlow, PyTorch, scikit-learn), пакувати його в Docker-контейнер і розгортати на керованому "залізі" від Google. Це забезпечує максимальну свободу дій.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

AutoML: Це набір рішень, що дозволяє створювати висококласні ML-моделі майже без написання коду. Користувач просто завантажує свій датасет (наприклад, для категоризації тексту або передбачення чисел) та вказує, що саме треба прогнозувати. Vertex AI самостійно обирає структуру моделі та оптимальні налаштування. Ця функція критично важлива для веб-сервісу, оскільки вона дозволить надати можливості ШІ навіть тим, хто не має глибоких знань у машинному навчанні.

3. Vertex AI Endpoints (Розгортання)

- Це ключовий елемент для мого проекту. Після завершення тренування (як власних моделей, так і AutoML), модель можна одразу ж опублікувати як захищену кінцеву точку (API) одним кліком.
- Веб-додаток надсилатиме інформацію (скажімо, новий запис даних для аналізу у форматі JSON) на цю API-адресу та миттєво отримуватиме назад результат (прогноз, згенерований моделлю). Це ідеально втілює ідею "Хмарного ШІ" - сервіс не виконує обчислень сам, а лише спілкується з готовим API.

4. Готові (Попередньо навчені) API

Google Cloud також пропонує набір високопродуктивних моделей, які взагалі не вимагають навчання. Веб-сервіс може безпосередньо користуватися ними:

- Natural Language API: Для оцінки настроїв тексту, виявлення ключових сутностей, категоризації контенту.
- Vision AI: Для ідентифікації об'єктів на зображеннях.
- Speech-to-Text API: Для перетворення мовлення на текст.

Схема взаємодії з веб-сервісом: З огляду на тематику, архітектура складатиметься так:

- Клієнт (Фронтенд): Користувач завантажує дані через інтерфейс нашого веб-додатку.
- Веб-сервіс (Бекенд): Написаний на Python (використовуючи Flask/Django), він отримує ці дані та здійснює їх перевірку.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

- Зв'язок із Google Cloud: Бекенд, застосовуючи клієнтські бібліотеки Google (Google Cloud Client Libraries for Python), посилає підготовлені дані на Vertex AI Endpoint.

- Vertex AI: Модель, що працює під керуванням платформи, виконує необхідний аналіз (формує прогноз).

Попри його потужність, Vertex AI має свої мінуси. Для тих, хто не має досвіду роботи з хмарними сервісами та машинним навчанням, інтерфейс та велика кількість налаштувань можуть видатися заплутаними. Модель тарифікації "плата за фактичне використання" може спричинити непередбачувані фінансові витрати при обробці великих масивів даних або тривалих етапах навчання. Більше того, тісна інтеграція в екосистему Google створює ризик "залежності від постачальника хмарних послуг" (vendor lock-in), що ускладнить перехід на інші платформи у майбутньому.

Підсумок: Google Vertex AI являє собою зрілу, всеохоплюючу платформу, яка надає всі необхідні інструменти "зразу" - від середовища для розробки до масштабованої кінцевої точки API. Його сильні сторони - висока ефективність AutoML та легкість інтеграції з рішеннями Google. Водночас, слабкими місцями залишаються можлива заплутаність для новачків, невизначеність витрат за високих навантажень та ризик "прив'язки" до одного хмарного провайдера.

Microsoft Azure Machine Learning (Azure ML)

Рішення від Microsoft Azure для машинного навчання, відоме як Azure Machine Learning, є прямим конкурентом сервісу від Google і надає аналогічне всеохоплююче хмарне середовище, розроблене для повного циклу життєдіяльності MLOps. Ця платформа забезпечує гнучкий набір інструментів, що орієнтовані як на досвідчених фахівців (Data Scientists), так і на аналітиків з меншим досвідом у кодуванні.

Ядром цієї системи є Azure ML Workspace – єдиний простір, де концентруються усі елементи проєкту: збірки даних, навчені моделі, програмні скрипти та історія всіх проведених експериментів.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Для етапів вивчення даних та розробки моделей Azure ML пропонує вбудоване середовище Notebooks (що працює на базі Jupyter), яке дає можливість інтерактивно створювати й виконувати Python-код, використовуючи потужності підключених обчислювальних ресурсів (Compute Instances).

Головною перевагою Azure ML є різноманітність підходів до навчання моделей. Фахівці мають змогу створювати власні скрипти для тренування (Custom Training), застосовуючи будь-які бажані бібліотеки (будь то Scikit-learn, PyTorch чи TensorFlow), та запускати їх на кластерах із можливістю масштабування (Compute Clusters).

Водночас, для впровадження у вигляді веб-сервісу, особливу цінність мають два інструменти, що значно спрощують процес розробки. Першим є Automated ML (AutoML). Аналогічно до пропозиції Google, цей механізм дозволяє завантажити датасет, а система автоматично підбере найбільш оптимальну модель для завдань класифікації, регресійного аналізу чи передбачення часових рядів. Другий інструмент – це Azure ML Designer, що являє собою візуальний редактор, побудований за принципом drag-and-drop; тут ML-пайплайн можна зібрати з готових модулів (таких як імпорт даних, їхня попередня обробка, вибір алгоритму) без необхідності писати код.

Критичним етапом для програмної реалізації сервісу є процес впровадження моделі (Deployment). Після того, як модель було успішно навчено (будь-яким із згаданих методів), Azure ML надає простий механізм для її "пакування" та розміщення як спеціальної кінцевої точки (Endpoint). Це функціональна, масштабована та захищена кінцева точка (API), яка здатна приймати запити через протокол HTTP.

Отже, структура інтеграції веб-застосунку з сервісами Azure матиме значну схожість із підходом до Google. Користувач завантажує необхідну інформацію через інтерфейс веб-сторінки. Накопичувач на Python (застосовуючи Azure ML SDK для Python) трансліює цю інформацію, упаковану у формат JSON, до розгорнутої Кінцевої Точки (Endpoint) в Azure. Хмарна модель здійснює

необхідний аналіз та повертає результат, який згодом обробляється та відображається користувачеві цим сервісом.

Додатково, компанія Microsoft надає розлогий арсенал попередньо налаштованих моделей у межах Azure Cognitive Services (таких як обробка тексту, машинний зір, розпізнавання мовлення), якими веб-сервіс також може послуговуватися як джерелом ШІ-можливостей, викликаючи відповідні API.

Проте, Azure ML не позбавлена й певних недоліків. Подібно до інших масштабних хмарних екосистем, платформа може виявитися доволі заплутаною для новачків через надлишок доступних сервісів та опцій конфігурації. Фінансові витрати на експлуатацію, особливо у частині обчислювальних потужностей для тренування складних алгоритмів, здатні стати відчутною перепорою для невеликих ініціатив чи індивідуальних розробників. Крім того, існує феномен "прив'язки до постачальника" (vendor lock-in), оскільки застосування унікальних інструментаріїв Azure (наприклад, Designer чи наборів Cognitive Services) суттєво ускладнює подальший перехід на інші хмарні рішення.

Підсумок: Azure ML являє собою гнучку платформу, міцно зчеплену з продуктивним портфоліо Microsoft. Її вагомі переваги полягають у наявності візуального дизайнерського інструментарію та потужного AutoML. Проте, як і у випадку з конкурентами, вона може бути складною для швидкого освоєння, а фактори вартості обчислень та залежності від постачальника хмарних послуг є помітними обмежувальними чинниками.

Amazon SageMaker (AWS)

Amazon Web Services (AWS) вважається першопрохідцем у сфері хмарних платформ, а їхнім ключовим продуктом для машинного навчання є повністю керуєний сервіс Amazon SageMaker. На відміну від цілісних рішень, які пропонують Google чи Azure, SageMaker традиційно надає більш гнучку, компонентно-орієнтовану стратегію. Він дає розробникам набір окремих, спеціалізованих інструментів ("цеглин") для кожного етапу життєвого циклу MLOps, які можна вільно поєднувати.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

нового клієнта, бекенд надішле JSON-запит, що містить дані клієнта, на API SageMaker та отримує відповідь із прогнозом (скажімо, "ймовірність відтоку: 85%").

Також, як і його конкуренти, AWS має потужну добірку готових сервісів ШІ (наприклад, Amazon Comprehend для роботи з текстом, Amazon Rekognition для обробки візуальних даних), якими веб-сервіс може скористатися безпосередньо, якщо поставлене завдання має стандартне рішення.

Попри свою гнучкість, модульний підхід SageMaker може обернутися й недоліком, оскільки він вимагає від користувача глибшого розуміння взаємодії між окремими елементами, що підносить поріг початкового освоєння у порівнянні з більш уніфікованими альтернативами. Модель ціноутворення AWS, хоч і гнучка, може бути непрозорою для фінансового планування, особливо коли задіяна велика кількість різноманітних сервісів SageMaker (навчання, хостинг, моніторинг). Ризик "прив'язки до постачальника хмарних послуг" (vendor lock-in) також існує, хоча модульність частково його нівелює.

Підсумок: SageMaker - це вже усталена та надійна платформа, інтегрована у велику екосистему AWS. Її переваги полягають у компонентній структурі та видатних можливостях для розгортання API. Водночас, залежність від цієї модульності може ускладнити первинне впровадження, а структура витрат інколи буває важко прогнозованою.

Open-Source Альтернатив (Hugging Face)

На противагу тим складним, проте комерційно "зачиненим" екосистемам, що пропонуються "великою трійкою" (Google, Azure, AWS), існує вагома альтернатива - рух відкритого програмного забезпечення (open-source). Замість того, аби платити за готову послугу ШІ (MLaaS), є можливість узяти широкодоступну, безкоштовну модель і розгорнути її самостійно, використовуючи орендовану хмарну інфраструктуру (IaaS, скажімо, звичайний віртуальний сервер).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Сьогодні головним осередком у цій сфері виступає Hugging Face. Це меншою мірою платформа, а більше - колосальне центральне сховище, "GitHub світу машинного навчання". Воно надає розробницькій спільноті дві фундаментальні складові.

Перша - це сам Hugging Face Hub. Це масивне сховище, де налічуються десятки тисяч вже натренованих моделей, здатних вирішувати будь-які завдання: від визначення емоційного забарвлення тексту, перекладу, відповідей на запитання, до категоризації зображень та ідентифікації об'єктів. Ці моделі, підготовлені такими гігантами, як Google, Meta, Microsoft, а також академічними установами, стають загальнодоступними.

