МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАДИМА ГЕТЬМАНА

Навчально-науковий інститут

«Інститут інформаційних технологій в економіці»

Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА **«Системний аналіз»**

СПЕЦІАЛЬНІСТЬ **124 «Системний аналіз»**

Форма навчання: денна

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему: «Удосконалення технології прийняття управлінських рішень шляхом впровадження концепції Data-Driven Management»

Здобувачки: Карєвої Анжеліки Андріївни

_____

(підпис)

Науковий керівник: д.ф.-м.н., проф. Джалладова І.А.

_____

(підпис)

Робота допущена до захисту перед екзаменаційною комісією з атестації здобувачів вищої освіти (ЕК)

Завідувач кафедри: д.ф.-м.н., проф. Джалладова І.А.

_____

(підпис)

Київ 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАДИМА ГЕТЬМАНА

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА **«Системний аналіз»**

СПЕЦІАЛЬНІСТЬ **124 «Системний аналіз»**

ПОГОДЖЕНО	**ЗАТВЕРДЖУЮ**
Керівник проєктної групи (гарант) освітньо-професійної програми	Завідувач кафедри системного аналізу та кібербезпеки
Колєчкіна Л.М.	Джалладова І.А.
(підпис)	*(підпис)*
2026 р.	2026 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувачці вищої освіти Карєвій Анжеліці Андріївні

денної форми навчання

на підготовку кваліфікаційної магістерської роботи

На тему: «Удосконалення технології прийняття управлінських рішень шляхом впро-вадження концепції Data-Driven Management»

Тему затверджено наказом ректора Університету від «27» квітня 2026 р. № 540-ст

Кваліфікаційна магістерська робота виконується на матеріалах

_____

План кваліфікаційної магістерської роботи

Розділ 1	Теоретико-методологічні засади удосконалення технології прийняття рішень.
Розділ 2	Аналітична оцінка стану технології прийняття рішень.
Розділ 3	Проєктування та реалізація системи Data-Driven підтримки прийняття рішень.

Об'єкт дослідження:	Процес прийняття управлінських рішень в організації.
Предмет дослідження:	Методи вдосконалення технології прийняття рішень на основі концепції Data-Driven Management у межах системного аналізу.
Мета кваліфікаційної магістерської роботи:	Обґрунтування та розроблення підходу до удосконалення технології прийняття управлінських рішень шляхом впровадження концепції Data-Driven Management із використанням методів системного аналізу та математичного моделювання.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі 1

1. Розглянути прийняття рішень як системний процес;
2. Проаналізувати роль Data-Driven Management у формуванні управлінських технологій;
3. Формалізувати задачу вибору альтернатив у вигляді оптимізаційної моделі.

У розділі 2

1. Здійснити аналітичну оцінку існуючої практики прийняття рішень у R&D-команді.
2. Сформувати постановку задачі на розробку моделі автоматизованого прийняття рішень.

У розділі 3

1. Запропонувати прикладну модель інтеграції Data-Driven підходу в управлінський процес.
2. Розробити мінімально життєздатний продукт для системи прийняття рішень.

Завдання підготував науковий керівник

_____ Джалладова І.А.
(підпис) *(ініціали, прізвище)*
«___»_____2026 р.

Завдання одержав здобувач

_____ Карєва А.А.
(підпис) *(ініціали, прізвище)*

«___»_____2026 р.

РЕФЕРАТ

Кваліфікаційна магістерська робота містить 78 сторінок, 18 рисунків, 7 таблиць, список використаних джерел з 23 найменувань, додатки.

«Удосконалення технології прийняття управлінських рішень шляхом впровадження концепції Data-Driven Management»

Об'єктом дослідження є процес прийняття управлінських рішень в організації.

Предметом дослідження є методи вдосконалення технології прийняття рішень на основі концепції Data-Driven Management у межах системного аналізу.

Метою роботи є обґрунтування та розроблення підходу до удосконалення технології прийняття управлінських рішень шляхом впровадження концепції Data-Driven Management із використанням методів системного аналізу та математичного моделювання.

Для досягнення мети передбачено *завдання*:

- розглянути прийняття рішень як системний процес;
- проаналізувати роль Data-Driven Management у формуванні управлінських технологій;
- формалізувати задачу вибору альтернатив у вигляді оптимізаційної моделі;
- здійснити аналітичну оцінку існуючої практики прийняття рішень у R&D-команді;
- запропонувати прикладну модель інтеграції data-driven підходу в управлінський процес.

Проаналізовано існуючу технологію прийняття управлінських рішень у R&D-команді, що працює з NLP-рішеннями та data-driven підходами. Проведено формалізацію процесу вибору альтернатив із використанням методів системного аналізу, багатокритеріальної оцінки та імовірнісного моделювання. Для вирішення виявлених проблем розроблено інтерактивний веб-додаток підтримки прийняття рішень на основі Python та Streamlit, який реалізує нормалізацію критеріїв, оцінку корисності, Monte Carlo симуляції, Байєсівський аналіз ризиків і генетичний алгоритм оптимізації.

Рік виконання кваліфікаційної магістерської роботи 2026.

Рік захисту роботи 2026.

Ключові слова: Data-Driven Management, системний аналіз, підтримка прийняття рішень, теорія прийняття рішень, багатокритеріальна оптимізація, функція корисності, Monte Carlo, Байєсівський аналіз, генетичний алгоритм, управління ризиками, R&D, бізнес-аналітика, Python, Streamlit, моделювання рішень, аналітика даних.

В і д г у к

про кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми « Системний аналіз»

Карєвої А. А.
(прізвище, ініціали)

на тему **«Удосконалення технології прийняття управлінських рішень шляхом впро-ваджения концепції Data-Driven Management»**
(назва теми)

1. Актуальність теми:

Тема кваліфікаційної магістерської роботи є актуальною у зв'язку зі зростанням обсягів даних та необхідністю формалізації процесів прийняття управлінських рішень в українських підприємствах. Використання методів багатокритеріальної оптимізації, імовірнісного моделювання та оцінки ризиків у поєднанні з концепцією Data-Driven Management відповідає сучасним тенденціям розвитку аналітичних та інтелектуальних систем підтримки прийняття рішень.

2. Позитивні риси кваліфікаційної роботи:

Робота характеризується достатнім рівнем теоретичного опрацювання та практичної реалізації математичних методів підтримки прийняття рішень. Автором проведено аналіз підходів теорії прийняття рішень, багатокритеріальної оптимізації та стохастичного моделювання. Позитивною рисою є формалізація процесу вибору альтернатив за допомогою функції корисності, інтеграція Monte Carlo моделювання, Байєсівського оцінювання ризику та евристичних методів оптимізації.

3. Наявність самостійних розробок автора:

Автором самостійно розроблено математичну модель підтримки прийняття рішень, що базується на багатокритеріальній функції корисності та імовірнісних обмеженнях ризику. Реалізовано алгоритми нормалізації критеріїв, оцінювання очікуваної корисності, Monte Carlo симуляції та генетичного алгоритму пошуку оптимальної альтернативи. Також виконано програмну реалізацію моделі засобами Python та Streamlit.

4. Цінність теоретичних висновків та практичних рекомендацій:

Теоретичне значення роботи полягає у поєднанні методів Data-Driven Management та теорії прийняття рішень в межах єдиної концептуальної моделі. Практична цінність полягає у можливості застосування розробленої системи для підтримки багатокритеріального вибору, оцінювання ризиків та аналізу сценаріїв у R&D та Data Science командах. Запропонований підхід може бути використаний для задач оцінки ML-моделей, пріоритезації експериментів та аналізу альтернатив в умовах невизначеності.

5. Наявність недоліків:

До недоліків роботи можна віднести використання припущення про незалежність критеріїв та застосування спрощених імовірнісних розподілів у Monte Carlo моделюванні. Подальшо-

го розвитку потребує інтеграція механізмів автоматичного налаштування ваг критеріїв і врахування кореляцій між параметрами моделі.

6. Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК:

Кваліфікаційна магістерська робота виконана на високому науково-методичному рівні, відповідає вимогам до магістерських робіт за спеціальністю «Системний аналіз» та заслуговує на оцінку «відмінно». Робота рекомендується до захисту перед екзаменаційною комісією.

Науковий керівник д.ф.-м.н., проф. Джалладова І.А._____
(посада, учене звання, науковий ступінь)

_____ _____
(підпис) (прізвище, ініціали)

“15” травня 2026 р.

Рецензія

на кваліфікаційну бакалаврську роботу

здобувачки вищої освіти

Карєвої Анжеліки Андріївни

Тема: **Удосконалення технології прийняття управлінських рішень шляхом впровадження концепції Data-Driven Management**

Актуальність теми кваліфікаційної роботи і доцільність її розроблення
Актуальність теми кваліфікаційної магістерської роботи визначається зростаючою потребою сучасних компаній у використанні data-driven підходів для підтримки управлінських рішень в умовах невизначеності, високої динаміки ринку та швидкого розвитку технологій штучного інтелекту. Особливої ваги ця проблема набуває у R&D-командах, де процес вибору рішень пов'язаний із багатокритеріальною оцінкою альтернатив, ризиками та необхідністю швидкого аналізу великих обсягів даних. Розроблення моделей, що поєднують аналітичні методи, математичний апарат теорії прийняття рішень і сучасні інструменти аналізу даних, є доцільним та має значну практичну цінність для цифрової трансформації управлінських процесів.

Якість проведеного дослідження
Робота характеризується комплексним та системним підходом до дослідження процесу прийняття рішень у середовищі R&D. Авторка здійснила ґрунтовний аналіз існуючих підходів до Data-Driven Management, методів багатокритеріальної оптимізації та моделей оцінки ризиків. Значну увагу приділено формалізації задачі вибору альтернатив, побудові функції корисності та інтеграції імовірнісних методів аналізу, зокрема Monte Carlo моделювання та Байєсівського підходу. Практична частина роботи містить реалізацію програмного прототипу системи підтримки прийняття рішень із використанням Python та Streamlit, що свідчить про високий рівень володіння сучасними аналітичними інструментами та технологіями розробки.

Позитивні риси кваліфікаційної магістерської роботи
Позитивною рисою роботи є поєднання теоретичного системного аналізу з практичною реалізацією прикладної моделі підтримки прийняття рішень. Авторка продемонструвала здатність формалізувати складні управлінські процеси, структурувати критерії оцінки та інтегрувати кількісні методи аналізу в єдину систему. Окремо слід відзначити якісний опис архітектури програмного рішення, реалізацію сценарного аналізу, оцінки ризиків та використання генетичного алгоритму для оптимізації вибору альтернатив. Робота вирізняється логічною структурою, академічним стилем викладення та належним рівнем обґрунтованості висновків. Наукова новизна полягає у запропонованому підході до інтеграції концепції Data-Driven Management із математичними методами теорії прийняття рішень для задач управління в R&D-середовищі.

Зауваження
У цілому робота є змістовною та виконаною на високому рівні. Водночас окремі аспекти моделі могли б бути додатково розширені, зокрема питання врахування залежностей між критеріями, використання більш складних моделей прогнозування та порівняння запропонованого підходу з альтернативними методами оптимізації. Проте зазначені зауваження не

знижують загальної наукової та практичної цінності дослідження й можуть розглядатися як перспективи подальшого розвитку роботи.

Практична значимість висновків і рекомендацій

Практична значимість роботи полягає у можливості використання розробленої моделі для підтримки управлінських рішень у командах, що працюють із даними, аналітикою та машинним навчанням. Запропонований підхід дозволяє підвищити прозорість вибору альтернатив, формалізувати критерії оцінки та забезпечити системне врахування ризиків. Розроблений програмний прототип може бути адаптований для інтеграції з BI-системами, платформами аналітики та внутрішньою інфраструктурою компаній. Особливо цінним є те, що робота базується на практичному досвіді авторки, отриманому під час проходження переддипломної практики у команді R&D, що забезпечує прикладний характер дослідження та відповідність сучасним потребам IT-сфери.

Робота відповідає вимогам, що висуваються до кваліфікаційних магістерських робіт, і може бути рекомендована до захисту та оцінена на високому рівні.

Місце роботи та посада рецензента: ТОВ «Темабіт», власник продуктів ШІ

_____ Вітковська Є.М.
(підпис)

Дата: «15» травня 2026 р.

ЗМІСТ

ВСТУП 3

Розділ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ УДОСКОНАЛЕННЯ ТЕХНОЛОГІЇ ПРИЙНЯТТЯ РІШЕНЬ 9

1.1 Прийняття управлінських рішень як системний процес 9

1.2 Концепція Data-Driven Management у системі сучасного менеджменту 10

1.3 Теорія прийняття рішень як математична основа управління 11

1.4 Синергетична інтеграція DDM і теорії прийняття рішень 20

РОЗДІЛ 2. АНАЛІТИЧНА ОЦІНКА СТАНУ ТЕХНОЛОГІЇ ПРИЙНЯТТЯ РІШЕНЬ (НА ПРИКЛАДІ КОМАНДИ R&D)..... 23

2.1. Характеристика об’єкта дослідження 23

2.2. Аналіз існуючої технології прийняття рішень у R&D-команді .. 28

2.3. Постановка задачі 36

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ DATA-DRIVEN ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ..... 39

3.1 Архітектура моделі 39

3.2 Вимоги до функціоналу моделі 40

3.3 MVP (мінімальний життєздатний продукт) системи 43

ВИСНОВКИ..... 69

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 71

ВСТУП

Управлінське рішення є ключовим механізмом впливу на розвиток організації. Від того, як саме побудована технологія його прийняття, залежить ефективність використання ресурсів, швидкість реагування на зміни та рівень досягнення стратегічних цілей. З позиції системного аналізу прийняття рішення розглядається як елемент цілісної керованої системи, що функціонує на основі інформаційних потоків, зворотних зв’язків і багатокритеріальних цілей.

Розвиток цифрових технологій і аналітичних платформ суттєво змінив інформаційне середовище управління. Організації накопичують великі обсяги даних про бізнес-процеси, клієнтів, фінансові показники та результати експериментів. Проте наявність даних не гарантує якісних рішень. Вирішальним є те, наскільки ці дані інтегровані в саму технологію вибору альтернатив.

Концепція Data-Driven Management передбачає систематичне використання аналітики та кількісних показників для обґрунтування управлінських дій. Водночас теорія прийняття рішень забезпечує математичний інструментарій формалізації альтернатив, оцінювання їхніх наслідків і вибору оптимальної стратегії з урахуванням ризику та невизначеності. Поєднання цих підходів дозволяє перейти від інтуїтивного вибору до формалізованої моделі, де рішення ґрунтується на функції корисності, імовірнісних оцінках і чітко визначених критеріях.

Особливої ваги ця проблема набуває в середовищі R&D-команд, де управлінські рішення пов’язані з вибором моделей машинного навчання, пріоритезацією задач, розподілом обчислювальних ресурсів та оцінюванням експериментів. Такі рішення приймаються в умовах невизначеності, обмежених ресурсів і високої варіативності результатів. За відсутності формалізованого підходу зростає ризик суб’єктивності, неузгодженості та неефективного використання потенціалу команди.

Актуальність теми:

У сучасних умовах управлінські рішення дедалі частіше приймаються в середовищі високої невизначеності, швидких змін та значного обсягу даних, що потребує переходу від інтуїтивних підходів до формалізованих моделей управління. Концепція Data-Driven Management набуває особливого значення завдяки можливості використовувати аналітичні дані як основу для оцінювання альтернатив і прогнозування результатів управлінських дій. Водночас у багатьох організаціях, зокрема в українських IT- та R&D-командах, процес прийняття рішень залишається недостатньо структурованим і значною мірою залежить від суб'єктивного досвіду учасників. Це ускладнює ефективний розподіл ресурсів, пріоритезацію задач та оцінювання результативності рішень. Тому удосконалення технології прийняття управлінських рішень на основі концепції Data-Driven Management і методів системного аналізу є актуальним як для розвитку сучасних управлінських підходів, так і для практичної діяльності українських організацій.

Питання використання аналітики даних у процесах управління розглядається у **працях багатьох зарубіжних і вітчизняних науковців**. Зокрема, G. Shmueli, P. Bruce, M. Stephens та N. Patel у праці Data Mining for Business Analytics досліджують застосування методів data mining і business analytics для підтримки управлінських рішень та аналізу бізнес-процесів [1]. T. Hastie, R. Tibshirani та J. Friedman у роботі The Elements of Statistical Learning сформували теоретичну основу сучасних методів статистичного навчання, прогнозування та обробки даних, які широко використовуються в data-driven системах [5].

У дослідженні G. Cao, Y. Duan та G. Li [3] проаналізовано взаємозв'язок між використанням бізнес-аналітики та ефективністю прийняття управлінських рішень. Автори доводять, що впровадження аналітичних інструментів позитивно впливає на якість управління та швидкість реагування організацій на зміни середовища. J. Yin та V. Fernández у систематичному огляді [6] узагальнили сучасні напрями розвитку business analytics і визначили ключові тенденції інтеграції ана-

літичних платформ у процеси управління підприємствами.

Практичні аспекти впровадження Data-Driven Management у діяльність організацій розглянуто у працях S. Kurpiela та інших авторів [7], де досліджується вплив аналітичних можливостей організації на результати стратегічного планування. У матеріалах Harvard Business School Online [12] наголошується, що data-driven підхід дозволяє підвищити обґрунтованість рішень і знизити рівень управлінської невизначеності. Аналогічні підходи до використання аналітики в управлінні бізнес-процесами наведено у роботах [13], [14], [16].

Теоретичну основу формалізації процесу вибору альтернатив становлять дослідження з теорії прийняття рішень та очікуваної корисності. Зокрема, L. Mao у роботі [2] розглядає концепцію Maximum Expected Utility як основу раціонального вибору альтернатив в умовах ризику. Практичні приклади застосування expected utility theory у прикладних інженерних задачах наведені у дослідженні C. Cappello та співавторів [8]. Байєсівські підходи до обчислення максимальної очікуваної корисності та оптимізації рішень досліджує N. Polson [9]. Окремі аспекти побудови функції корисності та оцінювання ризику описані у праці S. Hanna [4].

Серед українських дослідників увагу привертає робота Н. Гольонко [11], у якій розглянуто трансформацію сучасної управлінської парадигми під впливом концепції Data-Driven Management та цифровізації бізнес-процесів. Також питання використання data-driven підходів у практичному управлінні персоналом і процесами організації висвітлено у публікаціях [16], [17].

Попри значну кількість досліджень у сфері business analytics і теорії прийняття рішень, недостатньо опрацьованим залишається питання інтеграції концепції Data-Driven Management із математичними моделями вибору альтернатив у діяльності R&D-команд. Наявні дослідження переважно зосереджені або на аналітиці даних як окремому управлінському інструменті, або на математичних аспектах теорії прийняття рішень без урахування специфіки data-driven середовища. Це зумовлює необхідність розроблення прикладної моделі, яка поєднувала б методи

системного аналізу, математичного моделювання та концепцію Data-Driven Management у межах єдиної технології підтримки управлінських рішень.

Метою роботи є обґрунтування та розроблення підходу до удосконалення технології прийняття управлінських рішень шляхом впровадження концепції Data-Driven Management із використанням методів системного аналізу та математичного моделювання.

Для досягнення мети передбачено **завдання**:

– розглянути прийняття рішень як системний процес;

– проаналізувати роль Data-Driven Management у формуванні управлінських технологій;

– формалізувати задачу вибору альтернатив у вигляді оптимізаційної моделі;

– здійснити аналітичну оцінку існуючої практики прийняття рішень у R&D-команді;

– запропонувати прикладну модель інтеграції data-driven підходу в управлінський процес.

Об’єктом дослідження є процес прийняття управлінських рішень в організації.

Предметом дослідження є методи вдосконалення технології прийняття рішень на основі концепції Data-Driven Management у межах системного аналізу.

Робота складається з теоретичної, аналітичної та прикладної частин, у яких послідовно розкрито методологічні засади проблеми, проведено аналіз існуючої практики та запропоновано модель її удосконалення.

У роботі використано комплекс загальнонаукових, математичних і прикладних **методів дослідження**. Метод системного аналізу застосовано для розгляду процесу прийняття управлінських рішень як цілісної системи взаємопов’язаних елементів, що функціонує на основі інформаційних потоків, критеріїв оцінювання та механізмів зворотного зв’язку. Методи аналізу та синтезу використано під час

дослідження концепції Data-Driven Management, опрацювання наукових джерел і формування архітектури моделі підтримки прийняття рішень.

Для формалізації процесу вибору альтернатив використано методи математичного моделювання та теорії прийняття рішень. Зокрема, застосовано метод багатокритеріального оцінювання альтернатив із використанням вагових коефіцієнтів, що дозволило врахувати різну важливість критеріїв ефективності, вартості, ризику та затримки виконання.

У роботі використано метод нормалізації даних для приведення різнорідних критеріїв до єдиної шкали оцінювання, що забезпечило можливість коректного порівняння альтернатив. Для визначення інтегральної оцінки альтернатив застосовано метод функції корисності, на основі якого обчислювалася детермінована та очікувана корисність управлінських рішень.

Для оцінювання невизначеності та варіативності результатів використано метод статистичного моделювання Монте-Карло, який дозволив здійснити серію симуляцій із випадковими змінами параметрів і визначити очікувану корисність альтернатив, а також імовірність досягнення заданого рівня ефективності.

Оцінювання ризику виконано із застосуванням байєсівського підходу та методу Bayesian Risk Assessment, що дало змогу оновлювати ймовірність успішності альтернатив на основі отриманих результатів симуляції та враховувати невизначеність у процесі прийняття рішень.

Для пошуку оптимальної альтернативи використано метод оптимізації за критерієм максимальної очікуваної корисності, а також евристичний метод генетичного алгоритму, який застосовувався для аналізу можливих рішень у складному просторі альтернатив та вибору найбільш доцільного варіанту з урахуванням ризикових обмежень.

Теоретична значущість роботи полягає в узагальненні підходів до прийняття управлінських рішень у межах системного аналізу та розвитку концепції Data-Driven Management шляхом її поєднання з математичними моделями теорії

прийняття рішень. У роботі уточнено роль функції корисності, критеріїв оцінювання та кількісного аналізу даних у процесі вибору управлінських альтернатив.

Методична значущість полягає у формуванні підходу до побудови моделі підтримки прийняття рішень, яка може бути використана для формалізації управлінських процесів у командах, що працюють із даними та аналітичними системами. Запропонована структура моделі дозволяє поєднати data-driven підхід, багатокритеріальне оцінювання та елементи математичного моделювання в єдиній системі підтримки рішень.

Практична значущість результатів полягає у можливості використання запропонованої моделі для підвищення обґрунтованості управлінських рішень у R&D-командах. Розроблений MVP системи може бути адаптований для оцінювання альтернатив, пріоритезації задач, аналізу ризиків та підтримки процесу розподілу ресурсів у діяльності організацій.

Інформаційну базу дослідження становлять праці вітчизняних і зарубіжних науковців з питань системного аналізу, теорії прийняття рішень, Data-Driven Management, business analytics та математичного моделювання. У роботі використано матеріали наукових статей, монографій, систематичних оглядів, конференційних публікацій і професійних аналітичних ресурсів.