Друга складова - це бібліотека під назвою transformers. Це надзвичайно затребуваний інструмент на Python, який дає змогу лише кількома рядками коду завантажити будь-яку з тисяч моделей, доступних у Hub, і миттєво розпочати її експлуатацію для генерації прогнозів.

Для мого проекту це докорінно перевертає підхід до архітектури. Замість того, щоб бекенд ініціював API-виклик до AWS SageMaker, тепер сам бекенд (який функціонує на орендованих потужностях) візьме на себе обов'язок: він імпортує бібліотеку `transformers`, завантажує необхідну модель (скажімо, `bert-base-multilingual-cased` для обробки тексту) безпосередньо у свою оперативну пам'ять і виконує аналіз локально, у межах того ж самого серверного середовища.

Переваги цього підходу очевидні:

1. Безкоштовність. Самі моделі та бібліотеки є безкоштовними.
2. Контроль. Повний контроль над моделлю та даними. Дані не залишають сервер і не йдуть до стороннього сервісу Google чи Microsoft.
3. Гнучкість. Ми не прив'язані до одного хмарного провайдера. Ми можемо запуснути цей код на будь-якому сервері (AWS EC2, Google Compute Engine, Azure VM, DigitalOcean).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Недоліки також суттєві:

1. Це не готова послуга. Ми самі відповідаємо за інфраструктуру. Нам потрібно орендувати сервер, налаштувати середовище Python, встановити бібліотеки.

2. Якщо ми оберемо велику, потужну модель (наприклад, для аналізу зображень чи великих текстів), звичайного сервера може не вистачити. Нам доведеться орендувати дорогий сервер з GPU (графічним процесором).

3. Якщо сервісом почнуть користуватися тисячі людей одночасно, нам доведеться вручну налаштовувати балансування навантаження та створювати копії сервера. Платформи "великої трійки" роблять це автоматично.

Ключовим мінусом у використанні програмного забезпечення з відкритим кодом є той факт, що відсутні надійні готові інфраструктурні рішення та належна технічна підтримка. Відповідальність за підбір, конфігурацію та подальше обслуговування серверного обладнання (включно із затратним устаткуванням, як-от GPU, якщо це необхідно), а також за забезпечення масштабованості та захисту, повністю лежить на кінцевому користувачеві. Це, своєю чергою, висуває високі вимоги до знань у сфері DevOps чи MLOps, що унеможлиблює застосування такого підходу для широкого кола споживачів (наприклад, аналітиків чи фахівців із маркетингу), які не володіють глибокими IT-навичками.

Підсумовуючи: Хоча відкритий код є відмінним варіантом для швидкого створення прототипів або проєктів, де критично важливим є повний суверенітет над інформацією або прагнення не залежати від комерційних постачальників послуг, він залишається гнучкою та безоплатною (з точки зору ліцензій на ПЗ) опцією. Проте, необхідність залучення значних технічних ресурсів для розгортання та забезпечення сталої роботи інфраструктури робить його недоцільним для тих, хто не має спеціалізації в інформаційних технологіях.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Databricks

Рішення компанії Databricks вирізняється на тлі інших пропозицій у сфері аналітики. Якщо "велика хмара" (провідні гравці інфраструктурних послуг) фокусується на загальній інфраструктурі (IaaS) чи платформах (PaaS), то Databricks є високоспеціалізованим інтегрованим рішенням, розробленим першочергово для роботи з масивами даних (Big Data) та застосування методів машинного навчання.

Сам сервіс заснований розробниками Apache Spark - провідного інструменту з відкритим кодом для розподіленої обробки значних обсягів інформації. По суті, Databricks - це комерційна, значно оптимізована та спрощена до користувача версія Spark, що функціонує поверх хмарної основи "великої трійки" (AWS, Azure чи GCP).

Центральне поняття Databricks - це архітектура "Data Lakehouse". Цей підхід об'єднує низьку ціну за зберігання інформації (характерну для "Озер даних") із високою надійністю транзакцій та продуктивністю (властивою "Сховищам даних"). Для будь-якого проєкту це означає наявність єдиного екосистемного осередку, де дані можуть бути не тільки збережені, але й піддаватися обробці, всебічному аналізу та застосовуватися для тренування систем штучного інтелекту.

У складі цього веб-сервісу користувачам доступні кілька важливих функціональних блоків:

1. SQL та Інженерія Даних Databricks: Ці інструменти призначені для попередньої підготовки масивів даних (процеси ETL). Вони дають змогу сервісу (або нам на етапі дослідження) імпортувати необроблені дані, очищати їх, трансформувати та готувати до подальшого аналізу, використовуючи весь потенціал Spark.

2. Машинне Навчання Databricks: Це інтегроване середовище, що охоплює увесь цикл існування моделі ШІ. Воно містить керований сервіс MLflow (широко використовуваний інструмент з відкритим кодом для трасування

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

експериментів) та функціонал AutoML, який автоматично виконує тренування, точне налаштування параметрів та вибір найбільш ефективних моделей.

3. Розгортання Моделей (Model Serving): Аналогічно до конкурентних рішень, Databricks дозволяє виводити будь-яку навчену модель (включно з тими, що створені через AutoML) у вигляді високопродуктивної, масштабованої кінцевої точки (API) з мінімальною затримкою.

Схема взаємодії з нашим веб-сервісом буде подібною: функціонал на базі Python (back-end) буде взаємодіяти з REST API платформи Databricks, надсилаючи запити у форматі JSON до розгорнутої моделі для отримання результатів прогнозування.

Важливою перевагою для тих, хто навчається або розробляє, є версія Databricks Community Edition. Це цілком безкоштовний варіант платформи, хоча він має певні обмеження щодо виділених ресурсів, але ідеально підходить для ознайомлення, наукових досліджень та реалізації дипломних робіт, оскільки надає доступ до робочих середовищ (ноутбуків) та кластерів Spark.

Незважаючи на те, що Databricks Community Edition не потребує оплати, вона суттєво обмежена щодо обчислювальної потужності та часу безперервної роботи кластерів, що може виявитись недостатнім для вирішення дійсно складних завдань. Повністю функціональна версія Databricks є доволі затратним рішенням, орієнтованим переважно на великі комерційні установи. Крім того, платформа має власну специфічну екосистему, що вимагає окремого вивчення та несе певний ризик "прив'язки до постачальника хмарних послуг".

Підсумок: Databricks утримує лідерські позиції у сфері просунутої обробки даних, пропонуючи глибоку інтеграцію Spark та MLflow. Це відмінний вибір для проєктів аналітики, що потребують складних маніпуляцій із даними. Однак безоплатна версія накладає значні обмеження, а комерційна платформа обтяжлива у фінансовому плані та вимагає додаткового навчання, створюючи залежність від даного вендора.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Здійснений огляд демонструє: попри наявність на ринку високоякісних програмних рішень та засобів для проведення AI-досліджень, кожне з них має певні вади, що стримують їхнє широке впровадження та зручне використання, особливо серед осіб, які не володіють ґрунтовними технічними навичками. Ключові труднощі полягають у зависокій планці для старту, заплутаному дизайні користувацького середовища, непередбачуваних фінансових витратах та можливості потрапити в залежність від одного постачальника хмарних послуг.

З огляду на це, створення власного веб-застосунку стає логічним та своєчасним кроком. Цей сервіс слугуватиме єдиною точкою входу, спрощеною оболонкою для доступу до потужностей хмарних систем штучного інтелекту (зокрема, я обрав Google Vertex AI). Його мета - інкапсулювати усю технічну заплутаність базових інструментів, пропонуючи користувачам лише критично важливі функції: завантаження інформаційних масивів, вибір аналітичної моделі та отримання результатів у прозорому форматі. Це дасть змогу зробити передові методи аналізу доступними для невеликих підприємств, фахівців з аналізу та науковців, звільняючи їх від необхідності опановувати складнощі хмарних платформ.

2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування

Базуючись на висновках, отриманих із попереднього етапу дослідження вже існуючих пропозицій, для втілення поставленої мети необхідно визначити основний набір технологій розробки. Цей вибір мусить ґрунтуватися на оцінці таких аспектів: легкість початку роботи (включно з наявністю безкоштовних ліцензій), відповідність сучасним промисловим стандартам, а також потенціал до змін та розширення функціоналу в майбутньому.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Обґрунтування вибору хмарної стратегії та AI-платформи

Оцінивши чотири основні шляхи (комплексні сервіси MLaaS від "великої трійки", спеціалізовані платформи на кшталт Databricks, а також рішення з відкритим кодом), для поточного проекту було обрано стратегію використання платформи «Машинне навчання як послуга» (MLaaS).

Дане рішення зумовлене тим, що воно усуває необхідність турбуватися про керування інфраструктурою (налаштування апаратного забезпечення, графічних процесорів, операційних систем) і дозволяє зосередити зусилля виключно на розробці логіки веб-додатку та тренуванні моделей. Хоча рішення на базі відкритого коду (наприклад, з використанням Hugging Face) є дуже функціональним, воно потребує значних навичок у галузі DevOps для впровадження та масштабування, що є зайвим для завдань цієї роботи.

Серед доступних MLaaS провайдерів (Google, AWS, Azure) основу для розгортання AI-компонентів визначено як Google Cloud Platform, зокрема їхній сервіс Vertex AI.

Обґрунтування цього вибору базується на таких ключових перевагах:

1. Стартовий капітал та доступність (Free Tier та Кредити). Google Cloud надає надзвичайно сприятливі умови для початківців, включаючи, як правило, \$300 безкоштовних бонусних коштів на рік, доповнених постійним безкоштовним рівнем (Free Tier) для багатьох основних сервісів. Для магістерської кваліфікаційної роботи, яка не передбачає комерційної експлуатації, цих ресурсів цілком достатньо для всього циклу - від розробки й тестування до фінальної демонстрації, уникнувши будь-яких грошових витрат.