Також використано дані офіційних вебресурсів, матеріали професійних блогів і платформ, присвячених business analytics та data-driven управлінню, зокрема Harvard Business School Online, DataCamp, TechnologyAdvice, GraphAware та інших. Практична частина дослідження ґрунтується на узагальнених даних щодо процесів прийняття рішень у R&D-команді, результатах аналізу управлінських сценаріїв і прикладному моделюванні процесу вибору альтернатив.

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ЗАСАДИ УДОСКОНАЛЕННЯ ТЕХНОЛОГІЇ ПРИЙНЯТТЯ РІШЕНЬ

1.1 Прийняття управлінських рішень як системний процес

Прийняття управлінського рішення є центральним механізмом впливу на діяльність організації. Воно не зводиться до одноразового вибору між кількома варіантами, а виступає складовою безперервного процесу управління, що поєднує аналіз інформації, формування альтернатив, оцінювання наслідків і контроль результатів.

Організація функціонує як відкрита система, яка взаємодіє із зовнішнім середовищем, використовує ресурси та досягає визначених цілей. У цьому контексті рішення виконує роль регулятора: воно спрямовує ресурси, змінює параметри процесів і впливає на майбутній стан системи. Кожне управлінське рішення має вхідні дані (показники діяльності, інформацію про ризики, обмеження), процедуру обробки (аналіз, моделювання, порівняння альтернатив) та вихід – конкретну дію або стратегію. Після реалізації формується зворотний зв’язок у вигляді фактичних результатів, що дозволяє оцінити правильність вибору.

Процес прийняття рішення зазвичай включає кілька взаємопов’язаних етапів:

– виявлення проблемної ситуації або можливості;
– формування множини допустимих альтернатив;
– прогнозування наслідків кожної альтернативи;
– вибір варіанта за визначеним критерієм;
– аналіз результатів і коригування дій.

Складність цього процесу визначається багатокритеріальністю та невизна-

ченістю. Управлінське рішення впливає одночасно на фінансові показники, операційну ефективність, ризики та стратегічні позиції. Частина параметрів має імовірнісний характер, тому вибір здійснюється за неповної інформації. Це зумовлює необхідність формалізації критеріїв і використання кількісних методів оцінювання.

Традиційна практика часто спирається на експертний досвід і інтуїцію. Такий підхід може бути швидким, однак він ускладнює перевірку та відтворюваність рішень. Формалізований процес дозволяє структурувати проблему, чітко визначити критерії оцінювання та забезпечити логічну обґрунтованість вибору.

У загальному вигляді управлінське рішення можна представити як задачу вибору оптимальної альтернативи на множині допустимих варіантів за наявності обмежень і цільової функції. Це створює основу для інтеграції аналітики, моделей прогнозування та оптимізаційних методів.

1.2 Концепція Data-Driven Management у системі сучасного менеджменту

Data-Driven Management (DDM) розглядається як підхід, за якого рішення ухвалюються на основі систематично зібраних даних, аналітики та об’єктивних показників, а не лише на інтуїції керівників. За словами Shmueli, G., Bruce, P. C., Stephens, M. L., & Patel, N. R. (2017), DDM дозволяє компаніям «перетворювати дані на цінну інформацію та знання, що безпосередньо впливають на ефективність управління» [1]. У той же час теорія прийняття рішень (ТПР) забезпечує формальну основу для обґрунтування вибору між альтернативами, визначення оптимальних стратегій і прогнозування наслідків кожного рішення (Cao, G., Duan, Y., & Li, G., 2022) [3].

Однією з ключових переваг такого підходу є підвищення обґрунтованості

управлінських рішень. Аналіз тенденцій, виявлення закономірностей і моделювання сценаріїв дозволяють зменшити рівень невизначеності. Це особливо важливо в умовах швидких змін ринку та високої конкуренції.

Data-Driven Management також змінює роль керівника. Його завдання полягає не лише у прийнятті остаточного рішення, а й у формуванні запитів до даних, визначенні релевантних показників ефективності та інтерпретації результатів аналітики. Водночас аналітичні інструменти не замінюють управлінський досвід – вони доповнюють його.

Впровадження цього підходу потребує відповідної інфраструктури: систем збору та зберігання даних, аналітичних платформ, інструментів візуалізації. Не менш важливою є організаційна культура – готовність працівників працювати з показниками, відкритість до перевірки гіпотез і постійного вдосконалення процесів.

Разом із перевагами існують і обмеження. Дані можуть бути неповними або викривленими, їх інтерпретація – помилковою. Надмірна орієнтація на кількісні показники іноді ігнорує якісні аспекти, зокрема мотивацію персоналу чи репутаційні ризики. Тому ефективне управління передбачає поєднання аналітики з професійним судженням.

Крім того, ефективність DDM безпосередньо залежить від якості даних, а помилки збору, неповнота інформації чи викривлення вибірки можуть призводити до хибних управлінських висновків. Це свідчить про необхідність поєднання data-driven підходу з методами формального оцінювання ризику та підтримки прийняття рішень.

1.3 Теорія прийняття рішень як математична основа управління

Теорія прийняття рішень сформувалася на перетині математики, економіки та теорії ймовірностей і стала інструментом формалізації управлінських виборів. Її головна ідея полягає в тому, що будь-яке рішення можна подати у вигляді моделі: є набір альтернатив, можливі стани середовища та наслідки, пов’язані з кожною комбінацією.

Базовими елементами теорії є альтернативи, критерії оцінки, обмеження та ризики. Формалізація дозволяє описати ситуацію через математичні залежності й обрати варіант, що максимізує вигоду або мінімізує втрати. У найпростішому випадку застосовуються критерії оптимальності – максимізація прибутку, мінімізація витрат чи часу виконання.

В умовах визначеності рішення приймаються за наявності повної інформації про наслідки кожної альтернативи. Коли ж середовище є невизначеним або ризиковим, застосовуються ймовірнісні моделі. Тут використовуються очікувана корисність, критерії Вальда, Севіджа, Байєса та інші підходи, що враховують можливі сценарії розвитку подій.

Важливим напрямом є багатокритеріальна оптимізація, коли рішення оцінюється не за одним, а за кількома показниками. У таких випадках формується система ваг або пріоритетів, що відображає стратегічні цілі організації. Це дає змогу узгодити фінансові, соціальні та операційні аспекти діяльності.

Теорія прийняття рішень також лежить в основі методів лінійного програмування, теорії ігор, мережевого планування та імітаційного моделювання. Завдяки цим інструментам керівник може аналізувати складні виробничі, логістичні чи інвестиційні задачі з урахуванням обмежених ресурсів.

Попри формальність моделей, їх застосування не зводиться до механічних розрахунків. Будь-яка модель є спрощенням реальності, а її результат залежить від припущень і якості вихідних даних. Тому математичні методи виступають інструментом підтримки рішень, а не повною заміною управлінського мислення.

Розглянемо кейс практичного застосування. Компанія, що займається логі-

стикою, отримує дані про завантаження транспортних маршрутів, витрати на доставку та рівень обслуговування клієнтів. На етапі оцінки альтернатив TPR дозволяє сформалізувати функцію корисності $U(a_i)$, яка включає прибутковість, ризик затримок та інші критерії. Вибір оптимальної стратегії відбувається через оператор argmax $U(a_i)$, що формально обґрунтовує управлінське рішення.

Багатокритеріальна функція корисності

Більш того, інтеграція DDM і ТПР дозволяє враховувати множинні критерії одночасно. Класична функція корисності може бути представлена у вигляді зваженої суми (часто її використовують для багатокритеріальних задач) за формулою:

$$U(a_i) = \sum_{j=1}^{m} w_j u_j(a_i),$$

де $u_j(a_i)$ – нормалізовані показники ефективності,

w_j – їхні ваги, що відображають пріоритети компанії.

Такий підхід дозволяє менеджерам балансувати між прибутком, ризиком та іншими стратегічними цілями.

Як ілюстрацію поведінки функції корисності у ситуаціях невизначеності, автори деяких досліджень подають графік, на якому показано саму криву корисності, очікуваний рівень корисності та пов’язану з ним певну «гарантовану» виплату (рис. 1.1). Аналіз таких графічних представлень дозволяє не лише побачити форму функції $U(x)$, а й зрозуміти, чому алгоритм argmax $U(a_i)$ обирає саме ту альтернативу, яка забезпечує найвище значення корисності (тобто оптимальний вплив на цільові показники) при заданих ймовірностях виходів.

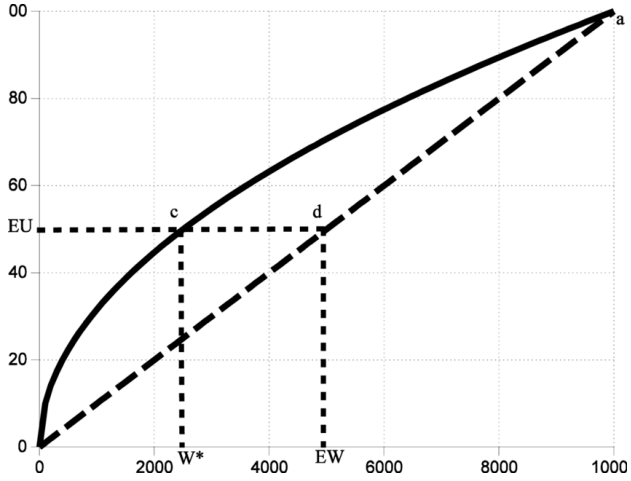


Рисунок 1.1. Функція корисності $U = \sqrt{Wealth}$, очікуване багатство (EW) та очікувана корисність (EU) за умови ймовірності 50 % нульового доходу та 50 % доходу в розмірі $10 000.

Джерело: ResearchGate, [4]

Як зазначає Mao, L. (2021) [2], використання максимізації очікуваної корисності дозволяє не лише формально оцінити варіанти рішень, а й проводити сценарне моделювання. Наприклад, у випадку невизначеного попиту можна розраховувати ймовірності різних станів і передбачати, як зміна попиту або витрат вплине на вибір оптимальної альтернативи. Це робить процес прийняття рішень прозорим, передбачуваним та адаптивним до змін у бізнес-середовищі.

Імовірнісні моделі та оцінка ризику

Реальні бізнес-рішення рідко реалізуються в детермінованому середовищі. Значна частина управлінських задач пов’язана з невизначеністю, що стосується попиту, фінансових ризиків, операційних збоїв або поведінки клієнтів. У таких умовах доцільним є використання принципу **очікуваної корисності**, відповідно до якого оптимальною вважається дія, що максимізує математичне сподівання корисності, за формулою:

$$a^* = arg \max_{ai \in A} \sum_{s \in S} p(s)\, U(a_i, s),$$

де S – множина можливих станів середовища,

$p(s)$ – ймовірність реалізації стану s.

Data-Driven Management забезпечує емпіричну оцінку $p(s)$ на основі істо-

ричних даних або моделей прогнозування, що дозволяє інтегрувати математичні рішення у практичний процес управління.

Для врахування ризику вводиться порогове правило за формулою:

$$a_i \in A \text{ приймається, якщо } P(U(a_i) \geq U_{min}) \geq \alpha,$$

де U_{min} *– мінімально допустимий рівень корисності,*

$\alpha \in (0,1)$ *– заданий поріг ймовірності.*

Така постановка поєднує оптимізацію з імовірнісними обмеженнями, що зближує модель з задачами chance-constrained optimization. Для бізнесу це означає, що рішення обирається не лише як найефективніше в середньому, а й як достатньо надійне з позиції допустимого ризику.

Наприклад, компанія може обрати альтернативу з дещо меншою очікуваною корисністю, але з імовірністю не нижче 0,9 досягти заданого рівня прибутковості. У цьому випадку Data-Driven Management забезпечує оцінку розподілу результатів, а теорія прийняття рішень – формальний критерій вибору.

Методи оптимізації

Для знаходження оптимальної альтернативи застосовуються різні алгоритмічні підходи (табл. 1.1):

Таблиця 1.1. Методи оптимізації

Метод	**Призначення**	**Особливості**
Argmax	Класична максимізація функції корисності	Проста, інтуїтивно зрозуміла, підходить для детермінованих умов
Евристики	Пошук наближеного оптимуму	Використовуються при великій кількості альтернатив або складній функції корисності
Monte Carlo	Імітаційне моделювання ймовірнісних сценаріїв	Дозволяє враховувати ризики та невизначеність, оцінювати розподіл результатів

Джерело: розроблено автором на основі [9].

Табл. 1.1 дає змогу зробити висновок, що вибір методу оптимізації залежить від рівня складності задачі, структури множини альтернатив та ступеня невизначеності середовища. Для детермінованих умов доцільним є використання класичних методів максимізації, тоді як у задачах із ризиками та багатофакторністю більш ефективними стають евристичні та імовірнісні підходи, зокрема Monte

Carlo моделювання.

Якщо альтернативи описуються набором параметрів, множина A може бути інтерпретована як область у просторі R^n. Тоді задача прийняття рішення зводиться до пошуку глобального максимуму функції корисності на допустимій області за формулою:

$$\max_{x \in A} U(x), \qquad \text{за умови } P\left(U(x) \geq U_{min}\right) \geq \alpha$$

де x - вектор параметрів (альтернатива), що описує можливе рішення;

A - множина допустимих альтернатив;

U(x) - функція корисності (інтегральна оцінка ефективності альтернативи);

Umin - мінімально допустимий рівень корисності;

P(·) - ймовірність виконання умови;

α - заданий рівень довіри (допустима ймовірність прийняття ризику).

Це дозволяє застосовувати апарат математичного аналізу, методи оптимізації та чисельного моделювання. Для дослідників-математиків особливий інтерес становить аналіз чутливості оптимального рішення до змін параметрів моделі – зокрема ймовірнісного порогу α, що безпосередньо пов'язано з управлінням ризиками.

Поєднання Data-Driven Management з формальними моделями ТПР забезпечує:

– прозорість і відтворюваність рішень;
– можливість адаптивного реагування на зміни даних та сценаріїв;
– баланс між ефективністю та ризиком;
– інтеграцію емпіричних даних у формальні критерії вибору.

Математична модель стає інструментом прийняття рішень, де алгоритмічні методи та багатокритеріальні функції корисності забезпечують стратегічний компроміс між прибутковістю, ризиком і надійністю управлінських дій.

Генетичні алгоритми в задачах вибору альтернатив

У задачах, де кількість можливих альтернатив є надто великою або струк-

тура функції корисності має складний характер, класичні методи оптимізації можуть бути недостатньо ефективними. У таких випадках доцільним є використання генетичних алгоритмів, які належать до евристичних методів оптимізації та імітують принципи природного добору.

Ідея методу полягає у формуванні початкової множини рішень (популяції), для яких обчислюється значення функції корисності. Надалі альтернативи з кращими результатами мають вищу ймовірність «успадкування» своїх характеристик у наступних поколіннях. За допомогою операцій схрещування та мутації формуються нові варіанти рішень, що дозволяє поступово знаходити більш оптимальні комбінації параметрів.

У контексті Data-Driven Management генетичний алгоритм може використовуватися для вибору управлінської стратегії за наявності великої кількості критеріїв, обмежень та сценаріїв розвитку подій. Наприклад, в R&D-команді алгоритм може допомагати знаходити оптимальний розподіл ресурсів між експериментами з урахуванням очікуваної ефективності, ризику невдачі та часових обмежень.

Формально задача оптимізації може бути представлена у вигляді формули:

$$a^* = arg \max_{a^i \in A} U(a^i),$$

де a∗ - оптимальна альтернатива (найкраще рішення);

ai - і-та альтернатива з множини допустимих рішень;

A - множина всіх допустимих альтернатив;

U(ai) - функція корисності, що відображає інтегральну ефективність альтернативи за всіма критеріями;

argmax - оператор вибору альтернативи, для якої функція корисності досягає максимального значення.

Множина альтернатив A досліджується не повним перебором, а через ітеративний пошук найбільш придатних рішень.

Перевагою генетичних алгоритмів є здатність працювати з нелінійними

функціями корисності та складними просторами пошуку, де традиційні методи можуть втрачати ефективність.

Байєсівський підхід до оцінювання ризику

У практичних задачах управління оцінка ризику часто змінюється після отримання нових даних. Саме тому в Data-Driven Management важливе місце займає байєсівський підхід, який дозволяє оновлювати ймовірність подій у процесі накопичення інформації.

На відміну від статичних моделей, байєсівський підхід враховує як попередні припущення, так і нові результати спостережень. Це особливо корисно для R&D-середовища, де управлінські рішення приймаються в умовах високої невизначеності, а результати експериментів можуть суттєво змінювати оцінку ефективності альтернатив.

Базою такого підходу є формула Байєса:

$$P\ (A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

де P(A) - *апріорна ймовірність події;*

$P(B \mid A)$ - *імовірність отримання нових даних за умови настання події;*

$P(A \mid B)$ - *оновлена (апостеріорна) ймовірність після врахування нової інформації.*

У межах даної роботи байєсівське оцінювання використовується для уточнення ймовірності успішності альтернатив після проведення симуляцій Монте-Карло. Це дозволяє не лише оцінювати середню ефективність рішень, а й враховувати рівень впевненості у прогнозованому результаті.

Нормалізація критеріїв у багатокритеріальних моделях

Оскільки критерії оцінювання альтернатив можуть мати різні одиниці виміру та масштаб значень, перед розрахунком функції корисності виникає необхідність їх нормалізації. Без цього показники з більшими числовими значеннями можуть непропорційно впливати на результат моделі.

Для критеріїв типу Benefit, де більше значення є кращим, використовуєть-

ся нормалізація за формулою:

$$u_j(a_i) = \frac{x_{ij} - \min x_j}{\max x_j - \min x_j}$$

де uj(ai) - нормалізоване значення j-го критерію для альтернативи ai;

x_{ij} - вихідне значення j-го критерію для альтернативи ai;

xj^{min} - мінімальне значення j-го критерію серед усіх альтернатив;

xj^{max} - максимальне значення j-го критерію серед усіх альтернатив;

нормалізація приводить різнорозмірні критерії до уніфікованого діапазону [0;1] для подальшого агрегування в функції корисності.

Для критеріїв типу Cost, де менше значення є бажанішим, формула:

$$u_j(a_i) = \frac{\max x_j - x_{ij}}{\max x_j - \min x_j}$$

Такий підхід дозволяє привести всі критерії до єдиної шкали в інтервалі від 0 до 1 та забезпечити коректне багатокритеріальне порівняння альтернатив.

Імітаційне моделювання Monte Carlo

У процесі прийняття рішень значна частина параметрів має невизначений характер. Наприклад, майбутній попит, тривалість виконання задач або фінансові витрати можуть змінюватися залежно від зовнішніх умов. Для врахування такої невизначеності використовується метод Monte Carlo.

Суть підходу полягає у багаторазовому генеруванні випадкових сценаріїв на основі заданих імовірнісних розподілів параметрів. Для кожного сценарію окремо обчислюється функція корисності, після чого формується розподіл можливих результатів.

Очікувана корисність визначається як середнє значення результатів симуляцій за формулою:

$$EU(a_i) = \frac{1}{N}\sum_{k=1}^{N} U_k(a_i)$$

де N - кількість симуляцій;

$U_k(a_i)$ - значення корисності в k-му сценарії.

Перевагою Monte Carlo є можливість оцінити не лише середній результат, а й варіативність, ризик та ймовірність несприятливих сценаріїв. У задачах управління це дозволяє переходити від статичного оцінювання до аналізу поведінки системи в умовах невизначеності.

Важливим етапом оцінювання якості математичної моделі є аналіз чутливості. Його мета полягає у визначенні того, наскільки зміна окремих параметрів впливає на кінцевий результат прийняття рішення.

У багатокритеріальних задачах це особливо важливо, оскільки зміна ваг критеріїв може змінити пріоритетність альтернатив. Наприклад, збільшення ваги ризику може зробити менш привабливою альтернативу з високою прибутковістю, але нестабільними результатами.

Формально аналіз чутливості можна розглядати як оцінювання зміни функції корисності за формулою:

$$\frac{\partial U}{\partial \omega_j}$$

де w_j - вага відповідного критерію.

Практичне значення такого аналізу полягає у можливості оцінити стійкість управлінського рішення до змін зовнішніх умов, пріоритетів або вхідних даних. Це дозволяє підвищити надійність моделі та зменшити ризик помилкового вибору альтернативи.

1.4 Синергетична інтеграція DDM і теорії прийняття рішень

Поєднання Data-Driven Management (DDM) і теорії прийняття рішень формує цілісну основу для управління, де дані та математичні моделі взаємодопов-

нюють одне одного. Основна ідея синергії DDM і ТПР полягає в тому, що дані стають основою для кількісного оцінювання альтернатив, а формальні моделі ТПР забезпечують логіку і обґрунтованість вибору. Іншими словами, DDM відповідає за емпіричну складову («що відбувається?»), тоді як ТПР забезпечує аналітичну структуру для відповіді на питання «що робити і чому це оптимально?». Така інтеграція дозволяє уникнути двох ключових проблем сучасного менеджменту: суб'єктивності та фрагментарності інформації.

У класичній ТПР розрізняють чотири етапи прийняття рішення:

– Ідентифікація проблеми – визначення ситуації, яка потребує втручання.

– Формулювання альтернатив – генерація можливих стратегій дій.

– Оцінка наслідків – аналіз потенційних результатів кожної альтернативи, включаючи фінансові, операційні та ризикові аспекти.

– Вибір оптимальної альтернативи – застосування критерію максимізації корисності або іншого формалізованого методу для обрання найкращого варіанту.

DDM на кожному з цих етапів інтегрує інструменти збору і обробки даних: BI-системи, Big Data-платформи, моделі машинного навчання, симуляційні моделі та алгоритми прогнозування. Наприклад, при ідентифікації проблеми система може автоматично виявляти відхилення ключових показників KPI або попереджати про потенційні затримки в ланцюзі постачання. Паралельно, при формулюванні альтернатив алгоритми кластерного аналізу чи генетичні алгоритми можуть генерувати оптимальні варіанти рішень, враховуючи історичні дані та тренди ринку.

Така інтеграція сприяє переходу від реактивного управління до проактивного, коли рішення ґрунтуються не лише на фіксації поточних показників, а й на прогнозуванні майбутніх змін та оцінці альтернатив розвитку.

У межах даної роботи інтеграція DDM і теорії прийняття рішень реалізу-

ється через побудову моделі підтримки прийняття рішень, яка поєднує багатокритеріальне оцінювання альтернатив, розрахунок функції корисності, оцінювання ризику та механізми оптимізаційного вибору. Це дозволяє формалізувати процес прийняття рішень у середовищі R&D-команди та зменшити залежність від суб’єктивних оцінок.

РОЗДІЛ 2. АНАЛІТИЧНА ОЦІНКА СТАНУ ТЕХНОЛОГІЇ ПРИЙНЯТТЯ РІШЕНЬ (НА ПРИКЛАДІ КОМАНДИ R&D)

2.1. Характеристика об’єкта дослідження

Об’єктом дослідження є процес прийняття управлінських рішень у команді R&D, що займається розробкою та вдосконаленням моделей обробки природної мови. Специфіка такої команди полягає у високому рівні невизначеності, експериментальному характері роботи та швидкій зміні технологічних підходів.