2. Просунутий AutoML. Сервіс Vertex AI AutoML є одним із лідерів на ринку автоматизованого машинного навчання. Це дасть змогу реалізувати в рамках веб-сервісу функцію, де користувач може завантажити власний датасет, а система автоматично навчить, оптимізує та виставить для нього високопродуктивну модель (для завдань класифікації, регресії тощо).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

3. Спрощене виведення моделей у продакшн (Endpoints). Vertex AI дозволяє лише кількома кроками розгорнути будь-яку навчену модель (як кастомну, так і AutoML) як захищену, горизонтально масштабовану кінцеву точку (API). Це ідеально вписується в архітектуру веб-сервісу, якому потрібно буде лише надсилати HTTP-запити до цього API для отримання готових прогнозних даних.

4. Високоякісні готові API (Pre-trained). Google пропонує найкращі у своєму класі набори попередньо навчених інтерфейсів (Natural Language API, Vision AI, Speech-to-Text), які легко інтегруються в сервіс. Це дозволить надати користувачам функції аналізу тексту чи зображень "без додаткових зусиль", без потреби у тренуванні власних рішень.

Отже, вибір Google Cloud (Vertex AI) забезпечує реалізацію всіх поставлених завдань на сучасному промисловому технологічному стеку, при цьому не потребуючи фінансових вкладень та суттєво спрощуючи інфраструктурні конфігурації.

Обґрунтування вибору мови програмування та Back-end фреймворку

Для створення серверної частини (back-end) веб-сервісу, яка функціонально виступатиме "посередником" між кінцевим користувачем та хмарною платформою штучного інтелекту, було ухвалено рішення використовувати мову програмування Python.

Такий вибір сьогодні вважається галузевим стандартом для завдань, тісно пов'язаних із науковими обчисленнями (Data Science) та ШІ. Це рішення підкріплене низкою вагомих факторів:

1. Багатство екосистеми Data Science: Python володіє найбільш розширеним та зрілим набором бібліотек для роботи з даними. До них належать Pandas (для маніпуляцій у табличному форматі), NumPy (для векторно-матричних обчислень) та SciPy (для реалізації наукових методів). Оскільки наш back-end неминуче матиме справу з попередньою підготовкою та перевіркою вхідних

даних перед їхнім надсиланням до Google Cloud, ці інструменти є неперевершеними.

2. Пріоритетна підтримка від хмарних сервісів: Усі провідні хмарні екосистеми, включно з Google Cloud, надають першочергову підтримку саме для Python. Офіційний набір Google Cloud Client Libraries for Python є досить функціональним, добре оформлений документацією та дозволяє безпосередньо з коду реалізувати підключення до всіх сервісів Vertex AI (включаючи моделювання, деплоймент та виклики кінцевих точок API).

3. Простота опанування та швидкість розгортання: Python славиться своїм чистим кодом, що легко сприймається, та лаконічністю. Це суттєво прискорює фазу розробки та внесення коректив, що є критичним моментом магістерської кваліфікаційної роботи.

Серед доступних веб-фреймворків для Python, наша увага зупинилася на мікро-фреймворку Flask.

Незважаючи на існування більш масштабних рішень, як-от Django, саме Flask виявився оптимальним для поставлених завдань. Django позиціонується як комплексне рішення, що диктує власну архітектуру (включаючи вбудоване ORM, адміністративну панель тощо). Flask, натомість, відзначається мінімалізмом та високою гнучкістю. Він надає лише фундаментальні можливості для визначення маршрутів (routing) та обробки запитів HTTP. Це дає змогу створити максимально легкий, високопродуктивний та адаптований під конкретні потреби API-сервер, чия єдина мета - приймати дані з фронтенду, трансформувати їх та перенаправляти до Google Vertex AI. Така мінімалістична філософія ідеально відповідає потребам створення вузькоспеціалізованого веб-сервісу.

Обґрунтування вибору технологій Front-end та Бази Даних

Аби веб-сервіс запрацював у повному обсязі, окрім серверної логіки (back-end), необхідно задіяти ще два основні елементи: інтерфейс для користувача (front-end) та систему керування базами даних (СУБД).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

У якості інструментарію для розробки клієнтської частини було обрано бібліотеку React.js. Цей фреймворк є одним із найпоширеніших та найактуальніших рішень сучасної веб-розробки. React, який підтримується гігантом Meta, дає змогу будувати складні, динамічні й "живі" інтерфейси, спираючись на компонентну архітектуру. Для нашого проекту це життєво необхідно, адже сервіс має функціонувати як повноцінна програма (Single Page Application, SPA), а не просто як статичний сайт. Нам належить втілити в життя низку непростих функцій: механізми завантаження файлів, відображення поточного статусу обробки запитів, а також динамічне формування графіків та візуалізацій на основі отриманих аналітичних даних. Компонентна парадигма React разом із його ефективним механізмом оновлення віртуального DOM ідеально підходить для таких вимог, гарантуючи швидку та чуйну реакцію користувацького середовища.

Щодо СУБД, наш вибір зупинився на PostgreSQL. Хоча для створення найпростішого прототипу можна було б використати менш вимогливі бази даних, наприклад, SQLite, рішення на користь PostgreSQL одразу закладає міцний ґрунт для майбутньої масштабованості та загальної надійності системи. Це потужна, об'єктно-реляційна система з відкритим кодом, всесвітньо відома своєю стійкістю та багатofункціональністю. У рамках мого сервісу, вона відповідатиме за архівацію всіх ключових відомостей: облікових даних користувачів, інформації, пов'язаної з їхніми проектами, службових даних завантажених наборів даних, і, що найважливіше, - за зберігання результатів аналізу, які надходять від Google Vertex AI. PostgreSQL відмінно взаємодіє з мовою Python (за допомогою таких бібліотек як psycopg2) і традиційно рекомендується як база даних для роботи у зв'язці з веб-фреймворками Flask та Django.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

2.3 Розгорнута постановка завдання

Базуючись на детальному вивченні сфери застосування та наявних технологічних напрацювань, а також аргументації щодо вибору апаратно-програмного комплексу, ми викладаємо чітке й детальне технічне завдання для магістерської кваліфікаційної роботи. Основний вектор роботи спрямовано на проектування та подальшу програмну розробку інтернет-сервісу, який би дав змогу автоматизувати повний цикл тренування моделей машинного інтелекту, використовуючи табличні дані, надані користувачем, та спираючись на хмарні потужності Google Cloud Platform.

Для досягнення поставленої вершини, розроблювана система мусить задовольнити низку обов'язкових умов щодо функціональності, які стосуються як роботи з даними, так і контролю за навчальними процедурами. Зокрема, критично важливим є впровадження механізму, що дозволяє завантажувати файли, збережені в CSV-форматі на персональному комп'ютері користувача, з подальшою автоматичною міграцією цих даних у захищене хмарне сховище Google Cloud Storage. Одночасно з цим процесом, система повинна зареєструвати ці набори даних у службі Vertex AI, що неминуче включає перевірку коректності їхньої структури. Доволі значущою ланкою є реалізація можливості запустити автоматизований процес навчання (AutoML), де користувачеві надається вибір змінної, яку необхідно передбачати; при цьому система має забезпечити моніторинг і відображення статусу цього запуску у режимі реального часу, повідомляючи про кожну стадію: підготовку, власне навчання чи появу будь-яких збоїв.

Окремо виділяється блок вимог, що стосуються забезпечення комунікації користувача з навченим алгоритмом. На системі має бути реалізована функція отримання та графічного представлення показників якості моделі, на кшталт точності чи матриці розходження, що є необхідним для валідації успішності тренування. Для практичного застосування сервісу потрібно спроектувати

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

інтерфейс, який дасть змогу як ручного внесення параметрів для нових сутностей, так і автоматичного створення зразків даних за допомогою рандомізації. Ці набори даних мають бути спрямовані на розгорнуту модель (через Endpoint) для отримання передбачення, результат якого потім слід представити користувачеві у доступній формі. Важливою складовою також є необхідність протоколювання всіх виконаних операцій та фіксації історії отриманих прогнозів у структурованій базі даних для подальшого аналітичного опрацювання.

Архітектурне рішення системи мусить брати до уваги нефункціональні критерії, зокрема аспекти, пов'язані зі структурою та захищеністю. Веб-сервісна частина має бути збудована на принципах клієнт-серверної взаємодії із застосуванням RESTful API для зв'язку між модулями, причому операції, що вимагають значних обчислювальних ресурсів, мають оброблятися асинхронно у хмарному середовищі, аби не перешкоджати роботі інтерфейсу. Клієнтська частина бажано має являти собою односторінковий додаток (SPA) із спрощеним та логічним візуальним оформленням. Пріоритетною вимогою є гарантування безпечного доступу до хмарної інфраструктури за допомогою механізму сервісних облікових записів, виключаючи збереження конфіденційних ключів на стороні клієнта, а також забезпечення адекватної реакції на будь-які нештатні ситуації, що можуть виникнути з боку хмарного провайдера.

Технічна реалізація проєкту передбачає використання визначеного технологічного стеку. Серверна частина (Back-end) розроблятиметься мовою програмування Python з використанням фреймворку Flask, що забезпечить гнучкість та простоту інтеграції з хмарними SDK. Клієнтська частина (Front-end) базуватиметься на бібліотеці React.js для створення динамічного інтерфейсу. Для збереження метаданих проєктів та історії запитів обрано систему керування базами даних PostgreSQL. Хмарна інфраструктура базуватиметься на сервісах Google Cloud Storage для зберігання об'єктів та Vertex AI для виконання задач машинного навчання. Реалізація сформульованих вимог дозволить створити повноцінний програмний продукт для автоматизованого аналізу даних.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Ця платформа являє собою односторінковий веб-додаток (SPA), що функціонує за принципом "клієнт-сервер" і залучає зовнішні хмарні сервіси ШІ (Machine Learning as a Service, MLaaS) для здійснення аналітичних операцій.

Роботу системи можна простежити через ключовий сценарій взаємодії користувача з платформою (User Flow):

1. Вхід до системи. Користувач (наприклад, фахівець із даних чи маркетолог) звертається до веб-інтерфейсу. Йому необхідно пройти процедуру реєстрації з наданням своїх даних, які потім фіксуються у сховищі PostgreSQL. Після цього він отримує доступ до свого робочого простору.

2. Створення аналітичного проекту та імпорт даних. У персональному кабінеті ініціюється новий "Проект аналізу". Центральним моментом тут є завантаження масиву даних (датованого), найчастіше у форматах .csv або .json. Клієнтська частина (розроблена на React) передає цей файл на серверний рівень.