Ключові ролі, що займаються оцінкою ризику на проєкті та прийняттям рішень, тут є продуктові менеджери (Product Managers) та бізнес-аналітики (Business Analysts).

Роль продуктового менеджера полягає у визначенні стратегічного напряму розвитку продукту, формуванні пріоритетів та оцінюванні бізнес-цінності запропонованих рішень. Product Manager приймає рішення щодо доцільності запуску нових функцій, пріоритезації експериментів, розподілу ресурсів між задачами та вибору компромісу між швидкістю реалізації, якістю продукту і витратами. У процесі оцінювання ризиків він аналізує можливий вплив технічних або організаційних рішень на бізнес-показники, користувацький досвід і стратегічні цілі компанії.

Роль бізнес-аналітика (BA) у цьому середовищі виходить за межі класичного збору вимог. BA виступає посередником між бізнес-цілями та технічними можливостями команди. Він формалізує запити замовників, трансформує їх у вимірювані критерії успіху, визначає метрики оцінки моделей та структурує вимоги до даних.

Проблеми з прийняттям рішення існують на кожному етапі життєвого ци-

клу продукту та доволі часто превентивно вирішуються детально описаною проєктною документацією. Список документів є затвердженим та переважно сталим, його представлено нижче:

1. Продуктові та бізнес-документи
    1.1. User Stories
        • Класичні (“Як користувач… Я хочу… Щоб…”)
        • INVEST-перевірка
        • Розбиття на підсторі (pipeline, інтеграції тощо)
    1.2. Acceptance Criteria
        • Pre-Conditions (передумови)
        • Conditions (функціональні (FR) / нефункціональні (NFR) вимоги)
        • Post-Conditions (післяумови)
    1.3. Бізнес-вимоги/Product Requirements
        • Опис функціоналу
        • Очікувана цінність для користувача
        • Use cases (сценарії використання)
2. Аналітична та системна документація
    2.1. Functional Requirements (FR)
        • Що саме система повинна робити (наприклад: зчитувати скани, витягувати дані, відображати в системі замовника)
    2.2. Non-Functional Requirements (NFR)
        • Продуктивність (час обробки)
        • Надійність (обробка помилок)
        • Безпека (доступи, ролі)
        • Масштабованість
    2.3. Data Requirements
        • Формати даних

- Джерела (скани, аудіо, API)
- Валідація та мапінг

3. Архітектурні документи

3.1. UML-діаграми

- Activity diagrams (алгоритми обробки)
- Sequence diagrams (взаємодія сервісів)
- Component diagrams (архітектура системи)

3.2. Опис пайплайнів

- Preprocessing pipeline
- OCR + LLM extraction
- Data mapping & validation
- Integration flow

3.3. Інтеграційна документація

- API (вхід/вихід)
- Інтеграції (служба підтримки, аудіосервіси)
- Формати обміну даними

4. Data/AI документація

4.1. NLP/LLM design

- Інтенти (працівники, новини, коментарі)
- Логіка обробки тексту
- Використання токенів (і їх вартість)

4.2. OCR/Voice processing

- Джерела (скани, аудіо з носія даних)
- Алгоритм обробки
- Якість розпізнавання

4.3. Data Science задачі

- Підготовка даних
- Моделі

- Оцінка якості

5. Операційна та технічна документація
    5.1. Dev/DevOps задачі
        - CI/CD
        - Інфраструктура (SRE/Platform)
        - Деплой сервісів
    5.2. QA документація
        - Тест-кейси
        - Acceptance testing
        - Edge cases (дублі, пропущені записи)
    5.3. Моніторинг та підтримка
        - Логи
        - Алерти
        - Метрики
6. Документи для роботи з даними
    6.1. Data Quality/Data Issues
        - Дублікати
        - Відсутні записи в джерелі
        - Розсинхронізація
    6.2. Унікальні ідентифікатори
        - Логіка генерації ID
        - Причини відмови від певних підходів
7. Інструкції та гайдлайни
    7.1. User Guides
        - Інструкції до користування системою
        - Робота з системою
    7.2. Внутрішні гайдлайни
        - Як писати user stories

• Як працювати з NLP/LLM

• Як обробляти дані

8. Аналітичні панелі (BI/Dashboard docs)

8.1. Опис панелі

• Мета (модерація, аналітика)

• Користувачі (ролі, доступи)

8.2. Метрики

• Кількість записів

• Дублікати

• Якість обробки

Бізнес-аналітик формує ключову частину рішень саме на етапі документації. Перелік шаблонів документів, де бізнес-аналітик є автором, представлено в Додатку 1. Однак часто бувають проблеми, які потребують швидшого аналізу та більшої формалізованості. У таких випадках є сенс розглядати математичні підходи до прийняття рішень та їх впровадження у робочу діяльність менеджерів та аналітиків.

У межах R&D-команди приймаються як стратегічні, так і операційні рішення. До типових належать:

- Пріоритезація функціональності (фіч) – визначення, які можливості продукту розробляти першочергово з урахуванням бізнес-цінності, складності реалізації та ресурсних обмежень.
- Вибір моделей або архітектур – порівняння альтернативних підходів (fine-tuning існуючих моделей, використання open-source рішень, розробка власних моделей) за показниками якості, вартості та масштабованості.
- Оцінка результатів експериментів – прийняття рішення про впровадження, доопрацювання або відмову від певної гіпотези на основі статистично значущих показників.
- Баланс між якістю та витратами – визначення прийнятного компромі-

су між точністю моделі та обчислювальними ресурсами.

Такі рішення часто приймаються в умовах неповної інформації та обмеженого часу, що підвищує значення структурованого аналізу та чітких критеріїв оцінки.

Основними джерелами даних у R&D-команді є:

– результати експериментів (метрики якості моделей);
– лог-файли використання продукту;
– дані користувацької поведінки;
– бізнес-показники (конверсія, retention, LTV);
– внутрішні технічні метрики (час відповіді, навантаження, стабільність системи).

Аналітична інфраструктура зазвичай включає системи збору та зберігання даних (data warehouse, data lake), інструменти обробки (Python, SQL, ML-фреймворки), платформи візуалізації та моніторингу. Важливою складовою є автоматизація експериментів і ведення історії моделей, що забезпечує відтворюваність результатів.

2.2. Аналіз існуючої технології прийняття рішень у R&D-команді

У сучасних R&D-командах процес прийняття рішень поступово трансформується у напрямі більшої формалізації та використання data-driven підходів. Для кращого розуміння цієї динаміки нижче наведено узагальнення ключових тенденцій розвитку у період 2022–2025 років (табл. 2.1).

Таблиця 2.1. Тенденції розвитку R&D-команд та data-driven підходів у 2022–2025 рр.

Тенденція	Прояв у R&D-командах	Джерело
Збільшення використання LLM	У 2023–2025 роках великі мовні моделі (LLM) стали базовим інструментом для NLP, автоматизації аналітики, генерації коду та підтримки прийняття рішень. R&D-команди переходять від експериментального використання до інтеграції моделей у внутрішні сервіси та бізнес-процеси.	[18]
Зростання кількості експериментів	Команди дедалі частіше працюють у форматі безперервних експериментів: тестують нові моделі, промпти, архітектури та сценарії використання. Це підвищує навантаження на процеси оцінювання та пріоритезації альтернатив.	[19]
Збільшення витрат на inference	Після переходу від пілотних рішень до production-систем основні витрати зміщуються з навчання моделей на inference-обчислення. Це особливо помітно при використанні LLM та RAG-архітектур.	[20], [21]
Перехід до Data-Driven Management	Організації поступово відмовляються від інтуїтивного управління на користь рішень, заснованих на даних, KPI та аналітичних моделях. Це особливо характерно для технологічних і R&D-команд.	[22]
Зростання ролі BI та MLOps	Поширення AI-рішень стимулює розвиток BI-систем, MLOps-платформ, моніторингу моделей та автоматизації пайплайнів. Відтворюваність експериментів і контроль якості моделей стають критично важливими.	[23]
Потреба у швидшому прийнятті рішень	Через швидку зміну ринку та технологій R&D-команди змушені скорочувати цикл прийняття рішень. Це підвищує попит на автоматизовану аналітику та системи підтримки рішень.	[18]

Джерело: розроблено автором на основі [18–23].

З таблиці 2.1 видно, що у 2022–2025 роках спостерігається активний перехід R&D-команд до data-driven підходів, широкого використання LLM та автоматизованої аналітики. Одночасно зростає складність управлінських рішень: збільшується кількість експериментів, витрати на inference-обчислення та потреба у швидкій оцінці альтернатив. Це підтверджує актуальність розробки формалізованих моделей підтримки прийняття рішень, які поєднують аналітичні дані, оцінку ризику та математичні методи оптимізації.

У команді R&D, що розглядається у поточному дослідження, управлінські рішення приймаються здебільшого на основі комбінації експертної оцінки та часткових даних (табл.1). Типовий сценарій виглядає так:

1. Ініціація (“навіщо взагалі це робимо”)

Артефакти:

- Product Vision/Problem Statement
- High-level User Stories (чернетки)
- Бізнес-вимоги (високорівневі)
- Stakeholders list (категорії стейкхолдерів)
- Use cases (на рівні ідеї)
- Value proposition

Результат фази:

є розуміння проблеми + чи варто це робити

2. Дослідження (Discovery) (“як це може працювати”)

Артефакти:

- Деталізовані User Stories
- INVEST-аналіз сторі
- Domain analysis (дані, джерела, обмеження)
- Data Requirements (джерела: скани, аудіо, API)
- Technical exploration:
    - OCR підходи
    - LLM/NLP можливості
    - Voice-to-text
- Spike tasks (дослідницькі)
- High-level architecture (чернетка)
- Ризики (наприклад: дублікати, відсутні записи)

Результат:

Зрозуміло, чи це реально зробити і як приблизно

3. Планування (“як саме будемо будувати”)

Артефакти:

- Повноцінні User Stories (готові до розробки)
- Acceptance Criteria:
    - Pre-Conditions

- Conditions (FR/NFR)
- Post-Conditions

- Functional Requirements (FR)
- Non-Functional Requirements (NFR)
- Архітектура:
  - UML (component, sequence, activity)
- Декомпозиція:
  - Preprocessing pipeline
  - OCR + extraction
  - Mapping & validation
  - Integration
- Data contracts/API specs
- План спринтів/roadmap
- Оцінка задач

Результат:

є чіткий план що і як робимо

4. Імплементація - Вимоги (Analysis/Design in-flight)

Артефакти:

- Уточнення User Stories (iteration)
- Деталізація Acceptance Criteria
- Data mapping rules
- Логіка унікальних ID
- Правила обробки:
  - дублікати
  - відсутні записи
- NLP design:
  - інтенти
  - поля (email, телефон і т.д.)

- Prompt/LLM interaction design
- Edge cases

Результат:

розробка отримує конкретні, не двозначні вимоги

5. Імплементація - Розробка/Тестування

Артефакти:

Для DEV

- Код
- API інтеграції
- Pipeline реалізації
- Інтеграція з іншими системами

Для Data/NLP

- Моделі/промпти
- Обробка OCR/аудіо
- Evaluation results

Для QA

- Test cases
- Test scenarios
- Edge case tests (дублі, пропуски)
- Acceptance testing

Для DevOps

- CI/CD pipeline
- Інфраструктура (SRE/Platform)
- Deployment configs

Результат:

працюючий функціонал + підтверджена якість

6. Впровадження (Release)

Артефакти:

- Release notes
- Deployment plan
- Migration scripts (якщо є)
    - User guides/інструкції
- Доступи/ролі (для панелей, модерації)
- Навчальні матеріали

Результат:

система доступна користувачам

7. Підтримка (Support/Operations) (“життя після релізу”)

Артефакти:

- Моніторинг:
    - логи
    - алерти
    - метрики
- Аналітичні панелі:
    - якість даних
    - дублікати
    - обробка
- Data Quality reports
- Інциденти/баг-репорти
- Оптимізації:
    - продуктивність
    - токени LLM (особливо вартість!)
- Backlog нових задач

Результат:

система стабільно працює і покращується

Для аналізу балансу між формалізованими та експертними підходами у процесі прийняття рішень нижче наведено узагальнене порівняння ключових

компонентів (табл 2.2).

Таблиця 2.2. Спостереження щодо ролі інтуїції та метрик

Компонент	**Характеристика**	**Переваги**	**Недоліки**
Інтуїція/експертний досвід	Рішення часто ухвалюються на основі знань і досвіду учасників	Швидкість, врахування контексту, нестандартних факторів	Суб'єктивність, ризик упередженості, важко відтворювати
Метрики та аналітика	Використовуються експериментальні показники (F1, latency, cost)	Об'єктивність, можливість формалізації	Часто неповні, не враховують стратегічні пріоритети, не інтегровані у загальну систему

Джерело: розроблено автором за конфіденційними даними підприємства, що досліджувалось.

З табл. 2.2 видно, що поточна практика прийняття рішень у R&D-командах базується на поєднанні інтуїтивних та аналітичних підходів, однак жоден із них окремо не забезпечує повної об'єктивності та системності оцінки.

Виявлені проблеми формалізації критеріїв:

– Неузгодженість метрик – різні команди оцінюють ефективність моделей за різними показниками, що ускладнює порівняння.

– Складність багатокритеріальної оцінки – труднощі у формалізації пріоритетів між якістю, вартістю та часом розробки.

– Обмежена історія даних – недостатня база для статистичної оцінки ризиків і прогнозування результатів.

На основі виявлених обмежень існуючого підходу виконано систематизацію основних ризиків управлінського процесу (табл. 2.3).

Таблиця 2.3. Ризики досліджуваної технології управління

Тип ризику	**Причини**	**Потенційні наслідки**
Помилки в оцінці	Недостатня статистична база, упередженість експертів	Впровадження неефективних моделей, зниження якості продукту
Затримки у прийнятті рішень	Ручний аналіз, багатоступеневе погодження	Втрата конкурентних переваг, пропуск дедлайнів
Неузгодженість рішень	Різні пріоритети команд і стейкхолдерів	Конфлікти між стратегічними цілями, повторна робота

Джерело: розроблено автором за конфіденційними даними підприємства, що досліджувалось.

Із табл. 2.3 видно, що ключові ризики пов’язані з недостатньою формалізацією процесів оцінки та високою залежністю від експертного судження. Подальший аналіз дозволяє деталізувати основні проблемні аспекти існуючої технології прийняття рішень.

Аналіз поточного процесу дозволяє виділити ключові дисфункції, що знижують ефективність прийняття рішень.

1. Інформаційна асиметрія

Команди часто мають неповний або розрізнений доступ до даних.

Метрики не інтегровані між бізнес-аналітикою та технічними експериментами.

2. Невизначеність у прогнозуванні результатів моделей

Експериментальні результати складно переносити на реальні сценарії через змінність даних або обмеженість тренувальних вибірок.

Прогнози мають високу дисперсію та малу відтворюваність.

3. Недостатня інтеграція аналітики в процес прийняття рішення

Метрики часто збираються після проведення експериментів, без безпосереднього впливу на вибір альтернатив.

Рішення частково опираються на інтуїцію, що знижує прозорість і обґрунтованість.

4. Вузькі місця (bottlenecks) (табл. 2.4)

Таблиця 2.4. Вузькі місця досліджуваної технології управління

Вузьке місце	**Причина**	**Наслідки**
Затримка у зборі та обробці даних	Ручний аналіз, розрізнені системи	Погіршення своєчасності управлінських рішень
Неузгодженість критеріїв	Відсутність єдиної функції корисності	Конфлікти між командами, субоптимальні рішення
Обмежена інтеграція моделей прогнозування	Роз’єднані експерименти і бізнес-процеси	Невикористані можливості оптимізації, високий ризик помилок

Джерело: розроблено автором на основі [3], [6], [10], [14], [16].

З табл. 2.4 видно, що основні обмеження пов’язані з фрагментованістю даних, відсутністю єдиної моделі оцінки та недостатньою інтеграцією аналітики у

процес прийняття рішень.

2.3. Постановка задачі

Аналіз існуючої технології прийняття рішень у команді R&D дозволяє сформулювати ключову задачу удосконалення: забезпечити більш обґрунтований, прозорий та прогнозований процес вибору стратегій за допомогою інтеграції аналітики та формалізованих моделей.

Метою є створення моделі прийняття рішень, яка поєднує:

– Емпіричні дані (результати експериментів моделей, метрики продуктивності, бізнес-показники);

– Кількісні критерії (прибутковість, точність моделей, ризик затримок);

– Пріоритети команди та стейкхолдерів (стратегічні цілі, обмеження ресурсів).

У термінах системного аналізу задача формулюється як оптимізаційна проблема з обмеженнями та багатокритеріальною функцією корисності, де необхідно вибрати оптимальний набір дій $a*$ із множини альтернатив A, що максимізує цільову функцію ефективності.

Оскільки середовище прийняття рішень характеризується невизначеністю, функція корисності може розглядатися як випадкова величина або оцінюватися через очікуване значення.

Визначені ключові і цільові критерії ефективності представлено в таблиці 2.5.

Таблиця 2.5. Цільові критерії ефективності

Критерій	**Пояснення**	**Метрика/показник**
Точність моделі	Наскільки обрана модель задовольняє вимоги бізнесу	F1-score, Accuracy, BLEU, ROUGE
Вартість розробки	Ресурси та час, необхідні для реалізації рішення	Люд.-години, обчислювальні витрати
Прогнозована ефективність	Очікуваний вплив на бізнес-процеси	ROI, підвищення KPI, скорочення часу обробки
Ризик	Можливість невідповідності результатів прогнозам	Дисперсія результатів, ймовірність відхилень
Своєчасність	Швидкість прийняття та впровадження рішення	Час від постановки задачі до релізу моделі

Джерело: розроблено автором на основі [3], [6], [12].

Із табл. 2.5 видно, що оцінювання альтернатив у досліджуваній системі ґрунтується на поєднанні технічних, економічних і часових критеріїв, що дозволяє забезпечити багатовимірний підхід до прийняття управлінських рішень та врахування якості, вартості й ризиків одночасно.

Тож інтегральна функція корисності формується як агрегована оцінка вказаних критеріїв із урахуванням їх вагових коефіцієнтів.

Важливо, що критерії мають бути не лише вимірюваними, а й узгодженими між командами, щоб забезпечити єдину систему пріоритезації та оцінки альтернатив.

При розробці моделі вдосконалення слід враховувати реальні обмеження:

– Ресурси – обмежена кількість аналітиків, розробників та обчислювальних потужностей;

– Час – необхідність ухвалювати рішення в межах спринтів або бізнес-циклів;

– Дані – неповнота, розрізненість та шумність наявних даних;

– Компетенції – рівень навичок команди у статистичному аналізі, машинному навчанні та використанні BI-інструментів.

Формалізація проблеми з урахуванням цих критеріїв та обмежень дозволяє перейти до розробки інтегрованої технології Data-Driven Management, яка забезпечить більш структурований та прозорий процес прийняття рішень у команді

R&D.

Запропонована модель має багаторівневу структуру, що поєднує етапи підготовки даних, стохастичного моделювання, оцінки ризику та оптимізаційного вибору рішення.

Модель може реалізовуватися у вигляді програми/скрипту, що автоматизує підбір оптимальної альтернативи:

– Збір даних – автоматичне підключення до внутрішніх BI-систем, експериментальних результатів моделей, логів продукту;

– Побудова критеріїв ефективності – нормалізація показників, зважування за пріоритетами бізнесу;

– Розрахунок функції корисності – використання багатокритеріальної оцінки та імовірнісних моделей (наприклад, симуляцій, Monte Carlo, Bayesian approaches);

– Вибір оптимальної альтернативи – застосування алгоритму argmax або евристичних методів пошуку, які враховують ризики та обмеження;

– Візуалізація результатів – графіки, таблиці, інтерактивні дашборди для менеджменту;

– Механізми адаптації – оновлення даних і критеріїв у режимі реального часу, що дозволяє відстежувати зміни в попиті або ефективності моделей.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ DATA-DRIVEN ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

3.1 Архітектура моделі

Модель може бути побудована як багаторівнева система, що інтегрує дані, аналітику та логіку прийняття рішень. Основні компоненти та їх функції наведено у таблиці 3.1:

Таблиця 3.1. Компоненти системи оптимізації

Компонент	**Призначення**	**Деталі**
Джерела даних	Забезпечують інформаційний фундамент	- **Внутрішні системи R&D:** результати експериментів, метрики моделей, логи продукту - **Бізнес-дані:** KPI, показники ефективності релізів, ресурси - **Зовнішні дані:** тренди ринку, відгуки користувачів, конкурентна інформація
Модуль обробки та підготовки даних	Створює базу для багатокритеріальної оцінки	- Нормалізація, фільтрація та агрегація метрик - Формування узгоджених таблиць для розрахунку функції корисності - Перевірка якості та повноти даних
Модуль аналітики та моделювання	Формалізує оцінку альтернатив та ризиків	- Розрахунок багатокритеріальної функції - Моделювання ризиків та невизначеності через ймовірнісні оцінки та сценарний аналіз - Побудова таблиць порівняння альтернатив, графіків чутливості, діаграм очікуваної ефективності - Використання методів оптимізації: argmax, евристики, Monte Carlo
Модуль прийняття рішень	Вибір оптимальної стратегії на основі критеріїв	- Формалізація пріоритетів бізнесу, ресурсних та часових обмежень, допустимого ризику - Алгоритмічний підбір оптимальної альтернативи з урахуванням порогових умов (chance-constrained rules) - Підготовка рекомендацій для менеджме-

		нту та команди розробки
Інтерфейс аналітика/BA	Забезпечує доступність та зрозумілість рішень	- Візуальні таблиці та графіки для оцінки альтернатив - Інструменти «що-якщо» для перевірки сценаріїв - Можливість інтеграції з BI-дашбордами та звітами для стейкхолдерів

Джерело: розроблено автором на основі [23].

З табл. 3.1 видно, що система оптимізації прийняття рішень має модульну архітектуру, у якій кожен компонент виконує окрему функціональну роль - від збору та підготовки даних до аналітичного моделювання, вибору оптимальної альтернативи та візуалізації результатів для кінцевого користувача.

Логіка роботи моделі:

– Дані з внутрішніх, бізнес- та зовнішніх джерел надходять у модуль обробки.

– Оброблені метрики нормалізуються і формуються у багатокритеріальні таблиці.

– Модуль аналітики розраховує функцію корисності для кожної альтернативи, враховуючи ризики та ймовірності станів.

– Алгоритми оптимізації (argmax, евристики, Monte Carlo) знаходять оптимальну альтернативу з урахуванням порогових умов та обмежень.

– Інтерфейс надає візуалізацію результатів, сценарії «що-якщо» та рекомендації для ВА і менеджменту.

Схематичне представлення

Джерела даних ⟶ Модуль обробки та підготовки ⟶ Модуль аналітики та моделювання ⟶ Модуль прийняття рішень ⟶ Інтерфейс ВА/менеджмент

3.2 Вимоги до функціоналу моделі

Use Case 1: Підготовка даних і визначення критеріїв

Actor: BA/системний аналітик

Preconditions: Наявні дані з внутрішніх, бізнес та зовнішніх джерел. Визначені первинні KPI та стратегічні цілі.