3. Опрацювання та перевірка даних. Серверний рівень (створений на Flask) отримує файл. З метою економії ресурсів він не зберігає цей файл у повному обсязі, а опрацьовує його "на льоту" за допомогою інструментарію Pandas. Сервер виконує валідацію, перевіряючи структуру даних, відповідність типів у стовпцях та наявність пропусків. Інформація про метадані цього масиву (перелік колонок, кількість записів, типи значень) фіксується у PostgreSQL, прив'язана до проекту користувача, згідно з технічною документацією СУБД PostgreSQL.

4. Вибір та налаштування механізму ШІ-аналізу. Користувачеві в інтерфейсі надається можливість обрати потрібний тип аналізу:

- Скористатися готовим API: Наприклад, для тексту можна замовити "Аналіз емоційного забарвлення" (сентимент-аналіз).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- Запустити процес навчання власної моделі (AutoML): Приміром, "Прогнозування значення (Регресія)" або "Розподіл за категоріями (Класифікація)". Користувач повинен чітко вказати, яку саме змінну він бажає передбачати (цільовий параметр).

5. Активування аналітичного процесу (Комунікація з Vertex AI). Це є ключовою складовою платформи.

- Якщо обрано стандартний API: Серверний додаток на Flask здійснює послідовну відправку запитів до Google Natural Language API для кожного елемента обраного стовпця у датасеті.

- Якщо обрано AutoML: Сервер формує запит до Google Vertex AI API для ініціації створення та тренування нової моделі, передаючи йому посилання на попередньо очищені дані, які були завантажені у хмарне сховище Google Cloud Storage. Весь процес навчання бере на себе Vertex AI. Це фоновий, неблокуючий процес, час виконання якого може коливатися від декількох хвилин до кількох годин.

6. Отримання та графічне представлення результатів.

- Після завершення роботи аналізу (чи тренування моделі та отримання прогнозу), Vertex AI надсилає результат назад на сервер.

- Сервер обробляє отримані висновки та зберігає їх у PostgreSQL.

- Клієнтська частина (React) періодично запитує стан результатів у сервера. Коли вони готові, отримує їх та візуалізує для користувача у зрозумілому форматі: за допомогою графіків (використовуючи бібліотеки на кшталт Chart.js або D3.js), діаграм (наприклад, матриці невідповідностей для моделі класифікації) та зведених таблиць.

Таким чином, платформа повністю приховує від кінцевого користувача усі складнощі, пов'язані з машинним навчанням та хмарною інфраструктурою, пропонуючи зручний інтерфейс для ефективного аналізу даних.

3.2 Розробка структурної схеми

Структурна схема системи програмного комплексу наведена на рисунку 3.1. Вона відображає ієрархічну архітектуру, центром якої є веб-сервер (REST API). Він виступає посередником, що приймає запити від клієнтського інтерфейсу (React.js), зберігає метадані у реляційній базі даних (PostgreSQL) та керує обчислювальними процесами у хмарному середовищі (Google Cloud Platform).

Систему функціонально поділено на чотири основні блоки: Клієнтська частина (Фронт-Енд), Серверна частина (Бек-Енд), База Даних (PostgreSQL) і зовнішній ресурс Google Cloud Platform.

1. Клієнт (Фронт-Енд): Це інтерфейс користувача, реалізований як вебдодаток на технології React. Користувач взаємодіє саме з цим елементом: проходить ідентифікацію, створює проєктні завдання, завантажує масиви інформації та переглядає кінцеві результати. Уся комунікація із серверною частиною відбувається не синхронно, за допомогою HTTP-запитів (за схемою REST API), обмінюючись даними у форматі JSON.

2. Сервер (Бек-Енд): Центральне ядро обробки, розроблене на Python із застосуванням фреймворку Flask. Він керує основною бізнес-логікою: автентифікацією осіб, керуванням проєктами, перевіркою та попередньою підготовкою даних (із залученням бібліотеки Pandas), а також взаємодією з іншими вузлами. Варто зазначити, що сам бекенд не виконує обчислювально складних операцій, пов'язаних зі штучним інтелектом.

3. База Даних (PostgreSQL): Система керування базами даних PostgreSQL скомпонована з бекенд-сервером. Вона відповідає за надійне зберігання даних, які не змінюються часто: інформації про облікові записи користувачів, метаданих проєктів, завантажених файлів, а також фінальних висновків, отриманих від ШІ-сервісу.

4. Google Cloud Platform (MLaaS): Це зовнішня, повністю керована послуга. Бекенд зв'язується з нею через захищений інтерфейс Google Cloud API,

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок–схеми та опис алгоритмів функціонування системи

Основним процесом, що визначає функціональність системи є алгоритм обробки запиту на аналіз даних, який виконується на стороні back-end (Python/Flask). Цей алгоритм (деталізований на Рис 4.1 – Блок-схема основної програми) активується, коли користувач на front-end натискає кнопку "Аналізувати".

Опис алгоритму

1. Початок. Сервер отримує HTTP-запит від клієнта (React).
2. Автентифікація. Сервер перевіряє токен автентифікації користувача.
3. Валідація даних: Сервер аналізує дані, що надійшли: чи існує проект, чи коректно вказані параметри аналізу.
4. Рішення (Перевірка валідації):
 - Ні (Помилка): Якщо дані невалідні, алгоритм негайно формує відповідь про помилку (наприклад, "Error 400: Bad Request") і відправляє її користувачеві. Процес завершується.
 - Так (Добре): Якщо дані коректні, процес продовжується.
5. Рішення (Тип аналізу). Алгоритм перевіряє, який тип аналізу обрав користувач.
 - "Готове API" (напр., аналіз тексту). Формується запит до відповідного Google API (напр., *Natural Language API*).
 - "AutoML / Власна модель". Формується запит до розгорнутої кінцевої точки (Endpoint) у *Google Vertex AI*.
6. Виклик Google Cloud API. Виконується мережевий запит до хмарної

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Ключовим етапом основного алгоритму є "Виклик Підпрограми: Обробка Аналізу". Ця підпрограма (деталізована на Рис. 4.2 – Блок-схема підпрограми обробки аналізу) відповідає за безпосередню взаємодію з хмарними сервісами Google Cloud.

Робота підпрограми включає наступні кроки:

1. Визначення типу аналізу, обраного користувачем (AutoML чи готове API).
2. Формування відповідного API-запиту до Google Vertex AI або іншого сервісу (наприклад, Language API).
3. Виконання мережевого виклику до Google Cloud API.
4. Перевірка успішності запиту:
 - У разі помилки: Логування помилки та підготовка відповідного повідомлення для користувача.
 - У разі успіху: Отримання та розбір JSON-відповіді з результатами аналізу.
5. Збереження отриманих результатів (прогнозів, метрик) у базу даних PostgreSQL.
6. Підготовка фінальних даних для передачі назад основній програмі (для відправки клієнту).
7. Завершення підпрограми та повернення результату або повідомлення про помилку.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Алгоритм шифрування AES-256

AES (Advanced Encryption Standard) - це алгоритм симетричного блокового шифрування, який визнано стандартом у США та активно використовується для захисту даних. У даній системі застосовано версію з 256-бітним ключем, що дозволяє забезпечити стійкість до спроб злому методом повного перебору (Brute-force attack).

В основі математичного апарату AES лежать операції у скінченному полі Галуа $GF(2^8)$. Алгоритм працює з блоками даних фіксованого розміру (128 біт), які формують матрицю стану (State) розмірністю 4×4 байти.

Процес шифрування для AES-256 складається з 14 раундів перетворень. Кожен раунд (окрім останнього) включає чотири основні трансформації:

1. SubBytes. Нелінійна заміна кожного байта матриці стану з використанням таблиці підстановок (S-Box). S-Box побудований на основі обернених елементів у полі GF та афінного перетворення, що забезпечує захист від лінійного та диференціального криптоаналізу.

2. ShiftRows. Циклічний зсув рядків матриці стану. Нульовий рядок не зсувається, перший зсувається на 1 байт вліво, другий - на 2, третій - на 3. Це забезпечує дифузю даних між стовпцями.

3. MixColumns. Перемішування даних у кожному стовпці матриці стану. Кожен стовпець розглядається як поліном над $GF(2^8)$ і множиться по модулю $x^4 + 1$ на фіксований поліном $c(x)$.

4. AddRoundKey. Додавання раундового ключа до матриці стану за допомогою операції XOR (виключне АБО). Раундові ключі генеруються з основного ключа шифрування за допомогою процедури розширення ключа (Key Expansion).

Вибір алгоритму AES-256 пояснюється його високою швидкістю на сучасному обладнанні (за рахунок підтримки інструкцій AES-NI), а також відсутністю практичних вразливостей, які дозволили б обійти повний перебір паролів. Для файлів CSV, що знаходяться у Google Cloud Storage, застосовується

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

автоматичне шифрування на стороні сервера. Це дозволяє зберегти конфіденційність інформації навіть у гіпотетичному випадку фізичного вилучення дисків з дата-центру.

Захист даних при передачі (Data in Transit)

Для захисту каналів зв'язку між клієнтською частиною, сервером та хмарною платформою використано протокол TLS 1.3. Ця версія відрізняється від попередніх тим, що в ній прибрано підтримку застарілих алгоритмів хешування (MD5, SHA-1).

Натомість для обміну ключами обов'язково застосовується алгоритм Діффі-Геллмана на еліптичних кривих (ECDHE). Такий підхід забезпечує властивість прямої секретності (PFS) навіть якщо в майбутньому ключ сервера буде викрадено, зловмисник не зможе розшифрувати раніше перехоплений трафік. Автентифікація компонентів системи (Service Accounts).

Авторизація серверної складової в середовищі Google Cloud базується на використанні сервісних акаунтів (Service Accounts). Для перевірки легітимності запитів застосовано асиметричний алгоритм шифрування RSA-2048.

Приватний ключ зберігається виключно на стороні сервера у захищеному файлі конфігурації (key.json) і не передається по мережі. Публічна частина ключа знаходиться в системі керування доступом Google IAM. При кожній взаємодії з API (скажімо, для старту AutoML) сервер створює токен стандарту JWT, підписуючи його своїм приватним ключем.