Main Flow:

1) Аналітик імпортує дані в систему.
2) Нормалізує та фільтрує метрики для уніфікації.
3) Визначає критерії оцінки альтернатив (прибутковість, ризик затримок, вплив на клієнта).
4) Присвоює ваги критеріям згідно пріоритетів бізнесу.
5) Формує таблицю багатокритеріальної оцінки, готову для розрахунку функції корисності.

Alternative Flows:

A1: Дані частково відсутні → аналітик позначає пропуски, використовуючи прогнозні оцінки або евристики.

A2: Критерії змінюються під час перегляду стратегії → аналітик оновлює ваги та таблиці без переробки всіх даних.

Use Case 2: Розрахунок функції корисності та оцінка альтернатив

Actor: BA/системний аналітик

Preconditions: Таблиця багатокритеріальної оцінки сформована, ваги критеріїв визначені.

Main Flow:

6) Система розраховує функцію корисності (багатокритеріальна).
7) Виконується ймовірнісне моделювання для сценаріїв невизначеності (попит, ризик, ресурсні обмеження).
8) Система визначає оптимальні альтернативи через argmax, Monte Carlo або евристичні алгоритми.

9) Формується таблиця порівняння альтернатив із очікуваною корисністю та ризиком.

Alternative Flows:

A1: Виявлено високий ризик у деяких альтернативах → додається порогова умова.

A2: Критерії пріоритетів змінюються після попереднього розрахунку → повторний розрахунок функції корисності без зміни вихідних даних.

Use Case 3: Візуалізація результатів

Actor: BA/системний аналітик

Preconditions: Функція корисності розрахована, альтернативи оцінені.

Main Flow:

10) Система будує графіки очікуваної корисності та криві «гарантованої виплати».

11) Генерує діаграми чутливості для перевірки впливу ваг критеріїв.

12) Формує інтерактивні таблиці для порівняння альтернатив.

13) Аналітик аналізує результати та відзначає критичні точки для ухвалення рішень.

Alternative Flows:

A1: Потреба у сценарному аналізі → аналітик створює «що-якщо» сценарії, змінюючи ймовірності та параметри моделей.

A2: Для керівництва потрібні агреговані результати → система генерує консолідовані графіки та зведені таблиці.

Use Case 4: Надання рекомендацій та пояснення логіки

Actor: BA/системний аналітик

Preconditions: Візуалізація та сценарний аналіз завершені.

Main Flow:

14) Аналітик готує рекомендації щодо вибору альтернатив для

менеджменту та команди.

15) Пояснює логіку вибору: як ваги критеріїв, ймовірності та порогові умови вплинули на оптимальне рішення.

16) Вказує на можливі ризики та компроміси між ефективністю та надійністю.

17) Забезпечує документування процесу для подальшого аудиту та аналізу.

Alternative Flows:

A1: Не всі альтернативи прийнятні → аналітик виділяє допустимі і пропонує компромісні рішення.

A2: Нові дані або зміни у стратегії → рекомендації оновлюються без зміни базової моделі.

Переваги такої архітектури:

- Модель формалізує весь процес, зменшуючи роль інтуїції.
- Забезпечує прозорість та відтворюваність рішень.
- Підтримує багатокритеріальні та ймовірнісні оцінки одночасно.
- Дає змогу проводити сценарне моделювання та аналіз чутливості.
- Відповідає принципам найкращих практик BA: use case, альтернативні сценарії, traceability до бізнес-цілей.

3.3 MVP (мінімальний життєздатний продукт) системи

Було розроблено мінімальний життєздатний продукт системи та надано опис його функціоналу нижче.

Документ MVP (мінімально життєздатного продукту)

1. Вступ

Цей документ надає опис **Мінімально Життєздатного Продукту (MVP)** системи підтримки управлінських рішень, розробленої на основі фреймворку Streamlit. Система призначена для сприяння прийняттю рішень у складних умовах за рахунок інтеграції методів множинно-критеріального аналізу, симуляції невизначеності та оптимізації (рис. 3.1).

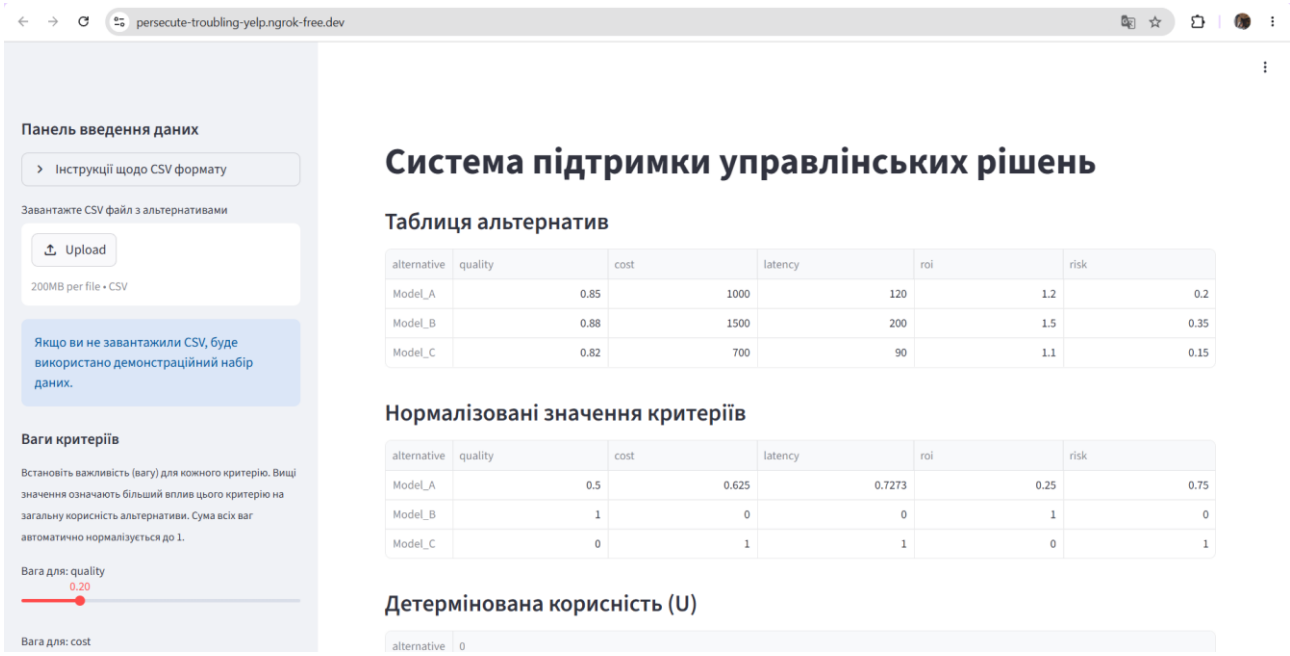


Рисунок 3.1 - Головна сторінка розробленої системи

Джерело: розроблено автором

2. Призначення Системи

Основна мета системи – надати користувачам інструмент для об'єктивної оцінки альтернатив за низкою визначених критеріїв. Вона дозволяє:

- **Кількісно оцінювати альтернативи**: шляхом нормалізації та зважування критеріїв.
- **Враховувати невизначеність та ризики**: використовуючи симуляцію Монте-Карло та Байєсівську оцінку ймовірностей.
- **Оптимізувати вибір**: знаходити найкращу альтернативу, яка задовольняє заданим обмеженням ризику, включно з використанням генетичного алгоритму для евристичного пошуку.
- **Візуалізувати результати**: для кращого розуміння розподілу корис-

ності та чутливості до параметрів.

3. Архітектура та Основні Блоки Системи

Система організована у вигляді послідовних блоків, кожен з яких виконує специфічну функцію:

- **Блок введення даних (Data Input)** (рис. 3.2):

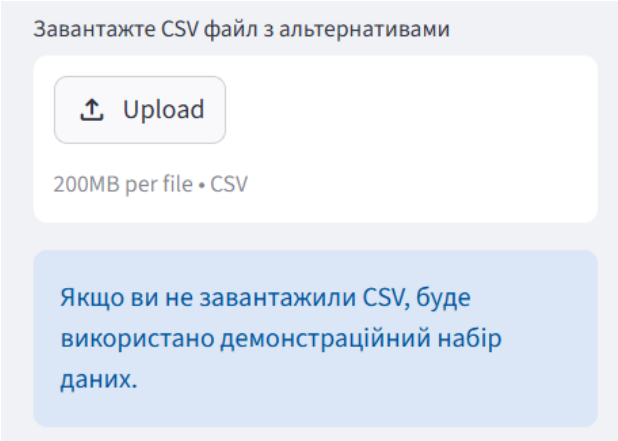


Рисунок 3.2 - Завантаження датасету в системі

Джерело: розроблено автором

  - **Призначення**: Завантаження та відображення вихідних даних про альтернативи та їхні критерії.
  - **Функціонал**: Підтримує завантаження CSV-файлів з визначеним форматом (alternative, quality, cost, latency, roi, risk) або використовує демонстраційний набір даних за замовчуванням. Користувач може налаштовувати ваги для кожного критерію та визначати, чи є критерій «вигодою» (Benefit) чи «вартістю» (Cost).

- **Блок нормалізації та розрахунку корисності (Normalization & Utility Calculation)** (рис. 3.3 і 3.4):

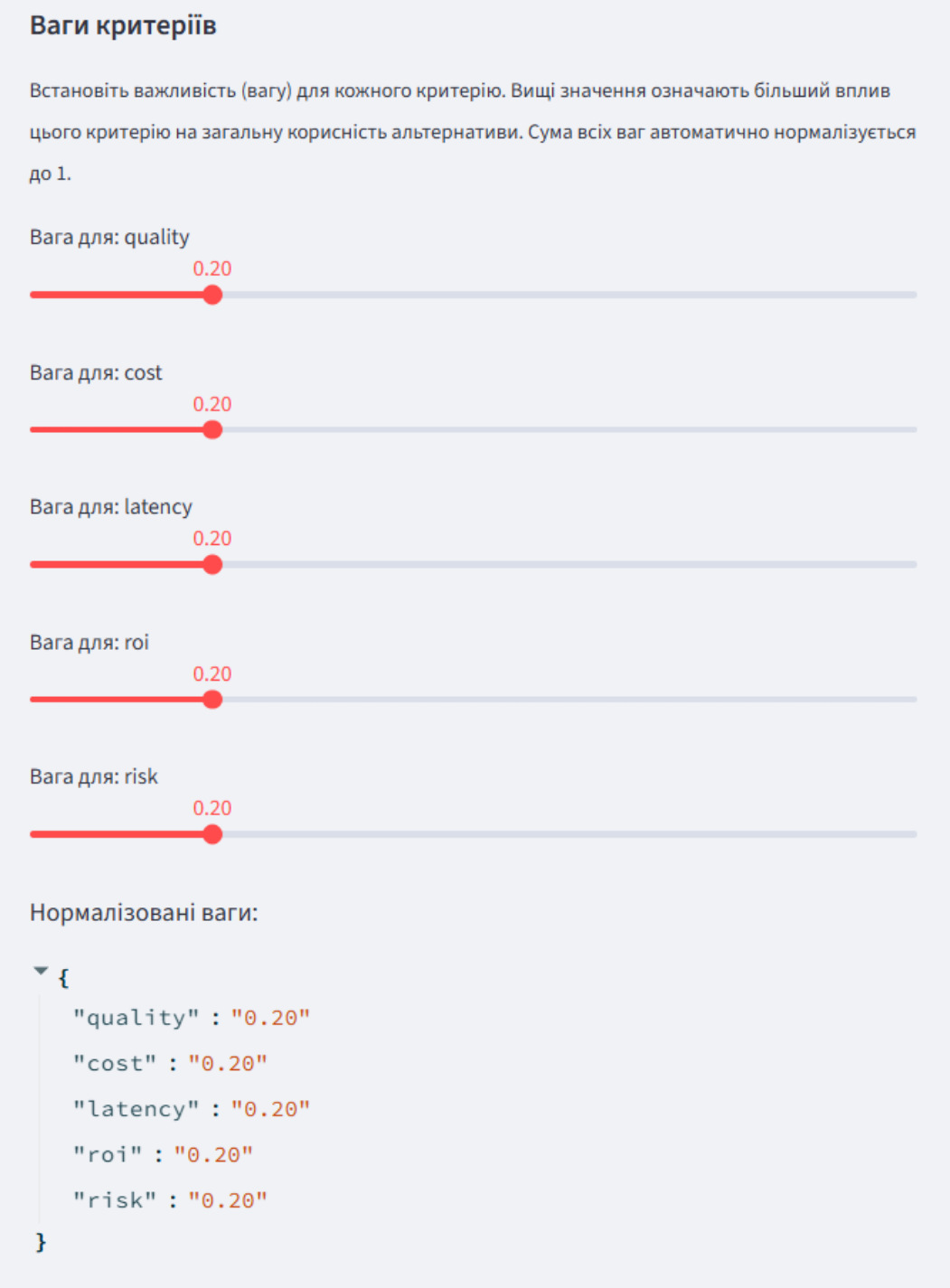


Рисунок 3.3 – Нормалізація

Джерело: розроблено автором



Рисунок 3.4 - Вибір типу критеріїв

Джерело: розроблено автором

- **Призначення**: Перетворення вихідних значень критеріїв до єдиної шкали та розрахунок детермінованої корисності.
- **Функціонал**: Нормалізує критерії (максимізує Benefit, мінімізує Cost) та обчислює загальну корисність для кожної альтернативи на основі заданих ваг.

- **Блок симуляції Монте-Карло (Monte Carlo Simulation)** (рис. 3.5 і 3.6):

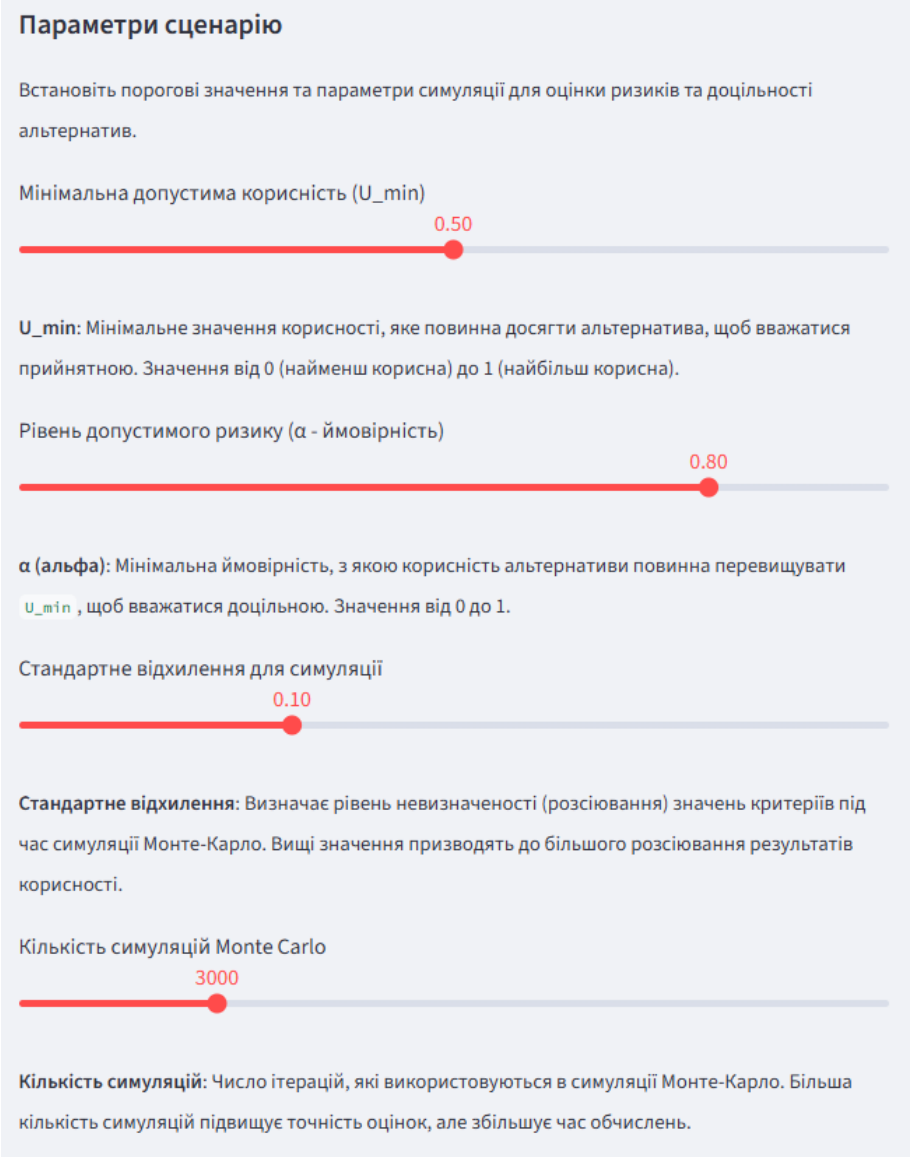


Рисунок 3.5 - Параметри сценарію

Джерело: розроблено автором

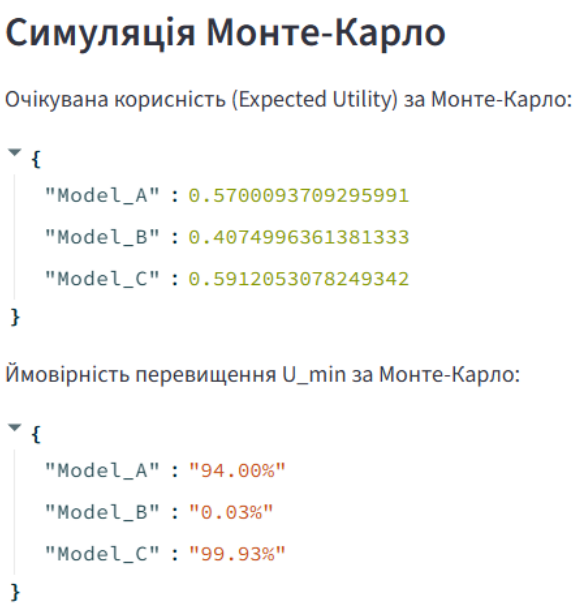


Рисунок 3.6 - Симуляція Монте-Карло

Джерело: розроблено автором

- **Призначення**: Оцінка впливу невизначеності на корисність альтернатив.
- **Функціонал**: Генерує випадкові варіації критеріїв на основі заданого стандартного відхилення та проводить n_sim симуляцій для кожної альтернативи. Розраховує очікувану корисність (Expected Utility) та ймовірність перевищення мінімальної допустимої корисності (U_min).

- **Блок Байєсівської оцінки ризику (Bayesian Risk Assessment)** (рис. 3.7 і 3.8):

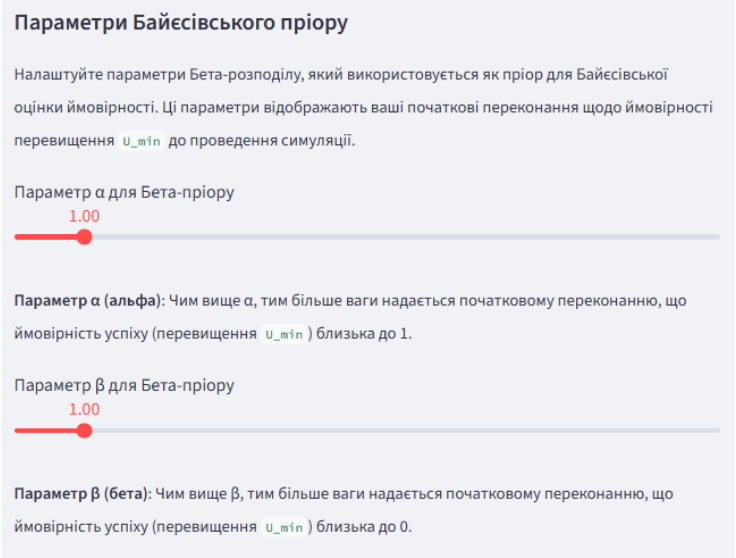


Рисунок 3.7 - Параметри Байєсівського пріору

Джерело: розроблено автором

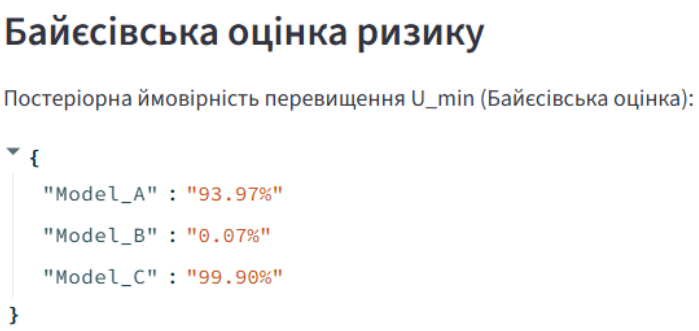


Рисунок 3.8 - Байєсівська оцінка ризику

Джерело: розроблено автором

- **Призначення**: Уточнення оцінок ймовірності успіху з урахуванням попередніх переконань.
- **Функціонал**: Використовує результати симуляції Монте-Карло як «докази» для оновлення Бета-пріору, надаючи постеріорну ймовірність перевищення U_min.

- **Блок обмеження ризику та доцільності (Risk Constraint & Feasibility)** (рис. 3.9):

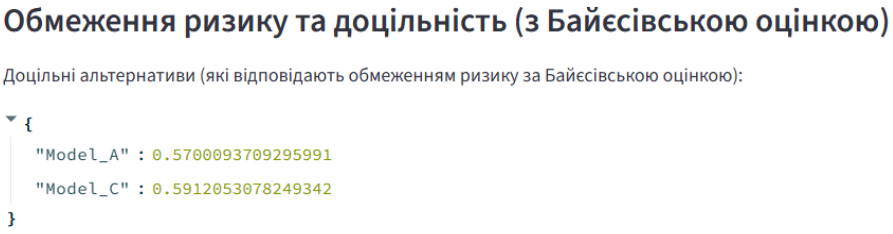


Рисунок 3.9 - Обмеження ризику та доцільність (Байєсівська оцінка)

Джерело: розроблено автором

- **Призначення**: Визначення альтернатив, які відповідають заданому рівню допустимого ризику.
- **Функціонал**: Фільтрує альтернативи, залишаючи лише ті, чия постеріорна ймовірність перевищення U_min є вищою за встановлений поріг (\alpha).

- **Блок оптимізації (Optimization)** (рис. 3.10):

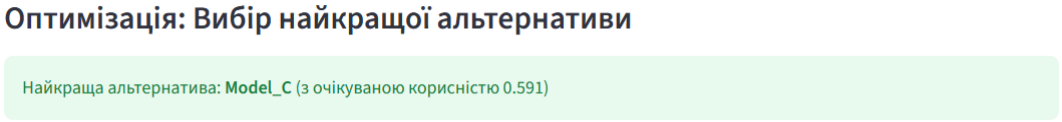


Рисунок 3.10 - Вибір найкращої альтернативи

Джерело: розроблено автором

- **Призначення**: Вибір найкращої альтернативи серед доцільних.
- **Функціонал**: Обирає альтернативу з найвищою очікуваною корисністю серед доцільних альтернатив.

- **Блок оптимізації за допомогою Генетичного Алгоритму (Genetic Algorithm Optimization)** (рис. 3.11 і 3.12):

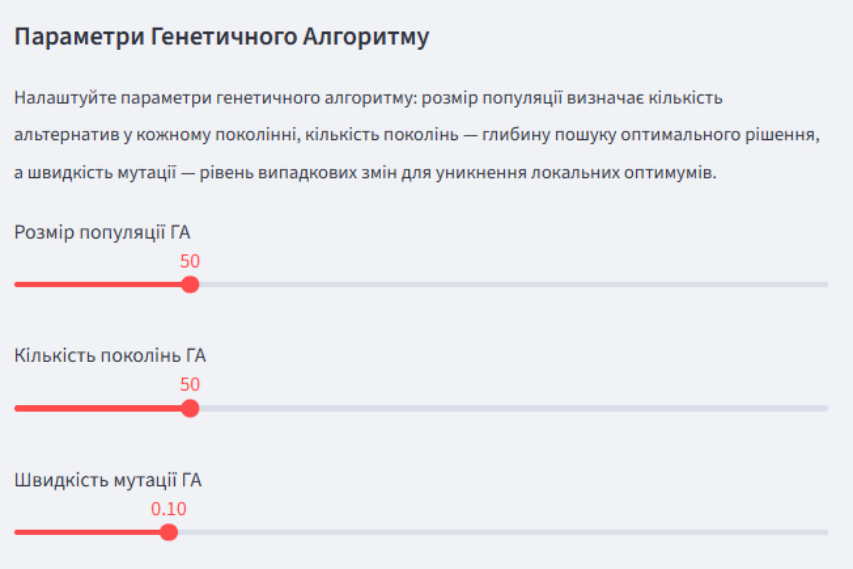


Рисунок 3.11 - Параметри Генетичного алгоритму

Джерело: розроблено автором

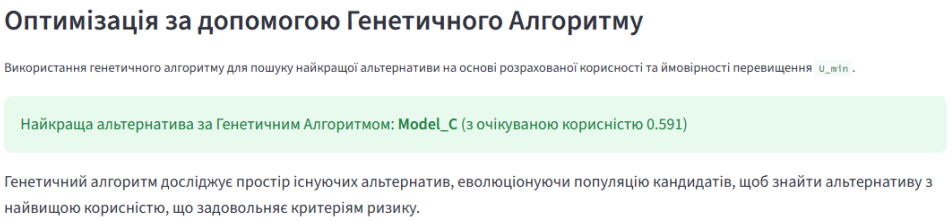


Рисунок 3.12 - Оптимізація за допомогою Генетичного алгоритму

Джерело: розроблено автором

- **Призначення**: Альтернативний евристичний метод для пошуку оптимальних рішень, особливо корисний у складних пошукових просторах.
- **Функціонал**: Реалізує генетичний алгоритм для вибору найкращої альтернативи, яка задовольняє критеріям ризику, використовуючи задані параметри популяції, поколінь та швидкості мутації.

- **Блок візуалізації (Visualization)** (рис. 3.13, 3.14, 3.15):

Розподіл корисності (Utility Distribution)

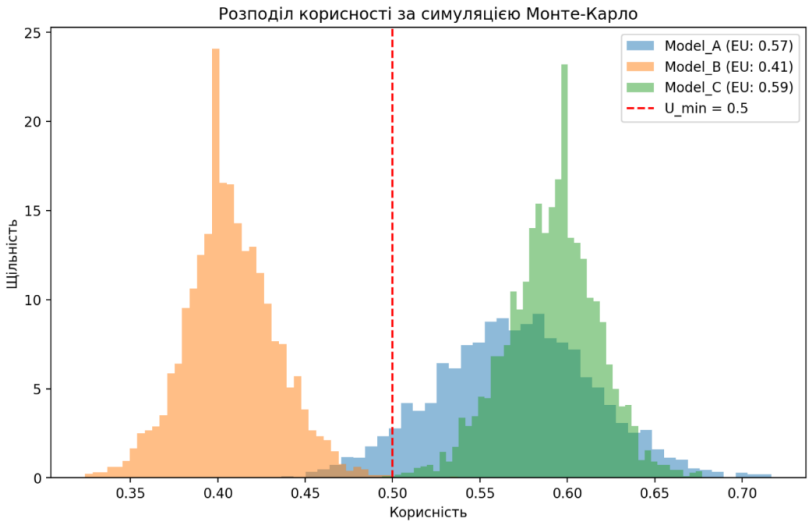


Рисунок 3.13 - Візуалізація: Розподіл корисності

Джерело: розроблено автором

Аналіз чутливості (вплив ваг критеріїв)

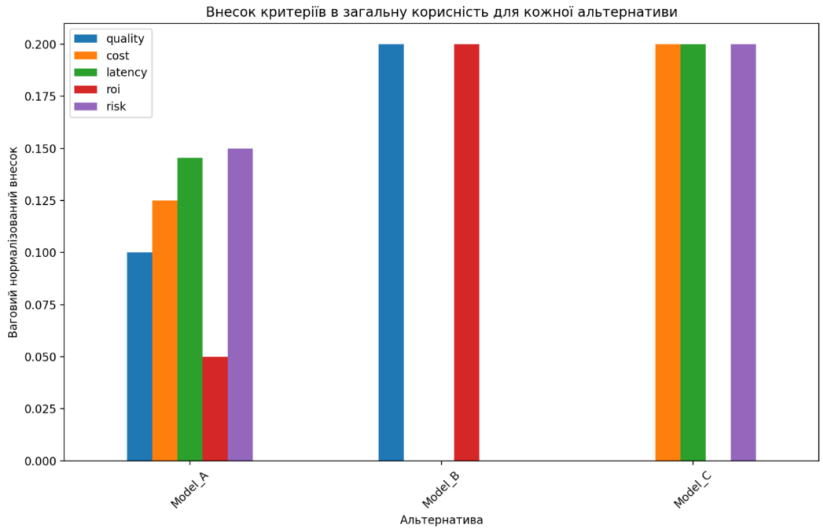


Рисунок 3.14 - Візуалізація: Аналіз чутливості

Джерело: розроблено автором

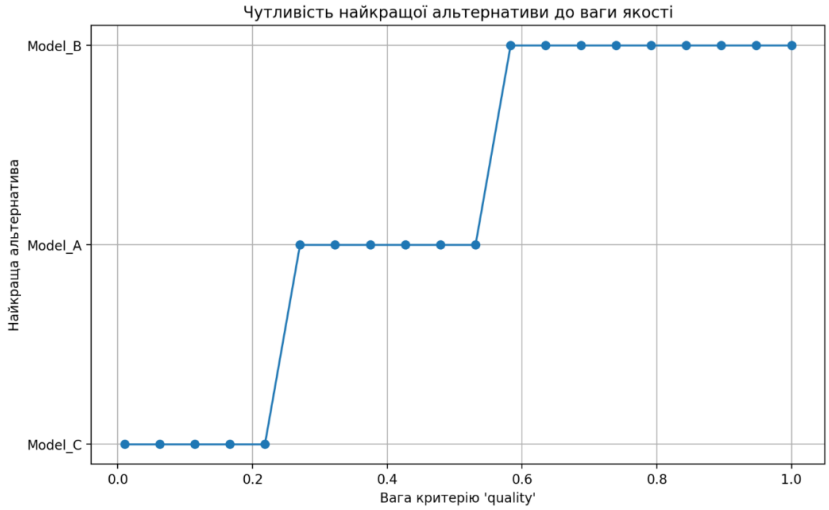
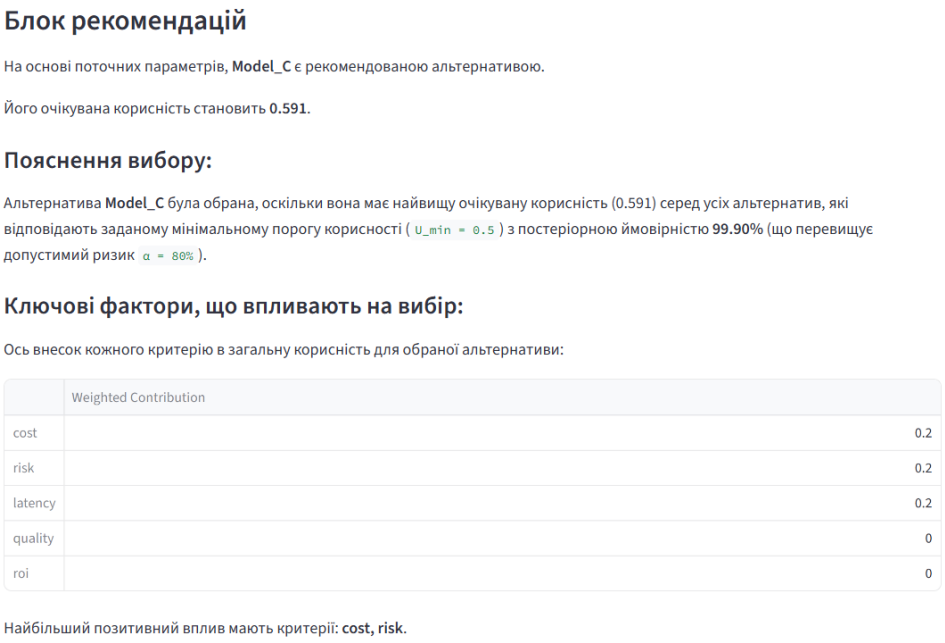


Рисунок 3.15 - Візуалізація: Чутливість найкращої альтернативи до ваги якості

Джерело: розроблено автором

- **Призначення**: Графічне представлення результатів аналізу.
- **Функціонал**: Включає гістограми розподілу корисності, стовпчасті діаграми внеску критеріїв та графіки чутливості до змін ваг критеріїв.

- **Блок рекомендацій (Recommendations)** (рис. 3.16 і 3.17):



Блок рекомендацій

На основі поточних параметрів, **Model_C** є рекомендованою альтернативою.

Його очікувана корисність становить **0.591**.

Пояснення вибору:

Альтернатива **Model_C** була обрана, оскільки вона має найвищу очікувану корисність (0.591) серед усіх альтернатив, які відповідають заданому мінімальному порогу корисності (`U_min = 0.5`) з постеріорною ймовірністю **99.90%** (що перевищує допустимий ризик `α = 80%`).

Ключові фактори, що впливають на вибір:

Ось внесок кожного критерію в загальну корисність для обраної альтернативи:

	Weighted Contribution
cost	0.2
risk	0.2
latency	0.2
quality	0
roi	0

Найбільший позитивний вплив мають критерії: **cost, risk**.

Рисунок 3.16 – Рекомендації

Джерело: розроблено автором

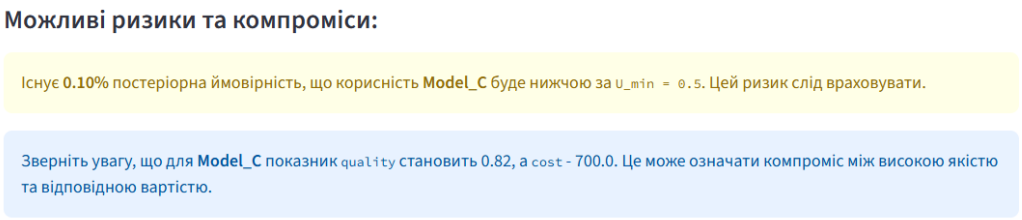


Рисунок 3.17 - Можливі ризики

Джерело: розроблено автором

- **Призначення**: Формулювання висновків та обґрунтування вибору.
- **Функціонал**: Надає текстову рекомендацію щодо найкращої альтернативи, пояснює причини вибору, вказує на ключові фактори впливу та можливі ризики.

4. Використання Системи

1. **Запуск**: Запустіть Streamlit-додаток (зазвичай командою streamlit run app.py або через наданий публічний тунель ngrok).
2. **Введення даних**: Завантажте власний CSV-файл або скористайтеся демонстраційними даними.
3. **Налаштування параметрів**: За допомогою бічної панелі встановіть:
   - **Ваги критеріїв**: Визначте відносну важливість кожного критерію.
   - **Тип критеріїв (Benefit/Cost)**: Вкажіть, чи потрібно максимізувати (Benefit) чи мінімізувати (Cost) значення критерію.
   - **Параметри сценарію**: Встановіть мінімальну допустиму корисність (U_min), рівень допустимого ризику ((\alpha)), стандартне відхилення для симуляції та кількість симуляцій Монте-Карло.
   - **Параметри Байєсівського пріору**: Налаштуйте параметри Бета-розподілу для Байєсівської оцінки.

- **Параметри Генетичного Алгоритму**: Встановіть розмір популяції, кількість поколінь та швидкість мутації.

4. **Аналіз результатів**: Ознайомтеся з висновками в розділах «Детермінована корисність», «Симуляція Монте-Карло», «Байєсівська оцінка ризику», «Обмеження ризику та доцільність», «Оптимізація» та «Оптимізація за допомогою Генетичного Алгоритму».

5. **Візуалізація**: Вивчіть графіки для отримання глибшого розуміння поведінки системи.

6. **Рекомендації**: Ознайомтеся з підсумковою рекомендацією та її обґрунтуванням.

5. Корисні блоки та функціонал

- **Динамічне визначення Benefit/Cost**: Дозволяє гнучко адаптувати модель до різних типів критеріїв без зміни коду.
- **Звітність у реальному часі**: Усі розрахунки та візуалізації оновлюються миттєво при зміні параметрів, забезпечуючи інтерактивний аналіз.
- **Інтуїтивно зрозумілі пояснення**: Додано детальні описи для кожного параметра в бічній панелі, що робить систему доступною для користувачів без глибоких технічних знань.
- **Евристична оптимізація**: Інтеграція генетичного алгоритму доповнює аналітичні методи, надаючи потужний інструмент для пошуку рішень, особливо при значній кількості альтернатив.

6. Програмна реалізація та інженерні особливості системи підтримки прийняття рішень

Архітектурна організація застосунку

Архітектура розробленої системи підтримки прийняття рішень побудована за принципом логічного розділення функціональних компонентів із орієнтацією на прозорість обчислювального процесу та інтерактивність взаємодії з користува-

чем. Програмна реалізація виконана у вигляді односторінкового веб-застосунку, що поєднує інтерфейс введення параметрів, аналітичне ядро обробки даних та модулі візуалізації результатів.

У загальному вигляді архітектуру системи можна представити як послідовність взаємопов'язаних рівнів: рівень представлення (інтерфейс користувача), рівень обробки даних та рівень обчислювальної логіки. Такий підхід забезпечує модульність системи та спрощує її розширення або модифікацію.

Рівень представлення реалізовано з використанням інтерактивного середовища, що дозволяє користувачу задавати параметри моделі, завантажувати вхідні дані та отримувати результати у зручній візуальній формі. Взаємодія з користувачем організована через бічну панель параметрів, де здійснюється налаштування ваг критеріїв, вибір типів критеріїв (вигода або витрати), а також задання параметрів невизначеності та ризику. Основна частина інтерфейсу відображає результати обчислень у вигляді таблиць і графіків, що сприяє підвищенню інтерпретованості моделі.

Рівень обробки даних відповідає за прийом, валідацію та попередню підготовку вхідної інформації. Дані можуть надходити у вигляді CSV-файлу або використовуватися з наперед визначеного демонстраційного набору. На цьому етапі здійснюється приведення даних до уніфікованого формату, зокрема встановлення індексів альтернатив, перевірка коректності структури та забезпечення числового типу значень критеріїв. Таким чином формується узгоджене представлення даних, придатне для подальшого аналізу.

Ключовим елементом архітектури є обчислювальне ядро, яке реалізує основні аналітичні процедури. До них належать нормалізація критеріїв, обчислення детермінованої функції корисності, імітаційне моделювання методом Монте-Карло, байєсівська оцінка ризику та оптимізація вибору альтернатив. Обчислювальний процес організовано як послідовний конвеєр трансформацій, де вихід одного етапу слугує вхідними даними для наступного. Така структура забезпечує логі-

чну цілісність моделі та спрощує трасування результатів.

Окрему роль в архітектурі відіграє модуль візуалізації, який забезпечує графічне представлення результатів аналізу. Використання гістограм, діаграм внесків критеріїв та графіків чутливості дозволяє не лише відобразити числові результати, але й виявити приховані закономірності, що є важливим для прийняття обґрунтованих управлінських рішень.

Особливістю реалізації є реактивна модель виконання, за якої будь-яка зміна вхідних параметрів автоматично ініціює повний перерахунок усіх залежних величин. Це забезпечує миттєвий зворотний зв’язок та дозволяє проводити інтерактивний аналіз сценаріїв без необхідності ручного перезапуску обчислень.

Обробка вхідних даних

Етап обробки вхідних даних у межах системи відіграє роль первинного фільтра, який забезпечує придатність інформації для подальших розрахунків. Оскільки модель працює з багатокритеріальними альтернативами, будь-які неточності у вхідній структурі безпосередньо впливають на коректність результатів, тому цей етап фактично визначає надійність усієї системи.

Вхідні дані подаються у вигляді CSV-файлу, що містить перелік альтернатив та набір числових характеристик для кожної з них. Після завантаження файл перетворюється у табличну структуру DataFrame, яка є базовим форматом для подальших обчислень. Ключовим моментом є те, що стовпець alternative використовується як індекс. Це дозволяє розглядати кожну альтернативу як унікальний об’єкт, до якого надалі прив’язуються всі операції нормалізації, зважування та оцінювання корисності.

Перед початком аналітичної обробки виконується перевірка структури вхідних даних. Вона включає контроль наявності необхідних стовпців, узгодженість їхніх назв із заданою схемою та перевірку можливості приведення значень до числового типу. У випадку виявлення помилок використовується механізм виключень try/except, який дозволяє локалізувати проблему без зупинки всього застосу-

нку. З практичної точки зору це важливо, оскільки в реальних умовах користувацькі дані часто містять неточності або порушення формату.

Окремою проблемою є некоректність або неоднорідність вхідних даних. Це може проявлятися у вигляді пропущених значень, текстових символів у числових полях або змішаних одиниць вимірювання. У межах реалізації прийнято спрощене припущення, що всі критерії, крім ідентифікатора альтернативи, є числовими та вже приведеними до порівнюваного вигляду. Таке припущення дозволяє сфокусуватися на логіці багатокритеріального аналізу, не перевантажуючи модель додатковими процедурами очищення даних.

У разі відсутності завантаженого файлу система переходить до використання вбудованого демонстраційного набору даних. Це рішення забезпечує стабільну роботу застосунку незалежно від зовнішніх джерел і дозволяє відтворювати результати навіть у тестовому середовищі.

Реалізація нормалізації критеріїв

Нормалізація вихідних критеріїв є необхідним етапом, який забезпечує приведення різнорідних показників до єдиної безрозмірної шкали. Це дозволяє коректно агрегувати критерії різної природи (вартість, якість, ризик, час тощо) у межах єдиної функції корисності. У межах реалізованої системи нормалізація виконується як окрема процедурна функція, що застосовується до кожного стовпця вхідної матриці даних.

З технічної точки зору реалізація побудована на послідовній ітерації по всіх критеріях. Для кожної колонки обчислюються мінімальне та максимальне значення, після чого виконується перетворення значень відповідно до типу критерію. Такий підхід забезпечує гнучкість обробки та дозволяє однаково працювати як із вигідними (Benefit), так і з витратними (Cost) показниками.

Для критеріїв типу Benefit застосовується класична min-max нормалізація, що інтерпретується як відносне положення значення в межах діапазону спостережень. Для критеріїв типу Cost використовується обернене перетворення.

Такий підхід забезпечує узгодженість інтерпретації: у всіх випадках більші значення нормалізованого показника відповідають кращим характеристикам альтернативи.

Окрему увагу в реалізації приділено ситуації, коли розмах значень критерію дорівнює нулю, тобто:

$$\max(x) - \min(x) = 0.$$

Така ситуація виникає, коли всі альтернативи мають однакове значення певного показника. У цьому випадку пряме застосування формули нормалізації призводить до ділення на нуль і втрати числової стійкості. Для уникнення цієї проблеми в системі застосовано інженерне правило заміщення: усім альтернативам присвоюється фіксоване нормалізоване значення 0,5. Це дозволяє зберегти нейтральний внесок критерію в загальну функцію корисності без викривлення результатів.

Додатково реалізовано динамічний вибір типу критерію, який задається користувачем через інтерфейс. Це означає, що класифікація «Benefit/Cost» не є жорстко зашитою в коді, а формується на основі контекстного налаштування. Такий підхід підвищує універсальність системи та дозволяє застосовувати її до різних предметних областей без модифікації логіки обчислень.

У підсумку, реалізований механізм нормалізації виконує не лише математичну функцію приведення даних до спільної шкали, але й забезпечує числову стабільність, адаптивність до типу критеріїв та коректність подальших етапів багатокритеріального аналізу.

Векторизація обчислень

У реалізації моделі ключовим інженерним рішенням є перехід від ітеративних (циклічних) операцій до векторизованих обчислень із використанням NumPy та pandas. Це дозволяє суттєво підвищити продуктивність системи, особливо при збільшенні кількості альтернатив або симуляцій у Монте-Карло процедурі.

Формально розрахунок корисності можна подати у вигляді формули:

$$U = X * w$$

де X - матриця нормалізованих значень критеріїв,

w - вектор ваг.

У коді це реалізовано через:

- np.dot(simulated_values, weights)
- df.mul(pd.Series(weights))
- sum(axis=1) для агрегації результатів

Замість поелементного обходу структури даних використовується матрична форма обчислень, що дозволяє бібліотекам низького рівня виконувати операції оптимізовано (через C-реалізацію та SIMD-інструкції).

Практичний ефект такого підходу полягає в тому, що складність обчислень переходить від явної ітеративної O(n « m) логіки до більш ефективної векторизованої форми, де основні операції виконуються пакетно. Це особливо критично для блоку Монте-Карло, де виконується тисячі ітерацій:

$$U(k) = X(k) \quad w, k=1,\ldots, N$$

Де U(k) - значення функції корисності на k-ій ітерації Монте-Карло;

X(k) - матриця значень нормалізованих критеріїв (або вектор параметрів) на k-ій ітерації;

w - вектор ваг критеріїв;

N - кількість ітерацій (сценаріїв) моделювання;

k - індекс поточної ітерації.

Генерація випадкових величин (Monte Carlo симуляція)

У системі для моделювання невизначеності використовується підхід Монте-Карло, який дозволяє отримати не одне фіксоване значення корисності, а розподіл можливих результатів для кожної альтернативи.

Технічно генерація випадкових значень реалізована через scipy.stats.multivariate_normal.rvs. Цей інструмент створює багато випадкових сценаріїв для кожної альтернативи, імітуючи можливі коливання критеріїв (якість, вартість, ризик тощо).

Прийняті припущення:

У поточній версії моделі використовується спрощене припущення, що всі критерії є статистично незалежними. Це реалізовано через діагональну коваріаційну матрицю, де варіативність кожного критерію задається окремо і не впливає на інші.

Практично це означає, що:

- зміна одного показника не змінює поведінку інших;
- усі критерії мають однаковий рівень випадкових коливань;
- модель не враховує приховані взаємозв’язки між метриками.