Математична модель алгоритму RSA

Криптостійкість механізму авторизації базується на складності задачі факторизації великих цілих чисел. Алгоритм генерації ключів RSA, що використовується в системі, описується наступною послідовністю:

1. Генеруються два великих простих числа p та q .
2. Обчислюється модуль системи $n = p \cdot q$, довжина n у бітах визначає довжину ключа (2048 біт).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Захист від ін'єкцій коду (SQL Injection Prevention)

Для убезпечення бази даних PostgreSQL від SQL-ін'єкцій (спроб впровадження шкідливого коду) застосовано метод параметризованих запитів. У кодї відсутня пряма конкатенація рядків при створенні SQL-команд. Натомість використовується функціонал бібліотеки `psycopg2`, що забезпечує автоматичне екранування спеціальних символів.

Санітизація вхідних даних (Input Sanitization)

Всі файли, що завантажуються користувачем (зокрема у форматі CSV), підлягають суворій валідації. Перед обробкою бібліотекою Pandas сервер аналізує MIME-тип та структуру заголовків файлу. Це унеможлиблює завантаження шкідливих скриптів або виконуваних файлів (наприклад, розширень `.exe` чи `.py`), замаскованих під табличні дані.

Аудит та логування подій безпеки

Для моніторингу безпеки та подальшого розслідування інцидентів ведеться журнал аудиту подій. У ньому фіксуються факти невдалої авторизації, спроби завантаження некоректних файлів та помилки доступу до хмарних ресурсів. Важливо, що логи зберігаються ізольовано від основної бази даних, що значно ускладнює їх видалення чи модифікацію зловмисником.

Висновки до підрозділу

Загалом система захисту побудована за принципом ешелонованої оборони. Комплексне застосування шифрування AES-256 для зберігання даних, ключів RSA-2048 для авторизації та протоколу TLS 1.3 для передачі трафіку дозволяє забезпечити рівень безпеки, необхідний для сучасних веб-сервісів, що працюють з конфіденційною інформацією.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Методика розгортання (Deployment) моделі на серверах Google Cloud Endpoint

Впровадження розробленої системи у промислове використання здійснюється на засадах концепції MLOps (Machine Learning Operations), котра передбачає автоматичне доведення навчених моделей до роботи у "хмарі" з метою надання доступу до них кінцевим споживачам.

Для етапу Деплойменту у цій роботі було обрано сервіс Google Vertex AI Endpoints, що дає змогу розміщувати моделі на інфраструктурі, яка може динамічно нарощувати потужності, забезпечуючи при цьому мінімум затримок (низьку латентність) під час обробки запитів.

Процедура виведення моделі у режим промислової експлуатації має таку послідовність кроків

1. Унесення моделі до реєстру (Model Registration). По завершенню автоматичного навчання (AutoML), найкраща версія моделі, що продемонструвала найвищий показник AuROC, автоматично фіксується у внутрішньому сховищі моделей Google Cloud (Model Registry). Це критично важливо для обліку версій та швидкого повернення до попереднього стану, якщо виникне така потреба.

2. Визначення параметрів обчислювальних потужностей (Compute Resources). Для обслуговування моделі потрібне виділення віртуальних серверів. Під час проектування було проведено аналіз потреб системи та встановлено таку конфігурацію:

Типовий розмір машини (Machine Type): n1-standard-2 (2 віртуальні процесори, 7.5 ГБ оперативної пам'яті). Такий вибір обґрунтовано оптимальним співвідношенням між вартістю оренди та швидкістю, необхідною для обробки табличних даних. Для завдань бінарної класифікації (як от у випадку з відтоком

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

клієнтів банку – Bank Churn) використання графічних процесорів (GPU) є надмірним і фінансово невиправданим.

Локація розгортання us-central1. Цей регіон хмарного провайдера обрано завдяки найнижчим тарифам на обчислювальні ресурси та наявності повного спектру сервісів Vertex AI.

3. Встановлення параметрів управління навантаженням-автомасштабуванням (Scaling Strategy). Для забезпечення сталості функціонування веб-сервісу налаштовано наступні механізми горизонтального автомасштабування:

Мінімальний ліміт активних копій (Min Replica Count): 1. Це гарантує, що сервіс завжди готовий прийняти запит, уникаючи затримки "холодного старту".

Максимальна кількість копій (Max Replica Count): 1. В межах магістерської кваліфікаційної роботи та для демонстраційного зразка встановлено обмеження задля контролю бюджету (Cost Management). У реальних умовах цей параметр має бути підвищено відповідно до очікуваного обсягу трафіку.

4. Запуск у роботу та валідація коректності (Health Check). Після застосування усіх налаштувань модель завантажується на сконфігурований Endpoints. Системі присвоюється унікальний ідентифікатор (Endpoint ID), який потім інтегрується у програмний код клієнтського додатка. Обмін даними відбувається через безпечний протокол HTTPS, використовуючи формат JSON.

Застосування цієї методології дозволило отримати надійний веб-сервіс, доступний цілодобово, здатний оперативно надавати прогнози в режимі реального часу.

Інструкція користувача веб-сервісу

Створена мною платформа пропонує простий для освоєння графічний інтерфейс, представлений у вигляді веб-додатку. Робота з цією системою не вимагає від користувача жодних знань у сфері програмування, а взаємодія відбувається згідно з таким порядком кроків:

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

1. Завантаження початкових файлів. На старті роботи необхідно скористатися секцією «1. Завантаження даних». Тут слід активувати кнопку для вибору файлу та вказати розташування файлу у форматі CSV, який містить дані, необхідні для аналізу (скажімо, історія банківських операцій). Після натискання кнопки «Відправити в Хмару» файл безпечно транспортується до сховища Google Cloud Storage, а користувачеві повертається унікальний код цього завантаження.

2. Реєстрація даних у системі ШІ. Секція «2. Реєстрація в AI» дозволяє розпочати процес створення керованого набору даних (Managed Dataset) безпосередньо на платформі Vertex AI. Клік на «Створити Датасет» ініціює автоматичну перевірку структури даних, підтвердження відповідності типів колонок та їх індексацію провайдером хмарних послуг.

3. Налаштування та старт тренування. Секція «3. Запуск Навчання» призначена для конфігурації параметрів для AutoML. Користувачу необхідно чітко вказати, яку колонку потрібно передбачити (Target Column) (наприклад, поле Exited у завданнях прогнозування відтоку). Запуск процесу («START AutoML») призводить до автоматичного підбору найкращого алгоритму машинного навчання. Оскільки цей етап вимагає значних ресурсів і часу, він виконується у фоновому режимі.

4. Контроль стану системи. Для відстеження прогресу використовується розділ «4. Моніторинг і Результати». Користувач може перевірити актуальний статус тренування моделі, використовуючи номер задачі (Job ID). Якщо процес завершено успішно (статус SUCCEEDED), система надає пряме посилання на звіт, де детально описані усі метрики точності отриманої моделі.

5. Отримання прогнозу (Аналітика). Останній етап роботи відбувається у блоці «5. Аналітика». Тут необхідно ввести ідентифікатор вже розгорнутого сервера (Endpoint ID) та заповнити всі необхідні параметри тестового клієнта (вік, залишок на рахунку, кредитний рейтинг тощо). Для швидкої перевірки передбачено функцію генерації випадкового профілю («Рандом»). Після

надсилання запиту («Прогноз») в режимі реального часу відображається результат класифікації та ступінь вірогідності події у відсотковому виразі.

Загальний вигляд інтерфейсу

Інтерфейс веб-сервісу спроектовано за принципом «єдиного вікна» (Dashboard), що дозволяє користувачеві мати доступ до всіх функціональних блоків на одній сторінці без необхідності переходу між вкладками. Така організація робочого простору значно підвищує ергономічність системи та пришвидшує процес аналізу даних. Повний вигляд головного вікна розробленого програмного продукту з усіма активними модулями наведено на рисунку 5.1.

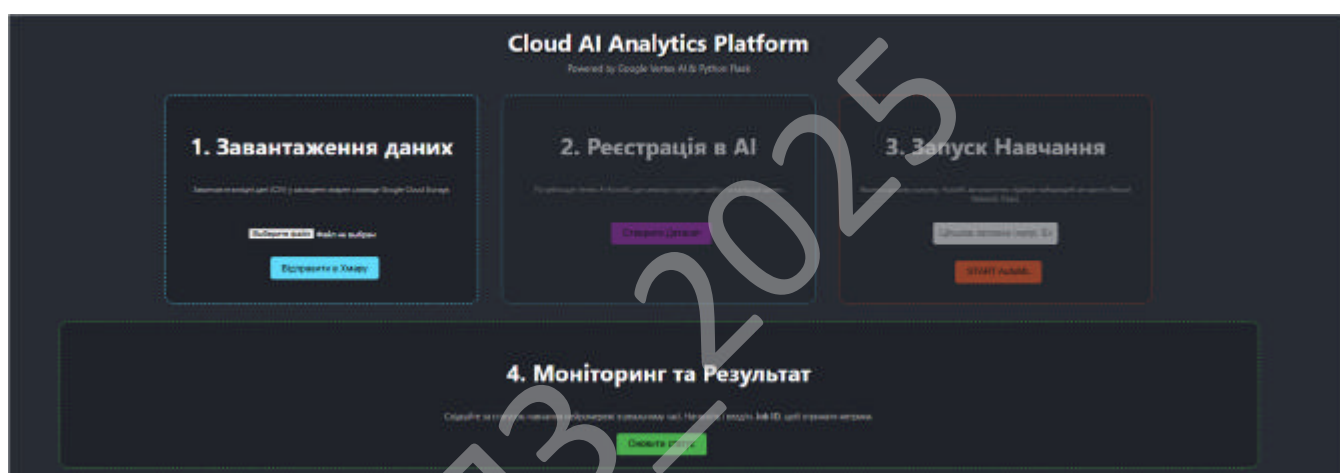


Рисунок 5.1 – Загальний вигляд інтерфейсу веб-сервісу

Слідування цій чіткій послідовності гарантує коректну роботу усіх компонентів системи та отримання надійних результатів аналізу.

На рисунку 5.2 зображено вікно «Про розробника», яке містить інформацію про авторські права на розроблений програмний продукт.



Рисунок 5.2 – Інформація про розробника та авторське право

Для розробленого веб-сервісу обрано модель розповсюдження SaaS (Software as a Service) з використанням ліцензії Freemium. Ця модель передбачає наступні умови використання:

1. Базовий доступ. Користувачі мають можливість безкоштовно використовувати основний функціонал системи (завантаження даних, створення датасетів) з обмеженням на обсяг файлів та кількість запитів до API. Це дозволяє потенційним клієнтам оцінити можливості системи без фінансових ризиків.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

2. Комерційна ліцензія. Для доступу до розширених функцій (навчання моделей, необмежена кількість прогнозів, пріоритетна підтримка) передбачено платну підписку.