Таке спрощення використовується свідомо, оскільки:

- воно зменшує складність обчислень;
- робить модель стабільною при невеликій кількості даних;
- відповідає ситуації, коли реальні кореляції між метриками невідомі або не зафіксовані.

Можливе розширення моделі

У реальних R&D-системах критерії часто взаємопов’язані. Наприклад, підвищення якості моделі може збільшувати вартість, а зменшення затримки - впливати на продуктивність.

Щоб врахувати це, модель можна розширити за рахунок введення кореляцій між критеріями. Це дозволяє симулювати більш реалістичну поведінку системи, де зміни одного параметра впливають на інші.

Основні підходи до покращення:

- Використання історичних даних

Кореляції можна оцінити на основі результатів попередніх експериментів і логів системи.

- Експертне налаштування

Бізнес-аналітики або ML-інженери можуть задавати логічні зв’язки між

метриками.

- Комбінований підхід

Поєднання статистичних даних і експертних припущень для формування більш стабільної моделі.

У результаті модель переходить від простого припущення незалежності до більш реалістичного сценарію, де критерії взаємодіють між собою, що підвищує точність оцінки ризиків і якості рішень.

Контроль числової стабільності

У процесі реалізації стохастичної моделі важливим етапом є забезпечення числової стабільності, оскільки результати Монте-Карло симуляцій можуть виходити за межі допустимого діапазону або містити некоректні значення.

Для цього застосовується механізм обмеження значень (clipping), який приводить усі згенеровані дані до інтервалу від 0 до 1. Це особливо важливо, оскільки нормалізовані критерії інтерпретуються як відносні показники якості альтернативи.

Основні проблеми, які вирішуються на цьому етапі:

- вихід значень за допустимі межі через випадкову варіативність;
- поява некоректних чисел (NaN) у результаті обчислень;
- ризик числового переповнення при великій кількості симуляцій.

Застосування np.clip забезпечує контрольовану поведінку моделі та гарантує, що всі проміжні та фінальні значення залишаються в інтерпретованому діапазоні. Це підвищує стійкість алгоритму та робить результати придатними для подальшого аналізу.

Реалізація Байєсівського оновлення

Байєсівський компонент моделі використовується для уточнення ймовірності досягнення мінімального рівня корисності на основі результатів Монте-Карло симуляції.

У даній реалізації результати симуляції інтерпретуються як серія спосте-

режень, де кожна ітерація відповідає умовному “експерименту”: чи перевищує згенерована корисність заданий поріг U min.

Таким чином, Монте-Карло виступає як наближений біноміальний процес:

- «успіх» - перевищення порогу корисності;
- «невдача» - значення нижче порогу.

На основі частоти таких подій формується початкова оцінка ймовірності, яка далі уточнюється через Байєсівське оновлення з використанням апріорного розподілу.

Особливістю реалізації є використання так званих “дробових успіхів”. Оскільки Monte Carlo повертає не дискретні події, а ймовірнісну оцінку, кількість успіхів інтерпретується як частка від загальної кількості симуляцій. Це дозволяє плавно переходити від емпіричної частоти до параметрів апостеріорного розподілу без втрати інформації.

Такий підхід має кілька переваг:

- дозволяє використовувати повну інформацію про симуляції, а не лише дискретні результати;
- зменшує шум у оцінці ймовірностей;
- забезпечує стабільність при великій кількості альтернатив.

У результаті модель формує апостеріорну оцінку ймовірності перевищення порогу корисності, яка враховує як симуляційні дані, так і початкові припущення, задані експертом.

Реалізація генетичного алгоритму

Генетичний алгоритм у даній системі використовується як евристичний метод пошуку найкращої альтернативи в умовах багатокритеріальності та обмежень за ризиком. Його основна роль - не точне оптимізаційне рішення, а ефективне дослідження простору можливих варіантів.

Представлення рішення

У межах задачі кожна альтернатива розглядається як окремий «ген». Тоб-

то:

- один ген = одна бізнес-альтернатива (наприклад, модель або стратегія);
- хромосома фактично вироджується до одного елемента, оскільки кожна альтернатива вже є повністю описаним рішенням.

Таким чином, класична структура “набір параметрів” замінюється на вибір з дискретної множини готових варіантів.

Популяція

Популяція формується як набір випадково обраних альтернатив із загального списку. Це дозволяє стартувати алгоритм без додаткового припущення про початкову якість рішень і забезпечує різноманітність пошуку.

Фітнес-функція

Фітнес-функція побудована на поєднанні двох компонентів:

очікувана корисність (отримана з Монте-Карло симуляції);

обмеження за ризиком (ймовірність перевищення порогу U min).

Альтернативи, які не задовольняють ризикове обмеження, отримують нульову або близьку до нуля придатність. Це дозволяє одразу відсікати неприйнятні рішення ще на рівні еволюційного відбору.

Особливості реалізації

Класичні генетичні алгоритми зазвичай включають оператори кросоверу та мутації. У даній реалізації використовується спрощена схема:

- кросовер відсутній
- основним оператором є мутація

Мутація полягає у випадковій заміні поточної альтернативи на іншу з множини допустимих варіантів.

Причина такого підходу полягає у природі задачі: простір рішень є дискретним і не має внутрішньої параметризації. Кожна альтернатива є завершеним об’єктом, а не набором ознак. Тому класичний кросовер, який передбачає комбі-

нування параметрів двох рішень, втрачає зміст і не дає інтерпретованого результату.

Використання мутації як основного механізму дозволяє:

- зберегти цілісність альтернатив;
- забезпечити випадковий пошук у просторі рішень;
- уникнути некоректних “гібридних” альтернатив.

Обчислювальна складність

Обчислювальна складність системи визначається двома основними компонентами: Монте-Карло симуляцією та генетичним алгоритмом.

Монте-Карло симуляція

Для кожної альтернативи виконується генерація N випадкових сценаріїв, кожен з яких включає m критеріїв. Для n альтернатив загальна складність формула має вигляд:

$$O(n * N * m)$$

де n - кількість альтернатив;

N - кількість симуляцій;

m - кількість критеріїв.

Цей компонент є найбільш обчислювально затратним, оскільки працює з великим числом ітерацій.

Генетичний алгоритм

Генетичний алгоритм виконується протягом G поколінь із популяцією розміру P. На кожному кроці обчислюється фітнес для всіх елементів популяції.

Складність оцінюється за формулою:

$$O(G * P)$$

де G - кількість поколінь;

P - розмір популяції.

Оскільки кожна ітерація працює з дискретним набором альтернатив без складних операцій кросоверу або багатовимірної оптимізації, обчислювальна

складність залишається лінійною відносно параметрів алгоритму.

Особливості Streamlit-реалізації

Інтерфейс системи реалізовано на базі Streamlit як легкого інструменту для побудови інтерактивних аналітичних застосунків. Такий підхід дозволяє швидко створити робочий прототип системи підтримки прийняття рішень без використання повноцінного веб-фреймворку.

Ключовою особливістю є реактивна модель виконання: будь-яка зміна параметрів у інтерфейсі автоматично запускає повний перерахунок моделі. Це забезпечує динамічну взаємодію користувача з аналітичною системою без необхідності ручного оновлення або перезапуску.

Sidebar використовується як основна панель керування параметрами. Через неї задаються ваги критеріїв, рівні ризику, параметри симуляції та налаштування оптимізаційних алгоритмів. Така організація інтерфейсу розділяє аналітичну частину (основне вікно) та керування моделлю (бічна панель), що покращує структурованість взаємодії.

Важливою перевагою Streamlit є можливість реалізації повноцінного інтерактивного аналітичного середовища без окремого фронтенду. Уся логіка, візуалізація та обробка даних знаходяться в одному Python-середовищі, що значно спрощує розробку та супровід прототипу.

Інтеграція з ngrok

Оскільки Streamlit за замовчуванням запускає локальний сервер, він доступний лише в межах локального середовища або хост-машини. Це створює обмеження для демонстрації системи у зовнішньому середовищі, зокрема під час захисту або віддаленого перегляду.

Для вирішення цієї проблеми використовується ngrok, який створює захищений публічний тунель до локального сервера. Таким чином, локальний Streamlit-додаток стає доступним через зовнішній URL без необхідності деплою на окремий сервер.

Цей підхід дозволяє швидко розгортати демонстраційні версії системи та забезпечує доступність інтерфейсу з будь-якого пристрою, підключеного до інтернету.

Обмеження реалізації

Запропонована система, незважаючи на інтеграцію кількох аналітичних методів, має низку методологічних та технічних обмежень.

По-перше, у моделі закладено припущення про незалежність критеріїв. Це спрощує обчислення, але не відображає реальних взаємозв'язків між показниками ефективності в R&D-середовищі.

По-друге, використовується нормальний розподіл для моделювання невизначеності. У реальних даних розподіли можуть бути асиметричними або мати важкі хвости, що не враховується в поточній реалізації.

По-третє, ваги критеріїв задаються як статичні значення користувачем. Це означає відсутність механізму автоматичного навчання або адаптації пріоритетів на основі історичних даних чи результатів системи.

По-четверте, генетичний алгоритм працює у спрощеному дискретному просторі альтернатив і не виконує повноцінного комбінаторного пошуку. Відсутність кросоверу та обмежена структура популяції зменшують його дослідницьку потужність.

У сукупності ці обмеження визначають систему як прототипну реалізацію decision-support інструменту, який орієнтований на демонстрацію підходу, а не на промислову точність моделювання.

ВИСНОВКИ

У ході виконання кваліфікаційної магістерської роботи було досліджено особливості прийняття управлінських рішень у R&D-команді, що працює з технологіями обробки природної мови та аналітичними системами. Проведений аналіз показав, що існуючі підходи до вибору стратегій і технологічних альтернатив значною мірою базуються на експертних оцінках та фрагментарному використанні аналітики, що ускладнює забезпечення прозорості, повторюваності та обґрунтованості рішень. Виявлено ключові проблеми поточного процесу, серед яких - неузгодженість критеріїв оцінювання, недостатня інтеграція даних у процес прийняття рішень, складність оцінки ризиків та залежність від суб'єктивного досвіду учасників команди.

На основі результатів дослідження було сформульовано задачу вдосконалення технології прийняття рішень шляхом інтеграції принципів Data-Driven Management та методів теорії прийняття рішень. У роботі запропоновано концептуальну модель, яка поєднує багатокритеріальний аналіз, математичні методи оптимізації, оцінку ризиків і використання емпіричних даних для вибору оптимальної альтернативи в умовах невизначеності.

У межах роботи формалізовано процес вибору альтернатив через багатокритеріальну функцію корисності, що враховує показники якості моделей, витрат, ризиків, часових обмежень та прогнозованої ефективності. Застосування принципу максимізації очікуваної корисності дозволило перейти від інтуїтивного підходу до кількісно обґрунтованого механізму прийняття рішень. Додатково до оптимізаційної моделі інтегровано імовірнісні підходи, зокрема Monte Carlo симуляцію та Байєсівське оновлення, що забезпечує можливість оцінки стійкості альтернатив та аналізу ризиків в умовах неповної інформації.

Практичним результатом роботи стала програмна реалізація прототипу системи підтримки прийняття рішень із використанням Python та Streamlit. Реалізований застосунок включає модулі завантаження та перевірки даних, нормалізації критеріїв, розрахунку функції корисності, стохастичного моделювання, Байєсівської оцінки ризику, генетичного алгоритму пошуку оптимальної альтернативи та інтерактивної візуалізації результатів. Архітектура системи забезпечує можливість гнучкого налаштування критеріїв і параметрів моделі відповідно до потреб користувача.

Проведене тестування моделі на різних сценаріях підтвердило її працездатність та здатність адаптуватися до зміни вхідних умов. Отримані результати показали, що інтегрований підхід дозволяє підвищити прозорість і повторюваність управлінських рішень, покращити якість оцінки ризиків та знизити залежність від суб'єктивних факторів. Порівняння з існуючою практикою продемонструвало переваги моделі в аспектах формалізації процесу, системності аналізу та швидкості отримання аналітичних висновків.

Практична цінність роботи полягає у можливості використання запропонованої моделі як основи для створення внутрішніх decision-support систем у R&D та Data Science командах. Запропонований підхід може бути адаптований для задач пріоритезації функціональності, оцінки ML-моделей, планування експериментів та підтримки стратегічних управлінських рішень у середовищах із високим рівнем невизначеності.

Для подальшого розвитку моделі доцільним є розширення механізмів оцінки залежностей між критеріями, інтеграція реальних історичних даних для автоматичного налаштування параметрів, удосконалення евристичних алгоритмів пошуку та підключення системи до корпоративної BI-інфраструктури. Перспективним напрямом також є використання методів машинного навчання для адаптивного коригування ваг критеріїв і прогнозування ефективності альтернатив на основі накопичених даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Shmueli, G., Bruce, P. C., Stephens, M. L., & Patel, N. R. Data Mining for Business Analytics: Concepts, Techniques, and Applications with JMP Pro. – Hoboken, New Jersey : John Wiley & Sons, 2017. – ISBN 9781118877432. – URL: https://content.e-bookshelf.de/media/reading/L-7771267-0e2ce5cf22.pdf (дата звернення: 18.12.2025).
2. Mao, L. Maximum Expected Utility and Decision Making. // Blog. – 2021. – URL: https://leimao.github.io/blog/Maximum-Expected-Utility-Decision-Making (дата звернення: 18.12.2025).
3. Cao, G., Duan, Y., & Li, G. Linking Business Analytics to Decision Making Effectiveness: A Path Model Analysis. // Journal of Industrial Engineering and Management. – 2022. – Vol. 15, No. 2. – URL: https://doi.org/10.3926/jiem.3030 (дата звернення: 18.12.2025).
4. Hanna, S. D. Utility Function U= Square Root of Wealth, Expected Wealth (EW) and Expected Utility (EU) Based on 50% Chance of Zero Wealth and 50% Chance of Wealth = $10,000. // ResearchGate. – 1989. – URL: https://www.researchgate.net/figure/Utility-Function-U-Square-Root-of-Wealth-Expected-Wealth-EW-and-Expected-Utility_fig1_279197095 (дата звернення: 18.12.2025).
5. Hastie, T., Tibshirani, R., & Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2009 (2nd ed.). – URL: https://hastie.su.domains/ElemStatLearn/ (дата звернення: 18.12.2025).
6. Yin, J., & Fernández, V. A Systematic Review on Business Analytics. (Systematic review/working paper). – 2020. – URL: https://www.econstor.eu/bitstream/10419/261719/1/1741210682.pdf (дата звернення: 18.12.2025).

7. Kurpiela, S. et al. Linking business analytics affordances to corporate strategic planning outcomes. Information Systems and e-Business Management (Springer), 2024. – URL: https://link.springer.com/article/10.1007/s10257-023-00661-z (дата звернення: 18.12.2025).
8. Cappello, C., et al. Expected utility theory for monitoring-based decision making. (IEEE/conference/journal paper), 2016 – приклад застосування EUT у практичних інженерних задачах. – URL: https://strathprints.strath.ac.uk/55231/ (дата звернення: 18.12.2025).
9. Polson, N., Generative Bayesian Computation for Maximum Expected Utility – методи обчислення MEU з байєсівським підходом (MDPI, 2024). – URL: https://www.mdpi.com/1099-4300/26/12/1076 (дата звернення: 18.12.2025).
10. Cruz, R. N., Data-Driven Decision-Making in Marketing: A Systematic Review. Systems (MDPI), 2025 – огляд сучасних практик DDDM у маркетингу. – URL: https://www.mdpi.com/2079-8954/13/12/1114 (дата звернення: 18.12.2025).
11. Golionko, N. H. Зміна парадигми управління організацією на основі Data-Driven підходу. – матеріали міжнародної конференції/КНЕУ, 2023. (українською) – URL: https://ir.kneu.edu.ua/items/c5bbb6a1-8bf0-4b35-8375-8555d5f5ee07 (дата звернення: 18.12.2025).
12. Harvard Business School Online. The Advantages of Data-Driven Decision-Making. // HBS Online Blog. – 2024. – URL: https://online.hbs.edu/blog/post/data-driven-decision-making (дата звернення: 18.12.2025).
13. DataCamp Team. What is Business Analytics? Everything You Need to Know. // DataCamp Blog. – 2022. – URL: https://www.datacamp.com/blog/what-is-business-analytics (дата звернення: 18.12.2025).

14. Tull, D. Data-Driven Decision-Making: A Complete Guide in 2025. // TechnologyAdvice Blog. – 2025. – URL: https://technologyadvice.com/blog/business-intelligence/data-driven-decision-making/ (дата звернення: 18.12.2025).
15. GraphAware Team. Competing on Analytics with Knowledge Graphs #1. // GraphAware Blog. – 2022. – URL: https://graphaware.com/blog/know-what-you-know-competing-analytics-with-kg/ (дата звернення: 18.12.2025).
16. Data-Driven process management: як використовувати метрики, а не інтуїцію. // E5.ua Blog. – 2025. – URL: https://e5.ua/uk/blogpost-2/data-driven-process-management-yak-vykorystovuvaty-metryky-a-ne-intuyitsiyu (дата звернення: 18.12.2025).
17. Data-Driven підхід у HR: як аналітика ментального стану допомагає зберегти команди. – Рабінович, М. // ЛІГА.Блоги. – 31.07.2025. – URL: https://blog.liga.net/user/mrabinovych/article/57403 (дата звернення: 18.12.2025).
18. OpenAI. The State of Enterprise AI 2025 Report. – URL: https://openai.com/business/guides-and-resources/the-state-of-enterprise-ai-2025-report/ (дата звернення: 15.05.2026).
19. McKinsey & Company. The State of AI 2025. – URL: https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai (дата звернення: 15.05.2026).
20. State of Generative AI in the Enterprise Q4 2025. – URL: https://www.pertamapartners.com/insights/research/state-of-generative-ai-in-the-enterprise-q4-2025 (дата звернення: 15.05.2026).
21. Lex Data Labs. AI Total Cost of Ownership: The Hidden Cost of Inference. – URL: https://www.lexdatalabs.com/post/ai-total-cost-of-ownership-tco-the-hidden-cost-of-inference-2 (дата звернення: 15.05.2026).
22. The State of AI+BI Analytics Global 2025 Report. – URL:

https://www.strategy.com/software/survey (дата звернення: 15.05.2026).

23. Impact and Implications of Generative AI for Enterprise Architects in Agile Environments. – arXiv, 2025. – URL: https://arxiv.org/abs/2510.22003 (дата звернення: 15.05.2026).
24. van Dijk, P. AI-Driven Business Intelligence: Leveraging Predictive Analytics for Data-Driven Decision Making. // International Journal of AI, BigData, Computational and Management Studies. – 2024. – Vol. 5, No. 3. – URL: https://ijaibdcms.org/index.php/ijaibdcms/article/view/63 (дата звернення: 15.05.2026).
25. Davenport, T. H., Harris, J. G. Competing on Analytics: The New Science of Winning. – Boston : Harvard Business School Press, 2007. – URL: https://www.tomdavenport.com/book/competing-on-analytics/ (дата звернення: 15.05.2026).
26. Davenport, T. H., Bean, R. Five Key Trends in AI and Data Science for 2024. // MIT Sloan Management Review. – 2024. – URL: https://sloanreview.mit.edu/article/five-key-trends-in-ai-and-data-science-for-2024/ (дата звернення: 15.05.2026).
27. Schrage, M., Kiron, D. Intelligent Choices Reshape Decision-Making and Productivity. // MIT Sloan Management Review. – 2024. – URL: https://sloanreview.mit.edu/article/intelligent-choices-reshape-decision-making-and-productivity/ (дата звернення: 15.05.2026).
28. Schrage, M., et al. The Future of Strategic Measurement: Enhancing KPIs With AI. // MIT Sloan Management Review & Boston Consulting Group. – 2024. – URL: https://sloanreview.mit.edu/projects/the-future-of-strategic-measurement-enhancing-kpis-with-ai/ (дата звернення: 15.05.2026).
29. Gartner, Inc. Gartner Identifies Top Trends in Data and Analytics for 2025. // Gartner Newsroom. – 2025. – URL: https://www.gartner.com/en/newsroom/press-releases/2025-03-05-gartner-identifies-top-trends-in-data-and-analytics-for-2025 (дата звернення:

15.05.2026).

30. Gartner, Inc. Gartner Announces the Top Data & Analytics Predictions for 2025 and Beyond. // Gartner Newsroom. – 2025. – URL: https://www.gartner.com/en/newsroom/press-releases/2025-06-17-gartner-announces-top-data-and-analytics-predictions (дата звернення: 15.05.2026).
31. Deloitte Insights. Decision-Making with AI: Human Capital Trends 2026. // Deloitte Insights. – 2026. – URL: https://www.deloitte.com/us/en/insights/topics/talent/human-capital-trends/2026/decision-making-with-ai.html (дата звернення: 15.05.2026).
32. Prawira, et al. Exploring the Role of Machine Learning and Big Data Analytics in Enhancing Decision-Making Processes: A Systematic Literature Review. // JOIV: International Journal on Informatics Visualization. – 2025. – URL: https://joiv.org/index.php/joiv/article/view/3244 (дата звернення: 15.05.2026).
33. Khawaldeh, K., Awamleh, F. T., Al-Momani, A. M. Data-Driven Strategic Decisions: Leveraging Business Analytics and Big Data to Improve Decision-Making Insights in International Organizations. // Journal of Project Management. – 2025. – Vol. 10, No. 1. – P. 61–70. – URL: http://growingscience.com/beta/jpm/7354-data-driven-strategic-decisions-leveraging-business-analytics-and-big-data-to-improve-decision-making-insights-in-the-international-organizations.html (дата звернення: 15.05.2026).
34. Al-Refaie, A., et al. A Systematic Review of the Benefits and Challenges of Data Analytics in Organizational Decision Making. // International Journal of Advanced Computer Science and Applications (IJACSA). – 2025. – Vol. 16, No. 2. – URL: https://thesai.org/Publications/ViewPaper?Volume=16&Issue=2&Code=ija

csa&SerialNo=21 (дата звернення: 15.05.2026).

35. Kurniawan, R., et al. Prescriptive Analytics Systems Revised: A Systematic Literature Review from an Information Systems Perspective. // Information Systems and e-Business Management. – Springer, 2024. – URL: https://link.springer.com/article/10.1007/s10257-024-00688-w (дата звернення: 15.05.2026).
36. Fanelli, S., et al. Big Data Analysis for Decision-Making Processes: Challenges and Opportunities for the Management of Health-Care Organizations. // Management Research Review. – 2023. – Vol. 46, No. 3. – P. 369–389. – URL: https://doi.org/10.1108/MRR-09-2021-0648 (дата звернення: 15.05.2026).
37. Weller, J., et al. Towards a Systematic Approach for Prescriptive Analytics Use Cases in Smart Factories. // Machine Learning for Cyber-Physical Systems. ML4CPS 2023. – Springer, Cham, 2024. – URL: https://link.springer.com/chapter/10.1007/978-3-031-47062-2_9 (дата звернення: 15.05.2026).
38. Adebiaye, R., Alshami, M., Owusu, T. Machine Learning Models for Extrapolative Analytics as a Panacea for Business Intelligence Decisions. // International Journal of Engineering Technologies and Management Research. – 2023. – Vol. 10, No. 6. – P. 13–32. – URL: https://doi.org/10.29121/ijetmr.v10.i6.2023.1333 (дата звернення: 15.05.2026).
39. Kyivstar Business Hub. Штучний інтелект в Україні: як розвивається галузь. // Kyivstar Business Hub. – 2025. – URL: https://hub.kyivstar.ua/articles/galuzevi-trendi-shtuchnyj-intelekt-v-ukrayini-yak-rozvivayetsya-galuz (дата звернення: 15.05.2026).
40. Двуліт, З. П. та ін. Вплив бізнес-аналітики на корпоративне управління. // Вісник Національного університету «Львівська політехніка». – 2024. – URL: https://science.lpnu.ua/sites/default/files/journal-

paper/2024/dec/37228/dvulit.pdf (дата звернення: 15.05.2026).