3. Захист інтелектуальної власності. Авторські права на вихідний код, алгоритми інтеграції з Google Cloud та дизайн інтерфейсу належать розробнику. Копіювання, модифікація або комерційне використання програмного коду без дозволу автора заборонено.

Такий підхід до ліцензування відповідає сучасним тенденціям ринку хмарних послуг і дозволяє ефективно монетизувати розробку, зберігаючи при цьому доступність для широкого кола користувачів.

КБПЗ – 2025

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

6 НАУКОВА НОВИЗНА

У магістерській кваліфікаційній роботі отримано нові науково-практичні результати у галузі інформаційних технологій та систем штучного інтелекту, які полягають у наступному:

Удосконалено архітектурний підхід до побудови систем хмарного аналізу даних. Запропоновано та реалізовано модель взаємодії компонентів за схемою «Тонкий клієнт – Сервер-оркестратор – Хмарна платформа» (React – Flask – Google Cloud). Новизна підходу полягає у створенні проміжного шару абстракції, який повністю інкапсулює складність налаштування інфраструктури машинного навчання (Vertex AI). На відміну від існуючих рішень, де налаштування середовища вимагає високої кваліфікації DevOps-інженера, запропонована архітектура дозволяє автоматизувати процеси Data Ingestion та Model Deployment через уніфікований API, забезпечуючи при цьому високий рівень безпеки завдяки використанню механізму Service Accounts без передачі ключів доступу на клієнтську сторону.

Набуло подальшого розвитку застосування технології AutoML для задач бізнес-аналітики. На основі проведеного експериментального дослідження доведено ефективність використання методів автоматизованого машинного навчання для вирішення задачі прогнозування відтоку клієнтів (Bank Customer Churn) в умовах обмежених вибірок даних. Встановлено, що використання запропонованого комплексу дозволяє досягти показників якості моделі AuROC = 0.906, що співставно з результатами роботи спеціалізованих команд Data Science, але при значно менших часових витратах на етапи Feature Engineering та підбору гіперпараметрів.

Дістало подальшого розвитку методичне забезпечення процесів MLOps для малого бізнесу. Сформовано та апробовано методику повного циклу розробки ML-рішень (від завантаження CSV до отримання REST Endpoint), яка, на відміну від

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

стандартних корпоративних пайплайнів, адаптована для використання неспеціалістами. Це дозволяє суттєво знизити поріг входження у використання технологій штучного інтелекту для підприємств малого та середнього бізнесу.

КБПЗ – 2025

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Працюючи над проєктом веб-сервісу для аналізу даних із використанням ШІ у хмарному середовищі, я постійно думав, кому він буде справді корисним. І перше, що спадає на думку - це компанії, які збирають велику кількість даних, але не мають гнучкого інструменту для швидкого їх аналізу. Наприклад, у маркетингу, логістиці або фінансах.

Ще одна очевидна група - це стартапи, які не мають ресурсу створювати власну аналітичну інфраструктуру з нуля. Для них можливість підключитися до хмарного рішення, яке вже має побудовану логіку, ШІ-алгоритми і зручний інтерфейс, - це справжній порятунок. Особливо якщо сервіс підтримує обробку неструктурованих даних або працює з великими обсягами.

Також, я вважаю, що мій сервіс може зацікавити викладачів і студентів технічних спеціальностей. Їм часто бракує простих інструментів, які дозволяють вивчати машинне навчання на практиці без складних розгортань локально. Тож освітні установи теж можуть стати потенційними користувачами.

Не можна забути й про державні структури, які все активніше впроваджують цифрові інструменти. Аналіз великих обсягів статистики, опитувань або звітів потребує не просто Excel, а сучасного інструменту з елементами автоматизації й прогнозування. І тут мій сервіс може стати в пригоді.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Щоб не робити висновки лише зі своєї бульбашки універсу, я вирішив протестувати ідею через метод експертних оцінок. Це дозволило мені подивитися на свій проєкт очима людей, які щодня працюють із даними або розробляють схожі сервіси.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Я звернувся до трьох аналітиків із комерційних компаній, викладача з кафедри кібернетики, фахівця з Big Data та одного стартапера з Data Science спільноти. Кожному я надіслав коротку презентацію про функціонал, архітектуру і потенційні варіанти застосування веб-сервісу.

Усі експерти отримали анкету, де треба було оцінити проєкт за критеріями: інноваційність, масштабованість, легкість інтеграції, комерційний потенціал і практична цінність. Оцінювання проводилося за шкалою від 1 до 10, а потім я обрахував середній бал по кожному критерію.

Найвищі оцінки отримала саме частина, що стосується автоматизації аналізу без глибокого залучення програмістів. В результаті середній показник привабливості склав 8,7 з 10. Це підтвердило, що розробка має сенс не лише як диплом, а й як інструмент, що дійсно може бути використаний у реальному житті.

7.3 Вибір методу оцінки вартості ПЗ

Коли постало питання оцінки вартості мого веб-сервісу, я зрозумів, що просто рахувати затрачені години - це не зовсім справедливо. Але й оцінювати проєкт виключно за майбутнім прибутком - поки що рано. Тому я зупинився на комбінованому підході: метод витрат плюс метод аналогів.

Спочатку я порахував усе, що витратив: свій час на розробку, вартість хмарного хостингу, домену, а також ліцензії на деякі бібліотеки. Сумарно вийшло приблизно 35 000 грн, якщо умовно монетизувати робочий час. Це вже дало мені уявлення про «собівартість» продукту.

Потім я проаналізував ціни на аналогічні сервіси - зокрема на продукти, які використовують AutoML або простий інтерфейс для обробки CSV-файлів через хмару. Там середня підписка на місяць коливається від 10 до 50 доларів залежно від функціоналу.

Це дало мені змогу вивести справедливую ринкову ціну. За такого функціоналу, як мій, навіть базова версія може коштувати 15–20 доларів на

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

місяць. Якщо врахувати, що підписку можуть взяти хоча б 100 користувачів, то економічний потенціал виглядає вже доволі реалістично.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Типове підприємство або організація, яка працює з великими масивами даних, зазвичай використовує розрізнені аналітичні інструменти, встановлені на локальних комп'ютерах або в рамках корпоративних ліцензій. Це призводить до: високої залежності від людського фактору при обробці даних; затримок у прийнятті управлінських рішень через повільну аналітику; дублювання інформації та неузгодженості звітності між підрозділами; витрат на ПЗ, яке не масштабується під нові запити; відсутності можливості швидко інтегрувати машинне навчання та прогнозування у звичайні бізнес-процеси.

Впровадження розробленого хмарного веб-сервісу із вбудованими модулями штучного інтелекту дозволяє: централізувати аналітику даних та забезпечити швидкий доступ до результатів з будь-якого пристрою; скоротити витрати на сторонні ліцензії завдяки моделі SaaS; зменшити навантаження на внутрішній IT-відділ; підвищити якість управлінських рішень завдяки прогнозним моделям і візуалізації; знизити бар'єр входу для нетехнічних працівників у роботу з даними.

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	До впровадження	Після впровадження	Економічний ефект
Середня кількість годин, витрачених на аналітику даних у відділі, год/місяць	180	90	-90 год/міс
Кількість фахівців, що залучені до обробки та підготовки звітів	3	1	-2 особи
Середня вартість 1 робочої години фахівця з аналітики, грн	400	400	-

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування мого веб-сервісу я би почав з простого, але чесного ознайомлення потенційних користувачів із його функціоналом. Створити лендінг - це перший крок. На ньому я б коротко показав, що сервіс вмiє, як виглядає інтерфейс і чим він вигідніший за аналоги. Без перебільшень і маркетингового шуму.

Наступним кроком я би зробив серію коротких відеоуроків на YouTube або в TikTok (як не дивно), де б показував - як завантажити файл, як побудувати візуалізацію, як зробити прогноз. Такий формат, до речі, дуже популярний серед студентів і молодих підприємців.

Також я би написав кілька гостьових статей або кейсів у професійні блоги, зокрема на DOU, Medium або Dev.to. Мій проєкт міг би стати темою статті "як з нуля зробити свій аналітичний SaaS", і це було б цікаво навіть самим розробникам.

Окремо я бачу сенс у запуску проєкту на платформах типу ProductHunt або IndieHackers - там аудиторія якраз цінує MVP з потенціалом. А для розширення охоплення - можна таргетовано запускати рекламу в LinkedIn серед маркетологів, викладачів або спеціалістів з аналітики.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Коли мова заходить про збут, я розумію, що у випадку з веб-сервісом усе залежить від того, як просто користувач може почати. Тому я вважаю, що найкращий шлях - це фріміум-модель. Дати можливість спробувати аналіз невеликого обсягу даних безкоштовно, щоб люди відчули цінність.

Ще один момент - це інтеграції. Якщо сервіс дозволяє легко підключати Google Sheets, Airtable або інші звичні системи, користувачі не будуть відчувати бар'єру входу. Тому я би акцентував на таких простих шляхах підключення.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Щодо каналів реалізації - я б не намагався одразу йти в корпоративний сегмент. Натомість краще вийти на спільноти - ті самі Reddit-гілки про data analysis, студентські спільноти в Telegram, Slack-групи для розробників. Там можна не продавати, а просто ділитися - і це викликає більше довіри.

Також варто подумати про партнерства з освітніми платформами або невеликими онлайн-школами, які могли б включати мій сервіс як інструмент для практики. Це не дає прямого прибутку, але формує аудиторію і бренд.

7.7 Визначення ключових факторів успіху конкретного проєкту

Найголовніше для успіху цього проєкту - це, напевно, реальна корисність. Якщо сервіс справді допомагає заощадити час і дає розуміння даних людям без технічної освіти - це вже половина перемоги. Бо багато хто просто не хоче "копатися" в Python чи R.

Друге - це простота. Люди не хочуть вчитися новому інтерфейсу з нуля. Якщо сервіс виглядає знайомо, інтуїтивно і не потребує годин налаштувань - шанси, що ним користуватимуться, набагато більші.

Також важливо регулярно оновлювати й розвивати продукт. Якщо сервіс стоїть на місці - користувачі йдуть. Я бачив багато інструментів, які «вигоріли» через відсутність підтримки. Тому навіть маленькі апдейти створюють відчуття живого проєкту.

Ну і не менш важливо - прозорість. Люди хочуть розуміти, куди йдуть їхні дані, як вони обробляються, і хто за всім цим стоїть. Якщо все буде чесно і відкрито - це значно підвищить рівень довіри, а отже й шанси на успішне зростання проєкту.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1. Вступ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Правовою основою організації робіт з охорони праці є Конституція України, Закон України «Про охорону праці», Кодекс законів про працю України (КЗпП) та державні нормативні акти з охорони праці.

Метою даного розділу магістерської роботи є аналіз умов праці інженера-програміста під час розробки веб-сервісу для аналізу даних, виявлення потенційно небезпечних та шкідливих виробничих факторів, а також розробка комплексних заходів щодо їх усунення або мінімізації до нормативних значень.

Актуальність питань охорони праці для працівників ІТ-сфери обумовлена специфікою їхньої діяльності: тривалою роботою з відеодисплейними терміналами (ВДТ), високим нервово-емоційним напруженням та гіподинамією. Створення безпечних умов праці є запорукою високої продуктивності розробника та запобігання професійним захворюванням.

8.2. Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Об'єктом дослідження є робоче місце розробника програмного забезпечення, обладнане персональним комп'ютером. Робота виконується у приміщенні кафедри.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Продовження таблиці 8.1

Група факторів	Найменування фактору	Джерело виникнення	Нормативний документ
	Небезпека ураження електричним струмом	Електрична мережа 220В, 50Гц	ПУЕ, НПАОП 40.1-1.21-98
Психофізіологічні	Нервово-емоційне перенапруження	Розв'язання складних алгоритмічних задач, дефіцит часу	ДСанПіН 3.3.2.007-98
	Перенапруження зорового аналізатора	Тривала робота з об'єктами розрізнення на екрані	-//-
	Монотонність праці та гіподинамія	Тривале перебування у фіксованій позі (сидячи)	-//-

Робота програміста відноситься до категорії 1а (легка фізична робота), оскільки енерговитрати становлять до 139 Вт, а робота виконується сидячи і не потребує систематичного фізичного напруження. Параметри мікроклімату повинні відповідати вимогам ДСН 3.3.6.042-99. Оптимальні та допустимі норми наведено в таблиці 8.2.

Таблиця 8.2 – Норми мікроклімату для категорії робіт 1а

Період року	Параметр	Оптимальні величини	Допустимі величини
Холодний	Температура повітря, °С	22 – 24	21 – 25
	Відносна вологість, %	40 – 60	не більше 75
	Швидкість руху повітря, м/с	0,1	не більше 0,1

Продовження таблиці 8.2

Період року	Параметр	Оптимальні величини	Допустимі величини
Теплий	Температура повітря, °С	23 – 25	22 – 28
	Відносна вологість, %	40 – 60	55 (при 28°С)
	Швидкість руху повітря, м/с	0,1	0,1 – 0,2

Для підтримання цих параметрів приміщення обладнане системою централізованого водяного опалення (взимку) та побутовим кондиціонером (влітку). Вентиляція – природна, через вікна та кватирки.

Аналіз виробничого освітлення

Робота з комп'ютером вимагає високої концентрації зору і належить до розряду зорових робіт III (високої точності). Згідно з ДБН В.2.5-28-2018 «Природне і штучне освітлення», у приміщенні передбачено суміщене освітлення:

1. Природне. бічне, здійснюється через два вікна розміром 1,5 · 1,8м. Коефіцієнт природної освітленості (КПО) становить $e = 1,5\%$.
2. Штучне. загальне рівномірне освітлення забезпечується стельовими світильниками з LED-лампами.

Нормована освітленість робочої поверхні столу повинна становити не менше 300–500 лк. Також важливим параметром є обмеження прямої блискучості джерел світла та відбитої блискучості на екрані монітора. Коефіцієнт пульсації освітленості не повинен перевищувати 5%.

Аналіз шуму та вібрації

Джерелами шуму в приміщенні є вентилятори охолодження комп'ютерної техніки (кулери процесора, відеокарти, блоку живлення) та зовнішній шум з вулиці.

Відповідно до ДСН 3.3.6.037-99, гранично допустимий рівень звуку на робочих місцях програмістів становить 50 дБА.

Фактичний рівень шуму від сучасного ПК складає близько 35–40 дБА, що не перевищує нормативних значень. Вібрація на робочому місці відсутня.

8.3. Розробка заходів з поліпшення стану охорони праці

Основним напрямком поліпшення умов праці розробника є раціональна організація робочого місця відповідно до вимог ергономіки та фізіології праці. Неправильне розміщення техніки або незручні меблі можуть призвести до швидкої втомлюваності та зниження продуктивності.

Ергономічні вимоги до організації робочого місця

Робоче місце організовано за принципом функціонального зонування. Згідно з ДСанПіН 3.3.2.007-98, конструкція робочого столу та крісла повинна забезпечувати підтримання раціональної робочої пози (кут нахилу корпусу – 15–20° вперед або назад, кут згинання ліктів – 90°).

1. Робочий стіл. Використовується стіл модульної конструкції. Висота робочої поверхні становить 725 мм. Поверхня столу має матове покриття світло-сірого кольору з коефіцієнтом відбиття 0,4, що запобігає утворенню відблисків. Простір для ніг має розміри: висота – 600 мм, ширина – 500 мм, глибина – 450 мм.

2. Робоче крісло. Використовується підйомно-поворотне крісло, яке регулюється за висотою та кутом нахилу спинки.

- Висота сидіння регулюється в межах 400–550 мм.
- Ширина і глибина сидіння становлять не менше 400 мм.
- Спинка має анатомічну форму, що підтримує поперековий відділ хребта.

3. Розміщення засобів відображення інформації. Екран відеомонітора розташовано на оптимальній відстані 600–700 мм від очей користувача (але не ближче 500 мм). Верхній край екрану знаходиться на рівні очей або трохи нижче, що забезпечує кут огляду 15–20° вниз. Це дозволяє розвантажити м'язи ший та зменшити висихання рогівки ока (зменшується площа відкритої поверхні ока).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

4. Клавіатура та маніпулятор «миша». Клавіатура розміщена на поверхні столу на відстані 100–300 мм від краю, зверненого до користувача. Це створює опору для рук. Використовується килимок для миші з гелевою подушечкою під зап'ястя для профілактики тунельного синдрому.

Режим праці та відпочинку

Для запобігання перевтомі розроблено раціональний режим праці. При 8-годинному робочому дні передбачені:

- Обідня перерва – 45 хвилин.
- Регламентовані перерви тривалістю 15 хвилин через кожні 2 години роботи. Під час перерв рекомендується виконувати комплекси вправ для очей (офтальмотренаж) та фізичні вправи для зняття напруги з м'язів спини та шиї.

8.4. Техніка безпеки та протипожежна профілактика

Електробезпека Обладнання робочого місця (ПК, монітор, периферія) живиться від однофазної мережі змінного струму напругою 220 В (50 Гц). Згідно з ПУЕ (Правила улаштування електроустановок) та НПАОП 40.1-1.21-98, приміщення кафедри за ступенем небезпеки ураження електричним струмом відноситься до класу приміщень без підвищеної небезпеки (сухе приміщення, струмонепровідна підлога, відсутність агресивного середовища).

Для забезпечення електробезпеки реалізовано такі заходи:

1. Захисне заземлення. Усі металеві неструмопровідні частини обладнання (корпус системного блоку) заземлені через третій провідник (РЕ) в кабелі живлення та євророзетку із заземлюючим контактом. Опір заземлюючого пристрою не перевищує 4 Ом.

2. Ізоляція. Використовуються кабелі з подвійною ізоляцією. Проводиться регулярний візуальний огляд на предмет пошкоджень.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

3. Захист від перевантажень. Лінія живлення обладнана автоматичними вимикачами, які розмикають ланцюг при виникненні струмів короткого замикання.

4. Організаційні заходи. До роботи допускаються особи, які пройшли інструктаж з електробезпеки (І група допуску).

Протипожежна профілактика Згідно з НАПБ Б.03.002-2007, приміщення з комп'ютерною технікою відноситься до категорії «В» (пожежонебезпечні), а за класом зони згідно ПУЕ – до класу П-Іа. Основними причинами можливої пожежі є: перегрів електронних компонентів, коротке замикання, велике перехідний опір у контактах.

Система протипожежного захисту включає:

1. Пожежна сигналізація. Приміщення обладнане автоматичними димовими сповіщувачами типу СПД-3, підключеними до загального пульта спостереження.

2. Первинні засоби пожежогасіння. У приміщенні наявний вуглекислотний вогнегасник типу ВВК-2 (або ВВК-3). Вибір вуглекислотного вогнегасника зумовлений тим, що він не пошкоджує електронне обладнання (на відміну від порошкового) і дозволяє гасити електроустановки під напругою до 1000 В.

3. Евакуаційні заходи. План евакуації розміщено на видному місці біля виходу. Ширина проходів (не менше 1 м) забезпечує вільну евакуацію персоналу. Двері відчиняються у напрямку виходу з приміщення.

Виконання цих заходів зводить ризик виникнення пожежі до мінімуму та забезпечує безпеку персоналу у разі надзвичайної ситуації.

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

8.5. Розрахункова частина. Розрахунок штучного освітлення

Метою розрахунку є визначення необхідної кількості світильників для забезпечення нормованого рівня освітленості на робочому місці програміста $20,0 \text{ м}^2$ $60,0 \text{ м}^3$

Розрахунок проводиться методом коефіцієнта використання світлового потоку.

1. Вихідні дані для розрахунку:

- Довжина приміщення (А): 5,0 м.
- Ширина приміщення (В): 4,0 м.
- Висота приміщення (Н): 3,0 м.
- Висота робочої поверхні (роб): 0,8 м (висота столу).
- Висота підвісу світильників (св): 0,2 м (від стелі).