41. Economyandsociety. Бізнес-аналітика як інструмент обробки даних. // Економіка та суспільство. – 2024. – Вип. 62. – URL: https://economyandsociety.in.ua/index.php/journal/article/view/3889 (дата звернення: 15.05.2026).
42. Economyandsociety. Роль аналітико-інструментальних засобів у прогнозуванні розвитку підприємства в умовах зростаючої цифровізації. // Економіка та суспільство. – 2025. – URL: https://economyandsociety.in.ua/index.php/journal/article/view/5690 (дата звернення: 15.05.2026).
43. Economyandsociety. Цифрова трансформація управління підприємством: світові тренди та українська практика. // Економіка та суспільство. – 2025. – Вип. 79. – URL: https://economyandsociety.in.ua/index.php/journal/article/view/6747 (дата звернення: 15.05.2026).
44. Економічна правда. Змінюйся або помри: як штучний інтелект став невід'ємною частиною світової економіки. // Економічна правда. – 2024. – URL: https://epravda.com.ua/publications/2023/07/24/702480/ (дата звернення: 15.05.2026).
45. Андрощук, Г. О. Політика і стратегії розвитку штучного інтелекту в країнах світу. // Наука, технології, інновації. – 2023. – № 1. – URL: https://nti.ukrintei.ua/wp-content/uploads/2024/03/Андрощук_1-2023.pdf (дата звернення: 15.05.2026).
46. Davenport, T. H., Bean, R. Five Trends in AI and Data Science for 2025. // MIT Sloan Management Review. – 2025. – URL: https://sloanreview.mit.edu/article/five-trends-in-ai-and-data-science-for-2025/ (дата звернення: 15.05.2026).
47. Bari, M. H., Ara, A. The Impact of Machine Learning on Prescriptive Analytics for Optimized Business Decision-Making. // SSRN. – 2024. –

URL: https://doi.org/10.2139/ssrn.5050060 (дата звернення: 15.05.2026).

48. Gartner, Inc. What Is Data and Analytics: Everything You Need to Know. // Gartner. – 2024. – URL: https://www.gartner.com/en/topics/data-and-analytics (дата звернення: 15.05.2026).
49. Ransbotham, S., et al. Artificial Intelligence and Business Strategy 2025 Report. // MIT Sloan Management Review & Boston Consulting Group. – 2025. – URL: https://sloanreview.mit.edu/big-ideas/artificial-intelligence-business-strategy/ (дата звернення: 15.05.2026).
50. Coherent Solutions. The Future of Data Analytics: Trends in 7 Industries [2025]. // Coherent Solutions Blog. – 2025. – URL: https://www.coherentsolutions.com/insights/the-future-and-current-trends-in-data-analytics-across-industries (дата звернення: 15.05.2026).
51. Hydrogen BI. Data-Driven Decision Making (2025): Latest Stats, Trends & Benchmarks. // Hydrogen BI. – 2025. – URL: https://www.hydrogenbi.com/data-driven-decision-making-2025-stats (дата звернення: 15.05.2026).
52. Лявинець, Г., Люлька, О., Ткачук, Ю. Інтелектуальні, орієнтовані на знання технології у адаптивному менеджменті готельно-ресторанного бізнесу. // Економіка та суспільство. – 2024. – № 67. – URL: https://doi.org/10.32782/2524-0072/2024-67-91 (дата звернення: 15.05.2026).
53. Economyandsociety. Особливості аналізування бізнес-процесів підприємств у сучасних умовах. // Економіка та суспільство. – 2024. – Вип. 62. – URL: https://economyandsociety.in.ua/index.php/journal/article/view/3963 (дата звернення: 15.05.2026).

ДОДАТКИ

Додаток А. Перелік шаблонів документів, за заповнення яких відповідальний бізнес-аналітик

Шаблон: User Story

---

1. Загальна інформація

ID:

Назва:

Епік / Feature:

Пріоритет: (High / Medium / Low)

Статус: (Draft / In Progress / Done)

Автор:

Дата створення:

---

2. Опис (класичний формат)

Як [тип користувача / роль]

Я хочу [дія / функціонал]

Щоб [бізнес-цінність / результат]

---

3. Детальний опис

Короткий опис функціоналу:

Контекст використання:

---

4. Бізнес-цінність

Яку проблему вирішує:

Очікуваний результат:

Вплив на користувача / бізнес:

---

5. Acceptance Criteria (коротко)

[ ]

[ ]

[ ]

---

6. Декомпозиція (за потреби)

Підсторі / задачі:

---

7. Дані (якщо застосовно)

Джерела даних:

Вхідні дані:

Вихідні дані:

---

8. Залежності

Інші сторі / задачі:

Системи / інтеграції:

---

9. Обмеження / припущення

Обмеження:

Припущення:

---

10. Edge cases (граничні випадки)

---

11. INVEST перевірка

Критерій	Опис	Статус
Independent	Незалежна від інших	✔ / ✘
Negotiable	Може обговорюватись	✔ / ✘
Valuable	Має цінність	✔ / ✘
Estimable	Можна оцінити	✔ / ✘
Small	Достатньо мала	✔ / ✘
Testable	Можна протестувати	✔ / ✘

---

12. Додатково

Коментарі:

Посилання (документи, макети):

Шаблон: Acceptance Criteria

---

1. Загальна інформація

ID User Story:

Назва User Story:

Епік / Feature:

Автор:

Дата створення:

Версія:

---

2. Опис мети Acceptance Criteria

Коротко описати, що саме ці критерії перевіряють:

---

3. Acceptance Criteria (структурований список)

3.1 Functional Criteria (FR)

[ ]

[ ]

[ ]

3.2 Non-Functional Criteria (NFR)

- ☐ Performance (час виконання / обробки):
- ☐ Reliability (стабільність / обробка помилок):
- ☐ Security (доступи / ролі):
- ☐ Scalability (навантаження / масштабування):

3.3 Business Rules

[ ]

[ ]

[ ]

---

4. Pre-Conditions (передумови)

Умови, які мають бути виконані ДО запуску функціоналу:

---

5. Post-Conditions (післяумови)

Стан системи ПІСЛЯ виконання сценарію:

---

6. Validation Rules (правила перевірки)

---

7. Edge Cases (граничні сценарії)

Що відбувається при відсутніх даних:

Що відбувається при дублях:

Що відбувається при некоректному форматі:

---

8. Acceptance Scenarios (Gherkin)

Scenario 1:

Given

When

Then

Scenario 2:

Given

When

Then

---

9. Notes / Assumptions

Припущення:

Обмеження:

Коментарі:

---

10. Залежності

Системи / сервіси:

Інші user stories:

Шаблон: Бізнес-вимоги / Product Requirements (PRD)

---

1. Загальна інформація

ID документа:

Назва продукту / функціоналу:

Епік / ініціатива:

Замовник / стейкхолдер:

Автор (BA):

Дата створення:

Версія:

Статус: (Draft / Review / Approved)

---

2. Контекст і проблема (Problem Statement)

Опис проблеми, яку вирішує продукт:

Поточна ситуація:

Проблема:

Наслідки проблеми:

Чому це важливо зараз:

---

3. Мета продукту (Product Goals)

Бізнес-цілі:

Продуктові цілі:

Очікуваний ефект:

KPI / success metrics (якщо є):

---

4. Опис рішення (High-level Solution)

Загальний опис функціоналу без технічної деталізації:

Основна ідея:

Ключові можливості:

Основні сценарії використання:

---

5. Обсяг функціоналу (Scope)

5.1 In Scope

5.2 Out of Scope

---

6. User Personas

Роль	Опис	Потреби

---

7. Use Cases (сценарії використання)

Use Case 1:

Use Case 2:

Use Case 3:

---

8. Бізнес-вимоги (Business Requirements)

BR-1:

BR-2:

BR-3:

(Кожна вимога повинна описувати *що потрібно бізнесу*, а не як це реалізувати)

---

9. Функціональні очікування (High-level Features)

Функція 1:

Функція 2:

Функція 3:

---

10. Обмеження та припущення

Технічні обмеження:

Бізнес-обмеження:

Припущення:

---

11. Залежності

Внутрішні системи:

Зовнішні інтеграції:

Команди / сервіси:

---

12. Ризики

Ризик 1:

Ризик 2:

Ризик 3:

---

13. Критерії успіху (Success Criteria)

---

14. Додатки

Посилання на макети:

Діаграми:

Додаткові матеріали:

Шаблон: Functional Requirements (FR)

---

1. Загальна інформація

ID документа:

Назва функціоналу:

Пов’язаний PRD / User Stories:

Епік / Feature:

Автор (BA):

Дата створення:

Версія:

Статус: (Draft / Review / Approved)

---

2. Опис функціоналу (Scope of Functionality)

Короткий опис, що саме система повинна робити:

---

3. Functional Requirements (деталізований список)

Кожна вимога має бути атомарною (одна дія = одна вимога)

FR-1:

FR-2:

FR-3:

FR-4:

---

4. Опис поведінки системи (System Behavior)

Для кожного ключового сценарію:

4.1 Сценарій 1

Вхідні дані:

Обробка:

Вихід:

4.2 Сценарій 2

Вхідні дані:

Обробка:

Вихід:

---

5. Business Rules (бізнес-логіка)

BR-1:

BR-2:

BR-3:

---

6. Вхідні дані (Input Requirements)

Типи даних:

Формати:

Джерела:

Валідація:

---

7. Вихідні дані (Output Requirements)

Формат результату:

Структура даних:

Куди передається:

---

8. Інтеграції

Системи:

API:

Протоколи обміну:

Частота взаємодії:

---

9. Обробка помилок (Error Handling)

Типи помилок:

Поведінка системи:

Логування:

---

10. Edge Cases

Випадок 1:

Випадок 2:

Випадок 3:

---

11. Залежності

Інші модулі:

Дані:

Сервіси:

---

12. Обмеження

Технічні:

Бізнесові:

Архітектурні:

---

13. Прийнятні результати (Expected Outcomes)

Шаблон: Data Requirements

---

1. Загальна інформація

ID документа:

Назва функціоналу / системи:

Пов’язаний PRD / FR:

Епік / модуль:

Автор (BA):

Дата створення:

Версія:

Статус: (Draft / Review / Approved)

---

2. Мета документа

Короткий опис, для чого визначаються вимоги до даних:

---

3. Джерела даних (Data Sources)

Опис усіх вхідних систем/каналів:

Джерело	Тип даних	Формат	Частота

---

4. Типи даних (Data Types)

Структуровані дані:

Неструктуровані дані:

Напівструктуровані дані:

---

5. Вхідні дані (Input Data Specification)

Для кожного типу даних:

5.1 Поля даних

Поле	Тип	Обов’язкове	Опис

5.2 Формати

JSON / XML / CSV / Text / Audio / Image:

Стандарти:

---

6. Вихідні дані (Output Data Specification)

Формат результату:

Структура:

Куди передається:

Приклад структури:

---

7. Валідація даних (Data Validation Rules)

Правила перевірки:

Обов’язкові поля:

Форматні обмеження:

Бізнес-валідації:

---

8. Мапінг даних (Data Mapping)

Source Field Target Field Transformation Rule

---

9. Обробка відсутніх / некоректних даних

Missing values:

Invalid format:

Partial data:

Recovery strategy:

---

10. Якість даних (Data Quality Rules)

Дублікати:

Повнота:

Актуальність:

Консистентність:

---

11. Унікальні ідентифікатори (IDs)

Тип ID:

Правила генерації:

Гарантія унікальності:

Причини вибору підходу:

---

12. Обмеження на дані

Об’єм даних:

Швидкість обробки:

Обмеження джерел:

---

13. Залежності

Зовнішні API:

Внутрішні системи:

Моделі / сервіси обробки:

---

14. Безпека та доступ

Рівні доступу:

Чутливі дані:

Маскування / анонімізація:

---

15. Приклади даних (Optional)

Input example:

Output example:

Шаблон: Data Quality / Data Issues

---

1. Загальна інформація

ID документа:

Назва системи / модуля:

Пов’язаний Data Requirements / FR:

Епік / ініціатива:

Автор (BA):

Дата створення:

Версія:

Статус: (Draft / In Review / Approved)

---

2. Мета документа

Опис проблем якості даних та визначення правил їх виявлення, обробки та запобігання:

---

3. Категорії проблем якості даних

Категорія	Опис	Приклад
Дублікати	Повторні записи	однакові записи з різними timestamps
Відсутні дані	Missing values	null у критичних полях
Некоректний формат	Format mismatch	неправильна дата / email
Розсинхронізація	Data mismatch між системами	різні значення в API та DB
Застарілі дані	Outdated records	неактуальна інформація
Неповні дані	Partial records	відсутні обов’язкові поля

---

4. Виявлені проблеми (Data Issues Log)

ID	Опис проблеми	Джерело	Частота	Вплив

---

5. Дублікати (Duplicate Handling Rules)

Критерії визначення дублікатів:

Поля для порівняння:

Правило вибору “головного” запису:

Стратегія об’єднання / видалення:

---

6. Відсутні дані (Missing Data Handling)

Обов’язкові поля:

Стратегія обробки (ignore / default / reject):

Логіка fallback:

---

7. Некоректні дані (Invalid Data Handling)

Типи помилок:

Правила валідації:

Поведінка системи:

---

8. Розсинхронізація даних (Data Sync Issues)

Джерела конфліктів:

Пріоритет джерел:

Частота синхронізації:

Правила вирішення конфліктів:

---

9. Метрики якості даних (Data Quality Metrics)

% дублікатів:

% відсутніх даних:

% помилкових записів:

Latency синхронізації:

Повнота даних:

---

10. Причини виникнення проблем

Проблеми інтеграцій:

Людський фактор:

Помилки парсингу / OCR:

Обмеження API:

---

11. Правила моніторингу

Логи перевірок:

Алерти:

Частота перевірок:

Інструменти моніторингу:

---

12. Рішення / Remediation Actions

Автоматичне очищення даних:

Ручна валідація:

Backfill процеси:

Покращення pipeline:

---

13. Відповідальність

Команда Data:

BA роль:

Dev / QA роль:

---

14. Залежності

Системи-джерела:

API інтеграції:

Моделі обробки даних:

Шаблон: User Guides

---

1. Загальна інформація

ID документа:

Назва системи / продукту:

Цільова аудиторія: (ролі користувачів)

Пов’язаний функціонал / модуль:

Автор (BA):

Дата створення:

Версія:

Статус: (Draft / Review / Approved)

---

2. Мета документа

Короткий опис мети інструкції:

---

3. Опис системи / функціоналу

Загальний опис того, що робить система:

Основні можливості:

Основні сценарії використання:

---

4. Ролі користувачів

Роль	Опис	Доступні функції

---

5. Передумови використання

Необхідні доступи:

Логін / авторизація:

Технічні вимоги:

---

6. Основні сценарії використання

6.1 Сценарій 1

Назва:

Крок 1:

Крок 2:

Крок 3:

Очікуваний результат:

6.2 Сценарій 2

Назва:

Крок 1:

Крок 2:

Крок 3:

Очікуваний результат:

---

7. Робота з функціоналом (детально)

7.1 Опис інтерфейсу / функції

Елемент 1:

Елемент 2:

Елемент 3:

7.2 Пояснення дій користувача

---

8. Помилки та їх обробка

Типові помилки:

Що робити користувачу:

Повідомлення системи:

---

9. Обмеження

Що користувач НЕ може робити:

Системні обмеження:

---

10. FAQ (часті питання)

Q1:

A1:

Q2:

A2:

---

11. Залежності

Системи:

Ролі доступу:

Інтеграції:

---

12. Додаткові матеріали

Посилання:

Скриншоти / макети:

Пов’язані документи:

Додаток Б. Код розробленої програми

```
!pip install streamlit

# встановлюємо pyngrok, щоб створити публічний тунель
!pip install pyngrok -q

import streamlit as st
from pyngrok import ngrok

# введіть ваш ngrok authtoken тут
# ви можете отримати його, зареєструвавшись на https://ngrok.com/
# зберігайте його в секретах Colab для кращої безпеки, якщо використовуєте багато разів.
# наприклад, ngrok.set_auth_token(userdata.get('NGROK_AUTH_TOKEN'))
NGROK_AUTH_TOKEN = "«««««««««««««««««««««««««««««««««««««««««««««««««««««"
ngrok.set_auth_token(NGROK_AUTH_TOKEN)

%%writefile app_improved.py
import streamlit as st
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns # import seaborn for heatmaps
from scipy.stats import multivariate_normal
import random # added for Genetic Algorithm

st.set_page_config(page_title="Decision Model", layout="wide")

st.title("Система підтримки управлінських рішень")

# -------------------------
# 1. DATA INPUT
# -------------------------
st.sidebar.header("Панель введення даних")

# CSV Instructions as an expandable section in the sidebar
with st.sidebar.expander("Інструкції щодо CSV формату"):
    st.subheader("Формат CSV-файлу")
    st.markdown("Будь ла ласка, переконайтеся, що ваш CSV-файл має наступний формат (заголовки чутливі до регістру):")
    st.code("alternative,quality,cost,latency,roi,risk")
    st.markdown("Де:")
    st.markdown("- ««alternative»«: Унікальна назва альтернативи (текст). Цей стовпець буде використовуватися як індекс.")
    st.markdown("- ««quality»«: Показник якості альтернативи (число).")
    st.markdown("- ««cost»«: Вартість альтернативи (число).")
```

```
   st.markdown("- ««latency»«: Затримка або час виконання (число).")
    st.markdown("- ««roi»«: Повернення інвестицій (число).")
    st.markdown("- ««risk»«: Рівень ризику (число).")
    st.markdown("««Приклад CSV-файлу:»«")
    st.code("""alternative,quality,cost,latency,roi,risk
Model_X,0.9,1200,100,1.3,0.1
Model_Y,0.7,1800,250,1.0,0.4
Model_Z,0.8,900,80,1.4,0.05""")
    st.markdown("Переконайтеся, що всі значення є числовими, окрім стовпця `alternative`.")

# можливість завантаження CSV
uploaded_file = st.sidebar.file_uploader("Завантажте CSV файл з альтернативами", type=["csv"])

if uploaded_file is not None:
    try:
        data = pd.read_csv(uploaded_file).set_index("alternative")
        st.success("CSV файл успішно завантажено!")
    except Exception as e:
        st.error(f"Помилка при читанні CSV файлу: {e}")
        st.stop()
else:
    # хардкодовані дані, якщо файл не завантажено
    st.sidebar.info("Якщо ви не завантажили CSV, буде використано демонстраційний набір даних.")
    data = pd.DataFrame({
        "alternative": ["Model_A", "Model_B", "Model_C"],
        "quality": [0.85, 0.88, 0.82],
        "cost": [1000, 1500, 700],
        "latency": [120, 200, 90],
        "roi": [1.2, 1.5, 1.1],
        "risk": [0.2, 0.35, 0.15]
    }).set_index("alternative")


st.subheader("Таблиця альтернатив")
st.dataframe(data)

# задаємо критерії та ваги
st.sidebar.subheader("Ваги критеріїв")
st.sidebar.markdown(
    "<span style='font-size: small;'>Встановіть важливість (вагу) для кожного критерію. "
    "Вищі значення означають більший вплив цього критерію на загальну корисність альтернативи. "
    "Сума всіх ваг автоматично нормалізується до 1.</span>" , unsafe_allow_html=True)
weights = {}
for col in data.columns:
    weights[col] = st.sidebar.slider(f"Вага для: {col}", 0.0, 1.0, 0.2, key=f"weight_slider_{col}")

# нормалізація ваг
total_weights = sum(weights.values())
if total_weights == 0:
```

```
st.sidebar.warning("Сума ваг не може бути нулем. Будь ласка, встановіть ваги.")
    st.stop()
weights = {k: v / total_weights for k, v in weights.items()}
st.sidebar.write("Нормалізовані ваги:", {k: f"{v:.2f}" for k, v in weights.items()})


# --- dynamic criterion type selection ---
st.sidebar.subheader("Вибір типу критеріїв (Benefit/Cost)")
st.sidebar.markdown(
    "<span style='font-size: small;'>Вкажіть, чи є кожен критерій показником, який ми максимізуємо
(Benefit) "
    "або мінімізуємо (Cost).</span>", unsafe_allow_html=True
)
criterion_types = {}
for col in data.columns:
    default_index = 0 # Default to Benefit
    # suggest initial defaults based on common sense for the hardcoded data
    if col in ["cost", "latency", "risk"]:
        default_index = 1 # Default to Cost
    criterion_types[col] = st.sidebar.selectbox(
        f"Критерій '{col}' є:",
        ["Benefit", "Cost"],
        index=default_index,
        key=f"type_select_{col}"
    )

benefit_criteria = [col for col, type_ in criterion_types.items() if type_ == "Benefit"]
cost_criteria = [col for col, type_ in criterion_types.items() if type_ == "Cost"]

st.sidebar.write("Критерії-вигоди (Benefit):", benefit_criteria)
st.sidebar.write("Критерії-вартості (Cost):", cost_criteria)
# --- end dynamic criterion type selection ---


# параметри сценарію
st.sidebar.subheader("Параметри сценарію")
st.sidebar.markdown(
    "<span style='font-size: small;'>Встановіть порогові значення та параметри симуляції для оцінки
ризиків та доцільності альтернатив.</span>", unsafe_allow_html=True
)
U_min = st.sidebar.slider("Мінімальна допустима корисність (U_min)", 0.0, 1.0, 0.5)
st.sidebar.markdown(
    "<span style='font-size: small;'>««U_min»«: Мінімальне значення корисності, яке повинна досягти
альтернатива, "
    "щоб вважатися прийнятною. Значення від 0 (найменш корисна) до 1 (найбільш корисна).</span>",
unsafe_allow_html=True
)
alpha = st.sidebar.slider("Рівень допустимого ризику (α - ймовірність)", 0.0, 1.0, 0.8)
st.sidebar.markdown(
```

```
"<span style='font-size: small;'>««α (альфа)»«: Мінімальна ймовірність, з якою корисність "
    "альтернативи повинна перевищувати `U_min`, щоб вважатися "
    "доцільною. Значення від 0 до 1.</span>", unsafe_allow_html=True
)
std_dev = st.sidebar.slider("Стандартне відхилення для симуляції", 0.01, 0.3, 0.1)
st.sidebar.markdown(
    "<span style='font-size: small;'>««Стандартне відхилення»«: Визначає рівень невизначеності (розсію-
вання) "
    "значень критеріїв під час симуляції Монте-Карло. Вищі значення "
    "призводять до більшого розсіювання результатів корисності.</span>", unsafe_allow_html=True
)
n_sim = st.sidebar.slider("Кількість симуляцій Monte Carlo", 1000, 10000, 3000)
st.sidebar.markdown(
    "<span style='font-size: small;'>««Кількість симуляцій»«: Число ітерацій, які використовуються в "
    "симуляції Монте-Карло. Більша кількість симуляцій підвищує "
    "точність оцінок, але збільшує час обчислень.</span>", unsafe_allow_html=True
)

# -------------------------
# 2. NORMALIZATION & UTILITY CALCULATION
# -------------------------
# using dynamically defined benefit_criteria and cost_criteria here

def normalize(df, benefit_cols, cost_cols):
    norm = df.copy().astype(float)
    for col in df.columns:
        min_v = df[col].min()
        max_v = df[col].max()
        if max_v - min_v == 0:
            norm[col] = 0.5
        elif col in benefit_cols:
            norm[col] = (df[col] - min_v) / (max_v - min_v)
        elif col in cost_cols:
            norm[col] = (max_v - df[col]) / (max_v - min_v)
        else: # fallback: if not explicitly defined, treat as benefit
             norm[col] = (df[col] - min_v) / (max_v - min_v)
    return norm

norm_data = normalize(data, benefit_criteria, cost_criteria)

st.subheader("Нормалізовані значення критеріїв")
st.dataframe(norm_data)

def calculate_utility(df_normalized, weights_dict):
    weighted_values = df_normalized.mul(pd.Series(weights_dict, index=df_normalized.columns)) # align
weights by index
    return weighted_values.sum(axis=1)

U_deterministic = calculate_utility(norm_data, weights)
```

```
st.subheader("Детермінована корисність (U)")
st.write(U_deterministic)

# -------------------------
# 3. MONTE CARLO SIMULATION
# -------------------------
st.subheader("Симуляція Монте-Карло")

# adjusted to use the actual number of columns from norm_data
num_criteria = len(norm_data.columns)
corr_matrix = np.eye(num_criteria)
cov_matrix = corr_matrix « std_dev»«2

mc_results = {}
probabilities_mc = {} # store MC probabilities separately
for alt in norm_data.index:
    mean_values = norm_data.loc[alt].values
    simulated_values = multivariate_normal.rvs(
        mean=mean_values,
        cov=cov_matrix,
        size=n_sim
    )
    simulated_values = np.clip(simulated_values, 0, 1)
    simulated_utilities = np.dot(simulated_values, list(weights.values())) # use list of weights for
dot product
    mc_results[alt] = simulated_utilities
    probabilities_mc[alt] = np.mean(simulated_utilities >= U_min) # calculate MC probability

EU_mc = {alt: np.mean(utils) for alt, utils in mc_results.items()}
st.write("Очікувана корисність (Expected Utility) за Монте-Карло:")
st.write(EU_mc)

st.write("Ймовірність перевищення U_min за Монте-Карло:")
st.write({k: f"{v:.2%}" for k, v in probabilities_mc.items()})


# -------------------------
# 4. BAYESIAN RISK ASSESSMENT
# -------------------------
st.subheader("Байєсівська оцінка ризику")

st.sidebar.subheader("Параметри Байєсівського пріору")
st.sidebar.markdown(
    "<span style='font-size: small;'>Налаштуйте параметри Бета-розподілу, який використовується як "
    "пріор для Байєсівської оцінки ймовірності. Ці параметри "
    "відображають ваші початкові переконання щодо ймовірності "
    "перевищення `U_min` до проведення симуляції.</span>" , unsafe_allow_html=True)
prior alpha beta = st.sidebar.slider("Параметр α для Бета-пріору", 0.1, 10.0, 1.0,
key="prior_alpha_beta_slider")
```

```
st.sidebar.markdown(
    "<span style='font-size: small;'>«Параметр α (альфа)»: Чим вище α, тим більше ваги надається "
    "початковому переконанню, що ймовірність успіху (перевищення `U_min`) "
    "близька до 1.</span>", unsafe_allow_html=True
)
prior_beta_beta = st.sidebar.slider("Параметр β для Бета-пріору", 0.1, 10.0, 1.0,
key="prior_beta_beta_slider")
st.sidebar.markdown(
    "<span style='font-size: small;'>«Параметр β (бета)»: Чим вище β, тим більше ваги надається "
    "початковому переконанню, що ймовірність успіху (перевищення `U_min`) "
    "близька до 0.</span>", unsafe_allow_html=True
)

posterior_probabilities = {}
for alt, prob_mc in probabilities_mc.items(): # use probabilities_mc here
    # treat MC simulation results as "observed data" from a binomial process
    # n_sim is the number of trials, prob_mc is the observed success rate
    k_successes = prob_mc « n_sim # effective number of "successes"
    n_failures = n_sim - k_successes # effective number of "failures"

    # update Beta prior with observed data
    posterior_alpha = prior_alpha_beta + k_successes
    posterior_beta = prior_beta_beta + n_failures

    # mean of the posterior Beta distribution is the Bayesian estimate of the probability
    posterior_mean_prob = posterior_alpha / (posterior_alpha + posterior_beta)
    posterior_probabilities[alt] = posterior_mean_prob

st.write("Постеріорна ймовірність перевищення U_min (Байєсівська оцінка):")
st.write({k: f"{v:.2%}" for k, v in posterior_probabilities.items()})

# -------------------------
# 5. RISK CONSTRAINT & FEASIBILITY (USING BAYESIAN PROBABILITIES)
# -------------------------
st.subheader("Обмеження ризику та доцільність (з Байєсівською оцінкою)")

feasible_alternatives_bayes = {}

for alt, prob_bayes in posterior_probabilities.items(): # use posterior_probabilities here
    if prob_bayes >= alpha:
        feasible_alternatives_bayes[alt] = EU_mc[alt] # still use MC expected utility for ranking

st.write("Доцільні альтернативи (які відповідають обмеженням ризику за Байєсівською оцінкою):")
if feasible_alternatives_bayes:
    st.write(feasible_alternatives_bayes)
else:
    st.warning("Не знайдено доцільних альтернатив, які відповідають заданим критеріям ризику за Байєсівською оцінкою.")
```

```
# -------------------------
# 6. OPTIMIZATION
# -------------------------
st.subheader("Оптимізація: Вибір найкращої альтернативи")

best_alternative = None
if feasible_alternatives_bayes:
    best_alternative = max(feasible_alternatives_bayes, key=feasible_alternatives_bayes.get)
    st.success(f"Найкраща альтернатива: ««{best_alternative}»« (з очікуваною корисністю
{feasible_alternatives_bayes[best_alternative]:.3f})")
else:
    st.warning("Немає доцільних альтернатив для вибору.")

# -------------------------
# 6.5 GENETIC ALGORITHM OPTIMIZATION
# -------------------------
st.subheader("Оптимізація за допомогою Генетичного Алгоритму")
st.markdown(
    "<span style='font-size: small;'>Використання генетичного алгоритму для пошуку найкращої альтерна-
тиви "
    "на основі розрахованої корисності та ймовірності перевищення `U_min`."
    "</span>", unsafe_allow_html=True
)

st.sidebar.subheader("Параметри Генетичного Алгоритму")
st.sidebar.markdown(
    "<span style='font-size: small;'>Налаштуйте параметри генетичного алгоритму: "
    "розмір популяції визначає кількість альтернатив у кожному поколінні, "
    "кількість поколінь - глибину пошуку оптимального рішення, "
    "а швидкість мутації - рівень випадкових змін для уникнення локальних оптимумів.</span>",
    unsafe_allow_html=True
)
ga_pop_size = st.sidebar.slider("Розмір популяції ГА", 10, 200, 50, key="ga_pop_size_slider")
ga_generations = st.sidebar.slider("Кількість поколінь ГА", 10, 200, 50, key="ga_generations_slider")
ga_mutation_rate = st.sidebar.slider("Швидкість мутації ГА", 0.01, 0.5, 0.1,
key="ga_mutation_rate_slider")

def run_genetic_algorithm_for_selection(
    eu_mc_scores,
    posterior_probs_scores,
    alpha_threshold,
    all_alternatives_ids,
    population_size,
    num_generations,
    mutation_rate
):
    # fitness function: Maximize EU_mc, consider feasibility based on posterior probability
    def calculate_fitness(alternative_id):
        if alternative_id not in eu_mc_scores or alternative_id not in posterior_probs_scores:
```

```
            return 0.0 # should not happen if alternatives_list is correct

        prob_feasible = posterior_probs_scores[alternative_id]
        if prob_feasible >= alpha_threshold:
            return eu_mc_scores[alternative_id]
        else:
            return 0.0 # not feasible, very low fitness

    # initialize population: a list of alternative_ids
    population = random.choices(all_alternatives_ids, k=population_size)

    best_overall_alternative_ga = None
    best_overall_fitness_ga = -1.0

    for generation in range(num_generations):
        fitness_scores = [calculate_fitness(alt_id) for alt_id in population]

        # track the best individual in the current generation and overall
        if len(fitness_scores) > 0 and max(fitness_scores) > best_overall_fitness_ga:
            current_best_in_gen_idx = np.argmax(fitness_scores)
            best_overall_fitness_ga = fitness_scores[current_best_in_gen_idx]
            best_overall_alternative_ga = population[current_best_in_gen_idx]

        # selection (Roulette Wheel - individuals with higher fitness are more likely to be chosen)
        total_fitness = sum(fitness_scores)
        if total_fitness == 0:
            # if all fitness is 0 (no feasible alternatives found yet or ever),
            # re-initialize population randomly to try new candidates
            population = random.choices(all_alternatives_ids, k=population_size)
            continue

        selection_probabilities = [score / total_fitness for score in fitness_scores]
        parents = random.choices(population, weights=selection_probabilities, k=population_size) #
select `population_size` parents

        new_population = []
        for i in range(population_size):
            # for this simplified GA (selecting one from existing alternatives), crossover is not
straightforward.
            # we'll just take a parent and potentially mutate it.
            child = parents[i]

            # mutation: randomly replace child with another alternative from the full list
            if random.random() < mutation_rate:
                child = random.choice(all_alternatives_ids)

            new_population.append(child)

        population = new_population
```

```
    return best_overall_alternative_ga, best_overall_fitness_ga

alternatives_ids_for_ga = list(data.index)

if not alternatives_ids_for_ga:
    st.warning("Немає альтернатив для застосування генетичного алгоритму.")
elif not EU_mc or not posterior_probabilities:
    st.warning("Результати симуляції Монте-Карло або Байєсівської оцінки відсутні. Неможливо запустити ГА.")
else:
    ga_best_alternative, ga_best_fitness = run_genetic_algorithm_for_selection(
        EU_mc,
        posterior_probabilities,
        alpha,
        alternatives_ids_for_ga,
        ga_pop_size,
        ga_generations,
        ga_mutation_rate
    )

    if ga_best_alternative:
        st.success(f"Найкраща альтернатива за Генетичним Алгоритмом: ««{ga_best_alternative}»« (з очікуваною корисністю {ga_best_fitness:.3f})")
        st.write("Генетичний алгоритм досліджує простір існуючих альтернатив, еволюціонуючи популяцію кандидатів, щоб знайти альтернативу з найвищою корисністю, що задовольняє критеріям ризику.")
    else:
        st.warning("Генетичний алгоритм не знайшов доцільних альтернатив.")

# -------------------------
# 7. VISUALIZATION (Аналітичний блок - графіки)
# -------------------------
st.subheader("Розподіл корисності (Utility Distribution)")

fig_hist, ax_hist = plt.subplots(figsize=(10, 6))
for alt, vals in mc_results.items():
    ax_hist.hist(vals, bins=50, alpha=0.5, label=f"{alt} (EU: {EU_mc[alt]:.2f})", density=True)
ax_hist.axvline(U_min, color='r', linestyle='--', label=f'U_min = {U_min}')
ax_hist.set_title("Розподіл корисності за симуляцією Монте-Карло")
ax_hist.set_xlabel("Корисність")
ax_hist.set_ylabel("Щільність")
ax_hist.legend()
st.pyplot(fig_hist)


st.subheader("Аналіз чутливості (вплив ваг критеріїв)")

fig_bar, ax_bar = plt.subplots(figsize=(12, 7))
```

```
normalized_weighted_values = norm_data.mul(pd.Series(weights, index=norm_data.columns)) # Align weights
by index
normalized_weighted_values.plot(kind='bar', ax=ax_bar, rot=45)
ax_bar.set_title("Внесок критеріїв в загальну корисність для кожної альтернативи")
ax_bar.set_xlabel("Альтернатива")
ax_bar.set_ylabel("Ваговий нормалізований внесок")
st.pyplot(fig_bar)

st.subheader("Чутливість найкращої альтернативи до ваги якості")

weights_range_sa = np.linspace(0.01, 1.0, 20)
sa_results = []
best_alts_over_range = []

# ensure 'quality' is a valid column before proceeding
if 'quality' in weights:
    original_quality_weight = weights["quality"]
    # re-evaluate all weights for sensitivity analysis
    for w_val in weights_range_sa:
        w_temp = weights.copy()
        w_temp["quality"] = w_val # Modify quality weight

        # re-normalize all weights for current sensitivity step
        current_total_weights = sum(w_temp.values())
        if current_total_weights == 0:
            normalized_w_temp = {k: 0 for k in w_temp}
        else:
            normalized_w_temp = {k: v / current_total_weights for k, v in w_temp.items()}

        U_temp = calculate_utility(norm_data, normalized_w_temp)
        best_alts_over_range.append(U_temp.idxmax())

    # reset quality weight for other calculations to its original value after sensitivity analysis
    weights["quality"] = original_quality_weight

    sa_df = pd.DataFrame({
        "Вага якості": weights_range_sa,
        "Найкраща альтернатива": best_alts_over_range
    })
    st.dataframe(sa_df)

    fig_sens, ax_sens = plt.subplots(figsize=(10, 6))
    ax_sens.plot(weights_range_sa, best_alts_over_range, marker='o', linestyle='-')
    ax_sens.set_title("Чутливість найкращої альтернативи до ваги якості")
    ax_sens.set_xlabel("Вага критерію 'quality'")
    ax_sens.set_ylabel("Найкраща альтернатива")
    ax_sens.grid(True)
    st.pyplot(fig_sens)
else:
```

```
    st.info("Критерій 'quality' відсутній у даних, аналіз чутливості за його вагою неможливий.")


# -------------------------
# 8. RECOMMENDATIONS (Блок рекомендацій)
# -------------------------
st.subheader("Блок рекомендацій")

if best_alternative:
    st.write(f"На основі поточних параметрів, ««{best_alternative}»« є рекомендованою альтернативою.")
    st.write(f"Його очікувана корисність становить ««{EU_mc[best_alternative]:.3f}»«.")

    st.markdown("#### Пояснення вибору:")
    st.write(f"Альтернатива ««{best_alternative}»« була обрана, оскільки вона має найвищу очікувану корисність ({EU_mc[best_alternative]:.3f}) серед усіх альтернатив, які відповідають заданому мінімальному порогу корисності (`U_min = {U_min}`) з постеріорною ймовірністю ««{posterior_probabilities[best_alternative]:.2%}»« (що перевищує допустимий ризик `α = {alpha:.0%}`).")

    st.markdown("#### Ключові фактори, що впливають на вибір:")
    alt_data = norm_data.loc[best_alternative]
    weighted_contributions = (alt_data « pd.Series(weights, index=alt_data.index)).sort_values(ascending=False) # Align weights by index
    st.write("Ось внесок кожного критерію в загальну корисність для обраної альтернативи:")
    st.dataframe(weighted_contributions.to_frame(name='Weighted Contribution'))
    st.write(f"Найбільший позитивний вплив мають критерії: ««{', '.join(weighted_contributions.head(2).index.tolist())}»«.")

    st.markdown("#### Можливі ризики та компроміси:")
    if posterior_probabilities[best_alternative] < 1.0:
        st.warning(f"Існує ««{(1 - posterior_probabilities[best_alternative]):.2%}»« постеріорна ймовірність, що корисність ««{best_alternative}»« буде нижчою за `U_min = {U_min}`. Цей ризик слід враховувати.")
    else:
        st.info("Згідно з Байєсівською оцінкою, ймовірність досягнення мінімальної корисності `U_min` для цієї альтернативи становить 100%.")

    original_alt_data = data.loc[best_alternative]
    # check if 'cost' and 'quality' exist in the original_alt_data index
    if "cost" in original_alt_data.index and "quality" in original_alt_data.index:
        st.info(f"Зверніть увагу, що для ««{best_alternative}»« показник `quality` становить {original_alt_data['quality']}, а `cost` - {original_alt_data['cost']}. Це може означати компроміс між високою якістю та відповідною вартістю.")
else:
    st.write("Наразі немає рекомендацій, оскільки не знайдено доцільних альтернатив за Байєсівською оцінкою.")

# примусово завершуємо всі процеси Streamlit, щоб уникнути конфліктів
!pkill -f streamlit
```

```
# зупиняємо будь-які попередні тунелі ngrok, якщо вони запущені
ngrok.kill()

# запускаємо Streamlit та створюємо ngrok тунель
public_url = ngrok.connect(8501)
print(f"Ваш Streamlit-додаток доступний за публічним посиланням: {public_url}")

# запускаємо Streamlit у фоновому режимі, щоб не блокувати виконання комірки
!nohup streamlit run app_improved.py > output_improved.log 2>&1 &
```

Короткий звіт подібності

StrikePlagiarism.com

Звіт не був оцінений

Метадані

ДОКУМЕНТ

Заголовок
КМР Карєва А.А.

Автор
Карєва Анжеліка

Науковий керівник / Експерт
Джалладова Ірада Агаєвна

ІД документу
333898612

ОРГАНІЗАЦІЯ

Назва організації
Kyiv National Economic University named after Vadym Hetman KNEU

підрозділ
кафедра системного аналізу та кібербезпеки

ЗВІТ

Дата звіту
5/17/2026

Дата редагування

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

0.4% 0.4% КП 1

0% 0% КП 2

0.76% 0.76% КЦ

25
Довжина фрази для коефіцієнта подібності 2

10878
Кількість слів

88525
Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	ß	6
Інтервали	A→	0
Мікропробіли	¶	8
Білі знаки	ß	0
Парафрази (SmartMarks)	a	3

Джерела

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз — Колір тексту

#	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://jase-web.org/documents/SER.I/SER.I19(1)_Francois.pdf21583007154	13 (0.12 %)

2	Analiza determinantów wydatków na opiekę medyczną w Filipinach za pomocą technik data miningowych [redacted] 11/26/2019 Szkoła Główna Handlowa (Szkoła Główna Handlowa)	11 (0.1 %)
3	Горбань [redacted] 12/10/2025 West Ukrainian National University (West Ukrainian National University)	10 (0.09 %)
4	Analiza determinantów wydatków na opiekę medyczną w Filipinach za pomocą technik data miningowych [redacted] 11/26/2019 Szkoła Główna Handlowa (Szkoła Główна Handlowa)	9 (0.08 %)

База даних RefBooks

#	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)

Домашня база даних

#	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)

Програма обміну базами даних (0.28 %)

#	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Analiza determinantów wydatków na opiekę medyczną w Filipinach za pomocą technik data miningowych [redacted] 11/26/2019 Szkoła Główna Handlowa (Szkoła Główna Handlowa)	20 (2) (0.18 %)
2	Горбань [redacted] 12/10/2025 West Ukrainian National University (West Ukrainian National University)	10 (1) (0.09 %)

Інтернет (0.12 %)

#	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
3	https://iase-web.org/documents/SERJ/SERJ19(1)_Francois.pdf?1583007154	13 (1) (0.12 %)

Список прийнятих фрагментів

#	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)

Копія тез доповіді

Джалладова І.А., Карєва А.А. «Синергія Data-Driven Management і теорії прийняття рішень у сучасному бізнесі». Інтелектуальні методи аналізу загроз кібербезпеки: Зб. матеріалів XV міжнародної науковопрактичної конференції, 10-14 листопада 2025 р. [Електронний ресурс]. – К. : КНЕУ, 2025. – с. 108-109

Джалладова І. А.
д.ф.-м.н., професор,
завідувач кафедри системного аналізу та кібербезпеки,
КНЕУ ім. В. Гетьмана, м. Київ, Україна
ORCID ID: https://orcid.org/0000-0003-3158-6844

Карєва А. А.
здобувачка магістерського рівня,
КНЕУ ім. В. Гетьмана, м. Київ, Україна

СИНЕРГІЯ DATA-DRIVEN MANAGEMENT І ТЕОРІЇ ПРИЙНЯТТЯ РІШЕНЬ У СУЧАСНОМУ БІЗНЕСІ

Вступ. У бізнес-середовищі, що характеризується високою динамічністю та невизначеністю, здатність компаній приймати ефективні рішення, ґрунтуючись на достовірних даних, набуває визначального значення. Концепція Data-Driven Management (DDM) полягає в управлінні, що базується на аналітиці, фактичних даних та об'єктивних показниках. Паралельно, теорія прийняття рішень (ТПР) пропонує формалізовані принципи й моделі, необхідні для обґрунтування вибору оптимальної дії серед наявних альтернатив. Синергетична взаємодія цих двох підходів суттєво підвищує їхню ефективність: кількісні методи ТПР отримують емпіричне підґрунтя (дані від DDM), а DDM, своєю чергою, здобуває необхідну методологічну основу.

Ціллю роботи є дослідження взаємодії концепцій Data-Driven Management і теорії прийняття рішень, визначення переваг їх поєднання для сучасного бізнесу та формування узагальненої моделі прийняття рішень на основі даних.

Виклад основного матеріалу дослідження. Теорія прийняття рішень розглядає процес вибору як послідовність етапів: ідентифікація проблеми, визначення альтернатив, оцінка наслідків і вибір оптимальної дії. У підході Data-Driven Management ці етапи автоматизуються за допомогою інструментів збору, оброблення та аналізу даних — систем BI, Big Data-платформ і моделей машинного навчання [1]. Синергія двох підходів проявляється в тому, що дані стають основою для оцінювання альтернатив, а математичні моделі ТПР забезпечують об'єктивний вибір серед них. Узагальнено процес прийняття рішень можна подати як оптимізаційну задачу (формула 1.1, [2]):

$$a^* = \arg\max_{a_i \in A} U(a_i), \qquad (1.1)$$

де: A = {a_1, a_2, …, a_n} — множина альтернатив, тобто можливих дій або рішень, які розглядає керівник/система; $U(a_i)$ — функція корисності (або ефективності) дії a_i. Вона відображає, наскільки ця дія є бажаною — на основі даних, аналітики, управлінських критеріїв. Оператор arg max означає: знайди туди дію a^* з множини A, для якої $U(a_i)$ є максимальною.

Дані (з BI-систем, систем аналітики, Big Data) дають значення показників, які потім формують функцію $U(a_i)$. Наприклад, для дії «збільшити обсяг логістики через конкретний маршрут» може бути показник скорочення витрат, підвищення рівня обслуговування клієнтів, зниження ризику затримок — все це