Розрахункова висота підвісу (h):

$$h = H - h_{\text{роб}} - h_{\text{св}} = 3,0 - 0,8 - 0,2 = 2,0 \text{ м}$$

2. Визначення індексу приміщення (i):

Індекс приміщення характеризує геометрію кімнати і впливає на те, як світло відбивається від стін.

$$i = \frac{S}{h \cdot (A + B)} = \frac{20}{2,0 \cdot (5,0 + 4,0)} = \frac{20}{18} \approx 1,11$$

3. Вибір коефіцієнтів:

- Коефіцієнти відбиття. Стеля – біла 70%, стіни – світло-бежеві 50%, підлога – темна 30%.

- Коефіцієнт використання світлового потоку η . Для індексу $i = 1,1$ та обраних кольорів інтер'єру, згідно з довідковими таблицями, $\eta = 0,45$ (або 45%).

- Коефіцієнт запасу (k). Враховує запилення ламп у процесі експлуатації. Для офісних приміщень з LED-лампами $k = 1,4$.

- Коефіцієнт нерівномірності освітлення (z): Для люмінесцентних та LED ламп $z = 1,1$.

4. Розрахунок необхідного світлового потоку (заг):

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

Загальний світловий потік, який повинні випромінювати всі лампи в кімнаті, розраховується за формулою:

$$F_{\text{заг}} = \frac{E_{\text{норм}} \cdot S \cdot k \cdot z}{\eta}$$
$$F_{\text{заг}} = \frac{400 \cdot 20 \cdot 1,4 \cdot 1,1}{0,45} = \frac{12320}{0,45} \approx 27377 \text{ лм}$$

5. Вибір світильників та ламп:

Для освітлення обираємо сучасні растрові LED-світильники, що вбудовуються у підвісну стелю (тип «Армстронг»).

Обрано світильник: Philips CoreLine Panel.

- Потужність одного світильника: 36 Вт.
- Світловий потік одного світильника (F_1): 3400 лм.

6. Визначення кількості світильників (N):

$$N = \frac{F_{\text{заг}}}{F_1} = \frac{27377}{3400} \approx 8,05$$

Приймаємо кількість світильників $N = 8$ шт.

7. Перевірка фактичної освітленості ($E_{\text{факт}}$):

$$E_{\text{факт}} = \frac{F_1 \cdot N \cdot \eta}{S \cdot k \cdot z} = \frac{3400 \cdot 8 \cdot 0,45}{20 \cdot 1,4 \cdot 1,1} = \frac{12240}{30,8} \approx 397 \text{ лк}$$

Фактична освітленість (397 лк) відрізняється від нормованої (400 лк) менше ніж на 1%, що є допустимим (допускається відхилення -10%...+20%).

Таким чином, для приміщення площею 20 м² необхідно встановити 8 LED-світильників (розміщення у два ряди по 4 світильники), що забезпечить комфортні та безпечні умови праці розробника.

Висновки до розділу

У даному розділі було проведено комплексний аналіз умов праці при розробці програмного забезпечення.

1. Встановлено, що приміщення кафедри за своїми геометричними параметрами відповідає санітарним нормам (20 м² на одну особу).

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

2. Проаналізовано шкідливі фактори (електромагнітне випромінювання, шум, зорове напруження) та запропоновано заходи щодо їх нейтралізації.

3. Розроблено заходи електробезпеки (захисне заземлення) та пожежної профілактики (наявність вуглекислотних вогнегасників).

4. Проведено розрахунок штучного освітлення, в результаті якого визначено необхідність встановлення 8 растрових LED-світильників для забезпечення нормованої освітленості 400 лк.

Запропоновані заходи гарантують безпечні умови праці, збереження здоров'я розробника та високу ефективність роботи над дипломним проєктом.

КБПЗ_2025

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, являє собою веб-сервіс для автоматизованого аналізу даних із використанням методів штучного інтелекту у хмарному середовищі.

Рішення завдання полягало у проведенні комплексного аналізу сучасних підходів до обробки великих даних та огляду ринку хмарних платформ AIaaS. На основі отриманих результатів досліджень було обґрунтовано вибір екосистеми Google Cloud Platform, яка забезпечує найкращі показники інтеграції сервісів автоматизованого навчання (AutoML) та співвідношення ціна/продуктивність.

При створенні програмного забезпечення було використано сучасний сервіс-орієнтований підхід до проектування архітектури. Система реалізована на базі стеку технологій Python (фреймворк Flask) для серверної частини та бібліотеки React.js для клієнтського інтерфейсу, що відповідає актуальним тенденціям у галузі веб-розробки. Для надійного зберігання метаданих проектів використано систему керування базами даних PostgreSQL.

Розроблені під час виконання роботи алгоритми дозволяють успішно вирішувати задачі повного циклу машинного навчання (MLOps): від захищеного завантаження «сирих» CSV-файлів у хмарне сховище Google Cloud Storage до автоматичного розгортання навченої моделі на віддаленому сервері (Endpoint). Для підвищення рівня безпеки реалізовано механізм автентифікації через сервісні акаунти з використанням криптографічних ключів, що унеможливорює передачу конфіденційних даних у відкритому вигляді.

В ході експериментального дослідження на реальному наборі даних банківських транзакцій було підтверджено ефективність запропонованих рішень. Результати автоматичного навчання показали високу здатність системи до узагальнення даних: отримана модель класифікації досягла показника якості

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

AuROC 0.906. Це доводить, що сучасні хмарні інструменти здатні створювати високоточні моделі без безпосередньої участі вузькопрофільних спеціалістів.

Розроблене програмне забезпечення володіє простим, зрозумілим та дружнім для користувача інтерфейсом, який забезпечує простоту використання та дозволяє виконувати моделювання бізнес-ситуацій у реальному часі без необхідності написання програмного коду. В цілому створена система підтверджує правильність використаних проектних рішень, повністю відповідає вимогам технічного завдання та має потенціал для впровадження у фінансовому секторі та інших галузях економіки.

КБПЗ_2025

					ВКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

15. Google Cloud. (2024). Vertex AI Documentation. Retrieved from <https://cloud.google.com/vertex-ai/docs>
16. Google Cloud Skills Boost. (2024). Introduction to Vertex AI Studio. Retrieved from https://www.cloudskillsboost.google/course_templates/552
17. Medium (Ankshuk Ray). (2023). Vertex AI and Document AI Features of GCP for Generative AI. Retrieved from <https://medium.com/@ankshukray/vertex-ai-and-document-ai-features-of-gcp-for-generative-ai-8f84970b474>
18. Amazon Web Services. (2024). Amazon SageMaker Documentation. Retrieved from <https://docs.aws.amazon.com/sagemaker/>
19. Microsoft Azure. (2024). Azure Machine Learning Documentation. Retrieved from <https://learn.microsoft.com/en-us/azure/machine-learning/>
20. Databricks. (2023). Introduction to Databricks Lakehouse Platform. Retrieved from <https://docs.databricks.com/>
21. Hugging Face. (2023). Transformers Documentation. Retrieved from <https://huggingface.co/docs/transformers>
22. Boto3 (AWS SDK for Python). (2024). Developer Guide. Retrieved from <https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>
23. Flask Official Documentation. (2024). Welcome to Flask Documentation (3.0). Retrieved from <https://flask.palletsprojects.com/>
24. React Official Docs. (2024). React Documentation – Learn React. Retrieved from <https://react.dev/>
25. PostgreSQL Global Development Group. (2024). PostgreSQL Documentation (v16). Retrieved from <https://www.postgresql.org/docs/>
26. Python Software Foundation. (2024). Python Documentation (v3.12). Retrieved from <https://docs.python.org/3/>
27. Pandas Development Team. (2024). Pandas Documentation. Retrieved from <https://pandas.pydata.org/docs/>
28. Chart.js Community. (2024). Chart.js Documentation (v4). Retrieved from <https://www.chartjs.org/docs/>

					БКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

29. D3.js Foundation. (2023). Data-Driven Documents Library Documentation. Retrieved from <https://d3js.org/>
30. Firebase Documentation. (2024). Firebase Hosting for Web Apps. Retrieved from <https://firebase.google.com/docs/hosting>
31. Murphy, Kevin P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012.
32. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 3rd ed. O'Reilly Media, 2023.
33. Chollet, François. Deep Learning with Python. 2nd ed. Manning Publications, 2021.
34. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
35. Google Cloud. Vertex AI Documentation. URL: <https://cloud.google.com/vertex-ai>
36. AutoML Documentation. Google Cloud. URL: <https://cloud.google.com/automl>
37. Flask - Official Documentation. URL: <https://flask.palletsprojects.com>
38. React - Official Documentation. URL: <https://react.dev>
39. PostgreSQL - Official Documentation. URL: <https://www.postgresql.org/docs>
40. Docker - Official Documentation. URL: <https://docs.docker.com>
41. Python - Official Documentation. URL: <https://docs.python.org>
42. Scikit-learn Documentation. URL: <https://scikit-learn.org/stable>
43. TensorFlow - Official Documentation. URL: <https://www.tensorflow.org>
(дата звернення: 25.11.2025).
44. Keras Documentation. URL: <https://keras.io>
(дата звернення: 25.11.2025).
45. ДСТУ 2226-93. Автоматизовані системи. Терміни та визначення.

					БКРМ-122.25.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

46. Sculley, D., et al. (2015). Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28.
47. Red Hat. (2024). What is MLOps? Retrieved from <https://www.redhat.com/en/topics/devops/what-is-mlops>
48. IBM Cloud Education. (2023). What is SaaS (Software as a Service)? Retrieved from <https://www.ibm.com/topics/saas>
49. Pallets Projects. (2024). Flask-CORS Documentation. Retrieved from <https://flask-cors.readthedocs.io/en/latest/>
50. REST API Design Guidelines - Google API Style Guide. URL: <https://cloud.google.com/apis/design>
51. Abadi, Martín et al. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.” 2016.
52. Google Cloud. Cloud SQL for PostgreSQL Documentation. URL: <https://cloud.google.com/sql>
53. Kubernetes - Official Documentation. URL: <https://kubernetes.io/docs>
54. NIST. Machine Learning Framework for Safety and Trust. National Institute of Standards and Technology, 2023.
55. European Commission. Ethics Guidelines for Trustworthy AI. 2021.
56. IBM. AutoAI Documentation. URL: <https://www.ibm.com/cloud/autoai>
57. Microsoft Azure. Azure Machine Learning Documentation. URL: <https://learn.microsoft.com/azure/machine-learning>
58. Pandas - Official Documentation. URL: <https://pandas.pydata.org/docs>
59. Plotly - Official Documentation. URL: <https://plotly.com/python>